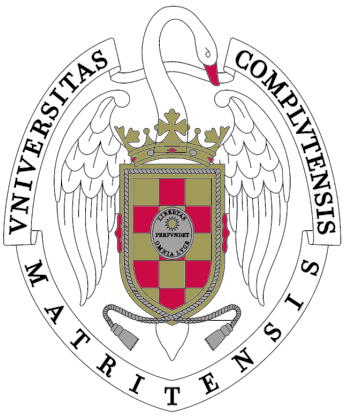
Extracción automática de tópicos en biología a partir de la literatura científica

Departamento de Arquitectura de Computadores y Automática



Universidad Complutense de Madrid

Rubén Nogales Cadenas

Director: Alberto Pascual Montano

Trabajo de investigación de doctorado Madrid, septiembre de 2007

Resumen

Los recientes avances en Biología Molecular y en Informática son responsables de la acumulación de muchos y cada vez más complejos tipos de datos. Este incremento se ha visto también reflejado en el elevado número de publicaciones relacionadas. Todo esto se debe a los experimentos a gran escala que ahora se pueden llevar a cabo en este tipo de investigación. Genomas completos pueden ser secuenciados en meses o semanas, métodos computacionales permiten la identificación de miles de genes en el DNA secuenciado y se han desarrollado herramientas que analizan automáticamente las propiedaddes de los genes y las proteínas.

No obstante, no sólo los resultados de los distintos experimentos sirven para encontrar información biológica, actualmente es posible explorar la literatura biomédica en busca de evidencias biológicas. Sin embargo, ese proceso de extracción de información a partir de las publicaciones es, en su gran mayoría, manual. Un grupo de anotadores se encarga de leer todos los artículos científicos, extraer evidencias biológicas y almacenarlas en las bases de datos y ontologías biológicas públicas accesibles a través de internet.

Debido a la gran acumulación de documentos científicos, se necesita desarrollar métodos y herramientas que automaticen el proceso de extracción de información.

En este contexto se propone un método de extracción de información biológica a partir de la literatura biomédica basado en la extracción de anotaciones enriquecidas en términos encontrados en publicaciones y bases de datos. Un posterior análisis estadístico, utilizando varios test como el de χ^2 o el de la distribución hipergeométrica y corrigiendo el problema de la hipótesis múltiple, nos permitirá evaluar el nivel de relevancia de las anotaciones recuperadas. Esta metodología permite integrar datos obtenidos de la literatura con otras fuentes de información como anotaciones funcionales o reguladores transcripcionales y es de gran utilidad para el descubrimiento de asociaciones entre información biológica de los genes y proteínas y documentos o conjuntos de palabras.

Lista de palabras clave

Minería de datos, Minería de Textos, Extracción de la Información, Bioinformática, Reglas asociativas, Bases de datos, Análisis estadístico.

Agradecimientos

Quisiera agradecer en primer lugar y muy especialmente a Alberto Pascual-Montano, Mónica Chagoyen y Pedro Carmona por su inestimable ayuda y sus valiosos consejos y sugerencias, imprescindibles a la hora de realizar este trabajo.

También quiero dar las gracias a Carolina Bónacic, César Vicente, Enrique de la Torre, Xiaoyuan Yang, Edgardo Mejía, Miguel Vázquez y Mariana Lara, ya que en los muchos momentos en los que he precisado su ayuda no han dudado un momento en facilitarme todo lo que necesitaba. Muchas gracias chicos.

Por último agradecer las financiaciones de los proyectos BIO2007-67150-C03-02, GR/SAL/p653/2004, PR27/05-13964-BSCH, CYCIT-TIN-2005-5619 y CAM-P2006/Gen-0166 dentro de las que ha sido realizado este trabajo. Alberto Pascual-Montano quiere agradecer también al programa "Ramón y Cajal".

Indice

Indice de figuras

Capítulo 1

Introducción

La última década se ha caracterizado por un crecimiento sin precedentes en la obtención de datos biomédicos que pueden ir desde secuencias biológicas derivadas de experimentos genéticos hasta datos estructurales de distintas biomoléculas. Este incremento se ha visto también reflejado en el elevado número de publicaciones relacionadas. Todo se debe al tipo de experimentos a gran escala que ahora pueden llevarse a cabo en este tipo de investigación, gracias a los avances en los campos de la informática y de la biología. Genomas completos pueden ser secuenciados en meses o semanas, métodos computacionales permiten la identificación de miles de genes en el DNA secuenciado y se han desarrollado herramientas que analizan automáticamente las propiedaddes de los genes y las proteínas. Técnicas como los microarrays de DNA permiten medir simultáneamente el nivel de expresión de todos los genes o proteínas de un sistema biológico. Estos experimentos a gran escala producen enormes cantidades de datos que, cuando son procesados, ofrecen los patrones de expresión de los genes estudiados ante determinadas condiciones experimentales (distintos tejidos, enfermedades, fases celulares, etc...). El último objetivo de esta cadena es traducir esa gran cantidad de información al conocimiento de los complejos procesos biológicos que ocurren dentro del ser humano, y utilizar ese conocimiento en favor del avance de la medicina.

Casi todos los conocimientos que se adquieren en los distintos trabajos de genómica o proteómica son publicados en la, ya de por sí, vasta colección de literatura biomédica debido a que es muy utilizada por la comunidad científica para diseminar los resultados. El avance de las técnicas de secuenciación del genoma, aunque muy importante, ha propiciado una desbordante acumulación de información, además del descubrimiento de nuevos genes y funciones o propiedades biológicas. Esta abundancia de genes, productos genéticos y literatura, en definitiva de información, es responsable de que al

interpretar los resultados de los experimentos genómicos, o incluso al planear dichos experimentos, se produzca un importante cuello de botella. Se necesita poder procesar toda esta información de manera efectiva y rápida para poder diseñar e interpretar los experimentos a gran escala que nos permiten llevar a cabo las técnicas más actuales. Además de todo esto, es interesante el desarrollo de métodos y herramientas que posibiliten la integración de fragmentos de información de diferentes campos de estudios, pudiendo de esta manera ofrecer un panorama general en el que se dibujen los roles de varios genes, proteínas y reacciones químicas en células y organismos.

Durante los últimos años se ha incrementado el interés del uso de la literatura biomédica. Dado que la literatura cubre todos los aspectos de la biología, química y medicina, no hay casi límites en los tipos de información que pueden ser recuperados a través de una minería exaustiva y cuidada. Entre las posibles aplicaciones tenemos por ejemplo la reconstrucción y predicción de vías metabólicas, establecer conexiones entre genes y enfermedades, encontrar relaciones entre genes y funciones biológicas específicas y muchas más. En este sentido, uno de los campos de estudio más importantes es la caracterización de las funciones de cada gen y proteína. Por otra parte, es indudable que una única estrategia de minería no es suficiente para poder abarcar el amplio espectro de objetivos y necesidades que surgen a este respecto.

Para poder abarcar el creciente número de tipos de datos, para poder procesar toda esa información y para poder almacenarla se han desarrollado muchos métodos informáticos en las areas de la bioinformática y la biología computacional. El procesado automático de textos es una area de investigación formada por diversas disciplinas. Incluyen Recuperación de la Información (IR), que se ocupa de encontrar documentos que satisfagan una determinada información o consulta dentro de una gran base de datos de documentos, como pudiera ser, por ejemplo, Internet; Procesamiento Natural del Lenguaje (NLP), que es una disciplina que abarca todos las técnicas de procesamiento automático tanto de lenguaje escrito como hablado; la Extracción de la Información (IE), que puede ser considerada un campo de NLP y está centrada en encontrar entidades explícitas y hechos dentro de un texto no estructurado. Por ejemplo, encontrar dentro de un texto todas las veces que aparece una determinada proteína. Finalmente, la Minería de Texto es el proceso de analizar el lenguaje natural escrito para descubrir información o conocimientos que son comunmente difíciles de recuperar.

El creciente interés en Recuperación de la Información (IR), Extracción de la Información (IE) y la Minería de Texto centrándose en la literatura biomédica está relacionado por una parte con el incremento y acumulación de literatura científica (PubMed contiene actualmente más de 12.000.000 entradas) y por otra con ese acelerado proceso de descubrimiento de información

biológica. El tipo de técnicas informáticas que procesan literatura biomédica son muy útiles para facilitar el acceso a textos relevantes a biólogos, bioinformáticos e incluso a anotadores de bases de datos. Sin embargo, el proceso actual de extracción de la información es en su gran mayoría manual. Se extrae dicha información de las publicaciones científicas pertinentes y se almacena en las grandes bases de datos y ontologías biológicas que hay repartidas por la red. Dichas bases de información son de gran importancia ya que los resultados de las distintas técnicas experimentales y bioinformáticas han de ser, en muchos casos, interpretados recurriendo a la información que contienen.

Una de las herramientas más importantes para representar y procesar información acerca de los genes y sus funciones es Gene Ontology (GO) [2]. Se trata de una de las ontologías públicas del campo de la biología más importantes y provee un vocabulario controlado de más de 22.000 términos que se utiliza para describir componentes celulares, funciones moleculares y procesos biológicos en cualquier organismo. La ontología de componentes celulares está compuesta por localizaciones o estructuras físicas (*flagellum, chromosome*), la ontología de funciones moleculares comprende actividades o tareas elementales (*mitosis, purine metabolism*) y la ontología de procesos biológicos contiene términos que representan objetivos o metas biológicas (*glycolysis, death*).

Asociada a GO se encuentra GOA (Gene Ontology Annotations), una base de datos que relaciona el genoma de determinados organismos con términos de GO. Además de establecer dicha relación (*genes - GOterms*), proporciona la publicación que la evidencia. El método de extracción de información e incorporación en la base de datos es totalmente manual, existe un cuerpo de anotadores encargados de leer todas las publicaciones biomédicas, concluir las relaciones entre genes y términos de GO e introducir la información en la base de datos de GOA. Debido a la creciente acumulación de información biomédica, se hace necesario el desarrollo de una aplicación que automatice el proceso.

Uno de los temas que más interesa a la comunidad científica es precisamente la de clasificar documentos de acuerdo a los procesos biológicos y las funciones moleculares que describen. Sabiendo de qué procesos biológicos o funciones moleculares habla un documento, podremos establecer ascociaciones entre las entidades (genes o proteínas) que aparecen en el documento y los tópicos encontrados. Dicho de otra manera, es posible caracterizar entidades biológicas a través de sus funciones y del papel que desempeñan en distintos procesos biológicos a través del estudio de la literatura biomédica y, más en concreto, de la clasificación de documentos.

Precisamente, dentro de este contexto se encuentra el trabajo presentado.

Se propone un sistema que utiliza una fuente de información específica de contenido biológico, de amplio espectro y plenamente aceptada dentro de la comunidad científica: Gene Ontology. El sistema trata de encontrar todos los términos de GO relacionados con el texto que sirve como entrada. De esta manera se etiqueta el texto y se resume su contenido basado en un vocabulario controlado. El número de apariciones de cada categoría de GO es entonces determinado tanto para el conjunto de palabras del texto de entrada como para el corpus que se utilice de referencia (por defecto, el conjunto total de abstracts de PUBMED junto con la información propia de Gene Ontology), y un test estadístico - usualmente la hypergeométrica, χ^2, binomial o el test de Fisher - es usado para calcular los *p valores*.

El trabajo está enmarcado dentro de un proceso de tres etapas en el que la primera etapa clasifica los textos de PUBMED y escoge sólo aquellos que tengan relación con procesos biológicos. La segunda etapa extrae de ése conjunto los documentos que referencien las proteínas o genes de una lista que se toma como entrada. La tercera fase, la que nos ocupa, se encargaría de recuperar los términos de Gene Ontology relacionados con ese subconjunto de documentos. De esta manera podríamos establecer relaciones entre las proteínas o genes de la lista de entrada y términos de Gene Ontology, pudiendo aportar además los textos que evidencien dicha relación.

Existen otras aproximaciones que intentan resolver este problema o problemas parecidos. En [3] se aprenden modelos de n-gramas para cada término de GO mediante un modelo estadístico y se utiliza esta información para encontrar anotaciones en los documentos, en [4] utilizan redes de términos y nodos de GO y en [5] agrupan diversas palabras en conjuntos y juegan con las probabilidades de pertenencia o no al conjunto para sacar conclusiones.

En la sección siguiente se presenta la Minería de Datos como método de extracción de la información, así como algunos métodos y trabajos relacionados. En el capítulo 3 se explican los distintos métodos de Minería de Texto y procesamiento textual, incluyendo una exposición de áreas como Recuperación de la Información, Extracción de la información y Procesamiento de Lenguaje Natural. Y en el capítulo 4 se desarrolla el trabajo de la minería de literatura, pero en el contexto de la Bioinformática. En el capítulo 5 se hablará acerca de la ontología Gene Ontology, utilizada en este trabajo como fuente de información, de la estructura de datos y de su utilidad.

A partir de aquí, nos centraremos en la metodología propuesta. En el capítulo 6 se especifican los objetivos perseguidos en este trabajo, en el capítulo 7 se exponen los métodos estudiados y utilizados y en 8 se explica la implementación llevada a cabo de la herramienta desarrollada. En último lugar se explicarán los resultados obtenidos en 9 y las conclusiones finales del trabajo serán expuestas en 10.

Capítulo 2

Minería de Datos

Tradicionalmente, Minería de Datos se define como "un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos" [6]. Realmente se trata de una etapa dentro de un proceso mayor llamado extracción de conocimiento en bases de datos (Knowledge Discovery in Databases o KDD). Lo que en verdad hace el Minería de Datos es reunir las ventajas de varias áreas como la Estadística, la Inteligencia Artificial, la Computación Gráfica, las Bases de Datos y el Procesamiento Masivo de Datos, principalmente usando como materia prima fuentes de información como las bases de datos o las ontologías.

En una era en la que se ha producido un crecimiento explosivo de la información biológica generada por la comunidad científica (Véanse las figuras 2.1, 2.2 y 2.3, donde se aprecia el increíble aumento de entradas en tres de las bases de datos biológicas más importantes actualmente, las del NCBI, SwissPort y PDB), debido al desarrollo de técnicas expermientales muy poderosas capaces de producir en un solo experimento la información equivalente a cientos de miles de experimentos tradicionales las técnicas de minería de datos se han convertido en una herramienta muy importante.

Las principales fuentes de datos utilizadas son ficheros planos, bases de datos relacionales, base de datos de transacciones, bases de datos objeto-relacionales, bases de datos espaciales, series de tiempo, textos, literatura e incluso multimedia (video, audio) o datos en Internet. De ellos se pretende extraer información que abarca desde caracterización de entidades, discriminación, clasificación, agrupamiento, descubrir tendencias, calcular la desviación, detección de datos anómalos, etc.

Las técnicas de minería de datos son muy utilizadas en distintas áreas y tienen diversas aplicaciones. Evidentemente son muy útiles en investigación científica, pero también en telecomunicaciones o en la banca. También es usada por determinados organismos para detección de fraudes y es muy

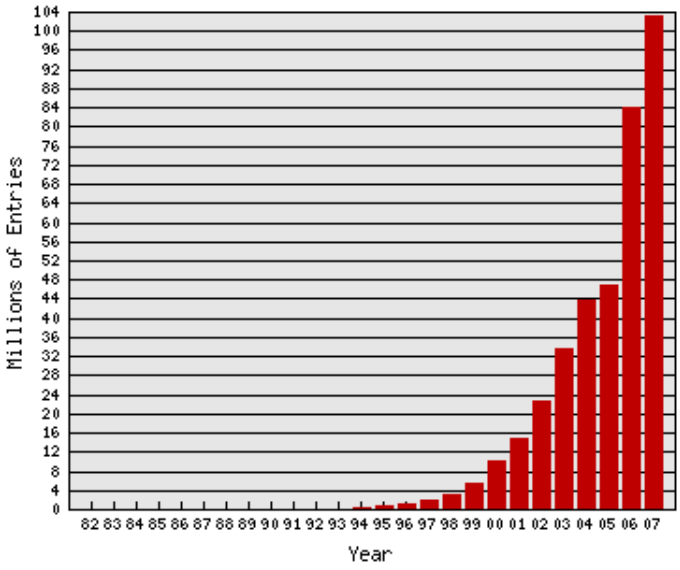


Figura 2.1: Crecimiento de la base de datos de nucleotidos de EMBL

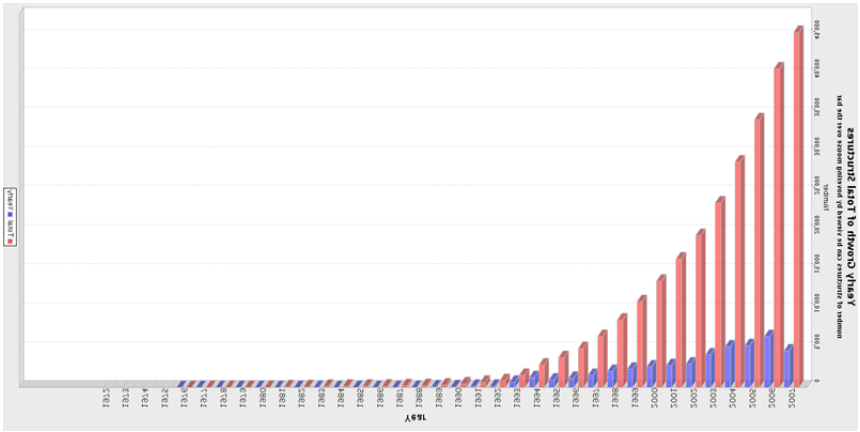


Figura 2.2: Crecimiento de la base de datos de estructuras (coordenadas atómicas) de PDB

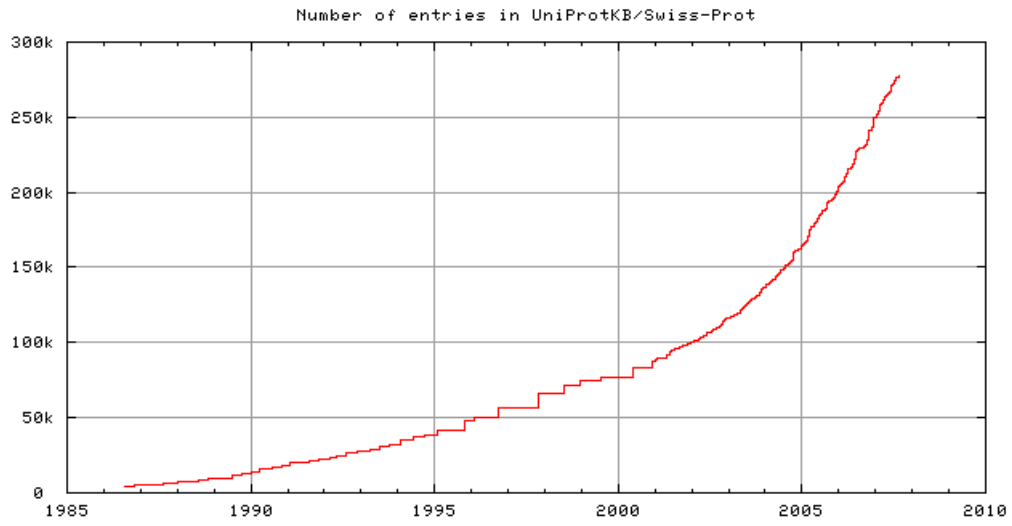


Figura 2.3: Crecimiento de la base de datos de secuencias de proteínas de SwissProt

apreciada en el mundo de los negocios, ya que permite hacer análisis de mercado o análisis de bolsa. Además ocupan una posición especial dentro del área de la Bioinformática, donde, por ejemplo, permite la extracción de conocimiento biológico a partir de anotaciones de genes o proteínas. Un análisis muy común es la identificación de anotaciones biológicas que aparecen frecuentemente relacionadas con un listado de genes (por ejemplo un cluster) con respecto a una lista de referencia (un microchip de ADN o un genoma completo). Formalmente, si queremos saber la probabilidad de que i genes de una lista de n tengan una anotación común (por ejemplo, un término GO [Véase capítulo 5]), dada una lista de referencia con M genes donde N tienen comparten dicha anotación, tendríamos que calular el resultado de la ecuación (2.1).

$$P = \frac{\binom{M}{i}\binom{N-M}{N-i}}{\binom{N}{i}} \tag{2.1}$$

Aunque esta aproximación tiene el problema de que analiza cada anotación de manera independiente. En [7] se desarrolla una herramienta que descubre co-ocurrencias de anotaciones en genes, permitiendo así analizar múltiples anotaciones en un solo paso, integrar distintos tipos de anotaciones en el mismo análisis (GO, rutas metabálicas, etc.) y, en definitiva, reportar información mucho más completa para entender los mecanismos celulares

que aparecen en un experimento determinado. En cuanto a su metodología, sin entrar mucho en detalle, en primer lugar se encuentran combinaciones de términos que aparecen en al menos x genes mediante extracción de reglas. De esta manera, ahora tenemos x genes de un grupo de n que comparten una determinada combinación de anotaciones y M genes de un grupo de N que comparten una determinada combinación de anotaciones, aplicando la ecuación (2.1) podemos realizar el mismo tipo de análisis.

Dentro de la minería de datos, uno de los campos más importantes es la minería de texto. Se trata de un tipo especial de Minería de Datos en el que la información es extraída a partir de textos y de la literatura. Debido a su extensión y a su relación con el trabajo, será explicado en un capítulo a parte mostrando las distintas alternativas, métodos y técnicas más usados y repasando en el capítulo 4 el uso de la Minería de Texto aplicada a la Bioinformática.

Aunque existen muchos y muy diversos métodos en la Minería de Datos, dado que se trata de un campo de investigación multidiscipinar, existen algunas técnicas clásicas o que son muy utilizadas dentro de la Bioinformática, sobre todo en el contexto del análisis de micorarrays de ADN. Se trata de algoritmos de agrupamiento, clasificación y métodos de extracción de características como pueden ser SVD o PCA.

2.1 Algoritmos de agrupamiento

Una de las metodologías que se usa con más frecuencia en la minería de datos son los algoritmos de agrupamiento o clustering. Este tipo de algoritmos divide un conjunto de elementos en grupos que satisfacen las condiciones de homogeneidad (alta similitud entre los elementos de un mismo grupo) y separación (baja similitud entre elementos de grupos distintos). Por ejemplo son muy usados en el contexto del análisis de datos de expresión génica, donde el principal objetivo al usar este tipo de algoritmos es encontrar conjuntos de genes, o condiciones experimentales, que muestran perfiles de expresión parecidos. Este tipo de análisis tiene un claro significado biológico ya que genes que muestran un patrón de expresión similar es probable que estén implicados en los mismos procesos biológicos o regulados por los mismos mecanismos y, del mismo modo, condiciones experimentales con perfiles de expresión similares es probable que estén relacionadas con un mismo estado fisiológico, por ejemplo muestras procedentes del mismo tipo de tumor. Esta familia de técnicas permite subdividir el problema en diferentes grupos y abordar el análisis individual de cada uno de ellos, dividiendo así la dimensionalidad del problema. Entre los algoritmos de agrupamiento más utilizados

están el algoritmo de agrupamiento jerárquico, el de las k-medias y los mapas auto-organizativos.

2.1.1 Agrupamiento jerárquico

El agrupamiento jerárquico ordena los elementos de una población en base a un árbol de distancias que refleja la similitud que hay entre los elementos y grupos. Los algoritmos aglomerativos se inician asignando cada elemento individual a un grupo, se calculan las distancias de todos contra todos y los dos elementos más similares se unen para formar un nuevo grupo. Finalizado este proceso, se vuelve a recalcular la matriz de distancias considerando el nuevo grupo y se vuelven a unir los dos elementos más similares. Este proceso se repite hasta que se unen los dos últimos grupos. Por el contrario, los algoritmos divisivos comienzan con un solo grupo que engloba al conjunto total de elementos, y en cada paso se subdivide en grupos de menor tamaño hasta llegar a los elementos únicos. Este tipo de algoritmos fue introducido al análisis de datos de expresión génica por Eisen et. al [54] y se han convertido en uno de los métodos más populares en este contexto. Presentan las ventajas de que es una metodología simple y los resultados pueden ser fácilmente visualizados. Sin embargo, también pueden presentar ciertos problemas como es el que, al ir creciendo en tamaño, los vectores representativos de un grupo puede que no se asemejen a los elementos englobados en el mismo. Además, con este tipo de técnicas si se comete un error de asignación en estadios iniciales del proceso este se arrastrará hasta el final .

2.1.2 K-medias

El algoritmo de k-medias es un algoritmo de agrupamiento clásico que divide un conjunto de elementos en un número predefinido de grupos. Este método requiere por tanto especificar el número de grupos (k) a priori. Dado un valor de k, el algoritmo de k-medias divide el conjunto de datos en k grupos minimizando la siguiente función:

$$E = \sum_{i=1}^{k} \sum_{O \in C_i} |O - \mu_i|^2 \tag{2.2}$$

donde O es un elemento en el grupo C_i y μ_i es el centroide (media de los elementos de un grupo) del grupo C_i. De forma resumida, este algoritmo trabaja de la siguiente manera: Los datos son asignados de forma aleatoria a k grupos. A continuación los centroides de cada grupo son calculados y cada dato es asignado a su centroide más cercano formando k nuevos grupos. Este

proceso es repetido hasta que se alcanza algún criterio de parada, usualmente cuando las variaciones de los centroides entre distintas iteraciones sean muy pequeñas o cuando se alcanza un número prefijado de las mismas.

El algoritmo de k-medias es rápido y sencillo, pero presenta también ciertas limitaciones para el análisis de datos de expresión como, por ejemplo, que normalmente el número de grupos no se conoce a priori. Además, este algoritmo no garantiza que se alcance un mínimo global en la función de optimización, por lo que los resultados obtenidos en muchas ocasiones pueden no ser óptimos.

2.1.3 Mapas auto-organizativos

Los mapas auto-organizativos constituyen un método de agrupamiento basado en redes neuronales desarrollado por Teuvo Kohonen. Un SOM asigna los elementos a una serie de vectores, o neuronas, dentro de una red que presenta una topología predefinida. El algoritmo de SOM fue introducido para análisis de datos de expresión por Tamayo et al. [55] y Toronen et al. [56] y tiene algunas propiedades que lo hacen interesante para este tipo de análisis: facilita la visualización e interpretación de datos multidimencionales en espacios usualmente bidimensionales, organiza los grupos de forma que los más cercanos en la red son los más parecidos y es relativamente más robusto al ruido en los datos que otros algoritmos como el de k-medias. Las desventajas de este método es que requiere determinar a priori el tamaño y la estructura del mapa, aunque este parámetro no es tan crítico como establecer el número de grupos en el algoritmo de k-medias. Además si los datos contienen una gran cantidad de elementos irrelevantes, como por ejemplo genes con poca variación en sus perfiles de expresión, este método generará unos resultados en los cuales este tipo de datos serán asignados a la gran mayoría de las neuronas y los patrones más interesantes pueden ser asignados y mezclados en unos pocos grupos .

2.2 Algoritmos de Clasificación

La diferencia entre los algoritmos de clasificación y los de agrupamiento radica en que los primeros conocen a priori el número de grupos que se van a formar y utilizan esta información mientras que los segundos no. Debido a esto se considera a los algoritmos de clasificación como algoritmos de aprendizaje supervisado (es decir, que cuentan con información previa que les ayuda a resolver el problema).

Formalmente se pueden definir como una función en la que dada un conjunto de instancias del problema a resolver, devuelve la categoría a la que pertenecen (de un conjunto de categorías predefinidas). A pesar de necesitar conocer previamente el número de categorías son algoritmos muy utilizados y de gran utilidad. Incluso pueden combinarse con otros algoritmos que deducen el numero de clases existentes dentro de un conjunto de instancias del problema (selección del rango de factorización).

2.2.1 Árboles de decisión

Los sistemas de aprendizaje basados en árboles de decisión son quizás el método más fácil de utilizar y de entender. Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos, y son especialmente apropiados para expresar procedimientos médicos, matemáticos, lógicos, etc.

Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y, siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar. Estos algoritmos se llaman algoritmos de partición o algoritmos de "divide y vencerás", donde la elección del criterio de partición puede llevar a un buen o mal resultado.

Existen muchos tipos de árboles de decisión, en función del algoritmo que utilizan para ser generados o por ejemplo el tipo de datos con los que se trabaje. En el campo de la Bioinformática, últimamente están siendo muy utilizada la técnica conocida como *Random Forests*, que implica el uso de diversos árboles de decisión para llevar a cabo la clasificación. Por ejemplo en [57], donde se utiliza para clasificar genes en función de sus patrones de expresión en microarrays.

2.2.2 Máquinas de soporte vectorial

Las máquinas de vectores de soporte (SVM, por sus siglas en inglés) han mostrado conseguir buen desempeño de generalización sobre una amplia variedad de problemas de clasificación, destacando en problemas de clasificación de textos, donde se aprecia que SVM tiende a minimizar el error de generalización, i.e. los errores del clasificador sobre nuevas instancias.

En términos geométricos, SVM puede ser visto como el intento de encontrar una superficie (σ_1) que separe a los ejemplos de un tipo u otro por el

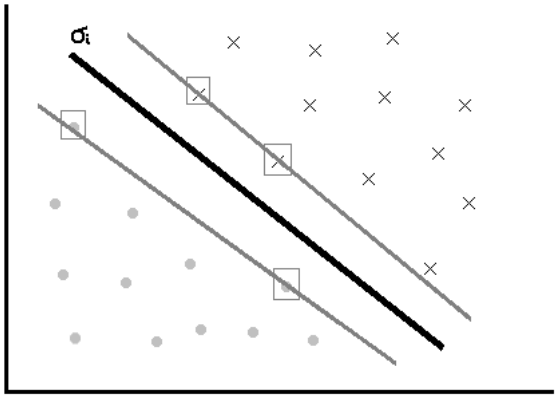


Figura 2.4: Problema de clasificación mediante SVD

margen más amplio posible.

La búsqueda de σ_1 que cumple que la distancia mínima entre él y un ejemplo de entrenamiento sea máxima, se realiza a través de todas las superficies $\sigma_1, \sigma_2, ...$ en el espacio n-dimensional que separan a los ejemplos de diversos tipos en el conjunto de entrenamiento (conocidas como superficies de decisión). Gráficamente, el método de SVM se explica en la figura 2.4 donde se ve un espacio de 2 dimensiones con dos tipos de casos, positivos y negativos representados por puntos y cruces. Se trata de encontrar la superficie σ_i capaz de separar unos de otros de la mejor manera posible, en este caso esa superficie se trata de la línea marcada en negrita.

En el campo de la Bioinformática se han publicado diversos trabajos de minería de datos uilizando este método, como por ejemplo [58] y [59], donde se utiliza SVM en la minería de datos biomédicos y para clasificar genes en función de sus nombres respectivamente.

2.3 Extracción de características

Como se ha mencionado anteriormente, en un experimento típico con microchips de ADN se cuantifica la expresión de miles de genes, o incluso genomas completos, a lo largo de varias condiciones experimentales. Algunas de estas condiciones experimentales pueden mostrar una alta correlación y sólo una pequeña parte de todos los genes incluidos en el chip serán importantes para explicar la mayor parte de variabilidad entre los distintos experimentos. Esto hace que las matrices de expresión génica contengan información redundante y ruidosa.

Métodos tales como el análisis de componentes principales (PCA), la descomposición en valores singulares (SVD) o el análisis de componentes independientes (ICA) son muy útiles para reducir la dimensionalidad de los datos reteniendo los principales patrones de los mismos. Dada una matriz inicial de m variables (genes) y n observaciones (experimentos), estos métodos permiten encontrar k nuevas variables donde $k < m$ mediante una descomposición de la matriz inicial como un producto de matrices de menor rango:

$$A_{m\times n} \sim (WH)_{m\times n} = \sum_{i=1}^{k} W_{m\times i} H_{i\times n} \tag{2.3}$$

Las k columnas de W son denominadas componentes o factores. Las columnas de H están en correspondencia uno a uno con los n experimentos en la matriz A y contienen los coeficientes por los cuales cada experimento de la matriz original es representado como una combinación lineal de los k-factores. En PCA y SVD estos componentes capturan la mayor varianza de los datos y son ortogonales entre si mientras que en ICA los componentes son estadísticamente independientes entre si.

Esta descomposición se puede usar para descubrir patrones en los datos, eliminar ruido y transformar los datos para una mejor visualización y análisis . También se suele utilizar como un paso previo antes de aplicar otros métodos de análisis, como los algoritmos de agrupamiento.

Capítulo 3

Métodos de procesamiento de texto y Text Mining

Esta sección introduce las disciplinas involcuradas en el procesamiento de texto así como con las técnicas y métodos que usan. Empezamos con técnicas generales de Procesamiento Natural de Lenguaje (NLP) y text mining. Después procederemos con áres más específicas de Information Retrieval e Information Extraction. La primera está centrada alrededor de tareas de alto nivel de identificar documentos relevantes que satisfagan una determinada información o consulta, no involucrándose demasiado en tareas de representación o comprensión de lenguaje natural. Por otra parte, Information Extraction se ocupa de la extracción de entidades específicas, hechos y eventos de dentro del texto, y está más relacionada con técnias NLP. Concluimos la sección con una revisión corta de los métodos estándares de evaluación empleados en estos campos.

3.1 Procesamiento de Lenguaje Natural: Técnicas Generales

Las técnicas de NLP cubren todos los aspectos y etapas necesarias para convertir el lenguaje escrito o hablado en información que pueda ser usada por otros humanos o agentes automatizados. En el contexto de la bioinformática se suele hacer referencia sólo al texto escrito que suele ser accesible en formato electrónico. Esto implica que sólo nos concentremos en las operaciones comunes de procesamiento de texto usadas por los sistemas tipicos de text mining. Esto incluye tokenización, part of speech, lematización y parsing.

3.1.1 Tokenización

Este primer paso en el análisis de texto es el proceso de separar el texto en unidades, los denominados tokens. Los tokens pueden variar su granularidad en función de las necesidades. De esta manera, la tokenización se puede dar en distintos niveles: el texto puede ser dividido en capítulos, secciones, párrafos, frases, palabras, sílabas o fonemas. Existen muchos algoritmos diferentes para cualquier nivel de tokenización aunque generalmente el texto suele fragmentarse en frases o palabras y en algunos sistemas en sílabas. No se trata de una tarea especialmente complicada, pero sí que hay que tener en cuenta una serie de problemas, por ejemplo:

- Combinación de letras y números en el nombre de determinados genes, proteínas u otras entidads biológicas: ACC1, SPO1, CWP1
- Números: Para los números se suele hacer otro tipo de indexación. Además, debe tenerse en cuenta que no todos los números significan lo mismo: Motorola 68000 (nombre propio), 68000 euros (cantidad), 2003 (año). ¿Cómo reconocer los números que son relevantes? En general, los números no se consideran términos índice, en el contexto que nos ocupa tampoco son especialmente relevantes como para añadir o restar significado al documento que se esté analizando.
- Guiones y signos: Los guiones se suelen eliminar para evitar inconsistencias de uso. Sin embargo, hay muchas palabras (generalmente nombres del campo de la biología o química) que poseen guiones que forman parte ingtegral de las mismas: AP-1-luciferase, FR-antigen. Para estos casos, se puede recurrir al uso de reglas que especifiquen excepciones.
- Palabras compuestas: Neuronal Network

En general, no resulta complejo implementar estas operaciones de texto. Sin embargo, deben estudiarse las distintas excepciones con cuidado ya que pueden provocar un importante impacto en el momento de la recuperación de documentos.

Existe otro tipo de tokenización especial, los n-gramas (n-grams), que son subsecuencias de n elementos (caracteres) de un texto dado. Así por ejemplo si tenemos "decaboxylase" los trigramas correspondientes serían "dec", "eca", "cab", etc. Los n-gramas se emplean a menudo en sistemas de reconocimiento de patrones para determinar la probabilidad de que una palabra dada aparezca en un texto (útil a la hora de encontrar menciones de distintas entidades biológicas en un texto) o en el proceso de recopilación de información cuando es necesario encontrar documentos similares dado un documento y una base de datos de documentos de referencia.

3.1.2 Eliminación de stopwords

Las palabras que son más frecuentes en los textos de una colección no son buenos discriminantes y se denominan stopwords. Artículos, preposiciones y conjunciones, así como algunos verbos, adverbios y adjetivos son candidatos naturales para formar parte de la lista de stopwords. Son característicos de cada lenguaje por lo que se requiere detectar el idioma de cada documento tratado. En bioinformática generalmente eso no supon ningún problema al estar casi toda la literatura en inglés.

La eliminación de stopwords permite reducir el tamaño de la estructura de indexación que se use. Sin embargo, hay controversia sobre sus beneficios. La eliminación de stopwrods puede empeorar el resultado de la consulta que se haga o de la información que se pretenda buscar dentro de un texto (si, por ejemplo buscamos la expresión "to be or not to be", puede que la lematización deje únicamente el término "be"), aunque ese problema es quizá de menor grado dentro del campo de la bioinformática.

3.1.3 Lematización

El propósito de la lematización o stemming es obtener un único término de indexación a partir de las diferentes variaciones morfológicas de una palabra (por ejemplo, representar "analysis", "analyzer" o "analyzing" mediante "analy"). Frecuentemente, una palabra no aparece exactamente en un documento, pero sí alguna variante gramatical de la misma como plurales, gerundios, sufijos de tiempo verbal, etc. Este problema puede resolverse con la sustitución de las palabras por su raíz (stem).

Un stem es la porción de una palabra que resulta de la eliminación de sus afijos (prefijos y sufijos). Los stems son interesantes ya que pemiten reducir variantes de la misma raíz gramtical a un concepto común. Consecuentemente, el stemming permite reducir el tamaño de la estructura de indexación ya que el número de términos índice se reduce. Además, permite ampliar la definicion de la información que poseemos o la consulta que se pretende satisfacer con las variantes morfológicas de los términos usados, mejorando así el performance de recuperación. Sin embargo, hay controversia en la literatura acerca de sus beneficios.

Se pueden distinguir varios tipos de estrategias de stemming: mediante un diccionario, n-gramas y eliminación de afijos. La aproximación mediante diccionario consiste en la bésqueda del stem en una tabla. Es un proceso simple pero la construcción del diccionario es costosa, por lo que esta aproximación no suele ser práctica. El stemming mediante n-gramas se basa en la identificación de diagramas y trigramas y se trata más de un procedimiento

de clustering que de stemming como tal. La eliminación de afijos es intuitiva, simple y se puede implementar eficientemente, Por ello la vemos en detalle.

En eliminacion de afijos, la parte más importante es la eliminación de sufijos porque la mayoría de las variantes de una palabra se generan con su introducción. El algoritmo más popular de eliminación de sufijos es el algoritmo de Porter [8]. Este algoritmo usa una lista para la detección de sufijos. La técnica se basa en aplicar una serie de reglas a los sufijos de las palabras del texto. Por ejemplo la regla $\{s \rightarrow \emptyset\}$ se utiliza para convertir las formas plurales en singulares sustituyendo la "s" por "nulo". siempre se busca el sufijo más largo de la palabra que empareje con los antecedentes en un conjunto de reglas. Las reglas de Porter están separadas en 5 grupos distintos.

Al aplicar lematización podemos provocar, sin embargo, dos tipos de errores:

- Infraradicación (understemming): Obtener distintas formas canónicas para una palabra.
- Sobrerradicación (overstemming): Obtener la misma forma canónica para dos palabras distintas.

3.1.4 Part of Speech

Consiste en el uso de etiquetas que representen conjuntos de categorías de palabras, basándose en el papel que las palabras pueden desempeñar en la frase en la que aparecen. El etiquetado *Part of Speech (PoS)* es la anotación de las palabras con su etiqueta correspondiente en función del contexto de la frase. Las etiquetas almacenan información del contenido semántico de la palabra. Los sustantivos denotan comunmente entidades tangibles o intenagibles mientras que las preposiciones expresan relaciones entre entidades. Aunque las etiquetas pueden variar de un sistema a otro, existen normalmente unas categorías básicas: artículo, nombre, verbo, adjetivo, preposición, número y nombre propio, aunque por supuesto las etiquetas pueden ser mucho más complicadas y elaboradas. Por ejemplo, el Corpus Brown [9] contiene 87 etiquetas básicas.

Existen muchas aproximaciones que pueden llevar a cabo este tipo de análisis. Los análisis más comunes están basados en reglas o son estadísticos basados en los modelos ocultos de Markov (HMM) Los etiquetados basados en modelos de Markov ([10][11][12]) estiman la probabilidad de que una secuencia de etiquetas pueda ser asignada a una secuencia de palabras. Con el fin de estimar los parámetros del modelo utilizado, se entrena el sistema en

una etapa anterior, usando un corpus anotado, como el corpus WSJ de Pen TreeBank [13].

Por otra parte, las aproximaciones basadas en reglas ([14][15][16]) usan información contextual para asignar etiquetas a palabras ambiguas o desconocidas mediante reglas, por ejemplo "Si la palabra X es precedida por un determinante y seguida por un nombre se trata de un adjetivo". También pueden usar información morfológica, por ejemplo que la palabra termine en "ing" indica que se trata de un verbo ([17]) o en las letras mayúsculas o la puntuación, por ejemplo si es un nombre y empieza por mayúscula, se trata de un nombre propio.

Los sistemas basados en reglas suelen requerir un conjunto de entrenamiento que esté etiquetado previamente, por lo que se consideran sistemas de aprendizaje supervisado, aunque sí que existen algunos sistemas no supervisados ([18]).

3.1.5 Análisis sintáctico

Es el proceso de determinar la estructura sintáctica completa de una frase. Los sistemas que llevan a cabo este tipo de análisis toman como entrada una secuencia de tokens extraídos del texto original. La salida suele ser un árbol sintáctico, cuyas hojas corresponden con las palabras del texto y cuyos nodos internos representan estructuras sintácticas, identificadas por etiquetas gramaticales, como: *sustantivo, verbo, sujeto, predicado, etc.* Actualmente no existe un sistema lo suficientemente eficiente que analice sintáticamente un texto sin ningún tipo de restricción. Los algoritmos estándar suelen consumir muchos recursos en corpora grandes y no son lo suficientemente robustos.

Una alternativa es realizar el mismo análisis que identifique las partes de una oración, pero sin especificar su estructura interna ni su papel en la oración principal, es decir, realizando el análisis con menor granuralidad. Esta opción tiene la ventaja de ser más rápida y robusta, aunque evidentemente el análisis realizado es menos profundo, ha de alcanzarse un compromiso entre ambas cuestiones. Suele emplearse en un paso previo de preproceso y después llevar a cabo un análisis más exahustivo. También permite identificar relaciones entre objetos, el sujeto y complementos espaciales o temporales dentro de la oración.

3.2 Minería de Textos

La Minería de Textos o Text Mining tiene como objetivo examinar una colección de documentos no estructurados escritos en lenguaje natural y

descubrir información no contenida en ningún documento individual de la colección; en otras palabras, trata de obtener información sin haber partido de algo ([19]). Aunque se apoya en técnicas de minería de datos ([20]) al trabajar con textos, se invierte un mayor porcentaje del esfuerzo en el preproceso de la colección de documentos, así se puede decir que la minería de textos es un área multidisciplinaria basada en la recuperación de información, minería de datos, aprendizaje automático, estadísticas y NLP. Además del preproceso de documentos, la minería de texto cubre también el almacenamiento de representaciones intermedias, las técnicas para analizarlas (tales como clustering ([21], [22]), análisis de tendencias ([23]) o mediante reglas asociativas ([24]) y visualización de los resultados ([25], [26]).

Un sistema típico de minería de texto comienza con una colección de documentos, sin ningún tipo de etiqueta. Los documentos son etiquetados en primer lugar por categorías, o por términos o relaciones extraídos directamente de los documentos. Este proceso se denomina *categorización de textos*, y divide enormes colecciones de textos en subconjuntos que estén interrelacionados por algún criterio predefinido. Ésta es una subárea de Information Retrieval que se verá más adelante. En la siguiente fase se utilizan operaciones de minería de datos sobre los documentos en base a las categorías asignadas y a las entidades y relaciones encontradas dentro del texto (mediante IE).

Una aplicación muy popular del text mining es explicada en [27]. Se intenta extraer información derivada de colecciones de texto. Teniendo en cuenta que los expertos sólo pueden leer una pequeña parte de lo que se publica en su campo, por lo general no se dan cuenta de los nuevos desarrollos que se suceden en otros campos. Así, se ha demostrado cómo cadenas de implicaciones causales dentro de la literatura médica pueden conducir a hipótesis para enfermedades poco frecuentes, algunas de las cuales han recibido pruebas de soporte experimental. Investigando las causas de la migraña, se extrajeron varias evidencias a partir de títulos de artículos presentes en la literatura biomédica. Algunas de esas claves fueron:

- El estrés está asociado con la migraña.
- El estrés puede conducir a la pérdida de magnesio.
- Los bloqueadores de canales de calcio previenen algunas migrañas.
- El magnesio es un bloqueador natural del canal de calcio.
- La depresión cortical diseminada (DCD) está implicada en algunas migrañas.
- Los niveles altos de magnesio inhiben la DCD.

- Los pacientes con migraña tienen una alta agregación plaquetaria.
- El magnesio puede suprimir la agregación plaquetaria.

Esta información sugieren que la falta de magnesio podría representar un papel importante en algunos tipos de migraña, una hipótesis que no existía en la literatura y que se encontró mediante esas claves. De acuerdo con Swanson ([28]), estudios posteriores han probado experimentalmente esta hipótesis obtenida por text mining con buenos resultados.

3.3 Recuperación de la Información

La recuperación de información (Information Retrieval o IR) ([29], [30]) se ocupa de la representación, almacenamiento, organización y acceso a la información ([31]). Dada una base de datos de documentos extensa (bases de datos, ontologías, diccionarios, internet), y una información específica (que generalmente es expresada como una consulta por un usuario), el objetivo de los métodos de recuperación de información es rescatar los documentos de la base de datos que satisfagan la información dada. Naturalmente, ésto se debe conseguir rápidamente y de manera eficaz. Estos documentos pueden ser textos, pero también sonidos, imágenes o cualquier otro tipo de datos, aunque en el campo de la bioinformática, lo que interesa genralmente es la bésqueda de textos. Dentro de este tipo de búsqueda, existen distintas aproximaciones para recuperar información. Las tres clásicas (los métodos más usados) son el modelo booleano, el vectorial y el probabilístico, sin embargo existen muchas más como las redes Bayesianas, redes neuronales, redes de inferencia, etc.

3.3.1 Modelo booleano

Existen muchas maneras de expresar una determinada información que se necesita satisfacer (una consulta). Una manera simple y muy común de hacerlo es a través de una consulta de tipo booleano. El modelo booleano es un modelo de recuperación simple basado en la teoría de conjuntos y el álgebra de Boole. El usuario proporciona un termino (*DNA*) o una combinacion booleana de términos (*DNA* and *lipid* utilizando operadores tales como AND (los documentos han de contener todas las palabras) u OR (los documentos han de contener alguna de las palabras). De esta manera el grado de relevancia de un documento es binario, es decir, una información determinada es relevante o no lo es. El resultado es el conjunto de todos los documentos de la base de datos que satisfagan las restricciones de la consulta, por ejemplo, que contengan los términos *DNA* y *lipid*. Esta estrategia es la seguida por la

base de datos de literatura biomédica PUBMED y por otras bases de datos de textos, incluso por los motores de búsqueda de internet. Se implementa mediante una estructura de índices que apunta a todos los términos en la base de datos de documentos entera. Cada término puede ser una única palabra ("polymerase") o un conjunto de ellas ("polymerase chain reaction"). Una práctica común es la de omitir del índice de términos aquéllas palabras que sean muy frecuentes y carentes de significado, como las preposiciones (Tal y como se contó en la sección 3.1.2). La estructura de índices cotiene todos los términos, tipicamente ordenados alfabéticamente para facilitar el acceso, y por cada término guarda una referencia a todos los documentos de la base de datos que lo contienen.

Cuando un usuario realiza una consulta, se busca en la estructura de índices y se devuelven los documentos que contienen el término o combinación de términos que se buscan. Existen varios métodos para crear índices y usarlos.

aunque este tipo de estrategias tienen la ventaja de ser muy rápidas, tienen algunas limitaciones:

- El número de documentos recuperado puede llegar a ser prohibitivamente grande.
- Una parte substancial de los documentos recuperados puede ser irrelevante para el usuario.
- Muchos documentos que sí son relevantes pueden no ser devueltos. Por ejemplo, si buscamos en PubMed "OLE1", muchos documentos que hablan de *OLE1* pero a través de un sinónimo (por ejemplo, "DNA repair protein" o "fatty-acid desaturase 1") no serían recuperados.

En este punto es importante hablar de polisemia y sinonimia. Polisemia es el conocido fenómeno por el cual una palabra puede tener muchos significados diferentes, en función del contexto. Debido a esto, si por ejemplo buscamos en PubMed por el término "Cytosine Deaminase" bajo su acrónimo "CD", nos encontraremos con un buen número de documentos que hagan referencia al concepto que buscamos, pero también recuperaremos muchos documentos que hablen de "Crohn's Disease" (también conocido por "CD") que no tiene nada que ver. Esta es la causa del segundo problema relatado anteriormente. Por otra parte, el tercer problema es debido a la sinonimia, que hace que un mismo pueda ser referido en distintos documentos mediante diferentes nombres.

Sin embargo, el modelo booleano es muy popular, sobre todo debido a sus sencillez y a que es una de las primeras ideas que surgen en el diseño de un sistema IR. Su sencillez hace que sea muy fácil de formalizar e implementar.

3.3.2 Modelo vectorial

El modelo vectorial descarta las soluciones binarias y propone un sistema en el que las coincidencias parciales son posibles, gracias a que se asignan pesos no binarios a los términos que aparecen en el texto. Tanto los documentos de la base de datos como la consulta el usuario son ahora vistos como vectores de términos. La tarea de recuperación busca dentro de la base de datos los documentos que son mas similares al vector consulta. Existen diversas maneras de medir la similitud entre vectores de documentos ([32], [22]).

Se elaboran vectores de términos a partir de los documentos seleccionando un conjunto de palabras que sea útil para discriminar unos textos de otros (se denominan términos o keywords). En los sistemas modernos todas las palabras del texto se consideran términos, excepto las stopwords o palabras vacías. Se puede enriquecer esto con procesos de lematización (stemming), etiquetado e identificación de frases. A cada uno de los términos que aparecen en el vector hay que asignarle un peso en función de la frecuencia con la que aparece la palabra en el documento o en la colección de documentos entera.

Sea $\{t_1, ..., t_k\}$ el conjunto de términos y $\{d_1, ..., d_N\}$ el de documentos. Un documento d_i se modela como un vector

$$d_i \longrightarrow \vec{d_i} = (w(t_1, d_i), ..., w(t_k, d_i))$$

donde $w(t_r, d_i)$ es el peso del término t_r en el documento d_i. Dicho peso representa la frecuencia de aparición del término en el documento o su nivel de importancia. La elección de los pesos de los términos puede influir significativamente en los resultados de la búsqueda, de esta manera han aparecido diferentes maneras para calcularlos.

Una representación intuitiva es la binaria, donde el peso es o bien 1 o bien 0, correspondiendo con la presencia o ausencia del término en el documento. Dicha representación se ve en la ecuación (3.1).

$$w(t_r, d_i) = w_{ri} = \begin{cases} 1 & \text{si } t_r \in d_i, \\ 0 & \text{en otro caso.} \end{cases} \tag{3.1}$$

A pesar de ser una representación clara y simple, no tiene en cuenta diversas propiedades de los documentos y términos que pueden mejorar la calidad de la búsqueda. Por ejemplo, una simple extensión del sistema binario usa como peso el número de veces que aparece el término en el documento. De manera intuitiva se puede llegar a la conclusión de que aquel documento en el que algún término de la consulta aparezca muchas veces va a ser considerado como relevante para el usuario. Formalmente sería obtendríamos la ecuación (3.2).

$w_{ri} = n_{di} \iff t_i$ aparece en el documento d_i un número de veces igual a n_{di} , $0 \leq n_{di}$ (3.2)

Esta aproximación tiene en cuenta la frecuencia de aparición del término en el documento, pero no considera el tamaño del documento. Un documento pequeño pero muy relevante puede contener menos apariciones de los términos de la consulta que un documento mucho mas extenso pero menos relevante. Para corregir esto se puede normalizar el peso calculado en la ecuación (3.2), dividiendo por el número total de términos en el documento, que denotamos con N_d. De esta manera, la fórmula para calcular del peso sería la que indica la ecuación (3.3).

$$w_{ri} = \frac{n_{di}}{N_d} \tag{3.3}$$

Ahora podemos hacer otra consideración, si un término t_1 de la consulta aparece de manera frecuente en muchos documentos de la base de datos, mientras que otro t_2 es más *raro* o específico, los documentos que contengan el término t_2 deberían ser considerados más relevantes para el usuario frente a los que contengan el término t_1, más frecuente. Esto contado de manera intuitiva, es lo que formalizan una familia de esquemas de peso comunmente conocidos como TFIDF (acrónimo de "Term Frequency x Inverse Document Frequency"). Bajo este esquema general, el peso se calcula como indica la ecuación (3.4).

$$w_{ri} = tf_{ri} \cdot idf_r \tag{3.4}$$

donde tf_{ri} es la medida local de la frecuencia del término t_r en el documento d_i, y idf_r es la medida global, inversamente proporcional al número de documentos que contienen t_r en toda la base de datos.

Existen muchas maneras de calcular la medida local tf_{ri}. Por ejemplo, hemos visto $tf_{ri} = 1$ (Eq. 3.1) o $tf_{ri} = n_{ri}$ (Eq. 3.2), donde en ambos casos $idf_r = 1$. Otras alternativas serían las que se muestran en las ecuaciones (3.5) y (3.6).

$$tf_{ri} = 1 + ln(n_{ri}) \tag{3.5}$$

$$tf_{ri} = k + (1 - k) \cdot \frac{n_{ri}}{max_j[n_{ji}]} \tag{3.6}$$

donde k es una constante, $0 \leq k \leq 1$, y el denominador es la moda del documento d_i, es decir, la frecuencia del término que aparece más veces en el documento.

De la misma manera, existen varias opciones para calcular la medida global, idf_r. Por ejemplo, denotamos por N_r al número total de documentos que contienen el término t_r en la base de datos. Una expresión simple para idf_r sería entonces la de la ecuación (3.7).

$$idf_r = \frac{1}{N_r} \tag{3.7}$$

Otras alternativas serían (3.8) o (3.9), donde N denota el número total de documentos en la base de datos.

$$idf_r = ln(1 + \frac{N}{N_r}) \tag{3.8}$$

$$idf_r = ln(\frac{N - N_r}{N_r}) \tag{3.9}$$

Se pueden encontrar muchos estudios en la literatura de recuperación de la información al respecto de qué esquema de peso es mejor ([32], [33], [34]). En concreto, se puede ver que uno de los más usados es el representado por la ecuación (3.10).

$$w_{ri} = \frac{n_{ri} \times idf_r}{|\vec{d_i}|} = \frac{n_{ri} \times ln\frac{N}{N_r}}{\sqrt{\sum_{s=1}^{k}(n_{si} \times ln\frac{N}{N_s})^2}} \tag{3.10}$$

Si un término aparece mucho en un documento, se supone que es importante en ese documento (n_ri crece), pero si aparece en muchos documentos, entonces no es útil para distinguir a un documento de los demás (idf_r decrece).

Hemos visto varios métodos usados para representar documentos y consultas mediante vectores. Usando esta representación, podemos aplicar medidas de similitud de vectores para calcular la similitud entre un par de documentos o entre una consulta y cada documento de la base de datos.

Existen muchas edidas de similitud entre vectores n-dimensionales. Sin embargo la mas conocida, fuera del ámbito de recuperación de información, es la distancia Euclídea. Cuanto menor sea la distancia, más similares serán los documentos. Formalmente, la distancia Euclídea entre dos vectores de dimensión n, $V_1 = \langle v_{11}, ..., v_{1n} \rangle$ y $V_2 = \langle v_{21}, ..., v_{2n} \rangle$ se define como en la ecuación (3.11).

$$d_{Euc}(V_1, V_2) = \sqrt{\sum_{i=1}^{n}(v_{1i} - v_{2i})^2} \tag{3.11}$$

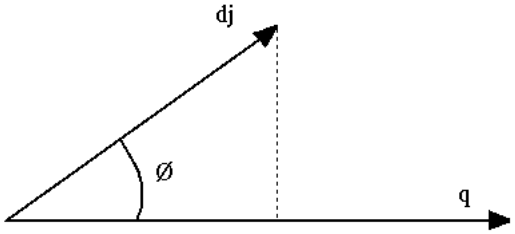


Figura 3.1: Coseno de dos vectores como medida de similitud entre documentos

Se puede ver gráficamente en la figura 3.1 la distancia Euclídea entre dos vectores de dimensión 2, en ese caso el coseno de Ø es la similitud entre dj y q. Se aprecia que la longitud de los vectores afecta significativamente a la distancia que existe entre ellos. En el contexto de los documentos, esto significa que dos documentos que contienen muchos términos tienden a divergir más que otros que contienen menos términos.

La medida de similitud más extendida dentro de los sistemas de recuperación de la información y que no depende de la longitud de los vectores es la distancia coseno ([33]). Se trata del coseno del ángulo que forman dos vectores, formalmente, sean dos vectores V_1 y V_2, cuyas respectivas longitudes son $\|V_1\|$ y $\|V_2\|$, el coseno del ángulo que forman se define como en la ecuación (3.12).

$$cos(V_1, V_2) = \vec{V_1} \cdot \vec{V_2} = \frac{\sum_{j=1}^{n} v_{1j} \cdot v_{2j}}{\|V_1\| \cdot \|V_2\|} \tag{3.12}$$

Al contrario que la medida Euclídea, no se trata de *distancia* sino de *similitud*. Esto se traduce en que su valor, que está contenido dentro del rango [0,1], cuanto más cerca está de 1, más similares serán los dos vectores y cuanto más cerca de 0, más divergen los vectores el uno del otro (más perpendiculares son).

Bajo una representación binaria, una simple consulta booleana de tipo disyuntivo (e.g. "DNA" or "AIDS") puede ser transformada en un vector con un 1 en las posiciones correspondientes a los términos de la consulta y 0 en todas las demás. Una búsqueda de vectores de documentos que encajen con el vector consulta, usando la medida del coseno, devolverá exactamente los mismos documentos que un sistema booleano basado en índices. Sin embargo, diferentes sistemas de pesado devolverán distintos resultados. Además, para consultas que contienen varios términos, el uso de un sistema de búsqueda por similitud tiene dos ventajas principales. Primero, no es necesario especificar

una consulta mediante una expresión booleana complicada que puede no corresponder con lo que se quiere buscar, de esta manera, se puede utilizar como consulta incluso un documento entero. Y en segundo lugar, devuelve documentos que encajan mejor con la información necesaria que un sistema booleano, y además ordena los resultados acorde con el nivel de similitud que guarda con el conjunto de términos de la consulta. Esto es porque es la combinación de palabras de la consulta la que determina el resultado, y no una determinada palabra específica.

Por ejemplo, consideramos la consulta "paciente cáncer Kaposi Sarcoma VIH" en una base de datos biomédica cuyo sistema de búsqueda es vectorial. Mediante esta consulta se pretende recuperar todos los artículos médicos acerca de los pacientes con VIH que tienen un tumor de sarcoma de Kaposi y no los artículos acerca de gente que sea paciente o que pertenezca al signo zodiacal de cáncer. Las palabras "VIH", "kaposi", "sarcoma", sirven en este caso para desambiguar las palabras "paciente" y "cáncer", dando así mayor puntuación a los documentos que traten de pacientes con VIH que sufren un Sarcoma de Kaposi. Nótese que no es necesario especificar ninguna consulta mediante operadores booleanos y que ninguno de los documentos devueltos tiene que tener todos los términos especificados en la consulta. Este ejemplo muestra que la polisemia de las palabras "cáncer" y "paciente" se resuelve de manera implícita por la presencia de los otros términos, sin necesitar que ninguno de los términos deba de aparecer en los documentos un número elevado de veces.

Aunque es sin lugar a dudas muy útil, el modelo vectorial basado en el cálculo del coseno tiene algunos inconvenientes. Ya se ha hablado de la polisemia y la sinonimia como los problemas principales que impone el lenguaje natural en los sistemas de recuperación de la información. El modelo vectorial ataja esos problemas hasta cierto punto, como se ve en el ejemplo anterior. Sin embargo, la presencia de las palabras de manera explícita sigue siendo un problema. Si ningún documento de la base de datos biomédica del ejemplo contiene el término "VIH" para referirse a los pacientes con el virus de inmunodeficiencia (por ejemplo, utilizan la palabra SIDA), se podrán recuperar todos los documentos que hablan de los pacientes con Sarcoma de Kaposi, aun cuando no estén infectados con el VIH, simplemente porque ningún documento encaja con el término "VIH" de la consulta.

3.3.3 Modelo probabilístico

Una manera de relajar la dependencia entre los resultados recuperados y los términos explícitos de la consulta es usando el modelo probabilístico. El modelo probabilístico clásico fue propuesto en 1976 por Robertson y Sparck

Jones ([35]) y más tarde sería conocido como modelo de recuperación binaria independiente (BIR). Este modelo trata de abarcar el problema de la recuperación de información dentro del marco de la Probabilidad. Dada una cosulta de un usuario, el modelo presupone que existe un conjunto de documentos que contiene todos los documentos que son relevantes y ninguno más, a este conjunto le vamos a llamar respuesta ideal. Si tuviésemos una descripción de cómo debe ser esa respuesta ideal, no habría muchos problemas en recuperar el conjunto de documentos, el problema es que inicialmente no disponemos de esa información. así que se propone una respuesta ideal inicial, y se da al usuario la oportunidad de decir qué documentos son relevantes y cuáles no del conjunto propuesto. Repitiendo este proceso, el sistema debería ser capaz de conocer la descripción de la respuesta ideal y devolverla.

El modelo probabilístico se basa en el siguiente supuesto: dada una consulta q y un documento d_j en la colección, el modelo probabilístico trata de estimar la probabilidad de que el usuario encuentre a dicho documento relevante. El modelo asume que esta probabilidad de relevancia depende únicamente de la consulta hecha y del propio documento. De esta manera, se asume que existe un subconjunto de entre todos los documentos que el usuario quiere como respuesta a la consulta q. Ese conjunto R que forma la respuesta ideal maximiza la probabilidad de relevancia para el usario. Todos los documentos que se encuentren en el conjunto R se dice que son relevantes para la consulta y los que no están en la consulta son no relevantes.

El problema es que no se dice de qué manera calcular la probabilidad de que un determinado documento sea relevante o no. Dada una consulta q, la relevancia de un documento d_j se calcula como indica la ecuación (3.13)

$$sim(d_j, q) = \frac{P(d_j \ relevante \ para \ q)}{P(d_j \ no \ relevante \ para \ q)} \tag{3.13}$$

Para el modelo probabilístico, los pesos de los términos son siempre binarios. Una consulta q es un subconjunto del índice de términos. Sea R el conjunto de documentos conocidos (o propuesto inicialmente) que son relevantes. Sea $\bar{R}$ el conjunto complementario de R (el conjunto de todos los documentos que no son relevanes). Sea $P(R|\vec{d_j})$la probabilidad de que el documento d_j sea relevante para la consulta q y $P(\bar{R}|\vec{d_j})$ la probabilidad de que d_j nos sea relevante para q. De esta manera, la similitud $sim(d_j, q)$ de el documento d_j y la consulta q se define como en (3.14).

$$sim(d_j, q) = \frac{P(R|\vec{d_j})}{P(\bar{R}|\vec{d_j})} \tag{3.14}$$

Mediante Bayes llegamos a (3.15)

$$sim(d_j, q) = \frac{P(\vec{d_j}|R) \times P(R)}{P(\vec{d_j}|\bar{R}) \times P(\bar{R})} \tag{3.15}$$

donde $P(\vec{d_j}|R)$ es la probabilidad de que el documento d_j sea seleccionado de el conjunto R de documentos relevantes y $P(R)$ la probabilidad de que un documento seleccionado aleatoriamente de la colección entera sea relevante. El razonamiento con $P(\vec{d_j}|\bar{R})$ y $P(\bar{R})$ es análogo. Como $P(R)$ y $P(\bar{R})$ son iguales para todos los documentos de la colección, podemos escribir:

$$sim(d_j, q) \sim \frac{P(\vec{d_j}|R)}{P(\vec{d_j}|\bar{R})}$$

Asumiendo independencia del índice de términos:

$$sim(d_j, q) \sim \frac{(\Pi_{g_i(\vec{d_j})=1} P(k_i|R)) \times (\Pi_{g_i(\vec{d_j})=0} P(\bar{k_i}|R))}{(\Pi_{g_i(\vec{d_j})=1} P(k_i|\bar{R})) \times (\Pi_{g_i(\vec{d_j})=0} P(\bar{k_i}|\bar{R}))}$$

donde $P(k_i|R)$ es la probabilidad de que el término k_i esté presente en un documento seleccionado aleatoriamente de el conjunto R y $P(\bar{k_i}|R)$ es la probabilidad de que el término k_i no esté presente en un documento seleccionado aleatoriamente del conjunto R. De nuevo, las probabilidades asociadas a $\bar{R}$ tienen una explicación análoga.

Haciendo cálculos, sabiendo que $P(k_i|R)+P(\bar{k_i}|R) = 1$, llegamos a (3.16).

$$sim(d_j, q) \sim \sum_{i=1}^{t} W_{iq} \cdot W_{ij} \cdot \left(log \frac{P(k_i|R)}{1 - P(k_i|R)} + log \frac{1 - P(k_i|\bar{R})}{P(k_i|\bar{R})} \right) \tag{3.16}$$

$$= \sum_{k_i \in q \bigcap d_i} \left(log \frac{P(k_i|R)}{1 - P(k_i|R)} + log \frac{1 - P(k_i|\bar{R})}{P(k_i|\bar{R})} \right) \tag{3.17}$$

donde W_{ij} es 1 si k_i aparece en d_j y 0 en otro caso. Como no conocemos el conjunto R desde el principio, necesitamos algún método para calcular tanto $P(k_i|R)$ como $P(k_i|\bar{R})$. Existen muchas alternativas para hacer esto, aunque sólo veremos dos. Al comienzo, se pueden hacer una serie de suposiciones básicas, como asumir que $P(k_i|R)$ es constante para todos los términos índice k_i (típicamente se supone $P(k_i|R) = 0{,}5$) y que la distribución de los términos índice en el conjunto $\bar{R}$ es igual a la distribución de los términos índice en toda la colección de documentos (tendríamos que $P(k_i|\bar{R}) = n_i/N$, donde n_i es el número de documentos que contienen el término k_i y N es el número total de documentos que tenemos). Tras una primera iteración se

recuperan V documentos, fijamos un umbral r y nos quedamos con el subcojunto de los r documentos con mayor probabilidad. Sea v_i el número de documentos recuperados que contienen el término k_i. Ahora debemos mejorar los valores de $P(k_i|R)$ y $P(k_i|\bar{R})$ para mejorar los resultados, podemos asumir que podemos aproximar $P(k_i|R)$ por la distribución de los términos índice k_i en los documentos recuperados y que podemos aproximar $P(k_i|\bar{R})$ considerando que los documentos no recuperados son no relevantes. Con eso llegaríamos a:

$$P(k_i|R) = \frac{v_i}{V}$$
$$P(k_i|\bar{R}) = \frac{n_i - v_i}{N - V}$$

Este proceso se repite recursivamente, mejorando $P(k_i|R)$ y $P(k_i|\bar{R})$ sin la necesidad de la intervención de un humano, sin embargo, se puede usar también la asistencia del usuario para definir el subconjunto V, tal y como sugería la idea original del modelo.

La mayor ventaja de este modelo es que los documentos son ordenados de manera decreciente respecto a la probabilidad que tienen de ser relevantes. Pero tiene una serie de desventajas como por ejemplo:

- La necesidad de suponer inicialmente la separación de documentos relevantes y no relevantes, es decir que se comienza adivinando y luego se refina esa apuesta iterativamente.
- El método no considera la frecuencia con la que un término aparece en un documento, sino que ve cada documento como un conjunto de términos (la información es binaria).
- Necesita presuponer que los términos son independientes.

Sin embargo tiene una base teórica que es distinta al del modelo vectorial y permite algunas extensiones que sí son bastantes populares. En ([36]) se ve cada documento de la colección como un *modelo de lenguaje* donde los términos siguen aproximadamente una distribución multinomial. Los documentos devueltos por el sistema son aquellos que se consideran firmes candidatos a ser el *modelo de lenguaje fuente* de la consulta hecha. Otra aproximación fue la desarrollada en [37], donde los documentos son vistos como si hubiesen sido generados por algún modelo probabilístico en el que la semántica de los términos seleccionados fuese determinada estocásticamente por un conjunto de variables escondidas. Otra aproximación más es la identificación probabilística de temas por Shatkey et. al. ([38]) donde se ve a los documentos de la misma manera que en [36], como un modelo de lenguaje donde los términos, en este caso, siguen la distribución de Bernuilli.

3.3.4 Latent Semantic Indexing

Como ya se ha dicho antes, resumir el contenido de los documentos y las consultas a un conjunto de términos puede ocasionar problemas a la hora de recuperar información debido a que muchos documentos no relevantes pueden ser incluidos dentro del conjunto de respuesta por compartir términos con la consulta y a que documentos que sí son relevantes, pero que no tengan ninguno de los términos que aparecen en la consulta, no son recuperados.

La sinonimia y la polisemia son las dos principales causantes de estos problemas. Al hablar de sinonimia nos referimos al hecho de que existen diferentes maneras de llamar a una misma cosa. Los usuarios en diferentes contextos, con diferentes necesidades, conocimientos, hábitos lingüísticos describirán la misma información usando distintos términos. La sinonimia es la principal culpable de disminuir el valor de "recall" de los sitemas de recuperación.

Por polisemia se entiende al hecho de que muchas palabras puedan tener más de un único significado. Una misma palabra usada en diferentes contextos o por distintas personas puede llegar a significar cosas completamente distintas. De esta manera, el hecho de que un determinado término aparezca en una consulta no significa necesariamente que un documento que contenga dicho término sea de interés. La polisemia hace que los sistemas de recuperación obtengan una baja "precisión".

Los sistemas de recuperación e indexado no son capaces de superar el problema de la sinonimia y la polisemia debido principalmete a tres factores:

- Los términos indice identificados no son suficientes. sólo se emplea uan fracción de todos los posibles términos que existen para describir a un documento correctamente. Eso es debido en parte a que los mismo documentos no contienen todas esas palabras y a que determinados sistemas omiten algunas de las palabras o simplemente las desechan.

- El segundo factor es el propio método usado para intentar solventar el problema de la polisemia. Algunos sistemas reducen su campo a un vocabulario controlado e incluso utilizan la intervención del ser humano para traducir las palabras a los términos conocidos. Esto no sólo es muy caro y poco eficiente, sino que ni si quiera es necesariamente efectivo. Otros intentan desambiguar las palabras de una consulta a través del resto de palabras mediante expresiones booleanas, pero en este caso se necesita que el usuario conozca el álgebra de Boole o que utilice los términos necesarios en la consulta para que la desambiguación sea efectiva, y no siempre se da el caso.

- El tercer factor es algo más técnico. Este tipo de sistemas tratan cada término como si fuera independiente de todos los demás [Veáse Van Rijsbergen [9]]. De esta manera las palabras que aparecen casi siempre juntas en un documento son tratadas o "puntuadas" de la misma manera que aquellas palabras que sólo aparecen en el mismo documento en raras ocasiones.

El método *Latent Semantic Indexing* (LSI), propuesto inicialmente en [39], utiliza la relación implícita que existe en términos y documentos, pero a nivel semántico, prentendiendo así mejorar la detección de aquellos documentos que sean relevantes en función de los términos que se hayan encontrado en la consulta. Se vale de un método matemático conocido como Singular Value Descomposition (SVD), cuyo cometido es el de factorizar matrices. En este caso se trata de una matriz de términos por documentos, que una vez factorizada representa la *estructura semántica latente* entre la colección de documentos y los términos contenidos. El motivo de usar SVD es el de reducir la dimensionalidad del espacio de términos, que terminan agrupándose como conceptos (ideas más generales que pueden englobar uno o más términos). De esta manera se reducen los efectos de la sinonimia y la polisemia.

Sea t el número total de términos índice y N el número total de documentos en la colección. Se define $\vec{M} = (M_{ij})$ como la matriz de términos por documentos asociada con t filas y N columnas. A cada elemento M_{ij} de la matriz se le asigna un peso W_{ij} asociado al par término-documento $[k_i, d_j]$. El peso W_{ij} puede ser binario o generado mediante una técnica de pesado, como la TFIDF comentada en el modelo vectorial. LSI propone la descomposición de $\vec{M}$ mediante SVD obteniendo otras tres matrices, de la siguiente manera:

$$\vec{M} = \vec{T_0} \cdot \vec{S_0} \cdot \vec{D_0^t}$$

de tal manera que tanto $\vec{T_0}$ como $\vec{D_0}$ tienen columnas ortonormales y que $\vec{S_0}$ es diagonal de $r \times r$ donde $r = min(t, N)$ es el rango de $\vec{M}$. $\vec{T_0}$ y $\vec{D_0}$ son las matrices de los vectores singulares de la izquierda y la derecha y $\vec{S_0}$ es la matriz diagonal de los valores singulares. La descomposición SVD es única (salvo permutaciones de filas o columnas)y además, por convenio, los elementos de la diagonal de $\vec{S_0}$ han de ser todos positivos y ordenados de mayor a menor.

El método SVD además permite aproximar el modelo mediante matrices más pequeñas de una manera muy sencilla. Si lo valores singulares de la matriz $\vec{S_0}$ estén ordenados por magnitud, los k primeros elementos mayores deben ser conservados y el resto puestos a cero junto con las correspondientes columnas en $\vec{T_0}$ y $\vec{D_0}$. El producto de las matrices resultantes es la matriz $\vec{M_k}$, de rango k y aproximadamente igual a $\vec{M}$

$$\vec{M}_k = \vec{T}_k \cdot \vec{S}_k \cdot \vec{D}_k^t$$

donde k, $k < r$, es la dimensionalidad del *espacio de conceptos* reducido que se utiliza para representar los datos. Aunque la elección de k es un punto crítico para que el algoritmo funcione bien, generalmente se elige una k tal que la suma de los k primeros elementos de la diagonal de $\vec{S}_0$ represente al menos el 80% de la suma total de todos los elementos.

La relación entre dos documentos cualesquiera en el espacio reducido de dimensionalidad k se obtiene de la matriz $\vec{M}_k^t \vec{M}_k$.

$$\begin{aligned} \vec{M}_k^t \vec{M}_k &= (\vec{T}_k \vec{S}_k \vec{D}_k^t)^t \vec{T}_k \vec{S}_k \vec{D}_k^t \\ &= \vec{D}_k \vec{S}_k \vec{T}_k^t \vec{T}_k \vec{S}_k \vec{D}_k^t \\ &= \vec{D}_k \vec{S}_k \vec{S}_k \vec{D}_k^t \\ &= (\vec{D}_k \vec{S}_k)(\vec{D}_k \vec{S}_k)^t \end{aligned}$$

En la matriz resultante, el elemento (i, j) cuantifica la relación entre los documentos d_i y d_j. Para tener la similitud de todos los documentos con respecto a una consulta dada por un usuario, podemos simplemente modelar dicha consulta como si fuese un pseudo-documento que se encontrase en la matriz $\vec{M}$ original, por ejemplo el documento d_0. De esta manera, la primera fila de la matriz $\vec{M}_k^t \vec{M}_k$ contendrá el grado de similitud de todos los documentos de la colección respecto de la consulta.

Como las matrices usadas en el modelo LSI son de rango k, $k \ll t$ y $k \ll N$, indexan de manera muy eficiente a los documentos de la colección. Además, eliminan ruido y redundancias.

El método LSI, a pesar de ser una muy buena opción, tiene también sus desventajas:

- Es muy efectivo en colecciones pequeñas de documentos, pero no tanto en colecciones grandes.
- La transformación algebraica que se lleva a cabo hace que el método no sea capaz de devolver qué términos son responsables de la similitud de los documentos.

3.3.5 Modelo de redes neuronales

En un sistema de Recuperación de la Información, los vectores de documentos son comparados con los vectores consulta para calcular el grado de similitud

entre ellos. Esto se hace capturando y pesando los términos índice que aparecen en los documentos y en las consultas y comparando los patrones de unos y otros. Como las redes neuronales son conocidas por ser buenas encontrando patrones, es natural considerar su uso como un modelo alternativo para la recuperación de información.

Es un hecho constatado que nuestro cerebro esta compuesto por billones de neuronas. Cada neurona puede ser entendida como una unidad básicas de proceso que al recibir una determinada señal, es estimulada y reacciona emitiendo una serie de señales. Las señales emitidas por una neurona son enviadas a otras neuronas que pueden a su vez emitir nuevas señales de salida y así propagarse la señal inicial durante varias capas de neuronas. La señal será finalmente procesada en el cerebro y puede producir alguna reacción física en respuesta (por ejemplo, una acción motora).

Una red neuronal representa de manera muy simplificada mediante un grafo el conjunto de neuronas interconectadas en un cerebro humano. Los nodos del grafo son las unidades de proceso mientras que las aristas serían las conexiones sinápticas. Para simular el hecho de que la intensidad de las conexiones sinápticas en el cerebro humano cambian constantemente se asignan pesos a las aristas del grafo que forma la red neuronal. En cada momento el estado de un nodo se define por su nivel de activación (que es una función de su estado inicial y de las señales que recibe como entrada). Dependiendo de su nivel de activación, un nodo A enviará una señal a su vecino B. La intensidad de la señal enviada dependerá del peso asociado a la arista que conecta el nodo A y el nodo B.

Una red neuronal empleada en recuperación de la información puede ser definida como ilustra la figura 3.2. Se observa que la red neuronal está compuesta por tres capas: una para los términos de la consulta, otra para los términos de los documentos y la tercera para los documentos mismos. En este modelo los nodos de la primera capa (los términos de la consulta) son los que inician el proceso de inferencia enviando señales a los nodos de los términos de los documentos (segunda capa). Seguidamente, los nodos de la segunda capa propagan la señal (o no) hasta los nodos que representan a los documentos (tercera capa). así se completa la primera fase en la que una señal viaja desde los nodos de los términos de la consulta hasta los nodos de los documentos.

La red neuronal, sin embargo, no termina tras esta primera fase, sino que los nodos de los documentos generan nuevas señales y las propagan hacia atrás, hacia los nodos de los términos de los documentos (esa es la razón por la cual las aristas que conectan la segunda y a tercera capa son bidireccionales). El proceso se repite recursivamente mientras la señal se hace cada vez más débil, hasta que llega un momento en el que el proceso de activación se

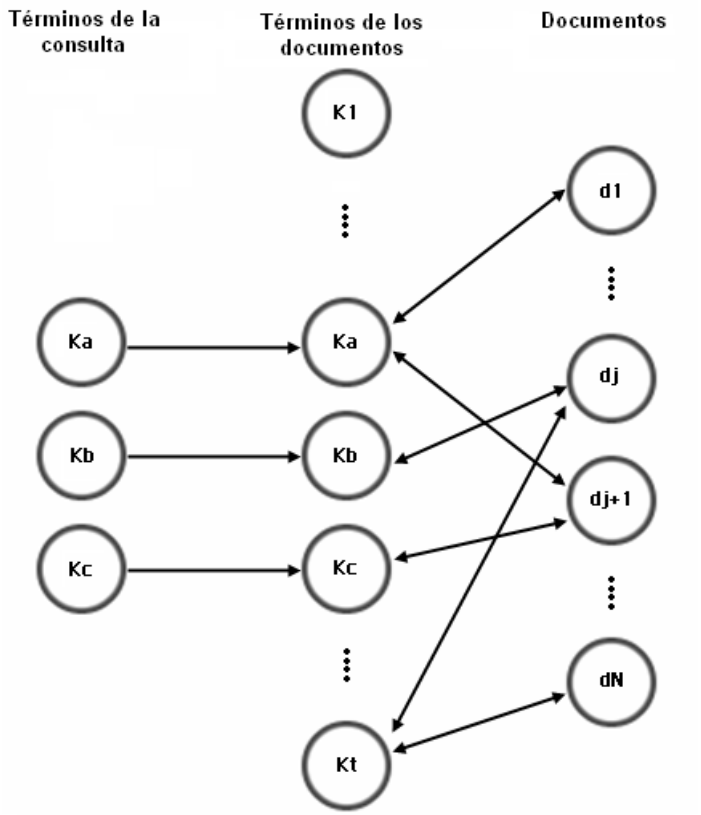


Figura 3.2: Un modelo de red neuronal para Recuperación de la Información, extraído de [31]

termina parando. Este proceso puede activar a un documento d_l aun cuando dicho documento no contenga ningún término de la consulta. así, el proceso entero puede ser interpretado como la activación de un tesauro integrado.

En primer lugar se asigna un nivel de activación igual a 1 a los nodos de los términos de consulta. éstos propagan a los nodos de la segunda capa señales que son atenuadas por los pesos normalizados de los términos de la consulta, $\bar{W_{iq}}$. Los pesos W_{iq} se definen del mismo modo que en el modelo vectorial, de tal manera que tenemos:

$$\bar{W_{iq}} = \frac{W_{iq}}{\sqrt{\sum_{i=1}^{t} W_{iq}^2}}$$

donde la normalización se hace usando la norma del vector consulta.

Una vez que las señales llegan a los nodos de los términos de los documentos, éstos envían nuevas señales a los nodos de los documentos. Estas señales son atenuadas por los pesos de los términos de los docuemntos normalizados W_{ij} que derivan de los pesos W_{ij} definidos en el modelo vectorial.

$$\bar{W_{ij}} = \frac{W_{ij}}{\sqrt{\sum_{i=1}^{t} W_{ij}^2}}$$

donde la normalización se hace usando la norma de los vectores de los documentos. De esta manera, en una primera vuelta, el nivel de activación del nodo asociado al documento d_j es el dado por:

$$\sum_{i=1}^{t} \bar{W_{iq}}\bar{W_{ij}} = \frac{\sum_{i=1}^{t} W_{iq}W_{ij}}{\sqrt{\sum_{i=1}^{t} W_{iq}^2} \times \sqrt{\sum_{i=1}^{t} W_{ij}^2}}$$

que encaja exactamente con la ecuación (3.12) descrita en el modelo vectorial.

Para mejorar la eficacia de la recuperación, la red cotinua con la difusión de la señal de activación. Esto modifica el ranking inicial del mismo modo que si el usuario especificara qué documentos son relevantes y cuales no dada una respuesta inicial por parte del sistema (método parecido al explicado en el modelo probabilístico). Para hacer que el proceso sea más efectivo, se puede asociar un umbral mínimo de activación que haga que aquellos nodos de documentos que reciban una señal tal que no supera dicho umbral, no la propaguen (Véase [40]).

No hay evidencia de que las redes neuronales consigan resultados superiores con colecciones de documentos generales. Sin embargo, una red neuronal presenta un modelo paradigmático alternativo y además, consigue recuperar

documentos aun cuando éstos no hayan sido relacionados inicialmente a los términos consulta.

3.3.6 CBR para recuperación de la información

Aunque no se incluye exactamente dentro de las técnicas de recuperación de la información, se puede citar la técnica de CBR como método de recuperación de documentos. El Razonamiento Basado en Casos (CBR) es una técnica de resolución de problemas que se basa en la manera de razonar del ser humano. Muchos estudios en Psicología afirman que la mente del ser humano trata de resolver determinadas situaciones utilizando información específica de experiencias anteriores (e.g. [41]). De esta manera, en CBR un nuevo problema será resuelto encontrando casos pasados similares y reutilizándolos, adaptando la solución a la situación del caso nuevo. Una de las ventajas es que el caso nuevo se incorpora a la base de conocimiento del sistema, poniándolo a disposición de problemas futuros, de esta manera a medida que se resuelven problemas mejorarán los resultados obtenidos. Además CBR permite combinar la información de problemas anteriores con conocimiento adicional del contexto en el que se trabaje, por ejemplo si es un sistema relacionado con la abogacía se puede incorporar información acerca de la legislación vigente. A los sistemas de este tipo se les considera CBR de conocimiento intensivo.

En [42] se explican las cuatro fases que generalmente componen el ciclo de un sistema CBR: recuperación, reutilización, revisión y recuerdo.

- La fase de recuperación se ocupa por una parte de determinar qué características del caso nuevo son las que van a permitir encontrar casos relevantes de la base de casos del sistema. Después se accederá a memoria para recuperar los casos más similares, se ordenarán según el grado de similitud y se escogerán aquellos que pasen de un determinado valor umbral o simplemente se selecciona aquel que presente más semejanza.

- En la fase de reutilización se utiliza el conocimiento incluido en el caso recuperado para resolver/clasificar el problema/situación actual. Aqué existen dos posibles alternativas, se puede ofrecer sin más la solución recuperada sin modificar o se puede adaptar previamente a la situación actual. Esto último requiere encontrar las diferencias entre el caso recuperado y el actual y aplicar algún mecanismo que sugiera cambios en función de esas diferencias encontradas.

- En la fase de revisión se pone a prueba la solución propuesta, ya sea siendo evaluada por un experto, o aplicándola directamente a un sistema real. En caso de no ser una solución adecuada se incorpora de

alguna manera la información al respecto, reparando la solución o incorporando algún mecanismo en el sistema que se encargue de llevar a cabo alguna estrategia de reparación.

- La fase de recuerdo es la que integra el nuevo caso con sus características más relevantes y su solución en la base de casos del sistema. Es esta fase la que hace que el sitema CBR mejore su funcionamiento a medida que va adquiriendo nuevas experiencias.

En el caso de recuperación de la información los documentos han de ser representados como conjuntos de características que se establecen durante la adquisición del conocimiento (fase necesaria) y la medida de similitud dependerá del dominio en que nos encontremos. Esta es la principal desventaja del uso de CBR, que se necesita poseer conocimiento previo del dominio en el que se está trabajando, lo que hace que sólo se pueda aplicar en campos muy limitados. Por otra parte, además de tratarse de una alternativa a los métodos tradicionales, el uso de CBR permite integrar información no textual que puede ayudar a mejorar los resultados obtenidos por éstos.

Algunos sistemas han sido implementados siguiendo esta línea como por ejemplo en [43], donde se implementa un sistema de recuperación de documentos al modo de una FAQ para la empresa privada, o en [44], donde el sitema SPYRO diseñado recupera en una primera fase documentos relevantes mediante CBR y una segunda fase utiliza técnicas de IR. En general, se trata de trabajos muy limitados y orientados a campos muy concretos.

3.3.7 Categorización de textos

Se trata de una tarea que a veces llevana cabo los sistemas de recuperación de la información. Se etiquetan los textos de lenguaje natural con categorías temáticas que se extraen de un conjunto previamente definido. Existen dos maneras de hacerlo, en la primera (*Ingeniería de Conocimiento*) ([45], [46]) el usuario define una serie de reglas manualmente que codifican la información de los expertos, estas reglas hacen que los textos sean después etiquetados correctamente. La otra aproximación esta basada en técnicas de Aprendizaje Automático o Machine Learning ([47], [48], [49], [50]) donde un proceso previamente entrenado clasifica automáticamente los textos a partir de un conjunto de textos preclasificados. Dentro de esta categoría podríamos enmarcar algunas de los trabajos mediante CBR comentadas en el punto anterior.

Un ejemplo de *Ingeniería de Conocimiento* es el sistema CONSTRUE ([45],[46]) implementado por el *Carnegie Group* para la agencia de noticias *Reuters*. Una regla del sistema consiste en definir una codición como una

disyunción de cláusulas conjuntivas seguida de una categoría como resultado. Por ejemplo, la siguiente regla identifica artículos que deben ser considerados como relevantes para "wheat":

```
If  ((wheat & farm) or
     (wheat & commodity) or
     (bushels & export) or
     (wheat & tones) or
     (wheat & winter & soft))
then Wheat
else ~Wheat
```

La principal desventaja de este método es el cuello de botella que supone la adquisición de conocimiento (como en el caso de la técnica de CBR comentada antes). Las reglas deben ser definidas manualmente por un ingeniero de conocimiento a través de la información que reciba de los expertos en el dominio. Si se modifica el conjunto de categorías, se necesita de nuevo la participación de ambas partes. Hayes et. al. ([45], [46]) consiguió un 90% de *recall* y *precisión* en un conjunto de test reducido (cerca de 723 documentos). Sin embargo, el proceso de desarollo fue demasiado costoso (tomó varios años) y el conjunto de test no era lo suficientemente significativo como para validar los resultados, no está claro si los resultados escalarían en un sistema más grande.

3.4 Extracción de la Información

Opuestamente a la recuperación de información, encargada de seleccionar los documentos más relevantes en función de las necesidades del usuario, la extracción de la información (*Information Extraction o IE)* es el nombre dado a cualquier proceso que recupera información que se encuentren de manera explícita o implícita en uno o más textos [51]. Se trata de una técnica de text mining que combinada con herramientas NLP, recursos léxicos y restricciones semánticas, proporciona modulos efectivos para identificar hechos y relaciones en la literatura.

Los sistemas de extracción buscan entidades, relaciones entre ellas u otros hechos específicos dentro de los textos. Permite además el etiquetado de los documentos, pero no tal y como se comentaba en el apartado correspondiente a *Text Categorization*, utilizando un conjunto de categoría predefinidas, sino que identifica conceptos explícitos y relaciones dentro de los textos, y asocia

partes específicas del documento con algún asunto que sea de interés, es utilizando estas entidades específicas, hechos y eventos encontrados como se puede etiquetar al documento, y no mediante categorías fijadas de antemano.

3.4.1 Arquitectura de los sistemas de extracción de la información

Según Shatkay et al. ([52]), un sistema de extracción de la información tiene tres o cuatro fases principales. La primera fase consiste en la tokenización, dividir el documento en bloques básicos. Estos bloques suelen ser palabras, oraciones o párrafos, en raras ocasiones se elige tener unos bloques más grandes (como capítulos o secciones). La segunda fase consiste en el análisis morfológico y léxico, asignar etiquetas PoS (Part of Speech) a las palabras, creación de sintagmas básicos (nominales o verbales) y desambig?ación de palabras o expresiones. La tercera fase trata del análisis sintáctico, estableciendo la conexión entre las diferentes partes de cada oración, explicado en una sección previa. La cuarta fase consiste en el *análisis de dominio*, donde se combina toda la información extraída en las fases anteriores para describir las relaciones entre las distintas entidades. El análisis de dominio lleva a cabo también un proceso de resolución de anáforas.

3.4.2 Resolución de anáforas

Uno de los principales desafíos que tienen los sistemas de text-mining es la resolución de anáforas, esto es, la habilidad para resolver co-referencias (varias palabras distintas refiriéndose a la misma entidad dentro del texto) (Hobbs, 1986).

Se ha concluido (Lappin y Leass, 1994) que, en general, resolver el problema entre nombres propios y alias o pseudónimos es algo más fácil, resolver el problema de los pronombres personales como *it, this, theses, he, she*, etc. es más difícil y resolver el problema en sintagmas nominales como "the two genes" es la tarea más complicada y propensa a errores.

Lo más común parece ser utilizar una técnica basada en conocimiento ([53]) donde todos los antecedentes de cada frase que haga referencia que se quiera estudiar son tenidos en cuenta. Estos antecedentes son computados basíndose en el tipo de frase que se está observando. Para nombres propios, todas las entidades previas sirven como candidatos. Para pronombres, se miran las entidades que aparecen en oraciones anteriores pero del mismo párrafo. Para definir los sintagmas nominales, se tienen en cuenta todas las entidades que aparezcan tanto en el párrafo actual como el anterior. una

excepción son las entidades de la forma "the X" donde X es el nombre de una compañía, corporación, organización, etc. cuyo alcance abarca todo el texto previo. con el fin de seleccionar el antecedente correcto del conjunto de posibles candidatos, se eliminan en primer lugar aquellos que sean incompatibles con la expresión que se está estudiando (por cuestión de género, número, tipo, etc). De los candidatos que quedan, se selecciona uno acorde a un orden de importancia, cuanto más cerca de la oración actual mejor.

Parece ser que este sistema consigue resultados relativamente buenos logrando encajar correctamente entidades con expresiones que hacen referencia a ellas en el texto en porcentajes alrededor del 80%.

3.5 Métodos de evaluación de los resultados

Cuando se aplica algún tipo de análisis textual sobre una colección de documentos determinada o, más importante, cuando se desarrolla una herramienta nueva, es fundamental saber si los resultados obtenidos son fiables. Dado que es imposible conocer todos los posibles casos con los que se puede encontrar una herramienta de estas características (por ejemplo, todos los posibles artículos que pueden aparecer) y, por lo tanto, evaluar los resultados de manera anticipada no es viable, lo más razonables es medir la efectividad de una determinada herramienta comparándola con otra técnica candidata que haga el mismo tipo de análisis, utilizando en ambos casos el mismo dominio. El dominio consiste en un corpus anotado o etiquetado que está compuesto por elementos textuales. Además de eso, será necesario una medida o métrica para denotar la efectividad del sistema ejecutad sobre ese dominio.

Una buena manera para evaluar tanto sistemas de recuperación de la información como sistemas de extracción de la informacion es midiendo los valores de *recall* y *precisión*. Tenemos un conjunto de N elementos (ya sean terminos, oraciones o documentos) y tenemos un sistema qe esencialmente se encarga de etiquetar positiva o negativamente a dicho eslementos siguiendo algún criterio determinado, por ejemplo si son relevantes para una determinada consulta, o si pertenecen a una categoría de documentos determinada o a alguna clase de término. Mediante este etiquetado, que no es perfecto, se divide el conjunto original en 4 sunconjuntos:

- Verdaderos positivos: A elementos correctamente etiquetados como positivos.
- Falsos positivos: B elementos etiquetados como positivos erróneamente.
- Verdaderos negativos: C elementos etiquetados como negativos correctamente.

- Falsos negativos: D elementos etiquetados como negativos de manera incorrecta

de tal manera que el número de elementos en el conjunto es $N = A + B + C + D$

La Precisión, P, es la proporción de verdaderos positivos con respecto a todos los elementos que el sistema ha considerado positivos, es decir, en recuperación de la información, la proporción existente entre el número de documentos que eran relevantes para la consulta y que ha devuelto el sistema y el número total de documentos que ha devuelto el sistema. El recall, R, es la proporción de verdaderos positivos con respecto a todos los elementos que deberían haber sido etiquetados como positivos, en el caso anterior, la proporción entre los documentos relevantes que ha extraído el sistema y el número total de documentos que deberían haber extraído.

$$P = \frac{A}{A+B} \quad y \quad R = \frac{A}{A+D} \tag{3.18}$$

Por ejemplo, suponemos que tenemos un conjunto de 50 documentos, y deseamos que nuestro sistema marque como positivos aquellos documentos que hablan de expresión gúnica y como falsos todos los demás. Suponemos además que 30 documentos del total tratan la expresión gúnica y que nuestro sistema marca como positivos 40 documentos de los cuales sólo 25 realmente lo son. De esta manera, la precisión calculada P será igual a 25/40 ($P = 0.625$) y el recall será igual a 25/30 ($R = 0.83$).

Una medida que combina los valores de precision y recall es el valor *F-score*, propuesto en **??** que de manera simple se trata de:

$$F = \frac{2PR}{P+R}$$

F es un número entre 0 y 1, pero sólo llega a ser 1 cuando el sistema no produce ni falsos negativos ni falsos positivos. Si se define el valor de F-score de una manera más generl, se puede asignar más o menos peso al valor de precisión o recall.

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

de tal manera que si $\beta = 1$, se le da el mismo peso a la precisión y al recall y F_β produce el mismo resultado que F.

Existe otra medida que evalúa la exactitud del sistema, el ratio entre las respuestas correctas con respecto al número total de respuestas. Usando la

misma notación que en la ecuación (3.18), este valor (*acc*) se calcula de la siguiente manera:

$$acc = \frac{A + C}{A + B + C + D} = \frac{A + C}{N}$$

Cuando se trata de recuperar documentos (ordenados según su grado de relevancia), se puede limitar la medida a los documentos de la lista que están al principio (los más relevantes), calculando la precision y el recall teniendo en cuenta sólo esos documentos. Obviamente, si estudiamos la lista entera el valor de recall será muy alto pero el de precisión será muy bajo. Si embargo, si sólo miramos los documentos que están muy al principio de la lista (estableciendo un umbral para el grado de relevancia muy alto) la precisión aumentará pero el recall será más bajo. Para tener en cuenta esta relación que existe entre el recall y la precisión en función del número de documentos examinados, es común dibujar una curva recall-precisión en función de dicho número. De esta manera, muchos de los sistemas de recuperación son comparados basíndose en sus curvas de recall y precisión.

Obviamente, para poder comparar correctamente el rendimiento de dos herramientas se necesita un corpus de referencia de algún dominio específico donde poder tomar las medidas antes comentadas. En este sentido existen varias colecciones de documentos estandarizadas e incluso varias tareas o problemas concretos de recuperación y extracción, también estandarizados.

Un ejemplo de colección de documentos es el conjunto *Reuters* de artículos clasificados dentro de categorías temáticas ([60]). Esta colección se utiliza mucho a la hora de evaluar sistemas de categorización de textos. Otra colección es el corpus de *Brown* (**??**) de ejemplos de textos americanos, categorizados por tipos de litratura (por ejemplo, prensa, escritos religiosos, narrativa de misterio, etc..). El corpus esta además etiquetado hasta el nivel de las palabras y de varias maneras: Part of Speech (nombre, verbo, etc), función (determinante, preprosición), etc. Este corpus ha sido muy utilizado en módulos de proceso de lenguaje natural, tanto probándolos como entrenándolos.

Un forum donde se evalúan distintos métodos de recuperación es el TREC, conferencia de recuperación de texto *(Text Retrieval Conference)*, ([62]). Se formó en 1992 con el fin de evaluar sisemas de recuperacion a gran escala. Cada año se ofrecen distintos programas, en cada uno se proporciona un conjunto de datos y se proponen una serie de tareas que tienen que ser llevadas a cabo sobre esos datos. Los participantes utilizan su propio sistema y los resultados son evaluados por un jurado. Los campos temáticos que abarca ese concurso son muy variados, en los últimos años ha surgido *TREC Genomics*, cuyo objetivo es recuperar datos genómicos de la literatura.

Otro forum parecido a TREC es el concurso de *BioCreAtIvE* ((Critical

Assessment of Information Extraction systems in Biology)) que celebra el EMBL-EBI (European Molecular Boiology Laboratory-European Bioinformatics Institute) [63]. Se han celebrado dos ediciones de BioCreAtIvE. En la primera se trataban dos temas principales, los dos relacionados con la extracción de información útil y relevante en el campo de la Biología. El primer tema tenía que ver con la detección de entidades biológicas (nombres) como genes o proteínas y relaciones entre ellas. El segundo tema estaba reacionado con la detección de asociaciones de entidades con determinados hechos o eventos (por ejemplo, relacionar una proteína con los términos que describen su función). Para ambas tareas, se proporcionaban dos conjuntos de datos, uno de pruebas y otro de entrenamiento.

Esta primera edición de BioCreAtIvE, celebrada en 2003-2004, atrajo una considerable atención por parte de la comunidad bioinformática y biomédica, llegándose a presentar 27 grupos de 10 pa?ses distintos, y se organizó gracias a la colaboración de grupos de NLP, anotadores de bases de datos biológicas e investigadores bioinformóticos.

La segunda edición de BioCreAtIvE se celebró en 2006-2007, y esta vez se trataron tres problemas principales:

1. *Gene mention tagging*

2. *Gene normalization*

3. Extracción de interacciones proteína-proteína del texto.

La primera tarea trata de encontrar menciones de genes y proteínas en oraciones dentro de los *abstracts* de artículos de MEDLINE. El segundo problema está implicado en la creación de una lista de identificadores *EntrezGene* (que permite el acceso a la base de datos del NCBI, *National Center for Biotechnology Information*) compuesta por todos los genes y proteínas del ser humano mencionados en una colección de *abstracts* también de MEDLINE. El tercer y último tema tratado se centra en la identificación de interacciones proteína - proteína de textos (artículos) enteros, incluyendo la extracción de fragmentos de estos artículos que describen esas interacciones, para anotar los resultados en dos bases de datos de interacciones: IntAct [64] and MINT [65].

Capítulo 4

Minería de Textos en Bioinformática

La secuenciación de el genoma humano marcó el comienzo de la era de la genómica y la proteómica a gran escala. Los experimentos a gran escala estudian la conducta de miles de genes y proteínas. Sin embargo, la interpretación de sus resultados puede llegar a ser un problema. Por ejemplo, muchos de los análisis a gran escala de datos genómicos estudian los patrones de expresión de los genes, y particularmente, tratan de esblacer grupos de genes en base a su nivel de expresión ([75],[55]). Aunque este tipo de herramientas proporcionan una información aproximada de las correlaciones existentes entre grupos de genes en función a su patrón de expresión, tienen una serie de limitacines ([38]). En primer lugar, dos genes pueden funcionar de manera antagónica dentro de un mismo proceso biológico y esto se traduciría en falta de correlación en sus niveles de expresión, a pesar de estar funcionalmente relacionados. Por otra parte, hay genes que a pesar de mostrar patrones de expresión similares, no participan en los mismos procesos biológicos. Además, un mismo gen puede participar en más de un proceso biológico por lo que no debería ser agrupado únicamente dentro de un mismo cluster. y, más importante aún, incluso cuando los patrones de expresión están perfectamente bien diferenciados y relacionados con los distintos grupos de genes que se hayan formado, las relaciones funcionales que existen entre los genes no pueden ser determinadas simplemente por los datos extraídos de los clusters, sino que se precisa levar a cabo mucho más análisis posterior.

La información que se necesita para estos tipos de análisis puede ser la mayor parte de las veces encontrada en la literatura publicada. Sin embargo, no es un tipo de búsqueda que pueda llevar a cabo una persona mirando en la literatura relacionada con un determinado gen o grupo de genes a pequeña escala, sino que al estar implicados miles de genes, se necesita un tipo de

búsqueda Automática que recopile toda la información respecto de ellos, las relaciones existentes entre ellos y el papel que juegan en las reacciones bioquimicas.

La fuente de información on-line más importante se trata de PUBMED [76] una base de datos de literatura biomédica mantenida por el *National Center of Biotechnology Information* (NCBI). Contiene más de 12.000.000 abstracts científicos, y es accedida por millones de usuarios de todo el mundo diariamente. Una búsqueda tipica de documentos relevantes en PUBMED comienza con una consulta de tipo booleano, el usuario proporciona un conjunto de términos y el sistema devuelve todos los documentos que cree que satisfacen la consulta. En PUBMED podemos, sin embargo, encontrarnos con el problema de la sinonimia debido a la falta de uniformidad que siguen los autores al esribir sus artículos. Des esta manera, si uno busca por el gen "AGP1" no va a recuperar todos los abstracts de los documentos que hablen de ese gen, porque en algunos documentos se le referencia mediante un sinónimo (por ejemplo, "YCC5"). Aun así, si el usuario identifica un documento verdaderamente relevante entre los resultados devueltos, PUBMED permite acceder a todos los documentos que están relacionados con éste.

Aunque PUBMED sea un recurso indispensable hoy en día, está claro que un metodo de búsqueda que vaya de gen a gen no es viable cuando hablamos de hacer una minería de la literatura a gran escala. Para mejorar la efectividad y eficiencia del estudio de la literatura, se han desarrollado muchos métodos que procesan la literatura automáticamente. Se puede distinguir entre dos tipos de herramientas en este sentido, las basadas en extracciónd e la información y NLP y las que se basan en recuperación de la información.

4.1 Extracción de la información en Bioinformática

Muchos de los esfuerzos centrados en la minería de la literatura biomedica hasta la fecha han sido destinados a la extracción de la información Automática, empleando mecanismos de NLP para identificr entidades, expresiones o hechos relevantes en el texto. Sin entrar demasiado en detalle, merece la pena mencionar las principales fuentes de información de términos relacionados con los genes. Por una parte tenemos las bases de datos de genomas y proteomas de distintos organismos como LocusLink ([77]), SwissProt ([78]) y HUGO ([79]), que contienen muchos de los nombres y sinónimos de los distintos genes conocidos. Por otra parte, tenemos vocabularios controlados de términos biomédicos como el *National Library of Medicine's MeSH (Medical*

Subject Heading) ([80]) y *UMLS (Unified Medical Language System)*([81]). La ontología más importante provista de un vocabulario controlado del papel biológico, quimico y celular de los genes y del producto de los genes (ARN, proteínas, etc) es *Gene Ontology, GO* ([1]), de la que hablaremos más adelante.

En [82], uno de los trabajos más antiguos en este dominio, usa modelos ocultos de Markov (HHMs) para extraer oraciones que hablaran de la localización de los genes en los cromosomas (los HMMS son usados frecuentemente por las técnicas NLP para representar la estructura de una oración). En el caso de las oraciones que describían la localización de los genes en los cromosomas, estaban compuestas por el nombre de los genes y los cromosomas, palabras que describian la localización, y términos que denotaban los métodos experimentales que validaban la localización del gen en el cromosoma. Los nombres de genes y cromosomas se identificaban por heurísticas simples (por ejemplo, términos que tengan todas las letras mayúsculas con algún número son vistos como genes), y los métodos experimentales se identifican comparándolos con los que aparecen en una lista predefinida. Los conjuntos de entrenamiento y test consistían en cientos de oraciones. Los resultados se evaluaban en función de los valores obtenidos de recall y precisión, en conjuntos relativamente pequeños de test se obtenía un ratio de éxito de 0.6 en el punto donde recall y precisión coincidían.

Craven et. al. ([83], [84]) han extendido esta linea de trabajo, desarrollando sistemas que distinguen oraciones que contienen hechos relevantes de aquellas que no. Los sistemas fueron diseñados para identificar dos tipos de hechos: localización subcelular de las proteínas y asociación entre genes y enfermedades. El primer trabajo ([83]) consistía en clasificadores que aprendían, con o sin uso de reglas gramaticales, a reconocer oraciones que discutían acerca de la localización de las proteínas dentro de la célula. Usando la lexica predefinida de localizaciones y proteínas y varios cientos de oraciones de entrenamientos derivadas de YPD (Yeast Proteome Database), se entrenaban los clasificadores y se probaban después con un corpus formado por unos 3000 abstracts de PUBMED. En el test se intentaba evauar la capacidad del sistema para ditinguir correctamente las oraciones que hablaban de la localización de las proteínas, más que extraer cual era esa localización exactamente. Sin utilizar reglas basadas en gramatica, la mejor precisión obtenida era del 77% con un recall del 30%. Usando reglas gramaticales y parseando las oraciones, se obtenía una precisión del 92% pero un recall del 21%. El segundo método decide que una oración proporciona una localización celular si aparece el nombre de una proteína y de una localización dentro de la frase. Este método tan simple, que es actualmente de los más populares en el contexto de la literatura Bioinformática, obtenía una precisión más baja que el

sistema anterior basado en clasificadores (cerca de un 35% de precisión con un recall de 30% y un 45% de precisión con un recall de 21%). El método basado en la ocurrencia conjunta en la misma oración puede alcanzar mejores resultados de recall (˜70%) sin perder mucha precisión (˜40%). Sin embargo, con el mismo nivel de recall, un sistema basado en clasificadores también puede alcanzar el mismo nivel, o incluso algo mayor, de precision (˜45 - 50%). El estudio sugiere que los clasificadores a nivel de oración puede mejorar la precisión con respecto a los métodos que se basan en la co-ocurrencia en la misma frase, siempre hablando en el contexto biomédico.

Este trabajo fue extendido más ([84]), utilizando HMMs para representar la estructura de la oración e identificando las frases que hablaban de las relaciones entre genes y enfermedades. En este caso, se utilizaron varios cientos de oraciones preetiquetadas como ejemplos positivos y miles de oraciones como ejemplos negativos para aprender los modelos ocultos de Markov. La identificación correcta de oraciones que hicieran referencia explícita a genes y proteínas se limitaba a aquellas contuvieran los nombres previamente usados en los ejemplos de entrenamiento.

Una aproximación simple que se basaba en la coocurrencia de genes/proteínas dentro de una misma oración, sin utilizar ningún mecanismo avanzado de Aprendizaje Automático o NLP, fue la usada por Blaschke et. al [85]. Su objetivo era extraer información acerca de interacciones entre proteínas de un conjunto predefinido de proteínas relacionadas. Usando una lista de nombres de proteínas y una lista de palabras que indicaran alguna interacción de algún tipo, se buscaban frases donde aparecieran los nombres de dos proteínas separadas por una de esas palabras, para identificar el tipo de relación entre las proteínas. Una extensión de este trabajo es descrita en [86], donde se usa un módulo de detección de nombres de proteínas y se excluyen las negaciones, es decir que sólo se recupera información de oraciones que hablan afirmativmente de una interacción.

La exclusión de la negación es un punto interesante y merece ser discutido. Si tenemos por ejemplo una oración como "We have found *no evidence* that a protein A is involved in the regulation of gene B", si nuestro sistema está extrayendo rutas reguladoras Automáticamente de la literatura, no debería relacionar nunca la proteina A con la proteína B. Sin embargo, en un escenario diferente, es posible que la información negativa pueda ser útil, si por ejemplo finalmente somos capaces de establecer una relación entre las proteínas A y B mediante un método experimental, gracias a ese documento podremos saber que hemos hecho un descubrimiento relevante. De esta manera la omisión de la información negativa debe ser considerada según el caso.

El trabajo de Jensen et. al [87] fue llevado a cabo más a gran escala. Usando una lista predefinida de nombres de genes y símbolos, se ejecutó

una búsqueda booleana sobre PUBMED, encontrando todos los abstracts que mencionaran a esos genes. Entonces se construyó un grafo con un nodo por cada gen y arcos conectando a aquellos genes que fuesen nombrados en el mismo abstract. El peso asociado a cada arco consistía en el número de coocurrencias. El resultado fue una red a gran escala de genes interrelacionados por la literatura donde los abstracts justificaban cada uno de los arcos. Esta red consistía en una herramienta sin precedentes para los investigadores.

Han aparecido muchos otros sistemas basados en coocurrencia, todos referidos a la extracción de la información de textos bomédicos de hechos acerca de entidades biológicas. Todos tienen en común que intentan identificar coocurrecias de nombres o identificadores de entidades, comúnmente junto con términos de dependencia o de activación. Las diferencias entre los distintos sistemas suelen radicar en la extensión del uso que le den a los métodos de análisis sintáctico y métodos NLP, y a los vocabularios o tesauros que utilicen ([26],[88], [89], [90])

Sin embargo, todos los métodos citados antes tienen diversas limitaciones. Por una parte casi todos necesitan que en las consultas deban ser puestos los nombres de los genes o proteínas explícitamente si se quieren obtener buenos resultados. Por otra parte, y esto es más importante, todos hacen suposiciones acerca del uso dellenguaje natural, como qué terminos implican necesariamente relación, la estructura típica de una oración, los nombres de proteínas y genes y su formato y la manera en la que esos nombres son usados dentro de las oraciones. Evidentemente dichas suposiciones simplifican mucho el tipo de lenguaje que se puede encontrar a lo largo de toda la literatura y limita la eficacia de estos métodos.

Además, estos métodos se basan en la coocurrencia de genes o proteínas dentro de abstracts publicados, es decir, que no van a revelar relaciones que no haya sido ya publicads en la literatura aunque se puede hacer un matiz. Y es que por ejemplo se puede seguir la metodología de Swanson ([91], [92], [93]) y usar las relaciones transitivas para detectar nuevas relaciones. Esto sí, si en la literatura aparecen relacionados el gen A con el gen B y el gen B con el gen C, un sistema de estas características debería poder inferir una relación entre el gen A y el gen C.

También cabe reseñar que, aunque la mayoría de los trabajos realizados al respecto confían en la aparicion de nombres de genes o proteínas en el texto, y esos nombres son extraídos previamente de bases de datos públicas, sí que existen también trabajos de detección Automática de nombres de genes o proteínas en textos ([94], [95]).

Por último, se ha visto el esfuerzo puesto en los métodos de extracción de la información y NLP en los trabajos desarrollados, aunque estos métodos dependen en gran medida de información predefinida, que por regla general es

difícil de obtener. Es necesario un sistema que relaje esos requerimientos. Una alternativa o complemento a ese análisis tan exhaustivo y a bajo nivel como es la búsqueda de nombres o sinónimos dentro de los textos, es mediante la recuperación de abstracts más relevantes. En este sentido, la recuperación de información, que trabaja a mayor nivel al tratar con documentos y abstracts tiene mucho que ofrecer.

4.2 Recuperación de la información en Bioinformática

La maera más común y simple de recuperación de la información ya se usa de manera regular por todos los investigadores a la hora de buscar artículos. Como se comentóal principio de la sección, PUBMED permite tanto consultas de tipo booleano como consultas basadas en la similitud (aunque de una manera limitada). Aunque PUBMED es una herramienta efectiva para recuperar artículos de interés (bien etiquetados), no se puede pretender usar el mismo sistema para recuperar o explicar relaciones entre genes y entidades biológicas a gran escala. Sin embargo, sí que se han desarrollado varios métodos para llevar a cabo esto mismo.

En Shakay et. al ([38],[52]) se trataba de encontrar relaciones funcionales entre genes, sin que imporatara demasiado la nomenclatura de los genes o de la estructura de las oraciones. El trabajo se basa en la hipótesis de que muchos genes individuales y sus funciones aparecen ya en la literatura. Se usaron decenas de miles de abstracts extraídos de PUBMED del dominio que se estuviera tratando (por ejemplo, todos los abstracts que tuvieran relación con los genes de la levadura). Para encontrar relaciones entre grandes conjuntos de genes, se buscaba para cada gen un abstract que hablara de su función biológica. Este abstract era tratado como el representante del gen y se le daba el nombre de *kernel abstract* para ese gen.

Entonces se aplicaba un algoritmo probabilístico ([38]). Dicho algoritmo, dado un documento de ejemplo, encuentra un conjunto de documentos más relevantes para él y produce un conjunto de términos resumiendo el contenido de dicho conjunto. Aplicando este algoritmo a cada kernel, se produc?a para cada gen un cuerpo de literatura relacionada junto con un conjunto de términos que caracterizaba a dicha literatura relacionada, siempre basíndose en la información contenida en el kernel de cada gen. Una vez hecho esto, se aplicaba un algoritmo que comparaba los conjuntos de abstracts y extra?a relaciones funcionales entre los genes.

Otros grupos han aplicado métodos de clustering y clasificación para re-

cuperación de la información. En [96] se sugería un método de clustering de anotaciones de proteínas. La idea básica era que mediante el clustering de proteínas dentro de grupos, uno podía inferir la función común que las proteínas podráan tener. El método se basaba en agrupar en primer lugar los términos que aparecían en las anotaciones de las proteínas dentro de conjuntos, de acuerdo a su tendencia a coocurrir. Se utilizaba entonces una medida de similitud que se basaba en la proporción de los términos que tenían en común unos grupos con otros.

En [97] se aplicaba un clustering de k-means sobre un conjunto de abstracts e PUBMED relativamente pequeño (alrededor de 2000 documentos) con el fin de encontrar subconjuntos significativos donde cada uno tratara un asunto determinado. Cada uno de esos asuntos era entonces representado por los términos extraídos mediante un análisis estadístico de las frecuencias de los términos dentro de los clusters formados. En [98] se aplicaba un clasificador de Bayes que se basaba en la discriminación de términos para identificar abstracts que discutieran acerca de interacciones entre proteínas.

El trabajo presentado en [26] representaba las proteínas a través de los abstracts que las mencionaran. Se utilizaba entonces el algoritmo SVM (Support vector Machine) para llevar a cabo una clasificación distinguiendo los abstracts que hablaran de unas u otras proteínas, basíndose en las diferentes localizaciones celulares de las proteínas mencionadas en el texto. Su propósito era el de determinar el orgánilo donde se ubica cada proteína dentro de la célula.

Stephens et. al. ([99]) deduce relaciones entre genes basíndose en la coocurrencia de sus nombres (donde los nombres son dados por un tesauro) pero mediante métodos de information retrieval. Se representaba a los documentos como vectores con pesos, donde los términos eran los genes mencionados en el texto. Mirando la matriz traspuesta, cada gen es entonces visto como un vector cuyos elementos son los documentos que le mencionan. La asociación entre dos genes era entonces calculada mediante el producto escalar de los vectores que les representaban. De esta manera se cuantificaban ls coocurrencias de los genes dentro de los coumentos de manera efectiva.

En [100] se propuso el sistema PreBind/Textomy en el que se combinan técnicas de recuperación de la información y extracción de la información para recuperar interacciones entre proteínas de la literatura. En la fase de recuperación de la información, se entrenaba un clasificador SVM para distinguir entre los abstracts de PUBMED que hablaban de interacciones de proteínas y aquellos que no lo hacen. El clasificador se usaba para recuperar los abstracts relevantes respecto a las interacciones de las proteínas y una vez hecho esto, se aplicaba técnicas de extracción de la información para buscar la información concreta en los textos. Se usaba entonces SVM para recuperar

aquellas oraciones donde se encontrara la información de las interacciones. Se buscaba el nombre de las proteínas en cada una de esas oraciones (los nombres estaban contenidos en una lista de nombres y sinónimos).

Un trabajo más reciente es el propuesto por Chagoyen et. al [101]. En él se presenta un método para crear *perfiles literarios* de grandes grupos de genes o proteínas basíndose en la semántica común extraída de un gran corpus de documentos relevantes. Para conseguirlo proponen usar un método de análisis, *non-negative matrix factorization (NMF)*, introducido en [102] en un contexto distinto, pero usado después en análisis de expresión génica ([103],[104]], secuenciación de datos ([105]) y anotaciones funcionales de genes ([106]). La idea es crear por cada gen un *documento* concatenando todos los términos de títulos y abstracts relevantes para dicho término. Se representa cada uno de estos documentos artificiales en el espacio vectorial, mediante vectores de términos con pesos asociados y obtenemos una matriz V de genes (documentos de genes) por términos. Se aplica el algoritmo de NMF sobre dicha matriz. Formalmente, la factorización no negativa de matrices (NMF) se describe como sigue:

$$V \approx WH$$

donde V es una matriz positiva de pxn elementos, W es una matriz positiva de compuesta por k *vectores básicos* o *factores* y H es una matriz de k x n elementos, que contiene los coeficientes de la combinación lineal de los vectores básicos para reconstruir la matriz original, $k \leq p$ y adicionalmente las columnas de W están normalizadas (suman 1). Evidentemente, la elección de una k correcta es un asunto crítico en este método.

Para la aplicación descrita, se demuestra que cada columna de W es representada por un conjunto pequeño de términos, que de alguna manera identifica a cada uno de los grupos formados mediante NMF (un grupo de términos relacionados semánticamente que representan un determinado perfil literario). Por otra parte, el análisis de los vectores de H proporciona información acerca de cómo la combinación de esos perfiles literarios describe semánticamente a cada gen o proteína. De esta manera, dado un gran grupo de genes o proteínas, podemos extraer informacción semántica o latente que estuviese contenida en la literatura biomédica relevante.

Capítulo 5

Gene Ontology

Gene Ontology ([1]) es un proyecto que se gestó a partir de la idea de que todos los organismos eukaryotas compartían un elevado porcentaje de genes y proteínas. De esta manera, se pensó que toda la información acerca de dichos genes y proteínas ayudaráan a entender el comportamiento de todos los organismos que los comparten. Por otra parte, la existencia de muchos sistemas diversificados para nombrar tanto a los genes como a sus productos y la falta de un estándar impedían la interoperabilidad entre las distintas bases de datos, lo que de alguna manera obstaculizaba el desarrollo o progreso de la Bioinformática.

La propuesta del *GO Consortium* consistía en producir un vocabulario controlado, estructurado bien definido y común que describiera el papel de los genes y sus productos dentro de cualquier organismo ([1]) y se crearon tres ontologías independientes, accesibles a través de internet: procesos biológicos, funciones moleculares y componentes celulares.

Cada nodo de las ontologías GO sería enlazado por otros tipos de bases de datos de genes y proteínas como SwissPROT, GeneBank, EMBL, PDB, NCBI, etc. Una razón para esto es que el conocimiento biológico que se tiene de los genes y proteínas cambia rápidamente y todo los descubrimientos necesarios para entender el papel y funcionamiento de los genes y las proteínas se publican en este tipo de bases de datos.

Por otra parte, el conocimiento que se tiene de unos genes o proteínas y otros es muy distinto en cuanto a profundidad. De esta manera era necesario organizar, describir y visualizar la información en estos diferentes niveles de conocimiento. Cualquier sistema debe ser además flexible y tolerante a los continuos cambios y actualizaciones de la información.

La ventaja de usar ontologías es que son capaces de representar las distintas entidades que aparecen dentro de un determinado área, además de las relaciones existentes entre ellas. Precisamente, una ontología se trata de

un conjunto de términos y de relaciones definidas entre esos términos. La estructura en sí representa el conocimiento biológico actual y a la vez permite organizar los nuevos conocimientos que se vayan adquiriendo. Los datos pueden ser anotados en diferentes niveles de la jerarquéa en función de su grado de profundidad. Por último, permite a los investigadores acceder de manera fácil a la información y ser una fuente de información útil a la hora de desarrollar herramientas Bioinformáticas.

Las tres categorías de GO son procesos biológicos, funciones moléculares y componentes celulares. En [1] explican el significado de cada categoría: por proceso biológico se entiende el objetivo biológico en el que contribuye un gen o un producto genético. Cada proceso es una ruta compleja en la que intervienen una o más funciones moleculares. Una función molecular es definida como una actividad bioquémica de un producto genético, describiendo sólo que es lo que ocurre, sin especificar donde o cuando ocurre. Por componente celular se entiende el lugar de la célula (eukaryota) donde un producto genético es activo.

Procesos biológicos, funciones moleculares y componentes celulares son todos atributos de genes, productos genéticos o grupos de productos genéticos, fácilmente reconocibles e independientes entre sí. Las relaciones entre un gen, producto genético o grupo de éstos con procesos biológicos, funciones moleculares y componentes celulares es *uno a muchos*, reflejando la realidad biológica de que una misma proteína puede verse involucrada en más de un proceso.

Por ejemplo, en la figura 5.1 podemos observar tres ejemplos del tipo de estructura utilizada en Gene Ontology para representar y asociar la información y los genes. Las ontologías están construidas en base a un vocabulario controlado. Por simplicidad, no se incluyen todos los genes en la figura. La figura 5.1.a, muestra una porción de la ontología de procesos biológicos que describe describing el metaboilismo del ADN (*DNA metabolism*). Se puede observar que un mismo nodo puede tener más de un padre, por ejemplo "DNA ligation" tiene tres padres: "DNA-dependent DNA replication", "DNA repair" y "DNA recombination". La figura 5.1.b, muestra un estracto de la ontología de funciones moleculares. Esta ontología no está pensada para representar la ruta de una reacción, sino reflejar las categorías conceptuales de las funciones de los productos genéticos. Un producto genético puede asociarse con más de un nodo de la ontología, como ilustran las proteínas MCM. Se sabe que estas proteínas están relacionadas con varias funciones moleculares y por lo tanto aparecen asociadas a varios nodos. La figura 5.1.c muestra la ontología de componentes celulares. Las ontologías han sido concebidas para una célula eukaryota genérica y son lo suficientemente flexibles como para representar las diferencias entre los distintos organismos.

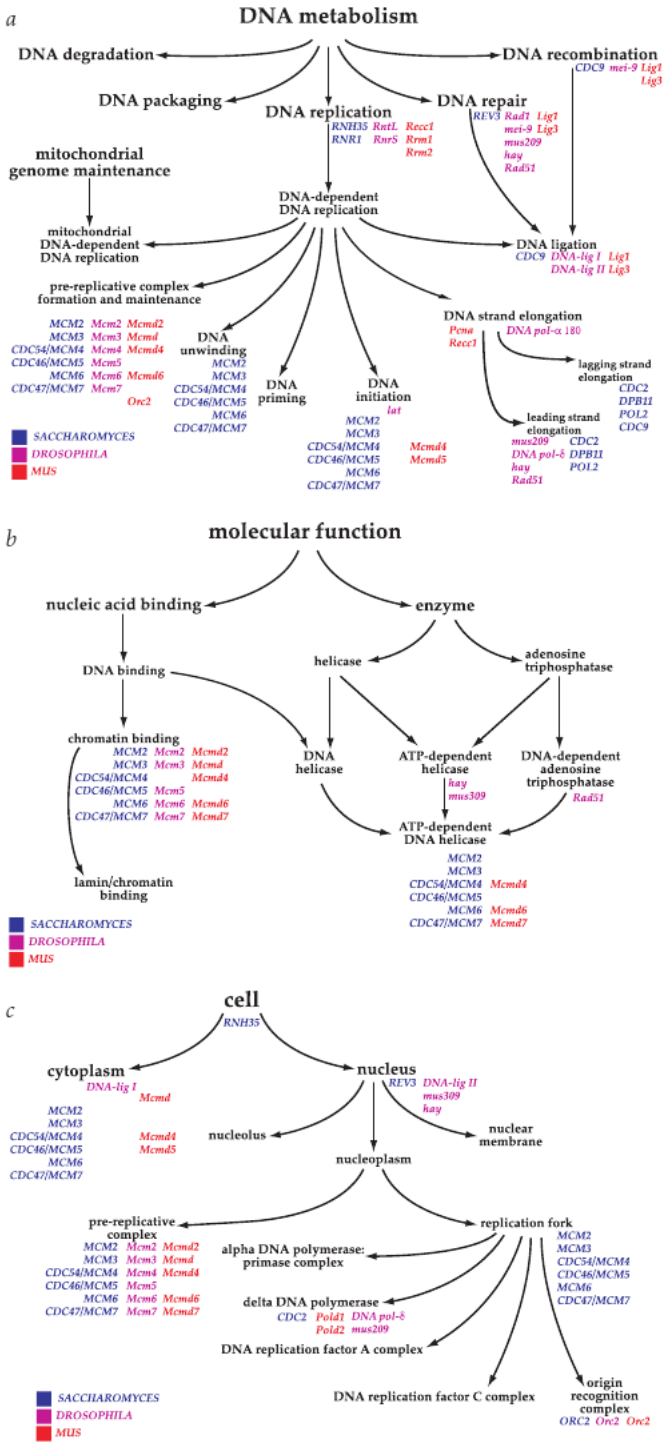


Figura 5.1: Ejemplos de Gene Ontology. Imágen extraída de [1]

Asociada a GO se encuentra GOA (Gene Ontology Annotations), una base de datos que relaciona el genoma de determinados organismos con términos que aparecen en GO. Además de establecer dicha relación (genes - GOterms), proporcionan la publicación biomédica que recoge la evidencia. El método de extracción de información e incorporacón en la base de datos es totalmente manual, existe un cuerpo de anotadores encargados de leer todas las publicaciones biomédicas, concluir las relaciones entre genes y términos de GO e introducir la información en la base de datos de GOA. Esto, debido a la creciente acumulación de información biomédica ya comentada, hace que sea una tarea cada vez más ardua y que sea necesario el desarrollo de herramientas que automaticen el proceso.

A pesar de todo, Gene Ontology es hoy por hoy una de las principales fuentes de información biológicas y una herramienta indispensable para los investigadores.

Capítulo 6

Objetivos

El objetivo general de este trabajo se centra en el estudio, desarrollo y aplicación de nuevas metodologías para el análisis de datos biológicos a través de la literatura biomédica. Se han abordado los métodos clásicos de Minería de Datos, Procesamiento de Textos, Extracción de la Información y Recuperación de la Información y se han estudiado las distintas alternativas propuestas en distintos ámbitos de la Bioinformática y las bases de datos y recursos disponibles en la web.

En concreto, los objetivos desglosados son:

1. Desarrollo y evaluación de una metodología basada en un método usado en extracción de información biológica a partir de grandes listas de genes resultantes del análisis de estos experimentos, pero en el ámbito de la literatura biomédica, con dos claros propósitos:

    - Permitir, por una parte, establecer relaciones entre genes o proteínas e información biológica como anotaciones funcionales o reguladores transcripcionales.
    - Por otra parte, se puede realizar el proceso de categorización de documentos, llevado a cabo actualmente por los anotadores de las bases de datos, automáticamente.

2. Desarrollo de una herramienta gratuita accesible a través de la web que implemente dicho método y que sea de utilidad para la comunidad científica.

Capítulo 7

Materiales y métodos

En esta sección se detallan los métodos y algoritmos que han sido propuestos para el análisis y extracción de información biológica a partir literatura biomédica e información biológica de distintas bases de datos. Dichos métodos y algoritmos se basan en los trabajos propuestos durante los últimos años centrados en el análisis funcional de genes, como [66], [67], [68], [69] y [70].

En primer lugar, en la sección 8.5 se describen las metodologías propuestas para la extracción de información biológica a partir de la literatura biomédica basado en la extracción de reglas asociativas. Esta metodología permite integrar datos obtenidos de la literatura con otras fuentes de información como anotaciones funcionales o reguladores transcripcionales y es de gran utilidad para el descubrimiento de asociaciones entre información biológica de los genes y proteínas y documentos o conjuntos de palabras.

La creación de bases de datos, a partir de la extracción de reglas asociativas que relacionan términos con anotaciones, son el medio utilizado para recuperar posteriormente las anotaciones enriquecidas en un determinado conjunto de palabras.

En segundo lugar, en la sección 7.2 se explican los distintos tipos de análisis estadísticos implementados para evaluar estadísticamente las anotaciones concurrentes procedentes de distintas bases de datos recuperadas: el test de la distribución hipergeométrica y el test de χ^2.

Por último, en la sección 7.3 se presenta el problema de las comparaciones multiples que aparece cuando se baraja un elevado número de hipótesis (como es el caso, donde el número de anotaciones evaluadas puede llegar a ser muy grande) y los distintos métodos propuestos para corregir los p-valores calculados y obtener así datos más fiables.

7.1 Uso del análisis del enriquecimiento para el análisis integrado de datos

La extracción de reglas asociativas (ARD) es una técnica de minería de datos, propuesta originalmente por Agrawal et al. [71], que ha sido ampliamente utilizada para encontrar asociaciones o relaciones entre conjuntos de elementos presentes en una base de datos de transacciones. Este método extrae conjuntos de elementos que ocurren frecuentemente en la misma transacción, y formula reglas que caracterizan esas relaciones. Esta técnica se ha utilizado tradicionalmente en el análisis de matrices de expresión con el objetivo de extraer relaciones entre genes en base a sus patrones de expresión génica. En este trabajo se ha desarrollado una novedosa aplicación de esta técnica para el análisis de literatura biomédica capaz de integrar y extraer asociaciones entre términos de documentos científicos y características biológicas de los genes (categorías de Gene Ontology).

7.1.1 Definición de reglas asociativas

La definición formal de una regla asociativa se puede expresar como:

Sea $I = \{i_1, i_2, i_3, \cdots, i_n\}$ un conjunto de n elementos en una base de datos S. Una transacción T, perteneciente a dicha base de datos, está compuesta de un conjunto de elementos que satisface $T \subseteq I$, es decir, es una subconjunto de elementos de I. Se puede decir que una transacción T contiene un conjunto de elementos X en I si $X \subseteq T$. Una regla asociativa es una expresión de la forma $\{X \rightarrow Y\}$, donde $X \subseteq I$, $Y \subseteq I$ y $X \cap Y = \emptyset$. La parte izquierda de la regla se denomina *antecedente* y la parte derecha *consecuente*. Estas reglas se interpretan de la siguiente forma: cuando ocurre X es probable que también ocurra Y en la misma transacción.

Dada una regla asociativa , hay dos medidas que definen la calidad de la regla;

- Su soporte, el cual es definido como $P(X \cup Y)$, o sea, la probabilidad de que X e Y aparezcan juntos.
- Su confianza, que se define por la probabilidad condicional de que ocurra Y dado X, y se expresa como $P(Y|X) = \frac{P(X \cup Y)}{P(X)}$.

El soporte y la confianza son las medidas más comunes y, en muchos casos, las únicas utilizadas para cuantificar la relevancia de este tipo de asociaciones. Sin embargo, el uso de estas dos medidas puede conllevar que en ciertos casos se generen reglas que en principio pueden parecer significativas

por presentar altos valores de soporte y confianza, pero sin embargo reflejan asociaciones entre conjuntos de elementos no correlacionados. Esto pasa cuando los elementos del consecuente son muy frecuentes en la base de datos. Por ejemplo, imaginemos la asociación $\{A \cdot B, C\}$ en la que el valor del soporte sea del 70% y el de la confianza del 80%. Esta regla indica que el 70% de todas las transacciones contienen los elementos A,B y C y que el 80% de las veces ocurre A también ocurren B y C. Aunque esta regla parece que ofrece una fuerte correlación entre los elementos A con B y C, esto no es necesariamente cierto si B y C están presentes en el 100% de las transacciones. Se necesita por lo tanto una medida de correlación entre el antecedente y el consecuente que evalúe fielmente la calidad de una regla asociativa. Esta medida es:

- La mejora de la regla, la cual es definida como $\frac{P(X \cup Y)}{P(X) \times P(Y)}$, esto es, la confianza de la regla dividida por el soporte del consecuente.

Cualquier regla con un valor de mejora menor que 1 indica que no hay una correlación real entre el antecedente y el consecuente y, por el contrario, reglas con valores mayores que uno reflejan reglas con mejores propiedades para predecir el consecuente. Esta metodología ha sido muy utilizada para la búsqueda de patrones entre artículos de venta en transacciones comerciales, lo que se ha venido a denominar análisis de la "cesta de la compra". Las reglas extraídas en este contexto tienen como objetivo descubrir hábitos de compra de los consumidores, lo cual tiene una aplicación directa en estrategias de marketing tales como la disposición y ubicación de los productos en unos grandes almacenes, diseño de catálogos o publicidad personalizada.

Un ejemplo sencillo que ilustra este tipo de patrones es el siguiente: Imaginemos una base de datos de transacciones en la que cada transacción representa un cliente y los productos comprados por cada cliente el conjunto de elementos (tabla 7.1). Una regla asociativa que se podría extraer de esta base de transacciones es: $\{manzanas \rightarrow peras, naranjas\}$, con un soporte del 40% y una confianza del 50%. Esta regla indicaría que el 50% de las personas que compran manzanas también compran peras y naranjas y la compra de estos tres productos ocurre en el 40% de las transacciones.

Esta metodología puede extenderse a cualquier tipo datos en los que interese extraer este tipo de asociaciones. El requisito es poder estructurar la base de datos en forma de transacciones que reflejen la presencia o ausencia de cada uno de los elementos de la misma. En el campo de la bioinformática este método se ha usado en numerosos contextos, como por ejemplo la extracción de asociaciones entre motivos de secuencia en promotores , entre características estructurales y funcionales de proteínas que interaccionan entre sí, entre elementos de secuencia y función biológica, entre motivos de

Table 7.1: Ejemplo de una base de datos de transacciones comerciales

Transacción (clientes)	**Elementos (productos)**
Transacción 1	Pan, queso, manzanas, refrescos
Transacción 2	Pan, manzanas, peras, naranjas
Transacción 3	Pan, leche, manzanas, peras
Transacción 4	Leche, peras, naranjas
Transacción 5	Manzanas, peras, azúcar, naranjas

InterPro y clases enzimáticas o entre conjuntos de genes en base a datos de expresión.

7.1.2 Bases de datos de transacciones a partir de literatura biomédica y Gene Ontology Annotations

Del procesado de la literatura biomédica y la base de datos de Gene Ontology Annotations se puede extraer una base de datos de transacciones. Para poder encontrar relaciones entre términos y categorías GO, el sistema propuesto bebe de dos fuentes distintas. Por una parte, se busca información en la misma base de datos de Gene Ontology. Cada una de las entradas de GO tiene una serie de campos asociados que proporcionan información de la categoría, nosotros extraemos su nombre, su definición y sus sinónimos, resaltadas en al figura 7.1.

Los términos de estos tres campos son entonces procesados, salvo los términos de parada o stopwords, que no son tenidos en cuenta. En primer lugar se les aplica el algoritmo de stemming de Porter con el fin de trabajar sólo con la raíz de cada palabra, y que las distintas formas verbales, los plurales y otras declinaciones lingüísticas no sean problema para reconocer los términos. De esta manera, creamos un vector por cada categoría GO donde almacenamos todos las raíces de los términos relacionados junto con las respectivas frecuencias con las que aparecen en total en los campos parseados (ver esquema en figura 7.2).

Para saber qué artículos están relacionados con cada una de los nodos de GO, recurrimos a la base de datos de Gene Ontology Annotations. Como se ha explicado en la sección anterior, las anotaciones publicadas en esta base de datos establecen asociaciones entre genes y entradas de Gene Ontology, proporcionando además (entre otro tipo de información) los artículos publicados en la literatura que presentan la evidencia de dicha relación. De esta manera, podemos establecer tuplas de tres componentes *gen - entrada GO -*

Name
anaphase-promoting complex-dependent proteasomal ubiquitin-dependent protein catabolic process
Accession
GO:0031145
Ontology
biological process
Synonyms
exact: anaphase-promoting complex-dependent proteasomal ubiquitin-dependent protein breakdown
exact: anaphase-promoting complex-dependent proteasomal ubiquitin-dependent protein catabolism
exact: anaphase-promoting complex-dependent proteasomal ubiquitin-dependent protein degradation
exact: APC-dependent proteasomal ubiquitin-dependent protein catabolic process
exact: APC-dependent proteasomal ubiquitin-dependent protein catabolism
Definition
The chemical reactions and pathways resulting in the breakdown of a protein or peptide by hydrolysis of its peptide bonds, initiated by the covalent attachment of ubiquitin, with ubiquitin-protein ligation catalyzed by the anaphase-promoting complex, and mediated by the proteasome. [source: PMID:15380083]
Comment
None

Figura 7.1: Ejemplo de información contenida en Gene Ontology por cada anotación

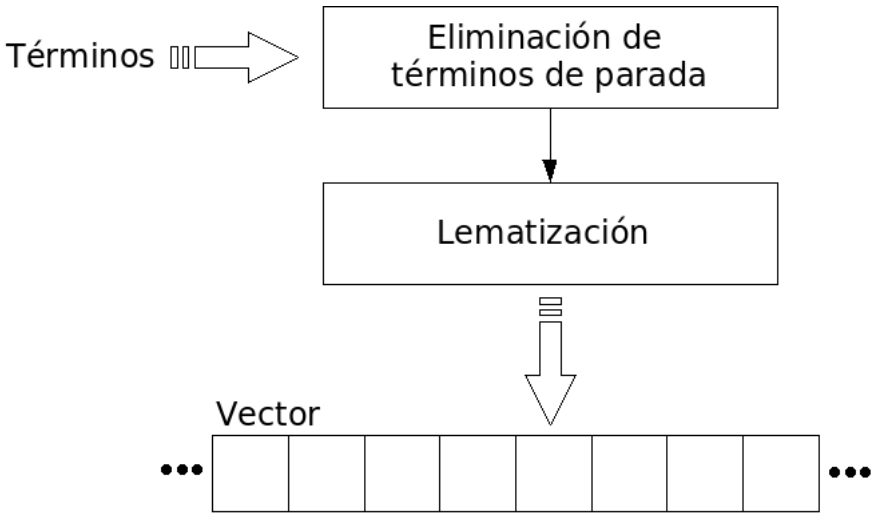


Figura 7.2: Esquema del procesado del texto

artículo.

Aunque es muy interesante contar con la información que relaciona los genes con las entradas de GO o con los artículos y, siguiendo la misma metodología que la propuesta se podría hacer un análisis análogo al explicado en este trabajo, la información que extraemos es la concerniente a la relación establecida entre las entradas de GO y los artículos publicados. Evidentemente, si una determinada publicación evidencia una asociación entre un gen y una determinada categoría de GO, se puede concluir que dicha publicación habla tanto de ese gen como de esa categoría de GO y podemos relacionar los términos contenidos en el texto tanto con uno como con otro.

De esta manera, se extrae del fichero de anotaciones (único para cada organismo) los artículos relacionados con cada una de las categorías de GO. El siguiente paso es acudir a PUBMED, de donde se extrae tanto el abstract como el título de cada publicación. Se extraen todos los términos y aquellos que no estén incluídos dentro de una lista de stopwords se procesan de la misma manera que se ha comentado anteriormente. Por último, con estas palabras, se completan los vectores de términos creados para cada categoría de GO. Cada uno de estos términos tiene una frecuencia asociada, que indica el número de veces que aparece relacionada la palabra en cuestión con la categoría de GO, sea cual sea la fuente de información de donde haya sido extraído. De esta manera, tenemos cada una de las categorías de Gene Ontology definidas con un conjunto de palabras (un metadocumento), extraídas tanto de la propia base de datos de Gene Ontology como de los artículos relacionados en PUBMED. En la imágen 7.3 se puede ver un ejemplo del metadocumento creado para la categoría *GO:0031145*

El siguiente paso consiste en el filtrado de esas palabras, ya que no todas son igual de relevantes ni aportan la misma información. Un método muy común es eliminar las palabras que aparecen de manera recurrente, en este caso, para todas las categorías de GO, ya que no aportan demasiada información ni sirven para distinguir unas categorías de otras. Para ello, aquellos términos que aparecen en más del 80% de las categorías estudiadas no son tenidos en cuenta. Por ejemplo, al trabajar con artículos relacionados con la levadura, el término "yeast", que no tiene porque ser considerado como un témino de parada, aparecerá en la gran mayoría de los artículos sino en todos, considerarla sólo implicaría más trabajo de computación en la siguiente fase además de añadir ruido y emperorar los resultados.

Por otra parte, también es común eliminar las palabras que no aparecen en al menos un determinado umbral de documentos (se suele usar el 20% como umbral). De esta manera sabremos a ciencia cierta que no estamos tomando palabras poco relevantes (o excepcionales). Aunque este filtro puede traer controversia, dado que según el tipo de análisis que se esté realizando

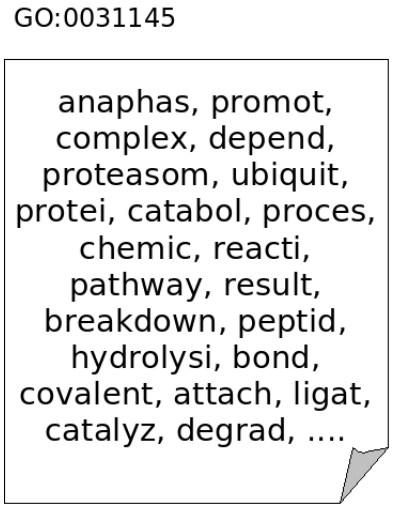


Figura 7.3: Ejemplo de metadocumento para una categoría de GO

podemos estar eliminando información realmente importante, en este caso parecía preferible hacerlo. Por una parte, es una medida de reducción la potencial magnitud del corpus (facilitando así el cómputo posterior) y por otra parte, dado que la intención del proceso es la de caracterizar una entidad (una categoría GO), no sería buena idea hacerlo utilizando palabras, como se ha dicho antes poco relevantes o incluso excepcionales, que quizá sólo sirven para añadir ruido.

Tenemos cada documento representado por un vector de pesos siguiendo el modelo de espacio vectorial clásico en sistemas de extracción de información a partir de textos: cada metadocumento $D_i \in i^p$, donde p es el número total de términos en el vocabulario del corpus entero, es representado como un vector numérico donde cada elemento D_{ij} representa la importancia relativa del término j en el metadocumento i.

Varios criterios han sido propuestos para definir la importancia de cada término en un documento. El esquema más frecuente es el comúnmente conocido como $TF \times IDF$, donde TF es la frecuencia del término (*term frequency*). Este esquema de pesos penaliza la importancia de términos comunes que aparecen en muchos documentos y que por tanto no son significativos del contenido de los mismos. Formalmente, el IDF para el término j-ésimo es calculado como:

$$idf_j = log(\frac{T}{t_j})$$

Table 7.2: Matriz de categorías GO por términos

Categorías GO	**Term. 1**	**Term. 2**	**Term. 3**	**Term. 4**	**Term. 5**	**Term. 6**
Categoría A	1	1	0	1	1	0
Categoría B	1	1	0	1	1	0
Categoría C	1	1	0	1	1	1
Catergoría D	0	0	0	1	0	0
Categoría E	1	1	0	0	0	0
Categoría F	1	1	0	0	0	0

donde T es el número total de metadocumentos (número total de categorías GO en nuestro caso) y t_j es el número de metadocumentos que contienen el término j. Por lo tanto, el peso asignado al término j en el metadocumento i bajo el esquema $TF \times IDF$ queda definido como:

$$D_{ij} = idf_{ij} \times \frac{tf_{ij}}{max_k[tf_{ik}]}$$

donde $max_k[tf_{ik}]$ es la moda del metadocumento i , es decir, la frecuencia con la que aparece la palabra más recurrente del metadocumento. Dividir la frecuencia de un término por la moda del documento es muy frecuente y se utiliza para normalizar los valores.

El último paso de esta fase consiste en el filtrado del corpus total en función de los pesos asignados. El propósito básico de este proceso es el de ir depurando el corpus paso a paso para poder identificar cada categoría con las palabras que mejor la definen. De esta manera, eliminamos del sistema todas aquellas palabras que no alcanzan un determinado peso umbral, es decir, aquellas que son poco representativas y no han de tenerse en cuenta. De esta manera, hemos conseguido crear un conjunto de palabras (un metadocumento) que representa a cada categoría de GO.

Una vez hecho esto, sólo hay que saber si un término aparece o no dentro de un metadocumento. Por ejemplo, consideremos la tabla binaria 8.9, que indica si una palabra está contenida dentro de una caegoría GO o no.

Este tipo de matrices pueden ser transformadas fácilmente en una base de datos de transacciones en la cual cada término representa una transacción (un cliente, en analogía con el caso de las transacciones comerciales) y el conjunto de categorías GO representa el conjunto de elementos que aparecen en cada transacción (los productos que compra cada cliente) (ver tabla 7.3).

Table 7.3: Base de datos de transacciones para extraer reglas asociativas entre conjuntos de términos

Transacciones	**Conjunto de elementos**
Term. 1	Categoría A, Categoría B, Categoría C, Categoría E, Categoría F
Term. 2	Categoría A, Categoría B, Categoría C, Categoría E, Categoría F
Term. 4	Categoría A, Categoría B, Categoría C, Categoría D
Term. 5	Categoría A, Categoría B, Categoría C
Term. 6	Categoría C, Categoría F

7.1.3 Extracción de anotaciones enriquecidas en la base de datos

La base de datos creada consta de un conjunto de términos anotados con categorías de Gene Ontology. Dada una consulta del usuario, compuesta por una combinación de palabras (que puede ser el resultado de un experimento anterior, puede ser un conjunto de palabras relacionadas con un determinado gen o proteína o simplemente el abstract de un documento), el sistema procesa las palabras de la lista de entrada, elimina los términos de parada, aplica sobre ellas el algoritmo de stemming de Porter y las busca en la lista de términos del sistema.

En el siguiente paso, por cada palabra encontrada, se recuperan todas las categorías GO asociadas a, como mínimo, un número determinado de palabras. Dicho número, que por defecto el sistema considera que es 3, puede ser modificado por el usuario. De esta manera, sólo se recuperarán aquellas anotaciones que aparezcan relacionadas con al menos 3 términos de la consulta del usuario.

Por ejemplo, si utilizamos el caso de la tabla 7.3 con un soporte umbral de 3 y la entrada del usuario es una lista compuesta por los términos {*Term 1., Term. 6, Term. 5*} , el sistema devolvería la salida {*Categoría C*}. Si observamos la tabla, notaremos que *Categoría C* es la única anotación compartida por los tres términos. Sin embargo, si el soporte umbral hubiese sido de 2, la salida del sistema hubiese sido {*Categoría A, Categoría B, Categoría C, Categoría F*} , el conjunto de todas las anotaciones compartidas por al menos 2 términos de la lista de entrada.

7.2 Análisis estadístico

Una vez extraídas todas las anotaciones (categorías GO) enriquecidas en la lista de términos, el siguiente paso es realizar el análisis estadístico. Para esto el método debe contar las ocurrencias de cada anotación en la lista de palabras de entrada y en la lista de palabras que se tome por referencia (que por defecto es la base de datos entera). Nótese que la frecuencia calculada de cada anotación es calculada como el número de palabras que están simultaneamente co-anotadas con ella.

A partir de esta información se aplica un análisis estadístico para identificar las categorías que están significativamente enriquecidas en la lista de palabras. Existen varios tests estadísticos para calcular la suficiencia estadística *(p-valor)* de cada anotación. En este trabajo se han implementado dos análisis distintos: el basado en la distribución hipergeométrica y el test de indepencia de χ^2.

En este caso, el significado biológico de los p-valores calculados no es difícil de entender. Si tenemos una lista de términos de entrada y observamos que "mitosis" aparece enriquecida y con un valor de p-value muy bajo (próximo a cero), es que la entrada está relacionada de manera significativa con el concepto de "mitosis" por alguna razón. Aunque no se incluye información negativa en esta primera implementación del trabajo, no sería descabellado, del mismo modo que se hace en los análisis de patrones de expresión genéticos, incorporar información acerca de qué términos aparecen "inhibidos" ante determinadas categorías de GO (extrayendo dicha información de la literatura o a través de varios expertos), de tal manera que se pudiese incorporar como un filtro. En ese caso una anotación que aparece relacionada negativamente con un grupo de términos, y lo hace de manera significativa, nunca podría ser relacionada con dichos términos, mejorando así la precisión del sistema.

En algunos análisis, los investigadores desean clasificar un documento o conjunto de palabras, pero en función de un determinado subgrupo de documentos. Por ejemplo, sólo desea análizar unos resultados previos comparándolos con los documentos relacionados con la "meiosis". En este caso se permite que la lista de referencia utilizada no sean los términos de toda la base de datos sino los términos que el investigador indique (en este caso, los extraídos de todos los documentos de PUBMED relacionados con la "meiosis").

7.2.1 Test basado en la distribución hipergeométrica

La distribución hipergeométrica es una distribución de probabilidad discreta. Es el modelo que se aplica en experimentos del tipo *En una urna hay bolas de*

dos colores (blancas y negras), ¿cuál es la probabilidad de que al sacar 2 bolas las dos sean blancas?. Son experimentos donde, al igual que en la distribución binomial, en cada ensayo hay tan sólo dos posibles resultados: o sale blanca o no. Pero se diferencia de la distribución binomial en que los distintos ensayos son dependientes entre sí, *si en una urna con 5 bolas blancas y 3 negras, en un primer ensayo saco una bola blanca, en el segundo ensayo hay una bola blanca menos, por lo que las probabilidades son diferentes (hay dependencia entre los distintos ensayos).*

Para nuestro problema, consideremos que existen N términos en total en nuestra base de datos. Un determinado término puede estar anotado o no por una determinada categoría de GO que vamos a llamar F. En otra palabras, podemos decir que nuestros N términos pueden ser de dos tipos: los que estan anotados con la categoría F y los que no lo están *(F y NF)*. Supongamos ahora que en la entrada el usuario ha introducido un subconjunto de K términos. Observamos que x de esos K términos son de tipo F y queremos saber cuál es la probabilidad de que eso sea fruto del azar. De esta manera, podemos plantear nuestro problema de la siguiente manera: tenemos N términos de los cuales M son de tipo F y $N - M$ son de tipo NF, si cogemos aleatoriamente K términos, cuál es la probabilidad de que exactamente x de esos K términos sean de tipo F. Una vez que se ha cogido un término de la base de datos, evidentemente no se puede volver a coger, así que está claro que no hay reemplazamiento.

La probabilidad de que una cierta categoría GO ocurra x veces sólo por azar en una lista de términos se puede calcular mediante la distribución hipergeométrica con los parámetros (N,M,K) (7.1).

$$P(X = x|N, M, K) = \frac{\binom{M}{x}\binom{N-M}{K-x}}{\binom{N}{K}} \tag{7.1}$$

Basándonos en esto, el p-valor de tener x términos o menos anotados con F puede ser calculado sumando las probilidades de que en una lista aleatoria de K términos haya 1, 2, $\cdots$, x términos de la categoría F (7.2).

$$p = \sum_{i=0}^{x} \frac{\binom{M}{i}\binom{N-M}{K-i}}{\binom{N}{K}} \tag{7.2}$$

Este test corresponde evalúa el p-valor de categorías poco enriquecidas, sin embargo si quisiéramos calcular el p-value de categorías muy enriquecidas la ecuación sería distinta (7.3)

$$p = 1 - \sum_{i=0}^{x} \frac{\binom{M}{i}\binom{N-M}{K-i}}{\binom{N}{K}} \tag{7.3}$$

$$= \sum_{i=x}^{N} \frac{\binom{M}{i}\binom{N-M}{K-i}}{\binom{N}{K}} \tag{7.4}$$

Aunque la distribución hipergeométrica es en ocasiones difícil de calcular, se sabe que tiende a la binomial cuando el valor de N es muy elevado. Si se usa la binomial, la probabilidad de tener x términos anotados con F en un conjunto de K términos extraídos al azar es dada por la clásica fórmula de la probabilidad binomial (7.6)

$$P(X = x|K, M/N) = \binom{K}{x}\left(\frac{M}{N}\right)^{x}\left(1 - \frac{M}{N}\right)^{K-x} \tag{7.5}$$

y, de manera análoga a antes, el p-valor sería calculado por (**??**)

$$p = \sum_{i=x}^{N} \binom{K}{i}\left(\frac{M}{N}\right)^{i}\left(1 - \frac{M}{N}\right)^{K-i} \tag{7.6}$$

7.2.2 Test de χ^2

Existen, sin embargo, otras alternativas para llevar a cabo este test, como es el test de la χ^2, la prueba exacta de fisher, la prueba de McNemar o la prueba Q de Cochran, entre otras. El test de la χ^2 permite determinar si dos variables cualitativas están o no asociadas. Si al final del estudio se concluye que las variables no están relacionadas podremos decir con un determinado nivel de confianza, previamente fijado, que ambas son independientes. Para llevar a cabo un análisis mediante el test de la χ^2, los datos deben ser organizados en tablas de contingencia como la que se muestra en la tabla 7.4. La notación de un punto en el subíndice indica la suma de todos los elementos de esa fila o columna. Mediante esta notación, el número de términos en la base de datos (o en la lista de referencia usada) $N = N_{.1}$, el número de términos anotados en F en la base de datos (o en la lista de referencia usada) es $M = n_{11}$, el número de términos de la lista de entrada $K = N_{.2}$ y el número de términos en la lista de entrada anotados en F $x = n_{12}$. La relevancia de una anotación F concreta puede ser calculada usando una tabla de contingencia de dimensiones 2×2. Los N términos de la lista de referencia pueden ser divididos en dos grupos: los que están anotados en F ($n_{11} = M$) y los que no lo están (n_21). Los K términos de la entrada son a su vez también divididos en dos grupos, los

Table 7.4: Tabla de contingencia

	Términos en lista de referencia	**Términos en entrada**	
F	n_{11}	n_{12}	$N_{1.} = \sum_{j=1}^{2} n_{1j}$
No F	n_{21}	n_{22}	$N_{2.} = \sum_{j=1}^{2} n_{2j}$
	$N_{.1} = \sum_{i=1}^{2} n_{i1}$	$N_{.2} = \sum_{i=1}^{2} n_{i2}$	$N_{..} = \sum_{i,j} n_{ij}$

anotados en F ($n_{21} = x$) y los que no lo están (n_{22}). Usando esta notación, el valor del estadístico χ^2 sería el de la ecuación (7.7)

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{(n_{ij} - E_{ij})^2}{E_{ij}} \tag{7.7}$$

donde E_{ij} son las frecuencias esperadas para cada celda de la tabla de contingencia y se calculan como las frecuencias totales en la fila y columna divididas por la frecuencia total. Así, el estadístico χ^2 mide la diferencia entre el valor que debiera resultar si las dos variables fuesen independientes y el que se ha observado en la realidad. Cuanto mayor sea esa diferencia (y, por lo tanto, el valor del estadístico), mayor será la relación entre ambas variables. El hecho de que las diferencias entre los valores observados y esperados estén elevadas al cuadrado convierte cualquier diferencia en positiva. Si se hacen cálculos se puede llegar a la ecuación (7.8)

$$\chi^2 = \frac{N_{..}\left(|n_{11}n_{22} - n_{12}n_{21}| - \frac{N_{..}}{2}\right)^2}{N_{1.}N_{2.}N_{.1}N_{.2}} \tag{7.8}$$

donde $N_{..}/2$ en el numerador es un factor de corrección que puede ser omitido cuando la muestra es muy grande.

Cada valor de χ^2 puede ser comparado con los valores críticos obtenidos de una distribución con grado de libertad 1.

7.3 Corrección de p-valores en comparaciones múltiples

El problema de las comparaciones múltiples aparece cuando un test estadístico es usado repetidamente para evaluar un número relativamente elevado de observaciones. En un experimento estadístico, se considera que un p-valor es significativo siempre que sea menor que un determinado *valor alpha*. El valor alpha consiste simplemente el umbral a partir del cual se acepta

que un determinado hecho no es producto del azar. Por ejemplo, en un experimento usando un valor alpha de 0.05, existe una posibilidad entre 20 de que un determinada hipótesis nula pueda ser dada por significativa por pura probabilidad. Cuando se realiza un test de comparaciones múltiple, cada hipótesis tiene una determinada probabilidad de ser considerada como verdadera a pesar de ser falsa. Si se prueban 10 hipótesis y el valor de alpha es de 0.05, entonces la probabilidad de encontrar al menos una diferencia aparentemente significativa debido a una posibilidad arbitraria es de 0.4 (que es $1 - 0.95^10$).

Por tanto este problema es crucial en el análisis de datos que estamos haciendo, ya que el número de anotaciones de GO es del orden de unos pocos miles para un determinado organoismo. Cuando se aplica un test estadístico a un número elevado de casos la probabilidad de encontrar falsos positivos (lo que se denomina error tipo 1) aumenta significativamente. Por ejemplo, si en nuestro análisis evalúamos 5000 anotaciones de Gene Ontology, si consideramos como estadísticamente significativos aquellas anotaciones con un p-valor de 0.01 tendríamos que esperar una tasa de error del orden de 50 falsos positivos.

Existen varios métodos para corregir este tipo de errores y su uso es más que recomendable en cualquier análisis donde se lleven a cabo comparaciones múltiples.

Las técnicas se explicarán en el contexto de un análisis de expresión diferencial donde se ha calculado un determinado p-valor (utilizando cualquier análisis estadístico) para cada una de las N anotaciones entiquecidas en una lista de términos.

7.3.1 Corrección de Bonferroni

Es el método de corrección más común. Puede ser descrito de manera muy simple. Cuando un tests evalúa varias hipótesis nulas $H_i(i = 1, \cdots, n)$, con el fin de corregir el error de tipo 1 de manera global, cada uno de los correspondientes p-valores P_i es comparado con el valor de alpha dividido por el número de hipótesis (anotaciones). De esta manera, la probabilidad global de encontrar un falso positivo es la misma que la de encontrar un falso positivo en un experimento con una única hipótesis, asumiendo que las pruebas son independientes.

La corrección de Bonferroni se considera extremadamente conservativa, de tal manera que cuando se analizan muchas anotaciones, como es el caso, puede que al aplicar la corrección por Bonferroni no quede ninguna anotación significativa. Además, en nuestro caso, no estaría claro si las hipótesis (anotaciones de GO) son independientes, porque los mismos nodos de GO

Table 7.5: Corrección de Holm

términos	t_{i1}	t_{i2}	$\cdots$	t_{iN}
p-values crecientes	p_1	p_2	$\cdots$	p_N
p-values ajustados	$p_1 \times N$	$p_2 \times N - 1$	$\cdots$	p_N

están estructurados en forma de grafo. En cualcuier caso, el procedimiento de la corrección de Bonferroni es equivalente a corregir los p-valores multiplicándolos por el número de anotaciones evaluadas (7.9)

$$\text{p-valor}_{ajustado} = \text{p-valor} \times N \tag{7.9}$$

7.3.2 Corrección de Holm

Holm propuso un método que se aplicaba en los mismos casos que el procedimiento de Bonferroni, pero que es más potente. En este caso el procedimiento sería de la siguiente manera. En primer lugar se deben ordenar los p-valores de menor a mayor. Diremos que P_1 es el más pequeño y P_N el mayor de todos. En segundo lugar, cada p-valor es comparado con $\alpha/(n-i+1)$, empezando por P_1 y continuando hasta llegar al primer p-valor no rechazado. De esta manera, las hipótesis H_i rechazadas serán aquellas para las que $P_j \leq \frac{\alpha}{n-j+1} <$ para toda $j \leq i$.

El procedimiento a seguir sería análogo a multiplicar cada p-valor por $N - K + 1$, donde k es la posición (rango) que ocupa en la lista 7.5.

Esta corrección es un poco menos conservativa que la de Bonferroni aunque también asume independencia entre las variables.

7.3.3 FDR propuesto por Benjamini y Hochberg

Las correcciones clásicas en comparaciones múltiples suelen corregir sólo el error de tipo 1 (un falso positivo). Esto es a veces insuficiente y además puede no controlar la aparición de falsos negativos. Un método alternativo de corrección es calcular el ratio de descubrimiento falso (FDR), que consiste en la proporción entre las hipótesis nulas ciertas rechazadas y todas las hipótesis nulas rechazadas (Benjamin y Hochberg, 1995 [72]), en otras palabras, es la proporción de todas las hipótesis que se estima que serán significativas, pero que actualmente no lo son.

Sean $H_1 \ldots H_N$ las hipótesis nulas y $P_1 \ldots P_N$ sus correspondiente p-valores. En primer lugar hay que ordenar los p-valores, en este caso de mayor a menor. Les denotaremos como $P_{(1)} \ldots P_{(N)}$. Para un valor de α dado, hay

Table 7.6: FDR

términos	t_{i1}	t_{i2}	$\cdots$	t_{iN}
p-values crecientes	p_1	p_2	$\cdots$	p_N
p-values ajustados	$\frac{p_1 \times N}{1}$	$\frac{p_2 \times N}{2}$	$\cdots$	$\frac{p_N \times N}{N}$

que encontrar la k mayor tal que $P_{(k)} \leq \frac{k}{N}\alpha$. De esta manera, se rechazan (se declaran positivas) todas las hipóteis $H(i)$ para $i = 1, \ldots, k$.

En este caso, el método es análogo a multiplicar cada p-valor por N/K 7.6

7.3.4 Corrección basada en permutaciones

Por último, se decribirá la corrección basada en permutaciones. En primer lugar, se deben calcular los p-valores para cada anotación en los datos, como en los métodos anteriores. Una vez que se ha hecho esto, lo siguiente es permutar aleatoriamente las clases, y entonces volver a calcular los p-valores en estos datos. Este paso se repite n veces, en algunos sistemas se debe repetir del orden de unas 1000 veces. El valor de n depende de la magnitud de la base de datos con la que se esté trabajando y el número de hipótesis. Para cada anotación, el p-valor ajustado se calculará como el número de permutaciones en las que se ha encontrado un p-valor menor o igual que el p-valor real para esa anotación, dividido por el número total de permutaciones hechas (n).

Este método tiene en cuenta las posibles correlaciones entre anotaciones, aunque es computacionalmente costoso y lento dado el elevado número de iteraciones que se necesitan. Una alternativa es implementar este algoritmo y que su ejecución pueda realizarse en paralelo, aliviando de esa manera esa carga computacional.

Capítulo 8

Implementación

La aplicación que se propone es sencilla en su planteamiento: se toma una lista de términos como entrada (por ejemplo de un abstract de un determinado artículo) y se determinan todas las anotaciones que tengan relación con ellas. No sólo es una manera de etiquetar temáticamente un determinado texto, sino que puede ser utilizado como fase posterior de análisis que devuelvan documentos o conjuntos de palabras, por ejemplo, si caracterizamos un determinado gen mediante un conjunto de palabras, podremos establecer directamente relaciones entre genes y anotaciones (por ejemplo, entradas de Gene Ontology), es decir, podremos caracterizar la función de un gen y los procesos biológicos con los que esté relacionado.

El proceso de implementación del sistema se desarrolló en dos etapas bien definidas. La primera de ellas se centraba en la adquisición de conocimiento y creación de una estructura de datos que contuviera toda la información necesaria. Para ello se accedió a distintas bases de datos Bioinformáticas a través de la red y se llevó a cabo toda la parte de procesamiento de texto. La segunda etapa tiene que ver con el desarrollo del algoritmo de extracción de anotaciones enriquecidas, el análisis estadístico y la corrección de los p-valores, así como la presentación de los resultados.

8.1 Etapa de entrenamiento: adquisición de la información

En esta fase se recupera toda la información de PUBMED y Gene Ontology. Para ello nos servimos de la base de datos de anotaciones GOA. En esta primera aproximación, nos centramos únicamente en la base de datos dedicada a la levadura (*Saccharomyces cerevisiae*), pero el sistema es tan flexible que más adelante se podrá ampliar fácilmente utilizando las distintas bases

de anotaciones de otros organismos como *Arabidopsis thaliana, Bos taurus, Caenorhabditis elegans, Danio rerio, Drosophila melanogaster, Gallus gallus, Homo sapiens, Mus musculus, Rattus norvegicus, Schizosaccharomyces pombe* y *Vibrio Cholerae*.

El flujo de ejecución de esta primera fase puede verse en la figura 8.1. Gene Ontology Annotations (GOA) es una base de datos generada con el fin de anotar cada gen con entradas de Gene Ontology, determinando así el papel que juega dicho gen en el organismo al que haga referencia la base de datos (exisen distintos ficheros de anotaciones, uno por organismo). Dentro de GOA, no sólo podemos ver asociaciones entre genes y términos de Gene Ontology, sino que, además de otro tipo de información, se puede ver qué artículo publicado en PUBMED evidencia dicha anotación. El sistema recorre toda la base de datos y recupera todas las tuplas del tipo {*Identificador de GO, Identificador de PUBMED*}. De esta manera conocemos todos los documentos publicados relacionados con cada categoría de Gene Ontology que tiene algo que ver con el organismo de la levadura. Una vez conocidas dichas relaciones, el sistema se encarga de extraer información de Gene Ontology y de PUBMED. En primer lugar accede a Gene Ontology y por cada identificador recupera los campos {*Nombre; Definición; Sinónimos*}. Por ejemplo, para el identificador *GO:0006200* tendríamos {*ATP catabolic process; The chemical reactions and pathways resulting in the breakdown of ATP, adenosine 5'-triphosphate, a universally important coenzyme and enzyme regulator; ATP breakdown, ATP catabolism, ATP degradation, ATP hydrolysis*}. El sistema se encarga de dividir cada uno de estos campos en palabras. Las palabras de parada son eliminadas y a las restantes se les aplica el algoritmo de lematización de Porter, quedándose únicamente con la raíz de cada término. Una vez hecho esto, el sistema crea un metadocumento relacionado a la categoría *GO:0006200*, compuesto por todas las raíces de los términos recuperados.

Una vez extraída la información de Gene Ontology, la siguiente base de datos a analizar es PUBMED. PUBMED es una base de datos compuesta por documentos biomédicos publicados. Establecidas las relaciones entre identificadores de GO y de PUBMED gracias a la base de datos de GOA, sabemos qué documentos están relacionados con qué categorías. Recuperamos por cada uno de los documentos los campos {*Título, Abstract*}. De la misma manera que antes, se divide cada campo en términos, y aquellos que no se encuentren en la lista de palabras de parada son lematizados aplicándoles el algoritmo de Porter. Por último, añadimos las raíces de los términos recuperados de PUBMED a los metadocumentos creados para cada categoría de Gene Ontology.

De esta manera, una misma palabra puede estar contenida en más de un metadocumento. En este punto, y con el fin de disminuir la cantidad

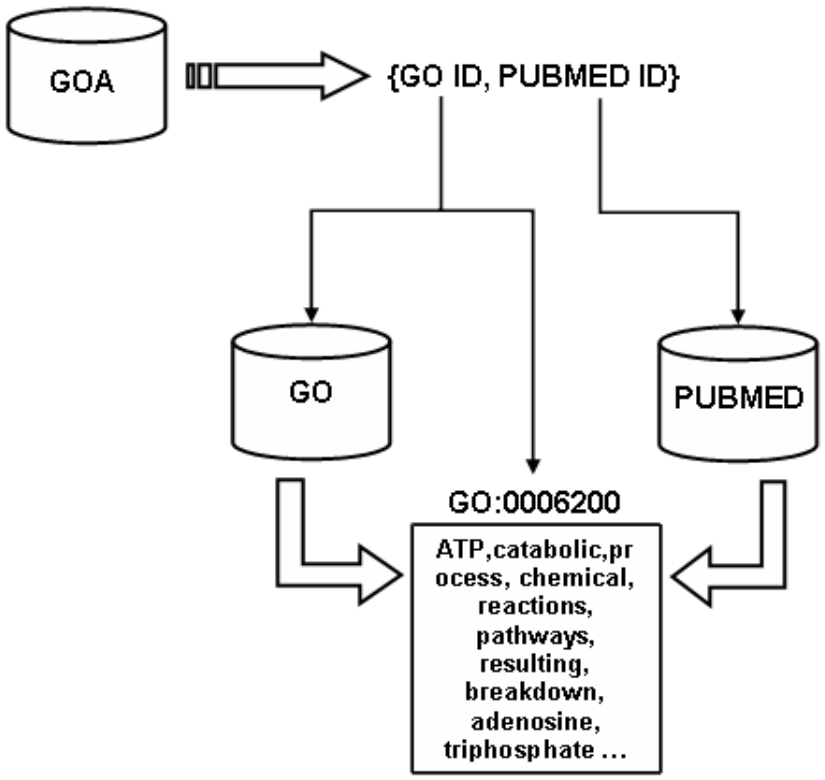


Figura 8.1: Flujo de información en el proceso de extracción de datos de las bases de datos

de cómputo necesaria en la siguiente etapa, nos disponemos a reducir en la medida de lo posible el tamaño de la base de datos creada. La idea es que cada categoría GO quede representada por una serie de términos que reflejen fielmente sus características, es decir, aquellas palabras que sean más importantes dentro de la categoría. Esto lo conseguimos mediante dos pasos. En primer lugar hacemos pasar las palabras por un filtro, aquellas palabras que aparezcan en más del 80% de los metadocumentos no serán tenidas en cuenta. De esta manera, evitamos tener palabras poco representativas en nuestra base de datos, por ejemplo, es de esperar que el término "yeast" (levadura en inglés) aparezca en todos o casi todos los documentos publicados acerca del organismo de la levadura y sin embargo no es una palabra que aparezca tradicionalmente en una lista de stopwords.

Una vez hecho esto, es el momento de calcular el peso que tiene cada una de las palabras de cada uno de los metadocumentos, es decir, asignamos pesos a las palabras en función a su relación con la categoría de GO. Para ello seguimos el esquema TFIDF ya comentado, por una parte dividimos cada frecuencia de cada palabra por la moda del metadocumento (la frecuencia de la palabra más frecuente) y, una vez hecho esto, se calcula el peso mediante la ecuación (8.1)

$$D_{ij} = log(\frac{T}{t_j}) \times tf_{ij_{normalizada}} \tag{8.1}$$

donde T es el número total de metadocumentos (número total de categorías GO en nuestro caso), t_j es el número de metadocumentos que contienen el término j y $tf_{ij_{normalizada}}$ es la frecuencia con la que aparece el término en el metadocumento, normalizada tal y como se ha explicado antes.

El cálculo de los pesos se utiliza como segundo paso para filtrar las palabras, sólo aquellas cuyo peso alcance un determinado umbral será utilizadas para representar cada categoría GO, el resto de palabras no será tenido en cuenta.

Una vez creados los metadocumentos y filtrado las palabras, el siguiente paso es crear la base de datos de transacciones. Si lo que tenemos hasta este punto es una serie de categorías GO representadas por un conjunto de palabras, nuestra base de datos estará formada por una serie de palabras, cada una relacionada con una serie de categorías GO (figura 8.2). Esta es la manera de que en la siguiente etapa podamos recuperar aquellas anotaciones enriquecidas en un conjunto de palabras determinado.

Todo el proceso desarrollado en este punto se lleva a cabo tres veces. Como se ha comentado en el capítulo 5, Gene Ontology es un sistema que en realidad está compuesto por tres ontologías distintas: *procesos biológicos, funciones moleculares y componentes celulares*. El sistema que presentamos

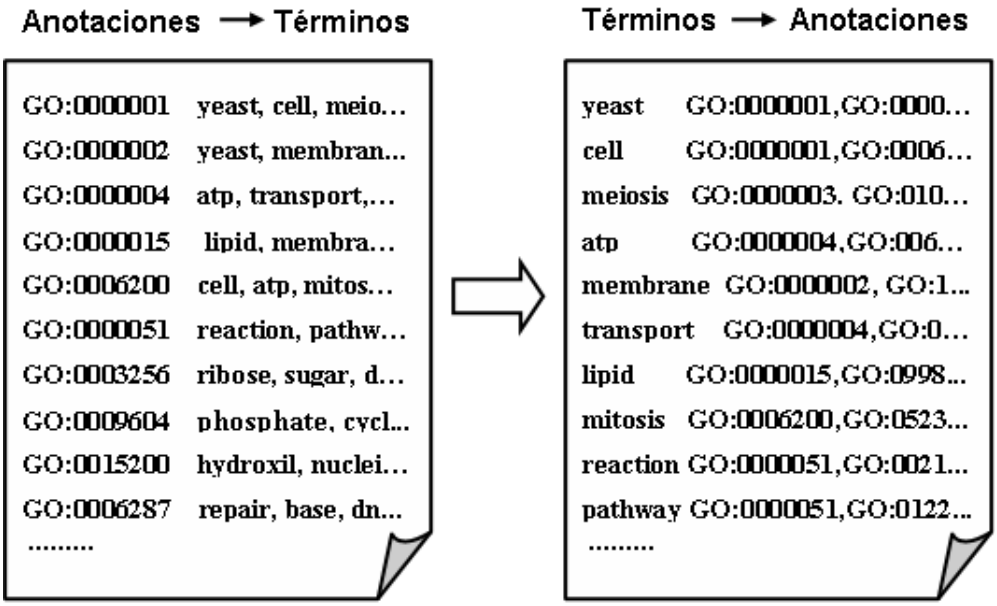


Figura 8.2: A partir de los metadocumentos de cada anotación creamos las bases de transacciones, compuestas por palabras anotadas

crea una base de datos por cada una de estas ontologías, es decir, una base de datos compuesta por términos relacionados con categorías GO de procesos biológicos, otra con funciones moleculares y otra con componentes celulares. El usuario podrá elegir al hacer la consulta qué base de datos debe ser utilizada. Por último, y con el fin de dar uniformidad a todo el sistema y acelerar la respuesta una vez el usuario introduzca su consulta, se asigna un identificador único a cada palabra que esté presente en alguna de las tres, o las tres, bases de transacciones. Este identificador permanecerá guardado en una estructura a parte y apuntará a una única palabra, esté en la base de transacciones que esté (figura 8.3)

8.2 Etapa de análisis

En este trabajo se ha desarrollado una herramienta que fuese accesible a través de la web de manera gratuita. Inicialmente dicha herramienta sólo proporciona anotaciones de Gene Ontology, pero en un futuro incorporará también *rutas de KEGG, Interpro Motifs, SwissProt Keywords o términos GO Slim*. Además, aunque en esta primera versión sólo se trabaje con el organismo de la levadura, la herramienta desarrollada es fácilmente escalable

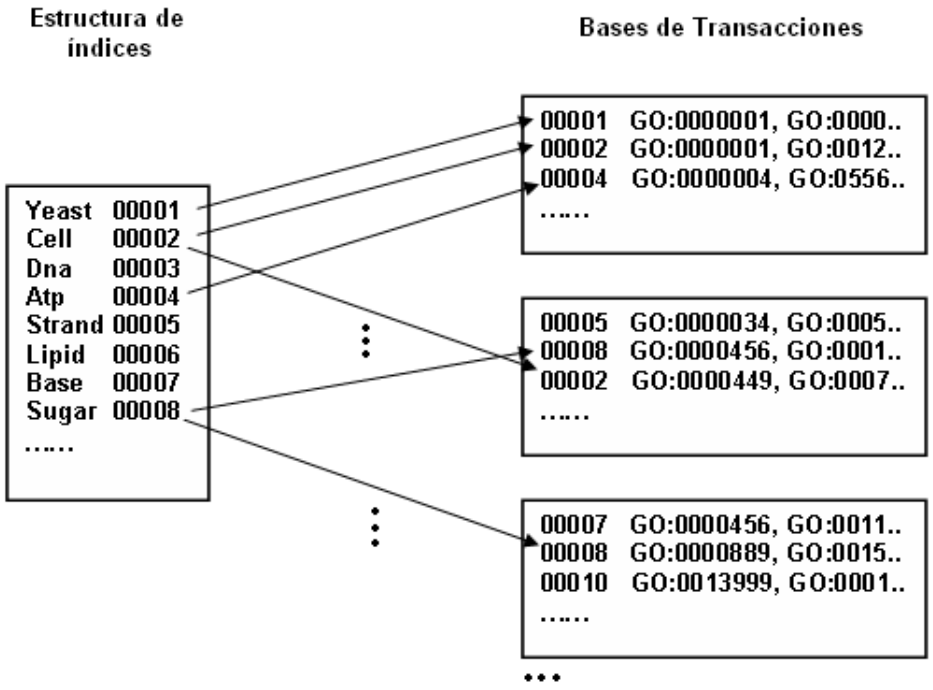


Figura 8.3: Estructura de Índices que enlaza con las bases de transacciones

y soportará en breve organismos como *Arabidopsis thaliana, Bos taurus, Caenorhabditis elegans, Danio rerio, Drosophila melanogaster, Gallus gallus, Homo sapiens, Mus musculus, Rattus norvegicus, Schizosaccharomyces pombe* y *Vibrio Cholerae*.

El interfaz permite elegir el organismo y tipo de anotación que se pretende analizar, elegir el test estadístico que se quiere llevar a cabo, si se quiere corregir los p-valores calculados, la inclusión de una lista de términos de referencia y, por su puesto, la introdcucción del documento de entrada.

Una vez hecha una consulta, el sistema recupera el documento de entrada y lo divide en palabras. Se analiza cada palabra, si está incluída dentro de nuestra lista de stopwords se descarta, sino, se le aplica el algoritmo de stemming de Porter y nos quedamos sólo con la raíz. Una vez hecho esto, se acude al esqueleto de la base de datos para recuperar los identificadores de las palabras introducidas en la consulta. Esos identificadores nos permitirán acceder de manera más rápida a la base de transacciones adecuada (en función del organismo y el tipo de anotaciones elegidos, figura 8.4).

En este punto es necesario hacer una breve aclaración. Si la entrada no contiene ninguna palabra contemplada por el sistema, indudablemente no se podrá devolver ninguna salida. En este caso, no se considera que el documento introducido por el usuario esté suficientemente relacionado con ninguna anotación. Es posible que una relajación en el filtrado comentado en la sección anterior pudiera aliviar este resultado, pero en ese caso empeoraría considerablemente la precisión de los resultados y además se podrían relacionar documentos y anotaciones que en un principio tienen poco o nada en común.

Una vez identificadas las palabras en la base de transacciones correspondiente, el sistema procede a extraer las anotaciones enriquecidas. Como se ha mostrado, la base de transacciones creada consta de un conjunto de términos anotados. Se recuperan todas las anotaciones asociadas a cada palabra encontrada. Una vez hecho esto, se analiza cada anotacion, se considera enriquecida si aparece relacionada a, como mínimo, un número determinado de palabras. Dicho número, que por defecto el sistema considera que es 3, puede ser modificado por el usuario. De esta manera, por defecto sólo se recuperarán aquellas anotaciones que aparezcan relacionadas con al menos 3 términos de la consulta del usuario (figura 8.5).

Una vez extraídas todas las anotaciones enriquecidas el sitema realiza el análisis estadístico elegido por el usuario. Se han implementado dos algoritmos de test estadístico: el de la distribución hipergeométrica y el test de χ^2. Sea cual sea el método elegido, el sistema debe contar las ocurrencias de cada anotación en la lista de palabras de entrada y en la lista de palabras que se tome por referencia (que por defecto es la base de transacciones entera). Nótese que la frecuencia calculada de cada anotación es calculada como

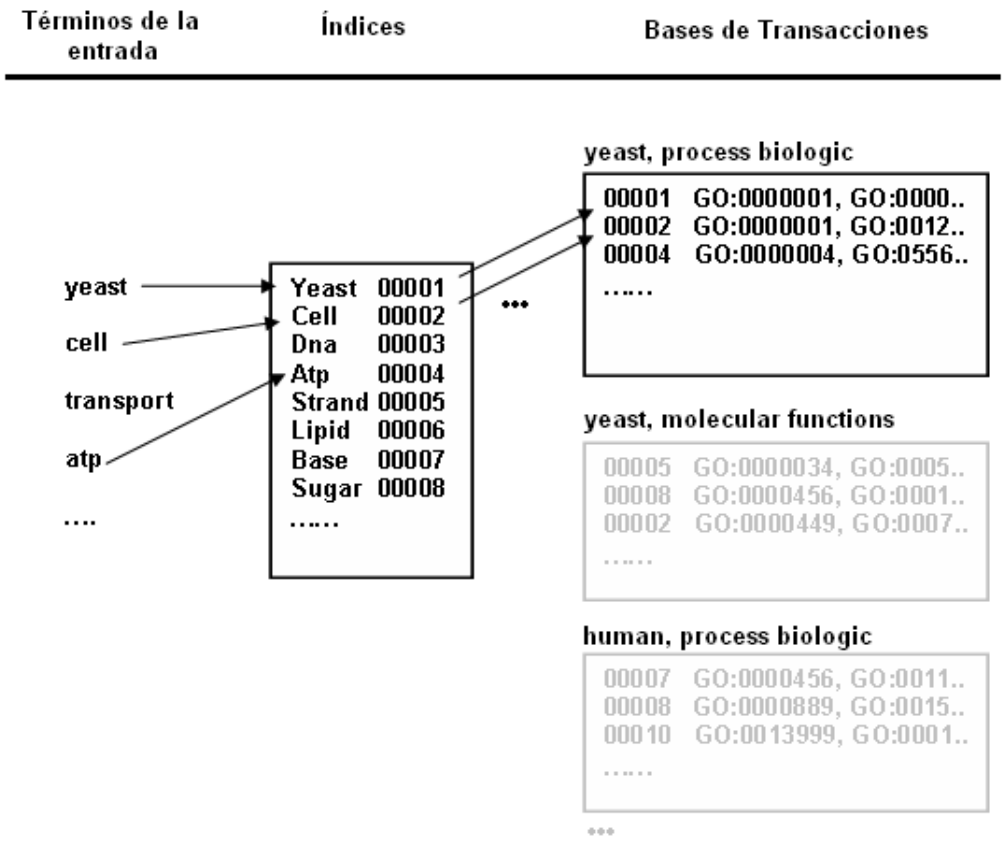


Figura 8.4: Una vez hecha la consulta, se buscan los identificadores en la estructura de índices y se acude a la base de transacciones adecuada, en este caso la base de datos correspondiente al organismo de la levadura y anotaciones de procesos biológicos de GO

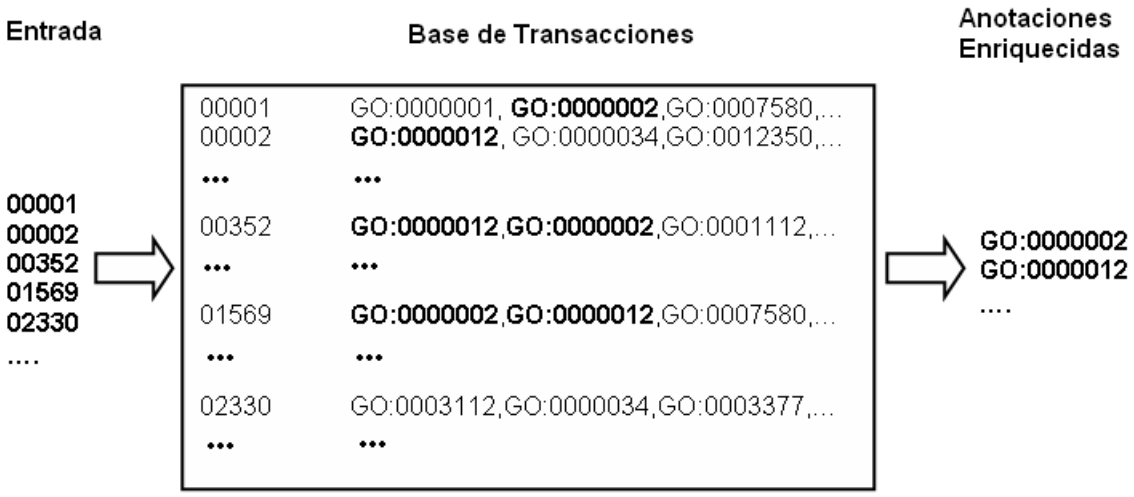
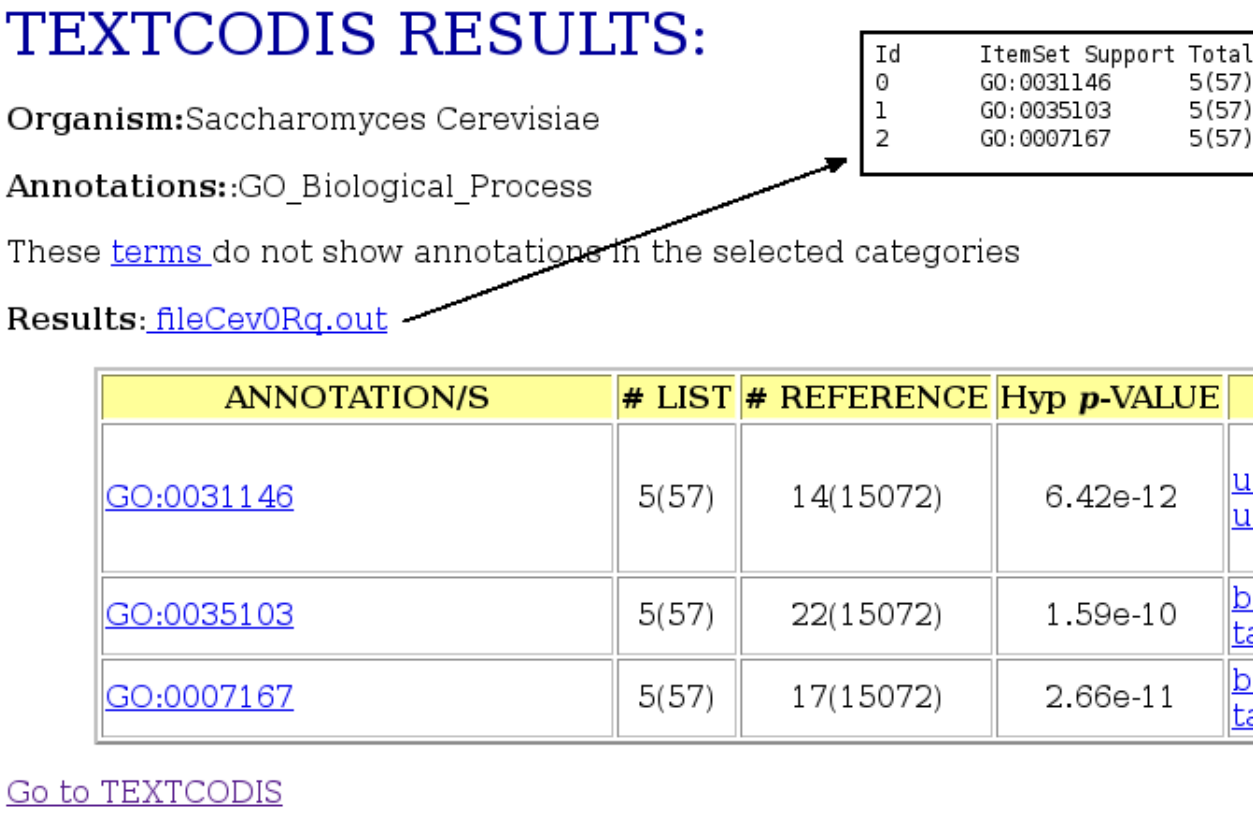


Figura 8.5: El sistema devolverá aquellas anotaciones que estén enriquecidas en el conjunto de palabras de entrada. En este caso el soporte mínimo es de 3

el número de palabras que están simultaneamente co-anotadas con ella. En la figura 8.5 podemos ver que la salida del sistema está compuesta por las anotaciones *GO:0000002 y GO:0000012*, que aparecen co-anotadas en tres palabras, mientras que otras anotaciones como *GO:00075800 o GO:0000034* no forman parte de la salida al estar anotadas sólamente 2 veces.

Según el test elegido, el p-valor de cada anotación será calculado de una manera u otra. Como ya se ha visto en el punto 7.2, si se ha elegido el test de la distribución hipergeométrica, el p-valor se calcula según la ecuación (7.3); en cambio si se elige la opción del test de χ^2 se crea una tabla de contingencia del mismo tipo que la tabla 7.4 y se calcula el valor del estadístico siguiendo la ecuación (7.7). De esta manera, el sistema ya tiene todas las anotaciones enriquecidas en la lista de palabras introducidas por el usuario con sus respectivos p-valores calculados según uno de los dos métodos implementados.

El siguiente paso es la correcion de los p-valores calculados. En el punto 7.3 se explica el problema de la hipótesis múltiple y se exponen una serie de métodos para corregirlo. En el sistema se han implementado tres opciones: el usuario tiene la posibilidad de elegir entre no corregir los p-valores o corregirlos mediante el método FDR o el método basado en permutaciones. En el primer método se ordenan las anotaciones en función de su p-valor calculado de mayor a menor, y se ajusta cada p-valor multiplicando por N/K donde N es el número total de anotaciones extraídas y K la posición que ocupa la



TEXTCODIS RESULTS:

Id	ItemSet	Support	Total
0	GO:0031146		5(57)
1	GO:0035103		5(57)
2	GO:0007167		5(57)

Organism: Saccharomyces Cerevisiae

Annotations::GO_Biological_Process

These terms do not show annotations in the selected categories

Results: fileCev0Rq.out

ANNOTATION/S	# LIST	# REFERENCE	Hyp *p*-VALUE	
GO:0031146	5(57)	14(15072)	6.42e-12	u u
GO:0035103	5(57)	22(15072)	1.59e-10	b t
GO:0007167	5(57)	17(15072)	2.66e-11	b t

Go to TEXTCODIS

Figura 8.6: Salida final de TEXTCODIS

anotación una vez se han ordenado todas, véase tabla 7.6.

En el segundo método se permutan las transacciones, de tal manera que cada término queda anotado aleatoriamente. Se extrae de esta manera el grupo de anotaciones asignadas a los términos de entrada por puro azar, y se calculan sus p-valores. El p-valor ajustado se calcula como el número de permutaciones en las que se ha encontrado un p-valor menor o igual que el p-valor real para ese término, dividido por el número total de permutaciones. El número de permutaciones que realiza el sistema es de 1000.

Una vez hecho esto, el sistema ya tiene calculadas las anotaciones enriquecidas en la lista de términos, con sus p-valores calculados y posteriormente corregidos si el usuario así lo ha querido. La salida final del sistema será una tabla donde se muestran los resultados, en código html y accesible a través

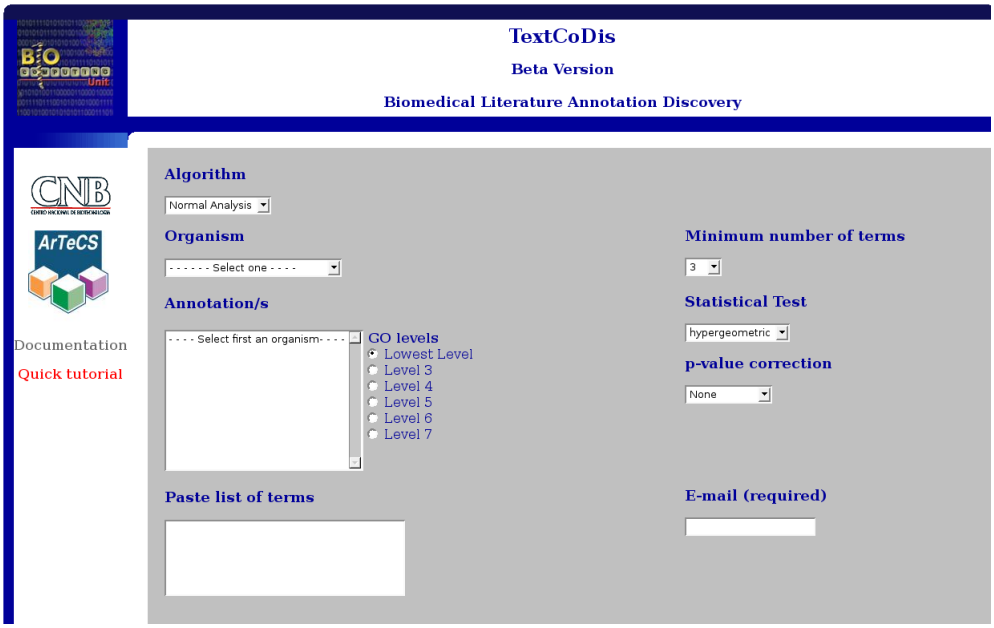


Figura 8.7: Interfaz de TEXTCODIS

de la web, y en formato ascii, vease figura 8.6. Si el usuario así lo ha decidido, le llegará un correo electrónico a su dirección de email avisándole de que el análisis ha terminado e indicándole donde puede ver los resultados.

8.3 Software desarrollado

TEXTCODIS es una herramienta web accesible pública y gratuitamente que actualmente se encuentra ya implementada y disponible en su primera versión beta. La interfaz de la aplicación se muestra en la figura 8.7. Como se puede observar, el sistema cuenta con diversos cuadros de texto y listas desplegables para configurar las diversas opciones.

En primer lugar 8.8 se puede elegir el tipo de algoritmo que se desea ejecutar, buscar anotaciones simples (análisis comentado en este trabajo) o conjuntos de anotaciones que co-ocurren en una determinada lista de términos (aún en fase de desarrollo). Una vez hecho esto, se debe elegir el orgnaismo para el que se desea realiza el análisis. Aunque para este trabajo sólo se ha incluído el análisis para el organismo de la levadura (*Saccharomyces cerevisiae*), está prevista la inclusión a corto plazo de otros organismos (*Arabidopsis thaliana, Bos taurus, Caenorhabditis elegans, Danio rerio, Drosophila melanogaster, Gallus gallus, Homo sapiens, Mus musculus, Rattus norvegi-*

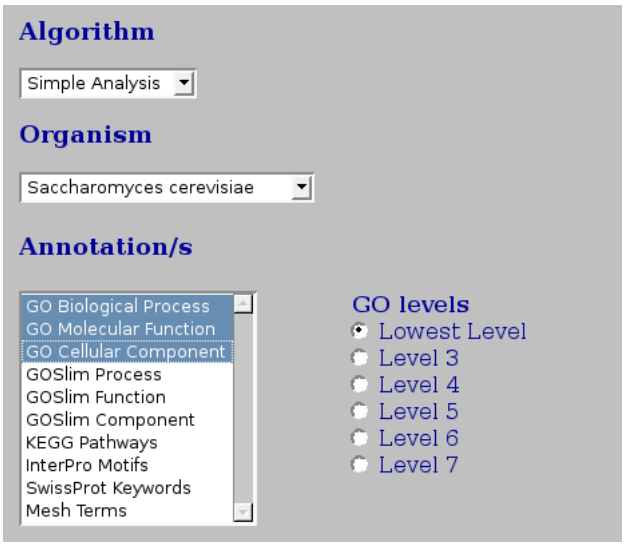


Figura 8.8: Selección de algoritmo, organismo y anotaciones en TEXTCODIS

cus, Schizosaccharomyces pombe y *Vibrio Cholerae.*) Para cada organismo el sistema proporcionará un análisis para diferentes anotaciones, incluyendo las tres categorías de Gene Ontology (biological process, cellular component, and molecular function), rutas KEGG, motivos de InterPro y keywords de SwissProt, pudiendo elegir más de una anotación de manera simultánea. Como se ha explicado anteriormente, en esta primera versión se han incluído las tres categorías de Gene Ontology.

Una vez seleccionado el tipo de algoritmo, el organismo y las anotaciones, el siguiente paso es pegar el documento o conjunto de palabas que se quieren analizar en el campo de texto indicado 8.9. Más adelante se permitirá subir directamente un fichero con el contenido del documento. También es posible pegar un conjunto de documentos de referencia en el siguiente campo de texto, por si el usuario desea conocer las anotaciones enriquecidas con respecto a un corpus determinado (y no a toda la base de datos de PUBMED y la información recogida de Gene Ontology).

En el campo "Número mínimo de términos" *(Mininum number of terms)* se puede elegir el soporte mínimo que deben tener las anotaciones recuperadas 8.10. Si se elige 3, el sistema recuperará aquellas anotaciones que estén co-anotadas en al menos 3 términos. En "Test estadístico" *(Statistical test)*, se puede seleccionar el test estadístico que se quiere llevar a cabo para calcular el nivel de significancia de las anoaciones recuperadas. En este sentido, se han implementado dos posibilidades, el test de χ^2 y el de la distribución hiper-

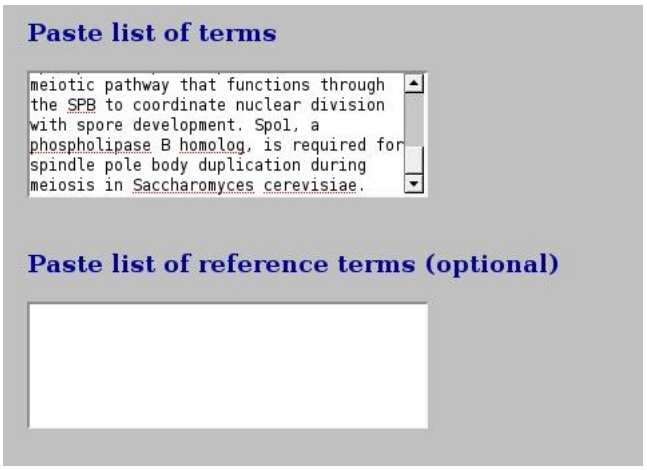


Figura 8.9: Campo indicado para introducir el documento y una lista de términos de referencia en TEXTCODIS

geométrica. En el campo "Correción de p-valores" *(p-value correction)* se puede elegir el algoritmo para corregir el problema de las múltiples hipótesis. Las tres opciones permitidas son no corregir los p-valores, corregirlos mediante el algoritmo de FDR o hacerlo mediante el algoritmo basado en permutaciones. ambos algoritmos fueron comentados en las secciones 7.6 y 7.3.4 respectivamente.

Por último, se permite al usuario proporcionar su dirección de correo electrónico por si desea que se le avise una vez el análisis haya terminado y se le facilite la dirección donde acceder a los resultados.

Una vez seleccionados todos los parámetros correctamente, se lanza el trabajo pulsando el botón *"submit"* y aparece una pantalla como la de la figura 8.11 en la que se indica el estado del proceso. Cuando el análisis esté completo, el navegador te redirecciona automáticamente a la página que contiene los resultados de tu análisis 9

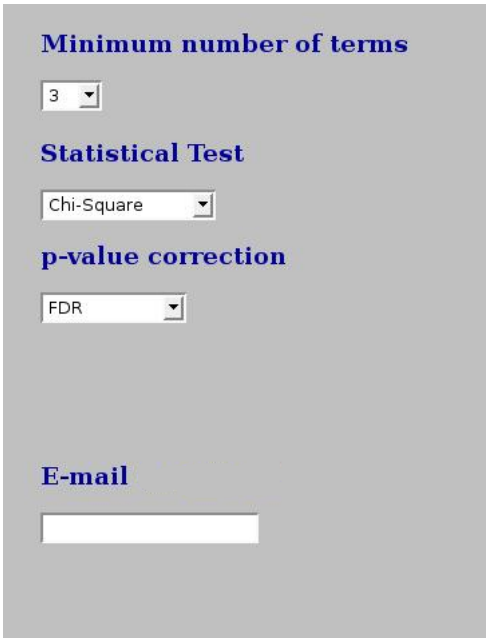


Figura 8.10: Selección de parámetros de análisis en TEXTCODIS

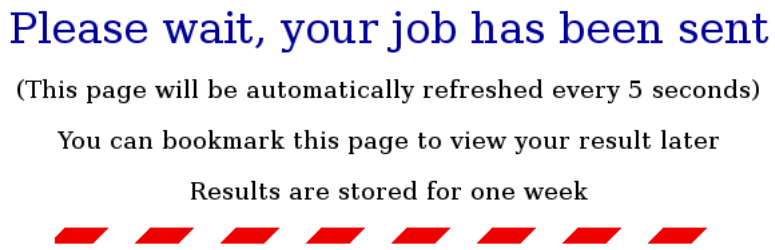


Figura 8.11: Pantalla que indica el estado de el análisis en TEXTCODIS

TEXTCODIS RESULTS:

Organism: Saccharomyces Cerevisiae

Annotations::GO_Biological_Process

These terms do not show annotations in the selected categories

Results: filekLyAN9.out

ANNOTATION/S	# LIST	# REFERENCE	Hyp *p*-VALUE	Hyp CORRECTED *p*-VALUE	TERMS	DESCRIPTION/S
GO:0045141	9(119)	28(15072)	0.00e+00	0.00e+00	progress, nuclear, pole, meiotic, spindl, meiosi, format, chromosom, homolog	(BP)telomere clustering
GO:0000135	7(119)	23(15072)	4.92e-12	6.56e-12	function, presenc, saccharomyc, cerevisia, chromosom, segreg, delai	(BP)septin checkpoint
GO:0007131	7(119)	32(15072)	1.06e-10	1.06e-10	meiotic, recombin, product, meiosi, chromosom, homolog, segreg	(BP)meiotic recombination
GO:0000089	7(119)	13(15072)	0.00e+00	0.00e+00	progress, it, mitosi, pole, spindl, chromosom, segreg	(BP)mitotic metaphase

Go to TEXTCODIS

Figura 8.12: Pantalla de resultados de TEXTCODIS

Capítulo 9

Resultados

Dada la dificultad de encontrar un corpus de documentos anotados lo suficientemente flexible y completo como para poder evaluar los resultados de la herramienta implementada, se recurrió a la opinión de expertos biológos para que introdujeran los documentos que creyeran convenientes y evaluaran los resultados obtenidos.

Podemos comentar, por ejemplo, los resultados de dos de los documentos evaluados para extraer conclusiones. Para el documento con *PMID 10089879* ([73]), los resultados con soporte 5 y test de la distribucion hipergeométrica se obtuvieron los resultados mostrados en la tabla 9.1

Las dos primeras anotaciones, *poliubiquitinación de proteínas* y *catabolismo de proteínas dependiente de ubiquitina*, (cuyo p-valor les asigna mayor significancia) se corresponden efectivamente con los procesos biológicos descritos en el *abstract* del artículo. La tercera anotación, *SCF-dependent proteasomal ubiquitin-dependent protein catabolism*, matiza el segundo proceso señalado, mediante los términos *proteasomal* y *dependiente de SCF* (siglas de *Skp1/Cul1/F-box protein*), siendo efectivamente, dicha anotación un nodo descendiente de la segunda anotación en la jerarquía de GO. Sin embargo, el conjunto de términos identificados en la regla (ver tabla 9.2) sólo soportarían la calificación de proteasomal, mientras que no evidencian ninguna dependencia con SCF.

Esto es una indicación de que parece ser necesario considerar la jerarquía (la estructura interna de Gene Ontology), bien en la propia creación de la base de datos de transacciones, bien en un proceso posterior de filtrado de reglas, o en ambos lugares.

Los dos procesos restantes son considerados a priori como falsos positivos. Por ejemplo, la identificación de "enzyme linked receptor protein signaling pathway" se hace en base a, entre otros términos, 'link', que en el texto del abstract aparece utilizado en un contexto totalmente diferente.

Table 9.1: Resultados de TEXTCODIS para el documento *PMID:10089879* con soporte 5 y test de distribución hipergeométrica

Anotaciones	**Entrada**	**Referencia**	**p-valor**	**Descripción**
GO:0000209	5(67)	10(15072)	1.21^{-12}	protein polyubiquitination
GO:0006511	5(67)	11(15072)	2.72^{-12}	ubiquitin-dependent protein catabolism
GO:0031146	5(67)	14(15072)	1.78^{-11}	SCF-dependent proteasomal ubiquitin-dependent protein catabolism
GO:0035103	5(67)	22(15072)	4.33^{-10}	sterol regulatory element binding-protein cleavage
GO:0007167	5(67)	17(15072)	7.29^{-11}	enzyme linked receptor protein signaling pathway

Table 9.2: Términos identificados en el análisis de TEXTCODIS para el documento *PMID:10089879*

Anotaciones	**Términos**
GO:0000209	multipl, chain, ubiquitin, protein, moieti
GO:0006511	ubiquitinprotein, catalyz, ubiquitin, proteasom, protein
GO:0031146	bind, activ, protein, regulatori, target
GO:0035103	bind, enzym, link, protein, target
GO:0007167	multipl, proteolysi, ubiquitin, protein, moieti

Table 9.3: Resultados de TEXTCODIS para el documento *PMID:10329624* con soporte 6 y test de distribución hipergeométrica

GO:0031929	7(81)	17(15072)	0.00e+00	TOR signaling pathway
GO:0002768	6(81)	16(15072)	9.01e-13	immune response-regulating cell surface receptor signaling pathway
GO:0007167	6(81)	17(15072)	1.64e-12	enzyme linked receptor protein signaling pathway

Table 9.4: Términos identificados en el análisis de TEXTCODIS para el documento *PMID:10329624*

GO:0031929	signal, kinas, rapamycin, avail, tor, target, nutrient
GO:0002768	signal, bind, inhibit, activ, respons, target
GO:0007167	signal, bind, kinas, catalyt, protein, target

De esta manera parece plausible que un tipo de análisis alternativo basado en frases o bifrases pudiera ser menos ruidoso, mejorando los resultados.

Para el documento con *PMID 10329624* ([74]), los resultados con soporte 6 y test de la distribucion hipergeométrica se obtuvieron los resultados mostrados en la tabla 9.3

De nuevo, la primera anotación (la de mejor p-valor) es correcta, correspondiento con la ruta de señalización mediada por proteínas TOR *(Target of rapamycin)* que se describe en el *abstract* del documento utilizado.

Las dos anotaciones siguientes sería falsos positivos. Se trata de dos procesos hijos del término GO que describe las rutas de señalización iniciadas por receptores de la superficie celular. Sin embargo, ninguno de los términos de la regla o su conjunto (ver 9.4)parecen implicar que se trate específicamente de dichos tipos de rutas.

En este caso se evidencia la necesidad de tener cierta medida de especi-

fidad o bondad de los términos asignados a las anotaciones, en el caso del modelo vectorial está claro que se trataría del peso tfidf, pero en este caso parece ser necesario tener en cuenta de alguna manera si un determinado término describe a una determinada anotación o no. En este sentido, también es factible asignar una medida de puntuación a las propias anotaciones en función de lo bien representadas o no que están, esto puede hacerse en función del número de documentos que hablan de ellas, del número de palabras utilizadas, de la frecuencia con la que aparecen esas palabras relacionadas con otras anotaciones, etc.

En definitiva, el análisis de los expertos biólogos es favorable, mostrando que la medida hipotética de *recall* sería elevada, aunque sin embargo es necesario corregir la aparición de falsos positivos para mejorar la medida de precisión.

+

Capítulo 10

Conclusiones

En el presente trabajo se ha propuesto un nuevo método para la extracción de información biológica a partir de grandes listas de términos resultantes del análisis de la literatura biomédica y de bases de datos como Gene Ontology. Además, se ha desarrollado una herramienta que implementa dicho método, accesible de manera gratuita por los investigadores.

La principal conclusión que puede derivarse de este trabajo es que la metodología propuesta basada en la extracción de categorías funcionales, y sus combinaciones, enriquecidas significativamente en un conjunto de términos de texto libre, es capaz de extraer información biológica relevante latente en el documento y por lo tanto no solo es capaz de etiquetarlo sino que además permite el descubrimiento automático de nuevos términos relevantes para describir procesos biológicos descritos en la literatura científica.

El método propuesto en este trabajo ha sido inspirado en las técnicas existentes de análisis funcional de genes, pero hasta donde sabemos es la primera vez que se propone para la extracción de información en texto científico. Los resultados experimentales demuestran que el sistema es efectivo en la caracterización funcional de resúmenes de texto, a la vez que detecta la importancia de nuevos términos para la descripción de procesos biológicos.

Por otra parte, el análisis estadístico posterior y la corrección de los p-valores parece ser adecuado, siendo en este caso las anotaciones más significativas (con un p-valor mejor) aquellas que precisamente más relación guardan con el documento de entrada.

A pesar de esto, los resultados revelan que es necesario mejorar el valor de precisión del sistema con el fin de no devolver falsos positivos que no estén realmente relacionados con los documentos. A partir de la investigación llevada a cabo en este trabajo han surgido muchas ideas que permiten la mejora de esos resultados. Por una parte, parece hacerse imprescindible el uso de la información que proporciona la misma jerarquía de Gene Ontology,

pudiendo heredar los nodos padres la información concerniente a sus nodos hijos, podando la jerarquía a un determinado nivel o analizando los resultados y devolviendo los antecesores más cercanos a las anotaciones propuestas.

Por otra parte, también es factible establecer un valor que indique el nivel de representación de cada anotación, basándonos en información como el número de documentos y palabras relacionados o en la frecuencia de dichas palabras, tanto para la anotación en sí como para el resto de anotaciones. En este sentido, y con el fin de mejorar la *representación semántica* de cada anotación es posible el uso de algunos de los métodos estudiados, como por ejemplo la técnica de *Latent Semantic Indexing*, comentada en el apartado 3.3.4

Aunque a nivel computacional el análisis implementado no es ineficiente (su ejecución no demora demasiado tiempo), el desarrollo de la aplicación ha revelado que es factible pararelizar en algunos puntos el código y hacer, por ejemplo, uso de un clúster de varios nodos para acelerar el análisis, por ejemplo en la corrección de los p-valores mediante el método de las permutaciones. También se pretende *gridificar* la aplicación, esto es, poder lanzar más de una petición a una estructura grid, con el fin de mejorar la productividad y así ser capaces de atender simultáneamente muchas más peticiones.

Así pues, el trabajo futuro se encuentra enmarcado dentro de estos tópicos.

Bibliografía

[1] The Gene Ontology Consortium. (2000) *Gene Ontology: tool for the unification of biology.* Nature Genet.25: 25-29.

[2] Barry Smith, Jennifer Williams & Steffen Schulze-Kremer.(2003) *The Ontology of the Gene Ontology.* Proceedings of AMIA Symposium.

[3] Soumya Ray & Mark Craven. (2005) *Learning Statistical Models for Annotating Proteins with Function Information using Biomedical Text* BMC Bioinformatics Vol.6 (Suppl 1)

[4] Karin Verspoor, Judith Cohn, Cliff Joslyn, Sue Mniszewski, Andreas Rectsteiner, Luis M. Rocha & Tiago Simas.(2005) *Protein annotation as term categorization in the gene ontology using word proximity networks.* BMC Bioinformatics Vol.6 (Suppl. 1)

[5] Francisco M. Couto, Mario J. Silva & Pedro M. Coutinho. (2005) *Finding genomic ontology terms in text using evidence content.* BMC Bioinformatics Vol. 6 (Suppl 1)

[6] Fayyad, U., G. Piatetsky-Shapiro y P. Smyth. , (1996) *Data Mining and Knowledge Discovery in Databases: An overview.* Communications of ACM, 39:11.

[7] Carmona-Saez P, Chagoyen M, Tirado F, Carazo JM, Pascual-Montano A. (2007) *GENECODIS: A web-based tool for finding significant concurrent annotations in gene lists.* Genome Biology 8(1):R3

[8] Porter M F. (1980) *An algorithm for suffix stripping.* Program, 14 no. 3, pp 130-137.

[9] Francis, W. N. and Kucera, H. (1979) *Brown Corpus Manual.*

[10] Charniak, E. (1993) *Statistical Language Learning.* MIT Press.

[11] Dermatas, E. and Kokkinakis, G. (1995) *Automatic Stochastic Tagging of Natural Language Texts.* Computational Linguistics, 21 (2).

[12] Kupiec, J. (1992) *Robust Part-of-Speech Tagging Using a Hidden Markov Model.* Computer Speech and Language, 6.

[13] Marcus, M. *The Penn Treebank Project.* http://www.cis.upenn.edu/~treebank.

[14] Brill, E. (1992) *A Simple Rule-Based Part of Speech Tagger.* In Proceedigns of the Third Annual Conference on Applied Natural Language Processing, ACL.

[15] Brill, E. (1999) *Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging.* Kluwer Academic Publishers.

[16] Greene, B. B. and Rubin, G. M. (1972) *Automatic Grammatical Tagging of English.* Technical report, Brown University, Providence, RI.

[17] Maltese, G. and Mancini, F. (1991) *A Technique to Automatically Assign Parts-of-Speech to Words Taking into Account Word-Ending Information through a Probabilistic Model.* Proceedings of Eurospeech-91, pp. 753-756.

[18] Schutze, H. (1993) *Part-of-Speech Induction from Scratch.* Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 251 - 258.

[19] Nasukawa, T.; Nagano, T. (2001). *Text analysis and knowledge mining system.* IBM Systems Journal, knowledge management. Vol. 40 (4).

[20] Frawley, W. J. et al. (1991). *Knowledge Discovery in Databases: An Overview.* MIT Press.

[21] Sahami, M. et al. (1996). *Applying the Multiple Cause Mixture Model to Text Categorization.* Proceedings of the Thirteenth International Conference on Machine Learning.

[22] Goldszmidt, M. and Sahami, M. (1998). *A Probabilistic Approach to Full-Text Document Clustering.* Technical report ITAD-433-MS-98-044, SRI Int.

[23] Lent, B. et al. (1997). *Discovering Trends in Text Databases.* Proceedings of the 3rd Internacional Conference on Knowledge Discovery and Data Mining (KDD-97).

[24] Rajman, M. and Besancon, R. (1997). *Text Mining: Natural Language Techniques and Text Mining Applications.* Proceedings of the 7th IFIP 2.6 Working Conference on Database Semantics, DS-7. Chapam & Hall.

[25] Aumann, Y. et al. (1999). *Circle Graphs: New Visualization Tools for Text-Mining.* Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97), pp. 277 - 282.

[26] Stapley, B. J. and Benoit, G. (2000). *Bibliometrics: Information Retrieval and Visualization from Co-Occurrences of Gene Names in Medline Abstracts* Proceedings of the Pacific Symposium on Biocomputing (PSB), pp. 526 - 537.

[27] Hearst, M. A. (1999). *Untangling Text Data Mining.* Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 3-10, Maryland.

[28] Swanson, D. R. and Smalhaiser, N. R. (1994). *Assessing a gap in the biomedical literature: magnesium deficiency and neurologic disease.* Neuroscience research communications. Vol. 15, pág. 1-9.

[29] van Rijsbergen, C. J. (1979). *Information Retrieval.* Butterworth, London.

[30] Salton G. and McGill (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill.

[31] Baeza-Yates, R. and Ribeiro-Neto, B. (1999) *Modern Information Retrieval* Addison Wesley

[32] Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer.* Reading (MA): Addison-Wesley.

[33] Witten, I. H. et al. (1999). *Managing Gigabytes, Compressing and Indexing Documents and Images (2 edition).* Morgan-Kaufmann.

[34] Wilbur, W. J. and Yang, Y. (1996). *An Analysis of Statistical Term Strength and its Use in the Indexing and Retrieval of Molecular Biology Text.* Computers in Biology and Medicine, 26 (3), 209 - 222.

[35] Robertson, S. E. and Spark Jones, K. (1979) *Relevance weighting of search terms.* Journal of the American Society for Information Sciencies, 27(3) pp. 129-146.

[36] Ponte, J. M. and Croft, W. B. (1998). *A Language Modeling Approach to Information Retrieval.* Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR-98).

[37] Hofmann, T. (1999). *Probabilistic Latent Semantic Indexing.* Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR-99).

[38] Shatkay, H. et al. (2000). *Genes, Themes and Microarrays: Using Information Retrieval for Large Scale Gene Analysis.* Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB), pp. 317 - 328.

[39] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas and Richard A. Harshman (1990) *Indexing by Latent Semantic Analysis.* Journal of the American Society of Information Science, volume 41(6) pp. 391 - 407

[40] Wilkinson, R. and Hingston, P. (1991) *Using the cosine measure in a neural network for document retrieval.* Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 202-210.

[41] Ross, B.H (1989) *Some psychological results on casebased reasoning Case-Based* Reasoning Workshop , DARPA. Morgan-Kaufmann. pp. 144-147.

[42] Aamodt, A. and Plaza, E. (1994) *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches* AI Communications, Vol. 7 (1) pp. 39-59

[43] Lenz, M. and Burkhard, H. D. (1997) *CBR for Document Retrieval - The FallQ Project.* Lecture Notes in Artificial Intelligence, 1266. Springer Verlag, pp. 84-93.

[44] Daniels, J. J. and Rissland, E. L. (1997). *What You Saw Is What You Want: Using Cases to Seed Information.* Lecture Notes in Artificial Intelligence, 1266. Springer Verlag. pp. 325-336.

[45] Hayes, P. (1992). *Intelligent High-Volume Processing Using Shallow, Domain-Specific* Techniques in Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval, pp. 227-242.

[46] Hayes, P. and Weinstein, S. (1990). *CONSTRUE: A System for Content-Based Indexing of a Database of News Stories.* Proceedings of the 2nd Annual Conference on Innovative Applications of Artificial Intelligence.

[47] Cohen, W. W. and Singer, Y. (1999). *Context-Sensitive Learning Methods for Text Categorization.* ACM Transaction on Information Systems, Vol. 17 (2), pp. 141-173.

[48] Dumais, S. T. et al. (1998). *Inductive Learning Algorithms and Representations for Text Categorization.* Proceedings of the 7th Internacional Conference on Information and Knowledge Management (CIKM-98), pp. 148 - 155.

[49] Joachims, T. (1998). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features.* Proceedings of the European Conference on Machine Learning (ECML-98).

[50] Sebastiani, F. (2002). *Machine Learning in Automated Text Categorization.* ACM Computing Surveys, Vol. 34 (1), pp. 1-47.

[51] Cowie, J. and Wilks, Y. (1996) *Information Extraction.* Communications of the ACM

[52] Shatkay, H. and Feldman, R. (2003). *Mining the biomedical literature in the genomic era: An overview.* Journal of Computational Biology Vol. 10, 6, pp. 821-856.

[53] Mitkov, R. (1998). *Robust Pronoun Resolution with Limited Knowledge.* In COLING-ACL, pp. 869 - 875.

[54] Eisen, M.B., Spellman, P., Brown, P.O., and Botstein, D. (1998) *Cluster Analysis and Display of Genome-Wide Expression Patterns.* Proceedings of the National Academy of Sciences, 95(25): 14863–14868.

[55] Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR. (1999) *Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.* Proc Natl Acad Sci USA 1999, 96:2907-2912.

[56] Toronen P, Kolehmainen M, Wong G, Castren E. (1999). *Analysis of gene expression data using self-organizing maps.* FEBS Lett 1999, 451:142-146.

[57] Diaz-Uriarte, R., Álvarez de Andrés, S. (2006). *Gene selection and classification of microarray data using random forest.* Bmc Bioinformatics 6, 3-3.

[58] Thanh-Nghi Do and Poulet F. (2003) *Incremental SVM and Visualization Tools for Biomedical Data Mining*. Proceeding of the Europen workshop on data mining and text mining for bioinformatics (2003).

[59] Lin, S. Patel and A. Duncan (2003). *Using decision trees and support vector machines to classify genes by names.*. Proceeding of the Europen workshop on data mining and text mining for bioinformatics (2003).

[60] Lewis, D. D. (1997). *Test Collections: Reuters-21578.*. http://www.daviddlewis.com/resources/testcollections/reuters21578.

[61] Francis, W. N. and Kucera, H. (1979). *Brown Corpus Manual.*http://www.hit.uib.no/icame/brown/bcm.html.

[62] Voorhees, E. and Harman, D. K. (1993). *Text Retrieval Conference (TREC).* http://trec.nist.gov.

[63] Camon, E. B. and Barrell, D. G. and Dimmer, E. C. and Lee, V. and Magrane, M. and Maslen, J. and Binns, D. and Apweiler, R (2005) *An evaluation of GO annotation retrieval for BioCreAtIvE and GOA.* BMC Bioinformatics vol 6, suppl 1.

[64] Hermjakob, H., L. Montecchi-Palazzi, et al. (2004). *IntAct: an open source molecular interaction database.* Nucleic Acids Res 32(Database issue): D452-5.

[65] Zanzoni, A., L. Montecchi-Palazzi, et al. (2002). *MINT: a Molecular INTeraction database.* FEBS Lett 513(1): 135-40.

[66] Khatri P, Draghici S, Ostermeier GC, Krawetz SA (2002). *Profiling gene expression using onto-express.* Genomics 2002, 79:266-270.

[67] Khatri P, Draghici S (2005). *Ontological analysis of gene expression data: current tools, limitations, and open problems.* Bioinformatics 2005, 21:3587-3595.

[68] Draghici S (2003).*Data Analysis Tools for DNA Microarrays.* Chapman and Hall/CRC Press.

[69] Berriz GF, King OD, Bryant B, Sander C, Roth FP (2003). *Characterizing gene sets with FuncAssociate.* Bioinformatics 2003, 19:2502-2504.

[70] Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, Sherlock G. (2004) *GO::TermFinder - open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes.* Bioinformatics 2004, 20:3710-3715.

[71] Agrawal, R., Imielinski, T., Swami A. (1993) *Mining Association Rules between Sets of Items in Large Databases.* Proceedings of the ACM SIGMOD international conference on Management of data, pp. 207-216.

[72] Bnejamini, Y. and Hochberg, Y. (1995). *Controlling the false discovery rate: A practical and powerful approach to multiple testing.* J. royal Stat. Soc. B 57: 289-300

[73] M. Koegl, T. Hoppe, S. Schlenker, H. Ulrich, T. Mayer, S. Jentsch (2003) *A Novel Ubiquitination Factor, E4, Is Involved in Multiubiquitin Chain Assembly.* Cell, Volume 96, Issue 5, Pages 635-644

[74] Yu Jiang and James R. Broach (1999) *Tor proteins and protein phosphatase 2A reciprocally regulate Tap42 in controlling cell growth in yeast* The EMBO Journal 18, 2782-2792.

[75] Sharan, R. and Shamir, R. (2000). *CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis.* Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB), pp. 307-316.

[76] www.ncbi.nlm.nih.gov/sites/entrez

[77] Pruitt, K. D. and Maglott, D. R. (2001). *RefSeq and LocusLink: NCBI Gene-Centered Resources.* Nucleic Acids Research, 29 (1), 137-140. http://www.ncbi.nlm.nih.gov/LocusLink.

[78] Boeckmann, B. et al. (2003). *The SWISS-PROT Protein Knowledgebase and its Supplement TrEMBL in 2003.* Nucleic Acids Research, 31 (1), 365-370. http://www.expasy.org/sprot/.

[79] HUGO (2003). *HUGO (The Human Genome Organization) Gene Nomenclature Committee.* http://www.gene.ucl.ac.uk/nomenclature.

[80] NLM (2003). *Mesh: Medical Subject Headings.* http://www.nlm.nih.gov/mesh/.

[81] Lindberg, D. A. et al. (1993). *The Unified Medical Language System.* Meth. Inform. Med.,32 (4), 281-291. http://www.nlm.nih.gov/research/umls.

[82] Leek, T. R. (1997). *Information Extraction Using Hidden Markov Models.* Master's thesis, Department of Computer Science, University of California, San Diego.

[83] Craven, M. and Kumlien, J. (1999). *Constructing Biological Knowledge Bases by Extracting Information from Text Sources* . In Proc. of the AAAI Conf. on Intelligent Systems for Molecular Biology (ISMB), pp. 77-86.

[84] Ray, S. and Craven, M. (2001). *Representing Sentence Structure in Hidden Markov Models for Information Extraction.* In Proc. of the Int. Joint Conf. on Arti cial Intelligence (IJCAI-01).

[85] Blaschke, C. et al. (1999). *Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions.* In Proc. of the AAAI Conf. on Intelligent Systems for Molecular Biology (ISMB), pp. 60-67.

[86] Blaschke, C. and Valencia, A. (2002). *The Frame-Based Module of the SUISEKI Information Extraction System.* IEEE Intelligent Systems, Special Issue on Intelligent Systems in Biology, 17 (2), 14-20.

[87] Jenssen, T.-K. et al. (2001). *A Literature Network of Human Genes for High-Throughput Analysis of Gene Expression.* Nature Genetics, 28, 21-28.

[88] Tanabe, L. et al. (1999). *MedMiner: An Internet Text-Mining Tool for Biomedical Information, with Application to Gene Expression Profiling.* BioTechniques, 27 (6), 1210-1217.

[89] Yakushiji, A. et al. (2001). *Event Extraction from Biomedical Papers Using a Full Parser.* In Proc. of the Paci c Symposium on Biocomputing (PSB), pp. 408-419.

[90] Pustejovsky, J. et al. (2002). *Robust Relational Parsing over Biomedical Literature: Extracting Inhibit Relations.* In Proc. of the Paci c Symposium on Biocomputing (PSB), pp. 362-373.

[91] Swanson, D. R. (1986). *Fish-oil, Raynaud's Syndrome and Undiscovered Public Knowledge.* Perspectives in Biology and Medicine, 30 (1), 7-18.

[92] Swanson, D. R. (1988). *Migraine and Magnesium: Eleven Neglected Connections.* Perspectives in Biology and Medicine, 31 (4), 526-557.

[93] Swanson, D. R. (1990). *Somatomedin C and Arginine: Implicit Connections Between Mutually Isolated Literatures.* Perspectives in Biology and Medicine, 33 (2), 157-186.

[94] Fukuda, K. et al. (1998). *Toward Information Extraction: Identifying Protein Names from Biological Papers.* In Proc. of the Pacific Symposium on Biocomputing (PSB), pp. 705-716.

[95] Hanisch, D. et al. (2003). *Playing Biology's Name Game: Identifying Protein Names in Scientific Text.* In Proc. of the Paci c Symposium on Biocomputing (PSB), pp. 403-411.

[96] Renner, A. and Aszodi, A. (2000). *High-throughput Functional Annotation of Novel Gene Products Using Document Clustering.* In Proc. of the Pacific Symposium on Biocomputing (PSB).

[97] Iliopoulos, I. et al. (2001). *TEXTQUEST: Document Clustering of Medline Abstracts for Concept Discovery in Molecular Biology.* In Proc. of the Pacific Symposium on Biocomputing (PSB), pp. 384-395.

[98] Marcotte, E. M. et al. (2001). *Mining Literature for Protein-Protein Interactions.* Bioinformatics, 17 (4), 359-363.

[99] Stephens, M. et al. (2001). *Detecting Gene Relations from Medline Abstracts.* In Proc. of the Pacfic Symposium on Biocomputing (PSB), pp. 483-496.

[100] Donaldson, I. et al. (2003). *PreBind and Textomy - Mining the Biomedical Literature for Protein-Protein Interactions using a Support Vector Machine.* BMC (BioMed Central) Bioinformatics, 4 (11). http://www.biomedcentral.com/1471-2105/4/11.

[101] Chagoyen M, Carmona-Saez P, Shatkay H, Carazo JM and Pascual-Montano A. *Discovering semantic features in the literature: a foundation for building functional associations.* BMC Bioinformatics. 2006; 7: 41.

[102] Lee DD, Seung HS *Learning the parts of objects by non-negative matrix factorization.* Nature 1999, 401:788-791.

[103] Kim PM, Tidor B. *Subsystem identification through dimensionality reduction of large-scale gene expression data.* Genome Res 2003, 13:1706-1718.

[104] Brunet JP, Tamayo P, Golub TR, Mesirov JP. *Metagenes and molecular pattern discovery using matrix factorization.* Proc Natl Acad Sci U S A 2004, 101:4164-4169.

[105] Heger A, Holm L. *Sensitive pattern discovery with 'fuzzy' alignments of distantly related proteins.* Bioinformatics 2003, 19 Suppl 1:i130-i137.

[106] Pehkonen P, Wong G, Toronen P *Theme discovery from gene lists for identification and viewing of multiple functional groups.* BMC Bioinformatics 2005, 6:162.