PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

Inferencia de Interacciones Causales Génicas Usando Técnicas basadas en Manto de Markov

Tesis para obtener el título profesional de INGENIERO INFORMÁTICO

AUTOR

Sergio Andrés Del Río Cárdenas

ASESOR:

Dr. Edwin Rafael Villanueva Talavera

Lima, Octubre, 2019

Resumen

Conocer cómo interactúan los genes en las células es un objetivo importante en biología y medicina. Este conocimiento permitiría la creación de terapias celulares precisas para corregir disfunciones de los mecanismos moleculares detrás de condiciones patológicas como el cáncer [1,2]. El estudio de estas interacciones ha sido realizado tradicionalmente por medio de experimentos que involucran perturbaciones a los sistemas celulares, y con ello una alta demanda de tiempo y mano de obra. La premisa común para realizar estos costosos experimentos de intervención es que ellos permiten detectar relaciones de causalidad entre genes sin ambigüedad, a diferencia de realizar únicamente observaciones en los sistemas celulares que no permitirían distinguir de forma confiable relaciones causales de correlaciones estadísticas generadas indirectamente por mecanismos no observados. En redes génicas, es necesario distinguir entre una causa de un efecto y el efecto de una causa, ya que esto permitiría saber cómo funciona la regulación génica en las células.

No obstante, Maathius et al. [6] demostró que inferir relaciones causales en redes moleculares es posible usando datos de observaciones de los componentes del sistema (genes) y una metodología de análisis de datos. Estos trabajos generaron interés en el tema motivando diversos trabajos en consecuencia con el enfoque de estadística inferencial y causalidad. Sin embargo, las metodologías propuestas incorporan fuertes consideraciones en los modelos, como aciclicidad de las interacciones y gausianidad en los niveles de expresión de los genes, consideraciones que son biológicamente cuestionables, así como un elevado costo computacional para su procesamiento.

Es en dicho contexto donde el presente proyecto propone aplicar un enfoque basado en Aprendizaje Máquina (AM). Este campo estudia cómo generar modelos que aprendan a discriminar objetos o instancias en categorías o clases conocidas, con base a un conjunto

de instancias ya clasificadas (datos de entrenamiento). La idea de usar Aprendizaje Máquina en la detección de interacciones causales entre genes es aprender las diferencias mínimas que puedan existir dentro de las observaciones temporales de las expresiones de los genes que pueden caracterizar comportamientos causales entre genes.

Sin embargo, al aplicar Aprendizaje Máquina en problemas de alta dimensionalidad como el descrito, es común hallar un alto costo computacional para su ejecución, lo cual genera la necesidad de métodos de reducción de dimensionalidad. En el presente proyecto se propone investigar un enfoque basado en el concepto de Manto de Markov (MM), cuyos estimadores han probado ser teóricamente óptimos para la detección del conjunto de variables causalmente relevante respecto a una variable de interés.

Dedicatoria

A mi abuelo Oscar, por siempre impulsarme a más.

Agradecimientos

A mis padres, por nunca rendirse. A mi familia, por siempre apoyarme. A Diana, por siempre exigirme dar lo mejor. A las familias Puente Harada y Ramirez Robles, por recibirme con tanta calidez en sus hogares. A mi asesor, Edwin Villanueva, por su constante apoyo, guía y consejo durante el planteamiento, desarrollo y experimentación del proyecto que involucra el presente trabajo.

Tabla de contenido

Índice de Tablas

Índice de Figuras

Capítulo 1.Generalidades

1.1 Problemática

Conocer cómo interactúan los genes en las células es un objetivo importante en biología y medicina. Tener este conocimiento nos permitiría crear terapias celulares precisas para corregir las disfunciones de los mecanismos moleculares que están detrás de condiciones patológicas como el cáncer [1, 2]. El estudio de las interacciones génicas ha sido realizado tradicionalmente por medio de experimentos que involucran perturbaciones a los sistemas celulares, experimentos que involucran altas demandas de mano de obra y tiempo [1], [3]. La premisa común para realizar estos costosos experimentos de intervención es que ellos permiten detectar relaciones de causalidad entre genes sin ambigüedad, a diferencia de realizar únicamente observaciones en los sistemas celulares que no permitirían distinguir de forma confiable relaciones causales de correlaciones estadísticas generadas indirectamente por mecanismos no observados [4]. En redes génicas, es necesario distinguir entre una causa de un efecto y el efecto de una causa, ya que esto permitiría saber cómo funciona la regulación génica en las células.

Desafiando la aceptada práctica de realizar experimentos intervencionales para estudiar las interacciones génicas, Maathius et al [6] demostró que inferir relaciones causales en redes moleculares es posible usando datos de observaciones de los componentes del sistema (genes) y una metodología de análisis de datos. Esta metodología propuesta fue basada en combinaciones de inferencia estadística de causalidad e ingeniería inversa [7].

Estos trabajos generaron interés en el tema motivando diversos trabajos en consecuencia con el enfoque de estadística inferencial y causalidad [8]–[11]. Sin embargo, estas metodologías, a pesar de su fundamentación matemática, incorporan fuertes consideraciones en los modelos, como aciclicidad de las interacciones y gausianidad en los niveles de expresión de los genes, consideraciones que son biológicamente cuestionables. También, las implementaciones son complejas ya que

requieren ajustar un elevado número de parámetros de los modelos, implicando un elevado costo computacional [12].

En el presente proyecto de fin de carrera se plantea investigar un enfoque basado en una metodología de aprendizaje de máquina. Aprendizaje de maquina (AM) estudia cómo generar modelos que aprendan a discriminar objetos o instancias en categorías o clases conocidas, con base a un conjunto de instancias ya clasificadas (datos de entrenamiento) [26]. La idea de usar AM en la detección de interacciones causales entre genes es aprender las diferencias mínimas que puedan existir dentro de las observaciones temporales de las expresiones de los genes que pueden caracterizar comportamientos causales entre genes. Se espera que dichas diferencias sean detectables en los datos, ya que la expresión de los genes que interactúan causalmente es dirigida por mecanismos biológicos diferentes de los mecanismos que dirigen la expresión de genes sin interacción causal [13].

Una problemática frecuente al intentar aplicar AM en problemas de alta dimensión, como el del presente proyecto, es la elevada complejidad computacional. Es deseable, por lo tanto, buscar formas efectivas de reducir la dimensión para que las técnicas de AM sean aplicadas de forma efectiva. En esta línea, el presente proyecto propone investigar un enfoque basado en el concepto de Manto de Markov (MM) [15]. Las técnicas de inferencia del Manto de Markov han probado ser teóricamente óptimas para la estimación de variables relevantes en relación a la variable objetivo [15]. Ellas también han brindado resultados positivos en la construcción de predictores en el campo de la medicina molecular [16]. En el presente proyecto se pretende aplicar las técnicas más actuales de estimación del Manto de Markov con la finalidad de revelar las variables más relevantes para la predicción de interacciones causales entre genes.

1.2 Objetivos

1.2.1 Objetivo general

Diseñar, implementar y validar una metodología basada en el Manto de Markov para la predicción de interacciones génicas a partir de datos observacionales de expresión génica.

1.2.2 Objetivos específicos

1. Definir e implementar una metodología de muestreo de datos adecuada para las características del conjunto de datos a disposición sobre interacciones génicas y generar con ella una colección de muestras para entrenamiento y validación.
2. Implementar un método basado en la estimación del Manto de Markov, para obtener varios conjuntos de atributos discriminantes con las muestras generadas en O1.
3. Implementar un algoritmo genético adaptativo para la optimización de los Mantos de Markov generados en O2 a fin de encontrar atributos robustos para la predicción de interacciones génicas.
4. Definir y adaptar modelos de clasificación para los datos de interacciones génicas en posesión, filtrados por los conjuntos de atributos generados en O2 y O3.
5. Determinar el grado de precisión y esfuerzo computacional de la metodología desarrollada en relación a métodos de selección de atributos de referencia.

1.2.3 Resultados esperados

R 1. Técnica de muestreo de datos adecuado para los datos de interacción génica a disposición y el conjunto resultante de muestras de datos para el entrenamiento y validación de los métodos de selección de atributos y los modelos de clasificación.

R 2. Algoritmo de estimación del Manto de Markov diseñado e implementado para el conjunto de datos a analizar, selectores de atributos seleccionados y los

conjuntos de atributos relevantes o evaluados, resultantes de ejecutar los diversos algoritmos de selección de atributos propuestos.

R 3. Algoritmo genético para optimizar la inferencia del Manto de Markov.

R 4. Un conjunto de modelos predictivos definidos, que puedan recibir como entrada colecciones de muestras de datos filtrados en base a los resultados de los algoritmos de selección de atributos, obtenidos al aplicar R2 y R3.

R 5. Reporte de comparación de precisión y costo computacional, en cada conjunto de datos de R2 y R3 de cada combinación de método de selección de atributos y modelo de predicción implementado; con énfasis en las ventajas y desventajas del uso del Manto de Markov.

1.2.4 Mapeo de objetivos, resultados y verificación

Tabla 1: Tabla de Objetivo 1 (Elaboración propia)

Objetivo 1: Definir e implementar una metodología de muestreo de datos adecuada para las características del conjunto de datos a disposición sobre interacciones génicas y generar con ella una colección de muestras para entrenamiento y validación.		
Resultado	**Meta física**	**Medio de verificación**
Algoritmo adecuado para el preprocesamiento de los datos de interacción génica a disposición y el conjunto resultante de muestras de datos para el entrenamiento y validación de los métodos de selección de atributos y los modelos de clasificación.	Metodología	● Sustentación teórica respecto a los pasos dentro de la metodología. ● Comprobación de coherencia entre datos iniciales y datos resultantes.

Tabla 2: Tabla de Objetivo 2 (Elaboración propia)

Objetivo 2: Implementar un método basado en la estimación del Manto de Markov, para obtener varios conjuntos de atributos robustos con las muestras generadas en O1.		
Resultado	**Meta física**	**Medio de verificación**
Algoritmo de estimación del Manto de Markov diseñado e implementado para el conjunto de datos a analizar, selectores de atributos seleccionados y los conjuntos de atributos relevantes o evaluados, resultantes de ejecutar los diversos algoritmos de selección de atributos propuestos.	Algoritmo	● Sustentación del método diseñado en base a experimentación numérica. ● Análisis de consistencia entre conjuntos iniciales de datos, y los conjuntos de datos obtenidos al aplicar el algoritmo.

Tabla 3: Tabla de Objetivo 3 (Elaboración propia)

Objetivo 3: Implementar un algoritmo genético para optimizar los Mantos de Markov generados en O2 a fin de encontrar atributos robustos para la predicción de interacciones génicas.		
Resultado	**Meta física**	**Medio de verificación**
Algoritmo genético para optimizar la inferencia del Manto de Markov. La población inicial del algoritmo serán, utilizando como generadores de población, los métodos de estimación del Manto de Markov implementados en R2, que será aplicado sobre la colección de muestras de datos al aplicar R1 sobre el conjunto de datos inicial.	Algoritmo	● Sustentación teórica respecto a la lógica núcleo del algoritmo.

Tabla 4: Tabla de Objetivo 4 (Elaboración propia)

Objetivo 4: Definir y adaptar modelos de clasificación para los datos de interacciones génicas en posesión, filtrados por los conjuntos de atributos generados en O2 y O3.		
Resultado	**Meta física**	**Medio de verificación**
Un conjunto de modelos predictivos definidos, que puedan recibir como entrada colecciones de muestras de datos filtrados en base a los resultados de los algoritmos de selección de atributos, obtenidos al aplicar R2 y R3.	Algoritmos	● Sustentación práctica de cada algoritmo seleccionado con conjuntos de datos cuyos resultados sean conocidos de antemano.

Tabla 5: Tabla de Objetivo 5 (Elaboración propia)

Objetivo 5: Determinar el grado de precisión y esfuerzo computacional de la metodología desarrollada en relación a métodos de selección de atributos de referencia		
Resultado	**Meta física**	**Medio de verificación**
Reporte de comparación de precisión y costo computacional, en cada conjunto de datos de R2 y R3 de cada combinación de método de selección de atributos y modelo de predicción implementado; con énfasis en las ventajas y desventajas del uso del Manto de Markov.	Gráficos y Tablas Comparativas	● Los datos a comparar serán hallados a través de la realización de experimentación numérica.

1.3 Herramientas y Métodos

En esta sección, se describirán las distintas herramientas y metodologías a utilizar durante el desarrollo del presente proyecto de fin de carrera.

1.3.1 Herramientas

1.3.1.1 Servidor PUCP: Servidor provisto por la Universidad para la ejecución de rutinas de código de alta duración. Es la plataforma base del proyecto a realizar, por lo que se relacionaría con todos los resultados propuestos.

1.3.1.2 Jupyter Notebook: Editor de textos basado en el enfoque *cliente-servidor*. Permitirá la visualización y ejecución de código almacenado en el servidor provisto. Se relaciona con todos los resultados que comprenden procesamiento datos, implementación y evaluación de algoritmos: R2, R4, R6, R7.

1.3.1.3 Python 3.6: Lenguaje de programación con licencia gratuita, y gran adopción en el contexto de análisis de datos en Aprendizaje Máquina. Se utilizará en conjunto a distintas librerías de Aprendizaje Máquina para la implementación del método de selección de atributos y el modelo de predicción de interés. Se relaciona con todos los resultados que comprenden procesamiento datos, implementación y evaluación de algoritmos: R2, R4, R6, R7.

1.3.1.4 TensorFlow (Python): Librería para implementación de algoritmos de Aprendizaje Máquina utilizando los recursos que ofrece los procesadores gráficos (GPUs). Se diseñará una abstracción matricial del modelo de predicción de interés para lograr mejor aprovechamiento de esta librería. Se relaciona con todos los resultados que comprenden procesamiento datos, implementación y evaluación de algoritmos: R2, R4, R6, R7.

1.3.1.5 Scikit-Learn: Librería con amplia variedad de algoritmos de Aprendizaje Máquina ya implementados, así como herramientas para el preprocesamiento y gestión de datos. Se relaciona con todos los resultados que comprenden procesamiento datos, implementación y evaluación de algoritmos: R2, R4, R6, R7.

1.3.1.6 Pandas (Python): Librería para el procesamiento de conjuntos de gran cantidad de datos. Trabaja en sinergia con las demás librerías mencionadas para lograr un flujo de trabajo claro y simple, respecto al procesamiento y gestión de los

datos. Se relaciona con todos los resultados que comprenden procesamiento datos, implementación y evaluación de algoritmos: R2, R4, R6, R7.

1.3.1.7 Base de Datos IEEEX: Base de datos con gran cantidad de investigaciones referentes al campo de ciencias de la computación, entre otros. Apoyará los resultados R1, R3 y R5, en tanto estos requieren de investigación acerca de aplicaciones previas de algoritmos para el análisis de datos biológicos.

1.3.1.8 Base de Datos PubMed: Base de datos enfocada en investigaciones en el campo de la medicina. Su uso es de particular relevancia para los resultados R1, R3 y R5, en tanto estos requieren de investigación acerca de aplicaciones previas de algoritmos para el análisis de datos biológicos.

1.3.2 Métodos

1.3.2.1 Muestreo de datos: Es la división de un conjunto de datos en subconjuntos, a fin de facilitar la obtención y posterior análisis de los mismos, sin perder precisión en el análisis. Suele utilizarse este método para conjuntos con gran cantidad de datos. Debido a la naturaleza desbalanceada de los conjuntos de datos a disposición, es tentativa la aplicación de muestro aleatorio y muestreo balanceado de los datos, a fin de obtener un conjunto de atributos óptimo [37]. Se relaciona con R1 para la definición de una metodología adecuada para el análisis de los conjuntos de datos a disposición.

1.3.2.2 Experimentación numérica: Es la comparación científica entre dos conjuntos de datos bajo un mismo contexto, aplicando la prueba de hipótesis para lograr una comparación adecuada. Para este proyecto de fin de carrera, este método es vital para medir el rendimiento del modelo de interés respecto a otros modelos propuestos en investigaciones pasadas. Se relaciona con R7, siendo este el resultado de las comparaciones a realizar siguiendo este método.

1.3.2.3 Método de estimación del Manto de Markov MMPC: Este método se usará como base para el diseño del nuevo algoritmo de estimación del Manto de Markov, dada su robustez demostrada [23].

1.3.2.4 Algoritmo Genético propuesto: Para este proyecto de fin de carrera, se aplicará el uso de técnicas de inferencia del Manto de Markov dentro de un contexto evolutivo,a fin de inferir un conjunto robusto de variables pertenecientes al Manto de Markov asociado a la variable objetivo "*interacción causal"* evitando las desventajas que presentan las técnicas actuales de inferencia, siendo la más relevante su complejidad exponencial) [22].

1.4 Alcance y limitaciones

En esta sección, se describirá el alcance del presente proyecto de fin de carrera, así como las posibles limitaciones y riesgos a encontrar durante su desarrollo.

1.4.1 Alcance

El proyecto está fuertemente relacionado con las investigaciones acerca de datos biológicos, específicamente las investigaciones dedicadas al análisis de datos utilizando métodos de Aprendizaje Máquina. Se ha elegido este sector por la importancia en el avance del desarrollo de nuevas drogas y terapias génicas, así como la relevancia actual de este tipo de análisis sobre datos del genoma humano. Los datos que se pretende analizar son conjuntos de datos de expresión génica.

En particular, bancos de datos observacionales de expresión génica temporal de líneas celulares BL2 (Burkits linfoma), obtenidos en el Instituto de Genómica Funcional de la Universidad de Regensburg, Alemania.

La metodología del proyecto busca evaluar el impacto del uso del Manto de Markov para la selección de atributos, y la validez de una Red Bayesiana inferida para la predicción de nuevos datos de interacción entre pares de genes, en comparación a otros enfoques más populares dentro del contexto

biológico [24].

1.4.2 Limitaciones

Este proyecto de fin de carrera se encuentra limitado principalmente:

1. En cuanto a la calidad de los datos; determina la relevancia del modelo a diseñar para aplicaciones reales. Como los datos a disposición son observacionales, depende de la calidad de observación realizada por parte de los investigadores.
2. En cuanto a la cantidad de atributos originales de los datos iniciales; determina la cantidad de ruido existente inicialmente en los datos y la cantidad de ajustes que deberá aplicarse a cada modelo productivo para obtener salidas comparables de los mismos.
3. En cuanto a la disponibilidad de poder de procesamiento, en tanto más poder permitiría mayor experimentación con los datos disponibles.

1.4.3 Riesgos

Los riesgos asociados al proyecto son los relacionados a la disponibilidad de poder de procesamiento para las experimentaciones propuestas.

Tabla 6: Tabla de Riesgos (Elaboración propia)

Riesgo identificado	**Impacto en el proyecto**	**Medidas correctivas para mitigar**
Dificultad para la obtención de poder de procesamiento	Media	Utilizar los recursos ofrecidos por la Facultad de Informática para el desarrollo del proyecto.
Disponibilidad de plataforma de trabajo	Baja	Utilizar servicios en la nube para almacenar el código fuente y grabar “huellas” del progreso obtenido en cada paso de los algoritmos en ejecución.

1.5 Viabilidad

1.5.1 Viabilidad Técnica

Se cuenta con libre disponibilidad a las herramientas mencionadas previamente, así como con experiencia en el lenguaje a utilizar para las implementaciones, Python 3.6. Respecto al respaldo de información, se cuenta con acceso a un repositorio gratuito (GitHub), donde el código permanecerá bajo la licencia GNU, y a un servidor que el Grupo de Inteligencia Artificial de la Pontificia Universidad Católica del Perú ha puesto a disposición para los experimentos que se requieran realizar. También se cuenta con la disponibilidad de un gran conjunto de datos para el muestreo, aprendizaje y validación de las técnicas a implementar.

1.5.2 Viabilidad Temporal

Para la implementación de los métodos de estimación del Manto de Markov y Redes Bayesianas (tema núcleo de este proyecto de fin de carrera) se estima una duración de 3 meses. En conjunto con las demás tareas a realizar, se estima una duración total de 6 meses, reflejada en el Anexo 1.

1.5.3 Viabilidad Económica

Se posee libre acceso a las herramientas mencionadas previamente, lo cual exime a este proyecto de alguna limitación financiera. Además, el desarrollo de este proyecto de fin de carrera es parte del proyecto “Aplicación de técnicas de biclustering y comités de clasificadores en la predicción de interacciones causales gen-gen a partir de datos observacionales” liderado por el asesor de tesis y financiado por InnovatePeru bajo el convenio 334-INNOVATEPERU-BRI-2016. Tal proyecto contempla la adquisición de material bibliográfico relevante y hardware computacional para la implementación y validación de los desarrollos.

Capítulo 2. Marco Conceptual

2.1 Introducción

A continuación, se hace una breve descripción de los conceptos teóricos que se utilizarán en este proyecto de fin de carrera. Se mencionan definiciones sobre Genética, Aprendizaje Máquina y Complejidad Computacional. La finalidad de esta sección es ilustrar las bases conceptuales de los distintos elementos que componen las implementaciones y las comparaciones realizadas en este proyecto, la manera en que se relacionan entre sí y sus particularidades.

2.2 Conceptos genéticos

2.2.1 Gen

El gen es la unidad básica de información que determina qué proteína será producida en qué momento y en qué cantidad [17]. Los genes se encuentran distribuidos a lo largo de la cadena de ADN (ácido desoxirribonucleico) en los cromosomas. Los genes interactúan entre si para producir las diferentes proteínas necesarias para sustentar la vida. Estas en conjunto forman la red génica que se desea investigar (genoma).

2.2.2 Genoma

El genoma es el conjunto de genes que define las características de un ser viviente. Este concepto se refiere al contenido genético de un organismo [17].

2.2.3 Red Génica

Los genes interactúan (Figura 1) mutuamente unos con otros como se mencionó anteriormente; esto se logra a través de moléculas reguladoras (transcripts) que generan. Estas moléculas se unen de forma específica para regular los genes de manera que activa o reprimen su expresión [18].

2.3 Conceptos de Aprendizaje Máquina:

2.3.1 Conceptos y Modelos de Predicción en Aprendizaje Máquina

En esta sección se introducirá el método de interés de este proyecto de fin de carrera, así como métodos tentativos para realizar las comparaciones respectivas.

2.3.1.1 Red Bayesiana

Una red bayesiana para un set de variables aleatorias **V** es representado por el par *(G, θ)*; donde la estructura de G es la de un grafo acíclico dirigido (GAD), con nodos correspondientes a las variables aleatorias en ***V***. Los parámetros *θ* indican la distribución de probabilidad condicional de cada nodo dado sus padres. Si existe un camino dirigido entre X y Y , entonces X es un ancestro de Y y Y es un descendiente de X . Si dos nodos no adyacentes X y Y tienen un nodo hijo en común, entonces X y Y son esposos entre ellos [15].

Usualmente cada variable, representada como nodo, posee estados mutuamente exclusivos, siendo cada estado representado con una probabilidad; y estos nodos son relacionados entre sí de acuerdo al tipo de relación que existe entre las variables (directamente causal o correlación) [14].

2.3.1.2 Redes Neuronales

Una red neuronal se define como un procesador masivo distribuido en paralelo que tiene la propiedad natural de almacenar conocimiento empírico y disponerlo para su posterior uso, en cuanto este conocimiento es adquirido a través de información de experiencias previas respectivas al uso que se quiere brindar [31].

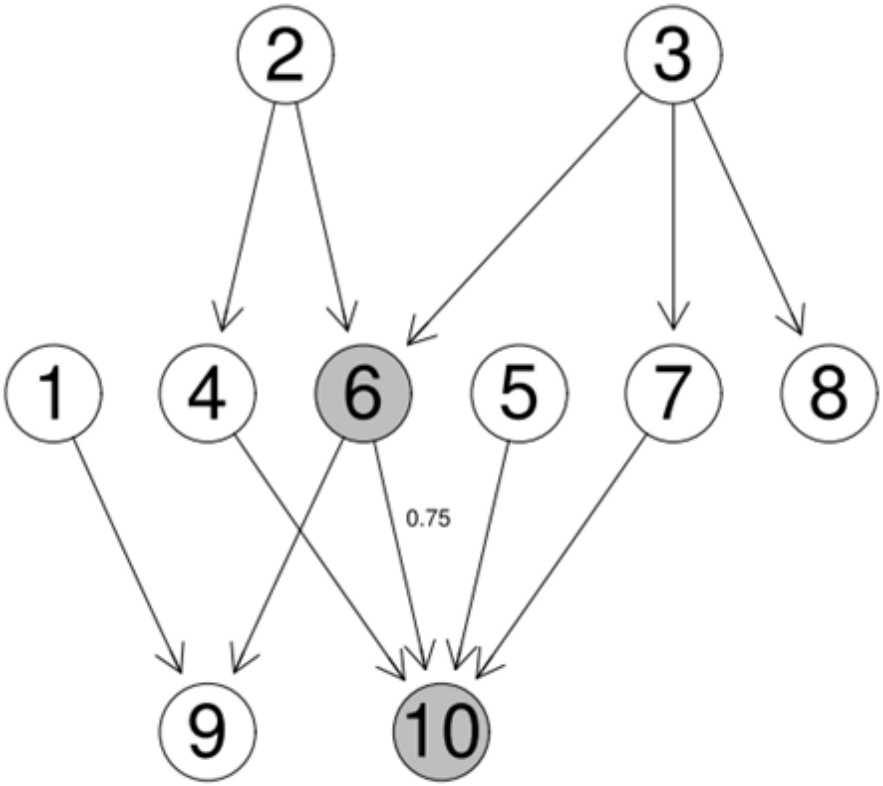

Figura 1. Ejemplo de cómo se comporta una red de interacción génica. Cada nodo representa un gen y cada arista una interacción directa [19]

Sus elementos básicos son:

Neurona: elemento básico de la red, contiene una función con la que procesa cada dato que se le ingresa.

Capa: Agrupa las neuronas, de manera que el resultado final también depende de la distribución, tipo y cantidad de neuronas a utilizar. Usualmente se puede encontrar una capa de entrada, una oculta y una de salida.

Para este proyecto, una opción tentativa para la comparación es un Perceptrón Multicapa (PMC), dado sus resultados prometedores en

aplicaciones parecidas [32]. El propósito general de este tipo de red neuronal es aproximar relaciones funcionales arbitrarias entre covariables y variables de respuesta [32].

La estructura interior de un PMC es un grafo dirigido y con pesos, cuyos vértices son llamadas neuronas y cuyos arcos (caminos) son llamados sinapsis [32]. Las neuronas están organizadas por capa, y cada capa está completamente conectada por medio de sinapsis con la siguiente capa [32]. La capa de entrada contiene todas las consideradas “covariables” y la capa de salida las respuestas positivas y negativas, para el caso de predicción de interacciones génicas.

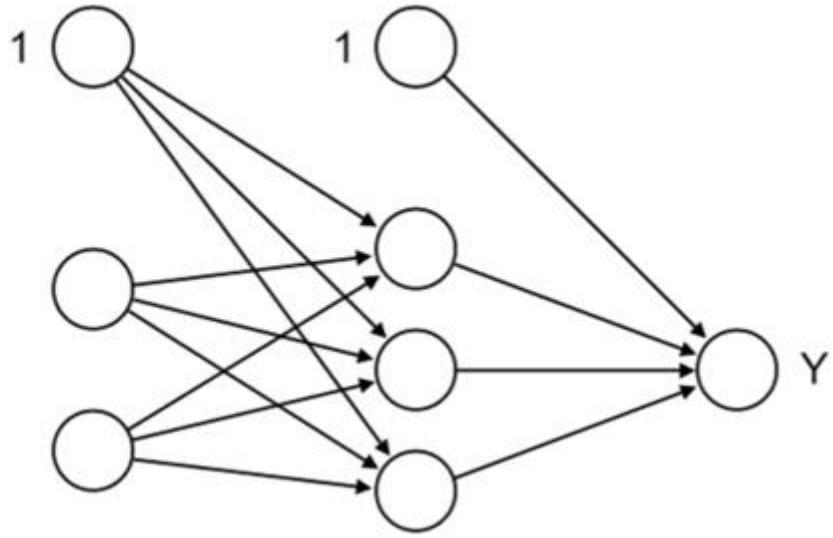

Figura 2. Ejemplo de una red neuronal. Cada nodo representa una neurona, cada flecha una sinapsis y cada agrupación vertical, una capa [32].

2.3.1.3 Máquina de Vectores de Soporte

Una Máquina de Vectores de Soporte puede definirse como un método de aprendizaje supervisado para clasificación y regresión, que utiliza el espacio de hipótesis de una función linear para construir un hiperplano óptimo respecto a los atributos de la información ingresada y las clases en las que se desea separar (en el caso de clasificación) [33]. Este método utiliza la minimización de un tipo de error previamente escogido para la optimización de sus resultados [33].

2.3.1.4 Bosques Aleatorios

En base a un set de entrenamiento X, compuesto de N casos, que pertenecen a dos clases, se podría construir un árbol de clasificación de la siguiente manera:
Un atributo *x* y un límite *t* que divide X en 2 subconjuntos que son máximamente distintos, de acuerdo a un criterio especificado previamente, son seleccionado de todos los atributos de *x* y todos los posibles valores de *t* [34]. En cada subconjunto generado se vuelve a realizar el mismo proceso, hasta que no se pueda dividir los subconjuntos obtenidos [34].
En el método de Bosques Aleatorios, en vez de utilizar el conjunto de entrenamiento en su totalidad, se selecciona una cantidad *N* de casos de manera aleatoria, así como una cantidad *g* de atributos (usualmente $g=G^{1/2}$) elegidos de manera aleatoria, participan en la división del conjunto.
Es una técnica popular de clasificación parecida a la de Árboles Clasificadores para un gran rango de tipo de data, particularmente en casos donde la cantidad de atributos sea grande [34]. Sin embargo, si los atributos realmente relevantes son de cantidad considerablemente menor a

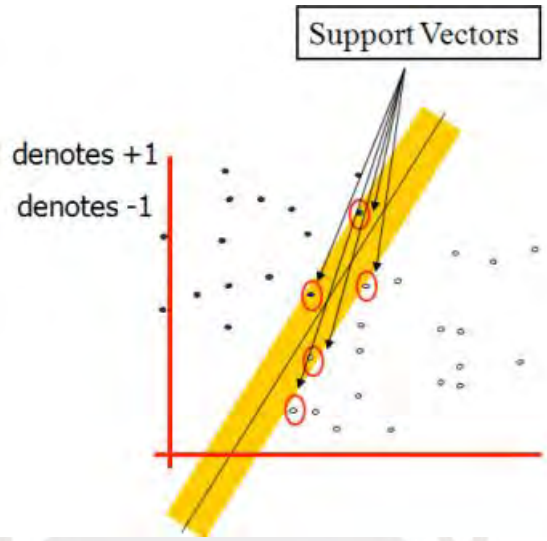

Figura 3. Ejemplificación de un hiperplano (resaltado en amarillo), en la ejecución de una Máquina de Vectores de Soporte para la clasificación de los datos (negros y blancos); los puntos más cercanos al hiperplano son los Vectores de Soporte [34].

la totalidad de atributos, los resultados obtenidos declinan rápidamente [34].

2.3.1.5 Métodos de Selección de Atributos

En esta sección se introducirá el método de interés de este proyecto y los métodos de selección de atributos tentativos a usar en la comparación.

2.3.1.5.1 Manto de Markov

En una red bayesiana, el manto de Markov de un nodo objetivo consiste de sus nodos padres, hijos directos y padres de los hijos (esposos); siendo este nodo independiente del resto de nodos de la red dado su manto de Markov [14].

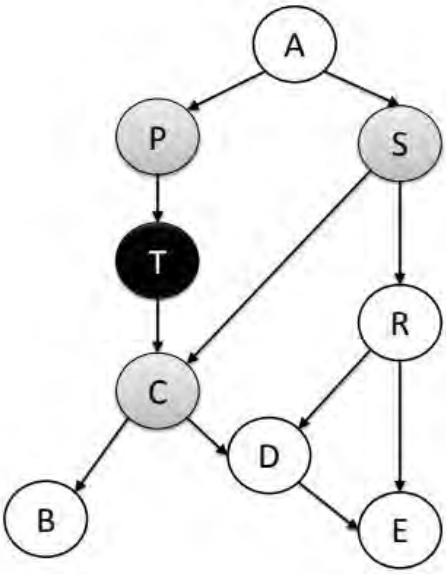

Figura 4. Ejemplo de una red bayesiana. Para el nodo T, los nodos P, S y C conformarían su Manto de Markov. [15]

2.3.1.5.1.1 Fidelidad

Una red bayesiana *G* y una distribución conjunta *P* son fieles uno con el otro, sí y solo sí cada relación de independencia condicional presente en *G* y en el Manto de Markov también está presente en *P* [23].

2.3.1.5.1.2 Restricción simétrica

Para que un nodo *X* sea padre o hijo de un nodo *T*, dentro de su Manto de Markov *X* debe pertenecer al conjunto de padres e hijos del nodo *T* y *T* debe pertenecer al conjunto de padres e hijos del nodo *X*.

2.3.1.5.2 Paseo Aleatorio

En el ámbito de Aprendizaje Máquina, puede definirse, para grafos, como la transición iterativa de un caminante de su nodo actual a uno vecino, seleccionado de manera aleatoria. Formalmente, el Paseo Aleatorio puede definirse como:

$$\mathbf{p}^{t+1} = (1 - r)\mathbf{W}\mathbf{p}^{t} + r\mathbf{p}^{0}$$

donde W es la matriz de adyacencia normalizada por columna del grafo y $\boldsymbol{p}^{t}$ es un vector en donde el *i* elemento contiene la probabilidad de estar en el nodo *i* en el paso *t* [36].

Ha sido utilizado en el pasado para la priorización de genes como candidatos [35].

2.3.2 Conceptos de Complejidad Computacional

La complejidad computacional representa el número de operaciones elementales que tomaría teóricamente un algoritmo diseñado para la resolución de algún problema elegido. Para su análisis e interpretación se utiliza la siguiente nomenclatura [28]:

- O(n): La letra *O* representa el peor tiempo computacional posible, y la letra al interior del paréntesis *n* representa la complejidad calculada. En este caso, se trataría de una complejidad lineal, proporcionalmente lineal al número de datos que ingresen al algoritmo.
- o(n): La letra *o* representa el mejor tiempo computacionalmente posible para un algoritmo dado.

Capítulo 3.Estado del Arte

3.1 Introducción

En esta sección se procederá a describir las investigaciones y avances realizados relacionadas al análisis de las interacciones génicas, en particular las soluciones que sean resultado de la aplicación de enfoques y metodologías de aprendizaje de máquina para la detección/predicción de interacciones génicas.

Para el desarrollo de esta sección, se usó el método de revisión sistemática, el cual se utiliza para identificar, evaluar e interpretar toda la literatura disponible relevante para un tema, pregunta de investigación o campo en particular [20].

Se seleccionaron las siguientes palabras clave relacionadas a este proyecto de fin de carrera:

- red bayesiana
- manto markov
- detección interacciones génicas
- soluciones informáticas médicas
- análisis complejidad computacional
- aprendizaje máquina

A continuación, se definieron las preguntas que se desean responder:

- ¿Cuáles son las metodologías usadas a la actualidad para la estimación de interacciones génicas causales?
- ¿Qué algoritmos existen a la actualidad para la estimación del manto de markov?
- ¿Cuál es la viabilidad de aplicación de cada uno de estos algoritmos, en base a su complejidad computacional?
- ¿Cuáles de estos algoritmos han sido previamente implementados en código abierto?

Luego, se procedió a elaborar las cadenas de búsqueda a utilizar dentro de las bases de datos, con el fin de encontrar bibliografía que responda las preguntas definidas.

El primer conjunto aborda la evolución de la metodología para la estimación del manto de Markov dentro de una red bayesiana, estrategia crucial para un uso eficiente de estos modelos con gran cantidad de datos:

- red bayesiana
- Manto Markov

El segundo conjunto aborda los enfoques usados para la inferencia de interacciones génicas hasta la actualidad:

- interacciones génicas
- predicción comportamiento genes

El tercer conjunto aborda la complejidad computacional presente en algoritmos, enfocándose en aquellos más usados en aprendizaje de máquina, y una unidad de medida para evaluar su viabilidad:

- análisis complejidad computacional
- aprendizaje máquina

A partir de los conjuntos establecidos, se formaron las siguientes cadenas de búsqueda:

- C1: (‘Red bayesiana” OR “manto markov”) AND (“interacciones génicas” OR “predicción comportamiento genes”)
- C2: (‘Red bayesiana” OR “manto markov”) AND (“análisis complejidad computacional” OR “aprendizaje máquina”)
- C3: (“análisis complejidad computacional” OR “aprendizaje máquina”) AND (“interacciones génicas” OR “predicción comportamiento genes”)

Las cadenas generadas fueron ingresadas en las siguientes bases de datos:

- PubMed
- ACM

- IEEE

Los criterios adicionales dentro las búsquedas realizadas fueron los siguientes:

- La fecha de publicación debe ser menor de 20 años.
- Como mínimo, deberán aparecer dos grupos de palabras claves dentro de las cadenas de búsqueda ingresadas.

Después de la aplicación de las cadenas de búsqueda en cada base de datos, se obtuvo:

- PubMed: se obtuvo 9 resultados para C1, 106 resultados para C2, y 40 resultados para C3
- ACM: se obtuvo 5 resultados para C1, 277 resultados para C2, y 18 resultados para C3
- IEEE: se obtuvo 39 resultados para C1, 1598 resultados para C2, y 1215 resultados para C3

De los artículos hallados, se seleccionaron 5 de mayor relevancia:

- Aliferis, C. F., Tsamardinos, I., & Statnikov, a. (2003). HITON: a novel Markov Blanket algorithm for optimal variable selection. AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium, 21–25.
- Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2017). Deep Learning for Health Informatics. IEEE Journal of Biomedical and Health Informatics, 21(1), 4–21.
- Gao, T., & Ji, Q. (2017). Efficient scored-based Markov Blanket discovery. International Journal of Approximate Reasoning, 80, 277-293.
- U., A., S., I., C., Ö., A., A., & H.H., O. (2013). A dynamic Bayesian framework to learn temporal gene interactions using external knowledge. 2013 8th International Symposium on Health Informatics and Bioinformatics, HIBIT 2013

- Myte, R., Gylling, B., Häggström, J., Schneede, J., Magne Ueland, P., Hallmans, G., … Van Guelpen, B. (2017). Untangling the role of one-carbon metabolism in colorectal cancer risk: a comprehensive Bayesian network analysis. Scientific Reports, 7, 43434. https://doi.org/10.1038/srep43434

De este modo, se presentan los resultados obtenidos en base a la metodología aplicada:

3.2 Resultados de la revisión sistemática

3.2.1 Evolución de los algoritmos para la estimación del Manto de Markov

3.2.1.1 Métodos de estimación del Manto de Markov

En la actualidad, existen varios tipos de algoritmos para la estimación del Manto de Markov dentro de una red bayesiana, ya que representa una estructura crucial en el uso de estos modelos de predicción. La mayoría de estos métodos son basados en pruebas de independencia condicional. Pueden ser divididos en dos subgrupos: los que se basan en la topología del nodo objetivo (métodos locales) y los que no (métodos globales). A continuación, se presentará algunos algoritmos relevantes del primer subgrupo, dado que los métodos contenidos en el segundo son poco eficientes para grandes cantidades de variables [21], como es el caso del presente proyecto.

3.2.1.1.1 Métodos de estimación del Manto de Markov basados en la topología del nodo objetivo

Los métodos del Mínimo-Máximo Manto de Markov utilizan pruebas de independencia para comparar el nodo objetivo de los demás, y su incremento de eficiencia radica en encontrar primero los nodos padres e hijos del nodo objetivo, y luego encontrar los nodos esposos del nodo objetivo (padres de los hijos del nodo objetivo). Los métodos clásicos, sin

embargo, realizan un paso adicional de revisión de simetría en el grupo de nodos hallados para retirar falsos positivos, pero añaden mayor complejidad de la ya existente (exponencial) [21]. Los métodos más conocidos son:

- **HITON-MB:** Desarrollado por Tsamardinos et. al. [22], utiliza la metodología mencionada previamente, insertando una por una las variables dentro del conjunto estimado de padres e hijos durante el descubrimiento de este, de tal manera que si la variable insertada resultara independiente respecto a la variable (nodo) objetivo, no sería considerada como padre ni como hijo [22].

- **IAMB:** Propuesto en el 2003 para problemas de clasificación en investigaciones de microarreglos donde lo común era la aparición de miles de variables. Ordena el ingreso de variables de mayor a menor fuerza de asociación con el nodo objetivo cuando aún el conjunto estimado de nodos pertenecientes al manto de markov está vacío, y reordena los nodos conforme se admitan más nodos durante su trayecto, lo que disminuye el número de falsos positivos respecto a sus predecesores [23].

- **MMPC:** Asume la condición de fidelidad y prueba estadística de independencia condicional correcta. Consta de dos fases principales: la fase de crecimiento, que utiliza el enfoque de “Máximo Mínimo” para la adición de un atributo al conjunto candidato de padres e hijos; y la fase de sustracción, donde descarta atributos calificados como “redundantes” del conjunto candidato[23].

- **PCMB:** Asume la condición de fidelidad y prueba estadística de independencia condicional correcta. Al utilizar los mismos pasos que **IAMB**, aún cuenta con ineficiencia a la hora de procesar miles de atributos, a pesar de que sus autores resaltan el alto valor de precisión que puede llegar a obtener [23].

- **IPCMB:** Se concentra principalmente en remover falsos positivos durante cada entrada de nuevas variables al manto de markov estimado, en tanto esto debería ayudar a disminuir la cantidad de pruebas de dependencia que se realicen posteriormente. En la revisión de Fu y Desmarais [23] fue apuntado como el algoritmo con mejor ganancia respecto a tiempo invertido, datos insertados e información descubierta.

- **S2MTB+:** Se basa principalmente en el descubrimiento simultáneo de nodos esposos dentro del conjunto hallado de padres e hijos, teniendo en consideración que los falsos positivos dentro de este conjunto son en parte los nodos esposos que faltan encontrar (Figura 5). Por el riesgo de que el conjunto de padres e hijos con falsos positivos sea demasiado largo, primero se separa los falsos positivos del conjunto de padres e hijos descubierto en un conjunto *S*, y en un segundo paso se procede a remover los falsos positivos esposos de este conjunto para así completar el manto de markov [21].

Algorithm 3 S^2TMB+ algorithm.

1: **Input:** dataset D, target node T
{step 1: find the PC set }
2: $\mathbf{PC}_T \leftarrow \emptyset$, $\mathbf{O} \leftarrow \mathbf{V} \setminus \{T\}$;
3: **while O** is nonempty **do**
4: choose $X \in \mathbf{O}$, $\mathbf{O} \leftarrow \mathbf{O} \setminus \{X\}$;
5: $\mathbf{Z} \leftarrow \{T, X\} \cup \mathbf{PC}_T$;
6: $G \leftarrow$ BNStructLearn $(\mathbf{Z}, D_Z)$;
7: $\mathbf{PC}_T \leftarrow findPC(G)$;
8: **end while**
{step 2a: remove false PC nodes and keep spouses}
9: $\mathbf{S}_T \leftarrow \emptyset$, $\mathbf{O} \leftarrow \mathbf{V} \setminus \mathbf{PC}_T$;
10: **while O** is nonempty **do**
11: choose $X \in \mathbf{O}$, $\mathbf{O} \leftarrow \mathbf{O} \setminus \{X\}$;
12: $\mathbf{Z} \leftarrow \{T, X\} \cup \mathbf{PC}_T$;
13: $G \leftarrow$ BNStructLearn $(\mathbf{Z}, D_Z)$;
14: $\mathbf{PC}_T \leftarrow findPC(G)$;
15: **spouse** $\leftarrow findSpouse(G)$;
16: $\mathbf{S}_T \leftarrow \mathbf{S}_T \cup$ **spouse**;
17: **end while**
{step 2b: shrink spouses}
18: $\mathbf{O} \leftarrow \mathbf{S}_T$, $\mathbf{S}_T \leftarrow \emptyset$;
19: **while O** is nonempty **do**
20: choose $X \in \mathbf{O}$, $\mathbf{O} \leftarrow \mathbf{O} \setminus \{X\}$;
21: $\mathbf{Z} \leftarrow \{T, X\} \cup \mathbf{PC}_T \cup \mathbf{S}_T$;
22: $G \leftarrow$ BNStructLearn $(\mathbf{Z}, D_Z)$;
23: $\mathbf{PC}_T \leftarrow findPC(G)$;
24: $\mathbf{S}_T \leftarrow findSpouse(G)$;
25: **end while**
26: **Return: MB** $\leftarrow \mathbf{PC}_T \cup \mathbf{S}_T$

Figura 5. Pseudocódigo del algoritmo S2TMB+. [23]

Dentro del artículo *Markov Blanket Based Feature Selection: a Review of Past Decade [23],* se menciona que, a pesar que fue Pearl quien comenzó el estudio de la importancia del Manto de Markov durante sus investigaciones iniciales acerca de las redes bayesianas, fueron Koller y Sahami quienes, gracias al algoritmo homónimo que crearon, lograron el reconocimiento de la relevancia que podía tener la estimación del manto de Markov de una clase objetivo para una mejor selección de atributos para una gran cantidad de datos y atributos, respectivamente [23]. Su algoritmo requiere dos entradas principalmente: el número de atributos a retener, y el número máximo de atributos que el algoritmo debería ser capaz de evaluar en un mismo instante (al realizar las pruebas de dependencia). Si bien no garantiza la certeza de sus resultados, este algoritmo despertó el interés de gran cantidad de investigadores, lo que contribuyó altamente al desarrollo de este campo [23].

En el artículo *HITON: a novel Markov Blanket algorithm for optimal variable selection, [22]*, se introduce el algoritmo HITON para el descubrimiento del manto de markov, y se propone como el primer enfoque viable para el análisis de gran cantidad de datos y variables, dado enfoques anteriores con muy poco resultado en este campo [22].

En el artículo *Efficient scored-based Markov Blanket discovery [21],* se brinda un mayor avance a los enfoques algorítmicos para la estimación del Manto de Markov, atacando principalmente el segundo paso de los enfoques clásicos, la búsqueda de nodos redundantes dentro del manto candidato, de complejidad super exponencial, para así disminuir la complejidad computacional que estos requerían.

3.2.2 Aplicación de enfoques de aprendizaje máquina para la detección o predicción de interacciones génicas:

Se ha podido observar la gran cantidad de herramientas de aprendizaje máquina

que han sido aplicados, sobre todo aprendizaje profundo [24]. Gracias a los fundamentos del aprendizaje profundo en el estudio de redes neuronales, se han desarrollado múltiples arquitecturas para abarcar distintos problemas presentados dentro de la medicina, siendo las arquitecturas de Redes Neuronales Convulsiónales, Máquinas de Boltzmann, Redes de Creencias Profundas y Redes Neuronales Recurrentes Profundas. Asimismo, gracias a los avances logrados en el desarrollo de unidades de procesamiento gráfico, la cantidad de datos viables a procesar ha crecido de manera exponencial [24], facilitando el avance y adaptación de estos enfoques dentro del área médica.

Entre las aplicaciones halladas, las de mayor relevancia para este proyecto de fin de carrera han sido las siguientes:

- Aplicación de redes bayesianas dinámicas para aprender interacciones génicas a través del tiempo insertando conocimientos externos; donde a partir de la noción de robustez contra la sobrealimentación y ruido de los datos ingresados que brindan los modelos de redes bayesianas, los autores utilizan el marco de trabajo BNP (Bayesian Network Prior) para incorporar información externa existente para el modelo. Los resultados obtenidos superaron a las redes bayesianas dinámicas comunes; sin embargo, también se aclara que el enfoque utilizado fue para una pequeña cantidad de datos en comparación a los potencialmente existentes acerca del genoma humano [27].

- Análisis de rol del metabolismo del carbón-uno en cáncer colon rectal, el cual ha sido extensamente estudiado, no ha logrado resultados concluyentes. Dada la complejidad del metabolismo del carbón-uno, se ha propuesto utilizar un enfoque de aprendizaje de máquina utilizando redes bayesianas. La red bayesiana combinada con datos utilizando 3 tipos diferentes de algoritmos de estimación, obtuvieron relaciones bioquímicas plausibles [29].

- Estimación de parámetros en una red bayesiana usando inteligencia de enjambre superpuesta; ya que se ha tomado en consideración el potencial de relaciones “escondidas” dentro de las variables de un conjunto de datos, y la facilidad que posee las redes bayesianas para encontrar los atributos más relevantes para la clasificación de datos. Realiza la estimación del Manto de Markov estableciendo competencia entre múltiples enjambres con regiones distintas (grupos de nodos) como objetivo; y finalmente muestra la superioridad de este método versus el enfoque de un solo enjambre [30].

3.3 Conclusiones

A través de la revisión sistemática realiza, se puede concluir que se ha aplicado los modelos de redes bayesianas para recuperar la red de interacciones entre genes, sea para entender una enfermedad en particular o conocer las interacciones en sí a partir de datos previamente obtenidos a través de ensayos. Se obtiene una gran retroalimentación de estos artículos, ya que permite conocer las debilidades con las que se encuentran los modelos de redes bayesianas al ser aplicados en este campo de la medicina, siendo los más relevantes la cantidad de datos que se puede procesar de manera viable y, por ende, la complejidad computacional de los algoritmos utilizados para realizar los ensayos respectivos.

Asimismo, se ha podido rescatar la evolución que ha sufrido la estimación del Manto de Markov dentro de una red bayesiana, factor determinante para su viabilidad dentro de conjunto de datos de gran tamaño, como son las inferencias génicas a través del genoma humano. Los algoritmos hallados cuentan con implementaciones usualmente en Python o R, lo que también abre campo a nuevas mejoras, sea en la implementación de los últimos algoritmos creados o en el lenguaje de elección para los módulos del mismo.

Finalmente, por lo resaltado previamente, se observa la posibilidad del desarrollo de un enfoque para el análisis y predicción de interacciones causales génicas contemplando su viabilidad para gran cantidad de datos, sin sacrificar la precisión a obtener. Este enfoque se basaría en el uso de algoritmos de estimación del Manto de Markov del

estado de arte a través de muestras estadísticas de un conjunto de datos de expresión génica disponible. Se realizará experimentación numérica para encontrar mantos de Markov estables y relevantes para predecir las interacciones génicas. De esta forma se contará con una base sólida para construir distintos modelos de predicción.

Capítulo 4. Metodología de muestreo de datos sobre interacciones génicas

4.1 Introducción

En este capítulo se presentará el desarrollo del resultado esperado 1. Para este fin, se analizaron un conjunto de datos sobre interacciones génicas. La metodología a plantear debe satisfacer las siguientes consideraciones: el tamaño de las muestras, la reserva de una fracción de estas para la validación de los modelos clasificadores y el algoritmo genético, y la agrupación en conjuntos de entrenamiento y validación según alguna característica en particular.

4.2 Descripción del resultado

El conjunto de datos usado corresponde a datos de expresión génica de líneas celulares BL2 (Burkitt's lymphoma) obtenidos en el Instituto de Genómica Funcional de la Universidad de Regensburg, Alemania. Los datos están disponibles en el repositorio Gene Expression Onnibus - GEO (http://www.ncbi.nlm.nih.gov/geo) en los accessions GSE71721 y GSE89936. Los datos constan de 2 series temporales por cada interacción gen-gen. La primera serie es relativa al gen causa y la segunda serie al posible gen efecto. En adición a las series temporales, se cuenta con información de experimentos intervencionales con inhibidores. Esta información es representada con p-values que indican el efecto de la inhibición del gen causa en el gen efecto (este valor oscila entre 0 y 1, a menor valor, mayor será la probabilidad de que el gen causa y el gen efecto esten realmente relacionados de forma causal). Asimismo, cada interacción ha sido etiquetada con el tipo de inhibidor usado para desvendar la interacción, por lo cual, contamos con una totalidad de 7 conjuntos de datos distinguidos principalmente por el inhibidor utilizado. Estos son: CTNNB1, ERK, IKK2, IRF4, JNK, LEF1 y MYC.

Es así como la metodología a proponer deberá usar los inhibidores como método de agrupación, y esta dependerá de los limites inferiores y superiores impuestos sobre el p-value para definir las clases.

Adicionalmente se desea que la metodología lidie con la característica más prominente de los conjuntos de datos a disposición, la cuál es la desproporcionalidad de instancias en la clase positiva (interacciones causales de acuerdo al límite superior impuesto en los p-values) respecto a las instancias de la clase negativa (interacciones no causales).

4.3 Desarrollo del resultado

A continuación, se presentarán los resultados obtenidos a través del desarrollo de este proyecto de fin de carrera:

4.3.1 Exploración de los datos a disposición

Se realizó una exploración de los datos a disposición, con la finalidad de aseverar las consideraciones a tener en cuenta durante el diseño de la metodología de trabajo a aplicar en estos.

- CTNNB1: 100146 instancias
- Ikk2: 200292 instancias
- IRF4: 100146 instancias
- Erk: 200292 instancias
- MYC: 100146 instancias
- Jnk: 300438 instancias
- LEF1: 100146 instancias

Se determinó aplicar un criterio estricto en los datos mencionados para determinar las instancias de la clase positiva y negativa. Para ello se estableció como instancias positivas aquellas con un pvalue menor a 0.01, y como instancias negativas aquellas con un pvalue mayor a 0.5. Con estos umbrales se filtró los datos, quedando su distribución de la siguiente forma:

- CTNNB1: 82038 instancias
- Ikk2: 108984 instancias
- IRF4: 96654 instancias
- Erk: 140730 instancias
- MYC: 64002 instancias
- Jnk: 213102 instancias
- LEF1: 68070 filas

Asímismo, se midió el grado de balance (proporción de número de instancias positivas a número de instancias negativas) en cada conjunto de los conjuntos de datos, brindando los siguientes resultados:

- CTNNB1: 0.0732%
- Ikk2: 4.3549%
- IRF4: 0.0683%
- Erk: 0.1879%

- MYC: 9.5849%
- Jnk: 3.4516%
- LEF1: 0.1943%

Como se puede observar, la clase positiva (interacciones causales) tiene un tamaño considerablemente menor a la clase negativa

4.3.2 Consideración de tamaño en muestras generadas

Como se pudo observar en los resultados de la exploración de datos, se cuentan con más de 1 millón de instancias en su totalidad, así como más de 50’000 instancias por cada grupo de muestras por inhibidor utilizado. Siendo uno de los propósitos alcanzar cierta eficiencia en el procesamiento de los datos, se escogió un tamaño de 6000 instancias para cada muestra aleatoria generada, en base a los siguientes criterios:

- La totalidad de métodos de estimación del Manto de Markov se apoyan en la lectura transversal de los atributos en forma de vectores para su posterior comparación; en consecuencia, el tiempo de procesamiento aumenta por dos factores principales: el número de atributos a evaluar y el tamaño de los vectores mencionados, que serían el tamaño de las muestras a generar. El proceso descrito en la Figura 6 y 7 se repite por cada combinación posible de valores entre los 2 atributos.
- Se cuenta con 540 atributos discretos de cada dato, lo cual presenta cierto nivel predeterminado de complejidad computacional (siendo el peor de los casos O(2**n), factor que podría aliviarse con la selección de un tamaño prudente para cada muestra.

Figura 6. Flujo de lectura de 2 vectores, representando 2 atributos distintos, para comparar su similitud en base al valor "3". Elaboración propia.

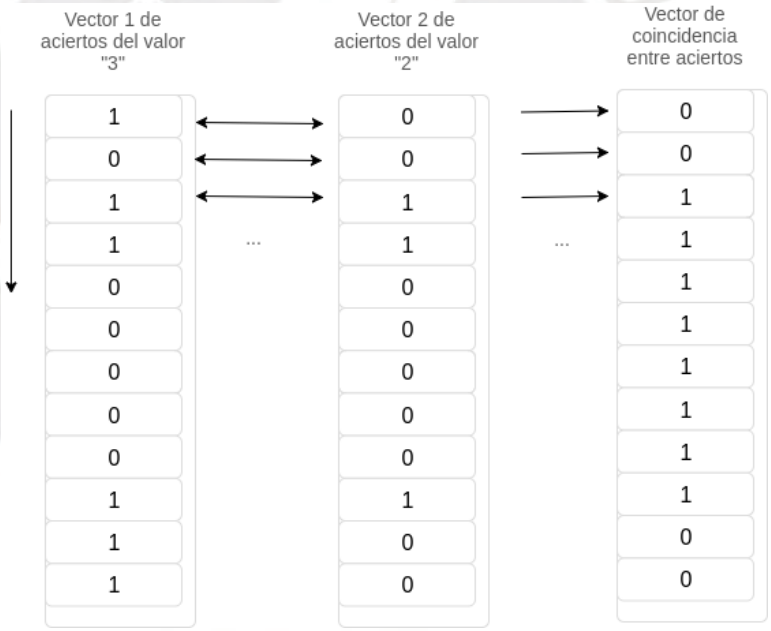

Figura 7. Flujo de análisis de atributos para el cálculo de la dependencia. Elaboración propia.

4.3.3 Consideración de agrupación de muestras según el inhibidor utilizado

Como se comentó previamente, los datos a disposición pueden clasificarse de acuerdo al inhibidor utilizado durante la experimentación. La metodología a proponer busca el entrenamiento de modelos predictivos para predecir la presencia de interacciones génicas ante nuevos inhibidores. Esto podría emularse con la data que se dispone de la siguiente manera:

- Crear conjuntos de datos que consten de todos menos un inhibidor, por cada inhibidor presente en el conjunto inicial de datos.

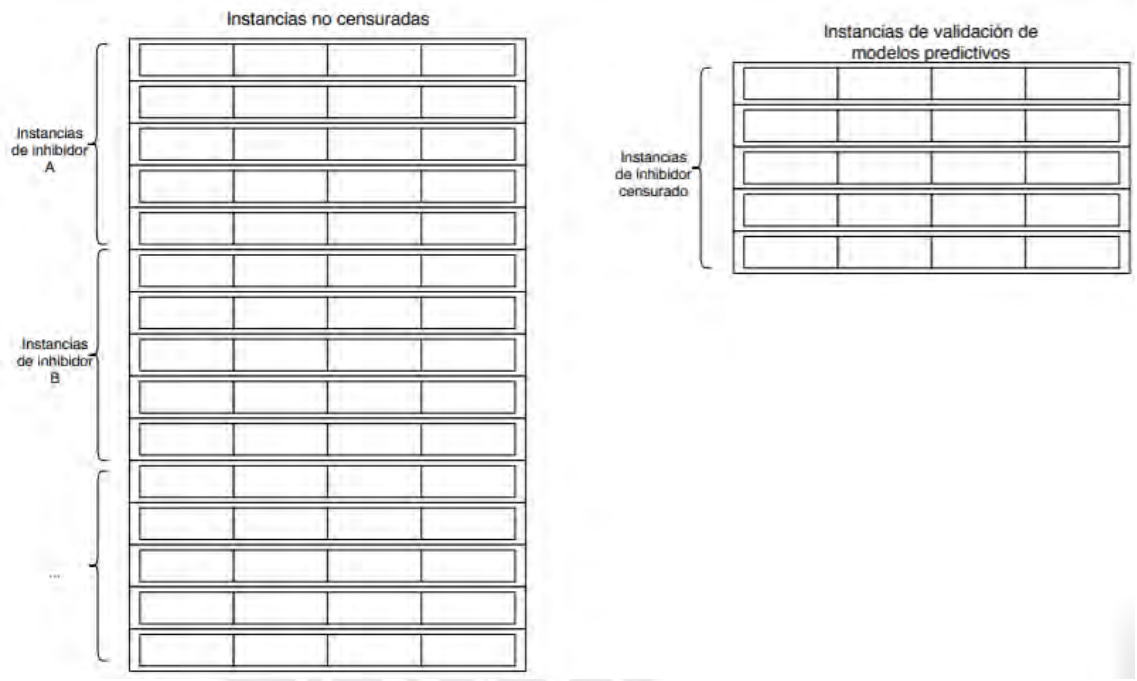

Figura 8. Organización inicial de los datos. Elaboración propia.

- Usar estos conjuntos para la generación de muestras aleatorias, y reservar los datos con el inhibidor censurado por cada conjunto para la validación final de los modelos predictivos.

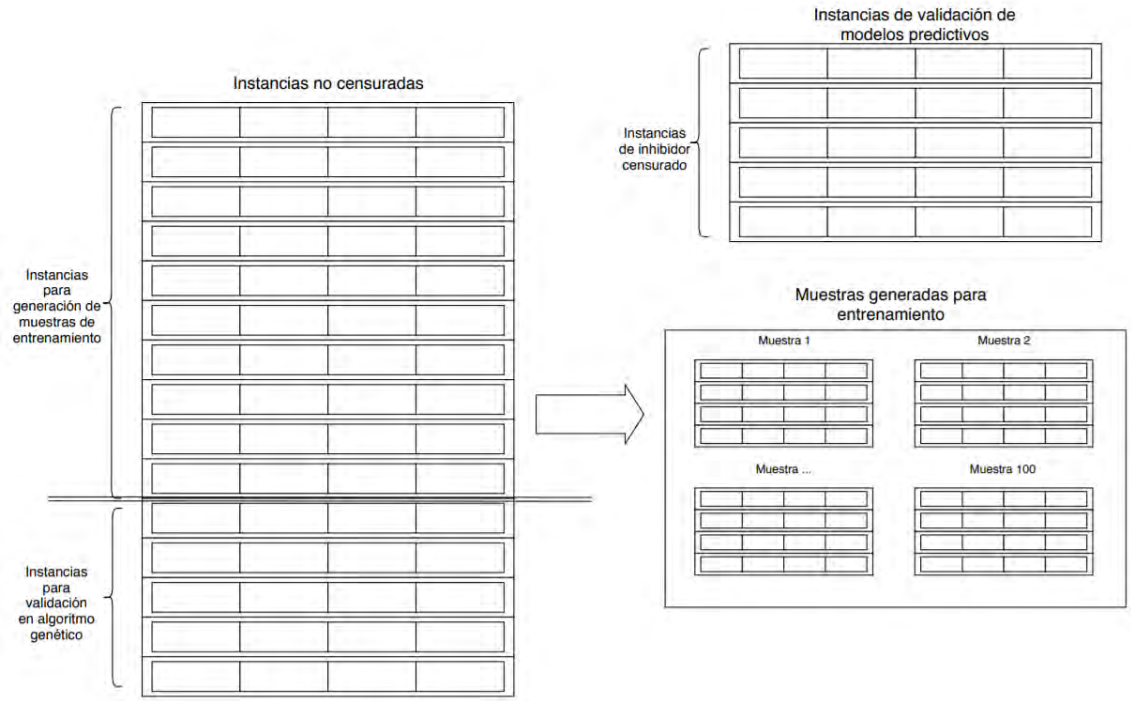

Figura 9. Generación de muestras balanceadas. Elaboración propia.

4.3.4 Separación de muestras en conjuntos de entrenamiento y de validación

En línea con las consideraciones propuestas anteriormente [38], y la metodología básica de experimentación en el campo de aprendizaje máquina, se propone la separación de los conjuntos de datos y muestras de la siguiente manera:

- Una fracción de las muestras aleatorias será reservada para evaluación por parte del método de estimación de Manto de Markov a diseñar y los métodos de selección de atributos adicionales seleccionados como referencia.

- Una fracción de las muestras aleatorias será reservada para la validación durante el procesamiento de los conjuntos de atributos obtenidos por los métodos selectores de atributos previos, dentro del algoritmo genético.

4.3.5 Ponderación de resultados de muestras por cada inhibidor

El primer enfoque que se ha escogido para el análisis de los resultados es a través de la ponderación de resultados de muestras por cada inhibidor. Para ello, se tomaron las siguientes consideraciones:

- Sumar el puntaje obtenido de cada atributo, respecto a la clase objetivo, con un selector/calificador de atributos, de cada muestra analizada, para obtener el puntaje “global” de cada atributo.
- Para los selectores de atributos, se ha considerado el conjunto de atributos resultante como único conjunto de análisis.
- Para los calificadores de atributos, se ha considerado los 2, 3, 5, 10 y 15 mejores atributos, de acuerdo con el puntaje total, para obtener una visión parcial de la progresión del rendimiento (medido como precisión) del calificador.
- En base a trabajos previos, donde se usó un algoritmo genético en conjunto con el concepto del Manto de Markov [42], se consideró filtrar los resultados obtenidos exclusivamente con los estimadores del Manto de Markov, con un algoritmo genético.

4.3.6 Filtración de resultados a través de un algoritmo genético

El segundo enfoque, que resulta una expansión parcial del primer enfoque de análisis propuesto, trabajará en base al trabajo de Zexuan [42] con los resultados obtenidos por los estimadores del Manto de Markov, centrando la relevancia del enfoque en la diversificación de los individuos dentro del universo inicial, para obtener resultados positivos. Así, este enfoque consta de los siguientes pasos:

- Usando la frecuencia con la que apareció cada atributo en el conjunto candidato del Manto de Markov, se ordenarán los atributos de mayor número de apariciones a menor número de apariciones.
- Como población inicial, se limitará la participación de los atributos que tuviesen X % de aparición, mayor a un límite inferior impuesto arbitrariamente. Los límites impuestos fueron 0%, 15% y 30%. Así, solo los atributos que hayan aparecido por lo menos 1 vez en los resultados de las muestras.
- Usando una fracción reservada de las instancias iniciales de entrenamiento (la concatenación de todas las instancias que no incluyan al inhibidor Y), se medirá el fitness de cada individuo de la población.
- Estos pasos se repetirán por cada inhibidor existente del banco de datos.

4.3.7 Generación de modelos predictivos por cada muestra generada

El tercer enfoque de evaluación de los selectores/calificadores de atributos, se basa en el trabajado realizado por Villanueva, donde se generó un modelo predictivo por cada muestra generada, usando estas últimas como fuente de entrenamiento, filtradas por los conjuntos de atributos resultantes de la aplicación de los métodos selectores/calificadores de atributos [38], para luego obtener la media geométrica de los modelos en sus respectivos conjuntos de prueba.

A través del flujo presentado en la figura 10, será posible analizar la influencia de los modelos predictivos en conjunción con los selectores/calificadores de atributos.

4.3.8 Agregación de resultados de cada modelo predictivo generado

El cuarto enfoque de evaluación resulta ser una expansión del tercer enfoque. A través de la ponderación de los resultados brindados por los modelos predictivos generados por cada muestra, se evaluará nuevamente la media geométrica de estos resultados sobre el conjunto de datos de prueba (del inhibidor censurado).

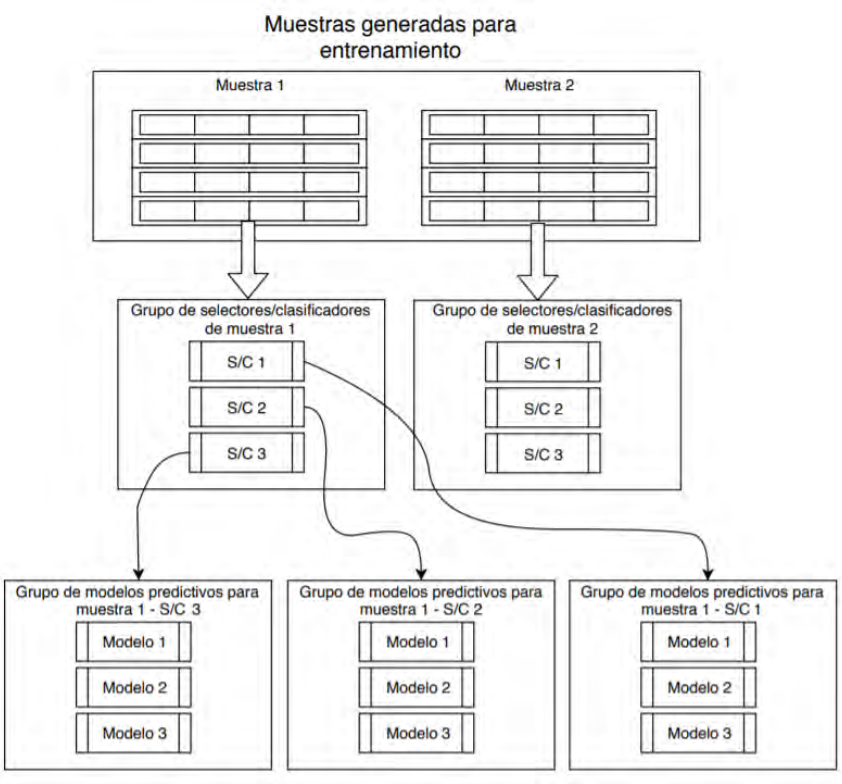

Figura 10. Figura de análisis de cada muestra generada, que será procesada por un grupo de S/C de atributos y un grupo de modelos predictivos. Elaboración propia.

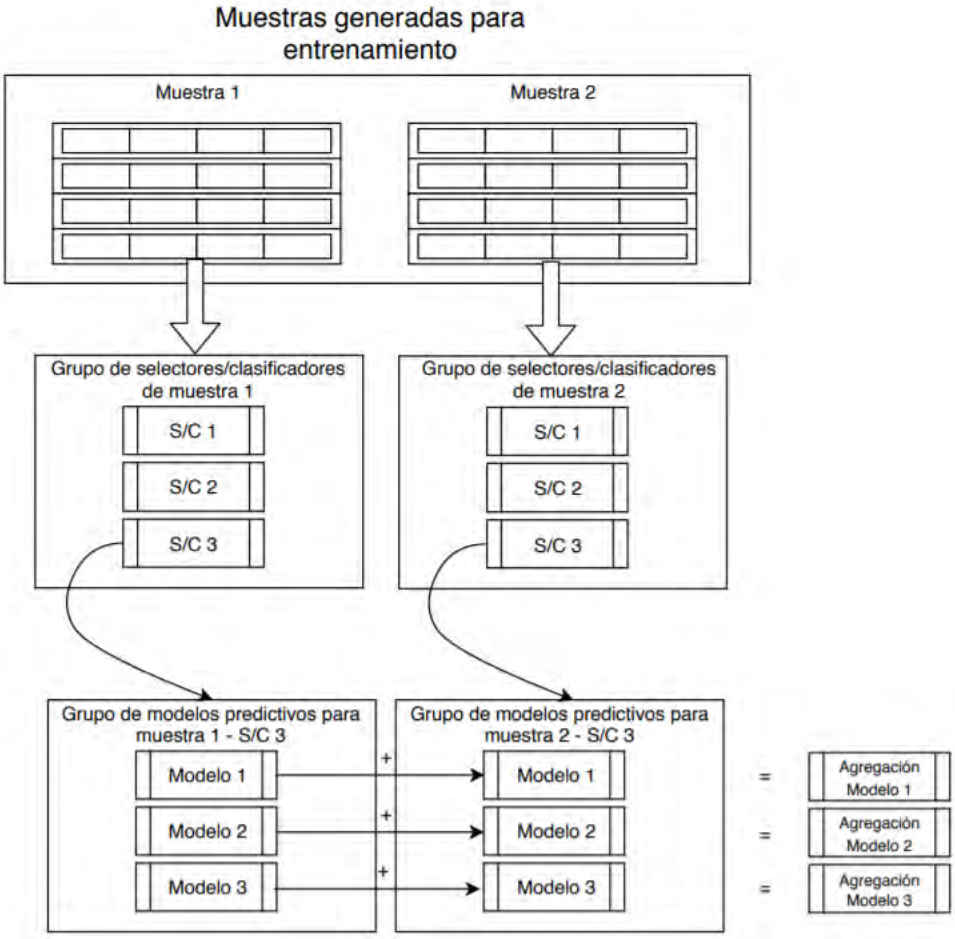

Figura 11. Figura de agregación de modelos por cada S/C, por cada muestra generada, respectivamente. Elaboración propia.

Capítulo 5. Diseño del algoritmo estimador del Manto de Markov propuesto y su desempeño

5.1 Introducción

En este capítulo, se presentará el desarrollo del estimador del Manto de Markov respecto a un algoritmo del estado del arte (MMPC). Asimismo, se describirán los resultados obtenidos de la aplicación de este algoritmo y otros algoritmos de selección de atributos sobre conjuntos de muestras seleccionados para entrenamiento respectivamente.

5.2 Descripción del resultado

Después de diversos experimentos, se ha diseñado un algoritmo que busca atacar las principales debilidades halladas en las implementaciones halladas de estimación del Manto de Markov, tomando como base la implementación clásica del MMPC (Max-Min Parents and Children) [23]. A su vez, la aplicación de este algoritmo, en conjunto con los algoritmos propuestos de selección de atributos para comparación, proveerá de un set de atributos relevantes por cada muestra analizada.

5.3 Desarrollo del resultado

5.3.1 Características comunes en estimadores de Manto de Markov y sus debilidades

En la totalidad de métodos de estimación de Manto de Markov encontrados en la revisión del estado del arte, se ha observado las siguientes características:

- Lectura transversal de columnas (atributos) para realizar mediciones de correlaciones entre atributos, un par vector-vector por cada combinación posible de valores entre ambas columnas.

- Una fase de búsqueda de atributos candidatos.

- Una fase de descarte de atributos candidatos con información redundante.

De estas 3 características, se encontraron las siguientes debilidades principales:

- La lectura transversal de atributos se realiza cada vez que exista una comparación entre atributos. Los métodos de estimación de correlación pueden considerar un grupo de atributos “separadores”, lo cual eleva el número de combinaciones a evaluar a *n potencia de 2,* y dentro de cada combinación evaluada, se realiza la lectura transversal en cada atributo por cada valor posible que contenga. Entonces, se puede decir que existe una gran redundancia en la lectura de datos, y por lo tanto, una ineficiencia computacional asociada.

- La fase de búsqueda de atributos candidatos, en los métodos menos robustos como GetMB [23], no se ven afectados en particular por el orden. Sin embargo, métodos como MMPC e IAMMB, sí se ven afectados por el orden de lectura de atributos, puesto que no consideran el caso donde, durante la adición de un atributo, exista más de un atributo que cumpla todos los requerimientos necesarios para ingresar al conjunto candidatos de atributos.

- La fase de búsqueda de información redundante usa la existencia de un grupo de atributos como “separador” del flujo de información de un atributo al atributo objetivo. El análisis de la existencia de este “separador” es de complejidad exponencial, y su aplicación depende del orden de ingreso de los atributos al grupo candidato [23]. Para que exista este separador, debe existir un atributo dentro del grupo candidato que contenga mayor cantidad de información acerca del atributo objetivo que el otro atributo analizado. Conforme se analiza la existencia de un separador para cada atributo, el grupo candidato va decreciendo, y la debilidad que se observa es el análisis de un separador entre dos atributos, considerando un conjunto de atributos como potenciales separadores, sin que la mayoría de estos contengan información

redundante, lo que conlleva a considerar combinaciones innecesarias de atributos, y por consecuente, una cantidad elevada de cálculos innecesarios.

5.3.2 Implementación clásica de MMPC y sus debilidades

La implementación básica del MMPC consta de 2 fases principales, mostradas en la Figura 12. Esta implementación toma en cuenta las 3 debilidades mencionadas de la mayoría de los métodos de estimación del Manto de Markov, que serán atacadas en el diseño del algoritmo propuesto en el acápite 5.3.3.

5.3.3 Manto de Markov como conjunto predictivo

Se buscó sustentar la factibilidad de aplicar métodos estimadores en los conjuntos de datos de interacciones génicas, a través de experimentación en conjuntos de datos artificiales. Al respecto, se utilizó el conjunto de datos artificial "Alarm", el cual se ha encontrado en la literatura [38]. La experimentación consistió en las siguientes fases:

- Se propuso un diseño alternativo al MMPC, para acelerar la evaluación de los datos a disposición. Consistió en desagregar los atributos de un conjunto de datos en subconjuntos de atributos, para aplicar la fase de crecimiento del MMPC sobre cada subconjunto, y obtener un conjunto de atributos que han aparecido por lo menos 1 vez en el resultado del análisis de cada subconjunto. Se descarta la segunda fase del MMPC (la poda o descarte de atributos redundantes), para analizar la pérdida de calidad de los conjuntos de atributos candidatos resultantes. El diseño del algoritmo se ha plasmado en la Figura 14.

- El *"Supercpc"* fue comparado con conjuntos de atributos analizados por otros selectores de atributos clásicos (T-test y Entropía), cuyo punto de corte fuera el tamaño del conjunto del *supercpc* obtenido.

Estos conjuntos fueron evaluados usando NaiveBayes y SVM, para obtener una comparación inicial de la calidad de cada conjunto. Pueden observarse los resultados de esta experimentación en la Figura 15.

Algorithm 1 Algoritmo MMPC

```
1: procedure MMPC
   Entrada: Atributo objetivo T, Conjunto de atributos a evaluar U
   Salida: Conjunto de atributos Padres e Hijos respecto al atributo objetivo
   –Fase 1
2: CPC = []
3: repetir
4:     < F, assocF >← MaxMinH(T,CPC, U)
5:     si assocF != 0 entonces:
6:         CPC = CPC U F
7:     fin si
8: hasta que el CPC deje de ser modificado
   –Fase 2
9: por todos los X atributos en el CPC:
10:    si Existe un separador para X en el CPC
11:        descartar X
12:    fin si

13: retornar CPC
```

Figura 12. Pseudocódigo del procedimiento principal del algoritmo Max-Min (adaptado de [39])

procedure MaxMinH
Entrada: T, CPC, U
Salida: Atributo A con mayor asocin del conjunto U
arregloDeAttEvaluados = []
por cada atributo X en U:
arregloDeAttEvaluados = arregloDeAttEvaluados **U** *MinAsoc(X, T, CPC, U)*
fin por
$< assocF, F >= max(arregloDeAttEvaluados)$
retornar $< F, assocF >$

Figura 13. Pseudocódigo de la heurística MaxMin del algoritmo Max-Min (adaptado de [39])

Algorithm 1 Algoritmo SUPERCPC

```
1:  procedure SUPERCPC
    Entrada: Atributo objetivo T, Conjunto de atributos a evaluar U
    Salida: Conjunto de atributos Padres e Hijos respecto al atributo objetivo
2:      –Fase 1 de MMPC
3:      –Fase de Generacin de Subconjuntos
4:      < SubconjuntosDeAtributos >← GenerarSubconjuntos(U)
5:      –Fase 1 de MMPC
6:      SUPERCPC = []
7:      repetir
8:          CPC = []
9:          repetir
10:             < F, assocF >← MaxMinH(T,CPC, subU)
11:             si assocF != 0 entonces:
12:                 CPC = CPC U F
13:             fin si
14:         hasta que el CPC deje de ser modificado
15:         SUPERCPC = SUPERCPC U CPC
16:     por cada subU en SubconjuntosDeAtributos
17:     retornar SUPERCPC
18:
19:
20: procedure MaxMinH
    Entrada: T, CPC, U
    Salida: Atributo A con mayor asocin del conjunto U
21:         arregloDeAttEvaluados = []
22:     por cada atributo X en U:
23:             arregloDeAttEvaluados = arregloDeAttEvaluados U MinAsoc(X,
    T, CPC, U)
24:     fin por
25:     < assocF, F >= max(arregloDeAttEvaluados)
26:     retornar < F, assocF >
```

Figura 14. Algoritmo SUPERCPC propuesto como solución a la complejidad superexponencial de la fase de poda o descarte del MMPC clásico.

- Estos resultados fueron comparados con el uso de los Mantos de Markov “reales”, así como el conjunto de padres e hijos del Manto de Markov verdadero, para conocer la calidad del Manto de Markov respecto a la predicción de su clase objetivo, y la diferencia de calidad en caso no se incluyan los atributos esposos (es decir, solo se use los padres e hijos de la clase objetivo).

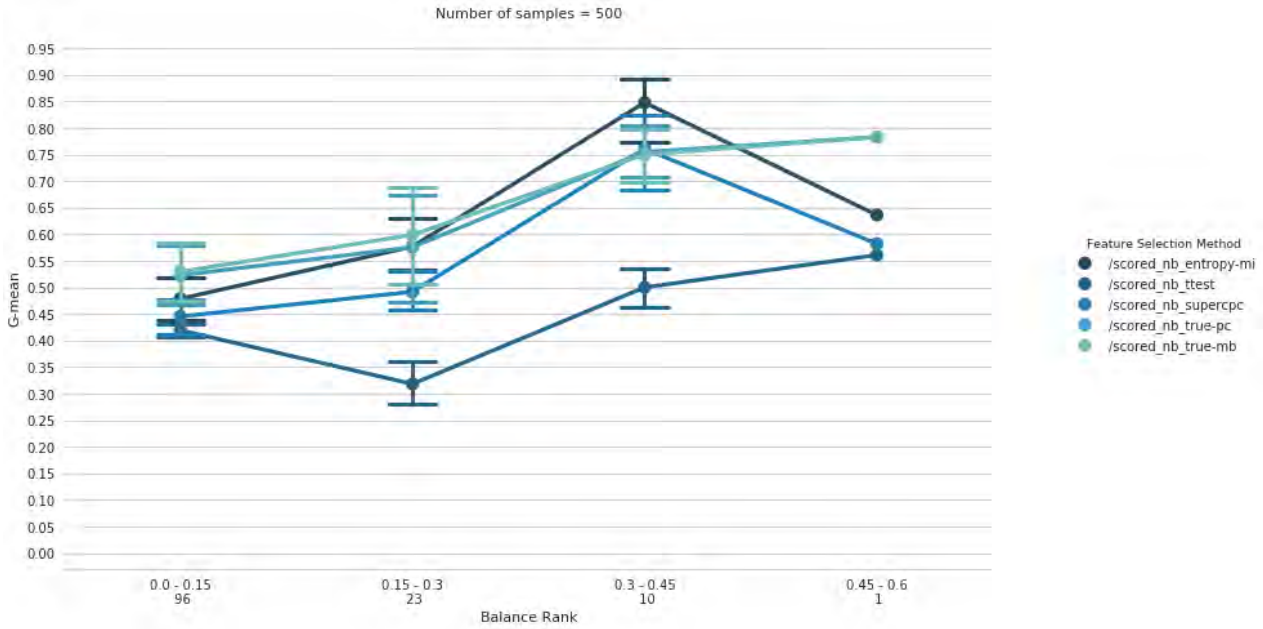

Figura 15. Resultados de en análisis de muestras de 500 instancias de Alarm.

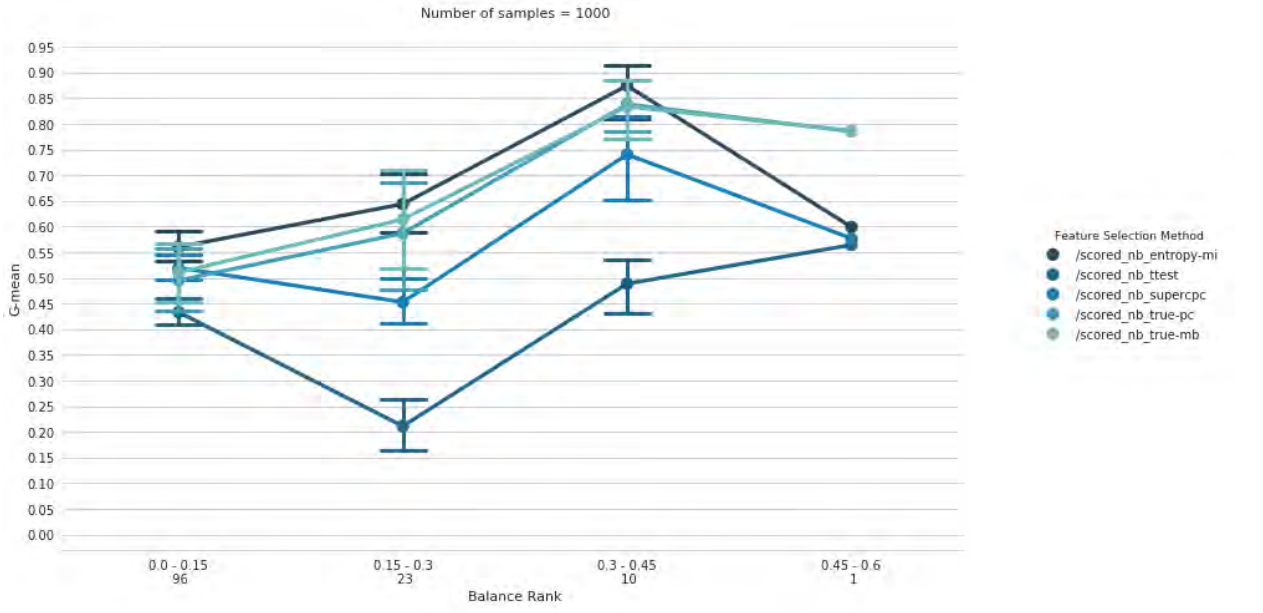

Figura 16. Resultados de en análisis de muestras de 1000 instancias de Alarm.

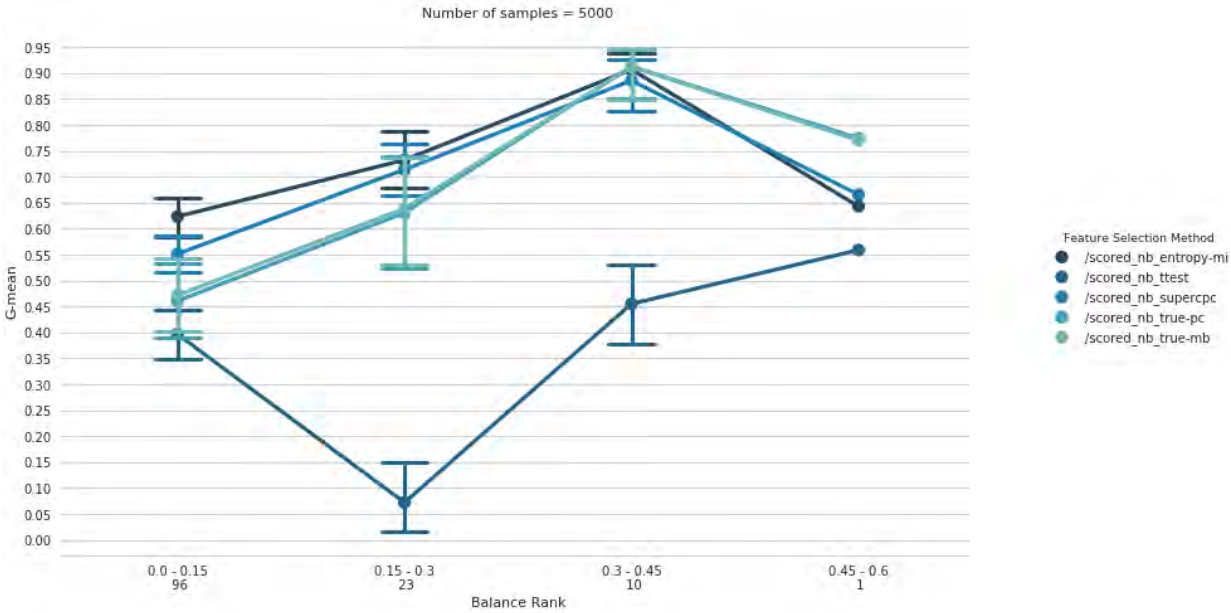

Figura 17. Resultados de en análisis de muestras de 5000 instancias de Alarm.

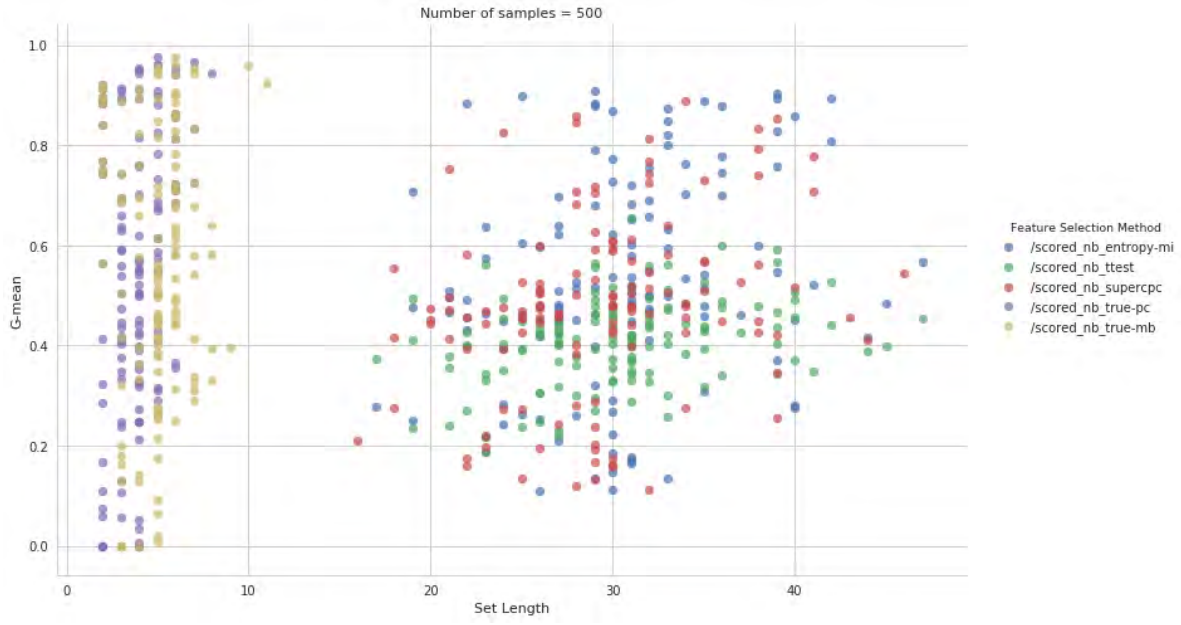

Figura 18. Resultados desagregados de las muestras de 500 instancias.

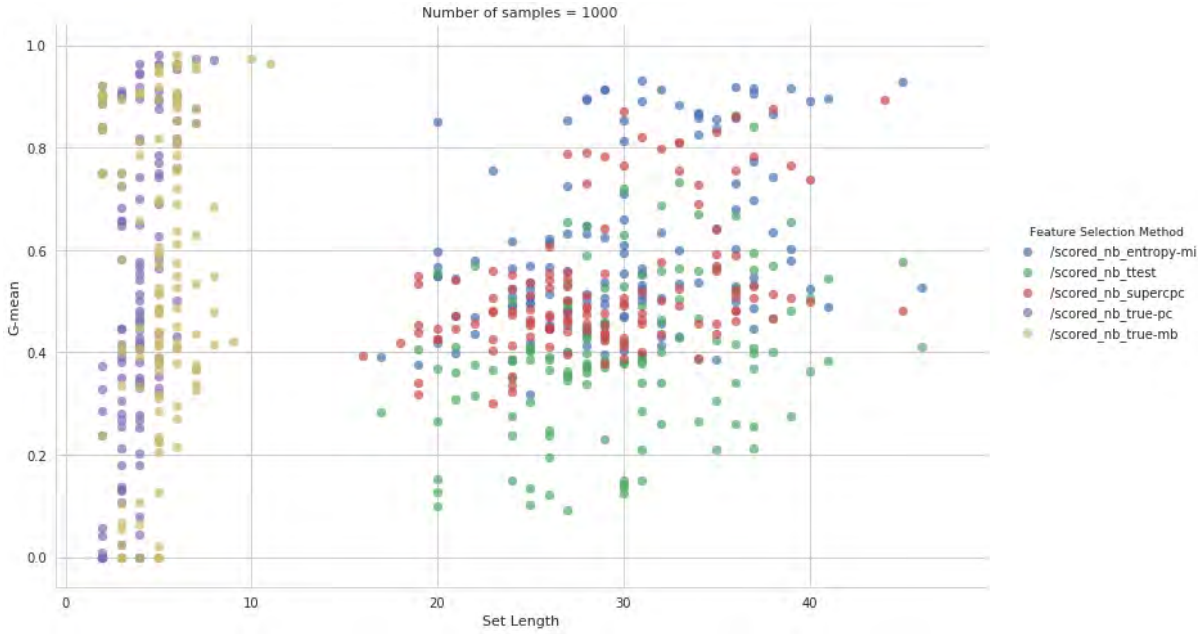

Figura 19. Resultados desagregados de las muestras de 1000 instancias.

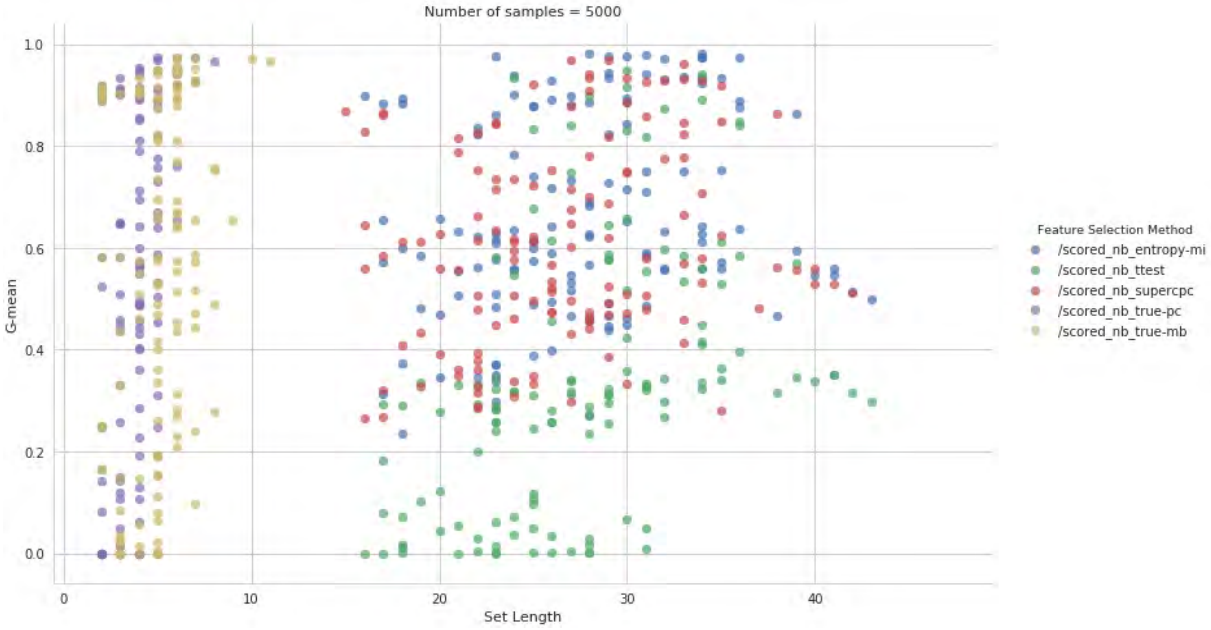

Figura 20. Resultados desagrados de las muestras de 5000 instancias.

- Las conclusiones que se obtuvieron de esta experimentación fueron las siguientes:
    - El *supercpc*, obtenido a través del algoritmo propuesto en esta sección, contiene cantidad de información similar a los métodos de selección de atributos clásicos con los que se comparó.
    - El Manto de Markov, como conjunto de atributos predictivo de la clase objetivo, obtiene resultados óptimos a pesar del tamaño mínimo de atributos que contiene, en comparación a los demás métodos.
    - La consideración de los atributos esposos no afecta de manera estadísticamente relevante la calidad del conjunto de atributos como predictor.

5.3.4 Sensibilidad al orden de los estimadores del Manto de Markov

Durante la experimentación del acápite anterior, así como en la revisión de la teoría, se pudo apreciar que la fase de crecimiento del MMPC es sensible al orden en que los atributos estén dispuestos dentro del conjunto de datos.

Para analizar con mayor profundidad el impacto de esta característica del MMPC en la calidad de los conjuntos resultantes, se realizó el análisis de la varianza de la distancia entre el conjunto hallado y el conjunto “verdadero”, bajo la siguiente fórmula, mencionada por E. Villanueva:

$$\sqrt[2]{(1 - truePositiveRateMB)^2 + (1 - trueNegativeRateMB)^2} = distance$$

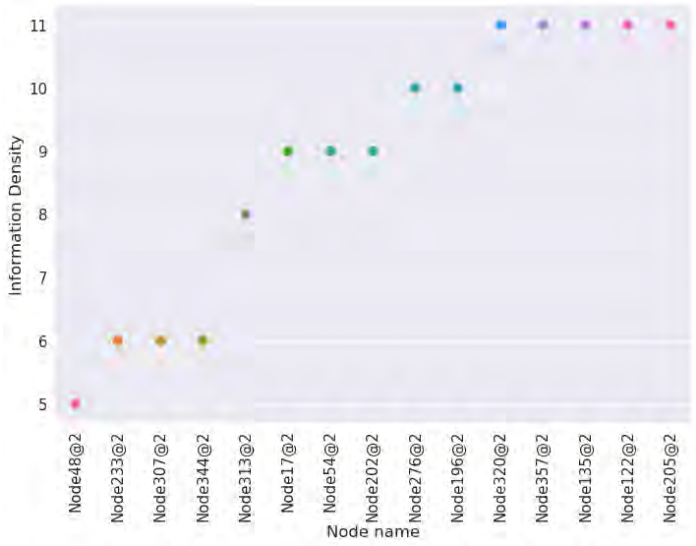

Figura 21. Gráfico de primero 15 nodos ordenados por sus nodos de segundo grado (es decir, son parte del Manto de Markov de un nodo del Manto de Markov del nodo objetivo)

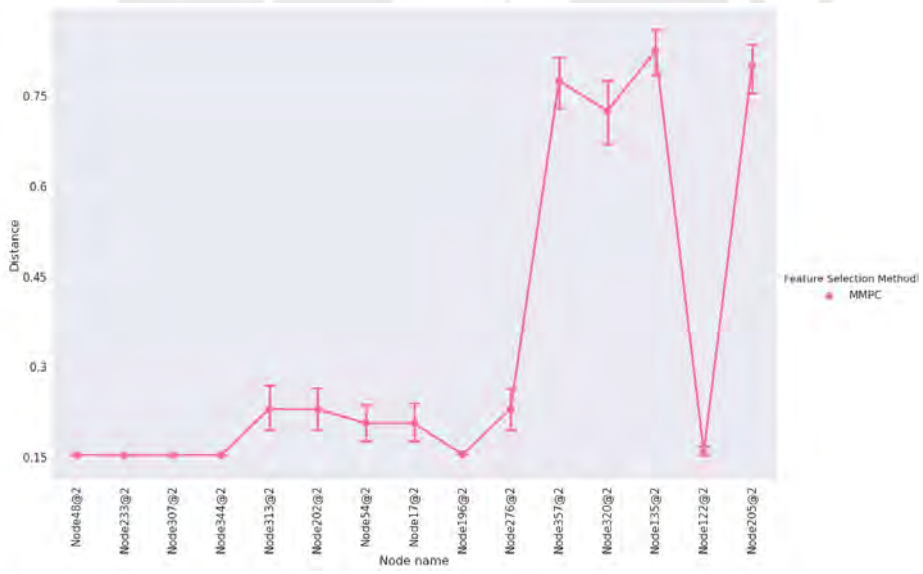

Figura 22. Progreso de distancia y varianza de la distancia obtenidos por el MMPC, por cada nodo

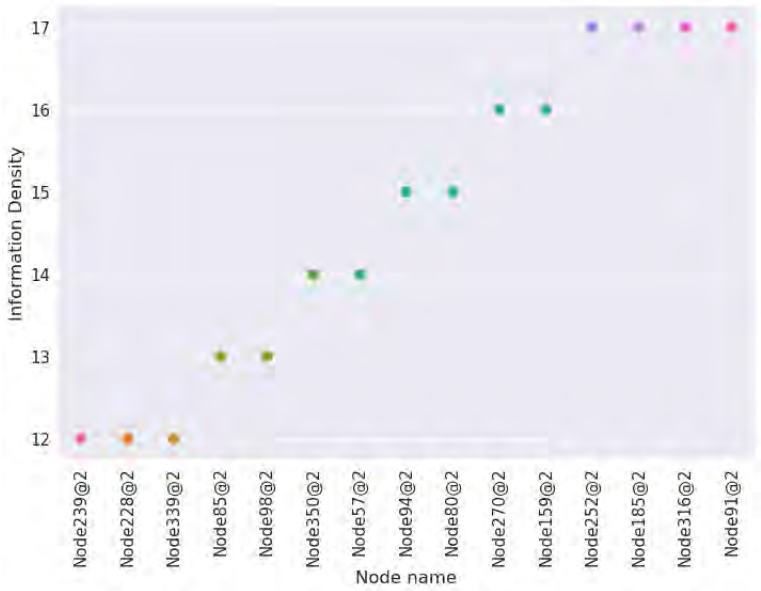

Figura 23. Progreso de cantidad de nodos de segundo grado de los siguientes 15 nodos (ordenados bajo este mismo criterio)

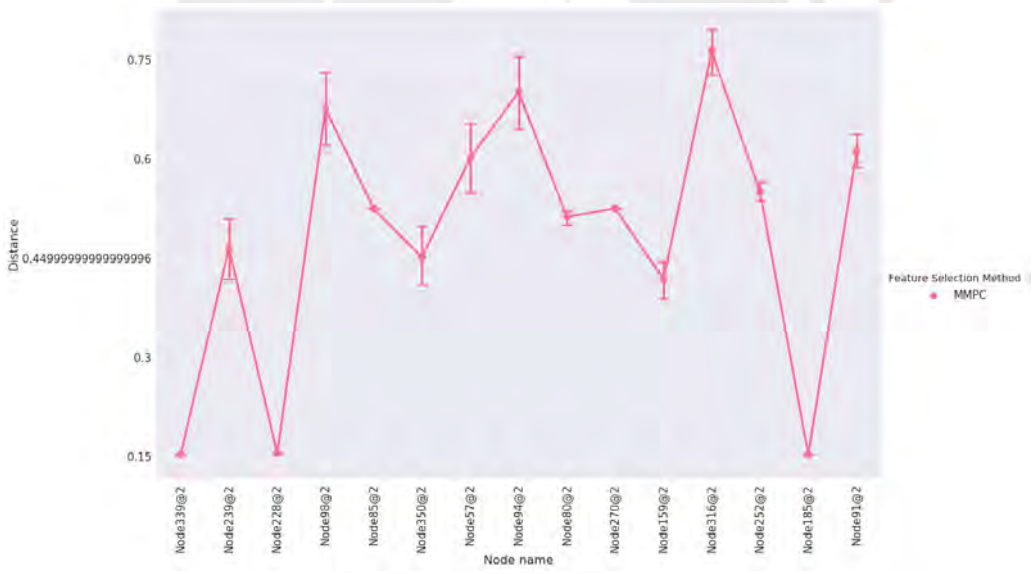

Figura 24. Progreso de distancia y varianza de la distancia obtenidos por el MMPC, por cada nodo

De estos resultados, se pudo concluir que la sensibilidad de estimadores del Manto de Markov como MMPC, puede impactar negativamente en los resultados obtenidos, tomando en cuenta la cantidad de atributos con información redundante que estén fuertemente relacionados con la clase objetivo, y entre sí. También se observó que el rendimiento del MMPC se vuelve errático conforme el número de nodos de segundo grado incrementa.

5.3.5 Diseño del algoritmo propuesto MMRWPC

El diseño para el nuevo algoritmo propuesto toma las características de los siguientes algoritmos clásicos:

- Descarte entrelazado: al final de la adición de un grupo de nodos en el conjunto de padres e hijos candidatos, se buscará separadores entre los atributos a fin de evitar cálculos posteriores con atributos no relevantes para el conjunto "Padres e Hijos" buscado por el algoritmo. La idea es mantener el conjunto de padres e hijos candidatos lo más relevante y no redundante posible, para mantener la cantidad de cálculos posteriores al mínimo posible.

- Heurística Máximo-Mínimo: la evaluación de asociación entre dos nodos se ejecutará, de manera nuclear, igual que en el algoritmo MMPC.

- Caminata Aleatoria: se usará este algoritmo como una aproximación de clustering espectral para la agrupación de atributos similares entre sí dentro del grupo candidato, para disminuir la cantidad de cálculos realizados durante la búsqueda de nodos relevantes y de separadores, respectivamente. La elección de este algoritmo está respalda por un estudio realizado sobre la selección de atributos para el progreso de enfermedades, enfocado en el cáncer y el VIH [41].

Bajo estas nuevas características, el algoritmo propuesto es el mostrado en Algorithm 1 (Figura 24).

Algorithm 1 Algoritmo MMRWPC

```
1: procedure MMPC
   Entrada: Atributo objetivo T, Conjunto de atributos a evaluar U
   Salida: Conjunto de atributos Padres e Hijos respecto al atributo objetivo
2:     –Fase 1
3:     CPC = []
4:     repetir
5:         ArrAsoc[< Fi, AssocFi >] ← MaxMinH(T, CPC, U)
6:         por cada X en ArrAssoc:
7:             si X < assocF >> 0 :
8:                 CPC = CPC U X < F >
9:             fin si
10:        fin por
11:        –Fase 2 entrelazada en Fase 1
12:        CPC ← FiltrarPorSeparador(T,CPC, U)
13:    hasta que el CPC deje de ser modificado

14:    retornar CPC
15:
16:
17: procedure MaxMinH
   Entrada: T, CPC, U
   Salida: Atributo A con mayor asociacion del conjunto U
18:        arregloDeAttEvaluados = []
19:    por cada atributo X en U:
20:            arregloDeAttEvaluados = arregloDeAttEvaluados U MinAsoc(X,
   T, CPC, U)
21:    fin por
22:    retornararregloDeAttEvaluados
23:
24:
25: procedure FiltrarPorSeparador
   Entrada: T, CPC
   Salida: CPC filtrado
26:    -Fase de agrupacion:
27:    agrupacionesCPS = []
28:    mientras CPC no vacio :
29:        NodoCentroide = EscogerAleatoriamenteNodo(CPC)
30:        agrupacion = CaminataAleatoria(NodoCentroide, CPC)
31:        agrupacionesCPC = agrupacionesCPC U agrupacion
32:    fin mientras
33:    -Fase de filtracion:
34:    por cada agrupacion en agrupacionesCPC:
35:        agrupacion = BuscarSeparador(T,agrupacion)
36:    fin por
37:    retornar agrupacionesCPC
```

Figura 24. Pseudocódigo del algoritmo propuesto MMRWPC. Elaboración propia.

5.3.6 Desempeño del algoritmo propuesto en comparación a la implementación clásica MMPC

Usando el conjunto de datos artificiales de “Alarm” [39], popular benchmark en la medición de desempeño entre algoritmos de estimación del Manto de Markov y otros estimadores de Redes Bayesianas, se utilizó un grupo de 5 nodos arbitrariamente elejidos para la evaluación. Se buscó evaluar el tiempo necesitado para la ejecución y la “distancia” del conjunto resultante de los algoritmos respecto al Manto de Markov real de cada nodo. Asimismo, se incluyeron permutaciones en el orden de los sets de datos para apreciar la sensibilidad al orden de atributos en los algoritmos evaluados.

Los resultados evidencian la plausibilidad del algoritmo propuesto MMRWPC en reducir la distancia de los mantos de Markov estimados en relación a los mantos de Markov reales, ofreciendo una mayor estabilidad en la estimativa (menor desviación estandard) en relación

Node	Algorithm	Mean Time	Max Time	Min Time	StdDev Time
Node14	MMPCOPT	0.9977	3.2227	0.2606	0.7027
Node5	MMPCOPT	3.7893	7.7171	1.4247	1.4146
Node351	MMPCOPT	0.4805	0.7836	0.2923	0.1547
Node169	MMPCOPT	0.2946	0.5379	0.1825	0.0839
Node31	MMPCOPT	0.0587	0.1107	0.0562	0.0057
Node14	MMRWPC	1.4814	5.1903	0.3860	1.0287
Node5	MMRWPC	4.3143	10.1978	2.1259	1.8744
Node351	MMRWPC	0.6327	1.0919	0.2966	0.2117
Node169	MMRWPC	0.3032	0.3627	0.2515	0.0312
Node31	MMRWPC	0.0618	0.1262	0.0579	0.0075

Tabla 7: Rendimiento temporal del algoritmo MMRWPC vs MMPCOPT (Algoritmo clásico del MMPC optimizado exclusivamente de manera computacional); para propósitos ilustrativos, se han incluido los valores máximos (Max Distance) y mínimos (Min Distance) de cada nodo analizado, así como la desviación estándar del análisis (StdDev). (Elaboración propia)

al clásico MMPC, siendo casi inmune al ordenamiento de atributos del banco de datos. Se observa también que el incremento de tiempo computacional debido a la caminata aleatoria no representa valores significativos.

Node	Algorithm	Mean Distance	Max Distance	Min Distance	StdDev Distance
Node14	MMPCOPT	0.0001	0.0027	0.0000	0.0005
Node5	MMPCOPT	0.2018	0.5006	0.0109	0.1419
Node351	MMPCOPT	0.1092	0.3335	0.0054	0.1504
Node169	MMPCOPT	0.0058	0.0109	0.0055	0.0010
Node31	MMPCOPT	0.0000	0.0000	0.0000	0.0000
Node14	MMRWPC	0.0046	0.0137	0.0000	0.0043
Node5	MMRWPC	0.2512	0.2525	0.2505	0.0004
Node351	MMRWPC	0.0090	0.0136	0.0054	0.0020
Node169	MMRWPC	0.0055	0.0055	0.0055	0.0000
Node31	MMRWPC	0.0000	0.0000	0.0000	0.0000

Tabla 8: Rendimiento respecto a la distancia entre el set recuperado y el verdadero Manto de Markov del nodo objetivo; para propósitos ilustrativos, se han incluido los valores máximos (Max Distance) y mínimos (Min Distance) de cada nodo analizado, así como la desviación estándar del análisis (StdDev). (Elaboración Propia)

5.3.7 Esquema de aplicación de algoritmo propuesto y otros selectores de atributos sobre las muestras de entrenamiento

Se ha seleccionado los métodos selectores de atributo de T-test, asumiendo y sin asumir correlación entre atributos analizados (librería “Scipy”), y Entropía (librería “Sci-kit Learn”), para la comparación de desempeño final, dado su uso en estudios anteriores de estimadores de manto de Markov [38].

Capítulo 6.Diseño del algoritmo genético adaptivo

6.1 Introducción

En este capítulo, se presentará la implementación propuesta de algoritmo genético para la filtración del total de atributos relevantes obtenidos por el análisis de las muestras generadas con el algoritmo propuesto de estimación del Manto de Markov, propuesto en el Capítulo 5. Asimismo, se describirá las principales problemáticas que desea atacar esta implementación.

6.2 Descripción del resultado

Se ha implementado un algoritmo que busca filtrar los atributos mencionados anteriormente, a partir de la frecuencia de aparición dentro del conjunto candidato de padres e hijos en cada muestra analizada. La aplicación de este algoritmo proveerá de un grupo de conjuntos de atributos a evaluar durante el capítulo final. El algoritmo utilizará la concatenación de las muestras generadas para cada inhibidor de prueba, siendo 100 la cantidad escogida para este proyecto de fin de carrera. Entonces, los conjuntos de entrenamiento del algoritmo genético contarán con 600,000 filas, con grado de balance del 100% para hacer énfasis en la clase positiva de la variable objetivo.

6.3 Desarrollo del resultado

6.3.1 Enfoque a usar en generación de población inicial del algoritmo genético

Por cada grupo de muestras analizadas para un experimento en particular, se calculó la relevancia de los atributos en base al número de apariciones en los conjuntos resultantes del análisis con el algoritmo estimador del Manto de Markov.

Asimismo, se estableció distintos límites de “mínima frecuencia” para limitar aún más la población inicial del algoritmo genético.

6.3.2 Implementación y adaptabilidad del algoritmo genético

Dado que uno de los objetivos principales de este proyecto de fin de carrera es alcanzar eficiencia en los algoritmos aplicados, la implementación de este algoritmo abarcó las siguientes características:

- Diversidad a través de incremento de probabilidad de mutación: característica usualmente estática en la implementación del algoritmo, permite la consideración de individuos con un gen de diferencia sin descartar al gen original, aportando mayor diversidad al grupo de hijos resultantes.

- Diversidad a través de erradicación de la población actual, con excepción del individuo más "fuerte" (es decir, con mejor fitness): A través de esta característica, el algoritmo tiene mayor probabilidad de escapar de mínimos locales alcanzados en etapas iniciales de su ejecución.

Las características principales del algoritmo presentado en la Figura 25 son las siguientes:

- A través de ***AumentarProbabilidadMutación()*** , se aumenta la potencial diversidad a obtener de los individuos, en cada iteración del algoritmo.

- Usando ***ReiniciarProbabilidadMutacion(), EliminarPoblacionSinElMejor(Población)*** y ***GenerarNuevaPoblacion()***, se implementa la técnica de erradicación total de la población, dejando como sobreviviente al mejor individuo, para obtener mayor diversidad de individuos dentro del algoritmo.

- Usando ***EvaluarPoblacion(****Población+Hijos+Mutaciones****)*** se obtiene el fitness promedio de los mejores individuos, medida que se usa como medida de referencia para conocer la calidad de la población.

Algorithm 1 Algoritmo Gentico

```
1: procedure AGA
   Entrada: Atributo objetivo T, Conjunto de atributos a evaluar U, Numero de iteraciones N
   Salida: Conjunto de atributos relevantes respecto al atributo objetivo
2:     –Fase Inicial
3:     Poblacion ← GenerarPoblacionInicial U)
4:     repetir
5:         si ProgesoPromedioPoblacion > Alpha :
6:             AumentarProbabilidadMutacion()
7:             si VecesProgresoPromedioMenor > 3 :
8:                 ReiniciarProbabilidadMutacion()
9:                 EliminarPoblacinSinElMejor(Poblacion)
10:                GenerarNuevaPoblacion(U)
11:            fin si
12:        fin si
13:        Poblacion ← EvaluarPoblacion(Poblacion, T)
14:        ParesPadres ← TorneoPoblacion(Poblacion)
15:        Hijos ← CruzarPadres(ParesPadres)
16:        Mutaciones ← Mutaciones(Poblacion+Hijos)
17:        Poblacion, PrecisionPromedio ← EvaluarPoblacion(Poblacion + Hijos + Mutaciones)
18:        ProgresoPromedioPoblacion = PrecisionPromedio - PrecisionPromedioAnterior
19:    hasta que nIteraciones sea igual que N

20:    retornar MejorIndividuo
```

Figura 25. Pseudocódigo de algoritmo genético implementado para la filtración de los atributos obtenidos al usar el algoritmo propuesto MMRWPC. Elaboración propria.

Capítulo 7.Selección de modelos de clasificación

7.1 Introducción

En este capítulo, se presentará la selección y adaptación de modelos de clasificación basados en Aprendizaje Máquina, para la medición de precisión de los conjuntos de atributos relevantes obtenidos por la aplicación de los algoritmos propuestos en capítulos anteriores.

7.2 Descripción del resultado

Se ha seleccionado y adaptado modelos de clasificación que se encontraron relevantes para la medición de precisión de los conjuntos de atributos relevantes obtenidos. Los criterios principales para escogerlos fueron: eficiencia, distancia respecto a la forma de clasificación que estos modelos efectúan y relevancia en el estado del arte.

7.3 Desarrollo del resultado

7.3.1 Selección de modelo Gaussiano

Se seleccionó un modelo de Naive Bayes Gaussiano de la librería “Sci-kit Learn” en base los siguientes criterios:

- ligereza de procesamiento y resultados robustos respecto al tiempo invertido en el procesamiento.

- considera que los atributos son independientes entre sí, hecho que se relaciona bastante al objetivo de hallar el Manto de Markov: los atributos que le involucren deberían ser independientes entre sí, y contener toda la información existente (no redundante) del conjunto de características global acerca de la variable objetivo.

7.3.2 Selección de modelo de Regresión Logística

Se seleccionó un modelo de Regresión Logística de la librería “Sci-kit Learn” por los siguientes motivos:

- Su cálculo se utiliza en diversas aplicaciones, siendo una de ellas redes neuronales, las cuales logran obtener resultados robustos dependiendo de la cantidad de información que se les alimente.

- A través de teoría y experimentación, se halló que sus resultados eran extremadamente parecidos a los que se podría obtener a través de la aplicación de un modelo de Máquina de Vectores de Soporte.

- El bajo costo de procesamiento que toma para el entrenamiento del modelo con un número variado de atributos.

7.3.3 Selección de modelo de Árboles de Decisión

Se seleccionó el modelo de Árbol de Decisión de la librería “Sci-kit Learn” por los siguientes criterios:

- Su popularidad en los métodos de ensemble existentes en la actualidad (Random Forest, por ejemplo), así como su clara diferenciación en la forma de “clasificar” las filas: de manera similar a una red neuronal, este modelo calcula la probabilidad de que una fila pertenezca a una clase; en este momento difiere de una red neuronal, dado que la clasificación utiliza cada atributo como un discriminante en cadena para conocer si efectivamente pertenece a una clase o no. Esta manera de clasificar le vuelve atractiva para este proyecto de fin de carrera, dado que el algoritmo propuesto de estimación del Manto de Markov busca los atributos discriminantes e independientes entre sí que permitan reconocer adecuadamente la clase objetivo.

- El poco poder de procesamiento que toma para el entrenamiento del modelo con un número variado de atributos.

Capítulo 8. Grado de precisión y esfuerzo computacional de combinaciones “Selector de atributos – modelo clasificador”

8.1 Introducción

En este capítulo, se presentarán los resultados obtenidos de la aplicación de la metodología y algoritmos propuestos en los anteriores capítulos, en los datos a disposición.

8.2 Descripción del resultado

Los resultados obtenidos son la precisión (media geométrica) y el esfuerzo computacional (en segundos) que los algoritmos propuestos y los algoritmos de referencia han tomado para efectuar exitosamente el flujo explicado en la metodología propuesta.

8.3 Desarrollo del resultado

8.3.1 Resultados obtenidos por la aplicación del algoritmo basado en la estimación del Manto de Markov, bajo el primer enfoque

Respecto al esfuerzo computacional, la totalidad de algoritmos de referencia tomaron menos de 5 segundos en promedio para ejecutarse en la totalidad de cada muestra (que contiene 6000 registros de 541 atributos cada uno, incluyendo el atributo objetivo de si la interacción génica es causal o no); mientras que el algoritmo propuesto de estimación del Manto de Markov tomó 2 minutos en promedio.

Respecto al grado de precisión obtenida, se tomaron mediciones respecto al número de atributos incluidos con mayor puntaje, luego de haber ponderado los resultados obtenidos en cada muestra, por cada método selector de atributos.

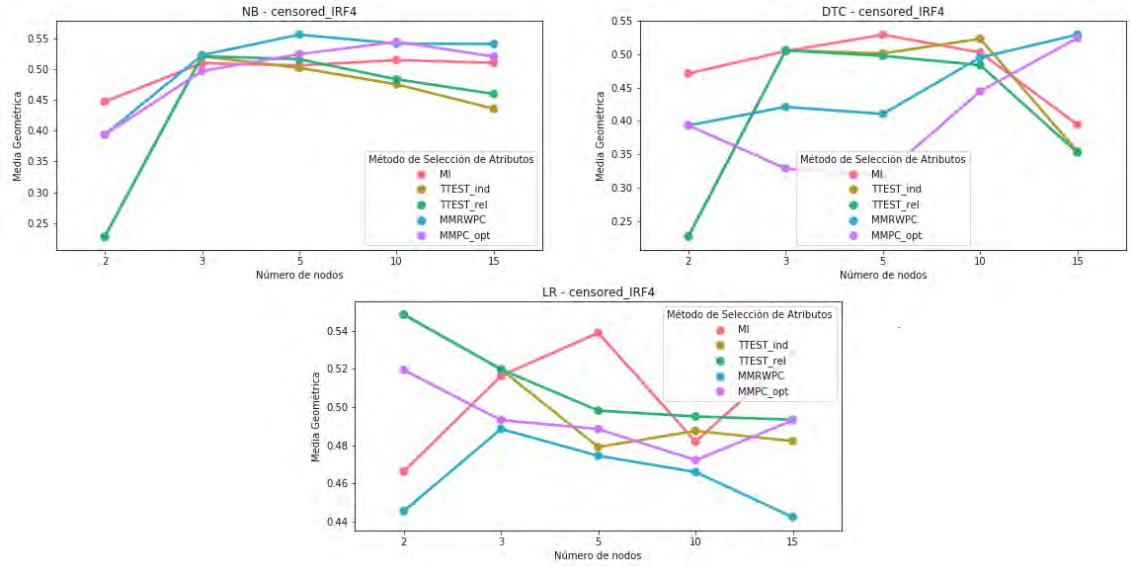

Figura 26. Resultados obtenidos censurando las instancias que usaron el inhibidor IRF4. Elaboración propia.

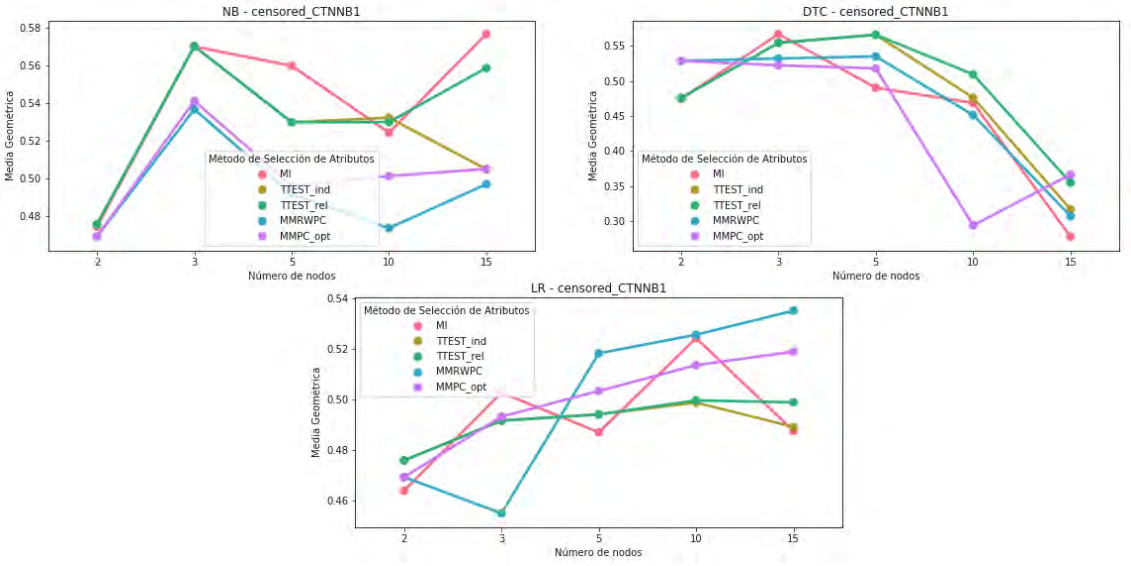

Figura 27. Resultados obtenidos censurando las instancias que usaron el inhibidor CTNNB1. Elaboración propia.

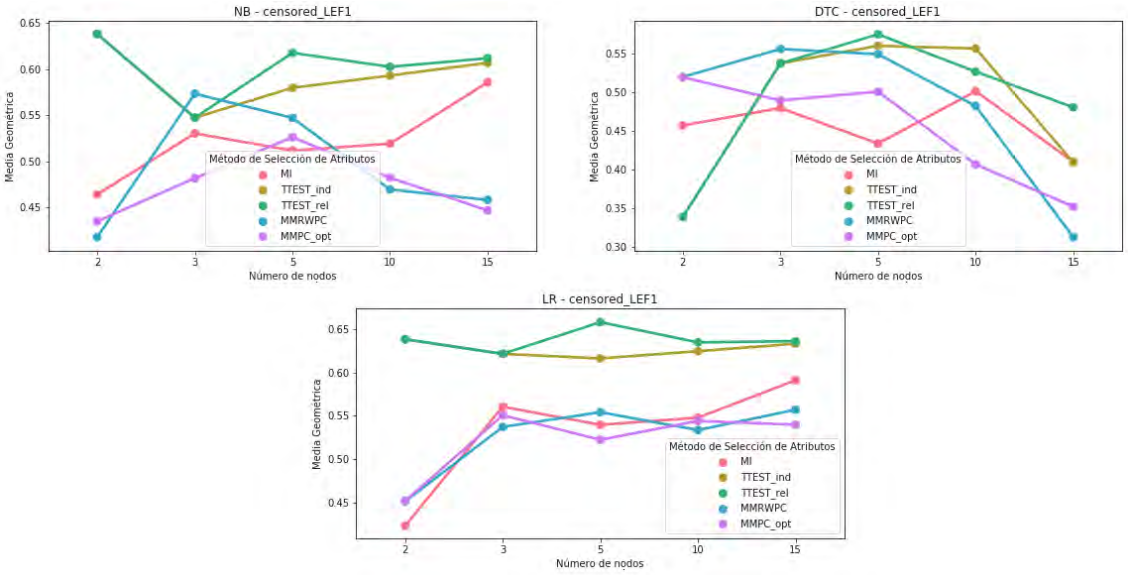

Figura 28. Resultados de precisión, obtenidos censurando las instancias que usaron el inhibidor LEF1. Elaboración propia.

En la figura 28, se puede apreciar que el rendimiento del algoritmo propuesto excede el del resto en 2 ocasiones (NaiveBayes – censored IRF4, LogisticRegression - CTNNB1), mientras que en el resto permanece cercano al promedio, manteniendo una tendencia parecida de progreso. Conociendo que el experimento CTNNB1 tiene 0.0732% de balance sobre positivos, se puede decir que el rendimiento errático del algoritmo propuesto es consecuencia del poco número de casos positivos a evaluar, en comparación de los negativos.

En las figuras 29 y 30, se puede observar que los métodos MMPC(MMPC_opt) y MMRWPC mantienen la misma tendencia a través de la adición de mayor número de atributos al conjunto seleccionado, siendo el MMRWPC el método que obtiene caídas menos abruptas, respecto a MMPC.

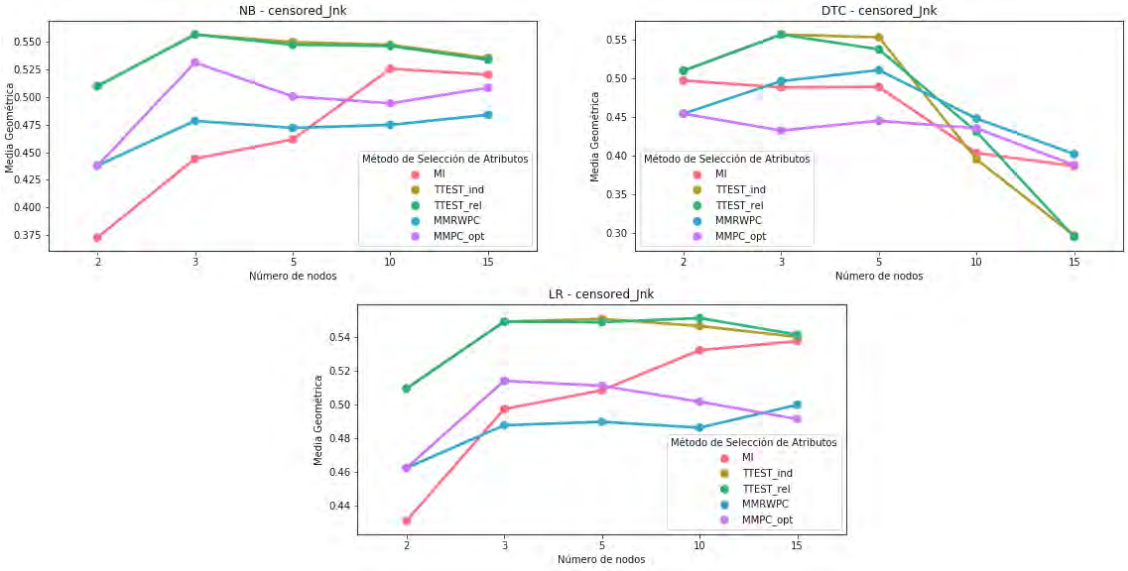

Figura 29. Resultados obtenidos censurando las instancias que usaron el inhibidor Jnk. Elaboración propia.

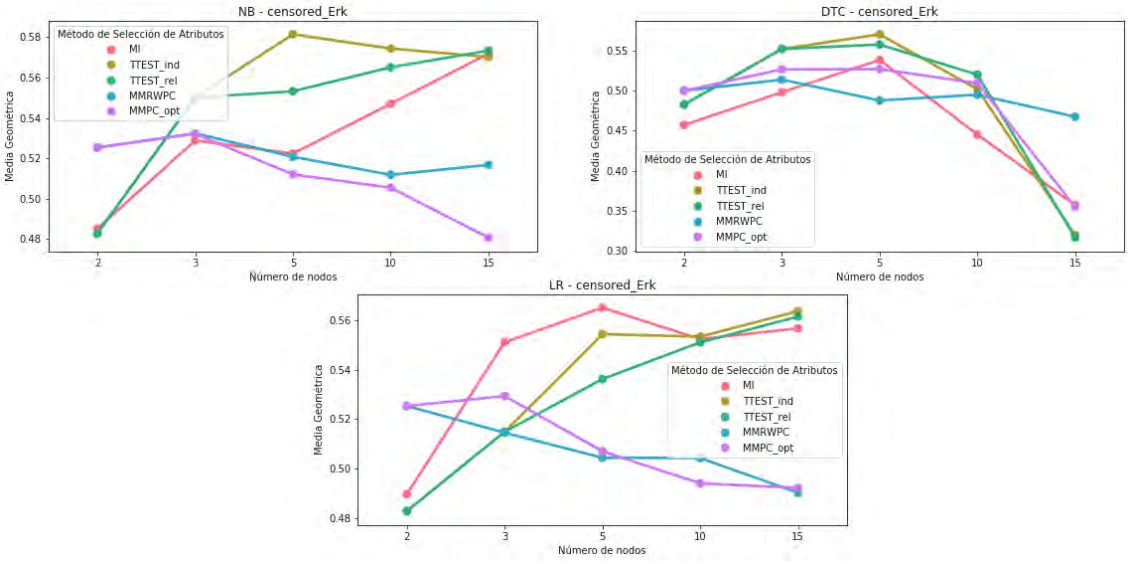

Figura 30. Resultados obtenidos censurando las instancias que usaron el inhibidor Erk. Elaboración propia

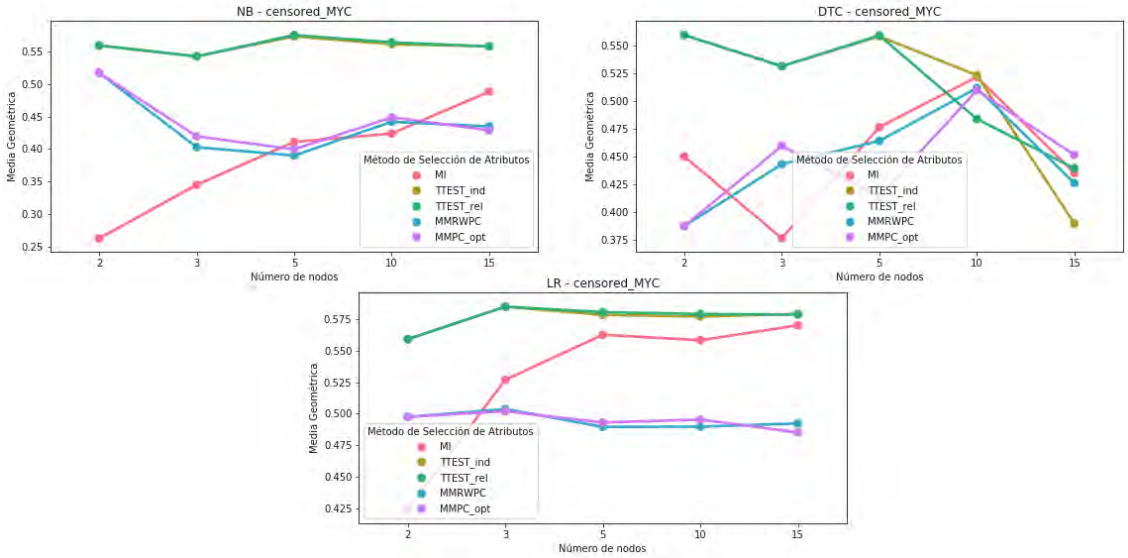

Figura 31. Resultados obtenidos censurando las instancias que usaron el inhibidor MYC. Elaboración propia.

En la Figura 31 también se logra apreciar que los resultados no son concluyentes, puesto que, según el clasificador, un método selector de atributo rinde mejor que el resto y no se logra apreciar una tendencia clara. Se puede atribuir esto a que los experimentos contenidos en esta imagen también constan de poco grado de balance.

En la Figura 32, nuevamente se aprecia que existen tendencias por clasificador de incremento o decremento de precisión, pero no de manera panorámica, respecto a la media geométrica obtenida. En particular, se puede observar que la combinación “Árbol de Decisión - MMRWPC” funciona de manera óptima a través de todos los conjuntos analizados, con excepción de aquellos con grado de balance menor al 0.2%

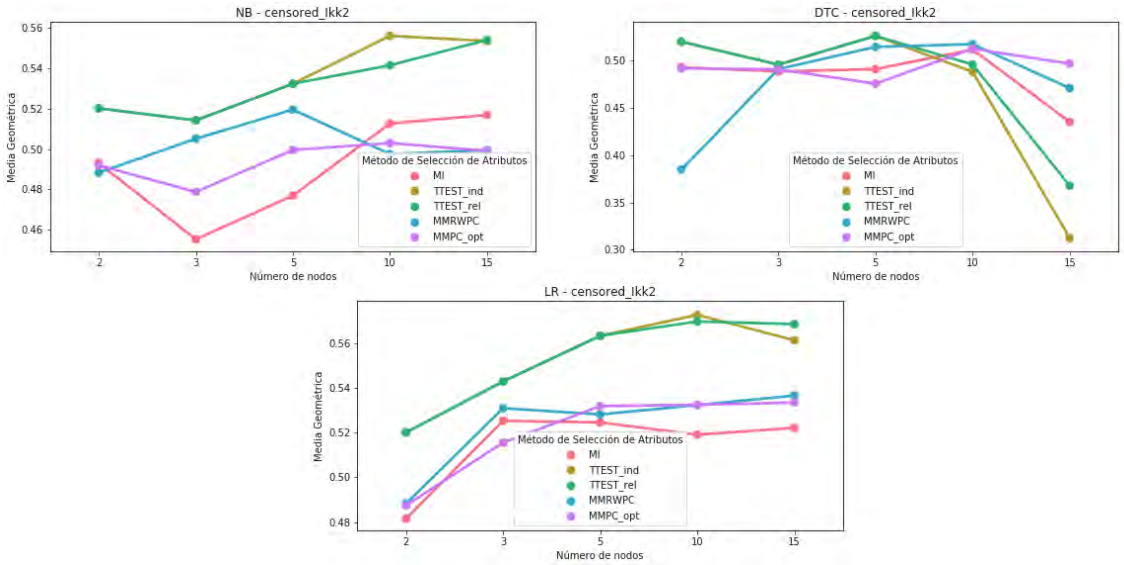

Figura 32. Resultados de precisión obtenidos al aplicar los selectores de atributos, el estimador del Manto de Markov y los modelos clasificadores seleccionados en dataset IKK2. Elaboración propia.

8.3.2 Resultados obtenidos del segundo enfoque, por la filtración adicional de atributos usando el algoritmo genético propuesto

A continuación, se presentan los resultados obtenidos de la aplicación del algoritmo genético propuesto en los conjuntos filtrados por el algoritmo de estimación del Manto de Markov propuesto, en tanto se buscaba realizar un filtrado adicional de atributos, en búsqueda de obtener grupos más robustos y discriminantes respecto a la clase objetivo. Este algoritmo tomó 5 minutos en promedio para procesar los conjuntos de atributos, siendo el conjunto de entrenamiento un set de 600,000 filas.

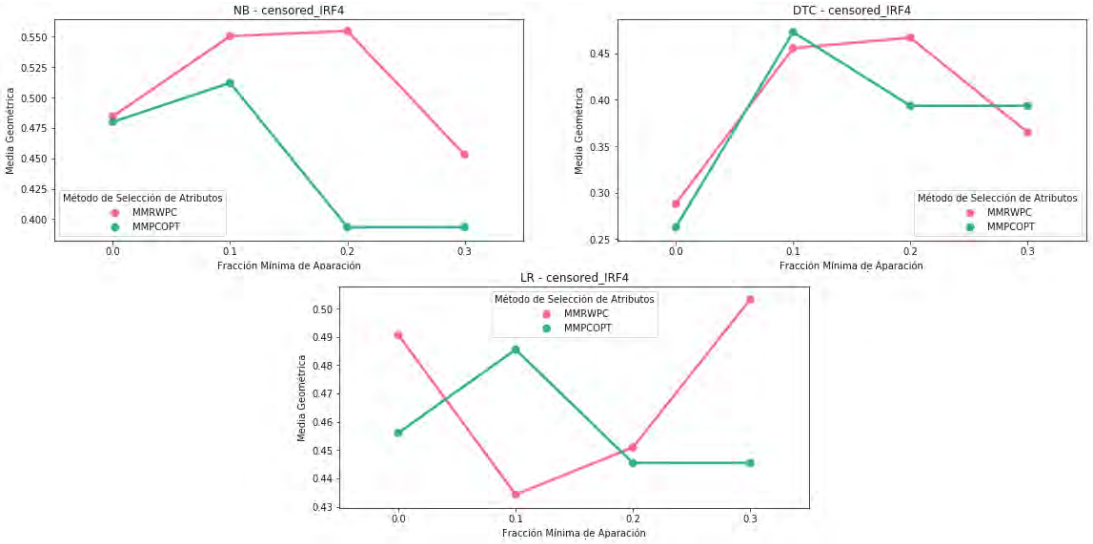

Figura 33. Resultados obtenidos censurando las instancias que usaron el inhibidor IRF4. Elaboración propia.

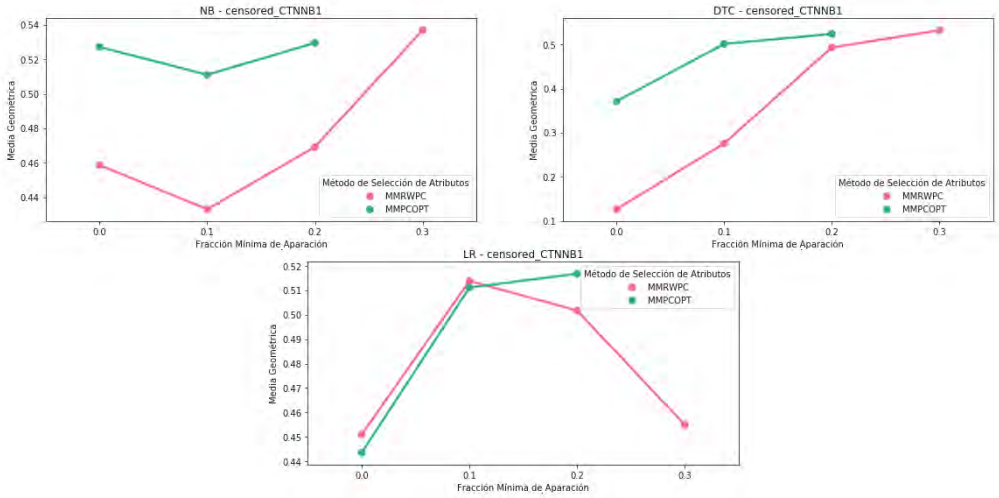

Figura 34. Resultados obtenidos censurando las instancias que usaron el inhibidor CTNNB1. Elaboración propia.

En la figura 33 y 34 se logra apreciar que, conforme crece el número de atributos a considerar como población inicial, el método MMRWPC posee parecida o mejor calidad (medida como la Media Geométrica obtenida) respecto al MMPC, en la mayoría de los casos (CTNNB1 – NB no ocurre esto).

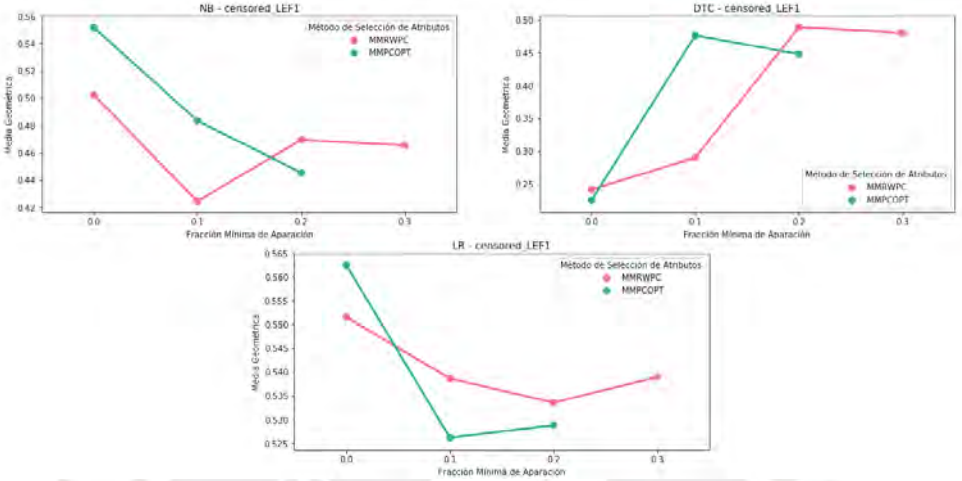

Figura 35. Resultados obtenidos censurando las instancias que usaron el inhibidor LEF1. Elaboración propia.

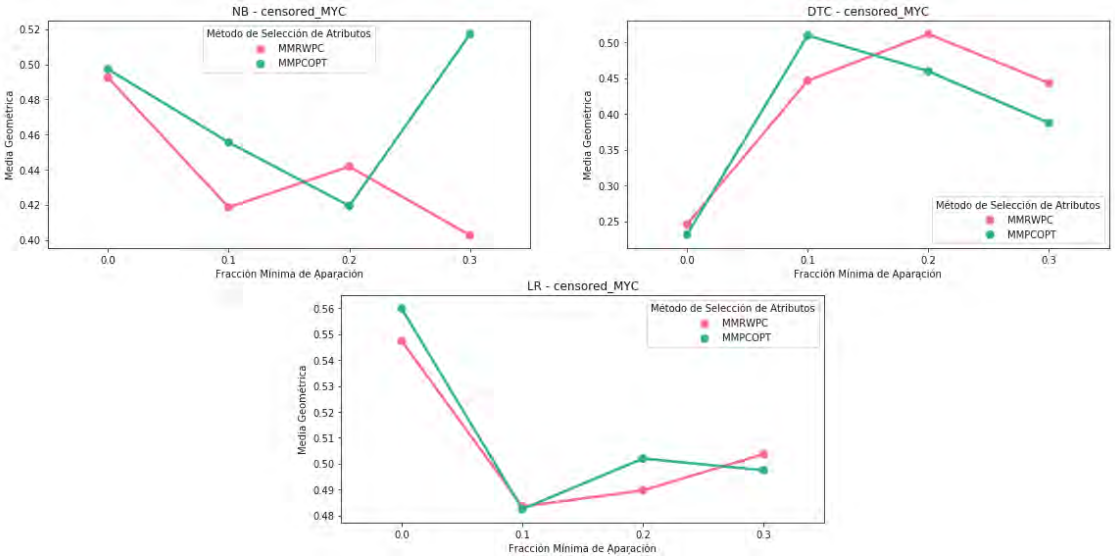

Figura 36. Resultados obtenidos censurando las instancias que usaron el inhibidor MYC. Elaboración propia.

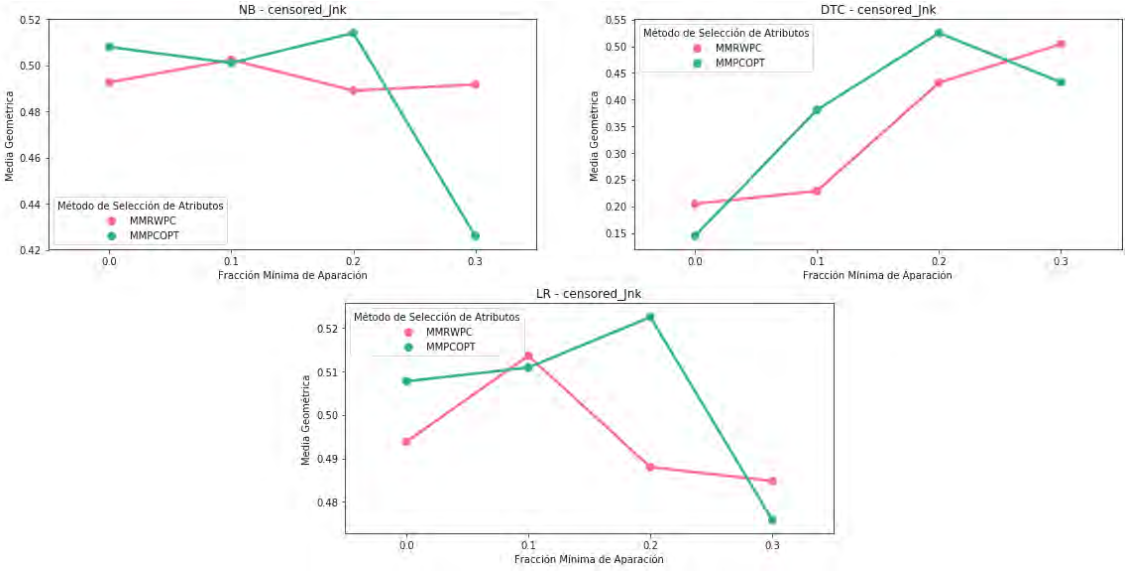

Figura 37. Resultados obtenidos censurando las instancias que usaron el inhibidor Jnk. Elaboración propia.

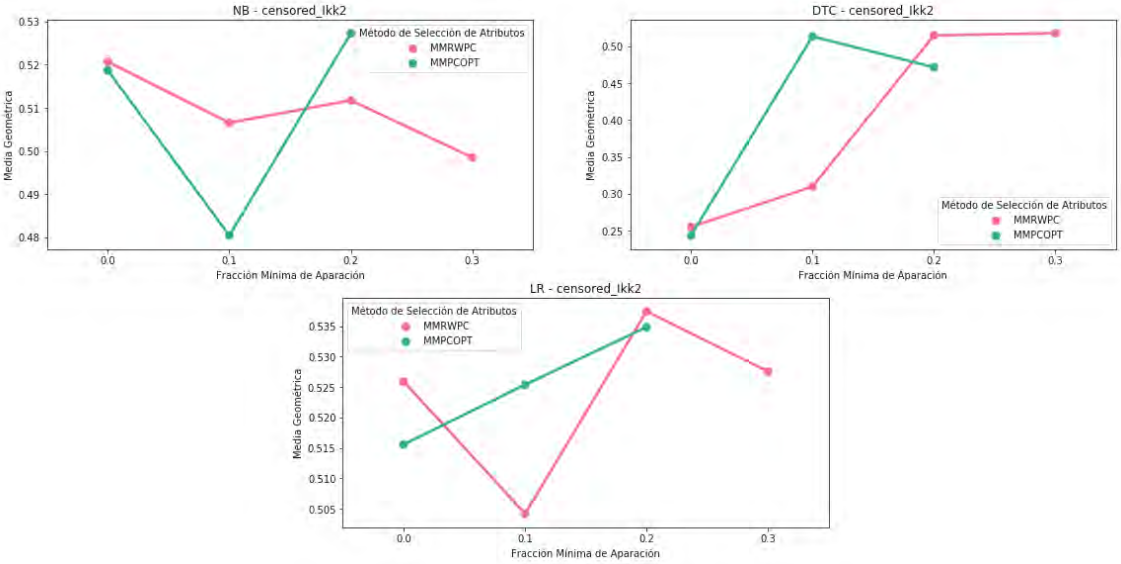

Figura 38. Resultados obtenidos censurando las instancias que usaron el inhibidor Ikk2. Elaboración propia.

A través de estos resultados, se ha podido apreciar que el porcentaje de aparición (es decir, el porcentaje mínimo de frecuencia de aparición de los nodos a incluir en el set de atributos predictores) donde la media geométrica empieza a converger es de 20%. Se debe tener en cuenta que los resultados presentados han sido de una sola ejecución de la metodología, y no se ha realizado experimentación numérica, lo cual podría arrojar resultados más robustos y alineados con lo esperado.

8.3.3 Resultados obtenidos del tercer enfoque, por la generación y entrenamiento de modelos predictivos por cada muestra

Se generó un modelo predictivo por cada Selector/Calificador de atributos, por cada muestra, por cada inhibidor a censurar. Los resultados han sido agrupados por inhibidor censurado y por método clasificador utilizado, por propósitos ilustrativos y de comparación (Figura 40 a Figura 46).

En las figuras 40 y 41 se logra apreciar la diferencia de variabilidad de los resultados de los métodos estimadores del Manto de Markov puestos a prueba. Asímismo, se puede observar la disputa entre MI y TTEST_rel como mejor calificador de atributos

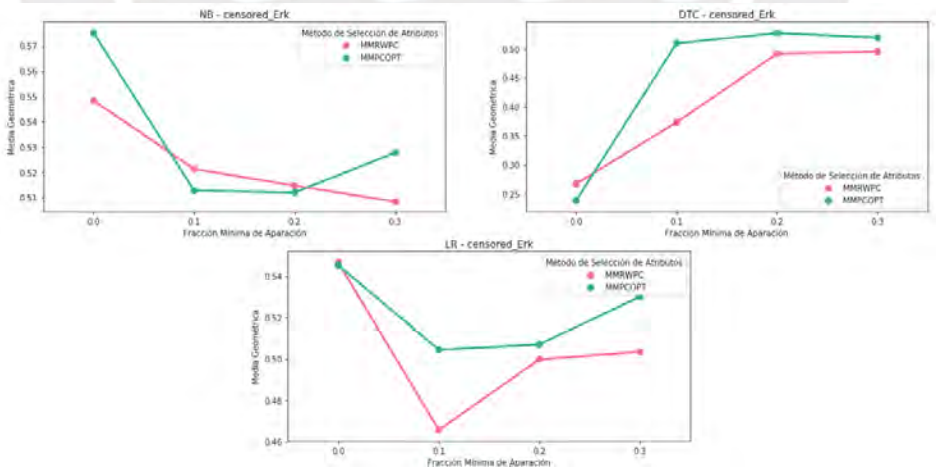

Figura 39. Resultados de precisión obtenidos al aplicar el algoritmo genético, censurando las instancias que usaron el inhibidor Erk. Elaboración propia.

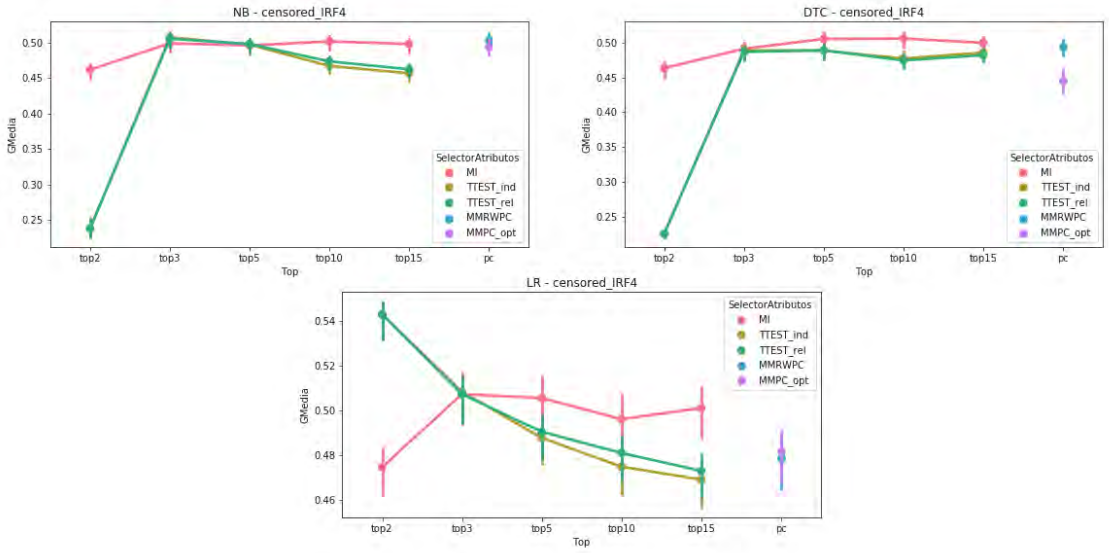

Figura 40. Resultados obtenidos censurando las instancias que usaron el inhibidor IRF4. Elaboración propia.

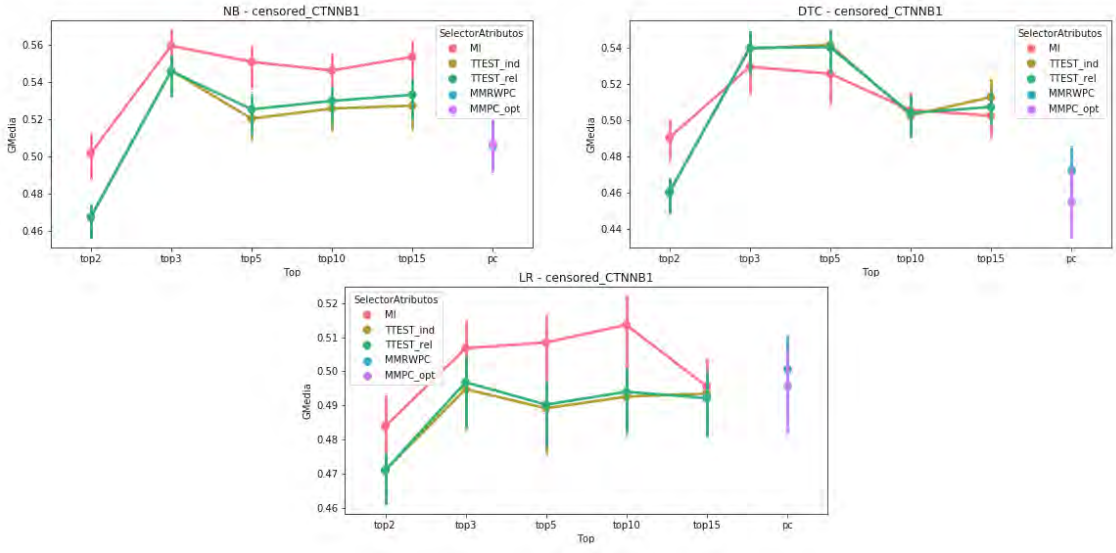

Figura 41. Resultados obtenidos censurando las instancias que usaron el inhibidor CTNNB1. Elaboración propia.

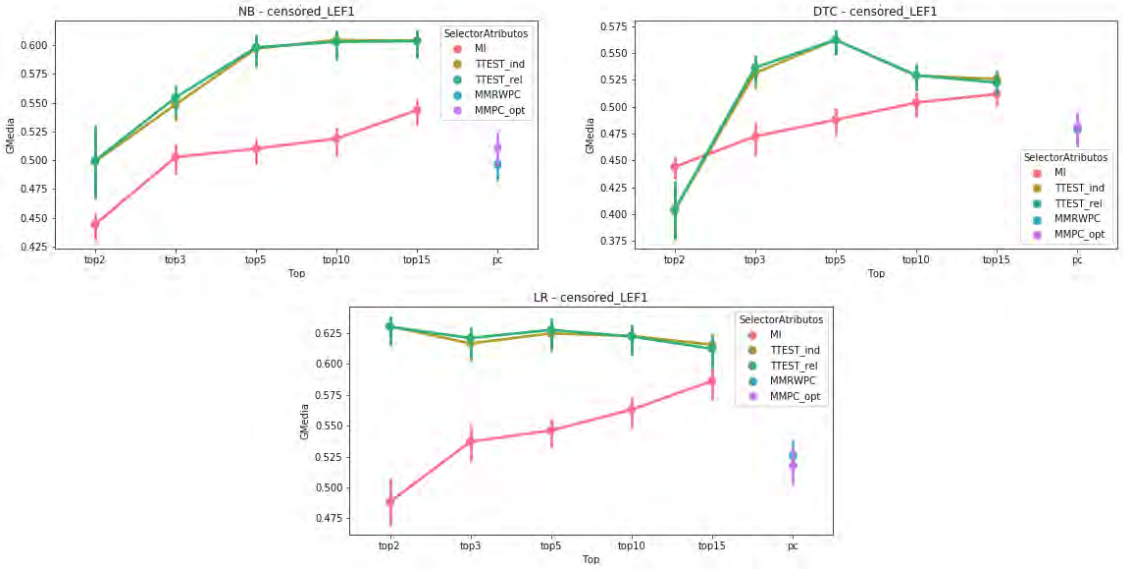

Figura 42. Resultados obtenidos censurando las instancias que usaron el inhibidor LEF1. Elaboración propia.

En la figura 42, se puede observar que el método TTEST empieza a establecer su rendimiento como el mejor, a través de los distintos clasificadores. En particular, se puede apreciar que el método TTEST_rel obtiene resultados óptimos respecto al resto de métodos considerados.

Realizando un análisis de las figuras 40 a 46, se puede observar dos claras tendencias: la variabilidad de los resultados del MMPC_opt es mayor a la de MMRWPC, y el método TTEST_rel sirve como calificador óptimo de atributos para los tipos de datos que se están procesando (interacciones génicas).

En comparativa, respecto a los dos primeros enfoques, observar resultados iguales o mejores en precisión, permite comprender que la clonación de instancias de clase positiva para lograr un conjunto de instancias balanceado no ha sido un acercamiento óptimo y eficiente, respecto al efectuado en esta sección.

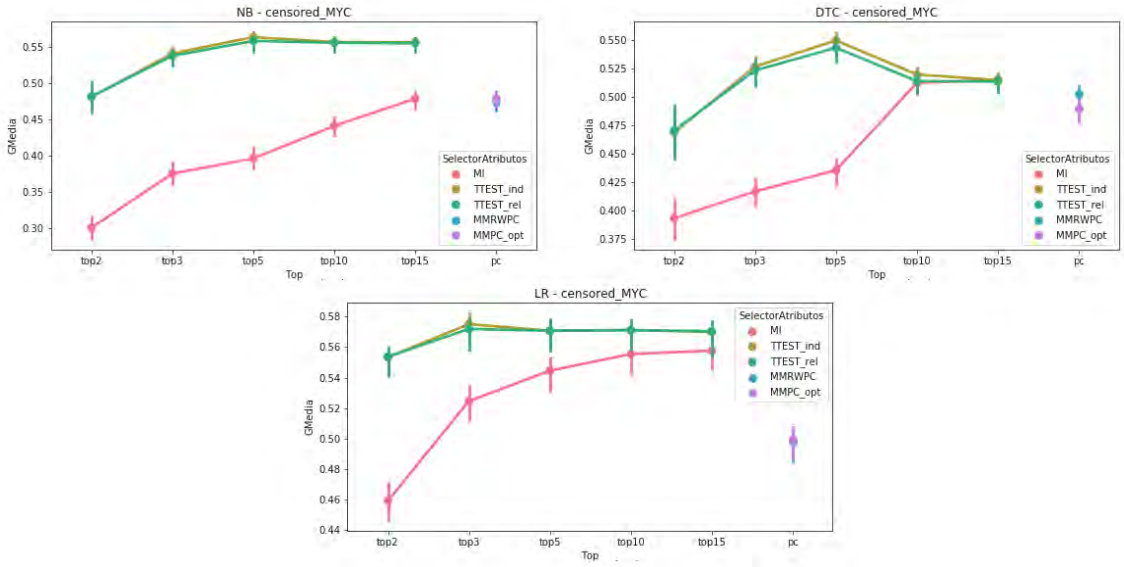

Figura 43. Resultados obtenidos censurando las instancias que usaron el inhibidor MYC. Elaboración propia.

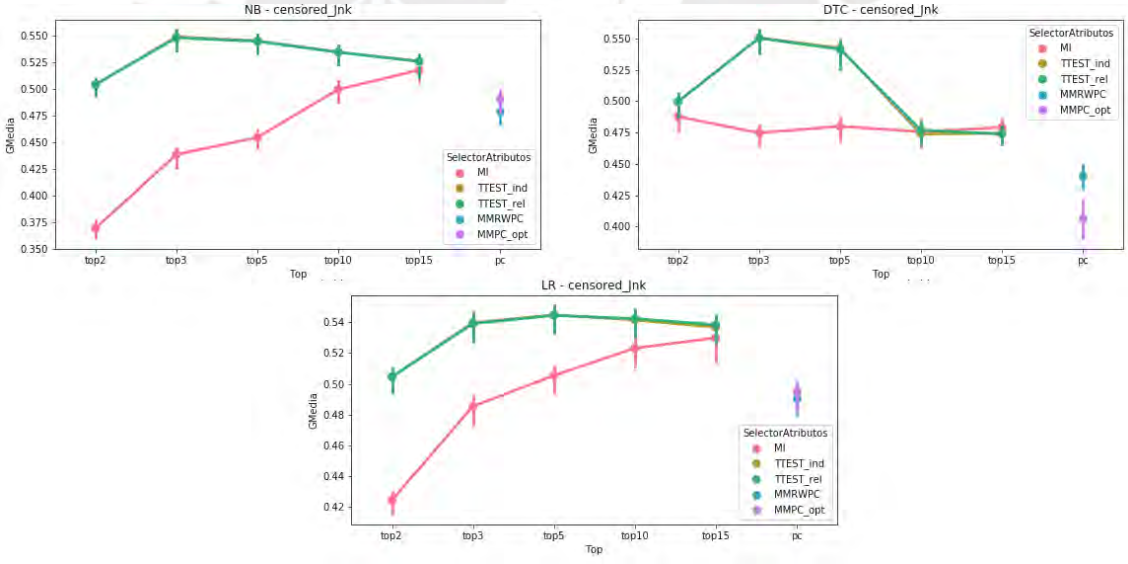

Figura 44. Resultados obtenidos censurando las instancias que usaron el inhibidor Jnk. Elaboración propia.

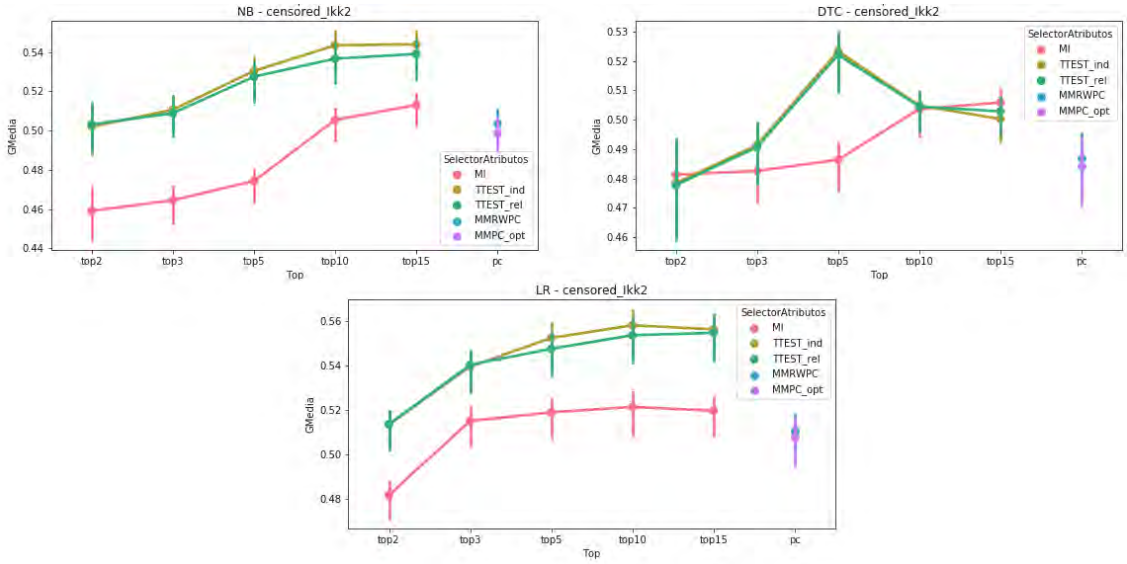

Figura 45. Resultados obtenidos censurando las instancias que usaron el inhibidor Ikk2. Elaboración propia.

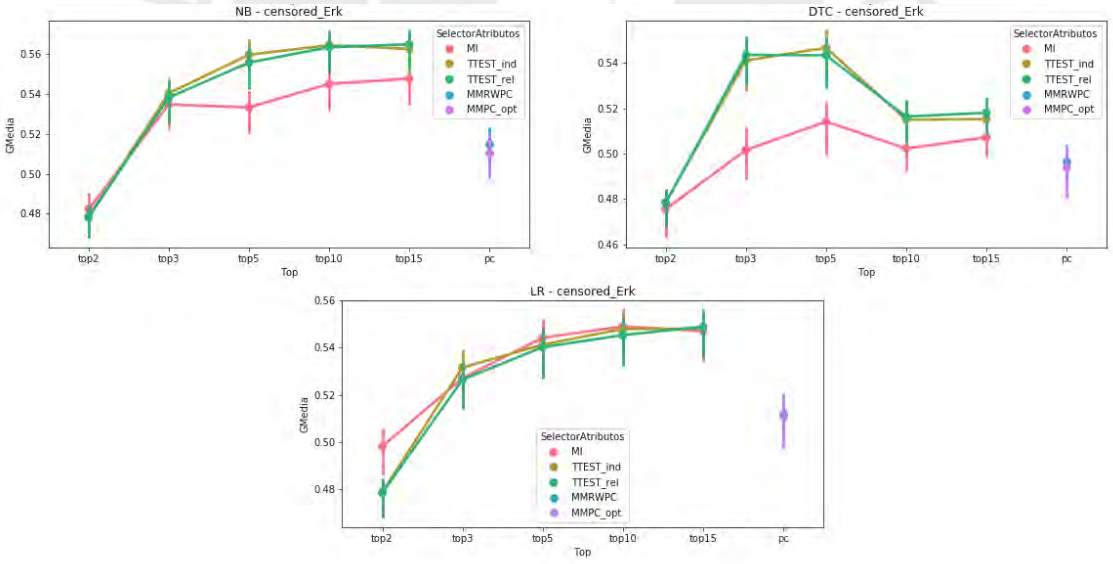

Figura 46. Resultados obtenidos censurando las instancias que usaron el inhibidor Erk. Elaboración propia.

8.3.4 Resultados obtenidos del cuarto enfoque, por la agregación de los modelos predictivos generados por cada muestra

Posterior al tercer enfoque, se agregaron todos los vectores de predicción generados por cada modelo por cada muestra, para explorar la posibilidad de armar un mejor predictor en base a los predictores anteriores. Se aplicó la agregación con pesos iguales, obteniendo los siguientes resultados:

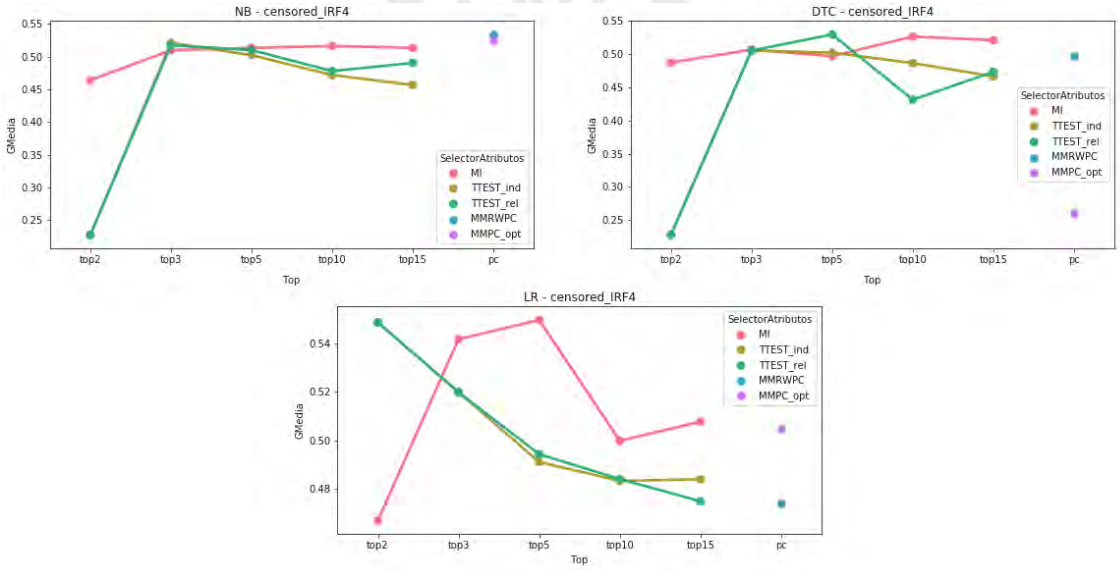

Figura 47. Figura de resultados de aplicación del cuarto enfoque al censurar censurado las instancias con inhibidor IRF4

En la figura 48 se logra apreciar la precisión más alta alcanzada en este proyecto de fin de tesis, mayor al 64%, usando el método TTEST_rel. A la par, se aprecia que, si bien el método MMRWPC logra resultados mejores al método clásico MMPC, no son resultados suficientemente buenos como para competir contra los calificadores de atributos clásicos incluídos en este proyecto de fin de carrera.

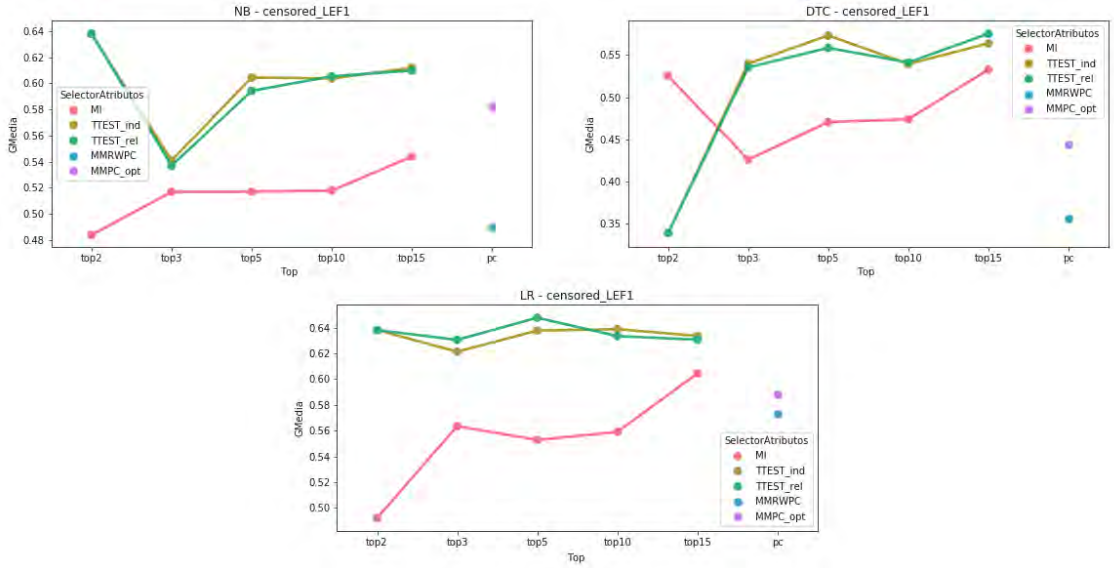

Figura 48. Figura de resultados de aplicación del cuarto enfoque al censurar las instancias con inhibidor LEF1. Elaboración propia.

En las figuras 49 a 51, se observa la primacía de TTEST_rel como mejor método, así como la continuidad en la tendencia, mencionada previamente, acerca de la calidad de los resultados del método MMRWPC respecto al método MMPC y a los demás calificadores de atributos clásicos.

Observando la totalidad de figuras en esta sección, se pudo concluir que la agregación de los predictores previamente generados es una opción óptima a ejecutar para el análisis de interacciones génicas, dado los atributos a disposición; respecto al resto de enfoques mostrados por este proyecto de fin de carrera.

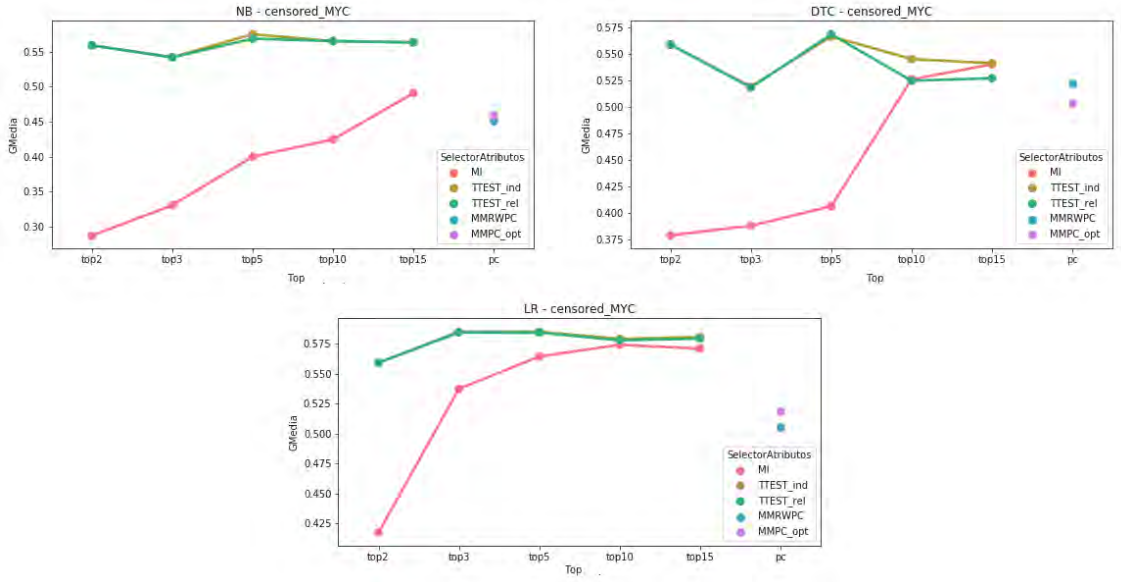

Figura 49. Figura de resultados de aplicación del cuarto enfoque al censurar las instancias con inhibidor MYC. Elaboración propia.

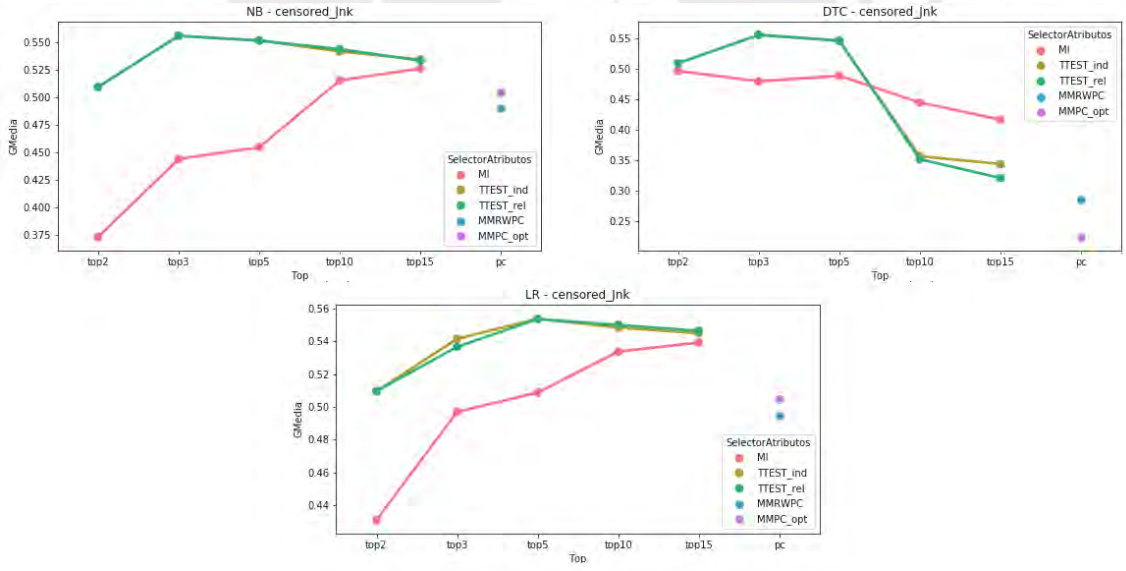

Figura 50. Figura de resultados de aplicación del cuarto enfoque al censurar las instancias con inhibidor Jnk. Elaboración propia.

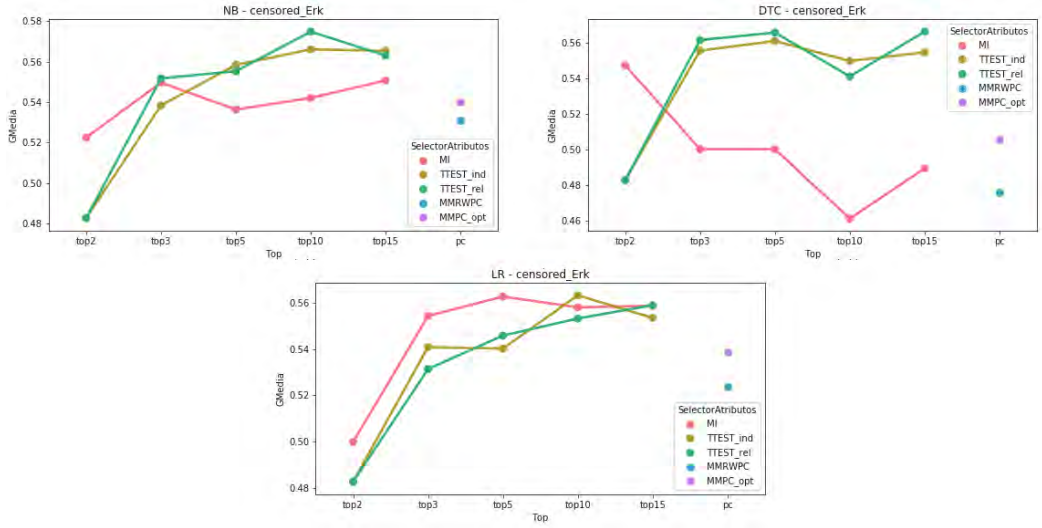

Figura 51. Figura de resultados de aplicación del cuarto enfoque al censurar las instancias con inhibidor Erk. Elaboración propia.

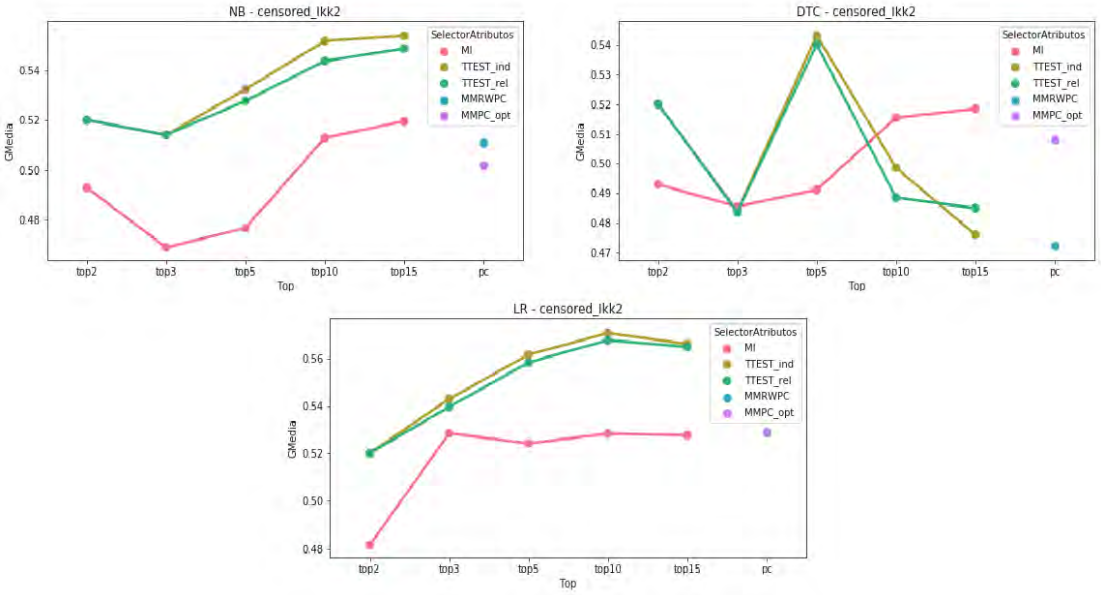

Figura 52. Figura de resultados de aplicación del cuarto enfoque al censurar las instancias con inhibidor Ikk2. Elaboración propia.

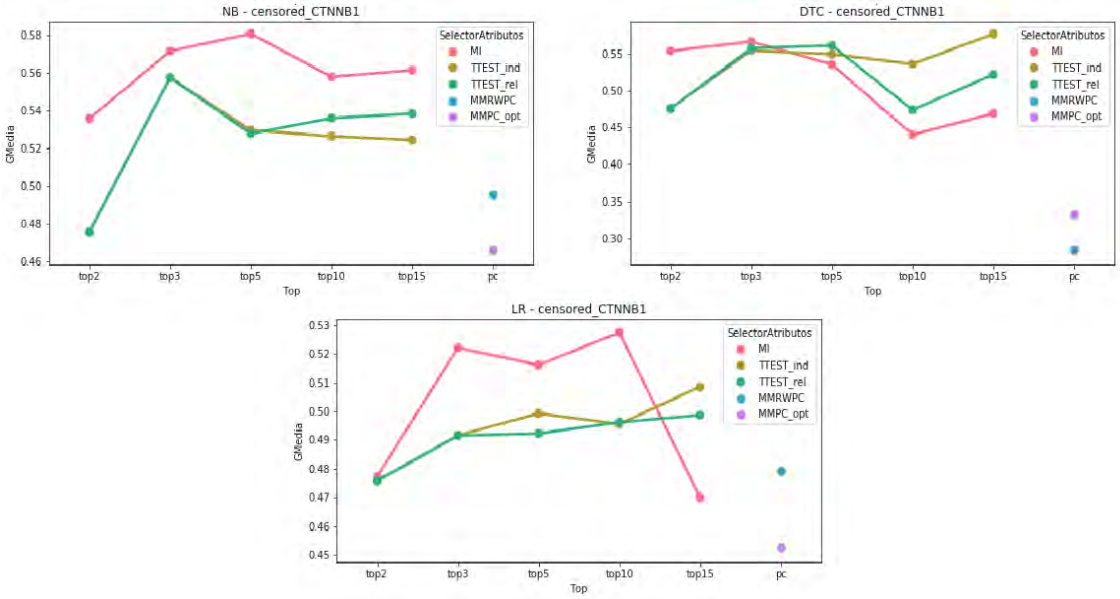

Figura 53. Figura de resultados de aplicación del cuarto enfoque al censurar las instancias con inhibidor CTNNB1. Elaboración propia.

Capítulo 9.Conclusiones y trabajos futuros

En este capítulo, se presentarán las conclusiones realizadas en base al desarrollo y resultado de este proyecto de fin de carrera, enfocado en la aplicación del concepto del Manto de Markov sobre conjuntos de interacciones génicas para su mejor predicción.

9.1 Conclusiones

Durante el desarrollo de este proyecto de fin de carrera, se pudo observar los siguientes resultados: para datos sintéticos, como lo fueron los sets de data utilizados para medir el rendimiento del algoritmo de estimación del Manto de Markov MMRWPC propuesto, se pudo hallar diferencias claras en la calidad de los atributos hallados para predecir un atributo objetivo cuando se compara con métodos de referencia de selección de atributos, siempre y cuando existiese un mínimo grado de información sobre todas las clases de ese atributo. En este sentido, los resultados mostraron que el grado de balance mínimo contenido en los datos biológicos a disposición son el principal obstáculo para la filtración adecuada de atributos discriminantes acerca de las interacciones génicas.

Asimismo, si bien se ha obtenido precisiones con un promedio superior al 50% (valor esperado de clasificación aleatoria), aún no se obtienen niveles suficientemente altos de precisión como para que la solución sea de adopción en la práctica laboratorial. Sin embargo, los resultados también apuntan a que es posible extraer cierta información causal de datos puramente observacionales de expresión génica, lo cual es de interés teórico y práctico en la comunidad científica, ya que la forma establecida de inferir causalidad es a través de experimentos intervencionales. Trabajos anteriores en el mismo problema de detección de causalidad con datos observacionales han mostrado lo desafiante que es esta tarea (Buhlmann et al., 2014; Colombo et al., 2012; Maathuis et al., 2009; Taruttis et al., 2015; Uhler et al., 2013), proponiendo complejos modelos estadísticos.

Los resultados encontrados en el presente trabajo muestran un camino alternativo basado en conceptos actuales de estimación de Manto de Markov y computación evolutiva. Definitivamente el uso de modelos de clasificación lineares y básicos ha afectado el potencial de precisión alcanzado y apuntan caminos para trabajos futuros.

9.2 Trabajos futuros

Se sugiere la realización de un mayor número de afinaciones a los modelos involucrados actualmente, así como adicionar modelos predictivos de mayor complejidad, en cuanto se cuente con la capacidad computacional para encontrar resultados en tiempos razonables.

Sería de interés aplicar técnicas de creación de datos sintéticos para lograr caracterizar toda la metodología de forma más controlada, ya que los resultados de este proyecto de fin de carrera evidencian claramente la influencia del grado de balance en la obtención de resultados óptimos.

Referencias

[1] J. J. Rossi and D. Castanotto, “The promises and pitfalls of RNA-interference-based therapeutics,” *Nature*, vol. 457, no. 7228, pp. 426–433, 2009.

[2] P. Zamore, “RNA interference: big applause for silencing,” *Stockholm*, no. 127, pp. 1083–1086, 2006.

[3] M. Zaratiegui, “Molecular biology: RNA interference hangs by a thread,” *Nature*, vol. 520, no. 7546, pp. 162–164, 2015.

[4] P. W. Holland, “Causal inference, path analysis and recursive structural equations modeling,” vol. 18, no. 1988, pp. 449–484, 1988.

[5] D. Lewis, “Causal Explanation,” *Philosophical Papers, Volume II*. pp. 214–240, 1986.

[6] M. H. Maathuis, M. Kalisch, and P. B??hlmann, “Estimating high-dimensional intervention effects from observational data,” *Ann. Stat.*, vol. 37, no. 6 A, pp. 3133–3164, 2009.

[7] J. Pearl, “Statistics and causal inference: A review,” *Test*, vol. 12, pp. 281–318, 2003.

[8] P. Buhlmann, M. Kalisch, and L. Meier, “High-Dimensional Statistics with a View Toward Applications in Biology,” *Annu. Rev. Stat. Its Appl.*, vol. 1, no. 1, pp. 255–278, 2014.

[9] D. Colombo, M. H. Maathuis, and M. Kalisch, “Learning high-dimensional directed acyclic graphs with latent and selection variables,” *Ann. Stat.*, vol. 40, no. 1, pp. 294–321.

[10] P. Spirtes, “Introduction to Causal Inference,” *J. Mach. Learn. Res.*, vol. 11, pp. 1643–1662, 2010.

[11] C. Uhler, P. Raskutti, and B. Yu, “Geometry of the Faithfulness Assumption in Causal Inference,” *Ann. Stat.*, vol. 41, no. 2, pp. 436–463, 2013.

[12] M. Kalisch and P. Buehlmann, “Causal Structure Learning and Inference: A Selective Review,” *Qual. Technol. Quant. Manag.*, vol. 11, no. 1, pp. 3–21, 2014.

[13] V. Bo, T. Curtis, A. Lysenko, M. Saqi, S. Swift, and A. Tucker, “Discovering Study-Specific Gene Regulatory Networks,” *PLoS One*, vol. 9, no. 9, 2014.

[14] Marcot, B. G. (2012). Metrics for evaluating performance and uncertainty of Bayesian network models. Ecological Modelling, 230, 50–62.

[15] Gao, T., & Ji, Q. (2017). Hybrid Markov Blanket discovery. In Proceedings - International Conference on Pattern Recognition (pp. 1653–1658). Institute of Electrical and Electronics Engineers Inc.

[16] Chai, L. E., Loh, S. K., Low, S. T., Mohamad, M. S., Deris, S., & Zakaria, Z. (2014, May 1). A review on the computational approaches for gene regulatory network construction. Computers in Biology and Medicine. Elsevier Ltd. https://doi.org/10.1016/j.compbiomed.2014.02.011

[18] M. A. González, “Estructura y dinámica de redes genéticas,” *Gac. Biomédicas. Órgano Inf. del Inst. Investig. Biomédicas la UNAM*, no. 2, pp. 1, 10–12, 2007.

[19] F. Taruttis, R. Spang, and J. C. Engelmann, “A statistical approach to virtual cellular experiments: Improved causal discovery using accumulation IDA (aIDA),” *Bioinformatics*, vol. 31, no. 23, pp. 3807–3814, 2015.

[20] Kitchenham, B. (2004). Procedures for performing systematic reviews. Keele, UK, Keele University, 33(TR/SE-0401), 28. https://doi.org/10.1.1.122.3308

[21] Gao, T., & Ji, Q. (2017). Efficient scored-based Markov Blanket discovery. International Journal of Approximate Reasoning, 80, 277-293.

[22] Aliferis, C. F., Tsamardinos, I., & Statnikov, a. (2003). HITON: a novel Markov Blanket algorithm for optimal variable selection. AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium, 21–25. https://doi.org/D030003616 [pii]

[23] Fu, S., & Desmarais, M. C. (2010). Markov Blanket Based Feature Selection: a Review of Past Decade. Proceedings of the World Congress on Engineering 2010, I, 321–328.

[24] Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2017). Deep Learning for Health Informatics. IEEE Journal of Biomedical and Health Informatics, 21(1), 4–21.

[25] Neapolitan, R. E. (2003). Learning Bayesian Networks. (S. Russel & P. Norvig, Eds.), Journal of Biomedical Informatics (Vol. 43, p. 674). Prentice Hall.

[26] Segaran, T. (2007). Programming Collective Intelligence: Building Smart Web 2.0 Applications. (M. Treseler O'brien, Ed.), Intelligence (p. 368). O'Reilly Media.

[27] U., A., S., I., C., Ö., A., A., & H.H., O. (2013). A dynamic Bayesian framework to learn temporal gene interactions using external knowledge. 2013 8th International Symposium on Health Informatics and Bioinformatics, HIBIT 2013

[28] Papadimitriou, C. (2003). Computational complexity. Addison-Wesley Publishing Company, Inc.

[29] Myte, R., Gylling, B., Häggström, J., Schneede, J., Magne Ueland, P., Hallmans, G., … Van Guelpen, B. (2017). Untangling the role of one-carbon metabolism in colorectal cancer risk: a comprehensive Bayesian network analysis. Scientific Reports, 7, 43434. https://doi.org/10.1038/srep43434

[30] Fortier, N., Sheppard, J., Strasser, S. (2015). Parameter Estimation in Bayesian Networks Using Overlapping Swarm Intelligence. Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (p. 9-16).

[31] Bishop, C. M. (1995). Neural networks for pattern recognition. *Journal of the American Statistical Association*, *92*, 482.

[32] Günther, F., Wawro, N., & Bammann, K. (2009). Neural networks for modeling gene-gene interactions in association studies. *BMC Genetics*, *10*(1), 87. https://doi.org/10.1186/1471-2156-10-87

[33] Jakkula, V. (2006). Tutorial on Support Vector Machine (SVM). *School of EECS, Washington State University*, 1–13.

[34] Amaratunga, D., Cabrera, J., & Lee, Y. S. (2008). Enriched random forests. *Bioinformatics*, *24*

[35] Li, Y., & Li, J. (2012). Disease gene identification by random walk on multigraphs merging heterogeneous genomic and phenotype data. *BMC Genomics*, *13*(Suppl 7), S27.

[36] Köhler S, Bauer S, Horn D, Robinson PN: Walking the Interactome for Prioritization of Candidate Disease Genes. Am J Hum Genet 2008, 82:949-958.

[37] Seiffert, C., Khoshgoftaar, T. M., & Van Hulse, J. (2009). Hybrid sampling for imbalanced data. In *Integrated Computer-Aided Engineering* (Vol. 16, pp. 193–210).

[38] Villanueva, E., & Maciel, C. D. (2014). Efficient methods for learning Bayesian network super-structures. *Neurocomputing*, *123*, 3–12. https://doi.org/10.1016/j.neucom.2012.10.035

[39] Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*(1), 31–78. https://doi.org/10.1007/s10994-006-6889-74

[40] Abraham K, Sameith, K, Falciani, F: Improving functional module detection. In: Ohio Collaborative Conference on Bioinformatics: 2009; Ohio; 2009: 110-115.

[41] Hainke, K., Szugat, S., Fried, R., & Rahnenführer, J. (2017). Variable selection for disease progression models: methods for oncogenetic trees and application to cancer and HIV. *BMC Bioinformatics*.

[42] Zhu, Zexuan et al. “Markov blanket-embedded genetic algorithm for gene selection.” *Pattern Recognition* 40 (2007): 32