VNIVERSIDAD NACIONAL AVTÓNOMA DE MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

IDENTIFICACIÓN *IN SILICO* DE OPERONES EN GENOMAS BACTERIANOS BASADA EN REDES NEURONALES

T E S I S

QUE PARA OBTENER EL GRADO DE:

DOCTORA EN CIENCIAS (COMPUTACIÓN)

P R E S E N T A

M.T.I. BLANCA ITZELT TABOADA RAMÍREZ

DIRECTOR DE TESIS: DR. ENRIQUE MERINO PÉREZ

CO-DIRECTORA DE TESIS: DRA. CRISTINA VERDE RODARTE

CIUDAD UNIVERSITARIA, MÉXICO D.F. 2012

Agradecimientos

A Armando, por ser no sólo mi esposo, sino mi mejor amigo. Le agradezco todo su amor, confianza, paciencia, cariño, risas, ternura, y motivación en todo momento. Pero sobre todo, por haberme enseñado a confiar y a soñar. Te amo.

A Isabella porque a pesar de que aún no naces me has llenado de felicidad e ilusión.

A mi madre quien siempre ha estado a mi lado. Le agradezco sus sacrificios, preocupaciones, regaños, y alegrías y por todo el amor y apoyo que me ha brindado a pesar de mis fallas. Muchas gracias madre.

A una mujer admirable, mi abuelita Eufrosina. Gracias por el apoyo y cariño que me ha brindado durante toda mi vida.

A mi hermano Alberto, por su amor, cariño y complicidad y a toda mi familia. Gracias por estar siempre ahí.

A Enrique Merino Pérez, mi director de tesis, por la asesoría, la enseñanza, el tiempo y confianza brindada para la realización de la presente tesis. Pero principalmente, porque más que un jefe siempre ha sido un amigo, gracias por tu apoyo incondicional, siempre encuentro algo nuevo que aprender de ti.

A Cristina Verde Rodarte, mi co-directora de tesis, por darme la oportunidad y confianza, por su ayuda y paciencia a lo largo de este difícil camino de plasmar en papel lo que se hace en la práctica y por los gratos momentos.

A los miembros de mi comité y jurado doctoral: Dr. Pedro Miramontes Vidal, Dra. Katya Rodríguez Vázquez, Dr. Christopher Stephens Stevens y Dr. Edgar Emmanuel Vallejo Clemente, por todo el apoyo que me brindaron siempre, por su inversión de tiempo y por sus valiosos comentarios y aportaciones.

Al personal administrativo del Posgrado en Ciencias Computacionales que siempre me ha brindado toda la ayuda en la realización de trámites y solicitudes. Gracias especialmente a Lulú y Diana por siempre estar en la mejor disposición.

Al Centro de Ciencias Aplicadas y Desarrollo Tecnológico-UNAM y al Instituto de Biotecnología-UNAM por la oportunidad y facilidades brindadas para el desarrollo de esta tesis, especialmente a todos mis compañeros de los laboratorios de Enrique Merino y Enrique Morett. En particular, gracias a Ricardo Ciria Merce, Leticia Olvera Rodriguez y a Leticia Vega Alvarado por compartir sus conocimientos.

Por supuesto, agradezco a todos mis amigos y compañeros que han estado junto a mí en los momentos buenos y malos durante esta fase de mi vida. En especial a Katy Juárez, Leticia Vega, Leopoldo Ruiz, Alberto Caballero, Rosario Colin, Maribel Acosta, Francisco Santana, Rosa María Gutiérrez, Alfredo Mendoza, Leticia Olvera, Maricela Olvera, Josué Reyes, Maria Soledad Córdova, Maria Luisa Tabche, Sonia Davila y Enrique Morett, muchas gracias por su amistad, apoyo y cariño.

Para Isabella y Armando. Porque me han enseñado a soñar.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Un operón es un conjunto de uno o más genes contiguos en la misma cadena del DNA que se expresan coordinadamente como una sola unidad de transcripción en respuesta a estímulos intracelulares y medioambientales. Debido a su relevancia biológica para coordinar la función o metabolismo de genes relacionados en organismos bacterianos, en los últimos años, se han desarrollado diversos métodos computacionales que utilizan varias características genómicas para el reconocimiento de operones, en el conjunto creciente de genomas completamente secuenciados. Hasta el 2009, el método que había reportado el mejor desempeño alcanzaba una precisión del 93.7 % en la bacteria de *Escherichia coli*, cuando dicho método utilizaba datos de la misma bacteria en el proceso de entrenamiento. Sin embargo, cuando el método era empleado para predecir operones en *Basilus subtilis*, se observaba una disminución del 11 % en su precisión. Reducciones más significativas, del 11 % al 30 %, han sido reportadas por la mayoría de los métodos cuando estos fueron empleados en datos de genomas diferentes a los cuales fueron entrenados, haciéndolos costosos y poco eficaces en términos de su comprobación experimental.

El motivo del presente proyecto de investigación es el de desarrollar un nuevo método computacional para identificar operones que tenga una mayor precisión a la de los métodos previamente reportados y que tenga la capacidad de ser utilizado en diversos genomas bacterianos sin que su precisión disminuya significativamente. Con tal fin, se realizó una selección de las características genómicas más relevantes, obteniendo un impacto positivo en la exactitud de la identificación de operones. Cabe mencionar, que anteriormente no se había realizado un análisis cuantitativo de la contribución relativa de las características comúnmente utilizadas en la identificación de operones, por lo que se considera que el estudio pionero realizado en esta rama, tiene un impacto positivo en la comprensión de la biogénesis de las unidades de transcripción de las bacterias. La estrategia general que se emplea en el análisis está basada en el uso de Redes Neuronales Perceptrón Multicapa (MLP, por sus siglas en inglés *Multilayer Perceptron*). El desempeño de la metodología propuesta se evaluó en varios conjuntos de operones experimentalmente validados en diferentes bacterias obteniendo siempre una mayor precisión que los métodos anteriormente reportados. Asimismo, con la finalidad de probar la generalización, se probó en genomas diferentes a los utilizados en su entrenamiento, obteniendo solamente decrementos de alrededor de 1 % en la precisión.

ABSTRACT

Operons can be defined as a gene or set of genes arranged contiguously on the same transcriptional strand of a genome sequence, which are co-transcribed in the same transcription unit. Due to the biological relevance of operons for coordinating the expression of metabolically or functionally related genes in bacterial organisms, different computational protocols, which have used different genomic characteristics, have been devised for identifying them in the fast growing set of fully-sequenced genomes. Until 2009, the method that had reported the highest performance has a 93.7 % accuracy when used to identify operons in *E. coli* bacteria using data from the same bacteria for training. However, when it was used in *B. subtilis*, there was an accuracy reduction of 11 %. More significant accuracy reductions, from 11 % to 30 %, have been observed with most of the published algorithms when the training data, and the operon predictions, corresponded to different organisms, making them expensive and ineffective in terms of its experimental verification.

The above mentioned results motivated our research project, which is aimed to the development of a new computational method for the *in silico* identification of operons. Our method has the highest accuracy ever obtained, in addition, it can be used to predict operons in various bacterial genomes without significant reduction in precision. Furthermore, we determined the most important genomic features used in the process of operons identification, which have a positive impact on the accuracy of the operon predictions. It should be noted that our study is the first one that quantitative evaluated the relative contribution of the commonly used features in the operon identification, which helps to understand the biogenesis of bacteria transcription units. The overall methodology used in our study was based on the use of Multilayer Perceptron Neural Networks (MLP). The performance of the proposed methodology was evaluated in several sets of experimentally validated operons in different bacteria, always getting better accuracy than the ones previously reported by other methods. Also, in order to prove its generalization our method was tested in different bacterial genomes other than those used in the training procedure, getting only a slightly decreases of around 1 %.

CAPÍTULO 1

INTRODUCCIÓN

La última década ha sido testigo del despertar de una nueva era en la Biología basada en el uso de las computadoras, generando la posibilidad de realizar investigación y análisis comparativo de genomas completos. Esto es de gran importancia en proyectos de secuenciación a gran escala como *el Proyecto Genoma Humano* (Cavalli-Sforza, 2006). En consecuencia, la información contenida en las bases de datos de secuencias ha tenido un crecimiento exponencial, dando origen a un inmenso volumen de datos imposible de analizar y manipular sin el uso de herramientas formales de computación que permitan, además, generar conocimiento biológico a partir de estos datos de secuencias.

El análisis genómico no se refiere únicamente a la implementación de herramientas que permitan el acceso, uso y manejo de varios tipos de información, sino que incluye el desarrollo de sistemas computacionales más complejos basados en el uso de herramientas tales como teoría estadística, matemáticas, teoría de aprendizaje, modelos matemáticos, clasificadores y tecnologías de la información, por mencionar sólo algunos. Estas herramientas pueden ser utilizadas para realizar diversos tipos de análisis, que pueden incluir desde el alineamiento y búsqueda de patrones en secuencias, hasta el alineamiento estructural de proteínas, modelado de la evolución, integración y ensamblado de mapas genéticos y predicciones a diversos niveles como genes, estructuras de proteínas, promotores y operones entre otros, representando cada uno de estos una amplia área de estudio. El objetivo final de los anteriores tipos de análisis es el generar nuevo conocimiento biológico y de proveer los medios para obtener respuestas a preguntas de inmensa relevancia en las ciencias de la vida. Sin embargo, esto no significa que de forma automática se conozca todos los detalles del funcionamiento de los seres vivos. De hecho, se está solamente al principio del entendimiento de cómo la regulación fina y sincronizada del genoma permite el desarrollo y funcionamiento de los organismos. Por otra parte, cuando se desea plantear una metodología haciendo referencia a campos de estudios interdisciplinarios, tal es el caso de la biología y la computación, se hace necesario poner en contexto el planteamiento del problema, en este caso, los operones bacterianos.

En genomas bacterianos, los operones se pueden definir como uno o un conjunto de genes contiguos dispuestos en la misma cadena de DNA que se expresan coordinadamente como una sola unidad de transcripción en respuesta a estímulos intracelulares y medioambientales. El conocer los operones en un genoma determinado es de gran importancia debido a la relevancia biológica de éstos para coordinar la función o metabolismo de genes relacionados. En principio, los operones podrían ser detectados si se conocieran los patrones de caracteres que delimitan el inicio (promotor) y el final de la transcripción (terminator) (Dam *et al*., 2007; Tjaden *et al*., 2002; Yada *et al*., 1999). No obstante los esfuerzos de distintos grupos de investigación, la capacidad actual de reconocimiento e identificación de estos patrones de manera precisa es limitada debido a que las secuencias de caracteres que los identifican son variantes e inciertas y se desconoce de manera precisa sus reglas de sintaxis. En este sentido, se han desarrollado varios métodos computacionales (descritos en detalle en la sección 2.2) para la identificación de operones basados en distintos algoritmos de reconocimiento de patrones y que han utilizado, de manera individual o en subconjunto, varias características genómicas disponibles, entre las que destacan: a) Dirección de transcripción, b) Distancias intergénicas, c) Vías metabólicas, d) Relaciones funciones, e) Conservación de vecindad y f) Perfiles filogenéticos, entre otras (Dam *et al*., 2007; Tran *et al*., 2007; Bergman *et al*., 2007; Zhang *et al*., 2006; Westover *et al*., 2005; Price *et al*., 2005; Jacob *et al*., 2005; Edwards *et al*., 2005; Chen *et al*., 2004a,b; Romero and Karp, 2004; De Hoon *et al*., 2004; Bockhorst *et al*., 2003; Salgado *et al*., 2000).

A pesar de la gran variedad de métodos desarrollados y características utilizadas, hasta el año de 2009, no existía un identificador de operones *in silico* eficiente que además, pudiera ser utilizado en diversos genomas bacterianos sin una pérdida significativa de precisión. Por ejemplo, el trabajo que mejor desempeño había reportado, fue el de Dam *et al*. (2007), el cual tenía una precisión de 93 % en la bacteria de *E. coli*. Sin embargo, cuando dicha metodología se probó con datos de *B. subtilis* su precisión disminuyó al $83\,\%$. Asimismo, no existía un trabajo donde se hubiera evaluado la importancia relativa de las características utilizadas para identificar operones, a pesar de que se sabía que una buena selección de éstas, generalmente resulta en una mejora en la precisión porque la información puede contener características ruidosas, redundantes o no importantes. Además, un proceso de selección de características por lo general proporciona otros beneficios como disminución en los tiempos de procesamiento de los datos, menor requerimiento en los espacios donde se almacena la información, mayor facilidad en el análisis y visualización de los datos, y lo más importante seleccionar el subconjunto de características relevantes que aportan la mayor cantidad de información para un problema en particular.

Lo anteriormente expuesto motivó el presente proyecto de investigación, donde se desarrolló un nuevo método computacional para la identificación de operones que tuviera una muy alta precisión, tanto en los genomas de donde se obtuvieron los datos para el entrenamiento, como para otros diversos genomas bacterianos, sin que por ello se disminuyera significativamente su precisión. Además, se planteó evaluar la contribución relativa de las características comúnmente utilizadas en la identificación de operones, lo cual ayuda a la comprensión de la biogénesis de las unidades de transcripción de las bacterias. Todo lo anterior, fue implementado mediante el uso de Redes Neuronales Perceptrón Multicapa (MLP, por sus siglas en inglés *Multilayer Perceptron*).

1.1. Objetivos

En esta tesis se plantearon los siguientes objetivos.

i) Desarrollar una metodología computacional universal para la identificación de operones que pueda ser utilizada en cualquier genoma bacteriano, independientemente que el proceso de aprendizaje del método incluya, o no, información de dichos genomas.

ii) Desarrollar una metodología para seleccionar el subconjunto de características relevantes para la predicción *in silico* de operones, de tal forma que se pueda entender mejor la naturaleza biológica que determinan su arquitectura.

iii) Generar una nueva característica genómica para identificar operones que no puedan ser caracterizados por ninguno de los métodos propuestos en los objetivos i y ii.

1.2. Metodología

Para alcanzar los objetivos propuestos en este proyecto, se planteó la siguiente metodología:

a) Como primer instancia para identificar un par de genes como *operón* o *no-operón*, se propuso el uso de un conjunto patrón formado por tres atributos obtenidos a partir de las relaciones entre características individuales de pares de genes. Los componentes considerados fueron: i) La dirección de transcripción de los genes, ii) La distancia intergénica entre los mismos y iii) La relación funcional que guardan las proteína codificadas por dichos genes y que había sido previamente definida en la Base de Datos (BD) llamada STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) (Jensen *et al*., 2009). Por lo tanto, el problema se definió como la asignación del conjunto patrón, mediante una red MLP, al espacio objetivo de la clase *operón* o *no-operón* con

una tasa de error baja. Cabe mencionar, que STRING es una BD cuidadosamente curada que contiene una recopilación extensa de datos para obtener la relación de distintos grupos de genes ortólogos (genes con la misma función en diferentes organismos originados por un proceso de especiación), ponderando en un sólo valor siete distintas características genómicas, algunas de las cuales han sido usadas de manera individual o en subconjunto en la identificación de operones. Sin embargo, se desconoce que dicha BD haya sido utilizada para la identificación de operones, por lo que su uso es una aportación en este trabajo de investigación. Al utilizar información de STRING, los genes considerados para el análisis fueron sólo aquellos que tienen asignado un grupo de genes ortólogos.

b) Para evaluar los beneficios de un proceso de selección de características y conocer cual es el subconjunto relevante para la identificación de operones, se propuso acoplar el método *Weight Explanatory* (WE) (Gevrey *et al*., 2003) de una red MLP para determinar la contribución relativa de cada una de éstas. Por lo tanto, ahora el objetivo fue minimizar la cardinalidad del espacio de características totales por medio del método WE, tal que existió otra red MLP que asignó el subconjunto seleccionado al espacio objetivo de *operones* y *no-operones*, con una tasa de error baja. Con ésto, se eliminaron los atributos menos influyentes que podían inducir a error (características ruidosas), no aportar mayor información (características irrelevantes), o incluir la misma información que otras (características redundantes) (Blum and Langley, 1997), al mismo tiempo que se mantenía la precisión alcanzada en la identificación de operones. Al igual que en el método anterior, al utilizar STRING, los genes considerados para el análisis fueron sólo aquellos que tienen asignado un grupo de genes ortólogos.

c) Se generó una nueva característica STRING-like basada en la conservación de vecindad y aproximada a los valores ponderados de STRING, la cual permitió analizar genes que no tenían asignado un grupo de ortólogos dentro de la base de datos STRING. Ésto debido a que con los métodos descritos en los puntos anteriores (1 y 2) sólo se pueden analizar aquellos genes que son elementos de un conjunto extenso de genes ortólogos. Sin embargo, con esta característica STRING-like, se logra discretizar aquellos genes que no pertenecen a un grupo de ortólogos, de tal forma que se pueda reconocer el conjunto total de operones de un genoma determinado.

1.3. Trabajos publicados

Como resultado de este proyecto de investigación se han generado los siguientes artículos de investigación que han sido publicados en revistas indexadas de circulación internacional:

- Taboada Blanca, Verde Cristina, Merino Enirque, 2010, High accuracy operon prediction method based on STRING database scores, *Nucleic Acids Research*, 38(12), e130, ISSN:0305-1048.
- Taboada B., Ciria R., Martinez-Guerrero C.E., Merino,E., 2012, ProOpDB: Prokaryotic Operon DataBase, *Nucleic Acids Research*, 40(D1), D627-D631, ISSN:0305-1048.

Asimismo, los resultados del presente proyecto fueron presentados en los siguientes congresos nacionales e internacionales:

- Taboada B., Merino E., Verde C., 2010, Evaluation of the relative contribution of each STRING feature in the overall accuracy operon classification, *1st International Congress on Instrumentation and Applied Sciences ICIAS* (Incorporating the 25th National Congress on Instrumentation), Sociedad Mexicana de Instrumentación, Cancún, Quintana Roo, México, 26 - 29 de octubre.
- Taboada Ramírez B., Verde Rodarte C., Merino Pérez E., 2009, Metodología computacional para la predicción de operones en procariontes, *XXIV Congreso Nacional de instrumentación*, Sociedad Mexicana de Instrumentación, Mérida, Yucatán, 16-21 de Octubre.
- Taboada Ramírez B., Verde Rodarte C., Merino Pérez E., 2010, Metodología computacional para la predicción de operones basada en las distancias intergénicas y los scores de la base de datos de STRING, *XXVIII Congreso Nacional de Bioquímica*, Sociedad Mexicana de Bioquímica, Tuxtla Gutiérrez, Chiapas, México, 7 - 12 de noviembre.
- Taboada Ramírez B., Verde Rodarte C., Merino Pérez E., 2008, Predicción computacional de promotores y operones en procariontes, *XXVII Congreso Nacional de Bioquímica*, Sociedad Mexicana de Bioquímica, Merida, Yucatan,16 - 21 de Noviembre.
- Taboada Ramírez B., Verde Rodarte C., Merino Pérez E., 2008, Clasificación computacional de operones en procariontes, *Primer Congreso Mexicano de Inteligencia Artificial (COMIA08)*, Sociedad Mexicana de Inteligencia Artificial, Edo. México, 29 - 30 de Octubre.

Cabe mencionar, que los trabajos “Evaluation of the relative contribution of each STRING feature in the overall accuracy operon classification”, “Metodología computacional para la predicción de operones en procariontes”, y “Clasificación computacional de operones en procariontes” fueron presentados en la respectiva sesión ordinaria del congreso.

1.4. Organización de la tesis

Este trabajo de investigación está organizado de la siguiente forma. En el capítulo 2, se describen algunos conceptos básicos de Biología Molecular relacionados con el proceso de la transcripción y unidades tanscripcionales, así como el estado del arte de los diversos métodos de identificación *in silico* de operones. En el capítulo 4, se define el problema de identificación *in silico* de operones de manera formal. Asimismo, se proporciona el conjunto de datos y medidas de desempeño utilizadas. En el capítulo 5, se describe el método de identificador de operones basado en distancias intergénicas y los valores ponderados de relación funcional de la BD de STRING, así como los resultados obtenidos con dicho método. En el capítulo 6, se detalla el método de identificación de operones basado en el subconjunto de características seleccionadas como relevantes. En el capítulo 6, se describe el procedimiento para estimar y estandarizar la característica utilizada para la identificación de operones cuando los valores de la BD STRING no pueden ser aplicados. Finalmente, en el capítulo 7, se presentan las conclusiones y las perspectivas para un trabajo futuro.

CAPÍTULO 2

MARCO REFERENCIAL DE IDENTIFICACIÓN DE OPERONES

En este capítulo, primero se describen brevemente algunos conceptos de Biología Molecular para entender el objetivo de estudio de este trabajo. Posteriormente, se realiza un análisis de los métodos de identificación *in silico* de operones previamente reportados.

2.1. Preliminares de Biología Molecular

En términos biológicos, el ácido desoxirribonucleico (DNA, por sus siglas en inglés) es el elemento donde se almacena la información genética de cualquier organismo. El DNA está formado por dos largas cadenas helicoidales de bases nitrogenadas que son: adenina (A), guanina (G), citosina (C) y timina (T). Las dos cadenas permanecen juntas por enlaces de hidrógeno entre los pares de bases, respetando una estricta complementariedad: A sólo se aparea con T mediante dos puentes de hidrógeno (y viceversa), y G sólo lo hace con C mediante tres puentes de hidrógeno. De ésto se desprende que una de las hebras de la molécula de DNA es la complementaria de la otra. Los extremos de cada una de las cadenas del DNA se denominan $5'$ y $3'$. Asimismo, las dos cadenas se alinean en forma paralela, pero en direcciones opuestas (una en sentido $5' \rightarrow 3'$ y la complementaria en el sentido inverso) (Fig. 2.1).

La información del DNA es codificada en bloques discretos, llamados genes, que son utilizados para dirigir la síntesis de proteínas. Para que la información de un gen sea "leída" por la célula, la información del DNA es copiada en moléculas de ácido ribonucleico (RNA, por sus siglas en inglés), en un proceso llamado *transcripción*. La transcripción del DNA es el primer proceso de la expresión génica, mediante el cuál se transfiere la información contenida en la secuencia del DNA hacia la secuencia de proteína, utilizando diversos RNA como intermediarios. Durante la transcripción genética, las secuencias de DNA son copiadas a RNA mediante una enzima llamada RNA polimerasa que sintetiza un RNA mensajero que mantiene la información de la secuencia del DNA. De esta manera, la transcripción del DNA también

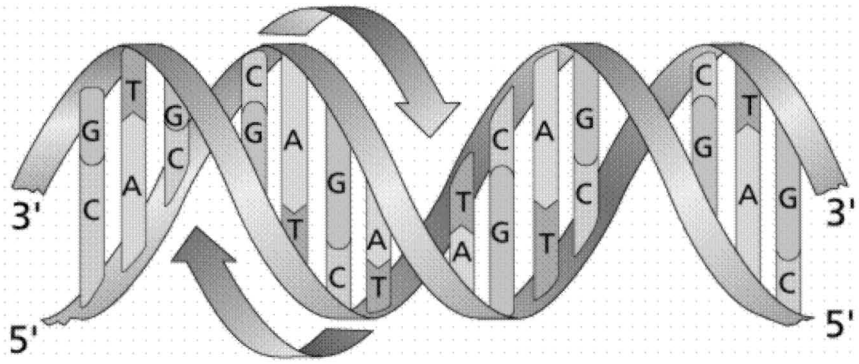

Figura 2.1: Representación gráfica del DNA.

*Imagen obtenida de Purves et al., Life: The Science of Biology , 4rd Edición, por Sinauer Associates y WH Freeman

podría llamarse síntesis del RNA mensajero. En términos computacionales, tanto el DNA como el RNA se pueden ver como una combinación de cuatro caracteres diferentes A, G, C, T (la T cambia a U -uracilo-,en el caso de moléculas de RNA), los cuales conforman un alfabeto que se utiliza para realizar la transcripción del DNA al RNA por medio de un lenguaje determinado. Incluso, el DNA se puede ver como una cadena de bits, donde cada uno de los elementos del alfabeto se representa en código binario, e.g. $A = 00$, $G = 01$, $C = 10$, $T = 11$.

Una vez que se transcribió el DNA a RNA, se realiza la síntesis de proteínas, en un proceso denominado *traducción*. La *traducción* utiliza un código genético que es estándar para la mayoría de los organismos (Ver Tabla 2.1). Éste permite codificar el RNA en una serie de codones o tripletes de tres nucleótidos, constituido por tres bases sucesivas en el RNA, donde cada uno corresponde a un aminoácidos en particular, siendo éstos los componentes básicos de las proteínas. Existen 20 diferentes tipos de aminoácidos de manera natural. La combinación total de tripletess, en un alfabeto de cuatro caracteres, es $4^3 = 64$ posibilidades, de las cuales 61 codones codifican a los 20 aminoácidos del código genético. Las restantes no son codificables sino que se utilizan como señales de terminación de la traducción. Algunos tripletes codifican a un mismo aminoácido (Tabla 2.1).

En resumen, durante el proceso de síntesis de proteínas, se transmite información desde los genes del DNA (la molécula que almacena la información) a RNA (molécula de transferencia de información), y finalmente a una proteína específica (producto funcional y no codificante). Cabe mencionarse que cada uno de estos pasos tiene un complicado sistema de elementos reguladores e interacciones que apenas se han comenzado a comprender.

Aminoácidos	**Codones**					
Alanina (A)	GCA	GCC	GCG	GCU		
Arginina (R)	AGA	AGG	CGA	CGC	CGG	CGU
Aspartato (D)	GAC	GAU				
Asparagina (N)	AAC	AAU				
Cisteína (C)	AGC	UGU				
Glutamato (E)	GAA	GAG				
Glutamina (Q)	CAA	CAG				
Glicina (G)	GGA	GGC	GGG	GGU		
Histidina (H)	CAC	CAU				
Isoleucina (I)	AUA	AUC	AUU			
Leucina (L)	CUA	CUC	CUG	CUU	UUA	UUG
Lisina (K)	AAA	AAG				
Metionina (M)	AUG					
Fenilalanina (F)	UUC	UUU				
Prolina (P)	CCA	CCC	CCG	CCU		
Selenocisteína (U)	UGA					
Serina (S)	AGC	AGU	UCA	UCC	UCG	UCU
Treonina (T)	ACA	ACC	ACG	ACU		
Triptofano (W)	UGG					
Tirosina (Y)	UAC	UAU				
Valina (V)	GUA	GUC	GUG	GUU		
Inicio	AUG					
Término	UAA	UAG	UGA			

Tabla 2.1: Tabla estándar del código genético

2.1.1. Operones

En una célula no todos los genes se transcriben o expresan simultáneamente, sino que la transcripción se da de manera sumamente organizada en respuesta a estímulos intracelulares y medioambientales. En organismos bacterianos, comúnmente, diferentes genes relacionados a una misma función (por ejemplo, formación de un flagelo) o en una misma vía metabólica (por ejemplo, la síntesis de un aminoácido determinado) se transcriben simultáneamente en una unidad llamada *operón*. A los elementos de regulación del DNA, que indican donde deben iniciar y finalizar los diferentes operones se les conoce como promotor y terminador, respectivamente. Se tiene que considerar que los genes de un genoma no están situados siempre de forma contigua en el DNA, sino que existe una cantidad de DNA (variable según los distintos genomas), en donde comúnmente se localizan dichas señales de inicio y término de la transcripción, así como otras regiones intergénicas no-codificantes.

Un buen ejemplo de un operón es el de lactosa de *E. coli*, el primero en ser descrito, que contiene tres genes (*lacZ*, *lacY* y *lacA*) que intervienen en la conversión de lactosa en unidades

de glucosa y galactosa. La glucosa y galactosa son sustratos para la vía de la glucólisis generadora de energía. De modo que la función de los genes del operón de lactosa es convertir lactosa en una forma que pueda ser utilizada por *E. coli* como fuente de energía. Como la lactosa no es un componente común del medio natural de *E. coli*, la mayor parte del tiempo el operón de lactosa no se transcribe. Cuando se dispone de lactosa, el operón se activa y se transcriben juntos los tres genes; lo que determina la síntesis coordinada para la utilización de lactosa. Este es un ejemplo clásico de regulación genética en las bacterias utilizando operones.

Como se mostró con el ejemplo anterior, el conocer todos los operones en un genoma de interés es importante debido a que es el primer paso para examinar el proceso de trascripción de los genes, determinar posibles relaciones funcionales entre los genes co-expresados en dichas unidades transcripcionales y entender la red de regulación de los genes en un genoma de interés particular.

2.2. Estado del arte de identificadores *in silico* de operones

La identificación de operones en un genoma determinado es de gran importancia debido a la relevancia biológica de éstos para coordinar la función o metabolismo de genes relacionados. En principio, los operones podrían ser detectados si se conocieran los patrones de caracteres que delimitan el inicio (promotor) y el final de la transcripción (terminador). Sin embargo, las secuencias de caracteres que los identifican son variables e inciertas y sus reglas de sintaxis no son del todo conocidas, haciendo que su reconocimiento sea limitado. Por lo tanto, en los últimos años se han desarrollado diversos métodos computacionales para la identificación de operones que han utilizado diversas características genómicas y aplicado diversas metodologías computacionales, las cuales son descritas en los siguientes párrafos.

2.2.1. Características

En cuanto a las características genómicas que han sido utilizadas para la identificación de operones destacan (Tabla 2.2):

- ***Dirección de transcripción:*** Esta característica determina la dirección de transcripción de un gen, que siempre es en sentido $5' \rightarrow 3'$, pero los genes pueden estar codificados en cualquiera de las dos hebras del DNA que se orientan de manera antiparalela. Dado que los genes de un operón están dispuestos a lo largo de la misma cadena de transcripción de DNA o dirección, esta característica origina que genes en direcciones opuestas no puedan ser parte del mismo operón.

- ***Distancias intergénicas:*** Esta característica determina la distancia en pares de bases nitrogenadas (pb) entre dos genes adyacentes a partir de sus posiciones izquierda y derecha respectivas correspondientes a la dirección $5' \rightarrow 3'$ del genoma. El valor de la distancia intergénica entre genes contiguos ha sido muy utilizada en los métodos computacionales de identificación de operones, ya que por lo general el tamaño de las regiones intergénicas de pares de genes de un mismo operón suelen ser más pequeñas en comparación con las de los pares de genes de diferentes operones (Dam *et al*., 2007; Tran *et al*., 2007; Bergman *et al*., 2007; Zhang *et al*., 2006; Westover *et al*., 2005; Price *et al*., 2005; Jacob *et al*., 2005; Edwards *et al*., 2005; Chen *et al*., 2004a,b; Romero and Karp, 2004; De Hoon *et al*., 2004; Bockhorst *et al*., 2003; Salgado *et al*., 2000).

- ***Vías metabólicas:*** Se entiende por ruta o vía metabólica la sucesión de reacciones químicas que conducen de un sustrato inicial a uno o varios productos finales, a través de una serie de metabolitos intermediarios. Por ejemplo, la ruta que incluye la secuencia de reacciones: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, A es el substrato inicial, E es el producto final, y B, C, D son los metabólicos intermediarios de la ruta. Las vías metabólicas se han utilizado como característica para identificara operones debido a que se ha comprobado que los genes que pertenecen a un mismo operón suelen estar involucrados en la misma vía (Tran *et al*., 2007; Zhang *et al*., 2006; Jacob *et al*., 2005; Romero and Karp, 2004; Zheng *et al*., 2002).

- ***Relaciones Funciones:*** Es frecuente que los genes de un mismo operón llevan a cabo funciones relacionadas, por ejemplo colaboran en la formación de un flagelo o que sus productos sean miembros de un mismo complejo proteico. Así, genes con una misma función o funciones relacionadas pueden ser considerados como parte de un operón (Dam *et al*., 2007; Tran *et al*., 2007; Westover *et al*., 2005; Price *et al*., 2005; Jacob *et al*., 2005; Chen *et al*., 2004a,b; Romero and Karp, 2004). En particular, en la identificación de operones, se han utilizado distintas clasificaciones funcionales existentes, tales como: Ontología Génica (Gene Ontology, en Inglés, cuya abreviación es GO) que provee un vocabulario controlado que describe el gen y los atributos del producto génico en cualquier organismo (Dam *et al*., 2007; Tran *et al*., 2007), anotación funcional de Riley (Romero and Karp, 2004), agrupamiento de genes en sus correspondientes grupos de ortología COG (por sus siglas en inglés, Cluster of Orthologous Genes) (Price *et al*., 2005; Chen *et al*., 2004a,b), entre otros.

- ***Conservación de vecindad:*** Esta característica se refiere a que si en un genoma, un par de genes (A,B) se encuentran contiguos, sus correspondientes genes ortólogos (A',B') en otro genoma, también serán contiguos. La razón de esta conservación es que existe

una presión selectiva para que dichos genes se transcriban conjuntamente y por lo tanto, formen parte del mismo operón (Dam *et al*., 2007; Bergman *et al*., 2007; Zhang *et al*., 2006; Westover *et al*., 2005; Price *et al*., 2005; Jacob *et al*., 2005; Edwards *et al*., 2005; Chen *et al*., 2004a,b; Ermolaeva *et al*., 2001).

- ***Co-ocurrencia filogenética:*** Esta característica evalúa la tendencia que tiene un par de genes y sus correspondientes ortólogos de estar presentes o ausentes de manera conjunta y no necesariamente contigua en un grupo de genomas para inferir una conexión biológica significativa, como la participación de dos proteínas diferentes en la misma vía metabólica, lo cual es común en genes de un mismo operón (Dam *et al*., 2007; Zhang *et al*., 2006; Westover *et al*., 2005).

- ***Co-expresión:*** No todos los genes de una célula se transcriben simultáneamente, si no que esta transcripción se da de manera sumamente organizada en respuesta a estímulos intracelulares y medioambientales. El número de veces que un gen es transcrito por unidad de tiempo en la célula, se le conoce como nivel de expresión. Varios estudios han utilizado la evaluación del nivel de expresión de genes en experimentos de microarrelgos de DNA para identificar operones (De Hoon *et al*., 2004; Bockhorst *et al*., 2003). Genes que son parte del mismo operón, comúnmente muestran patrones similares de expresión génica, es decir tienden a tener una correlación cercana a uno en su nivel de expresión. Por lo tanto, la correlación de la expresión de genes en múltiples experimentos de microarreglos ha sido usada para predecir las estructuras de los operones.

2.2.2. Métodos computacionales

En cuanto a los métodos de reconocimiento de patrones que han sido utilizados para la identificación de operones (ver Tabla 2.2) sobresalen los Modelos Ocultos de Markov (Dam *et al*., 2007; Bergman *et al*., 2007), Probabilidades Bayesianas (Price *et al*., 2005; Westover *et al*., 2005; De Hoon *et al*., 2004; Bockhorst *et al*., 2003), Redes Neuronales Artificiales (Tran *et al*., 2007; Chen *et al*., 2004a,b), Máquinas de Soporte Vectorial (Zhang *et al*., 2006), Algoritmos Genéticos (Jacob *et al*., 2005), Árboles de Decisión (Dam *et al*., 2007), entre otros. El desempeño de estos métodos de identificación de operones comúnmente se calculó sobre la base de los resultados obtenidos al identificar operones experimentalmente validados de las bacterias de *E. coli* y *B. subtilis*. En los siguientes párrafos, se describen brevemente los trabajos más relevantes.

Dos de los primeros trabajos de identificación de operones fueron los reportados por Salgado *et al*. (2000) y Ermolaeva *et al*. (2001). Dichos trabajos se basaron en algoritmos de

Trabajo	Características usadas										Método
	DI	VM	COG	GO	ORF	CV	PF	CG	NE	MV	
Salgado *et al*. (2000)	√										Max. verosimilitud
Ermolaeva *et al*. (2001)						√					Max. verosimilitud
Sabatti *et al*. (2002)	√								√		Bayesiano
Zheng *et al*. (2002)		√									Bayesiano
Chen *et al*. (2004a,b)	√		√			√			√		Redes Neuronales
De Hoon *et al*. (2004)	√								√		Bayesiano
Romero and Karp (2004)	√	√			√						Max. verosimilitud
Edwards *et al*. (2005)	√					√					Estadístico
Jacob *et al*. (2005)	√	√			√	√					Algoritmo Genético
Westover *et al*. (2005)	√				√	√					Bayesiano
Price *et al*. (2006)	√		√			√				√	Bayesiano
Zhang *et al*. (2006)	√	√			√	√	√				MSV
Bergman *et al*. (2007)	√					√					Markov-Bayesiano
Charaniya *et al*. (2007)	√								√	√	MSV
Tran *et al*. (2007)	√	√		√							
Dam *et al*. (2007)	√			√	√		√			√	Estadístico
Roback *et al*. (2007)	√										Regresión logarítmica
Li *et al*. (2009)	√					√					Teoría de grafos

Tabla 2.2: Propiedades principales de los métodos computacionales de identificación de operones

** DI, distancia intergénica; VI, vias Metabólicas; COG, agrupamiento de genes ortólogos; GO, ontología de genes; ORF, Otras relaciones funciones de proteínas; CV, conservación de vecindad; PF, perfiles filogenéticos; NE, niveles de expresión; MV, Patrones o motivos

máxima verosimilitud, el primero utilizó las distancias intergénicas entre genes como única característica de entrada, mientras el segundo la conservación de vecindad de genes en diversos genomas bacterianos. Una limitante del trabajo de Salgado *et al*. (2000) es que se calcularon las

distancias intergénicas sólo en pares de genes de *E. coli*, asumiéndose distribuciones iguales en todos los genomas bacterianos. Por otra parte, en el trabajo de Ermolaeva *et al*. (2001) se impuso las restricciones de adyacencia inmediata y corte de distancia intergénica de 200 pb en los pares de genes candidatos para el calculo de conservación de vecindad. Sin embargo, está documentado que puede haber inserciones, deleciones o reorganizaciones de genes dentro de un mismo operón (Wolf *et al*., 2001). Ambos trabajos, se probaron solamente en la bacteria de *E. coli*, reportando precisiones de 74 % y 70 %, respectivamente.

Posteriormente, en el año 2004, Chen *et al*. (2004a,b) reportaron un método basado en redes neuronales que utilizaba como características de entrada las distancias intergénicas, conservación de vecindad, relaciones funcionales basadas en su clasificación de COG y número de genes en los operones experimentalmente validados. En este método, se relajó la restricción de adyacencia inmediata para el calculo de conservación de vecindad de pares de genes, pero siguió siendo limitada a un máximo de un gen intermedio. El método mostró una precisión en *E. coli* del 83.8 %, cuando todas las características eran utilizadas, habiendo probado cada una de éstas de manera independiente con una disminución en el desempeño. Con estos resultados, los autores probaron que al utilizar varias características de manera conjunta se podía mejorar la precisión alcanzada en la identificación de operones. Sin embargo, no probaron su método en otros genomas bacterianos.

En el año 2005, el trabajo desarrollado por De Hoon *et al*. (2004) ya reportaba una precisión del 88.6 % en la identificación de operones de *E. coli*. Este trabajo utilizó un clasificador bayesiano que tenía como características de entrada el tamaño de los operones experimentalmente validados, distancia intergénica y niveles de expresión de los genes de *E. coli*. Primero utilizaron cada una de las características de manera independiente evaluando su desempeño. La precisión de su método mejoró considerablemente cuando todas las características fueron usadas al mismo tiempo. Sin embargo, al haber evaluado las tres características sólo en *E. coli*, se asumieron distribuciones semejantes para todos los genomas bacterianos restantes, lo cual no es verdadero en todos los casos. Otro trabajo publicado el mismo año fue el de Jacob *et al*. (2005), en el cual se hizo uso de un algoritmo genético y lógica difusa como función de aptitud para evaluar la solución. Este método consideró las distancias intergénicas, conservación de vecindad de genes, vías metabólicas y funciones de los genes como características de entrada. Al igual que en trabajo anterior, probaron cada una de manera independiente para corroborar que al utilizarlas de manera conjunta lograban alcanzar una mayor precisión. Sin embargo, una limitante de este trabajo es que las reglas difusas generadas fueron intuitivas sin ninguna base biológica, probándolo sólo en datos de *E. coli* obteniendo una precisión del 88.2 %.

En ese mismo año, se reportaron otros dos métodos que utilizaron clasificadores bayesianos, el primero desarrollado por Price *et al*. (2006) que utilizó como características de entrada distancias intergénicas, conservación de vecindad, relación funcional basada en COG, y búsqueda de codones de inicio y término de la transcripción. Sin embargo, a pesar de haber utilizado todas estas características, reportó sólo 84 % de precisión en la identificación de operones de *E. coli*. El segundo trabajo lo publicó Westover *et al*. (2005), obteniendo una precisión del 83 % en *E. coli* utilizando distancias intergénicas, conservación de vecindad y relaciones funcionales. En este trabajo, a parte de utilizar las tres características de manera conjunta, realizaron otras tres pruebas donde utilizaban solamente dos, variando la característica que no era utilizada. Ésto lo desarrollaron con la finalidad de probar que al descartar una de las características el desempeño del clasificador disminuía, siendo necesario utilizar las tres de manera simultánea. Una restricción de este trabajo, es que utilizaron los pares de genes contiguos y en direcciones de transcripción diferentes (genes divergentes y convergentes) como el conjunto de negativos positivos, asumiendo que las propiedades de las características en estos pares de genes eran similares a las de pares de genes que no son parte de un mismo operón (genes contiguos y en la misma dirección), lo cual pudo introducir un bias en ciertas características al no ser siempre cierta esta afirmación.

Para el año 2006, su publicó el trabajo de Zhang *et al*. (2006) que utilizó una máquina de soporte vectorial, teniendo como características de entrada las distancias intergénicas, vías metabólicas, conservación de vecindad, perfiles filogenéticos y relaciones funcionales basadas en la clasificación GO. A pesar de haber utilizado cinco características de entrada, su precisión en *E. coli* fue de 85.5 %, la cual no estaba por arriba de trabajos previos que habían utilizado menos características. En el 2007, se reportó el trabajo desarrollado por Tran *et al*. (2007), el cual implementó una red neuronal artificial para combinar los resultados generados por tres métodos previamente reportados (Price *et al*., 2006; Westover *et al*., 2005; Chen *et al*., 2004b) y descritos en los párrafos anteriores. Adicionalmente, este trabajo consideró otra vez las distancias intergénicas, vías metabólicas y relaciones de funcionales, basadas en su clasificación GO, para mejorar su desempeño. Sin embargo, a pesar de haber utilizado el resultado de tres métodos previos como entradas y repetir varias características de éstos de manera individual, su desempeño no mejoró significativamente, reportando una precisión en *E. coli* de 86.5 %.

En el mismo año, el trabajo desarrollado por Bergman *et al*. (2007) utilizó las características de distancias intergénicas, perfiles filogenéticos y conservación de vecindad mediante un modelo oculto de markov bayesiano, logrando una precisión en *E. coli* de 86.5 %.

Una limitante de este trabajo es que asume distribuciones semejantes en todos los organismos de las características utilizadas, lo cual es incorrecto. Por otra parte, se publicó el trabajo de Wang *et al*. (2007), el cual utilizó como entradas las distancias intergénicas, relaciones funcionales, rutas metabólicas y niveles de expresión. Este método se basó en un algoritmo genético para realizar la identificación de operones obteniendo una precisión en *E. coli* de 85.9 %. Sin embargo, una restricción es que se utilizó las distancias intergénicas de *E. coli* como característica principal, asumiendo distribuciones semejantes en el resto de los organismos bacterianos, y el resto como características suplementarias.

Otro trabajo publicado en el mismo año y uno de los que más alta precisión había alcanzado, fue el reportado por Dam *et al*. (2007) el cual incorporó también varias características genómicas que incluía las distancias intergénicas, perfiles filogenéticos, conservación de la vecindad, relación funcional basada en GO, tamaño de los pares de genes adyacentes y patrón o motivo que delimita el inicio de la transcripción. En este trabajo, se probaron varias funciones estadísticas de una herramienta de reconocimiento de patrones de Matlab, siendo un árbol de decisión la que mejor desempeño tuvo, alcanzando una precisión en *E. coli* del 93.2 %. En este trabajo, en un intento de evaluar la contribución de las características utilizadas en la a identificación de operones, realizaron varias pruebas. Dichas pruebas se basaron en utilizar de manera independiente cada una de las características hasta ir utilizándolas de manera combinada para evaluar el error del clasificador. Sin embargo, no hicieron todas las combinaciones posibles ni se explica porque seleccionaron las cinco combinaciones realizadas. Finalmente, en el 2009, se publicó el método desarrollado por Li *et al*. (2009) el cual utilizó teoría de grafos para realizar la identificación de operones donde los vértices eran representados por genes y las aristas ponderadas por las características de entradas las distancias intergénicas y conservación de vecindad. Este método reportó una precisión de 93.3 % en *E. coli*. Una gran limitante de este método es que usaron umbrales de corte para las dos características utilizadas basándose únicamente en datos de *E. coli*.

A pesar de la gran variedad de métodos de identificación de operones mencionados en los párrafos anteriores, no existía un método satisfactorio ya que todos tenían la limitante de no poder ser utilizados en diversos genomas bacterianos sin perder precisión. En los casos en que estos métodos se entrenaban con datos de un genoma bacteriano y se usaban para reconocer operones de otro genoma, los porcentajes de precisión disminuían significativamente. Por ejemplo, el trabajo que mejor desempeño había reportado, fue el de Dam *et al*. (2007) el cual, como se mencionó, tenía una precisión de 93.2 % en la bacteria de *E. coli*. Sin embargo, cuando dicha metodología se probó en datos de *B. subtilis* su precisión disminuyó al 83 %. La forma de atacar este problema, fue desarrollar métodos '**universales**' que utilizaban

información genómica genérica (perfiles filogenéticos y conservación de vecindad, entre otros) de un conjunto de genomas bacterianos completamente secuenciados (Li *et al*., 2009; Yan and Moult, 2006; Edwards *et al*., 2005; Westover *et al*., 2005). Desafortunadamente, estos métodos tenían una baja sensibilidad debido al alto número de falsos negativos detectados, por ejemplo en el trabajo de Edwards *et al*. (2005) se reportó una sensibilidad de $49\,\%$ o en el de Westover *et al*. (2005) de 69 %. Por otra parte, los métodos que generalizan relativamente bien de un genoma a otro, es decir que no perdían tanta precisión cuando eran usados en otro genoma diferente al utilizado para su entrenamiento, son en general aquellos cuyas precisiones son en si bajas, por abajo del 89 % (Jacob *et al*., 2005; Price *et al*., 2005; Zhang *et al*., 2006).

Cabe hacer notar que en el 2009, salió publicado un artículo donde se estimó el valor real de todos los métodos reportados de identificación de operones (Pertea *et al*., 2009). En este trabajo, se evaluaron y compararon dichos métodos bajo un criterio uniforme y en un mismo conjunto de datos. Esto debido a que el conjunto de operones, experimentalmente validados de *E. coli* y *B. subtilis*, utilizados para estimar el desempeño de cada método pudo haber diferido, ya que las bases de datos utilizadas son continuamente actualizadas por la nueva información que se va generando. Además, la estimación del conjunto de falsos positivos había sido calculado de diversas maneras. Los resultados mostraron que las precisiones de los métodos reportados hasta ese momento disminuían cuando éstos eran probados con datos actuales y generados bajo el mismo criterio. Por ejemplo, dos de los primeros métodos publicados por Salgado *et al*. (2000) y Ermolaeva *et al*. (2001), que reportaban una precisión del 74 % y 70 % en *E. coli* respectivamente, en el trabajo de Pertea *et al*. (2009) tienen una del 40 %. Incluso, cuando el método de Ermolaeva *et al*. (2001) fue probado en datos de *B. subtilis* su precisión disminuía hasta el 18 %. Otro ejemplo, es el método publicado por Jacob *et al*. (2005) que reportó una precisión del 88.2 % en *E. coli*, evaluando en Pertea *et al*. (2009) una del 55 % y 70 % en *B. subtilis* y *E. coli*, respectivamente. Trabajos más recientes, como el de Dam *et al*. (2007) muestran una disminución en la precisión menos significativa, de alrededor del 10 %.

2.2.2.1. Conclusiones

Con base al estado del arte de los métodos de identificación de operones descritos en esta sección, se llegó a la conclusión de que ningún método había alcanzado el 100 % de precisión por lo que todavía existía la posibilidad de desarrollar un nuevo método que tuviera un mejor desempeño. Adicionalmente, se comprobó que una deficiencia común en los métodos desarrollados era su incapacidad de generalizar de un genoma a otro ya que sus precisiones disminuían significativamente cuando se entrenaban con datos de un organismo y se prueban con datos de otro.

Asimismo, se constató que ninguno de los métodos previamente reportados había realizado un proceso de selección de características relevantes evaluado la importancia relativa de las mismas en la identificación de operones. Algunos solamente habían evaluado la precisión alcanzada cuando utilizaban cada una de las características de manera independiente (De Hoon *et al*., 2004; Jacob *et al*., 2005) o quitando sólo una del conjunto total (Westover *et al*., 2005), o combinando sólo algunas de manera aleatoria (Dam *et al*., 2007), pero ninguno ha realizando un proceso completo. Esto a pesar de que se sabe que un proceso de selección de características relevantes generalmente resulta en una mejora en la precisión porque la información puede contener características ruidosas, redundantes o no importantes. Además, el conocer las características importantes en la identificación de operones ayudaría a la comprensión de la biogénesis de las unidades de transcripción de las bacterias.

CAPÍTULO 3

MARCO TEÓRICO DE REDES NEURONALES PERCEPTRÓN MULTICAPA

En este capítulo, primero se da una breve introducción a las redes neuronales artificiales perceptrón multicapa (MLP por sus siglas en inglés, MultiLayer Perceptron), realizando un análisis comparativo con las máquinas de soporte vectorial (SVM por sus siglas en inglés, Support Vector Machine). Posteriormente, se proporcionan los conceptos y bases de las redes MLP.

3.1. Introducción

En los últimos años, el área de aprendizaje de máquinas ha tenido un desarrollo impresionante, teniendo como objetivo desarrollar algoritmos y técnicas que permitan a las computadoras resolver problemas mediante datos de ejemplos o experiencias obtenidas con anterioridad. Estas técnicas tienen una amplia gama de aplicaciones tales como procesamiento de lenguaje natural, diagnóstico medico, bioinformática, análisis de mercados, reconocimientos de patrones, entre otras. Una de las aplicaciones más importantes del aprendizaje de máquinas es la clasificación, que es cuando se tienen dos o más clases a las que se debe asignar un caso no visto anteriormente. La experiencia anterior ayuda a entrenar un sistema que encuentra automáticamente la salida dada una entrada. Actualmente, se cuenta con múltiples técnicas de clasificación y reconocimiento de patrones que han sido desarrolladas bajo distintos enfoques, pero que persiguen objetivos similares. Dos de las más utilizadas y versátiles han sido las redes MLP y las SVM (Cybenko, 1989; Haykin, 1999).

Las redes MLP pueden ser consideradas como aproximadores universales de funciones en el espacio multidimensional, formadas por elementos no lineales conocidos como neuronas. El

conjunto de neuronas que conforman la red neuronal, se conectan de tal manera que la salida de una neurona cualquiera sirve como entrada de otras neuronas. El número de neuronas, la disposición de las mismas y las conexiones entre ellas determinan su estructura, denominada arquitectura o topología de la red. La topología de una red neuronal está formada por capas de neuronas y las conexiones entre éstas también se caracterizan por valores llamados pesos: i) *La capa de entrada* que recibe los valores de entrada o conjunto de entrenamiento que procesa la red. Estos valores son generalmente vectores que representan las características de un problema y son enviados a la capa intermedia o a la capa de salida dependiendo de la topología que tiene la red. ii) *La(s) capa(s) intermedia(s)* u oculta(s) recibe la información de la capa de entrada y evalúa la función que determina como modificar los pesos de sus conexiones. Además proporciona el resultado que se envía a otra capa oculta o la capa de salida. Esto depende de la topología de la red, porque la red puede tener más de una o ninguna capa oculta. iii) *La capa de salida* proporciona el resultado final del procesamiento que lleva a cabo la red. Esta capa recibe la información de la capa de entrada o de la última capa oculta y realiza las operaciones para modificar y ajustar los pesos entre las conexiones, de tal manera que proporciona una solución.

Por otra parte, las SVM son una técnica no paramétrica que fueron concebidas, en un principio, para resolver problemas linealmente separables. Modificaciones de las mismas han permitido que ahora tengan la capacidad de ser usadas en problemas de regresión no lineal y de clasificación de patrones no separables. Una SVM primero asigna los datos de entrada a un espacio de características de una dimensión mayor (e.g. si los datos de entrada están en R^2 entonces pueden ser asignados por la SVM a R^3) encontrando un hyperplano que los separe y maximice el margen m entre las clases en este espacio. Para ésto, utiliza el producto punto con funciones en el espacio de características que son llamados kernels. Las SVM pueden ser vistas como una red neuronal con una sola capa oculta de unidades no lineales, definidas por funciones kernel.

La ventaja más significativa de las SVM sobre las redes MLP es que en su espacio de búsqueda de soluciones sólo existe un mínimo global mientras que en las redes MLP existen múltiples mínimos locales, asegurando encontrar la solución óptima. En este sentido, en varias aplicaciones se han realizado diversos estudios comparativos entre las redes MLP y las SVM (Wu *et al*., 2008; Chen *et al*., 2005; Hsinchun *et al*., 2004; Byvatov *et al*., 2003; Jack and Nandi, 2002; Ding and Dubchak, 2001), considerando en general la precisión alcanzada, complejidad en la arquitectura y velocidad de convergencia. En cuanto a la precisión alcanzada en algunos casos y condiciones se ha concluido que las redes MLP son ligeramente mejores que las SVM (Wu *et al*., 2008; Hsinchun *et al*., 2004; Jack and Nandi, 2002), y en otros casos sucede lo contrario (Chen *et al*., 2005; Byvatov *et al*., 2003; Ding and Dubchak, 2001), siendo

estas diferencias mínimas y no significativas. En cuanto a la arquitectura, las redes MLP han demostrado ser menos complejas, empleando un número muy pequeño de neuronas ocultas mientras las SVM utiliza gran número de unidades ocultas (Wu *et al*., 2008; Chen *et al*., 2005; Hsinchun *et al*., 2004; Byvatov *et al*., 2003; Jack and Nandi, 2002; Ding and Dubchak, 2001). Por lo tanto, en problemas con conjuntos de datos de entrenamiento limitados, es recomendable el uso de redes MLP. En cuanto a la velocidad de convergencia, las SVM han demostrado ser más rápidas. En conclusión, se ha demostrado que tanto las redes MLP como las SVM son buenas soluciones en problemas de clasificación, regresión y tareas de predicción.

En este trabajo se ha optado usar redes MLP en vez de SVM como una alternativa eficiente para la identificación de operones, debido a que el conjunto de datos de entrenamiento no es muy grande (sección 4.4) y para tener una arquitectura sencilla que nos permita entender más fácilmente el problema. Además, las redes MLP son utilizadas para determinar la contribución relativa de cada una de las características empleadas en la identificación de operones con la finalidad de realizar una selección de las más relevantes (sección 6.3.1). En la siguiente sección se describe en detalle las redes MLP.

3.2. Redes Neuronales Perceptrón multicapa

Una red MLP con una estructura de dos capas, sin capa oculta, correspondiente a un perceptrón simple (Haykin, 1999). Se tiene una función F de $\Re^m$ que aplica un patrón de entrada $\mathcal{Y} = (Y_1, \ldots, Y_m)$ en la salida deseada $z \in \{-1, 1\}$ es decir $F(Y) = z$. La información que se dispone sobre dicha fusion está dada por n pares de patrones de entrenamiento $\big(\{Y_{1,1}, z_1\}, \ldots, \{Y_{m,n}, z_n\}\big)$, donde $F(\mathcal{Y}_i) = z_i \in \{-1, 1\}, i = 1, \ldots, n$. Dicha función realiza una partición en el espacio $\Re^m$ de los patrones de entrada, teniendo por una parte los patrones de salida $+1$ y por otra parte los patrones con salida -1, es decir, los patrones de entrada se clasifican en dos clases (Ver Figura 3.1). Por consiguiente, una red MLP simple puede ser vista como un dispositivo que aprende dicha función a partir del conjunto conocido de patrones de entrenamiento $\mathcal{Y}$, el cual es ampliado con una entrada interna constantemente conectada al valor $+1$ y multiplicada escalarmente por un vector de pesos $w \in \Re^{(n+1)}$ tal que:

$$v_{i,l} = \sum_{j=1}^{M} w_{j,l}\, y_{j,i} + \theta_l \tag{3.1}$$

donde θ_l es el umbral de la neurona de salida l, M hace referencia al número de neuronas de entrada, $w_{j,l}$ es el peso entre la neurona de entrada j y la de salida l y $y_{j,i}$ es el valor de la

neurona de entrada j para el patrón i.

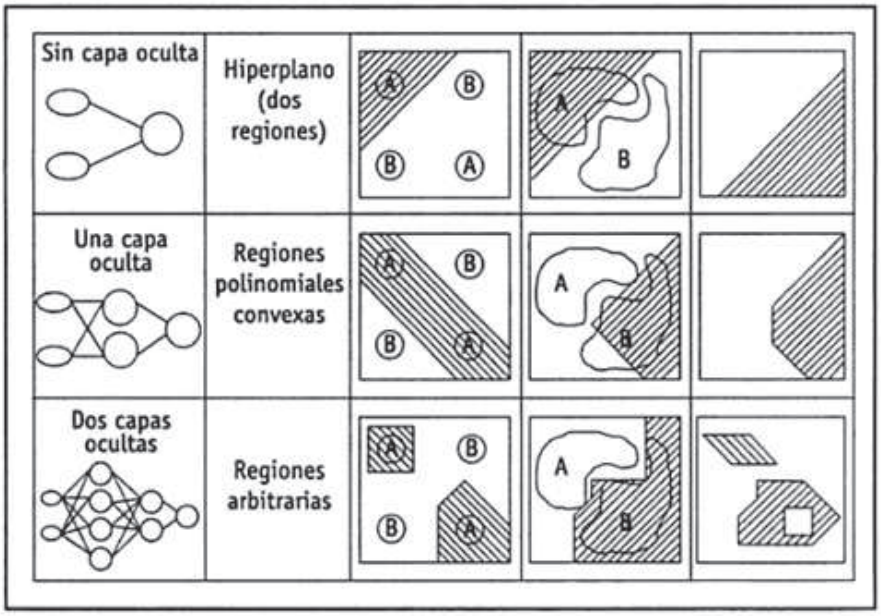

Figura 3.1: Tipo de regiones de decisión en las redes MLP (Fuente: Martin del Brío y Sanz Molina, 2001)

Al utilizar una función de activación lineal en la neurona de salida, $v_{i,l}$, la MLP puede ser vista como un modelo de regresión lineal (Haykin, 1999), donde si hay más de una neurona de salida, se convierte en un modelo de regresión multivariado. Por otra parte, un perceptrón simple con función de activación logística en la neurona de salida es similar a un modelo de regresión logística (Haykin, 1999) cuyo modelo matemático esta dado por:

$$v_{i,l} = \frac{1}{\left(1 + exp\big(\theta + \sum_{j=1}^{M} w_{j,l}\, y_{j,i}\big)\right)} \tag{3.2}$$

Lo que se puede apreciar es que mediante un perceptrón simple, la región de decisión es un hiperplano que separa los dos espacios de las variables, es decir, se divide los patrones de entrada en dos conjuntos A y B si estos son linealmente separables en un espacio dimensional (Figura 3.1. Para lo cual, debe existir $m + 1$ números reales $w_1, \ldots, w_m$ de manera que cada punt $y_{1,i}, \ldots, y_{m,i} \in A$ satisface $\sum_{j=1}^{m} w_{j,l}\, y_{j,i} \geq \theta$ y cada punto $y_1, \ldots, y_n \in B$ satisface $\sum_{j=1}^{M} w_{j,l}\, y_{j,i} \leq \theta$.

Por otra parte, se han realizado múltiples investigaciones para determinar el potencial de las redes MLP. Unas de las más importantes son las descritas en (Hornik *et al*., 1989) y la de (Cybenko, 1989), donde se demostró la capacidad de las MLP *feedforward* con una sola capa oculta como aproximadores universales. Esto es, dada cualquier función continua derivable y acotada en un hipercubo, existe un MLP que puede aproximar dicha función con un nivel de error. La base matemática de esta afirmación se debe a una ampliación del teorema de superposición de Kolmogorov (1957), donde se constató que una función continua de diferentes variables puede ser representada por la concatenación de varias funciones continuas de una misma variable. El teorema es el siguiente:

Teorema del Aproximador Universal

"Sea $\phi(\cdot)$ una función continua, no-constante, acotada y monótonamente creciente. Sea I_n un hipercubo unitario p-dimensional $[0,1]^n$. Sea $C(I_n)$ el espacio de funciones continuas definidas en I_n. Entonces, dada cualquier función $F \in C(I_n)$ y $\epsilon > 0$, existe un entero M y un conjunto de constantes α_j, θ_j y $w_{j,k}$ donde $j = 0, \ldots, M$ y $i = 1, \ldots, n$ tales que se puede definir:

$$F(y_{j,1}, \ldots, y_{j,n}) = \sum_{j=1}^{M} \alpha_j \phi\Big(\sum_{i=1}^{n} w_{j,i}\, y_{j,i} - \theta \Big) \tag{3.3}$$

donde F es una aproximación de la función $f(\cdot)$, esto es: $\big|F(y_1, \ldots, y_p) - f(y_1, \ldots, y_p)\big| \leq \epsilon$ para cualquier $y_1, \ldots, y_p \in I_p$"

En este sentido en una red MLP multicapa, la entrada total que recibe una neurona oculta j es:

$$v_{pj} = \theta_j + \sum_{l=1}^{N} w_{jl}\, y_{pl} \tag{3.4}$$

El valor de salida de la neurona oculta j se obtiene aplicando una función $f(\cdot)$ sobre su entrada neta:

$$b_{pj} = f(v_{pj}) \tag{3.5}$$

de igual forma, la entrada que recibe una neurona de salida k,u_{pk}, es:

$$v_{pk} = \theta_k + \sum_{j=1}^{L} v_{kj}\, b_{pj} \tag{3.6}$$

donde el valor de salida de la neurona de salida k, b_{pk} es: $b_{pk} = f(v_{pk})$

Por consiguiente el teorema del ***Aproximador Universal***, se puede aplicar directamente a

una red MLP utilizando una función de activación sigmoidal (logística) o hiperbólica como una función no-lineal en el modelo neuronal de la capa oculta. Esto debido a que estas funciones satisfacen la condición impuesta en $\phi(\cdot)$ al ser no-constantes y acotadas. Por otra parte, la ecuación 3.1 representa la salida de una MLP, donde: a) La red tiene p nodos de entrada y está constituida por una sola capa oculta de M neuronas, teniendo $y_1, \ldots, y_p$ entradas. b) La neurona l tiene pesos sinápticos $y_{l1}, \ldots, x_{lp}$ y un umbral θ_l. c) La salida de la red es una combinación lineal de las salidas de las neuronas ocultas, con $\alpha_l, \ldots, \alpha_M$ definiendo los coeficientes de esta combinación.

Esto significa que una MLP conteniendo al menos una capa oculta con suficientes unidades no lineales, tiene la capacidad de aprender, con cierto error, cualquier tipo de función siempre que pueda ser aproximada en términos de una función continúa (Figura 3.1), ya que cada neurona oculta se activa en una región distinta del espacio de entrada. Además, se ha demostrado que utilizando más de una capa oculta, la red puede aproximar relaciones que impliquen funciones discontinuas (Haykin, 1999) (ver Figura 3.1).

3.2.1. Aprendizaje y función de error

El aprendizaje se realiza mediante el algoritmo de *Backpropagation*, donde el objetivo es minimizar el error entre la salida obtenida por la red y la salida deseada ante la presentación de un conjunto de patrones denominado grupo de entrenamiento (Haykin, 1999). La función de error que se pretende minimizar para cada patrón p viene dada por:

$$E_p = \sum_{k=1}^{M} \left(d_{pl} - v_{pk}\right)^2 \tag{3.7}$$

donde d_{pk} es la salida deseada para la neurona de salida k ante el patrón p. A partir de esta expresión, se puede obtener una medida general del error mediante:

$$E = \sum_{p=1}^{P} E_p \tag{3.8}$$

Para ésto, el algoritmo busca determinar el vector de pesos w_l que minimice el error E. Lo anterior se logra modificando los pesos mediante una técnica conocida como gradiente decreciente. Teniendo en cuenta que el gradiente de E_p es un igual a la derivada parcial de E_p respecto a cada uno de los pesos. Éste toma la dirección que determina el incremento más rápido en el error, mientras que la dirección opuesta (dirección negativa) determina el decremento más rápido en el error. Por lo tanto, el error puede reducirse ajustando cada peso en

la dirección:

$$E = -\sum_{p=1}^{P} \frac{\partial E_p}{\partial w_{jl}} \tag{3.9}$$

La forma de modificar los pesos de forma iterativa, hasta alcanzar los que minimicen el error, consiste en aplicar la regla de la cadena a la expresión del gradiente y añadir una tasa de aprendizaje η. Por ejemplo, para peso de una neurona de salida:

$$\delta v_{kj}(n+1) = \eta \sum_{p=1}^{P} \delta_{pk}\, b_{pj} \tag{3.10}$$

donde $\delta_{pk} = (d_{pk} - v_{pl}) f(v_{pk})$, n indica la iteración. Cuando se trata del peso de una neurona oculta:

$$\delta w_{kj}(n+1) = \eta \sum_{p=1}^{P} \delta_{pj}\, y_{pl} \tag{3.11}$$

El valor de η tiene un papel importante en el proceso de entrenamiento de una red neuronal, ya que controla el tamaño del cambio de los pesos en cada iteración. De este modo, el error asociado a una neurona oculta j está dado por la suma de los errores que se comenten en k neuronas de salida que reciben como entrada la salida de esa neurona oculta j. De ahí que el algoritmo también se denomine propagación del error hacia atrás o *Backpropagation*, donde la modificación de los pesos se realiza después de haber presentado todos los patrones de entrenamiento. De esta manera, al inicio los pesos son inicializados con valores aleatorios pequeños y el proceso de ajuste continúa iterativamente. La parada del proceso de aprendizaje puede ser llevado a cabo por medio de uno de los siguientes criterios: a) Elegir un número de pasos fijos; b) El proceso de aprendizaje continua hasta que la cantidad $v_{kj} = v_{kj}(n+1) - v_{kj}$ está por debajo de algún valor específico; c) Parar cuando el error total alcanza un mínimo en el conjunto de prueba.

Mediante este algoritmo, las MLP perceptrón multicapa logran determinar los valores de los pesos que minimicen el error. Sin embargo, si el problema se ve de forma gráfica el conjunto de pesos que forma una MLP puede ser representado por un espacio compuesto por tantas dimensiones como pesos se tengan. Por ejemplo, para una red formada por dos pesos, éstos se pueden visualizar como un espacio de dos dimensiones donde el error cometido es función de los pesos de la red y donde a cualquier combinación de los dos pesos, le corresponderá un valor de error para el conjunto de entrenamiento. Estos valores de error se pueden visualizar como una superficie, que se denomina superficie del error. El algoritmo de aprendizaje se basa en obtener información local de la pendiente (gradiente) de la superficie, y a partir de esa información modificar iterativamente los pesos de forma proporcional a dicha

pendiente, a fin de asegurar el descenso por la superficie del error hasta alcanzar el mínimo más cercano desde el punto de partida (Masters, 1993).

Para una superficie de error de tipo paraboloide o parábola tridimensional de una MLP con una sola capa y un perceptrón, sólo existe un mínimo en toda la superficie (Duda, R.O. and Stork, 2000), ya que éstas sólo tienen un mínimo. Sin embargo, la superficie de error que se genera por combinación de pesos en una red no-lineal es completamente distinta. La superficie del error puede tener una topografía arbitrariamente compleja, dependiendo de la cantidad de pesos involucrados y de las funciones de activación de las neuronas, donde en este caso tienden a tener muchos mínimos locales.

Por lo tanto, en las técnicas de gradiente descendente se recomienda avanzar por la superficie de error con incrementos pequeños en los pesos (Duda, R.O. and Stork, 2000; Haykin, 1999). Esto se debe a que se tiene una información local de la superficie y no se sabe lo lejos o lo cerca que se está del punto mínimo. Con incrementos grandes, se corre el riesgo de pasar por encima del punto mínimo sin conseguir estacionarse en él, mientras con incrementos no tan pequeños, se trata de evitar el caer en un mínimo local. El elegir un incremento adecuado influye en la velocidad con la que converge el algoritmo. Este control se realiza mediante el parámetro η. Asimismo, otra manera de incrementar la velocidad de aprendizaje, consiste en utilizar otro parámetro llamado Momento (α), cuando se calcula el cambio de peso se le añada una fracción del cambio anterior. De este modo, cuando se trata, la ecuación 3.10, queda:

$$\delta v_{kj}(n+1) = \eta\Big(\sum_{p=1}^{P} \delta_{pk}\, b_{pj}\Big) + \alpha \Delta v_{kj}(n) \tag{3.12}$$

y de una neurona oculta (3.11) de:

$$\delta w_{lk}(n+1) = \eta \sum_{p=1}^{P} \delta_{pj}\, y_{pi} + \alpha \Delta w_{kj}(n) \tag{3.13}$$

donde α permite filtrar las oscilaciones en la superficie de error provocadas por la tasa de aprendizaje y acelera considerablemente la convergencia de los pesos, debido a que si en el momento n el incremento de un peso era positivo y en $n+1$ también, entonces el descenso por la superficie de error en $n+1$ será mayor. Sin embargo, si en n el incremento es positivo y en $n+1$ negativo, el paso que se da en $n+1$ es más pequeño, lo cual es adecuado ya que ésto significa que se ha pasado por un mínimo.

Sin embargo, a pesar de aplicar una tasa de aprendizaje (η) y un factor momento (α) en el

algoritmo del gradiente descendiente, no se puede garantizar en ningún momento que el mínimo que se encuentre en una superficie de error con topografía arbitrariamente compleja sea global. Además, el algoritmo se puede quedar estancado en una meseta, donde la pendiente de la superficie es muy pequeña. Una vez que la red se asienta en un mínimo, sea local o global, cesa el aprendizaje, aunque el error siga siendo demasiado alto y los pesos no sean los óptimos por haber alcanzado un mínimo local. En conclusión, no se puede garantizar la obtención de los pesos óptimos al menos que se esté trabajando sobre una superficie cóncava. Sin embargo, si se satisface el porcentaje de error permitido en el conjunto de datos de entrenamiento y de prueba, se puede deducir que el entrenamiento fue un éxito.

3.2.2. Arquitectura y generalización

A la hora de evaluar el desempeño de una red MLP no sólo importa saber si la red aprendió con éxito los patrones utilizados durante el aprendizaje, sino conocer el comportamiento de la red frente a patrones que no fueron utilizados durante el entrenamiento. Para tal fin, es necesario disponer de dos conjuntos de patrones: el conjunto de entrenamiento, que entrena y modifica los pesos y umbrales de la red, y el conjunto de validación que mide la capacidad de la red para responder correctamente a los patrones que no fueron ingresados durante el entrenamiento. En este sentido, la generalización es la habilidad de una red neuronal de almacenar en sus pesos sinápticos características que le son comunes a todos los patrones que fueron usados durante la fase de entrenamiento, es decir:

- Cuando una red generalizara clasifica o aproxima de manera correcta, con el mínimo error ajuste, los datos de prueba que nunca han sido utilizados para su entrenamiento.
- Cuando la red aproxima correctamente los patrones de aprendizaje, pero no responde bien a los patrones de validación, se dice que hubo subaprendizaje de la red o memorización de los patrones de entrenamiento.

El proceso de aprendizaje puede ser visto como un problema de ajuste de curva, donde la generalización puede ser vista como el efecto de una buena interpolación no lineal de los datos de entrada. La generalización está influida principalmente por la arquitectura de la MLP y el número de ciclos de entrenamiento. Respecto a la arquitectura de la red MLP, al ser estas aproximadores donde cada neurona de la capa oculta se activa en una región distinta del espacio de entrada, si se utilizan demasiadas neuronas ocultas en la red pueden conducir a una escasa capacidad de generalización Haykin (1999); Kaastra and Boyd (1996); Zhang *et al*. (1998). En estos casos, la red tiende a ajustar con mucha exactitud los patrones de entrenamiento. Particularmente, en problemas en los que las muestras poseen ruido la utilización de muchas neuronas ocultas harían que la red se ajuste al ruido de los patrones

impidiendo así la generalización, con lo que disminuiría el error de entrenamiento pero aumentará considerablemente el error de validación. Por otra parte, el poner pocas neuronas ocultas no es la solución ya que a veces no son suficientes debido a que el entrenamiento es pobre. En conclusión, es necesario un balance entre el número de neuronas ocultas y la generalización y desempeño del MLP.

En este sentido, uno de los problemas más importantes en el diseño de las redes neuronales es el determinar una buena arquitectura (número de capas ocultas y neuronas) en dependencia de la problemática (se puede consultar Gori and Tesi (1992) y Hecht-Nielsen (1995) para una explicación detallada de dichos criterios). En cuanto al número de capas ocultas, el incrementarlas, a parte de aumentar el tiempo computacional requerido, puede ocasionar que las MLP puedan sobreajustar y por consiguiente tener un mal desempeño. El sobreajuste ocurre cuando un modelo tiene pocos grados de libertad, es decir, tiene relativamente pocas observaciones en relación a sus parámetros y por lo tanto tiende a memorizar los datos de entrenamiento en vez de aprender los patrones generales de los mismos. En las redes MLP entre más pesos se tengan en relación al tamaño del conjunto de entrenamiento más posibilidad se tiene de memorización. Dichos pesos están en dependencia directa del número de capas ocultas que se utilicen, por lo que la recomendación es empezar con una capa oculta lo cual es lo mínimo para tener un aproximador universal e ir probando el desempeño alcanzado en el conjunto de prueba y entrenamiento conforme se aumentan las capas. En cuanto al número de neuronas, algunas reglas básicas se han presentado. Por ejemplo, en Masters (1993) para un MLP de tres capas con j neuronas de entrada y l neuronas de salida, se propuso una capa oculta con $sqrt(j * l)$ neuronas. El número real de neuronas ocultas puede variar de este valor a tres veces esta medida dependiendo de la complejidad del problema. Por otra parte en Baily and Thomson (1990) sugirieron que el número de neuronas ocultas en una red neuronal de tres capas debiera ser el $75\,\%$ del número total de neuronas de entrada. En Haykin (1999), 1999, se propone que el número de pesos y sesgo de una red debe ser $W = \epsilon \cdots N$, donde N es el número de patrones de entrada y ϵ el error que se quiere. Por ejemplo, si se quisiera un error del 10 por ciento, el número de pesos y bias deberá ser el número de patrones de entrada entre 10. Así como estos trabajos, existen muchos otros que han propuesto diversas medias para determinar el número de neuronas ocultas.

3.3. Método de validación

Los estimadores no paramétricos que se utilizan en este trabajo están basados en la idea del remuestreo (resampling). Uno de los métodos utilizados para determinar la precisión de la red

MLP es el de *Holdout* o validación cruzada (Stone, 1974), el cual particiona los datos aleatoriamente en dos conjuntos mutuamente exclusivos, denominados conjunto de entrenamiento y conjunto de validación (o conjunto de *holdout*). El conjunto de entrenamiento es usado para inducir un modelo clasificatorio, utilizando el conjunto de validación para estimar la predicción verdadera. El conjunto de entrenamiento viene a ser habitualmente las dos terceras partes de todos los datos, utilizando el resto para el conjunto de validación. Formalmente se tiene D_h como el conjunto de validación, un subconjunto de D de tamaño N_h, y D_t definido por D/D_h como el conjunto de aprendizaje siendo $N_t = N - N_h$.

Por otra parte, para evaluar la generalización de la red MLP, se hace uso del método de validación cruzada (*cross-validation*) que viene también a ser una generalización del *Holdout*. El conjunto de datos D se divide aleatoriamente en k subconjuntos mutuamente excluyentes $D_1, D_2, \ldots, D_k$ de aproximadamente el mismo tamaño. El clasificador es entrenado y validado k veces. Cada instante de tiempo $t \in l, 2, \ldots, k$ es entrenado en D/D_t y validado en $D - t$. La estimación de la exactitud por medio del método de validación cruzada es el número total de bien clasificados, dividido entre el número total de instancias del conjunto de datos.

Un caso particular de la validación cruzada y el utilizado en este trabajo, es el dejar-uno-fuera (*leave-one-out*), en el cual el parámetro k viene a ser igual al número de instancias N que existen para inducir el modelo final. De esta forma, los N subconjuntos de validación están formados por una única instancia y los de entrenamiento por los de la cardinalidad del conjunto total menos esa única instancia que ha sido llevada a la validación.

CAPÍTULO 4

PLANTEAMIENTO FORMAL DEL PROBLEMA DE IDENTIFICACIÓN DE OPERONES

4.1. Propiedades de los operones

La discriminación de cuándo un par de genes pertenecen al mismo operón o no, requiere de un conjunto adecuado de propiedades que describan a cada gen, junto con ciertos patrones de entrada que deben ser estimables y obtenidos a partir de las características propias de las parejas de genes a identificar. A continuación, se describen las características y funciones que definen los patrones de los operones a reconocer, indicando las suposiciones y consideraciones que llevaron a formular el problema de identificación con dichos patrones.

Sea G^b un genoma bacteriano que está compuesto por un conjunto finito de genes ordenados por su posición $5' \rightarrow 3'$ del DNA en el genoma, descrito como:

$$G^b = \left\{ g_1^b, g_2^b, \ldots, g_n^b \right\} \text{ con cardinalidad } |G^b| \tag{4.1}$$

donde el índice b está asociado a un genoma específico del conjunto total $\mathcal{G}$ de 1053 genomas completamente secuenciados. Además, considere que cada gen g_i^b puede describirse en términos de su vector de propiedades:

$$\mathcal{X}_i^b = \begin{pmatrix} x_{1,i}^b & x_{2,i}^b & \ldots & x_{c-1,i}^b & x_{c,i}^b \end{pmatrix}^T \tag{4.2}$$

en donde cada $x_{c,i}^b$ tiene su propio dominio $\mathcal{D}_i$, el cual puede se binario, entero, cadena de caracteres u otro, permitiendo ausencia de información. Para evitar ambigüedad, el elemento $x_{c,i}^b$ denota la característica $x_c \in \mathcal{D}_c$ del gen ${g_i}^b$ perteneciente al genoma G^b.

Como se mencionó en la sección 2.1.1, diferentes genes contiguos relacionados con una misma función (por ejemplo, formación de un flagelo) o con una misma vía metabólica (por ejemplo, la síntesis de un aminoácidos determinado) son transcritos comúnmente en una sola unidad llamada operón (Definición 1).

Definición 1 *Un operón O^b es definido como un subconjunto de genes contiguos de un genoma G^b:*

$$O^b = \left\{ g_i^b, \ldots, g_{n_o}^b \right\} \tag{4.3}$$

tal que sus genes son transcritos en la misma dirección y están relacionados de alguna manera para transcribirse juntos en la misma unidad transcripcional (TU), donde $1 \leq n_o \leq n$.

El conjunto total de operones de un genoma G^b está dado por:

$$\mathcal{O}^b = \left\{ {O_1}^b, {O_2}^b, \ldots, {O_p}^b \right\} \tag{4.4}$$

donde p es el número total de operones de G^b. Por consiguiente, para describir las posibles relaciones de genes de un mismo operón de manera genérica se propone la siguiente definición.

Definición 2 *Sea $\mathcal{X}_i^b$ y $\mathcal{X}_{i+1}^b$ los vectores de propiedades de dos genes contiguos g_i^b and g_{i+1}^b del mismo genoma G^b, entonces cualquier relación entre estos, en términos de una asociación de sus propiedades, pueden ser expresadas por:*

$$y_{j,i}^b = x_{r,i}^b \otimes x_{q,i+1}^b \quad \in \ D_j \tag{4.5}$$

donde el índice j identifica a la característica obtenida mediante la relación de las propiedades $x_{r,i}^b$ y $x_{q,i+1}^b$ (4.2) y se le puede asignar cualquier índice particular, el símbolo $\otimes$ denota una relación permitida de los genes entre sus elementos r y q y el dominio D_j depende de la asociación especifica entre estos.

Sólo algunas relaciones específicas (4.5) son útiles para determinar si diferentes genes se transcriben como una misma unidad mediante ciertos atributos de sus propios vectores $\mathcal{X}_i^b$, las cuales son consideradas como características en el problema de identificación de operones. Los elementos de $\mathcal{X}_i^b$ usados para este fin son:

- $x_{1,i}^b$: define la nomenclatura del gen, la cual es su denominación científica (nombre sistemático, nombre estándar o el nombre reservado).

- $x^b_{2,i}$: define el número de posición del gen en relación a su orden dentro de un genoma.
- $x^b_{3,i}$: define la dirección de transcripción del gen g^b_i y puede ser *Forward* lo cual significa que el gen es leído en la dirección $5' \rightarrow 3'$ del DNA (ácido desoxirribonucleico) del genoma G^b; o *Reverse* cuando es leído en la cadena complementaria.
- $x^b_{4,i}$: define la posición izquierda del gen ${g_i}^b$, en nucleótidos, dentro del genoma G^b con respecto a su dirección $5' \rightarrow 3'$.
- $x^b_{5,i}$: define la posición derecha del gen ${g_i}^b$, en nucleótidos, dentro del genoma con respecto a su dirección $5' \rightarrow 3'$.
- $x^b_{6,i}$: define al tipo de RNA (ácido ribonuclaico) al que el gen ${g_i}^b$ pertenece, definido por el conjunto $\{mRNA, tRNA, rRNA\}$. Donde, los elementos $tRNA$ (RNA de transferencia) y $rRNA$ (RNA ribosomal) representan las secuencias no-codificantes expresadas con elementos de bases nitrogenadas ($\mathcal{NU}$) del conjunto $\mathcal{NU}$, donde:

  $$\mathcal{NU} = \Big\{A, T, G, C\Big\}$$

  formado por los elementos: adenina (A), guanina (G), citosina (C) y timina (T). El elemento $mRNA$ representa a una secuencia codificante expresada con elementos del conjunto de aminoácidos $\mathcal{AM}$:

  $$\mathcal{AM} = \Big\{A, R, D, N, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\Big\}$$

  donde, como se mostró en la Tabla 2.1, cada elemento está formado por combinaciones de tres elementos del conjunto $\mathcal{NU}$, e.g. $A = \{GCA, GCC, CGC, GCU\}$ donde para formar un aminoácido, la T es cambiada por una U.
- $x^b_{7,i}$: define la cadena de DNA del gen ${g_i}^b$ compuesta de elementos de $\mathcal{NU}$.
- $x^b_{8,i}$: define la cadena de RNA del gen ${g_i}^b$ compuesta de elementos de $\mathcal{AM}$, respectivamente.
- $x^b_{9,i}$: define el identificador único del gen por cada base de datos diferente.
- $x^b_{10,i}$: define al grupo de genes ortólogos al que pertenece el gen ${g_i}^k$, tal que:

  $$x^b_{10,i} = \begin{cases} \mathcal{COG}_t & \text{si } \mathcal{COG}_t \,\epsilon\, \mathcal{COG} \\ \mathcal{ROG}_s & \text{si } \mathcal{ROG}_s \,\epsilon\, \mathcal{ROG} \\ NOG & \text{en cualquier otro caso} \end{cases}$$

  donde $\mathcal{COG}$ y $\mathcal{ROG}$ son conjuntos de grupos de genes ortólogos creados bajo diferentes

metodológicas, NOG es una etiqueta que denota que el gen g_i^b no pertenece a ningún de estos dos conjuntos.

Por definición, los genes ortólogos son aquellos que provienen de diferentes especies, originándose a partir de un ancestro común por lo que realizan funciones parecida (Mindell and Meyer, 2001). En este trabajo, los grupos de genes ortólogos $\mathcal{COG}_t$ pertenecientes al conjunto $\mathcal{COG}$ son obtenidos de la BD COG de NCBI (Tatusov *et al*., 2003), la cual contiene 4873 elementos, por lo tanto $t = 1, .., 4873$. Dado que no todos los genes a la fecha secuenciados tienen asignado un grupo de la BD COG, en este trabajo se generaron nuevas agrupaciones de genes ortólogos denominadas ROG (Remaining Orthologous Cluster) utilizando el método descrito en la sección 7.2. El conjunto *ROG* cuenta con 8450 elementos, por lo tanto $s = 1, .., 8450$. Finalmente, en el caso de genes que no se les ha podido definir un grupo de ortólogos $\mathcal{COG}_\sqcup$ o $\mathcal{ROG}_\in$, dado que su sobre representación en los genomas actualmente secuenciados es muy baja, se les asigna la etiqueta NOG .

Mediante (4.5), una característica para identificar los operones del genoma G^b puede ser descrita por el conjunto formado por la relación $y_{j,i}^b$ aplicada a todos los pares de genes contiguos en el genoma G^b:

$$Y_j^b = \begin{pmatrix} y_{j,i}^b \\ y_{j,i+1}^b \\ \vdots \\ y_{j,\tilde{n}}^b \end{pmatrix} \tag{4.6}$$

donde $\tilde{n}$ corresponde a un estimado del número de pares de genes contiguos en la misma dirección de transcripción del genoma G^b.

Por consiguiente, el arreglo patrón:

$$\mathcal{Y}^b = \begin{pmatrix} Y_1^b & Y_2^b & \dots & Y_m^b \end{pmatrix} \tag{4.7}$$

describe m características usadas para discriminar el conjunto total de operones (4.4) del genoma G^b, obtenidas con $\tilde{n}$ relaciones de pares de genes contiguos.

4.1.1. Identificación

Considerando el conjunto de características propias de un operón en un genoma determinado b, el problema básico de la identificación de operones se establece como sigue.

Dada una función desconocida $g : \mathbb{Y}^b \rightarrow \mathbb{C}$ que mapea el conjunto de entradas $\mathcal{Y}^b \in \mathbb{Y}^b$ asociadas a las características de estos, en el espacio binario de salida $\mathbb{C} = \{o, \overline{o}\}$ dado por las clases:

- o para pares de genes que pertenecen a la clase *operón*, y
- $\overline{o}$ para los que pertenecen a la clase *no-operón*.

Se busca generar una función $h^* : \mathbb{Y}^b \rightarrow \mathbb{C}$ que se aproxima tanto como sea posible al mapeo correcto para un conjunto de datos de entrenamiento $D = \{(Y_1, \mathbb{C}_1), ...(Y_n, \mathbb{C}_n)\}$ considerados representativos de las dos clases a identificar.

Para formalizar el problema el término *"tan próximo como sea posible"*, se emplea una función de costo que se incrementa en un valor positivo distinto de cero por cada dato del conjunto D cuyo mapeo $\mathbb{C}_k$ de la función h^* sea incorrecta. Esta formulación es equivalente a determinar la función h^* que maximiza el valor de confianza para el conjunto de datos disponibles de forma sistemática.

4.2. Preprocesamiento de las características

El propósito del preprocesamiento de los datos utilizados en cualquier problema de reconocimiento de patrones es principalmente corregir las inconsistencias para que dichos datos conserven su coherencia. Para ésto, antes de realizar cualquier procedimiento de identificación, el punto de partida es el pre-procesamiento de $\mathcal{Y}^b$ mediante la eliminación de los valores atípicos de cada $Y_j^b \in \mathcal{Y}^b$, lo cual evita errores grandes en el entrenamiento. Posteriormente, se realiza una normalización de los datos con la finalidad de igualar la importancia de cada una de las características.

Para eliminar los valores atípicos, primero se calcula la media (μ_j) y la desviación estándar (σ_j) de la característica Y_j^b para después analizar los valores del vector de características de acuerdo a:

$$y_{j,i}^b \text{ es } \begin{cases} \textit{se elimina} & \text{si } (\mu_j + \beta\sigma_j) < y_{j,i}^b < (\mu_j - \beta\sigma_j;) \\ \textit{se mantiene} & \text{de otro modo} \end{cases} \tag{4.8}$$

donde β es el coeficiente seleccionado para todos los j a fin de tener los mismos intervalos de confianza (Barnett and Lewis, 1994). Un β grande tiende a eliminar los casos más extremos mientras un β pequeño puede quitar los valores que no son realmente los valores atípicos.

Después de que los valores extremos son eliminados, cada Y_j^b es normalizada en el mismo rango con la finalidad de asegurar que todas las características tengan el mismo peso en la decisión a pesar de sus diversos dominios, es decir igualar la importancia de las características, por lo tanto:

$$\tilde{y}_{j,i}^b = R_{min} + \left(y_{j,i}^b - min(Y_j^b)\right)\left(\frac{R_{max} - R_{min}}{max(Y_j^b) - min(Y_j^b)}\right) \tag{4.9}$$

donde $y_{j,i}^b$, $min(Y_j^b)$, $max(Y_j^b)$, son los valores actual, mínimo y máximo de cada vector de la característica Y_j^b, respectivamente; R_{min} y R_{max} son el valor inferior y superior del rango a escalar, respectivamente. Para simplificar la notación, en este trabajo la tilde de la notación es eliminada.

4.3. Evaluación del desempeño

Estimar el desempeño de un método de clasificación inducido por un algoritmo de aprendizaje automático es importante para predecir su comportamiento. En este trabajo, para evaluar el desempeño del método propuesto se hizo uso de las matrices de confusión, las cuales en el campo de la inteligencia artificial son empleadas para evaluar métodos de aprendizaje supervisado (Kohavi and Provost, 1998). Esta matriz permite ver la distribución de los errores y aciertos y contiene información acerca de las clasificaciones actuales y predichas. Las columnas de la matriz se utilizan para las clases de referencia y las filas para las clases deducidas del resultado de la clasificación. La diagonal de la matriz expresa el número de casos de verificación en donde se produce acuerdo entre las dos fuentes, mientras los marginales suponen errores de asignación. La Tabla 4.1 muestra la matriz de confusión para un clasificador de dos clases.

		Predicho	
		Negativo	**Positivo**
Actual	**Negativo**	VN	FN
	Positivo	FP	VP

Tabla 4.1: Matriz de confusión

Las entradas de la matriz de confusión tienen el siguiente significado:

- VN (Verdaderos Negativos): es el número de predicciones correctas de que una instancia sea negativa.

- FN (Falsos Negativos): es el número de predicciones incorrectas de que una instancia sea positiva.
- FP (Falsos Positivos): es el número de predicciones incorrectas de que una instancia sea negativa.
- VP (Verdaderos Positivos): es el número de predicciones correctas de que una instancia sea positiva.

La evaluación de estos índices permite obtener varias medidas para evaluar el desempeño de un clasificador:

$$\text{Sensibilidad} = \frac{VP}{VP + FN} \tag{4.10}$$

la cual indica la capacidad del clasificador para dar como casos positivos los casos realmente positivos.

$$\text{Especificidad} = \frac{VN}{VN + FP} \tag{4.11}$$

la cual indica la capacidad del clasificador para dar como casos negativos los casos realmente negativos.

$$\text{Precisión} = \frac{VP + VN}{VP + FN + VN + FP} \tag{4.12}$$

la cual indica la capacidad del clasificador para dar como casos negativos y positivos los casos realmente negativos y positivos, respectivamente.

En el contexto de este estudio, VN representa el número de pares de genes no-operones bien identificados como no-operones, FN el número de pares de genes no-operones mal reconocidos como operones, FP el número de pares de genes operones asignados erróneamente como no-operones y finalmente, VP representa el número de pares de genes operones correctamente identificados como operones.

4.4. Conjunto de datos

Cualquier metodología de reconocimiento de patrones supervisada, para clasificar automáticamente una nueva muestra, considera la información que pueda extraer de un conjunto de datos disponibles divididos en clases y otro conjunto en el cual evaluar el

desempeño alcanzado. En este sentido, la mayoría de los métodos de reconocimiento de operones han utilizado a las bacterias *Escherichia coli* (G^{eco}) y *Bacilus subtilis* (G^{bsu}) como organismos modelos, debido a que han sido ampliamente estudiadas en diversas aplicaciones y cuentan con resultados experimentales. Por consiguiente, estos organismos fueron los que se utilizaron como el universo de verdaderos y falsos positivos para entrenar, validar y comparar la precisión del método propuesto con trabajos previos. La Tabla 4.2 da un resumen de los datos utilizados y en el Apéndice A se da información de donde se obtuvieron dichos datos.

Los datos de operones de *E. coli* fueron obtenidos de la BD de RegulonDB (Gama-Castro *et al*., 2008), la cual reporta cerca de 2663 operones de uno o más genes, cuya existencia está sustentada por diversos tipos de evidencia, algunas clasificadas como fuertes (e.g. experimentalmente comprobados) y otras como débiles (e.g. predicciones computacionales). En este trabajo, se consideró un subgrupo de 344 operones que cuentan con evidencia fuerte, tal que utilizando (4.4) se tiene:

$$\mathcal{O}^{eco} = \left\{ {O_1}^{eco}, \ldots, {O_{344}}^{eco} \right\}$$

De estos operones, se formó el conjunto de pares de genes operones $\mathcal{OP}^{eco}$ (genes contiguos que se transcriben en la misma dirección y que pertenecen a un mismo operón):

$$\mathcal{OP}^{eco} = \left\{ (g_i^{eco}, g_{i+1}^{eco}), \forall\ g_i^{eco} \text{ y } g_{i+1}^{eco} \in {O_l}^{eco}, \text{ con } l = 1, \ldots, 344 \right\}$$

donde $|\mathcal{OP}^{eco}| = 493$.

Por otra parte, el conjunto $\overline{\mathcal{OP}^{eco}}$ de pares de genes no-operones de *E. coli* (falsos positivos), está formados por genes bordes de operones con referencia a su dirección $5' \rightarrow 3'$ y sus correspondiente genes adyacentes de sus extremos $5'$ y $3'$ del DNA. Es decir, si los genes $g_{i+1}^b, g_{i+2}^b, g_{i+3}^b$ definen un operón, entonces el par de genes (g_i^b, g_{i+1}^b) y (g_{i+3}^b, g_{i+4}^b) son bordes y por lo tanto forman parte del conjunto no-operón. De esta manera, el conjunto $\left|\overline{\mathcal{OP}^{eco}}\right| = 386$.

Asimismo, el conjunto de operones de *Basilus subtilis* se obtuvo de la BD DBTBS (Sierro *et al*., 2008). En este caso, el número de operones reportado es de 1153 de los cuales 509 cuentan con evidencia fuerte, tal que $\mathcal{O}^{bsu} = \left\{ {O_1}^{bsu}, {O_2}^{bsu}, \ldots, {O_{509}}^{bsu} \right\}$. Al igual que para *E. coli*, se definió el conjunto $\mathcal{OP}^{bsu}$ de pares de genes operones con 698 elementos, por consiguiente $|\mathcal{OP}^{eco}| = 698$. Mientras tanto, el conjunto de pares de genes no-operones ($\overline{\mathcal{OP}^{bsu}}$) fue de 433 (Tabla 4.2).

Adicionalmente, para probar la generalización del método en cualquier organismo

		Pares de genes						
			Totales en clase		Con COG		Sin COG	
	ID	**Operones**	***Operón***	***No-operón***	***Operón***	***No-operón***	***Operón***	***No-operón***
E. coli	eco	344	493	386	435	309	58	77
B. subtilis	bsu	509	696	433	527	276	169	157
50 organismos	ots	202	443	NA	292	NA	151	NA
H. pylori	hpy	528	1199	NA	582	NA	617	NA
M. pneumoniae	mpn	341	384	NA	179	NA	205	NA
S. solfataricus	sso	2165	856	NA	454	NA	402	NA
L. monocytogenes	lmo	522	1157	NA	841	NA	316	NA

Tabla 4.2: Conjuntos de datos utilizados como verdaderos positivos y verdaderos negativos
*NA = No Aplica

bacteriano sin que la precisión disminuyera significativamente, también se probó en un conjunto de 202 operones determinados experimentalmente en 50 genomas parcialmente estudiados obtenidos de la BD ODB (Okuda and Yoshizawa, 2011), tal que $|\mathcal{OP}^{ots}| = 433$. Asimismo, se probó en 528 operones de la bacteria *Helicobacter pylori* (Sharma *et al*., 2010), 341 de *Mycoplasma pneumoniae* (Guell *et al*., 2009), 2165 de *Sulfolobus solfataricus* (Wurtze *et al*., 2010) y 522 de *Listeria monocytogenes* (Toledo-Arana *et al*., 2009), obteniéndose 1199, 341, 2165 y 1157 pares de genes operones, respectivamente.

Cabe señalar que mientras los operones de *E. coli*, *B. subtilis* y los de los 50 genomas parcialmente estudiados fueron determinados y curados manualmente, los de *H. pylory*, *M. pneumoniae*, *S. solfataricus* y *L. monocytogenes* fueron determinados de manera global utilizando información generada por tecnologías de secuenciación masiva. Dichas tecnologías aún se encuentran en una etapa inicial de desarrollo por lo que su precisión suele ser pobre y contener ruido. Esta falta de precisión en los datos imposibilitó la capacidad de definir bordes de operones y en consecuencia el conjunto de pares de genes no-operones no pudo ser determinado, denotado por ”NA” en la Tabla 4.2.

CAPÍTULO 5

IDENTIFICACIÓN BASADA EN DISTANCIAS INTERGÉNICAS Y STRING

El desempeño de un clasificador depende fuertemente de los atributos empleados para generar el conjunto patrón de entrada, donde discrepancias fuertes entre los atributos ayudan a la tarea de discriminar un objeto de otro. En particular, en primer lugar para identificar un par de genes en la clase *operón* (o) o *no-operón* ($\overline{o}$), se propuso un método basado en una Red Neuronal Perceptrón Multicapa (MLP, por sus siglas en inglés *Multilayer Perceptron*) que utiliza un conjunto patrón formado por dos vectores de atributos: distancias intergénicas (Y_1^b) y la relación funcional de proteína (Y_2^b) predefinida por la BD de STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) (Jensen *et al*., 2009). Considerando lo descrito en la sección4.1, los vectores son obtenidos a partir de las relaciones entre características individuales de cada par de genes de un genoma b, es decir $\mathcal{Y}^b = (Y_1^b, Y_2^b)$. Por lo tanto, utilizando lo descrito en la sección 4.1.1, el problema se define como la asignación de $\mathcal{Y}^b$ mediante una MLP al espacio objetivo $C = (o, \overline{o})$ con una tasa de error baja.

La BD STRING refleja la relación funcional de los diferentes grupos de genes ortólogos basada en la integración ponderada de siete variables genómicas. Por consiguiente, al utilizarse los valores ponderados de dicha BD, los genes g_i^b considerados en este análisis fueron elementos de los distintos elementos $\mathcal{COG}_t$, es decir con un grupo de genes ortólogos definido. Cabe señalar que este trabajo fue publicado en Taboada *et al*. (2010).

5.1. Características utilizadas

Inicialmente, se propuso una matriz patrón $\mathcal{Y}^{b*}$ (Ec. 4.7) formada por tres características Y_j^b (Ec. 4.6) de relaciones específicas entre pares de genes: distancias intergénicas (Y_1^b), valor

ponderado de STRING (Y_2^b) y dirección de transcripción (Y_3^b). Sin embargo, dado que un par de genes que pertenecen a un mismo operón forzosamente deben estar en la misma dirección de transcripción, Y_3^b fue usado en la pre-clasificación donde pares de genes que están en direcciones contrarias automáticamente se asociaron a la clase $\overline{o}$. De este modo, $\mathcal{Y}^b = \{Y_1^b, Y_2^b\}$ fue la matriz utilizada para discriminar pares de genes como o o $\overline{o}$.

5.1.1. Distancias intergénicas

Establece la distancia en pares de bases (bp) de dos genes adyacentes de un organismo específico, dada a partir de sus posiciones derecha $x_{5,i+1}^b$ e izquierda $x_{4,i}^b$ respectivas (propiedades de los genes definidas en la sección 4.1), correspondientes a la dirección $5' \rightarrow 3'$ ($x_{3,i+1}^b$) del genoma. Utilizando la ecuación 4.5, las distancias intergénicas pueden expresarse como:

$$y_{1,i}^b = x_{4,i}^b \otimes x_{5,i+1}^b = x_{5,i+1}^b - x_{4,i}^b \tag{5.1}$$

con dominio en los enteros $\mathbb{Z}$.

Para utilizar las distancias intergénicas en la clasificación de operones se requiere establecer un umbral y calibrar su grado de confianza. Para ello, se realizó un análisis de las distancias intergénicas del conjunto de datos de la bacteria de *E. coli* descrito en la sección 4.4 de pares de genes $\mathcal{OP}^{eco}$ y $\overline{\mathcal{OP}}^{eco}$, es decir que pertenecen a la clase o y a la $\overline{o}$, respectivamente. La Figura 5.1 muestra los resultados donde se puede observa que el $4\,\%$ de lo pares de genes que perecen al conjunto $\mathcal{OP}^{eco}$ tienen una distancia intergénica menor a 50 bp, mientras que para pares de genes $\overline{\mathcal{OP}}^{eco}$ es del $69\,\%$.

Asimismo, se determinó si la tendencia de distancias intergénicas menores para pares de genes que pertenece a la clase o y mayores para la clase $\overline{o}$ no estaba restringida sólo a *E. coli*, repitiendo el análisis en la bacteria de *B. subtilis* la cual es filogenéticamente distante a *E. coli* y en un conjunto de 914 genomas bacterianos completamente secuenciados y disponibles. Como se puede observar en la Figura 5.1, la distribución de los promedios de todas las medias de las distancias intergénicas de estos pares de genes directones (adyacentes y en la misma dirección de transcripción) de diversos organismos, así como de *B. subtilis*, es muy similar a la distribución de los de *E. coli*, estando éstas casi traslapadas. Estos resultados sustentaron a la distancia intergénica como una relación útil, ponderada con otras, para la discriminación de operones dado que es similar para otros organismos bacterianos y no solamente para *E. coli*. Sin embargo, al considerar solamente las distancias intergénicas, se corría el riesgo de seleccionar un umbral para Y_1^b que genera muchos falsos positivos, ya que por ejemplo en *E. coli* se podía llegar a pasar el $31\,\%$ de los pares de genes de la clase o a la $\overline{o}$. Este hecho

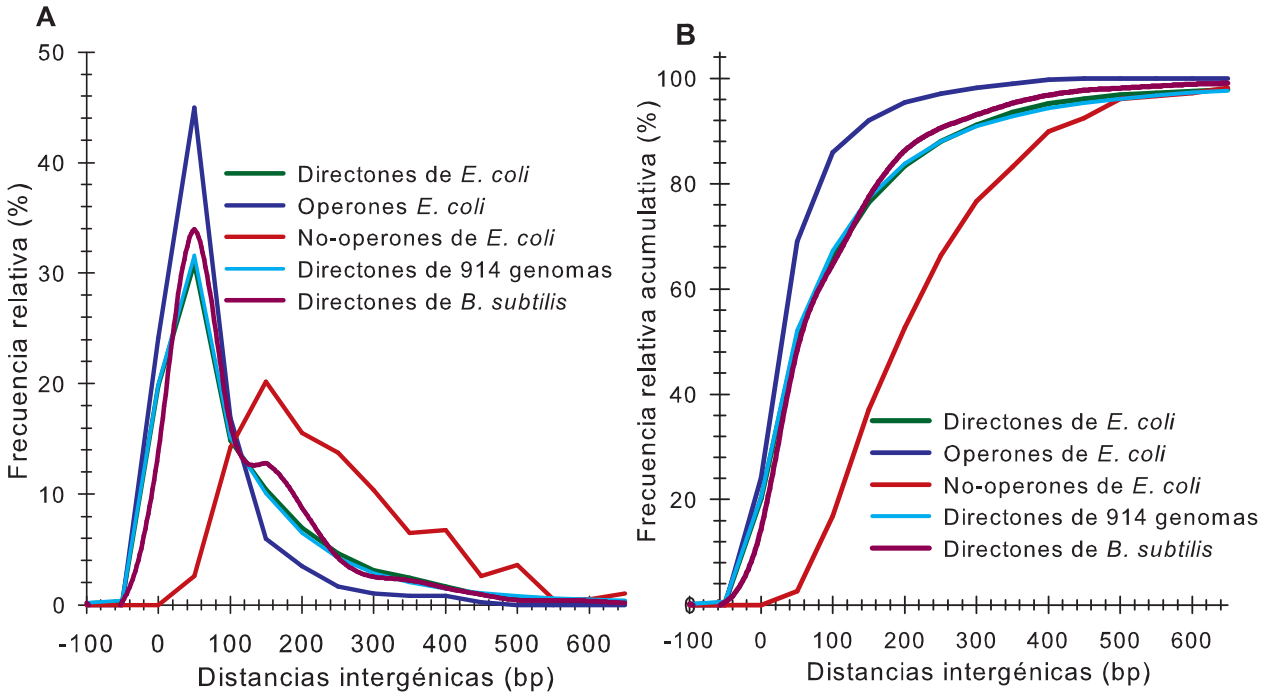

Figura 5.1: Distribución de las frecuencias de distancias intergénicas de pares de genes operones, no-operones y directones de *E. coli*, directones de *B. subitilis* y promedio de las medias de los directones de 914 organismos bacterianos. (A) Frecuencia relativa. (B) Frecuencia relativa acumulativa.

justificó la propuesta de usar además otras relaciones entre genes para la identificación de operones.

5.1.2. Relación funcional de STRING

Con el objetivo de tener un método de identificación de operones universal, se hizo uso como característica de entrada de los valores de asociación funcional de los diversos grupos de genes ortólogos definidos en la BD de STRING (Jensen *et al*., 2009). Estos valores de STRING están ponderados por siete tipos de características genómicas, descritas en detalle en la sección 6.2, que permiten inferir una relación de genes ortólogos en un proceso similar de la célula y son: **a)** ***Vecindad genómica conservada:*** genes que están en varias ocasiones en vecindad cercana en distintos genomas. **b)** ***Acontecimientos de fusión de genes:*** genes que se han unidos para codificar una sola proteína. **c)** ***Co-ocurrencia filogenética:*** concordancia en la presencia-ausencia de genes ortólogos en genomas diferentes para determinar similitudes en la historia de su evolución. **d)** ***Coexpresión:*** genes que muestran una respuesta transcripcional similar en condiciones variables, la cual puede ser evaluada mediante análisis de micro-arreglos (Bockhorst *et al*., 2003) **e)** ***Información experimental:*** relación que existe entre genes al

realizar una misma función, lo cual es probado experimentalmente. **f)** ***Información de diversas BD:*** STRING importa el conocimiento de la asociación de proteínas de diversas bases de datos. **g)** ***Minería de datos:*** Se busca co-mención relevante estadística de genes en resúmenes de trabajos de la base de datos de PUBMED.

En la BD de STRING para poder establecer los valores ponderados de asociación funcional, primero se considera que un grupo arbitrario de genes ortólogos t definido en la BD COG de NCBI (Tatusov *et al*., 2003) de un gen $g_i^b(x_{10,i}^b)$, está definido por:

$$\mathcal{COG}_t = \{g_i^b \in G^b \text{ con } b = 1, \ldots, |\mathcal{G}|\} \tag{5.2}$$

con todos los g_i^b descendiendo de un ancestro común y realizando funciones similares, a pesar de pertenecer a genomas diferentes.

Por consiguiente, los valores de STRING representan no sólo la relación funcional de un par de genes (g_i^b,g_{i+1}^b), si no que se extiende la noción para todos los pares de genes de distintos genomas que pertenecen a sus respectivos grupos $\mathcal{COG}_t$. Dado lo anterior y utilizando la ecuación 4.5, la segunda característica utilizada para identificar operones se define como:

$$y_{2,i}^b = x_{10,i}^b \otimes x_{10,i+1}^b = \{\mathcal{COG}_s, \mathcal{COG}_t, s_t\} \tag{5.3}$$

donde $x_{10,i}^b$ es el grupo de genes ortólogos al que pertenece el gen y las tripletas $\{\mathcal{COG}_s, \mathcal{COG}_t, s_t\}$ se obtienen directamente de la BD de STRING, definidas en este trabajo como:

$$\mathcal{S} = \big\{\{\mathcal{COG}_1, \mathcal{COG}_2, s_1\}, \ldots, \{\mathcal{COG}_s, \mathcal{COG}_t, s_a\}\big\} \tag{5.4}$$

donde el valor de s es la ponderación de las siete características genómica antes mencionadas, el índice a representa el número de combinaciones posibles con dos elementos del conjunto $\mathcal{COG}$ y s_t definido en el intervalo $[150, 999]$. Considerando que la BD COG tiene $4,873$ grupos de ortología, en teoría deberían existir $11,870,628$ diferentes combinaciones. Sin embargo, en la BD de STRING sólo $2,870,628$ tienen valores significativos de modo que $a = 1, \ldots, 2,870,628$, lo que implica que existen relaciones entre grupos de genes ortólogos que no tiene un valor s_a establecido por falta de información o información no significativa. Para estos casos, o para aquellos genes que no tienen un $\mathcal{COG}_t$ definido, la asignación de un par de genes a la clase o o $\overline{o}$ se realiza mediante la metodología descrita en el capítulo 7.

Como se ha mencionado a lo largo de esta tesis, los genes ortólogos típicamente tienen funciones semejantes a pesar de estar en diferentes especies debido a que descienden de un

ancestro común. Esto permite suponer, que el valor de STRING (Y_2^b) entre dos grupos de genes ortólogos es mayor para genes que pertenecen a un mismo operón que para aquellos que no. Al igual que en Y_1^b, se realizó un análisis de Y_2^b. Los resultados se muestran en la Figura 5.2 donde se identifica que para pares de genes de *E. coli* que pertenecen a la clase o, el valor Y_2^b está generalmente arriba de 900, mientras que para los de la clase $\overline{o}$ este valor está debajo de 400. Con ésto, se corroboró la tendencia de valores de STRING grandes para pares de genes operones y menores para no-operones.

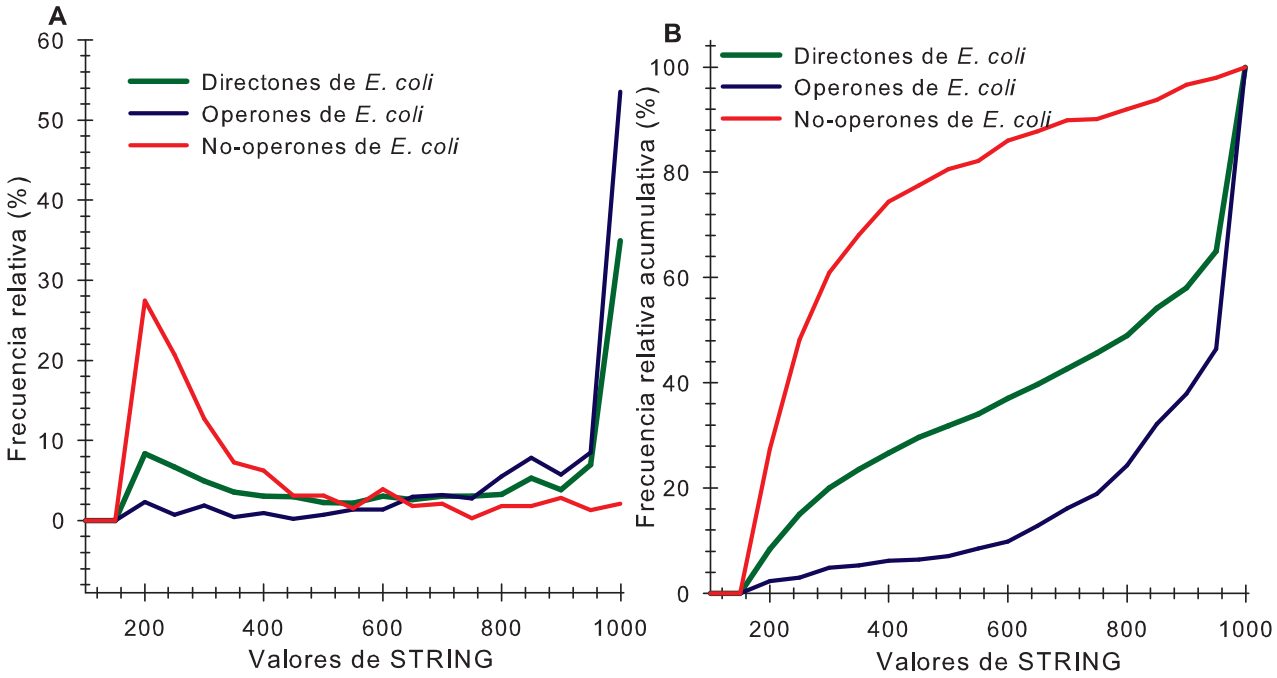

Figura 5.2: Distribución de las frecuencias de los valores ponderados de STRING de pares de genes operones, no-operones y directones de *E. coli*. (A) Frecuencia relativa. (B) Frecuencia relativa acumulativa.

5.1.3. Dirección de transcripción

Define la dirección de transcripción de pares de genes contiguos en el genoma denotado por:

$$y_{3,i}^b = \begin{cases} 1 & \text{si } x_{3,i}^b = x_{3,i+1}^b \\ 0 & \text{en otro caso} \end{cases} \tag{5.5}$$

con dominio $[0, 1]$. Así, esta característica estipula si dos genes están en la misma dirección y son contiguos.

Dado que un par de genes que pertenecen a un mismo operón forzosamente deben estar en la misma dirección de transcripción en el genoma, esta característica es usada para discriminar pares de genes operones ($y^b_{3,i} = 1$) de los no-operones ($y^b_{3,i} = 1$). Por lo tanto, Y^b_3 es usado inequívocamente en una pre-clasificación donde pares de genes que están en direcciones contrarías automáticamente se asocian a la clase *no-operón*.

5.2. Procedimiento de identificación

El procedimiento para identificar los operones de un genoma determinado, considerando como entrada a la red MLP el arreglo $\mathcal{Y}^{eco} = (Y^{eco}_1, Y^{eco}_2)$, se describe a continuación.

5.2.1. Pre-procesamiento

Como se mencionó en la sección 4.2, para cualquier problema de reconocimiento de patrones el punto de partida es el preprocesamiento de los datos de entrenamiento $\mathcal{Y}^b$ mediante la eliminación de los valores atípicos de cada Y^b_j y la normalización de los mismos. Para eliminar los valores atípicos, se hizo uso de la ecuación 4.8, donde se asumió un $\beta = 3$ para tener una cobertura del 99 %, removiendo sólo el 1 % de los datos tanto de Y^{eco}_1 como de Y^{eco}_2. Después de que los valores extremos fueron eliminados, cada Y^b_j fue normalizado en el intervalo de $[-1; 1]$ utilizando la ecuación 4.9, siendo $R_{min} = -1$ y $R_{max} = 1$.

5.2.2. Identificación de operones

Como se mencionó en el sección 3.1, en este trabajo se ha optado por utilizar redes neuronales perceptrón multicapa (MLP por sus siglas en inglés, MultiLayer Perception), las cuales han demostrado ser aproximadores universales de funciones (sección 3.2) (Cybenko, 1989; Haykin, 1999) y una alternativa eficiente para la identificación de operones. A pesar de que las maquinas de soporte vectorial (SVM por sus siglas en inglés, Support Vector Machine) tienen la ventaja de tener en su espacio de búsqueda de soluciones sólo un mínimo global mientras que en las redes MLP existen múltiples mínimos locales, los estudios comparativos realizados han demostrado que ambas son buenas soluciones en problemas de clasificación, regresión y tareas de predicción. Sin embargo, para conjuntos de datos de entrenamiento no muy grandes como el de operones (sección 4.4) las redes MLP han demostrado tener mejor desempeño. Además, las redes MLP tienen una más arquitectura sencilla empleando un número muy pequeño de neuronas ocultas mientras las SVM utiliza gran número de éstas.

Considerando lo descrito en el Capítulo 3, en este trabajo el diseño de la MLP para identificar operones consistió en:

i) **Selección del algoritmo de aprendizaje.** Al tener datos de entrenamiento de pares de genes de la clase *operón* (o) y *no-operón* ($\overline{o}$), se determinó utilizar un aprendizaje supervisado donde cada patrón o dato de entrada que recibe la red es comparado con un dato específico, de tal manera que la red modifica los pesos gradualmente. En cada paso de la fase de entrenamiento se actualizan los pesos hasta que el error entre la salida de la red y los datos establecidos se reduce. En este caso los pesos se obtienen minimizando la función de error, la cual mide la diferencia entre los valores establecidos y los valores calculados por la red. Para ésto, se pueden emplear varios algoritmos, el utilizado en este trabajo es el $Backpropagation$ (Kim *et al.*, 1996) (ver el Capítulo 3 para detalles). Básicamente, el algoritmo se divide en tres fases: a) recibir los datos de entrenamiento y enviarlos a las capas ocultas, b) evaluar la función de activación y calcular el error de la red, y c) y propagar el error hacia las capas intermedias anteriores y ajustar los pesos. En cuanto al valor de la tasa de aprendizaje (η) (Eq. 3.10 y Eq. 3.11), que está entre 0 y 1, se deben evitar dos extremos: un ritmo de aprendizaje demasiado pequeño que puede ocasionar una disminución importante en la velocidad de convergencia y la posibilidad de acabar atrapado en un mínimo local, o un ritmo de aprendizaje demasiado grande que puede conducir a inestabilidades en la función de error, lo cual evitará que se produzca la convergencia debido a que se darán saltos en torno al mínimo sin alcanzarlo. En general, el valor de la tasa de aprendizaje suele estar comprendida entre 0.05 y 0.5 (Rumelhart, D.E., 1986). En este trabajo se fijaron todos los parámetros que inciden en el entrenamiento de la red salvo η, y a base de prueba y error, variando el valor de η en 0.05 en el rango de 0.05 a 0.5, se llegó a la conclusión de que el mejor rendimiento se logró con un η =0.1. Por otra parte, siguiendo el mismo proceso, el factor momento α (Eq. 3.12 y Eq. 3.12) que suele tomar un valor próximo a 1 (Rumelhart, D.E. (1986)) se definió en 0.8.

ii) **Selección adecuada de una arquitectura.** Se probaron varias arquitecturas considerando lo descrito en el Capítulo 3.2.1 en cuanto al número de capas ocultas y neuronas por capa que se deben utilizar. En las redes MLPs, entre más pesos se tengan en relación al tamaño del conjunto de entrenamiento más posibilidad se tiene de memorización. Por lo tanto, entre menos capas y neuronas ocultas se tenga, con un alto desempeño, menos probabilidad de sobreajuste (Gori and Tesi, 1992; Hecht-Nielsen, 1995). En este sentido, se seleccionó una red MLP con una sola capa oculta que es lo mínimo para tener un aproximador universal de funciones (3.3). En cuanto al número de neuronas en la capa oculta, fue determinado según la heurística de Masters (1993) donde para un MLP de tres capas con M_j neuronas de entrada y M_l neuronas de salida, se

propone una capa culta con $sqrt(M_j * M_l)$ neuronas, pudiendo variar hasta tres veces esta medida. A partir de $sqrt(M_j * M_l)$, se fue ajustando el número de neuronas según los resultados del error obtenido en el conjunto de entrenamiento y prueba. La red MLP 2-3-1 neuronas(s) fue la que mejor desempeño tuvo con el menor número de neuronas y capas posible. Por lo tanto, $M_j = 2$, $M_k = 3$ y $M_l = 1$. En cuanto a la función de activación para las neuronas de la capa oculta se uso tangentes hiperbólicas, por lo tanto $F_k(x) = \frac{e^{2x}-1}{e^{2x}+1}$. Esta función es utilizada por su característica de derivación, ya que la propia función y su derivada son fáciles de computar lo cual es indispensable al utilizar el algoritmo $Backpropagation$. Además, tiene una amplia parte lineal para lograr velocidad de entrenamiento y convergencia en pocos ciclos y finalmente se recomienda para problemas de predicción o clasificación (Isa *et al*., 2010). Para la neurona de salida, la función es lineal para restringir los valores de salida al intervalo $[0, 1]$, entonces $F_j(x) = F_l(x) = x + b$.

iii) Evaluación de la MLP. Considerando lo descrito en el Capítulo 3.3, se realizó una validación cruzada (Stone, 1974) dividiendo al azar el arreglo patrón $\mathcal{Y}^b$ en 75 % para entrenamiento y 25 % para prueba, probando que la red no estuviera estancada en un mínimo local al satisface el porcentaje de error permitido en el conjunto de datos de entrenamiento y de prueba. Además, para estimar la generalización de la red MLP (Bengio and Grandvalet, 2004), se hizo una validación cruzada k-fold. Para ello, $\mathcal{Y}^b$ se dividió aleatoriamente en k particiones mutuamente excluyentes y de dimensiones aproximadamente similares. La red MLP fue entrenada y probada k veces, con k=7. En cada caso, una de las partes se consideró como datos de prueba y los restantes seis se añadieron a los datos de entrenamiento. Por lo tanto, existen k resultados diferentes de prueba.

iv) Estimación de un valor de confianza. El resultado del reconocimiento de operones es binario, es decir 0 ó 1; 1 para los pares de genes que pertenecen a la clase o y 0 para pares de genes que pertenecen a la clase $\overline{o}$. Sin embargo, se estimó un valor de confianza cv asociado a cada resultado, el cual está normalizado entre 0 y 1. Un valor superior a 0.5 indica que el par de genes pertenece a la clase o, mientras uno menor a la clase $\overline{o}$. En este sentido, los resultados de reconocimiento tienen mayor exactitud cuando los valores de confianza están cerca de 0 o de 1, y los de menor precisión están cerca de 0.5.

Para la implementación de las distintas redes MLP, se utilizó el software comercial NeuroIntelligence (Alyuda, 2005) debido a su potencialidad y fácil uso en la implementación de redes neuronales. NeuroIntelligence permite a los expertos en modelización predictiva dar soluciones a problemas del mundo real en áreas tales como minería de datos y reconocimiento

de patrones, como el caso de identificación de operones.

5.2.3. Evaluación del desempeño

Los resultados de la red MLP fueron evaluados utilizando las medidas de eficiencia descritas en la sección 4.3, donde los datos utilizados en los conjuntos TP, TN, FP y FN son especificados en la sección 4.4. Al utilizar la información de la BD de STRING, los genes g_i^b considerados en $\mathcal{Y}^b$ tenían que tener definido un grupo de genes ortólogos. Para ésto, de los conjuntos $\mathcal{OP}^{eco}$ y $\overline{\mathcal{OP}^{eco}}$ sólo se utilizaron los pares de genes, donde ambos genes tuvieran un $\mathcal{COG}_t$ (Ec. 5.2) determinado (Columna 6 de la Tabla 4.2). Para *E. coli* se obtuvieron 435 pares de genes de un total de 493 de $\mathcal{OP}^{eco}$ y 309 de un total de 386 de $\overline{\mathcal{OP}^{eco}}$, mientras para *B. subtilis* se obtuvieron 527 de 696 y 276 de 433 de $\mathcal{OP}^{bsu}$ y $\overline{\mathcal{OP}^{bsu}}$, respectivamente.

5.3. Resultados

En función a los objetivos planteados en este trabajo de doctorado: **i)** Se estimó la precisión para reconocer pares de genes operones y no-operones de *E. coli*. **ii)** Se evaluó si el método y las características genómicas tienen un desempeño generalizado para identificar operones en otros genomas bacteriano diferentes al utilizado como entrenamiento.

5.3.1. Desempeño con datos de *E. coli*

Como se mencionó en la 5.1.1, la distancia intergénica entre genes adyacentes se ha convertido en una de las características más utilizadas en la identificación de operones. La precisión más alta alcanzada en *E. coli* teniendo en cuenta esta característica, se ha reportado que es de 74 % (Salgado *et al*., 2000). En este trabajo, se obtiene una precisión similar de 80 % con una red MLP diseñada específicamente para sólo utilizar las distancias intergénicas (Y_1^{eco}) como patrón de entrada. Como era de esperarse, la precisión aumentó significativamente cuando además de Y_1^{eco}, el conjunto patrón $\mathcal{Y}^{eco}$ también incluía las relaciones funcionales entre grupos de genes ortólogos de la BD de STRING (Y_2^{eco}). La precisión alcanzada por la red MLP 2-3-1 en este conjunto de datos fue del 94.5 %.

Como se puede ver en la matriz de confusión de la Tabla 5.1, sólo en un número pequeño de casos se encontró que no fueron coherentes con la información recopilada en la BD de RegulonDB (Gama-Castro *et al*., 2008). El número de pares de genes de *E. coli* de la clase o identificados como $\overline{o}$ fue sólo de 22 de 435, dando una sensibilidad del 95.0 % (Ec. 4.10). Por otra parte, el número de pares de genes $\overline{o}$ identificados como o fue de 19 de 309, lo cual resulta

en una especificidad del 93.8 % (Ec. 4.11), siendo la precisión de 94.5 % (Ec. 4.12). Con este resultado se logró uno de los objetivos planteados en este trabajo de doctorado, el de obtener un método de reconocimiento de operones que fuera más preciso que lo reportado anteriormente.

		Predicho	
		No-operón	**Operón**
Actual	**No-operón**	285	19
	Operón	22	412

Tabla 5.1: Matriz de confusión de pares de genes operones y no-operones de *E. coli* en término de distancias intergénicas y valores de relación funcional de STRING

*Utilizando una MLP 2-3-1 entrenada con sólo datos de *E. coli*

Gene1	**Gene2**	**Clase**	**Clase predicha**	**Valor de confianza**
gabT	gabP	o	$\overline{o}$	0.00
truA	dedA	o	$\overline{o}$	0.01
gadX	gadA	o	$\overline{o}$	0.02
fic	papA	o	$\overline{o}$	0.02
gadE	mdtE	o	$\overline{o}$	0.03
dksA	afsA	o	$\overline{o}$	0.04
dxs	yajO	o	$\overline{o}$	0.09
rpmB	yicR	o	$\overline{o}$	0.11
mltC	nupG	o	$\overline{o}$	0.12
yiaL	yiaM	o	$\overline{o}$	0.14
yaiA	aroM	o	$\overline{o}$	0.22
mnmG	mioC	o	$\overline{o}$	0.32
relA	mazE	o	$\overline{o}$	0.33
glpK	glpX	o	$\overline{o}$	0.36
ydgK	rsxA	o	$\overline{o}$	0.39
IlsA	fkpB	o	$\overline{o}$	0.38
hypE	fhlA	o	$\overline{o}$	0.39
hydN	hypF	o	$\overline{o}$	0.43
yjcH	acs	o	$\overline{o}$	0.46

Tabla 5.2: Pares de genes operones en *E. coli* identificados incongruentemente con lo reportado en RegulonDB en términos de distancias intergénicas y valores de relación funcional de STRING

*El valor de confianza está normalizado entre 0 y 1. Un valor menor de 0.5 indica que el par de genes corresponde a la clase $\overline{o}$ en caso contrario a la clase o

Gene1	Gene2	Clase	Clase predicha	Valor de confianza
bglH	bglB	$\overline{o}$	o	0.50
tufB	secE	$\overline{o}$	o	0.51
ldcC	yaeR	$\overline{o}$	o	0.57
yfhH	yfhL	$\overline{o}$	o	0.58
yaeI	dapD	$\overline{o}$	o	0.60
efp	ecnA	$\overline{o}$	o	0.62
ygbE	cysC	$\overline{o}$	o	0.64
pcnB	yadB	$\overline{o}$	o	0.65
ubiB	tatA	$\overline{o}$	o	0.66
aat	cydC	$\overline{o}$	o	0.72
yhaO	tdcG	$\overline{o}$	o	0.86
lacZ	lacI	$\overline{o}$	o	0.88
yaaY	ribF	$\overline{o}$	o	0.90
uhpA	ilvN	$\overline{o}$	o	0.91
yadM	htrE	$\overline{o}$	o	0.99
glnB	glrR	$\overline{o}$	o	0.99
hscB	iscA	$\overline{o}$	o	0.99
folC	accD	$\overline{o}$	o	0.99
kdsD	kdsC	$\overline{o}$	o	0.99
folP	hflB	$\overline{o}$	o	1.00
yrdD	smg	$\overline{o}$	o	1.00
queA	tgt	$\overline{o}$	o	1.00

Tabla 5.3: Pares de genes no-operones en *E. coli* identificados incongruentemente con lo reportado en RegulonDB en términos de distancias intergénicas y valores de relación funcional de STRING

*El valor de confianza está normalizado entre 0 y 1. Un valor mayor de 0.5 indica que el par de genes corresponde a la clase o en caso contrario a la clase $\overline{o}$

La Figura 5.3 muestra el resultado de la identificación de los pares de genes $\mathcal{OP}^{eco}$ y $\overline{\mathcal{OP}}^{eco}$ de *E. coli*, donde dependiendo del valor de confianza son identificados como operones (o) en el intervalo de [0.5,1] o no-operones ($\overline{o}$) en el de [0,0.5). La Figura 5.3-A muestra la distribución de los pares de genes operones, mientras que la Figura 5.3-B corresponde a pares de genes no-operones. Como se puede observar, un gran número de pares de genes fueron correctamente clasificados en el área o intervalo que les corresponde de operones y no-operones, mientras que sólo pocos estuvieron en áreas incorrectas. En las Tablas 5.2 y 5.3 se muestra cada uno de estos casos mal identificados junto con su valor de confianza. En estos resultados, se puede observar que existen dos tipos de inconsistencia en términos del reconocimiento de operones:

a) Pares de genes cuyo valor de confianza de la predicción se encontró cerca de la frontera (valor de confianza de 0.5) donde la red MLP tiende a no distinguir correctamente entre la clase o y $\overline{o}$. Estas inconsistencias surgen como una consecuencia natural de estar cerca de los valores límite de un método de identificación binario. Vale la pena señalar, que

este conjunto de pares de genes es muy pequeño. Un ejemplo son el par de genes *lacI-lacZ* que son reconocidos por el método como pares de genes operones, sin embargo no lo son. Por una parte, existe el operón *lacZ-lacY-lacA* (de lactosa y el primero en ser descubierto) y por otra el operón *mhpR-lacI*. No obstante, el método define un sólo operón compuesto por cinco genes *mhpR-lacI-lacZ-lacY-lacA*. Como se menciona en el capítulo de Conclusiones y discusiones, existen terminadores transcripcionales que pudieran originar inconsistencias entre las predicciones del método de identificación y los operones que se generan *in vivo*.

b) Pares de genes cuyas predicciones de pertenecer a la clase *operón* o *no-operón* se encuentran cercanos a los valores de confianza 0 o 1 que corresponden a casos excepcionales que puedan surgir del uso incorrecto de anotaciones en el genoma o de la interpretación errónea de la datos experimentales. Un ejemplo de anotaciones erróneas en un genoma se encuentra en genes hipotéticos que no tienen datos experimentales para corroborar su existencia. Cuando este tipo de genes están cerca de genes reales, el método propuesto identifica estos pares de genes en la clase *operón*, aunque esto puede no ser así. Un ejemplo de este tipo de error corresponde a la inconsistencia encontrada en *ribF* y el gen hipotético *yaaY*, los cuales están separados por sólo 7 pb y por lo tanto el método los identifica como pares de genes operones. Sin embargo, en RegulonDB estos pares de genes son considerados como partes de diferentes operones debido a que el inicio de transcripción de *ribF* ha sido definido que está dentro de *yaaY*, lo caul apoya la posibilidad de que el gen *yaaY* no sea real (Kamio, Y. and Wu, 1985; Miller, K.W. and Wu, 1987). Por otra parte, un claro ejemplo de una inconsistencia debido a la preservación de datos imprecisos se encuentra en el par de genes *htrE-yadM* involucrados en el proceso de ensamblado de pilus. En *E. coli* estos genes están separados por sólo 16 pb y están comúnmente contiguos entre sí en diferentes genomas de proteobacterias. No obstante, basado en sólo un artículo con una caracterización parcial de *ecpD-htrE-yadM-yadL-yadK-yadC* (Raina, S. and Georgopoulos, 1993), estos genes están anotados en RegulonDB como parte de diferentes operones. Un segundo ejemplo de este tipo de datos probablemente mal curados o inexactos se encuentra en el par de genes *accD-folC* los cuales codifican para la subunidad carboxilasa acetil-CoA y la enzima folilpoliglutamato- sintetasa, respectivamente. En ciertas proteobacterias, la region intergénica de estos genes es muy pequeña o incluso inexistente, por lo tanto es muy probable que sean parte del mismo operón. Además de esto, la relación funcional entre sus correspondientes productos es muy alta, sin embargo en RegulonDB se consideran estos genes como parte de diferentes operones sobre la base de un sólo artículo en el que los autores sugieren sólo la naturaleza de esta relación (Li and Cronan, 1993).

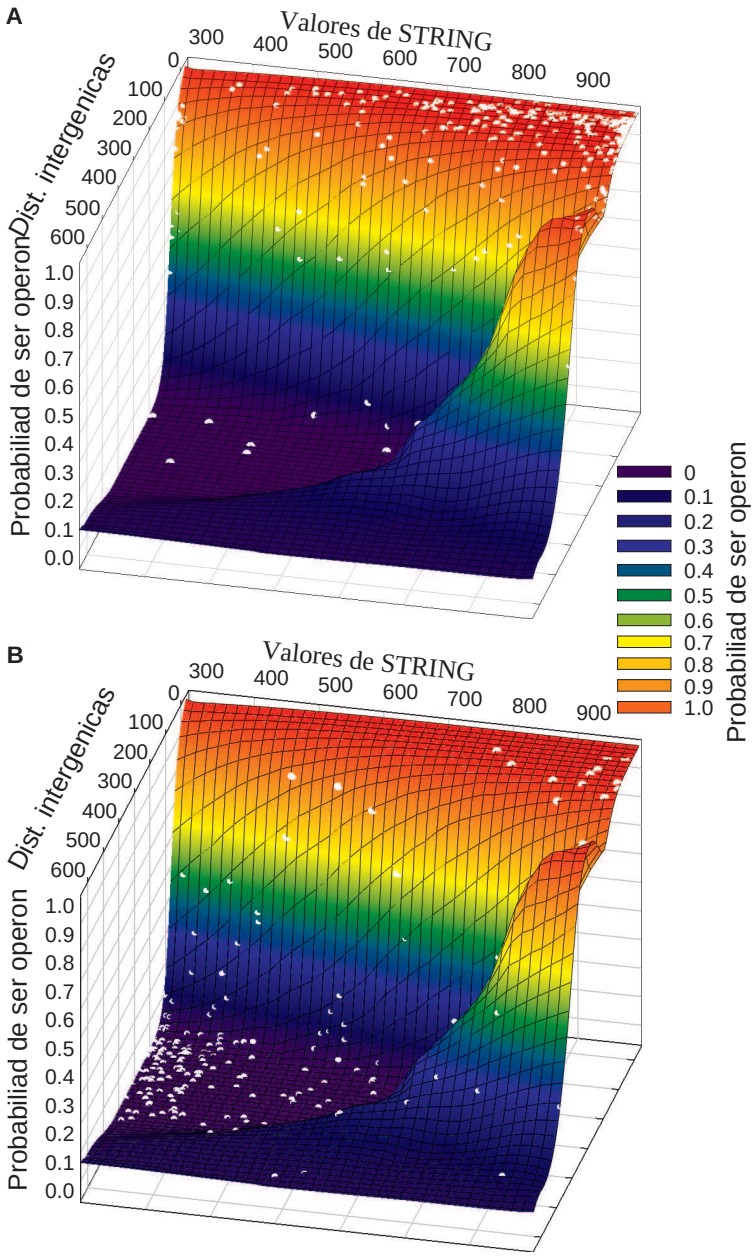

Figura 5.3: Resultados de identificación de operones en términos de distancias intergénicas y valores de relación funcional de STRING. (A) Distribución de pares de genes operones. (B) Distribución de pares de genes no-operones

Finalmente, en cuanto a la validación cruzada k-fold que se realizó (sección 5.2.2), el promedio de los errores en cada una de las siete pruebas fue de sólo 4.2 % con una desviación estándar de 0.7 %. Cabe mencionar, que esta validación cruzada se realizó independientemente siete veces, muestreando en cada una de éstas una fracción distinta del universo de datos. Con

lo que se garantizó, además de un error satisfactorio, que la red MLP no estuviera estancada en ningún caso en un mínimo local y sin un sobre-entrenamiento de los datos. Esto debido a que cuando una red generaliza y no esta en un mínimo local clasifica o aproxima de manera correcta, con el mínimo error ajuste, los datos de prueba que nunca han sido utilizados para su entrenamiento (Haykin, 1999).

5.3.2. Desempeño generalizado

Otro de los objetivos de este trabajo, era el desarrollar un método de reconocimiento de operones que pudiera ser utilizado en diversos genomas bacterianos sin que la precisión disminuyera significativamente, debido a que los trabajos reportados anteriormente no tienden a generalizar bien de un genoma a otro (ver sección 2.2.2).

Con el fin de probar el desempeño del método en otros organismos además de *E. coli*, se analizaron datos de diversos organismos bacterianos (Tabla 4.2). Para ésto, la red MLP fue primero entrenada exclusivamente con datos de *E. coli* y probada con datos de *B. subtilis*, *H. pylori*, *M. pneumoniae*, *S. solfataricus*, *L. monocytogenes* y en la recolección de operones de otros 50 organismos bacterianos. En el caso de *B. subtilis*, como se puede ver en la Tabla 5.4, se obtuvo una precisión ligeramente menor de 93.6 %, con una sensibilidad de 92.9 % y una especificidad del 94.9 %, lo cual sólo representa una reducción de 1.3 %. El peor de los casos se presentó en *L. monocytogenes* con una reducción del 2.6 % y en el mejor en *S. solfataricus* con un aumento del 3.3 %. Las disminución en la precisión representa un porcentaje pequeño en comparación de lo previamente reportado.

		Actual		
		E. coli	***B. subtilis***	***E. coli + B. subtilis***
Predicho	***E. coli***	94,5 %	92,1	94.9 %
	B. subtilis	93.6 %	93.9 %	94.3 %
	50 organismos	93.8 %	94.8 %	94.8 %
	H. pylori	93.6 %	94.1 %	94.3 %
	M. pneumoniae	94.4 %	93.9 %	94.4 %
	S. solfataricus	97.8 %	97.8 %	97.8 %
	L. monocytogenes	91.9 %	91.8 %	92.1 %

Tabla 5.4: Matriz de precisión en términos de distancias intergénicas y valores ponderados de STRING

Con el objetivo de evaluar la red MLP utilizando datos en el proceso de entrenamiento de un segundo organismo modelo diferente a *E. coli*, se utilizó a *B. subtilis*, probando con el mismo organismo y con el resto de los organismos. En el caso del mismo *B. subtilis* se obtuvo una precisión de 93.9 %, mientras en *E. coli* del 92.1 %. Al igual que cuando se entrenó con sólo datos de *E. coli*, la precisión más baja se obtuvo en la predicción de operones de *L. monocytogenes* y la más alta en *S. solfataricus*.

Por último, se utilizaron tanto los datos de *E. coli* como de *B. subtilis* con la finalidad de tener un conjunto de datos más grande y representativos de bacterias Gram positivas y Gram negativas en el proceso de entrenamiento. En este caso, como se observa en la Tabla 5.4, el desempeño generalizado es constante tanto en *E. coli*, *B. subtilis*, 50 organismos, *H.pylori* y *M. pneunoniae*, teniendo una reducción también sólo en *L. monocytogenes* y un aumento en *S. solfataricus*. Todos estos resultados demuestran que el método propuesto en este trabajo tiene un desempeño satisfactorio generalizado en la identificación de operones y que puede ser utilizado en diversos organismos bacterianos sin que la precisión disminuya inquisitivamente como en métodos reportados previamente.

CAPÍTULO 6

IDENTIFICACIÓN BASADA EN EL SUBCONJUNTO DE CARACTERÍSTICAS RELEVANTES

Hasta donde se sabe, la precisión más alta reportada para identificar operones en *E. coli* es de 94.6 %, obtenida por el método descrito en el capítulo anterior, usando como características de entrada el valor ponderado de STRING y las distancias intergénicas. Como se mencionó en la sección 5.1.2, los valores de STRING son ponderados mediate la integración de siete diferentes características genómicas. Desde el punto de vista de reconocimiento de patrones es importante determinar cuales de estas características son significativas para discriminar de la mejor manera posible la clase *operón* (o) de la *no-operón* ($\overline{o}$) a partir de la información que suministran. Esto permite eliminar las variables que puedan inducir errores (características ruidosas), no aporten información alguna que despeje la incertidumbre sobre la variable clase (características irrelevantes) o incluyen la misma información que otras, es decir que su valor puede ser determinado a partir de otras variables predictivas (características redundantes) (Blum and Langley, 1997).

En este sentido, a pesar de haber desarrollado un método para identificar operones que tiene una mayor precisión a lo reportado previamente, en este capítulo, se plantea un nuevo identificador que tiene como entrada el subconjunto de las características relevantes que aportan la mayor cantidad de información para discriminar la clase o de la $\overline{o}$. Para ésto, se utilizó un enfoque envolvente (sección 6.1) acoplando el método **Weight Explanatory** (*WE*) (Gevrey *et al*., 2003) de una red MLP para determinar la contribución relativa de cada una de las variables individuales de STRING y distancias intergénicas. Por lo tanto, ahora el problema a resolver es minimizar la cardinalidad del subespacio de características $\mathcal{Y}^{b*} \subset \mathcal{Y}^{b}$ por medio de una MLP, tal que exista otra MLP que asigne $\mathcal{Y}^{b*}$ al espacio objetivo C, con una tasa de error baja. Asimismo, se probaron otros dos métodos, uno de tipo filtro y uno híbrido (sección 6.1) mediante una correlación de Spearman y un árbol de decisión, respectivamente. Esto con

la finalidad de evaluar su desempeño y comparar los resultados con el método *WE*, mostrando la potencialidad de las redes MLP en la selección de características relevantes e identificación de operones en genomas bacterianos. Cabe señalar que al igual que en el método anterior, al utilizar los datos de STRING los genes g_i^b considerados en $\mathcal{Y}^b$ son elementos de los distintos grupos $\mathcal{COG}_t$.

6.1. Marco referencial de la selección de características

La selección o extracción de características se han convertido en interés de muchas investigaciones de reconocimiento de patrones en diversas áreas de aplicación como son la minería de datos, procesamiento de texto, análisis de expresión genética, entre otras, donde los datos contienen muchas variables potenciales que pueden ser reducidas con la finalidad de mejorar la precisión del modelo de reconocimiento. Mientras las técnicas de selección de características buscan el subconjunto óptimo eliminando variables redundantes o irrelevantes, las técnicas de extracción transforman el espacio de características original en un nuevo espacio que no es necesariamente un subespacio. En contraste con las técnicas de extracción, las técnicas de selección mantienen la representación original de las variables dado que simplemente seleccionan un subconjunto de éstas. De este modo, preservan la semántica original de las características, por lo tanto, ofrece la ventaja de interpretabilidad por un experto en el dominio (Guyon and Elisseeff, 2003; Lui and Yu, 2005). Además, generalmente los beneficios directos de seleccionar el subconjunto relevante de características abarca:

- Mejorar la comprensión del modelo de clasificación, ya que se induce el modelo con un gran número de variables.
- Disminuir los tiempos de procesamiento de los datos.
- Utilizar menor requerimiento en los espacios donde se almacena la información
- Tener la posibilidad de indagar en la naturaleza de distintos casos, debido al tamaño más manejable de cada instancia, identificar patrones en ciertos subconjuntos se vuelve más fácil.
- Tener un menor costo en la obtención de los datos.

Sin embargo, las técnicas de selección de características tienen algunas desventajas sobre las de extracción ya que éstas pueden proporcionar una mejor capacidad discriminatoria que las del mejor subconjunto seleccionado, resultando generalmente en un conjunto más pequeño y rico de atributos. Asimismo, las técnicas de selección son computacionalmente más costosas

debido a que para garantizar el mejor subconjunto se debe examinar todos los posibles subconjuntos. Por lo tanto, requieren de una estrategia de búsqueda y tener una función objetivo que evalué dichos subconjuntos. De este modo, en lugar de sólo optimizar los parámetros del modelo de reconocimiento de patrones para el subconjunto completo de características, ahora se tiene que encontrar el óptimo de los parámetros del modelo para el subconjunto de características relevantes, ya que no se puede garantizar que los parámetros óptimos para el conjunto completo de características sean igualmente óptimos para el subconjunto de características relevantes (Daelemans *et al*., 2003). Como resultado, la búsqueda en el espacio de hipótesis se complementa con otra dimensión: la de encontrar el subconjunto de características relevantes.

Las técnicas de selección de características difieren entre sí en la forma en que incorporan la búsqueda en el espacio añadido de subconjuntos del modelo de selección. En el conexo de clasificación utilizando aprendizaje supervisado donde las clases son conocidas de antemano, las técnicas de selección de características pueden ser organizadas en tres categorías dependiendo de cómo se combinan la función de búsqueda de selección con la construcción del modelo de clasificación: los métodos de filtrado, métodos envolventes y los métodos integrados o híbridos (Guyon and Elisseeff, 2003). La Tabla 6.1 presenta una taxonomía común de los métodos de selección de características, indicando para cada técnica las ventajas más importantes y desventajas, así como algunos ejemplos de las técnicas más influyentes.

Métodos indirectos o de filtrado (Dash *et al*., 2002; Liu and Setiono, 1996; Ben-Bassat, 1982): Evalúan la relevancia de las características por medio sólo de las propiedades intrínsecas de los datos, utilizando un heurístico o regla matemática para guiar su proceso de búsqueda hacia una solución. Es decir, el procedimiento de selección es realizado en forma independiente a la clasificación, filtrando las características irrelevantes antes que ocurra la etapa de clasificación. Posteriormente, este subconjunto de características es presentado como entrada a un algoritmo de clasificación. Como resultado, el proceso de selección de características es realizado una sola vez, y luego diferentes tipos de clasificadores pueden ser evaluados. Una de sus principales ventajas es que son simples y rápidos en el cálculo, hecho que hace que en conjuntos de datos con alta cardinalidad las aproximaciones de este tipo sean consideradas. Por otra parte, una desventaja común de estos métodos es que ignoran las interacciones con el clasificador (la búsqueda en el espacio de subconjuntos de características es separada de la búsqueda en el espacio de hipótesis), corriendo el riesgo de que la selección del subconjunto de características no sea óptima para el tipo de clasificador que se utiliza.

Métodos directos o envolventes (Kohavi and John, 1997; Caruana and Freitag, 1994):

Integra el modelo de búsqueda de hipótesis al modelo de búsqueda de selección del subcojunto de características. En estos métodos, varios subconjuntos de características se generan y evalúan. La evaluación de un subconjunto es generada por un modelo de clasificación específico que utiliza la tasa de error con dicho subconjunto de características. Sin embargo, como el espacio de subconjuntos de características crece exponencialmente con el número de características analizadas, los métodos heurísticos de búsqueda son utilizados para guiar la búsqueda a un subconjunto óptimo. Estos métodos de búsqueda se pueden dividir en dos clases: algoritmos de búsqueda deterministas y algoritmos de búsqueda aleatorios. Algunas ventajas de estos métodos evolventes incluyen la interacción entre la búsqueda del subconjunto de características y el modelo de selección y la habilidad de considerar las dependencias entre las variables. Algunas de sus desventajas son un costo computacional mayor a los indirectos y tener un cierto riesgo de sobreajuste.

Métodos híbridos (Das, 2001): Estos métodos son una combinación de los métodos de filtro y envolventes. La búsqueda del subconjunto relevante de características es realizada dentro de la construcción del clasificador y puede ser visto como la búsqueda en el espacio combinado de subconjuntos de características e hipótesis. Como los envolventes, este enfoque es específico para un algoritmo de aprendizaje.

6.2. Características a evaluar

La matriz patrón $\mathcal{Y}^b$ (Ec. 4.7) a evaluar está formada por las siete diferentes características genómicas de STRING: i) vecindad genómica conservada (Y_1^b), ii) fusion de genes (Y_2^b), iii) co-ocurrencia filogenética (Y_3^b), iv) co-expresión (Y_4^b), v) evidencia experimental (Y_5^b), vi) información de otras BD (Y_6^b), y vii) minería de datos (Y_7^b), además de las distancias intergénicas (Y_8^b). Para garantizar la generalización del método de identificación de operones, estas características son obtenidas a partir de las diferentes especies bacterianas y no sólo de una. Por otra parte, al igual que en el método descrito en el capítulo anterior, la dirección de transcripción fue usada en una pre-clasificación donde pares de genes que están en direcciones contrarias automáticamente se asociaron automáticamente a la clase $\overline{o}$ dado que un par de genes que pertenecen a un mismo operón forzosamente deben estar en la misma dirección de transcripción (sección 5.1.3). En los siguientes párrafos se describe cada una de las características, haciendo uso de las propiedades de los genes y operones definidos formalmente en 4.1. Cabe señalar que existen varias característica que utilizan la misma propiedad de los vectores de atributos $\mathcal{X}_i^b$ de los genes, cambiando únicamente la relación $\otimes$ que utilizan en

Tabla 6.1: Taxonomía de las técnicas de selección de características.

Modelo	Tipo	Ventajas	Desventajas	Ejemplos
Filtro	Univariable	Rápido Escalable Independiente del clasificador	Ignora dependencia de las características Ignora interacción con el clasificador	Chi-cuadrada Prueba T Distancia euclidiana
	Multivariable	Independiente del clasificador Bajo costo computacional	Más lento que los univariable Menos escalables que los univariable Ignora interacción con el clasificador	Divergencia entre clases Entropía de Shannon Filtro de correlación
Envolvente	Determinístico	Simple Interactúa con el clasificador Dependencia del modelo de selección Menos costo computacional que los aleatorizados	Riesgo de sobreajuste Más propensos a detenerse en un óptimo local que los aleatorizados Dependencia del clasificador	Redes neuronales con búsqueda hacia adelante atrás o exhaustiva Recorte de Caminos Distancia euclidiana
	Aleatorizado	Menos propensos a detenerse en un óptimo Interactúa con el clasificador Dependencia del modelo de selección	Cómputo intenso Dependencia del clasificador Ignora interacción con el clasificador Riesgo de sobreajuste	Algoritmos genéticos Recocido simulado Hill Climbing
Híbridos		Interactúan con el clasificador Menor costo computacional que los envolventes Dependencia del modelo de selección	Dependencia del clasificador	Árboles de decisión Modelos Bayesianos Maquinas de soporte vectorial

términos de la asociación de dichas propiedades (Eq. 4.5).

Conservación de vecindad: Tal como se mencionó en la sección 2.2.1, esta característica establece el grado de conservación genómica de un par de genes contiguos en la misma dirección de transcripción ($x^b_{3,i} = x^b_{3,i+1}$) a través de diferentes genomas bacterianos en relación a su posición en los genomas (x^b_2), por medio de:

$$y^b_{1,i} = x^b_{2,i} \otimes x^b_{2,i+1} \tag{6.1}$$

Fusión de genes: Ofrece información de la fusión de dos genes que codifiquen una sola proteína, proporcionando una fuerte asociación funcional, incluso en organismos en donde las dos proteínas están codificadas por genes diferentes (Marcotte *et al*., 1999). Esta característica se expresa como:

$$y^b_{2,i} = x^b_{6,i} \otimes x^b_{6,i+1} \tag{6.2}$$

Esto significa que es probable que los genes g^b_i y g^b_{i+1} de un organismo interactúen si sus homólogos se expresan como un sólo gen en otro organismo conteniendo información de ambos genes, demostrando una correlación entre los genes que interactúan y sus funciones.

Co-ocurrencia filogenética: Esta característica fue descrita en la sección 2.2.1 y evalúa la co-presencia o co-ausencia de un par de genes en un grupo de genomas de referencia usando x^b_6 para identificarlos y puede ser expresada como:

$$y^b_{3,i} = x^b_{6,i} \otimes x^b_{6,i+1} \tag{6.3}$$

Co-expresión: Establece si un par de genes es transcrito de manera similar, tal como se mencionó en la sección 2.2.1 y puede ser expresado como:

$$y^b_{4,i} = x^b_{9,i} \otimes x^b_{9,i+1} \tag{6.4}$$

Evidencia experimental: Establece la asociación de genes por medio de ensayos de interacciones experimentales directas, estableciendo la relación:

$$y^b_{5,i} = x^b_{9,i} \otimes x^b_{9,i+1} \tag{6.5}$$

La información experimental podría ser la característica más confiable para reconocer operones, pero su desventaja es la falta de datos experimentales para muchos pares de genes. STRING obtiene estos resultados mediante la importación de información de interacciones físicas de proteínas de otras bases de datos (Jensen *et al*., 2009): DIP: Database of Interacting Proteins (Salwinski *et al*., 2004), IntAct: Open Source Resource for Molecular Interaction Data (Kerrien *et al*., 2007) y PID: Pathway Interaction Database (Schaefer *et al*., 2009), entre otras.

Información de otras BDs: Apoya la asociación funcional de grupos de genes por interacciones indirectas, expresado como:

$$y^b_{6,i} = x^b_{9,i} \otimes x^b_{9,i+1} \tag{6.6}$$

Las interacciones indirectas pueden ser al compartir un substrato en una vía metabólica, por regulación transcripcional mutua o por participar en complejos multi-proteína grandes. STRING determina las interacciones indirectas utilizando otras BDs como:BioCyc: Pathway/Genome Databases (Karp *et al*., 2005), PID: Pathway Interaction Database (Schaefer *et al*., 2009) y KEGG: Kyoto Encyclopedia of Genes and Genomes (Kanehisa *et al*., 2008), entre otras.

Minería de datos: Define la co-ocurrencia de nombre de genes (x^b_1) en textos científicos:

$$y^b_{7,i} = x^b_{1,i} \otimes x^b_{1,i+1} \tag{6.7}$$

STRING incorpora minería de texto al analizar una gran cantidad de textos científicos buscando estadísticamente co-ocurrencias relevantes de los nombres de genes utilizando Procesamiento del Lenguaje Natural.

Distancias Intergénicas: Al igual que en el método descrito en el capítulo anterior (sección 5.1.1), establece la distancia en pares de bases (bp) de dos genes adyacentes de un organismo específico, que está dada a partir de sus posiciones derecha $x^b_{5,i+1}$ e izquierda $x^b_{4,i}$ respectivas, y es expresado como:

$$y^b_{8,i} = x^b_{4,i} \otimes x^b_{5,i+1} = x^b_{5,i+1} - x^b_{4,i} \tag{6.8}$$

6.3. Selección de un subconjunto características relevantes

Primero, se evaluó la asociación entre los vectores de características $Y_1^{eco}, Y_2^{eco}, \ldots Y_8^{eco}$ utilizando un método de filtro mediante un análisis de correlación de Spearman, para determinar cuales estaban altamente correlacionadas (Myers and Well, 2003). El coeficiente de correlación de Spearman es una medida no-paramétrica de dependencia estadística entre dos variables que evalúa que tan bien la relación puede ser descrita mediante una función monótona. La función f es monótona si y sólo si $x \leq y$ implica $f(x) \leq f(y)$ (es decir, la función es creciente), o bien $x \geq y$ implica $f(x) \geq f(y)$ (es decir, la función es decreciente).

El resultado de este análisis, sobre los datos de operones y no-operones de *E. coli* descritos en la sección 4.4, generó la matriz de coeficientes que se muestra en la Tabla 6.2. Como todos los coeficientes de correlación fueron menores de 0.6, se asumió que no existía una dependencia entre las características, es decir que las variables no eran redundantes, y por lo tanto no era necesario eliminar inicialmente ninguna. Asimismo, para tener un marco de referencia para el análisis de características relevantes se generó la Figura 6.1, la cual muestra la frecuencia relativa individual de cada variable de STRING y distancias intergénicas. En esta figura se puede observar que el procedimiento para discriminar pares de genes operones y no-operones no es una tarea sencilla, por lo tanto el uso de otras herramientas computacionales está justificada.

	$\mathbf{Y}_1^{eco}$	$\mathbf{Y}_2^{eco}$	$\mathbf{Y}_3^{eco}$	$\mathbf{Y}_4^{eco}$	$\mathbf{Y}_5^{eco}$	$\mathbf{Y}_6^{eco}$	$\mathbf{Y}_7^{eco}$	$\mathbf{Y}_8^{eco}$
$\mathbf{Y}_1^{eco}$	1.00	0.29	0.44	0.43	0.38	0.37	0.57	-0.52
$\mathbf{Y}_2^{eco}$	0.29	1.00	0.30	0.25	0.14	0.34	0.23	-0.22
$\mathbf{Y}_3^{eco}$	0.44	0.30	1.00	0.27	0.25	0.38	0.43	-0.35
$\mathbf{Y}_4^{eco}$	0.43	0.25	0.27	1.00	0.36	0.38	0.52	-0.30
$\mathbf{Y}_5^{eco}$	0.38	0.14	0.25	0.36	1.00	0.26	0.41	-0.18
$\mathbf{Y}_6^{eco}$	0.37	0.34	0.38	0.38	0.26	1.00	0.40	-0.28
$\mathbf{Y}_7^{eco}$	0.57	0.23	0.43	0.52	0.41	0.40	1.00	-0.36
$\mathbf{Y}_8^{eco}$	-0.52	-0.22	-0.35	-0.30	-0.18	-0.28	-0.36	1.00

Tabla 6.2: Matriz de correlación de Spearman de datos de *E. coli*

Posteriormente, se implementó el método *Weight Explanatory* (WE) (Gevrey *et al*., 2003) de una red MLP para evaluar la contribución relativa de cada característica obteniendo el subconjunto relevante de las variables más significativas. Después, se implementó un árbol de

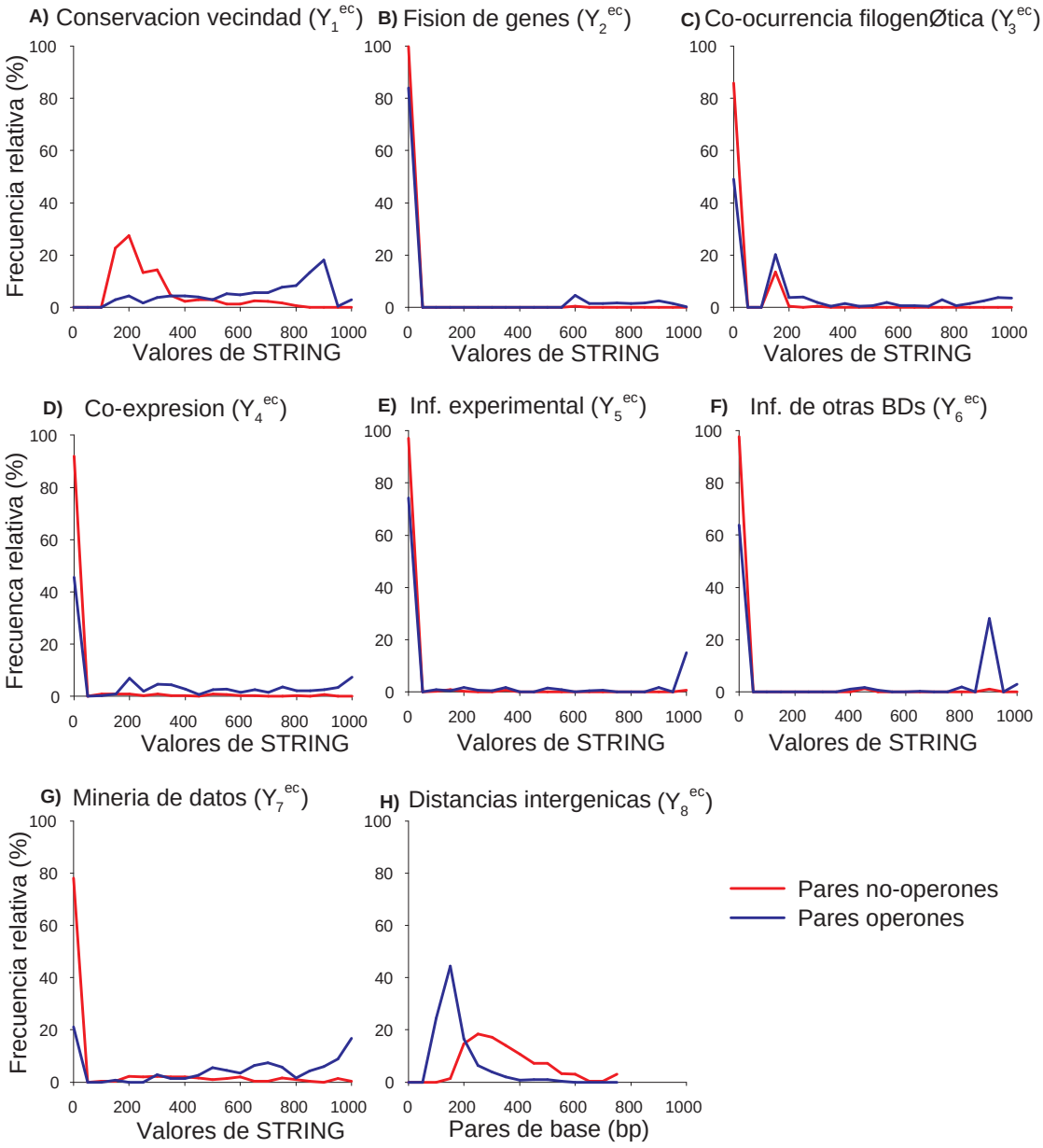

Figura 6.1: Frecuencias relativas de las características de STRING y distancias intergénicas de los conjuntos operones y no-operones de *E. coli*

decisión (DT, por sus siglas en inglés, *Decision Tree*), el cual es un método híbrido, para realizar un análisis exploratorio de las mejores asociaciones de las variables independientes y dependiente. Finalmente, se comparó los resultados de la identificación de operones obtenidos con ambos métodos para comparar sus desempeños y comprobar la potencialidad de las redes MLP en la selección de características e identificación de operones.

6.3.1. Subconjunto relevante generado por redes MLP

Como se describe en el Capítulo 3, se sabe que las redes MLP son aproximadores de funciones universales (Hornik *et al*., 1989) que han sido ampliamente utilizadas en problemas de clasificación. Las redes MLP también pueden ser utilizadas para cuantificar la contribución relativa de las características de entrada en un clasificador mediante diversos métodos. En Olden *et al*. (2004) se demostró que el método *Weight Explanatory* (*WE*) (Gevrey *et al*., 2003) exhibió el mejor desempeño de todos para estimar la verdadera importancia de todas las características en términos de su exactitud. Por lo tanto, fue el método seleccionado para estimar la contribución relativa de cada características de los operones.

Para ésto, la siguiente nomenclatura es usada: m_j son las neuronas de entrada indexadas por j, m_k son las neuronas ocultas indexadas por k (asumiendo que sólo existe una capa oculta) y m_l son las neuronas de salida indexadas por l. El peso entre la capa de entrada y la oculta, conectado la neurona de entrada j a la neurona oculta k es expresado como w_{jk}, y el peso entre la neurona oculta y la de salida, conectando la neurona oculta k a la neurona de salida l, es denotado como $w_{k,l}$. Para determinar la contribución relativa de las variables de entrada de una red MLP, el método *WE* se lleva a cabo en dos pasos:

a) Para estimar $Q_{j,k}$, que es el peso relativo de la característica j en la neurona oculta k, éste debe dividirse el valor absoluto del peso que conecta una neurona de la capa de entrada y otra de la oculta por la suma del valor absoluto de los pesos que conectan la capa de entrada y la oculta de todas las neuronas de entrada j para cada neurona oculta k. Esto significa:

Para $k = 1$ a m_k
  Para $j = 1$ a m_j

$$Q_{j,k} = \frac{\left|w_{j,k}\right|}{\sum_{j=1}^{m_j} \left|w_{j,k}\right|}$$

  Fin
Fin

b) Para estimar $\mathcal{RC}(j)$, que es una medida relativa de la importancia o contribución de la característica j con respecto al clasificador, ésta debe ser dividida para cada neurona de entrada j, la suma de $Q_{j,k}$ para cada neurona oculta por la suma de cada neurona oculta de la suma de cada neurona de entrada de $Q_{j,k}$, multiplicado por 100. Esto significa:

Para $j = 1$ a m_j

$$\mathcal{RC}(j)\,\% = \frac{\sum_{k=1}^{m_k} Qj,k}{\sum_{k=1}^{M_k}\sum_{j=1}^{m_j} Q_{j,k}} \times 100$$

Fin

Por lo tanto, si durante la fase de entrenamiento para identificar operones los ocho diferentes $\mathcal{RC}(j)$ son calculados, se obtiene la contribución relativa de cada característica Y_j^b. Entre más grande $\mathcal{RC}(j)$, más importante es la característica j que está asociada la neurona de entrada. Dependiendo del experimento en particular, el diseñador selecciona cuales características son relevantes para ajustar una buena clasificación.

Para la implementación de la red MLP, al igual que en el método anterior (ver sección 5.2.2), se hizo uso del software NeuroIntelligence (Alyuda, 2005). Se seleccionó una red MLP con una sola capa oculta que es lo mínimo para tener un aproximador universal de funciones (3.3). En cuanto al número de neuronas en la capa oculta fue determinado según la heurística de Masters (1993) donde para un MLP de tres capas con M_j neuronas de entrada y M_l neuronas de salida, se proponen $sqrt(M_j * M_l)$ neuronas ocultas, pudiendo variar hasta tres veces esta medida. A partir de $sqrt(M_j * M_l)$, se fue ajustando el número de neuronas según los resultados del error obtenido en el conjunto de entrenamiento y prueba. Una red neuronal de tres capas 8-9-1 fue la seleccionada, por lo tanto, $M_j = 8$, $M_k = 9$ y $M_l = 1$. Las funciones de activación para las neuronas de capa oculta fueron tangente hiperbólica y para la de salida lineal, con η =0.1 y α =.08.

6.3.1.1. Análisis de características relevantes

La contribución relativa $\mathcal{RC}(j)$ de cada característica $Y_j^{eco} \in \mathcal{Y}^{eco}$ en la identificación de operones de *E. coli* se muestra en la Columna 2 de la Tabla 6.3. La característica más importante fue la de conservación de vecindad (Y_1^{eco}), seguida de distancias intergénicas (Y_8^{eco}), minería de datos (Y_7^{eco}), co-expresión (Y_4^{eco}) e información de otras BD (Y_6^{eco}). Estas características juntas contribuyen en un 93.5 % en el rendimiento total del proceso de identificación de operones. Por otra parte, la contribución tanto de co-ocurrencia filogenética (Y_3^{eco}), información experimental (Y_5^{eco}) como fusión de genes (Y_2^{eco}) fue menor de 3.0 %, con valores de 1.7 % a 2.6 %, haciéndolas las características menos relevantes.

La Columna 4 de la Tabla 6.3 muestra que no todas las características tienen la misma cobertura en la BD STRING. STRING tiene cerca de $12,000,000$ registros que representan las relaciones funcionales entre grupos de genes ortólogos. De estos registros, por ejemplo, 54.1 % tienen un valor definido de co-ocurrencia filogenética (Y_3^{eco}), mientras sólo 0.1 % para fusión

Característica	Contribución relativa		Número de registros en STRING *	% en STRING DB
	E. coli	*B. subtilis*		
Conservación de vecindad (Y_1^{eco})	27.6 %	26.2 %	2'865,172	23.8 %
Fusión de genes (Y_2^{eco})	1.7 %	1.3 %	13,446	0.1 %
Co-ocurrencia filogenética (Y_3^{eco})	2.6 %	3.1 %	6'509,558	54.1 %
Co-expresión (Y_4^{eco})	10.2 %	9.8 %	965,938	8.0 %
Inf. experimental (Y_5^{eco})	2.2 %	2.8 %	473,736	3.9 %
Inf. de otras BDs (Y_6^{eco})	8.4 %	10.9 %	311,348	2.6 %
Minería de datos (Y_7^{eco})	22.4 %	20.6 %	2'223,096	18.5 %
Distancias intergénicas (Y_8^{eco})	24.9 %	25.3 %		

Tabla 6.3: Contribución relativa de las características y su cobertura relativa en la BD STRING
*La BD de STRING tiene en total 12,015,886 registros

de genes (Y_2^{eco}) (Tabla 6.3, columna 5). Esta gran diferencia en la cobertura es claramente un factor que influye en la contribución relativa de las variables en el proceso de identificación de operones. De la información mostrada en la Tabla 6.3, las siguientes observaciones pueden ser concluidas:

- La característica más informativa en el proceso de reconocimiento de operones fue la conservación de vecindad (Y_1^{eco}), con una contribución relativa de 27.6 %.
- Debido a su simplicidad y su importancia, las distancias intergénicas (Y_8^{eco}) ha sido la característica más utilizada en los métodos de identificación de operones e igualmente, en este trabajo, fue la segunda más importante (24.9 %).
- Minería de datos (Y_7^{eco}) tiene una contribución de 18.5 % y por consiguiente, fue la tercer característica más importante. A pesar de su alta contribución, hasta donde se sabe, esta es la primera vez que se ha utilizado en un estudio de reconocimiento de operones.
- Un caso particular es la co-ocurrencia filogenética (Y_3^{eco}) que es la más altamente representada en la BD STRING ya que está definida en 54.2 % de los registros. Sin embargo, la contribución de esta característica en la discriminación de pares de genes que pertenecen a la clase o o $\overline{o}$ fue de sólo 2.6 %, haciéndola irrelevante. Esto se puede corroborar en la Figura 6.1-c donde los valores para pares de genes operones y no-operones están casi traslapados. Es importante mencionar que esta característica ha sido ampliamente utilizada por varios métodos de identificación de operones (Dam *et al*., 2007; Zhang *et al*., 2006; Jacob *et al*., 2005; Westover *et al*., 2005).

- La baja contribución de fusión de genes (Y_2^{eco}) de 1.7 % es debido a que sólo el 0.1 % de los registros de STRING tienen definido este valor. En este momento, esta falta de información, no le permite discriminar si un par de genes pertenece a la clase o o a la $\overline{o}$.
- La información de otras BD (Y_6^{eco}) tiene una contribución importante en el proceso de identificación de operones (8.4 %) a pesar de su baja cobertura en STRING (sólo $2{,}6\,\%$ de sus registros). Hasta el momento, esta característica es la más confiable de todas las variables para discriminar entre pares de genes operones y no-operones.

Basándonos en los resultados arriba descritos, se decidió considerar a la conservación de vecindad (Y_1^{eco}), distancias intergénicas (Y_8^{eco}), minería de datos (Y_7^{eco}), co-expresión (Y_4^{eco}) e información de otras BD (Y_6^{eco}) como las características más relevantes y útiles, y descartar a Co-ocurrencia filogenética (Y_3^{eco}), información experimental (Y_5^{eco}), fusión de genes (Y_2^{eco}), que contribuyen en menos del 3 %. Al tener el valor de contribución relativo de cada una de estas variables, no fue fundamental el realizar una búsqueda exhaustiva para corroborar este subgrupo seleccionado. Por consiguiente, se decidió validar la selección mediante una modificación al método *forward stepwise* utilizando también redes MLP, añadiendo una nueva característica en cada paso de acuerdo a su relevancia. Al inicio, fue seleccionada Y_1^{eco} como la única característica de entrada debido a que es la de mayor relevancia. En cada paso siguiente, se añadió una nueva característica; la siguiente en función a su relevancia en la Tabla 6.3. Después de haber añadido Y_6^{eco}, no existió una mejora en la precisión, por lo que se comprobó que el resto de las características eran poco importantes en el proceso de clasificación de operones.

Finalmente, se realizó el mismo procedimiento pero ahora utilizando tanto los datos de *E. coli* como de *B. subilis* para corroborar que las características seleccionadas como relevantes no variaban al utilizar información de más organismos bacterianos. Los resultados comprobaron que a pesar de que existe una pequeña variación en la contribución de cada característica (± 1.5), el orden de importancia de cada una se mantuvo.

6.3.2. Subconjunto relevante generado por árboles CHAID DT

Los árboles de decisión (DT, por sus siglas en inglés, *Decision Tree*) son un método computacional que ha sido utilizados para la exploración de características relevantes en problemas de clasificación. Esto debido a su mecanismo de búsqueda de las mejores asociaciones de las variables independientes con la dependiente. Existen diversos algoritmos para diseñar los DT, tales como CART, CHAID, ID3, C4.5, entre otros. Estos algoritmos difieren uno de otro por el número de divisiones permitidas en cada nivel, como son

seleccionadas estas divisiones y como se limita el crecimiento del TD para evitar un sobre-entrenamiento. Sin embargo, lo característica más importante de los diversos algoritmos es el criterio para determinar las divisiones. En este trabajo, se decidió utilizar el algoritmo CHAID (por sus siglas en inglés, *Chi-squared Automatic Interaction Detector*) (Kass, 1980), ya que una diferencia importante del este algoritmo con respecto al resto es que busca la mejor combinación de las categorías de cada variable para dividir el nodo en curso (Ramaswami and Bhaskaran, 2010). Esto a pesar de ser computacionalmente costoso permite seleccionar las características más relevantes para la clasificación utilizando el estadístico *Chi-cuadrado*, el cual mide la divergencia entre clases Liu and Setiono (1995), asumiendo que la variable dependiente es categórica. Para ésto, la variable dependiente tiene $d \geq 2$ categorías y un predictor particular bajo análisis $c \geq 2$ categorías. Un subproblema en el análisis es la reducción de la tabla dada $c \times d$ a la tabla más significativa $j \times d$ mediante la combinación de las categorías de la variable predictora.

Los pasos lógicos que deben seguirse para realizar está tarea son los siguientes:

Paso 1 **Preparación de las variables**. Seleccionar una variable dependiente y un conjunto de posibles variables que permitan realizar una descripción y pronóstico óptimo de la primer variable.

Paso 2 **Agrupación de las categorías**. Agrupar las variables independientes en el caso que éstas tengan un perfil similar de la variable dependiente.

- Para cada posible par, se calcula el Chi-cuadrado correspondiente al cruce con la variable dependiente. El par con más bajo Chi-cuadrado, siempre que no sea significativo, formará una nueva categoría de dos valores fusionados. La condición de que no sea significativo es muy importante porque, en caso de que lo fuese, indicaría que las dos categorías que se pretenden fusionar no lo pueden hacer ya que son heterogéneas entre sí en los valores de la variable dependiente y el objetivo es justo lo contrario, asimilar categorías con comportamiento semejante.
- Si se ha fusionado un determinado par de categorías, se procede a realizar nuevas fusiones de los valores del pronosticador, pero esta vez con una categoría menos, pues dos de las antiguas han sido reducidas a una sola.
- El proceso se acaba cuando ya no pueden realizarse más fusiones porque los Chi-cuadrado ofrecen resultados significativos.

Paso 3 **Primera segmentación**. La segmentación es un proceso de partición recursiva, en el cual en cada nodo no terminal se toma la decisión de dividir la muestra de una cierta manera. En este caso, se divide en dependencia de la característica que mejor prediga la

variable dependiente, es decir la clase *operón-no-operón*. Para hacerlo, se calcula para cada característica su correspondiente Chi-cuadrada y se compara las significaciones obtenidas.

Paso 4 **Sucesivas segmentaciones**. Procede de forma similar al paso anterior en cada grupo formado por la segmentación previa. El proceso de segmentación debe ser examinado en sus distintas fases con el objetivo de valorar el comportamiento de los pronosticadores alternativos.

Paso 5 **Finalización del proceso de segmentación**. Definir límites al proceso de segmentación. Existen cuatro tipos de filtros que evitan la continuación de la segmentación: los de significación, los de asociación, los de tamaño y los de nivel.

- Filtros de significación. Su criterio consiste básicamente en no permitir segmentaciones que no sean estadísticamente significativas. Estos filtros pueden ser aplicados en la agrupación de categorías de una variables (fusión de valores) o bien en la selección del mejor pronosticador (segmentación de grupos). El primero consiste en determinar la significancia mínima para que dos categorías de una variable queden englobadas en el mismos segmento. El segundo afecta a las selección de variables. Este procedimiento es una forma directa de finalizar la segmentación, porque, después de encontrar el pronosticador con menor significación, si no es inferior al límite establecido, es obvio que no habrá otro pronosticador que cumpla también con esta propiedad, lo que el proceso de división termina.
- Filtros de asociación. Cumplen una función analógica a la de los filtros de significación de pronosticadores. Se trata de determinar la segmentación no porque un determinado cruce no obtenga un mínimo de significación, sino porque el coeficiente de asociación elegido no alcance un determinado nivel.
- Filtros de tamaño. Su principal objetivo consiste en evitar que se formen grupos muy pequeños durante el proceso de segmentación, dado el problema que supone la generalización en estos casos.
- Filtros de nivel. Consiste en arbitrar un nivel máximo de segmentación. Este filtro evita que se formen múltiples segmentaciones en segmentos desproporcionadamente grandes de la muestra. Asimismo, contribuye a simplificar los resultados en la medida en que reduce directamente el número de variables necesarias para predecir la variable dependiente.

En este caso, se implementó un método CHAID haciendo uso del software estadístico PASW (Norusis, 2008), para realizar una exploración de los atributos más significativos del conjunto

patrón $\mathcal{Y}^b$ para el reconocimiento de operones que sean capaces de discriminar de la mejor manera la clase o de la $\overline{o}$ ($\overline{o}$). Para ésto, se utilizó el conjunto total $\mathcal{OP}^{eco}$ y $\overline{\mathcal{OP}^{eco}}$ de *E. coli* definido en la sección 2.2. Se estipuló la variable dependiente a la clase $C = (o, \overline{o})$ y las variables dependientes a los vectores de características Y_l^{eco} con $l = 2, 3, \ldots, 8$. En cuanto a los filtros para terminar el proceso de segmentación, se utilizaron varios de éstos: el mínimo de casos en un nodo parental se estipuló en 35 y en uno filial de 20 y los valores de significativa para la agrupación y división en 0.0001. Estos parámetros fueron definidos de manera general para garantizar que en un nodo parental al menos existiera el 5 % de los datos, con lo cual se estaría descartando de antemano los atributos que contribuyan menos en la identificación de operones.

6.3.2.1. Análisis de características relevantes

En la Figura 6.2 se muestra los resultados de la selección de las características más relevantes para la identificación de operones en *E. coli* del conjunto patrón $\mathcal{Y}^{eco}$ utilizando un DT CHAID, mientras en el Apéndice B se proporcionan las reglas generadas por el mismo. Se puede observar que la profundidad del DT es sólo de dos y de las ocho características sólo tres, conservación de vecindad (Y_1^{eco}), minería de datos (Y_7^{eco}) y distancia intergénica (Y_8^{eco}) tienen significancia estadística. La variable más significativa es la de distancia intergénica (Y_8^{eco}) ya que es la más cercana al nodo raíz y de ahí parten los restantes ramificaciones, una para conservación de vecindad (Y_1^{eco})y la otra para minería de datos (Y_7^{eco}).

6.3.3. Comparación de las características relevantes

Como se describió en la sección 6.3.2.1, las características consideradas como relevantes en *E. coli* utilizando el método basado CHAID DT fueron tres, conservación de vecindad (Y_1^{eco}), minería de datos (Y_7^{eco}) y distancias intergénicas (Y_8^{eco}). De la columna 3 de la Tabla 6.3, se puede observar que las características Y_1^{eco} y Y_7^{eco} son también las más ampliamente representadas en la BD de STRING, sin considerar la co-ocurrencia filogenética (Y_3^{eco}) la cual es la más abundante a pesar de que no tiene ninguna relevancia significativa para discriminar entre las clases $C = \{o, \overline{o}\}$. La variable de mayor significativa estadística es Y_8^{eco} ya que es la más cercana a la raíz, sin embargo si sólo esta variable es utilizada para identificar operones en un DT, se obtendría una precisión del 81.4 %.

Por otra parte, como se puede observar en la Tabla 6.4, estas tres características seleccionadas como las relevantes por el DT, también son las más importantes seleccionadas por la red MLP (Columna 2 de la Tabla 6.3), representando el 73.1 % en la discriminación de clases. Sin embargo, con el método basado en DT se seleccionan menos características del subconjunto relevante obtenido por el análisis realizado con la red MLP, debido a que en éste

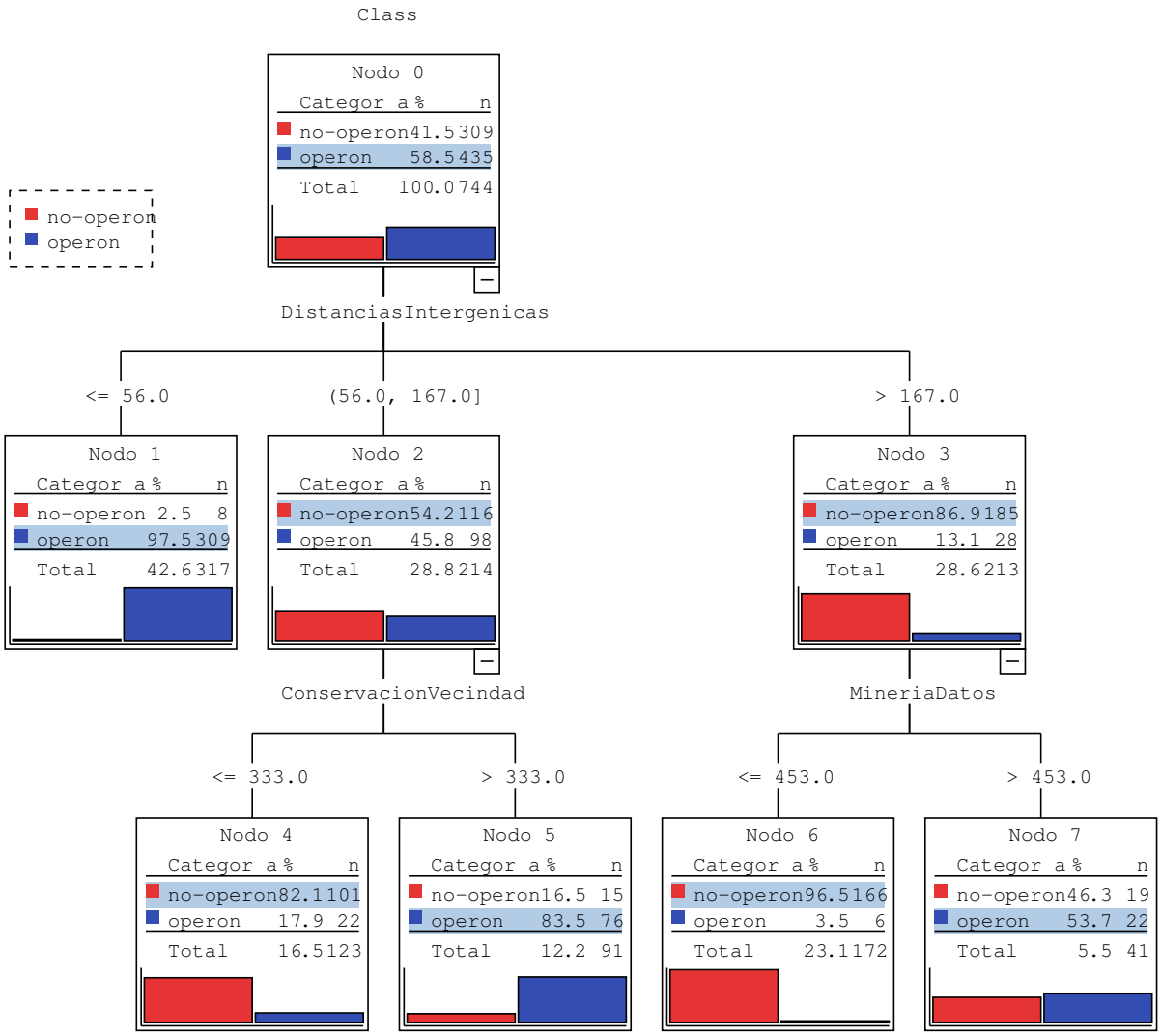

Figura 6.2: Árbol de decisión CHAID para la identificación de operones en *E. coli*

también se considera la co-expresión (Y_4^{eco}) e información de otras base de datos (Y_6^{eco}) como importantes no así en el DT.

6.4. Procedimiento de identificación

El procedimiento para identificar los operones de un genoma determinado, considerando como entrada a la red MLP el arreglo $\mathcal{Y}^{eco} = (Y_1^{eco}, Y_4^{eco}, Y_6^{eco}, Y_7^{eco}, Y_8^{eco})$, es el mismo descrito

Característica	MLP	CHAID DT
Conservación de vecindad,(Y_1^{eco})	si	si
Fusión de genes (Y_2^{eco})	no	no
Co-ocurrencia filogenética (Y_3^{eco})	no	no
Co-expresión (Y_4^{eco})	si	no
Inf. experimental, (Y_5^{eco})	no	no
Inf. de otras BD (Y_6^{eco})	si	no
Minería de datos (Y_7^{eco})	si	si
Distancias intergénicas (Y_8^{eco})	si	si

Tabla 6.4: Características relevantes seleccionadas por cada método

en el capítulo anterior (sección 5.2), cuando la red MLP tenía como características de entrada el valor ponderado de STRING y las distancias intergénicas y consistió en:

Pre-procesamiento. Se realizó el pre-procesamiento de las características de entrada $\mathcal{Y}^b$ mediante la eliminación de los valores atípicos de cada Y_j^b y la normalización de los mismos. Para eliminar los valores atípicos, se hizo uso de la ecuación 4.8, donde se asumió un valor de $\beta = 3$ para remover sólo el 1 % de los datos de cada una de las cinco características. Después cada Y_j^{eco} fue normalizado en rango de $[-1; 1]$ utilizando la ecuación 4.9, siendo $R_{min} = -1$ y $R_{max} = 1$.

Identificación de operones. Para realizar la identificación de operones se utilizó una red MLP con $Backpropagation$ como algoritmo de entrenamiento. En cuanto a la arquitectura, se seleccionó una red 5-7-1 neuronas(s) que fue la que mejor desempeño tuvo, tanto en el conjunto de entrenamiento como en el de prueba, con el menor número de neuronas y capas posible (Ver sección 5.2.2 para detalles del procedimiento). Por lo que $M_j = 5$, $M_k = 7$ and $M_l = 1$ donde las funciones de activación para las neuronas de la capa ocultas son tangente hiperbólicas y para la de salida lineal. En cuanto a la evaluación de la red, se realizó una validación cruzada (Stone, 1974) dividiendo al azar el arreglo patrón $\mathcal{Y}^b$ en 75 % para entrenamiento y 25 % para prueba, probando que la red no estuviera estancada en un mínimo local al satisface el porcentaje de error permitido en el conjunto de datos de entrenamiento y de prueba. Además, para estimar la generalización de la MLP (Bengio and Grandvalet, 2004), se hizo una validación cruzada k-fold con k=7. Finalmente, también se estimó un valor de confianza entre 0 y 1, donde un valor superior a 0.5 indica que el par de genes pertenece a la clase o, mientras uno menor a la clase $\overline{o}$.

Evaluación del desempeño. Los resultados de la red MLP fueron también evaluados utilizando

las medidas de eficiencia descritas en la sección 4.3, donde los datos utilizados en los conjuntos TP, TN, FP y FN son especificados en la sección 4.4. Al utilizar la información de la BD de STRING, los genes g_i^b considerados en $\mathcal{Y}^b$ para el análisis tenían que tener definido un grupo de genes ortólogos.

6.5. Resultados de la identificación

Utilizando los conjuntos de datos que se describen en la sección 4.4, se llevaron a cabo una serie de experimentos: **a)** Utilizando datos de *E. coli*, se comparó el desempeño de una red MLP que tenía como entrada el conjunto total de características (distancias intergénicas y las siete de STRING) con otra red que tenía como entrada el subconjunto de las características relevantes. Además, para analizar la ventaja de utilizar un método de selección de características, estos resultados fueron comparados con los del método descrito en el capítulo anterior. **b)** Se comprobó la generalización del método que utiliza el subconjunto de características, probándolo en datos de organismos diferentes a los utilizados para su entrenamiento. **c)** Finalmente, se realizó una comparación del método basado redes MLP con el CHAID DT, en función de la precisión alcanzada y generalización de los mismos.

6.5.1. Evaluación con características relevantes determinado por redes MLP

6.5.1.1. Desempeño con datos de *E. coli*

Hasta donde se sabe, la precisión más alta reportada para identificar operones en *E. coli* es de 94.8 % y fue obtenida por el método descrito en el capítulo anterior (y reportado en Taboada *et al*. (2010)). Este método utilizó una red MLP y como datos de entrada sólo dos características de pares de genes contiguos: **i)** sus distancias intergénicas y **ii)** la relación funcional, de los correspondientes productos de sus genes, definida por los valores ponderados de BD de STRING. Como se ha indicado anteriormente, estos valores de STRING son computados ponderando siete diferentes características genómicas ($Y_1^b, Y_2^b, \ldots, Y_7^b$). Cuando todas estas características son utilizadas como entradas en una red MLP 8-9-1 con datos de *E. coli* para identificar operones en el mismo organismo, se obtiene una precisión de 95.3 % (Tabla 6.5), lo cual está por arriba de la precisión obtenida con el anterior método (Tabla 5.4).

Con el fin de evaluar el beneficio de un proceso de selección de características, se repitió el mismo análisis de identificación operones en *E. coli* usando sólo el subconjunto de características consideradas como relevantes (sección 6.3.1.1) mediante una red MLP 5-7-1

Variables de entrada			Actual		
			E. coli	***B. subtilis***	***E. coli + B. subtilis***
Predicho	$Y_1^{eco}, Y_2^{eco}, \ldots, Y_8^{eco}$	***E. coli***	95,3 %	92,1	95.8 %
		B. subtilis	94.0 %	94.3 %	94.6 %
		50 organismos	95.9 %	95.9 %	96.6 %
		H. pylori	94.6 %	95.0 %	95.5 %
		M. pneumoniae	95.5 %	94.4 %	95.5 %
		S. solfataricus	98.2 %	97.8 %	98.2 %
		L. monocytogenes	92.5 %	92.8 %	93.1 %

Tabla 6.5: Matriz de precisión en términos del conjunto total de características

diseñada para este propósito. La Tabla 6.6 muestra que la precisión obtenida es de 96.5 %, aumentando un 2 % el desempeño del método que incluía las distancias intergénicas y el valor ponderado de las siete características de STRING (Tabla 5.4). Aunque este porcentaje puede ser visto como marginal, corresponde a casi el 40 % de las mejoras necesarias para alcanzar el 100 % a partir de la precisión del método anterior de 94.6 %. Además, al lograr esta mejora con un subconjunto de características permite tener una mejor compresión del modelo, disminuir los tiempos de procesamiento de los datos, utilizar menor requerimientos donde almacenar la información y lo más importante en este caso, tener un menor costo en la obtención de los datos.

Variables de entrada		Actual		
		E. coli	***B. subtilis***	***E. coli + B. subtilis***
$Y_1^{eco}, Y_4^{eco}, Y_6^{eco}, Y_7^{eco}, Y_8^{eco}$	***E. coli***	96.5 %	93.9 %	96.8 %
	B. subtilis	94.2 %	94.9 %	95.3 %
	50 organismos	96.2 %	96.3 %	97.2 %
	H. pylori	93.2 %	93.8 %	94.0 %
	M. pneumoniae	96.5 %	95.3 %	96.6 %
	S. solfataricus	97.9 %	97.8 %	98.2 %
	L. monocytogenes	92.7 %	92.7 %	93.1 %

Tabla 6.6: Matriz de precisión en términos del subconjunto de características relevantes

La contribución relativa final del subconjunto de características seleccionadas como

relevantes en el proceso de identificación de operones se muestran en la Tabla 6.7. Comparando esta Tabla con la columna 2 de la Tabla 6.3, se puede concluir que el orden de las características comunes, basado en su contribución relativa, es el mismo, siendo la más significativa conservación de vecindad (Y_1^{eco}), seguida por distancias intergénicas (Y_8^{eco}), minería de datos (Y_7^{eco}), co-expresión (Y_4^{eco}) e información de otras BD (Y_6^{eco}).

Característica	**Contribución relativa**
Distancias intergénicas, Y_1^{eco}	30.3 %
Co-expresión Y_4^{eco},	10.5 %
Información de otras BD, Y_6^{eco}	11.1 %
Minería de datos, Y_7^{eco}	21.8 %
Distancias intergénicas, Y_8^{eco}	26.3 %

Tabla 6.7: Contribución relativa en *E. coli* del subconjunto de características seleccionadas como relevantes

*Utilizando una MLP 5-7-1 entrenada con sólo datos de *E. coli*

Como se puede observar en la matriz de confusión de la Tabla 6.8, sólo un número pequeños de casos en la identificación de operones no fue consistente con la información compilada en la base de datos de RegulonDB (Gama-Castro *et al*., 2008). El número de pares de genes operones identificados como no-operones fue de sólo 16 de 434, dando una sensibilidad de 96.5 % (Eq. 4.10). Por otra parte, el número de pares de genes no-operones reconocidos como operones fue de sólo 11 de 309, resultando en una especificidad de 96.4 % (Eq. 4.11). En la Tabla 6.9 se muestran esos errores, donde al igual que en el método descrito en el capítulo anterior, se pueden observar dos tipos de inconsistencias: a) Pares de genes cuyo valor de confianza de la predicción se encontró cerca del umbral de 0.5, que es utilizado para discriminar la clase o y $\overline{o}$. b) Pares de genes cuyo valor de confianza estuvieron cercanos a 0 o 1 que corresponden a casos excepcionales en donde existía una alta confianza en la predicción y que por lo tanto, la inconsistencia podría ser resultado de una mala anotación en el genoma o interpretación errónea de los datos experimentales.

Prácticamente todos los pares de genes identificados inconsistentemente con lo reportado en RegulonDB (Gama-Castro *et al*., 2008), que se muestran en la Tabla 6.9, concuerdan con los errores del método que utiliza la distancia intergénica y el valor ponderado de STRING como variables de entradas (Capítulo anterior, Tablas 5.2 y 5.3). En el caso de pares que pertenecen a la clase *operón*, 13 de los 16 casos mal clasificados concuerdan, mientras en el de la clase *no-operón* todos son consistentes. Varios de estos casos que ahora se identificaron correctamente, anteriormente estaban en la zona frontera de las clases. Por ejemplo, en el caso

		Predicho	
		No-operón	**Operón**
Actual	**No-operón**	298	11
	Operón	16	418

Tabla 6.8: Matriz de confusión de *E. coli* en términos de distancias intergénicas y características relevantes de STRING

[**]Utilizando una MLP 5-7-1 entrenada con sólo datos de *E. coli*

de pares de genes *operón* identificados en la clase *no-operón*, los pares de genes *mnmG-mioC, relA-mazE, ydgK-rsxA,IlsA-fkpB, hypE-fhlA*, y *yjcH-acs* ahora son reconocidos correctamente (Ver Tabla 5.2 y Tabla 6.9). Mientras para lo de la clase *no-operón*, los casos son *bglH-bglB, tufB-secE, ldcC-yaeR, yfhH-yfhL, yaeI-apD, efp-ecnA, ygbE-cysC, ubiB-tatA* y *aat-cydC* (Ver Tabla 5.3 y Tabla 6.9). Por otra parte, los pares de genes *lsrG-tam, purF-ubiX, dut-slmA*, de la clase *operón*, que anteriormente estaban bien clasificados ahora arrojan resultados erróneos, con un valor de confianza en la zona frontera entre ambas clases (ver Tabla 6.9).

Por último, el promedio de los errores de la validación cruzada k-fold fue de sólo 3.4 % con una desviación estándar de 0.4 %, con $k = 7$. Estos resultados confirman que la red MLP no se estancó en un mínimo local y que no hubo un sobre-entrenamiento.

6.5.1.2. Desempeño generalizado

Como se mencionó en la sección 2.2.2.1, un problema común en los métodos de identificación de operones es que no tienden a generalizar bien de un organismo a otro. Se han reportado reducciones significativa del 11 % al 18 % en su precisión cuando son utilizados en otros organismos diferentes a los utilizados para su entrenamiento (Brouwer *et al*., 2008). Sin embargo, con el método descrito en el capítulo anterior, se comprobó que si un método de identificación operón utilizaba datos generales obtenidos a partir de características comunes observadas en el conjunto total de genomas completamente secuenciados, se garantiza la eficacia predictiva extensa (Taboada *et al*., 2010).

Con el fin de probar el desempeño generalizado de este nuevo método basado en el subconjunto de características relevantes, se realizó el mismo proceso de prueba que en el método descrito en el capítulo anterior (sección 5.3.2), es decir se analizaron datos de diversos organismos bacterianos. Para ésto, la red MLP fue primero entrenada exclusivamente con datos de *E. coli* y probada con datos de *B. subtilis*, *H. pylori*, *M. pneumoniae*, *S. solfataricus*, *L.*

Gene1	Gene2	Clase	Clase predicha	Valor de confianza
gabT	gabP	o	$\overline{o}$	0.00
dksA	afsA	o	$\overline{o}$	0.01
truA	dedA	o	$\overline{o}$	0.01
fic	papA	o	$\overline{o}$	0.02
gadX	gadA	o	$\overline{o}$	0.02
gadE	mdtE	o	$\overline{o}$	0.05
dxs	yajO	o	$\overline{o}$	0.09
mltC	nupG	o	$\overline{o}$	0.11
yaiA	aroM	o	$\overline{o}$	0.17
rpmB	yicR	o	$\overline{o}$	0.19
glpK	glpX	o	$\overline{o}$	0.21
hydN	hypF	o	$\overline{o}$	0.28
yiaL	yiaM	o	$\overline{o}$	0.32
lsrG	tam	o	$\overline{o}$	0.46
purF	ubiX	o	$\overline{o}$	0.48
dut	slmA	o	$\overline{o}$	0.49
yaaY	ribF	$\overline{o}$	o	0.60
pcnB	yadB	$\overline{o}$	o	0.65
uhpA	ilvN	$\overline{o}$	o	0.71
lacZ	lacI	$\overline{o}$	o	0.80
hscB	iscA	$\overline{o}$	o	0.850
folC	accD	$\overline{o}$	o	0.90
yrdD	smg	$\overline{o}$	o	95.00
folP	hflB	$\overline{o}$	o	1.00
yadM	htrE	$\overline{o}$	o	1.00
kdsD	kdsC	$\overline{o}$	o	1.00
queA	tgt	$\overline{o}$	o	1.00

Tabla 6.9: Pares de genes operones y no-operones en *E. coli* identificados inconsistentemente con lo reportado en RegulonDB en términos de distancias intergénicas y las características relevantes de STRING

*El valor de confianza está normalizado entre 0 y 1. Un valor menor de 0.5 indica que el par de genes corresponde a la clase $\overline{o}$ en caso contrario a la clase o

monocytogenes y en la recolección parcial de operones de otros 50 organismos bacterianos descritos en la sección 4.4. Los resultados se pueden observar en la Columna 1 de la Tabla 6.6. En el caso de *B. subtilis*, se obtuvo una precisión ligeramente menor de 94.0 %, lo cual sólo representa una reducción de 1.3 %. El peor de los casos se dio en *L. monocytogenes* con una reducción del 2.8 % y el mejor en *S. solfataricus* con un aumento de 2.9 %. Las variación en la precisión representa sólo un pequeño porcentaje en comparación de lo previamente reportado.

A fin de demostrar la robustez del subconjunto de características seleccionado, se entrenó con datos de *B. subitlis* probando con si mismo y con el resto de los organismos (Columna 2 de la Tabla 6.6). En el caso del mismo *B. subitlis*, se obtuvo una precisión de

		Actual	
		E. coli	***B. subtilis***
Predicho	***E. coli***	MLP=96.5 % DT=90.6 %	MLP=95.1 % DT=89.4 %
	B. subtilis	MLP=94.8 % DT=85.5 %	MLP=95.5 % DT=97.6 %

Tabla 6.10: Matriz de precisión, utilizando el subconjunto de características, de los métodos MLP y DT

94.9 %, mientras en *E. coli* del 93.9 %. Al igual que cuando se entrenó sólo con datos de *E. coli*, el peor de los casos se dio en *L. monocytogenes* y el mejor en *S. solfataricus*. Finalmente, se utilizaron de manera conjunta tanto los datos de *E. coli* como de *B. subtilis* con la finalidad de aumentar el conjunto de datos en el entrenamiento de organismos de grupos filogenéticamente distantes. En este caso, como se observa en la Columna 3 de la Tabla 6.6, el desempeño mejora ligeramente en todos los casos, con respecto a cuando la red MLP es entrenada con datos de un sólo organismo.

Todos estos resultados muestran una pequeña mejora en el desempeño en comparación con el método descrito en el capítulo anterior que consideraba a las distancias intergénicas y el valor ponderado de STRING como únicos datos de entrada.

6.5.2. Comparación del desempeño del método MLP y DT

Para comparar el desempeño de la red MLP con otro método anteriormente utilizado para identificar operones, se realizó un análisis usando árboles de decision (DT) CHAID obteniendo los siguientes resultados:

i) Cuando las distancias intergénicas y el valor ponderado de STRING de pares de genes de *E. coli* son utilizados como características de entrada, la precisión obtenida por el método basado en DT CHAID fue de 90.5 %, lo cual es un 4 % menor que lo obtenido con el método basado en redes MLP usando las mismas variables de entrada (Tabla 5.4).

ii) Cuando se utiliza en el análisis sólo el subconjunto de características relevantes Y_1^{eco},Y_7^{eco} y Y_8^{eco}), la precisión obtenida por el DT fue de 91.1 %, lo cual es 5 % menor a la precisión obtenida por la red MLP (Tabla 6.6) que utiliza a Y_1^{eco},Y_4^{eco},Y_7^{eco} y Y_8^{eco} como variables de entrada.

iii) Cuando el desempeño generalizable fue evaluado usando datos de *E. coli* para el entrenamiento y de *B. subtilis* para prueba, la precisión obtenida por el DT fue de 85.5 %, lo cual es 8.7 % menor a lo obtenido con el método basado en redes MLP (Tabla 6.10).

Estos resultados muestran claramente que el método basado en redes MLP para identificar operones es mejor que los basados en DT, además de que garantiza mejor la efectividad predictiva extensa.

CAPÍTULO 7

NUEVA CARACTERÍSTICA SEMEJANTE A STRING Y SU ESTIMACIÓN

Las características de entrada que se utilizan en los métodos de identificación de operones descritos en los capítulos 5 y 6 se basan en la relación funcional entre proteínas ortólogas definidas en la BD de STRING. Como se mencionó en la sección 5.1.2, existen pares de genes (g_i^b, g_{i+1}^b) que no tienen un valor definido en dicha BD porque al menos uno de los genes no tiene determinado un grupo de genes ortólogos $\mathcal{COG}_t$. En promedio, 25 % de los genes de genomas bacterianos no tienen definido un grupo porque o bien no se traducen en proteínas, o su producto polipéptido no tiene una familia ortóloga correspondiente. Por ejemplo, en la bacteria de *E. coli* de sus 4493 genes, sólo 3612 tienen definido uno. Esta falta de representatividad de grupos de ortólogos de la BD COG, es debido a que en el momento de su generación, solamente existían 66 genomas secuenciados en su totalidad (Tatusov *et al*., 2003). Actualmente, se cuenta con más de 1500 genomas, por lo que es posible la definición de nuevos grupos de genes ortólogos.

Por consiguiente, a pesar de que ya se definieron las características relevantes en la identificación de operones (capítulo 6), fue necesario diseñar una metodología complementaria para el reconocimiento de pares de genes donde al menos uno de los dos no tenía definido un grupo de genes ortólogos y por lo tanto, no habían podido ser analizados con los métodos descritos en los capítulos previos al no tener definido valores en la BD de STRING. Para ésto, primero se generó un nuevo conjunto de grupos genes ortólogos con los genes que no tenían definido uno en $\mathcal{COG}$ (Ec. 5.2), es decir en la BD COG de NCBI (Tatusov *et al*., 2003). A partir del nuevo conjunto, se propuso un procedimiento para estimar una característica similar al valor ponderado de STRING ($\mathcal{S}$) que se denominó STRING-like ($\hat{\mathcal{S}}$) basada en la conservación de vecindad, la cual es la característica de más relevancia en la identificación de operones (Tabla 6.7), haciendo posible la identificación del conjunto total de operones $\mathcal{O}^b$ (Ec. 4.4) de un

genoma G^b determinado. La metodología se describe a continuación, así como los resultados de la identificación de operones cuando STRING-like y las distancias intergénicas son usadas como características de entrada.

7.1. Definición de nuevos grupos de genes ortólogos ROGs

El nuevo conjunto de grupos de genes ortólogos se denominó ROGs (Remianing Orthologuous Groups), los cuales son complementarios a los definidos en la BD COG (Tatusov *et al*., 2003). Para determinar de si un par de genes son ortólogos se usa la propiedad de ser Biderectional Best Hit (BBH), descrita a continuación.

Definición 3 *Dos genes $g_i^b \in \hat{G}^b$ y $g_j^c \in \hat{G}^c$ son Bidirectional Best Hit (BBH), si para el gen ${g_i}^b \in G^b$ su máxima similitud, a nivel secuencia, es con el gen ${g_j}^c \in G^c$ y para este gen, su máxima similitud también es el gen ${g_i}^b$, donde dicha similitud es identificable a cierto nivel κ de identidad. La anterior definición ha sido ampliamente utilizada en estudios de genómica comparativa para identificar genes que potencialmente sean ortólogos (Overbeek* et al*., 1999).*

Para determinar la máxima similitud, todos los genes del genoma G^b son comparados contra todos los genes del genoma G^c y viceversa, es decir $G^b * G^c$ y $G^c * G^b$. No existe en el genoma G^b otro gen más similar que g_i^b para el gen $g_j^c \in G^c$ y viceversa. La similitud se basa en un valor obtenido al comparar sus secuencias de aminoácidos ($x_{8,i}^b, x_{8,j}^c$) o nucleótidos ($x_{7,i}^b, x_{7,j}^c$) mediante el algoritmo de BLAST (Altschul.S.F. et al*., 1990), donde dicho valor de similitud tiene que ser mayor a un κ dado.*

El proceso propuesto para la generación de los nuevos grupos $\mathcal{ROG}$ es el siguiente:

Paso 1. Seleccionar un subconjunto $\hat{\mathcal{G}} \subseteq \mathcal{G}$ de genomas no-redundantes, es decir filogenéticamente distantes, para que no existiera una sobre estimación de información de genes de genomas filogenéticamente cercanos que pudieran generar grupos de genes ortólogos falsos positivos (e.g. en la actualidad existen 12 sepas de la bacteria de *E. coli*). La cardinalidad del nuevo subconjunto $\hat{\mathcal{G}}$ dependió del valor de la distancia evolutiva entre los genomas de $\mathcal{G}$. Para ésto, los genomas considerados tenían que estar separados por una distancia ($d(i,j) = \left|G^b, G^c\right|$) mayor a un umbral, obteniendo dichas distancias entre todas las combinaciones de pares de genomas dadas por el número C_2^n, es decir:

$$\hat{\mathcal{G}} = \left\{G^b, G^c \in \mathcal{G} \mid d(G^b, G^c) > \varepsilon\right\} \tag{7.1}$$

donde ε =0.5 para garantizar que en el conjunto de organismos seleccionados no existieran cepas de un mismo organismo. En este caso, la distancia d fue evaluada

mediante el uso del programa PROTDIST del paquete de inferencia filogenética (Felsenstein, 1989) de modo que $\hat{\mathcal{G}} = \{\hat{G}^1, \ldots, \hat{G}^\mu\}$, siendo $\mu = 300$. La lista de estos organismos está disponible en el Apéndice C.

Paso 2. Obtener, a partir del conjunto $\hat{\mathcal{G}}$ (Eq.7.1), el conjunto $\mathcal{NC}$ de todos los genes que no tenían determinado un elemento de $\mathcal{COG}$ (Ec.5.2), es decir $x^b_{10,i} \neq \mathcal{COG}_t$, tal que:

$$\mathcal{NC} = \left\{ g_i^b \mid x^b_{10,i} \neq \mathcal{COG}_t, \ \forall \ {g_i}^b \in \hat{\mathcal{G}}^b, \text{ con } b = 1, \ldots, \mu \right\} \tag{7.2}$$

donde $\mu = 300$, obteniendo $\left|\mathcal{NC}\right| = 101,176$.

Paso 3. Separar el conjunto $\mathcal{NC}$ (Eq. 7.2), en el conjunto $\mathcal{CDS}$ formado por los genes de tipo mRNA ($x^b_{6,i} = mRNA$) y su complemento, denotado $\overline{\mathcal{CDS}}$ para cualquier otro tipo de genes, de modo que para cada $g_i^b \in \mathcal{NC}$:

$$\mathcal{CDS} = \left\{ {g_i}^b \mid x^b_{6,i} = mRNA, \forall \ {g_i}^k \in \mathcal{NC} \right\} \tag{7.3}$$

$$\overline{\mathcal{CDS}} = \left\{ {g_i}^k \mid x^k_{6,i} \neq mRNA, \forall \ {g_i}^k \in \mathcal{NC} \right\} \tag{7.4}$$

donde $x^b_{6,i}$ denota el tipo de gen, obteniendo $\left|\mathcal{CDS}\right| = 80,941$ y $\left|\mathcal{NCDS}\right| = 20,235$.

Paso 4. Encontrar los genes ortólogos de cada elemento de $\mathcal{CDS}$ (Ec. 7.3) y $\overline{\mathcal{CDS}}$ (Ec. 7.4), utilizando la propiedad de BBH (Definición 3). Para los genes del conjunto $\mathcal{CDS}$, al ser codificantes, la comparación se realizó a nivel de su secuencia de aminoácidos ($\mathcal{AM}$), es decir utilizando la característica $x^b_{8,i}$ y $x^c_{8,j}$, mientras que para los genes del conjunto $\overline{\mathcal{CDS}}$ a nivel nucleótidos ($\mathcal{NU}$) utilizando $x^b_{7,i}$ y $x^c_{7,j}$, formándose los conjuntos:

$$\mathcal{BBHCD} = \left\{ (g_i^b, g_j^c), \Leftrightarrow \left(x^b_{8,i}, x^c_{8,j} \text{ son BBH } \right), \ \forall \ g_i^b \text{ y } g_j^c \in \mathcal{CDS}, i \neq j, \right\} \tag{7.5}$$

$$\overline{\mathcal{BBHCD}} = \left\{ (g_i^b, g_j^c), \Leftrightarrow \left(x^b_{7,i}, x^c_{7,j} \text{ son BBH } \right) \forall \ g_i^b \text{ y } g_j^c \in \overline{\mathcal{CDS}}, i \neq j, \right\} \tag{7.6}$$

obteniendo $\left|\mathcal{BBHCD}\right| = 249,650$ y $\left|\overline{\mathcal{BBHCD}}\right| = 50,328$.

Paso 5. Obtener el conjunto total de pares de genes ortólogos que no tenían determinado un elemento del conjunto $\mathcal{COG}$ y que son BBH:

$$\mathcal{TBBH} = \mathcal{BBHCD} \cup \overline{\mathcal{BBHCD}} \tag{7.7}$$

obteniendo $\left|TBBH\right| = 312,594$.

Paso 6. Generar los nuevos grupos de genes ortólogos $\mathcal{ROG}$ complementarios a los ya definidos por la BD-COG y establecidos en el conjunto $\mathcal{COG}$. Para ésto, se presentó el problema en términos de teoría de grafos (Gross and Yellen, 1998; West, 2001). Cada gen g_i^b de $\mathcal{NC}$ (Ec. 7.2) correspondió a un vértice, y los pares de genes ortólogos de estos genes definidos en el conjunto $\mathcal{TBBH}$ (Ec. 7.7) son las aristas. De este modo, se obtuvo un grafo no dirigido $\mathcal{GRF} = (\mathcal{NC}, \mathcal{TBBH})$ con $|\mathcal{NC}| = 101,176$ nodos y $312,594$ vértices. Después, se construyó una matriz de adyacencia simétrica $A_G = |\mathcal{NC}| * |\mathcal{NC}|$, cuyo elemento $ag_{ij} = 1$ si existió la arista $tbbh_{ij}$ y cero en otro caso. Por medio del algoritmo de agrupamiento propuesto en (Hartuv and Shamir, 2000), el cual esta basado en la máxima conectividad de un a_{ij}, se crearon los nuevos grupos de genes ortólogos ($\mathcal{ROG}_t$), tal que:

$$\mathcal{ROG} = \{\mathcal{ROG}_1, \ldots, \mathcal{ROG}_s\} \tag{7.8}$$

obteniendo $s = 8901$, de los cuales 8539 correspondían a grupos de genes codificantes y 362 grupos de genes no-codificantes, es decir los genes que eran parte de los conjuntos $\mathcal{CDS}$ (Ec. 7.3) y $\overline{\mathcal{CDS}}$ (Ec. 7.4), respectivamente.

Paso 7. Asignar el correspondiente $\mathcal{ROG}_s$ a todos los genes que no tenían asignado un grupo $\mathcal{COG}_t$, es decir a los del conjunto $\mathcal{NC}$ (Eq. 7.2).

Paso 8. Definir, para los genes que no tenían determinado un elemento de $\mathcal{COG}$ y no se les pudo definir uno de $\mathcal{ROG}$, la etiqueta de NOG. Esta etiqueta indica que no tienen un grupo de genes ortólogos definido.

7.2. Cálculo de STRING-like basado en la conservación de vecindad

Una vez que se establecieron los nuevos grupos de ortólogos, fue necesario definirles un valor de relación funcional equivalente al valor ponderado de STRING, que se denominó STRING-like. Por lo cual, se propuso utilizar el valor de conservación de vecindad de genes contiguos a través de diferentes genomas bacterianos, siendo ésta la característica que más contribuyó en la identificación de operones (Tabla 6.2).

Como se mencionó en la sección 2.2.1, la conservación de vecindad ha sido ampliamente utilizada en varios trabajos previos para identificar operones. Se han propuesto diferentes modificaciones para su estimación, por ejemplo, en Tran *et al*. (2007), Jacob *et al*. (2005) y Janga and Moreno-Hagalsieb (2004) se consideró la conservación de vecindad sólo en genes contiguos separados por una distancia intergénica límite específica, en Price *et al*. (2005) en ventanas de cierto tamaño sin considerar el número de genes dentro de éstas , o en Zhang *et al*. (2006), Westover *et al*. (2005) y Edwards *et al*. (2005) en ventanas que incluyen un cierto número de genes independientemente de las distancias intergénica de éstos. Debido a que se han reportado diversos reordenamientos, supresiones e inserciones de genes en operones cuyo tamaño es de tres a cinco genes (Wolf *et al*., 2001), en este trabajo, se definió un nuevo concepto de genes *contiguos flexibles* (Definición 4) que permitió establecer la mayoría de las relaciones de vecindad de grupos de genes ortólogos de acuerdo a su comportamiento biológico, sin tener las limitantes de trabajos arriba mencionados.

Definición 4 *Dos genes*

$$\left(g_i^b, g_j^b\right) \textit{ son } \begin{cases} contiguos\ flexibles\ (cf) & \textit{si}\quad x_{3,i}^b = x_{3,j}^b \ \ y \ \ x_{5,j}^b - x_{4,i}^b < \kappa_{ij}^b \\ & \textit{con } j = (i+1), \ldots, (i+1+d_{ij}^b) \\ no - contiguos\ flexibles\ (\overline{cf}) & \textit{en cualquier otro caso} \end{cases} \tag{7.9}$$

donde:

- $x_{3,i}^b, x_{3,j}^b$ *establecen la dirección de transcripción de los genes* (g_i^b, g_j^b).
- $x_{5,i}^b, x_{4,i}^b$ *establecen la posición derecha e izquierda en nucleótidos del gen* g_i^b*, respectivamente dentro del genoma* G^b.
- d_{ij}^b *es el número de genes intermedios en la misma dirección de transcripción que los genes* (g_i^b, g_j^b)*. De acuerdo al trabajo de Wolf* et al. *(2001), se tiene* $0 \leq d_{ij}^b < 4$.
- κ_{ij}^b *es la distancia intergénica máxima permitida entre* g_i^b *y* g_j^b*, definida en este trabajo como:*

  $$\kappa_{ij}^b = tm_{ij}^b + (d_{ij}^b + 1)\rho \tag{7.10}$$

  donde ρ *es la distancia intergénica promedio de todos los pares de genes operones experimentalmente comprobados en E. coli y B. subtilis (sección 4.4) adicionandóle dos desviaciones estándar, obteniendo* $\rho = 375$ *y* ${tm_{ij}}^k$ *es el tamaño en nucleótidos de los*

genes intermedio, dado por:

$$tm_{ij}^{b} = \begin{cases} \left(x_{5,i}^{b} - x_{4,i}^{b}\right) & \textit{si } d_{ij}^{b} = 0 \\ \sum_{j_{min}}^{j_{max}} \left(x_{5,j}^{b} - x_{4,j}^{k}\right) & \textit{en cualquier otro caso} \end{cases} \tag{7.11}$$

donde $j_{min} = i + 1$ *y* $j_{max} = i + 1 + d_{ij}^{b}$.

Con la finalidad de implementar el valor de STRING-like: **i)** Primero, se calculó la conservación de vecindad entre pares de grupos de genes ortólogos de $\mathcal{COG}$, debido a que tenían con un valor de relación funcional de STRING. Por lo tanto, los valores de vecindad pudieron ser aproximados a los de STRING, obteniendo así el STRING-like para este conjunto de datos. Se definieron los coeficientes que permitieron que las funciones generadas por los valores de vecindad y STRING fueran lo más próximas posibles. **ii)** Segundo, se calculó la conservación de vecindad pero ahora sobre el conjunto de genes que no tenía definido un $\mathcal{COG}$. Para obtener el STRING-like de este conjunto de datos, al no tener un valor de STRING, los valores de vecindad fueron aproximados utilizando los mismo coeficientes definidos en el paso anterior. Para ésto, se implantó el siguiente proceso:

Paso 1. Definir el conjunto $\mathcal{GN}$ de todos los pares de genes *contiguos flexibles* que tenían determinado un elemento de COG, denotado como:

$$\mathcal{GN} = \left\{(x_{10,i}^{b}, x_{10,j}^{b}, d_{ij}^{b}, m^{b}),\ \text{con } i = 1, \ldots, m^{b} \text{ y } b = 1, \ldots, \mu; \right\} \tag{7.12}$$

donde:

- $\left(x_{10,i}^{b}, x_{10,j}^{k}\right) = cf$
- $x_{10,i}^{b} = \mathcal{COG}_{s}$
- $x_{10,j}^{k} = \mathcal{COG}_{t}$
- $m^{b} = \left|G^{b}\right|$

Paso 2 Definir un conjunto $\mathcal{N}$ que evalúa la relación de vecindad de los pares de $\mathcal{GN}$ (Ec. 7.12), tal que:

$$\mathcal{N} = \left\{\{\mathcal{COG}_{1}, \mathcal{COG}_{2}, n_{12}\}, \ldots, \{\mathcal{COG}_{s}, \mathcal{COG}_{t}, n_{st}\}\right\} \tag{7.13}$$

donde n_{st} es el valor de conservación de vecindad de un par de grupos ortólogos, definida como:

$$n_{st} = - \sum_{\forall\ x^b_{10,i},\, x^b_{10,j} \in \mathcal{GN}} L_{st}\big(x^b_{10,i}, x^b_{10,j}, d^k_{ij}, m^k\big) \tag{7.14}$$

donde $x^b_{10,i} = \mathcal{COG}_s$, $x^b_{10,j} = \mathcal{COG}_t$ y L_{st} es el logaritmo de la verosimilitud de que el par de grupos de genes ortólogos *contiguos flexible* $\big(x^b_{10,i}, x^b_{10,j}\big) \in \mathcal{GN}$ estén relacionados con una distancia ${d_{ij}}^b$ en un genoma de tamaño m^b y su valor se calculó como la probabilidad de ser vecinos, definido como:

$$L_{st} = \log\Big(p_s\, p_t\, p_{st}\, \frac{d^b_{st}(2m^b - d^b_{st} - 1)}{\big(m^b(m^b - 1)\big)}\Big) \tag{7.15}$$

donde:

- p_s y p_t son las frecuencias relativas de los grupos de ortólogos $\mathcal{COG}_s$ y $\mathcal{COG}_t$ en $\mathcal{GN}$ considerando el número de genes que conforman a cada grupo.
- p_{st} es la frecuencia de $\mathcal{COG}_s$ y $\mathcal{COG}_t$ en $\mathcal{GN}$ dividido entre el número de genomas diferentes G^b que tienen a estos dos grupos como vecinos y mutiplicada por el número total de genomas no redundantes, en este caso 300.

Paso 3 Generar los valores de STRING-like para pares de genes que tenían determinado un elemento de COG, con la finalidad de primero aproximar el valor de conservación de vecindad a los valores de STRING. Para ésto, se compararon los valores obtenidos de $\mathcal{N}$ (Ec. 7.13) con los valores de STRING $\mathcal{S}$ (Ec. 5.4).

La Figura 7.1-A muestra el conjunto de los datos de $\mathcal{N}$ donde se puede observar que sus valores pueden ser definidos en dos intervalos $[60, 200)$ y $[200, 999]$. Por otra parte, los valores de $\mathcal{S}$ se muestran en la Figura 7.1-B donde también se observa que pueden ser definidos en dos intervalos, [200,900) y [900,999]. Además, el primer intervalo establecido de $\mathcal{S}$ incluye el 86.7 % de la información, parecido también al 88.5 % del primer intervalo de $\mathcal{N}$ (Figura 7.1-C).

Por consiguiente, se definió el conjunto de datos que se denominó STRING-like, tal que:

$$\hat{\mathcal{S}} = \big\{\{\mathcal{COG}_1, \mathcal{COG}_2, \hat{s}_1\}, \ldots, \{\mathcal{COG}_s, \mathcal{COG}_t, \hat{s}_b\}\big\} \tag{7.16}$$

acoplando $\mathcal{N}$ con $\mathcal{S}$ mediante una aproximación lineal continua por partes, utilizando una

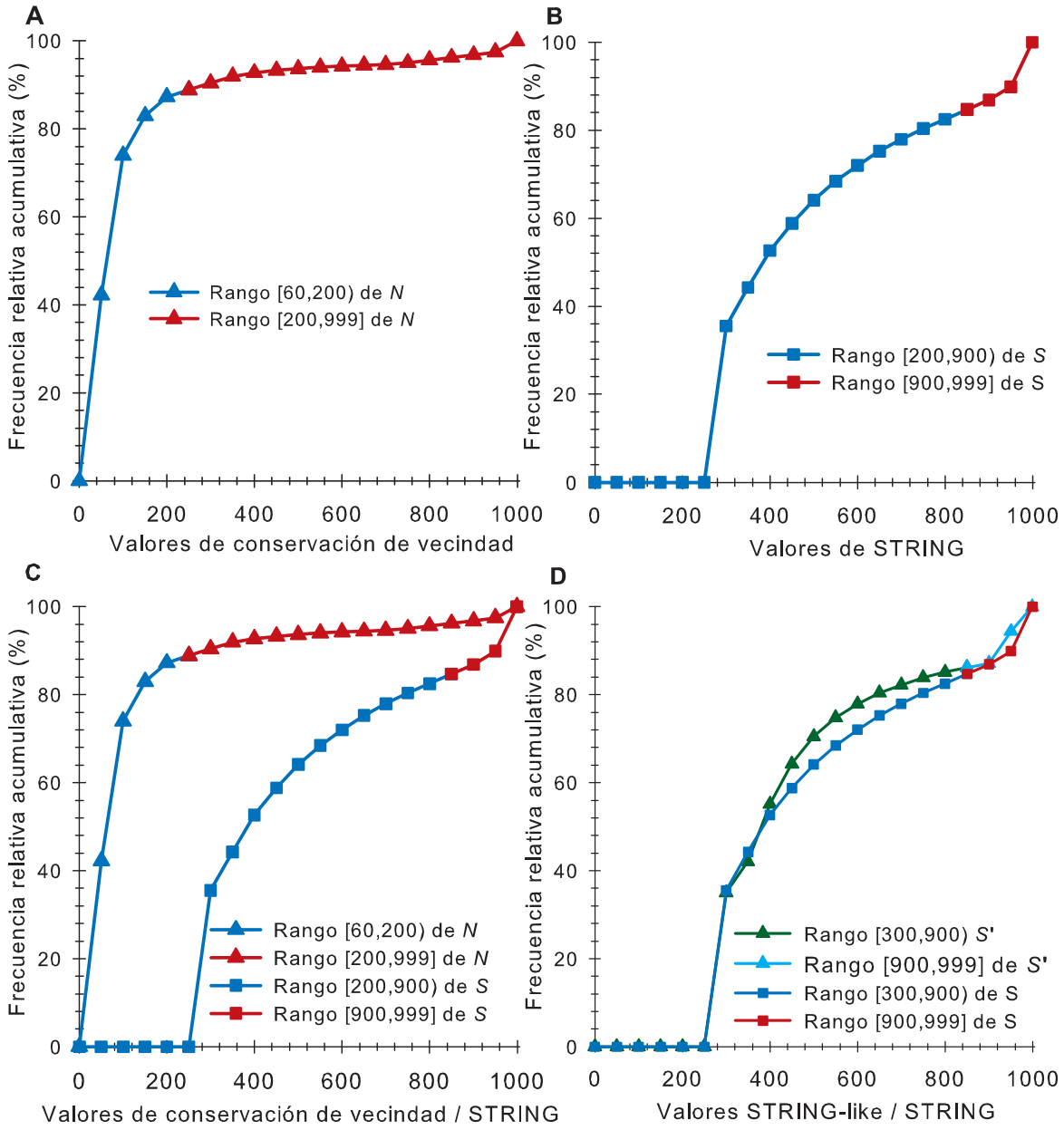

Figura 7.1: (A) Función de conservación de vecindad ($\mathcal{N}$). (B) Función de relación funcional de STRING ($\mathcal{S}$). (C) Función $\mathcal{N}$ que será aproximada a $\mathcal{S}$ por intervalos (azul con azul y rojo con rojo) usando una aproximación lineal continua por partes. (D) Función $\hat{\mathcal{S}}$ derivada de la aproximación de la función $\mathcal{N}$ a $\mathcal{S}$, la cual representa el valor STRING-like

.

ecuación para cada intervalo encontrado en $\mathcal{N}$ y $\mathcal{S}$ (Figura 7.1-d), tal que:

$$\hat{s}_b = \begin{cases} \frac{z_1(T_1-t)+z_2(t-t_1)}{T_1-t_1} & \text{si} \quad t_1 < t < T1 \\ \\ \frac{y_2(t_2-t)+z_3(t-T_1)}{t_2-T_1} & \text{si} \quad T_1 < t < t_2 \end{cases} \tag{7.17}$$

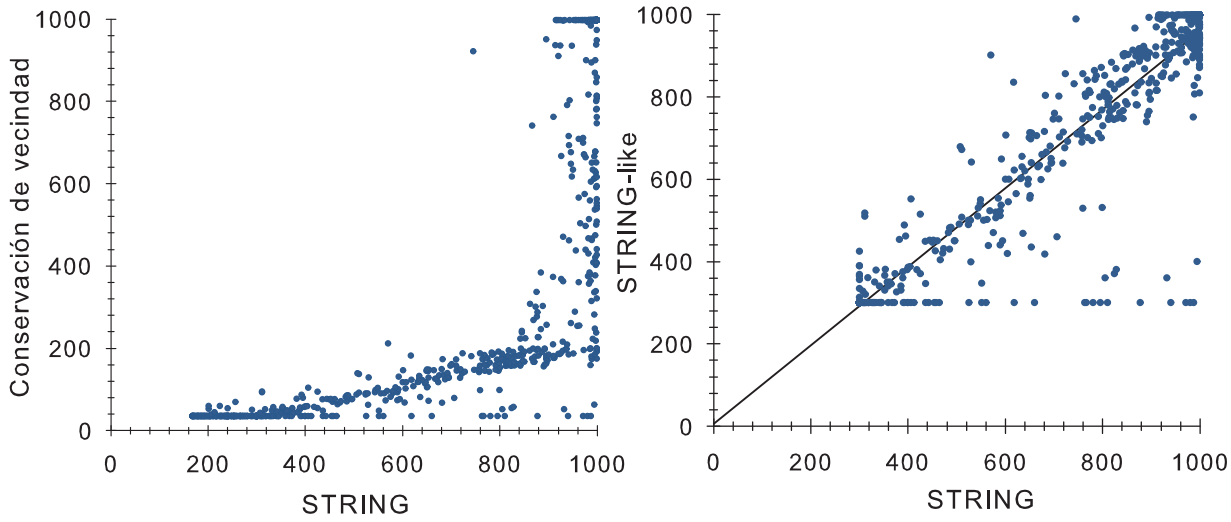

Figura 7.2: (A) Correspondencia de los valores de conservación de vecindad y STRING de pares de genes operones de *E. coli*. (B) Correspondencia de valores de STRING y STRING-like del mismo conjunto de datos.

donde:

- $t_1 = Min(\mathcal{N}),\ t_2 = Max(\mathcal{N})$
- $z_1 = 332,\ z_2 = 752,\ z3 = 950$
- $T_1 = 199$

Se obtuvo una correlación de $\hat{\mathcal{S}}$ y $\mathcal{S}$ del 82 %. La Figura 7.2-a, muestra la relación de conservación de vecindad y STRING de los pares de genes operones de *E. coli*, mientras en la Figura 7.2-b se muestra la relación de STRING-like con STRING donde se puede observar la correlación de dichos datos. Existen algunos casos de pares genes que tiene un valor de STRING-like bajo y uno alto para STRING, esto se debe a que cuentan con evidencia experimental que respalda su relación a pesar de que no conserva su vecindad a través de varios genomas bacterianos.

Cabe hacer notar que las variables utilizadas para calcular L_{st} (Eq. 7.15) permitieron evaluar la probabilidad de dos grupos de genes ortólogos del conjunto $\mathcal{COG}$ de estar relacionados funcionalmente en términos de su conservación de vecindad, considerando:

i) El número de veces que aparecen como vecinos; entre mayor sea la frecuencia, mayor es la conservación y por tanto es más importante su relación.

ii) El número de genes intermedios entre éstos; entre menor número de genes más significativa es su contribución.

iii) El número de elementos en los grupos COG de los genes; entre más elementos tengan los grupos mayor es la probabilidad de que aparezca por casualidad en cualquier genoma y, en consecuencia, su contribución es menos importante.

iv) El tamaño del genoma, en términos de número de genes; entre más grande es el genoma, menos significativa es su contribución.

Se ha observado que L_{st} es muy pequeño, cuando p_s, p_t, p_{st} o d_{st}^k son pequeños, lo cual coincide con lo reportado por Dam *et al*. (2007) al afirmar que valores pequeños de L_{st} generalmente están asociados con pares de genes que están funcionalmente relacionados.

Una vez que se definió la relación $\hat{\mathcal{S}}$ (Ec. 7.16) con elementos de $\mathcal{COG}$ que estaban caracterizados en la BD de STRING, se repitieron los Pasos 1 y 2 del proceso descrito arriba pero ahora para aproximar los valores de $\mathcal{N}$ en función de todos los pares de genes que no estaban caracterizados en dicha BD. Es decir, en pares de genes donde al menos uno no tenía definido un elemento de COG, pares de genes con $\mathcal{COG}$ y $\mathcal{ROG}$, así como los de $\mathcal{ROG}$ con $\mathcal{ROG}$. Para ésto, primero se definió el conjunto $\mathcal{GN}$ (Paso 1) para dichos pares de genes ($\mathcal{COG} \leftrightarrow \mathcal{ROG}$ y $\mathcal{ROG} \leftrightarrow \mathcal{ROG}$). Después, se desarrolló el Paso 2, utilizando este nuevo conjunto de $\mathcal{GN}$ para definir una nueva $\mathcal{N}$. Finalmente, se utilizó los mismos parámetros ya establecidos en la ecuación 7.16 pero en el nuevo cojunto $\mathcal{N}$. Al finalizar, 95.5 % de los genes del conjunto de 300 genomas no redundantes $\hat{\mathcal{G}}$ (Ec. 7.1) tuvieron un valor de relación funcional con otro gen, ya sea proveniente $\mathcal{S}$ o de $\hat{\mathcal{S}}$.

Finalmente, para el 4.5 % de los pares de genes que no pudieron ser caracterizados por STRING o STRING like se les estableció un valor mínimo de relación funcional de 300. Este filtro fue determinado considerando el método computacional seleccionado para la identificación de operones (sección 5.2.2) y el conjunto de datos (sección 4.4). Al realizar el entrenamiento de la red MLP utilizando el conjunto de datos de *E. coli*, se pudo determinar que los pares de genes con valores en $\hat{Y}_2^{eco} \geq 300$ y una distancia intergénica $Y_1^{eco} \leq 50$ bp, se clasificaban en la clase o; generando con este filtro, la menor cantidad de faltos positivos y falsos negativos.

7.3. Resultados de la identificación

Las características utilizadas para discriminar pares de genes como o o $\overline{o}$ fueron dos, al igual que en el método descrito en el capítulo 5.1. La diferencia, es que el valor de Y_2^b(Ec. 5.3) de STRING fue reemplazado por el de STRING-like ($\hat{Y}_2^b$) en la matriz patrón tal que $\mathcal{Y}^b = (Y_1^b, \hat{Y}_2^b)$, donde Y_1^b son las distancias intergénicas. Por otra parte, al igual que en los métodos descritos en los capítulos anteriores, la dirección de transcripción fue usada en una pre-clasificación donde pares de genes que están en direcciones contrarias automáticamente se asociaron a la clase $\overline{o}$ dado que un par de genes que pertenecen a un mismo operón forzosamente deben estar en la misma dirección de transcripción (sección 5.1.3).

En cuanto al método utilizado para identificar operones fue el mismo aplicado en el capítulo 5.2. Es decir, se empleó la misma red MLP 2-3-1 (sección 5.2.2) entrenada con datos de distancias intergénicas y valores de relación funcional de STRING ($\mathcal{Y}^{eco} = (Y_1^{eco}, Y_2^{eco})$) para sólo utilizar a este nuevo conjunto de datos como de prueba. Esto debido a que la cantidad de datos que no tenían un valor de STRING son pocos, por lo que fue imposible entrenar una nueva red y tener un conjunto de prueba con dicha información. Además, a pesar de que el vector de atributos $\hat{Y}_2^{eco}$ esta basado en la conservación de vecindad, éste está aproximado a los valores de STRING por lo que su desempeño no debería disminuir al utilizar una red entrenada con dichos datos.

Para evaluar la viabilidad de usar STRING-like en la identificación de operones, se llevaron a cabo los siguientes experimentos: **a)** Se comparó la precisión obtenida cuando se utilizó STRING y STRING-like en pares de genes de *E. coli* que tenían un grupo de genes ortólogos definido. **b)** Se comprobó la generalización del método que utiliza STRING-like, utilizando datos de *E. coli* en el entrenamiento y datos *B. subtilis* como prueba. **c)** Se evaluó el desempeño en pares de genes de *E. coli* donde al menos uno no tenía un elemento $\mathcal{COG}$ definido.

7.3.1. Desempeño de STRING-like con datos de *E. coli* con COGs

Con la finalidad de comparar el desempeño alcanzado con STRING-like en comparación con STRING, primero se realizó la identificación de operones en *E. coli* en aquellos pares de genes que tenían un elemento $\mathcal{COG}$ (Ec. 5.2) definido, es decir en el mismo conjunto de datos que en la sección 5.3.1. Para ésto, se utilizó la misma red MLP 2-3-1 (sección 5.2.2) entrenada con dichos datos, pero sustituyendo el valor de STRING (Y_2^{eco}) por el de STRING-like ($\hat{Y}_2^{eco}$) en la consulta. De manera relevante, se encontró que el desempeño alcanzado fue ligeramente inferior que cuando se utilizó STRING (94.5 %), ya que se alcanzó una precisión de 93.3 %, una

sensibilidad del 95.7 % y una especificidad del 92.5 %. Estos resultados obtenidos permitieron validar las modificaciones propuestas en este trabajo tanto a la característica de conservación de vecindad, así como su aproximación para obtener valores semejantes a los de STRING.

7.3.2. Desempeño generalizado de STRING-like

Para probar que STRING-like, al igual que STRING, tiene un desempeño generalizado satisfactorio primero se probó con la red MLP entrenada exclusivamente con pares de genes de *E. coli* con un $\mathcal{COG}$ definido (ver sección 5.3.2) y probada en el conjunto de datos *B. subtilis*, es decir, en 169 pares de genes operones y 157 no-operones (ver Tabla 4.2). Como se puede ver en la Tabla 7.1, se obtuvo una precisión ligeramente menor de 91.4 %, lo cual sólo representa una reducción de 1.5 %. Este resultado demuestra la efectividad del procedimiento para extrapolar los valores de STRING considerando únicamente la variable de la conservación de la vecindad entre genes en diferentes genomas y permite suponer que la obtención de los valores STRING-like para pares de genes que no tiene un valor de STRING, va a ser correcta.

		Actual	
		E. coli	***B. subtilis***
Predicho	***E. coli***	92.9 %	90.5 %
	B. subtilis	91.4 %	91.8 %

Tabla 7.1: Matriz de precisión en términos de distancias intergénicas y STRING-like

A fin de verificar con mayor profundidad los resultados antes obtenidos, se utilizó la red MLP entrenada con pares de genes de *B. subitlis* con un $\mathcal{COG}$ definido (sección 5.3.2) probando con el mismo organismo y con *E. coli*. En el caso del mismo *B. subitlis* se obtuvo una precisión de 91.8 %, mientras en *E. coli* del 90.5 %. Estos resultados muestran que los valores STRING-like usados en la red MLP tiene una muy alta capacidad de generalizar.

7.3.3. Desempeño de STRING-like con datos de *E. coli* con ROGs

Una vez que se validó la característica de STRING-like con un conjunto de datos conocido y que se comprobó su generalización, se evaluó el desempeño de ésta en el conjunto de pares de genes $\mathcal{OP}^{eco}$ y $\overline{\mathcal{OP}}^{eco}$ donde al menos uno tenía determinado un elemento del conjunto $\mathcal{ROG}$. Es decir en 58 pares de genes operones y 77 no-operones de *E. coli* (Tabla 4.2), los cuales no habían podido ser analizados con los métodos propuestos en los capítulos anteriores dado que

no tienen un grupo de ortólogos $\mathcal{COG}$ definido y por consecuencia valores de STRING que los relacionen funcionalmente. La precisión alcanzada fue de 92.9 %, la cual es ligeramente inferior a la obtenida en el conjunto de pares de genes con un elemento $\mathcal{COG}$. Esta pequeña disminución del desempeño se debe a este conjunto de datos corresponde a genes que se encuentran poco distribuidos entre los genomas, es decir, tienen baja cobertura ya que pueden representar rutas metabólicas o funciones muy específicas de ciertos grupos de organismos y por tanto, su estadística de conservación de vecindad es estimada con menor precisión.

CAPÍTULO 8

CONCLUSIONES Y DISCUSIONES

8.1. Conclusiones

En esta tesis se presenta por primera vez en la literatura, una descripción formal sobre el proceso de identificación de operones bacterianos en términos de los atributos, funciones y relaciones usados para describir a los genes y que permiten definir las características de la matriz patrón usada para la discriminación de operones. Ésto evita toda ambigüedad y permite sistematizar los algoritmos de identificación o refinarlos mediante el agregado de nuevas operaciones y/o atributos y/o características de manera sencilla.

Posteriormente, se desarrolló un método simple y altamente preciso basado en una red neuronal MLP que identificó exitosamente la estructura de casi todos los operones validados experimentalmente en los organismos modelos de *E. coli*, *B. subtilis*, *H. pylori*, *M. pneumoniae*, *S. solfataricus*, *L. monocytogenes* y otros cincuenta organismos parcialmente estudiados. Este método utilizó como características de entrada las distancias intergénicas entre genes contiguos y los valores ponderados de relaciones funcionales de la base de datos de STRING entre los diferentes grupos de proteínas ortólogas. Una de las ventajas fundamentales del método desarrollado en esta tesis, sobre otros previamente reportados, es el uso de los valores de STRING que integra la información procedente de diferentes tipos de fuentes genómicas, tales como conservación de vecindad, fusión de genes, co-ocurrencia filogenética, co-expresión de genes, información experimental de interacciones proteína-proteína, información de otras BD y minería de texto. Aunque los valores de STRING fueron inicialmente concebidos para la predicción de interacciones entre proteínas, dichos valores de STRING en la predicción de operones son de gran utilidad ya que esta base de datos integra la información de diferentes fuentes y tipos de evidencia y hace que las predicciones realizadas estén menos influenciadas por el sesgo impuesto cuando se entrena con datos de un organismo específico. Esto ha permitido realizar la identificación de operones de cerca de 1100 genomas totalmente secuenciados y disponibles públicamente.

En una segunda etapa del presente proyecto, se mejoró la identificación de operones realizando un proceso previo de selección de características relevantes. Para ésto, utilizando una red MLP, se adaptó el método *Weight Explanatory* (*WE*) para cuantificar la contribución relativa de las características de entrada (distancias intergénicas y las siete variables de STRING) en la identificación de operones, y así posteriormente seleccionar el subconjunto más relevante que fueron la conservación de vecindad, distancias intergénicas, minería de texto, co-expresión de genes e información experimental de otras BD. La característica más informativa fue la de conservación de vecindad, seguido de las distancias intergénicas que ha sido la característica más utilizada en los métodos de identificación de operones. Minería de datos fue la tercera característica más relevante y hasta donde se sabe, es la primera vez que se ha utilizado en un estudio de reconocimiento de operones. Un caso particular es la co-ocurrencia filogenética que ha sido ampliamente utilizada por varios métodos sin embargo su contribución relativa no fue significativa. La información experimental de otras BD tuvo una contribución importante, siendo hasta el momento la más confiable de todas las variables para discriminar entre pares de genes operones y no-operones. Un estudio comparativo del método de selección de características basado en redes MLP y un árbol decisión CHAID demostró un mejor desempeño de las redes MLP en la selección del subconjunto de característica relevantes, utilizando la precisión obtenida en la identificación de operones en diferentes organismos bacterianos como medida de desempeño. Esta versión mejorada del método logró identificar casi todos los operones determinados experimentalmente en todos los organismos modelos mencionados anteriormente, obteniendo precisiones de 96.5 %, 94.0 %, 93.2 %, 96.5 %, 97.9 %, 92.7 %, respectivamente. Por lo que se sabe, estás son las precisiones más altas obtenidas hasta el momento por un método de reconocimiento de operones. Además, el haber realizado un proceso de selección de características relevantes también ayudó a tener una mejor compresión del modelo, disminuir los tiempos de procesamiento de los datos, utilizar menor requerimientos donde almacenar la información, y lo más importante en este caso, tener un menor costo en la obtención de los datos. Adicionalmente, el proceso de selección de características tiene un impacto positivo en la comprensión de la biogénesis de las unidades de transcripción de las bacterias.

Por otra parte, la identificación de operones no se limitó a genes que tienen asignado un grupo de genes ortólogos COG y por lo tanto valores en la BD de STRING. Para genes que no lo tienen, se desarrolló un método en el cual, primero se les definió con éxito nuevos grupos de genes ortólogos y posteriormente, se definió una nueva característica basada en la conservación de vecindad de genes a través de diversos genomas bacterianos aproximada a los valores de STRING, que se denominó STRING-like. De manera relevante, se encontró que la precisión de una red MLP que usó como entrada el STRING-like y distancias intergénicas es mayor a la que

se obtiene cuando se usó los valores directos de la conservación de vecindad y distancias.

De esta manera, si los genes tienen asignado un grupo de genes ortólogos pueden ser analizados con la red MLP que utiliza el subconjunto de características relevantes, y si no, utiliza las distancias intergénicas y los valores de STRING-like. Con ésto, se logra tener una metodología de identificación de operones que es capaz de predecir la naturaleza *operón*/*no-operón* de todos y cada uno de los genes de cualquier genoma bacteriano, con una precisión mayor a lo reportado previamente. Además, se encontró que algunas de las inconsistencias entre la metodología de identificación de operones y el conjunto de datos que sirvió como referencia podrían surgir como consecuencia de una anotación incorrecta de genes inexistentes en el genoma, errores en la curación de los datos, o incluso por una interpretación errónea de los datos experimentales, y no necesariamente por un reconocimiento erróneo del método. Considerando este hecho, es probable que la precisión alcanzada por la metodología propuesta incluso sea mayor a la reportada en los resultados, llegando muy cerca del máximo teórico del 100 %.

8.2. Discusiones

La investigación queda abierta en varios aspectos que son manejados en este trabajo, algunos de ellos son los siguientes:

- Eliminar el sesgo en la característica de distancias intergénicas. Como se mencionó en la sección 5.1.1, sólo el $4\,\%$ de los pares de genes de la clase *no-operón* de *E. coli* tienen distancias menores a 50 bp, mientras que para pares de genes de la clase *operón* es del $69\,\%$. Esto hace evidente la tendencia de distancias intergénicas menores para pares de genes de la clase *operón*, tendencia que se mantiene en pares de genes *operón* de *B. subtilis*(Figura 5.1-B). Asimismo, al ser los promedios de las medias de las distancias intergénicas de pares de genes directores de un grupo de 914 bacterias muy similares a las distancias de los directores de *E. coli* (Figura 5.1-B), se asumió que esta tendencia se mantiene en todos los organismos bacterianos secuenciados hasta el momento.

  Sin embargo, considerando que se está utilizando el promedio de todas las medias de cada uno de los 914 organismos bacterianos, se tiene también que considerar las desviaciones estándar para conocer el grado de dispersión de dichas medias con respecto al valor promedio. La Figura 8.1-A muestra la media de las medias de las distancias intergénicas de los 914 genomas con sus respectivas más-menos dos desviaciones

estándar, siendo 66 bp la desviación estándar global. Estos resultados verifican que para ciertos organismos específicos, la media de sus distancias intergénicas pueden diferir y tener un sesgo importante con respecto a la de *E. coli* tal y como ha sido reportado en Price *et al*. (2005). Por ejemplo, como se puede observar en la Figura 8.1-B en *E. coli*, el 65 % de los pares de genes directones tienen una distancia intergénica menor a 100 bp, mientras que para *Blattabacterium sp* el porcentaje es de 91 %, un 26 % más. Por otro lado, en *Cyanobacterium UCYN-A* el porcentaje es de sólo 38 %, lo cual representa un 27 % menos que en relación a *E. coli*. Ésto indica una tendencia de distancias intergénicas menores para *Blattabacterium sp* y mayores para *Cyanobacterium UCYN-A* en comparación a las de *E. coli*, lo cual se ve reflejado en la media de sus distancias intergénicas de 28 bp, 187 bp y 120 bp, respectivamente.

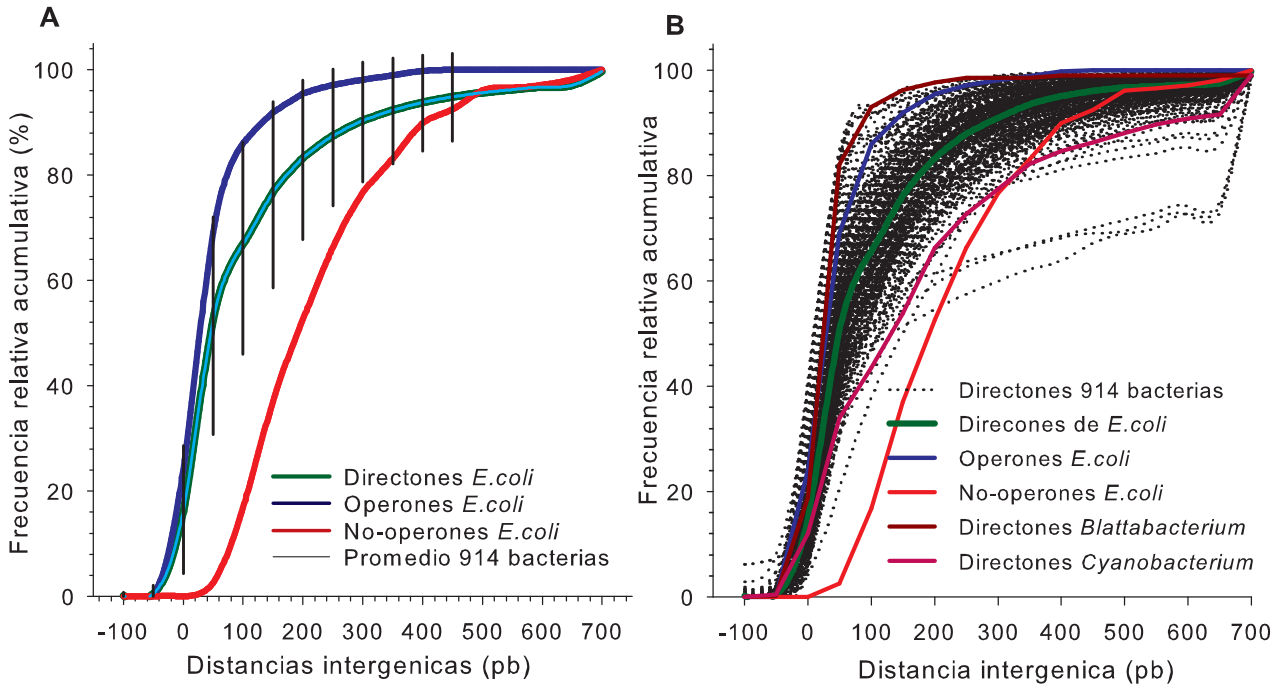

Figura 8.1: Distribución de las frecuencias de distancias intergénicas de pares de genes (A) Operones, no-operones y directones de *E. coli*, *B. subtilis* y promedio de las medias de los directones de 914 organismos bacterianos. (B) Operones, no-operones y directones de *E. coli*, *B. subtilis*, *Blattabacterium sp*, *Cyanobacterium UCYN-A* y otros 910 organismos bacterianos.

Con el método de identificación de operones propuesto en el capítulo 6, la característica de distancia intergénica contribuye en un 27 % en la clasificación de operones. Por consiguiente, para organismos con distancias intergénicas menores que las de *E. coli*

(serie verde de la Figura 8.1-B), se obtendrían más falsos positivos (*e.g. Blattabacterium sp*), mientras para los de mayor distancia, se espera obtener más falsos negativos. Ésto es en sí por la propia contribución de las distancias intergénicas en el reconocimiento de operones y al umbral mínimo de un valor de 300 establecido para los pares de genes que no tienen un grupo de genes ortólogos asignado ($\mathcal{COG}_t$) donde para distancias intergénicas menor a 60 pb se clasifican como operones.

Por lo tanto, para que el sesgo en la variable de distancia intergénica sea el menor posible en la determinación de cuando un par de genes pertenece a la clase *operón* o *no-operón*, en organismos específicos con distancias diferentes a las de *E. coli*, se podría: (a) Evaluar no sólo las características relevantes, sino la contribución de las distancias intergénicas en función a las características utilizadas y la combinación de éstas. (b) Diseñar varios modelos, basados en redes MLP, que sean utilizados en diferentes rangos de valores de distancias intergénicas esperando que la contribución de dicha variable en cada rango sea distinta. Por ejemplo, tres redes MLP que tengan como entrada las características más relevantes identificadas en la Sección 6.3.1, pero que sean entrenadas con distintos datos en función a tres rangos continuos de distancias intergénicas: i) menores a 75bp, ii) mayores a 75 pero menores a 150 y iii) mayores a 150.

- Mejorar los grupos ROG generados y el valor de STRING-like. Cuando se realizó el análisis para generar los nuevos grupos de genes ortólogos ROG complementarios a los COG, se contaba con 540 genomas bacterianos, sin embargo en la actualidad existen más de 1100 genomas completamente secuenciados. Ésto hace evidente que es necesario considerar, en un nuevo análisis, un mayor número de genomas no-redundantes a la hora de crear los grupos ROG y por consiguiente volver a calcular los valores del STRING-like, descrito en el capítulo 7. Ésto permitiría mejorar la predicción de operones en estos nuevos genomas bacterianos. Además, sería recomendable probar otros algoritmos de agrupamiento para determinar los grupos ROG, para proporcionar una manera de validar y mejorar los resultados que se obtuvieron al utilizar el algoritmo basado en la máxima conectividad de un nodo (Hartuv and Shamir, 2000).

- Usar características genómicas composicionales. Las variables composicionales están más enfocadas a determinar el punto de inicio y fin de la transcripción del DNA al RNA que contiene la información necesaria para la síntesis de proteínas. Como se mencionó en la introducción, la capacidad actual de reconocimiento e identificación de los patrones de

caracteres que delimitan el inicio (promotor) y el final de la transcripción (terminador) de manera precisa es limitada debido a que las secuencias de caracteres que los identifican son variantes e inciertas y se desconoce de manera precisa sus reglas de sintaxis. Sin embargo, después de realizar la identificación de operones pueden ser utilizadas en una post-clasificación utilizando solamente los casos con alta certeza, es decir tener una alta especificidad aunque la sensibilizad sea baja. La identificación de promotores permitiría reconocer, no sólo los operones sino las unidades transcripcionales internas de un operón. Por otra parte, la identificación de terminadores permitiría mejorar la precisión alcanzada hasta el momento al eliminar inconsistencias entre las predicciones del método de identificación y los operones que se generan *in vivo*. Tal es el caso del ejemplo que se mencionó en la sección 5.3.1 del capítulo 5 del par de genes *lacI-lacZ* que son reconocidos por el método como pares de genes operones, sin embargo no lo son. Por una parte, existe el operón *lacZ-lacY-lacA* (operón de lactosa y el primero en ser descrito) y por otra el operón *mhpR-lacI*. No obstante, el método define un sólo operón compuesto por cinco genes *mhpR-lacI-lacZ-lacY-lacA*. Dicha predicción en el método es debida a que los genes *lacI-lacZ* tienen un valor relativamente alto de relación funcional en STRING y una distancia intergénica de sólo 124 bp. Sin embargo, existe un terminador transcripcional entre ambos genes que hace que exista una inconsistencia en la predicción, pudiendo ser evitada con la identificación de terminadores.

- Mejorar el desempeño de la red MLP. Para disminuir los errores de clasificación de pares de genes cuyo valor de confianza de la predicción se encontró cerca de la frontera (valor de confianza de 0.5) donde la red MLP tiende a no distinguir correctamente entre la clase o y $\overline{o}$, se podría diseñar una nueva MLP que tuviera dos neuronas en la capa de salida en vez de una, con la finalidad de evaluar si disminuyen los errores en la clasificación.

- Probar otras herramientas computacionales de clasificación. En este trabajo, las características de entrada utilizadas en la identificación de operones son obtenidas a partir de las relaciones entre grupos de genes ortólogos a través de diversos genomas bacterianos, lo que hace que sean representativas no sólo de un organismo en particular. Este es un factor esencial en la alta precisión obtenida en diversos genomas bacterianos donde la metodología propuesta fue probada. Al utilizar las mismas características, las redes MLP mostraron ser una herramienta de clasificación que generalizan mejor que los árboles de decisión CHAID. Sin embargo, al hacer uso del algoritmo de gradiente descendiente no se puede garantizar en ningún momento que el mínimo que se haya encontrado sea global. Una vez que la red se asienta en un mínimo, sea local o global, cesa el aprendizaje, aunque el error siga siendo demasiado alto y los pesos no sean los

óptimos por haber alcanzado un mínimo local. Por lo tanto, a pesar de tener un error pequeño en la identificación de operones tanto en el conjunto de datos de entrenamiento como en el de prueba, sería importante probar otras herramientas computacionales tales como máquinas de soporte vectorial o probabilidades bayesianas para comparar su desempeño. De esta manera, se probaría no sólo la relevancia de las características genómicas seleccionadas en este trabajo, sino diversos métodos computacionales, lo cual podría mejorar el desempeño alcanzado.

APÉNDICE A

CONJUNTO DE DATOS

El conjunto de datos utilizados como verdaderos positivos, es decir que pertenecientes a la clase operón, de cada organismos fueron los siguientes:

Curados manualmente. Los operones fueron determinados por un grupo de curadores operón por operón. El proceso de curación comienza con la búsqueda de todos los artículos que contienen información acerca de la regulación transcripcional y la organización operón. El primer paso de esta búsqueda es reunir a los resúmenes de base de datos PubMed mediante el uso de un conjunto de palabras clave pertinentes. A continuación, los resúmenes de estos trabajos se leen y se selecciona para obtener los artículos completos con el fin de leerlos. El equipo de curadores sigue un conjunto unificado de criterios o directrices que se expanden como la experiencia se acumula.

***Escherichia coli*:** La lista de 344 operones de *E. coli* con evidencia fuerte (experimentalmente identificados) fueron obtenidos de la BD de RegulonDB versión 6.4 en el sitio web http://regulondb.ccg.unam.mx/.

***Basilus subtilis*:** La lista de 509 operones de *B. subtilis* con evidencia fuerte (experimentalmente identificados) fueron obtenidos de la BD de DBTBS version 5.0 en el sitio web http://dbtbs.hgc.jp/.

***50 organismos*:** La lista de 202 operones de los 50 organismos parcialmente estudiados fueron fueron obtenidos de la BD de DBTBS version 2011 en el sitio web http://www.genome.sk.ritsumei.ac.jp/odb2/index

APÉNDICE B

REGLAS DE DECISIÓN DEL ÁRBOL CHAID

Las siguientes son las reglas generadas del árbol CHAID para identificar operones utilizando datos de *E.coli*.

Nodo 1

HACER SI $\Big($ Value(Y_8^{ec}) $<= 56$ $\Big)$

COMPUTAR nod-001 $= 1$

COMPUTAR pre-001 $=' operonic'$

COMPUTAR prb-001 $= 0{,}97$

FIN SI

EJECUTAR

Nodo 4

HACER SI $\Big($ $\big($ Value(Y_8^{ec})> 56 AND Value(Y_8^{ec}) $<= 167$ $\big)$

AND $\big($No Value(Y_1^{ec}) OR Value(Y_1^{ec}) $<= 333$ $\big)$ $\Big)$

COMPUTAR nod-001 $= 4$

COMPUTAR pre-001 $=' no - operonic'$

COMPUTAR prb-001 $= 0{,}82$

FIN SI

EJECUTAR

Nodo 5

HACER SI $\Big($ $\big($Value(Y_8^{ec})$>$ 56 AND Value(Y_8^{ec}) $<=$ 167 $\big)$
AND Value(Y_1^{ec}) $>$ 333 $\Big)$

COMPUTAR nod-001 $= 5$

COMPUTAR pre-001 $=' operonic'$

COMPUTAR prb-001 $= 0{,}83$

FIN SI

EJECUTAR

Nodo 6

HACER SI $\Big($ Value(Y_8^{ec}) > 167 AND
$\big($No Value(Y_7^{ec}) OR Value(Y_7) < 453 $\big)$ $\Big)$

COMPUTAR nod-001 $= 6$

COMPUTAR pre-001 $=' no - operonic'$

COMPUTAR prb-001 $= 0{,}96$

FIN SI

EJECUTAR

Nodo 7

HACER SI $\Big($ Value(Y_8^{ec}) > 167 AND Value(Y_7^{ec}) > 453 $\Big)$

COMPUTAR nod-001 $= 7$

COMPUTAR pre-001 $=' operonic'$

COMPUTAR prb-001 $= 0{,}571429$

FIN SI

EJECUTAR

APÉNDICE C

ORGANISMOS DE REFERENCIA

Lista de los 300 organismos de referencia utilizados para calcular la conservación de vecindad de genes adyacentes. Cada fila describe un organismo.

Tabla C.1: Lista de 300 organismos de referencia utilizados para calcular la conservación de vecindad de genes adyacentes

Nombre del organismo bacteriano
Aquifex aeolicus VF5
Arthrobacter aurescens TC1
Acidobacteria bacterium Ellin345
Alcanivorax borkumensis SK2
Acinetobacter baumannii ATCC 17978
Acidothermus cellulolyticus 11B
Acidiphilium cryptum JF-5
Alkalilimnicola ehrlichei MLHE-1
Archaeoglobus fulgidus DSM 4304 (VC-16)
Anaeromyxobacter sp. Fw109-5
Acidovorax sp. JS42
Anaplasma marginale St. Maries
Alkaliphilus metalliredigens QYMF
Aeropyrum pernix K1
Anaplasma phagocytophilum HZ
Aeromonas salmonicida A449
Azoarcus sp. BH72
Buchnera aphidicola 5A endosymbiont of Acyrthosiphon pisum
Buchnera aphidicola Sg endosymbiont of Schizaphis graminum (greenbug)
Bdellovibrio bacteriovorus HD100
Bartonella bacilliformis KC583
Bradyrhizobium sp. BTAi1
Borrelia burgdorferi B31
Bacillus cereus ATCC 10987
Buchnera aphidicola Cc endosymbiont of Cinara cedri
Baumannia cicadellinicola Hc symbioint of Homalodisca coagulata
Bacillus clausii KSM-K16
Blochmannia floridanus endosymbiont of Camponotus floridanus
Bacteroides fragilis NCTC 9343
Continúa en la siguiente página

Nombre del organismo bacteriano
Bacillus halodurans C-125
Bradyrhizobium japonicum USDA110
Bifidobacterium longum DJO10A
Bordetella pertussis Tohama I
Candidatus Blochmannia pennsylvanicus BPEN endosymbiont of Camponotus pennsylvanicus
Bartonella quintana Toulouse Bacillus subtilis 168
Chlamydophila abortus S26/3
Clostridium acetobutylicum ATCC 824
Coxiella burnetii Dugway 5J108-111
Clostridium beijerinckii NCIMB 8052
Clostridium botulinum F Langeland
Chlorobium chlorochromatii CaD3
Caulobacter crescentus CB15
Campylobacter curvus 525.92
Clostridium difficile 630
Corynebacterium diphtheriae gravis NCTC13129
Campylobacter fetus subsp. fetus 82-40
Corynebacterium glutamicum ATCC 13032 (Kyowa Hakko)
Campylobacter hominis ATCC BAA-381
Cytophaga hutchinsonii ATCC 33406
Carboxydothermus hydrogenoformans Z-2901
Corynebacterium jeikeium K411
Campylobacter jejuni RM1221
Clostridium kluyveri DSM 555
Clavibacter michiganensis subsp. michiganensis NCPPB 382
Chlamydia muridarum Nigg (Chlamydia trachomatis MoPn)
Clostridium novyi NT Clostridium perfringens ATCC 13124
Chlorobium phaeobacteroides DSM 266
Chlamydophila pneumoniae CWL029
Colwellia psychrerythraea 34H
Candidatus Carsonella ruddii PV endosymbiont of Pachypsylla venusta
Chromohalobacter salexigens DSM 3043
Caldicellulosiruptor saccharolyticus DSM 8903
Clostridium tetani E88
Chlorobaculum tepidum TLS
Clostridium thermocellum ATCC 27405
Chromobacterium violaceum ATCC 12472
Cyanobacteria bacterium Yellowstone A-Prime (Synechococcus sp. JA- 3-3Ab)
Dechloromonas aromatica RCB
Desulfovibrio desulfuricans G20
Dehalococcoides ethenogenes 195
Deinococcus geothermalis DSM 11300
Dichelobacter nodosus VCS1703A
Desulfotalea psychrophila LSv54
Desulfotomaculum reducens MI-1
Desulfitobacterium hafniense Y51
Desulfovibrio vulgaris subsp. vulgaris Hildenborough
Ehrlichia chaffeensis Arkansas
Escherichia coli K-12 MG1655
Enterococcus faecalis V583 vancomycin-resistant clinical isolate
Erythrobacter litoralis HTCC2594
Ehrlichia ruminantium Welgevonden (France)
Flavobacterium johnsoniae UW101
Continúa en la siguiente página

Nombre del organismo bacteriano
Fervidobacterium nodosum Rt17-B1
Fusobacterium nucleatum ATCC 25586
Flavobacterium psychrophilum JIP02/86
Frankia sp. CcI3
Francisella tularensis subsp. tularensis Schu 4
Granulobacter bethesdensis CGDNIH1
Gramella forsetii KT0803
Geobacillus kaustophilus HTA426
Geobacter metallireducens GS-15
Gluconobacter oxydans 621H
Geobacter sulfurreducens PCA
Geobacter uraniumreducens Rf4
Gloeobacter violaceus PCC7421
Halobacterium salinarium NRC-1
Hyperthermus butylicus DSM 5456
Hahella chejuensis KCTC 2396
Haemophilus ducreyi 35000HP
Halorhodospira halophila SL1
Helicobacter hepaticus ATCC 51449
Hyphomonas neptunium ATCC 15444
Helicobacter pylori J99
Haloquadratum walsbyi DSM 16790 (HBSQ001)
Idiomarina loihiensis L2TR
Jannaschia sp. CCS1
Kineococcus radiotolerans SRS30216
Lactobacillus brevis ATCC 367
Lactobacillus casei ATCC 334
Lactobacillus delbrueckii subsp. bulgaricus ATCC 11842
Lactobacillus gasseri ATCC 33323
Lawsonia intracellularis PHE/MN1-00
Lactococcus lactis subsp. cremoris SK11
Leuconostoc mesenteroides subsp. mesenteroides ATCC 8293
Listeria monocytogenes F2365 (serotype 4b)
Legionella pneumophila subsp. pneumophila Philadelphia 1
Lactobacillus reuteri DSM 20016
Lactobacillus sakei 23K
Lactobacillus salivarius subsp. salivarius UCC118
Leifsonia xyli subsp. xyli CTCB07
Mycoplasma agalactiae PG2
Methanococcus aeolicus Nankai-3
Magnetospirillum magneticum AMB-1
Marinobacter aquaeolei VT8
Methanosarcina barkeri fusaro chromosome 1 Candidatus
Methanoregula boonei 6A8
Methanococcoides burtonii DSM 6242
Methylococcus capsulatus Bath
Methanoculleus marisnigri JR1
Mesorhizobium sp. BNC1
Methylobacillus flagellatus KT
Mesoplasma florum L1
Mycoplasma gallisepticum R
Mycoplasma genitalium G-37
Magnetococcus sp. MC-1
Continúa en la siguiente página

Nombre del organismo bacteriano
Methanospirillum hungatei JF-1
Mycoplasma hyopneumoniae 232
Methanococcus jannaschii DSM 2661
Methanopyrus kandleri AV19
Mycobacterium sp. KMS
Methanocorpusculum labreanum Z
Mesorhizobium loti MAFF303099
Methanosarcina mazei Goe1
Mycoplasma mobile 163K
Methanococcus maripaludis S2
Maricaulis maris MCS10
Minibacterium massiliensis (Janthinobacterium sp. Marseille)
Marinomonas sp. MWYL1
Mycoplasma mycoides subsp. mycoides SC PG1
Mycoplasma penetrans HF-2
Mycoplasma pneumoniae M129
Methylibium petroleiphilum PM1
Mycoplasma pulmonis UAB CTIP
Metallosphaera sedula DSM 5348
Methanobrevibacter smithii ATCC 35061
Methanosphaera stadtmanae DSM 3091
Mycoplasma synoviae 53
Moorella thermoacetica ATCC 39073
Mycobacterium tuberculosis F11
Methanobacterium thermoautotrophicum deltaH
Methanosaeta thermophila PT
Methanococcus vannielii SB
Myxococcus xanthus DK 1622
Novosphingobium aromaticivorans DSM 12444
Nocardioides sp. JS614
Nanoarchaeum equitans Kin4-M
Nitrosomonas eutropha C91
Nocardia farcinica IFM 10152
Nitrobacter hamburgensis X14
Nitratiruptor sp. SB155-2
Neisseria meningitidis Z2491 (serogroup A)
Nitrosospira multiformis ATCC 25196
Nitrosococcus oceani ATCC 19707
Natronomonas pharaonis DSM 2160
Nostoc punctiforme PCC 73102
Oceanobacillus iheyensis HTE831
Oenococcus oeni PSU-1
Orientia tsutsugamushi Boryong
Propionibacterium acnes KPA171202
Pseudomonas aeruginosa LESB58
Pyrobaculum aerophilum IM2
Psychrobacter arcticum 273-4
Pyrobaculum arsenaticum DSM 13514
Pseudoalteromonas atlantica T6c
Pelobacter carbinolicus DSM 2380
Pyrobaculum calidifontis JCM 11548 Candidatus
Protochlamydia amoebophila UWE25 endosymbiont of Acanthamoeba sp.
Parabacteroides distasonis ATCC 8503

Continúa en la siguiente página

Nombre del organismo bacteriano
Pyrococcus furiosus DSM 3638
Porphyromonas gingivalis W83
Pyrococcus horikoshii OT3
Psychromonas ingrahamii 37
Pyrobaculum islandicum DSM 4184
Parvibaculum lavamentivorans DS-1
Pelodictyon luteolum DSM 273
Prochlorococcus marinus SS120 (subsp. marinus CCMP1375)
Prochlorococcus marinus NATL1A
Prochlorococcus marinus MIT 9301
Prochlorococcus marinus MIT 9313
Pasteurella multocida PM70
Polynucleobacter necessarius subsp. asymbioticus QLW-P1DMWA-1
Polaromonas sp. JS666
Pelobacter propionicus DSM 2379
Pediococcus pentosaceus ATCC 25745
Pseudomonas putida F1
Psychrobacter sp. PRwf-1
Pelotomaculum thermopropionicum SI
Picrophilus torridus DSM 9790
Candidatus Pelagibacter ubique HTCC1062
Prosthecochloris vibrioformis DSM 265
Rhodopirellula baltica SH 1 (Pirellula sp. strain 1)
Rickettsia bellii RML369-C
Rickettsia conorii Malish 7
Roseobacter denitrificans OCh 114
Rhizobium etli CFN 42
Rhodoferax ferrireducens T118 (DSM 15236)
Rhodococcus jostii RHA1
Rhizobium leguminosarum bv. viciae 3841
Candidatus Ruthia magnifica Cm
Rhodopseudomonas palustris BisB18
Rhodopseudomonas palustris BisB5
Roseiflexus sp. RS-1
Rhodospirillum rubrum ATCC 11170
Ralstonia solanacearum GMI1000
Rhodobacter sphaeroides ATCC 17025
Rubrobacter xylanophilus DSM 9941
Syntrophus aciditrophicus SB
Streptococcus agalactiae A909 (serotype Ia)
Sphingopyxis alaskensis RB2256
Syntrophus aciditrophicus SB
Staphylococcus aureus subsp. aureus N315 meticillin-resistant (MRSA)
Streptomyces coelicolor A3(2)
Saccharophagus degradans 2-40 Saccharopolyspora erythraea NRRL 2338
Syntrophobacter fumaroxidans MPOB
Staphylococcus haemolyticus JCSC1435
Shewanella sp. W3-18-1
Silicibacter sp. TM1040
Sinorhizobium meliloti 1021
Staphylothermus marinus F1
Streptococcus pneumoniae D39
Salinibacter ruber DSM 13855

Continúa en la siguiente página

Nombre del organismo bacteriano
Sulfolobus solfataricus P2
Staphylococcus saprophyticus subsp. saprophyticus ATCC 15305
Streptococcus suis 05ZYH33
Symbiobacterium thermophilum IAM14863
Sulfolobus tokodaii strain7
Salinispora tropica CNB-440
Sulfurovum sp. NBC37-1
Solibacter usitatus Ellin6076
Sphingomonas wittichii RW1
Syntrophomonas wolfei subsp. wolfei str. Goettingen
Synechococcus elongatus PCC6301
Synechococcus sp. CC9605
Synechococcus sp. CC9311
Synechocystis sp. PCC 6803
Synechococcus sp. RCC307
Thermoplasma acidophilum DSM 1728
Thiobacillus denitrificans ATCC 25259
Thiomicrospira crunogena XCL-2
Treponema denticola ATCC 35405
Thermosynechococcus elongatus BP-1
Trichodesmium erythraeum IMS101
Thermobifida fusca YX
Thermococcus kodakaraensis KOD1
Thermotoga maritima MSB8
Thermosipho melanesiensis BI429
Treponema pallidum subsp. pallidum Nichols
Thermofilum pendens Hrk 5
Thermoanaerobacter tengcongensis MB4(T)
Thermus thermophilus HB27
Thermoplasma volcanium GSS1
Tropheryma whipplei TW08/27
Ureaplasma urealyticum serovar 10 ATCC 33699
Verminephrobacter eiseniae EF01-2
Candidatus Vesicomyosocius okutanii HA endosymbiont of Calyptogena okutanii
Wolbachia wBm endosymbiont of Brugia malayi TRS
Wigglesworthia glossinidia endosymbiont of Glossina brevipalpis (tsetse fly)
Wolbachia pipientis endosymbiont of Culex quinquefasciatus Pel (mosquito)
Wolinella succinogenes DSM 1740
Xanthobacter autotrophicus Py2
Xanthomonas campestris pv. campestris 8004 (Beijing)
Xylella fastidiosa Temecula1
Zymomonas mobilis subsp. mobilis ZM4

BIBLIOGRAFÍA

Altschul.S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. "Basic local alignment search tool." *Journal of molecular biology* 215, 3: (1990) 403–410. 82

Alyuda. "Alyuda NeuroIntelligence 2.2(5.77).", 2005. `http://www.alyuda.com/neural-networks-software.htm`. 48, 66

Baily, D., and D.M. Thomson. "Developing neural network applications." *AI Expert* 5(6): (1990) 33–41. 30

Barnett, V., and T. Lewis. *Outliers in Statistical Data, 3rd edition.* John Wiley & Sons, 1994. 36

Ben-Bassat, M. *Use of distance measures, information measures and error bounds in feature evaluation*, North-Holland Publishing Company, 1982, chapter Handbook of Statistics. 58

Bengio, Y., and Y. Grandvalet. "No Unbiased Estimator of the Variance of K-Fold Cross-Validation." *Journal of Machine Learning Research* 5: (2004) 1089–1105. 48, 73

Bergman, N.H., K.D. Passalacqua, P.C. Hanna, and S.Q. Zhaohui. "Operon Prediction for Sequenced Bacterial Genomes without Experimental Information." *Applied and environmental microbiology* 73, 3: (2007) 846–854. 4, 13, 14, 15, 17

Blum, A.L., and P Langley. "Selection of relevant features and examples in machine learning." *Artificial Intelligence* 97: (1997) 245–271. 6, 56

Bockhorst, J., M. Craven, D. Page, J. Shavlik, and J. Glasner. "Bayesian network approach to operon prediction. Bioinformatics." *Bioinformatics* 19, 10: (2003) 1227–1235. 4, 13, 14, 43

Brouwer, R.W., O.P. Kuipers, and S.A. van Hijum. "The relative value of operon predictions." *Briefings in Bioinformatics* 9: (2008) 367–365. 77

Byvatov, D., U. Fechner, J. Sadowski, and G. Schneider. "Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification." *Journal of Chemical Information and Modeling* 43: (2003) 1882–1889. 22, 23

Caruana, R., and D. Freitag. "Greedy Atributte Selection." In *International Conference on Machine Learning*. 1994. 58

Cavalli-Sforza, L.L. "The Human Genome Diversity Project: past, present and future." *Nature Reviews Genetics* 6: (2006) 333–340. 3

Charaniya, S., S. Mehra, W. Lian, K.P. Jayapal, G. Karypis, and W. Hu. "Transcriptome dynamics-based operon prediction and verification in Streptomyces coelicolor." *Nucleic Acids Research* 35: (2007) 7222–7236. 15

Chen, W., S.H. Hsu, and H.P. Shen. "Application of SVM and ANN for intrusion detection." *Expert Systems with Applications* 32: (2005) 1846–1856. 22, 23

Chen, X., Z. Su, P. Dam, B. Palenik, Y. Xu, and T. Jiang. "Operon prediction by comparative genomics: an application to the Synechococcus sp. WH8102 genome." *Nucleic Acids Research* 32, 7: (2004a) 2147–2157. 4, 13, 14, 15, 16

Chen, X., Z. Su, Y. Xu, and T. Jiang. "Computational prediction of operons in Synechococcus sp. WH8102." *Genome Informatics* 15, 2: (2004b) 211–222. 4, 13, 14, 15, 16, 17

Cybenko, G. "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals and Systems,* 2: (1989) 303–314. 21, 25, 46

Daelemans, D., V. Hoste, F. De Meulder, and B. Naudts. "Combined Optimization of Feature Selection and Algorithm Parameters in Machine Learning of Language." In *Lecture Notes in Computer Science: Machine Learning*. 2003. 58

Dam, P., V. Olman, K. Harris, Z. Su, and Y. Xu. "Operon prediction using both genome-specific and general genomic information." *Nucleic Acids Research* 35, 1: (2007) 288–298. 4, 13, 14, 15, 18, 19, 20, 67, 90

Das, S. "Filters, wrappers, and a boosting-based hybrid for feature selection." In *Proceedings of the Eighteenth International Congress of Machine Learning*. 2001. 59

Dash, M., K. Choi, P. Scheuermann, and H. Liu. "Feature Selection for Clustering: A Filter solution." In *Proceedings of the 2002 IEEE International Conference on Data Mining*. 2002. 58

De Hoon, M.J., S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano. "Predicting the operon structure of Bacillus subtilis using operón length, intergene distance and gene expression information." In *Pacific Symposium on Biocomputin*. PSB, 2004, volume 9, 276–287. 4, 13, 14, 15, 16, 20

Ding, C.H., and I. Dubchak. "Multi-class protein fold recognition using support vector machines and neural networks." *Bioinformatics* 17: (2001) 349–358. 22, 23

Duda, Hart P.E., R.O., and D.G. Stork. *Pattern clasification, Second Ed.,* . John Wiley & Sons, 2000. 28

Edwards, M.T., S.C. Rison, N.G. Stoker, and L. Wernisch. "A universally applicable method of operon map prediction on minimally annotated genomes using conserved genomic context." *Nucleic Acids Research* 33, 10: (2005) 3253–3262. 4, 13, 14, 15, 19, 85

Ermolaeva, M.D., O. White, and S.L. Salzberg. "Prediction of operons in microbial genomes." *Nucleic Acids Research* 29, 5: (2001) 1216–1221. 14, 15, 16, 19

Felsenstein, J. "PHYLIP – Phylogeny Inference Package (Version 3.2)." *Cladistics* 5: (1989) 164–166. 83

Gama-Castro, S., V. Jimenez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M.I. Penaloza-Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Muniz-Rascado, I. Martinez-Flores, and H. et al. Salgado. "RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation." *Nucleic Acids Research* 36: (2008) D120–D124. 39, 49, 76

Gevrey, M., L. Dimopoulos, and S. Lek. "Review and comparison of methods to study the contribution of variables in artificial neural network models." *Ecological Modelling* 160: (2003) 249–264. 6, 56, 63, 65

Gori, M., and A. Tesi. "On the problem of local minima in backpropagation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14: (1992) 76–86. 30, 47

Gross, J.L., and J. Yellen. *Graph Theory and Its Applications*. CRC Press, 1998. 84

Guell, M., V. van Noort, E. Yus, W.H. Chen, J. Leigh-Bell, K. Michalodimitrakis, T. Yamada, M. Arumugam, T Doerks, S. Kuhner, and et al. "Transcriptome complexity in a genome-reduced bacterium." *Science* 326: (2009) 1268–1271. 40

Guyon, I., and A. Elisseeff. "An Introduction to Variable and Feature Selection." *Journal of Machine Learning Research* 3: (2003) 1157–1182. 57, 58

Hartuv, E., and R. Shamir. "A clustering algorithm based on graph connectivity." *Information Processing Letters* 76: (2000) 4–6. 84, 98

Haykin, S. *Neural Networks: A comprehensive Foundation*. Prentince Hall, 1999. 21, 23, 24, 26, 28, 29, 30, 46, 54

Hecht-Nielsen, R. "Replicator neural networks for universal optimum source coding." *Science* 269: (1995) 1860–1863. 30, 47

Hornik, K., M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators,." *Neural Networks* 2(5): (1989) 359–366. 25, 65

Hsinchun, Z.H., H. Chen, C.J. Hsu, W.H. Chen, and S. Wu. "Credit rating analysis with support vector machines and neural networks: a market comparative study." *Decision Support Systems* 37: (2004) 543–558. 22, 23

Isa, I.S., S. Omar, Z. Saad, and M.K. Osman. "Performance Comparison of Different Multilayer Perceptron Network Activation Functions in Automated Weather Classification." In *Mathematical/Analytical Modelling and Computer Simulation*. 2010. 48

Jack, L.B., and K.A. Nandi. "Fault detection using support vector machines and artificial neural networks." *Mechanical Systems and Signal Processing* 16: (2002) 373–390. 22, 23

Jacob, E., R. Sasikumar, and K.N. Nair. "A fuzzy guided genetic algorithm for operon prediction." *Bioinformatics* 21, 8: (2005) 1403–1407. 4, 13, 14, 15, 16, 19, 20, 67, 85

Janga, S.C., and G. Moreno-Hagalsieb. "Conservation of adjacency as evidence of paralogous operons." *Nucleic Acids Research,* 32, 18: (2004) 5392–5397. 85

Jensen, L.J., M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, and M. et al Simonovic. "STRING 8–a global view on proteins and their functional interactions in 630 organisms." *Nucleic Acids Research* 37: (2009) D412–D416. 5, 41, 43, 62

Kaastra, I., and M. Boyd. "Design a Neural Network for Forecasting Financial and Economic Time Series." *Neurocomputing* 10: (1996) 215–236. 29

Kamio, Lin C.K. Regue M., Y., and H.C. Wu. "Characterization of the ileS-lsp operon in Escherichia coli. Identification of an open reading frame upstream of the ileS gene and potential promoter(s) for the ileS-lsp operon." *Journal of Biological Chemistry* 260: (1985) 5616–5620. 52

Kanehisa, M., M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, and T. Tokimatsu. "KEGG for linking genomes to life and the environment." *Nucleic Acids Research* 36, 1: (2008) D480–D484. 62

Karp, P.D., C.A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahre´n, S. Tsoka, N. Darzentas, V. Kunin, and N. Lopez-Bigas. "Expansion of the BioCyc collection of pathway/genome databases to 160 genomes." *Nucleic Acids Research* 33: (2005) 6083–6089. 62

Kass, G.V. "An Exploratory Technique for Investigating Large Quantities of Categorical Data." *Applied Statistics* 29: (1980) 119–127. 69

Kerrien, S., Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, and R. Huntley. "IntAcT: open source resource for molecular interaction data." *Nucleic Acids Research* 35: (2007) D561–D565. 62

Kim, H.B., S.H. Jung, T.G. Kim, and K.H. Park. "Fast learning method for back-propagation neural network by evolutionary adaptation of learning rates." *Neurocomputing* 11: (1996) 101–106. 47

Kohavi, R., and G. John. "Wrappers for feature subset selection." *Artificial Intelligence* 97: (1997) 273–324. 58

Kohavi, R., and F. Provost. *Machine Learning*, Kluwer Academic, 1998, chapter Glossary of Terms, 271–274. 37

Kolmogorov, A.N. "On the representation of continuous functions of several variables by means of superposition of continuous functions of one variable." *Doklady Akademii Nauk SSSR* 114: (1957) 953–956. 25

Li, G., D. Che, and Y. Xu. "Universal operon predictor for prokaryotic genomes." *Journal of Bioinformatics and Computational Biology* 7, 1: (2009) 19–38. 15, 18, 19

Li, S.J., and J.E. Jr. Cronan. "Growth rate regulation of Escherichia coli acetyl coenzyme A carboxylase, which catalyzes the first committed step of lipid biosynthesis." *Journal of Bacteriology* 175: (1993) 332–340. 52

Liu, H., and R. Setiono. "Proceedings of IEEE 7th International Conference on Tools with Artificial Intelligence." In *Chi-square: Feature Selection and Discretization of Numeric Attributes*. 1995. 69

———. "International Conference on Machine Learning." In *A Probabilistic Approach to Feature Selection: A Filter Solution*. 1996. 58

Lui, H., and L. Yu. "Toward integrating feature selection algorithms for classification and clustering." *IEEE Transactions on Knowledge and Data Engineering* 17(4): (2005) 491–502. 57

Marcotte, E.M., Pellegrini M., Leung H.N., D.W. Rice, T.O. Yeates, and D. Eisenberg. "Detecting protein function and proteint protein interactions from genome sequences." *Science* 285: (1999) 751–753. 61

Masters, T. *Practical Neural Network Reicpes in C++*. Academic Press, 1993. 28, 30, 47, 66

Miller, Bouvier J. Stragier P., K.W., and H.C. Wu. "Identification of the genes in the Escherichia coli ileS-lsp operon. Analysis of multiple polycistronic mRNAs made in vivo." *Journal of Biological Chemistry* 262: (1987) 7391–7397. 52

Mindell, D.P., and Meyer. "Homology evolving." *Trends in Ecology and Evolution* 16, 8: (2001) 434–440. 35

Myers, M., and A.D. Well. *Research Design and Statistical Analysis (second edition ed.)*. Lawrence Erlbaum, 2003. 63

Norusis, M.J. *PASW Statistics 18 Guide to Data Analysis.* Upper Saddle River, 2008. 70

Okuda, S., and A.C. Yoshizawa. "ODB: a database for operon organizations." *Nucleic Acids Research* 39: (2011) D552–D555. 40

Olden, J.D., M.K. Joy, and R.G. Death. "An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data." *Ecological Modelling* 178: (2004) 389–397. 65

Overbeek, R., M. Fonstein, M. D'Souza, G.D. Pusch, and N. Maltsev. "The use of gene clusters to infer functional coupling." *Proc. Natl. Acad. Sci. USA* 96: (1999) 2896–2901. 82

Pertea, M., K. Ayanbule, M. Smedinghoff, and S.L. Salzberg. "OperonDB: a comprehensive database of predicted operons in microbial genomes." *Nucleic Acids Research* 37(Database Issue): (2009) D479–D482. 19

Price, M.N., A.P. Arkin, and E.J. Alm. "The Life-Cycle of Operons." *PLoS Genetics* 2, 6: (2006) (e96):0859–0873. 15, 17

Price, M.N., K.H. Huang, E.J. Alm, and A.P. Arkin. "A novel method for accurate operon predictions in all sequenced prokaryotes." *Nucleic Acids Research,* 33, 3: (2005) 880–892. 4, 13, 14, 19, 85, 97

Raina, Missiakas D. Baird L. Kumar S., S., and C. Georgopoulos. "Identification and transcriptional analysis of the Escherichia coli htrE operon which is homologous to pap and related pilin operons." *Journal of Bacteriology* 175: (1993) 5009–5021. 52

Ramaswami, M, and R. Bhaskaran. "A CHAID Based Performance Prediction Model in Educational Data Mining." *Journal of Computer Science Issues* 17: (2010) 10–18. 69

Roback, P., J. Beard, D. Baumann, C. Gille, K. Henry, S. Krohn, H. Wiste, M.I. Voskuil, C. Rainville, and R. Rutherford. "A predicted operon map for Mycobacterium tuberculosis." *Nucleic Acids Research* 35: (2007) 5085–5095. 15

Romero, P.R., and P.D. Karp. "Using functional and organizational information to improve genome-wide computational prediction of transcription units on pathway-genome databases." *Bioinformatic* 20, 5: (2004) 709–717. 4, 13, 15

Rumelhart, Hinton G.E. Williams R.J., D.E. *Parallel distributed processing: Learning internal representations by error propagation*. MIT Press, 1986. 47

Sabatti, C., L. Rohlin, M.K. Oh, and J.C. Liao. "Co-expression pattern from DNA microarray experiments as a tool for operon prediction." *Nucleic Acids Research* 30: (2002) 2886–2893. 15

Salgado, H., G. Moreno-Hagelsieb, T.F. Smith, and J. Collado-Vides. "Operons in Escherichia coli: genomic analyses and predictions." *Proceedings of the National Academy of Sciences* 97, 12: (2000) 6652–6657. 4, 13, 14, 15, 19, 49

Salwinski, L., C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, and D. Eisenberg. "The Database of Interacting Proteins: 2004 update." *Nucleic Acids Research* 32: (2004) D449–D451. 62

Schaefer, C.F., K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K.H. Buetow. "PID: The Pathway Interaction Database." *Nucleic Acids Research* 37: (2009) D674–D679. 62

Sharma, C.M., S. Hoffmann, F. Darfeuille, J. Reignier, S. Findeiss, A. Sittka, S. Chabas, K. Reiche, J. Hackermuller, and R. et al Reinhardt. "The primary transcriptome of the major human pathogen Helicobacter pylori." *Nature* 464: (2010) 250–255. 40

Sierro, N., Y. Makita, M.J.L. de Hoon, and K. Nakai. "DBTBS: a database of transcriptional regulation in Bacillus subtilis containing upstream intergenic conservation information." *Nucleic Acids Research* 36: (2008) D93–D96. 39

Stone, M. "Cross-validatory choice and assessment of statistical predictions." *Journal of the Royal Statistical Society* b36: (1974) 111–133. 31, 48, 73

Taboada, B., C. Verde, and E. Merino. "High accuracy operon prediction method based on STRING database scores." *Nucleic Acids Research* 38, 12: (2010) 130–140. 41, 74, 77

Tatusov, R.L., N.D. Fedorova, J.D. Jackson, A.R. Jacobs, B. Kiryutin, E.V. Koonin, D.M. Krylov, R. Mazumder, S.L. Mekhedov, and A.N. et al. Nikolskaya. "The COG database: an updated version includes eukaryotes." *BMC.Bioinformatics* 4, 1: (2003) 41. 35, 44, 81, 82

Tjaden, B., D.R. Haynor, S. tolyar, C. Rosenow, and E. Kolker. "Identifying operons and untranslated regions of transcripts using Escherichia coli RNA expression analysis. Bioinformatics." *Bioinformatics* 18 Suppl 1, 90001: (2002) S337–S344. 4

Toledo-Arana, A., O. Dussurget, G. Nikitas, N. Sesto, H. Guet-Revillet, D. Balestrino, E. Loh, J. Gripenland, T. Tiensuu, and K. et al Vaitkevicius. "The Listeria transcriptional landscape from saprophytism to virulence." *Nature* 459: (2009) 950–956. 40

Tran, T.T., P. Dam, Z. Su, F.L. Poole, M.W. Adams, G.T. Zhou, and Y Xu. "Operon prediction in Pyrococcus furiosus." *Nucleic Acids Research* 35, 1: (2007) 11–20. 4, 13, 14, 15, 17, 85

Wang, S., Y. Wang, W. Dua, F. Suna, X. Wang, C. Zhoua, and Y. Liang. "A multi-approaches-guided genetic algorithm with application to operon prediction." *Artificial Intelligence in Medicine* 41, 2: (2007) 151–159. 18

West, D.B. *Introduction to Graph Theory - Second edition*. Prentice Hall, 2001. 84

Westover, B.P., J.D. Buhler, J.L. Sonnenburg, and J.I. Gordon. "Operon prediction without a training set." *Bioinformatics* 21, 7: (2005) 880–888. 4, 13, 14, 15, 17, 19, 20, 67, 85

Wolf, Y.I., I.B. Rogozin, A.S. Kondrashov, and E.V. Koonin. "Genome alignment, evolution of prokaryotic genome organization and prediction of gene function using genomic contex." *Genome Research* 11, 3: (2001) 356–372. 16, 85

Wu, T.K., S.C. Huang, and Y. M. ")Evaluation of ANN and SVM classifiers as predictors to the diagnosis of students with learning disabilities." *Expert Systems with Applications* 34: (2008) 1846–1856. 22, 23

Wurtze, O., R. Sapra, F. Chen, Y.W. Zhu, B.A. Simmons, and R. Sorek. "A single-base resolution map of an archaeal transcriptome." *Genome Research* 20: (2010) 133–141. 40

Yada, T., M. Nakao, Y. Totoki, and K. Nakai. "Modeling and predicting transcriptional units of Escherichia coli genes using hidden Markov models." *Bioinformatics* 15, 12: (1999) 987–993. 4

Yan, Y., and J. Moult. "Detection of operons." *PROTEINS: Structure, Function and Bioinformatics* 64: (2006) 615–628. 19

Zhang, G., B. Patuwo, and Y. Hu. "Forecasting with Artificial Neural Networks: The State of Art." *International Journal of Forecasting* 14: (1998) 35–62. 29

Zhang, G.Q., Z.W. Cao, Q.M. Luo, Y.D. Cai, and Y.X. Li. "Operon prediction based on SVM." *Computational Biology and Chemistry* 30: (2006) 233–240. 4, 13, 14, 15, 17, 19, 67, 85

Zheng, Y., J.D. Szustakowski, L. Fortnow, R.J. Roberts, and S. Kasif. "Computational identification of operons in microbial genomes." *Genome Research* 12: (2002) 1221–1230. 13, 15