



Aceleración del proceso de selección de características en entornos Big Data. Aplicación en biomarcadores oncológicos

Genaro Camele

Director: Dr. Hasperué, Waldo

Facultad de informática
Universidad Nacional de La Plata

Julio 2024

Agradecimientos

De todas las páginas que tiene la tesis, me gustaría tomarme solo dos para escribir de manera informal y sincera. Sería un robo no agradecerle a los autores de la mitad de esta tesis. Dichos autores se dividen en tres grupos:

Grandes de la informática, bioinformática y medicina. Trabajaron conmigo, aprendimos y me enseñaron muchas cosas, algunas importantes, y otras que tienen que ver con la tesis. Formaron parte de muchos desarrollos y conceptos de este trabajo y no puedo hacer más que hacerles honor agradeciéndoles con nombre y apellido (sin orden específico):

- Waldo Hasperué: por la paciencia, la experiencia, la dirección y la compañía durante tantos años. Siempre supo entender los problemas y guiar con consejos espectaculares. Jamás le faltó la simpleza (que tanto admiro) para avanzar en todas las cuestiones que se nos presentaron. Sin dudas voy a extrañar las charlas durante los almuerzos, los chistes y el chipá espectacular que prepara para el cumple.
- Matías Butti por el asesoramiento profesional en biología y bioinformática. Llevamos más de 5 años trabajando juntos, en todos ellos aprendí cosas nuevas, siempre con el foco en generar ciencia de impacto, un objetivo que recompensa tanto como cuesta. En el proceso formamos grupos de personas increíbles, batallamos presentaciones, ganamos concursos, y publicamos software y artículos... Y todo eso sin morir en el proceso! Gracias por aguantarme.
- Los chicos del box: Franco Ronchetti y Facu Quiroga, que nunca sé cuál es cuál pero fueron los primeros en recibirme en el LIDI y guiarme en el desarrollo del detector de ~~corrimos~~ peatones durante mi tesis de grado. Laura Lanzarini, César Estrebou, Pedro Dal Bianco, Gastón Ríos, Santi Ponte, (Ing.) Cacho, Juan Seery, Franco "2 (0-indexed)" Suelgaray. Todas personas de lo mejor, que hicieron súper placentero tanto el tiempo en el laboratorio como los almuerzos hablando de la singularidad. No solo aprendí muchísimo con ustedes, sino que estuvieron ahí para darme feedback de los artículos, las presentaciones, y para responderme (con infinita paciencia) las infinitas consultas que me surgieron a lo largo de la carrera. Son una masa!

- Los chicos de OmicsDataScience: Agustín Marraco, Sergio Calderón, Nicolás Di Giacomo, Juan Herrera, Seba Menazzi, Martín Abba, Mauri Brunner, Ramiro Lasorsa, Franco Bebczuk y Julián Muhlberger. 100% seguro que algún apellido escribí mal, pero no puedo dejar de agradecer a todos los cráneos que estuvieron detrás de Modulector, BioAPI y Multiomix, esta tesis es tan suya como mía.

Para el segundo grupo voy a omitir los nombres, porque saben quienes son y no necesitan que los escriba en ningún lado. Son la familia, los amigos y los mentores que tuve y tengo al lado, aunque la distinción no tiene sentido porque los tres tipos de vínculos son lo mismo. Tengo la suerte de contar en este grupo a muchos nombres del grupo anterior.

El tercer (y más importante) grupo tiene un solo integrante. Mis agradecimientos al clúster! Podrá detenerse en cualquier momento, apagarse por los cortes de luz en verano, perder los datos por fallas en el disco y hacerme ir a encenderlo durante la pandemia porque se agotó la pila del mother... Pero me dio un doctorado y sigue siendo mejor que programar en Java.

A los tres grupos (por igual), gracias!

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	4
1.4. Contribuciones derivadas de esta tesis	5
1.4.1. Concursos, honores y menciones	5
1.4.2. Publicaciones en revistas internacionales	6
1.4.3. Publicaciones en revistas nacionales	6
1.4.4. Publicaciones en congresos y workshops	6
1.4.5. Formación de recursos humanos	7
1.4.6. Desarrollo de herramientas	8
1.5. Organización del documento	9
2. Medicina de precisión	11
2.1. Biología del cáncer	11
2.2. Blancos terapéuticos	14
2.3. Descubrimiento de reguladores de expresión	15
2.4. Biomarcadores	16
2.4.1. Aplicaciones en la bioinformática	16
2.4.2. Análisis de supervivencia	17
2.5. Evaluación de biomarcadores	23
3. Selección de características	25
3.1. Motivación	26
3.2. Blind Search	28
3.3. Regresión de Cox penalizada	28
3.4. Metaheurísticas	29
3.4.1. Binary Black Hole	30

3.4.2.	Algoritmos genéticos	32
3.4.3.	Binary Particle Swarm Optimization	33
3.5.	Trabajo previo	33
3.6.	Ejecución distribuida de metaheurísticas	36
4.	Multiomix	39
4.1.	Descubrimiento de reguladores de expresión	39
4.2.	Identificación de biomarcadores	42
4.2.1.	Modelos entrenados	42
4.2.2.	Validaciones estadísticas	43
4.2.3.	Inferencia	43
4.2.4.	Multiomix AWS-EMR	43
4.3.	Abstracción en la obtención de datos	44
4.3.1.	Modulector	44
4.3.2.	BioAPI	45
4.3.3.	Datos subidos por el usuario	45
4.3.4.	cBioPortal	46
4.4.	Democratización de la tecnología	48
4.5.	Dificultades técnicas solventadas	48
5.	Optimización de metaheurísticas en Spark	51
5.1.	Apache Spark	52
5.2.	Balance de carga	53
5.3.	Estrategias de balance de carga propuestas	53
5.3.1.	Modelo de predicción del tiempo de ejecución de tareas	55
5.3.2.	Estrategia "Equally Distributed"	57
5.3.3.	Estrategia "Distribution Based on Predictions"	59
5.3.4.	Estrategia "Predictive Execution Load Algorithm with Delay Optimization"	62
5.3.5.	Generalización y aplicación del framework	66
6.	Experimentación	69
6.1.	Hardware y software	69
6.2.	Mediciones de tiempos y métricas	69
6.3.	Evaluación de las estrategias de balance de carga	76
6.3.1.	Simulador de distribución de tareas	76
6.3.2.	Experimentos	77

6.3.3.	Conjuntos de datos	80
6.3.4.	Metaheurísticas, modelos y métricas	80
6.3.5.	Estrategias de balance de carga	81
6.3.6.	Parámetros de PELADO y simulación	81
6.3.7.	Resultados Experimento 1: validación sobre el simulador	82
6.3.8.	Resultados Experimento 2: validación sobre Apache Spark	87
7.	Conclusión y trabajo a futuro	91
7.1.	Conclusiones generales	91
7.2.	Líneas de trabajo futuras	93
	Bibliografía	95
	Apéndice A. Medicina de precisión	107
A.1.	ADN	107
A.2.	Reguladores de expresión	108
A.2.1.	miRNA	108
A.2.2.	Copy Number Alteration/Variation (CNA o CNV)	109
A.2.3.	Metilación de ADN	111
A.3.	Pathways	112
A.4.	Blancos terapéuticos	115
A.5.	Métodos de correlación	116
A.5.1.	Pearson	116
A.5.2.	Spearman	116
A.5.3.	Kendall	117
A.6.	Ajuste de p-valor	117
A.6.1.	Bonferroni	117
A.6.2.	Benjamini-Hochberg	118
A.6.3.	Benjamini-Yekutieli	119
	Apéndice B. Multiomix	121
B.1.	Proceso de creación de un análisis de correlación	121
B.2.	Detalles del análisis de correlación	123
B.2.1.	Propiedades estadísticas de la combinación GEM-gen	123
B.2.2.	Gráfico de correlación	124
B.2.3.	Información de interacciones miRNA-gen	126
B.2.4.	Patologías y drogas asociadas a miRNA	127

B.2.5. Supuestos estadísticos	129
B.2.6. Gráfico de análisis de supervivencia	129
B.3. Proceso de creación de un biomarcador	130
B.4. Detalles del biomarcador	135
B.4.1. Detalles de las moléculas del biomarcador	136
B.4.2. Redes de asociaciones de genes	137
B.4.3. Información de Gene Ontology	138
B.4.4. Información adicional	143
B.5. Modelos entrenados	143
B.6. Validaciones estadísticas	144
B.6.1. Features más significativos	145
B.6.2. Curvas Kaplan-Meier	146
B.6.3. Heatmap	147
B.7. Inferencia	148
B.8. Abstracción en la obtención de datos	150
B.8.1. Modulector: bases de datos	150
B.8.2. Modulector: servicios	152
B.8.3. BioAPI: bases de datos	157
B.8.4. BioAPI: servicios	159

Capítulo 1

Introducción

1.1. Contexto

El contexto en el que se desarrolló la presente tesis doctoral se enmarca en el ámbito de la bioinformática y el análisis de datos biomédicos. A continuación, se detallan los aspectos más relevantes.

En primer lugar, la tesis surgió en el marco del desarrollo de dos plataformas previas: Bioplat [17] y Multiomics. La plataforma Bioplat tenía como objetivo acelerar la identificación de biomarcadores con poder pronóstico y predictivo en el cáncer (a lo largo del documento, se usará el término *biomarcador* para hacer referencia tanto a aquellos que sirven para pronóstico como para predicción). Por otro lado, Multiomics se enfocaba en la aceleración de hipótesis de blancos terapéuticos a partir de análisis de correlación entre genes y reguladores de expresión.

Sin embargo, ambas plataformas presentaban ciertas limitaciones, la principal de ellas era su carácter local, ya que eran sistemas instalables en un equipo específico, sin acceso desde Internet ni la posibilidad de contar con múltiples usuarios simultáneos. Asimismo, al tratarse de sistemas separados, no existía una integración de las bases de datos de los usuarios en un único sitio.

A partir de este contexto se planteó el objetivo principal de esta tesis: desarrollar un framework para la aceleración de metaheurísticas en entornos distribuidos utilizando Spark para optimizar la selección de características en la identificación de biomarcadores oncológicos. Estas técnicas fueron desarrolladas en una nueva plataforma llamada Multiomix que complementa las funcionalidades y objetivos de Bioplat y Multiomics, solucionando a su vez, sus limitaciones. La nueva plataforma se disponibiliza como Software as a Service (SaaS) accesible desde Internet, permitiendo a múltiples usuarios trabajar de manera simultánea,

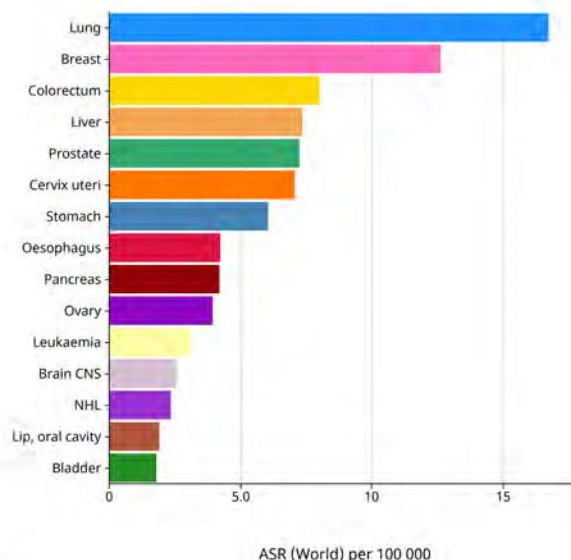
brindando a su vez con capacidades de cómputo avanzadas al alcance de la comunidad científica.

Frente a la problemática de ofrecer funcionalidades que hacen uso de grandes cantidades de datos o son computacionalmente costosas, esta tesis aporta un framework para la aceleración de la ejecución de metaheurísticas en entornos distribuidos durante el proceso de selección de características. Al distribuir las tareas entre los nodos de un clúster se busca optimizar dichos procesos y ofrecer los mejores tiempos de ejecución posibles para las investigaciones de los usuarios de la plataforma. Potenciando así el análisis de datos biomédicos y facilitando el descubrimiento de biomarcadores en el ámbito del cáncer.

1.2. Motivación

El cáncer es una enfermedad que se caracteriza por la proliferación celular descontrolada y la diseminación de células anormales en el cuerpo. Normalmente, las células del cuerpo crecen y se dividen de manera regulada para mantener un equilibrio y reemplazar aquellas que envejecen o se dañan. Sin embargo, en el caso del cáncer, este proceso se descontrola y las células comienzan a dividirse de forma acelerada e incontrolada, formando masas o tumores que pueden invadir y dañar los tejidos circundantes.

Age-Standardized Rate (World) per 100 000, Mortality, Both sexes, in 2022
Continents
(Top 15 cancer sites)



Cancer TODAY | IARC - <https://gco.iarc.who.int/today>
Data version : Globocan 2022
© All Rights Reserved 2024

ASR (World) per 100 000

International Agency
for Research on Cancer
World Health
Organization

Figura 1.1 Casos de muerte por localización tumoral en el mundo. Fuente: <https://gco.iarc.fr>

El cáncer puede originarse en prácticamente cualquier parte del cuerpo y cada tipo de cáncer se clasifica de acuerdo con el tejido u órgano en el que se origina. Algunos de los tipos más comunes son el cáncer de pulmón, de mama, de colon, de próstata y de piel, entre otros.

Las causas del cáncer son diversas y pueden estar relacionadas con factores genéticos, ambientales o hábitos de vida poco saludables, como el tabaquismo, la exposición a sustancias carcinógenas, la radiación ultravioleta excesiva, entre otros. Aunque el cáncer puede afectar a personas de cualquier edad, el riesgo aumenta con la edad debido a la acumulación de mutaciones genéticas a lo largo de la vida.

El informe de 2022 del IARC (Agencia Internacional para la Investigación sobre el Cáncer) muestra que hubo 20 millones de nuevos casos de cáncer y 9,7 millones de muertes por cáncer en todo el mundo. Los tipos de cáncer más comunes fueron el cáncer de pulmón y el cáncer de mama (Figura 1.1).

Se proyecta que los casos de cáncer aumentarán a 35 millones para el año 2050, particularmente en los países con índices de desarrollo humano bajos y medios (Figura 1.2), resaltando la necesidad de mejorar la infraestructura de atención médica y el acceso a servicios costo-efectivos para el tratamiento de pacientes oncológicos.

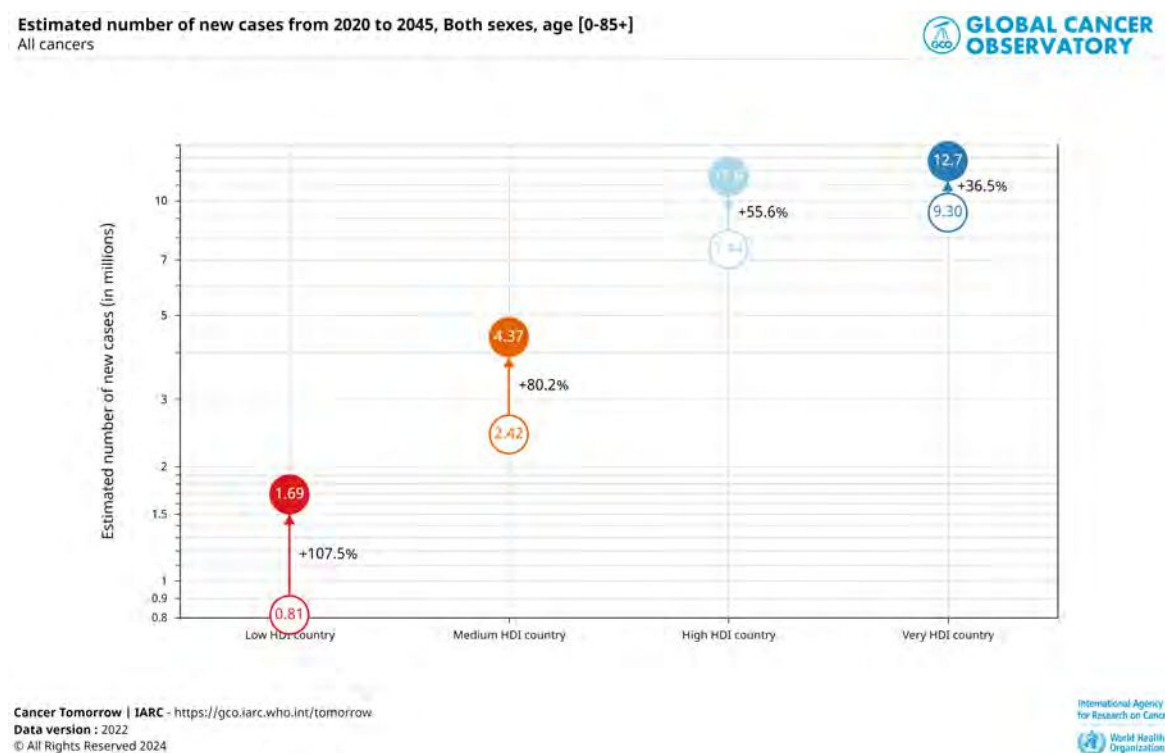


Figura 1.2 Incremento de incidencias de cáncer por nivel de ingresos estimados para el año 2050. Fuente: <https://gco.iarc.fr>

Los estudios genéticos y epigenéticos resultan cruciales para la detección temprana, el seguimiento y el tratamiento del cáncer debido a su capacidad para identificar alteraciones moleculares específicas que conducen al desarrollo y progresión de esta enfermedad. Los análisis genéticos permiten detectar mutaciones heredadas o adquiridas en genes clave que regulan el crecimiento y división celular, lo que puede alertar sobre un mayor riesgo de cáncer o incluso diagnosticar la enfermedad en etapas tempranas. Asimismo, los patrones epigenéticos anormales, como la metilación del ADN, pueden servir como biomarcadores para el diagnóstico precoz. Durante el tratamiento, el monitoreo de cambios genéticos y epigenéticos puede brindar información valiosa sobre la respuesta a la terapia y guiar la selección de fármacos más efectivos y personalizados para cada paciente.

Sin embargo, procesar volúmenes de información de gran tamaño o alta dimensionalidad sería imposible sin herramientas computacionales avanzadas, como el uso de técnicas de minería de datos, técnicas de procesamiento paralelo y distribuido, algoritmos de machine learning, entre otros. Estos, en su conjunto, permiten identificar patrones y alteraciones moleculares, nuevas asociaciones, biomarcadores y comprender mejor la enfermedad, contribuyendo al desarrollo de estrategias terapéuticas innovadoras. El carácter técnico de todos los métodos y tecnologías involucrados agrandan la brecha entre la informática y la medicina, dificultando el avance de la ciencia en la lucha contra estas patologías.

Por estas razones es crucial desarrollar soluciones avanzadas, libres y accesibles para toda la comunidad científica. Así, al democratizar estas herramientas y fomentar la colaboración, se aprovecharán óptimamente los algoritmos para un abordaje más preciso y personalizado, tanto del cáncer como de cualquier enfermedad.

Multiomix, no solo disponibiliza algoritmos de bioinformática para que sean utilizadas por la comunidad científica sin costo ni conocimiento técnico experto, sino también que acelera dichos algoritmos a través del cómputo distribuido inteligente.

1.3. Objetivos

El presente trabajo tiene como objetivo principal la implementación de un framework de distribución de metaheurísticas para la optimización de algoritmos de selección de características, permitiendo la aceleración en la identificación de biomarcadores oncológicos. Estas estrategias permiten distribuir el cómputo en un clúster de Apache Spark, aprovechando de manera más eficiente los recursos y ofreciendo tiempos de respuestas más cortos.

Además, todo el desarrollo realizado se pone a disposición a través de una plataforma bioinformática llamada Multiomix cuyo fin es el de democratizar el acceso a las herramientas bioinformáticas y los algoritmos de primer nivel para la comunidad científica. Dicha

plataforma pone a disposición técnicas para la identificación de mecanismos de regulación de expresión génica, que sirven como base para el descubrimiento de nuevos blancos terapéuticos, algoritmos de análisis de supervivencia y de selección de características para la aceleración de identificación de biomarcadores, entre otras.

Abordar estas problemáticas aporta soluciones hacia un mundo donde el diagnóstico y tratamiento del cáncer pueden ser menos costosos, más precisos y personalizados. Mejorando así la calidad de vida de los pacientes y aumentando sus posibilidades de supervivencia.

Los objetivos específicos son:

- Analizar diferentes algoritmos de selección de características para su uso en la identificación de biomarcadores oncológicos.
- Medir tiempos de ejecución de diferentes algoritmos de machine learning con el propósito de armar modelos que predigan los tiempos de ejecución de dichos algoritmos.
- Implementar estrategias de balance de carga inteligentes para la distribución de cómputo en un clúster de Apache Spark, minimizando el tiempo de ejecución total de los algoritmos de selección de características.
- Desarrollar herramientas que pongan a libre disposición todos los algoritmos y técnicas estudiadas para su uso por parte de la comunidad científica, democratizando así el acceso de la ciencia y tecnología.
- Presentar la comparación entre resultados obtenidos con los algoritmos implementados y las estrategias de distribución de tareas propuestas.

1.4. Contribuciones derivadas de esta tesis

1.4.1. Concursos, honores y menciones

- Mención especial por la presentación en el *Hackaton internacional 2024* organizado por la compañía The Hive de la Universidad Austral junto con la Universidad de Harvard.
- Premio estímulo otorgado al artículo "Multiomix: a cloud-based platform to infer cancer genomic and epigenomic events associated with gene expression modulation" [22], organizado por Facultad de Informática de la Universidad Nacional de La Plata, el día 1 de junio de 2022.

- Primer puesto en sección "Salud", segundo puesto global del proyecto *Innova CONI-CET – AWS 2021* por un premio de ≈ 12000 USD por la presentación del proyecto "Multiomix-AWS una plataforma de oncogenómica en la nube: implementación de Machine Learning para la identificación de biomarcadores oncológicos".

1.4.2. Publicaciones en revistas internacionales

- Camele, G., Menazzi, S., Chanfreau, H., Marraco, A., Hasperué, W., Butti, M. D., & Abba, M. C. (2022). Multiomix: a cloud-based platform to infer cancer genomic and epigenomic events associated with gene expression modulation. *Bioinformatics*, 38(3), 866-868.
- Camele, G., Hasperué, W., Ronchetti, F., & Quiroga, F. M. (2022). Statistical analysis of the performance of four Apache Spark ML algorithms. *Journal of Computer Science & Technology*, 22.

1.4.3. Publicaciones en revistas nacionales

- Camele, G., & Hasperué, W. (2023). Selección de características en entornos Big data. Aplicación en gene signatures. *Investigación Joven*, 10(3), 454-455.
- Marraco, A. D., Camele, G., Hasperué, W., Menazzi, S., Abba, M. C., & Butti, M. A. (2021). Modulector: una plataforma como servicio para el acceso a bases de datos de micro ARNs. *Innovación y Desarrollo Tecnológico y Social*, 3.
- Menazzi, S., Chanfreau, H., Nastasi, D., Martinez, D., Camele, G., & Butti, M. (2019). Identificación de biomarcadores con poder pronóstico en cáncer: una perspectiva desde la ciencia de datos biomédicos y la bioinformática. *Revista Abierta de Informática Aplicada*, 3(2), 5-14.

1.4.4. Publicaciones en congresos y workshops

- Hasperué, W., Estrebou, C. A., Camele, G., Rucci, E., Ronchetti, F., Castillo, D., ... & Fernández Bariviera, A. (2023). Sistemas inteligentes en el uso de aplicaciones de bioinformática y sistemas embebidos. In XXV Workshop de Investigadores en Ciencias de la Computación (Junín, 13 y 14 de abril de 2023).
- Hasperué, W., Estrebou, C. A., Camele, G., López, P., Peña, M., Zambrano, R., ... & Cerrada, M. (2022). Procesamiento inteligente de la información: aplicaciones en

bioinformática, trayectorias vehiculares, mantenimiento preventivo industrial y sistemas embebidos. In XXIV Workshop de Investigadores en Ciencias de la Computación (WICC 2022, Mendoza).

- Camele, G. (2022). Selección de características en entornos Big Data. In Encuentro de Becarios de Posgrado de la UNLP (EBEC 2022)(La Plata, 23 de noviembre de 2022).
- Camele, G., Hasperué, W., Ronchetti, F., & Quiroga, F. M. (2021). Comparative Study of the Performance of the Classification Algorithms of the Apache Spark ML Library. In XXVII Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 4 al 8 de octubre de 2021).
- Hasperué, W., Estrebou, C. A., Camele, G., López, P., Jimbo Santana, P. R., Reyes Zambrano, G., ... & Fernández Bariviera, A. (2021). Procesamiento inteligente de grandes volúmenes de información y de flujos de datos. In XXIII Workshop de Investigadores En Ciencias de La Computación (WICC 2021, Chilecito, La Rioja).
- Lanzarini, L. C., Hasperué, W., Estrebou, C. A., Villa Monte, A., Jimbo Santana, P. R., Reyes Zambrano, G., ... & Olivas Varela, J. Á. (2020). Sistemas inteligentes: aplicaciones en minería de datos y big data. In XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz).

1.4.5. Formación de recursos humanos

Durante el desarrollo de esta tesis y sus plataformas adyacentes, se reclutaron varios alumnos pertenecientes a la Universidad Nacional de La Plata y la Universidad Abierta Interamericana, que se vieron interesados en colaborar con dichas plataformas.

El grupo de investigación conformado por dichas personas recibió el nombre de *Omic-Data-science*¹. El autor de esta tesis fue el encargado de dirigir dicho grupo de investigación durante el periodo de trabajo particular que haya elegido cada alumno involucrado.

Además de aprender diferentes tecnologías de interés, bioinformática y biología básica, algunos alumnos han tenido la posibilidad de utilizar el trabajo realizado para finalizar las carreras en curso. Los títulos que han surgido de este grupo de trabajo fueron los siguientes:

- Franco Bebczuk: Licenciatura en Informática, Universidad Nacional de La Plata, diciembre 2023.
- Agustín Daniel Marraco: Ingeniería en Sistemas Informáticos, Universidad Abierta Interamericana, agosto 2021.

¹<https://omicsdatascience.org/>

1.4.6. Desarrollo de herramientas

Las siguientes herramientas fueron desarrolladas dentro del marco de esta tesis y se encuentran publicadas con licencias de código abierto a disposición de la comunidad:

- Multiomix: plataforma para la aceleración de identificación de biomarcadores. <https://github.com/omics-datascience/multiomix>.
- Modulector: plataforma para la obtención de información de miRNA y sitios de metilación. <https://github.com/omics-datascience/modulector>.
- BioAPI: plataforma para la obtención de información de genes y pathways. <https://github.com/omics-datascience/BioAPI>.
- GGCA: librería de alta performance y bajo consumo de memoria para el análisis de correlación entre dos conjuntos de datos de expresión. <https://github.com/jware-solutions/ggca>.
- Multiomix AWS-EMR: plataforma intermediaria entre Multiomix y el servicio EMR de Amazon Web Service para la ejecución distribuida de algunos procesos de Multiomix. <https://github.com/omics-datascience/multiomix-aws-emr>.
- Docker Big Data Cluster: configuración lista para el despliegue de un clúster de Apache Spark y Hadoop utilizando Docker Swarm. <https://github.com/jware-solutions/docker-big-data-cluster>.
- Neo-react-semantic-ui-range: implementación de un input de tipo *range* para Semantic-UI, el framework JS/CSS utilizado en Multiomix. <https://github.com/jware-solutions/neo-react-semantic-ui-range>.
- Repositorio con la experimentación de las estrategias de balance de carga implementadas: contiene los datos, algoritmos y experimentos reportados en [23]. <https://github.com/midusi/load-balancer-metaheuristics>.
- Repositorio con la experimentación de las mediciones de Survival SVM: contiene los datos, algoritmos y experimentos reportados en [19]. <https://github.com/midusi/paper-CACIC-2023>.
- Repositorio con la experimentación de las mediciones de 4 algoritmos de Apache Spark ML: contiene los datos, algoritmos y experimentos reportados en [20]. <https://github.com/midusi/classification-models-spark>.

Durante la tesis se han realizado aportes tanto de código como reportes de errores a los siguientes proyectos de código abierto:

- Python: <https://github.com/python/cpython/issues/88110>
- Pandas: <https://github.com/pandas-dev/pandas/issues/36116> y <https://github.com/pandas-dev/pandas/issues/35124>
- Scikit-survival: <https://github.com/sebp/scikit-survival/issues/277> y <https://github.com/sebp/scikit-survival/issues/454>
- Lifelines: <https://github.com/CamDavidsonPilon/lifelines/issues/1545> y <https://github.com/CamDavidsonPilon/lifelines/pull/1547>
- cBioPortal: <https://github.com/cBioPortal/cbioportal/issues/7477>
- Django REST Framework: <https://github.com/encode/django-rest-framework/issues/7532>
- Django/Daphne: <https://github.com/django/daphne/issues/338>
- Zolkko/Kendalls: <https://github.com/zolkko/kendalls/issues/2>, <https://github.com/zolkko/kendalls/pull/3>, <https://github.com/zolkko/kendalls/pull/4> y <https://github.com/zolkko/kendalls/pull/5>
- Rust-GSL: <https://github.com/GuillaumeGomez/rust-GSL/issues/94>
- PyO3/Maturin: <https://github.com/PyO3/maturin/issues/411>, <https://github.com/PyO3/maturin/issues/387> y <https://github.com/PyO3/maturin/issues/385>
- PyO3/PyO3: <https://github.com/PyO3/pyo3/issues/2063>
- Facebook Docusaurus: <https://github.com/facebook/docusaurus/issues/4932>
- Apexcharts.js: <https://github.com/apexcharts/apexcharts.js/issues/2178>
- Float-pretty-print: <https://github.com/vi/float-pretty-print/issues/1>

1.5. Organización del documento

Capítulo 2: Se explican los conceptos básicos de la genética y epigenética en el contexto de la medicina de precisión en cáncer. Luego se introduce el concepto de biomarcador, su utilidad en el pronóstico de la evolución de una enfermedad y por qué resulta de suma importancia hacer uso de algoritmos bioinformáticos para descubrir nuevos biomarcadores.

Capítulo 3: Se brinda una introducción al concepto de selección de características, sus aplicaciones y algunos algoritmos populares en el estado del arte. Entre los algoritmos definidos se hace foco en las metaheurísticas, y su uso para optimizar el rendimiento de este proceso. Se finaliza este capítulo introduciendo la programación distribuida como mecanismo para optimizar dichas metaheurísticas a través de la distribución de su cómputo.

Capítulo 4: Se presenta Multiomix, una plataforma de código abierto que pone a disposición de la comunidad científica numerosos algoritmos de bioinformática, entre ellos la posibilidad de ejecutar metaheurísticas y otros algoritmos de selección de características para la aceleración del descubrimiento de biomarcadores.

Capítulo 5: Se presenta un framework que utiliza tres estrategias inteligentes de balance de carga para acelerar los algoritmos de selección de características en un entorno Spark. Dos de ellas presentan enfoques innovadores que hacen uso de la predicción del tiempo de las tareas a ejecutar para realizar una distribución óptima entre los workers de un clúster de Spark.

Capítulo 6: En este capítulo se detallan los experimentos llevados a cabo. Desde la medición de tiempos para generar datos para el entrenamiento de los modelos predictores de tiempos de ejecución, hasta la evaluación y comparación de las tres estrategias de distribución propuestas frente a diferentes escenarios.

Capítulo 7: Se presentan las conclusiones de esta tesis. Además, se proponen posibles trabajos futuros y áreas de investigación que podrían derivarse de los descubrimientos y aportes realizados.

Capítulo 2

Medicina de precisión

La terapia de precisión es un enfoque innovador en el ámbito médico que se basa en la personalización de tratamientos para adaptarse a las características genéticas únicas de cada paciente, en lugar de aplicar enfoques generalizados [9] [146].

En el contexto de esta tesis, la terapia de precisión aprovecha la información contenida en el ADN de un paciente. Esto implica identificar mutaciones genéticas específicas, variaciones en el número de copias de genes y otros factores genéticos que puedan influir en el desarrollo de enfermedades. Con esta información, los médicos pueden seleccionar tratamientos que aborden directamente las causas subyacentes de la enfermedad a nivel molecular, o realizar un pronóstico del paciente al predecir la evolución de la patología.

La terapia de precisión también se basa en datos epigenéticos, como los miRNA y sitios de metilación, cuyo efecto en la expresión génica, permiten personalizar los tratamientos para revertir o modular estas alteraciones epigenéticas específicas causantes de la patología.

El éxito de la medicina de precisión requiere la identificación de objetivos terapéuticos validados y susceptibles de recibir fármacos, junto con biomarcadores funcionales precisos de respuesta a los medicamentos. Esta estrategia se ha visto impulsada principalmente por el desarrollo de las tecnologías de secuenciación de próxima generación (Next Generation Sequencing: NGS), en caracterización completa del genoma humano y la aparición de ensayos genómicos costo-eficientes [124].

2.1. Biología del cáncer

El cáncer es una patología compleja que abarca diferentes aspectos biológicos fundamentales. Desde la secuencia de nucleótidos que conforma la información esencial de la célula (ADN), hasta los mecanismos que regulan sus funciones (los llamados *Reguladores*

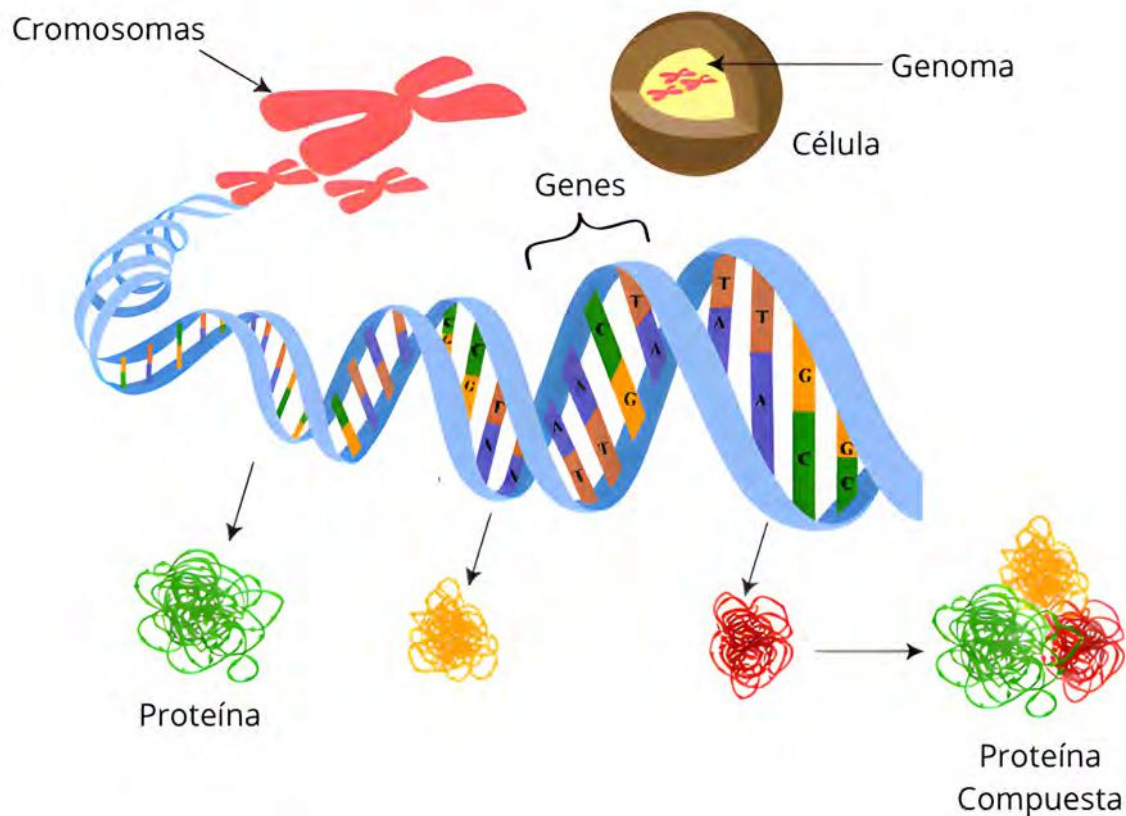


Figura 2.1 Figura con el esquema completo simplificado: dentro de la célula se encuentra el genoma que contiene los cromosomas. Cada cromosoma contiene en su interior ADN empaquetado, este consiste en una cadena de nucleótidos en la cual se encuentran regiones denominadas genes que contienen las instrucciones para sintetizar proteínas. Estas proteínas actúan solas o en complejos para realizar funciones biológicas específicas.

de Expresión o GEMs por sus siglas en inglés). Todos son engranajes cuya alteración puede tener una gran inferencia tanto en la ocurrencia de la enfermedad, como en su tratamiento.

En primer lugar, el ADN es la molécula que contiene la información genética necesaria para el desarrollo y funcionamiento de los organismos vivos. Esta información está organizada en segmentos específicos llamados genes.

Los genes son fragmentos de ADN que contienen instrucciones para la producción de proteínas, las cuales son fundamentales para el funcionamiento y desarrollo de los organismos (Figura 2.1). Todos los genes en su conjunto forman el genoma, es decir, la información genética de la especie.

La *expresión génica* es el proceso mediante el cual la información contenida en los genes se utiliza para sintetizar proteínas. Este proceso tiene lugar en dos etapas principales:

La *Transcripción*, proceso durante el cual se lee la información del ADN y se copia en ARN mensajero (mRNA en inglés). El mRNA lleva la información desde el núcleo de la célula al citoplasma, donde se llevará a cabo la siguiente etapa denominada *Traducción*.

Durante la Traducción, el mRNA se utiliza como plantilla para ensamblar una cadena de aminoácidos en una secuencia específica. Esta cadena de aminoácidos se pliega en una proteína funcional, cuya estructura tridimensional está estrechamente relacionada con la función de la proteína. Cuando esta se pliega correctamente, puede llevar a cabo su función específica, como catalizar reacciones químicas, transportar moléculas o actuar como componentes estructurales en las células. Sin embargo, si la proteína se pliega de manera incorrecta (por ejemplo, debido a mutaciones genéticas o condiciones ambientales), su función puede verse afectada, lo que puede dar lugar a enfermedades como el cáncer o disfunciones celulares.

Cuando un gen está produciendo más cantidad de mRNA (y en consecuencia, más cantidad de la proteína que codifica) de lo que sería típico para una célula o tejido en particular, se dice que el gen está *sobre-expresado*.

Por otro lado, cuando un gen está *sub-expresado*, significa que está produciendo menos mRNA y, por ende, menos cantidad de la proteína que codifica en comparación con las condiciones típicas. Tanto la sobre como la sub expresión de un gen puede deberse a diversos factores, como cambios en la regulación génica, mutaciones genéticas, o condiciones ambientales.

La sobre-expresión o sub-expresión de genes puede tener importantes implicaciones en la salud y el funcionamiento de un organismo, ya que puede afectar procesos biológicos clave y contribuir al desarrollo de enfermedades.

Los GEM son elementos que controlan cuándo y cómo se expresan los genes. Estos pueden ser proteínas que se unen a secuencias específicas del ADN para activar o reprimir la transcripción de un gen, o pueden ser moléculas de RNA que interfieren con la traducción de un gen a una proteína. En esta tesis solo se analizarán tres reguladores: microRNA (miRNA en inglés), Copy Number Alteration (CNA) y sitios de metilación del ADN.

La regulación de la expresión génica es esencial para el correcto funcionamiento celular, ya que permite a las células responder a cambios en su entorno y realizar funciones específicas en el momento adecuado.

Cuando ocurre una mutación en un gen, puede alterar la secuencia de la proteína que codifica, lo que puede resultar en una proteína que no funciona correctamente o que no se produce en absoluto. Asimismo, si los GEMs no funcionan correctamente, pueden causar que un gen se exprese en el momento equivocado o en la cantidad equivocada. Ambos escenarios pueden tener consecuencias perjudiciales para la célula y el organismo.

El cáncer es una enfermedad que se produce cuando las células adquieren la capacidad de dividirse y crecer de manera descontrolada. Esto puede ser el resultado de mutaciones en genes que controlan el crecimiento, la división o la muerte celular; o de alteraciones en los GEMs que controlan estos genes. Por ejemplo, si un gen que normalmente detiene la división celular se muta y deja de funcionar, la célula puede comenzar a dividirse sin control. Del mismo modo, si un regulador de expresión que normalmente reprime un gen que promueve la división celular deja de funcionar, el gen puede estar activo todo el tiempo, lo que también puede llevar a una división celular incontrolada.

Para más información detallada sobre los componentes mencionados en esta sección, se puede consultar la Sección A.

2.2. Blancos terapéuticos

En el ámbito médico, un blanco terapéutico se refiere a una o más moléculas específicas en el cuerpo humano que se elige como objetivo para un tratamiento particular. Este objetivo es crucial porque está directamente relacionado con la enfermedad o condición que se está tratando.

La elección cuidadosa del blanco terapéutico es fundamental para el desarrollo de tratamientos efectivos y específicos. Puede ser una proteína, un receptor celular, una enzima u otra molécula biológica que esté involucrada en la enfermedad. Por ejemplo, en el caso del cáncer, un blanco terapéutico común es una proteína específica que es generada por un gen que encuentra sobre expresado o mutado en las células cancerosas.

Los GEMs juegan un papel crucial en la medicina de precisión porque permiten un control más detallado y específico de la actividad génica. Estos pueden ser utilizados para ajustar la actividad de genes específicos que están implicados en la enfermedad. Por ejemplo, si un gen está sobre-expresado en una enfermedad particular, un regulador de expresión podría ser diseñado para disminuir la actividad de ese gen. Del mismo modo, si un gen está sub-expresado, un regulador de expresión podría aumentar su actividad.

Además, los GEMs pueden ser utilizados para dirigir terapias a células específicas. Por ejemplo, en el caso del cáncer, los GEMs podrían ser diseñados para activar genes que inducen la muerte celular sólo en las células cancerosas, dejando las células sanas intactas.

Si se desea obtener información más detallada acerca de diferentes enfoques para blancos terapéuticos y su importancia en el tratamiento y prevención de enfermedades, se puede consultar el Sección A.4.

2.3. Descubrimiento de reguladores de expresión

Uno de los problemas principales en el descubrimiento de los blancos terapéuticos radica en la complejidad asociada a la identificación de moléculas específicas que regulan genes particulares. A diferencia del genoma, que se refiere a la secuencia de ADN en los genes, la epigenómica se ocupa de las modificaciones químicas y estructurales que afectan la expresión génica sin cambiar la secuencia de ADN. Este nivel de complejidad hace que sea difícil determinar qué moléculas específicas están involucradas en la regulación de genes específicos.

Un blanco terapéutico puede ser cualquier mecanismo de modulación de la expresión génica. Este trabajo se enfoca únicamente en tres de ellos: las moléculas de miRNA, CNA, la metilación del ADN (todos explicados en detalle en el Sección A.2).

La complejidad radica en la interconexión y coordinación de estos mecanismos epigenéticos. La misma molécula de miRNA, por ejemplo, puede tener múltiples objetivos, y un gen específico puede ser regulado por diferentes mecanismos simultáneamente. Además, entre estos mecanismos se suceden interacciones altamente dinámicas y dependen de contextos específicos.

Una estrategia efectiva para identificarlos implica el análisis integrado de datos de expresión génica junto con los datos de los diferentes reguladores. El enfoque "todos vs todos" implica comparar todos los genes con todos los reguladores disponibles para determinar si hay correlaciones, especialmente correlaciones inversas, que puedan sugerir una relación regulatoria.

En este proceso, se recopilan conjuntos de datos que representan la expresión génica y la actividad de reguladores para un conjunto de pacientes. Estos datos pueden provenir de experimentos biológicos, estudios clínicos u otras fuentes. Luego, se utilizan algoritmos de correlación, como Pearson [104][125], Spearman [126] o Kendall [71], para calcular la relación estadística entre la expresión de genes y la actividad de los reguladores. Estos algoritmos también proveen un p-valor que sirve para saber si el resultado obtenido es significativo estadísticamente.

Los tres algoritmos de correlación utilizados para estudiar las relaciones gen-regulador y los métodos de ajuste de los p-valores en esta tesis se encuentran explicados en detalle en las secciones A.5 y A.6 respectivamente.

2.4. Biomarcadores

En el contexto que abarca esta tesis, un biomarcador consta de un conjunto de moléculas (mRNA, miRNA, CNA o sitios de metilación) con poder diagnóstico, pronóstico o predictivo para una enfermedad.

Un biomarcador está definido por datos genéticos, como cambios en el ADN, o alteraciones en la expresión génica; o epigenéticos, como información de miRNAs o metilación del ADN. Dichos datos, permiten indicar la presencia o el riesgo de una enfermedad específica, y/o proporcionar información sobre cómo podría progresar una enfermedad, cuando podría reaparecer el tumor, o cómo podría responder el paciente a un tratamiento particular.

Los biomarcadores genéticos pueden ayudar a identificar qué pacientes pueden beneficiarse de ciertos tipos de terapias dirigidas que atacan células cancerígenas que tienen ciertas mutaciones genéticas.

2.4.1. Aplicaciones en la bioinformática

El concepto de biomarcadores no es inherente a la informática. Se puede hacer uso de métodos manuales en laboratorio para evaluar la eficacia de un biomarcador para cierta patología. Por ejemplo, en [13] los autores evalúan el perfil genómico biomarcador diagnóstico en un tipo de metástasis; en [34] los autores descubrieron que la regulación de algunos genes se encuentra relacionada con la aparición de tumores en los ovarios.

Si bien estos métodos manuales son necesarios para la evaluación final de un biomarcador, resulta conveniente contar con herramientas que hagan uso de la informática para acelerar el tiempo total de dicho proceso. Realizar una evaluación informática (lo que se conoce como evaluación *in-silico*) no solo reduce los tiempos de pruebas, sino también abarata costos al disminuir la cantidad de posible biomarcadores a testear en un laboratorio. Por este motivo, los esfuerzos para poner la bioinformática a disposición de los investigadores toman más relevancia cada año.

En [75] los autores realizan un análisis de expresión diferencial en computadora para identificar los niveles normales de expresión frente a los que están correlacionados con el cáncer de ovarios. En [100] se evalúan 14 métodos de selección de características para quedarse con los atributos más relevantes provenientes de imágenes de tumores, con el fin de entrenar 12 modelos de aprendizaje supervisado para medir el poder pronóstico de estos atributos para el cáncer de cabeza y cuello. En [86] los autores entrenan un modelo Random Survival Forest para realizar regresiones sobre el tiempo de supervivencia y evaluar el poder pronóstico de diferentes genes para el cáncer de pulmón.

Los GEMs también pueden actuar como biomarcador pronóstico, por ejemplo en [47] se realiza un proceso de selección de características utilizando el algoritmo Boruta (que obtiene los atributos más significativos utilizando el modelo Random Forest para hacer clasificación). Este estudio llega a la conclusión que el miRNA hsa-miR-1343-3p podría servir para pronosticar la evolución de los pacientes que sufren cáncer gástrico. Otro ejemplo es el de [99], donde los autores identifican biomarcadores CNA que pueden contribuir a la aparición del cáncer de mama, para el reconocimiento propusieron y evaluaron un proceso de selección de características de Monte Carlo [35] sobre datos de las populares base de datos METABRIC y TCGA.

2.4.2. Análisis de supervivencia

Los biomarcadores no solo resultan de utilidad para el diagnóstico de enfermedades, sino también para predecir la evolución del paciente frente a la patología que sufre. Conocer la expresión tanto de los genes como de los reguladores de expresión permiten estimar la probabilidad de que un paciente sobreviva a una enfermedad durante un período de tiempo determinado. El análisis del tiempo de supervivencia de un paciente, o el riesgo de recurrencia de una enfermedad luego de curado (también llamado *recidiva*) se los conoce como *Análisis de supervivencia*.

El análisis de supervivencia es una herramienta crucial en la investigación médica y la práctica clínica. Se utiliza para estudiar el tiempo que transcurre desde el diagnóstico de cáncer hasta eventos críticos como la progresión de la enfermedad, la recurrencia del tumor o la muerte del paciente. Este enfoque es fundamental para comprender la eficacia de los tratamientos, identificar factores de riesgo y desarrollar estrategias de manejo de la enfermedad más efectivas.

En el análisis de supervivencia, se emplea la función de supervivencia para estimar la probabilidad de que un paciente sobreviva más allá de cierto periodo de tiempo después del diagnóstico. Además, se utilizan técnicas como la función de riesgo acumulado y los modelos de regresión, como el modelo de riesgos proporcionales de Cox [30], para explorar cómo diferentes factores, como la edad, el sexo, el tipo y estadio del cáncer, así como el tratamiento recibido, influyen en la supervivencia de los pacientes.

Este tipo de análisis aborda desafíos únicos, como el seguimiento a largo plazo de los pacientes y la presencia de datos censurados, donde la información sobre el tiempo de supervivencia puede no estar disponible para todos los sujetos del estudio.

Modelos de agrupamiento y curvas Kaplan-Meier

Las curvas de Kaplan-Meier [69] son gráficos que muestran la probabilidad de supervivencia de los pacientes con cáncer a lo largo del tiempo. Estas curvas se basan en datos reales observados y tienen en cuenta pacientes censurados (pacientes a los que ya no se les pudo hacer seguimiento, cualquiera sea el motivo, lo cual resulta en un problema a la hora de analizar los datos).

Estas curvas son útiles para comparar la supervivencia entre diferentes grupos de pacientes y para evaluar el impacto de variables como el tipo de cáncer o el tratamiento recibido.

Considerando un biomarcador del que se sospecha poder pronóstico, se podría realizar un agrupamiento por expresión génica de los pacientes de una base de datos. Luego de agrupar a los pacientes en dos o más grupos con expresiones similares, se puede realizar el procesamiento de las curvas Kaplan-Meier para observar la probabilidad de supervivencia de cada uno (Figura 2.2).

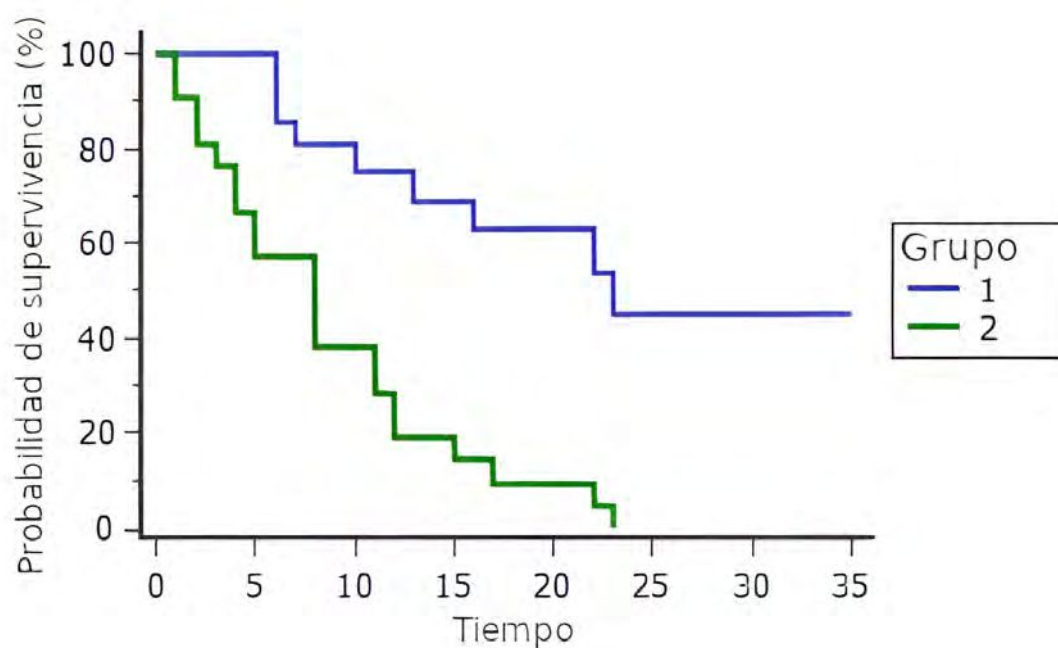


Figura 2.2 Curvas Kaplan-Meier donde se puede ver la probabilidad de supervivencia a lo largo del tiempo para dos grupos de pacientes.

Cuanto mayor sea la separación entre los grupos graficados, mejor poder pronóstico tendrá el biomarcador ya que logra hacer una distinción significativa en cuanto a la supervivencia de los diferentes grupos de pacientes. Sin embargo, la mera observación de las curvas no alcanza

para determinar que tan diferentes son, si no que se debe hacer uso de métricas objetivas que permitan medir la separación de los grupos, como LogRank Test o la regresión de Cox.

LogRank test El LogRank test [87] también conocido como prueba de rangos logarítmicos o prueba de Mantel-Cox, es un test estadístico no paramétrico que se utiliza para comparar las funciones de supervivencia de dos o más grupos.

El LogRank test compara las curvas de Kaplan-Meier de los diferentes grupos, evaluando si las diferencias observadas entre las curvas son estadísticamente significativas. La hipótesis nula es que los dos grupos tienen idéntica función de riesgo, por lo tanto, se busca obtener un p-valor significativo que nos permita descartar la hipótesis nula y, por lo tanto, determinar que se hallaron dos grupos significativamente diferentes en términos de probabilidad de ocurrencia del evento.

Regresión de Cox La regresión de Cox [54][55] es un método estadístico utilizado en el análisis de supervivencia para examinar cómo diversas variables afectan el riesgo de un evento, como la muerte o la recurrencia del cáncer. Esta técnica no requiere suposiciones específicas sobre la forma funcional de la relación entre las variables y el riesgo. En cambio, utiliza el concepto de hazard ratio (cociente de riesgos) para estimar cómo el riesgo relativo de un evento cambia en relación con los niveles de las variables predictoras. Los modelos de regresión de Cox ofrece dos funcionalidades.

En primer lugar está el Índice de concordancia (C-Index), que es una generalización del Área Bajo la Curva (AUC) para datos de supervivencia, incluyendo la censura. Este índice evalúa qué tan bien el modelo discrimina entre individuos que experimentan el evento de interés y aquellos que están censurados en un tiempo determinado. Un índice c de 0.5 indica que el modelo no tiene capacidad discriminatoria, mientras que un valor de 1 indica discriminación perfecta.

Por otro lado están los coeficientes individuales por cada variable o atributo, los cuales proporcionan estimaciones del impacto relativo de cada variable predictora sobre el riesgo de experimentar el evento de interés. Un coeficiente positivo indica que un aumento en el valor de la variable predictora está asociado con un aumento en el riesgo de experimentar el evento, mientras que un coeficiente negativo indica una asociación inversa. El valor absoluto del coeficiente refleja la magnitud del efecto de la variable en el riesgo, mientras que el signo indica la dirección de la asociación. Los intervalos de confianza asociados con los coeficientes proporcionan información sobre la precisión de estas estimaciones. Así, la regresión de Cox permite identificar qué variables están relacionadas con cambios en el riesgo de eventos de interés en el análisis de supervivencia.

Modelos penalizados de Cox

El modelo de regresión de Cox suele ser un modelo atractivo, porque sus coeficientes pueden interpretarse en términos de cociente de riesgos, lo que a menudo proporciona información valiosa y fácil de entender. Sin embargo, si queremos estimar los coeficientes de muchos grupos de pacientes, el modelo estándar de Cox no es aplicable, porque internamente intenta invertir una matriz que se vuelve no singular debido a las correlaciones entre las características. Para solventar el problema se puede hacer uso de otro tipo de modelos más complejos como *Ridge*, *LASSO* y *Elastic Net*.

Ridge

Una de las soluciones al problema mencionado anteriormente consiste en añadir un término de penalización ℓ_2 en los coeficientes que los reduzca a cero. El objetivo de Cox modificado tiene la forma:

$$\operatorname{argmax}_{\beta} = \log PL(\beta) - \frac{\alpha}{2} \sum_{j=1}^p \beta_j^2$$

Donde $PL(\beta)$ es la función de verosimilitud parcial del modelo de Cox, β_1, \dots, β_p son los coeficientes para p características, y $\alpha \geq 0$ es un hiperparámetro que controla la cantidad de contracción. El objetivo resultante suele denominarse *regresión de puente* [58] (por eso el término en inglés *Ridge*, que nace de "Regression" y "Bridge") y . Si α se fija en cero, obtenemos el modelo de Cox estándar, no penalizado.

LASSO

Aunque la penalización ℓ_2 (Ridge) resuelve el problema matemático de ajustar un modelo de Cox a múltiples grupos, seguiría la necesidad de medir todas las variables de un conjunto de datos para obtener la métrica C-Index. Idealmente, sería conveniente seleccionar un pequeño subconjunto de características que sean las más predictivas e ignorar el resto. Este es precisamente el enfoque adoptado por la penalización Least Absolute Shrinkage and Selection Operator (LASSO) [15][132]. En lugar de reducir los coeficientes a cero, realiza un tipo de selección continua de subconjuntos, en la que un subconjunto de coeficientes se establece en cero y se excluye de forma efectiva. Esto reduce el número de características necesarias para la predicción. En términos matemáticos, la penalización ℓ_2 se sustituye por una penalización ℓ_1 , lo que conduce al siguiente problema de optimización:

$$\operatorname{arg máx}_{\beta} \quad \log PL(\beta) - \alpha \sum_{j=1}^p |\beta_j|$$

El principal desafío es que no es posible controlar directamente el número de características que se seleccionan, sino que el valor de α determina implícitamente dicho número. Por lo tanto, se necesita un método basado en los datos para seleccionar un α adecuado y obtener un modelo parsimonioso. Esto se puede solventar calculando primero el α que ignoraría todas las características (todos los coeficientes son cero) y luego disminuyendo su valor de forma incremental, hasta alcanzar algún criterio de parada como el 1 % del valor original.

Elastic Net

El LASSO es una gran herramienta para seleccionar un subconjunto de características discriminativas, pero tiene dos inconvenientes principales. En primer lugar, no puede seleccionar más características que el número de muestras en los datos de entrenamiento, lo que es problemático cuando se trata de datos de muy alta dimensión. En segundo lugar, si los datos contienen un grupo de características muy correlacionadas, la penalización LASSO elegirá aleatoriamente una característica de este grupo. La penalización de Elastic Net supera estos problemas utilizando una combinación ponderada de la penalización ℓ_1 y ℓ_2 resolviendo:

$$\arg \max_{\beta} \log \text{PL}(\beta) - \alpha \left(r \sum_{j=1}^p |\beta_j| + \frac{1-r}{2} \sum_{j=1}^p \beta_j^2 \right)$$

donde $r \in [0; 1[$ es el peso relativo de la ℓ_1 y ℓ_2 penalización. La penalización Elastic Net combina la propiedad de selección de subconjuntos de LASSO con la fuerza de regularización de la penalización Ridge. Esto conduce a una mejor estabilidad en comparación con el modelo penalizado LASSO. Para un grupo de características altamente correlacionadas, este último elegiría una característica al azar, mientras que el modelo penalizado con Elastic Net tendería a seleccionarlas todas. Normalmente, basta con dar a la penalización ℓ_2 sólo un pequeño peso para mejorar la estabilidad del LASSO, por ejemplo, estableciendo $r = 0,9$.

Modelos de regresión

Tanto LogRank Test como la regresión de Cox permiten la evaluación de los diferentes grupos de riesgo. Este último además permite conocer los coeficientes de las característica del conjunto de datos para la inferencia de tiempo de supervivencia o función de riesgo.

Sin embargo, si se busca predecir el tiempo de ocurrencia de un evento, ninguno de los métodos anteriores sería de utilidad. Para realizar estas tareas de regresión de tiempo de supervivencia/recidiva se puede hacer uso de otros dos modelos con la capacidad de considerar patrones más complejos en la información, como el *Random Survival Forest* y *Survival Support Vector Machines*.

Random Survival Forest

Al igual que sus homólogos populares para la clasificación y la regresión, un Random Survival Forest (RSF) [62] es un conjunto de árboles de decisión que están modificados para considerar los datos de tiempo y evento (indispensables para el análisis de supervivencia).

Este modelo garantiza que los árboles individuales estén descorrelacionados mediante 1) la construcción de cada árbol en una muestra bootstrap diferente de los datos de entrenamiento originales, y 2) en cada nodo sólo evalúa el criterio de división para un subconjunto de características y umbrales seleccionados aleatoriamente. Las predicciones se forman agregando las predicciones de los árboles individuales del conjunto.

Survival Support Vector Machines

Las máquinas de vectores soporte de supervivencia (SSVM por sus siglas en inglés) [106] [107], son una extensión de la máquina de vectores soporte estándar para datos de tiempo y evento que también tienen en consideración la censura de los datos. Su principal ventaja es que puede tener en cuenta relaciones complejas y no lineales entre las características y la supervivencia mediante las denominadas funciones de kernel. Una función de kernel asigna implícitamente las características de entrada a espacios de características de alta dimensión en los que la supervivencia puede describirse mediante un hiperplano. Esto hace que SSVM sea extremadamente versátil y aplicable a una amplia gama de datos.

El análisis de supervivencia con SSVM puede describirse de dos formas diferentes:

- Como un problema de **clasificación**: el modelo aprende a asignar un rango inferior a las muestras con tiempos de supervivencia más cortos, y viceversa.
- Como un problema de **regresión**: el modelo aprende a predecir directamente el tiempo (logarítmico) de supervivencia.

En ambos casos, la desventaja es que las predicciones no pueden relacionarse fácilmente con cantidades estándar en el análisis de supervivencia, a saber, la función de supervivencia y la función de riesgo acumulativo.

En un SSVM con kernel lineal, los datos de entrenamiento consisten en n tripletes $(\mathbf{x}_i, y_i, \delta_i)$, donde, \mathbf{x}_i es un vector de características d -dimensional, $y_i > 0$ el tiempo de supervivencia o tiempo de censura, y $\delta_i \in \{0, 1\}$ el indicador de suceso binario. Utilizando los datos de entrenamiento, el objetivo es minimizar la siguiente función:

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\alpha}{2} \left[r \sum_{i, j \in \mathcal{D}} \max(0, 1 - (\mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \mathbf{x}_j))_+^2 + (1 - r) \sum_{i=0}^n (\zeta_{\mathbf{w}, b}(y_i, x_i, \delta_i))^2 \right]$$

$$\zeta_{\mathbf{w},b}(y_i, \mathbf{x}_i, \delta_i) = \begin{cases} \max(0, y_i - \mathbf{w}^T \mathbf{x}_i - b) & \text{if } \delta_i = 0, \\ y_i - \mathbf{w}^T \mathbf{x}_i - b & \text{if } \delta_i = 1, \end{cases}$$

$$\text{mathcal{P}} = \{(i, j) \mid y_i > y_j \wedge \delta_j = 1\}_{i,j=1,\dots,n}$$

El hiperparámetro $\alpha > 0$ determina la cantidad de regularización a aplicar: un valor menor aumenta la cantidad de regularización y un valor mayor la reduce. El hiperparámetro $r \in [0; 1]$ determina el equilibrio entre el objetivo de clasificación y el objetivo de regresión. Si $r = 1$ se reduce al objetivo de clasificación, y si $r = 0$ al objetivo de regresión.

Alcanza utilizar un $r < 1$ para llevar a cabo una función de regresión, donde un valor menor indica un tiempo menor de supervivencia y viceversa.

2.5. Evaluación de biomarcadores

Debido a la importancia de los biomarcadores para el diagnóstico y pronóstico de los pacientes frente a una patología, es crucial contar con métodos objetivos para comparar la utilidad de los mismos.

Frente a diferentes biomarcadores en estudio, los investigadores pueden hacer uso de los métodos de análisis de supervivencia (explicados en detalle en la Sección 2.4.2) para evaluar cuáles predicen mejor la evolución de un conjunto de pacientes a lo largo del tiempo. Ya sea evaluando la distancia entre las curvas Kaplan-Meier con LogRank o la Regresión de Cox, a partir de los coeficientes de los modelos penalizados de Cox, o a partir de la métrica C-Index realizando predicciones de tiempos con SSVM o RSF; los investigadores pueden analizar con diferentes métodos la utilidad práctica de los biomarcadores de interés.

Dichos biomarcadores pueden ser optimizados a través de algoritmos de FS (introducidos en el Capítulo 3), reduciendo la cantidad de moléculas que los conforman y mejorando las métricas obtenidas. Dichos algoritmos evaluarán algunos de los métodos de análisis de supervivencia para decidir qué subconjuntos de moléculas son los más prometedores.

A su vez, al contar con biomarcadores que permiten discriminar significativamente la evolución de diferentes grupos de pacientes para una patología, resulta interesante descubrir reguladores de expresión (como se detalla en la Sección 2.3) que afecten a las moléculas de dichos biomarcadores (lo que se define como Blanco Terapéutico en la Sección 2.2) para poder controlar la actividad de las mismas y mejorar así el pronóstico de los pacientes.

Capítulo 3

Selección de características

Este capítulo ofrece una introducción a la selección de características (FS por "Feature Selection") y su importancia en el análisis de datos. Se incluye una breve descripción de los principales métodos de FS, y se explica la motivación detrás de la FS en el contexto de la identificación de biomarcadores oncológicos, destacando sus beneficios en la reducción de costos, tiempo y mejora del rendimiento de los modelos. El capítulo profundiza en varias metaheurísticas utilizadas para FS, como Binary Black Hole, Algoritmos Genéticos y Binary Particle Swarm Optimization, explicando su funcionamiento y aplicaciones. Además, se presenta una revisión del trabajo previo en el campo, discutiendo diversas herramientas y enfoques para la FS en bioinformática. Finalmente, se describe el proceso de ejecución distribuida de metaheurísticas, que es el enfoque principal de la tesis para optimizar la identificación de biomarcadores utilizando técnicas de balance de carga en entornos de computación distribuida como Apache Spark [145].

La FS es un proceso clave en el análisis y modelado de datos que busca identificar las variables relevantes o más significativas para resolver una determinada tarea o predecir un resultado específico. Este procedimiento ayuda a:

- Reducir la dimensionalidad y por ende, la complejidad del problema.
- Mejorar la interpretabilidad del modelo y los resultados.
- Reducir los costes de obtención y almacenamiento de la información.
- Mejorar la eficiencia computacional de los algoritmos que hacen uso de la información.
- Evitar el sobreajuste en los algoritmos de machine learning.

El proceso de FS puede realizarse mediante diversos métodos que podrían ser clasificados en tres grandes grupos que se describen a continuación.

Métodos de filtrado Los métodos de filtrado (conocidos también como *Filter*) se centran en las propiedades intrínsecas de las características y miden su relevancia mediante estadísticas univariantes, como coeficientes de correlación o umbrales de varianza. Estos métodos son menos costosos desde el punto de vista computacional, pero pueden pasar por alto las relaciones entre las características. Un ejemplo es el método del umbral de varianza, que selecciona características cuyo nivel de varianza se encuentra por debajo de un umbral específico.

Métodos Empaquetados Los métodos empaquetados (conocidos también como *Wrapper*) evalúan la utilidad de las características mediante la optimización del rendimiento de un algoritmo clasificador (como SSVM o RSF), lo cual hace que sean más costosos debido a pasos de aprendizaje repetidos y validaciones cruzadas, pero es capaz de considerar patrones complejos y relaciones entre las características. Este es el enfoque principal utilizado en esta tesis.

Métodos híbridos Los métodos híbridos combinan las cualidades de los métodos de Filtro y Empaquetados. Se aplican mediante algoritmos que incorporan sus propios métodos de FS.

Algunos de los ejemplos más populares de estos métodos son la regresión LASSO y RIDGE (ambos métodos introducidos en la Sección 2.4.2), que incorporan funciones de penalización para reducir el sobreajuste.

3.1. Motivación

Los procesos de FS en el contexto de la identificación de biomarcadores son especialmente valiosos para reducir la dimensionalidad de los datos y mejorar así el rendimiento de los modelos de análisis de supervivencia. A continuación, se presentan dos aspectos principales de esta aplicación de FS:

- Reducción de costos y tiempo: la expresión genómica genera grandes volúmenes de datos, y el secuenciamiento de ADN y RNA es costoso. Realizar FS permite identificar los genes o moléculas más relevantes para predecir el resultado de interés, lo que permite trabajar con conjuntos de datos más pequeños y manejables. Esta reducción de la dimensionalidad no solo simplifica el trabajo sino que también reduce los costos asociados con el ensayo genético (cantidad de genes a evaluar experimentalmente en una muestra de un paciente) y los tiempos de ejecución de los algoritmos de análisis.

- Mejor rendimiento de los modelos: cuando se trabaja con grandes conjuntos de datos, hay un aumento en el ruido y la heterogeneidad, lo que puede afectar negativamente el rendimiento de los modelos de análisis de supervivencia. El proceso de FS permite eliminar las características irrelevantes o poco relevantes, lo que reduce el ruido y mejora la calidad de los datos disponibles para entrenar los modelos.

Esta tesis hace foco sobre los beneficios del segundo ítem: a partir de técnicas de FS se busca reducir la cantidad de características de un dataset de modo que al aplicar un algoritmo de análisis de supervivencia se obtenga un mejor resultado. Esta mejora se debe a la reducción de variables en el dataset que pueden no ser significativas para la predicción de la ocurrencia de un evento.

Durante todo este trabajo cuando se hable de características se hace referencia a genes o moléculas como los reguladores de expresión introducidos en el Capítulo 2. En la práctica, cada característica corresponde a una columna en un archivo con formato tabular.

El proceso que se describe en este capítulo se puede apreciar en la Figura 3.1, donde se aprecia la reducción de la cantidad de características, hecho que además permite mejorar la performance final del método que evalúa el poder pronóstico/predictivo del subconjunto de genes. Cabe mencionar que si bien en el ejemplo el conjunto de datos solo corresponde a expresión génica, las herramientas desarrolladas soportan múltiples ómicas (de ahí el nombre *Multiomix*), por ende el dataset podría contener datos de la expresión de miRNAs, cantidad de copias de un gen en particular (lo que se definió como CNA en esta tesis), la expresión de los sitios de metilación en el ADN o la combinación de varios de estos tipos de datos.

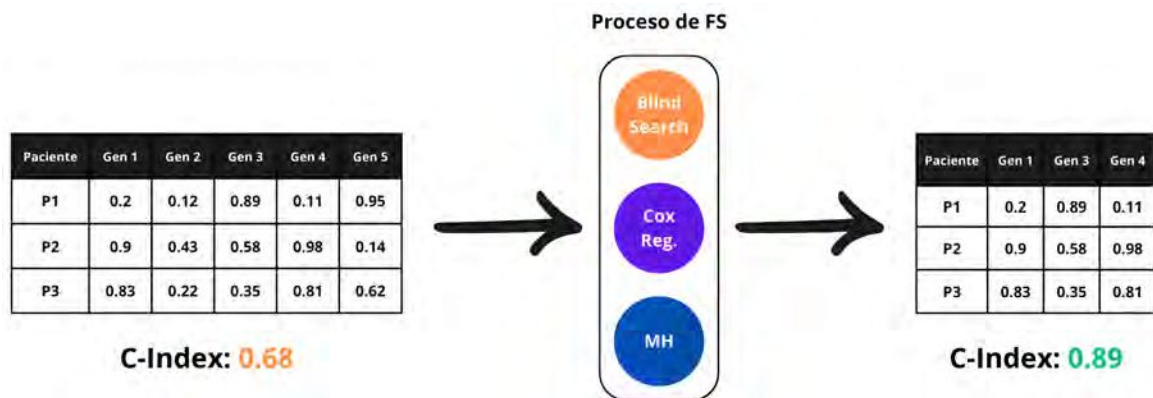


Figura 3.1 Esquema simple del proceso de FS. En este ejemplo, independientemente del algoritmo de FS utilizado, se llega a un conjunto de datos con menos genes que obtiene un valor de C-Index más alto en comparación con el conjunto de datos original.

3.2. Blind Search

La búsqueda ciega consiste en la evaluación del conjunto de soluciones *completo*. Es decir, dado un conjunto de N características se evalúan las $2^N - 1$ posibles combinaciones. Si el dataset a reducir contuviera los genes A , B y C , deberían evaluarse las $2^3 - 1 = 7$ combinaciones A , B , C , AB , AC , BC , ABC .

La ventaja de este método es que garantiza que se alcance el mejor poder predictivo global, obteniendo la respuesta óptima a cualquier problema. Sin embargo, cuando el número de características a evaluar es muy grande, el tiempo de ejecución requerido aumenta exponencialmente, impidiendo obtener una respuesta en un periodo de tiempo aceptable. Por ejemplo, en la actualidad se dispone de información de ≈ 25000 genes, lo que involucra la evaluación de $2^{25000} - 1$ combinaciones posibles. Este problema se acrecienta aún más cuando la evaluación de cada subconjunto en el espacio de soluciones requiere de la ejecución de un proceso costoso como el entrenamiento de un modelo de machine learning como es el caso de los métodos presentados en esta tesis.

3.3. Regresión de Cox penalizada

Como se describió en detalle en la Sección 2.4.2, algunos modelos de regresión de Cox (aunque no estén diseñado específicamente para FS) permiten clasificar y ordenar las variables según su efecto sobre la función de riesgo condicional cuando se aplica alguna penalización como LASSO o Elastic Net. Este proceso de clasificación y ordenamiento interno de las características puede utilizarse como una aproximación para realizar FS relevante con los datos de supervivencia [38].

En lugar de buscar directamente la mejor combinación de características, el uso del modelo de regresión de Cox para FS involucra la siguiente estrategia:

1. Calcular el coeficiente de regresión para cada característica independiente usando un modelo de regresión de Cox. Dicho coeficiente se puede interpretar como la *importancia* que tiene cada característica en la ocurrencia de un evento de interés (en este caso, de la muerte del paciente o la recidiva de un tumor).
2. Ordenar las características por sus valores absolutos de sus coeficientes en sentido descendente. Dejando así primero a las características con mayor influencia en la función de riesgo condicional.
3. Seleccionar aquellas características cuyos coeficientes de regresión son estadísticamente significativos y mantener solo esas características en el modelo final.

En la Figura 3.2 se puede apreciar gráficamente el coeficiente obtenido por cada uno de los genes de un conjunto de datos de ejemplo a partir de la regresión de Cox. Siguiendo con ese ejemplo, si se quisiera reducir el número de características de seis genes (número total de genes en el conjunto) a solo tres alcanza con quedarse con los primeros tres genes ya que son los que poseen coeficiente absoluto más alto. La cantidad de genes óptima a conservar dependerá del poder pronóstico/predictivo que se obtenga con cada cantidad particular utilizando cualquier modelo introducido en la Sección 2.4.2.

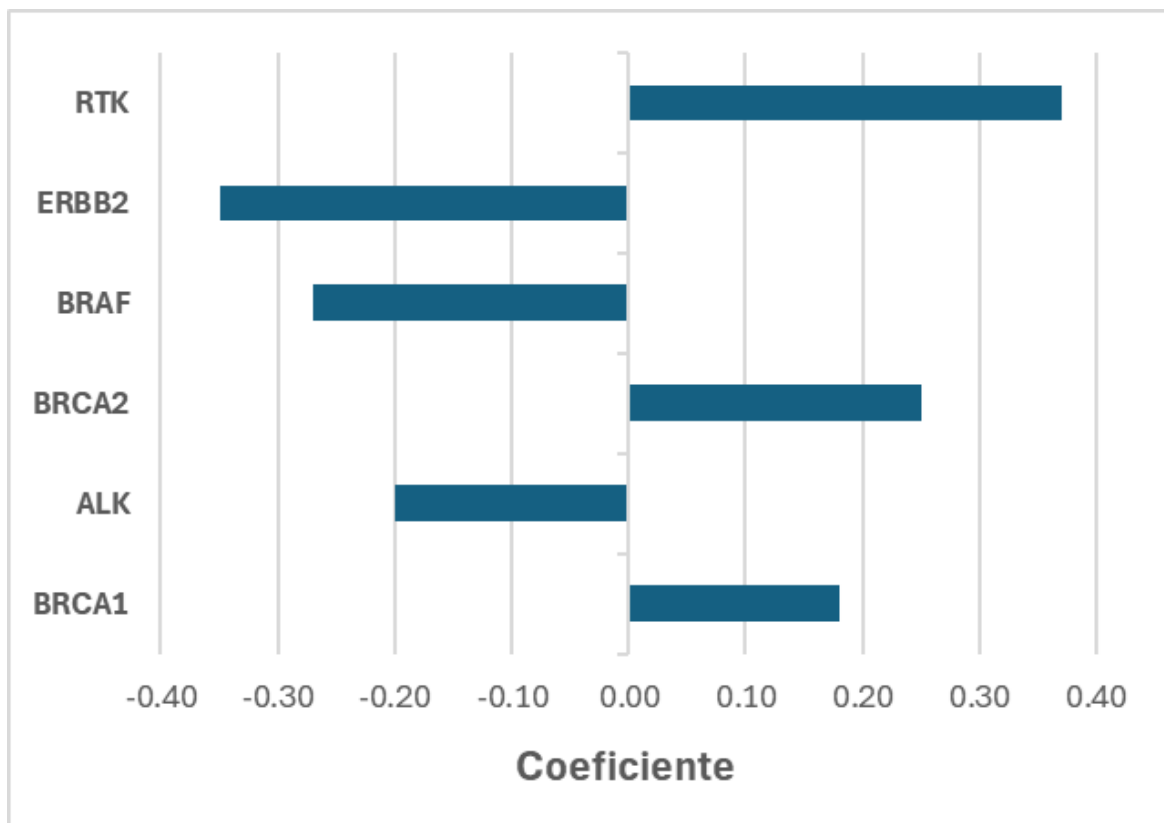


Figura 3.2 Visualización del coeficiente (eje X) obtenido por cada uno de los genes (eje Y), ordenados de manera decreciente por el valor absoluto de dichos coeficientes.

Es importante recordar que este método no siempre garantiza la elección óptima de características, pero puede ser útil cuando se trabaja con grandes cantidades de datos y se desea reducir la dimensión del espacio de características rápidamente.

3.4. Metaheurísticas

Las metaheurísticas son algoritmos de optimización que se utilizan para encontrar soluciones aproximadas de alta calidad a problemas complejos de optimización, donde una

búsqueda exhaustiva es computacionalmente inviable. En el contexto del descubrimiento de biomarcadores, utilizando datos genéticos, epigenéticos y/o de expresión génica, las metaheurísticas desempeñan un papel crucial debido a la naturaleza altamente compleja y de gran dimensionalidad de estos datos.

Este problema de optimización involucra miles o millones de combinaciones a evaluar. Realizar una búsqueda exhaustiva de todas ellas es computacionalmente intratable, incluso para conjuntos de datos moderadamente pequeños. El panorama empeora cuando la evaluación de cada una de estas combinaciones involucra el entrenamiento y evaluación de modelos de aprendizaje automático, los cuales podrían llegar a requerir un tiempo de ejecución mucho mayor al tolerable.

Las metaheurísticas ofrecen una alternativa eficiente para explorar este vasto espacio de búsqueda de manera inteligente y encontrar soluciones aproximadas de alta calidad en un tiempo razonable. Estas técnicas se basan en principios y estrategias de búsqueda inspiradas en fenómenos naturales, como la evolución biológica (algoritmos genéticos), el comportamiento de enjambres (optimización de colonia de partículas), entre otros. Estas técnicas explotan mecanismos de diversificación e intensificación para evitar quedar atrapadas en óptimos locales y, al mismo tiempo, aprovechar las mejores soluciones encontradas hasta el momento para guiar la búsqueda hacia regiones más prometedoras del espacio de soluciones.

Una de las principales ventajas de las metaheurísticas en este contexto es su capacidad para manejar problemas de optimización complejos, no lineales y con múltiples objetivos. Además, estas técnicas son flexibles y pueden adaptarse a diferentes tipos de problemas y restricciones, lo que las hace particularmente útiles para abordar la heterogeneidad y complejidad inherentes a los datos biológicos y clínicos relacionados con el cáncer.

Si bien existen numerosas metaheurísticas, esta tesis se enfocará en tres de ellas: Binary Black Hole, Algoritmos Genéticos y Binary Particle Swarm Optimization.

3.4.1. Binary Black Hole

La metaheurística Binary Black Hole (BBH) [101] y su versión mejorada [45] son metaheurísticas basadas en el concepto de los agujeros negros en el universo. Es una variante del algoritmo original Black Hole Algorithm [56], adaptada para resolver problemas de optimización con variables binarias.

El BBH simula el movimiento de un conjunto de estrellas (llamados *agentes* o soluciones candidatas) alrededor de un agujero negro (solución óptima) en un espacio de búsqueda binario. Cada estrella se representa como una cadena binaria de la misma longitud que el número de características disponibles, donde cada bit representa si una variable es considerada para evaluar o no en una estrella.

Si el conjunto de datos a evaluar consiste en [*Gen 1*, *Gen 2*, *Gen 3*] y una estrella posee la cadena 101 significa que solo considerará a las características *Gen 1* y *Gen 3*, ignorando por completo los datos de *Gen 2*.

El algoritmo comienza con una población inicial de estrellas generadas aleatoriamente. A continuación, se evalúa la función objetivo para cada estrella, y la que obtenga el mejor valor de fitness se selecciona como el agujero negro temporal.

En cada iteración, las estrellas se mueven hacia el agujero negro utilizando la siguiente fórmula:

$$X_i(t+1) = X_i(t) + \text{rand} * (X_{BH} - X_i(t)) \quad i = 1, 2, \dots, N, \quad (3.1)$$

Donde $X_i(t)$ y $X_i(t+1)$ son las ubicaciones de la estrella i^{th} en las iteraciones t y $t+1$, respectivamente. *rand* indica un número aleatorio dentro de una distribución con un rango de 0 a 1. N indica el número de estrellas. X_{BH} señala la "ubicación" del agujero negro en el espacio de soluciones. Esta ecuación determina la dirección y la magnitud del movimiento de cada estrella, lo que se traduce en cambios en los bits de su cadena binaria. La magnitud del movimiento se calcula en función de la distancia entre la estrella y el agujero negro, así como de un factor de aceleración que decrece gradualmente a medida que el algoritmo avanza.

Después de que todas las estrellas se han movido, se evalúa nuevamente la función objetivo y se actualiza el agujero negro temporal si se encuentra una mejor solución. Este proceso se repite iterativamente hasta que se alcance un criterio de parada predefinido, como un número máximo de iteraciones o una convergencia satisfactoria.

Una característica importante del BBH es el operador de cruce binario, que se aplica a las estrellas con cierta probabilidad. Este operador intercambia bits entre dos estrellas seleccionadas aleatoriamente, introduciendo diversidad en la población y ayudando a evitar la convergencia prematura en óptimos locales. Además, el BBH puede incluir mecanismos adicionales, como la mutación binaria, que invierte aleatoriamente algunos bits de las estrellas para promover aún más la diversidad y exploración del espacio de búsqueda.

El BBH ha demostrado ser eficaz en una variedad de problemas de optimización binaria: desde FS sobre datasets de genes para la clasificación de pacientes que recidivaron [103]; la identificación de patrones para la detección de robo de electricidad en Brasil [108]; hasta la mejora en el rendimiento de la extracción del subconjunto óptimo de características de alta dimensión en diferentes bases de datos de texto [143].

3.4.2. Algoritmos genéticos

Los algoritmos genéticos [93] (GA, por sus siglas en inglés) son metaheurísticas de optimización inspiradas en los principios de la evolución biológica y la genética. En estas técnicas, los agentes son estructuras de datos que codifican las variables de decisión del problema. Estas representaciones pueden ser binarias, enteras, reales o una combinación de ellas, dependiendo del problema. El algoritmo comienza con una población inicial de agentes, generalmente creados de manera aleatoria o utilizando alguna heurística específica del problema.

Cada individuo de la población se evalúa mediante una función de fitness que mide qué tan buena es esa solución para el problema. Esta función de fitness está directamente relacionada con la función objetivo del problema de optimización. A continuación, se seleccionan agentes de la población actual para ser "padres" en la siguiente generación. Esto se realiza mediante los denominados operadores de selección, como la selección por ruleta, selección por torneos o selección por ranking, que favorecen a los agentes con mayor aptitud.

Los agentes seleccionados se combinan mediante un operador de cruce para generar nuevos agentes "hijos". El cruce implica el intercambio de material genético entre dos padres, creando así nuevas soluciones potenciales. Para mantener la diversidad genética y explorar nuevas regiones del espacio de búsqueda, se aplica un operador de mutación a los agentes resultantes del cruce. La mutación modifica aleatoriamente algunos elementos de la representación genética de un individuo.

Una vez generados los nuevos agentes, se seleccionan los mejores agentes de la población actual y de la nueva generación para formar la población de la siguiente iteración, reemplazando a los agentes menos aptos. El proceso evolutivo se repite iterativamente hasta que se alcance un criterio de parada predefinido, como un número máximo de generaciones, un valor de fitness objetivo o una convergencia satisfactoria.

Los algoritmos genéticos son ampliamente utilizados en una variedad de problemas de optimización debido a su capacidad para explorar eficientemente espacios de búsqueda complejos y encontrar soluciones cercanas al óptimo global. Desde su uso en enrutamiento de redes encontrando la ruta más corta como se probó en [116], pasando por la mejora de la calidad visual de imágenes ajustando los niveles de brillo y oscuridad de manera más eficiente que la ecualización de histogramas [134]; hasta la identificación de sitios favorables de unión al agua en las superficies de las proteínas, un problema importante en bioquímica y diseño de fármacos [111].

3.4.3. Binary Particle Swarm Optimization

La metaheurística Binary Particle Swarm Optimization [73] (BPSO) es una variante del algoritmo de Particle Swarm Optimization [36] [72] [37], adaptada para resolver problemas de optimización con variables binarias.

En esta metaheurística, análogamente a las estrellas en BBH, sus agentes se llaman partícula y cada una representa una solución candidata codificada como una cadena binaria, donde cada bit corresponde a la característica a considerar. El BPSO comienza con una población inicial de partículas generadas aleatoriamente en el espacio de búsqueda binario. Cada partícula se mueve a través del espacio de soluciones siguiendo una trayectoria determinada por su propia experiencia y la experiencia conjunto de partículas general.

En cada iteración, cada partícula actualiza su posición (solución candidata) en función de su propia mejor posición encontrada hasta el momento (llamada *pbest*) y la mejor posición global encontrada por todo el enjambre (llamada *gbest*). Al igual que BBH y debido a que el espacio de búsqueda es binario, las partículas no pueden moverse directamente a una nueva posición. En su lugar, se calcula una probabilidad de cambio para cada bit de la cadena binaria.

Esta probabilidad de cambio se determina mediante una función de transferencia sigmoide que toma en cuenta la velocidad de la partícula y una constante de aceleración. Si la probabilidad de cambio para un bit es mayor que un umbral predeterminado, ese bit se invierte (de 0 a 1 o de 1 a 0). De esta manera, las partículas se "mueven" en el espacio de búsqueda binario al actualizar sus posiciones bit por bit.

Después de actualizar sus posiciones, las partículas evalúan la función objetivo y actualizan sus valores *pbest* y *gbest* si encuentran soluciones mejores. Este proceso se repite iterativamente hasta que se alcanza un criterio de parada, como un número máximo de iteraciones o una convergencia satisfactoria.

Además del movimiento básico de las partículas, el BPSO puede incluir operadores adicionales, como mutación binaria o reinicialización de partículas, para promover una mayor diversidad y exploración del espacio de búsqueda.

3.5. Trabajo previo

La bioinformática, a través de la integración de datos biológicos y técnicas computacionales, brinda una profunda comprensión de los complejos sistemas biológicos, que van desde la genómica hasta el campo del descubrimiento de fármacos. En consecuencia, la comunidad científica está desarrollando activamente herramientas y algoritmos para este

propósito, incluyendo estrategias de optimización, para mejorar la eficacia y eficiencia de los análisis bioinformáticos.

Las metaheurísticas juegan un papel fundamental dentro de este contexto. Varios estudios demuestran la amplia gama de casos de uso en el ámbito de la salud. Por ejemplo, en [94] los autores aplican metaheurísticas para obtener un subconjunto de moléculas mRNA y miRNA para la clasificación de diferentes subtipos de cáncer. En [102] los autores utilizan un enfoque híbrido que combina las metaheurísticas Dragonfly y Black Hole Algorithm con datos transcriptómicos para realizar FS e identificar subgrupos de riesgo para COVID-19 utilizando el biomarcador ACE2. En [4] se presenta un enfoque híbrido que combina las metaheurísticas Cuckoo Search, Flower Pollination Algorithm, Whale Optimization Algorithm y Harris Hawks Optimization para obtener biomarcadores con poder pronóstico en pacientes con enfermedad cardíaca y predecir la supervivencia en insuficiencia cardíaca. Además, en [123] se revisan más de cien artículos sobre enfoques de metaheurística utilizando datos de transcriptoma, centrándose en comparar las diferentes metaheurísticas con sus éxitos y fracasos.

Visto la gran cantidad de metaheurísticas existentes y sus aplicaciones, resulta indispensable contar con herramientas que las pongan a disposición de la comunidad, reduciendo la necesidad de conocimiento técnico por parte de los investigadores para hacer uso de ellas de manera eficaz.

Algunas herramientas fáciles de utilizar como Bioplat [17] (introducida en la Sección 1.1), una plataforma de escritorio basada en Java, fueron diseñadas para realizar FS para el descubrimiento de biomarcadores utilizando datos de mRNA a través de búsqueda exhaustiva (Blind Search) o la metaheurística Particle Swarm Optimization (PSO). De manera similar, Galaxy [46] se posiciona como una plataforma versátil en el ámbito de la bioinformática, proporcionando a los investigadores un entorno accesible y fácil de usar para construir, ejecutar y compartir flujos de trabajo bioinformáticos. Su interfaz gráfica basada en web simplifica la creación de flujos de trabajo complejos, permitiendo a los usuarios, incluso aquellos sin una amplia experiencia en programación, conectar y utilizar sin problemas una diversa gama de herramientas bioinformáticas. Esta herramienta tiene un historial comprobado de más de 5 años y se ha adaptado con éxito para su uso clínico en el análisis de datos de secuenciación de alto rendimiento y otros usos en un entorno de laboratorio clínico [28].

Dado que las metaheurísticas poblacionales son altamente paralelizables, se podría hacer uso de diferentes técnicas de distribución y paralelización que reduzcan el tiempo de ejecución necesario y ofrecer a los investigadores la posibilidad de ejecutar estos algoritmos en grandes volúmenes de datos. Sin embargo, tanto Bioplat como Galaxy operan dentro de

un marco de un solo subproceso, careciendo de este tipo de optimizaciones para las valiosas características que ofrecen.

Algunos estudios han tratado de solventar esta problemática, pero en el proceso presentan ciertas limitaciones. Por ejemplo, en [59] se aborda el proceso de distribución de cómputo para la optimización de PSO utilizando SVM como función de fitness a través de un clúster de computadoras. Utilizan servicios web con XML, SOAP, WSDL y UDDI para lograr la interoperabilidad en diferentes entornos. El uso de XML y SOAP, si bien facilita el intercambio de datos a través de la red, introduce ineficiencias de rendimiento durante la transmisión de datos. Además, la complejidad asociada en la configuración de SOAP, WSDL y UDDI plantea desafíos para adaptarse a las tendencias tecnológicas en evolución. Esta estructura multicapa complica el proceso de distribución en comparación con herramientas existentes como Spark, el cual ofrece un mayor rendimiento a través del procesamiento paralelo, un modelo de desarrollo simplificado y una mejor adaptabilidad a los requisitos cambiantes, respaldado por representaciones de datos eficientes.

Otros estudios, enfocados en el rendimiento y la generalización en la aplicación de metaheurísticas a diferentes esquemas y problemas, han introducido herramientas como Paradiseo [18] o Chameleon [130]. Estas ofrecen abstracciones para la distribución y el paralelismo de metaheurísticas considerando enfoques donde las tareas pueden intercambiarse entre workers de forma dinámica para evitar tiempos de inactividad. Sin embargo, consisten en librerías de C++ escritas desde cero y no hacen uso de tecnologías ya populares en el campo de la computación distribuida como Spark, lo que involucraría grandes cambios en algoritmos ya existentes y una curva de aprendizaje por parte de los investigadores.

Tanto en [137] como en [117] los autores presentan frameworks de predicción de rendimiento para trabajos que se ejecutan en Spark. Proponen una serie de fórmulas basadas en el consumo promedio de memoria, E/S y ciclos de CPU por etapa de una tarea que se ejecuta en el framework. Utilizan el número de etapas de una tarea por ejecutar e intentan predecir cuánto tiempo tardará en terminar. De manera similar, en [95] los autores predicen los tiempos de ejecución utilizando los modelos de regresión Multivariate Adaptive Regression Splines, Non-Negative Least Square y Least Square Boosting basados en características como el tamaño de la entrada de una tarea, las instrucciones o etapas que componen una tarea y la configuración del clúster utilizada. [63] también propone una solución utilizando un conjunto más pequeño de características y realizando las predicciones con los modelos de regresión Gradient Boosting Machine, Random Forest y Redes Neuronales. En [120] los autores entrenan modelos de atención y convolucionales a partir de la información de los registros de varias tareas ejecutadas en un clúster de Spark obteniendo buenas métricas en la predicción de los tiempos de ejecución para programas como PageRank, WordCount,

Regresión Logística y el cálculo de Pi. Sin embargo, todos estos enfoques solo consideran la ejecución de las instrucciones de Spark pero no tienen en cuenta casos más complicados donde se ejecutan funciones más complejas definidas por el usuario. Además, deja la mejora de la asignación de recursos al framework pero no propone una solución aplicable al usuario final para mejorar los tiempos de ejecución de su programa.

Dado todo el trabajo citado, es crucial contar con herramientas accesibles para la comunidad que pongan a disposición métodos bioinformáticos, pero que a su vez apliquen técnicas de paralelismo y distribución que optimicen la ejecución de los mismos. Multiomix [22] es una plataforma de código abierto que permite a los investigadores cargar conjuntos de datos o utilizar una base de datos pública (ya integrada en la plataforma) para realizar varias funcionalidades bioinformáticas como análisis de supervivencia con clasificadores como SSVM o RSF [62]. También proporciona funciones de FS para el descubrimiento de biomarcadores utilizando información de mRNA, miRNA, CNA y metilación del ADN. Estas técnicas permiten a los investigadores utilizar fácilmente algoritmos avanzados y están disponibles de forma gratuita para la comunidad. Multiomix utiliza las estrategias de balance de carga propuestas en esta tesis para optimizar los algoritmos de FS ofrecidos, algunas de estas estrategias emplean modelos de aprendizaje automático para predecir el tiempo de ejecución y optimizar la distribución de tareas de Spark a un nivel superior.

3.6. Ejecución distribuida de metaheurísticas

Si bien en esta tesis se evalúan e implementan en la plataforma Multiomix todos los algoritmos de FS mencionados en este capítulo, el enfoque principal se da en las metaheurísticas.

Cuando se utiliza alguna de ellas, sus agentes evalúan la llamada *función de fitness*, que en este contexto hace referencia al poder pronóstico/predictivo de un biomarcador. Esta función retorna una métrica (por ejemplo, C-Index, la exactitud, etc) que permite conocer y comparar objetivamente a los biomarcadores.

A lo largo de este trabajo se evaluará y optimizará la identificación de biomarcadores siguiendo el siguiente procedimiento:

1. Se selecciona el conjunto de datos con los datos de expresión de diferentes tipos de moléculas (mRNA, miRNA, CNA o sitios de metilación).
2. Se selecciona uno de los métodos de análisis de supervivencia definidos en la Sección 2.4.2 como función de fitness a evaluar.

3. Se selecciona una metaheurística para recorrer el espacio de soluciones. Es decir, recorrer de manera inteligente diferentes subconjuntos de características del conjunto de datos seleccionado.
4. La metaheurística irá ejecutando por cada uno de sus agentes, el método de análisis de supervivencia sobre un subconjunto de características para obtener su valor de fitness. En cada iteración de la metaheurística las tareas de evaluación serán distribuidas entre los workers de un clúster Spark para reducir considerablemente los tiempos de ejecución. Esta distribución será definida a partir de las estrategias de balance de carga definidas en la Sección 5.3 de esta tesis.
5. Los subconjuntos con mejor fitness al finalizar el proceso de FS serán considerados como nuevos biomarcadores de interés.

Capítulo 4

Multiomix

Multiomix [22] es una plataforma de código abierto diseñada con el propósito específico de acelerar el descubrimiento de biomarcadores y reguladores de expresión para blancos terapéuticos. Esta plataforma se centra en la integración de datos genómicos y epigenómicos, y su objetivo es el de proveer a la comunidad científica algoritmos bioinformáticos avanzados de manera gratuita y fácil de utilizar.

4.1. Descubrimiento de reguladores de expresión

Multiomix ofrece herramientas para la aceleración en el descubrimiento de reguladores de expresión. El proceso consiste en cruzar datos de expresión de genes y de GEMs de un grupo de pacientes para encontrar algún par GEM-gen cuyo coeficiente de correlación sea significativo, lo que supondría un mecanismo de modulación.

El proceso completo para la ejecución de un análisis de correlación dentro de la plataforma se explica en detalle en la Sección B.1. Una vez que un análisis finaliza correctamente, la plataforma ofrece un amplio abanico de herramientas para poder evaluar las combinaciones GEM-gen resultantes. Desde gráficos de correlación y propiedades estadísticas de los datos, hasta información externa como relaciones de las moléculas con drogas o enfermedades reportadas en la literatura. Se explica en detalle cada una de las herramientas disponibles en la Sección B.2.

El proceso de correlación requerido para el descubrimiento de posibles blancos terapéuticos es un proceso costoso en términos de rendimiento y consumo de memoria. Este problema se magnifica particularmente al utilizarse grandes conjuntos de datos, donde el tiempo requerido para evaluar millones de combinaciones GEM-gen se convierte en un desafío tecnológico significativo.

El problema central se encuentra en el cómputo y el ajuste de los p-valores necesarios, ya que los métodos de ajuste como Benjamini-Hochberg (BH) y Benjamini-Yekutieli (BY), requieren el ordenamiento de todos los p-valores obtenidos. Esta exigencia se traduce en un consumo masivo de memoria que en ocasiones conlleva al colapso de las herramientas existentes debido a la falta de recursos.

Para abordar esta problemática, se llevaron a cabo pruebas exhaustivas de rendimiento y uso de memoria, utilizando la librería WGCNA [79], librerías de Python como Pandas, Pandarallel, y en algunos casos, implementaciones en lenguaje Python puro con diversas técnicas de paralelización ad-hoc. Aún así, ninguna de las estrategias implementadas logró cumplir satisfactoriamente con los requerimientos de velocidad cuando se evaluaban conjuntos de datos de dimensiones considerables.

Para solventar de manera efectiva este desafío, se desarrolló desde cero de una herramienta innovadora llamada *Gene GEM Correlation Analysis* (GGCA). Esta fue concebida en el lenguaje de programación Rust, lo que asegura una ejecución altamente eficiente en términos de rendimiento. La elección de Rust no solo se basa en su capacidad para proporcionar una ejecución rápida, sino también en su compilador, que contribuye a reducir la incidencia de errores relacionados con tipos de datos y las dependencias de los mismos en entornos paralelos.

GGCA incorpora las denominadas funciones *lazy*, característica esencial para la gestión eficiente de grandes conjuntos de datos, ya que permiten trabajar cargando en memoria solo aquellos datos necesarios para la computación en curso. Así, la lectura de los datasets de GEM y gen se realizan fila a fila y no cargando ambos conjuntos completos en memoria en un mismo instante de tiempo.

Además, para afrontar el consumo excesivo de memoria durante el proceso de ordenamiento de p-valores necesario para su ajuste, GGCA hace uso de una técnica denominada *External sorting*. Esta posibilita ordenar un vector de elementos utilizando un iterador lazy en disco, eliminando la restricción inherente al tamaño de la memoria RAM.

El proceso completo de la librería se puede descomponer en los siguientes pasos (Figura 4.1):

1. Se realiza un preprocesamiento de los datasets de genes y GEMs, donde se parsean los datos como valores numéricos de punto flotante y se informa con errores claros al usuario si el formato del archivo es inválido (por ejemplo, separadores de celdas no soportado, o la presencia de valores que no son numéricos).

2. Se realiza una unión cruzada donde los valores de expresión de cada paciente del conjunto de datos de GEM son unidos a los valores de expresión de los pacientes del conjunto de datos de genes.
3. Por cada combinación GEM-gen se evalúa alguno de los tres métodos de correlación (Pearson, Spearman o Kendall) junto con el cómputo de sus p-valores.
4. Si el usuario optó por realizar un ajuste de p-valores con el método BH o BY, entonces se realiza el ordenamiento de todos los p-valores obtenidos utilizando un algoritmo de External Sorting.
5. Se realiza el ajuste de los p-valores. El valor final del procesamiento consta de una tabla con el GEM, el gen, el resultado de correlación, el p-valor y el p-valor ajustado.
6. Se realiza un filtro de los resultados utilizando diferentes criterios, como un umbral de correlación (conservando aquellas combinaciones cuyo valor estadístico de correlación se encuentre por encima de un valor específico), o manteniendo las N combinaciones con mayor valor de correlación.

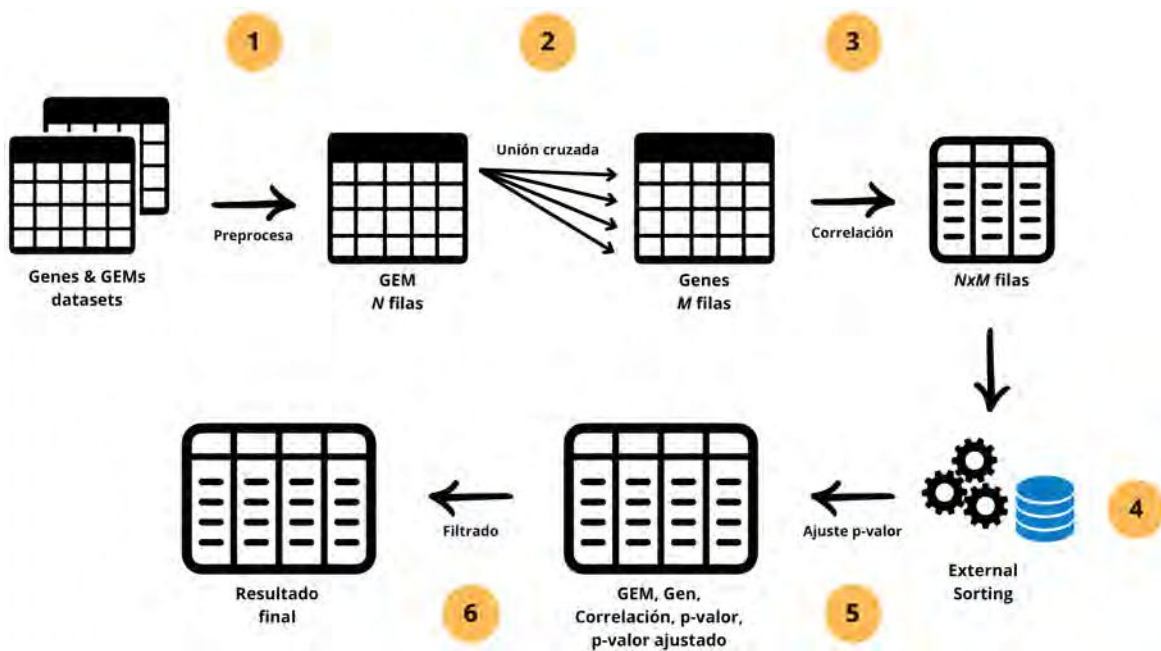


Figura 4.1 Proceso completo ejecutado por la librería GGCA.

Por último, GGCA utiliza dos librerías llamadas PyO3 y Maturin para extender su accesibilidad y facilitar su implementación en entornos de desarrollo versátiles.

La primera de ellas, PyO3, permite exponer la API implementada en Rust, permitiendo así que GGCA sea accesible no solo desde Rust, sino también desde Python. Esta interoperabilidad entre los dos lenguajes de programación se logra sin incurrir en un aumento significativo en los costos de mantenimiento. Gracias a PyO3, GGCA se vuelve fácilmente integrable en proyectos y flujos de trabajo que utilizan Python, proporcionando una capa de abstracción que facilita la interacción entre ambos lenguajes.

Por otro lado, Maturin se encarga del proceso de distribución y publicación de GGCA como una librería Python. Esto permite que GGCA pueda ser compartida a través del repositorio oficial de librerías de Python, conocido como PyPI (Python Package Index).

4.2. Identificación de biomarcadores

Multiomix tiene un panel dedicado a la gestión, optimización y evaluación de biomarcadores. Como se mencionó en la Sección 2.4, estructuralmente un biomarcador consiste en un conjunto de moléculas, la plataforma soporta cuatro tipos de dichas moléculas: mRNA, miRNA, CNA o sitios de metilación. De este conjunto de moléculas se sospecha la capacidad de pronosticar la probabilidad de supervivencia o evolución de un paciente para una enfermedad particular.

La plataforma ofrece un amplio conjunto de herramientas para poder crear biomarcadores por parte del usuario, y entrenar modelos de machine learning para analizar estadísticamente su poder pronóstico/predictivo, o realizar inferencia sobre diferentes bases de datos.

El proceso completo para la creación de un nuevo biomarcador se explica en detalle en la Sección B.3. Las herramientas para la evaluación del poder pronóstico de un biomarcador, y la obtención de información adicional de las moléculas que lo componen se desarrolla en la Sección B.4.

4.2.1. Modelos entrenados

Durante el proceso de FS para la creación de un biomarcador, múltiples algoritmos de ML (ya sean algoritmos de Clustering, SSVM o RSF) fueron entrenados para poder ejecutar la función de ajuste en cada uno de sus agentes y mejorar el poder pronóstico del biomarcador que estaba siendo optimizado. Una vez terminado el proceso, la plataforma almacena el subconjunto de moléculas que conforman el nuevo biomarcador, junto con la mejor métrica obtenida y dicho modelo de ML entrenado. Esto permite al usuario realizar nuevas validaciones estadísticas o inferencias sobre nuevos conjuntos de datos sin necesidad de entrenar un modelo nuevo.

La plataforma permite entrenar nuevos modelos utilizando diferentes parámetros y conjuntos de datos para poder realizar validaciones estadísticas e inferencia con las moléculas que conforman el biomarcador en estudio. Tanto el proceso de creación de modelos como los datos que se almacenan de ellos se explican en detalle en la Sección B.5.

4.2.2. Validaciones estadísticas

Con el fin de determinar el poder pronóstico/predictivo de un biomarcador, se dispone un panel de validaciones estadísticas con diferentes funciones que permiten al usuario comprender mejor la naturaleza de las moléculas que conforman el biomarcador, en relación con un conjunto de datos clínicos a elección.

Dichas herramientas permiten al usuario identificar las moléculas más relevantes del biomarcador en relación con la predicción de la ocurrencia de un evento. Todas las funciones disponibles se explican en detalle en la Sección B.6.

4.2.3. Inferencia

El último panel dentro de los detalles de un biomarcador es el de inferencia. Por inferencia se entiende al proceso en el que se hace uso de un modelo de ML para realizar predicciones sobre datos desconocidos (es decir, datos que no fueron utilizados para entrenar al modelo).

Esto nos permite utilizar los modelos entrenados en la práctica y evaluar que los mismos no hayan sufrido de sobre ajuste durante el proceso de entrenamiento. El proceso completo para la ejecución de inferencias se describe en detalle en la Sección B.7.

4.2.4. Multiomix AWS-EMR

Multiomix AWS-EMR es una plataforma minimalista diseñada para integrar de manera eficiente el servicio Elastic MapReduce (EMR) de Amazon Web Services (AWS). Su principal objetivo es proporcionar una capa de abstracción que permite a Multiomix interactuar con la API de AWS sin la necesidad de gestionar directamente la comunicación, las credenciales de autenticación, los scripts de configuración y algoritmos distribuidos que se ejecutan en la infraestructura de AWS-EMR.

Esta plataforma contiene los mismos algoritmos de FS que se implementan en Multiomix (como BBH, BPSO, GA, etc), pero con las adecuaciones necesarias para distribuir el cómputo de las metaheurísticas a través de los nodos del clúster de Spark con las técnicas abordadas en el Capítulo 5.

Multiomix AWS-EMR se encuentra disponible como plataforma de código abierto en GitHub ¹.

4.3. Abstracción en la obtención de datos

En el ámbito bioinformático, los investigadores a menudo se enfrentan a una serie de dificultades durante el proceso de investigación y análisis de datos. Estas deficiencias pueden manifestarse de diversas formas, incluyendo la obsolescencia de fuentes de datos, la falta de acceso a plataformas públicas, la falta de disponibilidad de código abierto y la falta de estándares para la representación de la información (los datos pueden estar incompletos o mal representados, con valores faltantes que se representan de diversas maneras, como cadenas vacías, "NA" o valores *null*). Cada una de estas falencias presenta desafíos significativos que afectan negativamente la eficiencia y la calidad de la investigación.

A continuación se detallan dos plataformas llamadas Modulector y BioAPI que fueron diseñadas con el objetivo de solventar estas carencias.

4.3.1. Modulector

Modulector [88] es una solución tecnológica que actúa como abstracción de información de miRNA y sitios de metilación del ADN, dos componentes cruciales en la regulación génica y epigenética. Esta plataforma de código abierto ² agiliza el acceso a información proveniente de diversas bases de datos, un desafío comúnmente enfrentado por investigadores en este campo.

Este sistema se distingue por su capacidad para integrar datos heterogéneos de los reguladores de expresión, procedentes de fuentes diversas (el listado de las bases de datos integradas se puede consultar en la Sección B.8.1), y normalizarlos a través de un proceso de importación de datos. Toda esta información se incorpora dentro de una bases de datos PostgreSQL a partir de una serie de scripts que elimina los datos faltantes, estandariza el valor de las celdas vacías a *null* y los guarda de manera tabular en una estructura fija.

Una de las características más notables de Modulector es su arquitectura basada en servicios REST, que proporciona una interfaz robusta y versátil para la recuperación de datos. Estos servicios REST, ofrecen una eficiente exposición de los conjuntos de datos estandarizados en formato JSON, un estándar ampliamente reconocido y utilizado en la transmisión de datos entre aplicaciones web. Además, dichos servicios ofrecen funciones de

¹<https://github.com/omics-datascience/multiomix-aws-emr>

²<https://github.com/omics-datascience/modulector>

paginación, ordenamiento, filtros y búsqueda, mejorando considerablemente la usabilidad por parte del usuario. Se pone a disposición, también, un archivo con la configuración predeterminada para poder realizar un despliegue de todos los servicios que conforman a Modulector (la base de datos, un servidor web con la lógica, y un servidor NGINX que actúa como proxy) con un único comando en cualquier entorno que el usuario requiera (permitiendo hacer uso de la plataforma con conocimiento técnicos mínimos y reduciendo la complejidad de la puesta en funcionamiento).

Por último, además de su capacidad para integrar y estandarizar datos, Modulector ofrece una funcionalidad adicional de gran valor para la comunidad científica: la posibilidad de suscribirse a notificaciones de nuevas publicaciones en PubMed relacionadas con miRNAs de interés. Esta función, que aprovecha la vasta base de datos de PubMed, permite a los investigadores mantenerse al día con los últimos avances en su área de estudio, proporcionando así una ventaja competitiva en la búsqueda de conocimiento y la identificación de nuevas direcciones de investigación.

La lista completa de los servicios que ofrece Modulector se puede consultar en la Sección B.8.2.

4.3.2. BioAPI

Al igual que Modulector, BioAPI también busca solucionar la problemática que representa obtener información de múltiples bases de datos de manera estandarizada y eficiente. La diferencia radica en que las bases de datos incorporadas por esta última corresponde a datos de genes y pathways, en vez de moléculas de miRNA o sitios de metilación.

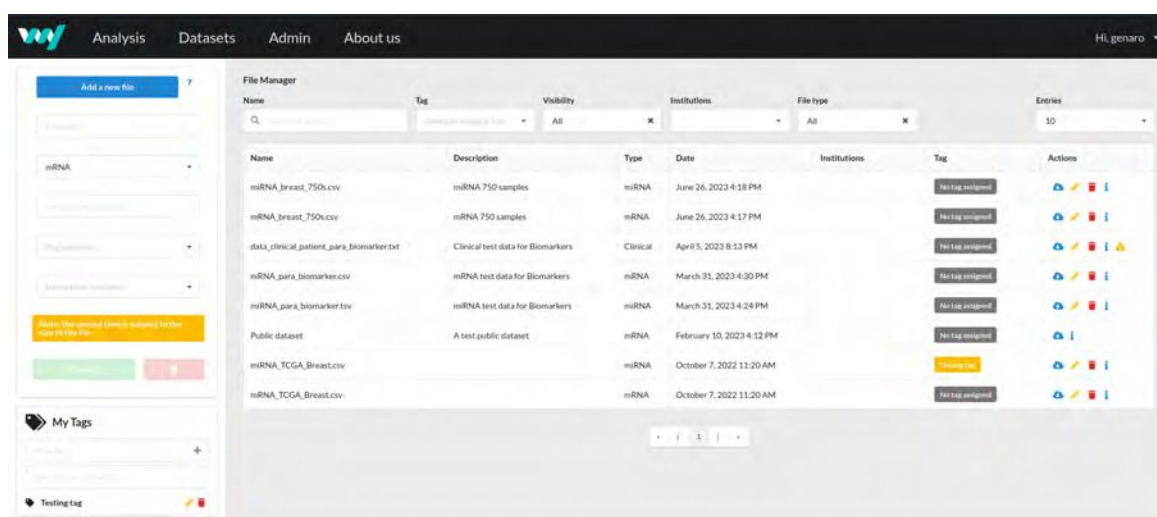
Las bases de datos integradas dentro de BioAPI abarcan una gran variedad de recursos destinados a facilitar diversos aspectos de la investigación y el análisis biológico. Al igual que Modulector, la forma en que se disponibiliza esta información es a través de APIs REST estandarizadas con funcionalidad extendida (como paginación, búsqueda y filtros).

La lista completa de bases de datos y los servicios incorporados se pueden consultar en las secciones B.8.3 y B.8.4 respectivamente.

4.3.3. Datos subidos por el usuario

Multiomix ofrece a los usuarios la capacidad de cargar sus propios datos de manera sencilla y eficiente. La plataforma realiza validaciones exhaustivas de los formatos de datos y verifica la consistencia de la información proporcionada. Para manejar conjuntos de datos de gran tamaño, Multiomix implementa una función de subida por lotes, permitiendo a los usuarios cargar grandes cantidades de datos de manera escalonada.

Además de la carga de datos, Multiomix facilita una gestión completa de los mismos a través de una tabla interactiva (Figura 4.2). Dicha tabla proporciona opciones de ordenamiento, filtros, paginación, búsqueda y categorización mediante el uso de etiquetas. Así, los usuarios pueden organizar y explorar sus datos de manera eficaz, optimizando la experiencia de gestión.



The screenshot shows the 'File Manager' interface of the Multiomix platform. It features a navigation menu with 'Analysis', 'Datasets', 'Admin', and 'About us'. A user profile 'Hi, genaro' is visible in the top right. The main area contains a table of datasets with the following columns: Name, Description, Type, Date, Institutions, Tag, and Actions. The table lists several datasets, including miRNA breast cancer samples, clinical patient data, and public datasets like TCGA Breast cancer data. Each row includes a 'Tag' button and a set of action icons (upload, edit, delete, share).

Name	Description	Type	Date	Institutions	Tag	Actions
miRNA_breast_750s.csv	miRNA 750 samples	miRNA	June 26, 2023 4:18 PM		No tag assigned	📄 🗑️ 🔄 📄
miRNA_breast_750s.csv	miRNA 750 samples	miRNA	June 26, 2023 4:17 PM		No tag assigned	📄 🗑️ 🔄 📄
data_clinical_patient_para_biomarker.txt	Clinical test data for Biomarkers	Clinical	Apr 9, 2023 8:13 PM		No tag assigned	📄 🗑️ 🔄 📄 ⚠️
miRNA_para_biomarker.csv	miRNA test data for Biomarkers	miRNA	March 31, 2023 4:30 PM		No tag assigned	📄 🗑️ 🔄 📄
miRNA_para_biomarker.tsv	miRNA test data for Biomarkers	miRNA	March 31, 2023 4:24 PM		No tag assigned	📄 🗑️ 🔄 📄
Public dataset	A test public dataset	miRNA	February 10, 2023 4:12 PM		No tag assigned	📄 🗑️ 🔄 📄
miRNA_TCGA_Breast.csv		miRNA	October 7, 2022 11:20 AM		Testing tag	📄 🗑️ 🔄 📄
miRNA_TCGA_Breast.csv		miRNA	October 7, 2022 11:20 AM		No tag assigned	📄 🗑️ 🔄 📄

Figura 4.2 Tabla para la gestión de los datasets accesibles por el usuario.

Además de permitir descargar datos directamente desde la tabla o compartirlos con colegas dentro de la plataforma. Esta capacidad de compartir se extiende a colaboradores de las mismas instituciones, fomentando la colaboración y facilitando el intercambio de información entre miembros de la comunidad en la plataforma Multiomix.

Cuando a un usuario se le comparte un dataset, puede utilizarlo para realizar análisis de correlación, crear biomarcadores o cualquier otra funcionalidad del sistema; pero no puede eliminarlo ni modificar su información. Esto facilita la replicabilidad de los experimentos para la comunidad científica.

4.3.4. cBioPortal

cBioPortal [25][33][43] es una plataforma inicialmente desarrollada en el Memorial Sloan Kettering Cancer Center (MSK) disponible bajo una licencia de código abierto a través de GitHub³. La responsabilidad del desarrollo y mantenimiento del software ha sido asumida por un equipo multiinstitucional que incluye a MSK, el Dana Farber Cancer Institute, Princess Margaret Cancer Centre en Toronto, Children's Hospital of Philadelphia, Caris Life

³<https://github.com/cBioPortal/>

Sciences, The Hyve y SE4BIO en los Países Bajos, así como Bilkent University en Ankara, Turquía.

Name	Description	Version	Sync	mRNA	miRNA	CNA	Methy.	Clinical P.	Clinical S.	State	Actions
Acral Melanoma (TCGA, Genome Res 2017)	Liang et al. Genome Res 2017	2		●	●	●	●	●	●	●	⚙️
Acral Melanoma (TCGA, Genome Res 2017)	Liang et al. Genome Res 2017	1	04/20/2021	■	■	■	■	■	■	■	⚙️
Acute Myeloid Leukemia (TCGA, Firehose Legacy)	TCGA, Firehose Legacy	2		●	●	●	●	●	●	●	⚙️
Acute Myeloid Leukemia (TCGA, Firehose Legacy)	TCGA, Firehose Legacy	1	04/20/2021	●	●	●	●	●	●	●	⚙️
Acute Myeloid Leukemia (TCGA, NEJM 2013)	TCGA, NEJM 2013	2		●	●	●	●	●	●	●	⚙️
Acute Myeloid Leukemia (TCGA, NEJM 2013)	TCGA, NEJM 2013	1	04/20/2021	●	●	●	●	●	●	●	⚙️
Acute Myeloid Leukemia (TCGA, PanCancer Atlas)	TCGA, Cell 2018	2		●	●	●	●	●	●	●	⚙️
Acute Myeloid Leukemia (TCGA, PanCancer Atlas)	TCGA, Cell 2018	1	04/20/2021	●	●	●	●	●	●	●	⚙️
Adrenocortical Carcinoma (TCGA, Firehose Legacy)	Adrenocortical Carcinoma (TCGA, Firehose Legacy)	2		●	●	●	●	●	●	●	⚙️
Adrenocortical Carcinoma (TCGA, Firehose Legacy)	Adrenocortical Carcinoma (TCGA, Firehose Legacy)	1	04/20/2021	●	●	●	●	●	●	●	⚙️

Figura 4.3 Tabla para la gestión de los datasets curados de cBioPortal por parte de un administrador del sistema. Los datasets importados estarán accesibles para todos los usuarios, independientemente de su rol.

cBioPortal está diseñada para la exploración interactiva de conjuntos de datos genómicos del cáncer multidimensionales. El objetivo fundamental es reducir significativamente las barreras entre datos genómicos complejos y los investigadores del cáncer. Logra esto al ofrecer un acceso rápido, intuitivo y de alta calidad a perfiles moleculares y atributos clínicos derivados de proyectos de genómica del cáncer a gran escala. La misión central de cBioPortal es capacitar a los investigadores, proporcionándoles las herramientas necesarias para traducir estos conjuntos de datos ricos en información en conocimientos biológicos y aplicaciones clínicas.

Uno de los recursos más valiosos es su extenso repositorio de datos de mRNA, miRNA, CNA, metilación de ADN y atributos clínicos⁴. Multiomix implementó un mecanismo de importación asincrónico que permite a sus administradores realizar la gestión completa de los estudios disponibles en cBioPortal para poder importarlos dentro de la plataforma (Figura 4.3). Esto permite sincronizar diferentes estudios en paralelo, y permite versionarlos, para no dejar inválidos los análisis que se hayan evaluado con versiones viejas de los datos.

Así, Multiomix permite realizar cualquier tipo de experimento con datos curados de primera calidad, sin necesidad de forzar al usuario a tener que descargar dichos datasets desde su fuente original y realizar un preprocesamiento para poder usarlos dentro de Multiomix.

⁴<https://www.cbioportal.org/datasets>

Esto facilita aún más la integración de datos y la replicabilidad de los experimentos dentro de los grupos de investigación.

4.4. Democratización de la tecnología

La elección de hacer tanto Multiomix como sus plataformas complementarias (Modulador, BioAPI, Multiomix AWS-EMR y GGCA) de código abierto no solo demuestra un compromiso con la transparencia, sino que también contribuye significativamente a la democratización de estas herramientas en el ámbito de la biomedicina.

Al adoptar este tipo de licencias, se permite a la comunidad científica acceder, estudiar, modificar y distribuir libremente el software. Esto elimina barreras significativas de acceso, fomentando la participación de investigadores, científicos y desarrolladores de todo el mundo.

La buena documentación, el cumplimiento de estándares y la creación de mecanismos de guías en el software facilita la comprensión y el uso del software, permitiendo a investigadores de diversos niveles de experiencia aprovechar al máximo estas herramientas. Cumplir con estándares establecidos asegura la interoperabilidad y la calidad del software, mientras que las guías proporcionan orientación valiosa para la implementación, el desarrollo y uso de las funciones frente a diferentes escenarios.

La estructura completa de la plataforma se puede apreciar en la Figura 4.4.

4.5. Dificultades técnicas solventadas

Multiomix ha demostrado ser una solución integral al abordar diversas dificultades técnicas en el análisis de datos biológicos. En primer lugar, simplifica la interacción del usuario al proporcionar una abstracción efectiva hacia funciones estadísticas avanzadas. Esto incluye la realización de análisis de correlación con métodos diversos, la aplicación de distintos enfoques para ajustes de p-valor, la selección de características mediante diversas técnicas y metaheurísticas, así como la implementación de algoritmos de inferencia para el análisis de supervivencia de los pacientes.

Además, Multiomix facilita la interpretación de los resultados a través de gráficos intuitivos. Estos gráficos son fundamentales para comprender y comunicar eficazmente los hallazgos derivados del análisis de datos biológicos complejos.

En términos de gestión de datos, Multiomix ofrece herramientas avanzadas como tablas de alta performance que admiten paginación, filtros, búsqueda y ordenamiento. La plataforma también simplifica la incorporación y compartición de datos al permitir la carga de archivos

validados para su uso interno y la importación de datos curados desde fuentes externas como cBioPortal.

Para optimizar los procesos, Multiomix ha desarrollado herramientas especializadas, como GGCA, que posibilita el análisis eficiente de correlación con conjuntos de datos extensos. Además, se ha puesto a prueba un framework (se presenta en detalle en el Capítulo 5) que implementa balance de carga en Spark para la distribución y paralelismo durante la ejecución de las metaheurísticas, asegurando un uso eficiente de los recursos computacionales y un tiempo de ejecución sin igual en el ámbito. Adicionalmente, todos los procesos que pueden requerir de un largo tiempo de ejecución se ejecutan de manera asincrónica en segundo plano, informando en tiempo real al usuario sobre el estado actual de los mismos. Esto permite a los investigadores seguir interactuando con el sistema sin bloquear la interfaz

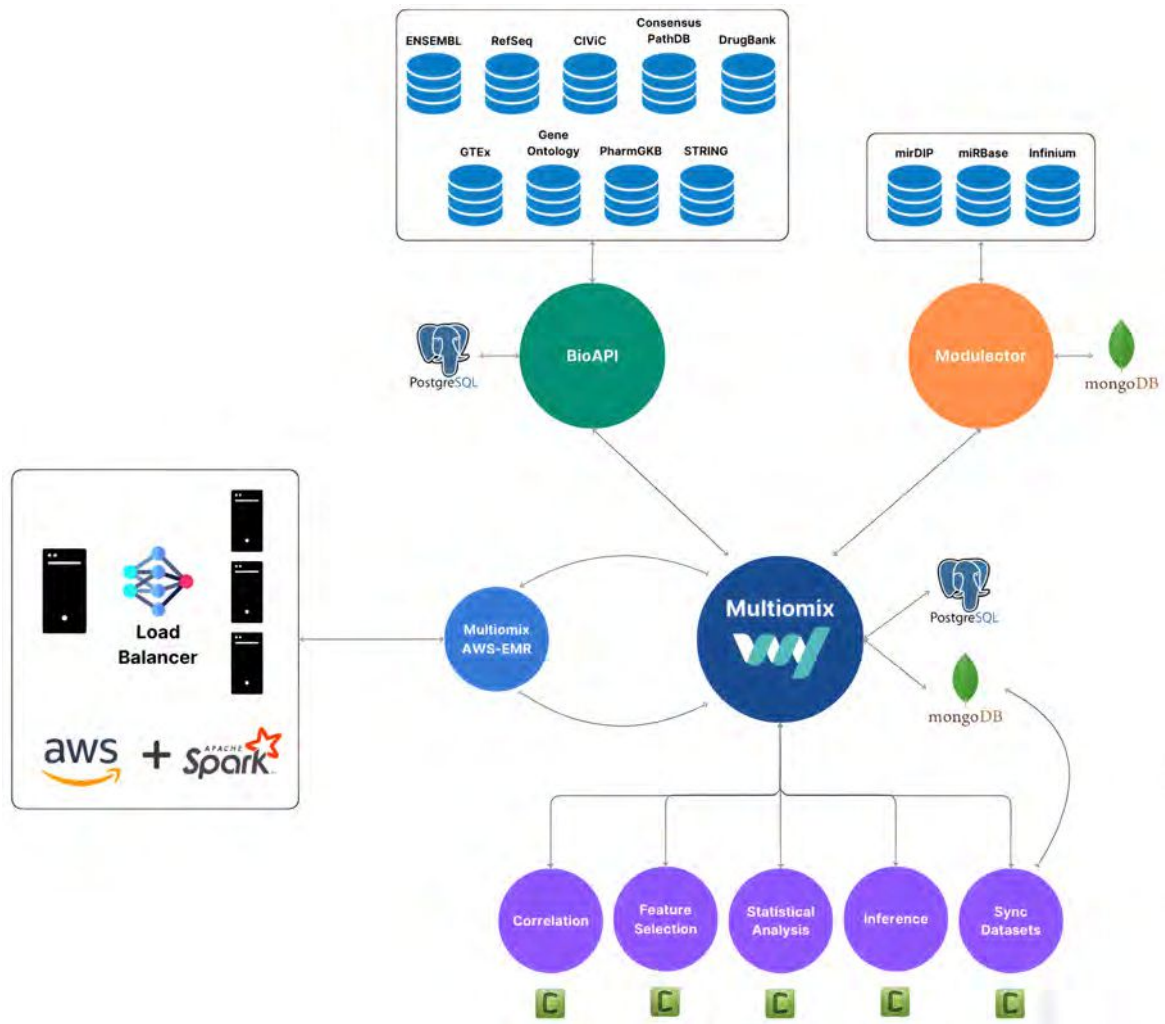


Figura 4.4 Estructura completa de Multiomix y todo su ecosistema complementario. Los íconos verdes con una letra "C" corresponden a la herramienta Celery.

y ejecutar varias funciones en paralelo, lo que brinda una experiencia de uso eficiente y agradable.

Finalmente, Multiomix ha logrado una abstracción efectiva en la obtención de información bajo demanda mediante las plataformas Modulector y BioAPI. Estas plataformas permiten la integración de diversas fuentes de datos de miRNA, genes, metilación del ADN y pathways, proporcionando una visión estandarizada y enriquecida para la investigación.

Capítulo 5

Optimización de metaheurísticas en Spark

Este capítulo se enfoca en la distribución de cómputo para la aceleración de metaheurísticas en Spark. Para lograrlo, se desarrolla un framework que hace uso de estrategias de balance de carga inteligentes con el fin de reducir el tiempo de ejecución de los procesos de FS. Se abordan diferentes estrategias de distribución, con diferentes grados de complejidad, hasta llegar al aporte original de esta tesis, donde se realiza el balance de carga utilizando el modelo Histogram-based Gradient Booster (HGB) [121] para predecir el costo de cada tarea a evaluar y así realizar la distribución de manera eficiente.

Cuando se habla de tareas o agentes de metaheurísticas se está haciendo referencia al mismo concepto, ya que cada agente debe realizar un cómputo para evaluar su función de fitness, lo que también podría representarse como una tarea a ejecutar por parte de Spark en uno de sus workers.

La elección de Spark como el framework objetivo, reconocido por su popularidad en la distribución computacional, elimina la necesidad de realizar cambios importantes en los algoritmos existentes, lo que facilita su integración y evita los costos asociados con la reimplementación de herramientas específicas del dominio. También ofrece la ventaja de ser generalizable a cualquier metaheurística en la que los agentes puedan ejecutarse en paralelo. Todas estas ventajas solventan la mayoría de las falencias que presentan algunas de las herramientas actuales mencionadas en la Sección 3.5.

Las estrategias presentadas en este capítulo forman parte de las optimizaciones implementadas en Multiomix a la hora de ejecutar un proceso de FS por parte de un usuario (explicado en detalle en la Sección 4.2).

5.1. Apache Spark

Apache Spark [145] es un framework de programación para procesamiento de datos distribuidos diseñado para ser rápido y de propósito general. Su virtud radica en cubrir una amplia gama de cargas de trabajo que antes requerían sistemas distribuidos diferentes, como procesamiento batch, algoritmos iterativos, consultas interactivas y procesamiento streaming. Spark es flexible y ofrece APIs en Python, Java, Scala, SQL y R, con un buen rendimiento en todas ellas, permitiendo trabajar con datos estructurados según las necesidades del usuario. Se integra fácilmente con otras herramientas Big Data, especialmente con Hadoop, ejecutándose en clústeres Hadoop y accediendo a datos almacenados en HDFS (un sistema de archivos distribuidos), S3 (servicio de almacenamiento de alta velocidad de AWS) y otras fuentes.

En cuanto a su funcionamiento, Spark es un motor de procesamiento distribuido que orquesta, distribuye y monitoriza aplicaciones con múltiples tareas de procesamiento. Utiliza un principio jerárquico primario-secundario (maestro-esclavo) para coordinar el nodo master y los nodos workers. Spark es más rápido que Hadoop, procesando trabajos hasta 100 veces más rápido en memoria y 10 veces más rápido en disco [145]. Su sistema de particiones permite distribuir y paralelizar el procesamiento de datos, optimizando la ejecución de consultas y almacenamiento en caché en memoria para un procesamiento más rápido. Spark se destaca por su flexibilidad, permitiendo escribir aplicaciones en varios lenguajes y trabajar con diferentes plataformas de datos como AWS S3, Apache Cassandra [78] o HBase.

A través de su abstracción de datos llamada Conjuntos de Resilient Distributed Datasets (RDD), Spark permite dispersar el cómputo entre diferentes nodos que componen un clúster de computadoras.

Un aspecto fundamental de todo el flujo de ejecución es la definición de particiones. Las particiones en Spark consisten en un mecanismo a partir del cual se asignan tareas a ser ejecutadas a los nodos workers. Si dos tareas (que en este contexto serían dos agentes de la metaheurística poblacional evaluando dos biomarcadores diferentes) fueron asignadas a la misma partición, entonces ambas serán delegadas al mismo worker del clúster. Esto permite una gestión de grano fino para designar conjuntos equilibrados de cómputos para todos los workers.

Si bien Spark fue concebido como un framework de procesamiento masivo de datos. Al considerarse este mecanismo de particiones puede verse también como un simple distribuidor de tareas, ya que permite generar una delegación de cómputo de manera simple, abstrayendo al desarrollador de todos los tecnicismos que conlleva la programación distribuida y paralela.

5.2. Balance de carga

El balance de carga es un tema fundamental en el campo de la computación distribuida y los sistemas de alta disponibilidad. A lo largo de los años, se han desarrollado diversas técnicas y enfoques para abordar este desafío, algunas de ellas son las siguientes:

- **Algoritmos basados en Round-Robin:** Estos algoritmos distribuyen las solicitudes de manera secuencial entre los servidores disponibles, asegurando una distribución equitativa de la carga. Estos algoritmos también reciben el nombre de *Algoritmos equitativos*.
- **Algoritmos basados en ponderación:** Estos algoritmos asignan pesos a los servidores en función de sus capacidades, lo que permite una distribución más eficiente de la carga.
- **Algoritmos basados en el estado del servidor:** Estos algoritmos monitorean el estado de los servidores (como la utilización de CPU, memoria y ancho de banda) y redirigen las solicitudes a los servidores menos cargados.
- **Algoritmos basados en aprendizaje automático:** Estos algoritmos utilizan técnicas de aprendizaje automático, como redes neuronales o algoritmos de aprendizaje por refuerzo, para predecir la carga de los servidores y tomar decisiones de enrutamiento más inteligentes.

La decisión de qué técnica usar está estrechamente ligada al problema a resolver y no resulta trivial, ya que debe analizarse el tipo de datos a manejar, los recursos disponibles y la tecnología empleada para realizar el cómputo.

En esta tesis se desarrollan nuevas estrategias de balance de carga en un clúster Spark para poder distribuir el cómputo de las metaheurísticas anteriores, reduciendo así el tiempo de ejecución requerido para obtener su resultado.

5.3. Estrategias de balance de carga propuestas

Como se explicó en más detalles en la Sección 3.4, una metaheurística poblacional es una técnica de optimización inspirada en procesos naturales como la evolución. Mantiene una población de soluciones diversas (también llamadas *agentes*), donde cada una representa una posible respuesta a un problema específico. A través de ciclos repetidos, crea nuevas soluciones, evalúa su efectividad y guía la población hacia mejores respuestas, eventualmente

convergiendo en la mejor solución posible dentro de un tiempo y recursos prácticos. La naturaleza de las metaheurísticas poblacionales podría enmarcarse dentro del pseudocódigo del Algoritmo 5.1.

Algoritmo 5.1 Pseudocódigo de una metaheurística poblacional.

```

1 Crear una poblacion inicial de  $N$  agentes aleatorios
2 Evaluar la funcion de fitness de cada nuevo agente
3 Mientras no se cumple el criterio de parada
4   Reemplazar agentes viejos por nuevos, basado en algun criterio
5   Evaluar la funcion de fitness de cada agente
6 Retornar la mejor solucion encontrada

```

El rendimiento podría mejorarse distribuyendo el cálculo de las líneas 2 y 5 del Algoritmo 5.1 siempre que la metaheurística utilizada no requiera que se respete un orden de ejecución específico para sus agentes.

Siendo N el número de agentes en la metaheurística, se crea un arreglo de N elementos que contiene (<ID>, <subconjunto de genes a seleccionar aleatoriamente>), luego se convierte en RDD a través del método *parallelize* de Spark. Se aplican dos funciones a ese RDD para definir dónde se ejecutará cada una de sus N tareas: *partitionBy* (que permite asignar una partición a la tarea) y *mapPartitions* (que define cómo se calculan las tareas de cada partición).

Asumiendo que el algoritmo que implementa la estrategia de balance de carga se llame *partition_assign_and_compute*, el Algoritmo 5.1 puede ser reescrito como el Algoritmo 5.2.

Algoritmo 5.2 Pseudocódigo de una metaheurística poblacional con la ejecución de evaluación distribuida.

```

1 Crear una poblacion inicial de  $N$  agentes aleatorios
2 partition_assign_and_compute()
3 Mientras no se cumple el criterio de parada
4   Reemplazar agentes viejos por nuevos, basado en algun criterio
5   partition_assign_and_compute()
6 Retornar la mejor solucion encontrada

```

En esta sección, y en respuesta a los inconvenientes planteados en la Sección 3.5, se presentan dos estrategias novedosas para mejorar el rendimiento de una metaheurística poblacional mediante la gestión de los tiempos de ejecución y de inactividad de los workers de un clúster Spark.

Estas estrategias de balance de carga serán comparadas con una implementación de la clásica técnica de distribución equitativa modificada para el problema a evaluar. Esta

estrategia recibe el nombre de *Equally Distributed (ED)* en esta tesis, y no utiliza predicción del tiempo de ejecución para el equilibrio de carga, ofreciendo simplicidad y potencial a cambio de un rendimiento subóptimo.

En contraste, la segunda estrategia *Distribution Based on Predictions (DBP)* aprovecha un modelo HGB para la predicción del tiempo de ejecución, permitiendo una distribución de tareas más justa entre los workers del clúster. Finalmente, la estrategia *Predictive Execution Load Algorithm with Delay Optimization (PELADO)* aborda las limitaciones de DBP, particularmente en entornos de clústeres heterogéneos, incorporando aprendizaje adaptativo y equilibrio de carga dinámico.

Teniendo en consideración las definiciones de las estrategias de balance de carga de la Sección 5.2, las presentadas en esta tesis resultan una solución híbrida, ya que si bien ED está basada en Round-Robin, tanto DBP como PELADO se basan en una distribución ponderada que utiliza aprendizaje automático y consideran datos como el estado de los servidores para la delegación de tareas durante el proceso de FS.

La elección de Spark como framework para la distribución de tareas se debe a que el mismo abstrae todos los aspectos técnicos del paralelismo y la distribución, permitiendo a los usuarios implementar fácilmente metaheurísticas en lenguajes populares como Python o Scala sin tener que preocuparse por los problemas técnicos involucrados. El sistema de particiones introducido en la Sección 5.1 permite implementar técnicas como ED con pocas líneas de código y concentrarse en aspectos más complejos como el entrenamiento del modelo HGB para la implementación de las estrategias más avanzadas como DBP y PELADO.

5.3.1. Modelo de predicción del tiempo de ejecución de tareas

Para implementar tanto las estrategias DBP como PELADO, se requiere un modelo de predicción del tiempo de ejecución de las tareas. Después de una experimentación sustancial, se optó por la técnica Gradient Boosting [40][41], que está diseñada para mejorar la precisión predictiva de los modelos. Su utilidad radica en construir modelos robustos a través de la combinación secuencial de múltiples modelos, típicamente árboles de decisión. La razón para seleccionar este tipo de modelo es que se destacan por su capacidad de manejar datos heterogéneos, capturar patrones complejos y su resistencia al sobreajuste, lo que lo convierte en una buena opción cuando hay poca disponibilidad de datos de entrenamiento. El modelo utilizado es el *HistGradientBoostingRegressor* de la popular librería Scikit Learn [105].

En esta estrategia, el tiempo de ejecución de cada agente se predice utilizando el modelo HGB. Para cada tipo de tarea a ejecutar (agrupamiento con Clustering o regresión con SSVM, ambos explicados en la Sección 6.3) se entrenó un HGB diferente utilizando diferentes

Tabla 5.1 Características utilizadas en los modelos HGB para predecir el tiempo de ejecución de tareas de análisis de supervivencia utilizando algoritmos de Clustering o SSVM.

	Parámetro	Posibles valores
Clustering	Número de pacientes	<i>número</i>
	Número de características	<i>número</i>
	Número de clústeres	<i>número</i>
	Algoritmo usado	K-Means
		Spectral
Métrica de score	Log Likelihood	
	C-Index	
SSVM	Número de pacientes	<i>número</i>
	Número de características	<i>número</i>
	Kernel	Linear
		Polynomial
		RBF
	Optimizador	AVLTree
RBTree		

características (Tabla 5.1). En ambos casos, el atributo a predecir es siempre el tiempo de ejecución.

Los datos para el entrenamiento de ambos modelos se obtuvieron de experimentos recopilados en la plataforma Multiomix utilizando el servicio EMR de AWS (introducido en la Sección 4.2.4). La recopilación de datos fue posible porque la implementación de la estrategia ED ya está implementada en Multiomix y permitió la generación del conjunto de datos de entrenamiento utilizando datos curados de cBioPortal en un corto tiempo. Para el entrenamiento de HGB para las tareas de Clustering se utilizaron 47790 registros, mientras que el de SSVM se utilizaron 23430 registros.

Vale la pena mencionar que el modelo HGB fue seleccionado después de un exhaustivo proceso de Grid Search y validación cruzada (CV por sus siglas en inglés) donde también se evaluaron modelos de regresión como Regresión Lineal, modelos SVM (con kernels Polinomial, RBF y Sigmoide) y redes neuronales; todos con diferentes combinaciones de parámetros particulares. En el caso de esta tesis, HGB fue el que obtuvo las mejores métricas durante el proceso de selección.

Los parámetros de HGB probados durante el proceso de GridSearch se pueden observar en la Tabla 5.2. Los valores en la última columna son los utilizados para entrenar el modelo que se usa como predictor durante las estrategias DBP y PELADO.

Tabla 5.2 Valores explorados durante el proceso de GridSearch para cada parámetro de HGB. La última columna muestra el valor que obtuvo las mejores métricas.

Parámetro	Valores evaluados	Valor óptimo
<i>max_iter</i>	[100, 200, 300, 400, 500]	300
<i>max_depth</i>	[2, 3, 4, 5, 6]	4
<i>learning_rate</i>	[0.01, 0.05, 0.1, 0.2, 0.3]	0.2
<i>max_leaf_nodes</i>	[31, 41, 51, 61, 71]	41
<i>min_samples_leaf</i>	[10, 20, 30, 40, 50]	20

5.3.2. Estrategia "Equally Distributed"

La estrategia ED consiste en la clásica estrategia de distribución equitativa, y sirve como base para la evaluación de las otras dos estrategias propuestas (DBP y PELADO). Utilizando el mecanismo de particiones de Spark, los diferentes agentes a evaluar entre cada iteración de la metaheurística se distribuyen de manera equitativa (en términos del número de tareas) entre todos los workers del clúster. En esta estrategia si, por ejemplo, se deben evaluar 30 agentes y hay 3 workers en el clúster, 10 agentes serán delegados a cada worker. La delegación se realiza de manera ordenada, asignando los primeros 10 a un worker, el segundo lote de 10 agentes a otro, y el último lote al worker restante.

Por lo tanto, asumiendo N agentes y W workers en el clúster, se generarán W particiones en Spark con N/W agentes a evaluar para cada una. De esta manera, cada worker recibe la misma cantidad de subconjuntos independientes de características a evaluar (Figura 5.1).

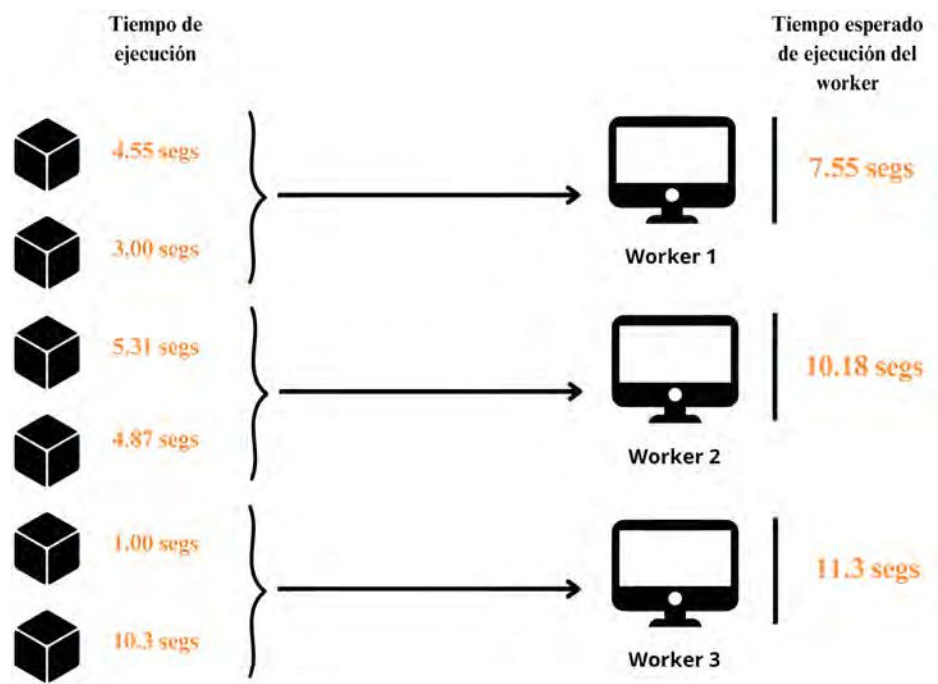


Figura 5.1 En el esquema ED, las tareas se dividen de manera equitativa. Se reparten la misma cantidad de tareas en cada nodo, ignorando el tiempo de ejecución de cada una de ellas.

El proceso de distribución equitativa entre los nodos de un clúster Spark podría definirse con el Algoritmo 5.3.

Algoritmo 5.3 Algoritmo de la estrategia ED.

```

def partition_f(key):
    return key * n_workers // len(stars_subsets)

def map_partition(fit_function, records):
    for key, elem in records:
        yield key, fit_function(elem)

stars_parallelized = sc.parallelize(stars_subsets)
result = stars_parallelized \
    .partitionBy(n_workers, partitionFunc=partition_f) \
    .mapPartitions(
        lambda records: map_partition(fit_function, records),
        preservesPartitioning=True
    ).collect()

```

Donde *star_subsets* es la lista de N agentes a computar, *n_workers* es el número de nodos workers en el clúster, *fit_function* es la función de fitness para evaluar (es decir, aquella que devuelve la métrica obtenida por el algoritmo de Clustering o el SSVM), *records* es el subconjunto de características a evaluar para cada tarea/agente, y *map_partition* es una función que devuelve el resultado de cada tarea que compone una partición. La razón de esto último es que las tareas se ejecutan en orden ya que Spark paraleliza la ejecución de las tareas de una partición y esto podría sesgar las métricas de tiempo durante los experimentos. En producción, esta sección de *mapPartitions* podría eliminarse para mejorar el paralelismo, teniendo en cuenta que se requeriría una configuración exhaustiva del nivel de paralelismo asignado a la ejecución de la CV, ya que la mala configuración de los recursos podría degradar el rendimiento final.

5.3.3. Estrategia "Distribution Based on Predictions"

La estrategia ED distribuye el cómputo entre los workers de un clúster de Spark basándose en el número de tareas, asignando en orden las particiones para cada una de estas tareas. El problema con este enfoque es que los subconjuntos de características de cada tarea son definidos de manera aleatoria, por lo que podría darse el caso de que las tareas asignadas para la primera partición sean las que tomen más tiempo en ejecutarse en comparación con el resto de las particiones, generando mucho tiempo de espera inactivo entre los workers que recibieron particiones con tareas que requerían menos tiempo para finalizar su ejecución (Figura 5.2).

El tiempo de ejecución de la función de fitness está correlacionado con el tamaño de los datos utilizados para el entrenamiento y el número de características a evaluar [20][21][19]. Por lo tanto, una estrategia más eficiente sería predecir el tiempo de ejecución aproximado requerido por cada una de las tareas y realizar la asignación de particiones a través de un algoritmo de Bin Packing [90] que se encarga de distribuir eficientemente objetos de diferentes tamaños en un número determinado de grupos o *bins*. En este caso particular, los bins son los workers del clúster, generando una asignación óptima de tiempos de ejecución a todos los nodos. Esta nueva estrategia de balance de carga recibe el nombre DBP y se representa gráficamente en la Figura 5.3. El código Python se puede apreciar en el Algoritmo 5.4.

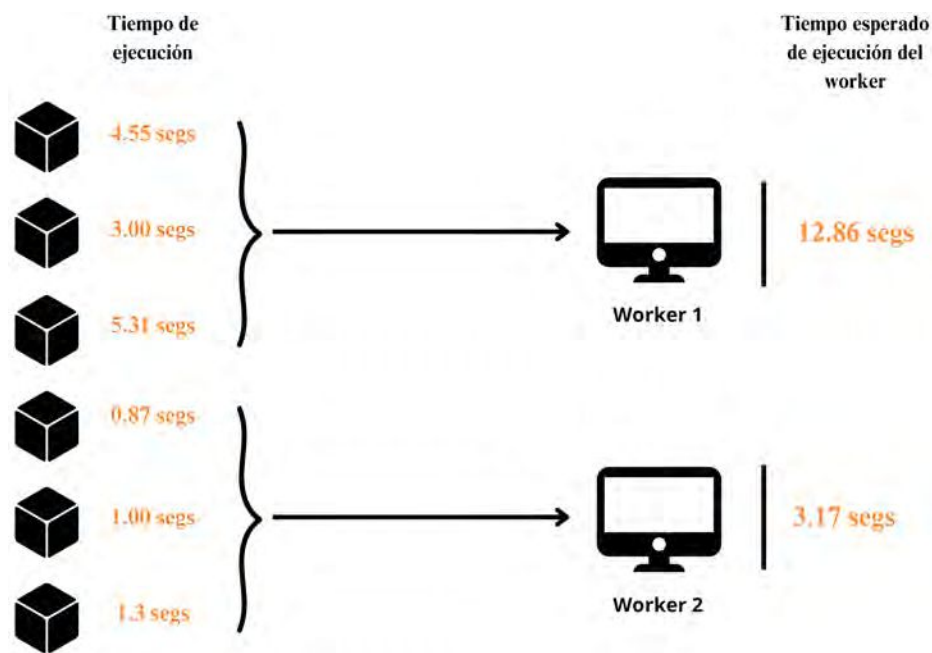


Figura 5.2 Representación gráfica del problema con la estrategia ED: las tareas asignadas a una partición requieren mucho más tiempo que las asignadas a otra partición, generando que un worker quede mucho tiempo ocioso esperando a que el que recibió mas carga de trabajo termine.

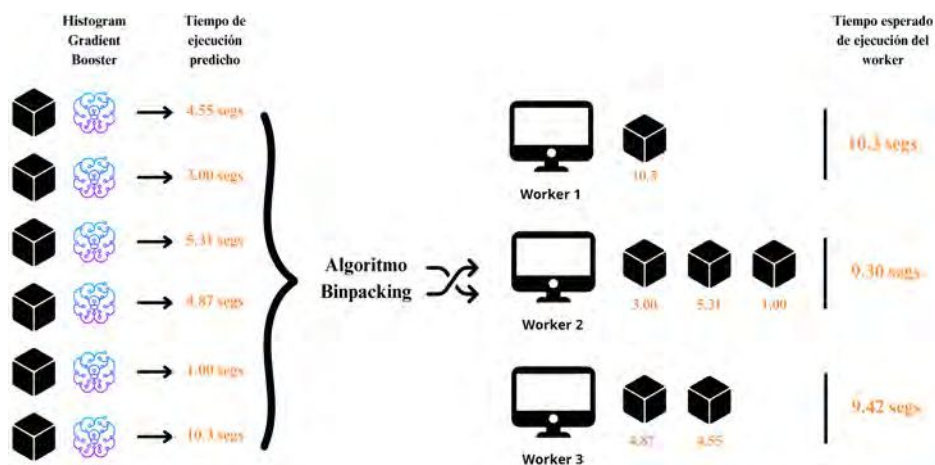


Figura 5.3 En DBP, el modelo HGB es responsable de predecir el tiempo de ejecución de las tareas. Luego, utilizando el algoritmo Bin Packing, se realiza una distribución de las tareas basada en los tiempos predichos.

 Algoritmo 5.4 Algoritmo de la estrategia DBP.

```

def map_partition(fit_function , records):
    for key, elem in records:
        yield key, fit_function(elem)

# Asigna todas las particiones.
predicted_times = predict(stars_subsets)
stars_and_times = {
    k: v
    for (k, v) in zip(range(n_stars), predicted_times)
}
bins = binpacking.to_constant_bin_number(
    stars_and_times ,
    number_of_workers
)
stars_partition = generate_stars_and_partitions_bins(bins)

# Igual que ED, pero cambiando la funcion partitionBy
stars_parallelized = sc.parallelize(stars_subsets)
result = stars_parallelized \
    .partitionBy(
        n_workers ,
        partitionFunc=lambda key: stars_partition[key]
    ).mapPartitions(
        lambda records: map_partition(fit_function , records) ,
        preservesPartitioning=True
    ).collect()
  
```

En el código, la función *predict* utiliza el modelo HGB descrito en la Sección 5.3.1 para hacer la inferencia de los tiempos de ejecución para cada una de las tareas. Este resultado se convierte en un diccionario y se pasa como parámetro a la función *to_constant_bin_number* de la librería de Python Binpacking ¹, que asigna de manera uniforme al número de workers en el clúster de Spark en el momento de la ejecución. Finalmente, la función *generate_stars_and_partitions_bins* devuelve un diccionario donde cada agente de BBH tiene asignada la partición específica en la que debe ser ejecutado.

¹<https://github.com/benmaier/binpacking>

5.3.4. Estrategia "Predictive Execution Load Algorithm with Delay Optimization"

Una vez que un algoritmo se ha optimizado mediante la distribución de las tareas por el tiempo de ejecución aproximado requerido, pueden surgir dos problemas:

- Clúster heterogéneo:** las estrategias presentadas hasta ahora consideran un clúster donde los workers tienen la misma capacidad computacional (clúster homogéneo) y, por lo tanto, requerirían un tiempo similar para la misma tarea. Sin embargo, en un entorno de producción típico, este factor podría no cumplirse debido a diferencias del hardware de los nodos o a diferentes cargas de trabajo (es decir, procesos externos) en el momento de la ejecución de la metaheurística.
- Restricciones de delegación de particiones:** Spark no proporciona un mecanismo para definir en qué worker se ejecuta una partición específica, si no que se basa en ciertas métricas como la localidad de los datos para la delegación de las particiones a los nodos. Esto podría generar que varias particiones sean asignadas al mismo nodo, dejando ociosos algunos workers del clúster. Esto resulta en un problema al enfrentar el inconveniente del ítem anterior (donde sería ideal asignar una partición con tareas más costosas a un worker con mejor rendimiento computacional). Además, en aquellas ocasiones donde múltiples particiones son asignadas a un mismo worker, se genera un 100% de tiempo de inactividad en aquellos que no recibieron tareas.

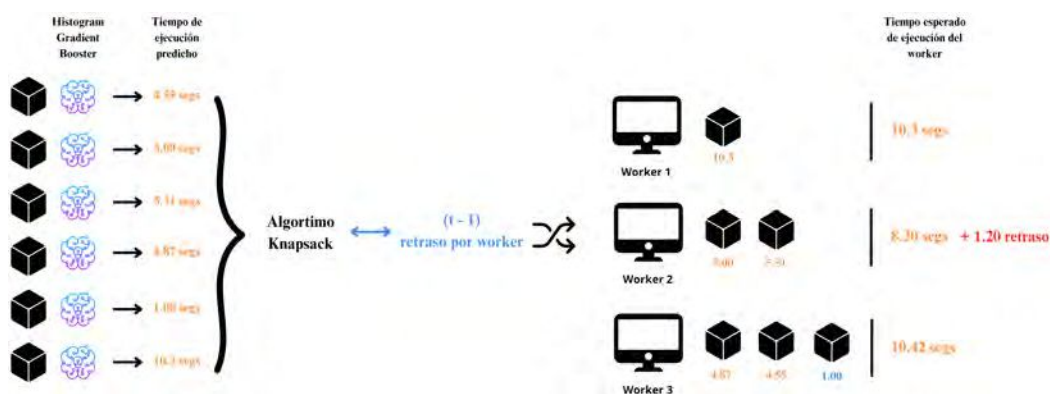


Figura 5.4 La estrategia PELADO introduce a la estrategia DBP la consideración de los retrasos presentados por los workers en la iteración anterior ($t-1$), de esta manera se puede realizar una mejor asignación de las tareas haciendo uso de Knapsack, lo que permite distribuir las tareas entre workers con diferentes capacidades.

Para resolver ambos problemas, se introduce la estrategia PELADO, que además de predecir los tiempos de ejecución de cada tarea, toma en consideración el retraso presentado

por cada uno de los workers en la iteración anterior para equilibrar la carga de trabajo de manera más eficiente (Figura 5.4).

Al comienzo de la estrategia, todos los workers tienen la misma capacidad, la cual se calcula a partir de la división del tiempo de ejecución predicho por el modelo HGB por el número de workers en el clúster. Este valor tendrá como valor mínimo el tiempo más bajo predicho para evitar que las capacidades de los workers estén por debajo del tiempo requerido por las tareas.

Luego, se realiza un ajuste para cada worker de acuerdo con la proporción del retraso que causó en la iteración anterior. Finalmente, entra en juego el proceso de redistribución a partir del algoritmo Knapsack [90] (utilizando el método heurístico MTHM definido en [89]), donde cada una de las tareas es delegada a los workers. El algoritmo completo en el lenguaje de programación Python se puede apreciar en el Algoritmo 5.5.

Algoritmo 5.5 Algoritmo de la estrategia DBP.

```
def pelado_strategy(agents, workers_delay):
    # Diccionario particion -> lista de agentes
    # Empieza por 1 ya que 0 significa que
    # la particion no fue asignada.
    worker_tasks_partitions = {
        i + 1: []
        for i in range(N_WORKERS)
    }
    n_elements = len(agents)

    # Genera un array de beneficio = 1 por cada agente.
    tasks = [1 for _ in range(n_elements)]

    # Obtiene los pesos de cada agente en funcion
    # de la prediccion.
    # El peso minimo es 1 para evitar la division por 0.
    weights = [max(round(predict(agent)), 1) for agent in agents]

    while len(tasks) > 0:
        # Inicia todos los workers con la misma capacidad:
        # (la suma de todos los pesos) / N_WORKERS.
        weight_per_worker_equal = round(
            sum(weights) / N_WORKERS
        )
```

```
min_weight = min(weights)
weight_per_worker = max(
    weight_per_worker_equal,
    min_weight
)

# Genera la misma capacidad para cada worker.
capacities = assign_capacities(
    weight_per_worker,
    N_WORKERS
)

# Aplica retardo (si es necesario).
if workers_delay is not None:
    for idx in range(N_WORKERS):
        delay_for_worker = get_worker_delay(worker_id)
        capacities[idx] *= delay_for_worker
        capacities[idx] = max(round(capacities[idx]),
                               min_weight)

# Asignar tareas a las workers maximizando
# los beneficios.
res_knp = compute_knapsack(
    tasks,
    weights,
    capacities
)

# Almacena el resultado del algoritmo Knapsack.
store_knapsack_result(res_knp, worker_tasks_partitions)

# Comprueba la tarea con bin_number != 0
# (ya que 0 significa que el elemento no fue
# asignado a ningun worker).
tasks_to_remove = get_already_assigned_tasks(res_knp)

# Elimina las tareas ya asignadas.
remove_assigned_tasks(
    tasks_to_remove,
```



```
        tasks ,  
        weights  
    )
```

```
# Asigna la particion/worker correspondiente a cada agente.  
assign_partitions_to_agents(agents , worker_tasks_partitions)
```

Donde *agents* es un RDD con los agentes de la metaheurística que se evaluarán para asignarles las particiones. *worker_delays* es un diccionario con el identificador de los workers del clúster como clave y el retraso que sufrieron en la iteración anterior como valor. En la primera iteración, este diccionario estará vacío; a partir de la segunda iteración, tendrá una proporción de retraso para cada worker. El retraso se calcula como TP/TR , donde TP es la suma del tiempo predicho para todas las tareas del worker y TR es la suma del tiempo de ejecución para todas las tareas del worker.

En la Figura 5.5 se puede apreciar la ejecución de las primeras dos iteraciones utilizando la estrategia de balance de carga PELADO. Al empezar, se realiza la predicción de los tiempos de ejecución requeridos para cada tarea utilizando el modelo HGB, luego se realiza una distribución ponderada de tareas utilizando el algoritmo Knapsack.

Es importante destacar que al ser la primera iteración (t_0) no hay ningún tipo de penalización para los workers del clúster. Una vez que las tareas fueron distribuidas y ejecutadas por los workers, se divide el tiempo de ejecución esperado (la suma de todas las predicciones de tiempos para las tareas de cada worker) por el tiempo real que conllevó (la suma de todos los tiempos reales para las tareas de cada worker). El resultado de esta división servirá en la siguiente iteración (t_1) para penalizar a aquellos workers cuya performance en tiempo haya estado por debajo de lo esperado, y para asignar más carga de trabajo a aquellos workers que rindieron mejor de lo esperado.

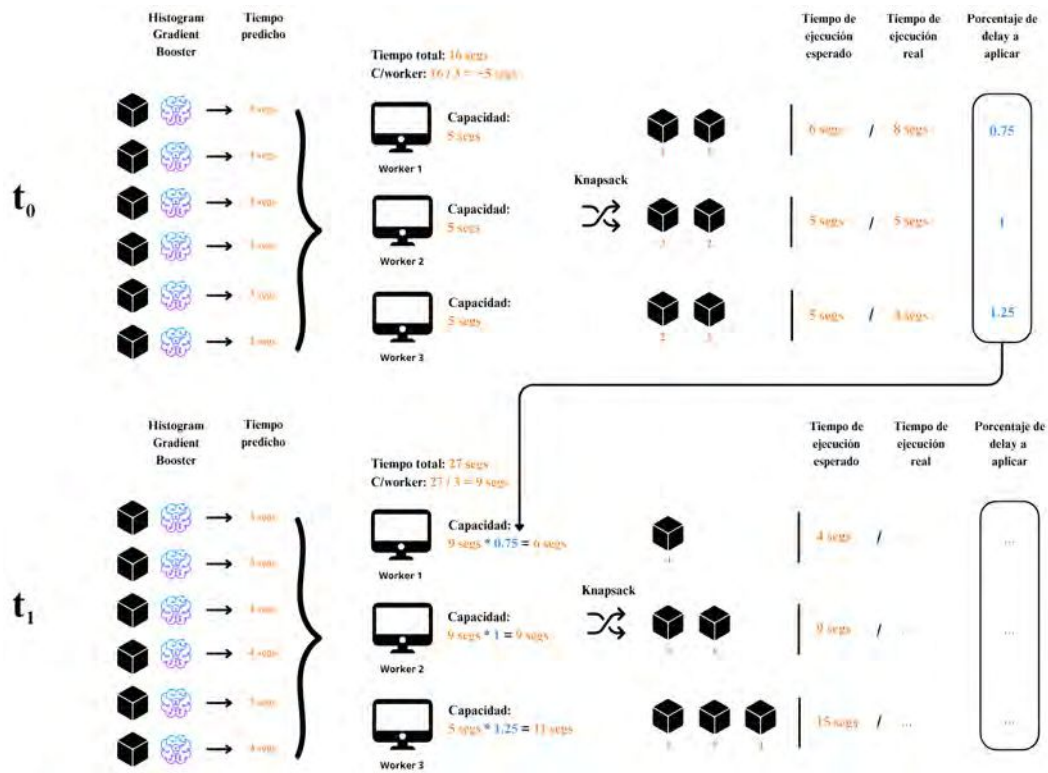


Figura 5.5 Representación gráfica de la ejecución de las primeras dos iteraciones de la estrategia de balance de carga PELADO.

5.3.5. Generalización y aplicación del framework

La principal ventaja del enfoque presentado es su alta generalización y facilidad de aplicación a una amplia gama de problemas de optimización basados en metaheurísticas. Este framework ofrece una solución flexible y eficiente para acelerar la ejecución de metaheurísticas poblacionales en entornos distribuidos, independientemente de la naturaleza específica del problema o la función de fitness utilizada. Para aplicar este framework a cualquier otra metaheurística poblacional o función de fitness, los pasos a seguir serían:

1. Elegir la función de fitness a evaluar durante la ejecución de la metaheurística.
2. Entrenamiento del modelo de predicción: en caso de utilizar las estrategias DBP o PELADO, se necesita entrenar un modelo de aprendizaje automático (ya sea un modelo HGB o cualquier otro que ofrezca mejores predicciones) para predecir los tiempos de ejecución de las tareas. Si se opta por la estrategia ED este paso puede omitirse.
3. Integración con el framework: una vez que se tienen estos componentes, la integración con el framework es relativamente sencilla. Basta con reemplazar el bloque

de código correspondiente a la ejecución de los agentes con el código de la estrategia elegida (ED, DBP o PELADO). Esto implica hacer uso de la función *partition_assign_and_compute()* del pseudocódigo 5.2 con el algoritmo de la estrategia seleccionada.

La ventaja clave de este enfoque es que, una vez realizada esta adaptación inicial, el investigador o desarrollador puede beneficiarse inmediatamente de las estrategias de balance de carga avanzadas sin necesidad de preocuparse por los detalles de la implementación distribuida o la optimización del rendimiento. El framework se encarga automáticamente de distribuir las tareas de manera eficiente entre los nodos del clúster, adaptándose dinámicamente a las características del hardware y a la carga de trabajo.

Además, al ser un framework de Spark permite aprovechar el vasto ecosistema de librerías disponibles para Python. Spark, al ser compatible con dicho lenguaje a través de PySpark, ofrece la posibilidad de integrar fácilmente bibliotecas populares de ciencia de datos y aprendizaje automático como NumPy, Pandas, Scikit-learn, Scikit-Surv, TensorFlow, PyTorch, entre otras. Esta característica es particularmente valiosa, ya que permite a los investigadores y desarrolladores utilizar herramientas y algoritmos familiares dentro del entorno distribuido, sin necesidad de reescribir código existente o aprender nuevas APIs.

Capítulo 6

Experimentación

En este capítulo se presenta una descripción de los experimentos llevados a cabo durante el desarrollo de esta tesis. Se incluyen mediciones de tiempos de ejecución de varios algoritmos de machine learning utilizando distintos conjuntos de datos. Se presentan resultados de métricas como exactitud, precisión, exhaustividad y F1-Score para estos algoritmos.

Además, se evalúan estrategias de balance de carga como ED, DBP y PELADO, tanto en un simulador como en un clúster Spark real. El capítulo ofrece análisis detallados de los tiempos de ejecución, tiempos de inactividad y ejecución obtenidas con estas estrategias, proporcionando una visión completa de su eficacia en diferentes escenarios.

6.1. Hardware y software

Todos los experimentos se realizaron en un clúster Spark que consta de un único nodo master y tres nodos workers. Los cuatro nodos poseen Ubuntu 20.04 LTS, una CPU Intel(R) Core(TM) i3-4160 funcionando a 3,60 GHz y 8 GB de RAM. En cuanto al software, se utilizó la versión de Spark 3.1.1.

6.2. Mediciones de tiempos y métricas

En [20] se realizó un análisis exhaustivo de la performance de los cuatro modelos de machine learning implementados en la librería Apache Spark ML: Máquina de vectores de soporte (SVM), Random Forest (RF), Naïve Bayes (NB) y Redes Neuronales (MLP). Estos modelos se evaluaron sobre diferentes cantidades de características (parámetro que se define en el artículo como ω) para estudiar el rendimiento frente a diferentes tamaños de información.

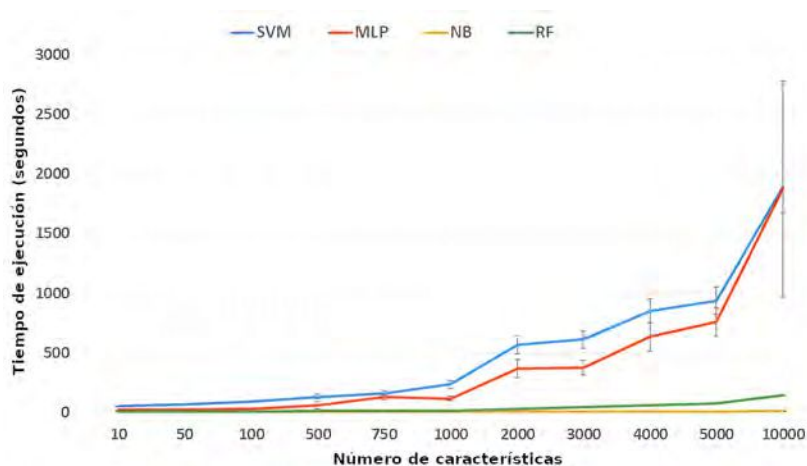


Figura 6.1 Media y desviación típica de los tiempos de ejecución de los cuatro algoritmos estudiados para los distintos subconjuntos de características utilizadas en el dataset de entrenamiento.

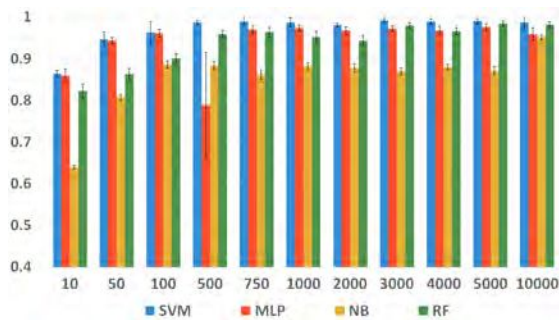
Se llegó a la conclusión de que los algoritmos que requieren más tiempo de cómputo son SVM y MLP (Figura 6.1). Sin embargo, en experimentos con un ω bajo, estos dos algoritmos obtuvieron los mejores modelos (Figura 6.2). RF resultó ser el algoritmo que requirió el menor tiempo de ejecución para lograr modelos con alta tasa de predicción.

En los algoritmos de FS, donde se debe ejecutar un algoritmo de clasificación cientos o miles de veces con diferentes valores de ω , la dicotomía presentada resulta en un gran desafío, especialmente considerando que, al seleccionar un subconjunto de características con alto poder pronóstico/predictivo, generalmente se espera que el subconjunto resultante sea lo más pequeño posible.

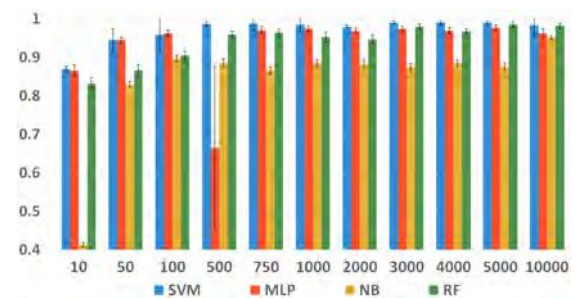
Los resultados demostraron la necesidad de encontrar un equilibrio entre un algoritmo con bajo tiempo de ejecución mientras se logran buenos modelos con un ω pequeño. Por un lado, se reportó que SVM produce muy buenos modelos independientemente de ω , pero requiere mayor tiempo de cómputo a medida que este valor crece. Por otro lado, RF requiere poco tiempo de ejecución, pero para obtener modelos similares a los de SVM, necesita miles de características.

Uno de los aspectos a mejorar del trabajo es que el conjunto de datos utilizado tiene una cantidad significativa de características ($\omega > 15000$), pero solo unas pocas muestras (253). Resulta interesante realizar experimentos con conjuntos de datos con más muestras, y estudiar el rendimiento de los algoritmos de clasificación en escenarios donde el volumen de datos sea mayor.

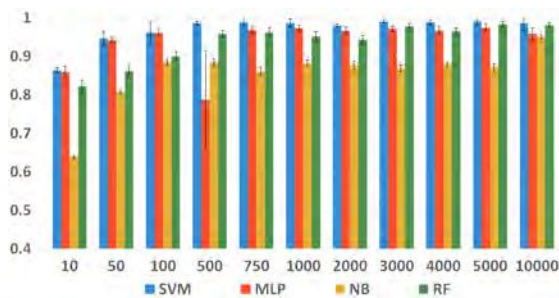
En [21] se midieron el tiempo de ejecución, la especificidad y el AUC para cuatro algoritmos de Spark ML, variando la cantidad de características en el conjunto de datos de



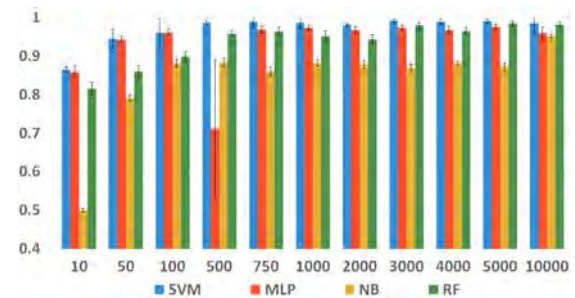
(a) Exactitud.



(b) Precisión.



(c) Exhaustividad.



(d) F1-Score.

Figura 6.2 Media y desviación estándar de las métricas de exactitud, precisión, recuperación y medida F1 de los cuatro algoritmos estudiados para los distintos subconjuntos de características utilizados.

entrenamiento; resultando en una extensión del trabajo previamente mencionado. Todas las comparaciones entre las métricas obtenidas por los algoritmos se compararon mediante una prueba de hipótesis para medir si sus diferencias son estadísticamente significativas o no.

Las figuras 6.3, 6.4 y 6.5 muestran que el algoritmo que tuvo el peor desempeño para todas las métricas es NB, aunque logró superar al resto de los modelos presentados en tiempo de ejecución. Los algoritmos que requieren más tiempo de cómputo son SVM y MLP; sin embargo, con ω bajo, estos dos algoritmos son los que obtuvieron los mejores modelos. RF resultó ser el algoritmo que requirió el menor tiempo de ejecución para lograr modelos con una alta tasa de predicción.

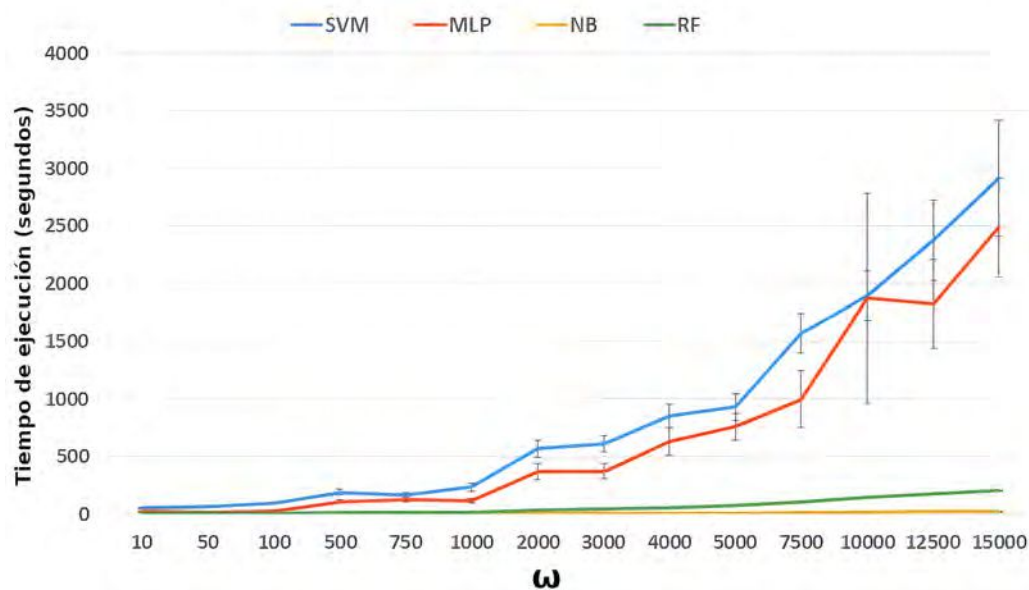


Figura 6.3 Media y desviación típica de los tiempos de ejecución de los cuatro algoritmos estudiados para los 14 valores de ω .

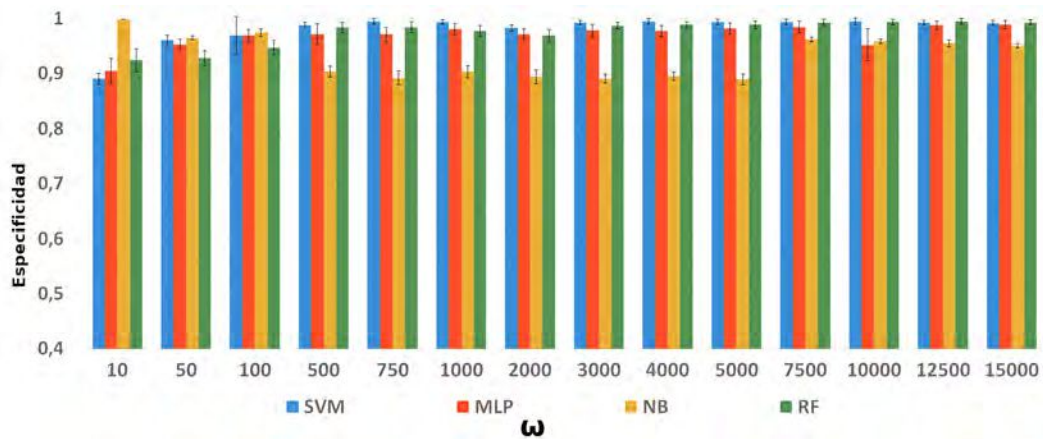


Figura 6.4 Media y desviación típica de la especificidad de los cuatro algoritmos estudiados para los 14 valores de ω .

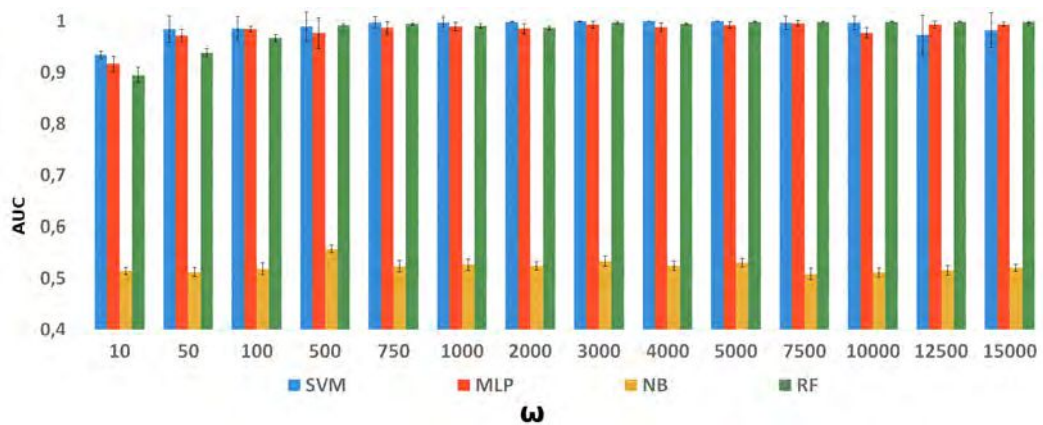


Figura 6.5 Media y desviación típica del AUC de los cuatro algoritmos estudiados para los 14 valores de ω .

SVM produce muy buenos modelos, independientemente del valor de ω en el conjunto de datos, pero requiere mucho tiempo de cómputo a medida que crece ω . En contraste, RF requiere poco tiempo de ejecución, pero miles de características para lograr un modelo que se asemeje al logrado por SVM (Tabla 6.1).

Tabla 6.1 Tabla de significancia estadística. Los valores de las filas representan, en orden respectivo, las veces que el modelo de la izquierda obtuvo mejores, iguales y peores resultados que el modelo de la columna para cada métrica dada. Para cada métrica y modelo, se destaca en negrita el mejor resultado.

Métrica	Modelo	RF	SVM	MLP
Tiempo	NB	14, 0, 0	14, 0, 0	14, 0, 0
	RF	-	14, 0, 0	14, 0, 0
	SVM	-	-	0, 1, 13
AUC	NB	0, 0, 14	0, 0, 14	0, 0, 14
	RF	-	2, 4, 8	9 , 2, 3
	SVM	-	-	9 , 4, 1
Especificidad	NB	3, 0, 11	2, 1, 11	3, 1, 10
	RF	-	1, 4, 9	10 , 2, 2
	SVM	-	-	12 , 1, 1
Precisión	NB	0, 0, 14	0, 1, 13	1, 0, 13
	RF	-	2, 3, 9	6 , 2, 6
	SVM	-	-	8 , 5, 1
Exhaustividad	NB	0, 0, 14	0, 0, 14	0, 1, 13
	RF	-	1, 2, 11	6 , 2, 6
	SVM	-	-	9 , 5, 0
Exactitud	NB	0, 0, 14	0, 0, 14	0, 1, 13
	RF	-	1, 2, 11	6 , 2, 6
	SVM	-	-	9 , 5, 0
F1-Score	NB	0, 0, 14	0, 1, 13	1, 0, 13
	RF	-	2, 1, 11	6 , 2, 6
	SVM	-	-	10 , 3, 1

En [19], ya poniendo en foco realizar un óptimo balance de carga en Spark para la aceleración en la identificación de biomarcadores, se evaluaron los tiempos de ejecución y el C-Index obtenidos por el modelo SSVM para diferentes valores del parámetro ω de un conjunto de datos de genes. Los experimentos se llevaron a cabo con diferentes kernels (Linear, Polynomial, RBF y Cosine) y optimizadores (RBTree y AVLTree), permitiendo una comparación directa entre diferentes configuraciones disponibles. Se concluye que, de los dos optimizadores disponibles, RBTree generó una degradación significativa en los tiempos de ejecución durante la fase de entrenamiento del modelo en todos los casos. En cuanto a la configuración del kernel, se observaron peores métricas de tiempo cuando el conjunto de datos tenía menos de 100 características (Figura 6.6). Aun así, los tiempos de iteración se mantuvieron directamente correlacionados con el tamaño del conjunto de datos para los kernels Linear y Cosine (Figura 6.8), dejando en claro que el alto tiempo requerido con pocas características se debe a la cantidad de iteraciones que el modelo necesita para alcanzar la convergencia. No fue el caso con los kernels Polinomial y RBF, cuyos tiempos por iteración también fueron erráticos para mediciones con pocas características de entrenamiento. Se analizó el comportamiento del modelo durante la etapa de entrenamiento, y se observó que la naturaleza desequilibrada de los datos (una gran cantidad de datos censurados) y la imposibilidad de separarlos linealmente resultaron en métricas peores a medida que aumentaba el tamaño del conjunto de datos (Figura 6.9).

Como era de esperar, la relación entre el tiempo de prueba y ω sigue una tendencia lineal (Figura 6.7), ya que no hay ninguna función de error que optimizar en el proceso, y el cálculo de la inferencia sólo consiste en la ejecución de la función kernel para el SSVM que está siendo evaluado.

Los resultados obtenidos muestran que, para subconjuntos con pocos genes, los kernels lineales y polinomiales tienen un tiempo de ejecución más corto y un C-Index aceptable. Por otro lado, si la cantidad de genes en el subconjunto excede los 2000, entonces los kernels Cosine y RBF obtienen un mejor equilibrio entre tiempo de ejecución y poder pronóstico. Ambas pruebas muestran que no resulta trivial obtener modelos que predicen el tiempo de ejecución de un modelo con N características, lo que permitiría mejorar la distribución de tareas en un entorno distribuido. Aunque los resultados presentados en este trabajo se obtienen del análisis de una base de datos específica, en el futuro se puede incluir la replicación del experimento en más bases de datos para determinar si existe variabilidad.

Los experimentos se llevaron a cabo en un clúster Spark para reducir el tiempo requerido para completarlos, además de establecer una configuración base que permitió llevar estos experimentos a otros modelos y continuar avanzando en el desarrollo de algoritmos que fueron puesto en producción en la plataforma Multiomix para la aplicación de técnicas de FS

con datos de supervivencia. Medir estos tiempos permitieron realizar un análisis exhaustivo y establecer una estrategia de balance de carga dentro del clúster de computadoras para obtener una distribución óptima del cómputo y reducir los tiempos de ejecución cuya evaluación se detallará a continuación.

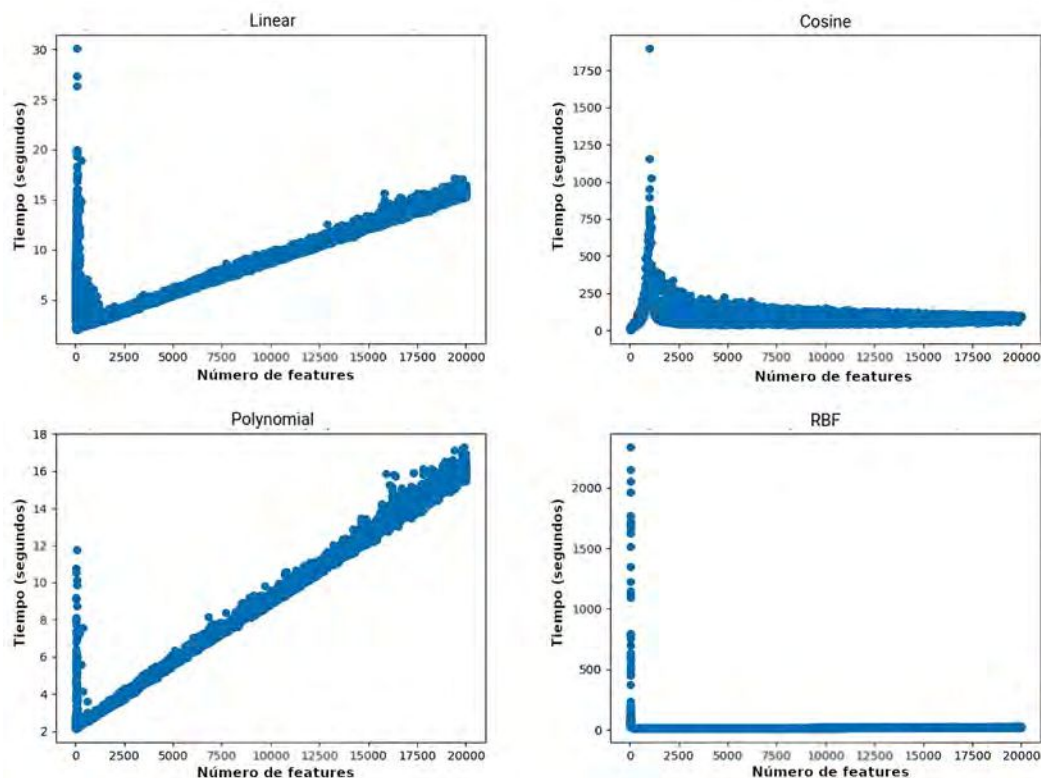


Figura 6.6 Tiempo total de ejecución del proceso de CV con 10 folds para los kernels Linear, Cosine, Polynomial y RBF.

6.3. Evaluación de las estrategias de balance de carga

En esta sección se detallan los experimentos llevados a cabo para la evaluación de las estrategias de balance de carga ED, DBP y PELADO (introducidas en la Sección 5.3).

6.3.1. Simulador de distribución de tareas

Debido a la limitación de Spark para la delegación de particiones de grano fino mencionada en la Sección 5.3.4, actualmente no se puede implementar la estrategia PELADO dentro del framework. Por lo tanto, se desarrolló un simulador que actúa como una prueba de concepto para dicha estrategia, gracias a la capacidad del simulador para definir qué worker

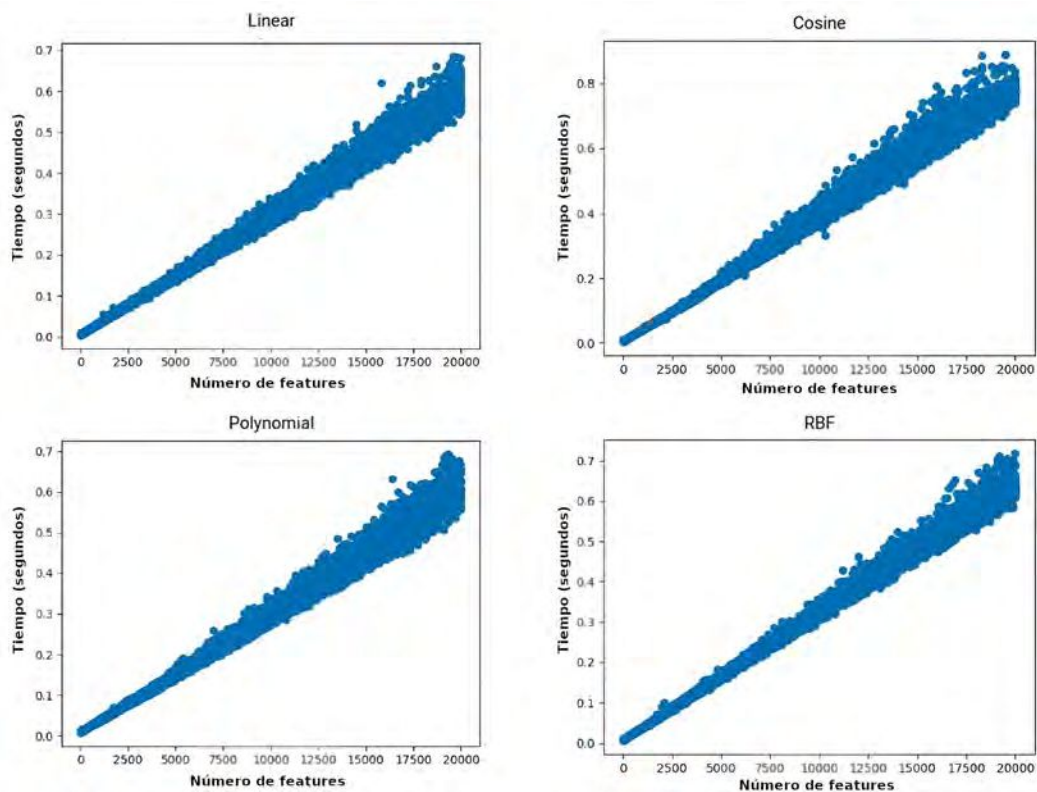


Figura 6.7 Tiempos medios de testing durante el proceso de CV para los kernels Linear, Cosine, Polynomial y RBF.

es responsable de cada partición. Este programa simula la ejecución de un RDD en Spark de manera más simplista: se asigna una partición a cada tarea de un RDD y luego el cómputo se distribuye entre los diferentes workers simulados del clúster. Sin embargo, no se lleva a cabo la distribución real de los datos ni el proceso de CV en cada uno de los agentes de la metaheurística, sino que todas las pruebas se ejecutan secuencialmente en el mismo nodo de computadora. Cada tarea sabe cuánto tiempo llevará y se agrega una diferencia ε (un pequeño valor aleatorio entre -1 y 1) para simular la aleatoriedad de un entorno real. El propósito de esto es medir cuánto tiempo toma ejecutar las tareas asignadas a cada worker definido en el simulador, incluso cuando la ejecución es secuencial (SEQ).

6.3.2. Experimentos

Para evaluar el rendimiento de las estrategias de balance de carga aplicadas a metaheurísticas, en [23] se realizaron dos experimentos:

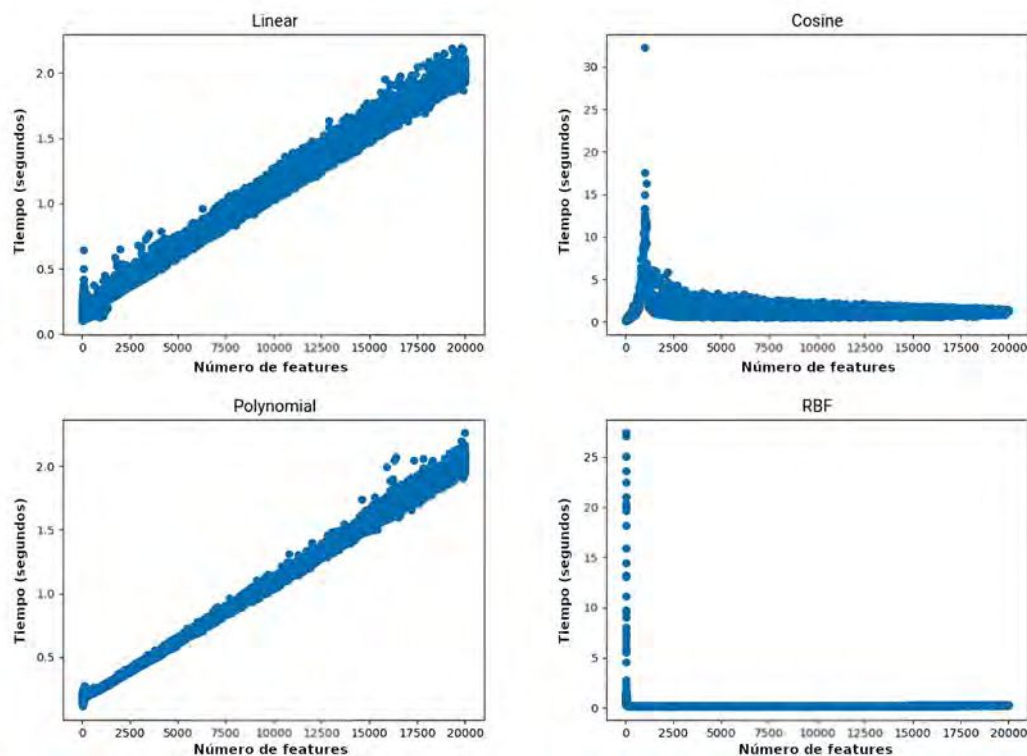


Figura 6.8 Tiempos por iteración para los núcleos Linear, Cosine, Polynomial y RBF.

1. **Experimento 1:** consiste en dos pruebas de concepto del simulador presentado en la Sección 6.3.1, donde se comparan las estrategias SEQ, ED, DBP y PELADO (Sección 6.3.7).
2. **Experimento 2:** consiste en una validación experimental en un clúster de Spark (Sección 6.3.8).

En el primer experimento, no se utilizan datos reales ni inferencia HGB, ya que se hizo foco en la validación de las estrategias de distribución en un simulador. Para someter a pruebas de estrés a las estrategias, la carga de cada tarea se simula generando un valor aleatorio con una gran variabilidad. Además, se simuló un gran número de workers para aumentar la probabilidad de una distribución ineficiente de tareas (más detalles sobre los parámetros utilizados se pueden encontrar más adelante en la Sección 6.3.6).

Para el segundo experimento, se abordó la optimización de un problema en un clúster de Spark real, involucrando el uso de información de expresión génica y datos clínicos relacionados con la supervivencia de pacientes con cáncer renal y de mama. Los datos clínicos incluyeron información sobre eventos de muerte y el tiempo en meses en que

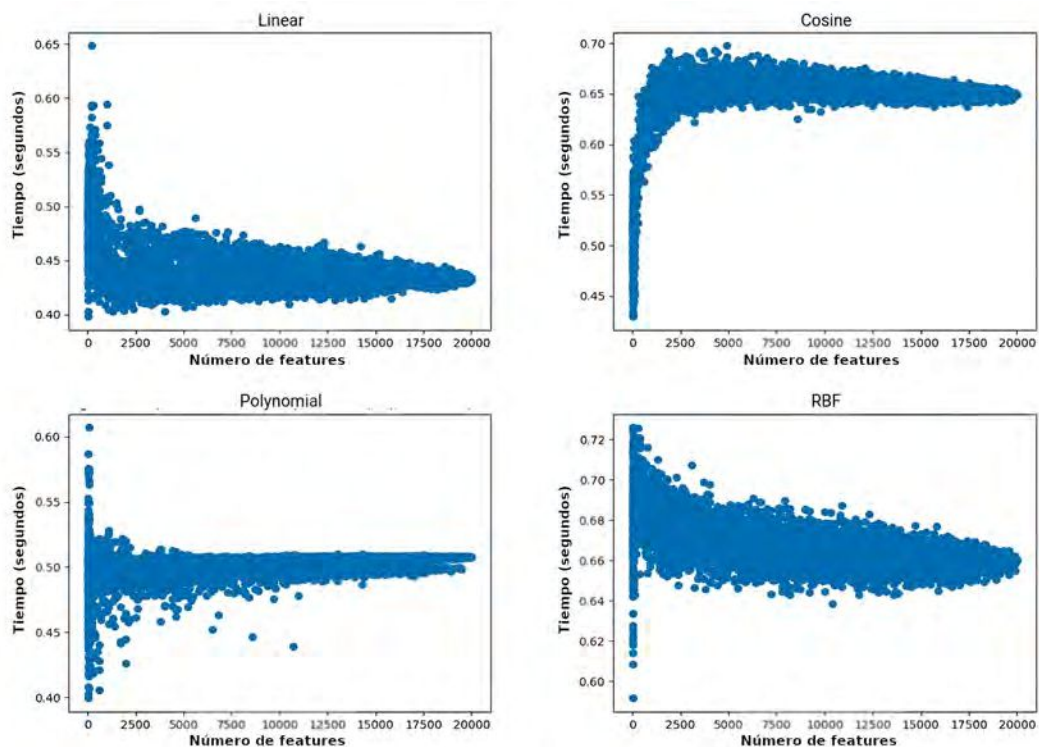


Figura 6.9 C-Index obtenido durante la fase de entrenamiento del SSVM para los kernels Linear, Cosine, Polinomial y RBF.

ocurrieron. El objetivo principal es identificar conjuntos de genes (biomarcadores) que exhiban poder pronóstico. Esto se puede obtener mediante dos enfoques:

- **Enfoque de agrupamiento:** se agrupan a los pacientes por expresión génica en dos grupos (enfoque introducido en la Sección 2.4.2) utilizando el algoritmo de agrupamiento K-Means. Luego, se emplea el método de Regresión de Cox para analizar la correlación de la expresión génica en grupos de pacientes para obtener la métrica de Log-Likelihood.
- **Enfoque de regresión de ocurrencia de eventos:** se utiliza la inferencia de un modelo SSVM para obtener el C-Index (enfoque introducido en la Sección 2.4.2), que indica la correlación entre cada uno de los pacientes y sus datos de expresión génica con la información clínica. Al abarcar datos categóricos y continuos, y tener un rango de valores interpretables entre 0.5 y 1.0, el C-Index permite comparar diferentes modelos y seleccionar el más adecuado para predecir eventos de interés a lo largo del tiempo.

Dado que el espacio de soluciones a evaluar es muy grande ($\approx 2^{2000}$ genes en el genoma humano), se utilizó la metaheurística BBH. Esta metaheurística evalúa, en paralelo, varios

subconjuntos de características en el espacio de soluciones. En cada uno de estos subconjuntos, se ejecuta uno de los enfoques mencionados anteriormente con el método de CV de muestreo estratificado [14] hasta que se completa un número predefinido de iteraciones. La estrategia es seleccionar el subconjunto de características que obtenga la mejor métrica de poder predictivo (ya sea log-likelihood en aquellos experimentos que utilizan el algoritmo de agrupamiento, o el C-Index en caso de que se esté realizando inferencia con SSVM).

6.3.3. Conjuntos de datos

Se utilizó el conjunto de datos *Breast Invasive Carcinoma (TCGA, PanCancer Atlas)* [12], que consiste en datos transcriptómicos de 19727 genes para 1082 pacientes con cáncer de mama; y el conjunto de datos *Kidney Renal Papillary Cell Carcinoma (TCGA, PanCancer Atlas)* [12], que proporciona datos transcriptómicos de 19291 genes para 282 pacientes con cáncer de riñón.

6.3.4. Metaheurísticas, modelos y métricas

La evaluación de los modelos K-Means y SSVM corresponde a la versión 1.0.2 de la librería Scikit-Learn y a la versión 0.17.2 de Scikit-Surv respectivamente. Los parámetros del modelo K-Means son los establecidos por defecto para esa versión, el número de clústeres se fija en 2 para representar a aquellos pacientes con la ocurrencia del evento y a los que aún están vivos. En cuanto al modelo SSVM, se realizaron experimentos con los kernels Linear, Polynomial y RBF. Independientemente del kernel utilizado, todos los experimentos SSVM utilizaron el optimizador AVLTREE y utilizando C-Index como la métrica a optimizar.

Para cada modelo evaluado (K-Means y SSVM con sus diferentes parámetros), se realizaron 10 ejecuciones ejecutando BBH con 25 iteraciones, 60 estrellas y una semilla aleatoria fija para garantizar la replicabilidad de los datos durante las 10 ejecuciones. Para evitar cualquier sesgo durante la evaluación de los resultados de sus agentes/estrellas, se realizó un proceso de CV de 10 pliegues con muestreo estratificado. Además, independientemente del modelo a entrenar y sus parámetros, el número de núcleos a utilizar se configuró en 1 para evitar cualquier sesgo de paralelismo ralentizado por tareas del sistema operativo que pudieran ocurrir durante los diferentes experimentos. Para eliminar el tiempo de transferencia de datos entre nodos, los datos se difundieron desde Spark. Esta difusión genera una copia de los datos utilizados por cada uno de los workers del clúster al comienzo del programa, reduciendo el sesgo del costo de dicha transferencia en los tiempos de ejecución resultantes de los experimentos.

6.3.5. Estrategias de balance de carga

Para comparar las estrategias propuestas en esta tesis, se realizan los mismos análisis con cuatro estrategias de distribución: SEQ, ED, DBP y, finalmente, PELADO.

En la estrategia SEQ, los agentes de BBH se ejecutan secuencialmente en el nodo maestro del clúster de Spark (para no sesgar la experimentación al no considerar los tiempos de inicialización y encapsulación del marco durante la ejecución).

6.3.6. Parámetros de PELADO y simulación

Prueba de estrés

Para realizar una prueba de estrés significativa, se evaluaron 30 iteraciones de la meta-heurística con 300 agentes de población con cada una de las tres estrategias de distribución en un conjunto de datos de 20000 características, realizándose el cómputo en 30 workers simulados. Para simular un entorno real donde diferentes nodos de computadora pueden tener diferentes capacidades computacionales, se agregó un retraso aleatorio entre el 10% y el 75% del tiempo de ejecución a 10 workers seleccionados aleatoriamente. Se estableció una semilla de aleatorización para que los tres experimentos fueran comparables. Luego, se realizó una prueba de Wilcoxon [140] para validar la significancia estadística entre la estrategia PELADO y la estrategia más cercana en términos de tiempo de ejecución y tiempo de inactividad. Este análisis es crítico para determinar si las diferencias observadas entre las dos estrategias son estadísticamente significativas o simplemente el resultado de una variación aleatoria.

Escenario con poca variabilidad

Para simular el entorno real, se ejecutaron 4 experimentos con 30 iteraciones y solo 3 workers. A su vez, contienen poca variabilidad en el tiempo de ejecución de sus tareas (el número de características se establece entre 1 y 200 en lugar de 20000) y, para verificar las ventajas de las estrategias planteadas, dos de los experimentos propuestos se llevan a cabo con 90 agentes de población con y sin retraso aleatorio en los workers del clúster, mientras que los otros dos experimentos restantes se ejecutan con 300 agentes de población (también alternando el retraso aleatorio en los workers del clúster).

Peor escenario para ED

Se llevó a cabo un experimento adicional en el simulador para demostrar las ventajas de las estrategias DBP y PELADO, que emplean modelos de predicción de tiempos de ejecución, en comparación con la estrategia ED. Este experimento se diseñó para recrear el peor escenario posible para la estrategia ED.

La estrategia ED distribuye las tareas a los workers de manera secuencial. Esto puede resultar en una situación donde algunos de estos workers reciban una carga de trabajo significativamente menor que la asignada a otros, lo que genera tiempos de inactividad considerables entre ellos durante la ejecución.

En contraste, las estrategias DBP y PELADO no sufren esta disminución en el rendimiento. Esto se debe a que implementan un sistema de redistribución de tareas, lo que les permite asignar el trabajo de manera más eficiente y equilibrada entre los workers disponibles.

Los parámetros para este experimento son iguales al experimento anterior, donde se evalúan las tres estrategias durante 30 iteraciones con 3 workers. Dos de las ejecuciones se realizan con 90 agentes con y sin retrasos aleatorios para un worker, y se repite la misma configuración con 300 agentes. Sin embargo, en este caso la asignación de la cantidad de características a evaluar en cada agente para cada worker está definida por el Algoritmo 6.1, donde i es el número de agente actual, $N_FEATURES$ es la cantidad máxima de features disponibles, $N_WORKERS$ es la cantidad total de workers en el clúster, y N_AGENTS es la cantidad de agentes a distribuir entre dichos workers.

En este escenario, si se contaran con 30 características totales, y 3 workers en el clúster de Spark, el primero recibiría tareas con una cantidad de características a evaluar entre 1 y 10, el segundo con tareas de entre 11 y 20 características, y el tercer worker con tareas de entre 21 y 30 características. Forzando una carga dispareja entre los workers, y la necesidad de realizar una distribución más inteligente para evitar la ociosidad de los mismos durante la ejecución.

Algoritmo 6.1 Algoritmo para forzar el peor escenario posible para la estrategia ED.

```

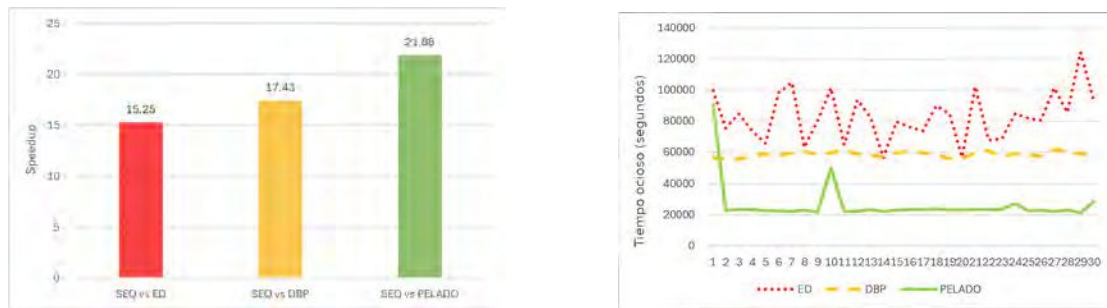
features_per_worker = N_FEATURES // N_WORKERS
agent_worker_idx = i * N_WORKERS // N_AGENTS
min_n_features = (agent_worker_idx * features_per_worker) + 1
max_n_features = min_n_features + features_per_worker - 1
features_to_evaluate = random(min_n_features, max_n_features)

```

6.3.7. Resultados Experimento 1: validación sobre el simulador

Prueba de estrés

Se analizaron las estrategias SEQ, ED, DBP y PELADO en el simulador. Los resultados obtenidos para el primer experimento indican que, para los períodos de inactividad durante las estrategias distribuidas, la estrategia PELADO exhibe un rendimiento superior (Figura 6.10b), aunque presenta una mayor varianza a lo largo de las iteraciones de la metaheurística. La Figura 6.10a muestra la aceleración obtenida por cada estrategia versus la estrategia SEQ.



(a) Speedup obtenido por las estrategias ED, DBP y PELADO vs. SEQ.

(b) Tiempos de ocio obtenidos en cada iteración, por las estrategias ED, DBP y PELADO.

Figura 6.10 Tiempos de ejecución y ocio para el primer sub experimento del Experimento 1.

Se puede ver que PELADO es la estrategia que más se acerca a la aceleración perfecta (30, que es el número de workers en la simulación).

En un entorno real como Spark, el tiempo de ejecución se calcula como el tiempo real que tarda entre la ejecución del método *collect()* de Spark. Dentro del simulador, no se posee dicho dato, ya que los tiempos se simulan sumando los tiempos de ejecución de las tareas asignadas a cada worker. Para obtener los datos en la Tabla 6.2, se sumaron todos los tiempos tomados por cada worker, y solo se dejó el máximo, ya que esa es la forma de simular que finalizó todo el proceso *collect()*. Por lo tanto, estos números siempre corresponden a la suma máxima de los tiempos de ejecución entre todos los workers para cada una de las iteraciones.

Observando las tablas tanto de los tiempos de ejecución (Tabla 6.2) como de los tiempos de inactividad (Tabla 6.3) por iteraciones, la estrategia PELADO presenta diferencias significativas frente a las otras estrategias evaluadas para todas las iteraciones, excepto para la primera. Esta excepción inicial se considera lógica, ya que en la primera iteración no hay información disponible sobre el trabajo pendiente de cada worker. En consecuencia, no es posible hacer un ajuste preciso de las capacidades para lograr una distribución igual y justa de la carga de trabajo entre los workers. Sin embargo, a partir de la segunda iteración, la estrategia PELADO demuestra consistentemente mejores tiempos de ejecución y de inactividad en comparación con las estrategias ED y DBP. Este hallazgo resalta la efectividad de la estrategia PELADO en integrar información de predicción, mejorando así la gestión y distribución de la carga de trabajo en cada iteración de la metaheurística. Respecto al pico observado en la iteración número 10, este es producto del retraso aleatorio de los workers durante la simulación. Aun así, el tiempo es menor que las otras dos estrategias.

Escenario con poca variabilidad

En cuanto a los 4 experimentos restantes, se puede observar que, cuando la variabilidad en el tiempo de ejecución de las tareas es insignificante e independiente del número de agentes

Tabla 6.2 Suma máxima de tiempos de ejecución (en segundos) de los workers para cada una de las tres estrategias de distribución. Todos los valores de la columna PELADO no muestran una diferencia significativa entre los tiempos comparados según la prueba de Wilcoxon.

Iteración	ED	DBP	PELADO
1	208860.982	166742.798	200821.920
2	180973.031	161558.942	126689.325
3	191676.677	162259.882	127483.264
4	184906.628	169791.113	132649.899
5	177349.099	170876.106	132000.625
6	209564.850	168253.178	130102.748
7	217880.426	172882.483	133018.452
8	177565.176	176788.458	136354.376
9	192854.703	170900.217	131602.794
10	215391.830	175338.356	163100.630
11	181598.595	178689.780	136933.039
12	210969.191	175362.467	135896.268
13	193719.520	169172.839	131854.237
14	165358.191	167838.125	129899.527
15	195830.710	175226.412	135947.934
16	192485.126	176953.791	136908.928
17	188083.629	174416.973	135262.494
18	204062.694	171241.215	133628.115
19	193412.966	164366.146	129286.420
20	165838.137	165792.138	129543.029
21	217991.871	173771.144	135069.606
22	185091.240	178965.334	138045.588
23	180105.956	168420.232	131924.848
24	200191.585	174189.641	139489
25	193484.9636	169128.0618	130857.077
26	190260.012	167538.460	130726.188
27	219492.275	180472.269	137978.421
28	201720.892	176168.462	136016.822
29	238127.471	172488.096	132010.958
30	206124.61	171671.768	138682

en la población, las estrategias DBP y PELADO no muestran diferencias significativas en un entorno de ejecución homogéneo (Figuras 6.11a y 6.11c). DBP puede obtener mejores resultados en términos de menor tiempo de inactividad al aplicar Bin Packing, que es más eficiente que Knapsack y realiza una distribución óptima (a diferencia de la distribución heurística MTHM realizada por Knapsack). Las diferencias aparecen cuando se introducen

Tabla 6.3 Tiempos medios de inactividad (en segundos) entre workers para cada una de las tres estrategias de distribución. El símbolo + en la columna PELADO indica que hay una diferencia significativa entre este tiempo y el valor más cercano de las otras dos estrategias (ED y DBP), el símbolo = indica que no hay diferencia significativa entre los tiempos comparados según la prueba de Wilcoxon.

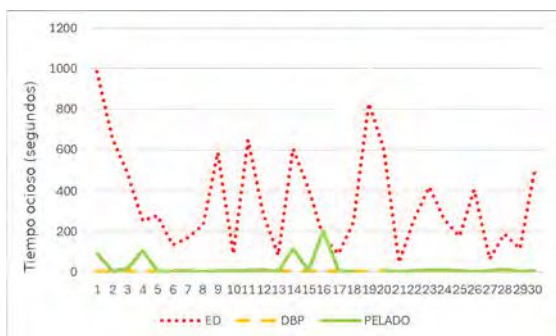
Iteración	ED	DBP	PELADO
1	99719,714	56851,696	90498.053 (=)
2	75212,325	55191,52	23122.345 (+)
3	84718,084	55341,101	23353.711 (+)
4	73789,804	58024,056	23687.263 (+)
5	65773,552	58715,559	22740.253 (+)
6	98331,147	57845,278	22469.891 (+)
7	104546,301	59067,619	22078.466 (+)
8	62900,892	60487,926	23045.289 (+)
9	80286,392	58417,537	21953.733 (+)
10	101162,899	59799,858	50115.502 (=)
11	64809,363	61308,879	22248.693 (+)
12	93773,678	58918,332	22401.269 (+)
13	82304,829	58013,361	23637.608 (+)
14	56055,525	57214,485	22234.058 (+)
15	79529,409	59552,869	22931.782 (+)
16	76326,081	60513,815	23433.64 (+)
17	73554,304	59543,994	23329.319 (+)
18	90043,027	58537,674	23933.292 (+)
19	84667,866	55754,824	23027.03 (+)
20	56681,927	56241,874	22908.847 (+)
21	102367,514	59439,366	23614.025 (+)
22	67854,272	61116,349	23333.777 (+)
23	68771,297	57213,15	23656.046 (+)
24	84842,439	58948,896	27348.519 (+)
25	81750,487	57803,766	22513.534 (+)
26	80393,539	57617,825	23222.976 (+)
27	101513,49	61697,528	22280.484 (+)
28	85595,442	60329,635	23133.623 (+)
29	124135,988	59007,545	21433.24 (+)
30	92690,207	58467,943	28501.036 (+)

retrasos aleatorios en la ejecución de las tareas, donde la estrategia PELADO se comporta de manera más eficiente que la estrategia DBP (Figuras 6.11b y 6.11d).

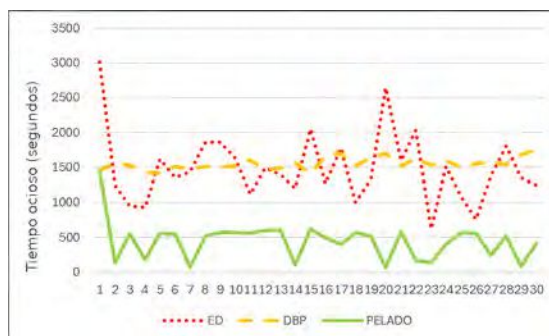
En todos los casos, cualquiera de las 3 estrategias distribuidas representa una mejora significativa en términos de aceleración sobre el procesamiento secuencial, como se puede

Tabla 6.4 Speedup obtenido para experimentos a lo largo de 30 iteraciones de prueba frente a la ejecución secuencial.

	90 (sin delay)	90 (con delay)	300 (sin delay)	300 (con delay)
ED	2.793	1.772	2.861	1.738
DBP	2.999	1.750	3.000	1.750
PELADO	2.986	2.132	2.955	2.255



(a) Tiempos de ocio obtenidos en cada iteración con 90 agentes y sin retrasos aleatorios en los workers.



(b) Tiempos de ocio obtenidos en cada iteración con 90 agentes y con retrasos aleatorios en los workers.



(c) Tiempos de ocio obtenidos en cada iteración con 300 agentes y sin retrasos aleatorios en los workers.



(d) Tiempos de ocio obtenidos en cada iteración con 300 agentes y con retrasos aleatorios en los workers.

Figura 6.11 Tiempos ociosos en cada iteración, por sub experimento en el Experimento 1.

ver en la Tabla 6.4. Esta tabla muestra que PELADO se desempeña significativamente mejor en entornos con retrasos de ejecución, y DBP en entornos homogéneos, aunque la diferencia es insignificante.

Peor escenario para ED

Como era de esperarse, en todos los escenarios ED rindió peor que DBP y PELADO (Figura 6.12), pero en aquellos entornos en donde los workers poseen tiempos de ejecución similares, las diferencias entre DBP y PELADO son despreciables, siendo DBP incluso más rápido ya que la distribución del algoritmos Bin Packing resulta más óptima que la realizada por el algoritmo Knapsack.

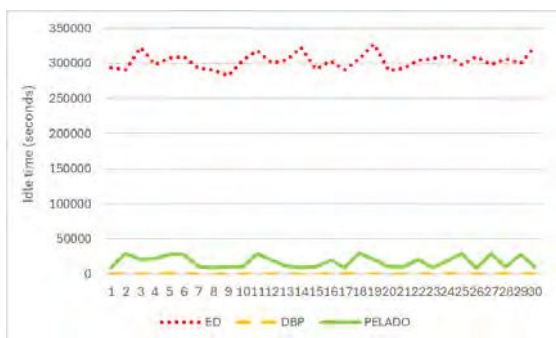
Frente a clústeres heterogéneos donde los workers podrían presentar retrasos en sus tiempos de ejecución, la estrategia PELADO ha demostrado rendir mejor, independientemente de la cantidad de agentes (figuras 6.12b y 6.12d). Cuando no se presentan retrasos, la diferencia entre los tiempos presentados por DBP y PELADO se hacen un poco más notables a medida que la cantidad de agentes aumenta, siendo DBP la que mejor distribuye la carga de cómputo (figuras 6.12a y 6.12c).

6.3.8. Resultados Experimento 2: validación sobre Apache Spark

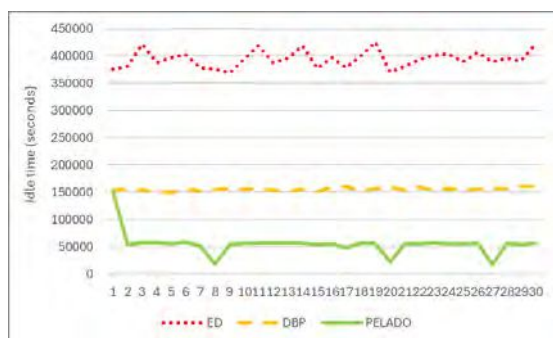
Respecto al segundo experimento, la Tabla 6.5 muestra, para cada uno de los modelos utilizados (Clustering y los tres modelos SSVM con diferentes kernels), las aceleraciones obtenidas para cada uno de los conjuntos de datos de entrenamiento y evaluación entre la estrategia SEQ y las estrategias distribuidas ED y DBP. En este experimento, no fue posible evaluar la estrategia PELADO debido a las limitaciones de Spark en la distribución de particiones, las cuales se explican en las Secciones 5.1 y 6.3.1.

La aceleración está en concordancia con el número de nodos workers disponibles en el clúster de computadoras (con tres nodos workers, la aceleración está cerca de 3). Aunque estos resultados son lógicos, cabe destacar que los cambios introducidos en los algoritmos originales son mínimos y se ajustan a cualquier modelo o al número de nodos que componen el clúster de Spark.

Un punto a tener en cuenta es que la diferencia entre la aceleración obtenida por la estrategia ED y el DBP es insignificante. Esto se debe a la baja variabilidad de los tiempos de ejecución de las tareas evaluadas durante las pruebas en el clúster. Con tiempos de ejecución similares, la distribución de las tareas, ya sea por número de tareas (estrategia ED) o por predicción del tiempo de ejecución (estrategia DBP), deja una carga de trabajo similar para todos los workers, además del sobrecoste de predicción requerido en la estrategia DBP. Además, debido a las limitaciones de hardware, las pruebas solo se ejecutaron con 3 nodos. Estos resultados son consistentes con lo observado durante las simulaciones en la Sección



(a) Tiempos de ocio obtenidos en cada iteración con 90 agentes y sin retrasos aleatorios en los workers en el peor escenario para ED.



(b) Tiempos de ocio obtenidos en cada iteración con 90 agentes y con retrasos aleatorios en los workers en el peor escenario para ED.



(c) Tiempos de ocio obtenidos en cada iteración con 300 agentes y sin retrasos aleatorios en los workers en el peor escenario para ED.



(d) Tiempos de ocio obtenidos en cada iteración con 300 agentes y con retrasos aleatorios en los workers en el peor escenario para ED.

Figura 6.12 Tiempos ociosos en cada iteración, por sub experimento en el Experimento 1.

6.3.7, donde se obtiene una mejora significativa solo en entornos de ejecución con retrasos aleatorios en las tareas.

En un entorno con tareas más heterogéneas y un mayor número de workers de Spark, la probabilidad de realizar una distribución desigual durante el proceso ED debería ser mayor, y por lo tanto, habría más diferencia en la aceleración final respecto a DBP.

El código y los datos para replicar ambos experimentos están disponibles en <https://github.com/midusi/load-balancer-metaheuristics>.

Tabla 6.5 Speedup obtenido en cada conjunto de datos para cada uno de los modelos utilizados a lo largo de las 30 iteraciones de prueba frente a la ejecución secuencial.

		K-Means	SSVM Linear	SSVM Poly	SSVM RBF
Breast	ED	2.88	2.83	2.81	2.85
	DBP	2.89	2.83	2.81	2.87
Kidney	ED	2.61	2.87	2.86	2.83
	DBP	2.61	2.87	2.89	2.82

Capítulo 7

Conclusión y trabajo a futuro

En este capítulo se expone un resumen de los logros obtenidos durante el desarrollo de la tesis. Por último, se detallan las líneas de investigación futuras que proponen extender los temas abarcados.

7.1. Conclusiones generales

La presente tesis doctoral ha centrado su aporte en el desarrollo de un framework que aplica estrategias de balance de carga inteligentes para la optimización de proceso de selección de características. Las estrategias presentadas han sido aplicadas particularmente al proceso de descubrimiento de biomarcadores con poder pronóstico/predictivo en cáncer. A continuación, se detallan los principales aportes y hallazgos obtenidos.

En primer lugar, se ha desarrollado e implementado la plataforma Multiomix, una herramienta de código abierto y gratuita que pone a disposición de la comunidad científica una amplia gama de funcionalidades para el análisis de supervivencia y la identificación de biomarcadores. Multiomix abstrae al usuario de la gestión de sus datos y facilita la incorporación de bases de datos externas, como los datos curados de cBioPortal. Además, esta plataforma conllevó el desarrollo de múltiples plataformas apéndice que abstraen la incorporación de datos actualizados y curados de genes y pathways (BioAPI), y miRNAs y sitios de metilación de ADN (Modulector).

Uno de los puntos fuertes de Multiomix radica en sus funcionalidades de selección de características para el descubrimiento de biomarcadores. En este sentido, se utilizan metaheurísticas que evalúan diferentes subconjuntos de moléculas que conforman el biomarcador a optimizar. Sin embargo, estos procesos son costosos computacionalmente, por lo que se implementaron diferentes estrategias de balance de carga para reducir los tiempos de

inactividad de los workers de un clúster de Apache Spark donde se ejecutan los procesos de FS de Multiomix.

Se propusieron tres estrategias de balance de carga: ED, DBP y PELADO. La estrategia ED distribuye equitativamente las tareas, mientras que DBP hace uso del modelo de inteligencia artificial HGB para predecir los tiempos de ejecución y realizar una distribución más ponderada utilizando el algoritmo Bin Packing. Por su parte, PELADO también utiliza HGB, pero emplea el algoritmo Knapsack para realizar una distribución que considere los retardos que presentan los workers a lo largo de las iteraciones de la metaheurística.

Los resultados obtenidos demostraron que las tres estrategias propuestas ofrecen mejoras significativas en los tiempos de ejecución en comparación con la ejecución secuencial. Además, una de las ventajas de estas estrategias es que los cambios introducidos son mínimos y pueden aplicarse a cualquier metaheurística que se quiera ejecutar de manera distribuida.

Cabe destacar que la estrategia PELADO ofrece una ventaja con respecto a las otras dos estrategias en clústeres heterogéneos (es decir, cuando los workers que lo conforman tienen diferentes prestaciones). Sin embargo, debido a limitaciones técnicas, no se puede implementar actualmente en Spark. Por este motivo, se desarrolló un simulador que pone a prueba las estrategias propuestas, demostrando la ventaja de PELADO en todos los experimentos llevados a cabo.

Los resultados obtenidos en dicho simulador para la evaluación de las estrategias SEQ, ED, DBP y PELADO, revelan que esta última presenta una mejora significativa en los tiempos de inactividad de los workers en un clúster de computadoras simulado. Aunque la estrategia PELADO mostró resultados ligeramente menos eficientes en la primera iteración, este fenómeno es lógico dada la ausencia de información para ajustar las capacidades de los workers en esa fase inicial. La superioridad consistente de PELADO en iteraciones posteriores resalta su capacidad para gestionar mejor la carga de trabajo, optimizando los tiempos de ejecución y, más importante aún, los períodos de inactividad de los workers. Con todas las implicaciones que esto conlleva para la eficiencia y la utilización efectiva de los recursos del clúster.

Otros dos experimentos en el simulador han demostrado que las estrategias de distribución ED, DBP y PELADO ofrecen una ventaja significativa en términos de aceleración en comparación con la estrategia SEQ. Sin embargo, no hay una diferencia sustancial entre estas estrategias cuando las tareas a ejecutar tienen una variabilidad mínima en el tiempo de ejecución. La ventaja de PELADO se hace evidente en entornos heterogéneos donde las tareas pueden experimentar retrasos en el tiempo de ejecución debido a diferencias en las capacidades de hardware en los nodos dentro de un clúster de computadoras. En tales

escenarios, PELADO logra una aceleración superior en relación con ED y DBP, minimizando el tiempo de inactividad entre los nodos.

Los experimentos en un clúster real han demostrado que las estrategias son efectivas para optimizar los tiempos de ejecución de la metaheurística BBH para el proceso de FS, en comparación con la ejecución secuencial. La estrategia ED, constituye una mejora sustancial en los tiempos de ejecución, distinguida por su simplicidad y su capacidad para generar una aceleración lineal con un costo prácticamente nulo de modificación del algoritmo. Aunque la adición de las predicciones del modelo HGB para la estrategia DBP no mostró mejoras distinguibles en la aceleración general durante la experimentación empírica en el clúster de Spark, es crucial resaltar que en un entorno de Big Data, es de esperar que la escala y variabilidad de los conjuntos de datos presenten mejoras más relevantes y justifiquen la implementación de las estrategias DBP y PELADO.

Es relevante enfatizar que las estrategias propuestas en este trabajo no se limitan a un contexto específico, sino que se presentan como un enfoque generalizable. Todas ellas pueden aplicarse con éxito a cualquier metaheurística que se quiera ejecutar de manera distribuida. Esta generalización consolida su posición como una valiosa contribución para optimizar los tiempos de ejecución y la inactividad en tareas computacionalmente intensivas. Además, al implementarse en el entorno Spark, se libera al usuario de las preocupaciones asociadas con la gestión de recursos y las complejidades inherentes a la coordinación de un clúster de computadoras y la distribución de cálculos.

Finalmente, las técnicas desarrolladas se implementaron en un entorno real utilizando la infraestructura de AWS, permitiendo su uso en la plataforma Multiomix. Además, todos los algoritmos propuestos en esta tesis y las plataformas que los emplean tienen implementaciones de código abierto.

7.2. Líneas de trabajo futuras

De la presente tesis se pueden desprender varios puntos de trabajo e investigación.

En primer lugar, hay numerosas funcionalidades técnicas a incorporar en la plataforma Multiomix:

- Nuevos métodos para creación de biomarcadores: podrían proponerse nuevos biomarcadores a partir de la expresión diferencial entre pacientes y controles sanos, o a partir de bases de datos de biomarcadores externas. También podrían proponerse nuevos biomarcadores a través de algoritmos de machine learning que hagan uso de resultados de experimentos ya ejecutados en la plataforma (encontrando características relevantes entre ellos).

- Incorporación de nuevos modelos de ML para la evaluación del poder pronóstico de los biomarcadores.
- Incorporación de nuevas fuentes de datos externas para su fácil uso desde Multiomix. Como podría ser la base de datos curada de UCSC Xena de la Universidad de California, Santa Cruz.
- Implementaciones para acortar la brecha tecnológica: el avance de la ciencia y la tecnología requiere que constantemente se estén actualizando el stack de herramientas de desarrollo e implementar nuevas características de la plataforma. Tanto de Multiomix como de los proyectos que lo rodean. Numerosas ideas han surgido a los largo del desarrollo, como nuevas visualizaciones de los datos, la posibilidad de compartir resultados de experimentos entre usuarios de una organización, recomendaciones personalizadas sobre los datos propios del usuario, opciones de accesibilidad, entre muchos otros.

Con respecto a las estrategias de balance de carga, un paso fundamental en esa dirección es la incorporación de nuevos mecanismos en el framework Spark que permita definir manualmente qué worker del clúster debe tomar cada partición. Dichos mecanismos permitirían implementar en un entorno real la estrategia PELADO.

Una vez asegurada la implementación, resultaría interesante evaluar la factibilidad y ventajas de todas las estrategias propuestas durante procesos de selección de características más complejos. Podrían ejecutarse experimentos con una cantidad de workers mucho más numerosa y mayor costo de cómputo (es decir, una mayor cantidad de agentes o iteraciones en la metaheurística, datasets con mayor cantidad de características, etc) que la evaluada durante este trabajo.

A su vez, las estrategias estudiadas podrían ser aplicables a nuevas funciones de Multiomix que sean computacionalmente costosas, pero que puedan ser optimizadas a través de la distribución de su ejecución.

Bibliografía

- [1] Agrawal, A., Balci, H., Hanspers, K., Coort, S. L., Martens, M., Slenter, D. N., Ehrhart, F., Digles, D., Waagmeester, A., Wassink, I., et al. (2024). Wikipathways 2024: next generation pathway database. *Nucleic acids research*, 52(D1):D679–D689.
- [2] Aleksander, S. A., Balhoff, J., Carbon, S., Cherry, J. M., Drabkin, H. J., Ebert, D., Feuermann, M., Gaudet, P., Harris, N. L., et al. (2023). The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031.
- [3] 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., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.
- [4] Ay, Ş., Ekinçi, E., and Garip, Z. (2023). A comparative analysis of meta-heuristic optimization algorithms for feature selection on ml-based classification of heart-related diseases. *The Journal of Supercomputing*, pages 1–30.
- [5] Belinky, F., Nativ, N., Stelzer, G., Zimmerman, S., Iny Stein, T., Safran, M., and Lancet, D. (2015). Pathcards: multi-source consolidation of human biological pathways. *Database*, 2015:bav006.
- [6] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.
- [7] Benjamini, Y., Krieger, A. M., and Yekutieli, D. (2006). Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, 93(3):491–507.
- [8] Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188.
- [9] Bode, A. M. and Dong, Z. (2018). Recent advances in precision oncology research. *NPJ precision oncology*, 2(1):11.
- [10] Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62.
- [11] Bonnal, S. C., López-Oreja, I., and Valcárcel, J. (2020). Roles and mechanisms of alternative splicing in cancer—implications for care. *Nature reviews Clinical oncology*, 17(8):457–474.

- [12] Bonneville, R., Krook, M. A., Kautto, E. A., Miya, J., Wing, M. R., Chen, H.-Z., Reeser, J. W., Yu, L., and Roychowdhury, S. (2017). Landscape of microsatellite instability across 39 cancer types. *JCO precision oncology*, 1:1–15.
- [13] Boons, G., Vandamme, T., Mariën, L., Lybaert, W., Roeyen, G., Rondou, T., Papadimitriou, K., Janssens, K., Op de Beeck, B., Simoens, M., et al. (2022). Longitudinal copy-number alteration analysis in plasma cell-free dna of neuroendocrine neoplasms is a novel specific biomarker for diagnosis, prognosis, and follow-up. *Clinical Cancer Research*, 28(2):338–349.
- [14] Botev, Z. and Ridder, A. (2017). Variance reduction. *Wiley statsRef: Statistics reference online*, pages 1–6.
- [15] Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384.
- [16] Breusch, T. S. and Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica: Journal of the econometric society*, pages 1287–1294.
- [17] Butti, M., Abba, M. C., Haedo, A. S., and Lacunza, E. (2011). Bioplat: A platform to discover and evaluate human cancer biomarkers. In *2nd Argentinian Conference on Bioinformatics and Computational Biology*. A2B2C.
- [18] Cahon, S., Melab, N., and Talbi, E.-G. (2004). Paradise: A framework for the reusable design of parallel and distributed metaheuristics. *Journal of heuristics*, 10(3):357–380.
- [19] Camele, G. and Hasperué, W. (2023). Performance analysis of the survival-svm classifier applied to gene-expression databases. In *XXVIII Congreso Argentino de Ciencias de la Computación (CACIC)*.
- [20] Camele, G., Hasperué, W., Ronchetti, F., and Quiroga, F. M. (2021). Comparative study of the performance of the classification algorithms of the apache spark ml library. In *XXVII Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 4 al 8 de octubre de 2021)*.
- [21] Camele, G., Hasperué, W., Ronchetti, F., and Quiroga, F. M. (2022a). Statistical analysis of the performance of four apache spark ml algorithms. *Journal of Computer Science & Technology*, 22.
- [22] Camele, G., Menazzi, S., Chanfreau, H., Marraco, A., Hasperué, W., Butti, M. D., and Abba, M. C. (2022b). Multiomix: a cloud-based platform to infer cancer genomic and epigenomic events associated with gene expression modulation. *Bioinformatics*, 38(3):866–868.
- [23] Camele, G., Quiroga, F. M., Muhlberger, J. M., Stanchi, O. A., Ponte, S. A., and Hasperué, W. (2024). Pelado: A load balancing algorithm for metaheuristics optimization applied to biomarker discovery.
- [24] Campbell, B. B., Light, N., Fabrizio, D., Zatzman, M., Fuligni, F., de Borja, R., Davidson, S., Edwards, M., Elvin, J. A., Hodel, K. P., et al. (2017). Comprehensive analysis of hypermutation in human cancer. *Cell*, 171(5):1042–1056.

- [25] Cerami, E., Gao, J., Dogrusoz, U., Gross, B. E., Sumer, S. O., Aksoy, B. A., Jacobsen, A., Byrne, C. J., Heuer, M. L., Larsson, E., et al. (2012). The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer discovery*, 2(5):401–404.
- [26] Chakravarty, D., Gao, J., Phillips, S., Kundra, R., Zhang, H., Wang, J., Rudolph, J. E., Yaeger, R., Soumerai, T., Nissan, M. H., et al. (2017). Oncokb: a precision oncology knowledge base. *JCO precision oncology*, 1:1–16.
- [27] Chapman, P. B., Hauschild, A., Robert, C., Haanen, J. B., Ascierto, P., Larkin, J., Dummer, R., Garbe, C., Testori, A., Maio, M., et al. (2011). Improved survival with vemurafenib in melanoma with braf v600e mutation. *New England Journal of Medicine*, 364(26):2507–2516.
- [28] Chappell, K., Francou, B., Habib, C., Huby, T., Leoni, M., Cottin, A., Nadal, F., Adnet, E., Paoli, E., Oliveira, C., et al. (2022). Galaxy is a suitable bioinformatics platform for the molecular diagnosis of human genetic disorders using high-throughput sequencing data analysis: Five years of experience in a clinical laboratory. *Clinical Chemistry*, 68(2):313–321.
- [29] Chou, C.-H., Chang, N.-W., Shrestha, S., Hsu, S.-D., Lin, Y.-L., Lee, W.-H., Yang, C.-D., Hong, H.-C., Wei, T.-Y., Tu, S.-J., et al. (2016). mirtarbase 2016: updates to the experimentally validated mirna-target interactions database. *Nucleic acids research*, 44(D1):D239–D247.
- [30] Cox, D. R. and Oakes, D. (1984). *Analysis of survival data*, volume 21. CRC press.
- [31] Csabai, L., Fazekas, D., Kadlecsek, T., Szalay-Bekő, M., Bohár, B., Madgwick, M., Módos, D., Ölbei, M., Gul, L., Sudhakar, P., et al. (2022). Signalink3: a multi-layered resource to uncover tissue-specific signaling networks. *Nucleic acids research*, 50(D1):D701–D709.
- [32] Cui, C., Zhong, B., Fan, R., and Cui, Q. (2024). Hmdd v4. 0: a database for experimentally supported human microrna-disease associations. *Nucleic Acids Research*, 52(D1):D1327–D1332.
- [33] de Bruijn, I., Kundra, R., Mastrogiacomo, B., Tran, T. N., Sikina, L., Mazor, T., Li, X., Ochoa, A., Zhao, G., Lai, B., et al. (2023). Analysis and visualization of longitudinal genomic and clinical data from the aacr project genie biopharma collaborative in cbiportal. *Cancer research*, 83(23):3861–3867.
- [34] De Caceres, I. I., Battagli, C., Esteller, M., Herman, J. G., Dulaimi, E., Edelson, M. I., Bergman, C., Ehya, H., Eisenberg, B. L., and Cairns, P. (2004). Tumor cell-specific brca1 and rassfla hypermethylation in serum, plasma, and peritoneal fluid from ovarian cancer patients. *Cancer research*, 64(18):6476–6481.
- [35] Dramiński, M., Rada-Iglesias, A., Enroth, S., Wadelius, C., Koronacki, J., and Komorowski, J. (2008). Monte carlo feature selection for supervised classification. *Bioinformatics*, 24(1):110–117.
- [36] Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, pages 39–43. Ieee.

- [37] Eberhart, R. C., Shi, Y., and Kennedy, J. (2001). *Swarm intelligence*. Elsevier.
- [38] Englebert, C., Quinn, T., and Bichindaritz, I. (2017). Feature selection for survival analysis in bioinformatics. In *Proceedings of the Workshop on Advances in Bioinformatics and Artificial Intelligence: Bridging the Gap Co-Located with 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia*, pages 19–25.
- [39] Flaherty, K. T., Puzanov, I., Kim, K. B., Ribas, A., McArthur, G. A., Sosman, J. A., O’Dwyer, P. J., Lee, R. J., Grippo, J. F., Nolop, K., et al. (2010). Inhibition of mutated, activated braf in metastatic melanoma. *New England Journal of Medicine*, 363(9):809–819.
- [40] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- [41] Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378.
- [42] Gamazon, E. R. and Stranger, B. E. (2015). The impact of human copy number variation on gene expression. *Briefings in functional genomics*, 14(5):352–357.
- [43] Gao, J., Aksoy, B. A., Dogrusoz, U., Dresdner, G., Gross, B., Sumer, S. O., Sun, Y., Jacobsen, A., Sinha, R., Larsson, E., et al. (2013). Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Science signaling*, 6(269):p11–p11.
- [44] Gauss, C. F. (1816). Bestimmung der genauigkeit der beobachtungen. *Ibidem*, pages 129–138.
- [45] Gholizadeh, S., Razavi, N., and Shojaei, E. (2018). Improved black hole and multiverse algorithms for discrete sizing optimization of planar structures. *Engineering Optimization*.
- [46] Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert, I., Taylor, J., et al. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome research*, 15(10):1451–1455.
- [47] Gilani, N., Arabi Belaghi, R., Aftabi, Y., Faramarzi, E., Edguenlue, T., and Somi, M. H. (2022). Identifying potential mirna biomarkers for gastric cancer diagnosis using machine learning variable selection approach. *Frontiers in genetics*, 12:779455.
- [48] Goldfeld, S. M. and Quandt, R. E. (1965). Some tests for homoscedasticity. *Journal of the American statistical Association*, 60(310):539–547.
- [49] Griffith, M., Spies, N. C., Krysiak, K., McMichael, J. F., Coffman, A. C., Danos, A. M., Ainscough, B. J., Ramirez, C. A., Rieke, D. T., Kujan, L., et al. (2017). Civic is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer. *Nature genetics*, 49(2):170–174.
- [50] Griffiths-Jones, S. (2004). The microrna registry. *Nucleic acids research*, 32(suppl_1):D109–D111.
- [51] Griffiths-Jones, S., Grocock, R. J., Van Dongen, S., Bateman, A., and Enright, A. J. (2006). mirbase: microrna sequences, targets and gene nomenclature. *Nucleic acids research*, 34(suppl_1):D140–D144.

- [52] Griffiths-Jones, S., Saini, H., van Dongen, S., and Enright, A. (2008). mirbase: Tools for microrna genomics. *ucleic acids research*, 36. *Database Issue*): D154-D68.
- [53] Griss, J., Viteri, G., Sidiropoulos, K., Nguyen, V., Fabregat, A., and Hermjakob, H. (2020). Reactomegsa-efficient multi-omics comparative pathway analysis. *Molecular & Cellular Proteomics*, 19(12):2115–2125.
- [54] Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., and Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546.
- [55] Harrell Jr, F. E., Lee, K. L., and Mark, D. B. (1996). Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387.
- [56] Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222:175–184.
- [57] Hoadley, K. A., Yau, C., Wolf, D. M., Cherniack, A. D., Tamborero, D., Ng, S., Leiserson, M. D., Niu, B., McLellan, M. D., Uzunangelov, V., et al. (2014). Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158(4):929–944.
- [58] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.
- [59] Huang, C.-L. and Dun, J.-F. (2008). A distributed pso–svm hybrid system with feature selection and parameter optimization. *Applied soft computing*, 8(4):1381–1391.
- [60] Huang, H.-Y., Lin, Y.-C.-D., Li, J., Huang, K.-Y., Shrestha, S., Hong, H.-C., Tang, Y., Chen, Y.-G., Jin, C.-N., Yu, Y., et al. (2020). mirtarbase 2020: updates to the experimentally validated microrna–target interaction database. *Nucleic acids research*, 48(D1):D148–D154.
- [61] Huang, Z., Shi, J., Gao, Y., Cui, C., Zhang, S., Li, J., Zhou, Y., and Cui, Q. (2019). Hmdd v3. 0: a database for experimentally supported human microrna–disease associations. *Nucleic acids research*, 47(D1):D1013–D1017.
- [62] Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, 2(3):841 – 860.
- [63] Javaid, M. U., Kanoun, A. A., Demesmaeker, F., Ghrab, A., and Skhiri, S. (2020). A performance prediction model for spark applications. In *International Conference on Big Data*, pages 13–22. Springer.
- [64] Jewison, T., Su, Y., Disfany, F. M., Liang, Y., Knox, C., Maciejewski, A., Poelzer, J., Huynh, J., Zhou, Y., Arndt, D., et al. (2014). Smpdb 2.0: big improvements to the small molecule pathway database. *Nucleic acids research*, 42(D1):D478–D484.
- [65] Jin, Z., Sato, Y., Kawashima, M., and Kanehisa, M. (2023). Kegg tools for classification and analysis of viral proteins. *Protein Science*, 32(12):e4820.

- [66] Kamburov, A. and Herwig, R. (2022). Consensuspathdb 2022: molecular interactions update as a resource for network biology. *Nucleic acids research*, 50(D1):D587–D595.
- [67] Kamburov, A., Wierling, C., Lehrach, H., and Herwig, R. (2009). Consensuspathdb—a database for integrating human functional interaction networks. *Nucleic acids research*, 37(suppl_1):D623–D628.
- [68] Kandasamy, K., Mohan, S. S., Raju, R., Keerthikumar, S., Kumar, G. S. S., Venugopal, A. K., Telikicherla, D., Navarro, J. D., Mathivanan, S., Pecquet, C., et al. (2010). Netpath: a public resource of curated signal transduction pathways. *Genome biology*, 11:1–9.
- [69] Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481.
- [70] Karapetis, C. S., Khambata-Ford, S., Jonker, D. J., O’Callaghan, C. J., Tu, D., Tebbutt, N. C., Simes, R. J., Chalchal, H., Shapiro, J. D., Robitaille, S., et al. (2008). K-ras mutations and benefit from cetuximab in advanced colorectal cancer. *New England Journal of Medicine*, 359(17):1757–1765.
- [71] Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- [72] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN’95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE.
- [73] Khanesar, M. A., Teshnehlab, M., and Shoorehdeli, M. A. (2007). A novel binary particle swarm optimization. In *2007 Mediterranean conference on control & automation*, pages 1–6. IEEE.
- [74] Knox, C., Wilson, M., Klinger, C. M., Franklin, M., Oler, E., Wilson, A., Pon, A., Cox, J., Chin, N. E., Strawbridge, S. A., et al. (2024). Drugbank 6.0: the drugbank knowledgebase for 2024. *Nucleic Acids Research*, 52(D1):D1265–D1275.
- [75] Konstantinopoulos, P. A., Spentzos, D., and Cannistra, S. A. (2008). Gene-expression profiling in epithelial ovarian cancer. *Nature clinical practice Oncology*, 5(10):577–587.
- [76] Krysiak, K., Danos, A. M., Kiwala, S., McMichael, J. F., Coffman, A. C., Barnell, E. K., Sheta, L., Saliba, J., Gridale, C. J., Kujan, L., et al. (2021). Evolution of the open-access civic knowledgebase is driven by the needs of the cancer variant interpretation community. *bioRxiv*, pages 2021–06.
- [77] Kundra, R., Zhang, H., Sheridan, R., Sirintrapun, S. J., Wang, A., Ochoa, A., Wilson, M., Gross, B., Sun, Y., Madupuri, R., et al. (2021). Oncotree: a cancer classification system for precision oncology. *JCO clinical cancer informatics*, 5:221–230.
- [78] Lakshman, A. and Malik, P. (2010). Cassandra: a decentralized structured storage system. *ACM SIGOPS operating systems review*, 44(2):35–40.
- [79] Langfelder, P. and Horvath, S. (2008). Wgcna: an R package for weighted correlation network analysis. *BMC bioinformatics*, 9:1–13.
- [80] Li, Y., Qiu, C., Tu, J., Geng, B., Yang, J., Jiang, T., and Cui, Q. (2014). Hmdd v2. 0: a database for experimentally supported human microRNA and disease associations. *Nucleic acids research*, 42(D1):D1070–D1074.

- [81] Liu, X., Wang, S., Meng, F., Wang, J., Zhang, Y., Dai, E., Yu, X., Li, X., and Jiang, W. (2013). Sm2mir: a database of the experimentally validated small molecules' effects on microRNA expression. *Bioinformatics*, 29(3):409–411.
- [82] Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., Hasz, R., Walters, G., Garcia, F., Young, N., et al. (2013). The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580–585.
- [83] Lu, M., Zhang, Q., Deng, M., Miao, J., Guo, Y., Gao, W., and Cui, Q. (2008). An analysis of human microRNA and disease associations. *PloS one*, 3(10):e3420.
- [84] Lu, Y., Chan, Y.-T., Tan, H.-Y., Li, S., Wang, N., and Feng, Y. (2020). Epigenetic regulation in human cancer: the potential role of epi-drug in cancer therapy. *Molecular cancer*, 19:1–16.
- [85] Lynch, T. J., Bell, D. W., Sordella, R., Gurubhagavatula, S., Okimoto, R. A., Brannigan, B. W., Harris, P. L., Haserlat, S. M., Supko, J. G., Haluska, F. G., et al. (2004). Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib. *New England Journal of Medicine*, 350(21):2129–2139.
- [86] Ma, B., Geng, Y., Meng, F., Yan, G., and Song, F. (2020). Identification of a sixteen-gene prognostic biomarker for lung adenocarcinoma using a machine learning method. *Journal of Cancer*, 11(5):1288.
- [87] Mantel, N. et al. (1966). Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemother Rep*, 50(3):163–170.
- [88] Marraco, A. D., Camele, G., Hasperu e, W., Menazzi, S., Abba, M., and Butti, M. (2021). Modulector: una plataforma como servicio para el acceso a bases de datos de micro arns. *Innovaci n y Desarrollo Tecnol gico y Social*, 3(1):89–114.
- [89] Martello, S. and Toth, P. (1981). Heuristic algorithms for the multiple knapsack problem. *Computing*, 27(2):93–112.
- [90] Martello, S. and Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.
- [91] Martin, F. J., Amode, M. R., Aneja, A., Austine-Orimoloye, O., Azov, A. G., Barnes, I., Becker, A., Bennett, R., Berry, A., Bhai, J., et al. (2023). Ensembl 2023. *Nucleic acids research*, 51(D1):D933–D941.
- [92] Milacic, M., Beavers, D., Conley, P., Gong, C., Gillespie, M., Griss, J., Haw, R., Jassal, B., Matthews, L., May, B., et al. (2024). The reactome pathway knowledgebase 2024. *Nucleic acids research*, 52(D1):D672–D678.
- [93] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- [94] MotieGhader, H., Masoudi-Sobhanzadeh, Y., Ashtiani, S. H., and Masoudi-Nejad, A. (2020). mRNA and microRNA selection for breast cancer molecular subtype stratification using meta-heuristic based algorithms. *Genomics*, 112(5):3207–3217.

- [95] Mustafa, S., Elghandour, I., and Ismail, M. A. (2018). A machine learning approach for predicting execution time of spark jobs. *Alexandria engineering journal*, 57(4):3767–3778.
- [96] Nishimura, D. (2001). Biocarta. *Biotech Software & Internet Report: The Computer Software Journal for Scient*, 2(3):117–120.
- [97] O’Leary, N. A., Wright, M. W., Brister, J. R., Ciufu, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., et al. (2016). Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. *Nucleic acids research*, 44(D1):D733–D745.
- [98] Paez, J. G., Janne, P. A., Lee, J. C., Tracy, S., Greulich, H., Gabriel, S., Herman, P., Kaye, F. J., Lindeman, N., Boggon, T. J., et al. (2004). Egfr mutations in lung cancer: correlation with clinical response to gefitinib therapy. *Science*, 304(5676):1497–1500.
- [99] Pan, X., Hu, X., Zhang, Y.-H., Chen, L., Zhu, L., Wan, S., Huang, T., and Cai, Y.-D. (2019). Identification of the copy number variant biomarkers for breast cancer subtypes. *Molecular Genetics and Genomics*, 294:95–110.
- [100] Parmar, C., Grossmann, P., Rietveld, D., Rietbergen, M. M., Lambin, P., and Aerts, H. J. (2015). Radiomic machine-learning classifiers for prognostic biomarkers of head and neck cancer. *Frontiers in oncology*, 5:272.
- [101] Pashaei, E. and Aydin, N. (2017). Binary black hole algorithm for feature selection and classification on biological data. *Applied Soft Computing*, 56:94–106.
- [102] Pashaei, E. and Pashaei, E. (2021). Gene selection using hybrid dragonfly black hole algorithm: A case study on rna-seq covid-19 data. *Analytical biochemistry*, 627:114242.
- [103] Pashaei, E., Pashaei, E., and Aydin, N. (2019). Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization. *Genomics*, 111(4):669–686.
- [104] Pearson, K. (1895). Vii. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, 58(347-352):240–242.
- [105] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- [106] Pölsterl, S., Navab, N., and Katouzian, A. (2015). Fast training of support vector machines for survival analysis. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II 15*, pages 243–259. Springer.
- [107] Pölsterl, S., Navab, N., and Katouzian, A. (2016). An efficient training algorithm for kernel survival support vector machines. *arXiv preprint arXiv:1611.07054*.

- [108] Ramos, C. C., Rodrigues, D., de Souza, A. N., and Papa, J. P. (2016). On the study of commercial losses in brazil: A binary black hole algorithm for theft characterization. *IEEE Transactions on Smart Grid*, 9(2):676–683.
- [109] Raney, B. J., Barber, G. P., Benet-Pagès, A., Casper, J., Clawson, H., Cline, M. S., Diekhans, M., Fischer, C., Navarro Gonzalez, J., Hickey, G., et al. (2024). The ucsc genome browser database: 2024 update. *Nucleic Acids Research*, 52(D1):D1082–D1088.
- [110] Rappaport, N., Twik, M., Plaschkes, I., Nudel, R., Iny Stein, T., Levitt, J., Gershoni, M., Morrey, C. P., Safran, M., and Lancet, D. (2017). Malacards: an amalgamated human disease compendium with diverse clinical and genetic annotation and structured search. *Nucleic acids research*, 45(D1):D877–D887.
- [111] Raymer, M. L., Sanschagrin, P. C., Punch, W. F., Venkataraman, S., Goodman, E. D., and Kuhn, L. A. (1997). Predicting conserved water-mediated and polar ligand interactions in proteins using a k-nearest-neighbors genetic algorithm. *Journal of molecular biology*, 265(4):445–464.
- [112] Romero, P., Wagg, J., Green, M. L., Kaiser, D., Krummenacker, M., and Karp, P. D. (2005). Computational prediction of human metabolic pathways from the complete human genome. *Genome biology*, 6:1–17.
- [113] Safran, M., Rosen, N., Twik, M., BarShir, R., Stein, T. I., Dahary, D., Fishilevich, S., and Lancet, D. (2021). The genecards suite. *Practical guide to life science databases*, pages 27–56.
- [114] Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., and Buetow, K. H. (2009). Pid: the pathway interaction database. *Nucleic acids research*, 37(suppl_1):D674–D679.
- [115] Seal, R. L., Braschi, B., Gray, K., Jones, T. E., Tweedie, S., Haim-Vilmovsky, L., and Bruford, E. A. (2023). Genenames. org: the hgnc resources in 2023. *Nucleic Acids Research*, 51(D1):D1003–D1009.
- [116] Selvanathan, N. and Tee, W. J. (2003). A genetic algorithm solution to solve the shortest path problem in ospf and mpls. *Malaysian Journal of Computer Science*, 16(1):58–67.
- [117] Shah, S., Amannejad, Y., Krishnamurthy, D., and Wang, M. (2019). Quick execution time predictions for spark applications. In *2019 15th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE.
- [118] Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.
- [119] Shaw, A. T., Kim, D.-W., Mehra, R., Tan, D. S., Felip, E., Chow, L. Q., Camidge, D. R., Vansteenkiste, J., Sharma, S., De Pas, T., et al. (2014). Ceritinib in alk-rearranged non-small-cell lung cancer. *New England Journal of Medicine*, 370(13):1189–1197.
- [120] Shen, C., Chen, C., and Rao, G. (2023). A novel multi-task performance prediction model for spark. *Applied Sciences*, 13(22):12242.

- [121] Shi, Y., Ke, G., Chen, Z., Zheng, S., and Liu, T.-Y. (2022). Quantized training of gradient boosting decision trees. *Advances in neural information processing systems*, 35:18822–18833.
- [122] Shirdel, E. A., Xie, W., Mak, T. W., and Jurisica, I. (2011). Navigating the microneome—using multiple microRNA prediction databases to identify signalling pathway-associated microRNAs. *PLoS one*, 6(2):e17429.
- [123] Shukla, A. K., Tripathi, D., Reddy, B. R., and Chandramohan, D. (2020). A study on metaheuristics approaches for gene selection in microarray data: algorithms, applications and open challenges. *Evolutionary intelligence*, 13:309–329.
- [124] Shyr, D. and Liu, Q. (2013). Next generation sequencing in cancer research and clinical application. *Biological procedures online*, 15:1–11.
- [125] Soper, H., Young, A., Cave, B., Lee, A., and Pearson, K. (1917). On the distribution of the correlation coefficient in small samples. appendix ii to the papers of "student." and Ra Fisher. *Biometrika*, 11(4):328–413.
- [126] Spearman, C. (1961). The proof and measurement of association between two things.
- [127] Stelzer, G., Rosen, N., Plaschkes, I., Zimmerman, S., Twik, M., Fishilevich, S., Stein, T. I., Nudel, R., Lieder, I., Mazor, Y., et al. (2016). The genecards suite: from gene data mining to disease genome sequence analyses. *Current protocols in bioinformatics*, 54(1):1–30.
- [128] Suehnholz, S. P., Nissan, M. H., Zhang, H., Kundra, R., Nandakumar, S., Lu, C., Carrero, S., Dhaneshwar, A., Fernandez, N., Xu, B. W., et al. (2024). Quantifying the expanding landscape of clinical actionability for patients with cancer. *Cancer Discovery*, 14(1):49–65.
- [129] Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K. P., et al. (2015). STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research*, 43(D1):D447–D452.
- [130] Thanh Chung, M., Weidendorfer, J., Förlinger, K., and Kranzlmüller, D. (2023). From reactive to proactive load balancing for task-based parallel applications in distributed memory machines. *Concurrency and Computation: Practice and Experience*, 35(24):e7828.
- [131] Thomas, P. D., Ebert, D., Muruganujan, A., Mushayahama, T., Albou, L.-P., and Mi, H. (2022). Panther: Making genome-scale phylogenetics accessible to all. *Protein Science*, 31(1):8–22.
- [132] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.
- [133] Tokar, T., Pastrello, C., Rossos, A. E., Abovsky, M., Hauschild, A.-C., Tsay, M., Lu, R., and Jurisica, I. (2018). mirDIP 4.1—integrative database of human microRNA target predictions. *Nucleic acids research*, 46(D1):D360–D370.

- [134] Verma, A. et al. (2012). A survey on image contrast enhancement using genetic algorithm. *International Journal of Scientific and Research Publications*, 2(7):1.
- [135] Vogelstein, B., Papadopoulos, N., Velculescu, V. E., Zhou, S., Diaz Jr, L. A., and Kinzler, K. W. (2013). Cancer genome landscapes. *science*, 339(6127):1546–1558.
- [136] Waks, A. G. and Winer, E. P. (2019). Breast cancer treatment: a review. *Jama*, 321(3):288–300.
- [137] Wang, K. and Khan, M. M. H. (2015). Performance prediction for apache spark platform. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, pages 166–173. IEEE.
- [138] Whirl-Carrillo, M., Huddart, R., Gong, L., Sangkuhl, K., Thorn, C. F., Whaley, R., and Klein, T. E. (2021). An evidence-based framework for evaluating pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics*, 110(3):563–572.
- [139] Whirl-Carrillo, M., McDonagh, E. M., Hebert, J., Gong, L., Sangkuhl, K., Thorn, C., Altman, R. B., and Klein, T. E. (2012). Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics*, 92(4):414–417.
- [140] Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*, pages 196–202. Springer.
- [141] Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., and Hassanali, M. (2008). Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl_1):D901–D906.
- [142] Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., and Woolsey, J. (2006). Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl_1):D668–D672.
- [143] Wu, X., Fei, M., Wu, D., Zhou, W., Du, S., and Fei, Z. (2023). Enhanced binary black hole algorithm for text feature selection on resources classification. *Knowledge-Based Systems*, 274:110635.
- [144] Yamamoto, S., Sakai, N., Nakamura, H., Fukagawa, H., Fukuda, K., and Takagi, T. (2011). Inoh: ontology-based highly structured database of signal transduction pathways. *Database*, 2011:bar052.
- [145] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauly, M., Franklin, M. J., Shenker, S., and Stoica, I. (2012). Resilient distributed datasets: A {Fault-Tolerant} abstraction for {In-Memory} cluster computing. In *9th USENIX symposium on networked systems design and implementation (NSDI 12)*, pages 15–28.
- [146] Zeng, C., Stroup, E. K., Zhang, Z., Chiu, B. C.-H., and Zhang, W. (2019). Towards precision medicine: advances in 5-hydroxymethylcytosine cancer biomarker discovery in liquid biopsy. *Cancer communications*, 39:1–9.

- [147] Zhang, B., Wang, J., Wang, X., Zhu, J., Liu, Q., Shi, Z., Chambers, M. C., Zimmerman, L. J., Shaddox, K. F., Kim, S., et al. (2014). Proteogenomic characterization of human colon and rectal cancer. *Nature*, 513(7518):382–387.

Apéndice A

Medicina de precisión

A.1. ADN

El ácido desoxirribonucleico, comúnmente abreviado como ADN, es una molécula fundamental que alberga la información genética en los seres vivos. Su descubrimiento revolucionó nuestra comprensión de la biología y ha sido un punto focal en numerosos campos científicos desde entonces.

Desde un punto de vista estructural, el ADN es una larga cadena polimérica formada por la repetición de unidades llamadas nucleótidos. Estos nucleótidos consisten en una base nitrogenada (**A**denina, **C**itosina, **T**imina o **G**uanina), un azúcar (desoxirribosa) y un grupo fosfato. La disposición específica de estas bases a lo largo de la cadena de ADN codifica la información genética. Dentro del núcleo de una célula eucariota, el ADN se organiza en estructuras más grandes llamadas cromosomas. Cada cromosoma consta de una única molécula de ADN altamente enrollada y asociada con proteínas histonas, formando una estructura compacta y densamente empaquetada. Este empaquetamiento permite que el ADN se organice y se compacte de manera eficiente, lo que facilita su transporte y distribución durante la división celular. Además, protege el ADN de daños y desgarros, lo que ayuda a preservar la integridad del material genético.

La función principal del ADN es llevar y transmitir la información genética de una generación a otra, así como proporcionar las instrucciones necesarias para el desarrollo, el crecimiento y el mantenimiento de los organismos vivos. La información del ADN es interpretada por la maquinaria molecular de la célula para producir proteínas y regular procesos biológicos.

A.2. Reguladores de expresión

Un regulador de expresión es una molécula que controla la actividad de los genes en una célula. Su función es modular la cantidad de mRNA producido a partir de un gen específico. Estos reguladores pueden aumentar o disminuir la expresión génica según las necesidades del organismo. Existen dos tipos principales de reguladores de expresión:

- **Activadores:** estos se unen al ADN cerca de un gen y aumentan su expresión. Es decir, que sobre-expresan al gen.
- **Represores:** estos también se unen al ADN, pero su función es reducir la expresión génica. Por ende, los represores sub-expresan al gen.

Hay múltiples tipos de reguladores de expresión, sin embargo, en este trabajo solo nos concentraremos en miRNA, Copy Number Alteration y metilación del ADN.

A.2.1. miRNA

Los miRNAs son unos RNAs pequeños, compuestos por 19 a 25 nucleótidos que desempeñan un papel crucial en la regulación de la expresión génica. El proceso de regulación (Figura A.1) consiste en los siguientes pasos:

1. Comienza con la transcripción de los genes miRNA, generando una molécula llamada *pri-microRNA*. Esta es una molécula de RNA larga y no procesada que contiene secuencias adicionales y estructuras secundarias que deben eliminarse antes de que se forme el miRNA maduro.
2. Una enzima llamada Drosha procesa el *pri-microRNA* para formar el *pre-microRNA*. Este es una forma intermedia antes de convertirse en el miRNA funcional.
3. El *pre-microRNA* se exporta al citoplasma, donde una enzima llamada Dicer forma un dúplex (una estructura formada por dos hebras complementarias) de miRNA.
4. La proteína Argonauta desenrolla dicho dúplex, generando lo que se conoce como miRNA maduro, que es una forma del miRNA de aproximadamente 20 a 22 nucleótidos de longitud lista para ejercer su función reguladora.
5. El miRNA maduro se asocia con la proteína Argonauta que al deshacer una parte del miRNA maduro, deja una cadena de nucleótidos que se conoce como complejo *RNA Induced Silencing Complex (RISC)*.

6. Este complejo es complementario a ciertas cadenas de ADN, cuando se unen a un gen generan la degradación del mRNA o la inhibición de la traducción, generando la sub-expresión de dicho gen.

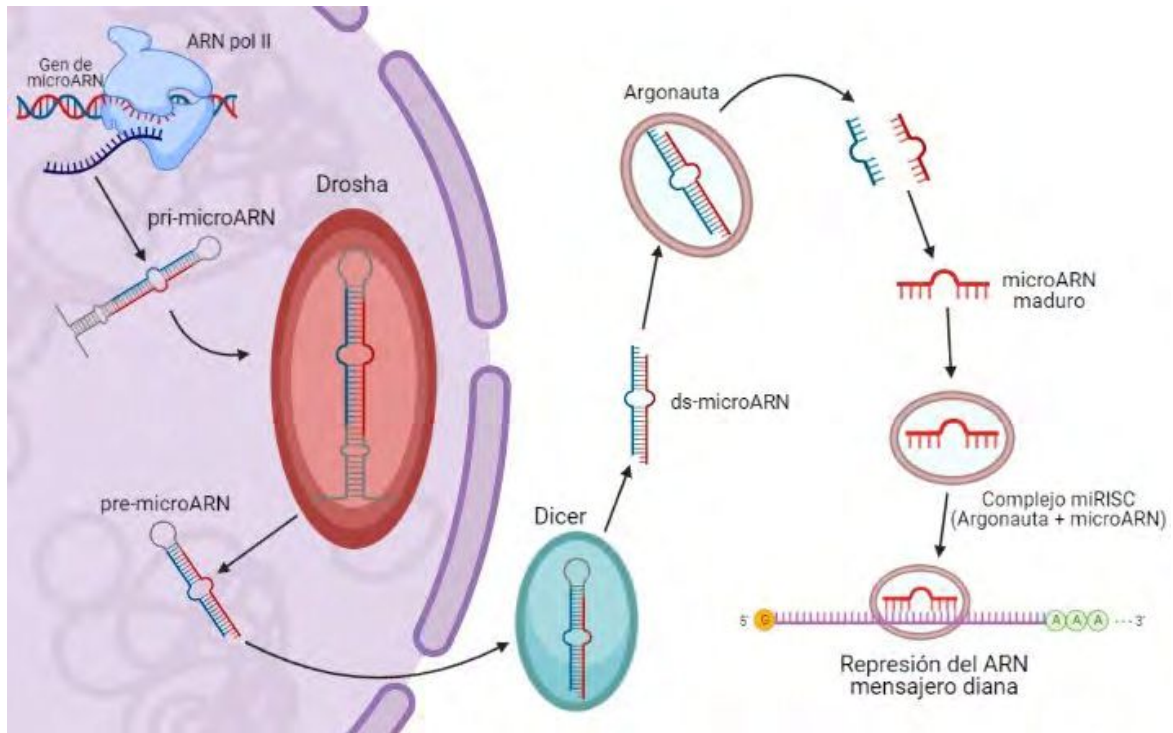


Figura A.1 Síntesis de los miRNAs. En primer lugar se transcriben los genes miRNA, se obtiene el pri-microRNA. A continuación Drosha procesa el miRNA primario para dar lugar a la molécula precursora del miRNA, el pre-microRNA. El precursor es exportado al citoplasma y, una vez allí, la enzima Dicer forma miRNA dúplex, que es posteriormente desenrollado por la proteína Argonauta. Finalmente, una de las cadenas de miRNA, que constituye el miRNA maduro, se queda junto a Argonauta y forman el complejo miRISC, listo para ejercer la represión de sus mensajeros diana.

A.2.2. Copy Number Alteration/Variation (CNA o CNV)

Durante el proceso de replicación del ADN, puede ocurrir que un fragmento del mismo (es decir, un gen) se replique varias veces. Este número de copias de genes está demostrado que puede generar efectos de regulación en la expresión de los genes involucrados.

La alteración en el número de copias (CNA por sus siglas en inglés) se refiere a una circunstancia en la que el número de copias de un segmento específico de ADN varía entre los genomas de diferentes individuos. Las CNA regulan la expresión génica de diferentes maneras [42], dependiendo del grado de solapamiento con los genes, la localización de la

CNA y el tipo de gen afectado. A continuación se listan algunos posibles mecanismos (Figura A.2):

- **Efecto dosis génica:** las duplicaciones o deleciones génicas pueden alterar la expresión de genes sensibles a la dosis. Si se duplica un gen, se incrementa su expresión, mientras que si se delecciona, disminuye.
- **Disrupciones estructurales:** las CNA que solo solapan parcialmente con un gen pueden inducir expresión reducida o generar nuevos transcritos al interrumpir la estructura del gen.
- **Modificación de elementos reguladores:** las CNA pueden regular la expresión de genes flanqueantes normales de manera distante, insertando o eliminando elementos reguladores como promotores o potenciadores a cientos de miles de nucleótidos de distancia.
- **Efectos de posición:** las CNA pueden alterar la proximidad física entre genes y sus elementos reguladores, afectando la expresión génica por efecto de posición.
- **Enmascaramiento de variantes reguladoras:** las CNA pueden desenmascarar el efecto de SNPs funcionales al modificar su número de copias.
- **Mecanismos epigenéticos:** las CNA pueden influir en la expresión génica a través de mecanismos epigenéticos como alteraciones en la arquitectura de la cromatina o su organización nuclear 3D.

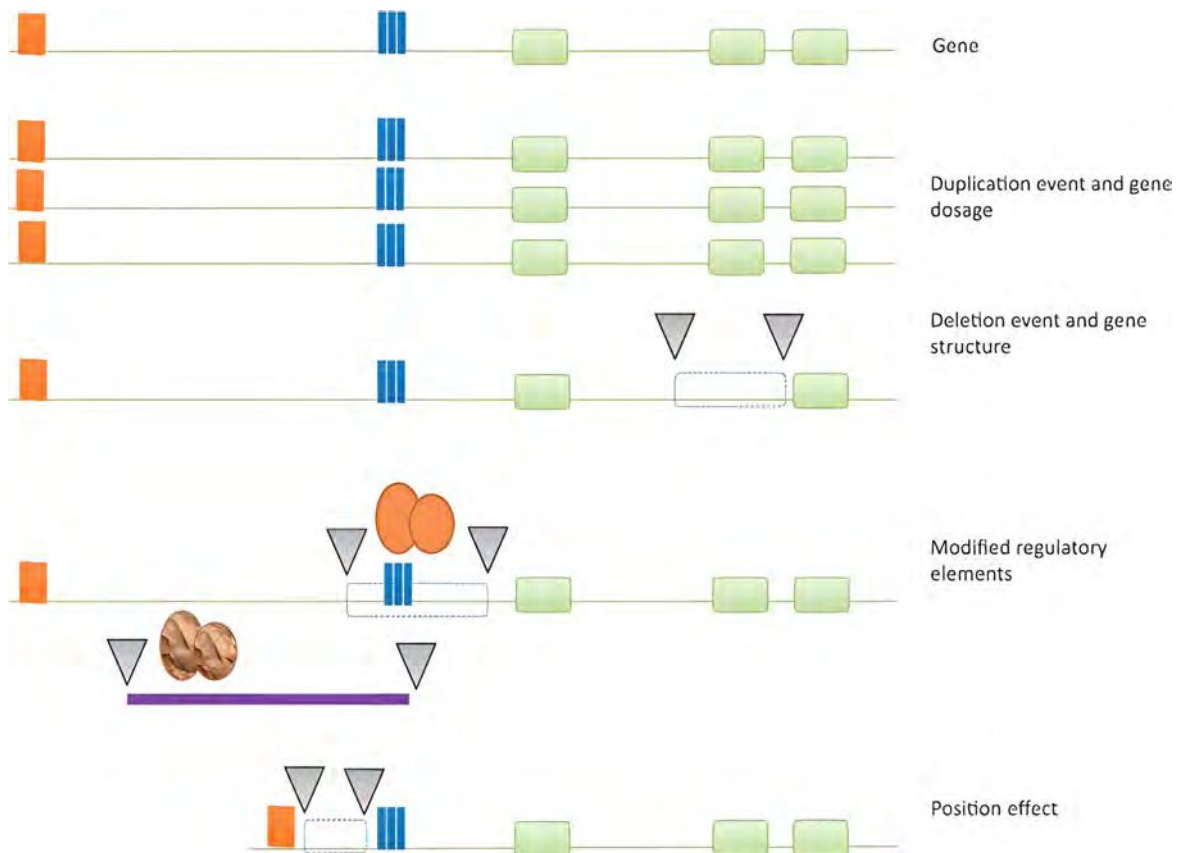


Figura A.2 Múltiples mecanismos a través de los cuáles las CNA regulan la expresión génica. Las cajas verdes representan exones. Las cajas azules representan promotores, mientras que las cajas naranjas representan potenciadores distales. Los triángulos marcan los puntos de ruptura de las CNA. Los círculos representan los factores de transcripción que pueden unirse a los elementos reguladores.

A.2.3. Metilación de ADN

La metilación del ADN es una modificación química que implica la adición de grupos metilo (una molécula compuesta por un átomo de carbono y tres átomos de hidrógeno) a ciertas posiciones del ADN (Figura A.3). Esta metilación inhibe la expresión génica de forma directa, ya que desplaza la unión habitual al ADN de los factores activadores de la transcripción y de forma indirecta, atrayendo a unas proteínas de unión a Citosinas (proteínas que controlan el crecimiento y la actividad de otras células del sistema inmunitario y las células sanguíneas) metiladas que actuarían reprimiendo la transcripción. Al alterar el proceso de transcripción de un gen, su expresión se ve reducida.

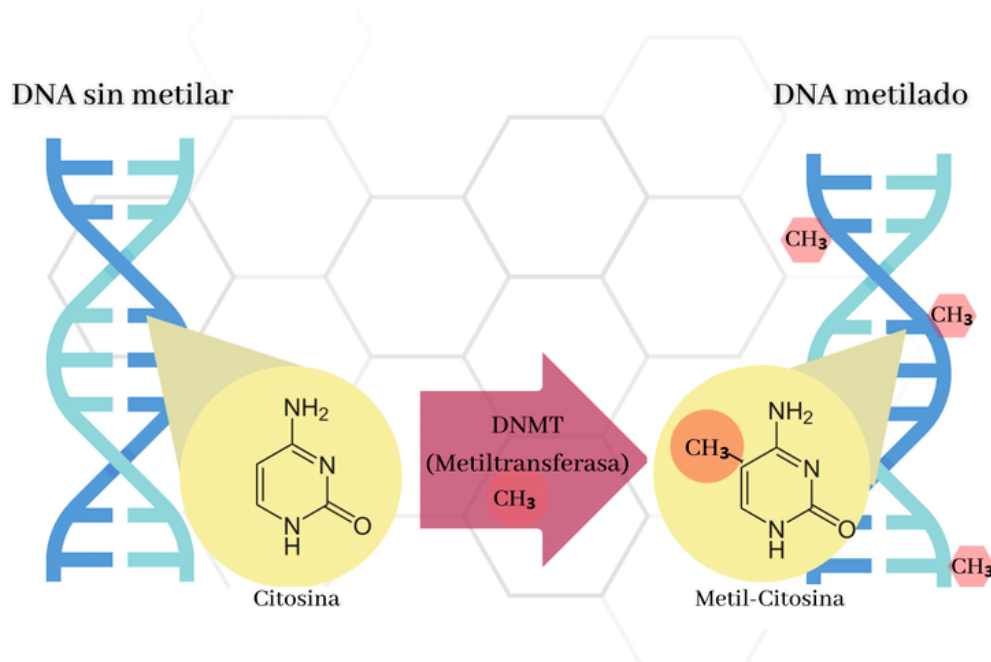


Figura A.3 En el proceso de metilación el grupo metilo se adhiere a la Citosina, impidiendo que los procesos de transcripción se adhieran a ella.

A.3. Pathways

Un pathway, también conocido como vía metabólica, es una serie de reacciones químicas interconectadas que ocurren dentro de una célula para llevar a cabo una función específica (Figura A.4). Estas vías metabólicas son esenciales para la vida, ya que permiten que las células produzcan energía, construyan y reparen tejidos, y realicen otras funciones importantes.

Cada pathway comienza con una molécula inicial, llamada sustrato, que se convierte en una serie de productos a través de una serie de reacciones químicas catalizadas por enzimas. Cada reacción química en el pathway es específica y está diseñada para producir un producto específico que se utiliza en la siguiente reacción.

Los pathways pueden ser lineales o ramificados, y pueden ser anabólicos o catabólicos. Los pathways anabólicos son aquellos que construyen moléculas más grandes a partir de moléculas más pequeñas, mientras que los pathways catabólicos son aquellos que descomponen moléculas más grandes en moléculas más pequeñas para liberar energía.

Los pathways también pueden ser regulados por la célula para asegurar que se produzcan los productos necesarios en las cantidades correctas. La regulación puede ocurrir a nivel de la

enzima, donde la actividad de la enzima se ajusta para aumentar o disminuir la velocidad de la reacción, o a nivel del pathway, donde se activan o desactivan las vías metabólicas enteras.

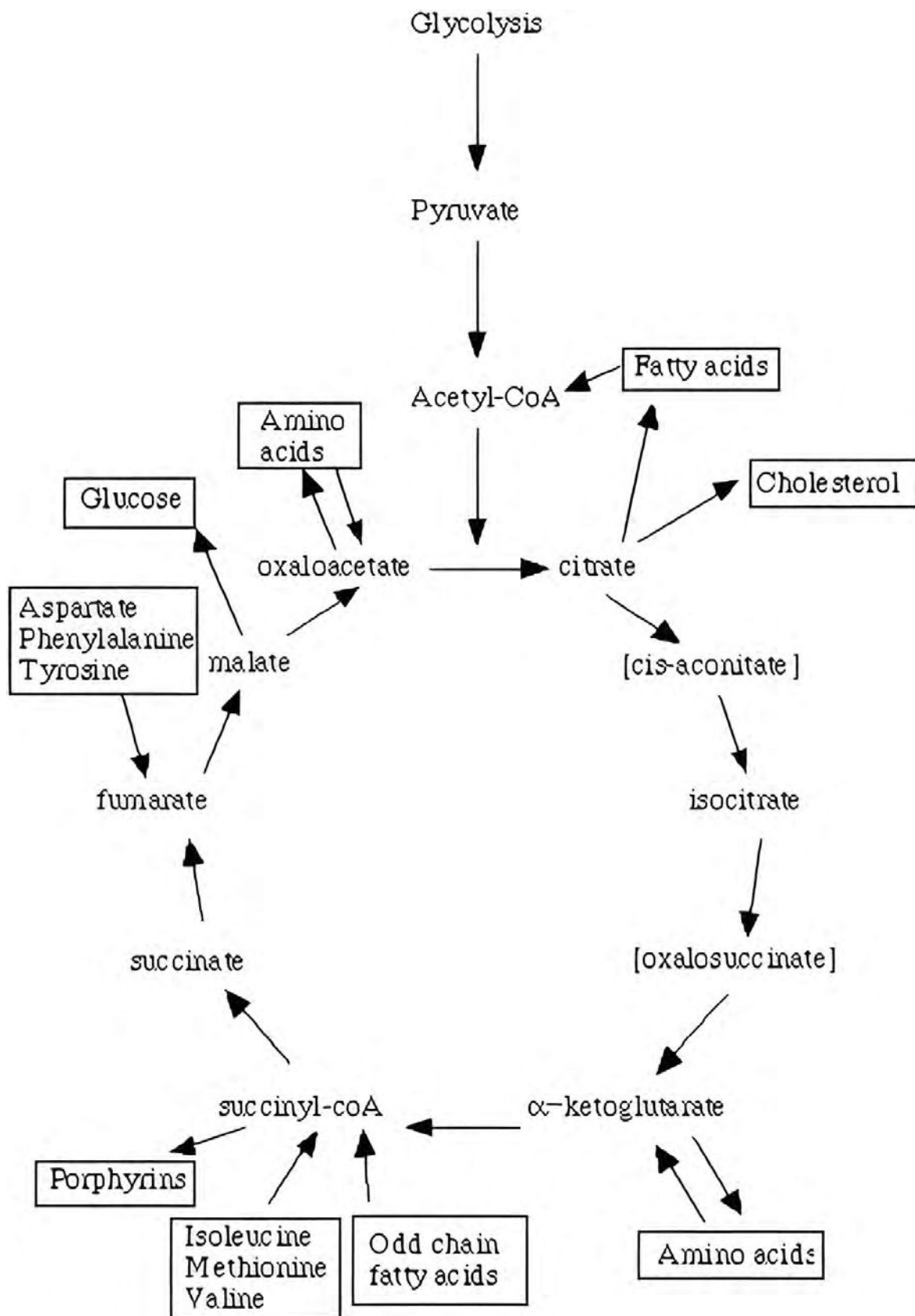


Figura A.4 Ejemplo del diagrama de un pathway. En este caso representa las propiedades anfóteras del ciclo del ácido cítrico. Este ciclo es una vía metabólica clave en la respiración celular. En él, participan reacciones tanto anabólicas como catabólicas, y ocurre en las mitocondrias de las células eucariotas y en el citoplasma de las células procariotas.

A.4. Blancos terapéuticos

Los enfoques tradicionales para el descubrimiento de blancos terapéuticos impulsados por la genómica/epigenómica se inician mediante la agrupación no supervisada de perfiles moleculares y genómicos de muestras de pacientes [24][57] (se realiza un análisis más exhaustivo sobre estas técnicas en la Sección 2.4.2). Estos métodos han llevado a la identificación de varias vulnerabilidades genómicas accionables, dando lugar a grandes éxitos en el tratamiento de diversos cánceres. Uno de los casos más resonados es la terapia para la inhibición del gen *BRAF* activado en el cáncer de pulmón, colon y melanoma [27][39]. Otra terapia apunta a la inhibición de los genes *RTK*, *L858R* y *ALK* en el cáncer de pulmón [85][98][119].

La NGS también se ha utilizado para guiar las decisiones terapéuticas en pacientes con cáncer de mama, las cuales son tratadas en función de su expresión del gen *ERBB2/Her2* [136]. En el caso del cáncer colorrectal, los pacientes con la mutación *K-RAS (G12D/G12V)* se asocian con la falta de respuesta al tratamiento con el medicamento Cetuximab [70] y probablemente reciban opciones terapéuticas alternativas. Estos ejemplos demuestran la utilidad de cómo las firmas mutacionales han guiado decisiones de tratamiento específicas en la clínica.

Así y todo, la información genética no es la única que sirve como biomarcador para el tratamiento de enfermedades. Se podría combinar esa información con los datos de reguladores de expresión, proteómica, pathways, perfiles de metabolitos, entre otros, para obtener una predicción más robusta para el tratamiento de los pacientes. Esta comprensión "multi-ómica" del cáncer es esencial, ya que el impacto acumulativo en el transcriptoma y el proteoma expresados no está determinado únicamente por las variantes genéticas. Por ejemplo, la reconfiguración epigenética de las células cancerígenas producto de la metilación del ADN [84], o el uso diferencial de potenciadores y promotores, pueden alterar sus transcriptomas que eventualmente gobiernan la progresión de la enfermedad, incluida la resistencia al tratamiento y la respuesta a varias quimioterapias [11]. Un nuevo estudio destaca la incoherencia entre la expresión de mRNA y proteínas en muestras de tumores de colon y recto [147], lo que sugiere que una comprensión multi-ómica de las células cancerígenas, puede ser crucial para identificar los mecanismos que controlan la progresión metastásica en la clínica.

A.5. Métodos de correlación

A.5.1. Pearson

Este método estadístico mide la relación lineal entre dos variables. Si hay una relación lineal positiva, el valor de la correlación es cercano a 1; si es negativa, el valor es cercano a -1. Un valor cercano a 0 indica una correlación débil o nula. La función de Pearson se puede definir de la siguiente manera:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

Donde:

- X_i e Y_i son los valores individuales de las dos variables.
- \bar{X} y \bar{Y} son las medias de X e Y , respectivamente.

A.5.2. Spearman

Este método evalúa la relación monotónica (al aumentar o disminuir los valores de una variable, los valores de la otra variable siempre siguen la misma dirección, ya sea creciendo o decreciendo respectivamente) entre dos variables, lo que significa que puede identificar correlaciones no lineales. Al igual que Pearson, asigna un coeficiente de correlación que puede variar de -1 a 1 (con la misma interpretación de dichos valores). Spearman es menos afectado por los outliers (valores que se alejan significativamente de la mayoría de los demás valores en un conjunto de datos), ya que se basa en los rangos y no en los valores exactos de los mismos. Su fórmula se detalla a continuación:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Donde:

- d_i son las diferencias entre los rangos de las dos variables para cada observación.
- n es el número de observaciones.

A.5.3. Kendall

Similar a Spearman, evalúa la correlación entre dos variables, pero se centra en la concordancia o discordancia del orden relativo de los datos. El coeficiente de correlación de Kendall también oscila entre -1 y 1 y su función se puede definir de la siguiente manera:

$$\tau = \frac{(\text{Número de pares concordantes}) - (\text{Número de pares discordantes})}{\frac{1}{2}n(n-1)}$$

Donde:

- n es el número de observaciones.
- *Pares concordantes*: son aquellos pares de observaciones en los que las dos variables mantienen el mismo orden relativo. Es decir, si una observación tiene un valor más alto que otra observación en una variable, también tendrá un valor más alto en la otra variable.
- *Pares discordantes*: son aquellos pares de observaciones en los que las dos variables tienen un orden relativo opuesto. Es decir, si una observación tiene un valor más alto que otra observación en una variable, tendrá un valor más bajo en la otra variable.

A.6. Ajuste de p-valor

El ajuste del p-valor es un paso crucial en el análisis estadístico para evitar errores de tipo I (falsos positivos) cuando se realizan múltiples pruebas de hipótesis.

Sin ajuste, el riesgo de obtener un resultado significativo por casualidad aumenta con cada prueba adicional. Esto se debe a que el p-valor individual solo refleja la probabilidad de obtener un resultado tan extremo o más extremo asumiendo que la hipótesis nula es verdadera. En cambio, al realizar el ajuste del p-valor, se toma en cuenta el número total de pruebas realizadas, lo que reduce la probabilidad de obtener falsos positivos. Existen diversas técnicas para ajustar el p-valor, en esta tesis se estudian las técnicas de Bonferroni, Benjamini-Hochberg y Benjamini-Yekutieli.

A.6.1. Bonferroni

El método Bonferroni [10] consiste en dividir el nivel de significancia original por el número de pruebas realizadas. Es simple de aplicar, pero puede ser demasiado conservador en algunos casos, porque al disminuir mucho el nivel de significancia, especialmente cuando

hay un gran número de pruebas, la probabilidad de cometer un error de tipo II (no rechazar la hipótesis nula cuando es falsa) aumenta considerablemente.

En otras palabras, el método de Bonferroni protege bien contra los falsos positivos (error tipo I), pero a costa de aumentar la probabilidad de falsos negativos (error tipo II), lo cual puede llevar a no detectar efectos o diferencias reales en los datos.

Sea H_1, \dots, H_m una familia de hipótesis, y p_1, \dots, p_m sus p-valores correspondientes. Sea m el número total de hipótesis nulas y m_0 el número de hipótesis nulas verdaderas. La *Tasa de Error Familiar* (FWER por sus siglas en inglés) es la probabilidad de rechazar al menos un H_i verdadero, es decir, de cometer al menos un error de tipo I. La corrección de Bonferroni rechaza la hipótesis nula para cada $p_i \leq \frac{\alpha}{m}$, controlando así el FWER en $\leq \alpha$. La prueba de este control se deriva de la desigualdad de Boole, como sigue:

$$\text{FWER} = P\left\{\bigcup_{i=1}^{m_0} \left(p_i \leq \frac{\alpha}{m}\right)\right\} \leq \sum_{i=1}^{m_0} \left\{P\left(p_i \leq \frac{\alpha}{m}\right)\right\} = m_0 \frac{\alpha}{m} \leq m \frac{\alpha}{m} = \alpha.$$

A.6.2. Benjamini-Hochberg

El método de Benjamini-Hochberg [6] es menos conservador que el de Bonferroni y permite controlar la FWER. BH se distingue por limitar la *Tasa de Descubrimiento Falso* (FDR por sus siglas en inglés) en lugar de la FWER. Cuando consideramos cada medición que se declara estadísticamente significativa como un descubrimiento, la FDR representa la proporción de estos hallazgos que son falsos, es decir, meramente ruido sin un impacto real. La sutil distinción entre la FWER y la FDR radica en que la FWER controla la fracción de experimentos que tienen uno o más descubrimientos falsos, mientras que la FDR regula la proporción de todos los descubrimientos que resultan ser falsos.

El procedimiento BH se puede definir con la siguiente serie de pasos:

1. Ordenar los p-valores de forma ascendente, de modo que el p-valor más pequeño sea el primero y el más grande sea el último.
2. Calcular el valor crítico q para un nivel de FDR deseado (α). La fórmula para calcular q es la siguiente:

$$q = k * \alpha / m$$

Donde:

- k es el índice de la prueba con el p-valor ordenado más pequeño.
- α es el nivel de FDR deseado.
- m es el número total de pruebas realizadas.

3. Identificar las pruebas significativas comparando los p-valores con el valor crítico q . Las pruebas donde se cumple que $p\text{-valor} \leq q$ se consideran significativas, mientras que las que tienen un p-valor mayor que q no se consideran significativas.

A.6.3. Benjamini-Yekutieli

El método Benjamini-Yekutieli [7][8] es similar a BH, pero es más flexible y permite controlar la FDR. La elección de la técnica de ajuste del p-valor depende de varios factores, como el número de pruebas realizadas, la dependencia entre las pruebas y el tipo de error que se desea controlar.

El procedimiento de BY es similar a BH, los p-valores también deben ordenarse de menor a mayor, pero q ahora se calcula como $q = k * \alpha' / m$ donde $\alpha' = \alpha / \sum_{i=1}^k 1/i$. Siendo este método más conservador que BH.

Apéndice B

Multiomix

B.1. Proceso de creación de un análisis de correlación

En la plataforma, este tipo de análisis comienza con la selección de un conjunto de datos de expresiones génicas, y otro dataset con los datos de un GEM específico, ya sea miRNA, CNA y metilación de ADN.

Ya seleccionados los conjuntos, esta evaluación se lleva a cabo mediante los métodos de correlación Pearson, Spearman o Kendall. Y con el fin de descartar resultados que hayan surgido del azar, se realiza un ajuste de p-valor, los tres métodos disponibles son Bonferroni, Benjamini-Hochberg, Benjamini-Yekutieli.

El formulario en cuestión consiste en la selección de ambos conjuntos de datos (mRNA y el de GEM: miRNA, CNA o metilación), un nombre de análisis y una descripción opcional (Figura B.1a). Además, se debe seleccionar por parte del usuario uno de los tres algoritmos de correlación a ejecutar y el mecanismo de ajuste de p-valor. Por defecto el sistema define una serie de filtros por desviación estándar (cualquier valor con una desviación menor a dichos valores serán descartados), dichos valores pueden ser cambiados por el usuario antes de la ejecución del análisis (Figura B.1b).

Una vez finaliza el análisis, se pueden visualizar todos los análisis de correlación ejecutados y sus estados (Figura B.2). Dicha tabla muestra el nombre, descripción, fecha de ejecución, estado, tipo de GEM, método de correlación, cantidad de combinaciones que fueron guardadas y evaluadas, dataset clínico asociado al análisis, un tag para una fácil identificación por parte del usuario, los datasets utilizados y una columna de acciones para ver los detalles, descargar los resultados, editar datos básicos de un análisis en particular y por último, eliminar dicho análisis. En esta tabla también se ponen a disposición diferentes filtros por tag, tipo de GEM, método de correlación y un buscador por nombre y descripción de los experimentos.

The image shows two parts of a web form for correlation analysis. Part (a) is the dataset selection section, and part (b) is the advanced settings section.

(a) Selección de los datasets.

This section is titled "miRNA" and contains two main input areas:

- mRNA profile:** Includes a DNA double helix icon, a "Select dataset..." dropdown menu, and a "Samples mRNA: 0" label.
- MiRNA profile:** Includes a miRNA hairpin icon, a "Select dataset..." dropdown menu, and a "Samples MiRNA: 0" label.

Below these are summary statistics: "Samples in common: 0".

Analysis info: A section with a text input field (marked with an asterisk), a "No tag assigned" button, and a "Upload files (optional)" area.

(b) Métodos de correlación y filtros.

This section is titled "Advanced Settings" and contains the following controls:

- Correlation method:** A dropdown menu set to "Pearson".
- Min. Correlation Threshold:** A slider set to 0.70, with buttons for 0.5 and 0.95.
- Genes Min. Standard Deviation:** A slider set to 0, with buttons for 0 and 0.95.
- MiRNA Min. Standard Deviation:** A slider set to 0, with buttons for 0 and 0.95.
- P-value adjustment:** A dropdown menu set to "Benjamini-Hochberg".
- * Required field:** A green "Run analysis" button.
- Warning:** A yellow box stating "The upload time is subject to the size of the file".
- Clear form:** A red "Clear form" button.

(a) Selección de los datasets.

(b) Métodos de correlación y filtros.

Figura B.1 Formulario completo para ejecutar un análisis de correlación.

Name	Description	Date	State	Type	Cor. Method	N° Combinations	Clinical	Tag	Sources	Actions
Kidney TCGA	Kidney Cancer TCGA	10/6/2021	✓	MIRNA	Pearson	235 / 3005000		--		
Breast TCGA	Breast Cancer TCGA	22/2/2021	✓	MIRNA	Pearson	213 / 6727394		breast		
breast cna	Copy number alterations for breast cancer	4/1/2021	✓	CNA	Pearson	18606 / 386384823		breast		
Colon	Colon Cancer Methylation	4/1/2021	✓	Methylation	Pearson	1904 / 223530570		colon		
met s k		15/12/2020	✓	Methylation	Kendall	0 / 206639004		--		
met s g	Breast cancer methylation Spearman Analysis	15/12/2020	✓	Methylation	Spearman	2811 / 206639004		breast		
met a p		15/12/2020	✓	Methylation	Pearson	2175 / 206639004		--		
met k		15/12/2020	✓	Methylation	Kendall	0 / 206639004		--		
met g s	Interesting result. Evaluate with wet lab!	15/12/2020	✓	Methylation	Spearman	34 / 206639004		breast		
cna g p	Cancer analysis for Methylation - Spearman	15/12/2020	✓	Methylation	Pearson	47 / 206639004		breast		

Figura B.2 Tabla de los análisis de correlación ejecutados con funciones de búsqueda, filtros, paginación, ordenamiento y acciones.

B.2. Detalles del análisis de correlación

La plataforma dispone de un panel para acceder a los detalles de un análisis de correlación particular. En él, se mostrará una tabla (Figura B.3) con las combinaciones GEM-gen que superaron los filtros configurados. En dicha tabla se ponen a disposición los datos de manera paginada, junto a un buscador, filtros por valor de correlación y botones para la descarga de los resultados. Los datos visualizados están conformados por el GEM, el gen, información del gen (cromosoma en el que se encuentra, la posición de inicio y fin en el mismo, el tipo de gen y una descripción breve), el coeficiente y p-valor obtenidos por el algoritmo de correlación, y el p-valor ajustado por el método de ajuste seleccionado por el usuario al momento de ejecutar el análisis.

En la última columna figura un botón para ver los detalles de una combinación GEM-gen particular. Si se ejecuta dicha acción se abrirá una ventana con múltiples paneles y opciones que se detallan en la siguiente Sección.

B.2.1. Propiedades estadísticas de la combinación GEM-gen

En el primer panel se aprecia diferentes propiedades estadísticas básicas de los datos de los pacientes de ambos datasets para ambas moléculas evaluadas, como el promedio, el

miRNA	mRNA	Chromosome	Gene start (bp)	Gene end (bp)	Gene type	Gene desc.	Correlation	P-value	Adj. P-value (Benj.)	Actions
hsa-miR-577	MLPH	2	238394071	238463961	protein_coding	melanophilin [Sour...	-0.7703	2.523e-148	8.714e-144	...
hsa-miR-934	FOXA1	14	38059189	38069245	protein_coding	forkhead box A1 [S...	-0.7412	1.260e-131	2.743e-127	...
hsa-miR-934	MLPH	2	238394071	238463961	protein_coding	melanophilin [Sour...	-0.7346	4.042e-128	7.991e-124	...
hsa-miR-934	CA12	15	63613577	63674360	protein_coding	carbonic anhydras...	-0.7221	8.147e-122	1.353e-117	...
hsa-miR-577	FOXA1	14	38059189	38069245	protein_coding	forkhead box A1 [S...	-0.7132	1.615e-117	2.356e-113	...
hsa-miR-934	AGR2	7	16831435	16873057	protein_coding	anterior gradient 2...	-0.7129	2.255e-117	3.274e-113	...
hsa-miR-934	GATA3	10	8095567	8117161	protein_coding	GATA binding prot...	-0.7119	6.196e-117	8.909e-113	...
hsa-miR-18a	MLPH	2	238394071	238463961	protein_coding	melanophilin [Sour...	-0.7108	2.001e-116	2.823e-112	...
hsa-miR-577	PRR15	7	29603427	29606911	protein_coding	proline rich 15 [Se...	-0.7092	1.191e-115	1.657e-111	...
hsa-miR-577	AGR2	7	16831435	16873057	protein_coding	anterior gradient 2...	-0.7015	3.791e-112	4.869e-108	...

Figura B.3 Tabla de las combinaciones resultantes de un análisis de correlación, con funciones de búsqueda, filtros, paginación, ordenamiento y acciones.

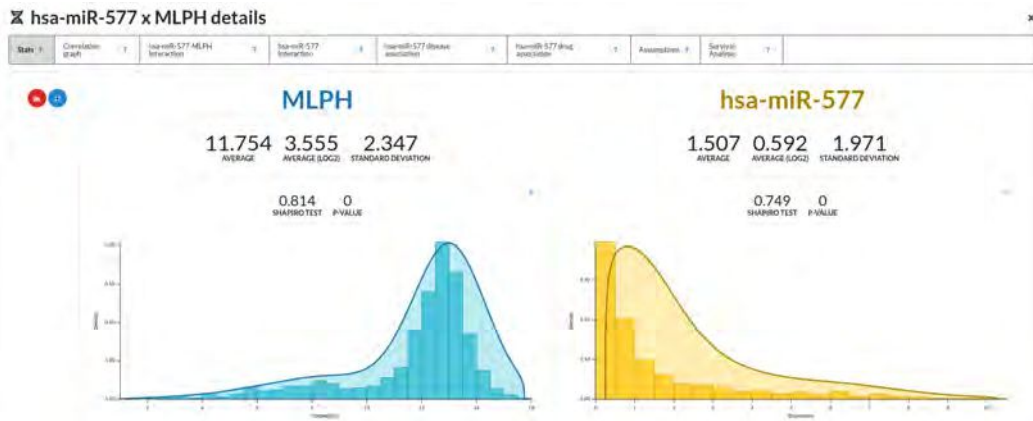
promedio (log), el desvío, y un gráfico de áreas de intervalos para visualizar la distribución (Figuras B.4a y B.4b).

Entre varios datos básicos como la cantidad de pacientes en común entren ambos datasets, se muestra el resultado de diferentes test estadísticos útiles para validar la naturaleza de los datos (Figura B.4c), como el test de Shapiro-Wilk [118] que evalúa si un conjunto de datos determinando sigue una distribución normal. El test de Breusch-Pagan [16], el cual verifica la homocedasticidad en regresiones, analizando si la varianza de los errores es constante. El test de Goldfeld-Quandt [48], que examina la homocedasticidad en regresiones, centrándose en la variabilidad de los errores en diferentes subgrupos de datos. Y por último el coeficiente de correlación de Spearman, que evalúa la relación monótonica entre dos variables.

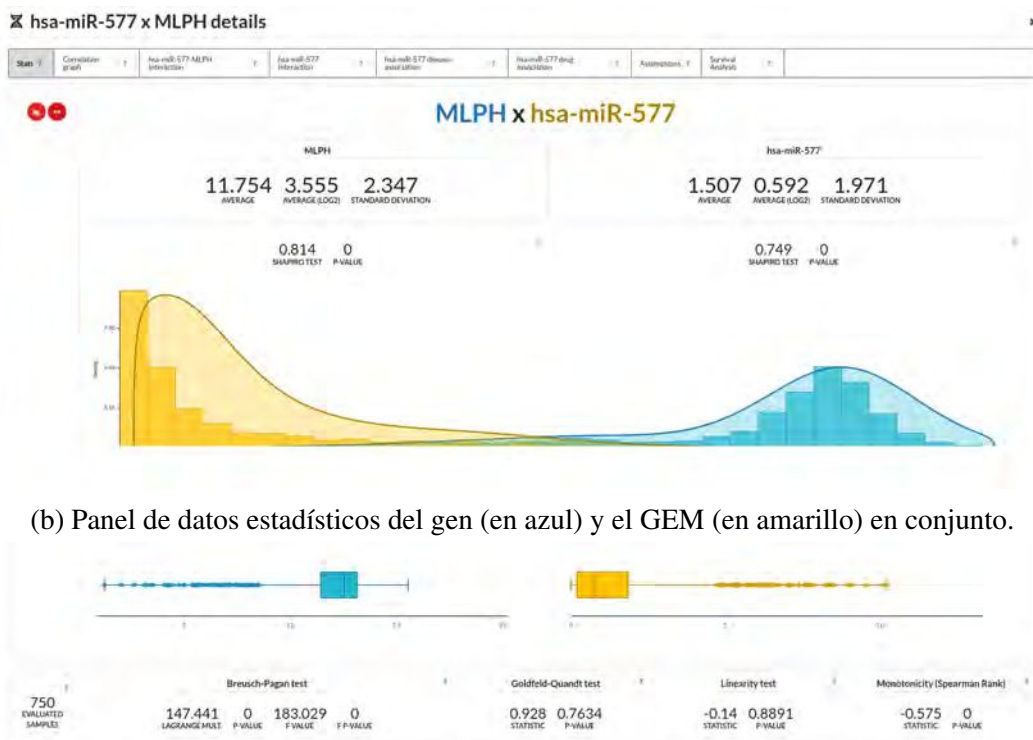
En el mismo sector se listan también los outliers, obtenidos a partir de la Desviación Absoluta Mediana (*MAD* por sus siglas en inglés) [44] mostrando la referencia de la media en un gráfico BoxPlot.

B.2.2. Gráfico de correlación

Siguiendo en orden los paneles dentro de la ventana de detalles, en la siguiente sección se puede encontrar un gráfico Scatter donde los valores de expresión del GEM se muestran sobre el eje X y los del mRNA sobre el eje Y para todos los pacientes. Además, se grafica junto una línea de regresión que se ajusta a todos los valores de expresión para visualizar mejor la correlación de los datos (Figura B.5).



(a) Panel de datos estadísticos del gen (en azul) y el GEM (en amarillo).



(c) Outliers para el dataset del gen (en azul) y el GEM (en amarillo) y otros tests estadísticos.

Figura B.4 Panel estadístico de ambas moléculas involucradas en el análisis de correlación. El sistema permite unificar el gráfico para realizar una comparación directa.

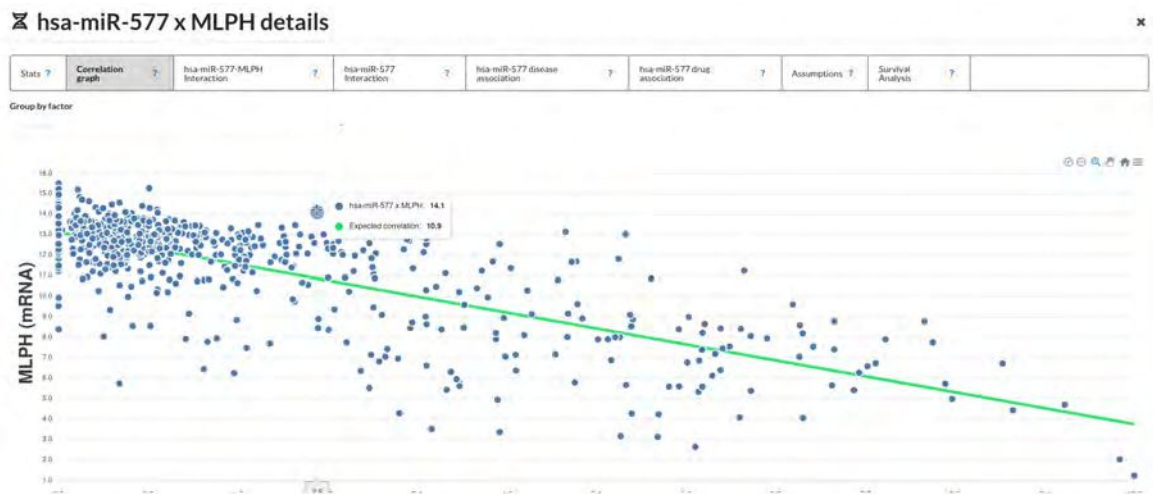


Figura B.5 Gráfico de correlación donde figuran todos los pacientes de los datasets utilizados en el análisis juntos a sus niveles de expresión para el GEM (eje X) y el gen (eje Y) en estudio.

B.2.3. Información de interacciones miRNA-gen

Multiomix no se limitan a las pruebas estadísticas y las herramientas gráficas, sino también que permite al investigador obtener información sobre algunos de los elementos involucrados. Por ejemplo, en el caso de que se estén utilizando datos de miRNA como GEM aparecerán cuatro paneles adicionales en el menú de la ventana. Uno con información sobre las interacciones entre la combinación miRNA-gen en estudio (Figura B.6), otro con todas las interacciones registrada para el miRNA pero para cualquier gen (no limitándose al que está siendo estudiado). Dichas interacciones en ambos paneles son obtenidas a partir de Modulector (Sección 4.3.1) y su integración de la base de datos mirDIP.



Figura B.6 Panel de detalles sobre una interacción miRNA-gen donde figura el puntaje asignado por mirDIP junto con las publicaciones que lo avalan. Arriba, el alias y la secuencia genética del miRNA en cuestión.

B.2.4. Patologías y drogas asociadas a miRNA

Los siguientes dos paneles (también únicos para el tipo de GEM miRNA), consisten en un listado de patologías y drogas asociadas al miRNA específico. En el primero (Figura B.7), se muestra la patología y las fuentes científicas que avalan dicha asociación. En el caso del panel de drogas (Figura B.8), se dispone de información más detallada, como la droga, el método de detección, la small molecule (compuestos orgánicos de bajo peso molecular que pueden interactuar con biomoléculas como proteínas o ácidos nucleicos), si dicha droga aumenta o disminuye la expresión del miRNA en cuestión, una referencia bibliográfica, la validación estadística sobre la que se basó la conclusión, si es una droga aprobada por la agencia regulatoria de medicamentos de Estados Unidos (*Food and Drug Administration* o FDA) y el enlace a la publicación en Pubmed.



Figura B.7 Panel de detalles sobre las patologías reportadas que involucran al miRNA en estudio.

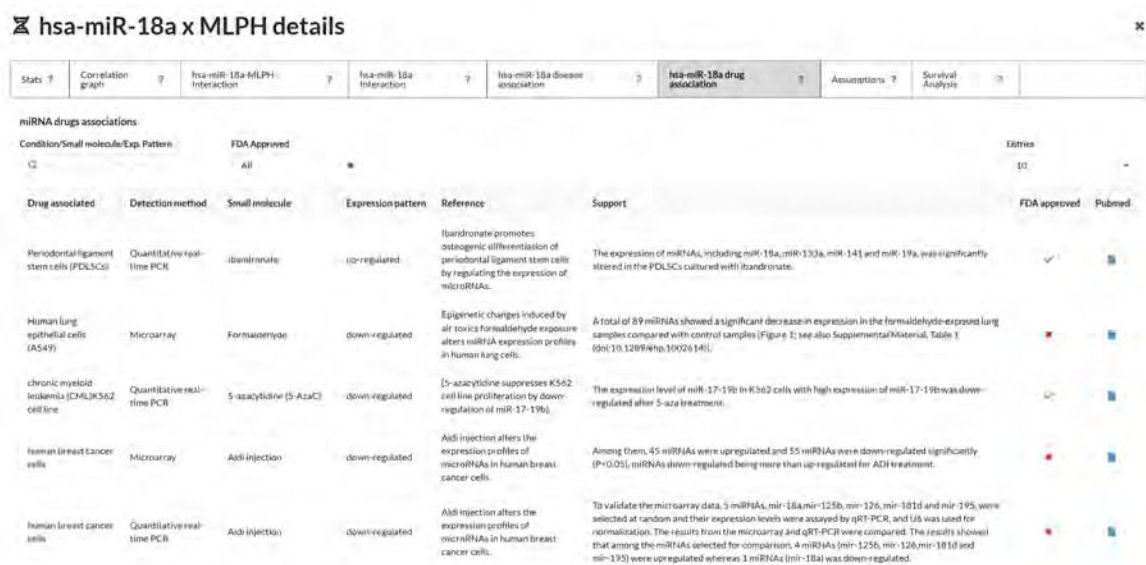


Figura B.8 Panel de detalles sobre las drogas reportadas que involucran al miRNA en estudio.

En los cuatro paneles anteriormente mencionados también se pone a disposición información relevante del miRNA en estudio, como el alias *MIMAT* del miRNA (que es un alias estandarizado) y la secuencia genética que lo conforma. Esta información es la misma que se observa en la Figura B.24 y se obtiene de Modulector que incorpora los datos de miRBase.

B.2.5. Supuestos estadísticos

Se dispone de un panel de supuestos estadísticos (Figura B.9) donde se explica al usuario algunas propiedades de sus fuentes de datos, como si los datos están normalizados, si posee outliers, si cumple con los tests de linealidad y homocedasticidad, siempre dentro del contexto del método de correlación seleccionado para realizar el análisis. De esta manera se proveen explicaciones para que el usuario pueda optar por un método de evaluación más propicio para la naturaleza de sus datos.



Figura B.9 Panel de supuestos estadísticos.

B.2.6. Gráfico de análisis de supervivencia

El último panel respecta la análisis de supervivencia, en él se debe seleccionar una fuente de datos clínicos para realizar un análisis del poder pronóstico al utilizar la información del gen y GEM seleccionados. Los datos clínicos consisten en un archivo tabular que puede contener algunos atributos clínicos como la edad del paciente a la hora de hacerse el estudio, si fuma, etc. Dicho dataset debe contener dos columnas, una que indique la ocurrencia del evento (el evento puede tratarse de la muerte del paciente o la recidiva del tumor), y otra columna que indique el tiempo en el que ocurrió el evento (si ocurrió). Con esa información, y a través del agrupamiento de los pacientes en dos grupos (aquellos cuyo nivel de expresión para el gen/miRNA está por encima de la media, y aquellos que se encuentran por debajo) se puede graficar las curvas Kaplan-Meier (Figura B.10).

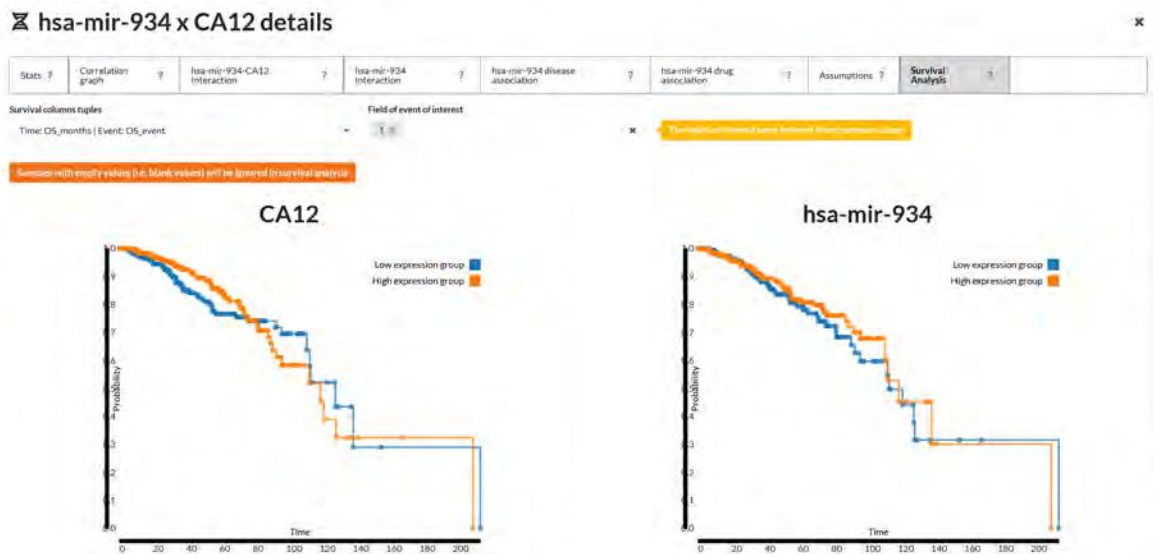


Figura B.10 Panel de análisis de supervivencia para un par GEM-gen particular.

B.3. Proceso de creación de un biomarcador

La funcionalidad de biomarcadores de Multiomix comienza por la creación de uno, y este proceso se puede dar de dos maneras diferentes (Figura B.11).

Por un lado, se puede crear manualmente, lo que permite seleccionar de forma específica las moléculas que conformarán el biomarcador. Por otro lado, existe la posibilidad de realizar FS a partir de un biomarcador ya existente. En este segundo caso, se aplican algoritmos de selección de características para identificar un subconjunto óptimo de moléculas a partir de un conjunto inicial más amplio. Esta opción permite refinar y optimizar un biomarcador existente basándose en criterios específicos.

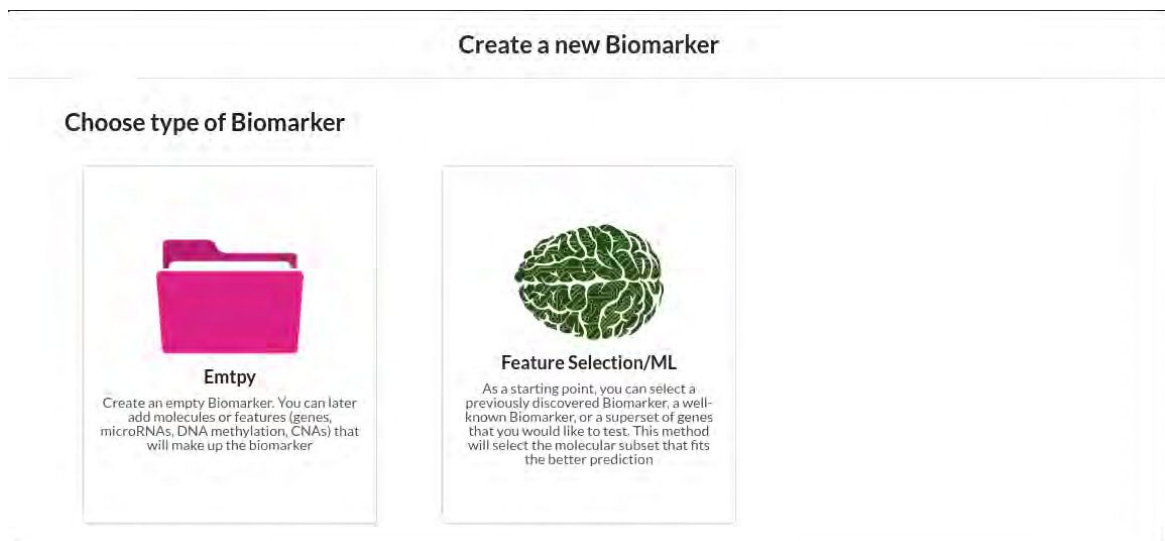


Figura B.11 Panel de creación de un nuevo biomarcador. Ofrece la opción manual a la izquierda, y la opción utilizando FS a la derecha.

Creación manual

Seleccionando la opción de creación manual (Figura B.12) se abre un menú con dos secciones principales. La primera involucra los datos básicos del biomarcador, que consta de un nombre obligatorio y una descripción opcional, junto a una pequeña sección para poder indicar qué tipo de molécula se está queriendo agregar y poder buscar por identificador de la misma. Además, ofrece la posibilidad de insertar varios identificadores de moléculas al mismo tiempo para ahorrar el tiempo de buscar cada molécula individualmente. En caso de agregar varios identificadores a la vez, Multiomix hace uso de BioAPI y Modulector para verificar la validez de los identificadores de moléculas insertados, indicando que la posibilidad de que el identificador insertado por el usuario sea erróneo (es decir, que no se encuentra en ninguna de las bases de datos incorporadas por alguna de las dos plataformas).

La segunda sección consta de cuatro paneles, uno por cada tipo de moléculas, para gestionar aquellas que el usuario agregó. Los colores indican la validez de cada moléculas: verde significa que el identificador fue encontrado en las bases de datos de BioAPI o Modulector, rojo significa que no fue encontrado (se permite al usuario avanzar de todos modos con el proceso de creación del biomarcador), y amarillo indica que hay varios alias que corresponden al identificador insertado (un gen, por ejemplo, puede cambiar de nomenclatura y ser referenciado de más de una manera).

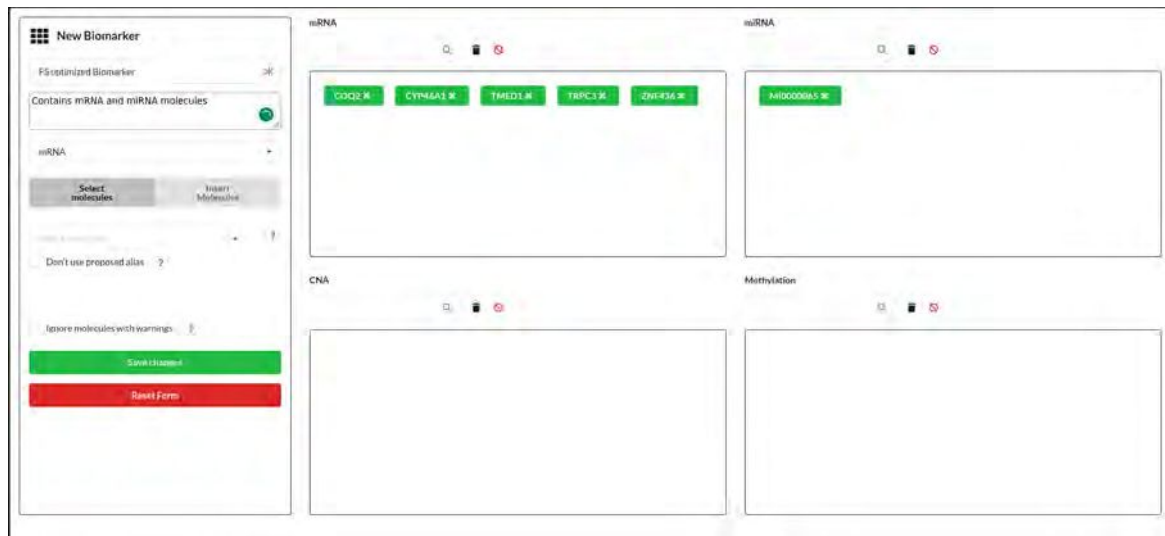


Figura B.12 Panel de creación manual de un biomarcador.

Creación a partir de Feature Selection

El segundo método de creación de un biomarcador consiste en reducir la cantidad de moléculas de un biomarcador existente que obtenga el mismo o mejor poder pronóstico/predictivo utilizando datos clínicos y de expresión molecular.

El proceso comienza con la selección de un biomarcador existente para realizar FS sobre el mismo (Figura B.13). La plataforma solo mostrará aquellos biomarcadores que contienen moléculas y se encuentran en un estado válido para ser optimizado (es decir, que no se encuentra en estado inconsistente producto de algún proceso de FS que haya finalizado con errores).



Figura B.13 Panel de selección del biomarcador a optimizar durante el proceso de FS.

Luego, el usuario debe seleccionar los datasets (Figura B.14) que serán utilizados para realizar un proceso de CV. Se debe seleccionar un dataset por cada tipo de molécula que

contenga el biomarcador seleccionado en el punto anterior. Es decir, si se seleccionó un biomarcador que contiene moléculas de mRNA y miRNA, entonces deberán seleccionarse tres datasets en total: uno para mRNA, otro para miRNA y uno de datos clínicos con las columnas de tiempo y evento. Este último dataset es indispensable para realizar análisis de supervivencia durante el proceso de FS a fin de conocer el poder pronóstico/predictivo del conjunto de moléculas que se esté evaluando. Los conjuntos de datos seleccionables pueden ser locales, subidos por el usuario previamente o datasets incorporados desde cBioPortal.



Figura B.14 Panel de selección de datasets a utilizar durante el proceso de FS.

El último paso consiste en seleccionar el método de FS a aplicar. Multiomix ofrece un amplio abanico de opciones para este último paso del procedimiento: el usuario puede optar por realizar FS a través de una búsqueda ciega (siempre y cuando la cantidad de moléculas a evaluar se encuentre por debajo de un umbral establecido), a partir del filtro de coeficientes obtenidos por cada molécula utilizando una regresión de Cox, o a través del uso de metaheurísticas.

Para todos los casos disponibles, se ofrece un panel con opciones básicas y avanzadas. En el caso de la búsqueda ciega o las metaheurísticas se debe seleccionar el modelo a ejecutar: Clustering (Figura B.15), SSVM (Figura B.16) o Random Survival Forest (Figura B.17). Para cada uno se permite seleccionar sus parámetros particulares.

En el caso de la regresión de Cox solo se permite seleccionar el top N de moléculas que se quieren conservar. Conservando los N elementos con mayor coeficiente obtenido por el método estadístico.

The screenshot shows the 'Step 3: Feature selection' configuration panel. At the top, three progress indicators are visible: 'Step 1: Selected Con muchos' (checked), 'Step 2: Datasets' (checked), and 'Step 3: Feature selection' (active). The main configuration area includes:

- Algorithms:** BBVA (with an 'Expert mode' button).
- Fitness function:** Clustering.
- Algorithms:** K-Means.
- Number of stars:** 60.
- Number of iterations:** 10.
- Original:** (with a dropdown arrow).
- Make:** (with a dropdown arrow).
- Scoring method:** C-index, Log Likelihood.
- Number of clusters:** 2.

At the bottom right, there are 'Cancel' and 'Confirm' buttons.

Figura B.15 Panel de creación de un biomarcador a partir de FS utilizando Clustering como modelo.

The screenshot shows the 'Step 3: Feature selection' configuration panel. At the top, three progress indicators are visible: 'Step 1: Selected Con muchos' (checked), 'Step 2: Datasets' (checked), and 'Step 3: Feature selection' (active). The main configuration area includes:

- Algorithms:** BBVA (with an 'Expert mode' button).
- Fitness function:** SVM.
- Kernel:** Linear.
- Max iterations:** 1000.
- Random state:** 2.
- Number of stars:** 60.
- Number of iterations:** 10.
- Original:** (with a dropdown arrow).

At the bottom right, there are 'Cancel' and 'Confirm' buttons.

Figura B.16 Panel de creación de un biomarcador a partir de FS utilizando SSVM como modelo.

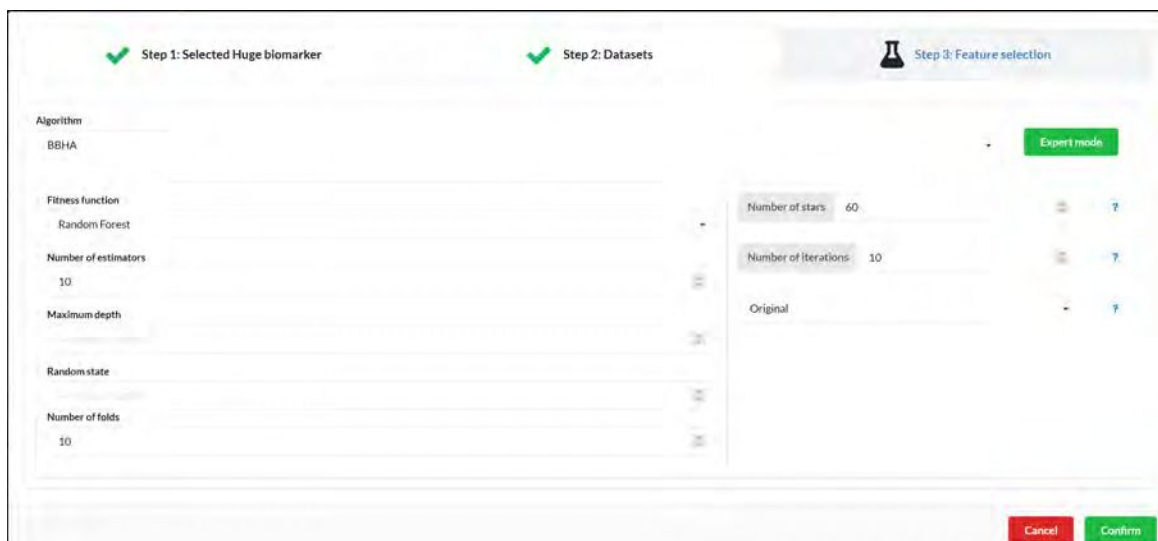
The screenshot shows a three-step process for biomarker creation. Step 1, 'Selected Huge biomarker', and Step 2, 'Datasets', are both completed, indicated by green checkmarks. Step 3, 'Feature selection', is the active step, marked with a flask icon. The interface is divided into two main sections. The left section, titled 'Algorithm', shows 'BBHA' selected. Below it, 'Fitness function' is set to 'Random Forest'. Other parameters include 'Number of estimators' (10), 'Maximum depth' (10), 'Random state' (empty), and 'Number of folds' (10). The right section, titled 'Expert mode', contains three input fields: 'Number of stars' (60), 'Number of iterations' (10), and 'Original' (empty). Each field has a help icon. At the bottom right, there are 'Cancel' and 'Confirm' buttons.

Figura B.17 Panel de creación de un biomarcador a partir de FS utilizando Random Survival Forest como modelo.

En el caso de haber seleccionado las metaheurísticas, el usuario podrá optar por utilizar BBH, BPSO o GA. También se ofrece la posibilidad de configurar los parámetros particulares de cada uno de los métodos:

- BBH: número de iteraciones, número de estrellas y la versión del algoritmo a correr (ambos fueron introducidos en la Sección 3.4.1).
- BPSO: número de iteraciones y número de partículas.
- GA: número de iteraciones, tamaño de población y tasa de mutación.

Una vez completados los tres pasos, Multiomix comenzará a ejecutar el proceso de FS de manera asincrónica enviando la tarea a una cola de ejecución del framework Celery que avisará al usuario en tiempo real cuando finalice.

B.4. Detalles del biomarcador

Una vez creado el biomarcador, independientemente del método utilizado para hacerlo, se puede realizar varias acciones con dicha entidad. Multiomix ofrece un panel con varias opciones útiles que permiten evaluar el biomarcador, desde consultar los detalles de las moléculas seleccionadas, hasta entrenar modelos de predicción de supervivencia para realizar inferencia sobre nuevos datos de pacientes.

B.4.1. Detalles de las moléculas del biomarcador

La primera pestaña del panel de detalles de un biomarcador consiste en una tabla con todas las moléculas que lo conforman (Figura B.18). A partir de ella, se pueden consultar datos específicos de cada una de las moléculas y dicha información es obtenida bajo demanda a través de BioAPI (en caso de haber seleccionado una molécula de mRNA o CNA, ya que consisten en genes) o Modulector (en caso de que sea una molécula de miRNA o un sitio de metilación).

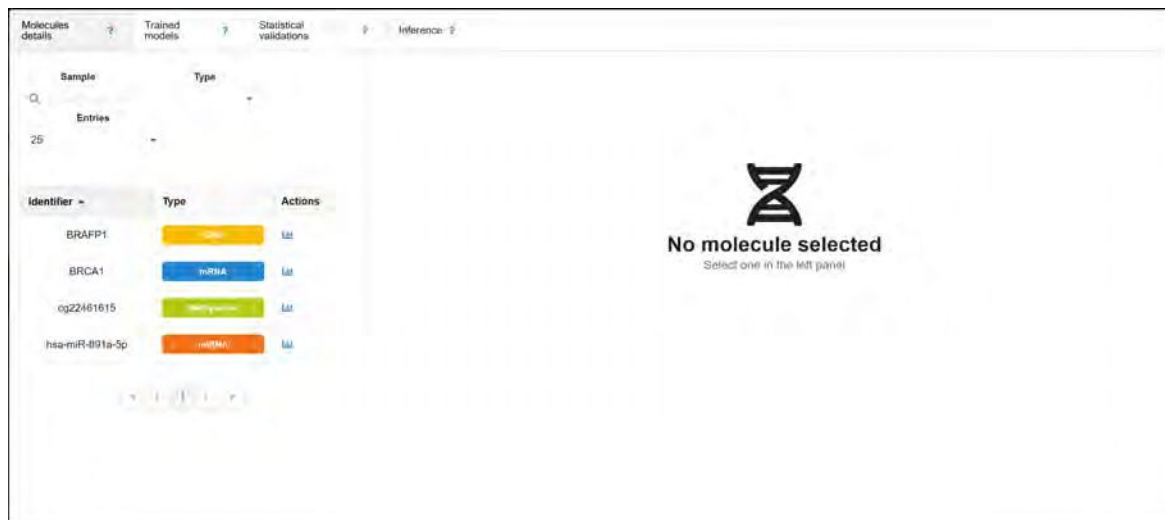


Figura B.18 Panel de información de las moléculas que conforman un biomarcador.

En el caso de seleccionar una molécula de mRNA o CNA, se muestra una serie de pestañas internas con diferentes tipos de información. La primera consiste en un panel de detalles básicos del gen seleccionado (Figura B.19) donde se muestra la posición y cromosoma del mismo, una descripción de las funciones asociadas y alias de diferentes bases de datos de referencia. Además, debajo se ofrece un listado de varias fuentes conocidas de información como PubMed, GeneCards [127][113], MalaCards [110], PathCards [5], Reactome [92][53], entre muchas otras.

The screenshot displays a web application interface for gene analysis. At the top, there are navigation tabs: 'Molecules details', 'Trained models', 'Statistical validations', and 'Inference'. Below these, a search bar is visible. A table on the left lists entries with columns for 'Identifier', 'Type', and 'Actions'. The entries include BRAFP1, BRCA1, cg22461615, and hsa-miR-691a-5p. The BRCA1 entry is highlighted. To the right, a detailed view for 'BRCA1 DNA repair associated' is shown. This view includes a 'Summary' section with a detailed description of the gene's function and its role in maintaining genomic stability. Below the summary, there are links to various databases: PubMed, Google, GeneCards, MetaCards, and PathCards. The interface is clean and professional, with a light blue and white color scheme.

Figura B.19 Panel con información de un gen en particular.

B.4.2. Redes de asociaciones de genes

Siguiendo al panel de detalles de moléculas se puede encontrar uno que muestra las asociaciones que el gen seleccionado tiene con otros genes (Figura B.20). Dichas asociaciones se pueden clasificar en:

- Evidencia de fusión proteica: puntuación que se deriva de proteínas fusionadas en otras especies.
- Evidencia de coocurrencia génica: es un tipo de puntuación que se deriva de patrones similares de ausencia/presencia de genes en diferentes especies.
- Evidencia experimental: es una puntuación que representa el nivel de confianza de una interacción de proteínas basada en datos experimentales.
- Evidencia de minería de textos: es una puntuación derivada de la coocurrencia de nombres de genes/proteínas en publicaciones científicas.
- Evidencia de bases de datos curadas: puntuación derivada de datos curados de varias bases de datos.
- Evidencia de coexpresión: es una medida del grado de coexpresión de dos genes basada en sus patrones similares de expresión de mRNA medidos por matrices de ADN y tecnologías similares.

Cada asociación tiene un peso que muestra qué tan fuerte es dicha asociación y permitirá obtener información sobre la misma. Estos y otros datos mostrados en este panel se detallan en el apartado del servicio de *STRING* de la Sección B.8.4.

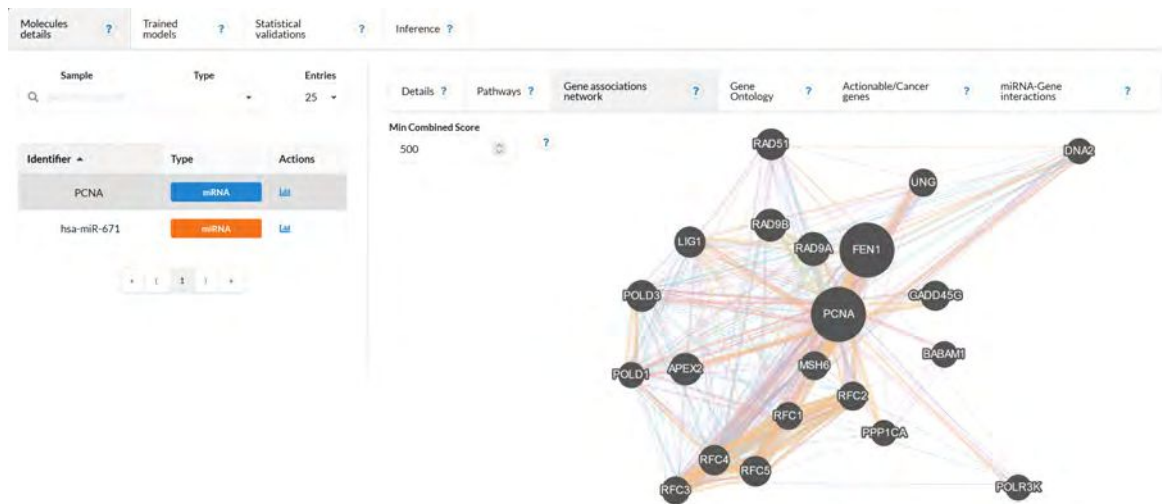


Figura B.20 Panel con gráfico de asociaciones de genes con respecto al gen seleccionado.

B.4.3. Información de Gene Ontology

Gene Ontology (GO) es un proyecto de bioinformática que tiene como objetivo desarrollar una representación computacional de nuestro conocimiento en constante evolución sobre cómo los genes codifican funciones biológicas a nivel molecular, celular y de tejidos (Figura B.21). Este recurso es fundamental para el análisis computacional de experimentos de biología molecular y genética a gran escala en la investigación biomédica.

El GO se basa en una ontología, que es una estructura jerárquica de términos definidos que representan las propiedades de los productos génicos. Esta ontología abarca tres dominios principales:

- **Componente celular:** describe las partes de una célula o su entorno extra-celular. Por ejemplo, los orgánulos celulares, la membrana plasmática y el citoplasma.
- **Función molecular:** representa las actividades específicas que realizan los productos génicos. Incluye funciones como la catálisis de reacciones químicas, la unión a moléculas y la transcripción del ADN.
- **Proceso biológico:** describe las secuencias de eventos que ocurren en una célula o un organismo. Estos procesos pueden incluir la división celular, la respuesta inmune y el desarrollo embrionario.

Las anotaciones funcionales de GO permiten priorizar genes y proteínas involucrados en procesos clave para el desarrollo y progresión del cáncer, como la proliferación celular descontrolada, la evasión de la apoptosis o la angiogénesis, representando blancos terapéuticos potenciales. Los perfiles de expresión génica asociados con determinados procesos biológicos anotados en GO pueden utilizarse como biomarcadores para predecir el pronóstico y la respuesta al tratamiento en diferentes tipos de cáncer, donde la expresión diferencial de genes involucrados en procesos como la invasión tumoral, la metástasis o la resistencia a fármacos puede tener valor pronóstico. GO también permite realizar análisis de enriquecimiento funcional, identificando procesos biológicos, funciones moleculares o componentes celulares sobre-representados en conjuntos de genes diferencialmente expresados en cáncer, revelando mecanismos biológicos subyacentes al desarrollo y progresión del cáncer, lo que puede conducir al descubrimiento de nuevos biomarcadores y blancos terapéuticos.

En el panel de Multiomix que abarca toda la información de GO se muestra, en primer lugar, los términos involucrados con el gen seleccionado (Figura B.22), en segundo lugar, diversos filtros para que el usuario pueda conservar aquellos términos biológicos dependiendo del tipo de ontología, del tipo de relación que conlleven, y un último filtro que ofrece la posibilidad de quedarse con aquellos términos por:

- Intersección: conserva solo los términos relacionados con todos los genes seleccionados.
- Unión: conserva solo los términos relacionados con al menos uno de los genes seleccionados.
- Gene Enrichment: conserva los términos de ontología que la herramienta PANTHER [131] considere significativos para los genes involucrados.

Una vez seleccionado el término de interés se podrá acceder a un gráfico con sus componentes (Figura B.23). En este panel se encuentran algunos mecanismos que permiten controlar la profundidad del gráfico, ocultar algunos componentes y filtrar por tipo de mecanismo biológico.

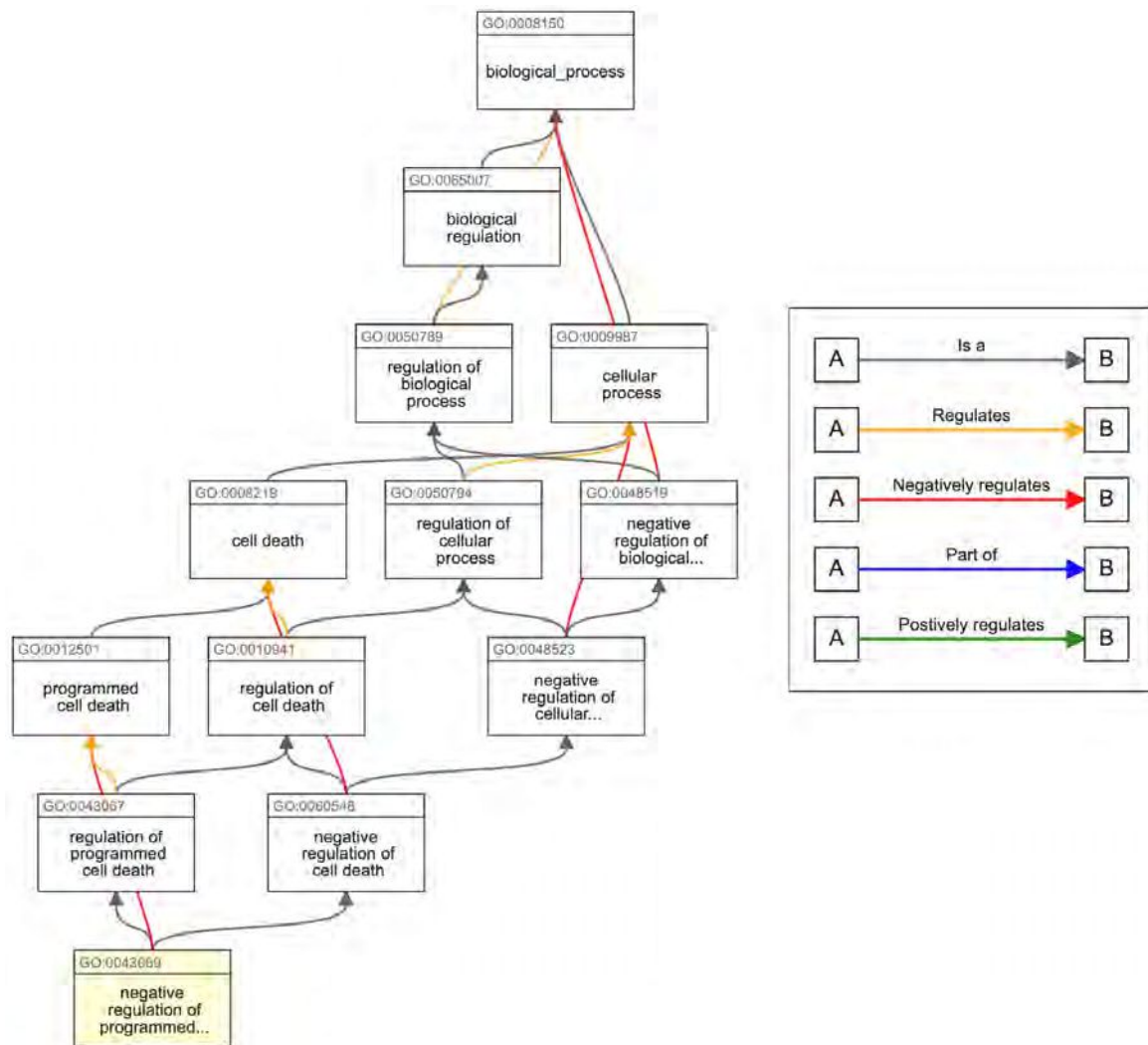


Figura B.21 Un ejemplo de GO, donde los procesos biológicos más simples (término en la sección inferior de la imagen) son relacionados con procesos cada vez más abarcativos, hasta llega al componente más general (término en la sección superior de la imagen).



Figura B.22 Panel con el listado de los términos biológicos involucrados con el o los genes seleccionados.

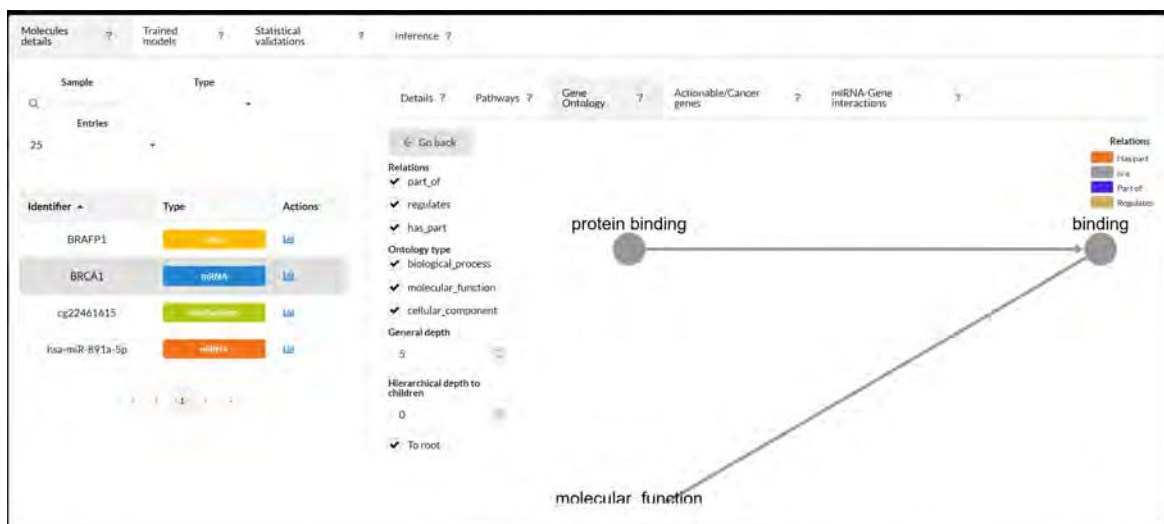


Figura B.23 Gráfico con la ontología completa seleccionada en el panel previo.

Información de miRNAs

En el caso de haber seleccionado una molécula de miRNA en lugar de un gen, aparecerá para seleccionar un panel con la información de la misma (Figura B.24). Estos detalles son los mismos que se muestran en los paneles introducidos en la Sección 4.1 y consta de los alias del miRNA, junto con su secuencia de aminoácidos.

The screenshot shows the Multiomix interface with a sidebar on the left containing a table of identifiers and their types. The main content area displays the details for a selected miRNA.

Identifier	Type	Actions
BRAF1	CpG	Li
BRCA1	miRNA	Li
cg22461615	Methylation	Li
hsa-miR-891a-5p	miRNA	Li

Main panel details:

MIMAT0004902 / hsa-miR-891a-5p / hsa-miR-891a-5p

UGCAACGAACCUAGCCACUGA

Figura B.24 Panel con información de un miRNA en particular.

Información de sitios de metilación

En caso de haber seleccionado un sitio de metilación, también se mostrará la información sobre el mismo en un panel dedicado a este tipo de moléculas (Figura B.25). En él se pueden observar los alias, la posición cromosómica, una lista de islas relacionadas con el sitio de metilación según la base de datos UCSC, y un listado de genes relacionados y las regiones donde se encuentra el sitio de metilación en cuestión.

The screenshot shows the Multiomix interface with a sidebar on the left containing a table of identifiers and their types. The main content area displays the details for a selected CpG site.

Identifier	Type	Actions
BRAF1	CpG	Li
BRCA1	miRNA	Li
cg22461615	Methylation	Li
hsa-miR-891a-5p	miRNA	Li

Main panel details:

Methylation Information

cg22461615 / cg22461615_TC11

Chr. Position: chr4:82900764 [+]

UCSC CpG Islands

CpG Island	Relation
chr4:82900535-82900912	Island

Related genes

Gene	Regions
THAP9	5'UTR / exon_1
THAP9-AS1	exon_1
SEC31A	TSS200

Figura B.25 Panel con información de un sitio CpG en particular.

B.4.4. Información adicional

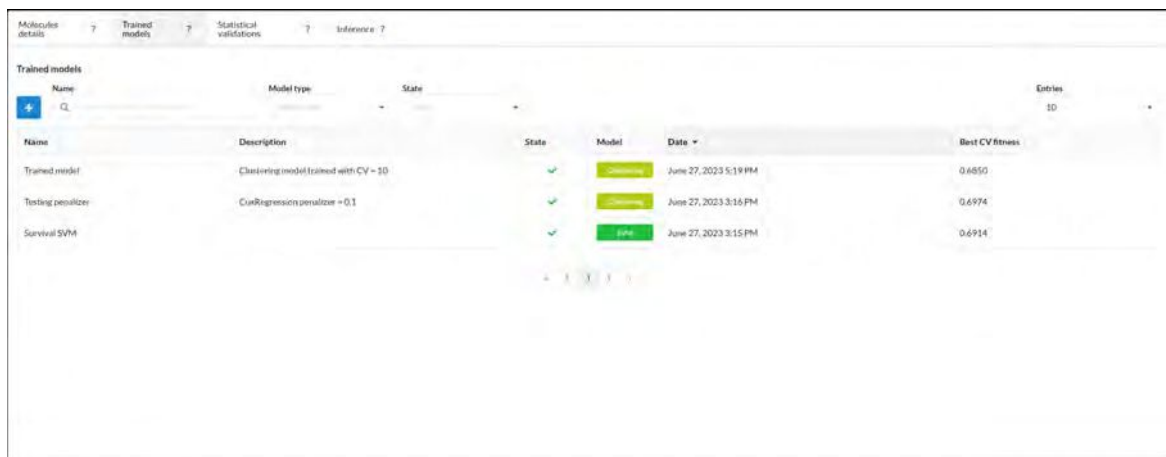
Similar a la página de análisis de correlación (Sección 4.1), en este panel de detalles de moléculas, cuando se selecciona algún gen o molécula miRNA se muestra el panel de interacción miRNA-Gen con el puntaje e información adicional que proporciona mirDIP.

En el caso de que la molécula en estudio sea un miRNA, serán accesibles los panel de asociaciones de dicho miRNA con patologías y drogas reportadas. También ofrecidos desde la página de análisis de correlación.

Los tres paneles de interacción miRNA-Gen, miRNA-Patología y miRNA-Droga fueron descritos en detalle en la Sección 4.1.

B.5. Modelos entrenados

Los modelos entrenados con las moléculas de un biomarcador seleccionado se listan en una tabla (Figura B.26) con un nombre, una descripción, el estado actual, qué tipo de modelo es, la fecha en la que se realizó el guardado del mismo, y la mejor métrica obtenida en el proceso de CV durante el entrenamiento.



Name	Description	State	Model	Date	Best CV fitness
Trained model	Clustering model trained with CV = 10	✓	Clustering	June 27, 2023 5:19 PM	0.6850
Testing penalizer	CsvRegression penalizer = 0.1	✓	Clustering	June 27, 2023 3:16 PM	0.6974
Survival SVM		✓	SVM	June 27, 2023 3:15 PM	0.6914

Figura B.26 Panel de modelos entrenados, con los datos básicos y métricas obtenidas durante su entrenamiento y evaluación.

Desde este panel también es posible entrenar nuevos modelos para utilizar en la posteridad. El formulario (Figura B.27) consiste en un nombre obligatorio para identificar el modelo, y una descripción opcional. Luego, se debe elegir el modelo a entrenar, las opciones son las mismas que se encuentran durante la configuración de experimentos de FS: Clustering, SSVM y RSF. Para cada uno se ofrece un amplio abanico de parámetros. Por último, se

permite seleccionar la cantidad de pliegues del proceso de CV, proceso que se realiza para el entrenamiento, independientemente del modelo a entrenar.

El segundo paso consiste en la selección de datasets con los que se realizará el entrenamiento y evaluación. Al igual que el formulario de FS, los datasets a seleccionar son los clínicos, y uno por cada tipo de molécula que conforma el biomarcador.

En el caso de que se presenten errores, se informará con mensajes claros sobre cómo podrían solventarse (por ejemplo, podría ocurrir que la cantidad de grupos en el modelo de Clustering sea mayor a la cantidad de grupos que genera la validación estratificada, indicándole al usuario que puede volver a entrenar el modelo, pero configurando una menor cantidad de grupos o utilizando otro conjunto de datos más diversificado).

The screenshot shows a web form titled "Create new trained model". It is organized into three main sections:

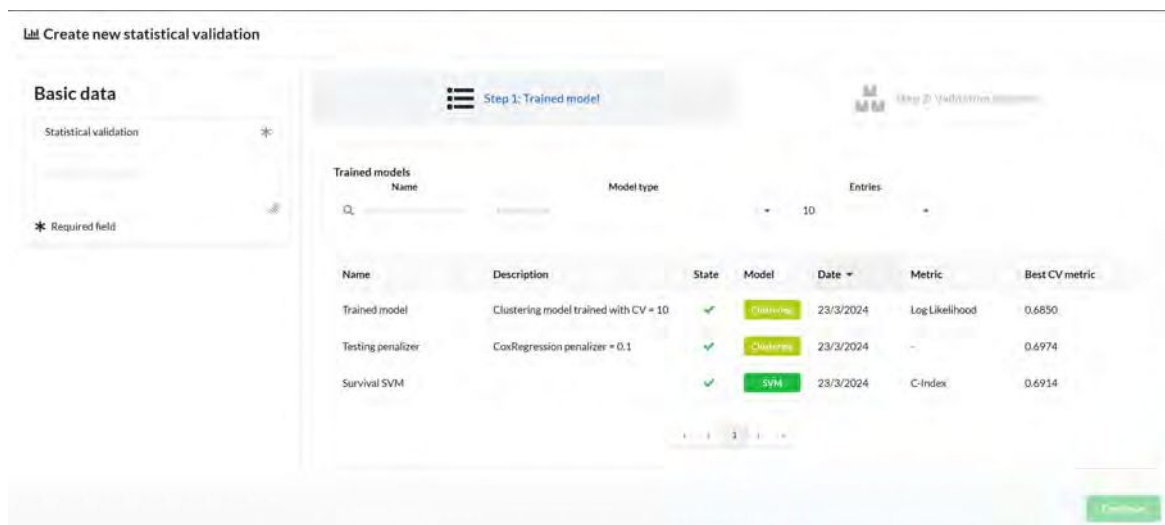
- Basic data:** Contains a "Trained model" input field with an asterisk indicating it is a required field.
- Step 1: Training parameters:**
 - Select a new model to train:** A dropdown menu currently showing "Clustering".
 - Select model parameters:** A list of parameters including "Algorithm" (K-Means), a checked checkbox for "Search for the optimal number of clusters", "Metric" (Cox-Regression), "Scoring method" (Log likelihood), "Random state", and "Penalizer".
- Step 2: Training datasets:** Contains "Select Cross Validation parameters" with a "Number of folds" input field set to "10".

At the bottom right of the form, there are "Cancel" and "Continue" buttons.

Figura B.27 Formulario de alta de un nuevo modelo de ML.

B.6. Validaciones estadísticas

Multiomix ofrece un formulario (Figura B.28) que permite seleccionar un modelo previamente entrenado (que se encuentre en estado completado), y datasets clínicos y de expresión (al igual que las otras funciones ya descritas, se debe seleccionar un dataset por cada tipo de molécula en el biomarcador) para poder realizar diferentes evaluaciones.



Basic data

Statistical validation

* Required field

Step 1: Trained model

Step 2: Validation

Name	Description	State	Model	Date	Metric	Best CV metric
Trained model	Clustering model trained with CV = 10	✓	Clustering	23/3/2024	Log Likelihood	0.6850
Testing penalizer	CoxRegression penalizer = 0.1	✓	Clustering	23/3/2024	-	0.6974
Survival SVM		✓	SVM	23/3/2024	C-Index	0.6914

Continue

Figura B.28 Formulario de alta de una nueva validación estadística.

B.6.1. Features más significativos

La primera funcionalidad una vez que ha finalizado el proceso de validación estadística, es el panel de features más significativos, donde se muestran un ranking con el coeficiente obtenido por el método de regresión de Cox en cuanto a la correlación con los datos de supervivencia (Figura B.29).

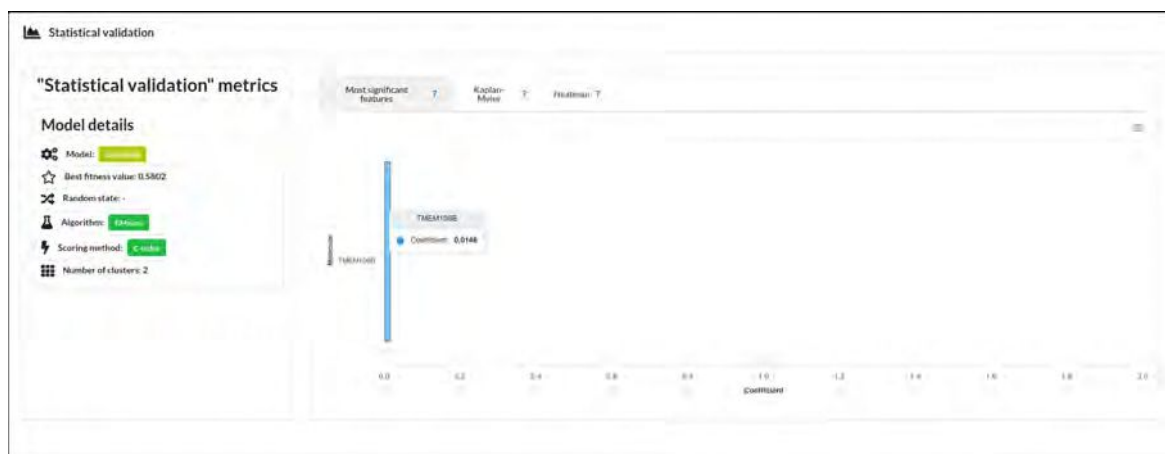


Figura B.29 Gráfico con las moléculas más significativas y su coeficiente según la regresión de Cox.

Conocer los features (variables predictoras) con los coeficientes más altos en un modelo de regresión de Cox es crucial en el análisis de datos de supervivencia, ya que indican cuales tienen un mayor impacto en la ocurrencia del evento.

B.6.2. Curvas Kaplan-Meier

El segundo panel disponible es el de supervivencia, donde se muestran curvas Kaplan-Meier. Como se explicó en la Sección 2.4.2, estas curvas permiten visualizar y comparar la probabilidad de supervivencia a lo largo del tiempo para diferentes grupos de pacientes. En este panel, se permite graficar múltiples curvas correspondientes a diferentes maneras de agrupar a los pacientes:

1. A partir de las predicciones del modelo de Clustering: si la validación estadística fue creada utilizando un modelo de Clustering, Multiomix permite realizar inferencia con el mismo para generar diferentes grupos de pacientes dependiendo de la expresión de las moléculas del biomarcador. Por ejemplo, si el modelo tiene un $K = 2$ entonces en el gráfico habrá 2 curvas, una por cada grupo generado por el modelo (Figura B.30).
2. Por atributo clínico: la plataforma mostrará una lista de atributos clínicos disponibles para seleccionar en el dataset asociado. Por ejemplo, si selecciona una columna "Es fumador", habrá 2 curvas en el gráfico (aquellos pacientes que son fumadores, y aquellos que no), si se agrupa por tipo de tumor, etnia u otro atributo multiclase, se graficarán tantas curvas como valores diferentes haya en esa columna del dataset clínico.

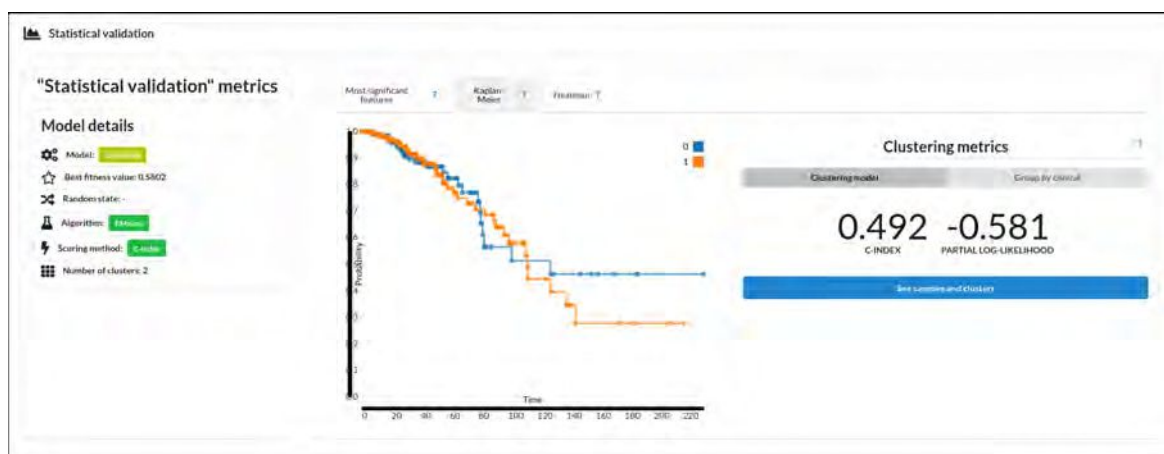


Figura B.30 Panel de graficación de curvas Kaplan-Meier. En este caso los grupos graficados son los inferidos por el modelo de Clustering seleccionado durante la creación de la validación estadística.

En el caso de que la validación estadística haya sido creada utilizando un modelo como SSVM o RSF, la función de agrupamiento por expresión no estará disponible y solo se podrá agrupar por atributo clínico. Independientemente del modelo de ML y el modo de

agrupamiento, Multiomix mostrará las métricas *C-Index* y *Partial Log-Likelihood* para poder realizar una comparación objetiva de la probabilidad de supervivencia de los grupos involucrados. Además, se permite ver una tabla con funciones de búsqueda y ordenamiento, los pacientes que pertenecen a cada uno de estos grupos.

B.6.3. Heatmap

El último panel corresponde a un mapa de calor (o Heatmap). Este es una representación gráfica de datos donde los valores individuales se codifican con colores. Típicamente, se utiliza una paleta de colores secuencial que va desde tonos claros hasta tonos oscuros para representar valores bajos y altos respectivamente. Esta técnica de visualización es especialmente útil para analizar y detectar patrones en grandes conjuntos de datos complejos.

Multiomix permite visualizar de manera intuitiva los niveles de expresión de múltiples moléculas para cada uno de los pacientes del dataset clínico en estudio. Cada fila del Heatmap representa una molécula específica, mientras que cada columna corresponde a un paciente. Los colores codifican los niveles de expresión (Figura B.31), permitiendo identificar fácilmente patrones de expresión similares o diferentes entre los pacientes. Esto puede revelar subgrupos o clústeres de individuos con perfiles de expresión moleculares similares, lo que puede tener implicaciones clínicas o biológicas relevantes. Además, el Heatmap destaca las moléculas que se expresan de manera significativamente diferente entre los pacientes, convirtiéndolas en candidatas prometedoras para investigar su relevancia en la enfermedad o respuesta al tratamiento.



Figura B.31 Heatmap con datos de expresión de las moléculas del biomarcador. En verde los valores más bajos, en color negro aquellos valores más altos.

B.7. Inferencia

El formulario de creación de un experimento de inferencia (Figura B.32) consiste en la selección de un modelo entrenado (el proceso de creación de un modelo entrenado fue detallado en la Sección 4.2.1), el cual definirá qué tipo de inferencia se está haciendo. En el caso de un algoritmo de Clustering, se realizará una tarea de naturaleza predictiva, ya que el modelo establece a qué grupo pertenece cada uno a partir de la información de expresión de las moléculas del biomarcador. Si el modelo seleccionado es un SSVM o RSF, la tarea será de pronóstico, ya que ambos modelos predicen el tiempo de ocurrencia de un evento de interés.

The screenshot shows a web interface for creating a new inference experiment. It is titled "Create new inference experiment". On the left, there is a "Basic data" section with a dropdown menu for "Inference" and a "Required field" label. The main area is divided into two steps: "Step 1: Trained model" and "Step 2: molecules datasets". Under "Step 1: Trained model", there is a table of trained models. The table has columns for Name, Description, State, Model, Date, and Best CV fitness. The models listed are:

Name	Description	State	Model	Date	Best CV fitness
Clustering (on 2)		✓	Substrate	June 27, 2023 7:13 PM	0.5802
Trained model	Clustering model trained with CV = 10	✓	Substrate	June 27, 2023 5:19 PM	0.6050
Testing penalizer	CoxRegression penalizer = 0.1	✓	Substrate	June 27, 2023 3:16 PM	0.6974
SurvivalSVM		✓	SVM	June 27, 2023 3:15 PM	0.6914

At the bottom right of the form, there is a green "Continue" button.

Figura B.32 Formulario de creación de un experimento de inferencia. En esta sección se debe seleccionar un modelo previamente entrenado de ML para realizar las predicciones.

El segundo paso, similar a los procesos de FS, entrenamientos de modelos, y validaciones estadísticas descriptos anteriormente; consta de la selección de los datos sobre los que efectuará la inferencia el modelo seleccionado. La única diferencia es que en este panel no se permite seleccionar un dataset clínico, ya que no se dispone de esa información. Por lo que basta con seleccionar un dataset por cada tipo de molécula que conforma un biomarcador para lanzar el proceso completo.

Figura B.33 Formulario del segundo paso en la creación de un experimento de inferencia. En los experimentos de inferencia no se debe seleccionar los datos clínicos ya que tanto el tiempo como ocurrencia del evento serán predichos por el modelo de ML.

Una vez que le proceso termine, se informará al usuario del estado del experimento de inferencia. En caso de que haya finalizado sin errores, el usuario podrá ver, a través de una tabla paginada con filtros y búsqueda, los valores predichos por el modelo para cada uno de los pacientes.

En el caso de los algoritmos de Clustering, el valor predicho es el grupo en el cual pertenece cada paciente dependiendo de sus niveles de expresión (Figura B.34). En el caso de SSVM o RSF, el resultado es un tiempo relativo en el que el modelo predice la ocurrencia del evento.

Sample	Cluster
TCGA-E2-A1BD	1
TCGA-E2-A1BC	1
TCGA-E2-A1B6	0
TCGA-E2-A1B5	1
TCGA-E2-A1B4	0
TCGA-E2-A1B1	1
TCGA-E2-A1B0	1
TCGA-E2-A1AZ	1
TCGA-E2-A1ST	0
TCGA-E2-A1S5	0
TCGA-E2-A1SR	0
TCGA-E2-A1SP	0
TCGA-E2-A1S0	1
TCGA-E2-A1SM	1
TCGA-E2-A1SL	1

Figura B.34 Listado de pacientes junto al grupo predicho por el modelo de Clustering seleccionado para hacer la inferencia.

Con el fin de hacer más amena la interpretación de resultados, se permite establecer etiquetas para los grupos, asignando un nombre y un color para poder realizar la distinción de los diferentes grupos de pacientes (Figura B.35).



Figura B.35 Listado de pacientes junto al grupo predicho por el modelo de Clustering seleccionado para hacer la inferencia. En este caso los diferentes grupos se muestran con un nombre y color seleccionados por el usuario para hacer la distinción más visible.

B.8. Abstracción en la obtención de datos

Numerosas funciones de Multiomix se nutren de información externa sobre diferentes tipos de moléculas. Para obtener dicha información de manera eficiente y estructurada hace uso de dos plataformas llamadas Moduletor y BioAPI. Cada una de ellas posee sus propias tecnologías, servicios y bases de datos integradas.

B.8.1. Moduletor: bases de datos

mirDIP

mirDIP (versión 5.2) [122][133] integra datos de predicción gen-miRNA con un puntaje unificado que califica la asociación, permitiendo comparar distintas asociaciones entre genes y estos reguladores. Como ocurre en las herramientas bioinformáticas en general, esta información no debe ser interpretada como determinante, pero es una gran guía para el investigador que la utiliza.

Además del puntaje, mirDIP provee un listado de fuentes y publicaciones científicas de Pubmed (repositorio que comprende más de 36 millones de citas de literatura biomédica de diferentes revistas y libros).

miRBase

miRBase (versión 22.1) [50][51][52][29][60] aporta secuencias y anotaciones de miRNA publicadas. Cada entrada de esta base de datos representa una porción de horquilla prevista de un transcrito de miRNA (denominada horquilla en la base de datos), con información sobre la ubicación y la secuencia del miRNA maduro (denominado maduro).

HMDD

Human microRNA Disease Database (HMDD) [83] es una base de datos que recopila evidencias experimentales que respaldan las asociaciones entre los miRNAs y las enfermedades. La información que aporta HMDD es crucial ya que los miRNA son una clase importante de RNA reguladores que reprimen la expresión génica a nivel postranscripcional, jugando un papel importante en varios procesos biológicos críticos y un amplio espectro de enfermedades.

HMDD fue la primera base de datos de enfermedades relacionadas con miRNA en el mundo, creada en diciembre de 2007. Y hasta entonces se ha mantenido actualizada a partir de diferentes versiones:

- La versión 1.0 de HMDD contenía el nombre del miRNA, el nombre de la enfermedad, el ID de PubMed de referencia y la evidencia que respaldaba la asociación miRNA-enfermedad.
- La versión 2.0 [80] presentó anotaciones más detalladas y completas, incluyendo datos de genética, epigenética, miRNA circulantes e interacciones miRNA-diana.
- La versión 3.0 [61] contenía un 200,2% más de entradas de asociaciones miRNA-enfermedad que la versión 2.0, con una clasificación de evidencias más específica.
- La versión 4.0 [32] es la más reciente y contiene 53530 entradas respaldadas experimentalmente, lo que significa un aumento de 1,5 veces en el número de asociaciones miRNA-enfermedad en comparación con la versión 3.0. Se agregaron categorías de miRNA exosomales y codificados por virus. También se integró el análisis de redes de enfermedades.

Modulector utiliza la versión 4.0, por lo que cuenta con la última información disponible.

SM2miR

Small Molecule - miRNA Association (SM2miR) [81] es una base de datos curada manualmente que recopila y incorpora los efectos validados experimentalmente de pequeñas moléculas (o fármacos) sobre la expresión de miRNA en 21 especies a partir de artículos publicados. Cada entrada contiene información detallada sobre pequeñas moléculas, miRNAs y sus relaciones, incluyendo especies (en Moduletor solo se importan las entradas correspondientes a la especie Homo Sapiens), nombre de la pequeña molécula, número de acceso a DrugBank, CID de PubChem, aprobado o no por la FDA, nombre del miRNA, número de acceso a miRBase, patrón de expresión del miRNA, método de detección experimental, tejidos o condiciones para la detección, evidencias en la referencia, ID de PubMed y año de publicación de la referencia.

La versión integrada no se define por un número, si no por la fecha en la que se realizó la actualización de los datos. En el caso de Moduletor, la versión corresponde a la actualización del 27 de abril del año 2015, que es la última versión publicada.

Illumina Infinium MethylationEPIC

Para la información de metilación, Moduletor incorpora los datos de Illumina Infinium MethylationEPIC 2.0 array ¹ que es una herramienta de secuenciación de metilación de todo el genoma que se dirige a más de 935000 sitios CpG en las regiones biológicamente más significativas del metiloma humano.

B.8.2. Moduletor: servicios

Los servicios que pone a disposición Moduletor son los siguientes:

MiRNA target interactions

Devuelve asociaciones entre genes y miRNA. Este servicio permite identificar regulaciones de expresión génica por parte de una molécula de miRNA. Esta información está compuesta de un score que genera miRDIP para dicha asociación y una lista de enlaces a las publicaciones científicas que la avalan. Los valores devueltos son los siguientes:

- **id:** identificador de registro en MirDIP.

¹<https://www.illumina.com/products/by-type/microarray-kits/infinium-methylation-epic.html>

- **mirna**: identificador de miRNA (miRBase MIMAT id o ID anterior). El recibido como parámetro de consulta.
- **gene**: gen objetivo.
- **score**: puntaje de interacción (según mirDIP). Rango de valor entre 0 y 1.
- **source_name**: base de datos de la cual se extrajo la interacción. Por ahora siempre se recibirá el valor *mirDIP*.
- **pubmeds**: array de URLs de PubMed para la interacción miRNA-gen (según mirTaR-Base).
- **sources**: fuentes de interacción miRNA-Gen. El puntaje de mirDIP se basa en los puntajes de esas fuentes. Este campo es un array que contiene los nombres de las fuentes de puntaje de interacción. Las diferentes bases de datos fuente se pueden encontrar en el sitio oficial de miRDIP.
- **score_class**: clase de puntaje según mirDIP. Los valores posibles son: *V* (Muy alto: Top 1%), *H* (Alto: Top 5%), *M* (Medio: Top 1/3) o *L* (Bajo: Fondo 2/3).

MiRNA details

Devuelve detalles de miRNA como la secuencia de nucleótidos, el identificador MIMAT (identificador estándar) y una lista de enlaces a fuentes externas que pueden aportar más información sobre el miRNA en cuestión. Los valores devueltos son los siguientes:

- **aliases**: array de alias de miRNA (IDs anteriores según miRBase).
- **mirna_sequence**: secuencia de nucleótidos de miRNA.
- **mirbase_accession_id**: identificador miRNA (MIMAT) según la base de datos miR-Base.
- **links**: lista de JSON que contiene la siguiente información:
 - **source**: nombre de la base de datos donde se puede encontrar información relacionada con el miRNA. Para esta versión siempre se recibirá el valor *mirbase*.
 - **url**: enlace para acceder a la base de datos fuente para el miRNA de interés.

MiRNA aliases

Devuelve el identificador de miRNA clásico y el identificador MIMAT (actualmente el estándar). La breve respuesta consta de los siguientes valores:

- **mirbase_accession_id**: identificador de miRBase (MIMAT) para el miRNA.
- **mature_mirna**: identificador de miRNA maduro en la base de datos miRBase.

MiRNA codes finder

Dado un criterio de búsqueda, devuelve todos los miRNAs que coinciden con él. Este servicio es de crucial importancia para poder ofrecer funcionalidad de autocompletado en Multiomix a la hora de estar trabajando con biomarcadores. La respuesta consta de un arreglo en JSON con los miRNAs que coinciden con el criterio de búsqueda.

miRNA codes

Retorna un listado de identificadores de miRNA y devuelve el identificador de acceso aprobado según miRbase DB. En caso de que el identificado no sea válido se devuelve un valor **null**.

Methylation sites finder

Dado un criterio de búsqueda, devuelve todos los sitios de metilación que coinciden con él. También se utiliza para autocompletar las moléculas que conforman un biomarcador en varias funciones de Multiomix. La respuesta consta de un arreglo en JSON con los sitios de metilación que coinciden con el criterio de búsqueda.

Methylation sites

Busca una lista de nombres o ID de sitios de metilación de diferentes versiones de arrays de Illumina para un identificador pasado por parámetro. En caso de que el identificado no sea válido se devuelve un valor **null**.

Genes of methylation sites

Busca a partir de una lista sitios de metilación (que pueden ser de diferentes versiones de arrays Illumina) y devuelve el gen o genes a los que pertenecen.

Methylation site details

Ofrece información sobre un sitio de metilación específico. Devuelve lista de otros nombres para el mismo sitio de metilación en otros arrays de Illumina (como los modelos EPIC v2, EPIC v1, Methyl450 y Methyl27); información sobre el cromosoma, la posición y la cadena en la que se encuentra el sitio; lista de islas relacionadas con el sitio de metilación según la base de datos UCSC [109]; y un listado de genes relacionados. Los valores específicos son los siguientes:

- **name:** nombre del sitio de metilación según la matriz Illumina Infinium MethylationEPIC 2.0.
- **aliases:** lista de otros nombres para el mismo sitio de metilación en otras matrices de Illumina (EPIC v2, EPIC v1, Methyl450 y Methyl27).
- **chromosome_position:** cadena con información sobre el cromosoma, posición y cadena en la que se ubica el sitio. Formato: *chr:posición [cadena]*
- **ucsc_cpg_islands:** lista de islas relacionadas con el sitio de metilación según la base de datos UCSC. Cada elemento en la vista es un JSON con el siguiente contenido:
 - **cpg_island:** coordenadas cromosómicas donde se ubica la isla. Formato: *chr:posición inicial-posición final*
 - **relation:** relación del sitio con la isla CpG. Los valores que puede tomar son *Island*=dentro de los límites de una isla CpG, *N_Shore*=0-2kb 5' de la isla, *N_Shelf*=2kb-4kb 5' de la isla, *S_Shore*=0-2kb 3' de la isla, *S_Shelf*=2kb-4kb 3' de la isla.
- **genes:** el valor es un JSON donde cada clave es un gen que está relacionado con el sitio de metilación. Los valores para cada gen son una lista que contiene la región del gen donde se ubica el sitio de metilación. Estas regiones, según la base de datos NCBI RefSeq, pueden ser: *5UTR*=región 5' no traducida entre el TSS y el codón de inicio ATG, *3UTR*=región 3' no traducida entre el codón de parada y la señal de poli A, *exon_#*, *TSS200*=1-200 pb 5' del TSS, o *TS1500*=200-1500 pb 5' del TSS. TSS=*Sitio de Inicio de la Transcripción*.

Diseases

Devuelve un listado de enfermedades que fueron reportadas en la literatura científica como relacionadas con un miRNA específico, junto con los enlaces a dichas publicaciones. Los valores devueltos son los siguientes:

- **id**: identificador interno del registro en la base de datos HMDD.
- **category**: códigos de categoría asignados por la base de datos HMDD para clasificar enfermedades. Los códigos posibles se pueden encontrar en la documentación de HMDD ².
- **disease**: nombre de la enfermedad asociada con el miRNA utilizado como parámetro.
- **pubmed**: enlace al artículo científico en la base de datos Pubmed donde se encuentra la evidencia que relaciona el miRNA con la enfermedad.
- **description**: breve descripción de por qué este miRNA está relacionado con esta enfermedad.

Drugs

Al igual que el servicio de *Diseases*, devuelve un listado de drogas que regulan la expresión del miRNA en cuestión. Este servicio ofrece información extra como: si la droga fue aprobada por la FDA, el método de detección utilizado, el tejido en el cual fue medida la interacción de la droga, patrón de expresión (es decir, si la droga sobre-expresa o sub-expresa al miRNA), y anotaciones de los autores de las publicaciones sobre la experimentación realizada. Los valores devueltos son los siguientes:

- **id**: identificador interno del registro en la base de datos SM2miR.
- **small_molecule**: nombre de la pequeña molécula (o fármaco).
- **fda_approved**: indica con un booleano si la pequeña molécula o fármaco está aprobado por la FDA.
- **detection_method**: método de detección experimental. Los diferentes métodos pueden ser: *Northern blot*, *Luciferase reporter assay*, *Illumina HiSeq2000*, *TaqMan low-density array*, *Microarray*, *Northern blot*, *MiRNA PCR array*, *Quantitative real-time PCR* o *Microarray*.
- **condition**: tejidos o condiciones para la detección.
- **pubmed**: enlace al artículo científico en la base de datos Pubmed donde se encuentra la evidencia que relaciona el miRNA con la pequeña molécula.

²<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10767894/table/tb11/?report=objectonly>

- **reference:** título del artículo científico donde se encuentra la evidencia que relaciona el miRNA con la pequeña molécula.
- **expression_pattern:** patrón de expresión del miRNA. Los diferentes métodos pueden ser: *up-regulated* o *down-regulated*.
- **support:** breve texto con información de respaldo para esta relación fármaco-miRNA.

B.8.3. BioAPI: bases de datos

Nomenclatura de Genes (Comité de Nomenclatura de Genes de HUGO - HGNC)

HGNC [115] se erige como el recurso autoritario para la nomenclatura estandarizada de genes humanos. Los datos de HGNC, adquiridos en septiembre de 2022, constituyen un componente fundamental dentro de BioAPI, proporcionando información esencial de identificación y nomenclatura de genes.

ENSEMBL

Aprovechando la herramienta de minería de datos BioMart, BioAPI extrae conjuntos de datos relacionados con genes de ENSEMBL [91], un navegador de genomas prominente que atiende a los genomas de vertebrados. Los datos de ENSEMBL, obtenidos en septiembre de 2022, enriquecen BioAPI con anotaciones integrales, facilitando la investigación en diversas áreas como la genómica comparativa, la evolución y la regulación transcripcional.

RefSeq

Mediante el paquete R GeneSummary, BioAPI obtiene resúmenes concisos de genes humanos de la base de datos RefSeq [97]. RefSeq, un repositorio mantenido por el Centro Nacional de Información Biotecnológica (NCBI), ofrece secuencias nucleicas y proteicas meticulosamente curadas. La adquisición de datos de RefSeq, basada en la versión 214, mejora la utilidad de BioAPI al proporcionar resúmenes concisos de genes, facilitando diversos emprendimientos analíticos.

CiVIC

BioAPI aumenta su repertorio con descripciones de genes orientadas a la interpretación clínica en el cáncer, obtenidas de la base de datos CiVIC [49, 76]. Como una plataforma de código abierto que respalda la curación de variantes de cáncer, CiVIC proporciona a BioAPI conocimientos especializados cruciales para la investigación oncológica. La incorporación de

datos de CiVIC, recuperados en abril de 2023, subraya el compromiso de BioAPI de facilitar avances en la terapéutica y el diagnóstico del cáncer.

Vías metabólicas (ConsensusPathDB-human)

BioAPI incorpora datos de vías metabólicas obtenidos de ConsensusPathDB-human [67, 66], una amalgama de diversas redes de interacción pertinentes a Homo sapiens. Esta integración, establecida en septiembre de 2022, enriquece BioAPI con una visión holística de las vías metabólicas, abarcando diversos mecanismos moleculares e interacciones regulatorias.

Expresión de Genes (Genotype-Tissue Expression - GTEx)

GTEx [82] se erige como una piedra angular dentro de BioAPI, ofreciendo conocimientos sobre patrones de expresión génica específicos de tejidos y mecanismos regulatorios. Con muestras recolectadas de numerosos sitios de tejidos no enfermos, GTEx, accedido en septiembre de 2022, proporciona un recurso rico para comprender la dinámica de la expresión génica en diversos contextos fisiológicos.

Terapias y Genes Accionables en el Cáncer (OncoKB)

OncoKB [26] sirve como un repositorio de conocimiento crucial para la oncología de precisión, ofreciendo información biológica y clínica integral sobre alteraciones genómicas en el cáncer. Al incorporar datos de OncoKB, descargados en noviembre de 2023, BioAPI potencia su capacidad para facilitar la toma de decisiones informada sobre terapias contra el cáncer, guiada por conocimientos basados en evidencia sobre alteraciones genéticas accionables.

Ontología de Genes (GO)

BioAPI integra datos del proyecto Gene Ontology [3, 2], enriqueciendo su funcionalidad con anotaciones estructuradas que delimitan funciones moleculares, procesos biológicos y componentes celulares asociados con genes y productos génicos. La inclusión de datos de GO, obtenidos en junio de 2023, mejora la interpretabilidad y la comprensión contextual de los datos biológicos dentro de BioAPI.

Fármacos Relacionados con el Cáncer (Pharmacogenomics Knowledge Base - PharmGKB)

PharmGKB [138, 139] sirve como un recurso valioso que aclara la interacción entre la variación genética humana y la respuesta a medicamentos, especialmente en el contexto del tratamiento del cáncer. Al integrar datos de PharmGKB, adquiridos en junio de 2023, BioAPI capacita a los investigadores con información sobre asociaciones gene-fármaco clínicamente accionables y relaciones genotipo-fenotipo, fomentando enfoques personalizados para la farmacoterapia del cáncer.

Red de Asociaciones Funcionales Predichas (STRING)

La base de datos STRING [129] contribuye a las capacidades de genómica funcional de BioAPI al proporcionar un repositorio integral de interacciones proteína-proteína conocidas y predichas. Esta integración, facilitada mediante predicciones computacionales y mecanismos de transferencia de conocimientos, permite a los usuarios de BioAPI explorar intrincadas redes de interacción de proteínas, elucidando relaciones funcionales entre entidades biológicas.

Farmacotranscriptómica (DrugBank)

DrugBank [142, 141, 74], una base de datos en línea integral que abarca información sobre fármacos y sus blancos, enriquece las capacidades farmacogenómicas de BioAPI. Al incorporar datos de Farmacotranscriptómica, BioAPI facilita la exploración de interacciones fármaco-blanco y conocimientos farmacogenómicos, cruciales para elucidar la base molecular de la respuesta a fármacos y los efectos adversos.

B.8.4. BioAPI: servicios

La integración de las bases de datos incorporadas en BioAPI hacen posible, en su conjunto, la siguiente serie de servicios:

Genes symbols validator

Busca el identificador de una lista de genes de diferentes bases de datos genómicas y devuelve los símbolos aprobados según la nomenclatura HGNC.

Genes symbols finder

Dado un criterio de búsqueda, devuelve todos los genes que coinciden con él. También se utiliza para autocompletar las moléculas que conforman un biomarcador en varias funciones de Multiomix.

Genes information

Obtiene información diferente para los genomas humanos de referencia GRCh38 y GRCh37 a partir de uno o más genes pasados por parámetro. La información de los mismos consta de los siguientes datos:

- **alias_symbol**: símbolos alternativos para un gen conocido
- **percentage_gene_gc_content**: proporción de nucleótidos de guanina y citosina en la secuencia de ADN del gen
- **oncokb_cancer_gene**: devuelve "Oncogene" o "Tumor Suppressor Gene" sólo si el gen tiene esta información en la base de datos OncoKB
- **name**: nombre del gen según la base de datos HGNC
- **band**: citobanda o localización específica en el genoma
- **chromosome**: cromosoma donde se localiza el gen
- **start_position**: posición cromosómica de inicio del gen para el genoma de referencia GRCh38
- **end_position**: posición cromosómica de los extremos del gen para el genoma de referencia GRCh38
- **start_GRCh37**: posición cromosómica del inicio del gen para el genoma de referencia GRCh37
- **end_GRCh37**: posición cromosómica de los extremos de los genes del genoma de referencia GRCh37
- **strand**: cadena de ADN que contiene la secuencia codificante del gen
- **gene_biotype**: tipo de gen, ya sea codificante de proteínas, no codificante, pseudo-gene, entre otros.

- **refseq_summary**: descripción completa del gen según la base de datos RefSeq (RefSeq : NCBI Reference Sequences)
- **civic_description**: descripción de la relevancia clínica del gen según la base de datos CIVIC (Clinical Interpretation of Variants in Cancer)
- **hgnc_id**: identificador del gen en la base de datos HGNC
- **uniprot_ids**: identificador del gen en la base de datos Uniprot
- **omim_id**: identificador de genes en la base de datos OMIM
- **ensembl_gene_id**: identificador de gen en la base de datos Ensembl
- **entrez_id**: identificador de gen en la base de datos NCBI Entrez

Gene Groups

Devuelve un listado de grupo de genes al que pertenece un gen específico según HGNC. Los valores devueltos son los siguientes:

- **gene_id**: símbolo de gen aprobado por el HGNC.
- **locus_group**: agrupa los tipos de locus en conjuntos relacionados. Las diferentes opciones para este campo pueden ser: *protein-coding gene*, *pseudogene*, *phenotype* o *other*.
- **locus_type**: especifica la clase genética de cada entrada de gen. Todos los tipos de locus y grupos de locus se pueden encontrar en la documentación del HGNC.
- **groups**:
 - **gene_group**: nombre del grupo de genes.
 - **gene_group_id**: identificador del grupo de genes.
 - **genes**: todos los demás genes para este grupo. Para una descripción de los grupos de genes y sus identificadores, puedes acceder al sitio web del Comité de Nomenclatura de Genes HUGO.

Genes of a metabolic pathway

Dada una fuente externa de información de pathways (pueden ser *kegg* [65], *biocarta* [96], *ehmn*, *humancyc* [112], *inoh* [144], *netpath* [68], *pid* [114], *reactome* [92], *smpdb* [64], *signalink* [31], *wikipathways* [1]), y un identificador de pathway, devuelve todos los genes que están involucrados en el mismo.

Metabolic pathways from different genes

Dada una lista de genes, devuelve una lista de pathways que los contienen a todos (junto con la base de datos que ofrecen dicha información). Por cada pathway pasado por parámetro, devuelve los siguientes valores:

- **source**: base de datos de la ruta metabólica encontrada. Los posibles valores se pueden encontrar en el servicio Genes de una ruta metabólica.
- **external_id**: identificador de la ruta metabólica en la fuente.
- **pathway**: nombre de la ruta metabólica.

Gene expression

Dada una lista de genes y un identificador de tejido de interés, devuelve una lista con valores de expresión para dicho tejido pero para pacientes *sanos*.

Therapies and actionable genes in cancer

Recupera información de terapias oncológicas de precisión aprobadas por la FDA, y genes y fármacos procesables obtenidos de la base de datos OncoKB, a nivel terapéutico, diagnóstico y pronóstico. Cabe mencionar, que entre la información que dispone este servicio, se puede encontrar un valor llamado *cancer_type* que corresponde a los tipos de cáncer introducidos en [77]. Por cada gen pasado por parámetro, devuelve los siguientes valores:

- **therapeutic**: evidencia del gen para terapéutica. El valor es una lista de elementos del tipo JSON, donde cada elemento es una evidencia diferente con la siguiente estructura:
 - **drugs**: fármaco terapéutico.
 - **level_of_evidence**: nivel de evidencia terapéutica. Los diferentes valores para los niveles de evidencia se pueden encontrar en la documentación de la base de datos OncoKB.

- **alterations**: alteraciones específicas del gen cancerígeno.
- **cancer_types**: tipo de cáncer. Los tipos de cáncer utilizan la nomenclatura Onco-Tree.
- **diagnostic**: evidencia del gen para diagnóstico (solo para neoplasias hematológicas). El valor es una lista de elementos del tipo JSON, donde cada elemento es una evidencia diferente con la siguiente estructura:
 - **level_of_evidence**: nivel de evidencia diagnóstica. Los diferentes valores para los niveles de evidencia se pueden encontrar en la documentación de la base de datos OncoKB.
 - **alterations**: alteraciones específicas del gen cancerígeno.
 - **cancer_types**: tipo de cáncer. Los tipos de cáncer utilizan la nomenclatura Onco-Tree.
- **prognostic**: evidencia del gen para pronóstico (solo para neoplasias hematológicas). El valor es una lista de elementos del tipo JSON, donde cada elemento es una evidencia diferente con la siguiente estructura:
 - **level_of_evidence**: nivel de evidencia pronóstica. Los diferentes valores para los niveles de evidencia se pueden encontrar en la documentación de la base de datos OncoKB.
 - **alterations**: alteraciones específicas del gen cancerígeno.
 - **cancer_types**: tipo de cáncer. Los tipos de cáncer utilizan la nomenclatura Onco-Tree.
- **oncokb_cancer_gene**: tipo de gen cancerígeno. *Oncogene y/o Tumor Suppressor Gene*.
- **refseq_transcript**: transcripción génica según la base de datos RefSeq.
- **sources**: lista de fuentes donde hay evidencia de la relación del gen con el cáncer. Estas pueden ser diferentes paneles de secuenciación, el Censo de Genes del Cáncer de Sanger³, o Vogelstein et al. (2013) [135].
- **precision_therapies**: terapias aprobadas por la FDA que se consideran terapias de oncología de precisión por OncoKB™. El valor es una lista de elementos del tipo

³<https://www.sanger.ac.uk/data/cancer-gene-census/>

JSON, donde cada elemento es una terapia de oncología de precisión diferente con la siguiente estructura:

- **precision_oncology_therapy**: un fármaco que es más efectivo en un subconjunto de pacientes definido molecularmente y para el cual se requiere un perfil molecular previo al tratamiento para una selección óptima de pacientes.
- **fda_first_approval**: año de la primera aprobación del fármaco por la FDA. El primer año en que el fármaco recibió la aprobación de la FDA en cualquier indicación, independientemente de si el biomarcador fue incluido en el fármaco de la FDA en ese momento.
- **drug_classification**: posibles clasificaciones son *first-in-class*, *mechanistically-distinct*, *follow-on*, o *resistance* basadas en [128]. Solo se clasifican los fármacos con un biomarcador especificado por la FDA que pueda ser detectado por un método de detección basado en ADN/NGS.
- **fda_recognized_biomarkers**: biomarcadores relacionados con la terapia según la FDA. Incluye biomarcadores patognomónicos y específicos de la indicación, que si bien no se enumeran específicamente en la sección Indicaciones y Uso de la etiqueta del fármaco de la FDA, son el objetivo de la terapia de oncología de precisión.
- **method_of_biomarker_detection**: método de detección de biomarcadores. Si hay un dispositivo de diagnóstico complementario aprobado o autorizado por la FDA para la identificación de biomarcadores, se enumera el método de detección asociado con este dispositivo; si el biomarcador puede ser detectado por un método de detección basado en ADN/NGS, este se enumera primero.

Gene Ontology terms related to a list of genes

Dado un listado de genes, devuelve los términos de GO relacionados. La información de cada uno de estos términos consiste en: el tipo de ontología, una descripción textual de lo que representa el término, referencia(s) a la fuente de información, relaciones con otros términos (para ofrecer la posibilidad de construir un gráfico ontológico a partir de los términos coincidentes con la búsqueda), y métricas de búsqueda por enriquecimiento (como p-valores corregidos para pruebas múltiples, el tamaño de la intersección entre los genes de la consulta y los términos anotados, cantidad de genes incluidos y anotados en la consulta, precisión y exhaustividad que indican la proporción de genes anotados funcionalmente en relación con el tamaño de la consulta y la capacidad de recuperación de la consulta de

genes funcionalmente anotados respectivamente). Este servicio ofrece la posibilidad de filtrar por intersección (quedan los términos en los que aparecen *todos* los genes pasados por parámetros), por unión (quedan los términos en los que aparecen *al menos uno* de los genes pasados por parámetros) o gene-enrichment (haciendo uso de gProfiler se filtran los resultados dejando únicamente los términos significativamente estadísticos). Los valores que devuelve son:

- **go_id**: identificador único.
- **name**: nombre del término legible por humanos.
- **ontology_type**: denota a cuál de las tres sub-ontologías (*biological_process*, *molecular_function*, *cellular_component*) pertenece el término.
- **definition**: una descripción textual de lo que representa el término, más referencia(s) a la fuente de la información.
- **synonyms**: palabras o frases alternativas estrechamente relacionadas en significado con el nombre del término, con indicación de la relación entre el nombre y el sinónimo dada por el alcance del sinónimo.
- **subset**: este campo se refiere a una categorización adicional de términos dentro de la ontología. Permite agrupar términos que comparten características o propiedades específicas en subconjuntos más pequeños y específicos.
- **is_a**: se refiere a una relación semántica entre términos dentro de la ontología. Indica que un término es un subtipo o subclase de otro término más general.
- **alt_id**: se refiere a identificadores alternativos o secundarios para un término específico en la ontología.
- **synonym**: los sinónimos son palabras o frases alternativas estrechamente relacionadas en significado con el nombre del término.
- **definition_reference**: este campo proporciona referencias bibliográficas o fuentes de las cuales se obtiene la definición del término en cuestión.
- **relations_to_genes**: lista de elementos de tipo JSON. Cada elemento corresponde a un gen y cómo está relacionado con el término.
 - **gene**: nombre del gen.

- **relation_type**: el tipo de relación entre el gen y el término GO. Pueden ser *enables*, *involved_in*, *part_of* o *located_in*. Cuando se está realizando un filtro por enriquecimiento, se recopilarán relaciones adicionales de la base de datos g:Profiler. Estas relaciones se mostrarán como "relación obtenida de gProfiler".
 - **evidence**: código de evidencia para indicar cómo se respalda la anotación a un término en particular.
- **enrichment_metrics**:
- **p_value**: valor p hipergeométrico después de la corrección por múltiples pruebas.
 - **intersection_size**: el número de genes en la consulta que están anotados al término correspondiente.
 - **effective_domain_size**: el número total de genes "en el universo" que se utiliza como uno de los cuatro parámetros para la función de probabilidad hipergeométrica de significancia estadística.
 - **query_size**: el número de genes que se incluyeron en la consulta.
 - **term_size**: el número de genes que están anotados al término.
 - **precision**: la proporción de genes en la lista de entrada que están anotados a la función. Definida como $intersection_size/query_size$.
 - **recall**: la proporción de genes anotados funcionalmente que recupera la consulta. Definida como $intersection_size/term_size$.

Gene Ontology terms related to another specific term

Similar al servicio anterior, pero en vez de filtrar a partir de un listado de genes, permiten obtener los términos a partir de una lista de términos, facilitando así la construcción de una ontología. Los valores devueltos son:

- **go_id**: identificador del término GO.
- **name**: nombre del término GO.
- **ontology_type**: denota a cuál de las tres sub-ontologías (*cellular_component*, *biological_process* o *molecular_function*) pertenece el término.
- **relations**: diccionario de relaciones. Las claves posibles dentro de este diccionario son *part_of*, *regulates* o *has_part*, y sus valores son listas de términos con identificadores de Ontología de Genes.

Cancer related drugs

Devuelve las drogas seleccionadas para un gen específico pasado por parámetro, esta información, similar al equivalente para moléculas de miRNA en Modulector, indica si dicha droga está aprobada por el FDA, en qué estadio está su aprobación, los químicos que la componen, variantes genéticas asociadas. El listado completo de valores devueltos es el siguiente:

- **pharmgkb_id**: identificador asignado a esta etiqueta de medicamento por PharmGKB.
- **name**: nombre asignado a la etiqueta por PharmGKB.
- **source**: la fuente que originalmente escribió la etiqueta. Las opciones válidas son *EMA*, *FDA*, *HCSC* o *PMDA*. Para una descripción detallada de cada valor, revise la documentación de PharmGKB.
- **biomarker_flag**: *On* si el medicamento en esta etiqueta aparece en la lista de Biomarcadores de la FDA; *Off (Formerly On)* si la etiqueta estuvo en la lista de Biomarcadores de la FDA en algún momento; *Off (Never On)* si la etiqueta nunca estuvo en la lista de Biomarcadores de la FDA (según el conocimiento de PharmGKB).
- **testing_level**: nivel de prueba PGx anotado por PharmGKB. Los valores posibles son: *Testing Required*, *Testing Recommended*, *Actionable PGx*, *Informative PGx* o *Criteria Not Met*. Para una descripción detallada de cada valor, revise la documentación de PharmGKB.
- **chemicals**: productos químicos relacionados.
- **genes**: lista de genes relacionados.
- **variants_haplotypes**: variantes y/o haplotipos relacionados.

Predicted functional associations network

Proporciona información sobre redes de interacción entre genes a partir de la base de datos STRING y un gen específico. Los valores devueltos son los siguientes:

- **gene_1**: gen 1 en la relación bidireccional.
- **gene_2**: gen 2 en la relación bidireccional.
- **neighborhood_transferred**: puntuación que refleja la fuerza de la evidencia de apoyo del vecindario realizado en otros organismos.

- **fusion**: puntuación que se deriva de proteínas fusionadas en otras especies.
- **cooccurrence**: es un tipo de puntuación que se deriva de patrones similares de ausencia/presencia de genes en diferentes especies.
- **homology**: es una medida del grado de homología entre los interactores en una interacción de proteínas.
- **coexpression**: es una medida del grado de coexpresión de dos genes basada en sus patrones similares de expresión de mRNA medidos por matrices de ADN y tecnologías similares.
- **coexpression_transferred**: es una medida del grado de coexpresión de dos genes basada en sus patrones similares de expresión de mRNA medidos por matrices de ADN y tecnologías similares, transferida de otras especies en función de la homología.
- **experiments**: es una puntuación que representa el nivel de confianza de una interacción de proteínas basada en evidencia experimental.
- **experiments_transferred**: puntuación que se calcula a partir de datos experimentales transferidos de otras especies en función de la homología.
- **database**: puntuación derivada de datos curados de varias bases de datos. Representa el nivel de confianza de las interacciones de proteínas basado en estos datos curados.
- **database_transferred**: puntuación derivada de datos curados de varias bases de datos, transferida de otras especies en función de la homología.
- **textmining**: es una puntuación derivada de la coocurrencia de nombres de genes/proteínas en publicaciones científicas.
- **textmining_transferred**: es una puntuación derivada de la coocurrencia de nombres de genes/proteínas en publicaciones científicas, transferida de otras especies en función de la homología.

Drugs that regulate a gene

Dado un gen pasado por parámetro, devuelve un simple enlace a la base de datos DrugBank con la información de el o los fármacos que sobre o sub expresan dicho gen.