Universidad de la República

Facultad de Ingeniería

Implementación de clasificadores jerárquicos multiclase para la predicción de función de genes a partir de su ubicación en el genoma

Lic. Diego Silvera

Tutor: Dr. Flavio Pazos Obregón

Cotutor: Dr. Gustavo Guerberoff

Tesis de Maestría en Ingeniería Matemática

Facultad de Ingeniería, Universidad de la República

Montevideo – Uruguay

Junio de 2022

Índice general

1. Introducción **1**

1.1. Predicción de función de genes 2

1.2. Gene Ontology 4

1.2.1. Ontologías de términos GO 5

1.2.2. Anotaciones de términos GO 8

1.2.3. Evidence Codes 9

1.3. Clasificadores Jerárquicos Multiclase 10

1.4. CAFA 13

1.5. Objetivos de la tesis 14

2. Métodos **16**

2.1. Anotaciones jerárquicas 16

2.2. Datasets 17

2.2.1. Ontologías 17

2.2.2. Anotaciones 18

2.2.3. Genomas 18

2.3. Modelado del genoma 19

2.4. Análisis de Enriquecimiento Local 19

2.5. Notación y definiciones básicas 22

2.6. Modelos Jerárquicos 24

2.7. Métricas Jerárquicas 29

2.8. Comparación con un modelo aleatorio 32

2.9. Implementación 33

3. Implementación y Resultados **35**

3.1. Implementación 35

3.1.1. Preprocesamiento . . . 35
3.1.2. Implementación de Modelos y Métricas . . . 36
3.2. Resultados . . . 36
3.2.1. Predicción de nuevas anotaciones . . . 36
3.2.2. Comparación con los métodos de referencia de CAFA . . . 41
3.3. Conclusiones . . . 44

Anexos **46**

A. Predicción automática de funciones **47**
A.1. Métodos basados en Similitud de Secuencias . . . 48
A.2. Métodos Probabilísticos . . . 48
A.3. Métodos de Aprendizaje Automático . . . 49

B. Random Forest **51**
B.1. Modelos tipo Ensamble . . . 51
B.2. Bagging . . . 52
B.3. Random Forest . . . 53

C. Gene function prediction in five model eukaryotes exclusively based on gene relative location through machine learning **55**

Abstract

The recent technological development is generating genomic data much faster than our ability to analyze it. In this context, it is essential to implement tools that reduce the time and cost necessary to determine the functions of genes experimentally, given that the function of most genes is still unknown.

To alleviate this problem, various gene function prediction methods have been developed in recent decades. Some are based on sequence alignments with proteins for which their function has been established experimentally [Clark and Radivojac, 2011, Martin et al., 2004, Engelhardt et al., 2005], and others exploit other types of data: protein structures [Pal and Eisenberg, 2005, Pazos and Sternberg, 2004], expression levels of genes [Huttenhower et al., 2006], temporal transcription profiles [Pazos Obregón et al., 2015], macromolecular interactions [Letovsky and Kasif, 2003, Nabieva et al., 2005], or a combination of several types of them.

Although genes with the same function are known to cluster in different ways in the genome, and their position in the genome is not independent of their biological function, the potential of a gene's position within the genome as a predictive variable of function remains unexplored in eukaryotic organisms.

In this work, a model is implemented to predict gene functions, using data generated from their position in the genome and from known functions, in five model organisms.

The results obtained indicate that, for some organisms and ontologies, the position of a gene is a better predictor of its function than its sequence.

Key words: Gene function prediction, Hierarchical Multiclass Classifier, Gene Ontology, Directed acyclic graph.

Resumen

El reciente desarrollo tecnológico está generando datos genómicos mucho más rápido que nuestra capacidad de analizarlos. Es imprescindible, en este contexto, implementar herramientas que permitan reducir el tiempo y el costo necesario para determinar las funciones de los genes experimentalmente, dado que para la mayoría de los genes aún se desconoce su función.

Para aliviar este problema, en las últimas décadas se han desarrollado varios métodos de predicción de funciones de genes. Algunos se basan en alineamientos de secuencia con proteínas para las cuales su función se ha establecido experimentalmente [Clark and Radivojac, 2011, Martin et al., 2004, Engelhardt et al., 2005], y otros explotan otros tipos de datos: estructuras de proteínas [Pal and Eisenberg, 2005, Pazos and Sternberg, 2004], niveles de expresión de genes [Huttenhower et al., 2006], perfiles temporales de transcripción [Pazos Obregón et al., 2015], interacciones macromoleculares [Letovsky and Kasif, 2003, Nabieva et al., 2005], o una combinación de varios tipos de ellos.

A pesar de que se sabe que los genes con la misma función se agrupan de diferentes maneras en el genoma, y que su posición en el mismo no es independiente de su función biológica, el potencial de la posición de un gen dentro del genoma como variable predictora de la función permanece poco explorado en organismos eucariotas. En este trabajo se implementa un modelo para predecir funciones de genes, utilizando datos generados a partir de su posición en el genoma y de funciones conocidas, en cinco organismos modelo.

Los resultados obtenidos indican que, para algunos organismos y ontologías, la posición de un gen predice mejor su función que la secuencia.

Palabras claves: Predicción de función de genes, Clasificador Jerárquicos Multiclase, Gene Ontology, Grafo acíclico dirigido.

Capítulo 1

Introducción

Un gen es una secuencia de nucleótidos de ADN o ARN que codifica la información necesaria para la síntesis de un producto génico, ya sea una molécula ARN o una proteína. Durante la expresión del ADN, este se copia primero en ARN; el ARN puede ser directamente funcional o ser la plantilla intermedia para una o distintas proteínas con una o varias funciones. El concepto de gen continúa siendo refinado a medida que se descubren nuevos fenómenos; en este trabajo nos referiremos por gen a un segmento de la cadena de ADN que codifica una proteína. Por genoma se entiende la cantidad total de ADN que porta un organismo, incluyendo la totalidad de sus genes.

Comprender la función de los genes (cómo los genes individuales contribuyen a la biología de un organismo a nivel molecular, celular y de organismos) es uno de los objetivos principales de la investigación biomédica, y a pesar de ello, la mayor parte de los genes secuenciados no poseen aún ninguna función conocida [Howe et al., 2020]. Avances en la predicción de las funciones de los genes y sus productos génicos pueden fomentar el progreso en el análisis de enfermedades [Kissa et al., 2015, Zeng et al., 2015, Zhang et al., 2019], desarrollo de fármacos [Barabási et al., 2011, Xuan et al., 2019] y en muchos otros campos que se beneficien del entendimiento de los procesos biológicos [Radivojac et al., 2013, Jiang et al., 2016, Zhou et al., 2019, Shehu et al., 2016]. Además, el conocimiento experimental obtenido en un organismo a menudo es aplicable a otros organismos, particularmente si estos comparten los genes relevantes porque los heredaron de algún ancestro en común.

Los experimentos presentados en este trabajo fueron llevados a cabo haciendo uso de la infraestructura de ClusterUY [Nesmachnow and Iturriaga, 2019], una pla-

taforma de computación de alto desempeño que posee la capacidad de gestionar en forma coordinada múltiples recursos de cómputo, y que es utilizada por científicos e investigadores de todo el país.

1.1. Predicción de función de genes

Los desarrollos tencológicos de los últimos años han provocado que los métodos experimentales para la determinación de las funciones de los genes no puedan acompasar el ritmo acelerado al que se producen datos genómicos. La brecha entre la cantidad de genes secuenciados y la cantidad de genes con función conocida no ha dejado de crecer en los últimos años. En la Figura 1.1 se muestra el número de secuencias de proteínas depositadas comparado con el número de proteínas con al menos una función conocida alojadas en la base de *UniProt* [Consortium, 2020], la mayor base de datos de proteínas secuenciadas. Por este motivo, los métodos computacionales, que con frecuencia incluyen técnicas de machine learning, han cobrado gran relevancia, pues resultan útiles para predecir las funciones de un gen, guiar experimentos en laboratorios para determinar fehacientemente sus funciones y ahorrar así tiempo y recursos.

Aunque los métodos basados en aprendizaje automático se han considerado como un "caja negra" en el pasado, pueden ser más precisos que los métodos estadísticos más simples. En los últimos años, el aprendizaje automático se ha desarrollado rápidamente y ha alcanzado un nivel sorprendente de rendimiento en diversas áreas, incluida la predicción de función de genes.

Dado que existen sólo cuatro nucleótidos (*Adenina*, *Citosina*, *Guanina* y *Timina*) a partir de los cuales los genes de ADN pueden formarse, es conveniente representar a los genes como un string con un alfabeto de cuatro letras; $\Sigma = \{A, C, G, T\}$. Entre los algoritmos más utilizados para la predicción de funciones de genes, que permiten utilizar esta estructura de cadena, destacan los métodos basados en la *homología de secuencia*, ver Sección A.1. La homología hace referencia a la situación en la que las secuencias de dos o más proteínas o ácidos nucleicos son similares entre sí. La homología de secuencia entre dos segmentos de ADN se debe generalmente a que ambas comparten un origen evolutivo común y puede darse principalmente por tres procesos:

- eventos de especiación (*secuencias ortólogas*): es el proceso evolutivo en el cual

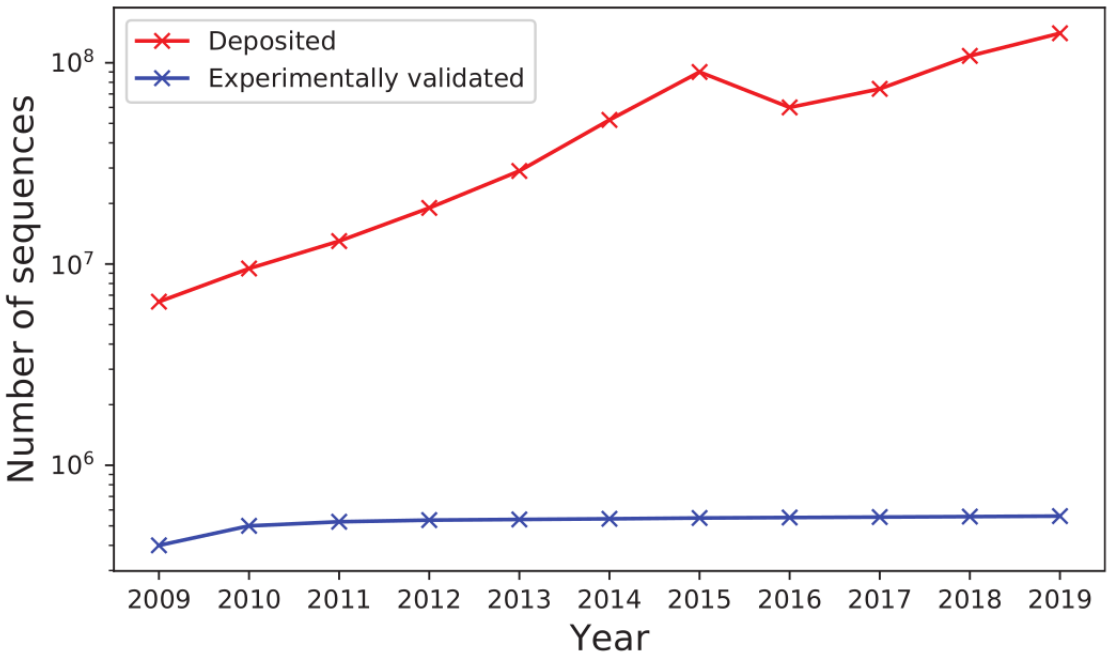


Figura 1.1: Cantidad de proteínas depositadas en UniProtKB durante la última década. En rojo, el total de secuencias depositadas y en azul la secuencias con alguna función asociada experimentalmente. La caída observada entre 2015 y 2016 se debe a los procedimientos implementados por los curadores para identificar y eliminar proteomas redundantes. Esta imagen fue extraída de [Bonetta and Valentino, 2019].

una población comienza a divergir genéticamente respecto a otra y deviene en una nueva especie. El mismo segmento de ADN acumula diferencias en entre una y otra poblacion .

- eventos de duplicación (*secuencias parálogas*): es el proceso en el que un gen de un organismo se duplica en un mismo genoma y una de las copias comienza a acumular mutaciones respecto a la otra.

- transferencia horizontal de secuencias (*secuencias xenólogas*): es el movimiento de material genético entre organismos unicelulares o pluricelulares.

Una similitud de secuencia significativa es una fuerte evidencia de que ambas secuencias de ADN derivan de un ancestro en común. Esto a su vez puede utilizarse para inferir nuevas funciones de un gen, ya que si un gen posee una cierta función conocida y este gen tiene una similitud significativa con otro (incluso perteneciente a otro organismo), entonces puede inferirse que este último posee las funciones del

primero.

Entre los algoritmos que se utilizan para determinar homología de secuencia podemos destacar a BLAST [Altschul et al., 1990] (Basic Local Alignment Search Tool), un algoritmo que cuantifica el parecido entre dos secuencias y que se ha convertido en el estandar para determinar si dos secuencias son homólogas o no.

No obstante, la evidencia resultante de métodos computacionales no constituye por si misma una prueba fehaciente para indicar que un gen tiene una determinada función biológica. Sin embargo, estos resultados son de gran ayuda para establecer y guiar experimentos que puedan, eventualmente, validar estas predicciones. Esto es de gran ayuda puesto que evita malgastar tanto tiempo como recursos, humanos o materiales, en la búsqueda de nuevas funciones de los genes.

La predicción de las funciones de los genes es una ardua tarea, ya que como veremos más adelante involucra predecir funciones sobre una estructura organizada de forma jerárquica. Estas estructuras jerárquicas son las *ontologías de términos GO*, que se describen a continuación en la Sección 1.2, y tienen varios miles de posibles nodos, por lo que clasificar un único gen supone distinguir entre miles de etiquetas posibles. Más aún, cada gen puede estar asignado o no a varios términos GO distintos y estos términos deben tener cierta coherencia con la estructura jerárquica de la ontología; si un determinado gen tiene asignada cierta función biológica a través de un término GO de la ontología, entonces este gen debe estar asociado también con cada término GO ancestro del primero. Otro inconveniente es la ausencia de instancias negativas para la clasificación: se puede determinar experimentalmente que un gen tiene asociada un cierta función, pero no se puede determinar tan fácilmente que no la tiene, esto se refleja en la escasa cantidad de anotaciones negativas en el archivo de anotaciones, ver Tabla 2.2. Este problema suele solventarse con una *Siblings Policy* [Vateekul et al., 2014, Feng et al., 2017, Feng et al., 2018], ver Sección 2.6. Además, los nodos en los niveles más bajos de la jerarquía usualmente tienen muy pocas instancias positivas y por lo tanto los datos para clasificar sobre estos nodos terminan siendo muy sesgados.

1.2. Gene Ontology

Para describir la función de los genes y sus productos génicos existen distintas herramientas como *Enzyme Commission* [Enz, 1993], *Functional Catalogue* (Fun-

Cat) [Ruepp et al., 2004] y *Kyoto Encyclopedia of Genes and Genomes* (KEGG) [Kanehisa et al., 2004]. Sin embargo, la termilogía más ampliamente utilizada es la provista por Gene Ontology (GO) [Ashburner et al., 2000, Consortium, 2018]. GO es la más amplia e importante iniciativa bioinformática para unificar la representación de las funciones de los genes y productos génicos en todas las especies. Es ampliamente utilizado por la comunidad científica y ha sido citado en miles de publicaciones distintas.

La misión del Consorcio GO es desarrollar un modelo computacional actualizado y completo de los sistemas biológicos, desde el nivel molecular hasta el nivel organismo, a través de multiples especies. Más específicamente, el proyecto apunta a:

1. Mantener y desarrollar un vocabulario de todas las posibles funciones y atributos de los productos de la expresión de los genes. Este vocabulario está formado por los términos GO;

2. Anotar genes y productos génicos; esto es, asociarlos con los términos GO, y asimilar y difundir datos de anotaciones;

3. Proporcionar herramientas para facilitar el acceso a todos los datos generados por el proyecto. Esto facilita la interpretación de muchos resultados experimentales, por ejemplo mediante el análisis de enriquecimiento funcional.

Gene Ontology es parte de un esfuerzo de clasificación más grande: las ontologías biomédicas abiertas, siendo uno de los miembros candidatos iniciales de la Fundación OBO [Smith et al., 2007].

1.2.1. Ontologías de términos GO

Una ontología es una representación formal de un cuerpo de conocimiento dentro de un dominio dado. Las ontologías generalmente consisten en un conjunto de clases (o términos o conceptos) con relaciones que operan entre ellas.

Gene Ontology intenta representar todo nuestro conocimiento sobre los genes y productos génicos mediante tres ontologías, cada una de las cuales captura diferentes aspectos del rol biológico de los genes:

- *Biological Process*: esta ontología describe los procesos biológicos en los que participan los productos génicos,

- *Cellular Component*: describe los componentes o estructuras anatómicas dentro de la célula en las que un producto génico desarrolla sus funciones.

- *Molecular Function*: describe a nivel molecular el mecanismo de acción a través del cual un producto génico lleva a cabo su función.

Cada término GO pertenece sólo a una de las tres ontologías. Por lo tanto, las anotaciones GO capturan declaraciones de los genes sobre qué procesos biológicos ayudan a llevar a cabo, en qué parte de la célula se expresan y cómo opera un gen a nivel molecular. Por ejemplo, el producto génico "cytochrome c" puede ser descripto por la función molecular *oxidoreductase activity* (GO:0016491), el proceso biológico *oxidative phosphorylation* (GO:0006119) y la componente celular *mitochondrial matrix* (GO:0031980). Los códigos GO:0016491, GO:0006119 y GO:0031980 son identificadores de cada término GO y en general se componen del prefijo "GO:" seguido de 7 dígitos decimales. En la Sección 2.2.1 se describe la información que el Consorcio GO provee para cada término GO.

Cada una de las tres ontologías tiene estructura de un *digrafo acíclico* (DAG por sus siglas en inglés) donde cada término GO es un nodo, y una arista de un término GO a otro significa que el primero es más específico que el segundo. Esta relación jerárquica entre términos GO se refleja en la ontología como ausencia de ciclos, puesto que en un ciclo cada uno de los términos sería más general que si mismo, lo cual es absurdo.

Por ejemplo, como puede verse en la Figura 1.2, el término GO *hexose biosynthetic process* (GO:0019319) tiene dos padres, *hexose metabolic process* (GO:0019318) y *monosaccharide biosynthetic process* (GO:0046364). Esto refleja el hecho de que el término GO:0019319 es un subtipo de los términos GO:0019318 y GO:0046364. Esta relación jerárquica entre términos GO es más gráfica en la ontología Cellular Component, ya que al decir que un término GO es más específico que otro equivale a decir que el primero es físicamente un componente del segundo. Una lista completa de todas las relaciones jerárquicas entre términos GO puede encontrarse en la web de Gene Ontology, en la Sección 2.2.1 se discute más a profundidad el contenido de estos datos. En este trabajo empleamos la versión con fecha de publicación 10/11/2018.

En cada ontología existe un único término GO que no tiene aristas hacia otros términos, este término constituye el término GO más general posible de la ontología

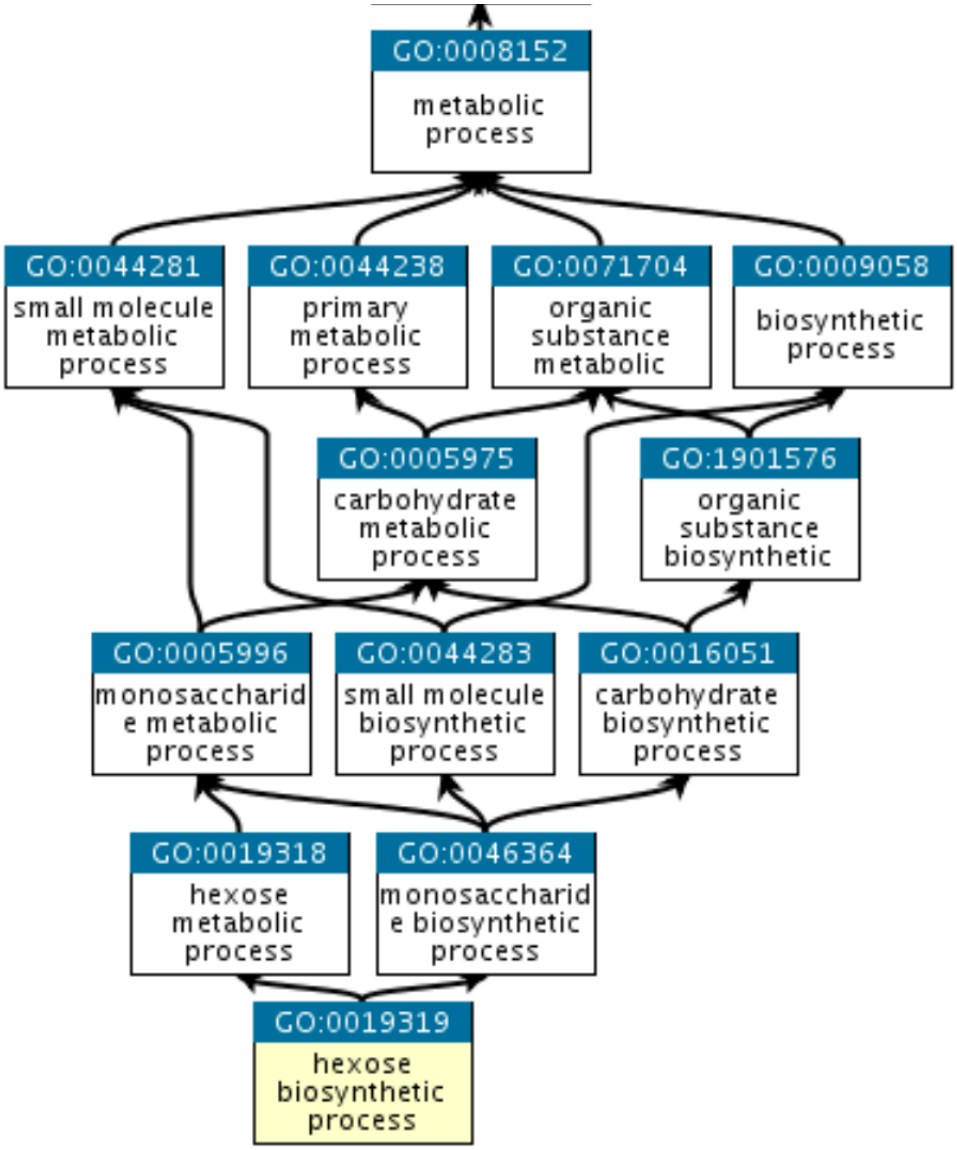


Figura 1.2: Ejemplo de la estructura de *digrafo acíclico* (DAG) de la ontología de término GO, en este caso un subgrafo de la ontología de Biological Process. En esta figura se muestra, con algunos término GO, la estructura jerárquica de la ontolgía, donde existen términos más específicos que otros y podemos encontrar términos con más de un término padre.

y lo denominamos como la *raíz*. Las raíces de Biological Process, Cellular Component y Molecular Function son GO:0008150, GO:0005575 y GO:0003674, respectivamente.

Con los datos obtenidos de la web de Gene Ontology generamos los grafos de las tres ontologías. En la Figura 1.3 tenemos representados estos grafos, siendo el nodo más central de cada figura el término GO más general de cada ontología. En la

Tabla 1.1 aparecen la cantidad de nodos y aristas de cada ontología, con la versión de las ontogías empleadas en este trabajo.

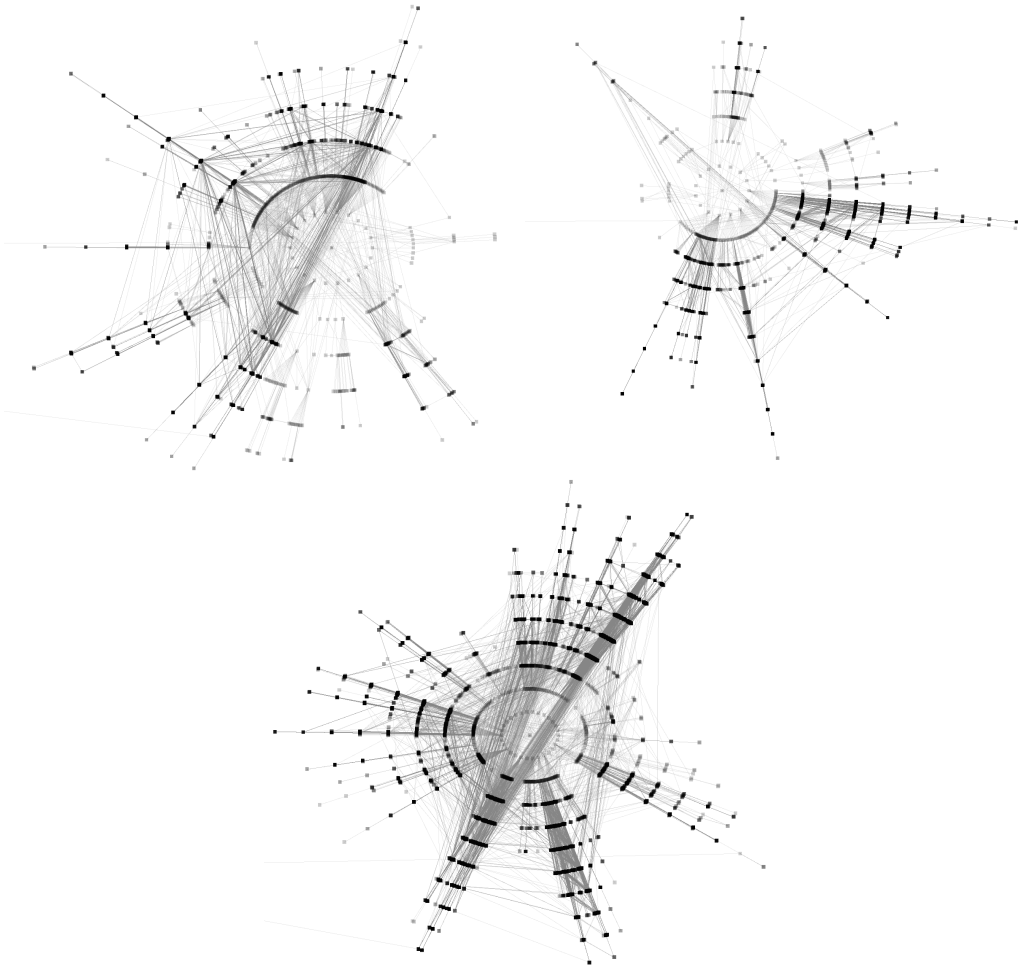


Figura 1.3: En la esquina superior izquierda se muestra el grafo de la ontología *Cellular Component*, a la derecha el de *Molecular Function* y en la parte inferior el de *Biological Process*. En cada una de estas figuras los nodos representan los términos GO y las aristas las relaciones jerárquicas entre distintos términos. Cada grafo está representado de forma tal que los nodos más centrales son los términos GO más generales.

1.2.2. Anotaciones de términos GO

La anotación de un gen o producto génico con un término GO es una declaración de su función biológica, creando asociaciones entre un gen o producto génico con un término GO. Los modelos desarrollados para la predicción de función de genes

	Vértices	Aristas
Cellular Component	4206	6080
Biological Process	29681	57459
Molecular Function	11118	13620

Tabla 1.1: Cantidad de nodos y aristas del grafo de cada ontología empleados en este trabajo.

buscan asociar a cada gen uno o varios términos GO. Gene Ontology provee, para varios organismos, una lista de anotaciones de términos GO, la cual puede descargarse desde su web, ver Sección 2.2.2. En este trabajo utilizamos la versión con fecha de publicación 08/10/2018.

Sin embargo, nuestro conocimiento actual sobre la taxonomía funcional de los productos génicos es todavía inmaduro. Por lo tanto, tanto la jerarquía de términos GO como las anotaciones se actualizan periódicamente con nuevos conocimientos y se archivan como referencia. Las anotaciones de GO recopiladas todavía son bastante incompletas, desbalanceadas y bastante superficiales [Rhee et al., 2008, Thomas et al., 2012, Dessimoz and Škunca, 2017]. Por ejemplo, diferentes especies tienen diferentes distribuciones de anotaciones GO; *zebrafish* es un organismo muy estudiado en términos de biología del desarrollo y embriogénesis, mientras que *rat* se constituye como el modelo estándar de toxicología, [Dessimoz and Škunca, 2017].

1.2.3. Evidence Codes

Cada anotación incluye un *evidence code* para indicar cuál es el tipo de evidencia que apoya esa anotación en particular. El Consorcio GO ha adoptado un total de 26 evidence codes agrupados en 6 categorías distintas:

1. *Experimental Evidence*: Inferred from Experiment (EXP), Inferred from Direct Assay (IDA), Inferred from Physical Interaction (IPI), Inferred from Mutant Phenotype (IMP), Inferred from Genetic Interaction (IGI), Inferred from Expression Pattern (IEP), Inferred from High Throughput Experiment (HTP), Inferred from High Throughput Direct Assay (HDA), Inferred from High Throughput Mutant Phenotype (HMP), Inferred from High Throughput Genetic Interaction (HGI) y Inferred from High Throughput Expression Pattern (HEP);

2. *Phylogenetic Evidence*: Inferred from Biological aspect of Ancestor (IBA), Inferred from Biological aspect of Descendant (IBD), Inferred from Key Residues (IKR) y Inferred from Rapid Divergence (IRD);

3. *Computational Evidence*: Inferred from Sequence or structural Similarity (ISS), Inferred from Sequence Orthology (ISO), Inferred from Sequence Alignment (ISA), Inferred from Sequence Model (ISM), Inferred from Genomic Context (IGC) y Inferred from Reviewed Computational Analysis (RCA);

4. *Author Statements*: Traceable Author Statement (TAS) y Non-traceable Author Statement (NAS);

5. *Curatorial Statements*: Inferred by Curator (IC) y No biological Data available (ND);

6. *Automatically Generated Annotations*: Inferred from Electronic Annotation (IEA).

Para más información sobre el significado de cada evidence code visitar la web de Gene Ontology.

A excepción de IEA, todo los evidence codes son revisados manualmente, aunque el método en sí suele estar sujeto a varias evaluaciones de calidad, por este motivo no hemos considerado para este trabajo las anotaciones con evidence code IEA. En la Figura 1.4 podemos ver la cantidad de anotaciones por evidence code a lo largo del tiempo. En la figura se puede observar que las anotaciones con evidence code IEA representan alrededor del 25 % del total de anotaciones. Este porcentaje varía según el organismo y la versión de las anotaciones que se considere.

Mientras que en la Figura 1.5 tenemos la cantidad de anotaciones por evidence code para los 5 organismos considerados en este trabajo.

1.3. Clasificadores Jerárquicos Multiclase

La predicción de las funciones de los genes se puede abordar como un problema de clasificación jerárquica multiclase (o por sus siglas en inglés: HMC, Hierarchical Multilabel Classification). Esta declaración tiene varias partes. En primer lugar, el término "Clasificación" obedece al problema de identificar, dentro un conjunto de categorías dadas, a cuál pertenece una nueva observación. En nuestro caso de estudio

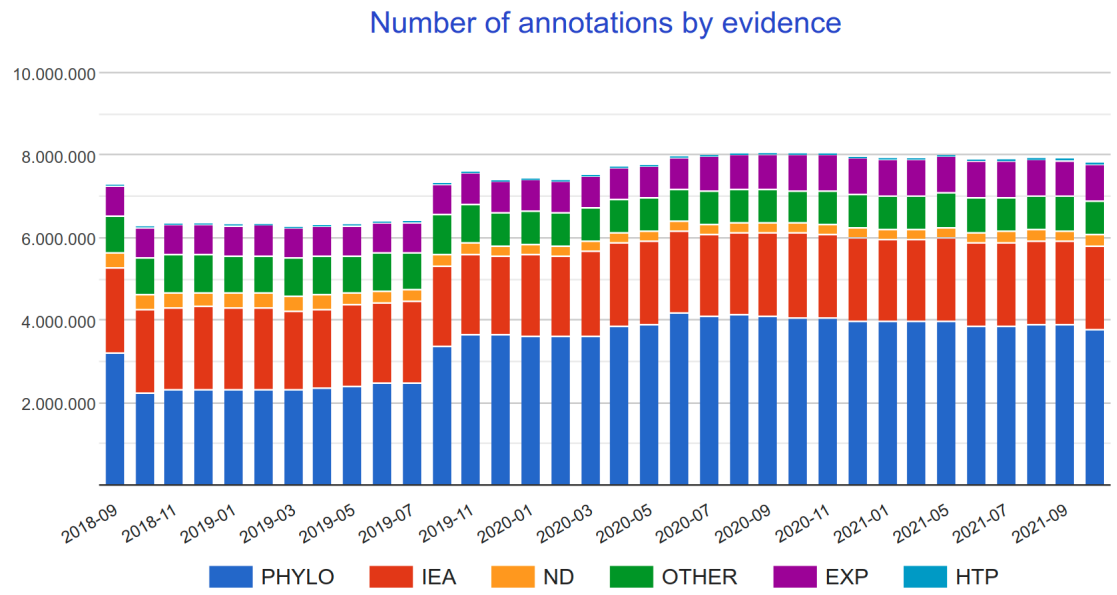


Figura 1.4: Número de anotaciones por evidence code a lo largo del tiempo de todos los organismos presentes en la base de datos de Gene Ontology. Este gráfico fue extraído de la sección de estadísticas de Gene Ontology.

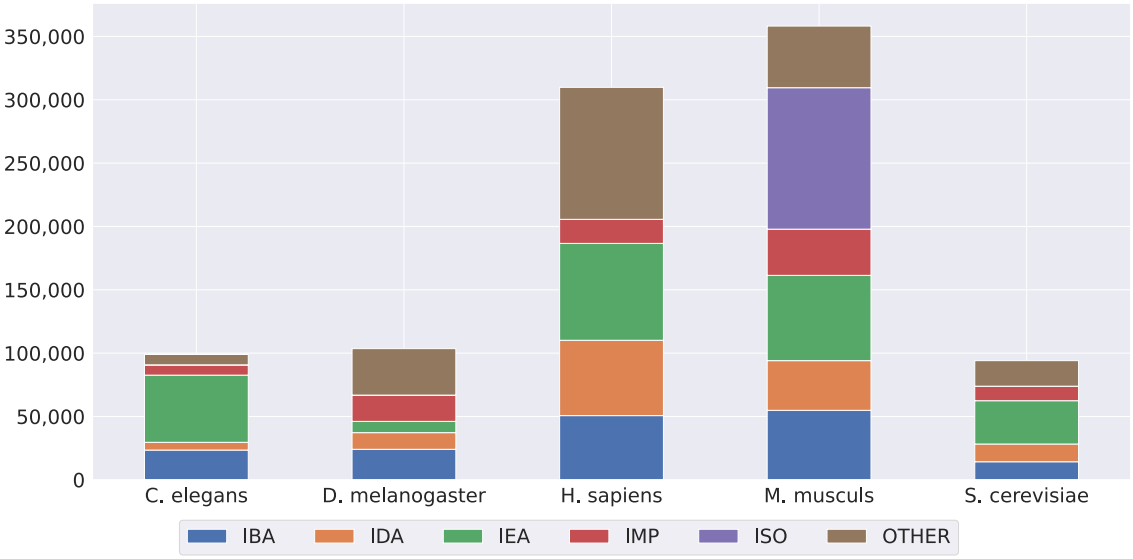


Figura 1.5: Número de anotaciones por evidence code para cada uno de los 5 organismos estudiados. Las anotaciones que se consideraron pra este gráfico fueron las empledas durante este trabajo, ver Sección 2.1.

el conjunto de categorías corresponde al conjunto de los términos GO, que además tiene una estructura de DAG. El término "Jerárquico" hace referencia a que el

conjunto de categorías o términos GO están organizadas jerárquicamente, esto obliga a que si un gen fue asociado con algún término GO, entonces está automáticamente asociado a todos sus términos GO ancestros. Por último, se trata de un problema de multietiqueta ya que un gen puede tener asociados uno o varios términos GO, en la misma o en distintas ontologías.

Las aplicaciones de HMC son diversas e incluyen categorización de textos [Rousu et al., 2006], predicción de funciones proteínas [Silla Jr. and Freitas, 2009], clasificación de géneros musicales [Silla and Freitas, 2009], clasificación de fonemas [Dekel et al., 2005] y, como veremos, predicción de funciones de genes.

De acuerdo con [Silla and Freitas, 2011], existen tres criterios principales para clasificar a los algoritmos que intentan resolver problemas de HMC:

- el tipo de estructura utilizada,
- la profundidad en la jerarquía con la que se realiza la clasificación,
- y cómo es explorada la estructura jerárquica.

El primer criterio considera si la estructura sobre la que se desea realizar la predicción es un árbol o un DAG. La principal diferencia entre un árbol y un DAG es que en un DAG los nodos pueden tener más de un nodo ancestro y que en un árbol cada nodo del grafo tiene un único camino hacia la raíz, ver Figura 1.6. El segundo criterio tiene en cuenta si la clasificación debe realizarze siempre sobre las hojas de la estructura, caso en el que se habla de *mandatory leaf-node prediction* (MLNP), o no, en cuyo caso se habla de un *non mandatory leaf-node prediction* (NMLNP). El tercer criterio separa los clasificadores según empleen un clasificador por cada nodo de la estructura *top-down* (o locales), o un único clasificador para toda la jerarquía *big-bang* (o globales).

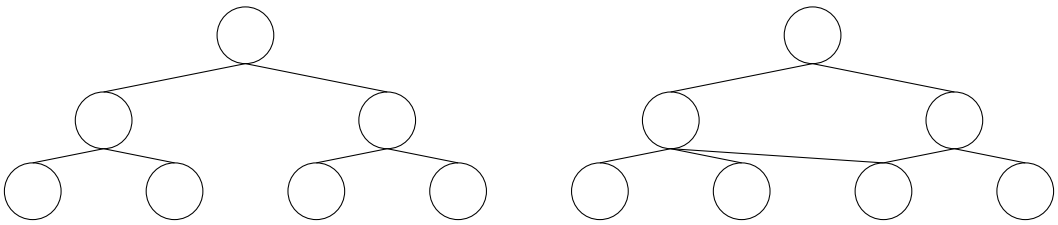


Figura 1.6: Un ejemplo de un árbol (izquierda) y de un DAG (derecha)

En el caso de estudio de este trabajo nos enfocaremos en predecir sobre un DAG

(sobre el digrafo de cada ontología), con un modelo local y NMLNP. Esto será detallado en la Sección 2.6.

Dado que el objetivo principal de este trabajo, ver Sección 1.5, consiste en explorar la posición relativa de un gen dentro del genoma como variable predictora de sus funciones, decidimos implementar un modelo HMC local ya que entendiamos que nos permitía introducir de una manera más natural y directa LEA (*Local Enrichment Analysis*), ver Sección 2.4, como input del sistema. A diferencia de los métodos más clásicos, ver Sección A, que en parte basan sus predicciones en interacciones en un grafo, los métodos basados en machine learning son más libres en cuanto al input que el modelo puede aceptar.

1.4. CAFA

El *"Critical Assessment of Protein Function Annotation"* (CAFA) es un desafío o competencia cuyo fin consiste en evaluar a gran escala los más recientes métodos computacionales diseñados para predecir funciones de proteínas o genes.

Los organizadores de CAFA proporcionan una gran cantidad de secuencias de proteínas. Luego, los participantes desarrollan modelos de predicción de funciones y envían sus predicciones. La predicción de las funciones debe ser llevada a cabo sobre términos de *Gene Ontology* (GO), términos de *Human Phenotype Ontology* (HPO) o (nuevo a partir de la cuarta edición) términos de *Disorder Ontology* (DO).

Después de un período de tiempo, las recepciones de nuevas predicciones se interrumpen y la competencia entra en la etapa de evaluación. Durante ocho meses se obtienen de la literatura nuevas anotaciones experimentales y luego de esa etapa de espera, las proteínas recientemente anotadas se utilizan como referencia para evaluar el desempeño de los modelos predictivos presentados.

CAFA es un esfuerzo de toda la comunidad cuyo objetivo es ayudar a comprender el estado del arte en la predicción de funciones de genes y proteínas e impulsar el campo hacia adelante. La primera edición de CAFA (CAFA1) se llevó a cabo entre 2010 y 2011 e incluyó 23 grupos de 14 países, los cuales propusieron 54 métodos computacionales de predicción que fueron evaluados en su desempeño. Este fue el primer esfuerzo a gran escala para proporcionar información sobre las fortalezas y debilidades de los modelos de predicción de funciones en la comunidad bioinformática. CAFA2 se realizó entre 2013 y 2014, con más del doble del número de grupos

(56) y métodos de participación (126), mientras que CAFA3 se efectuó entre 2016 y 2017 con 68 grupos y 144 métodos. Entre 2017 y 2018, para brindar a los equipos participantes de CAFA3 otra oportunidad para desarrollar o modificar los modelos de predicción se llevo a cabo CAFA-π.

Para obtener más información y resultados sobre estas primeras ediciones de CAFA pueden consultarse [Radivojac et al., 2013, Jiang et al., 2016, Zhou et al., 2019].

1.5. Objetivos de la tesis

El presente trabajo tiene como principal objetivo hacer aportes a la predicción de funciones de genes, implementando modelos de clasificación jerárquica multiclase entrenados exclusívamente con datos relativos a la posición de los genes en el genoma. Para ello utilizamos cinco organismos modelo (ver Tabla 1.2) ampliamente utilizados en la literatura: *Saccharomyces cerevisiae, Caenorhabditis elegans, Drosophila melanogaster, Mus musculus* y *Homo sapiens*.

En cada uno de estos organismos llevamos a cabo un análisis de enriquecimiento funcional local a escala genómica, a través del cual obtuvimos las variables predictoras con las que entrenamos nuestros modelos.

	Genes
S. cerevisiae	5892
C. elegans	7356
D. melanogaster	11122
M. musculs	20809
H. sapiens	17276

Tabla 1.2: Cantidad de genes que codifican proteínas en los organismos estudiados.

Los objetivos específicos son los siguientes:

1. Llevar a cabo una revisión bibliográfica de las principales técnicas de aprendizaje automático empleadas en la predicción de funciones de genes (ver Anexo A).

2. Llevar a cabo el análisis de enriquecimiento funcional local a escala genómica en las cinco especies consideradas (ver Secciones 2.4 y 3.1.1).

3. Implementar y entrenar un modelo de clasificación jerárquica multiclase que prediga nuevas funciones de genes para cada posible par formado por las cinco especies y las tres ontologías (ver Secciones 2.6 y 3.1.2).

4. Implementear métricas para la evaluación de estos modelos (ver Sección 2.7).

5. Hacer públicos los resultados de este trabajo para cada uno de los organismos considerados (ver gfpml-results y gfpml).

6. Evaluar el desempeño de nuestros modelos con el desempeño más básicos utilizados para la predicción de funciones de genes (ver Sección 3.2.2).

El trabajo aquí reunido se enmarcó en la tesis doctoral de uno de los orientadores de esta tesis. Mi principal aporte fue la implementación y ejecución computacional de todos los análisis aquí presentados, que a su vez forman parte de un artículo científico del cual soy primer autor y que recientemente ha sido aceptado para su publicación en la revista Nature Scientific Reports, ver Anexo 3 C.

Capítulo 2

Métodos

2.1. Anotaciones jerárquicas

Los archivos de anotaciones que pueden encontrarse en la web de Gene Ontology están planteados para ser lo más completos y a la vez lo menos redundantes posibles. Esto quiere decir que si un gen está anotado con cierto término GO, entonces en el archivo no se encontrarán anotaciones del mismo gen con términos GO menos específicos. Por este motivo debimos propagar las anotaciones hacia la raíz de la ontología, de manera tal que si un gen estaba asociado a un determinado término GO, lo asociamos automáticamente a todos los términos GO ancestros a este, incluyendo la raíz de la ontología.

En la Tabla 2.1 se muestra, para cada par de ontología y organismo, la cantidad de términos GO asociados con al menos un gen, antes y después da haber propagado las anotaciones. Como puede observarse, luego de propagar las anotaciones, se obtienen alrededor de un 30 % más de términos GO anotados.

	Biological Process	Cellular Component	Molecular Function
S. cerevisiae	2899 / 5074	780 / 1035	1791 / 2323
C. elegans	3254 / 5661	857 / 1110	1663 / 2226
D. melanogaster	4811 / 7416	1022 / 1277	2083 / 2599
M. musculs	11969 / 15318	1702 / 1953	3800 / 4269
H. sapiens	10152 / 13816	1556 / 1818	3737 / 4244

Tabla 2.1: Cantidad de términos GO anotados por ontología y organismo, con y sin la propagación de las anotaciones.

La Tabla 2.2 muestra la cantidad de anotaciones para cada organismo, antes y después de la propagación de anotaciones. Se obtienen alrededor de seis veces más anotaciones de esta manera. Así mismo, entre paréntesis se muestra la cantidad de anotaciones negativas, que como puede observarse, son escasas con respecto al total.

	Anotaciones	Anotaciones Jerárquicas
S. cerevisiae	43368 (124)	326902
C. elegans	40789 (75)	308680
D. melanogaster	75075 (446)	492157
M. musculs	229951 (679)	1368846
H. sapiens	195948 (950)	1231432

Tabla 2.2: Cantidad de anotaciones y anotaciones antes y después de la propagación de anotaciones en cada organismo. Entre paréntesis se muestran la cantidad de anotaciones negativas de cada organismo.

2.2. Datasets

En el repositorio gfpml-datasets se encuentran disponibles los conjuntos de datos empleados en este trabajo; en las siguientes secciones describimos estos datasets, los cuales son actualizados regularmente.

2.2.1. Ontologías

El archivo `go-basic.obo` provisto por Gene Ontology contiene las relaciones jerárquicas entre los términos GO a partir del cual se puede reconstruir los grafos de cada ontología. Es un archivo en texto plano que contiene la siguiente información:

- `id`: el identificador del término GO, un identificador único de siete dígitos con el prefijo "GO:", como por ejemplo: GO:0005739, GO:1904659, o GO:0016597.
- `name`: el nombre del término legible por humanos, como por ejemplo: *mitochondrion*, *mitochondrion*, o *amino acid binding*.
- `namespace`: denota a cuál de las tres ontologías (*Biological Process*, *Cellular Component* o *Molecular Function*) pertenece el término.

- `def`: una descripción textual de lo que representa el término, más referencias a la fuente de la información.
- `is_a`: esta etiqueta describe cómo se relaciona el término con otros términos de la ontología. Todos los términos, a excepción del términos raíz de cada ontología, tienen una relación de subclase con otro término.

Este archivo contiene otras etiquetas opcionales de las que podemos destacar `is_obsolete`, la cual indica que el término ha quedado obsoleto y no debe utilizarse. Un término GO puede quedar obsoleto por diversos motivos. En estos casos, aún persiste en el archivo de la ontología `go-basic.obo`, pero el término se etiqueta como obsoleto y se eliminan todas las relaciones con otros términos. Se agrega un comentario al término que detalla el motivo de la obsolescencia y se sugieren términos de reemplazo, cuando es posible.

La versión del archivo `go-basic.obo` empleado en este trabajo tiene fecha de publicación de 10/11/2018 y puede encontrarse en el siguiente enlace.

2.2.2. Anotaciones

Los archivos de anotaciones de genes en formato gaf que pueden encontrarse en la web de Gene Ontology. Estos archivos contienen las anotaciones de pares gen–término GO, estas anotaciones, como se describe en la Sección 2.1, están planteadas de forma de ser lo menos reduntante posibles y debemos propagarlas hasta la raíz de la correspondiente ontología.

Además este archivo provee el *evidence code*, ver Sección 1.2.3, el cual es un código que indica cómo la anotación de un gen o producto génico a un término GO fue admitida. Existen diversos evidence codes, cada uno describe una forma distinta como son anotados los término GO, ver la siguiente web para más información.

Lo archivos de anotaciones para los 5 organismos modelo estudiados en este trabajo puede encontrarse en el siguiente enlace.

2.2.3. Genomas

La información necesaria para definir la posición relativa de los genes dentro del genoma para los cinco organismos modelo considerados se obtuvo a partir de archivos descargados de Ensembl [Howe et al., 2020] en formato gtf. Estos archivos contienen una secuencia por línea, y para cada secuencia contienen además::

- `seqname`: nombre del cromosoma al cual pertenece la secuencia.
- `source`: nombre de la fuente o del programa que genero los datos.
- `feature`: nombre del tipo de secuencia, por ejemplo gen, variante.
- `start` y `end`: posición de inicio y de final de la secuencia.
- `strand`: valores posibles + y −, indican si la secuencia se codifica en el mismo sentido o en el sentido opuesto al que se codifica el genoma.

Para modelar los genomas también se tuvo en cuenta la eventual presencia de centrómeros en cada cromosoma. El centrómero es una estructura que puede interrumpir físicamente la transcripción y separar funcionalmente en dos unidades independientes al cromosoma, por lo que, cuando correspondía, consideramos a cada brazo cromosómico por separado.

Lo archivos de los genomas utilizados en este trabajo puede encontrarse en el siguiente enlace, también puede encontrarse el archivo que indica la posición de los centrómeros aquí.

2.3. Modelado del genoma

Como en [Pazos Obregón et al., 2018], modelamos cada genoma como una colección de secuencias de genes, en la que cada secuencia se corresponde con un cromosoma. En cada secuencia, los genes codificadores de proteínas se ordenan uno al lado del otro y sin superposición, según la posición del sitio de inicio de su transcripción en el cromosoma (sin considerar el sentido en el cual se codifica cada gen ni las regiones intergénicas), ver Figura 2.1. Por este motivo también podemos indentificar cada gen con un par $(\mathsf{seq}, \mathsf{pos})$, donde seq es el cromosoma al cual pertenece dicho gen y pos es la posición que ocupa dentro de la secuencia de este cromosoma.

En la Tabla 1.2 tenemos la cantidad de genes que codifican proteínas para los 5 organismos modelo estudiados.

2.4. Análisis de Enriquecimiento Local

Como comentamos previamente, el objetivo de este trabajo consiste en implementar una serie de modelos que predicen funciones de genes, entrenados exclusiva-

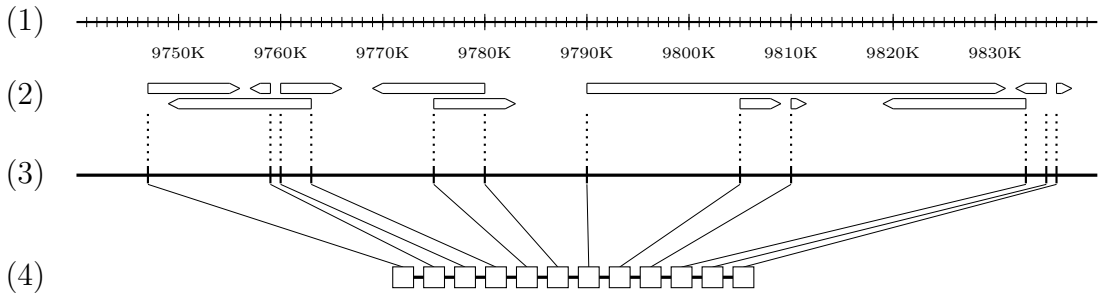


Figura 2.1: En (1) se muestra un porción de un genoma representado como una cadena de nucleótidos; en (2) se representan varios genes como bandas en las que se puede observar el solapamiento de los mismos, la dirección de su transcripción, su extensión y el espacio intergénico; en (3) se representan solamente las posiciones de inicio de transcripción de cada gen a lo largo de una línea que representa el genoma; y en (4) se representa cada gen como una unidad, ordenados según su posición de inicio, ignorando su largo, el sentido de su transcripción y el espacio intergénico. Esta es la represenatción usada en este trabajo.

mente con variables derivadas de la posición de los mismos dentro del genoma. Las variables utilizadas se obtuvieron mediante el *Análisis de Enriquecimiento Funcional Local*, que se describe a continuación.

Las técnicas de análisis de enriquecimiento funcional son habituales para caracterizar una lista de genes. Tienen como objetivo determinar si una lista de genes dada tiene alguna función biológica sobrerepresentada, es decir, si la lista contiene más genes con esta función de lo esperable por azar. En [Obregón, 2020] se propone el *Análisis de Enriquecimiento Funcional Local* (LEA, por su sigla en inglés). LEA propone analizar el enriquecimiento funcional en el entorno de cada gen, para todos los genes de un genoma, utilizando un entorno de tamaño variable.

Dada una lista de genes asociados a cierta función biológica y un valor del tamaño de ventana, el cual es un entero no negativo ws, para un determinado gen k el valor de LEA se calcula con la siguiente fórmula:

$$E_{\mathsf{ws},k} = \frac{b/n}{B/N},$$

donde b es el número de genes de la lista que se ubican dentro de la ventana (la ventana consiste de los genes del cromosoma que distan de k en a lo sumo ws unidades), n es el número de genes de la ventana ($n = 2\mathsf{ws} + 1$ si la ventana

no interseca los bordes del cromosoma), B el número de genes de la lista en el cromosoma o brazo cromoómico en el que se ubica el gen en cuestión y N el número total de genes del cromosoma. Así definido, el valor de LEA existe salvo en el caso en el cual la lista de genes no tiene representantes en el cromosoma; en este caso definimos $E_{\mathsf{ws},k} = 0$. En la Figura 2.2 se ejemplifica cómo es el cálculo de LEA en un cromosoma.

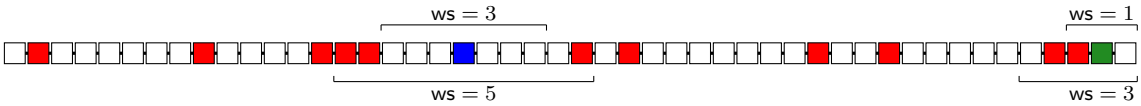


Figura 2.2: Ejemplo de cálculo de LEA. Cada uno de los cuadrados representa un gen de un cromosoma: los rojos son los genes de la función biológica que estamos evaluando y en azul y verde se indican dos genes para los cuales se ejemplifica el cálculo de LEA. El tamaño de la ventana es la cantidad de genes que se considera hacia un lado y otro del gen en cuestión. En el caso del gen azul, si el tamaño de la ventana es $\mathsf{ws} = 3$, entonces $E = \frac{0/7}{11/48} = 0$, y si $\mathsf{ws} = 5$, entonces $E = \frac{3/11}{11/48} \approx 1,19$. Mientras que para el gen verde, si el tamaño de la ventana es $\mathsf{ws} = 1$, entonces $E = \frac{1/3}{11/48} \approx 1{,}45$, y si $\mathsf{ws} = 3$, entonces $E = \frac{2/5}{11/48} \approx 1,74$.

En nuestro caso nos interesa evaluar la posición relativa de un gen como variable para predecir sus funciones y por este motivo las listas de genes que consideramos para llevar a cabo LEA son las listas de genes anotadas con cada término GO. Esto nos permitirá distinguir en cuáles porciones del genoma se encuentra enriquecida determinada función biológica. Si en una región del genoma los genes tienen un gran valor de LEA para un cierto término GO, entonces en esa región el término GO se encuentra sobrerepresentado. Este abordaje tiene un antecedente en la bibliografía [Tiirikka et al., 2014], en el que los autores llevan a cabo un procedimiento de este tipo con el objetivo de ubicar clusters de términos GO en el genoma de siete organismos modelo.

Para un cierto conjunto de términos GO, el cual discutiremos en la Sección 2.6, recorrimos cada cromosoma gen a gen, determinando, para una serie de distintos tamaños de ventana, el valor de LEA. Esto resulta, para cada término GO, en una matriz donde las filas se corresponden a los genes del genoma y las columnas a los tamaños de ventanas analizados. En la Figura 2.3 se ilustra el valor de LEA a lo largo del genoma para distintos tamaños de ventanas y distintas funciones biológicas. En

esta figura puede apreciarse cómo los genes con una misma función biológica tienden a concentrarse en determinadas secciones del cromosoma.

2.5. Notación y definiciones básicas

Sea $X = \{x_1, \ldots, x_n\}$ el conjunto de n genes de un organismo dado (recordar que en este trabajo examinanos cinco organismos modelo, ver Sección 1.5) y sea $G = \{g_1, \ldots, g_m\}$ el conjunto de términos GO de una ontología dada (la cual puede ser *Biological Process*, *Cellular Component* o *Molecular Function*, ver Sección 1.2). Como vimos en la Sección 1.2, cada ontología de términos GO tiene una estructura jerárquica y esto dota a G de una estructura de digrafo acíclico (DAG) dada por $D = (G, E)$, donde E es el conjunto de relaciones jerárquicas de la ontología. El conjunto E está compuesto por pares (k, l), donde $k, l \in G$ y l es un término GO más específico que k.

Además, para cada nodo $g \in G$ denotamos por $\mathsf{child}(g) = \{l : (g, l) \in E\}$ al conjunto de hijos de g; por $\mathsf{par}(g) = \{k : (k, g) \in E\}$ a los padres de g; y de forma análoga se definen $\mathsf{anc}(g)$ y $\mathsf{desc}(g)$ como los ancestros y descendientes, respectivamente. Otra notación que será de utilidad introducir es $\mathsf{sib}(g)$, el conjunto de hermanos de g, y se define de la siguiente manera:

$$\mathsf{sib}(g) = \left(\bigcup_{k \in \mathsf{par}(g)} \mathsf{chil}(k) \right) \bigcup \left(\bigcup_{l \in \mathsf{chil}(g)} \mathsf{par}(l) \right).$$

Dados un gen $x_i \in X$ y un término GO $g_k \in G$, notaremos por $x_i \in g_k$ cuando el gen x_i esté anotado con el término GO g_k y $x_i \notin g_k$ cuando no lo esté.

En este trabajo se implementan modelos locales (véase Sección 1.3), es decir un modelo clasificador para cada término GO de la ontología. Para distinguir entre los modelos por término GO y el modelo para la ontología nos referiremos a este último como el modelo jerárquico. El clasificador f_k del modelo para el término GO g_k, con $1 \leq k \leq |G|$, asigna para cada gen x_i una probabilidad $p^*_{ik} \in [0, 1]$ que representa la probabilidad del gen x_i de estar anotado con el término GO g_k. Con el conjunto de $|G|$ clasificadores se puede formar el vector $p^*_i = (p^*_{i1}, \ldots, p^*_{i|G|})$, el cual es el resultado preliminar de la clasificación para el gen x_i.

Diremos que una predicción $p^*_i = (p^*_{i1}, \ldots, p^*_{i|G|})$ es consistente con la jerarquía, o simplemente consistente, cuando para todo $(k, l) \in E$ se cumple que $p^*_{ik} \geq p^*_{il}$, es

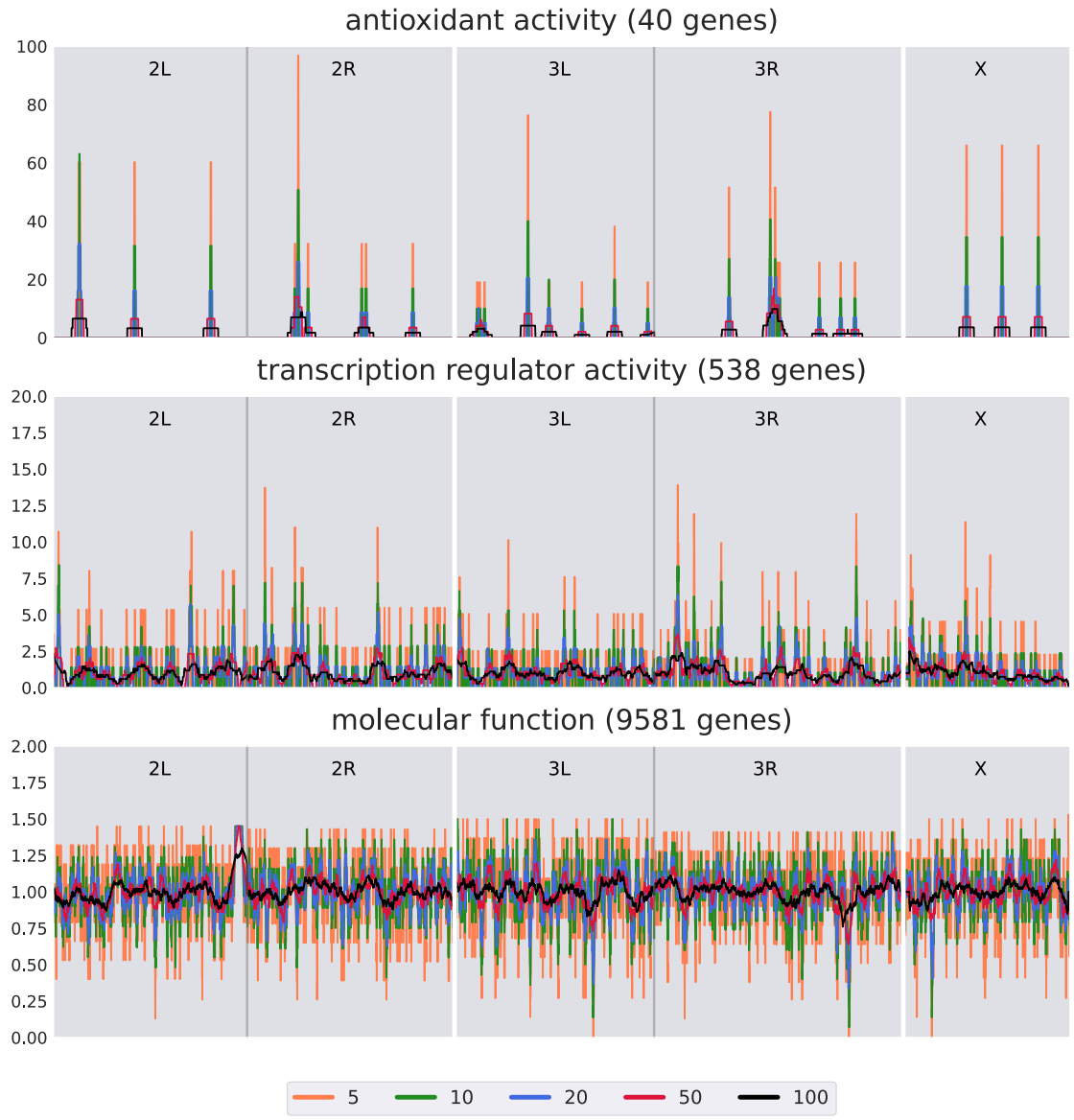


Figura 2.3: Enriquecimiento funcional local a lo largo del genoma de *Drosophila melanogaster* en tres términos GO de la misma rama de la ontología *Molecular Function*. El término "*molecular function*" (GO:0003674) (gráfico inferior) es el nodo raíz de la ontología, que contiene a "*transcription regulator activity*" (GO:0140110) (gráfico central), que a su vez contiene a "*antioxidant activity*" (GO:0016209) (gráfico superior). El número de genes que están anotados con cada término se indica entre paréntesis en la parte superior de cada gráfico. En el eje x se representan los genes que conforman cada brazo cromosómico, cuyos nombres están indicados en negro en la parte superior. Las líneas grises verticales indican la posición de los centrómeros y las líneas blancas separan cromosomas. En el eje y se representa el enriquecimiento encontrado en el entorno de cada gen utilizando cinco ventanas diferentes. Notar que los gráficos tienen escalas distintas en el eje y. Se excluyen los cromosomas 4 e Y por ser muy pequeños.

decir, cuanto más general sea el término GO mayor será la probabilidad de estar anotado con el mismo. Dado que las predicciones por término GO son independientes entre sí en un modelo local, no necesariamente son consistentes con la jerarquía, y por este motivo haremos un postprocesamiento al vector de probabilidades p_i^* para obtener un nuevo vector de probabilidades $p_i = (p_{i1}, \ldots, p_{i|G|})$ que sea consistente. Este postprocesamiento es una modificación al propuesto en [Feng et al., 2017, Feng et al., 2018].

Dada una predicción preliminar $p_i^* = (p_{i1}^*, \ldots, p_{i|G|}^*)$, podemos obtener otra consistente de la siguiente manera:

$$p_{ik} = \begin{cases} p_{ik}^*, & \text{si } k \text{ es el nodo raíz}, \quad (2.1) \\ \min\limits_{l \in \mathsf{par}(k)} p_{il}, & \text{si } p_{ik}^* > \min\limits_{l \in \mathsf{par}(k)} p_{il}, \quad (2.2) \\ p_{ik}^*, & \text{en otro caso}. \quad (2.3) \end{cases}$$

Las Ecuaciones 2.2 y 2.3 pueden resumirse en:

$$p_{ik} = \begin{cases} p_{ik}^*, & \text{si } k \text{ es el nodo raíz}, \\ \min\left\{ p_{ik}^*, \min\limits_{l \in \mathsf{par}(k)} p_{il} \right\}, & \text{en otro caso}. \end{cases} \quad (2.4)$$

La condición $p_{ik} \leq \min_{l \in \mathsf{par}(k)} p_{il}$ para todo término GO k supone que el vector de probabilidades $p_i = (p_{i1}, \ldots, p_{i|G|})$ es consistente con la jerarquía. Mientras que de la condición $p_{ik} \leq p_{ik}^*$ se deduce que este postprocesamiento no asigna una probabilidad mayor que la de cada modelo por término GO.

2.6. Modelos Jerárquicos

De aquí en adelante nos referiremos siempre a un modelo por organismo y ontología, por lo que analizaremos 15 modelos en simultáneo. Emplearemos un modelo muy similar al propuesto en [Feng et al., 2017, Feng et al., 2018], introduciendo algunas variantes para poder emplear LEA. A continuación se describe cada una de las etapas de estos modelos en el orden en el que deben ejecutarse dentro del modelo jerárquico.

1. *Partición del genoma*: en primer lugar separamos el genoma en dos conjuntos, uno que utilizamos para entrenar los modelos y que denominamos $\mathbb{T}$ y otro que utilizamos para evaluar los modelos y que denominamos $\mathbb{E}$, con el 80 % y 20 % del genoma respectivamente. Estos dos conjuntos se seleccionan de forma

aleatoria y son complementarios entre sí. En el paso 3, empleando únicamente el conjunto $\mathbb{T}$ calcularemos los valores de LEA, es decir, a la hora de calcular LEA para cada término GO sólo utilizaremos los genes anotados dentro de este 80 % del total del genoma. Este procedimiento tiene como fin que a la hora de evaluar nuestro modelo el valor de LEA sea independiente de nuestro conjunto de evaluación.

2. *Selección de términos GO*: en segundo lugar sólo consideramos aquellos términos GO que tengan por lo menos 40 anotaciones en $\mathbb{T}$ y 10 en $\mathbb{E}$. Ya que los conjuntos $\mathbb{T}$ y $\mathbb{E}$ se eligen al azar, los términos GO que consideraremos se verán alterados según la elección inicial de los mismos. En la Tabla 2.3 se muestran la cantidad de términos GO por ontología y organismo que se emplearon para construir el modelo jerárquico.

	Biological Process	Cellular Component	Molecular Function
S. cerevisiae	525	137	137
C. elegans	551	117	151
D. melanogaster	880	176	212
M. musculs	1040	285	364
H. sapiens	1212	338	369

Tabla 2.3: Cantidad de términos GO, agrupados por organismo y ontología, con al menos 40 anotaciones en el conjunto de entrenamiento $\mathbb{T}$ y 10 anotaciones en el conjunto de test $\mathbb{E}$. Este conjunto de términos GO es el que se empleo para construir el modelo jerárquico por ontología.

3. *Cálculo de las variables predictivas*: se calculó LEA para todos los términos GO descriptos en el paso anterior, considerando sólo aquellas anotaciones que pertenecen al conjunto de entrenamiento descripto en el paso 1. Se computó LEA para todos los términos GO juntos ya que, como detallaremos más adelante (ver Pasos 4 y 5), los valores de LEA de un mismo término GO fueron empleados en varios modelos locales y además al calcularlos a la misma vez pudimos sacar provecho de la vectorización en Python, reduciendo el tiempo total de computo.

   Los tamaños de ventana utilizados para llevar a cabo LEA fueron 5, 10, 20, 50, 100.

4. *Construcción de datasets y Siblings Policy*: dado que seguimos una estrategia local (en la que debemos entrenar un clasificador binario por cada término GO, ver Sección 1.3) debemos construir una muestra de entrenamiento por cada término GO, que incluya instancias positivas y negativas. Sin embargo, son excepcionales las ocasiones en las que un gen está anotado negativamente con un término GO (ver Tabla 2.2). En [Silla and Freitas, 2011] se propone resolver este problema explorando la estructura jerárquica con la *Siblings Policy*.

   La Siblings Policy consiste en considerar como instancias positivas para un término GO a los genes anotados con el mismo y como instancias negativas a los genes anotados con los términos GO del conjunto de nodos hermanos a este. En caso que este término GO no tenga hermanos o estos no tengan anotaciones, se seleccionan las instancias positivas pertenecientes a los hermanos de los nodos padres. En caso de haber genes anotados con un término GO y con algún término GO hermano del primero, lo consideraremos como una instancia positiva.

   De esta manera, para cada término GO se define un conjunto de genes con el cual entrenaremos nuestro modelo local, conformado por el conjunto de genes que resultan ser una instancia positiva o negativa para dicho término GO luego de aplicar la Siblings Policy.

   Observar que con la Siblings Policy si un gen esta anotado con un término GO y no lo está con ningun término GO hermano de este, entonces será una instancia positiva en el clasificador para el primer término y una instancia negativa de los términos hermanos a este. De la misma manera, si un gen está anotado con dos término hermanos, entonces será una instancia positiva en el clasificador de ambos términos. Por lo tanto, para distintos término GO, los conjuntos de entrenamiento y de test de los correspondientes clasificadores no serán los mismos. Más aun, las etiquetas de un mismo gen pueden no coincidir en distintos conjuntos de entrenamiento o de evaluación.

   Las variables predictivas asociadas a cada gen son obtenidas con LEA; son los valores de enriquecimiento fucnional local en los términos GO padres, hijos y hermanos, ver Sección 2.5 obtenidos usando ventanas de tamaño de 5, 10, 20, 50, 100. Observar que la cantidad de variables predictivas disponibles para cada clasificador binario depende de la cantidad de términos padres, hijos y

hermanos que tenga el término GO considerado.

5. *Entrenamiento de clasificadores*: a diferencia de [Feng et al., 2017], en lugar de máquinas de vectores soporte (SVM) utilizamos *Random Forests* (ver Anexo B para más detalles acerca de Random Forest) como algoritmo clasificador, que se suele desempeñar tan bien como SVM, pero cuyo entrenamiento tiene un menor costo computacional. Esta diferencia en el tiempo de entrenamiento es significativamente importante en nuestro caso, ya que estamos entrenando un clasificador por término GO (dado que desarrollamos un modelo local) y tenemos, como se muestra en la Tabla 2.3, miles de términos GO distintos. Además, debemos repetir este procedimiento para cada uno de los 5 organismos y 3 ontologías.

   Por el mismo motivo optamos por no aplicar SMOTE [Chawla et al., 2002], una técnica que se usa para generar artificialmente nuevos casos etiquetados cuando se dispone de muestras de entrenamiento pequeñas, como se describe en [Feng et al., 2017].

   Los hiperparámetros que ajustamos fueron profundidad, cantidad de árboles y medida de disimilitud, definidos tras realizar búsqueda en grilla y validación cruzada.

6. *Postprocesamiento de probabilidades*: por la adopción de la Siblings Policy, cada clasificador binario por término GO k tiene asociado un conjunto de test $\mathbb{E}_k$, el cual es un subconjunto del conjunto de test $\mathbb{E}$ definido en el Paso 1. Para cada término GO k, el correspondiente clasificador retorna la probabilidad de estar anotado con el mismo término para cada gen del conjunto $\mathbb{E}_k$. Para los genes de $\mathbb{E}$ que no pertenezcan a $\mathbb{E}_k$ asumiremos que esta probabilidad es 0.

   Dado que estos clasificadores son entrenados de forma independiente entre sí, resulta que sus predicciones no son necesariamente consistentes con la jerarquía. Para obtener predicciones consistentes se emplea el postprocesamiento descripto en la Sección 2.5.

   Observar que el postprocesamiento de las probabilidades nos permite acoplar todos las predicciones de los clasificadores por término GO en una única predicción consistente con la jerarquía, esto a pesar que los conjuntos de test de cada clasificador sean diferentes. En la Sección 3.2.1 consideraremos una va-

riante de esta configuración en la que clasificaremos sobre todo el conjunto de test $\mathbb{E}$ con cada clasificador por término GO.

7. *Selección del threshold*: finalmente, para obtener los términos GO predichos por el modelo jerárquico para cada gen, debemos seleccionar un valor umbral de clasificación $\theta \in [0, 1]$ y entonces diremos que el término GO k está asignado al gen x_i si $p_{ik} \geq \theta$, donde p_{ik} es la probabilidad del modelo jerárquico luego del postprocesamiento del paso anterior. El valor de θ se selecciona de forma tal de maximizar la métrica hF_1 que se definirá en la Sección 2.7.

   Observar que si $p_i = (p_{i1}, \ldots, p_{i|G|})$ es consistente, entonces al fijar un valor de θ si el modelo jerárquico predice que el gen x_i está anotado con un cierto término GO, entonces también estará anotado con los términos GO ancestros de este. Esto se debe a que si g_1 y g_2 son dos términos GO, siendo g_1 un término más general que g_2 (o dicho de otra manera, g_1 es un ancestro de g_2), entonces $p_{ig_1} \geq p_{ig_2}$.

En la Tabla 2.4 se muestra un resumen de la cantidad de genes y términos GO empleados en cada modelo jerárquico, ver Sección 1.3, por organismo y ontología.

Organismo	Genes que codifican proteínas	Ontología	Total de términos GO	Términos GO considerados
S. cerevisiae	5892	BP	5074	525
		CC	1035	137
		MF	2323	137
C. elegans	7356	BP	5661	551
		CC	1110	117
		MF	2226	151
D. melanogaster	11122	BP	7416	880
		CC	1277	176
		MF	2599	212
M. musculus	20809	BP	15318	1040
		CC	1953	285
		MF	4269	364
H. sapiens	17276	BP	13816	1212
		CC	1818	338
		MF	4244	369

Tabla 2.4: Algunas características de cada modelo jerárquico entrenado por organismo y ontología. En la segunda columna se muestra la cantidad de genes que codifican proteínas para cada organismo, en la tercera la ontología, en la cuarta se indica la cantidad de términos GO que presentan por lo menos algún gen anotado en ese organismo y ontología y en la quinta columna se muestra la cantidad de términos GO que, como se indica en el Paso 2, tienen al menos 40 anotaciones en el conjunto de entrenamiento y 10 en el conjunto de evaluación.

2.7. Métricas Jerárquicas

Las métricas más usadas en problemas de clasificación desbalanceados son *Precision* y *Recall*. Si denotamos por TP al número de verdaderos positivos, por FN al número de falsos negativos y por TN al número de verdaderos negativos, *Precision*

(*Prec*) y *Recall* (*Rec*) se definen de la siguiente manera:

$$Prec = \frac{TP}{TP + FP} \quad , \quad Rec = \frac{TP}{TP + FN}. \tag{2.5}$$

En palabras, podemos decir que *Prec* es la proporción de ejemplos verdaderamente positivos etiquetados por el clasificador, mientras que *Rec* es el porcentaje de ejemplos verdaderos que han sido etiquetados por el clasificador. Cuál de las dos es más importante depende de las características concretas del problema en cuestión.

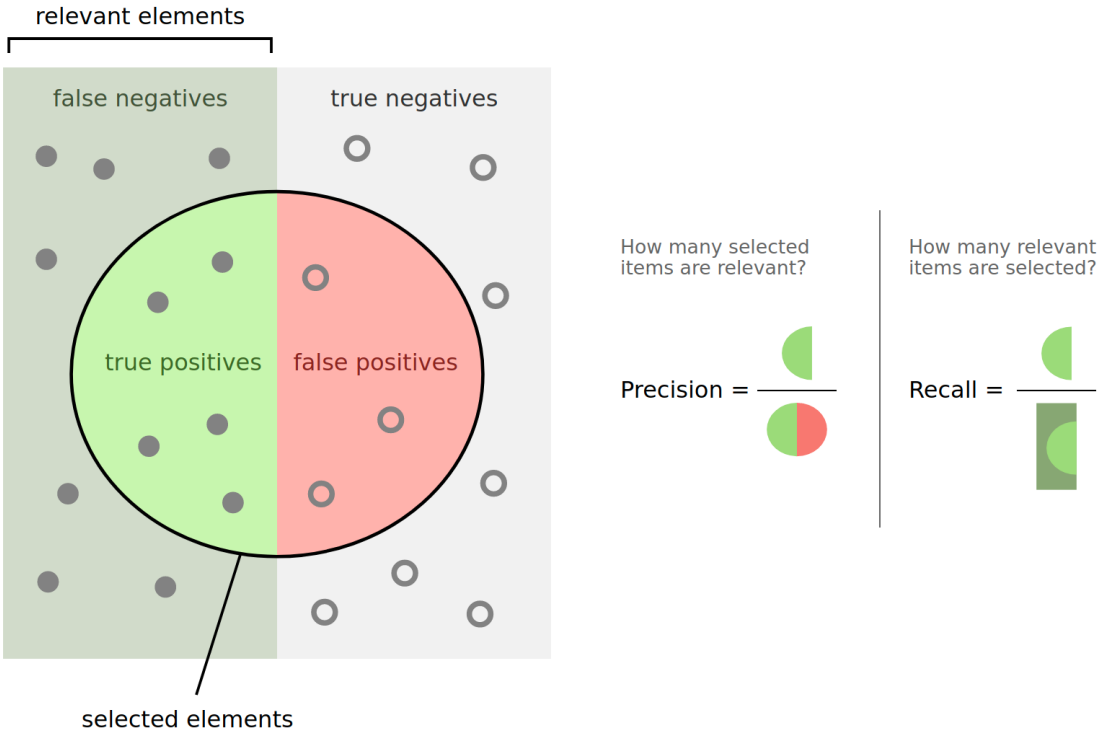


Figura 2.4: Representación esquemática de *Precision* y *Recall*, extraídas de Wikipedia.

Una métrica muy empleada en los problemas de predicción de función de genes es F_1, la media armónica entre *Prec* y *Rec*, que se define de la siguiente manera:

$$F_1 = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec}.$$

Sin embargo, el uso de métricas de clasificación planas no son suficientes para darnos una visión adecuada sobre el rendimiento de nuestro modelo al no tener en cuenta la estructura jerárquica de las clases. Algunos autores [Cerri et al., 2013, Silla and Freitas, 2011] han propuesto sus propias métricas para clasificaciones jerárquicas, pero suelen ser sólo empleadas por ellos mismos o no son válidas para cuando

las clases tienen estructura de DAG como es el caso de la ontología de términos GO. Sin embargo, Kiritchenko [Kiritchenko et al., 2006] propuso nuevas métricas que generalizan a *Prec* y *Recall* en este nuevo contexto y que son ampliamente utilizadas por la comunidad de investigadores, incluyendo las competencias CAFA (ver Sección 1.4), para analizar el desempeño de los modelos propuestos [Radivojac et al., 2013, Friedberg and Radivojac, 2017].

Estas últimas métricas se denominan *hierarchical precision* (*hPrec*), *hierarchical recall* (*hRec*) y *hierarchical* F_1 (hF_1) y se definen de la siguiente manera:

$$hPrec(\theta) = \frac{\sum_{i=1}^{n} |P_i(\theta) \cap T_i|}{\sum_{i=1}^{n} |P_i(\theta)|} \quad , \quad hRec(\theta) = \frac{\sum_{i=1}^{n} |P_i(\theta) \cap T_i|}{\sum_{i=1}^{n} |T_i|} \tag{2.6}$$

$$hF_1(\theta) = \frac{2 \cdot hPrec(\theta) \cdot hRec(\theta)}{hPrec(\theta) + hRec(\theta)},$$

donde $\theta \in [0, 1]$ es un valor umbral, n es el número de genes, T_i es el conjunto de términos GO anotados con el i-ésimo gen y $P_i(\theta)$ es el conjunto de términos GO predichos para el gen i y el valor de θ dado, es decir, el conjunto de términos GO para los cuales el modelo asigna una probabilidad mayor o igual que θ de estar anotado. Asumiremos que la raíz de la correspondiente ontología pertenece siempre a $P_i(\theta)$ para cualquier gen i y $\theta \in [0, 1]$. Sin embargo, para los genes i sin anotaciones de término GO tenemos que T_i es vacío.

Las métricas jerárquicas definidas en las Ecuaciones 2.6 son una generalización de las métricas planas definidas por las Ecuaciones 2.5 ya que el número de verdaderos positivos (TP) se corresponde con $|P_i(\theta) \cap T_i|$, el número TP + FP con $|P_i(\theta)|$ y TP + FN con $|T_i|$.

Otros autores [Alaydie et al., 2012, Vateekul et al., 2014] consideran una versión micro-averaging y otra macro-averaging, guardando la denominación micro-averaging para las métricas definidas por las Ecuaciones 2.6. Si denotamos por:

$$hPrec_i(\theta) = \frac{|P_i(\theta) \cap T_i|}{|P_i(\theta)|} \quad , \quad hRec_i(\theta) = \frac{|P_i(\theta) \cap T_i|}{|T_i|}$$

$$hF_{1,i}(\theta) = \frac{2 \cdot hPrec_i(\theta) \cdot hRec_i(\theta)}{hPrec_i(\theta) + hRec_i(\theta)},$$

entonces las versiones macro se definen de la siguiente manera:

$$hPrec^M(\theta) = \frac{1}{n} \sum_{i=1}^{n} hPrec_i(\theta) \quad , \quad hRec^M(\theta) = \frac{1}{n} \sum_{i=1}^{n} hRec_i(\theta)$$

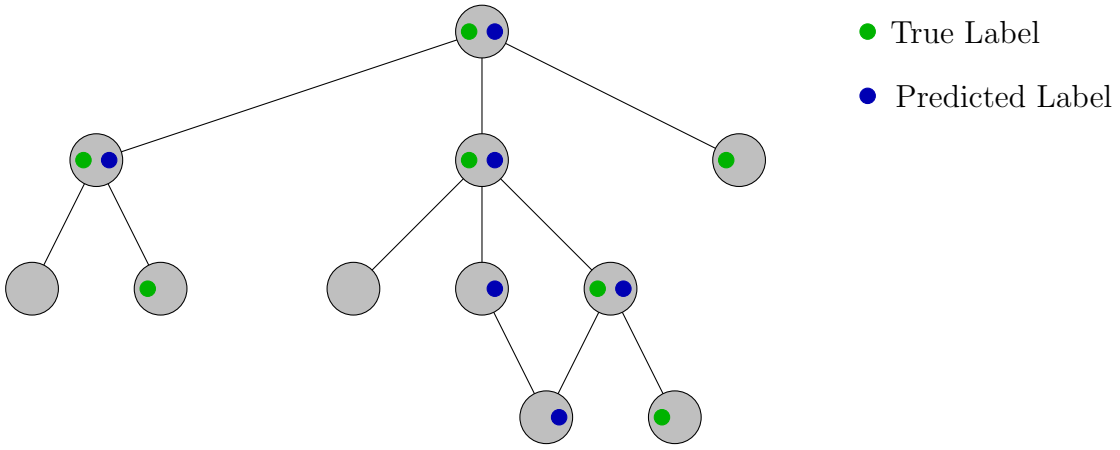


Figura 2.5: Cómputo de las métricas definidas en las Ecuaciones 2.6. Los nodos grises representan los términos GO de la ontología, cuanto más arriba esté el nodo menos específico es. Si para un determinado gen k los puntos verdes representan los términos GO anotados con este gen y los azules los que el modelo predice para un cierto valor de θ, entonces $|P_k(\theta)| = 6, |T_k| = 7$ y $|P_k(\theta) \cap T_k| = 4$.

$$hF_1^M(\theta) = \frac{1}{n} \sum_{i=1}^{n} hF_{1,i}(\theta).$$

Para proporcionar un número único para las comparaciones entre métodos, calculamos la medida F_1 para cada umbral y calculamos su valor máximo sobre todos los umbrales, tanto para las versiones micro como las macro. Más específicamente, tenemos que:

$$hF_{\mathsf{max}} = hF_{\mathsf{max}}^{\mu} = \max_{\theta \in [0,1]} hF_1(\theta) \qquad , \qquad hF_{\mathsf{max}}^{M} = \max_{\theta \in [0,1]} hF_1^M(\theta). \tag{2.7}$$

Todas estas métricas tienen en común que, cuanto mayor es su valor, mejor es el desempeño del clasificador. En este trabajo emplearemos las métricas definidas en las Ecuaciones 2.6, ya que también son las que se emplean en las competencias CAFA.

2.8. Comparación con un modelo aleatorio

Una forma habitual de evaluar el desempeño de un clasificador es comparándolo con un modelo con una distribución aleatoria. Sin embargo, en el caso particular de este trabajo, el clasificador puede tener una cantidad variable de predicciones y además estas predicciones deben ser consistentes con la jerarquía.

Por este motivo se propone un *modelo aleatorio* con el cual comparar los resultados del modelo jerárquico implementado. Este modelo aleatorio consiste en asignar a cada par gen–término GO una probabilidad con distribución uniforme en el intervalo $[0, 1]$ y luego realizar el postprocesamiento descripto en la Sección 2.5.

Con este modelo aleatorio se resuelven los dos problemas presentados al inicio de la sección: que el modelo aleatorio tenga una cantidad variable de términos GO por gen y que además sea consistente con la jerarquía.

2.9. Implementación

Todos los experimentos llevados a cabo en este trabajo fueron ejecutados en la plataforma de ClusterUY [Nesmachnow and Iturriaga, 2019]. Sin esta infraestrutura hubiera sido imposible implementar estos modelos, debido a la demandante necesidad de cómputo. Observar que se desarrollaron 15 modelos jerárquicos (ver Sección 2.6), uno por cada par organismo–ontología, y que para cada uno de estos modelos se deben entrenar cientos (ver Tabla 2.3) de clasificadores Random Forest con búsqueda de hiperparámetros (ver Paso 5 de la Sección 2.6), uno por cada término GO.

El más pequeño de estos modelos jerárquicos, empleando 10Gb de memoria RAM, 20 CPUs y corriendo de forma paralelizada sobre la arquitectura de ClusterUY llevaba un tiempo aproximado de 1 hora, mientras que el modelo más grande utilizando 50 GB de memoria RAM, 40 CPUs y también corriendo de forma paralelizada empleaba más de 12 horas.

A todo este tiempo de procesamineto, cabe agregar el tiempo necesario para la implementación, testeo y depuración de los modelos. Por este motivo nos fue imposible desarrollar y probar otros métodos con los cuales comparar el modelo que terminamos implementando.

Este trabajo fue implementado en Python 2.7.5 bajo la arquitectura de ClusterUY y las principales librerías utilizadas (ordenadas alfabéticamente) fueron las siguientes:

1. `Joblib` y `Multiprocessing`: un conjunto de herramientas que permite la ejecución de código de forma paralela en varios hilos de procesamiento simultáneamente.

2. `Matplotlib` y `Seaborn`: dos librerías para la visualización de datos en python.

3. `NetworkX`: un paquete de Python para la creación, manipulación y estudio de la estructura, dinámica y funciones de redes complejas. Esta librería fue escencial para la manipulación de los grafos de las ontologías.

4. `NumPy`: es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente de alta dimensionalidad, que permite la manipulación de datos de forma altamente eficiente.

5. `Pandas`: es una librería escrita como una extensión de NumPy para la manipulación y análisis de datos, que ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.

6. `Scikit-learn`: es una librería que cuenta con una amplia gama de algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad. Además, presenta compatibilidad con otras librerías de Python como NumPy, SciPy y Matplotlib.

Capítulo 3

Implementación y Resultados

3.1. Implementación

3.1.1. Preprocesamiento

En el repositorio gfpml-tools pueden encontrarse las herramientas necesarias para leer y dar formato a los datos descriptos en la Sección 2.2, estos incluyen:

1. Generar el modelo de cromosoma presentado en la Sección 2.3.
2. Constriur los grafos de las ontologías a partir de los datos descriptos en la Sección 2.2.1.
3. Extraer las anotaciones de los archivos gaf y generar las anotaciones jerárquicas, ver Sección 2.1.
4. Generar LEA para cada término GO con al menos veinte anotaciones y tamaños de ventanas de 5, 10, 20, 50 y 100, ver Sección 2.4. En el repositorio gfpml-datasets se pueden encontrar estos resultados. Tener en cuenta que estos datos de LEA dependen de las anotaciones, del cromosoma y de los términos GO empleados; diferentes versiones de los archivos descriptos en la Sección 2.2 resultarán en distintos valores de LEA.

Los resultados de estos métodos pueden observarse aquí.

3.1.2. Implementación de Modelos y Métricas

En el repositorio gfpml-models se puede encontrar la implementación del proceso descripto en la Sección 2.6, así como la implementación de las métricas de evaluación detalladas en la Sección 2.7. Finalmente, las predicciones de este modelo pueden encontrarse en el repositorio gfpml-result.

3.2. Resultados

3.2.1. Predicción de nuevas anotaciones

Como se detalló en la Sección 2.6, en este trabajo se implementó un modelo local, es decir, un modelo en el cual se entrena un clasificador binario para cada término GO. Si bien, cada uno de estos clasificadores predice sobre un conjunto de test $\mathbb{E}_k$ diferente, todas estas predicciones pueden combinarse mediante el postprocesamiento descripto en la Sección 2.5 (ver Ecuación 2.4).

En esta Sección emplearemos los parámetros aprendidos durante el entrenamiento de estos clasificadores y con ellos clasificaremos el total de genes del conjunto de test $\mathbb{E}$ (el conjunto de test consiste del 20 % del total del genoma, ver el Paso 1 de la Sección 2.6). Este nuevo procedimiento, al igual que antes, resulta en una predicción inconsistente con la jerarquía, por lo que aplicamos el postprocesamiento descripto en la Sección 2.5.

Las predicciones obtenidas mediante este procedimiento fueron reunidas en el sitio web de gfpml, en este sitio se puede consultar y descargar estas predicciones, siendo posible realizar búsquedas por organismo, ontología, cromosoma, gen o término GO.

En la Figura 3.1 se comparan los valor del $hF_1(\theta)$ (ver Sección 2.7) del modelo entrenado y el valor $hF_1(\theta)$ del modelo aleatorio para distintos valores del umbral $\theta \in [0, 1]$ para cada uno de los modelos descriptos (ver Sección 2.8). Mientras que en las Figuras 3.2 y 3.3 se muestra el ratio entre el valor $hF_1(\theta)$ del modelo entrenado y el valor $hF_1(\theta)$ del modelo aleatorio agrupados por ontología o por organismo, respectivamente. Todas estas métricas fueron calculadas usando el conjunto de test $\mathbb{E}$ descripto en la Sección 2.6. Cabe recordar que por cómo fueron construídos estos conjuntos, los mismos son independientes del conjunto de genes con los cuales se calculó LEA y se entrenó a cada modelo (ver Sección 2.4.)

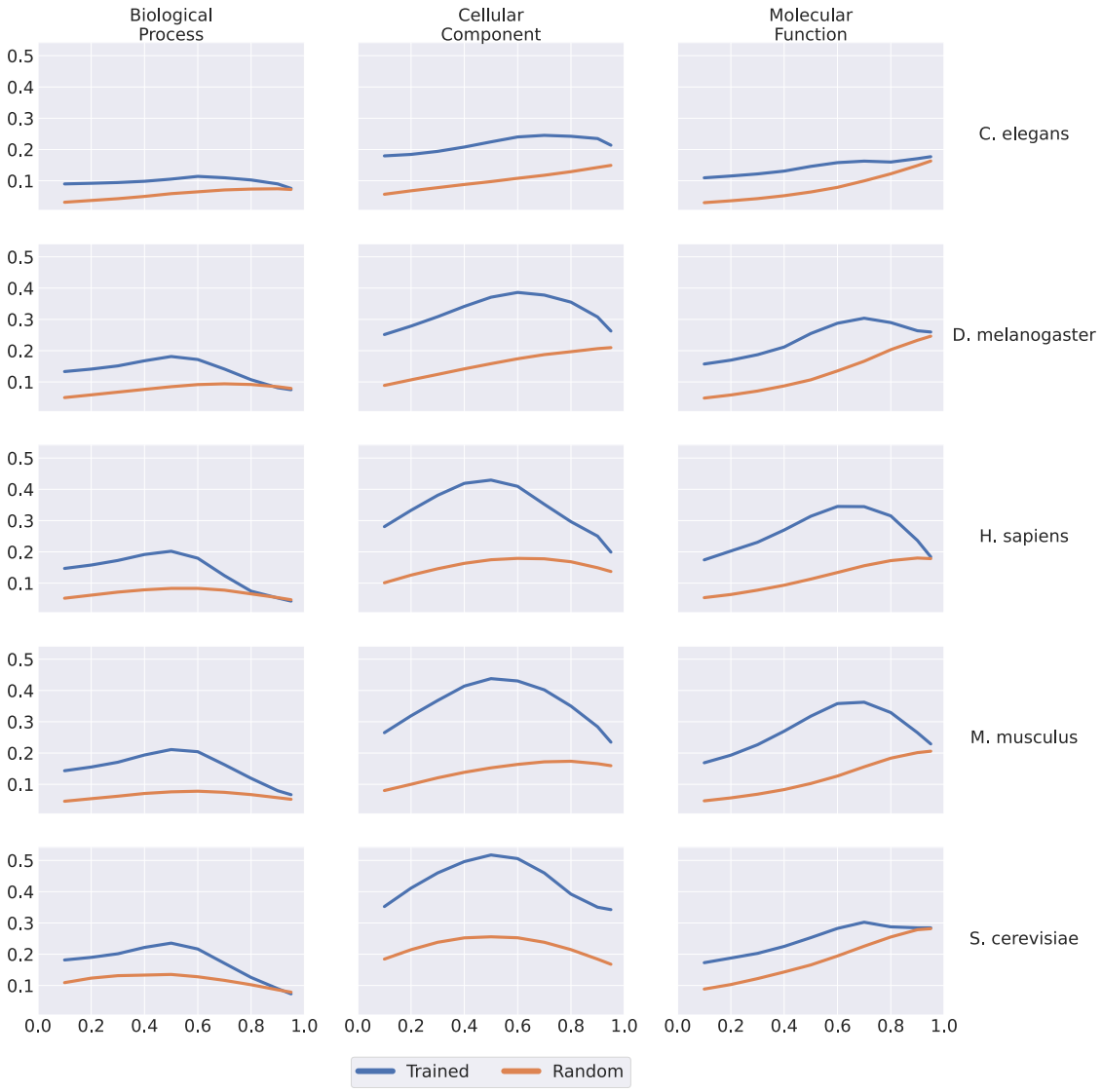


Figura 3.1: Comparación entre el hF_1 calculado con nuestro modelo entrenado y con el modelo aleatorio, ver Sección 2.8. En el eje horizontal tenemos el umbral $\theta \in [0, 1]$ para cada organismo y ontología. En este gráfico cada columna se corresponde con una ontología, mientras que cada fila con un organismo.

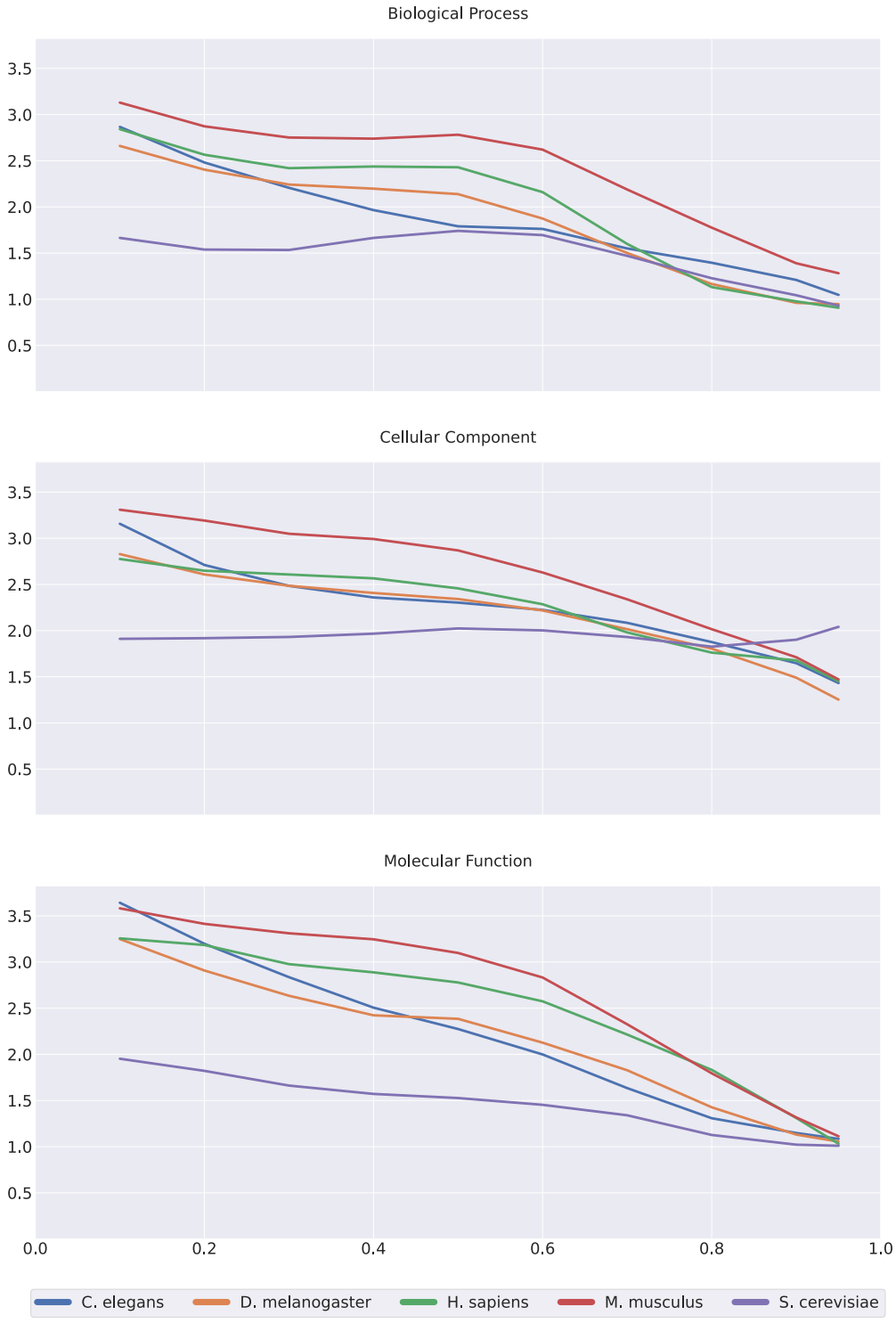


Figura 3.2: Ratios entre el valor hF_1 del modelo entrenado y del modelo aleatorio, ver Sección 2.8, en función del umbral $\theta \in [0, 1]$. En cada gráfico se muestra los resultados de cada ontología, representando cada organismo con un color diferente.

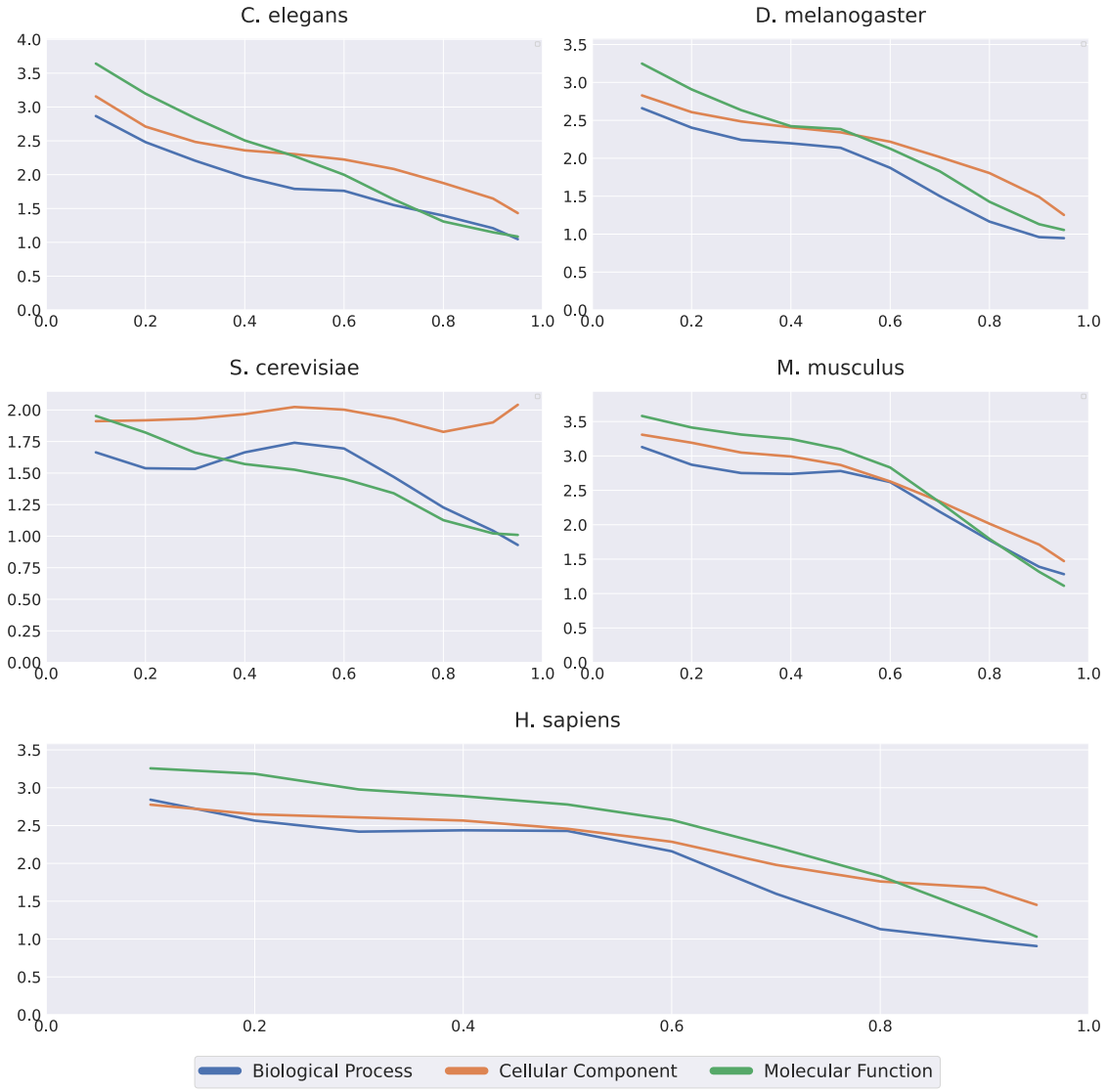


Figura 3.3: Ratios entre el valor hF_1 del modelo entrenado y del modelo aleatorio, ver Sección 2.8, en función del umbral $\theta \in [0, 1]$. En cada gráfico se muestra los resultados de cada organismo, representando cada ontología con un color diferente.

En la Tabla 3.1 se muestra el $hF_{\max}$, ver Ecuación 2.7, para cada modelo jerárquico junto a los correspondientes valores de *Precision* y *Recall* en sus versiones jerárquicas.

Organismo	Ontología	$hPrec$	$hRec$	$hF_{\max}$
S. cerevisiae	BP	0.24	0.23	0.24
	CC	0.51	0.52	0.52
	MF	0.69	0.19	0.30
C. elegans	BP	0.09	0.15	0.11
	CC	0.19	0.33	0.25
	MF	0.25	0.14	0.17
D. melanogaster	BP	0.17	0.20	0.18
	CC	0.41	0.37	0.39
	MF	0.47	0.22	0.30
M. musculus	BP	0.22	0.21	0.21
	CC	0.46	0.42	0.44
	MF	0.63	0.25	0.36
H. sapiens	BP	0.21	0.20	0.20
	CC	0.44	0.42	0.43
	MF	0.47	0.27	0.35

Tabla 3.1: $hF_{\max}$ para cada modelo jerárquico por organismo y ontología. Además se incluyen los valores de *Precision* y *Recall* en sus versiones jerárquicas ($hPrec$ y $hRec$) correspondiente al valor del umbral θ donde se alcanza el $hF_{\max}$. BP, CC y MF son las abreviaciones para *Biological Process*, *Cellular Component* y *Molecular Function*, respectivamente.

Como puede observarse, el ratio entre el valor hF_1 del modelo entrenado y el del modelo aleatorio varía según el organismo y la ontología que se considere. Esto puede deberse a varios factores, entre ellos: la cantidad y calidad de anotaciones, los tamaños de las ventanas y sus ajustes a la cantidad de anotaciones, el modelo de predicción empleado, etc. También es esperable que el poder predictivo de la posición respecto a la función sea variable dependiendo de la ontología y del organismo. Lo importante aquí es que en todos los casos el valor hF_1 del modelo entrenado es varias veces superior al valor hF_1 del modelo aleatorio, lo cual es una evidencia de que, al menos en estos organismos, la posición de un gen es informativa respecto a sus posibles funciones.

En la Tabla 3.2 se muestra el ratio entre el hF_{max} alcanzado por el modelo entrenado y el hF_1 alcanzado por el modelo aleatorio correspondiente sobre el conjunto de entrenamiento para cada posible par organismo–ontología.

Organismo	BP	CC	MF	Media
S. cerevisiae	1.74	2.02	1.34	1.70
C. elegans	1.76	2.08	1.15	1.66
D. melanogaster	2.14	2.22	1.83	2.06
M. musculus	2.78	2.87	2.33	2.66
H. sapiens	2.43	2.46	2.57	2.49
Media	2.17	2.33	1.84	

Tabla 3.2: Relación entre el hF_{max} alcanzado por el modelo entrenado y el hF_1 alcanzado por el modelo aleatorio correspondiente sobre el conjunto de entrenamiento para cada posible par organismo–ontología. BP, CC y MF son las abreviaciones para *Biological Process*, *Cellular Component* y *Molecular Function*, respectivamente.

3.2.2. Comparación con los métodos de referencia de CAFA

Las competencias CAFA, ver Sección 1.4, se han convertido en la principal iniciativa para evaluar el desempeño de los métodos para la predicción de función de genes. En estas competencias participan laboratorios especializados de todo el mundo, que predicen funciones a decenas de miles de proteínas de numerosos organismos. Las variables empleadas para la predicción son múltiples y variadas y cada equipo busca maximizar su performance. Dado que nuestro objetivo no es obtener el mo-

delo que mejor prediga funciones sino demostrar que es posible predecir funciones de genes solamente a partir de su ubicación en el genoma, decidimos cotejar la performance de nuestros modelos jerárquicos con la de uno de los modelos base usados en CAFA3 [Zhou et al., 2019]: BLAST [Altschul et al., 1990].

El enfoque de evaluación utilizado en las competencias de CAFA consiste en, una vez que todas las predicciones son presentadas, abrir un período de varios meses durante el cual se acumulan nuevas anotaciones experimentales de proteínas. Finalizado ese período, estas nuevas anotaciones son utilizadas como referencia para evaluar la performance de los métodos presentados. En el caso de CAFA3, este período de evaluación fue comprendido entre febrero y noviembre de 2017.

Decidimos comparar nuestros resultados utilizando este mismo enfoque, considerando las nuevas anotaciones generadas entre noviembre de 2018 (fecha de las anotaciones GO que utilizamos para entrenar nuestros modelos) y septiembre de 2021.

Sólo datos de 3 de los 5 organismos modelo que usamos en este trabajo se encuentran disponibles para CAFA3: *Drosophila melanogaster, Mus musculs* y *Homo sapiens*. Esta información se encuentra disponible como archivos suplementarios en [Zhou et al., 2019].

Las comparaciones fueron hechas entre las predicciones realizadas por BLAST y las predicciones con probabilidad positiva realizadas por nuestros modelos para estos 3 organismos. Nuestro objetivo con esta comparación es determinar si la ubicación de un gen detro del genoma aporta información relevante para predecir sus funciones. Lo que descubrimos es que para *Biological Process* en los 3 organismos nuestros modelos mejoran los resultados de BLAST, así como para los términos GO de *Cellular Component* en *Drosophila melanogaster* y *Mus musculus*, ver Figura 3.4.

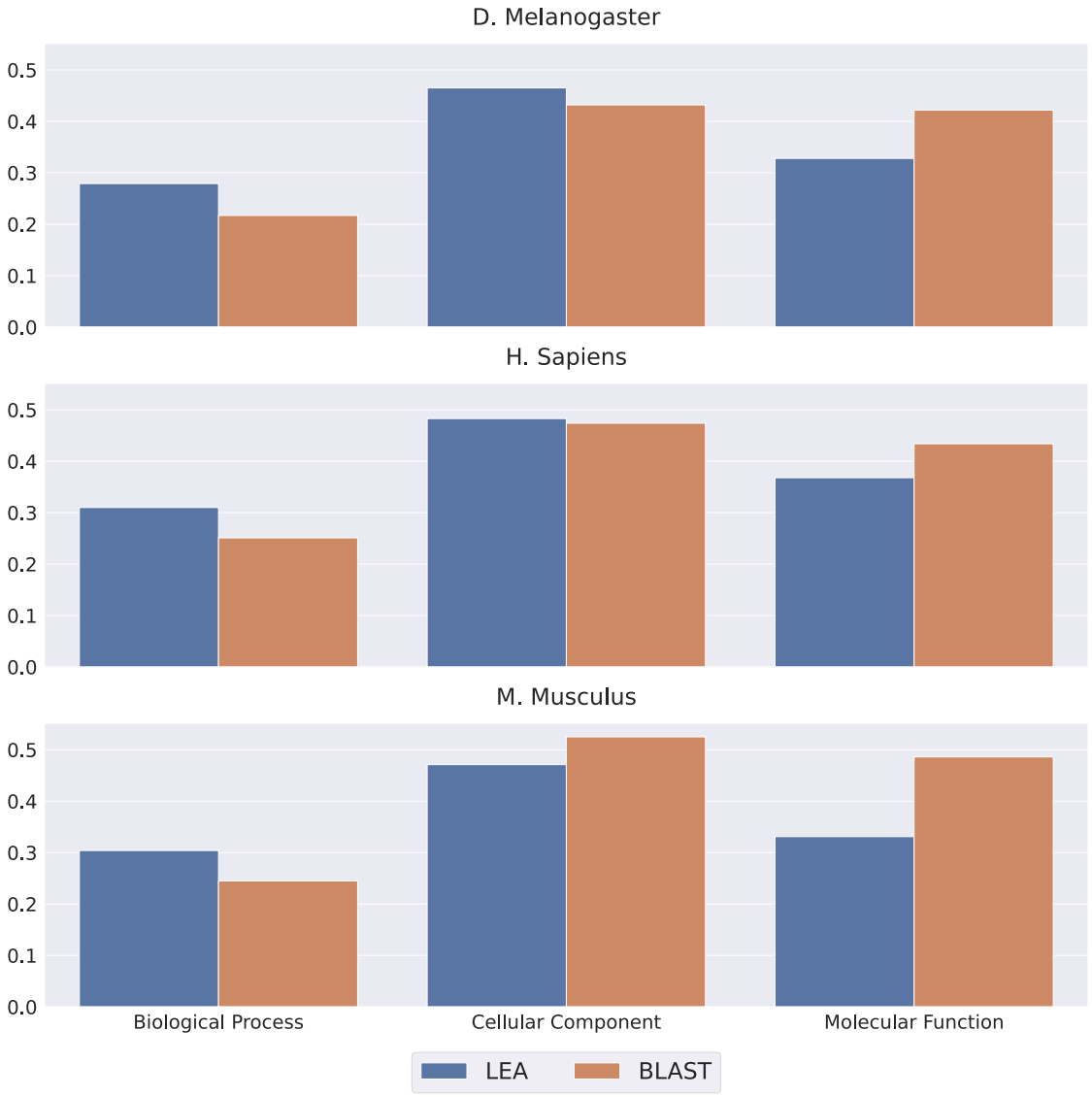


Figura 3.4: Comparación entre nuestro modelo (LEA) y los modelos de BLAST de CAFA3. Estos gráficos muestras los resultados para los organismos de *Drosophila melanogaster*, *Mus musculus* y *Homo sapiens* en las 3 ontologías.

3.3. Conclusiones

La identificación de variables predictivas relevantes para la implementación de modelos de aprendizaje automático es un área relevante en el campo de la predicción de función de genes. La secuencia nucleotídica de un gen determina en buena medida (aunque no por completo) la estructura tridimensional de la proteína que codifica y esta estructura explica generalmente el rol biológico de esa proteína. Esto motivó a que los métodos basados en homología de secuencias, ver Sección A.1, fuesen los pioneros en la prediccción de funciones de genes. Sin embargo, el abordaje tiene sus limitaciones, pues se ha demostrado que dos genes que comparten una misma función biológica no tienen en general un alto nivel de similitud de secuencia. Por ejemplo, en [Duan et al., 2006], se muestra que al menos en *S. Cerevisiae*, la mayoría de las secuencias de proteínas anotadas con el mismo término GO presentan baja homología de secuencias.

Existen otras características de la biología de los genes que se utilizan para la predicción de sus funciones (estructura 3D, patrones de expresión, redes de interacción), sin embargo, en la mayoría de los casos se dispone únicamente de información derivada de la secuenciación de los genomas [Shehu et al., 2016]. Por ello, la identificación de variables predictivas que se puedan extraer directamente de un genoma secuenciado y anotado pero que sean independientes de la homología de secuencia es particularmente importante.

En organismos eucariotas no existen métodos propuestos que se valgan de la distribución relativa de los genes anotados dentro del genoma para predecir sus funciones. En este trabajo buscamos evaluar el potencial predictivo de esa ubicación relativa, una información que se puede generar automáticamente a partir de cualquier genoma anotado utilizando LEA.

Existe abundante bibliografía que indica que la distribución de los genes con una misma función biológica no es aleatoria dentro del genoma, por lo que debería ser posible, al menos en algunos casos, inferir las funciones de los genes a partir de variables relativas a su distribución. Con este objetivo en mente fue que se ideó y desarrolló LEA, ver Sección 2.4: un análisis que permite cuantificar la distribución de la concentración de genes con una misma función a lo largo del genoma. Los resultados de LEA para los cinco organismos modelo se encuentra disponible en gfpml-datasets; mientras que el código de su implementación puede encontrarse en

gfpml-tools.

Se implementaron 15 modelos jerárquicos de aprendizaje automático (ver Sección 2.6), empleando como variables predictivas únicamente los resultados proporcionados por LEA, obteniendo así miles de nuevas asociaciones entre genes y términos GO. Estas predicciones fueron reunidas en el sitio web de gfpml, en este sitio es posible realizar una búsqueda por organismo, ontología, cromosoma, gen o término GO, así como descargar estas predicciones.

Los resultados conseguidos con estos modelos muestran que es posible predecir satisfactoriamente funciones de genes empleando solamente variables relativas a su distribución, ver Figura 3.4. En base a estos resultados creemos que es de fundamental importancia aprovechar la información procedente de la distribución de los genes para mejorar los resultados de futuros modelos que busquen inferir las funciones biológicas de los genes. Puesto que, en gran parte de los casos, la información que deriva de la secuenciación de los genomas es la única disponible.

Anexos

Anexo A

Predicción automática de funciones

Durante las últimas dos décadas, la cantidad de nuevos genes y proteínas secuenciadas viene creciendo a una tasa tan acelerada que lo métodos experimentales han quedado desfasados para determinar sus funciones biológicas (ver Figura 1.1), haciendo de los métodos de predicción automática de funciones (AFP, por sus siglas en inglés, *Automatic Function Prediction*) una necesidad en la biología moderna.

Más aún, al día de hoy, menos del 1 % de las proteínas secuenciadas de *UniProt* (la mayor base de datos de proteínas secuenciadas) [Consortium, 2020] tienen alguna función biológica conocida y conocer la función de estas proteínas es de fundamental importancia ya que promueve el desarrollo de nuevos y mejores fármacos [Barabási et al., 2011, Xuan et al., 2019], impulsa el progreso en el análisis de enfermedades [Kissa et al., 2015, Zeng et al., 2015, Zhang et al., 2019], así como beneficia muchos otros campos de la investigación biomédica.

Por lo tanto, uno de los principales retos en la bioinformática moderna involucra predecir el rol que desempeñan los genes en los procesos biológicos, así como predecir los mecanismos por los cuales tales funciones son llevadas a cabo.

Es difícil dar una categorización pura y exhaustiva de los distintos tipos de AFP desarrollados, ya que siempre hay superposiciones entre ellos, y el abanico de técnicas empleadas es muy amplio. Por este motivo, nos centraremos en describir tres tipos de métodos: *métodos basados en Similitud de Secuencias*, *Métodos Probabilísticos* y *Métodos de Aprendizaje Automático*.

A.1. Métodos basados en Similitud de Secuencias

En la década de los 90, las asignaciones funcionales eran efectuadas siguiendo un principio simple: un gen o producto génico era asociado a un determinado término GO si existía otra secuencia, homóloga o "*similar*" (similar en algún sentido predefinido), anotada con dicho término GO. Es decir, con estos métodos se buscaba transferir anotaciones funcionales de genes y productos génicos a otras secuencias con alto grado de similitud de secuencia.

La efectividad de estos métodos se basa en el hecho de que los genes que tienen un ancestro común; como los genes homólogos u ortólogos, más allá de que se vayan diferenciando entre sí al ir acumulando mutaciones a lo largo del tiempo, muchas veces conservan funciones similares .

La proliferación de este tipo de métodos fue posible gracias a la llegada de formatos de secuencia estandarizados, como FASTA [Pearson and Lipman, 1988] y de algoritmos eficientes para la alineación y comparación de secuencias, como Basic Local Alignment Search Tool (BLAST) [Altschul et al., 1990] y más tarde *Position-Specific Iterated BLAST* (PSI-BLAST) [Altschul et al., 1997].

OntoBlast [Zehetner, 2003], *GOFigure* [Khan et al., 2003], *GOblet* [Hennig et al., 2003] y *GOtcha* [Martin et al., 2004] son sistemas de anotación típicos que adoptan la similitud de secuencia determinada por la búsqueda BLAST. PFP [Hawkins et al., 2009] es otro método que utiliza información funcional asociada con homólogos remotos mediante el empleo de PSI-BLAST.

Sin embargo, la correspondencia entre función y similitud de secuencia tiene sus limitaciones. Por ejemplo, cuando se consideran secuencias cortas, puede encontrarse gran similitud por azar y, por lo tanto, incluso si la similitud de secuencia es alta, no siempre se puede transferir la función de forma certera. Además, a medida que ha crecido la anotación de distintos genomas se ha hecho evidente que la mayoría de las veces los genes que tienen la misma función no tienen secuencias similares [Duan et al., 2006].

A.2. Métodos Probabilísticos

Se han desarrollado diversos modelos probabilísticos para la predicción de función de genes y proteínas [Deng et al., 2003, Deng et al., 2004, Letovsky and Kasif, 2003,

Nariai et al., 2007]. Aquí repasamos algunos de ellos.

En [Letovsky and Kasif, 2003] se utilizó un gráfico de vinculación funcional construído a partir del grafo que representa todas las interacciones entre proteínas (grafo PPI) en el organismo modelo *Saccharomyces cerevisiae*. La hipótesis principal de este trabajo fue que la probabilidad de compartir funciones entre proteínas (nodos) muy próximas en el grafo PPI es mayor que la de los nodos que no están muy próximos. En este método, la probabilidad de estar asociado a un término GO dado se deriva de un modelo binomial que incorpora el algoritmo de Markov Random Fields (MRF). En [Nariai et al., 2007], los mismos autores integran múltiples fuentes de información (grafos PPI, datos de expresión génica, datos de fenotipo mutante y datos de localización proteica) y utilizando un modelo Bayesiano logran mejorar el rendimiento de la predicción, en comparación con el modelo que sólo usa PPI.

En [Lee et al., 2006] se desarrolló un método de kernel logistic regression (KLR) basado en kernels de difusión e incorporando los vecindarios indirectos de las redes PPI. En [Chua et al., 2006] también se consideran los vecinos indirectos con longitud 2, mientras que en este trabajo se calcula la puntuación de similitud funcional entre dos proteínas, que se deriva de la diferencia simétrica de los vecinos y la fiabilidad de las fuentes de datos utilizadas.

A.3. Métodos de Aprendizaje Automático

Entre los métodos de aprendizaje automático más utilizados para la predicción de funciones de genes se encuentran aquellos basados en las maquinas de vectores soporte (SVM por su sigla en inglés). Por ejemplo, en [Vinayagam et al., 2004], y más tarde también en GOPET [Vinayagam et al., 2006], los autores emplearon un sistema de votación basado en la combinación de múltiples clasificadores SVM para predecir anotaciones de términos GO sobre 13 organismos modelo y estimar intervalos de confianza para esas predicciones. En [Vateekul et al., 2014, Feng et al., 2017] se siguen modelos de clasificación jerárquica local (ver Sección 1.3) en donde cada clasificador binario por término GO es un clasificador SVM. Otro ejemplo es la herramienta FFPred [Lobley et al., 2008, Minneci et al., 2013, Cozzetto et al., 2016], la cual actualmente se encuentra en su tercera versión, que establece, de forma independiente de la homología de secuencia, asociaciones entre términos GO y proteínas de organismos eucariotas.

También se utilizan otros algoritmos, como vecinos más cercanos (kNN). Empleando kNN tenemos el ejemplo de PANNZER2 [Törönen et al., 2018], que proporciona un sistema de anotación funcional rápido basado en la homología de secuencia y otras variable que predicen función. Similar es el abrodaje de MS-kNN [Lan et al., 2013], en el que también se integran múltiples fuentes heterogéneas de datos para la predicción de funciones de proteínas.

En los último años los desarrollos dentro del aprendizaje automático aplicado a la AFP han sido monopolizados por los métodos basados en redes neuronales. Por ejemplo, entre las arquitecturas más simples de redes neuronales tenemos las redes de arquitecturas *feed–forward*, que incluyen aplicaciones como [Fa et al., 2018] y DEEPred [Sureyya Rifaioglu et al., 2019].

Las *Convolutional Neural Networks* (CNNs), que fueron originalmente desarrolladas para trabajar con datos en 2 dimensiones, tales como imágenes, han demostrado su efectividad en datos unidimensionales como secuencias genómicas. Las aplicaciones de CNNs para la AFP incluyen a SECLAF [Szalkai and Grolmusz, 2018], DeepSeq [Nauman et al., 2019], DeepGO [Kulmanov et al., 2017] y DeepGO-Plus [Kulmanov and Hoehndorf, 2019].

También se utilizan arquitecturas de redes neuronales especialmente desarrolladas para trabajar con datos unidimensionales, como las *Recurrent Neural Network* (RNN) y las *Long Short-Term Memory* (LSTM). Algunos ejemplos de estos tipos de redes aplicadas a la AFP son ProLanGO [Cao et al., 2017], GONET [Li et al., 2020] y DeepGOA [Zhang et al., 2020].

Las redes neuronales constituyen hoy en día un campo de investigación efervescente y los avances son contínuos. Constantemente se desarrollan nuevos tipos de arquitecturas, tales como GANs, Autoencoders, Transformers, etc., y sus aplicaciones para la AFP continuarán progresando.

Anexo B

Random Forest

Random Forest es un método de aprendizaje automático supervisado de tipo ensamble para la clasificación y regresión que opera mediante la construcción de una multitud de árboles de decisión en el momento del entrenamiento.

El método general de Random Forest fue propuesto por primera vez por Ho en 1995 [Ho, 1995]. En el año 2001 una extensión del algoritmo fue desarrollada por Leo Breiman y Adele Cutler [Breiman, 2001], quienes registraron "Random Forests" como marca comercial en 2006. La extensión combina la idea de *Bagging* de Breiman y la selección aleatoria de características, introducida primero por Ho y luego de forma independiente por Amit y Geman [Amit and Geman, 1997], para construir una colección de árboles de decisión con varianza controlada.

B.1. Modelos tipo Ensamble

Los métodos tipo *ensamble* están formados por un grupo de modelos de aprendizaje que permiten obtener un mejor rendimiento predictivo y estabilidad del modelo del que se podría obtener de cualquiera de los algoritmos de aprendizaje que lo constituyen. Como todos los modelos, los árboles de decisión también sufren el problema de equilibrio entre el sesgo y la varianza.

El término sesgo hace referencia a cuánto se alejan en promedio las predicciones de un modelo respecto a los valores reales. Refleja qué tan capaz es el modelo de aprender la relación real que existe entre los predictores y la variable respuesta.

El término varianza hace referencia a cuánto cambia el modelo dependiendo de los datos utilizados en su entrenamiento. Idealmente, un modelo no debería modificarse

demasiado por pequeñas variaciones en los datos de entrenamiento y si esto ocurre, es porque el modelo está sobreajustandose a los datos en lugar de aprender la verdadera relación entre los predictores y la variable respuesta.

A medida que aumenta la complejidad de un modelo, el mismo dispone de mayor flexibilidad para adaptarse a las observaciones, reduciendo así el sesgo y mejorando su capacidad predictiva. Sin embargo, alcanzado un determinado grado de flexibilidad, aparece el problema del sobreajuste, esto es, el modelo se ajusta tanto a los datos de entrenamiento que es incapaz de predecir correctamente nuevas observaciones. El mejor modelo es aquel que consigue un equilibrio óptimo entre sesgo y varianza.

Los modelos de tipo ensamble buscan reducir la varianza de las predicciones a través de la combinación de los resultados de varios clasificadores. Los tipos de ensambladores comunes son: *Bagging*, *Boosting* y *Stacking*, siendo Random Forest del primer tipo.

B.2. Bagging

El término Bagging es el acrónimo de ***b****ootstrap* ***agg****regat****ing***. En esta técnica cada uno de los modelos individuales es entrenado con cada uno de estos subconjuntos tomados de la misma población. Para predecir, todos los modelos que forman el agregado participan aportando su predicción. Como valor final, se toma la media de todas las predicciones (en caso de regresión) o la clase más frecuente (en caso de clasificación).

Dadas n observaciones independientes $Z_1, \ldots, Z_n$, cada una con varianza σ^2, la varianza de la media de las observaciones es σ^2/n. En otras palabras, promediando un conjunto de observaciones se reduce la varianza. Basándose en esta idea, una forma de reducir la varianza y aumentar la precisión de un método predictivo es obtener múltiples muestras de la población, ajustar un modelo distinto con cada una de ellas, y hacer la media (la moda en el caso de variables categóricas) de las predicciones resultantes. Como en la práctica no se suele tener acceso a múltiples muestras, se puede recurrir a generar múltiples subconjuntos de datos con reemplazo, con cada uno de estos subconjunto de datos se genera un modelo predictivo y finalmente se combinan todos estos submodelos en uno solo promediando o tomando la moda según si la variable a predecir es continua o categórica. A este proceso se le conoce

como Bagging y es aplicable a una gran variedad de métodos de regresión.

En el proceso de Bagging, el número de árboles creados no es un hiperparámetro crítico en cuanto a que, por mucho que se incremente el número, no se aumenta el riesgo de overfitting. Alcanzado un determinado número de árboles, la reducción del error se estabiliza. A pesar de ello, cada árbol ocupa memoria, por lo que no conviene almacenar más de los necesarios.

B.3. Random Forest

Como mencionamos anteriormente, Random Forest consiste en generar una determinada cantidad n de árboles de decisión (`n_estimators` en la librería de scikit-learn) y de estos tomar el valor que más represente a los valores predecidos por cada uno de estos modelos. A su vez, empleando Bagging, cada uno de estos árboles es entrenado con un subdataset diferente en cada iteración. El tamaño de cada uno de estos subdatasets es `max_samples` en scikit-learn.

Recordando la Sección anterior, los beneficios del Bagging se basan en el hecho de que, promediando un conjunto de modelos, se consigue reducir la varianza. Esto es cierto siempre y cuando los modelos agregados no estén correlacionados. Si la correlación es alta, la reducción de varianza que se puede lograr es pequeña.

Por este motivo, para evitar la correlación entre las features empleadas en el entrenamiento de cada árbol de decisión es que cada uno de estos árboles es entrenado con $m \leq p$ features seleccionadas al azar, siendo p el número total de features que se dispone. El número m (`max_features` en scikit-learn) es uno de los hiperparámetros más importantes de Random Forest. Hay varias estrategias para elegir el valor de este atributo que se pueden utilizar, algunos de los valores más recomendados son [Hastie et al., 2004]:

1. $m = \sqrt{p}$ para problemas de clasificación y
2. $m = p/3$ para problemas de regresión.

Para más valores recomendados de este hiperparámetro mirar la documentación de scikit-learn.

Otros hiperparámetros importantes que conciernen a los árboles de decisión que se generan son:

1. `criterion`: la función para medir la calidad de una división.

2. `max_depth`: la profundidad máxima del árbol.

3. `min_samples_split`: el número mínimo de muestras necesarias para dividir un nodo interno.

4. `min_samples_leaf`: el número mínimo de muestras que debe haber para ser en un nodo final (hoja).

5. `max_leaf_nodes`: el número máximo de nodos finales.

Anexo C

Gene function prediction in five model eukaryotes exclusively based on gene relative location through machine learning

En este Anexo adjuntamos el artículo *"Gene function prediction in five model eukaryotes exclusively based on gene relative location through machine learning"*, que recoge los resultados presentados en esta tesis y que ha sido aceptado para su publicación en la revista Nature Scientific Reports.

Gene function prediction in five model eukaryotes exclusively based on gene relative location through machine learning

Flavio Pazos Obregón [a, b] (+) *, Diego Silvera [a] (+), Pablo Soto [a], Patricio Yankilevich [c], Gustavo Guerberoff [d], Rafael Cantera [a]

a - Departamento de Biología del Neurodesarrollo, Instituto de Investigaciones Biológicas Clemente Estable, Montevideo, Uruguay.
b- Unidad de Bioquímica y Proteómica Analíticas, Instituto Pasteur de Montevideo, Montevideo, Uruguay.
c - Instituto de Investigación en Biomedicina de Buenos Aires (IBioBA), CONICET - Partner Institute of the Max Planck Society, Buenos Aires, Argentina.
d - Instituto de Matemática y Estadística “Prof. Ing. Rafael Laguardia”, Facultad de Ingeniería, UDELAR, Montevideo, Uruguay.

(+) These authors contributed equally to this work

* fpazos@iibce.edu.uy Av. Italia 3318, 11600 Montevideo, Uruguay

Abstract

The function of most genes is unknown. The best results in automated function prediction are obtained with machine learning-based methods that combine multiple data sources, typically sequence derived features, protein structure and interaction data. Even though there is ample evidence showing that a gene's function is not independent of its location, the few available examples of gene function prediction based on gene location rely on sequence identity between genes of different organisms and are thus subjected to the limitations of the relationship between sequence and function.

Here we predict thousands of gene functions in five model eukaryotes (*Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster*, *Mus musculus* and *Homo sapiens*) using machine learning models exclusively trained with features derived from the location of genes in the genomes to which they belong. Our aim was not to obtain the best performing method to automated function prediction but to explore the extent to which a gene's location can predict its function in eukaryotes. We found that our models outperform BLAST when predicting terms from Biological Process and Cellular Component Ontologies, showing that, at least in some cases, gene location alone can be more useful than sequence to infer gene function.

Contact: fpazos@iibce.edu.uy

1. INTRODUCTION

We witness a growing gap between the number of assembled genomes and the number of genes with known functions. Less than 1% of the protein sequences in UniProtKB [1] have an experimental Gene Ontology annotation [2] and even in well studied organisms, the majority of known genes have yet no assigned function [3]. Furthermore, well studied genes have frequently been assigned more than one function, so less studied genes, for which only one function is known, have probably more functions to be discovered [4]. In this context there is an increasing need to improve automated function prediction (AFP) [5–9].

The Critical Assessment of protein Function Annotation algorithms (CAFA) is a series of experiments designed to provide a large-scale assessment of computational methods dedicated to automated function prediction (AFP) [7,10,11]. In all CAFA editions so far, the best results were obtained with machine learning-based methods and combining multiple data sources, typically including sequence derived features, protein structure and molecular interaction data. The

performance of the methods evaluated by the CAFA challenges improved dramatically between the first (2013) and the second (2016) edition, but this improvement slowed down between the second and the third edition (2019). The authors hypothesized that including more varied sources of data will lead to additional large improvements in AFP [7].

Thus, finding new ways to extract relevant biological information from the available data is key to improve AFP. For around 99% of all known proteins, the only available information is the sequence encoded in the corresponding genome, highlighting the importance of sequence-based AFP [12]. But AFP based on sequence similarity is hindered by a highly variable correlation between sequence identity and gene function [13] and by the evolutionary distance of many genomes to the closest well-characterized genome [14]. Here we explore the hypothesis that the location of a gene relative to other annotated genes of the same genome, a feature that is independent of sequence homology and that can be directly extracted from any annotated genome, is sufficient to perform AFP on eukaryotic genomes, with a performance similar to that reached by sequence similarity alone.

Functionally related genes may be constrained to remain close to each other due to natural selection, forming conserved gene clusters [15]. Local clusters of co-expressed, co-regulated or functionally related genes have been documented in a wide range of organisms, including prokaryotes, yeast, insects, vertebrates and plants [16–23].

Equating conserved co-locality with co-functionality have been a fruitful approach for the prediction of gene function in prokaryotes for more than 20 years [15,24–28]. On the contrary, there are very few examples [14,29] of the use of this approach in eukaryotic organisms, although also gene functions are non-randomly distributed in their genomes [21]. However, these AFP studies were based on conserved gene neighborhoods, thus subjected to the limitations mentioned above regarding the relationship between sequence and function.

Here we performed AFP on eukaryotic genomes based exclusively on the relative location of genes. In particular, we tested the predictive power of a feature which represents the spatial organization of genes with respect to their annotated functions, which we term "functional landscape arrays" (FLAs). A FLA is an array associated to each gene, that contains the enrichment in a set of Gene Ontology terms (GO terms) found around the gene, considering different window sizes. These arrays contain information which is independent of sequence similarity between genes and that can be automatically extracted from any annotated genome.

We predicted associations between genes of five well-annotated eukaryote genomes (*Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster*, *Mus musculus* and *Homo sapiens*) and terms from the three ontologies of Gene Ontology (Biological Process, Cellular Component and Molecular Function) training a set of hierarchical multi-label classifiers with FLAs.

Then we compared the results of our 15 models, one for each pair organism/ontology, with equivalent models that randomly assign functions to genes. We found that our models, trained exclusively with location-derived features, performed better than chance in the five organisms and in the three ontologies, showing that there is useful information in the way in which genes are distributed along these genomes.

We also compared the performance of our models to the performance of BLAST, one of the baseline methods of CAFA 3 [7]. Using the same approach of the CAFA competitions, we used the updated annotations, released in September 2021, to evaluate the models that we had trained with the annotations released on November 2018. Our models outperformed BLAST when predicting terms form the Biological Process ontology in the three organisms for which specific data from the last CAFA is available and when predicting terms from the Cellular Component ontology our models also performed better in two of these organisms. These results demonstrate that gene location can be informative when performing AFP on eukaryotes. The results also support the idea that gene distribution patterns are tightly regulated in eukaryotic genomes. Finally, our results show that the use of FLAs as predictive feature could complement the annotation of partially annotated genomes.

2. METHODS

2.1 General procedure to predict associations between genes and GO terms

For each genome,

- Model the genome as a string of protein coding genes.
- Random split in sets T and E, containing 80% and 20% of the genes respectively.

For each Ontology,

- Train a binary classifier for each GO term X associated with at least 40 genes in T and 10 genes in E
    - Training set: genes in T annotated with GO term X (as positives) and its siblings (as negatives)
    - Predictive feature: a FLA for each gene, including enrichment in GO term X, its siblings and its ancestors
    - Hyper-parameters set by grid search & cross validation
- Combine all the binary classifications into one hierarchical multi-label classifier using the node interaction method.
- Evaluate performance calculating the hF1 score over the test set E

- Using the classification threshold that maximizes the ratio between the hF1 of the trained model and the hF1 of the random model, predict new associations between GO terms and all the genes in E.

2.2 Genome modeling

We modeled the genome as a collection of segments (the chromosomal arms) in which the protein coding genes -the only elements we considered- are located one next to the other, without intergenic regions or superpositions [30]. In this model, the position of a gene is defined by the location of its transcription starting point and the distance between two genes is the number of other genes located between them. The number of protein-coding genes considered in each genome is shown in Table 1.

2.3 Gene Ontology

Gene Ontology (GO) is an attempt to describe all the knowledge about the biological function of genes with three ontologies: Molecular Function, Cellular Component and Biological Process, each one representing different aspects of the biology of a gene product and organized as a directed acyclic graph [2]. Each "GO term" is a node of these graphs, with precise definition and relationships with other terms. A GO annotation occurs when an association between a gene product and a GO term is established. To train our models we used a version of the ontology downloaded on November 2018. To fulfill the true path rule [31], given the annotations of an organism within a given ontology, we up-propagated all the annotations, meaning that if a gene was annotated with a given GO term we associated that gene with all the ancestor terms up to the root of the graph.

2.4 Local enrichment analysis

Enrichment analysis is a method frequently used to determine if a given gene feature is overrepresented in a list of genes [32]. It assesses if the genes of a list associated with a given feature are more frequent than what should be expected in a list of genes of the same size but randomly picked from the same background list.

Given a gene of interest **j**, we define the Local Enrichment in the GO term **x** for the gene **j** and a window **w** centered in **j** as:

Eq. 1: $\mathbf{E_{jxw}} = ((\mathbf{k}/\mathbf{n}) / (\mathbf{M}/\mathbf{N}))$

where **N** is the number of genes in the chromosomal arm, **M** is the number of genes in the chromosomal arm associated with GO term **x**, **n** is the number of genes in the window and **k** is the number of genes in the window associated with GO term **x** (see Figure 1). In other words, $\mathbf{E_{jxw}}$ assess if the genes annotated with the GO term **x** are located in the surroundings of gene **j** more frequently than what could be expected by chance. This approach was successfully used to look for clusters of GO terms along the genome of seven eukaryotes [33].

2.5 Functional Landscape Arrays and Functional Enrichment Maps

To functionally characterize the surrounding of a gene we calculated its local enrichment in various GO terms. We considered a window **w**, centered in the gene under consideration, that includes 5, 10, 20, 50 or 100 genes to each side of the gene. The window was moved stepwise one gene at a time until the entire chromosome was covered (see Figure 1). Then, for each gene we defined a Functional Landscape Array (FLA): an array with a row for each window size and a column for each GO term whose enrichment was evaluated. Because of computational limitations, in the work we are reporting here, the GO terms included in each FLA depend on the GO term to be classified: we only included the enrichment found in that GO term, its father, its siblings and all its descendants.

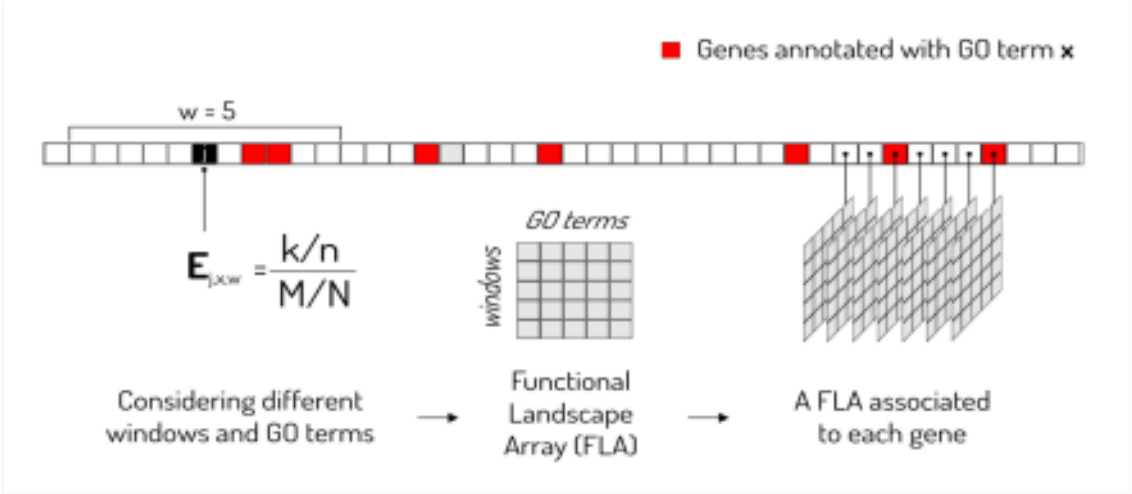


Figure 1. Local enrichment analysis and Functional Landscape Arrays. **k** is the number of genes in the window associated with GO term **x**, **n** is the number of genes in the window, **M** is the number of genes (squares) in the chromosomal arm (strip) associated with GO term **x**, and **N** is the total number of genes in the chromosomal arm.

Importantly: to train our models we did not consider the annotation of the genes in the set E, that was reserved for the evaluation of the models. This procedure guarantees an unbiased evaluation of the classifiers, in which the features used for training are not extracted from examples

used for testing. Nevertheless, because it is a useful result by itself, we also performed Local Enrichment Analysis along each genome considering all its current annotations. We calculated the local enrichment around all the genes in each genome using the same set of window sizes and for all those GO terms associated with at least 20 genes and obtained what we call "functional enrichment maps". The functional enrichment map of a given GO term shows which regions of a genome are enriched in that GO term, for various windows sizes.

2.6 Implementation of hierarchical multi label classifiers

We implemented a hierarchical multi label classifier for each pair organism/ontology using, with some modifications, the algorithm proposed in [34,35]. This is a local approach, since a binary classifier is trained for each GO term. Due to computational limitations, for the binary classification at each node, instead of a Support Vector Machine, we used a Random Forest classifier [36], that has comparable performance in gene function prediction but with lower computational cost. For the same reason we did not use SMOTE [37], a technique used to artificially generate new labeled data when training sets are too small. Depth, number of trees and measure of impurity for each classifier were set by grid search and 3-fold cross validation. Supplementary Table 1 includes the hyper parameters of the models.

First, we randomly split the genome into two sets: **T** and **E**. Set **T** included 80% of the genes and was used to define the training sets and to obtain the FLAs. Set **E** included the remaining 20% of the genes and was used to evaluate the models. We trained a binary classifier for each GO term that was associated with at least 40 genes in **T** and at least 10 genes in **E**. Table 1 shows the amount of GO terms meeting these conditions in each organism and ontology, i.e. the GO terms that could be predicted.

To define the training set for each classifier we applied the siblings policy [38]. We included as positive cases those genes associated with the GO term under consideration and as negative cases those genes associated with the siblings or uncles terms of the GO term under consideration and not associated to that term. Importantly, to construct the FLA associated to each gene, to be used as predictive feature, we only considered the annotations of the genes that belonged to **T**.

With each trained classifier we classified the genes in **E** and then post-processed the predictions using the node interaction method [35], to respect the restrictions imposed by the hierarchy of the ontology. Finally, we evaluated the performance of each hierarchical multi-label classifier using the hierarchical version of the F1 score. All calculations were carried out using ClusterUY (site: https://cluster.uy).

2.7 Evaluation of the models

To evaluate the performance of each trained model we used the complete set of annotations of the genes in **E**, that were not used in training. As evaluation metric we used the hierarchical version of the F1 score (hF1) proposed in [39] and used in the CAFA competitions. If we denote the true and false positives as TP and FP and the true and false negatives as TN and FN, Precision (Pre) and Recall (Rec) are defined as:

Eq. 2 $Pre = TP/(TP+FP)$

Eq. 3 $Rec = TP/(TP+FN)$

and their hierarchical versions, which we term hPre and hRec, are defined as:

Eq.4 $$hPrec(\theta)\frac{\sum_{i=1}^{n}|P_i(\theta)\cap T_i|}{\sum_{i=1}^{n}|P_i(\theta)|}$$

Eq. 5 $$hRec(\theta)\frac{\sum_{i=1}^{n}|P_i(\theta)\cap T_i|}{\sum_{i=1}^{n}|T_i|}$$

where $\theta \in [0,1]$ is the classification threshold, n is the number of genes, T*i* is the set of GO terms truly associated to gene *i* and P*i*(θ) is the set of GO terms predicted for gene *i* with the classification threshold set at θ. We assumed that the root of each ontology always is in P*i*(θ). The hF1 score is the harmonic mean of hPre and the hRec and is defined as:

Eq. 6 $$hF1(\theta) = \frac{2.hPrec(\theta).hRec(\theta)}{hPrec(\theta)+hRec(\theta)}$$

2.8 Comparison with random models

As a way to assess how far from randomness the distribution of gene functions along the genome is, we compared the hF1 of each of our trained models with the hF1 reached by an equivalent model that assigns the term frequency as the prediction score for any gene. In these "random models", if a given GO term occurs with relative frequency 0.25 in a given genome, the probability of association between each gene of that genome and that GO term is set to 0.25 (Radijovac 2013). For each organism and ontology, we obtained the ratio between the hF1 of the trained model and the hF1 of its random version.

2.9 Comparison to one of the CAFA baseline methods

We also compared the performance of our models to the performance of BLAST, one of the baseline methods used in CAFA 3. In this case, BLAST was based on search results using the Basic Local Alignment Search Tool software against the training database [40]. A term was predicted as the highest local alignment sequence identity among all BLAST hits annotated with the term. BLAST was evaluated during CAFA 3 using the new experimental annotations accumulated during the competition (from February 2017 to November 2017). We used the same approach to evaluate our models, using the annotations files released in September 2021 to evaluate the models that we had trained with the files released on November 2018 .

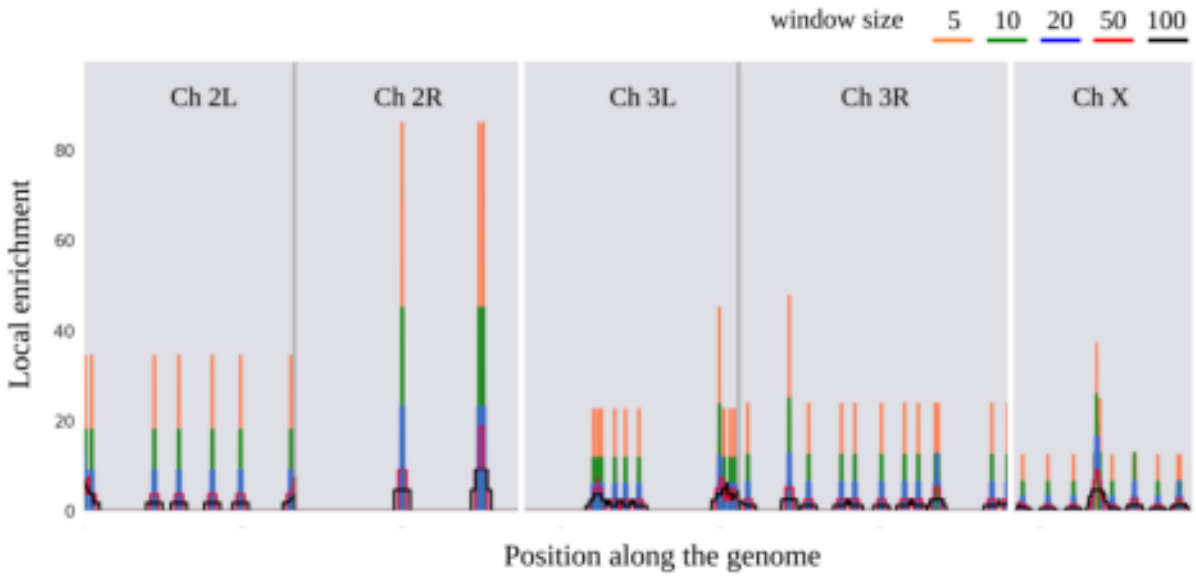
We compared the performance reached by our models with the performance of BLAST when predicting GO terms for individual species. This data is available as Supplementary files for CAFA 3 at: https://doi.org/10.6084/m9.figshare.8135393.v3 and includes performance evaluation for *H. Sapiens*, *M. musculus* and *D. melanogaster.* We compared our results with those obtained with the limited-knowledge benchmarks and under the full evaluation mode. For more details about the different CAFA evaluations modes please refer to CAFA 3, Additional file 1 [7] and CAFA2 [11]

3. RESULTS

3.1 Functional enrichment maps in five model eukaryotes

We performed Local Enrichment Analysis around each gene of a given genome considering windows of various sizes (See Methods). Local Enrichment Analysis of a given gene assess if the genes in the surroundings are annotated with any GO term more frequently than what could be expected by chance. Given a GO term, its functional enrichment map shows which regions of a genome are enriched in that GO term, considering various windows sizes. We obtained the functional enrichment map of all those GO terms associated with at least 20 genes in each of the five considered organisms. As an example, Figure 2 shows the functional enrichment map of the GO term "Golgi membrane" (GO:0000139) in the genome of *D. melanogaster*. The data to generate all the functional enrichment maps is available at: https://github.com/IIBCE-BND/gfpml-datasets/tree/master/lea

Figure 2. Functional enrichment map of the GO term "Golgi membrane" (GO:0000139) in the genome of *D. melanogaster.* There are 50 *Drosophila* genes annotated with this GO term that belongs to the Cellular Component ontology. The chromosomal position is represented in the x axis and the corresponding local enrichment at each position is shown in the y axis. Each light gray



block corresponds to a chromosome (only chromosomes 2, 3 and X are shown) and the vertical dark gray lines mark the position of the centromeres, which divide the chromosome 2 into arms 2L and 2R and chromosome 3 into arms 3L and 3R. The enrichment found using different windows is shown with the colors indicated in the figure.

3.2 Implementation of hierarchical multilabel classifiers

We trained fifteen hierarchical multilabel classifiers, one for each possible pair organism/ontology. As detailed in Methods, we randomly split each genome into two sets: **T**, that includes 80% of the genes and was used for training, and **E**, that includes the remaining 20% of the genes and was used for evaluation. Each model assigned probabilities of association between the genes of the set **E** and those GO terms associated with at least 40 genes of the set **T** and 10 genes of the set **E**. Table 1 shows, for each organism and each ontology, the number of GO terms fulfilling these conditions and for which we implemented a binary classifier.

Organism	Proetin coding genes	Ontology	Total GO terms	Considered GO terms	hPrec	hRec	hF-max
S. cerevisiae (R64)	*5,892*	BP	5,074	525	0.24	0.23	0.24
		CC	1,035	137	0.51	0.52	0.52
		MF	2,323	137	0.69	0.19	0.30
C. elegans (WBcel235)	*7,356*	BP	5,661	551	0.09	0.15	0.11
		CC	1,110	117	0.19	0.33	0.25
		MF	2,226	151	0.25	0.14	0.17
D. melanogaster (BDGP6)	*11,122*	BP	7,416	880	0.17	0.20	0.18
		CC	1,277	176	0.41	0.37	0.39
		MF	2,599	212	0.47	0.22	0.30
M. musculus (GRCm38.p6)	*20,809*	BP	15,318	1040	0.22	0.21	0.21
		CC	1,953	285	0.46	0.42	0.44
		MF	4,269	364	0.63	0.25	0.36
H. sapiens (GRCh38.p13)	*17,276*	BP	13,816	1212	0.21	0.20	0.20
		CC	1,818	338	0.44	0.42	0.43
		MF	4,244	369	0.47	0.27	0.35

Table 1. GO terms for which a binary classifier was trained and tested. The first column shows the assembly version used for each organism, the second column shows the number of protein coding genes in each genome, the third column indicates the ontology, the fourth column shows the number of GO terms associated with at least one gene for that organism and ontology and the fifth column shows the number of GO terms associated with at least 40 genes in the set T (used for training) and 10 genes in the set E (used for evaluation). These are the GO terms for which a binary classifier was trained and tested. For each organism and ontology, we implemented a hierarchical multilabel classifier combining these binary classifiers. Columns six, seven and eight show the hierarchical precision, recall and F-max reached by each of these models respectively.

3.3 Evaluation of the models

We evaluated the performance of our models using the hierarchical version of the F1 score (hF1). Figure 3 shows the hF1 reached by each trained model over the test set **E**, as well as the hF1 of the corresponding random model, as a function of the classification threshold.

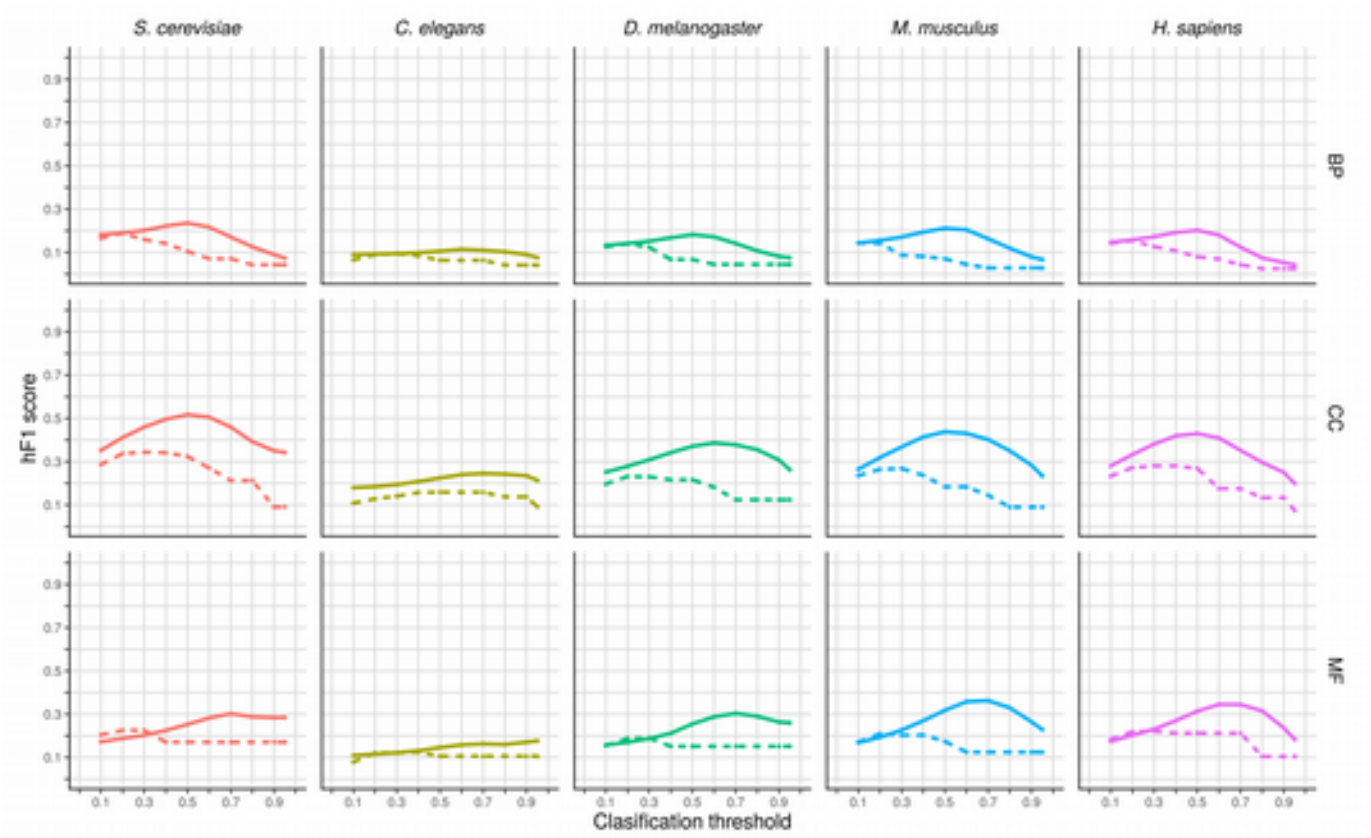


Figure 3. Hierarchical F1 over the test set for each trained and random model as a function of the classification threshold. In each plot the classification threshold, ranging from 0 to 1, is depicted in the x axis and the hF1, also ranging from 0 to 1, is depicted in the y axis. Trained models are represented by solid lines and random models by dotted lines. Each column of the panel corresponds to an organism and each row to an ontology (BP: Biological Process, CC: Cellular Component, MF: Molecular Function).

The hF-max is the highest hF1 score that the model reaches when varying the classification threshold and is a measure of the overall performance of the model. Table 1 shows the hF-max for each model along with the corresponding precision and recall.

3.4 Comparison with random models

To assess how far from randomness the linear organization of the genes along the genome with respect to its functions is, we calculated the ratio between the hF-max of the trained model and the hF-max of an equivalent random model, i.e. a model that assigns the term frequency as the prediction score for any gene (see Methods). Figures 4 and 5 show how this ratio varies with the classification threshold in each organism and ontology and Table 2 shows the max ratio between the

two models for each pair organism/ontology. The trained models consistently performed better than the random models.

Organism	Ontology	Threshold	Max Ratio
S. cerevisiae	BP	0,60	3,06
	CC	0,90	3,86
	MF	0,70	1,77
C. elegans	BP	0,80	2,49
	CC	0,95	2,33
	MF	0,95	1,68
D. melanogaster	BP	0,60	3,87
	CC	0,70	3,05
	MF	0,70	2,01
M. musculus	BP	0,70	5,83
	CC	0,80	3,90
	MF	0,70	2,92
H. sapiens	BP	0,80	3,03
	CC	0,90	2,67
	MF	0,80	3,02

Table 2. Max ratio between the hF1 reached by the trained model and the corresponding hF1 reached by the random model over the set E for each possible pair organism/ontology.

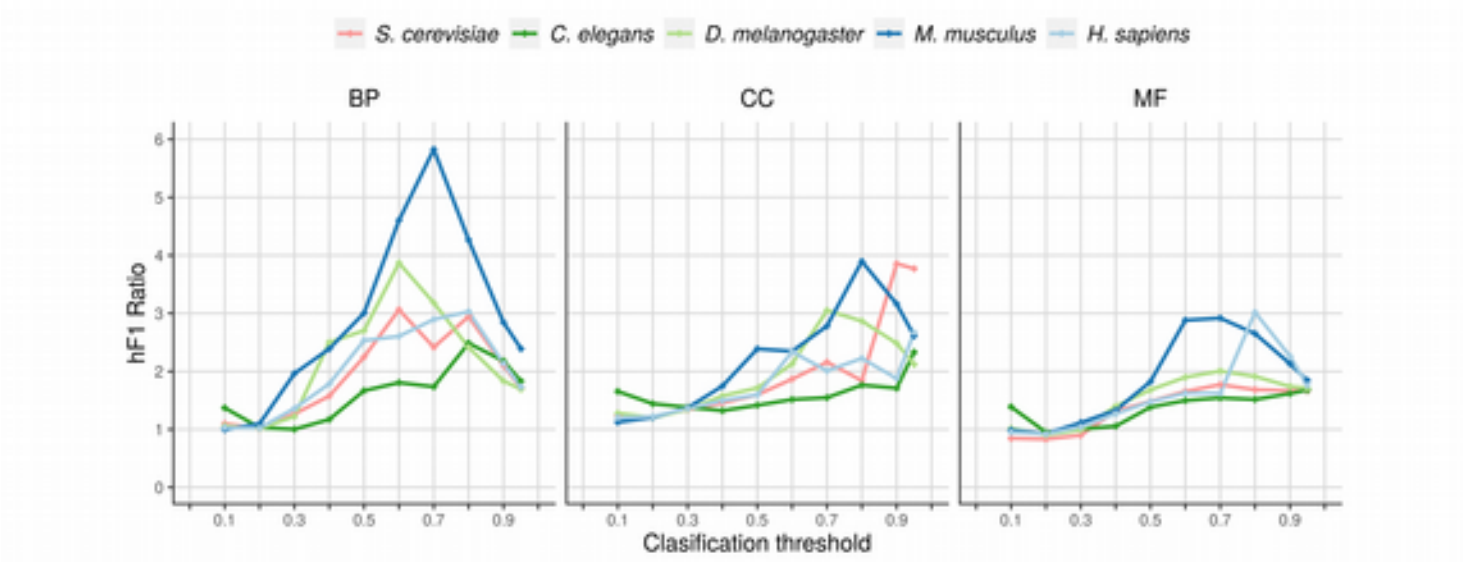


Figure 4. *Ratio between the hF1 score of the trained model and the hF1 score of the corresponding random model as a function of the classification threshold. Each graph shows the results for a given ontology, representing each organism with a different color.*

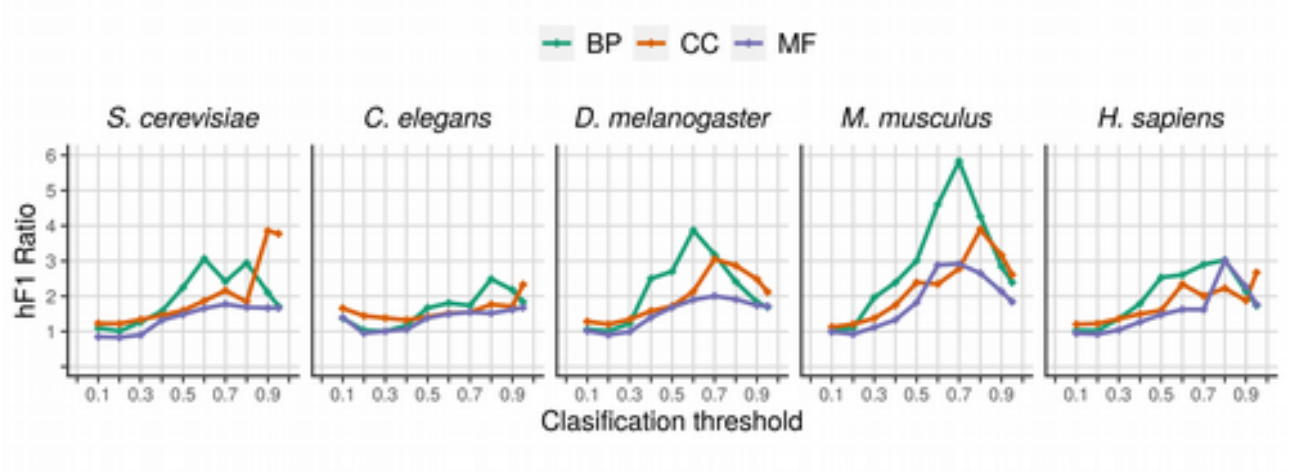


Figure 5. Ratio between the hF1 score of the trained model and the hF1 score of the corresponding random model as a function of the classification threshold. Each graph shows the results for a given organism, representing each ontology with a different color.

3.5 Comparison to one of the CAFA baseline methods

As a complementary way to evaluate our models, we also compared their performance with the performance reached by BLAST, one of the baseline methods used in CAFA 3 (see Methods). *To do so, we used the same approach used during CAFA competitions: we used the annotations released in September 2021 (i.e. after our predictions were generated) to evaluate the performance of the models that we had trained with the files released on November 2018. We compared the*

hFmax reached by our models with the hFmax reached by BLAST when making predictions for the same individual species (data that is only available for three of the five species we studied here: H. sapiens, M. musculus and D. melanogaster).".

With this comparison we aimed to asses if gene location alone can predict gene function with a performance comparable to that reached by sequence homology alone. We found that this is the case and Figure 6 shows the hFmax reached by the three models for each organism and ontology. Notably, for the three considered organisms, the models trained with FLAs outperforms BLAST when predicting GO terms from the Biological Process ontology. Our models also outperform BLAST when predicting GO terms from the Cellular Component ontology in *H. sapiens* and *D. melanogaster.*

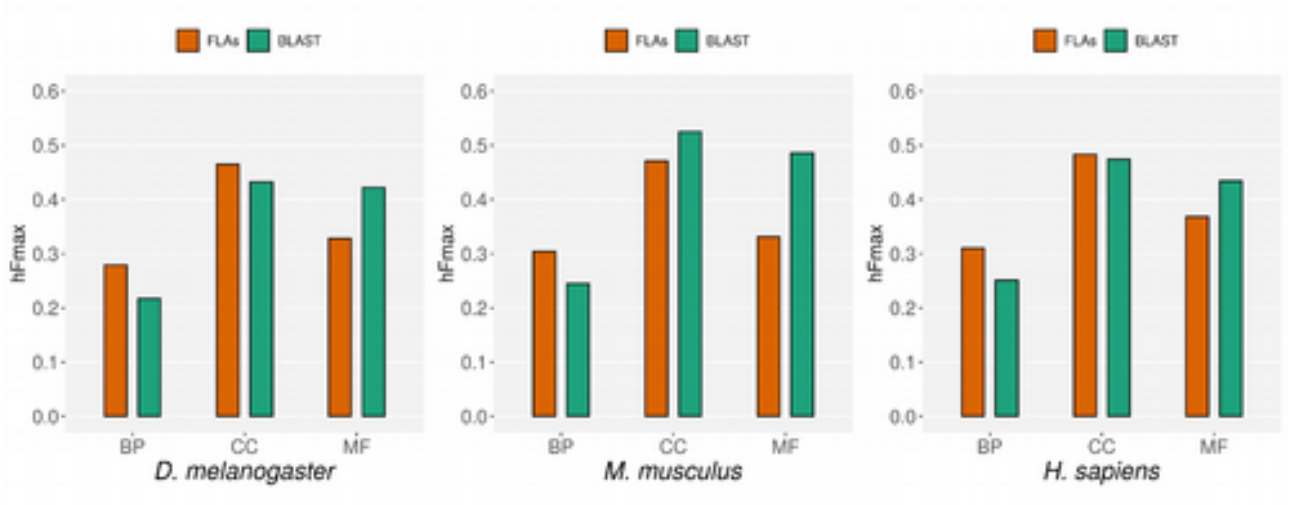


Figure 6. Comparison to one of the CAFA baseline methods. Each graph shows the hFmax of different models when predicting GO terms of the three ontologies in three organisms. In red, the hFmax of the models exclusively trained with FLAs, evaluated using the new experimental annotations accumulated from November 2018 to September 2021. In green, the hFmax of BLAST when making predictions on the same organisms and ontology as reported in CAFA 3 [7].

3.6 Prediction of new associations between genes and GO terms

In each organism, we classified the genes in the set **E** using the trained model. We obtained the probability of association between each gene in the set **E** and each GO term associated with at least 40 genes in **T** and 10 genes in **E**. We considered as new functional predictions all those associations with probabilities above the classification threshold that maximized the ratio between the hF1 score of the trained model and the hF1 score of the random model. For each gene in the set **E**, we only considered the most specific prediction within a given branch of the ontology. Figure 7

shows, for each ontology and organism, and at each depth of the ontology, the number of new predictions obtained. Because all annotations used for training were up-propagated, along each specific branch of the ontology more general GO terms were always annotated with more genes than more specific GO terms. As our predictions are based on the relative position of existing annotations, along the same branch of the ontology more predictions above the classification threshold should be expected for more general GO terms. The peaks observed in Figure 7 are a result of the better performance of our method when predicting certain branches of the ontologies.

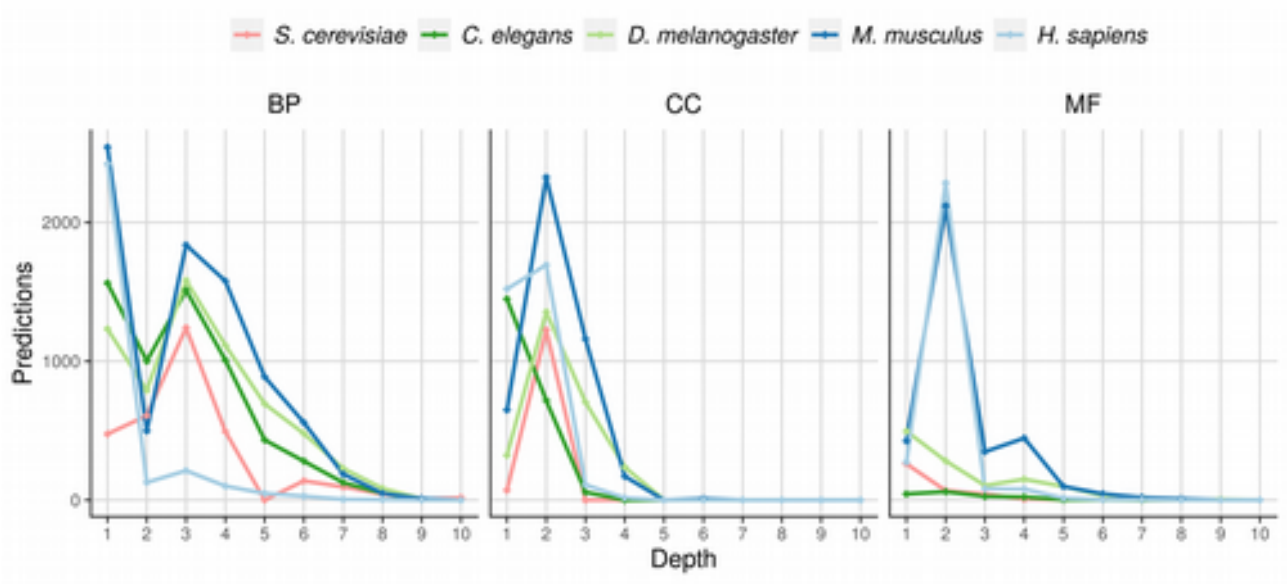


Figure 7. Predictions by depth in the ontology. Each graph corresponds to a different ontology and each organism is shown in a different color. The depth in the ontology is depicted in the x axis and the number of predicted associations above the classification threshold is depicted in the y axis.

The complete set of predicted associations with a probability above the threshold is provided as supplementary tables, with one table for each pair organism - ontology (see Supplementary Table S2 to Supplementary Table S16)

4. DISCUSSION

For the majority of the known genes, the only available information is their DNA sequence[12]. AFP based on DNA sequence similarity is a common approach, since it is known that two genes with very similar sequences probably have the same function. But the contrary is not always true. A thorough study of the correlation between similarity in protein sequence and function in yeast [13] found that, although sequence similarity can serve as a key measure in protein function prediction, the majority of the sequences of proteins annotated with the same GO term were non-similar. In general, within one branch of an ontology tree, the more specific a GO term is, the more similar the sequences of the genes annotated with that term are, but the degree of similarity is highly variable and is significant only for specific GO terms. When using orthology between genes, these methods face another limitation: the evolutionary distance of many genomes to the closest well-characterized genome. For example, only 25–50% of the proteins in any given algal genome have detectable sequence similarity to any defined domain in the Pfam database [14].

The localization of genes along the genome provides an alternative and complementary source of information that is independent of primary sequence [15]. Genomic context-based methods, including gene neighborhoods, gene-order and gene-teams based methods, make use of this information [12]. These methods rely on orthology between genes and thus are subject to the above exposed limitations. Probably because these limitations, the few examples of genomic context-based AFP in eukaryotes are limited to a small proportion of the genes of the organism being considered [29,41].

There is plenty of evidence pointing to the existence of distinctive patterns in the way in which functionally related genes distribute along eukaryotic genomes. If such patterns are biologically relevant it should be possible, at least in some cases, to predict the functions of a gene using as predictive feature its relative position with respect to other genes of known function in the same genome. As far as we know, here we have performed this task for the first time, using a new way to represent the information contained in these patterns: the Functional Landscape Arrays. This feature can be automatically extracted from any annotated genome and does not depend on orthology relations with other organisms.

Our aim was to explore the hypothesis that the functions of a gene can be predicted from its relative position with respect to other already annotated genes. For that reason, we compared the performance of our method with BLAST, one of the base-line methods used in the CAFA competitions [7] and not with any of the top performing methods of this competition nor with more sensitive methods as Blast2GO [42], the state of the art for GO-annotation based on sequence. Using

FLAs as the only predictive feature we trained a set of hierarchical multilabel classifiers that outperformed BLAST when predicting GO terms from the Biological Process ontology in *H. sapiens*, *M. musculus* and *D. melanogaster* (see Figure 6). Our models also outperformed BLAST when predicting GO terms from the Cellular Component Ontology in *H. sapiens* and *D. melanogaster*.

Our study resulted in the prediction of thousands of associations between several hundreds of GO terms and thousands of genes from five different organisms. It is thus not feasible to either validate or provide a theoretical justification in our publication for all those genes or even for a representative proportion of them. However, we hope the following examples makes a convincing argument in favor of our predictions:

- MYCT1 encodes a protein predicted to act upstream of or within hematopoietic stem cell homeostasis. Our model predicted the association between MYCT1 and the GO term "regulation of gene expression". Later on, a study published after the date of the annotation files we used to train our models, suggested that MYCT1 synergistically interact with MAX as a co-transcription factor or a component of MAX transcriptional complex, involved in enhanced apoptosis in laryngeal cancer cells [43]. The following year, another study found that MYCT1 significantly decreases the expression of miR-629-3p but increased the expression of ESRP2 in laryngeal cancer cells [44].
-Tmem132e encodes a transmembrane protein known to be involved in the posterior lateral line neuromast hair cell development. Our model had predicted the association between Tmem132e and the GO term "response to IFN-γ". A study published in 2019 included Tmem132e as one of the top genes dysregulated by Notch1 haploinsufficiency in the presence of LPS/IFN-γ [45].

All the predictions obtained with our trained classifiers are available at http://gfpml.bnd.edu.uy.

The relevance of our results stems from the fact that the performance of our models, assessed by standard metrics, shows that AFP exclusively based on features derived from the relative location of genes can be successfully performed on eukaryotic genomes. Even though, in AFP, it is common practice to integrate multiple types of information, information derived from gene location is rarely taken into account. Furthermore, according to the CAFA organizers, new improvements in gene function prediction should be expected from the incorporation of new kinds of predictive features [7]. We believe that including FLAs as predictive feature could significantly improve the performance of AFP models.

The use-case of our method is a partially annotated genome. When dealing with a novel genome with predicted genes/gene products, typically the first step is to annotate as many genes as possible based on sequence similarity. But because annotation based on sequence similarity has some drawbacks, a significant part of the genes will remain unannotated. For example, in yeast the

majority of the sequences of proteins annotated with the same GO term are non-similar [13]. Moreover, after using all other known sources of information (as phylogeny, interaction networks, etc.) to predict new annotations and after years of experimental work, the genomes of the most studied model organisms are still incompletely annotated, with thousands of genes without any annotation. We think the utility of our method is precisely to complement all other known sources of information used to predict gene function and improve annotations.

Our results are interesting from another point of view. The existence in eukaryotes of distribution patterns of functionally related genes so well defined as to allow good AFP points to levels of organization thought to be exclusive of prokaryotic genomes and its characteristic operons [46]. Diament and Tuller performed a comparative study of the organization of several genomes, analyzing the location of functionally related genes. Their results revealed that the prokaryote *Escherichia coli* exhibits a higher level of genomic organization than the eukaryote *S. cerevisiae*, as one would expect given its operon-based genomic organization. But when considering a higher order of genomic organization, analyzing the co-localization of pairs of different functional gene groups, the authors found that the genome of *S. cerevisiae* is markedly more organized than that of *E. coli*. Our results are consistent with this trend.

To estimate how far from randomness the distribution of the annotations corresponding to different ontologies and different genomes is, we used the hF-max ratio, i.e. the ratio between the hF-max reached by the trained model and the hF-max reached by an equivalent random model. Table 2 and Figure 4 show that although the relationship between the complexity of the organism and its hF-max ratio is not linear, simpler organisms reach lower hF-max ratios than more complex organisms. Figure 5 shows that, for the five considered organisms, hF-max ratio is higher for Molecular Function than for Biological Process, which in turn is higher than the ratio for Cellular Component. This result suggests that gene location has better predictive power over gene function when dealing with the Molecular Function ontology.

In sum, Functional Landscape Arrays have the potential to improve AFP, as they can be easily integrated into any model, can be automatically extracted from any annotated genome and are independent of sequence identity. To the best of our knowledge, this is the first work in which only features derived from the relative gene location of the genes within a genome are used to successfully predict gene function in eukaryotes.

Competing interests: The authors declare no competing interests.

Acknowledgments:

Funding: This work was supported by Agencia Nacional de Investigación e Innovación, Uruguay, [grant number FSDA_1_2017_1_14242]; Instituto de Investigaciones Biológicas “Clemente Estable”, MEC, Uruguay and Programa de Desarrollo de las Ciencias Básicas, Uruguay.

The experiments presented in this paper were carried out using ClusterUY (site: https://cluster.uy).

Data Availability

All data generated or analysed during this study are included in this published article (and its Supplementary Information files). The code and data used to train and evaluate the models is available at: https://github.com/IIBCE-BND/gfpml-models, https://github.com/IIBCE-BND/gfpml-tools and https://github.com/IIBCE-BND/gfpml-datasets. The data to generate all the functional enrichment maps is available at: https://github.com/IIBCE-BND/gfpml-datasets/tree/master/lea

References

1. UniProt Consortium, T. UniProt: the universal protein knowledgebase. *Nucleic Acids Res* **46**, 2699 (2018).
2. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* **25**, 25–29 (2000).
3. Zerbino, D. R. *et al.* Ensembl 2018. *Nucleic Acids Res* **46**, D754–D761 (2018).
4. Rubin, A. F. & Green, P. Expression-based segmentation of the Drosophila genome. *BMC Genomics* **14**, 812 (2013).
5. Bernardes, J. S. & Pedreira, C. E. A review of protein function prediction under machine learning perspective. *Recent Pat Biotechnol* **7**, 122–141 (2013).
6. Libbrecht, M. W. & Noble, W. S. Machine learning applications in genetics and genomics. *Nature Reviews Genetics* **16**, 321–332 (2015).
7. Zhou, N. *et al.* The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology* **20**, 244 (2019).
8. Zhao, Y. *et al.* A Literature Review of Gene Function Prediction by Modeling Gene Ontology. *Front Genet* **11**, (2020).

9. Bonetta, R. & Valentino, G. Machine learning techniques for protein function prediction. *Proteins* **88**, 397–413 (2020).
10. Radivojac, P. *et al.* A large-scale evaluation of computational protein function prediction. *Nat. Methods* **10**, 221–227 (2013).
11. Jiang, Y. *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology* **17**, 184 (2016).
12. Shehu, A., Barbará, D. & Molloy, K. A Survey of Computational Methods for Protein Function Prediction. in *Big Data Analytics in Genomics* (ed. Wong, K.-C.) 225–298 (Springer International Publishing, 2016). doi:10.1007/978-3-319-41279-5_7.
13. Duan, Z.-H., Hughes, B., Reichel, L., Perez, D. M. & Shi, T. The relationship between protein sequences and their gene ontology functions. *BMC Bioinformatics* **7**, S11 (2006).
14. Blaby-Haas, C. E. & Merchant, S. S. Comparative and Functional Algal Genomics. *Annu Rev Plant Biol* **70**, 605–638 (2019).
15. Ling, X., He, X. & Xin, D. Detecting gene clusters under evolutionary constraint in a large number of genomes. *Bioinformatics* **25**, 571–577 (2009).
16. Eisen, M. B., Spellman, P. T., Brown, P. O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* **95**, 14863–14868 (1998).
17. Niehrs, C. & Pollet, N. Synexpression groups in eukaryotes. *Nature* **402**, 483–487 (1999).
18. Cohen, B. A., Mitra, R. D., Hughes, J. D. & Church, G. M. A computational analysis of whole-genome expression data reveals chromosomal domains of gene expression. *Nat. Genet.* **26**, 183–186 (2000).
19. Boutanaev, A. M., Kalmykova, A. I., Shevelyov, Y. Y. & Nurminsky, D. I. Large clusters of co-expressed genes in the Drosophila genome. *Nature* **420**, 666–669 (2002).
20. Hurst, L. D., Williams, E. J. B. & Pál, C. Natural selection promotes the conservation of linkage of co-expressed genes. *Trends Genet.* **18**, 604–606 (2002).
21. Lee, J. M. & Sonnhammer, E. L. L. Genomic Gene Clustering Analysis of Pathways in Eukaryotes. *Genome Res* **13**, 875–882 (2003).
22. Hurst, L. D., Pal, C. & Lercher, M. J. The evolutionary dynamics of eukaryotic gene order. *Nat Rev Genet* **5**, 299–310 (2004).
23. Michalak, P. Coexpression, coregulation, and cofunctionality of neighboring genes in eukaryotic genomes. *Genomics* **91**, 243–248 (2008).
24. Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D. & Maltsev, N. The use of gene clusters to infer functional coupling. *Proc Natl Acad Sci U S A* **96**, 2896–2901 (1999).

25. Huynen, M., Snel, B., Lathe, W. & Bork, P. Predicting Protein Function by Genomic Context: Quantitative Evaluation and Qualitative Inferences. *Genome Res* **10**, 1204–1210 (2000).
26. Wolf, Y. I., Rogozin, I. B., Kondrashov, A. S. & Koonin, E. V. Genome Alignment, Evolution of Prokaryotic Genome Organization, and Prediction of Gene Function Using Genomic Context. *Genome Res.* **11**, 356–372 (2001).
27. Yanai, I., Mellor, J. C. & DeLisi, C. Identifying functional links between genes using conserved chromosomal proximity. *Trends Genet.* **18**, 176–179 (2002).
28. Zheng, Y., Roberts, R. J. & Kasif, S. Genomic functional annotation using co-evolution profiles of gene clusters. *Genome Biol.* **3**, RESEARCH0060 (2002).
29. Mihelčić, M., Šmuc, T. & Supek, F. Patterns of diverse gene functions in genomic neighborhoods predict gene function and phenotype. *Sci Rep* **9**, (2019).
30. Pazos Obregón, F. *et al.* Cluster Locator, online analysis and visualization of gene clustering. *Bioinformatics* **34**, 3377–3379 (2018).
31. Valentini, G. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Trans Comput Biol Bioinform* **8**, 832–847 (2011).
32. Boyle, E. I. *et al.* GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics* **20**, 3710–3715 (2004).
33. Tiirikka, T., Siermala, M. & Vihinen, M. Clustering of gene ontology terms in genomes. *Gene* **550**, 155–164 (2014).
34. Feng, S., Fu, P. & Zheng, W. A Hierarchical Multi-Label Classification Algorithm for Gene Function Prediction. *Algorithms* **10**, 138 (2017).
35. Feng, S., Fu, P. & Zheng, W. A hierarchical multi-label classification method based on neural networks for gene function prediction. *Biotechnology & Biotechnological Equipment* **32**, 1613–1621 (2018).
36. Breiman, L. Random Forests. *Machine Learning* **45**, 5–32 (2001).
37. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **16**, 321–357 (2002).
38. Silla, C. N. & Freitas, A. A. A survey of hierarchical classification across different application domains. *Data Min Knowl Disc* **22**, 31–72 (2011).
39. Kiritchenko, S., Matwin, S., Nock, R. & Famili, A. F. Learning and Evaluation in the Presence of Class Hierarchies: Application to Text Categorization. in *Advances in Artificial Intelligence* (eds. Lamontagne, L. & Marchand, M.) 395–406 (Springer, 2006). doi:10.1007/11766247_34.
40. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).

41. Foflonker, F. & Blaby-Haas, C. E. (ORCID:0000000215831291). Co-locality to co-functionality: Eukaryotic gene neighborhoods as a resource for function. *Molecular Biology and Evolution* (2020) doi:https://doi.org/10.1093/molbev/msaa221.
42. Conesa, A. *et al.* Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* **21**, 3674–3676 (2005).
43. Wang, H.-T. *et al.* MYCT1 represses apoptosis of laryngeal cancerous cells through the MAX/miR-181a/NPM1 pathway. *FEBS J* **286**, 3892–3908 (2019).
44. Yue, P.-J., Sun, Y.-Y., Li, Y.-H., Xu, Z.-M. & Fu, W.-N. MYCT1 inhibits the EMT and migration of laryngeal cancer cells via the SP1/miR-629-3p/ESRP2 pathway. *Cell Signal* **74**, 109709 (2020).
45. Hans, C. P. *et al.* Transcriptomics Analysis Reveals New Insights into the Roles of Notch1 Signaling on Macrophage Polarization. *Sci Rep* **9**, 7999 (2019).
46. Diament, A. & Tuller, T. Three-dimensional Genomic Organization of Genes' Function in Eukaryotes. in *Evolutionary Biology: Convergent Evolution, Evolution of Complex Traits, Concepts and Methods* (ed. Pontarotti, P.) 233–252 (Springer International Publishing, 2016). doi:10.1007/978-3-319-41324-2_14.

Author contributions

FPO conceived and supervised the project, performed analyses and wrote the manuscript. DS performed and analyzed the experiments. PS, GG, PY and RC discussed the results and corrected the manuscript. All authors approved the manuscript.

Bibliografía

[Enz, 1993] (1993). Enzyme nomenclature: Recommendations (1992) of the nomenclature committee of the international union of biochemistry and molecular biology. pp 862. academic press, san dieg. *Biochemical Education*, 21(2):102.

[Alaydie et al., 2012] Alaydie, N., Reddy, C., and Fotouhi, F. (2012). Exploiting label dependency for hierarchical multi-label classification. pages 294–305.

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

[Altschul et al., 1997] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402.

[Amit and Geman, 1997] Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588.

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature genetics*, 25(1):25–29. 10802651[pmid].

[Barabási et al., 2011] Barabási, A.-L., Gulbahce, N., and Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nature reviews. Genetics*, 12(1):56–68. 21164525[pmid].

[Bonetta and Valentino, 2019] Bonetta, R. and Valentino, G. (2019). Machine learning techniques for protein function prediction. *Proteins: Structure, Function, and Bioinformatics*, 88.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Cao et al., 2017] Cao, R., Freitas, C., Chan, L., Sun, M., Jiang, H., and Chen, Z. (2017). Prolango: Protein function prediction using neural machine translation based on a recurrent neural network. *Molecules (Basel, Switzerland)*, 22(10):1732. 29039790[pmid].

[Cerri et al., 2013] Cerri, R., Pappa, G., Carvalho, A., and Freitas, A. (2013). An extensive evaluation of decision tree-based hierarchical multilabel classification methods and performance measures. *Computational Intelligence*, 31.

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

[Chua et al., 2006] Chua, H. N., Sung, W.-K., and Wong, L. (2006). Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics*, 22(13):1623–1630.

[Clark and Radivojac, 2011] Clark, W. T. and Radivojac, P. (2011). Analysis of protein function and its prediction from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, 79(7):2086–2096.

[Consortium, 2018] Consortium, T. G. O. (2018). The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338.

[Consortium, 2020] Consortium, T. U. (2020). UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489.

[Cozzetto et al., 2016] Cozzetto, D., Minneci, F., Currant, H., and Jones, D. T. (2016). Ffpred 3: feature-based function prediction for all gene ontology domains. *Scientific Reports*, 6(1):31865.

[Dekel et al., 2005] Dekel, O., Keshet, J., and Singer, Y. (2005). An online algorithm for hierarchical phoneme classification. In *Machine Learning for Multimodal Interaction: First International Workshop*, pages 146–158. Springer LNAI 3361.

[Deng et al., 2003] Deng, M., Chen, T., and Sun, F. (2003). An integrated probabilistic model for functional prediction of proteins. In *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology*, RECOMB '03, page 95–103, New York, NY, USA. Association for Computing Machinery.

[Deng et al., 2004] Deng, M., Tu, Z., Sun, F., and Chen, T. (2004). Mapping gene ontology to proteins based on protein-protein interaction data. *Bioinformatics*, 20 6:895–902.

[Dessimoz and Škunca, 2017] Dessimoz, C. and Škunca, N. (2017). *The Gene Ontology Handbook*, volume 1446.

[Duan et al., 2006] Duan, Z.-H., Hughes, B., Reichel, L., Perez, D. M., and Shi, T. (2006). The relationship between protein sequences and their gene ontology functions. *BMC Bioinformatics*, 7(4):S11.

[Engelhardt et al., 2005] Engelhardt, B. E., Jordan, M. I., Muratore, K. E., and Brenner, S. E. (2005). Protein molecular function prediction by bayesian phylogenomics. *PLoS computational biology*, 1(5):e45–e45. 16217548[pmid].

[Fa et al., 2018] Fa, R., Cozzetto, D., Wan, C., and Jones, D. T. (2018). Predicting human protein function with multi-task deep neural networks. *bioRxiv*.

[Feng et al., 2017] Feng, S., Fu, P., and Zheng, W. (2017). A hierarchical multi-label classification algorithm for gene function prediction. *Algorithms*, 10.

[Feng et al., 2018] Feng, S., Fu, P., and Zheng, W. (2018). A hierarchical multi-label classification method based on neural networks for gene function prediction. *Biotechnology & Biotechnological Equipment*, pages 1–9.

[Friedberg and Radivojac, 2017] Friedberg, I. and Radivojac, P. (2017). *Community-Wide Evaluation of Computational Function Prediction*, pages 133–146. Springer New York, New York, NY.

[Hastie et al., 2004] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2004). The elements of statistical learning: Data mining, inference, and prediction. *Math. Intell.*, 27:83–85.

[Hawkins et al., 2009] Hawkins, T., Chitale, M., Luban, S., and Kihara, D. (2009). Pfp: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. *Proteins: Structure*, 74:566–82.

[Hennig et al., 2003] Hennig, S., Groth, D., and Lehrach, H. (2003). Automated gene ontology annotation for anonymous sequence data. *Nucleic acids research*, 31:3712–5.

[Ho, 1995] Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.

[Howe et al., 2020] Howe, K. L., Achuthan, P., Allen, J., Allen, J., Alvarez-Jarreta, J., Amode, M. R., Armean, I. M., Azov, A. G., Bennett, R., Bhai, J., Billis, K., Boddu, S., Charkhchi, M., Cummins, C., Da Rin Fioretto, L., Davidson, C., Dodiya, K., El Houdaigui, B., Fatima, R., Gall, A., Garcia Giron, C., Grego, T., Guijarro-Clarke, C., Haggerty, L., Hemrom, A., Hourlier, T., Izuogu, O. G., Juettemann, T., Kaikala, V., Kay, M., Lavidas, I., Le, T., Lemos, D., Gonzalez Martinez, J., Marugán, J. C., Maurel, T., McMahon, A. C., Mohanan, S., Moore, B., Muffato, M., Oheh, D. N., Paraschas, D., Parker, A., Parton, A., Prosovetskaia, I., Sakthivel, M. P., Salam, A. I. A., Schmitt, B. M., Schuilenburg, H., Sheppard, D., Steed, E., Szpak, M., Szuba, M., Taylor, K., Thormann, A., Threadgold, G., Walts, B., Winterbottom, A., Chakiachvili, M., Chaubal, A., De Silva, N., Flint, B., Frankish, A., Hunt, S. E., IIsley, G. R., Langridge, N., Loveland, J. E., Martin, F. J., Mudge, J. M., Morales, J., Perry, E., Ruffier, M., Tate, J., Thybert, D., Trevanion, S. J., Cunningham, F., Yates, A. D., Zerbino, D. R., and Flicek, P. (2020). Ensembl 2021. *Nucleic Acids Research*, 49(D1):D884–D891.

[Huttenhower et al., 2006] Huttenhower, C., Hibbs, M., Myers, C., and Troyanskaya, O. G. (2006). A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, 22(23):2890–2897.

[Jiang et al., 2016] Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D'Andrea, D., Lepore, R., Funk, C. S., Kahanda, I., Verspoor, K. M., Ben-Hur,

A., Koo, D. C. E., Penfold-Brown, D., Shasha, D., Youngs, N., Bonneau, R., Lin, A., Sahraeian, S. M. E., Martelli, P. L., Profiti, G., Casadio, R., Cao, R., Zhong, Z., Cheng, J., Altenhoff, A., Skunca, N., Dessimoz, C., Dogan, T., Hakala, K., Kaewphan, S., Mehryary, F., Salakoski, T., Ginter, F., Fang, H., Smithers, B., Oates, M., Gough, J., Törönen, P., Koskinen, P., Holm, L., Chen, C.-T., Hsu, W.-L., Bryson, K., Cozzetto, D., Minneci, F., Jones, D. T., Chapman, S., BKC, D., Khan, I. K., Kihara, D., Ofer, D., Rappoport, N., Stern, A., Cibrian-Uhalte, E., Denny, P., Foulger, R. E., Hieta, R., Legge, D., Lovering, R. C., Magrane, M., Melidoni, A. N., Mutowo-Meullenet, P., Pichler, K., Shypitsyna, A., Li, B., Zakeri, P., ElShal, S., Tranchevent, L.-C., Das, S., Dawson, N. L., Lee, D., Lees, J. G., Sillitoe, I., Bhat, P., Nepusz, T., Romero, A. E., Sasidharan, R., Yang, H., Paccanaro, A., Gillis, J., Sedeño-Cortés, A. E., Pavlidis, P., Feng, S., Cejuela, J. M., Goldberg, T., Hamp, T., Richter, L., Salamov, A., Gabaldon, T., Marcet-Houben, M., Supek, F., Gong, Q., Ning, W., Zhou, Y., Tian, W., Falda, M., Fontana, P., Lavezzo, E., Toppo, S., Ferrari, C., Giollo, M., Piovesan, D., Tosatto, S. C., del Pozo, A., Fernández, J. M., Maietta, P., Valencia, A., Tress, M. L., Benso, A., Di Carlo, S., Politano, G., Savino, A., Rehman, H. U., Re, M., Mesiti, M., Valentini, G., Bargsten, J. W., van Dijk, A. D. J., Gemovic, B., Glisic, S., Perovic, V., Veljkovic, V., Veljkovic, N., Almeida-e Silva, D. C., Vencio, R. Z. N., Sharan, M., Vogel, J., Kansakar, L., Zhang, S., Vucetic, S., Wang, Z., Sternberg, M. J. E., Wass, M. N., Huntley, R. P., Martin, M. J., O'Donovan, C., Robinson, P. N., Moreau, Y., Tramontano, A., Babbitt, P. C., Brenner, S. E., Linial, M., Orengo, C. A., Rost, B., Greene, C. S., Mooney, S. D., Friedberg, I., and Radivojac, P. (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, 17(1):184.

[Kanehisa et al., 2004] Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M. (2004). The KEGG resource for deciphering the genome. *Nucleic Acids Research*, 32(suppl$_1$) : $D277 - -D280$.

[Khan et al., 2003] Khan, S., Situ, G., Decker, K., and Schmidt, C. J. (2003). GoFigure: Automated Gene Ontology^{TM} annotation. *Bioinformatics*, 19(18):2484–2485.

[Kiritchenko et al., 2006] Kiritchenko, S., Matwin, S., Nock, R., and Famili, A. F. (2006). Learning and evaluation in the presence of class hierarchies: Application

to text categorization. In Lamontagne, L. and Marchand, M., editors, *Advances in Artificial Intelligence*, pages 395–406, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Kissa et al., 2015] Kissa, M., Tsatsaronis, G., and Schroeder, M. (2015). Prediction of drug gene associations via ontological profile similarity with application to drug repositioning. *Methods*, 74:71–82.

[Kulmanov and Hoehndorf, 2019] Kulmanov, M. and Hoehndorf, R. (2019). DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429.

[Kulmanov et al., 2017] Kulmanov, M., Khan, M. A., and Hoehndorf, R. (2017). DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668.

[Lan et al., 2013] Lan, L., Djuric, N., Guo, Y., and Vucetic, S. (2013). Ms-knn: protein function prediction by integrating multiple data sources. *BMC bioinformatics*, 14 Suppl 3(Suppl 3):S8–S8. 23514608[pmid].

[Lee et al., 2006] Lee, H., Tu, Z., Deng, M., Sun, F., and Chen, T. (2006). Diffusion kernel-based logistic regression models for protein function prediction. *Omics : a journal of integrative biology*, 10 1:40–55.

[Letovsky and Kasif, 2003] Letovsky, S. and Kasif, S. (2003). Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics (Oxford, England)*, 19(suppl$_1$) : $i197 - -i204$.

[Li et al., 2020] Li, J., Wang, L., Zhang, X., Liu, B., and Wang, Y. (2020). Gonet: A deep network to annotate proteins via recurrent convolution networks. *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 29–34.

[Lobley et al., 2008] Lobley, A. E., Nugent, T., Orengo, C. A., and Jones, D. T. (2008). FFPred: an integrated feature-based function prediction server for vertebrate proteomes. *Nucleic Acids Research*, 36(suppl$_2$) : $W297 - -W302$.

[Martin et al., 2004] Martin, D., Berriman, M., and Barton, G. (2004). Gotcha: A new method for prediction of protein function assessed by the annotation of seven genomes. *BMC bioinformatics*, 5:178.

[Minneci et al., 2013] Minneci, F., Piovesan, D., Cozzetto, D., and Jones, D. T. (2013). Ffpred 2.0: improved homology-independent prediction of gene ontology terms for eukaryotic protein sequences. *PloS one*, 8(5):e63754–e63754. 23717476[pmid].

[Nabieva et al., 2005] Nabieva, E., Jim, K.-C., Agarwal, A., Chazelle, B., and Singh, M. (2005). Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics (Oxford, England)*, 21 Suppl 1:i302–10.

[Nariai et al., 2007] Nariai, N., Kolaczyk, E. D., and Kasif, S. (2007). Probabilistic protein function prediction from heterogeneous genome-wide data. *PloS one*, 2(3):e337–e337. 17396164[pmid].

[Nauman et al., 2019] Nauman, M., Ur Rehman, H., Politano, G., and Benso, A. (2019). Beyond homology transfer: Deep learning for automated annotation of proteins. *Journal of Grid Computing*, 17(2):225–237.

[Nesmachnow and Iturriaga, 2019] Nesmachnow, S. and Iturriaga, S. (2019). Cluster-uy: Collaborative scientific high performance computing in uruguay. In Torres, M. and Klapp, J., editors, *Supercomputing*, pages 188–202, Cham. Springer International Publishing.

[Obregón, 2020] Obregón, F. P. (2020). *Predicción de función de genes mediante aprendizaje automático, con énfasis en el estudio de los patrones de ubicación de grupos funcionales de genes*. PhD thesis.

[Pal and Eisenberg, 2005] Pal, D. and Eisenberg, D. (2005). Inference of protein function from protein structure. 13(7):121–130.

[Pazos and Sternberg, 2004] Pazos, F. and Sternberg, M. J. E. (2004). Automated prediction of protein function and detection of functional sites from structure. *Proceedings of the National Academy of Sciences*, 101(41):14754–14759.

[Pazos Obregón et al., 2015] Pazos Obregón, F., Papalardo, C., Castro, S., Guerberoff, G., and Cantera, R. (2015). Putative synaptic genes defined from a drosophila whole body developmental transcriptome by a machine learning approach. *BMC genomics*, 16:694.

[Pazos Obregón et al., 2018] Pazos Obregón, F., Soto, P., Lavín, J. L., Cortázar, A. R., Barrio, R., Aransay, A. M., and Cantera, R. (2018). Cluster Locator, online analysis and visualization of gene clustering. *Bioinformatics*, 34(19):3377–3379.

[Pearson and Lipman, 1988] Pearson, W. and Lipman, D. (1988). Pearson wr, lipman dj. improved tools for biological sequence comparison. proc natl acad sci usa 85: 2444-2448. *Proceedings of the National Academy of Sciences of the United States of America*, 85:2444–8.

[Radivojac et al., 2013] Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., Pandey, G., Yunes, J. M., Talwalkar, A. S., Repo, S., Souza, M. L., Piovesan, D., Casadio, R., Wang, Z., Cheng, J., Fang, H., Gough, J., Koskinen, P., Törönen, P., Nokso-Koivisto, J., Holm, L., Cozzetto, D., Buchan, D. W. A., Bryson, K., Jones, D. T., Limaye, B., Inamdar, H., Datta, A., Manjari, S. K., Joshi, R., Chitale, M., Kihara, D., Lisewski, A. M., Erdin, S., Venner, E., Lichtarge, O., Rentzsch, R., Yang, H., Romero, A. E., Bhat, P., Paccanaro, A., Hamp, T., Kaßner, R., Seemayer, S., Vicedo, E., Schaefer, C., Achten, D., Auer, F., Boehm, A., Braun, T., Hecht, M., Heron, M., Hönigschmid, P., Hopf, T. A., Kaufmann, S., Kiening, M., Krompass, D., Landerer, C., Mahlich, Y., Roos, M., Björne, J., Salakoski, T., Wong, A., Shatkay, H., Gatzmann, F., Sommer, I., Wass, M. N., Sternberg, M. J. E., Škunca, N., Supek, F., Bošnjak, M., Panov, P., Džeroski, S., Šmuc, T., Kourmpetis, Y. A. I., van Dijk, A. D. J., Braak, C. J. F. t., Zhou, Y., Gong, Q., Dong, X., Tian, W., Falda, M., Fontana, P., Lavezzo, E., Di Camillo, B., Toppo, S., Lan, L., Djuric, N., Guo, Y., Vucetic, S., Bairoch, A., Linial, M., Babbitt, P. C., Brenner, S. E., Orengo, C., Rost, B., Mooney, S. D., and Friedberg, I. (2013). A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227.

[Rhee et al., 2008] Rhee, S., Wood, V., Dolinski, K., and Draghici, S. (2008). Use and misuse of the gene ontology annotations. *Nature reviews. Genetics*, 9:509–15.

[Rousu et al., 2006] Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classi?cation models. *Journal of Machine Learning Research*, 7:1601–1626.

[Ruepp et al., 2004] Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M.,

and Mewes, H.-W. (2004). The funcat, a functional annotation scheme for systematic classification of proteins from whole genomesrueppnucleic acids res200432185539554510.1093/nar/gkh89452430215486203. *Nucleic acids research*, 32:5539–45.

[Shehu et al., 2016] Shehu, A., Barbara, D., and Molloy, K. (2016). *A Survey of Computational Methods for Protein Function Prediction*, pages 225–298.

[Silla and Freitas, 2009] Silla, C. and Freitas, A. (2009). Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. pages 3499 – 3504.

[Silla and Freitas, 2011] Silla, C. N. and Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

[Silla Jr. and Freitas, 2009] Silla Jr., C. N. and Freitas, A. A. (2009). A global-model naive bayes approach to the hierarchical prediction of protein functions. In *2009 Ninth IEEE International Conference on Data Mining*, 2009 Ninth IEEE International Conference on Data Mining, pages 992–997.

[Smith et al., 2007] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R. H., Shah, N., Whetzel, P. L., Lewis, S., and Consortium, T. O. (2007). The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251–1255.

[Sureyya Rifaioglu et al., 2019] Sureyya Rifaioglu, A., Doğan, T., Jesus Martin, M., Cetin-Atalay, R., and Atalay, V. (2019). Deepred: Automated protein function prediction with multi-task feed-forward deep neural networks. *Scientific Reports*, 9(1):7344.

[Szalkai and Grolmusz, 2018] Szalkai, B. and Grolmusz, V. (2018). SECLAF: a webserver and deep neural network design tool for hierarchical biological sequence classification. *Bioinformatics*, 34(14):2487–2489.

[Thomas et al., 2012] Thomas, P. D., Wood, V., Mungall, C. J., Lewis, S. E., Blake, J. A., and Consortium, G. O. (2012). On the use of gene ontology annotations to

assess functional similarity among orthologs and paralogs: A short report. *PLoS computational biology*, 8(2):e1002386–e1002386. 22359495[pmid].

[Tiirikka et al., 2014] Tiirikka, T., Siermala, M., and Vihinen, M. (2014). Clustering of gene ontology terms in genomes. *Gene*, 550.

[Törönen et al., 2018] Törönen, P., Medlar, A., and Holm, L. (2018). Pannzer2: a rapid functional annotation web server. *Nucleic acids research*, 46(W1):W84–W88. 29741643[pmid].

[Vateekul et al., 2014] Vateekul, P., Kubat, M., and Sarinnapakorn, K. (2014). Hierarchical multi-label classification with svms: A case study in gene function prediction. *Intelligent Data Analysis*, 18:717–738.

[Vinayagam et al., 2006] Vinayagam, A., del Val, C., Schubert, F., Eils, R., Glatting, K.-H., Suhai, S., and König, R. (2006). Gopet: A tool for automated predictions of gene ontology terms. *BMC Bioinformatics*, 7(1):161.

[Vinayagam et al., 2004] Vinayagam, A., König, R., Moormann, J., Schubert, F., Eils, R., Glatting, K.-H., and Suhai, S. (2004). Applying support vector machines for gene ontology based gene function prediction. *BMC Bioinformatics*, 5(1):116.

[Xuan et al., 2019] Xuan, P., Sun, C., Zhang, T., Ye, Y., Shen, T., and Dong, Y. (2019). Gradient boosting decision tree-based method for predicting interactions between target genes and drugs. *Frontiers in Genetics*, 10:459.

[Zehetner, 2003] Zehetner, G. (2003). Ontoblast function: From sequence similarities directly to potential functional annotations by ontology terms. *Nucleic acids research*, 31(13):3799–3803. 12824422[pmid].

[Zeng et al., 2015] Zeng, X., Zhang, X., and Zou, Q. (2015). Integrative approaches for predicting microRNA function and prioritizing disease-related microRNA using biological interaction networks. *Briefings in Bioinformatics*, 17(2):193–203.

[Zhang et al., 2020] Zhang, F., Song, H., Zeng, M., Wu, F., Li, Y., Pan, Y., and Li, M. (2020). A deep learning framework for gene ontology annotations with sequence - and network-based information. *IEEE/ACM transactions on computational biology and bioinformatics*.

[Zhang et al., 2019] Zhang, J., Zhang, Z., Chen, Z., and Deng, L. (2019). Integrating multiple heterogeneous networks for novel lncrna-disease association inference. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 16(2):396–406.

[Zhou et al., 2019] Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., Lewis, K. A., Georghiou, G., Nguyen, H. N., Hamid, M. N., Davis, L., Dogan, T., Atalay, V., Rifaioglu, A. S., Dalkıran, A., Cetin Atalay, R., Zhang, C., Hurto, R. L., Freddolino, P. L., Zhang, Y., Bhat, P., Supek, F., Fernández, J. M., Gemovic, B., Perovic, V. R., Davidović, R. S., Sumonja, N., Veljkovic, N., Asgari, E., Mofrad, M. R., Profiti, G., Savojardo, C., Martelli, P. L., Casadio, R., Boecker, F., Schoof, H., Kahanda, I., Thurlby, N., McHardy, A. C., Renaux, A., Saidi, R., Gough, J., Freitas, A. A., Antczak, M., Fabris, F., Wass, M. N., Hou, J., Cheng, J., Wang, Z., Romero, A. E., Paccanaro, A., Yang, H., Goldberg, T., Zhao, C., Holm, L., Törönen, P., Medlar, A. J., Zosa, E., Borukhov, I., Novikov, I., Wilkins, A., Lichtarge, O., Chi, P.-H., Tseng, W.-C., Linial, M., Rose, P. W., Dessimoz, C., Vidulin, V., Dzeroski, S., Sillitoe, I., Das, S., Lees, J. G., Jones, D. T., Wan, C., Cozzetto, D., Fa, R., Torres, M., Warwick Vesztrocy, A., Rodriguez, J. M., Tress, M. L., Frasca, M., Notaro, M., Grossi, G., Petrini, A., Re, M., Valentini, G., Mesiti, M., Roche, D. B., Reeb, J., Ritchie, D. W., Aridhi, S., Alborzi, S. Z., Devignes, M.-D., Koo, D. C. E., Bonneau, R., Gligorijević, V., Barot, M., Fang, H., Toppo, S., Lavezzo, E., Falda, M., Berselli, M., Tosatto, S. C., Carraro, M., Piovesan, D., Ur Rehman, H., Mao, Q., Zhang, S., Vucetic, S., Black, G. S., Jo, D., Suh, E., Dayton, J. B., Larsen, D. J., Omdahl, A. R., McGuffin, L. J., Brackenridge, D. A., Babbitt, P. C., Yunes, J. M., Fontana, P., Zhang, F., Zhu, S., You, R., Zhang, Z., Dai, S., Yao, S., Tian, W., Cao, R., Chandler, C., Amezola, M., Johnson, D., Chang, J.-M., Liao, W.-H., Liu, Y.-W., Pascarelli, S., Frank, Y., Hoehndorf, R., Kulmanov, M., Boudellioua, I., Politano, G., Di Carlo, S., Benso, A., Hakala, K., Ginter, F., Mehryary, F., Kaewphan, S., Björne, J., Moen, H., Tolvanen, M. E., Salakoski, T., Kihara, D., Jain, A., Šmuc, T., Altenhoff, A., Ben-Hur, A., Rost, B., Brenner, S. E., Orengo, C. A., Jeffery, C. J., Bosco, G., Hogan, D. A., Martin, M. J., O'Donovan, C., Mooney, S. D., Greene, C. S., Radivojac, P., and Friedberg, I. (2019). The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, 20(1):244.