Universidad de Buenos Aires

Facultad de Ciencias Exactas y Naturales

Departamento de Ciencias de la Computación

Herramienta de selección de variables para vincular regiones del cerebro y genes relacionados con patologías psiquiátricas

Tesis presentada para optar al título de Magíster en Explotación de

Datos y Descubrimiento del Conocimiento

Ing. Guillermo Facundo Poblete Pantoni

Director: Dr. Ramiro Salas

8 de marzo de 2021

Índice general

1. Introducción 5
   1.1. Objetivo 5
   1.2. Motivación 5
   1.3. Antecedentes 6
      1.3.1. El proyecto *ProcessGeneLists (PGL)* 6
      1.3.2. El proyecto *Allen Human Brain Atlas* 6
      1.3.3. *A genoma-wide association study of attempted suicide* 7
      1.3.4. Spatiotemporal transcriptome of the human brain 7
   1.4. Marco teórico 7
      1.4.1. Marco teórico para genética 7
      1.4.2. Marco teórico para neurociencia 11
      1.4.3. Marco teórico para minería de datos 12
2. Datos y Métodos 20
   2.1. Datos 20
      2.1.1. Versión pre procesada del *Allen Human Brain Atlas* 20
      2.1.2. *Allen Human Brain Atlas* Transpuesto 23
      2.1.3. Datos para validación y prueba del método *GetROIs* 25
      2.1.4. Datos para prueba del método *GetGenes* 26
   2.2. Métodos 27
3. Resultados 27
   3.1. El método *GetROIs* 27
      3.1.1. Definición formal 28
      3.1.2. La subfunción *WilcoxonFS* 32
      3.1.3. La subfunción *RandomForestFS* 33
      3.1.4. La subfunción *SVMFS* 34
      3.1.5. La subfunción *rKNNFS* 35
      3.1.6. Fases de validación y prueba del método *GetROIs* 35
      3.1.7. Análisis de los parámetros y métricas del método *GetROIs* 41
   3.2. El método *GetGenes* 50
      3.2.1. Definición formal 50
      3.2.2. Prueba del método *GetGenes* 53

3.2.3. Análisis de los parámetros y métricas del método *GetGenes* 54
4. Conclusiones y trabajos futuros 56
4.1. Conclusiones 56
4.2. Trabajos futuros 56
5. Anexo 57
5.1. Caso de uso de *GetROIs* y *GetGenes* 57
5.2. Tablas de datos 59
5.2.1. Datos y resultados de los modelos finales en *GetROIs* 59
5.2.2. Datos y resultados de la función *AUCRF* de la subfunción *GetGenes* 60
6. Bibliografía 61

Índice de figuras

Figura 1-1. Representación gráfica de una porción de ADN. 8
Figura 1-2. Esquema simplificado de la técnica de microarray 9
Figura 1-3. Diferencia en la frecuencia de las variaciones génicas entre casos y controles 10
Figura 1-4. Uso de microarrays en la detección de variantes génicas 10
Figura 1-5. Matriz de confusión 18
Figura 1-6. Clasificadores discretos en el espacio ROC 19
Figura 1-7. Ejemplo de una curva ROC 19
Figura 2-1: Estructura interna de la versión pre procesada del Allen Human Brain Atlas 20
Figura 2-2: Estructura relacional del Allen Human Brain Atlas 22
Figura 2-3. Frecuencia de regiones del cerebro entre todos los donantes del Allen Human Brain Atlas 23
Figura 2-4. Cantidad de probes por cada gen en el Allen Human Brain Atlas 24
Figura 3-1. Diagrama de flujo del método GetROIs 29
Figura 3-2. Creación de la variable target para las subfunciones WilcoxonFS, RandomForestFS, SVMFS y rKNNFS 30
Figura 3-3. Descripción gráfica del procedimiento de sub-muestreo aleatorio 31
Figura 3-4. Exactitud media de los modelos iterativos para Random Forest y rKNN en la Etapa 1 de selección de variables 42
Figura 3-5. Cantidad de columnas de los datos utilizados para crear los modelos finales (Etapa 2) 43
Figura 3-6. Exactitud de los modelos finales en relación con la cantidad de variables utilizadas en los mismos (Etapa 2) 44
Figura 3-7. Exactitud de los modelos finales de los métodos RF, KNN y SVM para cada donante, sobre cada lista (Etapa 2) 45
Figura 3-8. Valores del parámetro mtry de cada modelo Random Forest final según exactitud y cantidad de columnas, para cada lista (Etapa 2) 46

Figura 3-9. Valores del parámetro kmax de cada modelo KNN final según exactitud y cantidad de columnas, para cada lista (Etapa 2) 47
Figura 3-10. Valores del parámetro loss de cada modelo SVM final según exactitud y cantidad de columnas, para cada lista (Etapa 2) 48
Figura 3-11. Valores del parámetro cost de cada modelo SVM final según exactitud y cantidad de columnas, para cada lista (Etapa 2) 49
Figura 3-12. Valores del parámetro weight de cada modelo SVM final según exactitud y cantidad de columnas, para cada lista. 49
Figura 3-13. Diagrama de flujo de la función GetGenes 52
Figura 3-14. Cantidad de variables vs área bajo la curva ROC de los modelos generados por AUCRF 55

Índice de tablas

Tabla 2-1: Disposición de los datos en la tabla Raw 21
Tabla 2-2: Descripción de la tabla Raw/Standarized 21
Tabla 2-3: Disposición de los datos en la tabla Raw/Standarized transpuestas 24
Tabla 2-4: Descripción de la tabla Raw/Standarized transpuestas 24
Tabla 2-5. Descripción de las listas de genes para validación de la función GetROIs. ... 25
Tabla 2-6. Descripción de la lista de genes para prueba de la función GetROIs. 25
Tabla 3-1: Regiones del cerebro asociadas a la lista de genes provenientes del estudio Spatiotemporal transcriptome of the human brain 36
Tabla 3-2. Resultado del método GetROIs para la lista de genes CBC 37
Tabla 3-3. Resultado del método GetROIs para la lista de genes HIP 38
Tabla 3-4. Resultado del método GetROIs para la lista de genes MD 38
Tabla 3-5. Resultado del método GetROIs para la lista de genes NCX 39
Tabla 3-6. Resultado del método GetROIs para la lista de genes STR 39
Tabla 3-7. Resultado del método GetROIs para la lista de genes Willour 41
Tabla 3-8. Cantidad de filas de las tablas de microarray en la Etapa 1 42
Tabla 3-9. Nombres de regiones del cerebro correspondientes al subículo en el Allen Human Brain Atlas 53
Tabla 3-10. Resultado de la prueba del método GetGenes utilizando la lista de genes Willour sobre el subículo. 53
Tabla 3-11. Dimensiones de las tablas de microarray transpuestas sobre las que se realiza la selección de genes 54
Tabla 3-12. Cantidad de variables óptimas para cada donante, según la función AUCRF 54
Tabla 3-13. Valores de mtry y exactitud media de los modelos finales de AUCRF 55
Tabla 5-1. Descripción de datos y parámetros usados junto a la exactitud lograda en los modelos finales 59
Tabla 5-2. Descripción de los datos utilizados y los valores de AUC logrados por la función AUCRF 60

Resumen

En el presente trabajo combino tres métodos de selección de variables usando los algoritmos de clasificación *Random Forest*, *SVM* y *KNN* junto con la prueba estadística *Wilcoxon* de suma de rangos sobre el conjunto de datos *Allen Human Brain Atlas*, para encontrar regiones del cerebro posiblemente relacionadas a patologías psiquiátricas en base a un grupo de genes cuya vinculación a dicha patología sea ya conocida o sugerida. Estas regiones encontradas sirven para reducir la cantidad de comparaciones múltiples en las pruebas estadísticas al momento de analizar datos provenientes de estudios de resonancia magnética. A su vez, dicha herramienta permite la operación en dirección opuesta permitiendo encontrar aquellos genes posiblemente relacionados a una patología psiquiátrica, en un conjunto de regiones del cerebro con conocidas vinculaciones a dicha patología. Así, es posible reducir la cantidad de estudios genéticos a realizarse, priorizando aquellos genes resultantes de la utilización de la presente herramienta. Muestro la efectividad de los métodos desarrollados mediante pruebas llevadas a cabo sobre 5 listas de genes que, según el estudio *Spatiotemporal transcriptome of the human brain*, están asociadas con regiones del cerebro puntuales. Finalmente, muestro resultados útiles para una investigación en curso, encontrando una relación entre el subículo y el gen AKAP7 en pacientes psiquiátricos con propensión al suicidio.

1. Introducción

1.1. Objetivo

El objetivo del presente trabajo es desarrollar una herramienta de minería de datos destinada a encontrar regiones del cerebro relacionadas con patologías psiquiátricas, a partir de un conjunto de genes también vinculados a dicha patología. Esto permite reducir el espacio de búsqueda al momento de realizar múltiples pruebas estadísticas sobre datos provenientes de estudios por imágenes del cerebro, mitigando los problemas estadísticos relacionados con la realización de una gran cantidad de comparaciones múltiples. También debe permitir identificar los genes más importantes a estudiar en regiones específicas del cerebro para una patología psiquiátrica determinada. Esta herramienta tiene como finalidad ser utilizada en una investigación en curso donde se busca establecer relaciones entre la función del cerebro y su genética en pacientes con propensión a cometer actos suicidas (ver sección 5.1).

1.2. Motivación

La principal motivación para realizar este trabajo está dada por la relevancia de las investigaciones realizadas por el Dr. Salas, las cuales están enfocadas en el estudio de imágenes del cerebro, especialmente dirigida a la investigación en adicciones a las drogas, ansiedad, depresión y otros problemas relacionados con el cerebro. Este campo de investigación cumple un rol social vital ya que, según las estadísticas de 2018 recopiladas por el *National Institute of Mental Health (NIMH)*[1], el suicidio fue la décima causa de muerte en Estado Unidos cobrando las vidas de 48.000 personas ese año. Mas precisamente, fue la segunda causa de muerte en individuos entre 10 y 34 años y la cuarta entre las edades de 35 y 54 años, habiendo más de dos veces y media más suicidios (48344) que homicidios (18830) en ese país en dicho año. Estos números proveen un escenario cuantitativo que justifica la inversión en numerosos proyectos de investigación científica para tratar esta problemática y otras asociadas, entre los que se encuentra el proyecto *ProcessGeneLists* (ver sección 1.3.1) sobre el cual se basa el presente trabajo. Por otro lado, en neuropsiquiatría en general existe un serio problema en cuanto a la confluencia de datos genéticos y datos de función cerebral. Estos dos campos son comúnmente estudiados en forma aislada, aun cuando es evidente que la función cerebral (alterada en patologías neuropsiquiátricas) está íntimamente relacionada a variabilidad genética. El proyecto *ProcessGeneLists* es un acercamiento al estudio de ambas modalidades al mismo tiempo: Si la genética tradicionalmente estudia qué genes y variantes genéticas están asociadas a una patología (o fenotipo), y las imágenes del cerebro tradicionalmente estudian qué regiones del cerebro están asociadas a una patología (o fenotipo), *ProcessGeneLists* estudia qué genes, en virtud de su modulación en las distintas regiones del cerebro, están asociados a la misma. Por lo tanto, *ProcessGeneLists* y las herramientas desarrolladas en este trabajo puede ser aplicado no solo a suicidio, sino a cualquier otro problema neuropsiquiátrico.

1.3. Antecedentes

Este trabajo se apoya sobre tres investigaciones precedentes, tanto para su desarrollo como para su validación y prueba.

1.3.1. El proyecto *ProcessGeneLists (PGL)*

ProcessGeneLists es el proyecto fundacional en que se basa esta tesis. El mismo es una iniciativa innovadora para resolver problemas en genética, neuroimágenes e investigación clínica, alineada con las prioridades del *NIMH*. Encabezado por el Dr. Ramiro Salas, el proyecto se desarrolla en *Baylor College of Medicine*. Las tres principales innovaciones logradas por el mismo son:

- Proveer una eficaz y eficiente solución para derivar hipótesis relacionadas a la anatomía y función del cerebro, promediando niveles de expresiones de *mRNA* (ver sección 1.4.1), en diferentes regiones de este, para descubrir aquellas *ROIs (Regions of Interest, regiones de interés del cerebro)* asociadas con dichos promedios de expresión génica.
- Disminuir el efecto de comparaciones múltiples proveyendo hipótesis sobre regiones del cerebro basadas en información genética. En lugar de analizar el cerebro entero, *PGL* permite enfocarse en aquellas regiones que estén genéticamente conectadas al fenotipo bajo estudio.
- Este es el primer proyecto en investigar la conexión entre variantes genéticas en todo el genoma y sus niveles de expresión, imágenes del cerebro y datos clínicos (diagnósticos, síntomas, etc.) en pacientes con tendencias suicidas.

1.3.2. El proyecto *Allen Human Brain Atlas*

Para la creación de este atlas se utilizaron los cerebros de seis donantes, hombres y mujeres entre 18 y 68 años, sin historial conocido de condiciones neuropsiquiátricas o neurológicas. Las condiciones claves de exclusión fueron: lesiones o enfermedades cerebrales, epilepsia, adicciones al alcohol o a las drogas, más de una hora con ventilación asistida, positivo para enfermedades infecciosas, falla renal crónica, historial familiar de cáncer, cáncer en el cerebro o haber transcurrido un tiempo de la muerte mayor a 30 horas. De estos donantes se recolectaron tejidos del cerebro, fluidos y sangre. A los tejidos se les realizó un control de calidad de *ARN*, serología, pH, toxicología, neuropatología y micro neuropatología. Un comité de revisión aprobó la inclusión de todo el material. Los tejidos recolectados fueron sometidos a análisis de *microarray*, cuyo objetivo fue perfilar sistemáticamente la expresión de genes a través de las principales regiones del cerebro típico de un humano adulto (ambos hemisferios, el *cortex*, *subcortex*, *cerebellum* y *brainstem*). Se usaron los métodos de macro disección con escalpelo para el cortical y otras regiones relativamente grandes y uniformes, y micro disección laser para aquellas regiones más pequeñas y con forma más irregular. Estas muestras fueron tomadas a lo largo de tres años, y las técnicas de disección más modernas fueron incorporadas en el último tiempo, y por ese motivo los primeros cerebros incluyen información sobre áreas más grandes, y las más pequeñas se incorporaron en los últimos cerebros ingresados. Luego de obtener los resultados de *microarray* se procedió a normalizar los datos de cada cerebro por separado y entre

cerebros, a fin de evitar múltiples motivos de introducción de sesgo tales como las variaciones y la pérdida de calidad en el *ARN* con el paso del tiempo, envejecimiento de los chips de *arrays*, diferentes métodos de muestreo, distintos laboratorios intervinientes en los análisis, etc. Dos procedimientos de normalización fueron llevados a cabo: el primero empleado en los primeros 4 cerebros, y un método actualizado luego sobre los seis cerebros[2].

1.3.3. *A genoma-wide association study of attempted suicide*

La investigación titulada *A genoma-wide association study of attempted suicide*[3] se utiliza en el presente trabajo para poner a prueba las herramientas desarrolladas. El mismo propone que existe un componente genético heredable relacionado con intentos de suicidio o su final concreción, que está parcialmente relacionado a desordenes psiquiátricos. Dos regiones cromosómicas habían sido ya asociadas a intentos de suicidio, pero aparte de la probabilidad de que existan muchas otras, una aproximación de más alta resolución sería necesaria para encontrar los genes específicos. Así es que la Dra. *Willour* condujo un estudio utilizando la técnica *GWAS* (ver definición en la sección 1.3.3) para comparar los genotipos de 1201 individuos bipolares con historial de intentos de suicidio con los de 1497 sujetos bipolares que no poseen dicho historial. Ningún *SNP* (ver definición en sección 1.4.1) fue identificado como asociado a suicidio cuando las comparaciones múltiples fueron tenidas en cuenta en el trabajo original. Sin embargo, usando un criterio estadístico más relajado nosotros identificamos 2507 *SNPs* que poseen evidencia de posible asociación a dicha patología. Los mismos fueron considerados por el Dr. Salas y tras obtener el nombre de los genes a los que pertenecían, los utilicé en el presente trabajo. Doy más precisiones al respecto en la sección 5.1.

1.3.4. Spatiotemporal transcriptome of the human brain

Este estudio se usa para extraer listas de genes relacionadas con las regiones del cerebro que se estudian en el mismo, y así utilizarlas como pruebas de validación de la herramienta *GetROIs* desarrollada en este trabajo. Esta investigación se basa en la obtención de 1340 muestras de tejido de 57 cerebros humanos de personas fallecidas, hombres y mujeres desde adultos mayores hasta embriones de múltiples orígenes étnicos. Las regiones estudiadas comprenden la corteza cerebral, el núcleo medio dorsal del tálamo, el estriado, la amígdala, el hipocampo y 11 áreas del neocórtex. Se realizó el genotipificado de 2.5 millones de *SNPs*[4].

1.4. Marco teórico

Esta sección contiene información relevante para entender los conceptos técnicos básicos sobre genética, neurociencia y minería de datos involucrados en el planteo del objetivo y en el posterior desarrollo de las herramientas para cumplir con ellos.

1.4.1. Marco teórico para genética

El *ADN*, siglas de ácido desoxirribonucleico, es la molécula que contiene la información genética en los seres vivos. Contiene cuatro compuestos: adenina (A), citosina (C), guanina (G) y timina (T) dispuestos en forma de hélice de doble cadena. Los compuestos de ambas tiras se unen entre sí por puentes de hidrógeno siguiendo una secuencia

determinada, y el orden en el que lo hacen codifica la información genética (el genoma). La adenina siempre se une con la timina, y la guanina siempre con la citosina. Un gen está hecho de *ADN*, y contiene información necesaria para hacer una molécula, usualmente una proteína. En los humanos y otros organismos complejos los genes están divididos en zonas codificantes (exones) interrumpidas por zonas no codificantes (intrones). Un genoma es el conjunto completo de instrucciones genéticas de un determinado organismo. Cada gen contiene información necesaria para construirlo, permitir que crezca y se desarrolle. Un genoma contiene a los genes de un organismo[5].

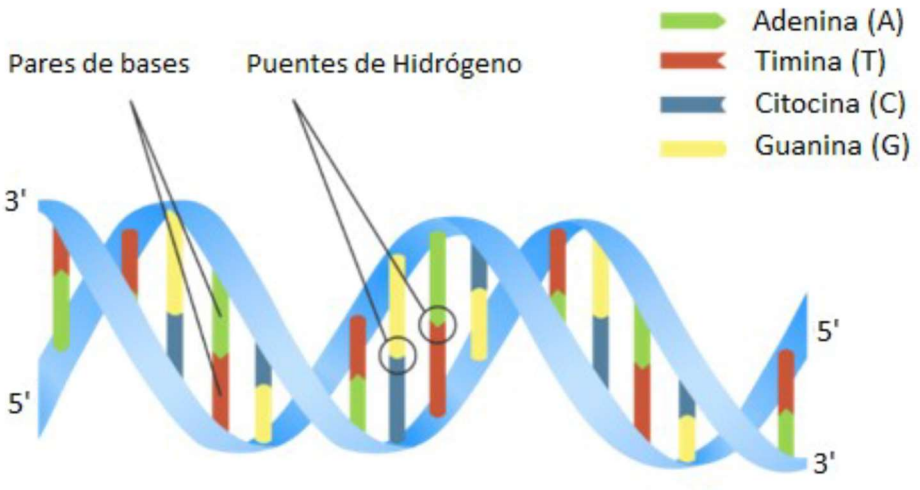

Figura 1-1. Representación gráfica de una porción de ADN.

La expresión génica es el proceso por el cual las instrucciones de un gen son convertidas en un producto funcional, como las proteínas u otras moléculas. Para esto, deben cumplirse dos pasos principales. El primero se llama transcripción, que se produce cuando el *ADN* de un gen es copiado para producir un tipo especial de *ARN* denominado *ARN* mensajero (*mARN*). El *mARN* es transportado desde el núcleo hasta los ribosomas (en el citoplasma), el lugar de la célula donde se producen las proteínas. El segundo paso comienza luego de que el *mARN* llega al ribosoma, donde es leído por una molécula denominada *tARN* de a tres letras por vez (un codón). Por ejemplo, los codones GGC y GGU codifican el mismo aminoácido, la glicina. Una serie de aminoácidos conectados es una proteína, que es una macromolécula funcional. La cantidad de producto funcional producido es llamado "nivel de expresión". El proyecto *Allen Human Brain Atlas* mide los niveles de *mARN* que luego forman las proteínas, utilizando una técnica denominada *microarray*. Esta es una técnica de laboratorio para detectar la expresión de miles de genes al mismo tiempo. Para ello se deben obtener pequeñas porciones de tejido de la parte del cuerpo de interés (en el caso de este trabajo, determinadas regiones del cerebro) que a su vez contienen *mARN* de los genes de los cuales queremos obtener los niveles de expresión. A cada una de estas piezas de tejido se las somete a una impresión de miles de pequeños puntos, los cuales contienen una secuencia específica que es parte de un gen sintetizado para el cual se conocen sus características. Este ácido nucleico sintetizado funciona como sonda (en inglés *probe,* término que utilizo a los largo de este trabajo para unificar la terminología con la original del atlas) para medir los niveles de

expresión génica de los genes de interés, uniéndose al *mARN* en un proceso conocido como hibridación. A estas piezas con impresiones se las conoce como chips de *microarray*. Cada punto impreso es un gen distinto, y la intensidad de color que toma dicho punto está directamente relacionada con el nivel de expresión génica del mismo. La Figura 1-2 muestra el proceso[6].

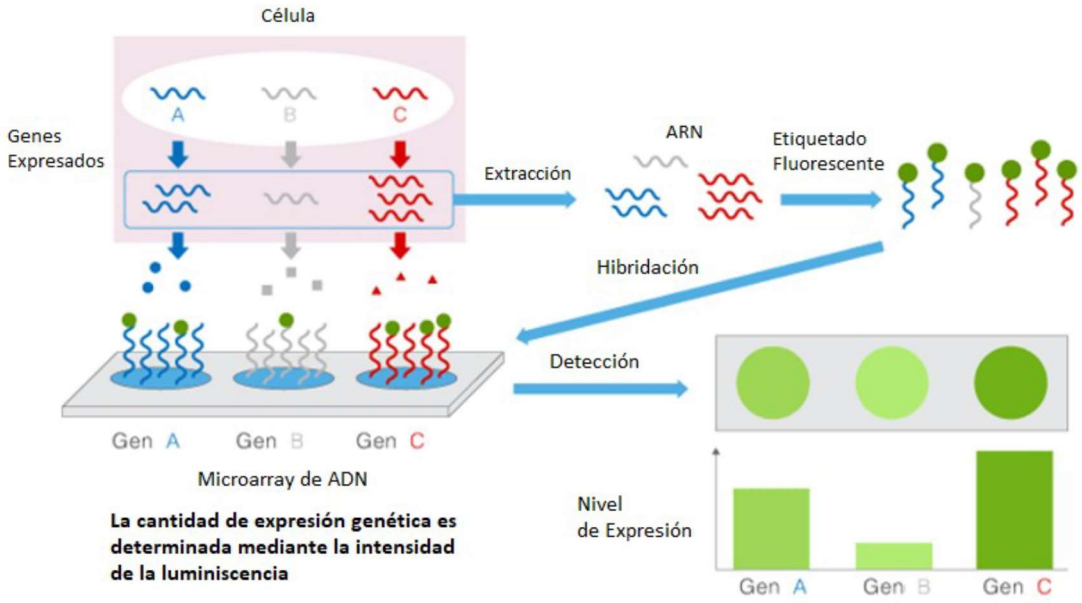

Figura 1-2. Esquema simplificado de la técnica de microarray

Los *SNPs*, siglas de *single nucleotide polymorphism,* son diferencias de un solo nucleótido en la secuencia de un gen, sobre el que se realizan estudios que permiten asociar enfermedades a dichas variaciones. *GWAS*, siglas de *genome-wide association studies*, es un tipo de estudio en el que se escanea el genoma de muchos individuos, con el fin de encontrar variaciones genéticas asociadas a una característica particular. Generalmente, se buscan cambios en una sola base en el *ADN* que componen los genes (un *SNP*). Se lo conoce como *genome-wide* porque se buscan *SNPs* a lo largo de todo el genoma. Por ejemplo, se pueden comparar *SNPs* en el genoma de personas con una enfermedad determinada contra los *SNPs* de personas que no la poseen, como ejemplifico en la Figura 1-3. *GWAS* es un método libre de hipótesis para identificar relaciones entre regiones genéticas y determinados rasgos y/o enfermedades. Las variaciones genéticas, que pueden causar diferencias en los fenotipos (rasgos visibles en el sujeto), se presentan con distinta frecuencia en los individuos que poseen estos rasgos (casos) que en aquellos que no los poseen (controles).

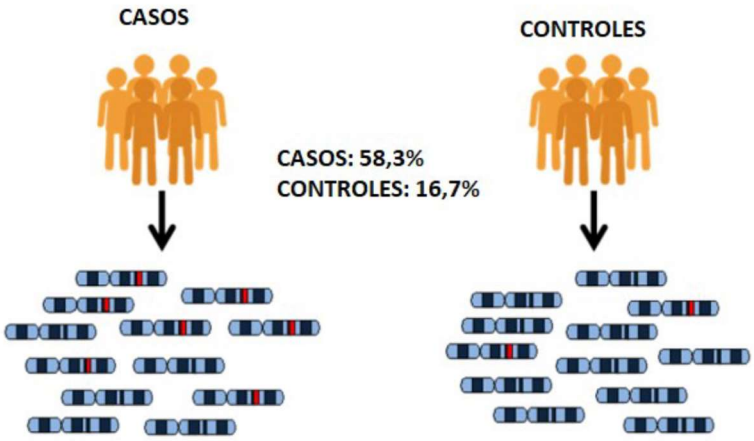

Figura 1-3. Diferencia en la frecuencia de las variaciones génicas entre casos y controles

Aquellas variantes asociadas con una enfermedad aparecerán con más frecuencia en los casos que en los controles. Esta búsqueda se realiza mediante *arrays* comerciales de *SNPs*, por lo que las variantes buscadas están bien definidas (no son casuales). La Figura 1-4 muestra un ejemplo de la detección de variantes mediante dichos *arrays*, y el diagrama *Manhattan* muestra los *p-values* que indican la significancia de la diferencia en frecuencia del alelo entre casos y controles con respecto a la posición de este en el cromosoma[7].

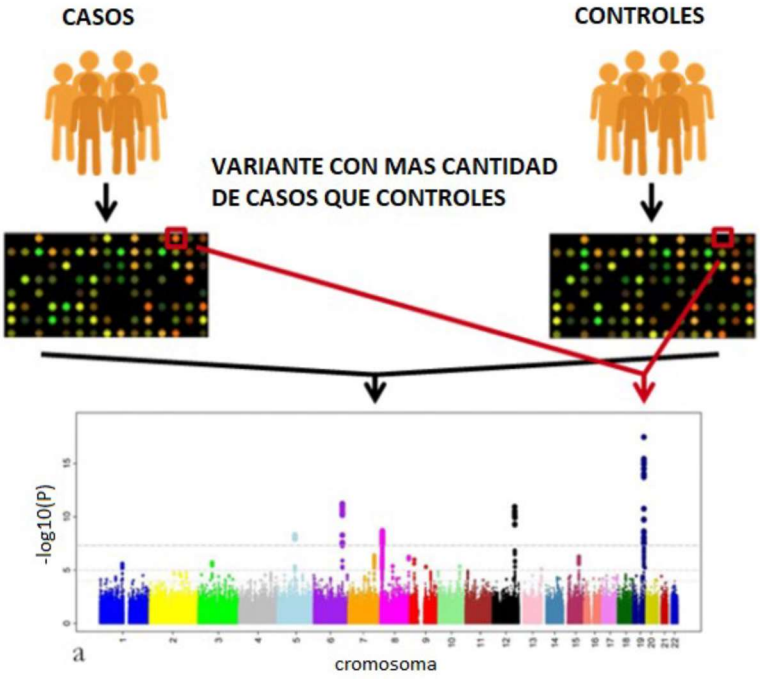

Figura 1-4. Uso de microarrays en la detección de variantes génicas

1.4.2. Marco teórico para neurociencia

Técnica de resonancia magnética

La técnica de resonancia magnética sirve para grabar información magnética proveniente del cerebro, realizando múltiples mediciones en un determinado período de tiempo (entre uno y dos segundos) y creando un modelo tridimensional. Fue creada por *Raymond Vahan Damadian*, y sus investigaciones al respecto comenzaron a principios de la década de 1970, cuando estudiaba la presencia de iones de potasio en las células y su tiempo de relajación (tiempo que transcurre hasta que una partícula estabiliza su campo magnético). Concluyó que el tiempo de relajación de dichos iones de potasio en las células era más corto que en solución acuosa. Más tarde, en 1971, publicó un estudio en el que demostraba que el tiempo de relajación en células tumorales era mayor que en células normales. En una imagen de resonancia magnética (*MRI*) se puede medir parámetros fisiológicos tales como densidad de la materia gris, densidad y localización de la materia blanca, tamaño, localización y curso de los vasos sanguíneos, medición del flujo sanguíneo y medidas relacionadas a la oxigenación sanguínea cerebral. La señal dependiente de la oxigenación sanguínea (*BOLD*, siglas de *Blood Oxygen Level-Dependent*) es el parámetro más utilizado en los estudios de resonancia magnética funcional (*fMRI*). Se estudian fluctuaciones de baja frecuencia, entre 0,01 y 0,08 Hz.[8]–[10]. La resonancia magnética funcional se utilizaba inicialmente para estudiar las respuestas a diversos estímulos, hasta que en 1995 *Bahrat Biswal* comenzó a realizar estudios funcionales sobre pacientes que no fueran sometidos a ningún estímulo. Aunque se esperaba encontrar con fluctuaciones aleatorias y desestructuradas, encontró correlaciones y estructuras activas con cierta organización en el sistema motor sensorial del cerebro. Ese fue el primer paso de lo que hoy se conoce como conectividad en estado de reposo (*resting state connectivity*). En 2001, *Marcus Raichle* utilizaba el mismo método cuando descubrió una red cerebral base, que se presentaba con mayor intensidad en estado de reposo. Y luego perdía fuerza cuando el paciente realizaba tareas específicas, al contrario de lo que sucedía con la red motor sensorial estudiada por *Biswal*. *Resting State* es actualmente utilizado para encontrar marcadores relacionados con el envejecimiento, psicopatología y síntomas clínicos. Normalmente, los participantes de este tipo de estudios deben permanecer quietos durante el experimento, y en algunos casos se solicita que cierren los ojos[11], [12].

Conectividad cerebral funcional

Nuestro cerebro es una compleja red estructural de regiones funcionalmente conectadas. Se presume que la comunicación funcional entre pares de regiones del cerebro juega un importante rol en los procesos cognitivos complejos. En el estudio del cerebro por imágenes, la conectividad funcional describe los patrones de activación neuronal entre regiones del cerebro anatómicamente separadas. La técnica de *fMRI* se conjuga con la de *resting state* para mapear los canales de comunicación funcional. Utilizando el *software Conn* se realizan las mediciones de conectividad funcional entre un grupo de *voxels* (mínima unidad cúbica del cerebro que puede analizarse) que toma el rol de semilla y otros grupos de *voxels* (análisis *seed-to-voxel*) o entre pares de regiones de interés (análisis *ROI-to-ROI*). Para ello se calcula la serie de tiempo *BOLD* promedio de todos los *voxels* dentro de una *ROI*, entre otros procedimientos. Esto permite el posterior cálculo de la conectividad *ROI-to-ROI* [13], [14].

1.4.3. Marco teórico para minería de datos

En esta sección repaso los conceptos sobre minería de datos utilizados en el presente trabajo. Comienzo abordando los tres métodos generales de selección de variable, y seguidamente explico breve y puntualmente los tres procedimientos específicos que usé en la herramienta desarrollada. Finalmente, concluyo describiendo las métricas utilizadas para evaluar los modelos generados.

Selección de variables: filtros, métodos empaquetados y embebidos.

La selección de dimensiones tiene como objetivos principales mejorar la capacidad predictiva, proveer modelos predictivos más rápidos y con mejor relación costo/beneficio, así como más sencillos de entender en su funcionamiento. En este trabajo la selección de variables es una cuestión central ya que normalmente, como sucede en la minería de datos aplicada a la salud, me encontré en situaciones donde debí utilizar conjuntos de datos con alta dimensionalidad con proporcionalmente muy pocos casos. Otro importante motivo para reducir la cantidad de variables es la capacidad de arribar a soluciones que permitan una explicación neurológica o genética, cuestión que se logra con un número acotado de características[15].

Existen tres métodos principales para realizar selección de variables: filtros, métodos empaquetados y métodos embebidos. Los filtros son métodos de preprocesamiento que se ejecutan previamente a la construcción de los modelos predictivos. Esto permite que la selección de variables sea independiente de la mecánica de estos últimos, aunque los modelos que luego son construidos con las variables seleccionadas de esta forma posean, generalmente, un poder predictivo menor comparado con los métodos empaquetados y embebidos. Los métodos basados en filtros usan una determinada medida para decidir la inclusión de las distintas variables, entre otras: distancia, similitud y medidas estadísticas. En este trabajo, la técnica utilizada como filtro es la prueba de *Wilcoxon* de suma de rangos, la cual usa una medida estadística para decidir la inclusión de cada una de las variables. Los métodos empaquetados evalúan los subconjuntos de variables de acuerdo con el desempeño que los mismos tienen en determinados

modelos predictivos. Estas evaluaciones se repiten para cada subconjunto de variables, las cuales se eligen mediante diferentes métodos de búsqueda. Los empaquetados son más lentos que los filtros debido a la demanda computacional en la ejecución de los modelos, y la selección final de variables está sesgada por el tipo de modelo utilizado. Aun así, generalmente se obtienen mejores resultados predictivos cuando la selección de variables se lleva a cabo con este método. Entre ellos se incluye *Random Forest*, utilizado en este trabajo. Los métodos embebidos son aquellos que realizan la selección de variables como parte de la construcción del modelo predictivo, o sea que son parte del algoritmo, como una extensión de su funcionalidad. Esta categoría abarca a aquellos algoritmos de aprendizaje automático tales como *SVM* penalizado, utilizado en este trabajo, el cual plantea un esquema de regularización que a fin de minimizar el error disminuye los coeficientes de las variables menos importantes[16]. En la sección 3.1 amplío la explicación sobre el funcionamiento de dichos métodos de regularización.

Selección de variables hacia atrás con Random Forest

La técnica de *bagging* tiene como objetivo reducir la varianza de un modelo promediando muchos de ellos con alta varianza y bajo sesgo, tal como sucede en aquellos basados en árboles. *Random Forest* mejora esta técnica construyendo una colección de muchos árboles no correlacionados, haciendo una selección de variables aleatoria para generar cada uno de ellos, para luego promediarlos como en la técnica de *bagging*. Muestro su funcionamiento en el siguiente pseudocódigo:

Sea B la cantidad de árboles a generar,

para $i \leftarrow 1$ **a** B **hacer**
  Tomar una muestra Boostrap Z* de tamaño N, del conjunto de entrenamiento;
  Generar un árbol Random Forest Tb en base a la muestra proveniente del Boostrap, repitiendo recursivamente los siguientes pasos por cada nodo terminal, hasta alcanzar el mínimo tamaño de nodo N.
  Seleccionar m cantidad de variables aleatoriamente de las p variables disponibles Elegir la mejor variable/punto de corte entre las m variables;
  Dividir el nodo en dos nodos hijos;
fin
Retornar el ensamblado de árboles T_{b1}^{B} ;

El método empaquetado de eliminación de variables hacia atrás usando *Random Forest* propone básicamente comenzar utilizando todas las variables para construir un modelo base, y luego ir quitando un porcentaje predefinido de las variables menos importantes según la métrica *Mean Decrease Gini*. Esta métrica se calcula de la siguiente forma: Cada vez que una variable es utilizada para dividir un nodo, se calcula el coeficiente Gini en cada uno de los nodos hijos (τ):

$$i(\tau) = 1 - p_1^2 - p_0^2$$

Siendo p_1 la probabilidad de que un elemento caído en dicho nodo hijo pertenezca a la clase 1 y p_0 la probabilidad de que pertenezca a la clase 0. Luego se calcula da diferencia

de este valor del índice con el de su nodo padre. Este cálculo se realiza para cada variable y por cada vez que la misma es utilizada en un nodo múltiples árboles generados. Las diferencias se promedian formando así el ranking de importancia que puede tener valores entre 0 y 1, siendo más importante la variable cuando mayor su valor de *Mean Decrease Gini*[17].

Se continúa quitando variables hasta que sólo quede un número mínimo de variables (también prefijado) y se genera un último modelo. Entre todos los modelos generados se conserva el de mayor desempeño con las variables que posee, siendo estas las finalmente consideradas[18], [19].

Uso este procedimiento en el método *GetROIs* (sección 3.1) y en el método *GetGenes* (sección 3.2).

La prueba de Wilcoxon de suma de rangos

Esta es una prueba estadística no paramétrica usada para determinar si dos muestras poseen distribuciones idénticas. La hipótesis nula es que ambas distribuciones son idénticas, o sea que los casos para ambas muestras fueron tomados al azar. La hipótesis alternativa es que las distribuciones no son idénticas o que una distribución se encuentra ubicada a un lado (derecha o izquierda) de la otra. Al ser una prueba no paramétrica, no requiere conocer a priori la distribución de los datos observados, pero requiere que ambas muestras sean independientes[20]

Dado que en este trabajo n_1 (genes relacionados a una patología psiquiátrica) y n_2 (el resto de los genes presentes en el *Allen Human Brain Atlas*) tendrán un tamaño mayor a 10, la distribución de T (la suma de los rangos de la muestra más pequeña) se aproxima a la distribución normal con media y desviación estándar:

$$\mu_T = \frac{n_1(n_1 + n_2 + 1)}{2}$$

$$\sigma_T = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$$

A continuación, detallo los pasos llevados a cabo para la realizar esta prueba:

1- Establecer la hipótesis nula y la alternativa. Para este trabajo en particular defino la hipótesis nula de la siguiente forma: La distribución de las expresiones génicas de un conjunto de genes de interés no difiere en su localización con respecto a la distribución de la expresión del resto de los genes disponibles en *Allen Human Brain Atlas* (sección 3.1.2).
2- Seleccionar la distribución a utilizar para calcular el estadístico Z. Para muestras mayores a 10 casos, es posible utilizar la distribución normal como aproximación al estadístico T.
3- Determinar la región de rechazo y de no rechazo de la prueba para el estadístico Z, según el α elegido.

4- Unificar las muestras en un solo conjunto de datos y calcular el rango de cada uno de los casos, para luego separar nuevamente las muestras y sumar los rangos para cada una de ellas. Calcular el valor del estadístico Z.

$$z = \frac{T - \mu_T}{\sigma_T}$$

5- Decidir si rechazar o no la hipótesis nula, comparando el valor del estadístico Z observado con el valor crítico del mismo determinado en el punto 3, para el α seleccionado.

Corrección por pruebas estadísticas múltiples

Al aplicar pruebas estadísticas sobre conjuntos de datos provenientes de estudios neurológicos o genéticos, como sucede en otras ramas de la ciencia, es común realizar múltiples comparaciones. Esto provoca un aumento en la probabilidad de rechazar incorrectamente la hipótesis nula. Para compensar este efecto aplicamos la corrección Bonferroni, que consta de dividir el *p-value* obtenido en cada prueba por la cantidad total de comparaciones múltiples realizadas[21].

Máquinas de vectores de soporte

Conceptualmente, la idea detrás de las máquinas de vector de soporte es la siguiente: vectores de entrada previamente etiquetados con una determinada clase son mapeados a un espacio de mayor dimensionalidad, utilizando funciones denominadas *kernels*. Un *kernel* es un producto interno entre dos vectores, que devuelve un valor real. Así es que, en lugar de representar los vectores utilizando todas las coordenadas (variables predictoras en nuestro caso), usamos un *kernel* para hacer el producto interno entre cada par de vectores de dicho espacio, y reformulamos la función de aprendizaje para que trabaje con estos nuevos valores reales. En este nuevo espacio se construye una superficie que separe a los vectores de distinta clase con un margen de separación óptimo, o sea maximizando la distancia perpendicular entre el punto más cercano de cada clase y el plano. Para construir esta superficie es necesario tener en cuenta solo un subconjunto de todos los vectores de entrenamiento, conocidos como vectores de soporte, que son aquellos que están ubicados más cercanamente a la mencionada superficie de separación. El objetivo es que los vectores estén lo más lejos posible de la superficie divisoria, y para lograrlo se minimiza la siguiente función de costo:

$$C\left[\sum_{i=1}^{m} y_i * \max(0,1 - \widehat{y_i}) + (1 - y_i) * \max(1 + \widehat{y_i})\right] + \ pen_\lambda(w)$$

La penalización puede adquirir dos formas: L_1 o L_2

La penalización L_2, la cual es utilizada en la subfunción *SVMFS* (0), adquiere la forma

$$pen_\lambda(w) = \lambda \sum_{1}^{d} w_i^2$$

que disminuye el valor de los coeficientes, pero no los lleva a cero.

En la misma sección, puede observar que esta forma de penalización puede ser acompañada de dos formas distintas de optimización a ser aplicada a la función de costo, primal o dual. La forma Primal de optimización es la mostrada arriba, y es utilizada cuando los casos son linealmente separables. Cuando los casos no son linealmente separables se utiliza la forma dual, donde básicamente se separa cada caso proyectándolo en una dimensión mayor agregándole variables relevantes al mismo, utilizando el multiplicador de Lagrange para resolver las restricciones del problema de optimización.

$$maximizar_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (X_i^T . X_j)$$

$$siendo\ \alpha_i \geq 0\ para\ todo\ i = 1, \dots, n\ y \sum_{i=1}^{n} \alpha_i y_i = 0$$

Si $\alpha_i > 0$ entonces X_i es un vector de soporte, y cuando $\alpha_i = 0$ entonces X_i no es un vector de soporte. La mayor ventaja de la forma dual es que solo depende de α[22].

Sea A el primer término de la suma en la ecuación de costo, y sea B el segundo término (el de regularización), entonces podemos reescribir la ecuación como

$$A + \lambda B$$

Siendo λ el término que determina el grado de regularización aplicado, por convención se escribe la suma anterior como

$$CA + B\ o\ sea\ C = \frac{1}{\lambda}$$

Este parámetro C se denomina costo y es uno de los que ajusto en la subfunción *SVMFS* del método *GetROIs* (ver sección 0).

En los casos en que haya un desbalance en la cantidad de casos de una clase a clasificar por sobre la otra, es útil asignar un peso al costo según la clase.

$$C_i = weight_i * C$$

Así se puede asignar una penalidad menor si se trata de un error de clasificación en un caso de la clase minoritaria, y un costo mayor si se trata de un caso de la clase mayoritaria. De esta forma se evita que el algoritmo ajuste demasiado la superficie divisoria por tratarse de un error en la clase minoritaria, siendo más flexible con dicha circunstancia y más rígido con la clasificación de la clase mayoritaria. Aunque el escenario abordado por la función GetROIs no representa un desbalance de clases, este parámetro es otro de los ajustados por la función *Caret* [23]en la subfunción *SVMFS* (ver sección 0)[24], [25]

Búsqueda aleatoria basada en K vecinos más cercanos

Básicamente, el algoritmo clasificatorio *KNN* (*K-nearest neighbors*, K-vecinos más cercanos) funciona de la siguiente forma: Dado un nuevo caso x_0 que se quiere clasificar se buscan los k casos más cercanos en distancia a este cuya clase ya es conocida, y luego se lo clasifica usando votación mayoritaria entre esos k casos vecinos. Si k=1 se asigna al nuevo caso la misma clase que el caso vecino más cercano[26]. En este trabajo, ajusto la máxima cantidad de vecinos utilizados (kmax) en la subfunción *rKNNFS* del método *GetROIs* (ver sección 3.1.5) Allí también uso la distancia Minkowski, definida como:

$$d^p = \left[(x_i - x_j)^p + (y_i - y_j)^p\right]^{\frac{1}{p}}$$

para 1<p<2, el cual se ajusta de acuerdo con el tipo de datos y la clase de agrupamiento conveniente para aumentar la exactitud de la clasificación[27].

Este tipo de modelos no requiere de una etapa de entrenamiento, dado que no se crea un modelo. Solo se tienen casos de los cuales ya se conoce su clase, y la evaluación de los nuevos casos se hace directamente sobre los conocidos. En este trabajo uso dos tipos de algoritmos distintos basados en este mismo concepto: *rKNN* para la selección de variables y *KKNN* para la creación del modelo final (sección 3.1.5). El algoritmo *rKNN* (*Random Forest KNN*) es propuesto como una alternativa más rápida, estable y fácil de implementar que *Random Forest*. El mismo crea múltiples instancias de modelos *KNN* usando en cada uno de ellos una determinada cantidad de variables elegidas al azar. Cada uno de los modelos emite la clasificación de cada una de las instancias y la clasificación final de estas se decide por voto mayoritario. Dado que *rKNN* puede llegar a utilizar gran parte de todas las variables disponibles, es necesario poseer un método para medir la importancia de cada variable. En este trabajo uso una medida denominada "soporte", definida como:

$$soporte(f) = \frac{1}{|C(f)|}\sum_{knn \epsilon C(f)} acc(knn)$$

Siendo f la variable a evaluar y $C(f)$ la cantidad de veces en la que dicha variable aparece en un clasificador, el soporte es un promedio de la exactitud de los modelos en la que determinada variable aparece. Usualmente, primero se estandariza cada una de las variables predictoras de todos los casos de entrenamiento para que posean media cero y varianza igual a uno, evitando así que diferencias en las escalas de medición influyan en el cálculo de la distancia. Para cada nuevo caso cuya clase es desconocida, el algoritmo *KKNN* realiza una clasificación basada en los k-vecinos más cercanos, encontrados mediante la distancia *Minkowski* [28].

Exactitud y área bajo la curva ROC

Los modelos clasificatorios binarios tales como *Naive Bayes* o los árboles de decisión clasifican las instancias como 1/0 o Si/No. Cuando estos modelos son aplicados a nuevos datos, para cada instancia se calcula la probabilidad de pertenencia a dicha clase. De acuerdo con un umbral de probabilidad predeterminado, se asigna a cada instancia la

clase correspondiente. Por ejemplo, si la probabilidad de pertenencia a la clase “1” es mayor a 0,5 entonces se le asigna dicha clase. Si es menor se le asigna la clase “0”. Luego se genera la matriz de confusión, una tabla de dimensión n x n, siendo n la cantidad de clases predichas. Dicha matriz se completa de acuerdo con la cantidad de verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos, como muestro en la Figura 1-5.

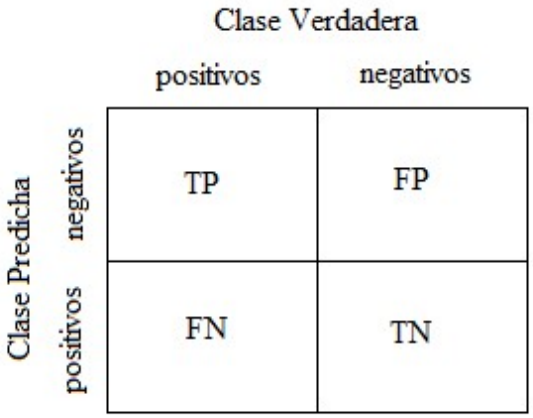

Figura 1-5. Matriz de confusión

TP = Verdadero Positivo (*True Positive*). Número de predicciones positivas correctas

FP = Falso Positivo (*False Positive*). Número de predicciones positivas incorrectas

FN = Falso Negativo (*False Negative*). Número de predicciones negativas incorrectas

TN = Verdadero Negativo (*True Negative*). Número de predicciones negativas correctas

A partir de los datos presentes en esta tabla de confusión se pueden calcular las siguientes métricas:

$$Accuracy = \frac{TP + TN}{P + N} \quad Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{P}$$

Siendo P la cantidad de predicciones positivas y N la cantidad de predicciones negativas. Uso la métrica de exactitud (*accuracy*) en el método *GetROIs* (sección 3.1).

La curva *ROC* es una representación gráfica bidimensional, en la que la tasa de verdaderos positivos ocupa el eje “Y” y la tasa de falsos positivos lo hace en el eje “X”. El punto (0,0) indica que nunca se ha realizado una clasificación positiva (ni verdadero positiva ni falso positivo). El punto (1,1), por el contrario, significa que todas las clasificaciones han sido positivas. El punto (0,1) representa una clasificación perfecta. En la Figura 1-6 muestro cinco clasificadores discretos representados en el espacio *ROC*.

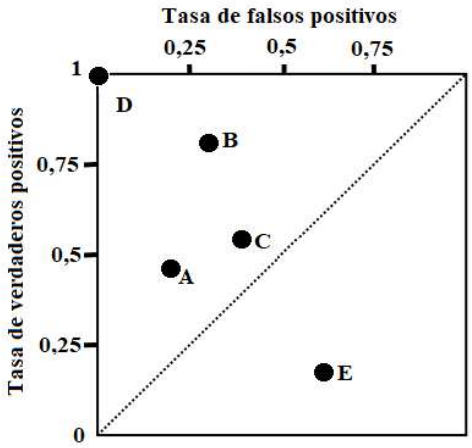

Figura 1-6. Clasificadores discretos en el espacio ROC

La línea punteada representa la región en la que se ubican los clasificadores aleatorios. Cada punto indica el desempeño de cada clasificador con respecto a la cantidad de los falsos y verdaderos positivos, según un umbral de corte para las probabilidades previamente determinado. Si, por ejemplo, para el clasificador "A" modificáramos dicho umbral (digamos de 0,5 a 0,6) obtendríamos un nuevo punto. Si repitiéramos dicho procedimiento N veces obtendríamos una curva que representaría el desempeño del modelo A para cada umbral de probabilidades.

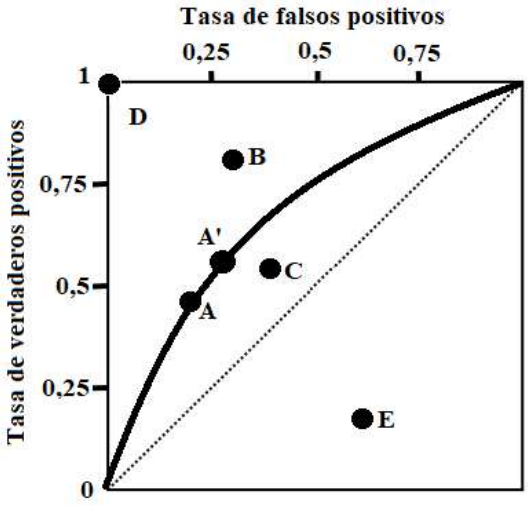

Figura 1-7. Ejemplo de una curva ROC

El área bajo la curva *ROC* (*AUC*, por *area under the curve*) es un valor entre 0 y 1, aunque para que un clasificador realice una clasificación mejor que al hacerla por azar su AUC debe ser siempre mayor que 0,5. Uso esta métrica para evaluar los modelos generados en el método *GetGenes* (sección 3.2)[29]

2. Datos y Métodos

2.1. Datos

En este trabajo utilizo 4 tipos de datos:

- Una versión preprocesada del *Allen Human Brain Atlas*, que utiliza el método *GetROIs* (sección 3.1).
- Una versión del atlas similar a la descripta en el punto anterior, pero con las tablas de *microarrays* transpuestas, como describo en la sección 3.2.
- Cinco listas de genes para validación y una para prueba del método *GetROIs* (sección 2.1.3).
- Una lista de regiones del cerebro junto a una lista de genes que utilizo para probar el método *GetGenes* (sección 3.2).

2.1.1. Versión pre procesada del *Allen Human Brain Atlas*

Se trata de una versión del *Allen Human Brain Atlas* modificada de tal forma que pudiera ser utilizada en lenguaje R por la herramienta que desarrollé para este trabajo. Uso esta versión del atlas en el método *GetROIs* para realizar la selección de variables sobre las regiones del cerebro.

Esta subsección está dividida en dos partes. En la primera, describo la estructura del atlas. En la segunda, analizo cuestiones relacionadas con la cantidad de información disponible para las distintas regiones del cerebro y genes presentes.

Estructura de la versión pre procesada del Allen Human Brain Atlas

En la Figura 2-1 muestro un diagrama descriptivo con la estructura de datos en cuestión.

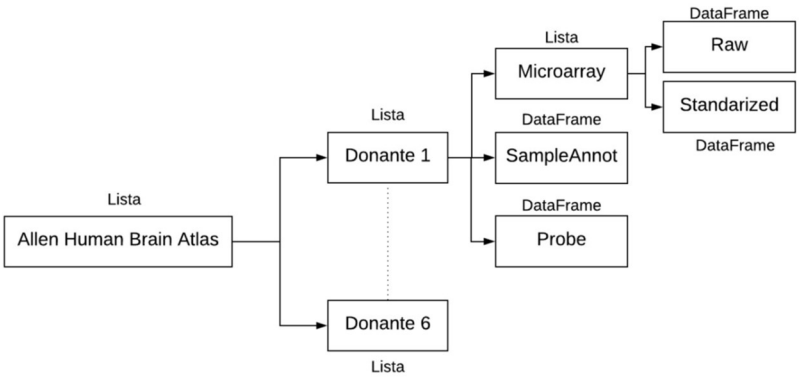

Figura 2-1: Estructura interna de la versión pre procesada del Allen Human Brain Atlas

A continuación, describo cada uno de los elementos mostrados:

- *Allen Human Brain Atlas*: Este objeto es una lista que contiene 6 objetos, uno por cada donante presente en el atlas.
- Donante 1 a Donante 6: Estos son 6 objetos, uno por cada donante, que contienen internamente 3 objetos cada uno (*Microarray*, *SampleAnnot* y *Probe*).

- *Microarray*: Es un objeto presente dentro de cada objeto Donante. El mismo contiene en su interior 2 tablas: *Raw* y *Standarized*.
- *Raw*: Es una tabla que contiene la información de expresión génica de cada gen en cada región del cerebro estudiada en el atlas, para cada una de las sondas utilizadas para estudiarlos. La tabla posee la siguiente estructura:

	Región 1	Región 2	…	Región M
Probe 1				
Probe 2				
…				
Probe N				

Tabla 2-1: Disposición de los datos en la tabla Raw

La relación de cardinalidad entre *probes* y genes es $n \rightarrow 1$. Esto significa que uno o más *probes* pueden ser utilizadas para estudiar un solo gen. La misma cardinalidad se presenta en las columnas, donde una o más columnas pertenecen a una misma región. La Tabla 2-2 muestra las dimensiones de las tablas de *microarrays* tanto *Raw* como *Standarized*. El atlas posee, tomando en cuenta a los 6 donantes juntos, un total de 414 regiones del cerebro únicas.

Donante	Cantidad de *Probes* (Filas)	Cantidad de regiones (Columnas)
Donante 1	58692	942
Donante 2	58692	893
Donante 3	58692	363
Donante 4	58692	529
Donante 5	58692	470
Donante 6	58692	501

Tabla 2-2: Descripción de la tabla Raw/Standarized

- *Standarized*: Son las que uso en este trabajo con el método *GetROIs*. Poseen las mismas características que las tablas *Raw*, pero sus filas están estandarizadas de la siguiente forma:

  Sea X el vector que contiene a todos los valores de expresión génica a través de todas las regiones del cerebro disponible en un donante y x_i un elemento de dicho vector:

$$x_i = \frac{x_i - mean(X)}{sd(X)}$$

  Realizo este procedimiento para todos los elementos de la fila.

- *SampleAnnot*: Es una tabla presente en cada objeto "Donante". La misma contiene información sobre las regiones del cerebro que fueron estudiadas en ese donante. En este trabajo utilizo solamente la información contenida en la columna *structure_name*. Cada una de las tablas *SampleAnnot* posee la misma cantidad de filas que cantidad de regiones (columnas) tiene su correspondiente tabla de *microarrays* (ver Tabla 2-2).

- *Probe*: Es una tabla que posee información sobre los *probes* utilizados para estudiar la expresión génica de los genes presentes en el atlas. En este trabajo utilizo la información de la columna *genes_symbol* para establecer cuáles de los genes que el usuario desea estudiar se encuentran efectivamente en el atlas. La tabla posee 58692 filas, la misma cantidad de filas que cada uno de los *microarrays* (ver Tabla 2-2).

La figura 2.2 muestra la estructura relacional que existe entre las tablas *Sample_Annot*, *Probe* y las de *microarrays*. Por cuestiones de simplicidad solo muestro la tabla de *microarrays* del Donante 1 pero existen 6 tablas de ese tipo.

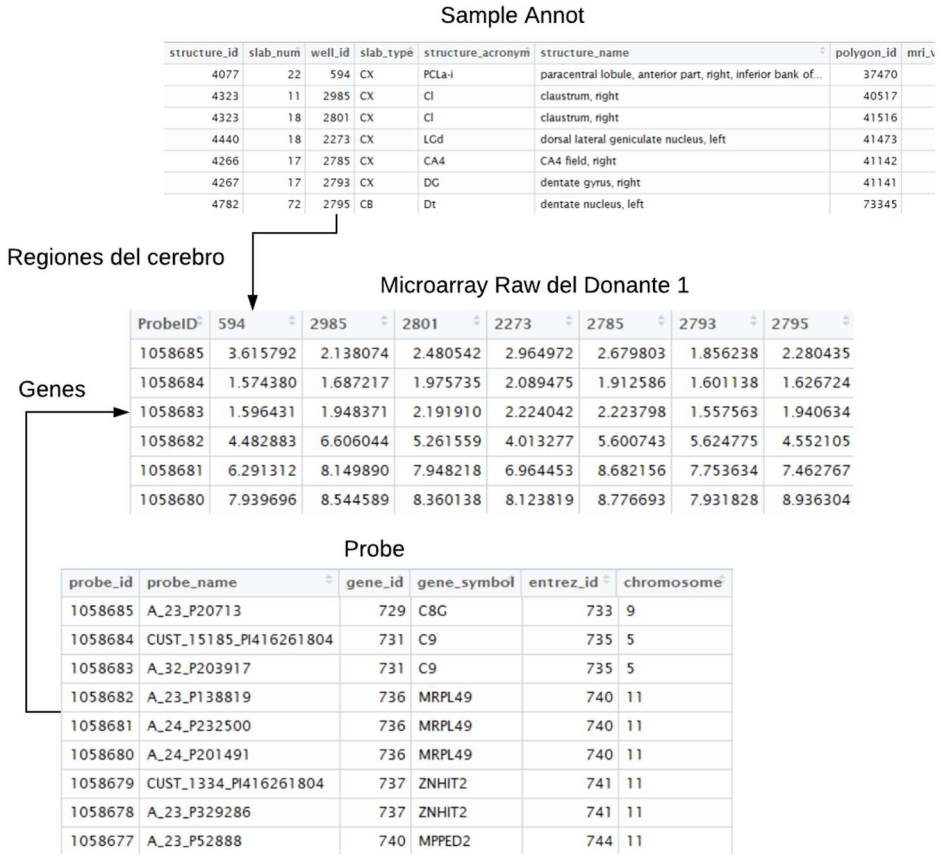

Figura 2-2: Estructura relacional del Allen Human Brain Atlas

Análisis del Allen Human Brain Atlas

En este apartado analizo la cantidad de información que el atlas posee sobre las regiones del cerebro, tal que podrían afectar a los resultados del método GetROIs que lo utiliza. Como expliqué anteriormente en esta sección, las tablas de *microarrays* poseen columnas con nombres de regiones con una cardinalidad $n \to 1, 1 < n < 48$. Esto significa que entre una y 48 columnas pueden pertenecer a una misma región del cerebro. En la Figura 2-3 muestro una representación de la cantidad de veces que cada

región del cerebro aparece en el atlas (número de veces que fue muestreada), tomando a los seis donantes juntos.

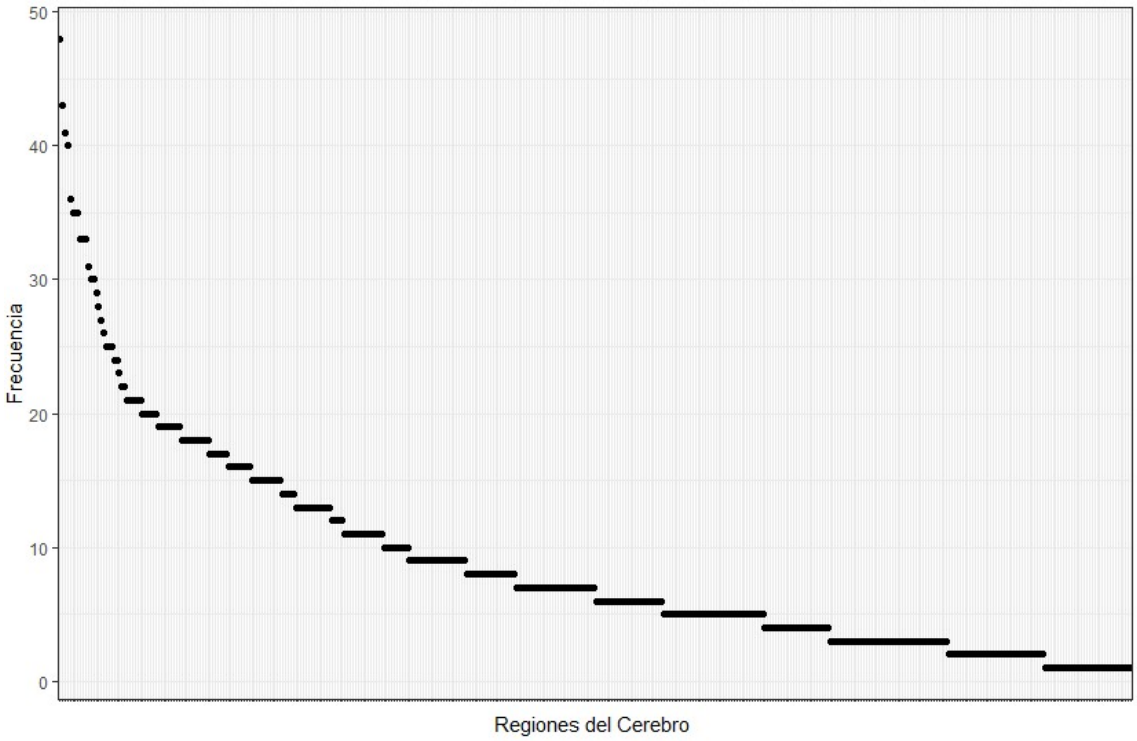

Figura 2-3. Frecuencia de regiones del cerebro entre todos los donantes del Allen Human Brain Atlas

El *putamen_left* es la región más muestreada, con 48 columnas de datos entre todos los donantes. Opuesto a esto, existen 34 regiones que solo poseen una muestra (o columna de datos) en todo el atlas. El promedio de cantidad de muestras para una región es de 8.94. Al momento de redactar este documento, estos datos no están incorporados en los algoritmos de los métodos desarrollados, ya que hacerlo requiere de una interpretación interdisciplinaria de los mismos que está más allá del alcance de este trabajo. Sin embargo, los expongo para que aquellos investigadores interesados puedan utilizar su propio criterio al respecto, sirviendo esto a futuras actualizaciones de las herramientas desarrolladas en este trabajo.

2.1.2. *Allen Human Brain Atlas* Transpuesto

Esta sección está dividida en dos partes. En la primera, describo la estructura de los datos. En la segunda, analizo la información que posee sobre los distintos genes.

Estructura del Allen Human Brain Atlas Transpuesto

Se trata de una versión similar del *Allen Human Brain* Atlas descripto en la sección 2.1.1., con la diferencia de poseer las tablas de *microarrays* transpuestas. Esto significa que las mismas poseen los identificadores de los *probes* en sus columnas y el nombre de las regiones del cerebro en sus filas. Uso esta versión del atlas en el método *GetGenes* (sección 3.2). En la Tabla 2-3 muestro la disposición de los datos en las tablas de *microarrays*, y en la Tabla 2-4 informo sus dimensiones para cada donante.

	Probe 1	*Probe* 2	…	*Probe* M
Región 1				
Región 2				
….				
Región N				

Tabla 2-3: Disposición de los datos en la tabla Raw/Standarized transpuestas

Donante	Cantidad de regiones (Filas)	Cantidad de *Probes* (Columnas)
Donante 1	942	58692
Donante 2	893	58692
Donante 3	363	58692
Donante 4	529	58692
Donante 5	470	58692
Donante 6	501	58692

Tabla 2-4: Descripción de la tabla Raw/Standarized transpuestas

Análisis del Allen Human Brain Atlas Transpuesto

En este apartado analizo la cantidad de información que posee el atlas, en lo que a genes respecta. La Figura 2-4 permite tener una idea global de la cantidad de *probes* que cada gen posee.

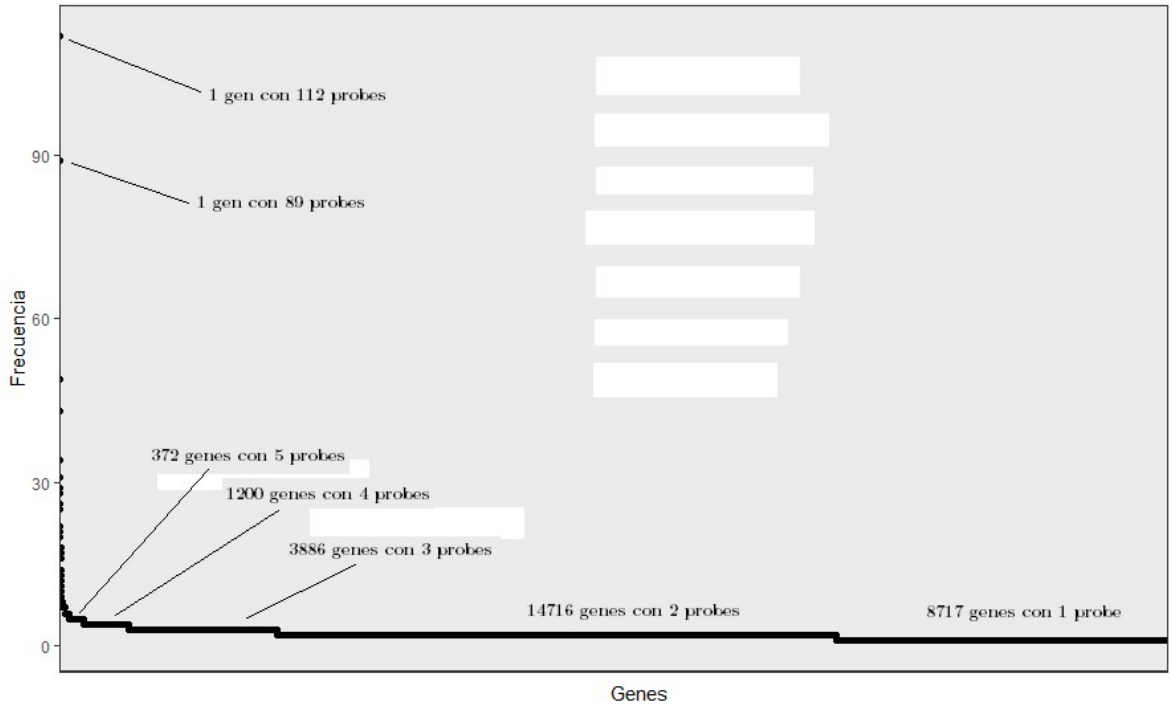

Figura 2-4. Cantidad de probes por cada gen en el Allen Human Brain Atlas

En promedio un gen posee 2 *probes*, como mínimo tiene uno y como máximo 112. Estos datos son importantes ya que el método *GetGenes* posee mecanismos que realizan operaciones sobre las columnas de los *microarrays* transpuestos (*probes*), en búsqueda de aquellos genes que cumplan determinados criterios como explico en la sección 3.2. Aunque en los resultados de la versión actual de *GetGenes* este dato no posee relevancia en los cálculos, la diferencia en la cantidad de *probes* sobre un gen sobre otro podría ser

parcialmente responsable del nivel de importancia que finalmente se le asigna al mismo. En tanto este método sea utilizado en futuras investigaciones, podremos realizar un análisis pormenorizado de esta cuestión lo que podría derivar en futuras actualizaciones que contemplen este aspecto.

2.1.3. Datos para validación y prueba del método *GetROIs*

Para probar el funcionamiento del método *GetROIs* utilizo listas de genes como entrada de esta, para obtener un conjunto de 10 regiones del cerebro ordenadas por puntaje de importancia para cada lista de entrada. Estas listas de genes provienen de dos investigaciones distintas.

Listas de genes para validación del método GetROIs

Para validar el funcionamiento del método *GetROI*s utilizo cinco listas de genes extraídas del estudio *Spatiotemporal transcriptome of the human brain* el cual detallo en la sección 1.3.4*.* Este estudio permite conocer la asociación entre un conjunto de genes y determinadas regiones del cerebro. Así es posible comparar esta información con los resultados del método *GetROIs* y evaluar sus coincidencias. Presento los detalles de las cinco listas en la Tabla 2-5. Las mismas pueden descargarse del sitio web www.brainai.science/pgl/getrois.

Nombre de la lista de genes	Cantidad total de genes	Cantidad de genes presentes en el atlas	Regiones del cerebro asociadas
CBC	**404**	**385**	***Cerebellar cortex***
HIP	**19**	**19**	***Hippocampus***
MD	**109**	**106**	***Mediodorsal nucleus of Thalamus***
STR	**103**	**100**	***Striatum***
NCX	**19**	**18**	***Neo Cortex***

Tabla 2-5. Descripción de las listas de genes para validación de la función GetROIs.

Lista de genes para prueba del método GetROIs

Para realizar una prueba sobre el método *GetROIs* uso una lista de genes provenientes del estudio *A genoma-wide association study of attempted suicide,* el cual detallo en la sección 1.3.3. En la Tabla 2-6 muestro los detalles de esta lista.

Nombre de la lista de genes	Cantidad total de genes	Cantidad de genes presentes en el atlas	Regiones del cerebro asociadas
Willour	**130**	**118**	**Desconocidas. A determinar mediante la función *GetROIs*.**

Tabla 2-6. Descripción de la lista de genes para prueba de la función GetROIs.

2.1.4. Datos para prueba del método *GetGenes*

El método *GetGenes* es probado utilizando dos tipos de datos distintos:

- Una lista de genes sobre los cuales quiero determinar la importancia de su expresión génica en un conjunto de regiones del cerebro. En este trabajo en particular uso la misma lista de genes utilizada para probar la función GetROIs proveniente del estudio *A genoma-wide association study of attempted suicide* (ver sección 2.1.3).
- Una lista de nombres de regiones del cerebro sobre las cuales quiero evaluar la importancia de la expresión génica de los genes del punto anterior. Dichos nombres deben poseer la denominación utilizada en el campo *structure_name* de la tabla *Sample_Annot* del *Allen Human Brain Atlas* (ver sección 2.1.1). Puntualmente, para realizar la prueba de la función *GetGenes* para este trabajo utilizo dos nombres de regiones del cerebro:
    - *subiculum, left*
    - *subiculum, right*

Estas dos regiones de interés provienen de un estudio que el Dr. Salas realizó sobre los resultados obtenidos tras la prueba del método *GetROIs* (ver sección 3.1.6 y 5.1).

2.2. Métodos

En este trabajo desarrollé cuatro métodos destinados a seleccionar variables sobre tablas de *microarrays* provenientes del *Allen Human Brain Atlas*, aplicando la prueba estadística *Wilcoxon* de suma de rangos, *Random Forest*, *SVM* y *KNN*.

- Uso la prueba estadística no paramétrica *Wilcoxon* de suma de rangos para seleccionar aquellas variables que poseen una distribución diferenciada entre los casos pertenecientes a una clase u otra, obteniendo así un puntaje final de importancia según el p-valor de la prueba estadística (mayor importancia de la variable a menor p-valor) sobre aquellas variables que hayan superado la corrección por comparaciones múltiples (ver sección 1.4.3).
- Uso *Random Forest*, *SVM* y *KNN* como métodos de selección de variables mediante procedimientos de eliminación de distintos tipos, a fin de alcanzar el valor más alto de exactitud o área bajo la curva *ROC* según corresponda, como explico en las secciones 0, 0 y 3.1.5 respectivamente.

3. Resultados

En esta sección describo los métodos *GetROIs* y *GetGenes* desarrollados para cumplir con los objetivos planteados (ver sección 1.1), explicando cómo uso los métodos mencionados en la sección 0 con los datos descriptos en la sección 2.1. Para el método *GetROIs* presento una definición formal y un diagrama de flujo con una explicación de cada paso. Luego detallo cada una de las cuatro subfunciones que lo componen. Luego, muestro la validación de este método utilizando listas de genes para las cuales ya conozco su asociación a regiones del cerebro puntuales. Para terminar, realizo una prueba utilizándolo en el contexto de una investigación dirigida por mi director de tesis. Para el método *GetGenes* presento una estructura similar, pero a diferencia de *GetROIs*, este método cuenta con una sola subfunción. No realizo procedimiento de validación porque hasta el momento no he tenido oportunidad de acceder a los medios para realizarlo (para más detalles, ver sección 4.2 sobre trabajos futuros). Aun así, presento los resultados de una investigación en curso (la misma en la que usé el método *GetROIs*) donde el Dr. Salas obtuvo resultados satisfactorios basados en la utilización del método *GetGenes*.

3.1. El método *GetROIs*

El método *GetROIs* sirve para encontrar regiones del cerebro posiblemente relacionadas a patologías psiquiátricas, en base a un grupo de genes cuya vinculación a dicha patología sea ya conocida. El funcionamiento de este método consiste en aplicar diversos métodos de selección de variables (regiones del cerebro) a las tablas de *microarrays* del *Allen Human Brain Atlas* para determinar cuáles de esas variables son las más apropiadas para clasificar correctamente a cada uno de los casos positivos (*probes* de los genes vinculados a una patología) del resto de los casos (*probes* de los genes sin vinculación conocida a dicha patología) según su expresión génica en cada región del cerebro.

3.1.1. Definición formal

Siendo

A: el conjunto de genes presentes en el *Allen Human Brain Atlas*

A': un subconjunto de genes relacionados con una patología psiquiátrica

X: el conjunto de todas las regiones del cerebro estudiadas en el mismo atlas

$$GetROIs(A')=X'$$

X' es un subconjunto ordenado de 10 regiones de X con las puntuaciones de importancia media más alta asignadas según las subfunciones *RandomForestFS*, *SVMFS* y *KNNFS* (que intentan clasificar correctamente A' del resto de los genes de A utilizando como variables al conjunto X) junto al método *WilcoxonFS* que también asigna una puntuación de importancia a cada elemento de X' como explico más adelante. Este enfoque de selección de variables proporciona información útil para investigadores en neurociencias, sobre cuán involucradas podrían estar ciertas zonas anatómicas del cerebro en determinadas patologías y así generar hipótesis de trabajo (ver ejemplo en sección 5.1). En la Figura 3-1 muestro un esquema general y un pseudocódigo simplificado para cada función.

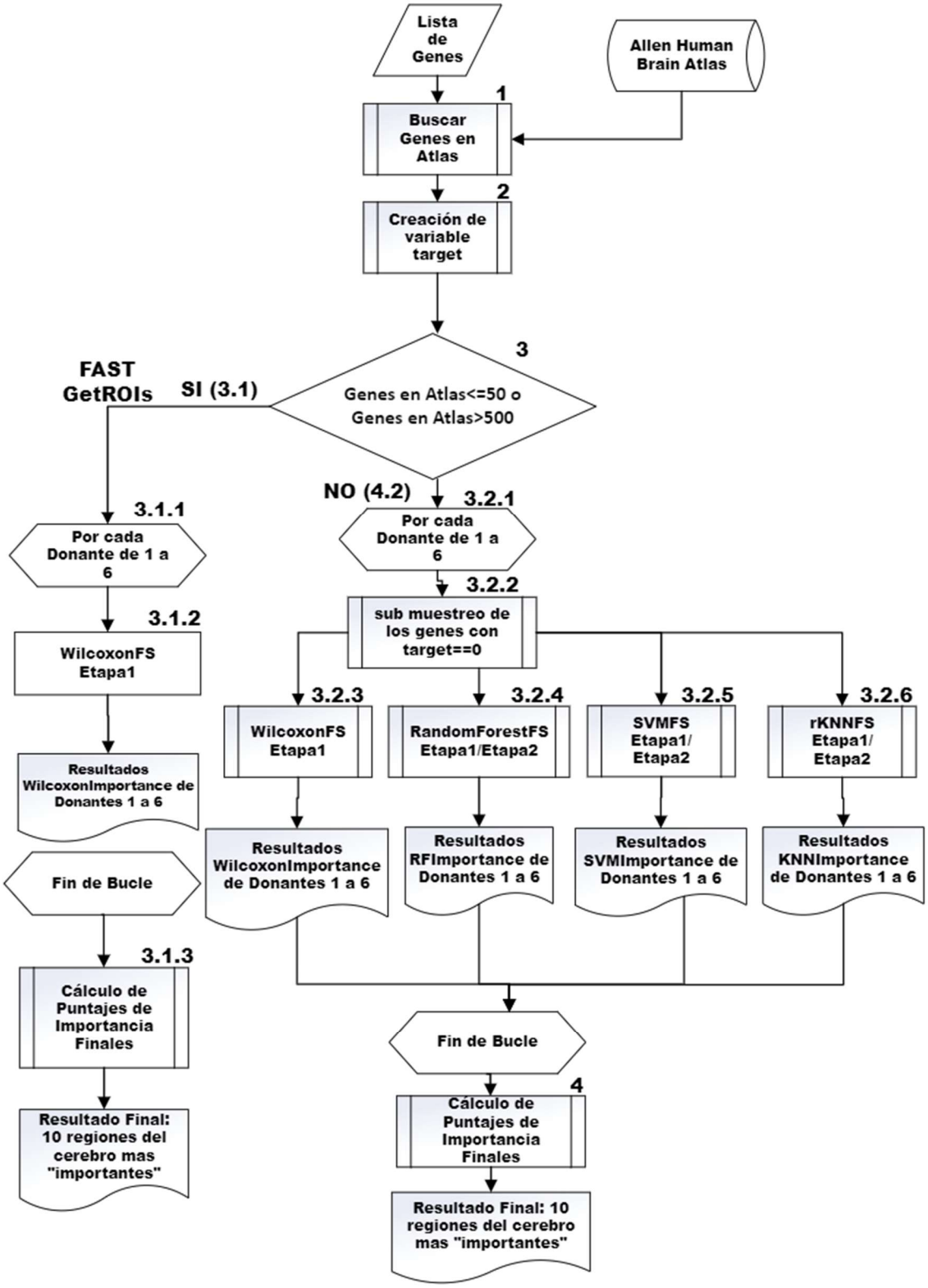

Figura 3-1. Diagrama de flujo del método GetROIs

En el primer paso cargo en memoria el *Allen Human Brain Atlas* y la lista de genes relacionados a la patología investigada. Utilizando la tabla *Probes* del atlas determino cuáles genes de la lista se encuentran efectivamente presentes en dicho atlas. A estos los denomino "genes encontrados".

1. Creo una nueva variable (*target*) en cada una de las tablas de *microarrays*, colocando un "1" en las filas donde el *ProbeID* coincide con los *ProbeIDs* de los genes de la lista encontrados en el atlas. En la Figura 3-2 grafico el proceso.

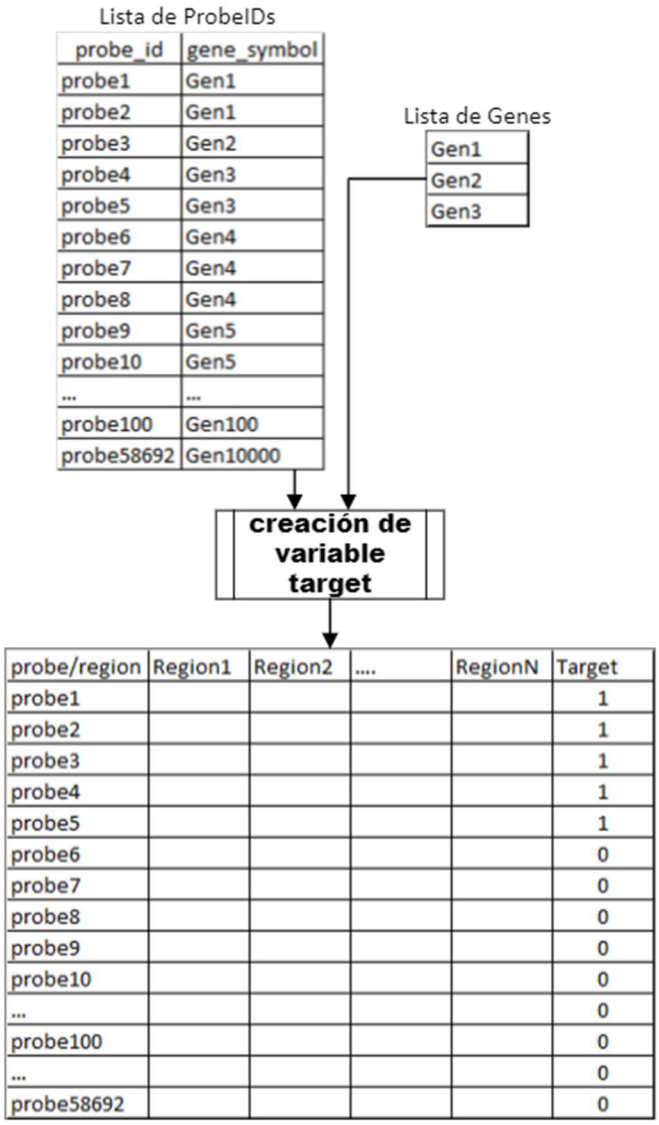

Figura 3-2. Creación de la variable target para las subfunciones WilcoxonFS, RandomForestFS, SVMFS y rKNNFS

2. De acuerdo con la cantidad de "genes encontrados" determino la utilización de la versión *FAST GetROIs* o la versión completa.

   3.1. Si la cantidad de "genes encontrados" es menor a 50 o mayor a 500, utilizo la versión *FAST GetROIs* que solo usa la subfunción *WilcoxonFS*.

      3.1.1. Por cada una de las 6 tablas de *microarrays* presentes en el atlas, ejecuto los procedimientos 3.1.2 y 3.1.3.

3.1.2. Ejecuto la subfunción *WilcoxonFS*, generando 6 listas con 10 regiones del cerebro cada una, ordenadas por puntaje de importancia.

3.1.3. Como resultado final del método *GetROIs* creo una lista de 10 regiones del cerebro (también ordenadas por puntaje de importancia) en base a las 6 listas del punto anterior y haciendo el siguiente cálculo para determinar el puntaje de importancia final de cada región:

$$RegionPuntajeFinal = \frac{1}{6}\sum_{i=1}^{6} WilcoxonFSScore_i$$

Donde $WilcoxonFSScore_i$ es el puntaje de importancia de una región determinada, para el donante i. Cuando una región no aparezca en algún donante, le asigno el valor "0".

3.2. Si la cantidad de "genes encontrados" es mayor a 50 y menor a 500, utilizo la versión de *GetROIs* que ejecuta las 4 subfunciones disponibles.

3.2.1. Por cada una de las 6 tablas de *microarrays* disponibles, ejecuto los procedimientos 3.2.2 a 3.2.6.

3.2.2. Realizo una sub-selección aleatoria (*under sampling*) de tantas filas con *target*=0 como filas con *target* = 1 haya, sobre cada una de las 6 tablas de *microarray*. En la Figura 3-3 muestro una descripción gráfica de este procedieminto.

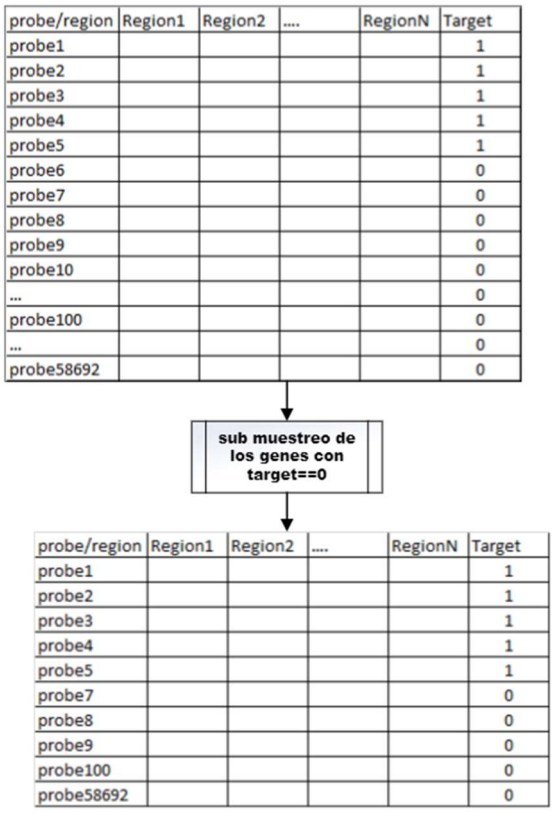

Figura 3-3. Descripción gráfica del procedimiento de sub-muestreo aleatorio

3.2.3, 3.2.4, 3.2.5 y 3.2.6. Ejecuto las subfunciones *WilcoxonFS*, *RandomForestFS*, *SVMFS* y *rKNNFS* sobre cada tabla de *microarray*, generando una lista con 10 regiones del cerebro ordenadas por puntaje de importancia por cada subfunción.

4. Realizo el cálculo final de los puntajes de importancia de cada región del cerebro, a partir de las cuatro listas de resultados creadas en los pasos 3.2.2 a 3.2.6. Para ello hago el siguiente cálculo sobre cada una de las regiones.

$$RegionPuntajeFinal = \frac{1}{6}\sum_{i=1}^{6} WilcoxonFSScore_i + RandomForesFS_i + SVMFS_i + rKNNFS_i$$

Donde $WilcoxonFSScore_i$, $RandomForesFS_i$, $SVMFS_i$ y $rKNNFS_i$ son los puntajes de importancia de una región determinada para el donante *i*, de cada una de las subfunciones. Cuando una región no aparezca el algún donante le asigno el valor "0". Luego de realizar este cálculo para cada región, conservo aquellas 10 con mejor puntaje final, ordenándolas de mayor a menor. En la sección 3.1.6 muestro los resultados finales de la función *GetROIs* calculados de esta forma (ver Tabla 3-2 a Tabla 3-7).

3.1.2. La subfunción *WilcoxonFS*

Este método utiliza la prueba estadística *Wilcoxon* de suma de rangos (ver sección 1.4.3), para identificar aquellas áreas del cerebro donde sea posible rechazar la siguiente hipótesis nula: La distribución de las expresiones génicas de un conjunto de genes de interés no difiere en su localización con respecto a la distribución de la expresión del resto de los genes disponibles en el *Allen Human Brain Atlas*. Para ello uso la función *Wilcox.test* de la biblioteca en leguaje R *Stats*[30] para realizar iterativamente la prueba de *Wilcoxon* sobre cada una de las columnas de cada uno de los *microarrays* pertenecientes a los seis donantes (analizo los resultados de cada donante individualmente). Recibe como argumento aquellas filas (genes) cuya posición en la variable objetivo posee un "1" (los genes bajo estudio) o un "0"(para el resto de los genes). La función *Wilcox.test* realiza la prueba de *Wilcoxon* comparando ambos conjuntos de genes, y esta operación se realiza sobre cada columna (regiones del cerebro) individualmente y en cada *microarray*. Tras dichas operaciones múltiples realizadas sobre cada donante, procedo a realizar la corrección de Bonferroni sobre el *p-value* derivado de cada prueba, para finalmente retener aquellas columnas cuyo *p-value* corregido sea menor o igual a 0,05. Esta es la única de las cuatro subfunciones que posee una sola etapa, ya que la función *Wilcox.test* realiza toda la tarea y no requiero de una función adicional.

3.1.3. La subfunción *RandomForestFS*

Esta subfunción requiere de la ejecución de 2 etapas distintas:

Etapa 1:

- Genero un modelo *Random Forest* base utilizando todas las variables (regiones del cerebro) disponibles para cada donante.

  - Número de iteraciones de validación cruzada = 3
  - Número de subconjuntos de validación cruzada =3
  - Métrica de comparación = exactitud en validación cruzada
  - Número de árboles generados = 500
  - mtry = Elegido mediante una búsqueda de optimización aleatoria, por la librería *Caret* (Tabla 5-1).
- Utilizo la función *rfcv* (*Random Forest Cross-Validation for feature selection*) de la librería de Lenguage R *randomForest*[31] para analizar la exactitud de la clasificación mientras va reduciendo la cantidad de variables utilizadas, según su importancia calculada sobre el modelo *Random Forest* base que generé en el paso anterior. La cantidad de variables que desecha en cada iteración es del 5%, hasta quedar con una sola variable.

  - Número de iteraciones de validación cruzada = 3
  - Número de subconjunto de validación cruzada =3
  - Porcentaje de variables eliminadas en cada paso = 5%
  - Métrica de comparación = exactitud en validación cruzada

Etapa 2:

- Creo un nuevo conjunto de datos con las variables participantes en el mejor modelo creado por la función *rfcv* en la etapa anterior. Con estos datos genero un nuevo modelo *Random Forest*. Los parámetros definidos son:
  - Número de iteraciones de validación cruzada = 3
  - Número de subconjuntos de validación cruzada =3
  - Métrica de comparación = exactitud
  - Número de árboles generados (*ntree*) = 500.
  - *mtry* = Elegido mediante una búsqueda de optimización aleatoria, por la librería *Caret* (Tabla 5-1).

3.1.4. La subfunción *SVMFS*

Esta subfunción cuenta con 2 etapas, ambas basadas en la creación de modelos de máquina de vector soporte penalizado L2.

Etapa 1:

- Realizo una transformación de la variable clase, la cual originalmente contiene los valores "1" y "0", para transformarlos en "1" y "-1" respectivamente de acuerdo con las necesidades del algoritmo utilizado.
- Genero el modelo *SVM* penalizado utilizando la librería de lenguaje R *penalizedSVM*[32].
    - Los parámetros de Lambda 1 y Lambda2 son ajustados automáticamente por el algoritmo utilizando búsqueda aleatoria.

Etapa 2:

- Obtengo las variables utilizadas en el modelo final luego del proceso de penalización.
- Genero un nuevo modelo *SVM* con penalización L2 usando la biblioteca de funciones *Caret*, con las variables obtenidas en el paso anterior. Los parámetros para ajustar son:
    - *Loss*: El algoritmo prueba las dos opciones disponibles, usando finalmente la mejor de acuerdo con la exactitud obtenida en combinación con los demás parámetros:
        - 1= *L2-regularized L2-loss support vector classification (dual)*
        - 2 = *L2-regularized L2-loss support vector classification (primal)*
    - *Cost*: El algoritmo utiliza valores entre 0 y 1 en busca de aquel que proporcione la mayor exactitud en combinación con el resto de los parámetros ajustados.
    - *Weight*: El algoritmo prueba con tres valores distintos y finalmente usa el que mejor exactitud haya reportado en combinación con el resto de los parámetros.

    Los valores finalmente utilizados para cada uno de los parámetros pueden observarse en la Tabla 5-1. Utilizo la penalización L2 para que ninguna de las variables seleccionadas en el paso anterior quede excluida del modelo como sí podría suceder si elijo la penalización L1 (ver sección 1.4.3). Los parámetros definidos son:

        - Número de iteraciones de validación cruzada = 3
        - Número de subconjunto de validación cruzada =3
        - Métrica de comparación = exactitud
- Calculo la importancia de cada variable en base al modelo creado en el paso anterior. Dado que *SVM* no posee un método de cálculo de importancia nativo, uso un método de filtrado mediante la función *varImp*[33]. Esta función calcula la curva *ROC* de cada variable predictora individualmente, y el área bajo dicha curva es utilizada como métrica de importancia.

3.1.5. La subfunción *rKNNFS*

Esta subfunción cuenta también de dos etapas distintas.

Etapa 1:

En esta etapa genero múltiples modelos del tipo k-vecinos más cercanos, eliminando variables de forma recursiva, haciendo uso de la función *rknnBeg* de la biblioteca de funciones en lenguaje R *rKNN*[34].

- Utilizo el método *knn.beg*, para crear modelos con eliminación recursiva de variables. Los parámetros para definir son:
  - Cantidad de vecinos más cercanos = 40 (definido de forma empírica de acuerdo con mi experiencia en la utilización de esta herramienta en otros casos).
  - Número de variables a ser elegidas en cada iteración = 50 (también definido empíricamente).
- Uso la función *bestset* de la misma biblioteca para extraer aquellas variables que generaron el mejor modelo de acuerdo con la métrica de exactitud media de los modelos generados en el paso anterior.

Etapa 2:

- Genero un modelo *KNN* usando la biblioteca *Caret*, con las mejores variables encontradas en el paso anterior. El parámetro para ajustar es el número máximo de vecinos considerados (*kmax*), para lo cual *Caret* genera automáticamente 3 valores y crea un modelo diferente para cada uno de ellos conservando aquel cuyo *kmax* haya reportado la mayor exactitud. La evaluación de los mencionados modelos la realizo utilizando el mismo criterio que en los demás métodos:
  - Número de iteraciones de validación cruzada = 3
  - Número de subconjuntos de validación cruzada =3
  - Métrica de comparación = exactitud
- Calculo la importancia de las variables utilizando la función *varImp*, ya que tampoco hay un método nativo implementado para este algoritmo que permita determinar la importancia de las variables utilizadas.

3.1.6. Fases de validación y prueba del método *GetROIs*

Esta sección está dividida en dos partes. En la primera, presento los procedimientos de validación del método *GetROIs* que me permiten comparar sus resultados contra información verificada por una investigación científica publicada. En la segunda parte, pruebo el método desarrollado en una investigación llevada a cabo por mi director de tesis (detalles en la sección 5.1), obteniendo resultados útiles que están pendientes de publicación.

Fase de Validación

Para validar el funcionamiento del método GetROIs procesé las cinco listas de genes provenientes del estudio titulado *Spatiotemporal transcriptome of the human brain* (ver sección 2.1.3) utilizando dicho método. Cada una de estas listas está asociada a un conjunto de regiones del cerebro según el mencionado estudio (al tamaño de este conjunto lo denomino *nROIs*). El resultado de *GetROIs* consta de 10 regiones ordenadas por importancia, para cada lista procesada. A la cantidad de regiones coincidentes entre el conjunto de regiones asociadas a la lista y las resultantes de GetROIs la denomino *nGetROIs*. Tomando en cuenta que el *Allen Human Brain Atlas* cuenta con 414 regiones distintas muestreadas en total, creé un índice llamado *ÍndiceGetROIs* a fin de obtener un valor numérico que me permita evaluar la calidad del resultado en función de los datos mencionados:

Sea,

nGetROIs la cantidad de regiones del cerebro efectivamente asociadas a la lista de genes (a las que llamo "regiones correctas"), según la investigación mencionada, que aparece entre las 10 regiones resultantes del método GetROIs.

nROIs el número total de regiones del cerebro asociadas a la lista de genes.

$$\acute{I}ndiceGetROIs_{(nGetROIs,nROIs)} = \begin{cases} 1 \; si \; nGetROIs = nROIs \\ \dfrac{nGetROIs}{10} * \left(1 - \dfrac{nROIs}{414}\right) \; si \; \text{nGetROIs} < \text{n}ROIs, \end{cases}$$

Por lo tanto, $0=<\acute{I}ndiceGetROIs=<1,$ siendo $0=<nGetROIs<=10$ y $1=<nROIs=<414$

La Tabla 3-1 muestra los valores de *nROIs* y *nGetROIs* junto al cálculo del *ÍndiceGetROIs* para cada lista de genes procesada. También detallo en qué casos utilicé la modalidad *FAST GetROIs* (ver sección 3.1) debido a la reducida cantidad de genes encontrados en el atlas sobre los que posee originalmente cada lista.

Lista	*nROIs*	*nGetROIs*	*ÍndiceGetROIs*	*FAST PGL*
CBC	49	10	0.88	SI
HIP	16	10	0.96	NO
NCX	136	10	0.67	SI
STR	13	8	0.77	NO
MD	1	1	1	NO

Tabla 3-1: Regiones del cerebro asociadas a la lista de genes provenientes del estudio Spatiotemporal transcriptome of the human brain

La validación utilizando la lista MD (Tabla 3-4) obtuvo el mayor valor para *ÍndiceGetROIs* ya que dicha lista de genes posee una sola región asociada, la cual apareció entre las 10 regiones resultantes de *GetROIs*, sobre 414 regiones presentes en el atlas. Y, como

efectivamente encontró todas las regiones asociada adquirió el mayor valor del índice. El menor valor la obtuvo la validación con la lista NCX (Tabla 3-5) que, aunque obtuvo 10 "regiones correctas", lo hizo sobre 136 regiones que estaban asociadas sobre 414 que tiene el atlas. Como la probabilidad de encontrar "regiones correctas" por azar es mayor, esto repercutió negativamente en su valor de *ÍndiceGetROIs*.

Al momento de utilizar este método no poseo aún un criterio de corte ni acción a tomar en función de este índice. Pero en tanto pueda realizar más pruebas de validación, espero poder sacar conclusiones sobre la efectividad del método en diversos escenarios y saber así qué calidad de resultados esperar en futuras investigaciones, como lo hice en el proceso de prueba que describo más adelante en esta sección. En las siguientes tablas las columnas "Posición", "Regiones del cerebro" y "Puntaje Final" muestran los resultados de procesar cada una de las cinco listas con la función *GetROIs*. De ellas obtuve el valor *nGetROIs* (usado en la fórmula *ÍndiceGetROIs*) contando cuántas regiones coinciden con las asociadas a cada lista de genes según la investigación (marcadas con *X* en la columna "Asociada"). La columna "Frec" indica la cantidad total de veces que cada región aparece en el *Allen Human Brain Atlas*, sumando las apariciones en las 6 tablas de *microarrays*. Aunque actualmente este dato no tiene incidencia en el *ÍndiceGetROIs*, lo informo para que los investigadores interesados en estos resultados puedan sacar sus conclusiones al respecto y, producto del intercambio de ideas y experiencias con ellos, en el futuro cercano pueda evaluar su posible incorporación en el índice y en el cálculo del resultado del método *GetROIs* en tanto el mismo sea también utilizado en otras investigaciones.

Posición	Región del cerebro	Puntaje Final	Asociada	Frec
1	*Crus I, left, paravermis*	4.54	X	11
2	*Crus II, left, lateral hemisphere*	4.50	X	17
3	*Crus I, left, lateral hemisphere*	4.42	X	19
4	*VI, left, lateral hemisphere*	2.79	X	15
5	*VIIIA, left, lateral hemisphere*	2.63	X	13
6	*VIIB, left, lateral hemisphere*	2.13	X	15
7	*Crus II, left, paravermis*	1.71	X	9
8	*V*	1.63	X	8
9	*III, left, lateral hemisphere*	1.60	X	4
10	*VIIIA, left, paravermis*	1.38	X	9

Tabla 3-2. Resultado del método GetROIs para la lista de genes CBC

Posición	Región del cerebro	Puntaje Final	Asociada	Frec
1	*CA2 field, left*	10.00	X	19
2	*subiculum, left*	7.16	X	25
3	*CA1 field, left*	6.50	X	24
4	*CA3 field, left*	6.00	X	21
5	*CA4 field, left*	3.66	X	20
6	*dentate gyrus, left*	3.66	X	21
7	*CA2 field, right*	2.50	X	9
8	*subiculum, right*	2.17	X	9
9	*CA1 field, right*	2.16	X	12
10	*parahippocampal gyrus, left, lateral bank of gyrus*	1.83	X	26

Tabla 3-3. Resultado del método GetROIs para la lista de genes HIP

Posición	Región del cerebro	Puntaje Final	Asociada	Frec
1	*rostral group of intralaminar nuclei, left*	7.79		15
2	*medial group of nuclei, left*	7.21	X	18
3	*lateral group of nuclei, left, dorsal division*	4.88		13
4	*posterior group of nuclei, left*	4.63		6
5	*medial geniculate complex, left*	4.21		9
6	*anterior group of nuclei, left*	3.29		19
7	*lateral group of nuclei, left, ventral division*	3.17		20
8	*rostral group of intralaminar nuclei, right*	2.67		7
9	*lateral group of nuclei, right, ventral division*	2.33		13
10	*lateral group of nuclei, right, dorsal division*	2.25		5

Tabla 3-4. Resultado del método GetROIs para la lista de genes MD

Posición	Región del cerebro	Puntaje Final	Asociada	Frec
1	*middle frontal gyrus, left, superior bank of gyrus*	3.83	X	33
2	*superior frontal gyrus, left, lateral bank of gyrus*	2.67	X	40
3	*angular gyrus, left, inferior bank of gyrus*	2.67	X	18
4	*inferior occipital gyrus, left, inferior bank of gyrus*	2.67	X	15
5	*angular gyrus, right, inferior bank of gyrus*	2.33	X	7
6	*inferior temporal gyrus, left, lateral bank of gyrus*	1.83	X	24
7	*superior parietal lobule, left, inferior bank of gyrus*	1.83	X	23
8	*middle frontal gyrus, left, inferior bank of gyrus*	1.67	X	30
9	*paracentral lobule, anterior part, left, inferior bank of gyrus*	1.67	X	12
10	*inferior temporal gyrus, left, bank of mts*	1.67	X	30

Tabla 3-5. Resultado del método GetROIs para la lista de genes NCX

Posición	Región del cerebro	Puntaje Final	Asociada	Frec
1	*putamen, left*	7.67	X	48
2	*head of caudate nucleus, left*	7.21	X	22
3	*body of caudate nucleus, left*	6.21		25
4	*tail of caudate nucleus, left*	5.58	X	15
5	*nucleus accumbens, left*	4.17	X	13
6	*head of caudate nucleus, right*	2.38	X	10
7	*body of caudate nucleus, right*	2.33		11
8	*olfactory tubercle, left*	2.17	X	5
9	*tail of caudate nucleus, right*	2.04	X	8
10	*putamen, right*	1.96	X	14

Tabla 3-6. Resultado del método GetROIs para la lista de genes STR

Fase de Prueba

Tras la fase validación, pruebo el método *GetROIs* utilizando la lista de genes *Willour* obtenida de la investigación que menciono en la sección 1.3.3 para la cual, y a diferencia de las listas de validación, no se conocen regiones del cerebro específicamente asociadas a la misma. Muestro el resultado obtenido en la Tabla 3-7 (columnas "Posición", "Regiones del cerebro" y "Puntaje Final").

Tras obtener estos resultados, el Dr. Salas realizó un estudio de conectividad funcional en estado de reposo sobre 410 pacientes (130 presentaron intentos de suicido y 280 no los presentaron). La conectividad entre las regiones resultantes del método *GetROIs* y el resto de las áreas del cerebro definidas anatómicamente (*ROI-to-ROI*) y entre las regiones resultantes del método *GetROIs* y posibles *clusters* de *voxels* en todo el cerebro (*seed-to-voxel*). En la sección 5.1 presento los detalles sobre este procedimiento.

Nota: Al momento de realizar esta investigación, usamos una versión inicial de la actual herramienta *GetROIs*, resultando en un listado de regiones parcialmente diferente al generado con la presente versión usada en este trabajo. De las diez regiones de la Tabla 3-7 solo tres coinciden con las generadas en aquel momento: *corpus callosum*, globus *pallidus, internal segment, left* y *subiculum, left* (marcadas con * en la columna "Asociada"). El resto de las regiones no coincidentes no fueron estudiadas aún. Sin embargo, esto no afecta la validez de los resultados en este trabajo, ya que llego a la misma conclusión con ambas versiones, porque la única región de interés encontrada está incluida en ambos resultados finales.

El estudio *ROI-to-ROI* realizado reveló un nivel de conectividad significativamente mayor entre el subículo izquierdo (la región con el quinto puntaje final más alto) y la habénula en pacientes con intento de suicidio en el pasado. En el estudio *seed-to-voxel*, la conectividad entre el subículo izquierdo y un *cluster* de *voxels* en el giro frontal medio también mostró mayor conectividad en pacientes con intento de suicidio en el pasado. Dado esto, el Dr. Salas consideró al subículo como una región involucrada en la propensión a cometer intentos de suicidio. Este resultado forma parte de una investigación pendiente de publicación de la cual presento detalles preliminares en la sección 5.1

Posición	Región del cerebro	Puntaje Final	Asociada	Frec
1	*corpus callosum*	7.04	*	13
2	*superior frontal gyrus, left, medial bank of gyrus*	1.79		43
3	*IX, left, paravermis*	1.75		9
4	*medial group of nuclei, left*	1.67		18
5	*subiculum, left*	1.58	X*	25
6	*middle frontal gyrus, left, inferior bank of gyrus*	1.33		30
7	*posterior group of nuclei, left*	1.25		6
8	*globus pallidus, internal segment, left*	1.25	*	11
9	*reticular nucleus of thalamus, left*	1.25		15
10	*putamen, left*	1.21		48

Tabla 3-7. Resultado del método GetROIs para la lista de genes Willour

3.1.7. Análisis de los parámetros y métricas del método *GetROIs*

En esta sección muestro y analizo datos relacionados a la construcción de modelos en Etapa 1 y Etapa 2 para las subfunciones que componen este método (en la sección 3.1 explico el funcionamiento por etapas). Así es que dividí esta sección en dos partes. En la primera detallo los parámetros y métricas referentes a la primera etapa de funcionamiento de las subfunciones *WilcoxonFS* (que solo funciona en una etapa), *RandomForestFS, SVMFS* y *rKNN*. Lo propio hago en la segunda parte, analizando la segunda etapa para las mencionadas subfunciones excepto para *WilcoxonFS* por lo ya comentado.

Etapa 1: Descripción de los conjuntos de datos utilizados.

Como describo en la sección 3.1, antes de ingresar a las subfunciones del método GetROIs, las tablas de *microarrays* pasan por un proceso de submuestreo aleatorio a fin de balancear la variable *target*. Por lo tanto, pasan de tener la cantidad de filas (*probes*) mostrado en la Tabla 2-1 a las cantidades mostradas en la Tabla 3-8. La cantidad de columnas (regiones del cerebro) de cada *microarray* no cambia con respecto a las originales (ver Tabla 2-2).

Listas	Cantidad de Filas
CBC	1876
MD	628
STR	626
Willour	622

Tabla 3-8. Cantidad de filas de las tablas de microarray en la Etapa 1

Etapa 1: Cantidad de variables vs exactitud media

En la Etapa 1 de las subfunciones *RandomForestFS*, *SVMFS* y *rKNNFS* se produce la selección de variables en las que se generan distintos modelos con diferente cantidad y combinaciones de variables, en búsqueda de la exactitud media más alta. La Figura 3-4 muestra la exactitud media (*mean accuracy*) de todos los modelos generados en las sub funciones *RandomForestFS* y *rKNNFS*, para todos los donantes, al momento de seleccionar distinta cantidad de variables para obtener la mayor exactitud posible usando las funciones *rfcv* (0 paso 2) y *knn.Beg* (3.1.5 paso 1) respectivamente. Para ambas funciones la exactitud media alcanza valores superiores a 0.80 utilizando menos de 100 variables en todas las listas de genes de validación en las que las usé (CBC, STR y MD), y dichos valores de métrica suben paulatinamente cuando utilizo un mayor número de variables alcanzando valores máximos de 0.95 con 285 variables para *RandomForestFS* y 0.92 con 30 variables para *rKNNFS* en ambos casos sobre la lista STR. Al utilizar la lista de prueba *Willour* (cuya exactitud media está graficada con puntos negros), observo que la función *rKNN* alcanza su máxima exactitud media de 0.59 con 30 variables, mientras *RandomForest* lo hace con 6 variables. Y la máxima exactitud media es de 0.66 y la alcanza *RandomForestFS*. El método *penalizedSVM* utilizado por la función *SVMFS* (0 paso 2) no cuenta con la información necesaria para poder realizar dicha comparación.

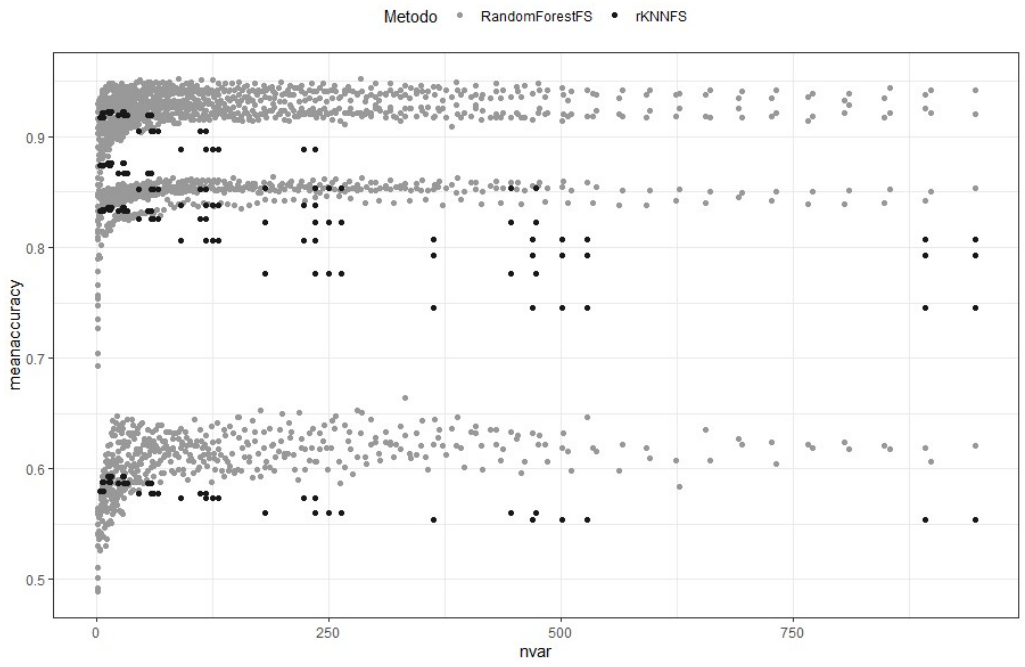

Figura 3-4. Exactitud media de los modelos iterativos para Random Forest y rKNN en la Etapa 1 de selección de variables

Etapa 2: Descripción de los conjuntos de datos utilizados

En la Etapa 2 los modelos *RandomForestFS*, *SVMFS* y *rKNNFS* trabajan con las variables (regiones del cerebro) elegidas en la Etapa 1, y conservan la misma cantidad de filas o *probes* (ver Tabla 3-8). La Figura 3-5 muestra ambos valores para cada donante y lista de genes junto al número de columnas tras ser seleccionadas en base a las elegidas en la Etapa 1.

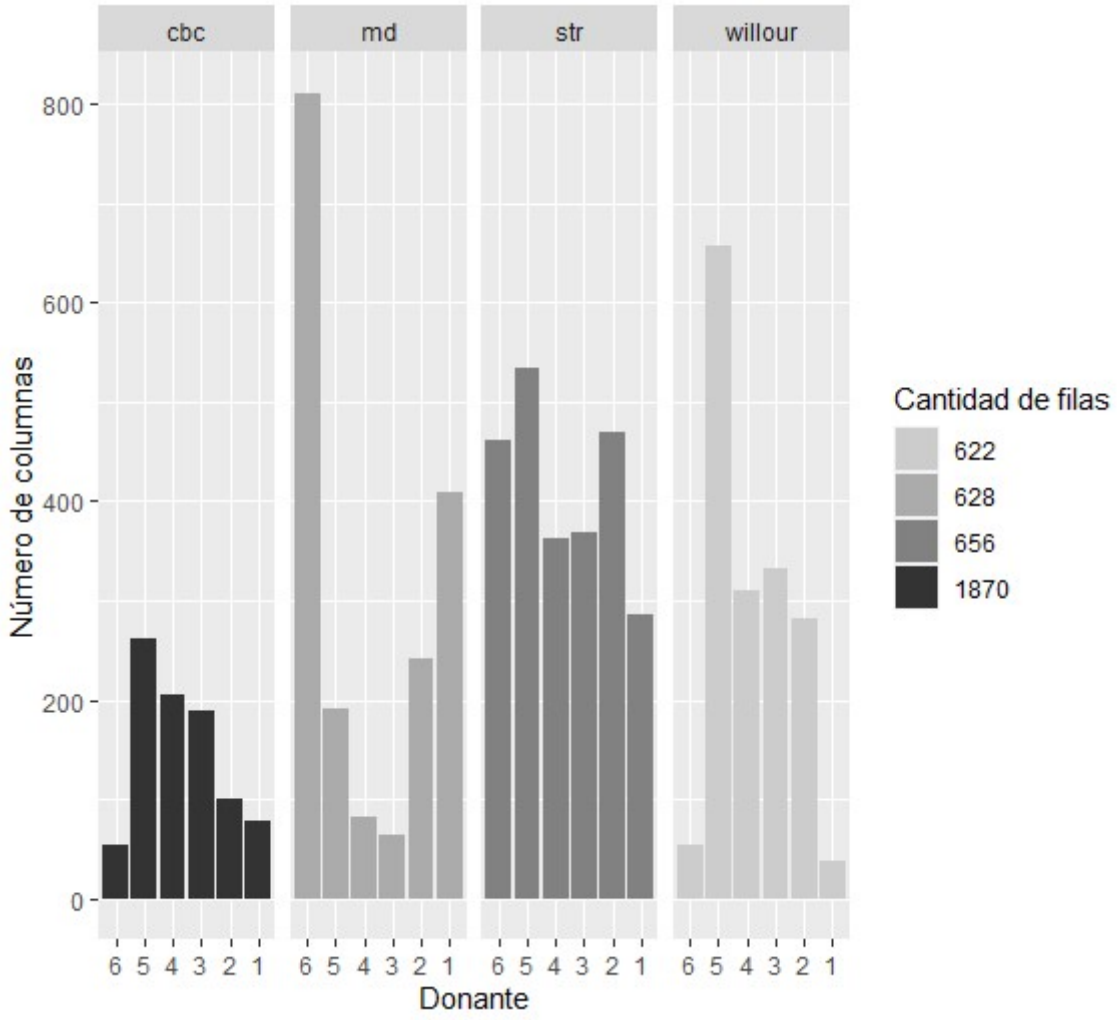

Figura 3-5. Cantidad de columnas de los datos utilizados para crear los modelos finales (Etapa 2)

Etapa 2: Métricas e hiperparámetros de los modelos finales

Tras haber determinado la cantidad óptima de variables por medio de las tres funciones encargadas de esa tarea (*rfcv, KNN.Beg* y *penalizedSVM*) genero los modelos definitivos para cada uno de los seis donantes. En la Figura 3-6 muestro la exactitud media de cada uno de ellos para cada donante, en cada una de las listas de validación donde se utilizaron (CBC, MD y STR) y en la lista de prueba *Willour*. Sobre la lista CBC, observo la utilización de menos de 400 variables para la mayoría los modelos sobre los seis donantes, similar a lo que sucede sobre la lista MD y *Willour*. Para la lista STR uso más de 400 variables en cerca de la mitad de los modelos generados, y la exactitud media es la más alta entre todos los modelos generados incluyendo los de las otras listas. La

principal observación de este análisis es que, al menos para las listas de genes procesadas hasta el momento, la mayor parte de los modelos utiliza menos de 400 variables, siendo que las tablas de *microarrays* de los donantes poseen un promedio de 616 variables.

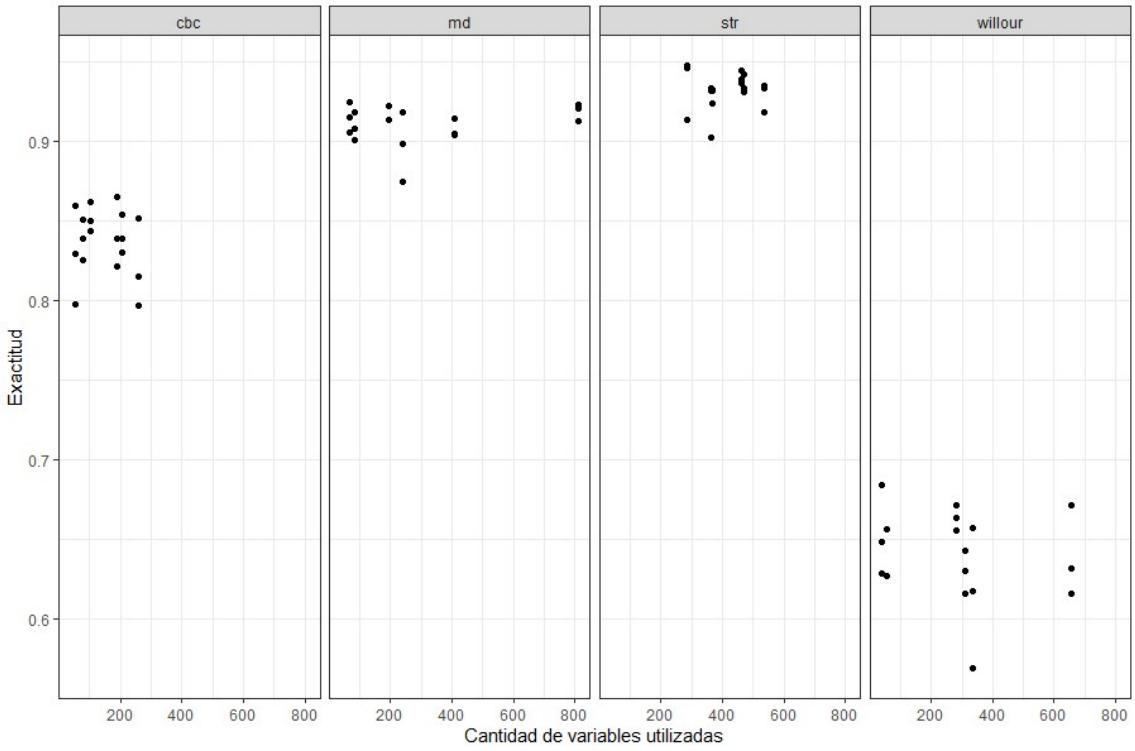

Figura 3-6. Exactitud de los modelos finales en relación con la cantidad de variables utilizadas en los mismos (Etapa 2)

La Figura 3-7 muestra la exactitud de cada uno de los modelos finales, sobre los que calculé la importancia de cada variable (región del cerebro), para cada donante y sobre cada una de las listas de validación y sobre la de prueba *Willour*. En la misma observo que las listas utilizadas a modo de validación poseen valores de exactitud más elevados que en la lista *Willour*. Esto es consistente con lo esperado, obteniendo los mayores valores de exactitud en la fase de validación (exactitud media de 0.89 entre todos los modelos de todos los donantes para las 3 listas de validación sobre las que se generan) donde utilicé listas de genes específicamente relacionadas con determinadas regiones del cerebro, en comparación con la lista *Willour* proveniente de un estudio *GWAS* que no implica la asociación con ninguna región del cerebro en particular (exactitud media de 0.64 entre todos los modelos generados sobre dicha lista).

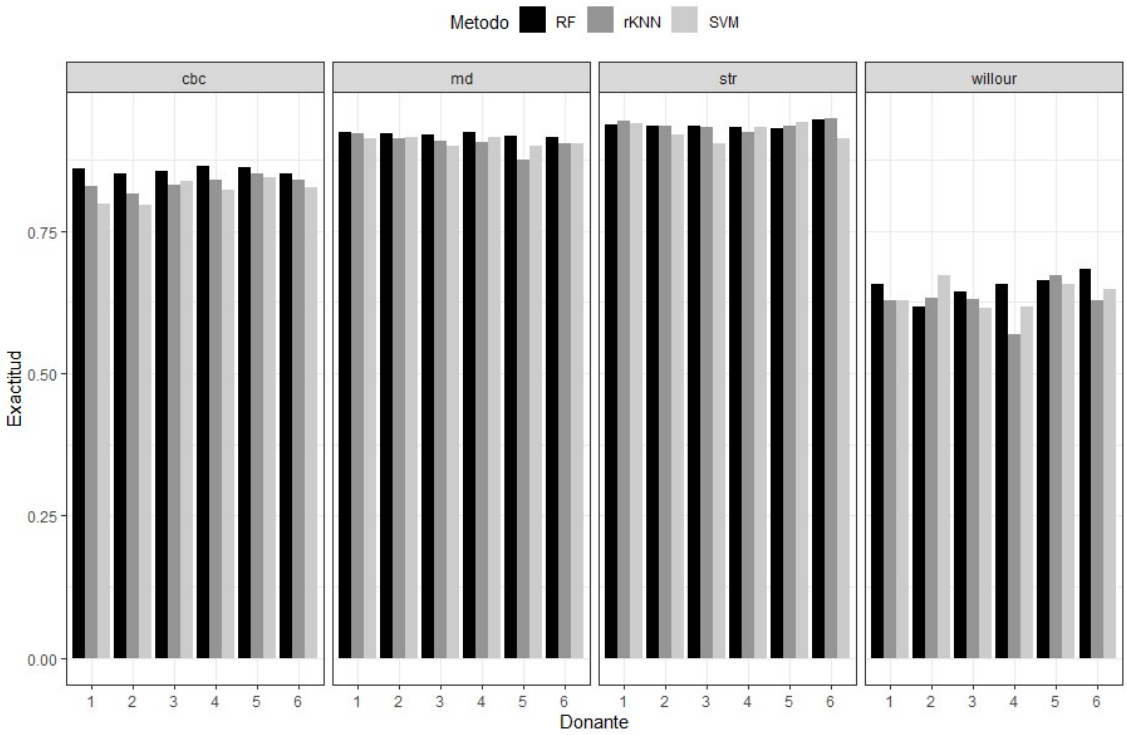

Figura 3-7. Exactitud de los modelos finales de los métodos RF, KNN y SVM para cada donante, sobre cada lista (Etapa 2)

A continuación, grafico los valores de los parámetros ajustados para cada uno de los tres tipos de modelos finales generados (*Random Forest, SVM* y *KNN*) para determinar la importancia de cada región del cerebro según cada donante. Cada gráfico cuenta con 24 puntos, ya que para cada una de las cuatro listas creé 6 modelos (uno por cada donante). Para el caso de *Random Forest*, el único parámetro ajustado es mtry. En la Figura 3-8 puede observarse el valor que adquirió este parámetro en cada modelo, luego del ajuste automático realizado por la función *Caret* en la subfunción *RandomForestFS* (ver sección 0). Excepto por dos modelos generados, todos usan menos de 200 variables, y 20 de ellos menores a 150.

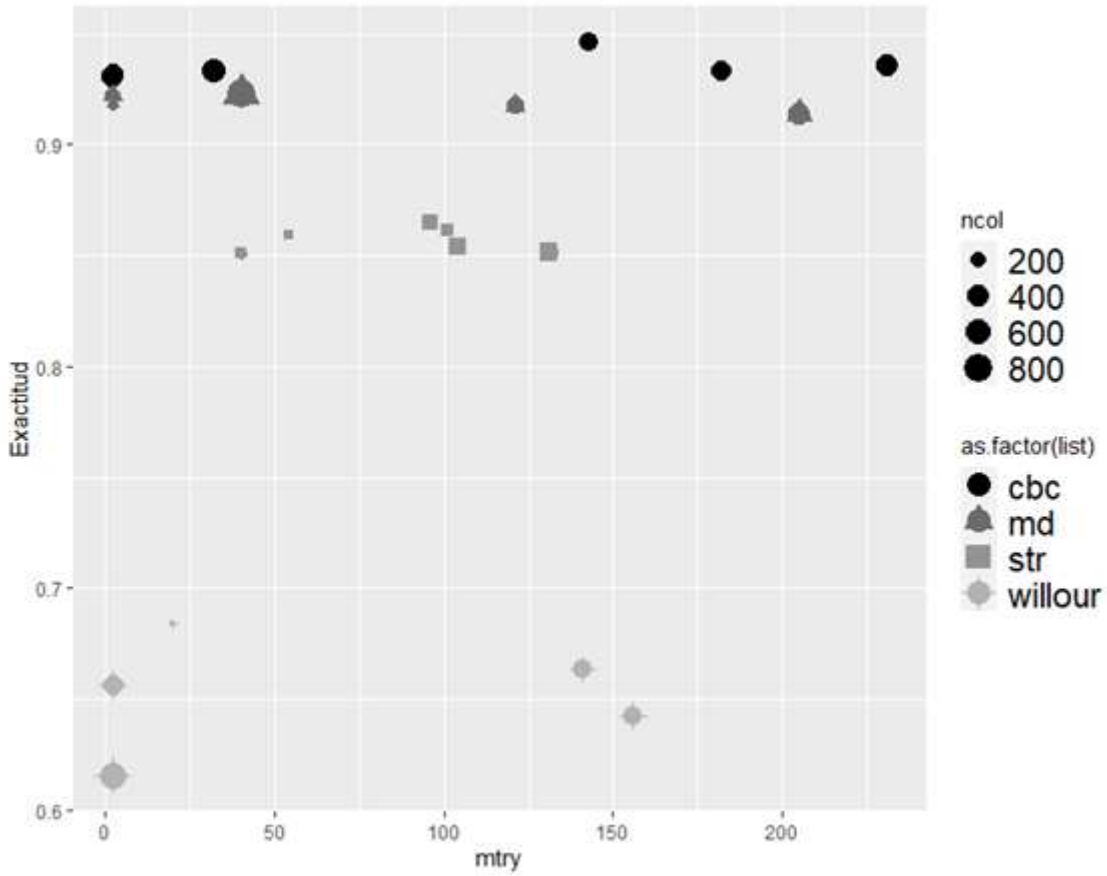

Figura 3-8. Valores del parámetro mtry de cada modelo Random Forest final según exactitud y cantidad de columnas, para cada lista (Etapa 2)

Para los modelos *KNN*, el único parámetro ajustado en *kmax* (sección 3.1.5), y en la Figura 3-9 se observa que en 20 de los 24 modelos utilizo *kmax*=9.

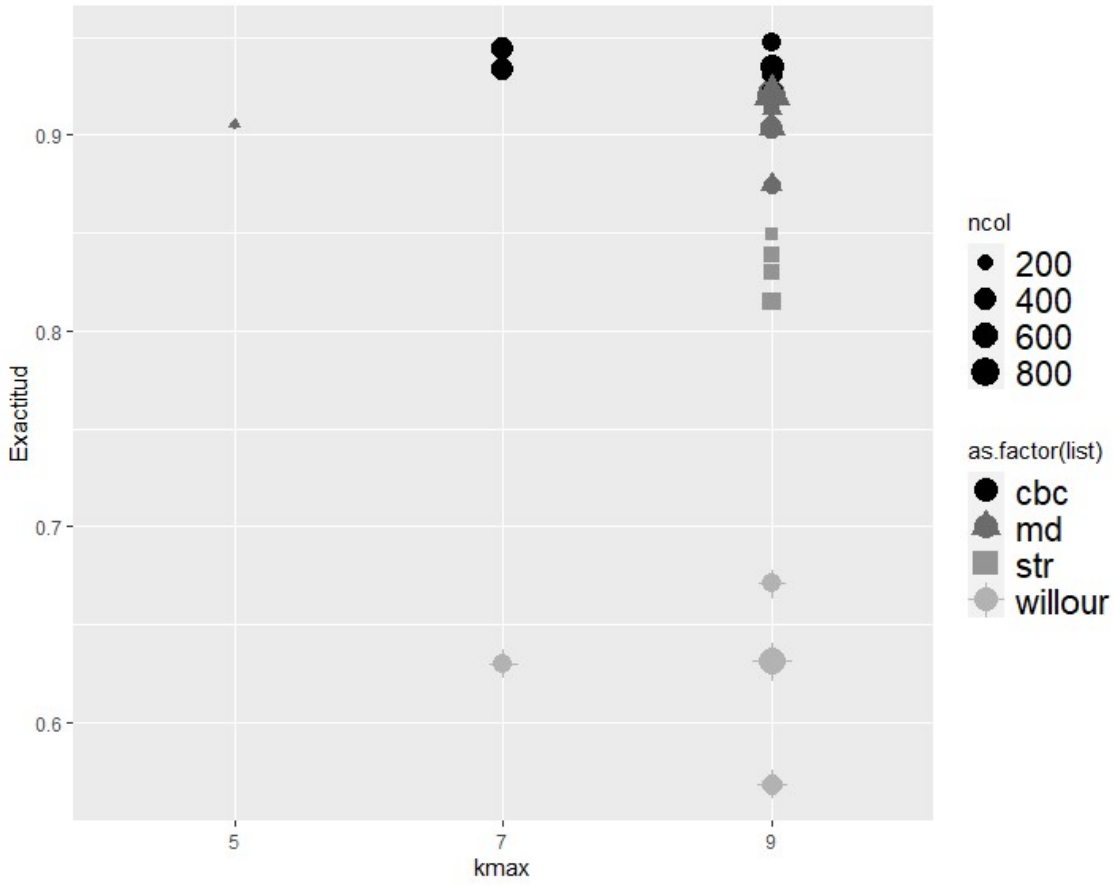

Figura 3-9. Valores del parámetro kmax de cada modelo KNN final según exactitud y cantidad de columnas, para cada lista (Etapa 2)

Los modelos *SVM* ajustan 3 parámetros: *Loss, Cost* y *Weight*. La Figura 3-10 muestra los valores usados para Loss. Como explico en la sección 0, el valor 1 hace referencia a la utilización de la penalización *L2-regularize L2-loss (dual)*, y 2 para la forma *primal*. Una explicación más detallada de estos conceptos se encuentra en la sección 1.4.3. Puede observarse la predominancia de la modalidad dual en los modelos de mayor exactitud, excepto para la lista STR donde uso únicamente la modalidad *primal*.

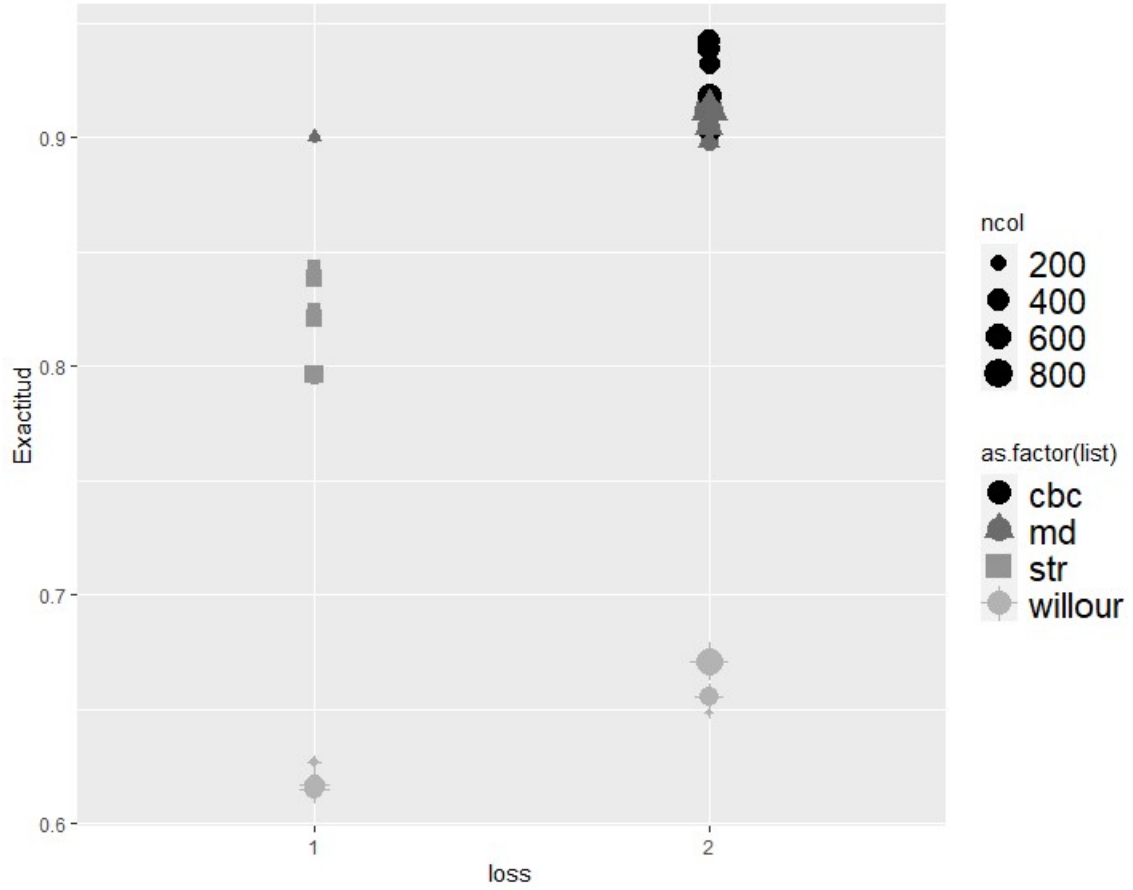

Figura 3-10. Valores del parámetro loss de cada modelo SVM final según exactitud y cantidad de columnas, para cada lista (Etapa 2)

La Figura 3-11 muestra la relación entre el valor tomado por el parámetro *cost* en cada modelo, y la exactitud lograda por los mismos. Lo propio sucede en la Figura 3-12 para el parámetro *Weight*.

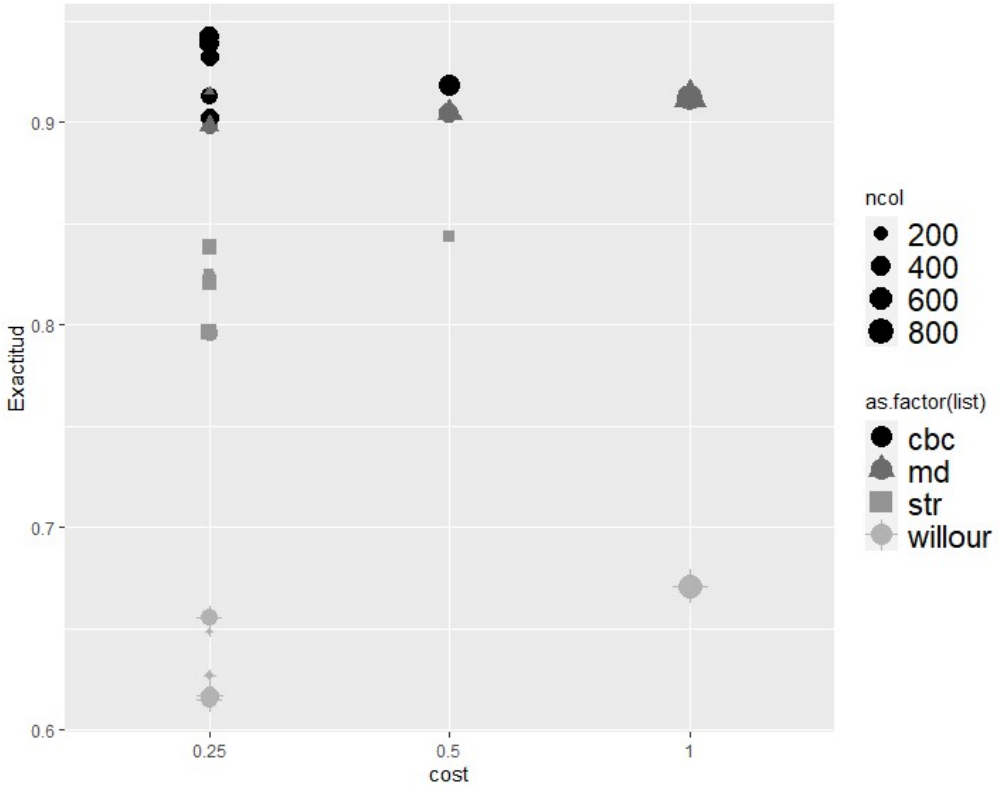

Figura 3-11. Valores del parámetro cost de cada modelo SVM final según exactitud y cantidad de columnas, para cada lista (Etapa 2)

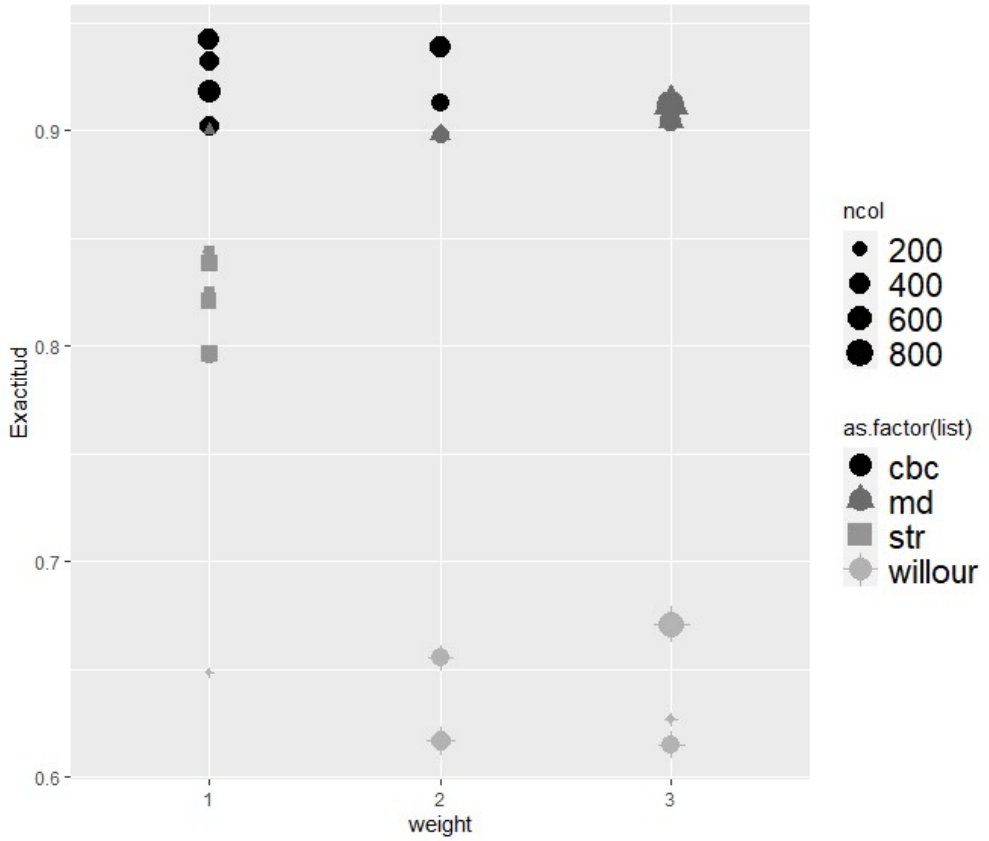

Figura 3-12. Valores del parámetro weight de cada modelo SVM final según exactitud y cantidad de columnas, para cada lista.

3.2. El método *GetGenes*

3.2.1. Definición formal

Sea

A: El conjunto de genes presentes en el *Allen Human Brain Atlas*

A': Un subconjunto de genes de A, relacionados con una patología psiquiátrica

X: El conjunto de todas las regiones del cerebro presentes en el mismo Atlas

X': Un subconjunto de regiones de interés de X, obtenido tras realizar estudios por imágenes del cerebro.

$$GetGenes(A', X') = A''$$

Siendo A'' un conjunto ordenado de 10 genes de A' con las puntuaciones de importancia más altas al intentar clasificar X' entre el resto de las regiones del cerebro de X. De esta forma es posible obtener un conjunto acotado de genes que permite reducir el espacio de búsqueda de genes a estudiar en la investigación de determinadas patologías psiquiátricas de las cuales conozco la importancia de las regiones del cerebro X'. En la función *GetGenes*, a diferencia de *GetROIs*, uso las tablas de *microarrays* del *Allen Human Brain Atlas* transpuestas. Esto significa que las filas representan regiones del cerebro y los nombres de las columnas son las de los *probes* de los genes disponibles en dicho atlas.

Esta función requiere dos parámetros:

- Lista de Genes A': Una lista de genes relacionados con una patología psiquiátrica, y sobre los cuales quiero determinar su nivel de importancia en un conjunto de regiones del cerebro también involucradas en la misma patología (y que perteneces a la "Lista de *ROIs*" del ítem siguiente).
- Lista de *ROIs* X': Una lista de regiones de cerebro sobre las que quiero analizar los niveles de expresión génica de los genes listados en el ítem anterior (Lista de Genes A').

El flujo de funcionamiento es el siguiente:

1. Cargo en una variable la lista de genes suministrada (Lista de Genes A'). junto a la versión transpuesta de *microarrays* del *Allen Human Brain Atlas*. Usando la tabla *ProbeIDs* busco los *IDs* de los *probes* que corresponden a los de la Lista de Genes A' y selecciono las columnas que los posean.
2. Por cada uno de los donantes del atlas ejecuto iterativamente los procedimientos 3 y 4.
3. Usando la Lista de ROIs X' suministrada, creo la variable *target* a clasificar colocando un "1" en aquellas filas donde las regiones del cerebro están presentes en la lista de regiones X' y un "0" en las restantes.
4. Utilizo la función *AUCRF*[35] de la biblioteca de lenguaje R homónima para realizar una selección de las variables más adecuadas para clasificar la

variable *target*, optimizando la selección usando el área bajo la curva *ROC*. Esta función realiza el proceso de eliminación de variables hacia atrás, quitando el 20% de las variables en cada iteración, hasta que solo quede el 0,2% de la cantidad de variables original. Dado esto, los parámetros a establecer son los siguientes:

- *pdel*: Cantidad de variables a eliminar en cada iteración = 20%
- *k0*: porcentaje de variables remanentes para parar el proceso de eliminación de variables = 20%

Finalmente creo una tabla con las 10 variables de mayor importancia de acuerdo con *AUCRF*, en orden descendente.

5. Unifico las seis tablas creadas en el punto anterior en una sola y conservo la instancia de cada *probe* con su mayor valor de importancia, creando una nueva tabla ordenada por este valor. Como resultado final, conservo los 10 primeros genes de dicha tabla ordenada por puntaje de importancia.

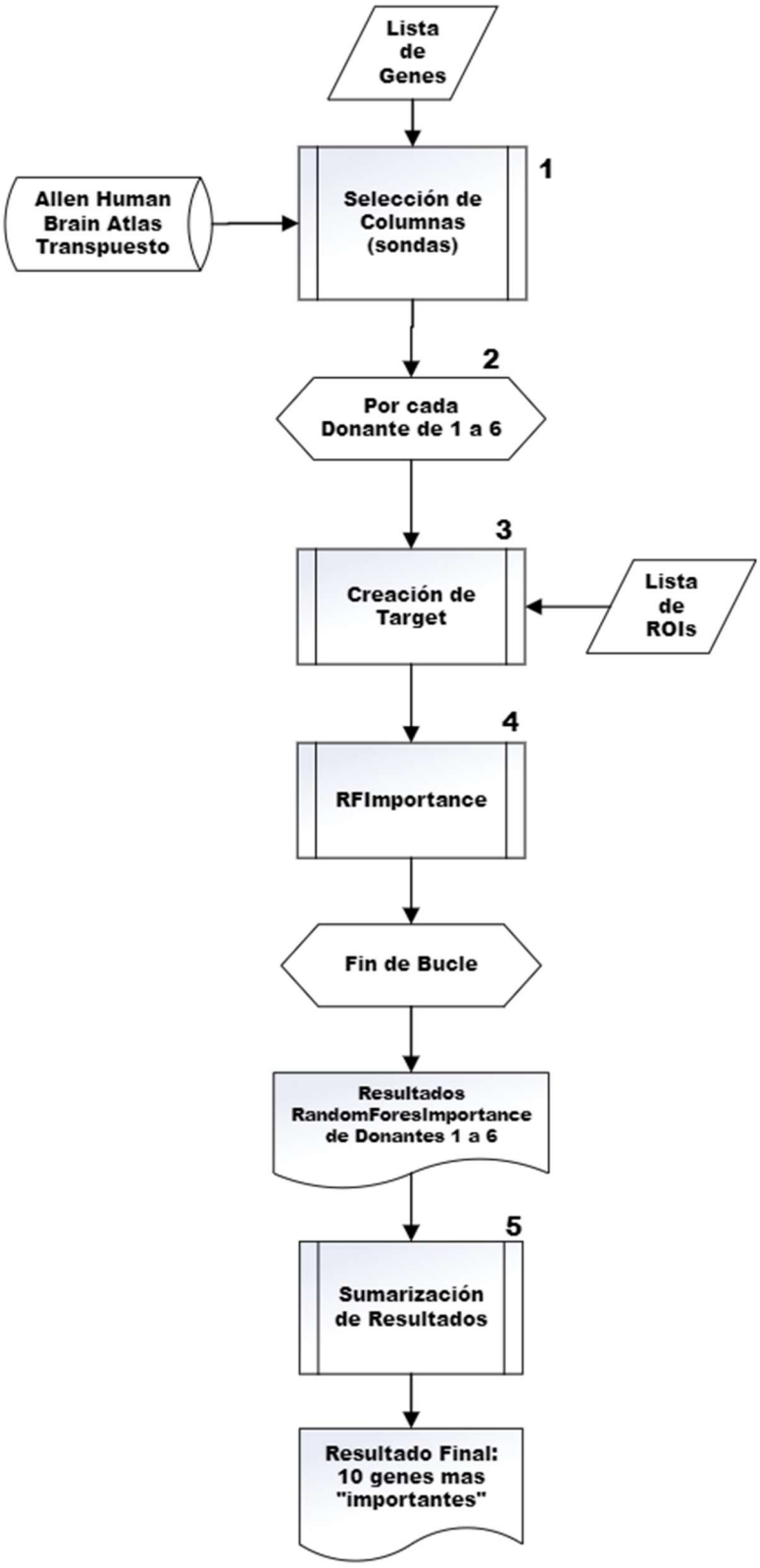

Figura 3-13. Diagrama de flujo de la función GetGenes

3.2.2. Prueba del método *GetGenes*

Utilicé la función *GetGenes* para determinar cuáles genes de la lista *Willour* son los más importantes para clasificar, de acuerdo con su expresión génica, al subículo del resto de las regiones del cerebro presentes en el atlas. Para ello le suministré a la función una lista con los dos nombres de regiones del cerebro que representan al subículo en el atlas (Tabla 3-9), y la lista de genes *Willour*.

Regiones del Cerebro
subiculum, left
subiculum, right

Tabla 3-9. Nombres de regiones del cerebro correspondientes al subículo en el Allen Human Brain Atlas

Como resultado, la función *GetGenes* devolvió la siguiente lista de 10 genes:

Nombre de Gen	Puntaje Final	Genotipificado
GRIK1	0.87	*
TSHZ2	0.72	
GRM8	0.57	*
NFIA	0.49	*
DPYS	0.34	
FARP1	0.33	*
AKAP7	0.30	*X
ARL15	0.27	
MAML3	0.27	
LDB2	0.26	

Tabla 3-10. Resultado de la prueba del método GetGenes utilizando la lista de genes Willour sobre el subículo.

Bajo la misma investigación en la cual el Dr. Salas determinó que el subículo tenía una actividad funcional mayor con el hipocampo, posteriormente genotipó un SNP de cada uno de los genes resultantes de la función *GetGenes* en los mismos 410 pacientes estudiados, como explico en la sección 5.1.

Nota: Dado que, al momento de realizar esta investigación usamos una versión inicial de *GetGenes* cuyo resultado consistía en solo 5 genes, solo la mitad de los 10 genes resultantes en la versión actual fueron genotipificados (aquellos que poseen un asterisco en la columna "Genotipificado" de la Tabla 3-10). Sin embargo, esto no afecta la validez de los resultados en este trabajo, ya que llego a la misma conclusión con ambas versiones de *GetGenes*, porque el único gen de interés encontrado está incluido en ambos resultados finales.

3.2.3. Análisis de los parámetros y métricas del método *GetGenes*

El método *GetGenes*, a diferencia de GetROIs, posee solo una subfunción que realiza selección de variables, por lo que no es necesario recurrir a dos etapas de funcionamiento. Como explico en la sección 3.2, este método trabaja con las tablas de *microarrays* transpuestas, donde las columnas son *probes*. Inicialmente, sobre estas tablas llevo a cabo un procedimiento de selección de columnas de acuerdo con las listas de genes (que surgen del estudio realizado por mi director de tesis tras el uso de la función *GetROIs* sobre la lista *Willour*, como explico en la sección 5.1) y que le suministro al método *GetGenes*. Previo a este procedimiento, las tablas de *microarrays* transpuestas poseen las dimensiones mostradas en la Tabla 2-4, y posteriormente la cantidad de filas son las mostradas en la Tabla 2-1. La cantidad de filas es la misma para todos los *microarrays* porque son seleccionadas en base a una misma lista de genes de prueba (lista *Willour*). La cantidad de columnas coincide con la cantidad de filas en las tablas de *microarrays* sin transponer (ver Tabla 2-4).

Donante	Cantidad de regiones (Filas)	Cantidad de *Probes* (Columnas)
Donante 1	312	942
Donante 2	312	893
Donante 3	312	363
Donante 4	312	529
Donante 5	312	470
Donante 6	312	501

Tabla 3-11. Dimensiones de las tablas de microarray transpuestas sobre las que se realiza la selección de genes

De acuerdo con la función *AUCRF* aplicada sobre las tablas de *microarrays* transpuestas de cada donante, muestro la cantidad de *probes* óptima utilizadas para obtener la mejor área bajo la curva *ROC* en la Tabla 3-12.

Donante	Cantidad óptima de variables
1	6
2	15
3	6
4	15
5	248
6	248

Tabla 3-12. Cantidad de variables óptimas para cada donante, según la función AUCRF

En la Figura 3-14 podemos apreciar el área bajo la curva *ROC* para cada donante, y cómo se alcanzan valores por encima de 0,9 tras utilizar menos de 10 variables, más allá de que la función haya utilizado 248 variables para obtener los valores óptimos de dicha métrica en los donantes 5 y 6.

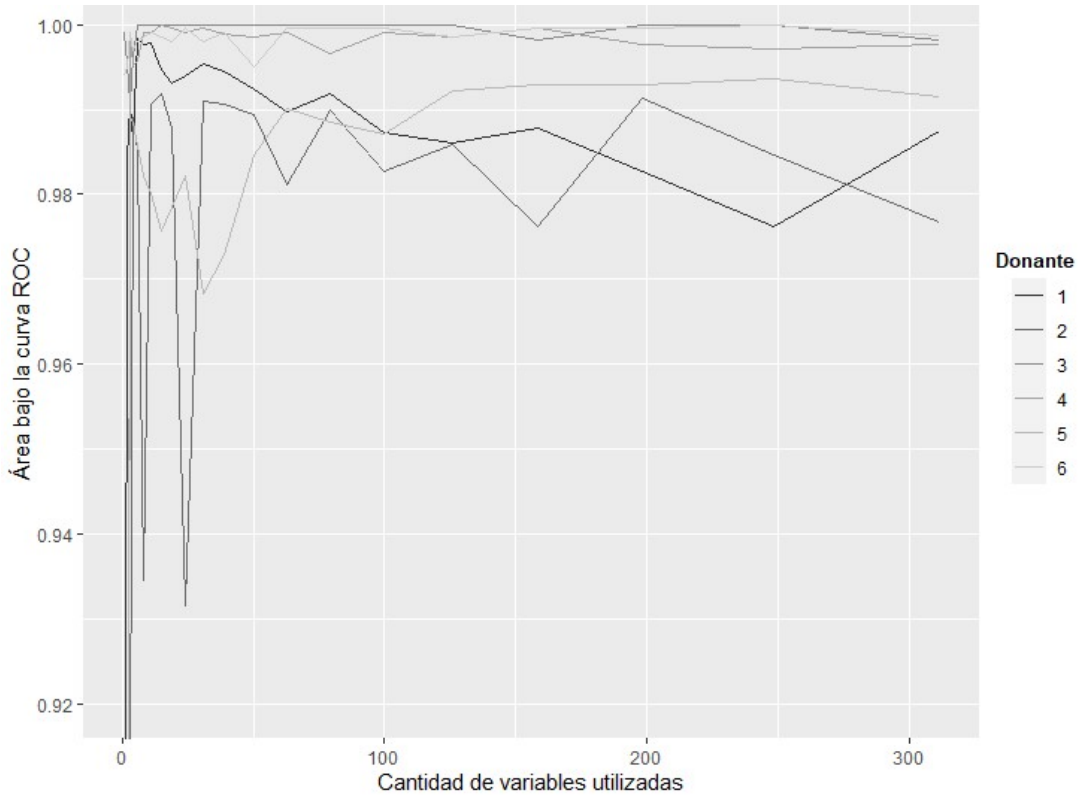

Figura 3-14. Cantidad de variables vs área bajo la curva ROC de los modelos generados por AUCRF

En la Tabla 3-13 muestro los valores de *mtry* para los modelos óptimos obtenidos para cada donante. En todos los casos usé *ntree*=500.

Donante	mtry	Exactitud media
1	2	0.81
2	3	0.77
3	2	0.82
4	3	0.75
5	15	0.66
6	15	0.66

Tabla 3-13. Valores de mtry y exactitud media de los modelos finales de AUCRF

4. Conclusiones y trabajos futuros

4.1. Conclusiones

- Los procedimientos de validación sobre el método *GetROIs* mostraron su efectividad para encontrar regiones del cerebro asociadas a un conjunto de genes. Y la prueba realizada durante una investigación sobre aspectos génicos relacionados a intentos de suicidio fue exitosa, al encontrar una región del cerebro asociada a un estudio genético independiente.
- Lo propio ocurrió con el método *GetGenes* que, aunque no contó con una fase de validación, logró un resultado positivo en la misma investigación de prueba donde utilicé la función *GetROIs*.

4.2. Trabajos futuros

- El principal objetivo a futuro es lograr la utilización de los métodos desarrollados en tantos escenarios de investigación como sea posible, especialmente para la función *GetGenes*, dado que el flujo de trabajo donde este método participa requiere el genotipificado de los genes que entrega como resultado. Esto último precisa no solo de la motivación, sino del acceso a un presupuesto aprobado para genotipificar.
- Realizaré un estudio con el objetivo de acortar los tiempos de ejecución y los recursos informáticos necesario para ejecutar los métodos *GetROIs* y *GetGenes*.

5. Anexo

5.1. Caso de uso de *GetROIs* y *GetGenes*

Las funciones *GetROIs* y *GetGenes* creadas en este trabajo fueron utilizadas durante la investigación en curso preliminarmente titulada *A novel approach to Link Genetics and Human Brain Imaging Identifies AKAP7-dependent Subicular Functional Connectivity as Altered in Suicidality* conducida por mi director de tesis Dr. Ramiro Salas en la cual oficié como autor principal, y que se encuentra pendiente de publicación. En la misma se realizó un estudio sobre 410 pacientes de *The Menninger Clinic*, 130 de los cuales presentaron intentos de suicidio (*AAT*) y 280 quienes no lo presentaron (*NAT*).

A continuación, enumero los pasos realizados por mi director de tesis y por mí, en diferentes etapas cada uno.

1) Obtuvimos un conjunto de genes posiblemente relacionados a intentos de suicidios, según el estudio GWAS *A genoma-wide association study of attempted suicide* (ver sección 1.3.3) correspondiente a los 2507 SNPs con un límite arbitrariamente elegido *de p-value*<=$1x10^{-3}$. Determiné el nombre de los genes usando la herramienta *scanDB*[36].
2) Usamos la función *GetROIs* para procesar dicha lista y obtuvimos 10 regiones del cerebro consideradas como posiblemente relacionadas con intentos de suicidio. Como explico en la sección 5.1, este estudio se llevó a cabo con una versión de la herramienta *GetROIs* que solo daba como resultado 8 regiones, de las cuales 3 coinciden con las 10 obtenidas con la última versión utilizada en este trabajo. Aun así, el subículo (una de las 3 regiones coincidentes) resultó una región de interés asociada a intento de suicidio (como explico más adelante) por lo que habríamos llegado a identificar a dicha región tanto con la región actual como con la anterior.
3) Utilizando estudios de conectividad funcional en estado de reposo sobre los 410 pacientes (*AAT vs NAT*), realizamos un estudio *ROI-to-ROI* desde cada región obtenida con el método GetROIs hacia el resto de las regiones del cerebro según el atlas *AAL* y también sobre un conjunto de pequeñas regiones para las cuales (según la bibliografía) hipotetizamos a priori que podrían ser importantes en el estudio de suicidio. Estas son:
    - *Habenula*
    - *Ventral tegmental area and substancia nigra compacta*
    - *Medial and dorsal raphe*
    - *Locus coeruleus.*

    Tras corregir por Bonferroni por la múltiple cantidad de comparaciones hechas, las conectividades que presentaron valores significativamente altos fueron entre las regiones *left subiculum* a *right* y *left habenula* en *ROI-to-ROI* y entre *left subiculum* y *middle frontal gyrus* en *seed-to-voxel*. *En ROI-to-ROI*, estudiamos y tabulamos la conectividad funcional entre todas las regiones en que el cerebro ha sido parcelado. Para este estudio, se comparan los valores de conectividad funcional entre cada una de las 10 regiones definidas como posiblemente

interesantes por *GetROIs* y todas las otras parcelas, usando Bonferroni para controlar el alto número de comparaciones (126 comparaciones, una por cada parcela). En *seed-to-voxel*, usamos cada una de las 10 regiones obtenidas por *GetROIs* como semillas (*seeds*) y buscamos *clusters* de *voxels* en los que la conectividad con la semilla sea estadísticamente distinta entre los dos grupos de pacientes (*ATT* y *NAT*).

4) Luego, utilizamos el método *GetGenes* para determinar cuáles genes extraídos del estudio GWAS presentan una conectividad funcional en el subículo que permita diferenciar dicha región (*subículum, left* y *subículum, right* según el *Allen Human Brain Atlas*) del resto de las regiones presentes en dicho atlas.
5) El siguiente paso consistió en genotipificar un *SNP* de cada uno de los genes obtenidos en el paso anterior. De los diez genes resultantes de la función *GetGenes* para este trabajo solo se estudiaron cinco ya que, como sucedió con la función *GetROIs*, al momento de realizar esta investigación utilizamos una versión inicial que solo entregaba como resultado cinco genes. Los *SNPs* se seleccionaron usando una combinación de la mayor frecuencia de menor alelo posible y el menor *p-value* en el *GWAS*.
6) Los datos significativos en *ROI-to-ROI* y *Seed-to-Voxel* (conectividad *left subiculum-habenula* y *left subiculum-middle frontal gyrus*, respectivamente) fueron estudiados otra vez, pero ahora separando las muestras de *ATT* y *NAT* en dos grupos, de acuerdo con sus genotipos en cada uno de los 5 genes genotipificados (marcados con asteriscos en la Tabla 3-10). Como un *SNP* fue genotipificado en cada gen, se hicieron 5 estudios *ROI-to-ROI (left subiculum-habenula* con cada uno de los 5 genes) y 5 estudios *Seed-to-Voxel* (usando *left subiculum* como semilla y buscando *clusters* de *voxels* en todo el cerebro). Estos estudios dieron como resultado que la variante genética (SNP) genotipificada en el gen AKAP7 (*A kinase anchoring protein 7*, un gen importante en actividad neuronal en corazón y cerebro) interactúa con la conectividad entre *left subículum* y *middle frontal gyrus* cuando los pacientes se dividen en *ATT* y *NAT*. Por lo tanto, es posible que la actividad de AKAP7 esté asociada al riesgo de intento de suicidio a través de la modulación de la conectividad entre esas dos regiones del cerebro. No se encontraron interacciones en la conectividad entre *left subiculum* y *habenula* en *ATT* vs *NAT*.

5.2. Tablas de datos

5.2.1. Datos y resultados de los modelos finales en *GetROIs*

En la Tabla 5-1 expongo los datos con los cuales fueron construidas las Figura 3-5 a laFigura 3-12.

Lista	Don.	#Filas	#Col	mtry	RFAcc	cost	loss	weight	SVMAcc	kmax	KNNAcc
Cbc	1	656	461	231	0.94	0.25	2	2	0.94	7	0.94
Cbc	2	656	535	32	0.93	0.5	2	1	0.92	9	0.93
Cbc	3	656	363	182	0.93	0.25	2	1	0.90	9	0.93
Cbc	4	656	369	2	0.93	0.25	2	1	0.93	9	0.92
Cbc	5	656	470	2	0.93	0.25	2	1	0.94	7	0.93
Cbc	6	656	285	143	0.95	0.25	2	2	0.91	9	0.95
Str	1	1870	54	54	0.86	0.25	1	1	0.80	9	0.83
Str	2	1870	261	131	0.85	0.25	1	1	0.80	9	0.82
Str	3	1870	206	104	0.85	0.25	1	1	0.84	9	0.83
Str	4	1870	190	96	0.86	0.25	1	1	0.82	9	0.84
Str	5	1870	101	101	0.86	0.5	1	1	0.84	9	0.85
Str	6	1870	79	40	0.85	0.25	1	1	0.83	9	0.84
Md	1	628	811	40	0.92	1	2	3	0.91	9	0.92
Md	2	628	192	2	0.92	1	2	3	0.91	9	0.91
Md	3	628	82	2	0.92	0.25	1	1	0.90	9	0.91
Md	4	628	65	2	0.92	0.25	2	3	0.92	5	0.91
Md	5	628	241	121	0.92	0.25	2	2	0.90	9	0.87
Md	6	628	408	205	0.91	0.5	2	3	0.90	9	0.90
willour	1	622	54	2	0.66	0.25	1	3	0.63	9	0.63
willour	2	622	656	2	0.62	1	2	3	0.67	9	0.63
willour	3	622	311	156	0.64	0.25	1	3	0.62	7	0.63
willour	4	622	333	2	0.66	0.25	1	2	0.62	9	0.57
willour	5	622	281	141	0.66	0.25	2	2	0.66	9	0.67
willour	6	622	39	20	0.68	0.25	2	1	0.65	7	0.63

Tabla 5-1. Descripción de datos y parámetros usados junto a la exactitud lograda en los modelos finales

5.2.2. Datos y resultados de la función *AUCRF* de la subfunción *GetGenes*

#Don	#Col	AUC
1	1	0.76
1	2	0.98
1	3	0.99
1	4	0.99
1	6	1.00
1	8	1.00
1	11	1.00
1	15	0.99
1	19	0.99
1	24	0.99
1	31	1.00
1	39	0.99
1	50	0.99
1	63	0.99
1	79	0.99
1	100	0.99
1	126	0.99
1	158	0.99
1	198	0.98
1	248	0.98
1	311	0.99
2	1	0.64
2	2	0.74
2	3	0.87
2	4	0.99
2	6	0.99
2	8	0.93
2	11	0.99
2	15	0.99
2	19	0.99
2	24	0.93
2	31	0.99
2	39	0.99
2	50	0.99
2	63	0.98
2	79	0.99
2	100	0.98
2	126	0.99
2	158	0.98
2	198	0.99
2	248	0.98
2	311	0.98
3	1	1.00
3	2	1.00
3	3	0.99
3	4	1.00
3	6	1.00
3	8	1.00
3	11	1.00
3	15	1.00
3	19	1.00
3	24	1.00
3	31	1.00
3	39	1.00
3	50	1.00
3	63	1.00
3	79	1.00
3	100	1.00
3	126	1.00
3	158	1.00
3	198	1.00
3	248	1.00
3	311	1.00
4	1	0.61
4	2	0.86
4	3	1.00
4	4	1.00
4	6	1.00
4	8	1.00
4	11	1.00
4	15	1.00
4	19	1.00
4	24	1.00
4	31	1.00
4	39	1.00
4	50	1.00
4	63	1.00
4	79	1.00
4	100	1.00
4	126	1.00
4	158	1.00
4	198	1.00
4	248	1.00
4	311	1.00
5	1	0.49
5	2	0.64
5	3	0.99
5	4	0.99
5	6	0.99
5	8	0.98
5	11	0.98
5	15	0.98
5	19	0.98
5	24	0.98
5	31	0.97
5	39	0.97
5	50	0.98
5	63	0.99
5	79	0.99
5	100	0.99
5	126	0.99
5	158	0.99
5	198	0.99
5	248	0.99
5	311	0.99
6	1	0.99
6	2	0.99
6	3	1.00
6	4	1.00
6	6	1.00
6	8	1.00
6	11	1.00
6	15	1.00
6	19	1.00
6	24	1.00
6	31	1.00
6	39	1.00
6	50	0.99
6	63	1.00
6	79	1.00
6	100	1.00
6	126	1.00
6	158	1.00
6	198	1.00
6	248	1.00
6	311	1.00

Tabla 5-2. Descripción de los datos utilizados y los valores de AUC logrados por la función AUCRF

6. Bibliografía

[1] National Institute of Mental Health, “Suicide Statistics,” 2021. https://www.nimh.nih.gov/health/statistics/suicide.shtml.

[2] Allen Human for Brain Science, “ALLEN Human Brain Atlas. Technical white paper: Microarrat Survey,” 2013. http://human.brain-map.org/.

[3] V. L. Willour *et al.*, “A genome-wide association study of attempted suicide,” *Molecular Psychiatry*, vol. 17, no. 4, pp. 433–444, 2012, doi: 10.1038/mp.2011.4.

[4] H. J. Kang *et al.*, “Spatiotemporal transcriptome of the human brain,” *Nature*, vol. 478, no. 7370, pp. 483–489, 2013, doi: 10.1038/nature10523.Spatiotemporal.

[5] Public Engagement team and Wellcome Genome Campus, “Your Genome. What is DNA?” www.yourgenome.org/facts/what-is-dna.

[6] “Scitable,” 2019. scitable.com.

[7] EMBL-EBI Training, “What are genome wide association studies (GWAS)?” https://www.ebi.ac.uk/training/online/courses/gwas-catalogue-exploring-snp-trait-associations/what-is-gwas-catalog/what-are-genome-wide-association-studies-gwas/.

[8] M. A. Lindquist, “The Statistical Analysis of fMRI Data,” vol. 23, no. 4, pp. 439–464, 2009, doi: 10.1214/09-STS282.

[9] F. Cope and R. Damadian, “Cell potassium by 39K spin echo nuclear magnetic resonance,” *Nature*, vol. 228, no. 5266, pp. 76–77, 1970.

[10] R. v. Damadian, “Tumor Detection by Nuclear Magnetic Resonance,” *Science*, vol. 171, no. 3976, pp. 1151–1153, 1971.

[11] B. Biswal, F. Z. Yetkin, V. M. Haughton, and J. S. Hyde, “Functional connectivity in the motor cortex of resting human brain using echo-planar MRI.,” *Magnetic Resonance in Medicine*, 2012, [Online]. Available: http://f1000.com/714597885.

[12] M. E. Raichle, A. M. MacLeod, A. Z. Snyder, W. J. Powers, D. A. Gusnard, and G. L. Shulman, “A default mode of brain function,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 676–682, Jan. 2001, doi: 10.1073/pnas.98.2.676.

[13] M. P. van den Heuvel and H. E. Hulshoff Pol, “Exploring the brain network: A review on resting-state fMRI functional connectivity,” *European Neuropsychopharmacology*, vol. 20, no. 8, pp. 519–534, Aug. 2010, doi: 10.1016/j.euroneuro.2010.03.008.

[14] S. Whitfield-Gabriel and A. Nieto-Castanon, “Conn: A Functional Connectivity Toolbox for Correlated and Anticorrelated Brain Networks,” *Brain Connectivity*, vol. 2, no. 3, 2012.

[15] I. Guyon and A. M. De, “An Introduction to Variable and Feature Selection André Elisseeff,” 2003. Accessed: Jul. 02, 2019. [Online]. Available: http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf.

[16] A. Jović, K. Brkić, and N. Bogunović, “A review of feature selection methods with applications,” *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*, pp. 1200–1205, 2015, doi: 10.1109/MIPRO.2015.7160458.

[17] B. H. Menze *et al.*, “A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data,” *BMC Bioinformatics*, vol. 10, pp. 1–16, 2009, doi: 10.1186/1471-2105-10-213.

[18] L. Breiman, “Random Forests,” pp. 1–33, 2001, [Online]. Available: http://oz.berkeley.edu/~breiman/randomforest2001.pdf%0Ahttp://medcontent.metapress.com/index/A65RM03P4874243N.pdf.

[19] J. Freidman, R. Tibsherani, and T. Hastie, “Springer Series in Statistics The Elements of Statistical Learning,” *The Mathematical Intelligencer*, 2009, doi: 10.1007/b94608.

[20] P. S. Mann, “Nonparametric Methods,” in *Introductory Statistics, 9th Edition*, 2002, pp. 631–696.

[21] C. E. Bonferroni, “Teoria statistica delle classi e calcolo delle probabilità,” *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 1936.

[22] A. Yadav, “Support Vector Machines (SVM),” 2018. https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589.

[23] A. Max *et al.*, “Caret R Package,” 2020.

[24] A. Mishra, “Decoding Support Vector Machines,” *Towards data science*, 2020. https://towardsdatascience.com/decoding-support-vector-machines-5b81d2f7b76f.

[25] S. Luo, “Loss Function(Part III): Support Vector Machine,” *Towards data science*, 2018. https://towardsdatascience.com/optimization-loss-function-under-the-hood-part-iii-5dff33fa015d#:~:text=The loss function of SVM,the raw model output%2C $\theta^T x$.

[26] L. Y. Hu, M. W. Huang, S. W. Ke, and C. F. Tsai, “The distance function effect on k-nearest neighbor classification for medical datasets,” *SpringerPlus*, vol. 5, no. 1, 2016, doi: 10.1186/s40064-016-2941-7.

[27] N. Sharma, “Importance of Distance Metrics in Machine Learning Modelling,” *Towards data science*, 2019. .

[28] S. Li, E. J. Harner, and D. A. Adjeroh, “Random KNN feature selection - a fast and stable alternative to Random Forests,” *BMC Bioinformatics*, vol. 12, no. 1, p. 450, Dec. 2011, doi: 10.1186/1471-2105-12-450.

[29] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.

[30] contributors worldwide and R Team, “wilcox.test,” *Stats v3.6.2*. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/wilcox.test.

[31] A. Liaw, “randomForest,” *Classification and regression based on a forest of trees using random inputs, based on Breiman (2001)*, 2018. https://www.rdocumentation.org/packages/randomForest.

[32] N. Becker, W. Werft, and A. Benner, “penalizedSVM R Package,” 2018.

[33] “varImp function.” https://www.rdocumentation.org/packages/caret/versions/6.0-84/topics/varImp (accessed Jul. 07, 2019).

[34] S. Li, “rKNN R Package,” 2015.

[35] A. V. Urrea, M. L. Calle, and M. V. Urrea, “Package ‘AUCRF,’” 2015.

[36] “SCANDB: SNP and CNV Annotation Database.” http://www.scandb.org/newinterface/about.html (accessed Jul. 05, 2019).