Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Departamento de Ciencias de la Computación

Tesina para optar por el título de
Licenciado en Ciencias de la Computación

"Selección de variables en problemas anchos con alta correlación"

Autor: Mauro Di Masso

Director: Dr. Pablo Granitto

Marzo 2014

Índice

1. Introducción

El aprendizaje automatizado o *machine learning* es una ciencia que nace a mediados del siglo XX y que ha estado en auge desde entonces. Entre sus objetivos se plantea la generación automática de modelos numéricos que, basados en datos de entrenamiento, permitan la correcta predicción sobre observaciones nuevas. Las principales dificultades de dicho modelado subyacen en la cantidad de variables medidas en los datos presentes, así como también en su calidad en términos de ruido, y en la cantidad de mediciones disponibles para llevar a cabo el entrenamiento.

Los avances en las tecnologías de medición y almacenamiento han facilitado ampliamente la obtención de cantidades masivas de datos, pero con ello también han surgido inconvenientes para obtener información útil a la hora de analizarlos. Cuando se comienza a trabajar en espacios de mayor dimensionalidad, se necesitan exponencialmente más mediciones para obtener la información necesaria a fin de crear un modelo de predicción. Este problema es conocido como "maldición de la dimensionalidad" o *curse of dimensionality* y afecta a los principales problemas del área hoy en día, tales como los microarrays de ADN y las espectrometrías de suero sanguíneo. Asimismo, con el correr de las décadas, el volúmen de variables que pueden ser medidas ha crecido a tal punto que su procesamiento es incluso costoso para las computadoras más potentes.

Estos problemas de grandes cantidades de variables suelen venir acompañados de una escasa colección de observaciones por lo que obtienen la denominación de "problemas anchos". Al desfazaje entre observaciones y variables por observación se le suma la existencia de una mayor probabilidad de variables correlacionadas y ruidosas dentro del conjunto de datos. Esto hace que la correcta construcción del modelo de datos por parte de los métodos

de aprendizaje automatizado sea costoso en términos temporales. Además, por lo general, la buena calidad de las predicciones queda en contraposición con la facilidad para interpretar la información obtenida.

La selección de variables o *feature selection* es una respuesta a los problemas planteados ya que su meta es simplificar el espacio de variables quitando de la ecuación aquéllas irrelevantes a la solución, redundantes y/o con altos niveles de ruido, etc. Para ello se utilizan métodos que generan un ranking de variables caracterizándolas por su importancia o poder predictivo o de clasificación. Este preprocesamiento sobre los datos contribuye a incrementar la eficiencia de los algoritmos de aprendizaje permitiéndoles centrar sus esfuerzos en un grupo reducido de variables.

Hoy en día existen múltiples algoritmos de selección de variables, entre los cuales el de Eliminación Recursiva de Variables (*Recursive Feature Elimination*) o RFE es uno de los más utilizados. RFE permite la obtención de modelos muy precisos ya que su foco está puesto en minimizar el error de clasificación en cada una de sus iteraciones. Este tipo de foco, paradójicamente, reduce la utilidad de la solución a ojos humanos, ya que si bien predice de forma superior, los conjuntos finales de variables elegidas suelen diferir de un experimento a otro debido a que este método no se preocupa por la llamada estabilidad de la solución. Como RFE minimiza el error en cada iteración, tal vez prefiera elegir una variable en lugar de otra cuando en realidad las mismas son redundantes y tan sólo quedaron ordenadas de forma diferente por una mera cuestión de azar durante el entrenamiento. La aparición de distintas variables "más importantes" en diferentes experimentos sobre un mismo conjunto de datos dificultan la interpretación de los resultados y llevan a la problemática que se aborda en esta tesina.

Como objetivo de la presente se propone entonces el diseño e implementación de un método de selección de variables, cuyo foco sea la obtención de conjuntos de variables más estables que permita a su vez una mayor interpretabilidad sobre la importancia y correlación de las variables obtenidas; aunque no alcance en algunos casos el mismo nivel de error que otros métodos más codiciosos como RFE.

Para un mejor entendimiento del desarrollo experimental se cubren en el capítulo dos los conceptos básicos relacionados con el aprendizaje automatizado, la selección de variables y la estabilidad. En el capítulo tres, se presentan, explican y exploran los métodos de selección de variables utilizados durante la experimentación. En el capítulo cuatro, se muestran los resultados de clasificar diferentes conjuntos de datos, artificiales y reales, con cada método; para luego, en el capítulo cinco, determinar las ventajas y las desventajas de cada uno, así como otras conclusiones generales. Por último, se plantean trabajos futuros para indagar en las potencialidades del método generado.

2. Conceptos básicos

2.1. Aprendizaje automatizado

El aprendizaje automatizado es una rama de la inteligencia artificial cuyo objetivo es el desarrollo de algoritmos que permitan a sistemas informáticos aprender conceptos o criterios de decisión que no son fácilmente programables de otra manera [1]. Esta dificultad se debe principalmente a una carencia en el desarrollo teórico del área de estudio en cuestión que no permite determinar las cualidades subyacentes del problema.

Formalmente, un algoritmo de aprendizaje automatizado genera una función de la forma

$$h : X \to Y$$

también llamada hipótesis, que transforma elementos del dominio de datos X al conjunto de datos Y de posibles valores objetivo definidos en base al problema en cuestión. El algoritmo de aprendizaje aprende de un subconjunto de datos D, llamado conjunto de entrenamiento, perteneciente al espacio X o $X \times Y$.

- $D = \{x_1, x_2, \ldots, x_m\} \in X$
- $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \in X \times Y$

En el primer caso, se dice que el problema es de aprendizaje no supervisado, donde h sólo puede basarse en los elementos de entrada x_i y debe descifrar las similitudes entre los datos presentes para dar luego una posible clasificación. En el segundo caso, se habla de problemas de aprendizaje supervisado. Aquí, h utiliza pares (x_i, y_i) donde y_i es el valor deseado para la entrada x_i. Si los valores de y_i son continuos, el problema se llama de regresión y el algoritmo busca ajustar una curva que satisfaga los valores de

entrenamiento; si los valores de y_i son discretos entonces se está en presencia de un problema de clasificación, en el que h subdivide el espacio de soluciones para determinar cómo asignar los posibles valores o etiquetas. Estos últimos problemas son los que trabaja esta tesina.

Dado un problema de clasificación concreto, existen infinitas soluciones consistentes con las observaciones dadas. Como no es posible conocer todos los valores a clasificar (no tendría sentido), todos los algoritmos de aprendizaje trabajan sobre el supuesto de que cualquier hipótesis h que pueda aproximar lo suficientemente bien la función ideal de clasificación c para un conjunto de datos lo suficientemente grande D con puntos clasificados x_i, también la aproximará lo suficientemente bien para los puntos no clasificados x_o. No es difícil concluir que si existen infinitas hipótesis, todo punto no clasificado tiene la misma probabilidad de pertenecer a cualquiera de las clases disponibles del problema. Hace falta que el algoritmo ignore algunas de estas hipótesis para hacer posible el aprendizaje, lo que implica que el algoritmo debe suponer algo arbitrario sobre la solución. Estas suposiciones se conocen como sesgo inductivo o *inductive bias*.

Para determinar la utilidad de una hipótesis se debe realizar algún tipo de validación sobre ella. Inicialmente se espera que clasifique bien los datos de entrenamiento, pero esto no garantiza precisión alguna sobre los datos no observados. Existen diversas técnicas de validación de modelos de predicción; entre ellas, la más común e intuitiva es particionar los datos de entrenamiento (muestreo o *subsample*) en dos conjuntos, uno para el entrenamiento y otro para la validación. De esta manera, se sacrifica parcialmente el aporte al aprendizaje para ganar certeza en la predicción. Tanto el aprendizaje como la validación son mejores si este procedimiento se repite varias veces con subconjuntos de validación disjuntos. La técnica de validación cruzada de k

iteraciones o *k-fold cross validation* subdivide el conjunto de entrenamiento en k subconjuntos, de los cuales utiliza k-1 para entrenar y uno para validar; así k veces hasta que se hubo utilizado cada uno para validar, y luego promedia los errores. De esta manera, todos los datos colaboran en el aprendizaje y la validación. El error de clasificación de una hipótesis h se define formalmente como

$$E(h) = \frac{N_e}{N_t}$$

donde N_t es la cardinalidad del conjunto de entrenamiento y donde

$$N_e = \sum_{i=1}^{N_t} neq(h(x_i), c(x_i)) \quad x_i \in D$$

siendo D el conjunto de datos sin particionar, c la función clasificadora ideal y

$$neq(x, y) = \begin{cases} 0 & \text{si } x = y \\ 1 & \text{si } x \neq y \end{cases}$$

Por último, existe una serie de consideraciones a tener en cuenta a la hora de entrenar un algoritmo de aprendizaje automatizado. Para comenzar, cabe recalcar que no existe un algoritmo óptimo para todo tipo de problema, por lo cual es necesario estudiar y conocer las singularidades del dominio con el que se trabaja. Cuando se emplean pocos datos de entrenamiento, o cuando la dimensionalidad del problema es tal que la cantidad de variables supera las observaciones, se corre el riesgo de que el algoritmo aprenda de los datos a tal punto que los errores de medición y demás ruidos se asimilen al modelo. Esto produce modelos que se ajustan muy bien a los datos de entrenamiento pero que tienen errores muy altos al momento de generalizar. Este fenómeno se denomina sobreajuste u *overfitting* y muestra que es preferible una hipótesis más simple que no se adapte tan bien a los datos de entrenamiento pero que generalice mejor para datos no observados todavía.

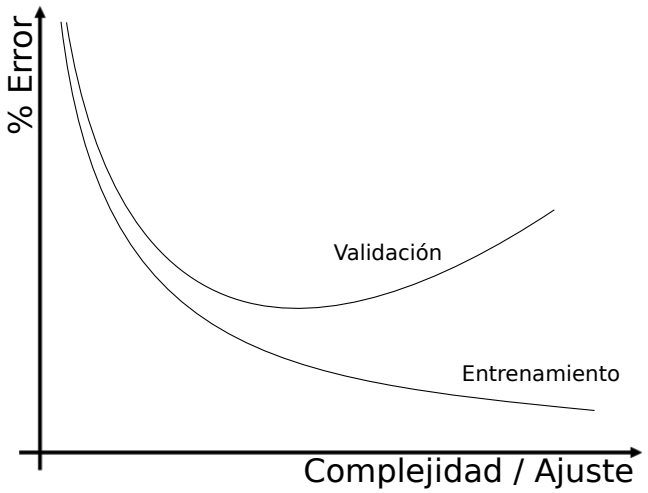


Figura 1: la peor predicción al aumentar la complejidad del modelo muestra el fenómeno de sobreajuste.

2.2. Support Vector Machines

Las máquinas de vectores soporte, o SVM por sus siglas en inglés, son el método de aprendizaje automatizado más importante de los últimos años. Fueron desarrolladas en 1992 y posteriormente mejoradas en numerosas ocasiones [2]. En su nivel más básico, son excelentes para problemas de clasificación binarios en un espacio de datos linealmente separable, pero pueden adaptarse para problemas no lineales, de multiclase y de regresión [3].

El principio que gobierna las SVM es la construcción de un hiperplano que maximice la distancia entre él y los puntos de entrenamiento de cada clase. Éste se denomina "hiperplano óptimo" y los vectores posición correspondientes a los puntos de entrenamiento más cercanos son denominados "vectores soporte". Cuanto mayor es la distancia o "margen" entre los puntos de entrenamiento y el hiperplano, mayor es el poder de generalización de la máquina. Para problemas de separación lineal, el módulo de cada componente del vector normal que define el hiperplano está en directa relación con la relevancia de la variable en el momento de la clasificación. Esto resulta

de gran importancia en el uso de algoritmos de selección de variables que se explican en la próxima sección.

Formalmente, el hiperplano óptimo y la función objetivo de la SVM se define como

$$\max_{w,b} \quad \min\{||x - x_i|| : w^T x + b = 0, \quad i = 1, \ldots, m\}$$

Las variables w y b pueden ser escaladas de tal forma que el punto más cercano al hiperplano $w^T x + b = 0$ cumpla con $w^T x + b = \pm 1$. De esta manera, para cada x_i se tiene que $y_i[w^T x + b] \geq 1$, por lo que el ancho del margen es $2/||w||$. Así, el problema de encontrar el hiperplano óptimo puede ser replanteado como el problema de optimización de la función objetivo $\tau(w)$ tal que

$$\min_{w,b} \quad \tau(w) = \frac{1}{2}||w||^2$$

con las restricciones

$$y_i[w^T x + b] \geq 1 \quad i = 1, \ldots, m$$

Para resolverlo se construye el lagrangiano

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i (y_i[x_i^T w + b] - 1)$$

donde α_i son multiplicadores de Lagrange, y su minimización lleva a

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \qquad w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

De acuerdo con las condiciones de Karush-Kuhn-Tucker [11] se concluye que

$$\alpha_i(y_i[x_i^T w + b] - 1) = 0 \quad i = 1, \ldots, m$$

Por lo tanto, los valores no nulos de α_i se corresponden con $y_i[x_i^T w+b] = 1$, lo que significa que los vectores que están en el margen cumplen un rol crucial en la solución del problema de optimización. Estos vectores son los vectores soporte del problema.

El problema puede trabajarse todavía más, llevándolo a su formal dual, que es de la forma

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

con restricciones

$$\alpha_i \geq 0 \quad i = 1, \ldots, m \quad \text{y} \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

Usando la solución a este problema, la función objetivo se puede escribir como

$$f(x) = sgn(\sum_{i=1}^{m} \alpha_i y_i x^T x_i + b)$$

Para problemas no linealmente separables en la dimensión original de los datos, el truco (literalmente, se lo llama *kernel trick*) es la utilización de una función *kernel* que construya una biyección de los puntos de entrenamiento hacia una dimensión superior donde sí sean separables. Esto se hace reemplazando el producto interno $x^T x'$ por la función kernel $k(x, x') = \Phi(x)^T \Phi(x')$, por lo que la función objetivo queda

$$f(x) = sgn(\sum_{i=1}^{m} \alpha_i y_i k(x, x_i) + b)$$

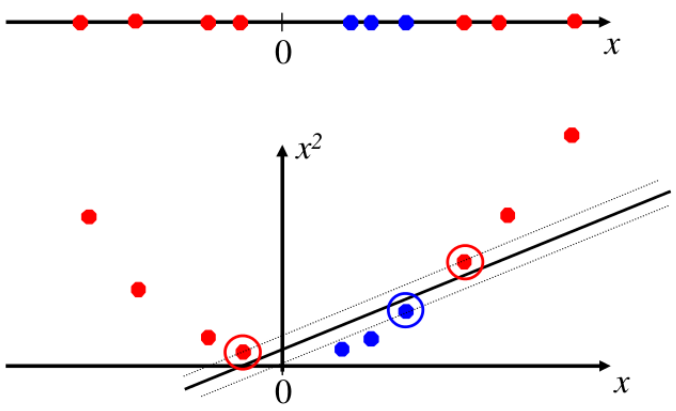


Figura 2: el panel superior muestra un problema donde los datos no son linealmente separables. En el panel inferior se utiliza un *kernel trick* para agregar una nueva dimensión al problema que efectivamente permite la separación lineal. Los puntos resaltados con halos definen los vectores soporte.

Ocurre a veces que las clases están muy superpuestas y no es posible construir un hiperplano que las separe aun usando funciones kernel, ya sea por errores de medición, datos equívocos o por la naturaleza misma de los datos de entrenamiento. En esta situación se relaja el problema de optimización utilizando variables *slack* y entra en juego un parámetro de las SVM llamado C, que determina el balance entre la precisión de entrenamiento de la máquina y el ancho del margen

$$\min_{w,b,\xi} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i$$

con restricciones

$$y_i[w^T x_i + b] \geq 1 - \xi_i \quad i = 1, \ldots, m$$

Y el problema dual queda definido como

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

con restricciones

$$0 \leq \alpha_i \leq C \quad i = 1, \ldots, m \quad \text{y} \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

Esto se traduce en que se le permite a la máquina desplazar aquellos puntos superpuestos hacia el subespacio que les pertenecería según el clasificador, desentrelazar los datos y poder trazar la división. El módulo de C determina la rigidez de dichos desplazamientos y lo que se llama margen blando o *soft margin*.

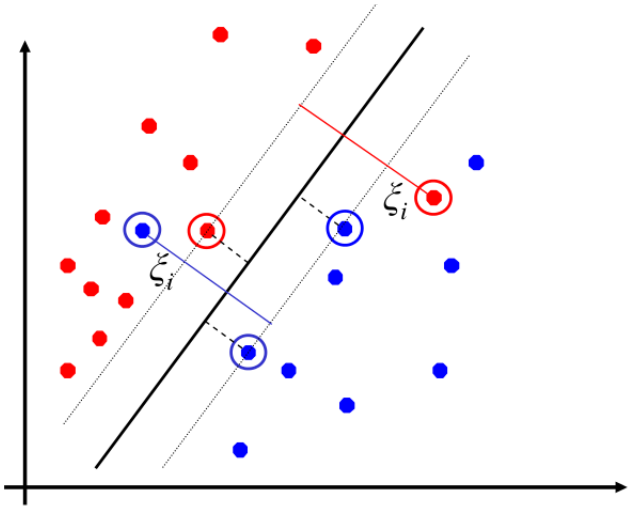


Figura 3: una SVM clasificando con clases parcialmente superpuestas.

Para problemas multiclase existen variadas técnicas, pero la idea es siempre entrenar varios clasificadores binarios y quedarse con la clasificación más correcta o más votada. Las más comunes son "uno contra todos" o *one vs all*

y "uno contra uno" o *one vs one*. En *one vs all*, se entrenan c clasificadores, uno para cada clase, considerando el resto de las clases como una sola; mientras que en *one vs one*, se entrena un clasificador para cada par de clases, para un total de $c(c-1)/2$ clasificadores. La clase se termina definiendo por votación de los clasificadores [6].

2.3. Selección de variables

La selección de variables o *feature selection* es el proceso por el cual se filtran las variables más influyentes de un problema de aprendizaje automatizado para poder construir mejores modelos [4]. Es una técnica de preprocesado de los datos que aporta grandes beneficios. Los algoritmos de selección de variables permiten reducir el número de dimensiones del espacio del problema mejorando la relación entre variables y observaciones, permitiendo un modelado de mayor calidad predictiva e interpretabilidad humana, ya que elimina variables que no aportan nada nuevo a las ya seleccionadas (variables redundantes) o que no aportan información en general al contexto del problema (variables irrelevantes).

Existen dos tipos de algoritmos de selección de variables: los de tipo filtro o *filter* y los de tipo envoltorio o *wrapper*, que si bien llevan a cabo diferentes estrategias, ofrecen como resultado un ranking de variables según su capacidad predictiva [7]. Los algoritmos de tipo *filter* son sencillos y muy eficientes computacionalmente. Se basan en mediciones estadísticas aplicadas directamente sobre los datos de entrenamiento. Como contrapartida, no tienen la posibilidad de detectar las situaciones en donde múltiples variables interactúan entre sí para describir el concepto objetivo. Los métodos *wrapper*, en cambio, trabajan sobre un algoritmo de aprendizaje automatizado, y realizan varias iteraciones para encontrar el subconjunto de variables que aumentan el

rendimiento de ese algoritmo en particular. Suelen tener un desempeño mejor que los filtros en cuanto a precisión, pero son computacionalmente costosos ya que construyen una gran cantidad de modelos de datos en cada iteración.

Los algoritmos de selección de variables parten de la base de que en todo problema de selección de variables, en especial en los de tipo ancho, se pueden encontrar tres tipos de variables:

- fuertemente relevantes
- débilmente relevantes
- irrelevantes

Sea F el conjunto de todas las variables, F_i una variable en particular, C una clase y $S_i = F - \{F_i\}$, las clasificaciones anteriores se definen como:

Relevancia fuerte: *una variable F_i es fuertemente relevante si y sólo si*

$$\mathbf{P}(C \mid F_i, S_i) \neq \mathbf{P}(C \mid S_i)$$

Relevancia débil: *una variable F_i es débilmente relevante si y sólo si*

$$\mathbf{P}(C \mid F_i, S_i) = \mathbf{P}(C \mid S_i) \quad \wedge \quad \exists S'_i \subset S_i : \mathbf{P}(C \mid F_i, S'_i) \neq \mathbf{P}(C \mid S'_i)$$

Irrelevancia: *una variable F_i es irrelevante si y sólo si*

$$\forall S'_i \subseteq S_i : \mathbf{P}(C \mid F_i, S'_i) = \mathbf{P}(C \mid S'_i)$$

Por ejemplo, sea $\{F_1, \ldots, F_5\} \in Bool$, con $F_2 = \bar{F}_3, F_4 = \bar{F}_5$ y el concepto a aprender $c(F_1, F_2)$, se puede determinar que F_1 es fuertemente relevante para la solución, F_2 y F_3 son débilmente relevantes (la solución no cambia mientras alguna esté presente) y F_4 y F_5 son completamente irrelevantes.

Para obtener la solución óptima a un problema de este tipo, el conjunto de variables seleccionadas debe incluir a todas aquellas fuertemente relevantes, ninguna de las irrelevantes y un subconjunto de variables débilmente relevantes. Sin embargo, no se dispone de un método para decidir qué subconjunto debe utilizarse, por lo que debe definirse algún tipo de redundancia entre las variables relevantes. Normalmente, la redundancia entre variables se mide en torno a la correlación entre ellas. Cuando dos variables están completamente relacionadas, como F_2 y F_3, la eliminación de una no afecta la precisión y mejora la solución. Sin embargo, en la práctica no siempre se dan casos tan sencillos, y puede que una variable esté parcialmente correlacionada con otra, o con un conjunto de otras variables.

Cobertura de Markov: *dada una variable F_i, y sea $M_i \subset F, F_i \notin M_i$, M_i es una cobertura de Markov de F_i si y sólo si*

$$\mathbf{P}(F - M_i - F_i, C \mid F_i, M_i) = \mathbf{P}(F - M_i - F_i, C \mid M_i)$$

La cobertura de Markov no sólo pide que M_i cubra toda la información que proporciona F_i sobre C, sino que además cubra la información sobre todas las otras variables contenidas en F. Si se procede desde el conjunto total de variables, si se encuentra una variable F_i tal que tenga una cobertura de Markov M_i en las variables restantes, es demostrable que la eliminación de F_i es segura y siempre se encontrará una cobertura en el subconjunto restante sin necesidad de volver hacia atrás a revisar eliminaciones previas. También

es demostrable que una variable fuertemente relevante no posee coberturas de Markov. Las variables irrelevantes quedan excluidas de la definición de redundancia. Es fácil ver que una variable redundante que es eliminada seguirá siendo redundante sin importar la sucesión de eliminaciones.

Variable redundante: *sea F' el conjunto actual de variables, una variable se considera redundante y debe ser eliminada de F' si y sólo si es débilmente relevante y tiene una cobertura de Markov dentro de F'.*

Las definiciones de relevancia dividen entonces a las variables en fuertemente relevantes, débilmente relevantes e irrelevantes. A su vez, la definición de redundancia divide las variables débilmente relevantes en redundantes y no redundantes. El objetivo es entonces encontrar el subconjunto de variables tal que contenga variables fuertemente relevantes o no redundantes.

2.3.1. Estabilidad

Un problema particular de los algoritmos de selección de variables es el problema de la estabilidad. Cuando un problema tiene un número elevado de variables y estas variables presentan altos niveles de correlación, puede que la aleatoriedad del muestreo necesario para la validación del modelo lleve a resultados igual de buenos pero con variables totalmente diferentes. La estabilidad de un algoritmo marca la robustez del mismo frente a cambios en los datos de entrenamientos, lo que se condice con la confiabilidad de la información extraída por su parte.

3. Métodos de selección de variables independientes

El problema de la correlación entre variables ya ha sido tratado por otros autores y múltiples algoritmos han surgido de dichos análisis. De ellos, dos métodos resultan particularmente interesantes de analizar por sus aportes conceptuales al problema. Estos son:

- Fast Correlation Based Filter de Lei Yu y Huan Liu [12].
- SVM-RFE with MRMR filter de Piyushkumar A. Mundra y Jagath C. Rajapakse [8].

Además, para el desarrollo de esta tesina se utilizó como referencia y control el método RFE con SVM como clasificador [5].

3.1. Recursive Feature Elimination

El algoritmo de Eliminación Recursiva de Variables, o RFE por sus siglas en inglés, es uno de los métodos de selección de variables más utilizados, tanto por su desempeño a nivel temporal como por la calidad de predicción de sus modelos.

RFE es un método del tipo *wrapper* que consiste en utilizar algún clasificador para generar un modelo de datos y con éste un ranking de variables. Luego se descarta un cierto número de las variables con menor puntuación en base a parámetros de entrada del algoritmo y se repite la iteración con el nuevo subconjunto de variables hasta llegar a un mínimo. Por lo general, RFE descarta muchas variables en las primeras iteraciones y se vuelve más cuidadoso cuando la cantidad de variables es menor.

RFE es un *wrapper* particularmente efectivo ya que en lugar de generar un modelo de datos para todas las combinaciones de variables posibles en

una iteración dada, como lo hacen los algoritmos de *backward elimination* y *forward selection* [7], su foco está puesto en estimar el cambio en la precisión debido a la eliminación de una o más de dichas variables. Esto no resulta tan preciso como generar los modelos, pero es lo que permite al método ser temporalmente factible para problemas con un gran número de variables a considerar. Al usar SVM lineales, está demostrado [5] que la reducción del error está dada por el desplazamiento en la dirección del vector perpendicular al hiperplano de separación. Por eso, el valor de una variable i en un ranking R queda definido como

$$r_i = w_i,$$

donde w_i es la componente correspondiente a i en el vector W perpendicular al hiperplano de separación.

RFE es un algoritmo que agresivamente minimiza el error en cada iteración para el clasificador que use; por lo cual, si bien las tasas de error serán bajas, puede sufrir el problema de la estabilidad descrito en la sección anterior para problemas de variables altamente correlacionadas.

3.2. Fast Correlation Based Filter

El Filtro Rápido Basado en Correlaciones [12], de ahora en más FCBF, es un método del tipo *filter* desarrollado para la selección eficiente de variables a través del análisis de la relevancia y redundancia de las mismas, a diferencia de los otros métodos de selección que suelen centrarse únicamente en encontrar las variables más relevantes para el problema. Primero, el análisis de relevancia determina el subconjunto de variables relevantes eliminando aquellas que no lo son. Segundo, el análisis de redundancia elimina las variables redundantes de las relevantes dejando el subconjunto final. Este tipo

de búsqueda tiene como ventaja sobre otros algoritmos que al separar ambos análisis evita la búsqueda exhaustiva del subconjunto óptimo y computa una buena aproximación de forma eficiente.

La correlación se utiliza ampliamente en el contexto del aprendizaje automatizado como medición de relevancia. Comúnmente, las mediciones de correlación se clasifican en lineales y no lineales. Al trabajar sobre mediciones del mundo real, no es seguro suponer correlaciones del primer tipo, por lo que suele trabajarse con el concepto de la teoría de la información llamado "entropía", que es una medida de incertidumbre de una variable aleatoria.

La entropía H de una variable discreta X se define como

$$H(X) = -\sum_i \mathbf{P}(x_i) \log_2 \mathbf{P}(x_i),$$

y la entropía de X luego de observar valores de otra variable aleatoria Y es

$$H(X \mid Y) = -\sum_j \mathbf{P}(y_j) \sum_i \mathbf{P}(x_i \mid y_j) \log_2 \mathbf{P}(x_i \mid y_j),$$

donde $\mathbf{P}(x_i)$ es la probabilidad a priori de todos los valores de X y $\mathbf{P}(x_i \mid y_j)$ es la probabilidad a posteriori de todos los valores de X dadas las ocurrencias de los valores de Y.

La reducción en la entropía de X dada la ocurrencia de Y se denomina ganancia de información o *information gain* y se define como

$$IG(X \mid Y) = H(X) - H(X \mid Y)$$

De acuerdo con esta medida se puede ver que una variable aleatoria Y está más correlacionada con una variable X que con otra variable Z si

$$IG(X \mid Y) > IG(Z \mid Y)$$

Se puede probar que IG es una medición simétrica y asegura que el orden de dos variables no altera el resultado de la medición [10]. Como IG tiende a favorecer variables con más valores posibles se la normaliza con respecto a la entropía, de modo que se obtiene lo que se llama incertidumbre simétrica o *symmetrical uncertainty*

$$SU(X \mid Y) = 2\frac{IG(X \mid Y)}{H(X) - H(Y)}$$

SU compensa el sesgo de la ganancia de información para con las variables con más valores y restringe el resultado al intervalo $[0, 1]$ en problemas de clasificación binaria. Un valor de 1 significa la completa predicción de los valores de una variable dada la otra, y un valor de 0 significa total independencia. Las mediciones basadas en entropía utilizan mediciones discretas, por lo que los conjuntos de datos continuos deberán discretizarse de alguna manera para poder trabajar con esta medida.

Al usar SU como medida de correlación ya se puede comenzar a definir el algoritmo de selección de variables. Primero se definen dos tipos de correlación para el problema.

C-correlación: *la correlación entre una variable F_i y una clase C, denotada por $SU_{i,c}$.*
F-correlación: *la correlación entre un par de variables F_i y F_j con $i \neq j$, denotada por $SU_{i,j}$.*

Para mantener la eficiencia en un primer plano se calcula la C-correlación para cada variable y se utiliza un método heurístico para el cálculo de las F-correlaciones. Se dice que F_i es relevante si está altamente correlacionada con la clase C ($SU_{i,c} > \delta$ definido por el usuario). Para el cómputo de la redundancia surge la dificultad de no calcular todas las combinaciones posibles por su alto costo computacional en problemas de alta dimensionalidad, razón por la que se busca aproximar el resultado. Partiendo de la definición de coberturas de Markov, en lugar de buscar una medida exacta de redundancia se busca ahora una aproximación. Se parte de la suposición de que una variable con mayor C-correlación que otra tiene más información sobre la clase, y se determina la existencia de una cobertura aproximada de Markov entre dos variables de la siguiente manera:

Cobertura aproximada de Markov: *para dos variables relevantes F_i y F_j con $i \neq j$, F_j forma una cobertura aproximada de Markov para F_i si y sólo si* $SU_{j,c} > SU_{i,c} \quad \wedge \quad SU_{i,j} > SU_{i,c}$

Al contrario que las coberturas de Markov, la cobertura aproximada no asegura que la eliminación de una variable siga siendo válida en futuras iteraciones. Sin embargo, esto puede solventarse si sólo se eliminan variables para las cuales se encuentre una cobertura aproximada de Markov formada por una variable predominante.

Variable predominante: *una variable relevante es predominante si y sólo si no tiene ninguna cobertura aproximada de Markov en el conjunto actual de variables.*

Una variable predominante no será eliminada en ningún caso. Si una variable F_i es eliminada debido a la presencia de una cobertura aproximada dada por F_j, siempre se encontrará una cobertura aproximada en las futuras iteraciones: la misma F_j.

En resumen, el algoritmo filtra las variables relevantes, las ordena, y por cada una procede a eliminar aquellas variables cubiertas. De esta manera, se aproxima el subconjunto óptimo de variables a partir del conjunto de variables predominantes.

FCBF es un filtro univariado, es decir, no toma en cuenta los casos donde se necesita más de una variable para determinar un concepto, como en el caso de *xor*. Como tal, su precisión de modelado está seriamente restringida. Además, el algoritmo no considera variables de relevancia similar como posibles variables equivalentes y descarta una de ellas arbitrariamente sin establecer cuidados sobre la estabilidad de la solución. Por último, la penalización de variables no es proporcional a su nivel de redundancia, sino que directamente se elimina y ya no se la considera, privando al usuario de información sobre la variable en cuestión.

Algoritmo FCBF

Entrada:

- $S = \{F_1, \ldots, F_n, C\}$: conjunto de variables y clasificación.
- δ: umbral de corte para variables relevantes.

Salida:

- R: ranking óptimo de variables.

Pseudocódigo

for $F_i \in S$ **do**

Calcular $SU_{i,c}$ para F_i.

if $SU_{i,c} > \delta$ **then**

$S' = S' \cup \{F_i\}$.

end if

end for

Ordenar S' por $SU_{i,c}$ decreciente.

$F_j = pop(S')$

repeat

$F_i = next(S', F_j)$

repeat

if $SU_{i,j} > SU_{i,c}$ **then**

$S' = S' - \{F_i\}$

end if

$F_i = next(S', F_i)$

until $F_i == NULL$

$F_j = next(S', F_j)$

until $F_j == NULL$

$R = S'$

return R

3.3. SVM-RFE con filtro MRMR

El método de Mínima Redundancia y Máxima Relevancia o MRMR [8] apunta, como su nombre lo indica, a seleccionar variables máximamente relevantes y mínimamente redundantes para la clasificación, mediante la combinación del filtro MRMR con el algoritmo SVM-RFE.

El filtro MRMR fue introducido por Peng et al. [9]. Sea $S = \{F_1, F_2, \ldots, F_n\}$ un conjunto indexado de variables. Sean C las clases objetivos de las muestras. La información mutua entre la variable F_i y la clase C determinará la relevancia de F_i para la clasificación. Entonces, la relevancia R_i está dada por:

$$R_i = I(C, F_i).$$

La redundancia de la variable F_i con las otras variables del subconjunto S está dada por:

$$Q_{S,i} = \frac{1}{|S|^2} \sum_{F_j \in S, F_j \neq F_i} I(F_i, F_j)$$

Con el filtro MRMR, el ordenamiento de las variables se realiza optimizando la proporción entre el valor de relevancia de una variable contra el valor de redundancia con las otras variables del conjunto. La variable máximamente relevante y mínimamente redundante F_i^* en el conjunto S está dada por

$$F_i^* = argmax_{F_i \in S} \frac{R_i}{Q_{S,i}}$$

El filtro MRMR usado por sí solo puede no ser capaz de obtener una precisión óptima ya que actúa independientemente del clasificador y no está involucrado en la verdadera selección de variables. Por otro lado, SVM-RFE no toma en cuenta la redundancia entre variables. El objetivo es mejorar la selección de variables de SVM-RFE combinándolo con el filtro MRMR para

que minimice la redundancia entre variables relevantes. Para eso, lo que se plantea es ordenar las variables por el resultado de la combinación convexa entre la relevancia marcada por los pesos de la SVM y los valores del filtro. Para la i-ésima variable se define

$$r_i = \beta|w_i| + (1-\beta)\frac{R_i}{Q_{S,i}},$$

donde $\beta \in [0,1]$ define la prevalencia del ranking SVM y el ranking MRMR.

Un problema que surge del estudio de este algoritmo es que la selección es inestable. Cuando se encuentra un par de variables redundantes se penaliza a las dos por igual en lugar de determinar un criterio por el cual una de ellas es mejor que la otra. Además, en esta situación una variable poco relevante y poco redundante puede llegar a ocupar un mejor lugar en el ranking final desplazando variables que sí son relevantes al problema. A su vez, el algoritmo no hace ningún esfuerzo por controlar la escala de los rankings a la hora de combinarlos.

Algoritmo SVM-RFE con filtro MRMR

Entrada:

- $S = \{F_1, \ldots, F_n, C\}$: conjunto de variables y clasificación.
- β: factor de importancia entre RFE y MRMR.

Salida:

- R: ranking óptimo de variables.

Pseudocódigo

repeat
  $W = rfe(S)$
  for $F_i \in S$ **do**
    Calcular $R_{S,i}$ y $Q_{S,i}$
    Calcular $r_{S,i}$
  end for
  $F_i^* = argmin(r_i)$
  $R = R \cup \{F_i^*\}$
  $S = S \backslash \{F_i^*\}$
until $S = \emptyset$
return R

3.4. Stable Recursive Feature Elimination

El algoritmo SRFE es el método de selección de variables desarrollado como eje de esta tesina. Es un algoritmo de la familia *wrapper* que intenta minimizar, combinando los puntos fuertes de los algoritmos ya descriptos, el problema de la estabilidad que se genera debido a altos niveles de correlación entre las variables. Combina un método para seleccionar la variable a penalizar de una manera estable siguiendo el ejemplo de FCBF con la base de selección precisa y eficiente de SVM-RFE.

En esencia, el algoritmo cuenta en primera instancia con un filtro inspirado en FCBF que se encarga de determinar los niveles de correlación y discriminar las variables intercambiables en el momento del aprendizaje. De esta manera, se genera un vector de penalizaciones siguiendo un criterio que considera la estabilidad de la solución, la cual luego se combina con el ranking obtenido a través de RFE para determinar qué variables eliminar. Luego, se itera hasta que todas las variables estén ordenadas.

La relevancia de una variable F_i en el problema queda determinada entonces por la ecuación

$$r_i = \beta I_i + (1 - \beta) P_i,$$

donde I_i es la importancia de la variable en cuanto a su aporte a la solución y P_i es la penalización a la misma por redundancia. Como el método utiliza el algoritmo SVM-RFE como base, I_i puede remplazarse por w_i, que representa la componente asociada a la variable F_i en el vector que define el hiperplano clasificador, quedando finalmente

$$r_i = \beta |w_i| + (1 - \beta) P_i$$

3.4.1. El vector de penalización P

La construcción del vector de penalización P se da en el bucle inicial del algoritmo, que consta del cálculo de la medida de SU para todos los pares de variables y de las variables con las clases. Si bien tiene complejidad $O(n^2)$, el cálculo es sencillo y se realiza una sola vez, con lo cual se amortiza frente a las n iteraciones posteriores del algoritmo de aprendizaje automatizado. Esto determina el vector de C-correlación $SU_{i,c}$ y la matriz de F-correlación $SU_{i,j}$. Con estos valores se procede a la construcción del vector P de penalización de variables mediante la búsqueda del par de variables tal que

$$(F_i, F_j) = argmax(SU_{i,j})$$

SRFE, a diferencia de MRMR, penaliza a una sola de las variables del par. Si los valores de correlación $SU_{i,c}$ y $SU_{j,c}$ difieren en menos de un umbral de tolerancia T_p entonces se considera que F_i y F_j aportan el mismo valor predictivo al modelo y, para mantener un criterio estable, se penaliza con un valor de $-SU_{i,j}$ a aquella variable con mayor subíndice y se la remueve del conjunto. Si la diferencia sobrepasa el umbral, se considera que las variables son redundantes pero hay una clara relevancia de una por sobre la otra y aquella con menor poder predictivo es penalizada con ese valor y removida. El procedimiento se itera hasta completar el vector P. La severidad de la penalización queda dada por el nivel de redundancia mismo entre las variables. Durante la experimentación, el umbral T_p fue establecido en 0.05 o 5 % de diferencia en la correlación para salvar perturbaciones dadas por el ruido y el muestreo aleatorio.

La iteración de este cómputo devuelve un vector P donde el índice correspondiente a una sola de las variables tiene valor 0 (la de mayor relevancia) y

las demás tienen un valor $p \in [-1, 0)$ dependiendo del nivel de redundancia medido. Sin embargo, es errado pensar que todas las variables deban ser penalizadas en base a una sola, ya que estaríamos implicando una redundancia entre todas las variables. El problema es que, en la práctica, $SU_{i,j}$ es siempre distinto de 0, aun para variables independientes. Para solventar esto y mejorar el filtro se planteó la inclusión de un umbral de correlación T_c, inspirado en los algoritmos de clustering jerárquico. El valor de T_c determina el valor de correlación en el cual dos variables ya no se consideran redundantes. De esta manera, T_c actúa como mecanismo de corte del bucle (o del árbol, si se sigue con la idea de clustering) y determina un factor de agrupamiento de las variables: $T_c = 0$ indica que todas las variables deben considerarse redundantes entre sí (un solo cluster), mientras que $T_c = 1$ indica la consideración de independencia total de las variables (cada variable es un cluster). Un valor adecuado de T_c generará varios conjuntos de variables redundantes en distinta medida con una de ellas especialmente relevante y sin penalizar.

3.4.2. El método SRFE

Una vez obtenido el vector de penalizaciones se procede al entrenamiento del algoritmo de aprendizaje automatizado. En la implementación de SRFE se decidió utilizar SVM-RFE por su eficiencia ya descrita. Al obtenerse el vector de pesos W se lo escala junto con el vector P en base a la cantidad restante de variables y se los combina convexamente en base a un parámetro β. Las peores variables son removidas de la lista y se repite el proceso (incluyendo el escalado) hasta que se finaliza el ranking.

Como ventaja potencial sobre los métodos ya mencionados, SRFE es menos avaro que RFE al ofrecer un control adicional, opcional y escalable sobre la eliminación de variables, brindando más estabilidad y coherencia en las

selecciones. Además, obtiene mejores resultados que FCBF en cuanto a error en la clasificación debido a su naturaleza *wrapper*, y supera a SVM-RFE con filtro MRMR, ya que considera una mejor escala al momento de combinar las penalizaciones y deja de manera estable una variable de cada grupo sin penalizar.

Algoritmo SRFE

Entrada:

- $S = \{F_1, \ldots, F_n, C\}$: conjunto de variables y clasificación.
- β: factor de balance entre importancia y redundancia.
- T_p: factor de tolerancia para determinar la equivalencia de dos variables.
- T_c: factor de corte en el cálculo de redundancia.

Salida:

- R: ranking óptimo de variables.

Pseudocódigo

for $F_i \in S$ **do**
  Calcular $SU_{i,c}$
end for
for $F_i, F_j \in S$ **do**
  Calcular $SU_{i,j}$
end for
$S' = S$
repeat
  $(F_i, F_j) = argmax(SU_{i,j})$
  if $SU_{i,c} > SU_{j,c}(1 + T_P)$ **then**

$P[j] = -SU_{i,j}$

$remove(F_j)$

else if $SU_{j,c} > SU_{i,c}(1 + T_P)$ **then**

$P[i] = -SU_{i,j}$

$remove(F_i)$

else

$w = min(i, j)$

$P[w] = -SU_{i,j}$

$remove(F_w)$

end if

until $S' = \emptyset \vee SU_{i,j} < T_C$

repeat

$W = rfe(S)$

Escalar W y $P[S]$

$F_i^* = argmin(\beta W + (1 - \beta)P[S])$

$R = R \cup \{F_i^*\}$

$S = S \backslash \{F_i^*\}$

until $S = \emptyset$

return R

4. Resultados

La experimentación se realizó sobre distintos conjuntos de datos o *data-sets* de diversas características. Sobre todos ellos se corrieron los algoritmos descriptos en el capítulo tres para la obtención de los rankings finales. A continuación, se utilizó el algoritmo SVM con los rankings como argumento para medir la tasa de error en base a la cantidad de variables seleccionadas. Asimismo, se evaluó la estabilidad de la selección. No se tuvo en consideración el factor temporal o de memoria para determinar la efectividad de los métodos.

Todos los métodos utilizaron la misma entrada aleatoria y realizaron las mimas validaciones para garantizar la equidad en la experimentación. La obtención de los cien conjuntos de datos de entrenamiento y de prueba se realizó con el método de *subsamples* del 75 % para entrenamiento y los modelos generados se validaron internamente con las SVM utilizando la técnica de *4-fold cross validation*. Las SVM utilizadas fueron configuradas con un kernel lineal y el parametro de costos C se eligió por validación interna entre $(0.01, 0.1, 1, 10, 100)$.

En cada una de las secciones siguientes se confeccionaron cuatro gráficas.

La primera de ellas muestra, utilizando un mapa de calor o *heatmap*, el nivel de correlación entre las variables. Cuanto mayor es la correlación, el color tiende del rojo oscuro al blanco. Esta gráfica sirve, en conjunción con las gráficas de estabilidad, para determinar si la selección fue redundante o no. Es común encontar soluciones perfectamente estables pero que tras un análisis de redundancia revelan su baja calidad. No sólo se busca hallar un conjunto de variables estable a lo largo de los experimentos, sino que éste esté, además, formado por variables independientes entre sí de manera que la información a extraer sea más abundante.

En la segunda, se pueden observar cuatro curvas en las que se comparan los errores de clasificación del modelo generado con un número dado de variables elegidas por los distintos métodos. Esta gráfica sirve para mostrar que el método SRFE obtiene modelos de predicción al menos tan buenos como el resto de los métodos ya publicados.

La tercera gráfica muestra, para cada método en forma independiente, la estabilidad de la selección para un número definido de variables. Para esto, se grafica la probabilidad de una variable de ser seleccionada entre las x primeras sobre todos los experimentos realizados. En esta gráfica, se busca que las probabilidades de las primeras x variables tiendan a 1, mientras que el resto tienda a 0, lo que determina una selección estable. El valor de x se determinó en todos los casos buscando un mínimo en las curvas de error de la primera figura.

La última gráfica muestra, para cada variable, la posición final luego del entrenamiento para cada uno de los rankings obtenidos por los métodos. Se presentan ordenadas de mejor a peor y se busca observar poca variación por lo menos en las primeras x posiciones. Cuando se grafica el dataset completo, cuanto más se asemeja la gráfica a una diagonal, más estable es el método de selección. La presencia de dos o más posiciones distintas y bien marcadas para una variable puede indicar una alta correlación con otras, lo que sugiere que habrían intercambiado lugares a lo largo de los experimentos.

El parámetro β, donde aplica, fue configurado en 0.5 dando igual importancia a ambas partes en el ranking. Para SRFE, el parámetro T_p se definió en 0.05 por no ser crítico y T_c se ajustó de manera empírica.

4.1. Datos artificiales

Para mostrar con claridad las bondades del método presentado en esta tesina, se procedió a la generación de un conjunto de datos que responda a la problemática que se intenta resolver: la selección estable de variables en problemas anchos con una gran cantidad de variables correlacionadas.

El dataset "artificial" consta de cien variables agrupadas en conjuntos altamente correlacionados de veinticinco variables. A su vez, el concepto objetivo, que es binario, está determinado por una conjunción de los cuatro grupos, lo que significa que es necesaria la presencia de al menos una variable de cada grupo para obtener la solución correcta. La Figura 4 muestra con un heatmap la estructura de correlaciones entre las variables.

Para este caso particular se hicieron tres experimentos con cada vez menos muestras para exponer cómo se va perdiendo la capacidad predictiva a medida que disminuye la información y el espacio se torna más ralo. Además, se muestra cómo la información redundante elude los criterios de selección de los métodos distintos a SRFE.

En la primera instancia, de mil variables para una relación $1 : 10$ de variables y muestras, se puede observar en la Figura 5 como únicamente el método SRFE logra obtener el modelo óptimo al minimizar el error con cuatro variables. Tanto RFE como MRMR mantienen variables redundantes en las últimas iteraciones y FCBF falla en obtener cualquier tipo de información de los datos porque elimina secuencialmente grupos enteros de variables (de ahí los grandes saltos en la gráfica).

Las Figuras 6 y 7 exponen la estabilidad de las soluciones. La primera figura muestra la probabilidad de que una variable dada sea seleccionada entre las más importantes para el problema dado. En este caso ese número se fijó en cuatro variables. La segunda, muestra la distribución completa

de posiciones que toma cada variable en el ranking a lo largo de todas las corridas efectuadas. Mientras más estable es un método, más angosta serán las distribuciones.

En dichas figuras de estabilidad se acentúa más aún la diferencia en la selección. MRMR muestra una estabilidad excelente, pero siempre elige dos pares de variables correlacionadas, con lo cual el modelo resultante es malo. RFE tiene una selección inestable y errónea que no aporta información útil a la interpretación de los resultados al igual que FCBF. SRFE elige de manera estable una variable de cada conjunto.

En las segunda y tercera instancias, con relaciones $1:1$ (Figuras 8 a 10) y $4:1$ (Figuras 11 a 13), respectivamente, se puede observar como progresivamente aumenta el error de los modelos de predicción generados y disminuye la estabilidad para todos los métodos excepto para SRFE que demuestra su robustez incluso cuando comienzan a surgir los errores de medición consecuencia de la escasez de muestras[1].

[1]Debido a que los cálculos con métricas de entropía o información requieren de variables discretas, es necesario realizar una discretizacion utilizando *bins* de datos. Para generar estos bins con una densidad útil se utiliza una cantidad bastante estándar de $\sqrt{n}$ bins donde n es la cantidad de muestras o puntos. Cuando la cantidad de muestras es muy baja (y se reduce más al usar subsamples), el número de bins no alcanza para obtener una medición fidedigna y los resultados tienen menos precisión.

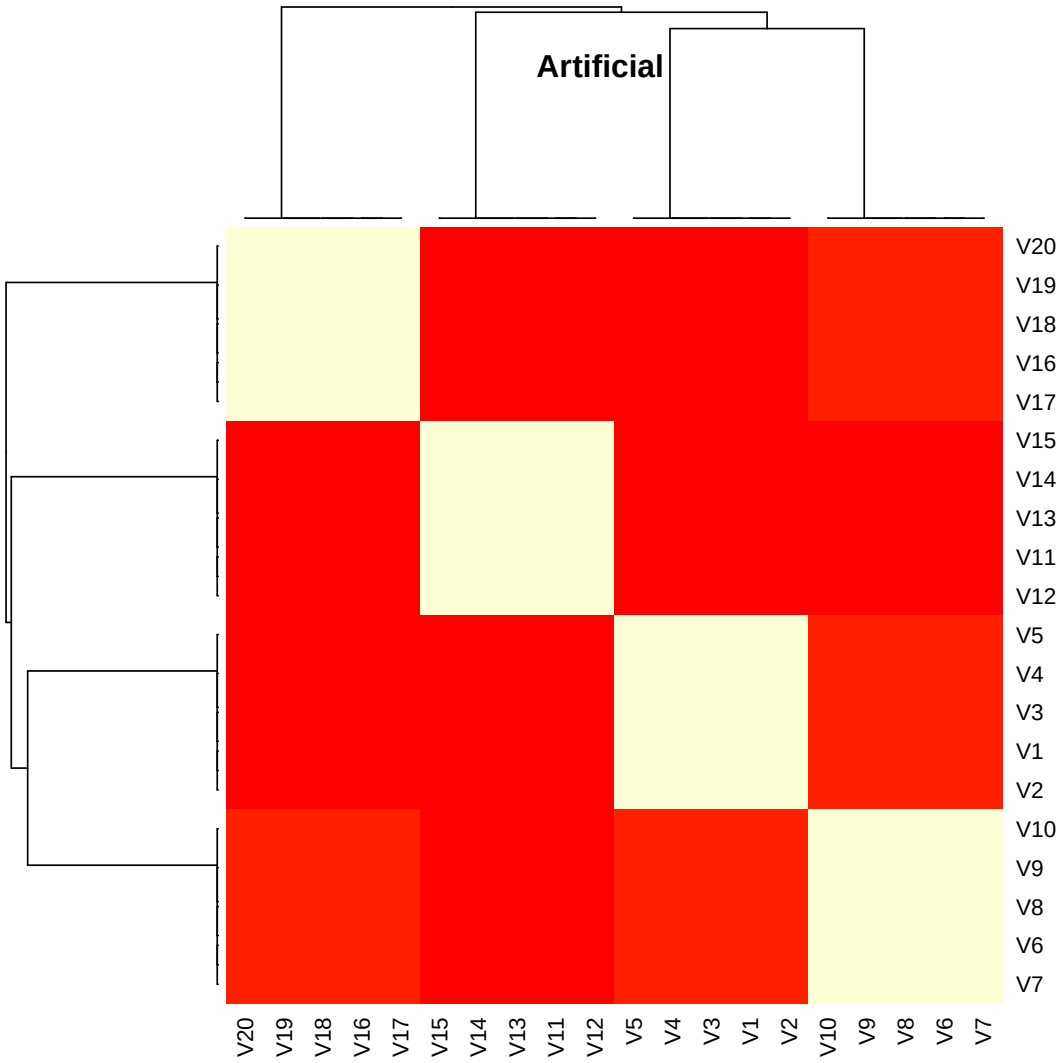


Figura 4: Dataset Artificial: mapa de calor que muestra la correlación entre las variables del dataset (sólo un subconjunto del total). Se pueden ver cuatro grupos de variables altamente correlacionadas dentro del grupo pero independientes de los otros grupos.

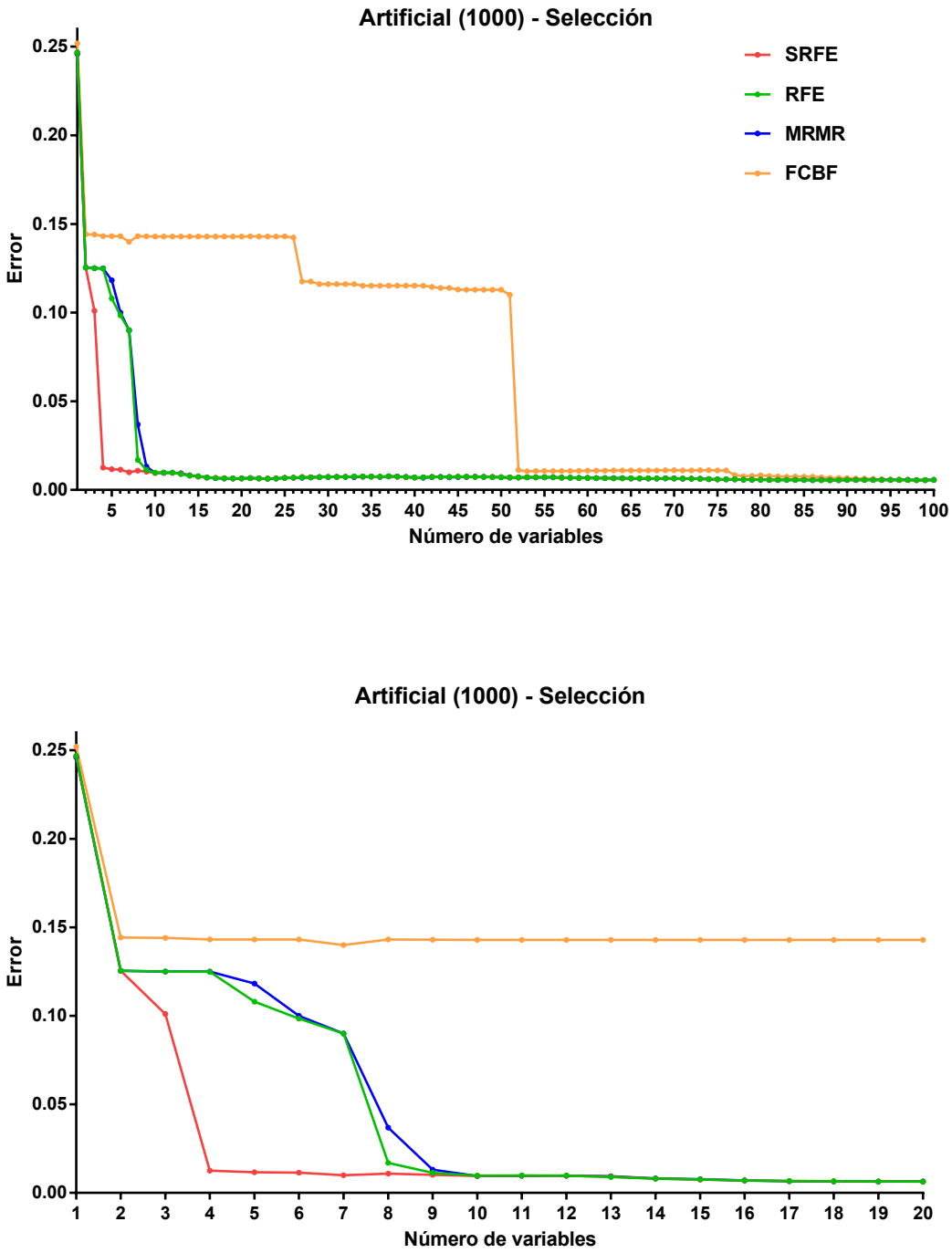


Figura 5: Dataset Artificial (1000 puntos): nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

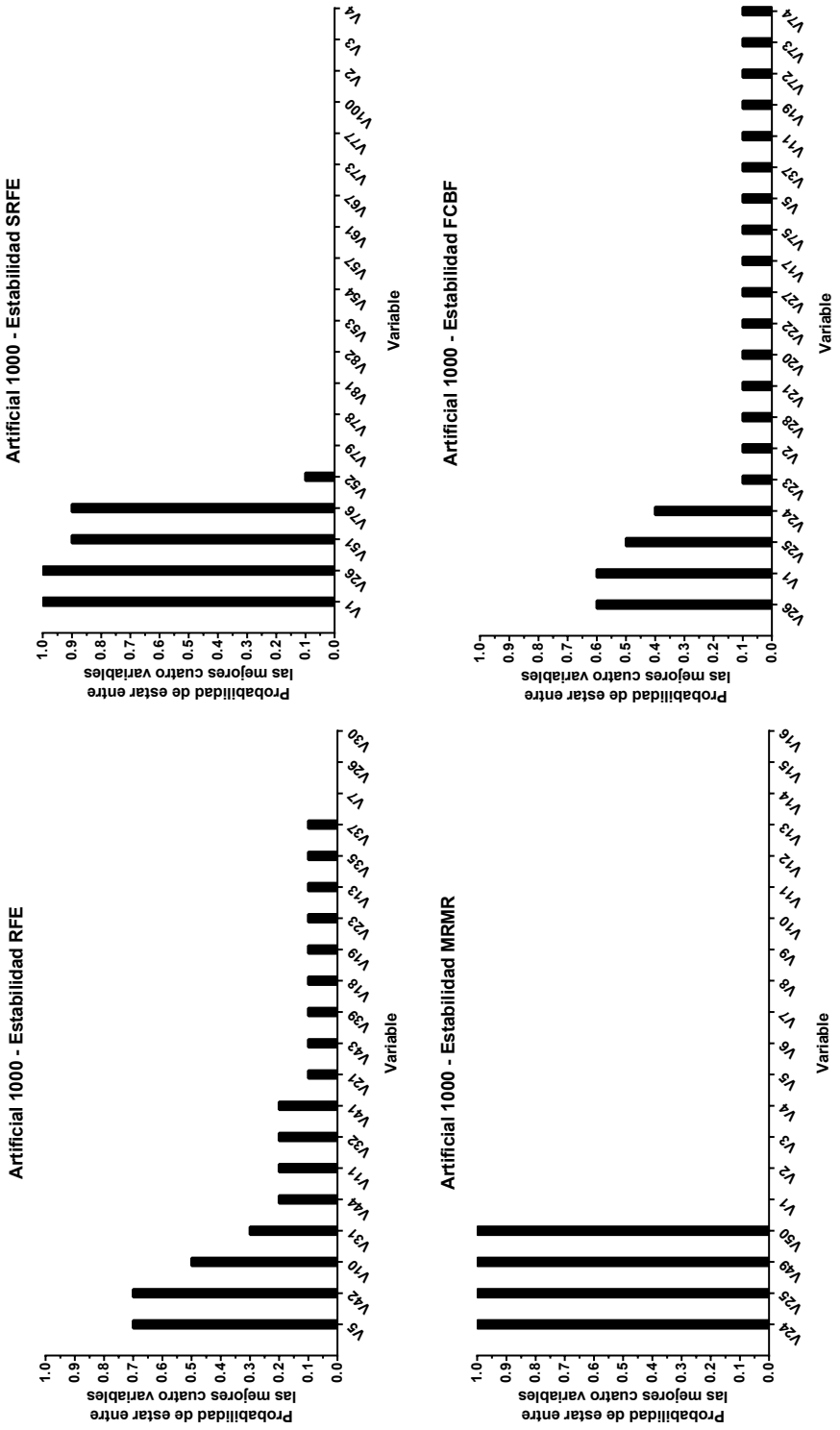


Figura 6: Dataset Artificial (1000 puntos): estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras cuatro.

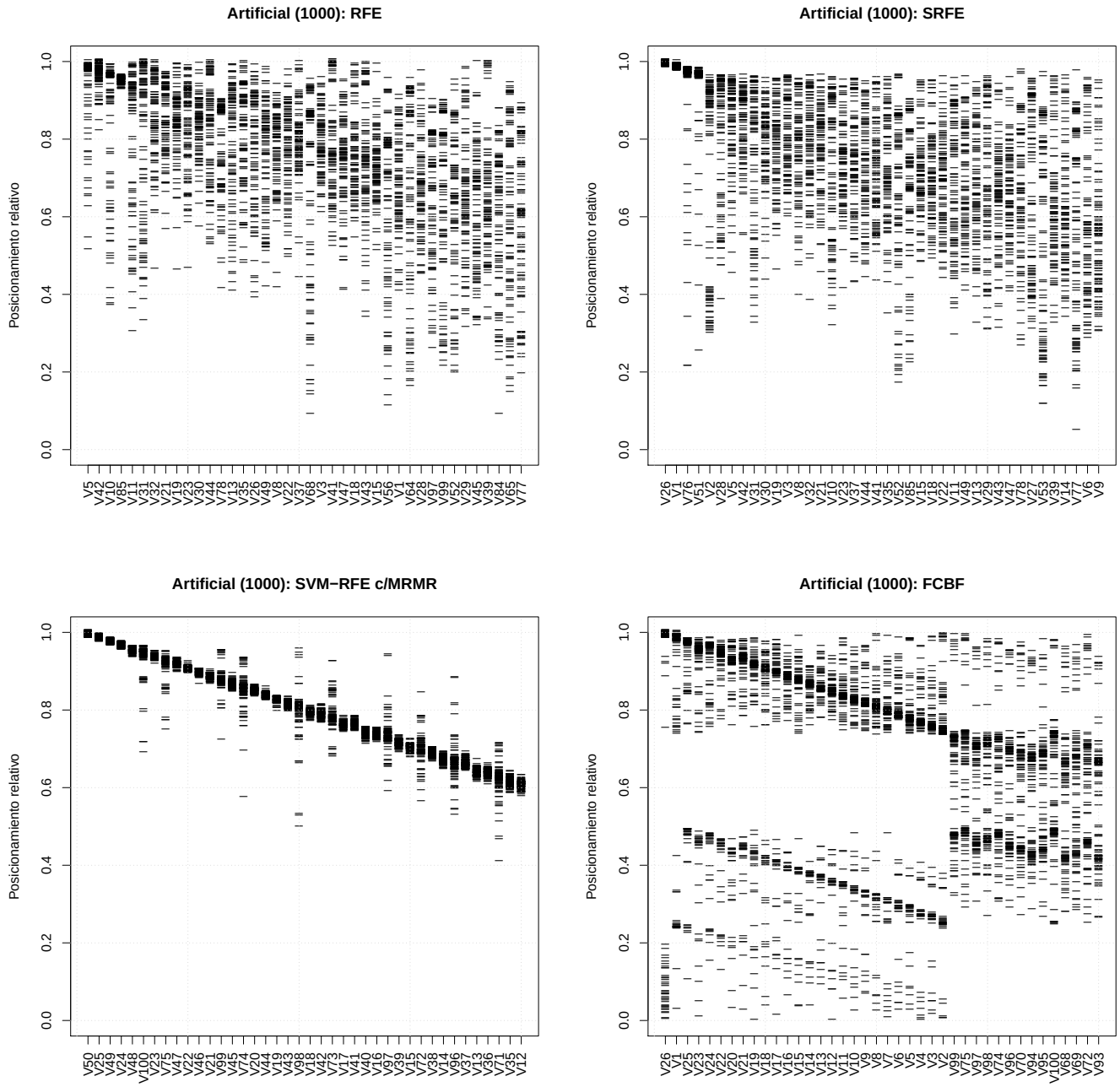
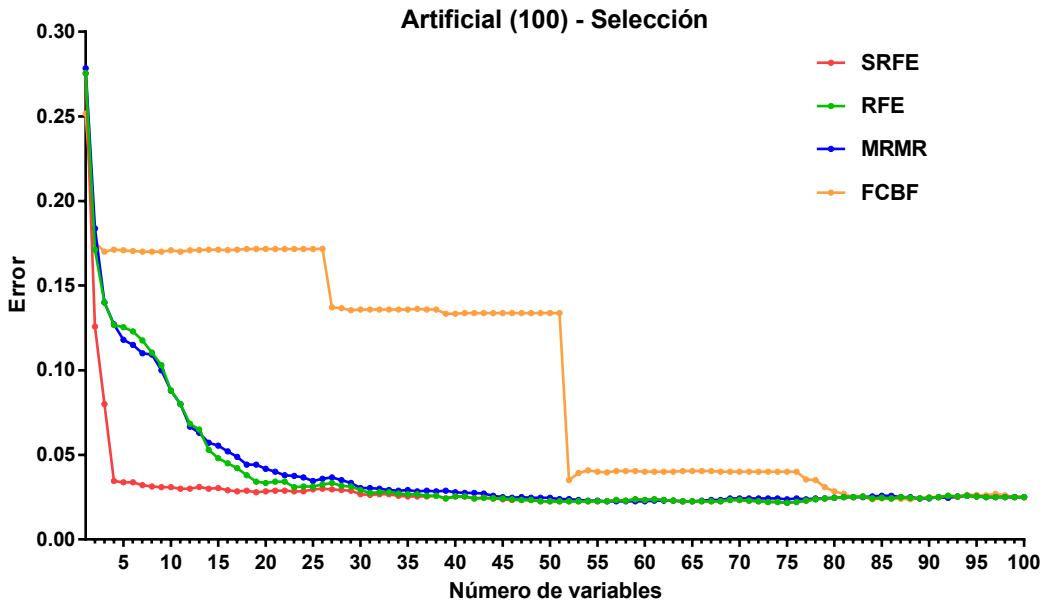


Figura 7: Dataset Artificial (1000 puntos): distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.



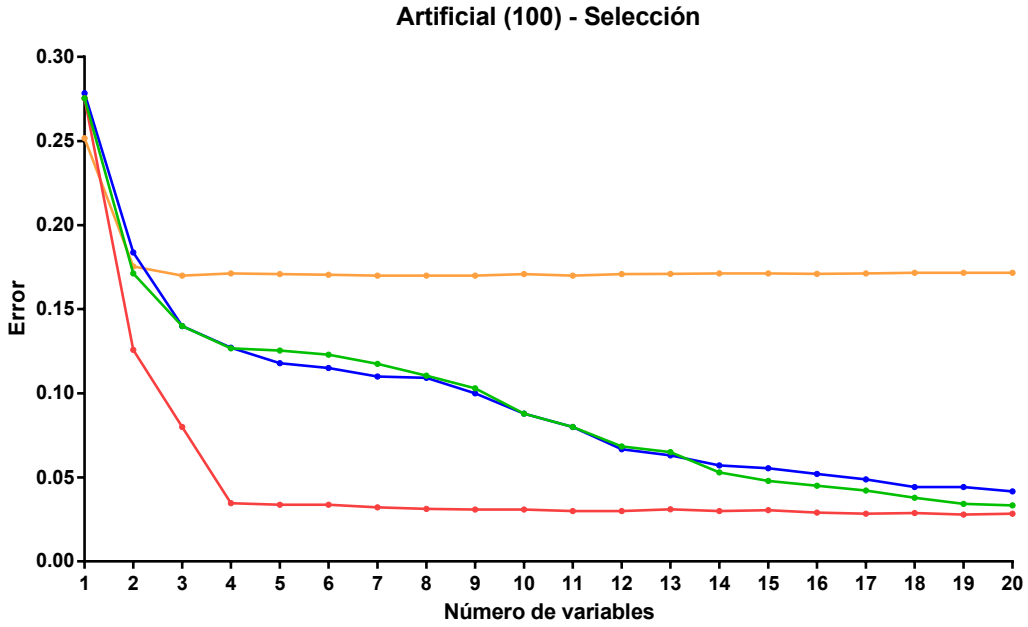


Figura 8: Dataset Artificial (100 puntos): nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

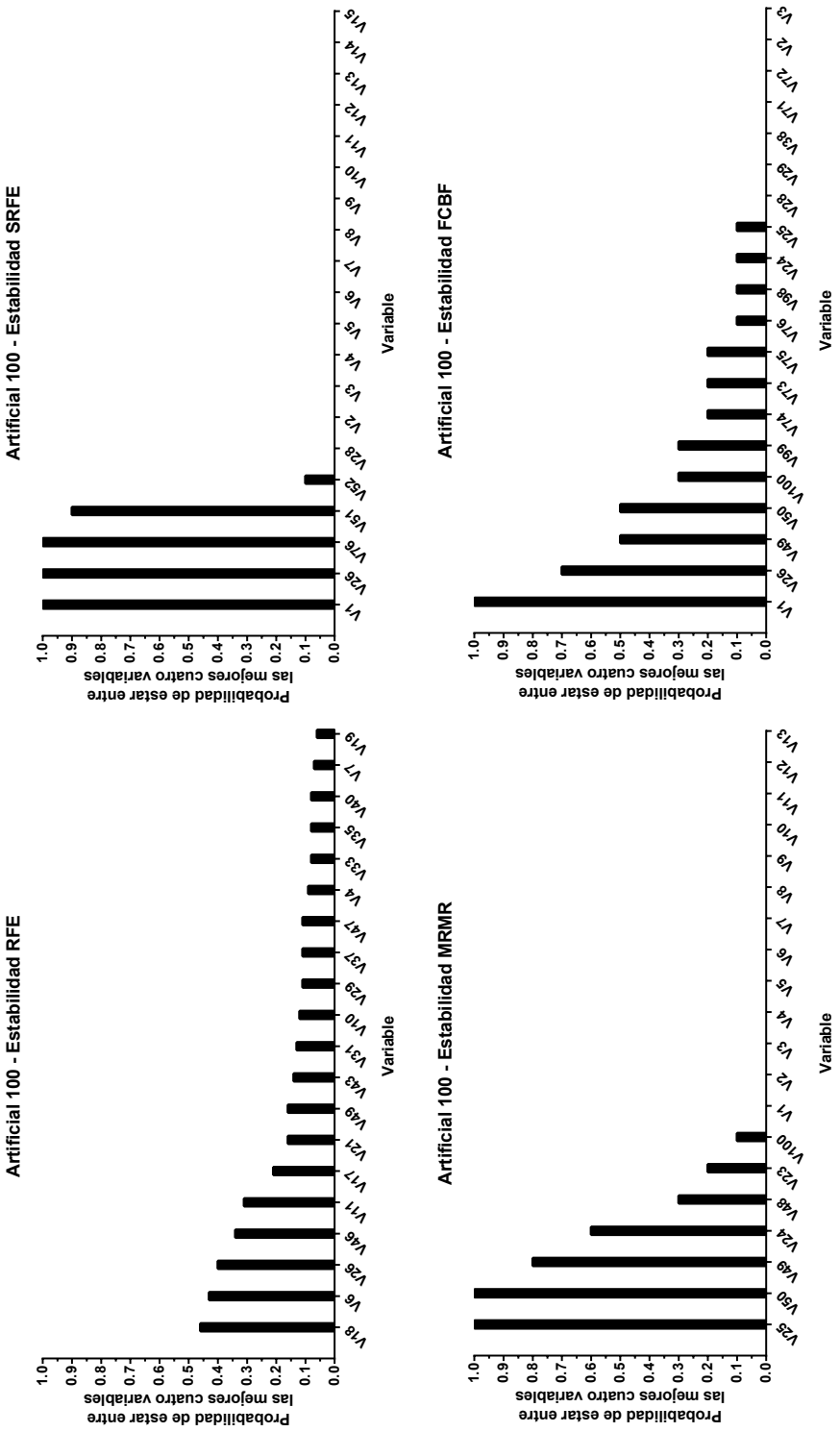


Figura 9: Dataset Artificial (100 puntos): estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras cuatro.

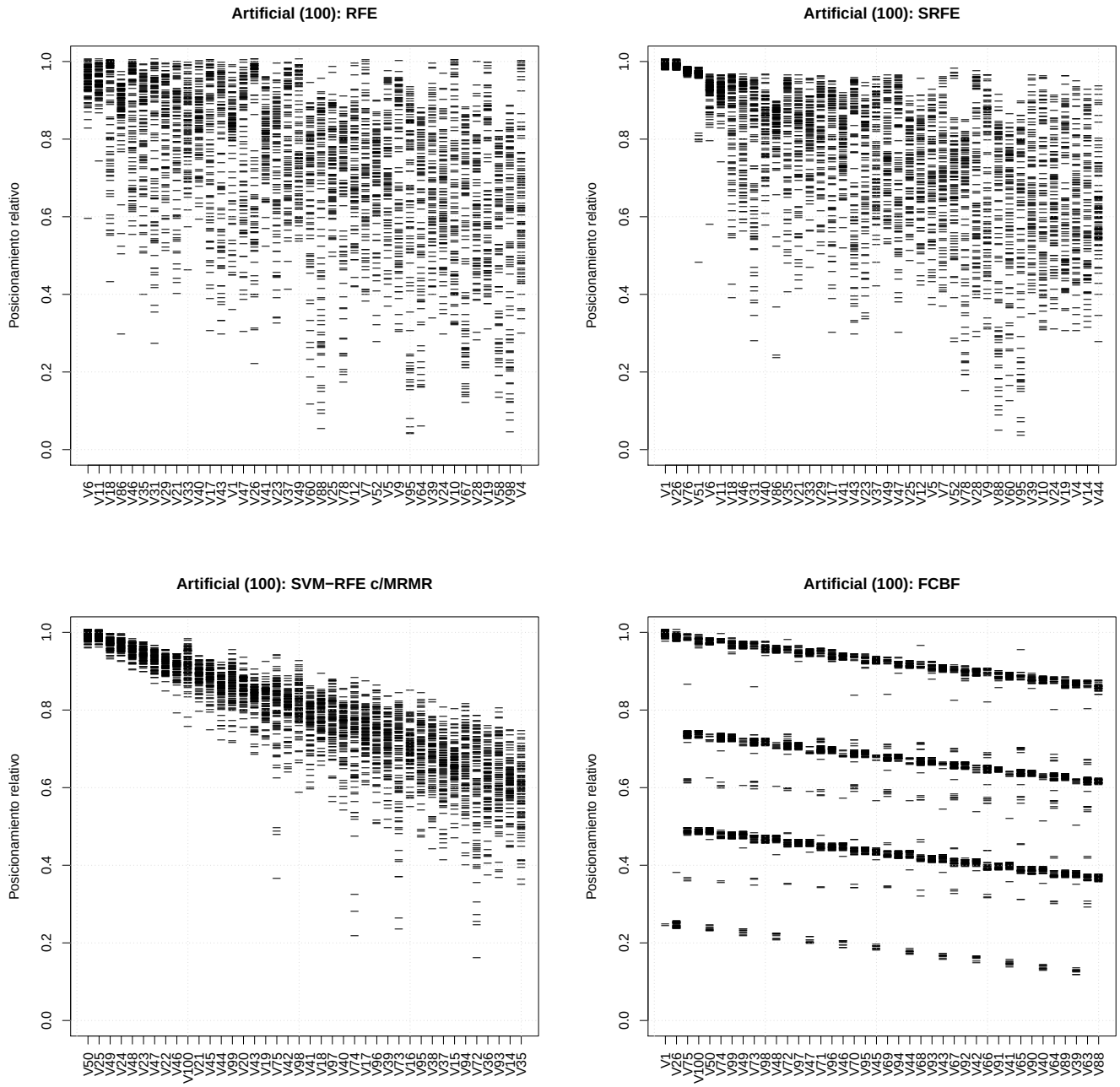


Figura 10: Dataset Artificial (100 puntos): distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

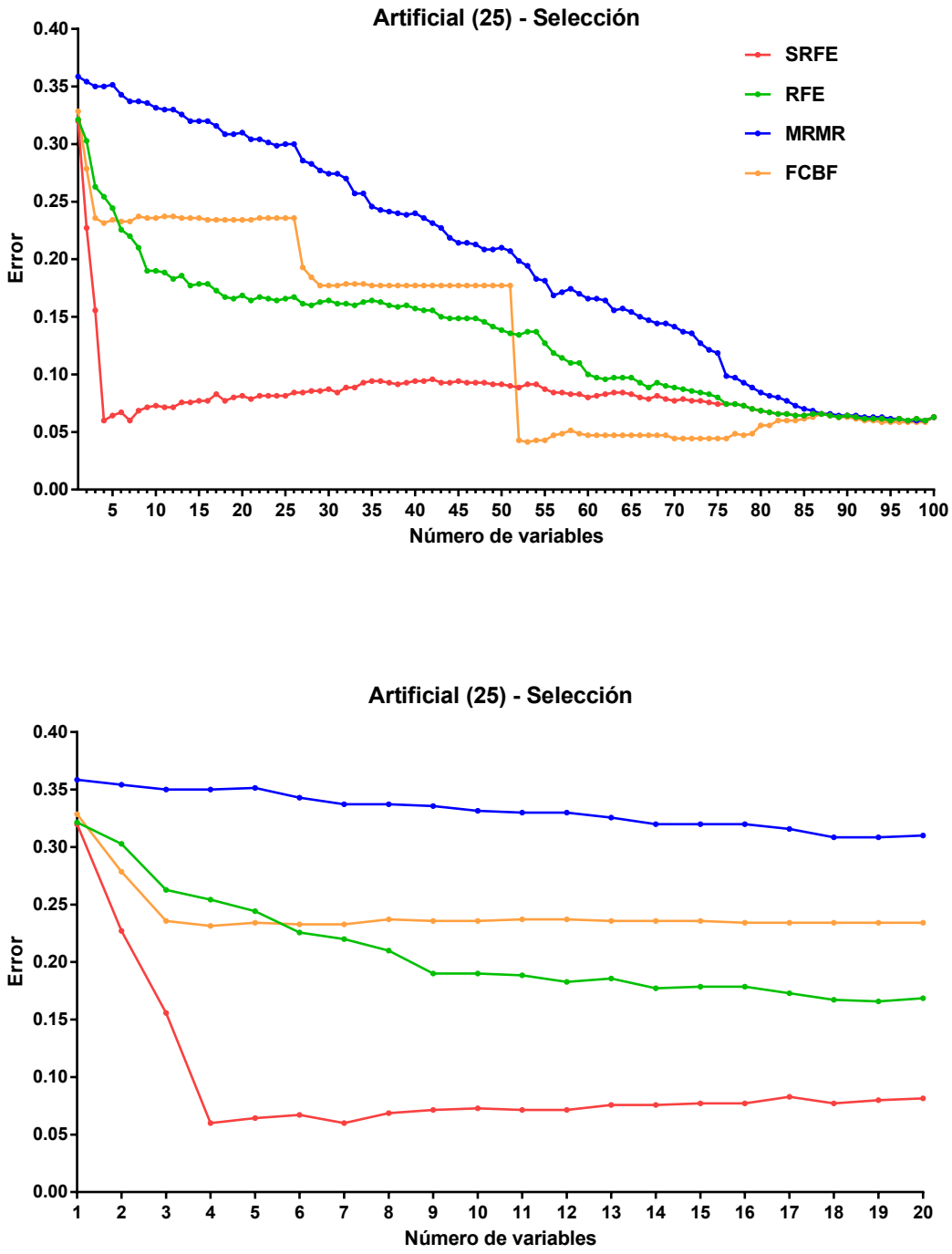


Figura 11: Dataset Artificial (25 puntos): nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

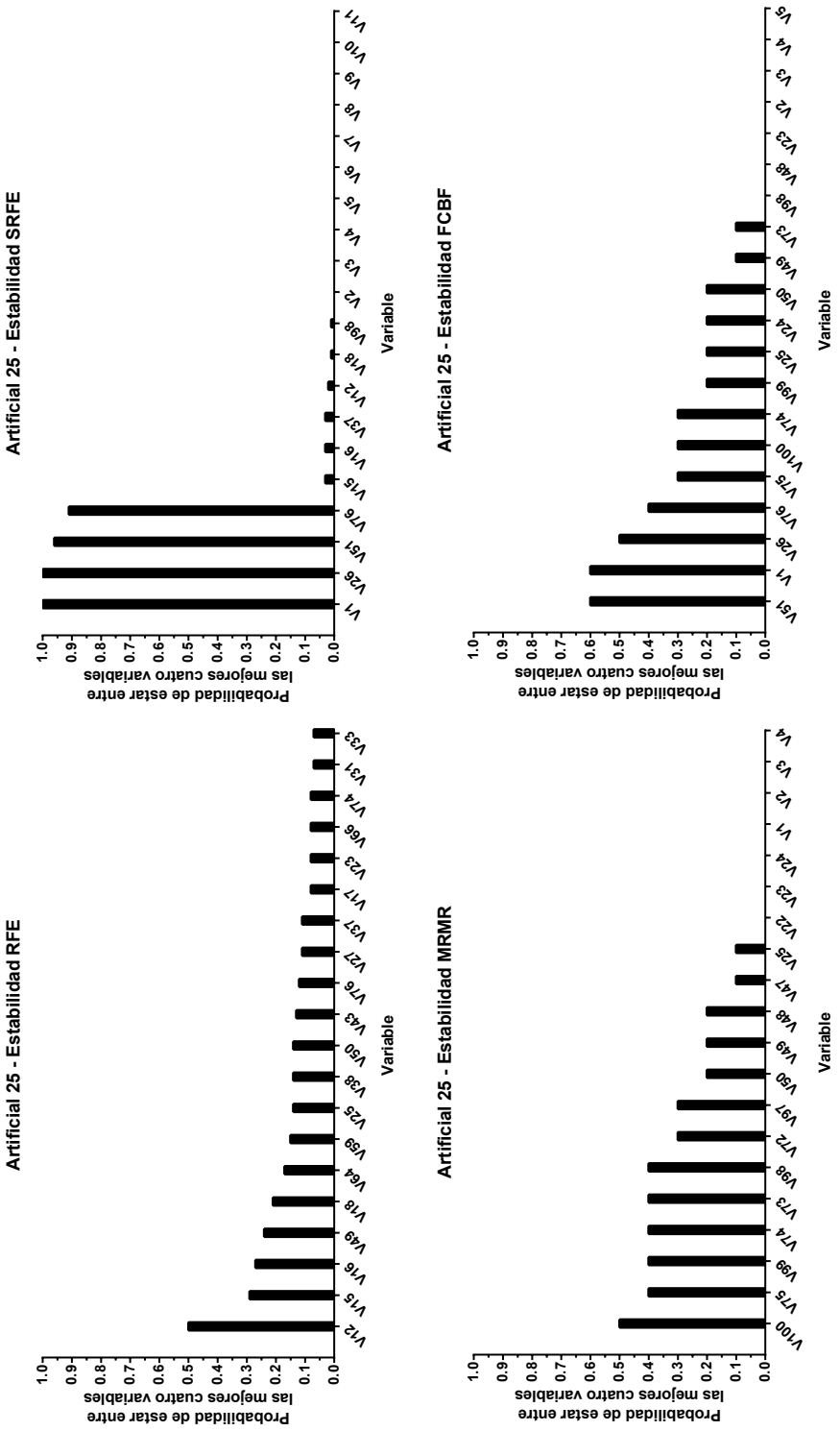


Figura 12: Dataset Artificial (25 puntos): estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras cuatro.

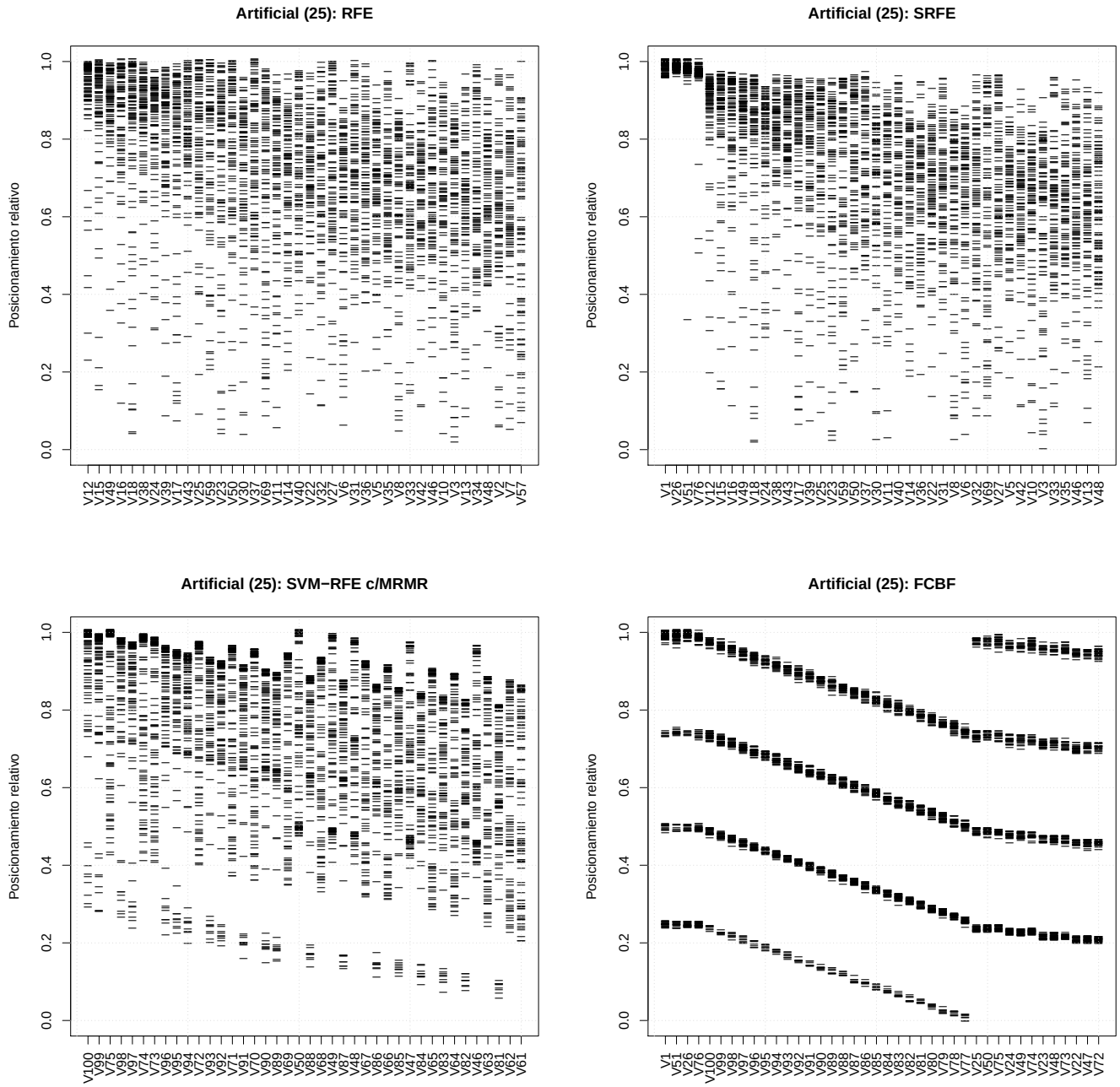


Figura 13: Dataset Artificial (25 puntos): distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

4.2. Datos reales

Los siguientes datasets corresponden a mediciones sobre problemas del mundo real. En particular, se tratan problemas de espectrometría por su cantidad de variables y problemas de colorimetría por sus altos niveles de correlación.

4.2.1. Espectrometría

Mele

Este dataset contiene mediciones sobre muestras de distintas especies de manzanas realizadas con un espectrómetro de masa de reacción de transferencia de protón (PTR-MS) acoplado con un detector de tipo tiempo de vuelo (TOF). El espectrómetro mide la composición del aire alrededor de la manzana (el "olor") y da como respuesta la proporción de compuestos para cada masa atómica desde 1 hasta 250 AMU, aproximadamente.

En este caso particular se analizaron 150 muestras de tres variedades distintas de manzanas. Se descartaron las masas con lectura cero y las masas correspondientes al agua e isótopos, quedando un total de 714 variables en el dataset. El objetivo fue discriminar las variedades de manzana en base al perfil químico de su olor. La Figura 14 muestra el heatmap de correlaciones para una parte de las variables, donde se observa la típica correlación en espectrometría de masa entre pequeños grupos de variables, en general isótopos o fracciones de un mismo compuesto.

Los resultados de este experimento muestran un subconjunto final óptimo de unas tres variables encontrado por RFE y SRFE, seguido de cerca por FCBF y por último por MRMR, que finaliza con un poco más de nivel de error (Figura 15).

Siendo un verdadero problema ancho, y si bien los niveles de error son siempre bajos, ninguno de los métodos utilizados presenta una gran estabilidad en la selección salvo el método MRMR (Figuras 16 y 17). Este método, sin embargo, produce casi el doble de error que RFE y SRFE con pocas variables. Se puede ver también como el único método que no penaliza las redundancias (RFE) presenta un par de variables con alta correlación entre las seleccionadas (26.01 y 42.01).

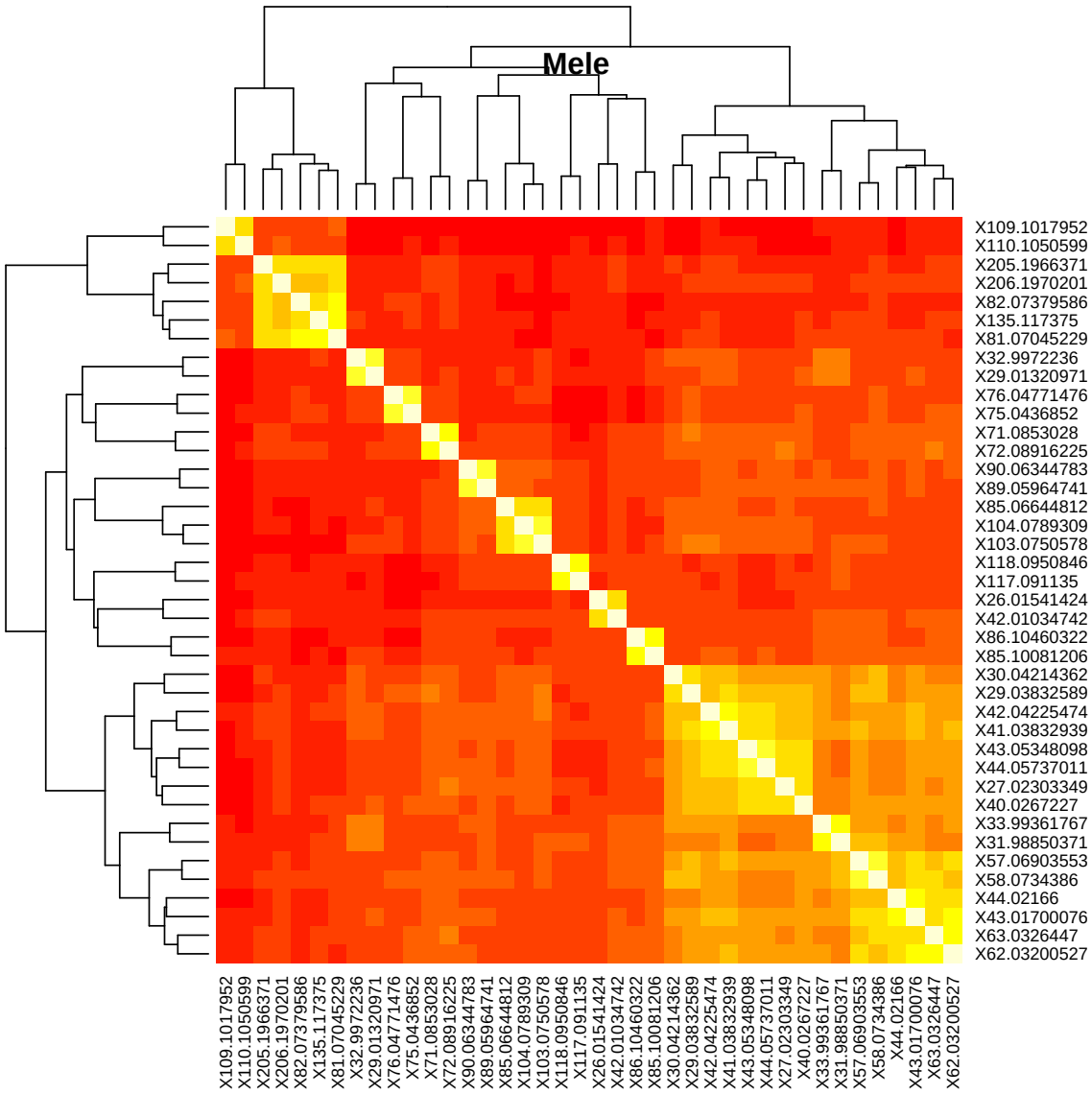


Figura 14: Mele: mapa de calor que muestra la correlación entre las variables del dataset (sólo un subconjunto del total). Se pueden distinguir varios grupos de dos variables, generalmente isótopos del mismo compuesto, y un par de grupos de mayor tamaño.

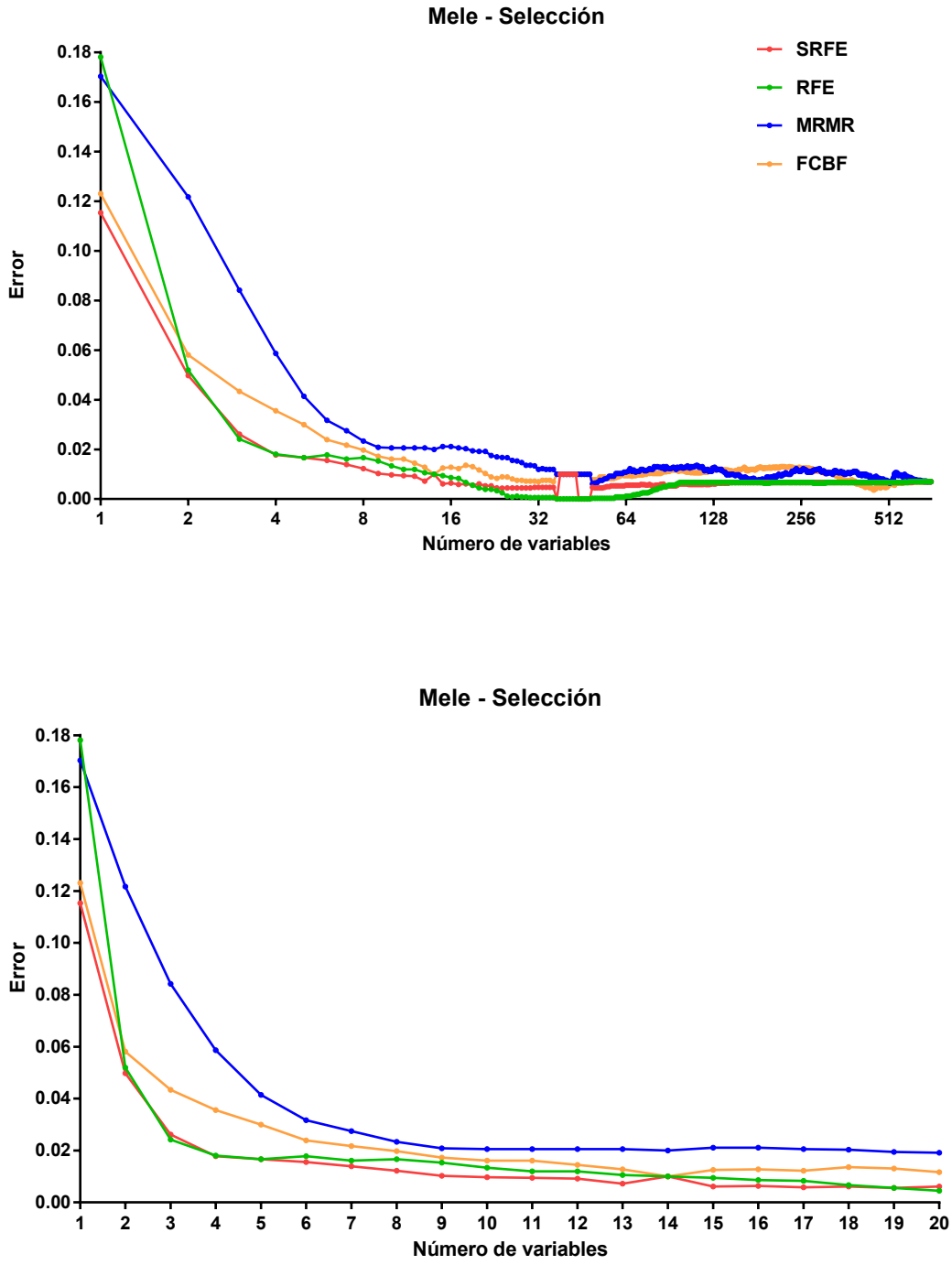


Figura 15: Mele: nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

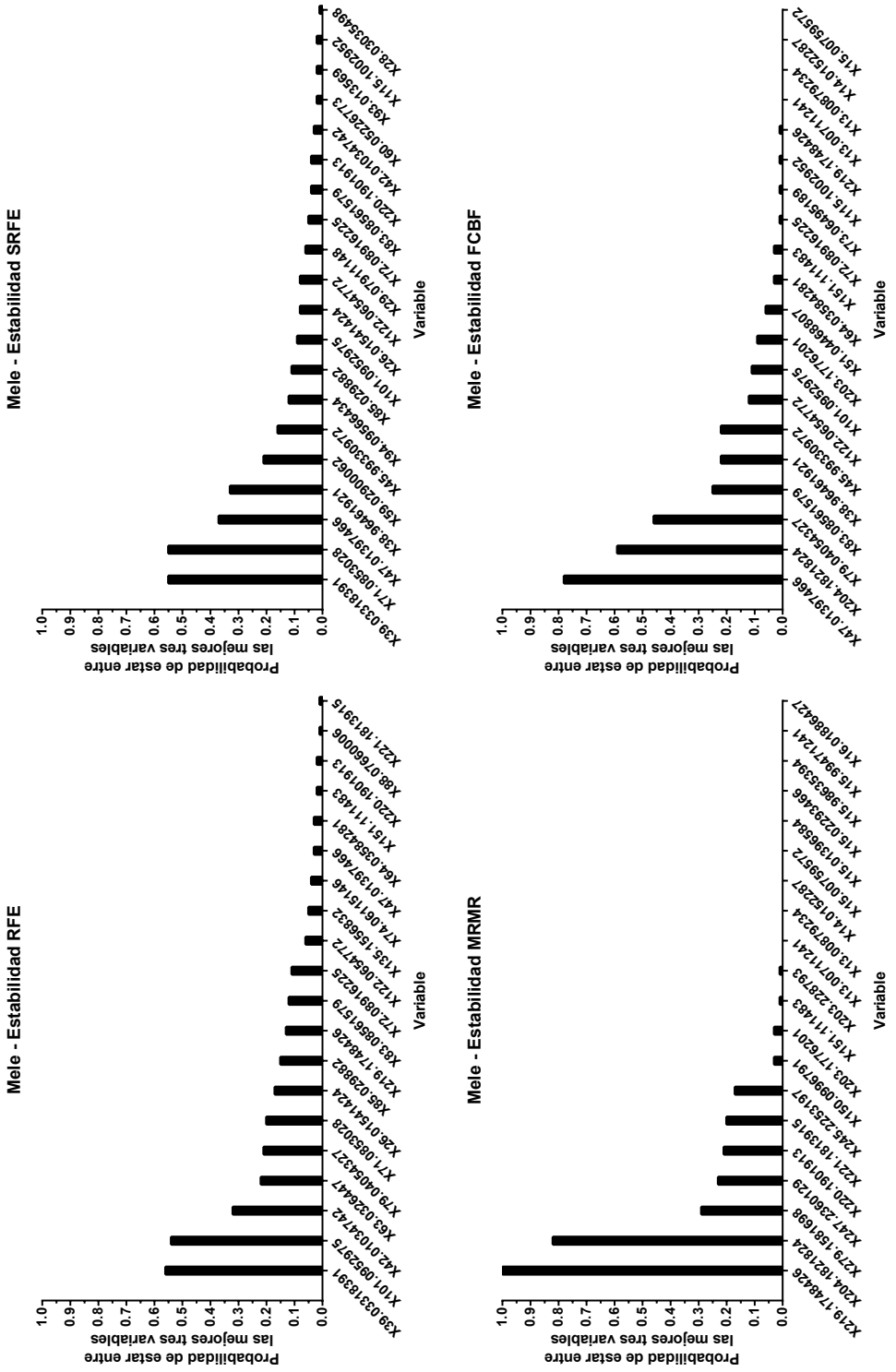


Figura 16: Mele: estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras tres.

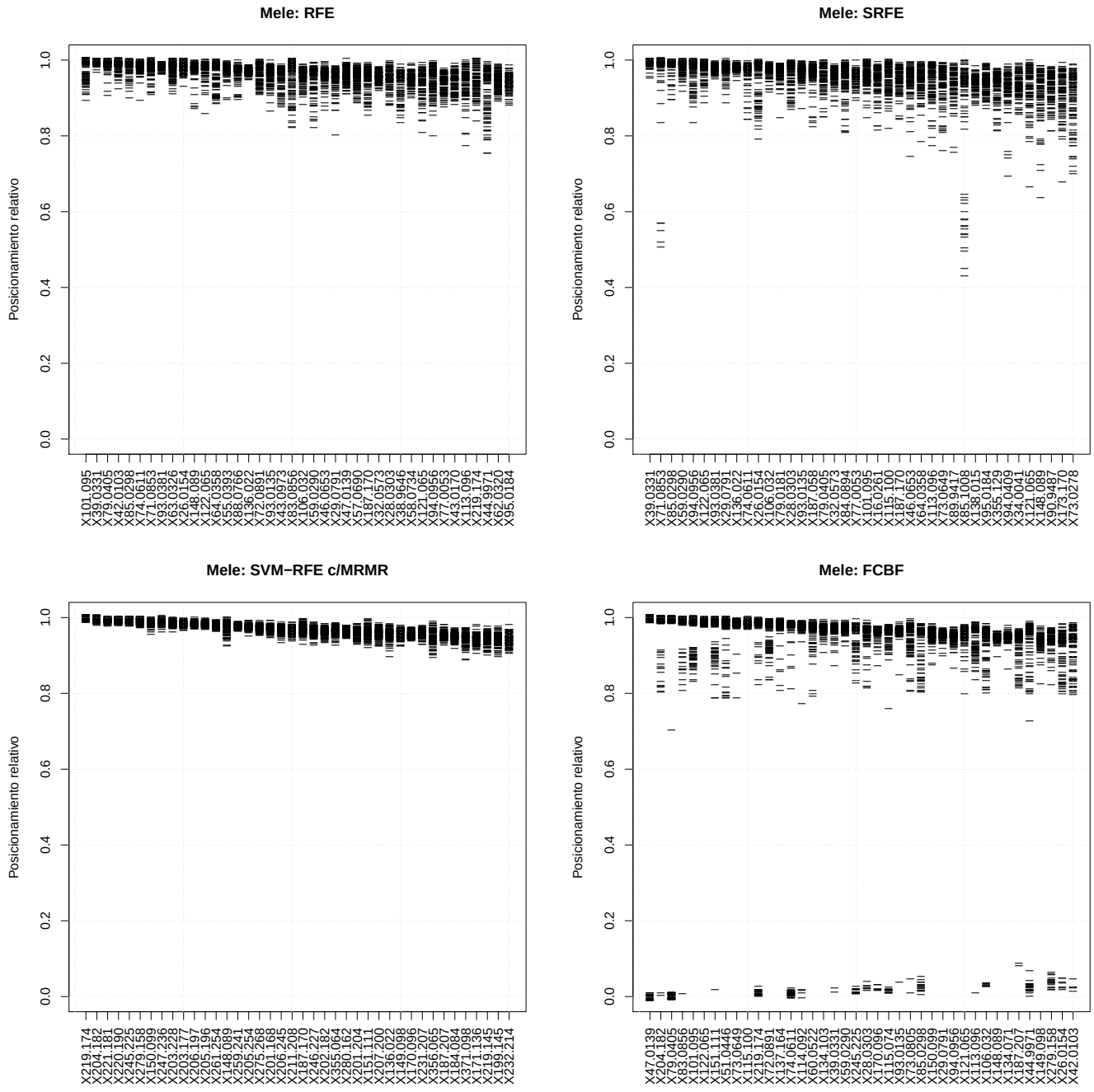


Figura 17: Mele: distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

Fragola

Como en Mele, se midieron variedades de frutillas utilizando el PTR-MS, en este caso con un detector de tipo cuadrupolo de menor resolución. El dataset contiene 231 registros de nueve clases de frutillas medidos sobre 233 variables. El heatmap correspondiente, Figura 18, muestra nuevamente la típica correlación en espectrometría entre pequeños grupos de variables.

Los resultados de este experimento muestran a SRFE y a FCBF como los mejores modeladores con seis y siete variables respectivamente y un error promedio del 10 %, como se muestra en la Figura 19.

En cuestión de estabilidad, lo ancho del problema hace que el resultado no sea tan definitivo como en los casos artificiales y de colorimetría (que se muestran a continuación). Aun así, se destaca que SRFE y MRMR lideran, seguidos por RFE y finalmente FCBF (Figura 20).

El análisis de correlación en las selecciones, considerando los resultados de las Figuras 18 y 20, muestra que tanto RFE como MRMR eligen variables idénticas (isótopos) como las masas 103-104, y 131-132; mientras que SRFE y FCBF reconocen y eliminan dichas redundancias. SRFE logra, en este caso, el mejor rendimiento ya que obtuvo variables independientes.

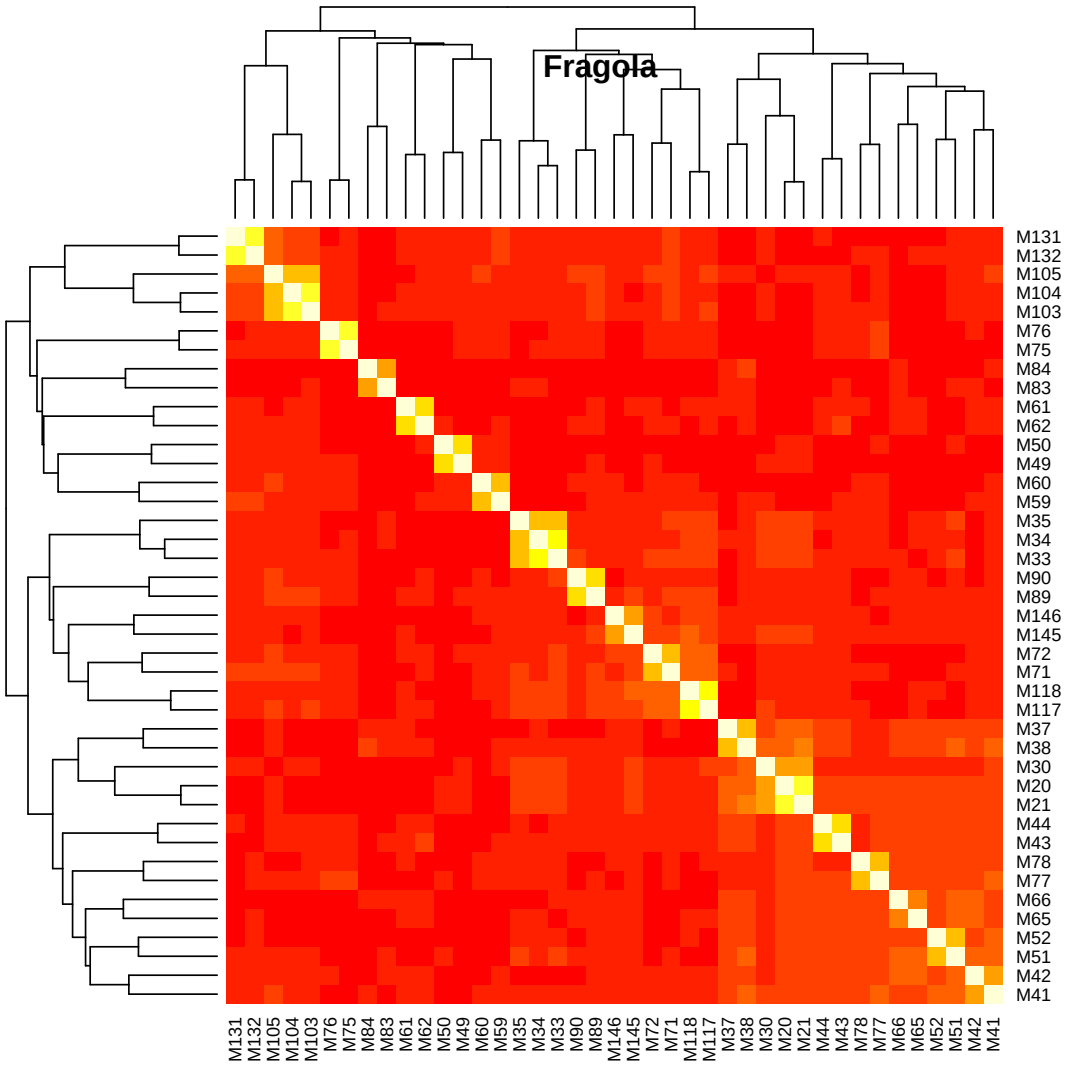


Figura 18: Fragola: mapa de calor que muestra la correlación entre las variables del dataset (sólo un subconjunto del total). Como en Mele, se pueden distinguir varios grupos de dos variables, generalmente isótopos del mismo compuesto, y algunos grupos de mayor tamaño.

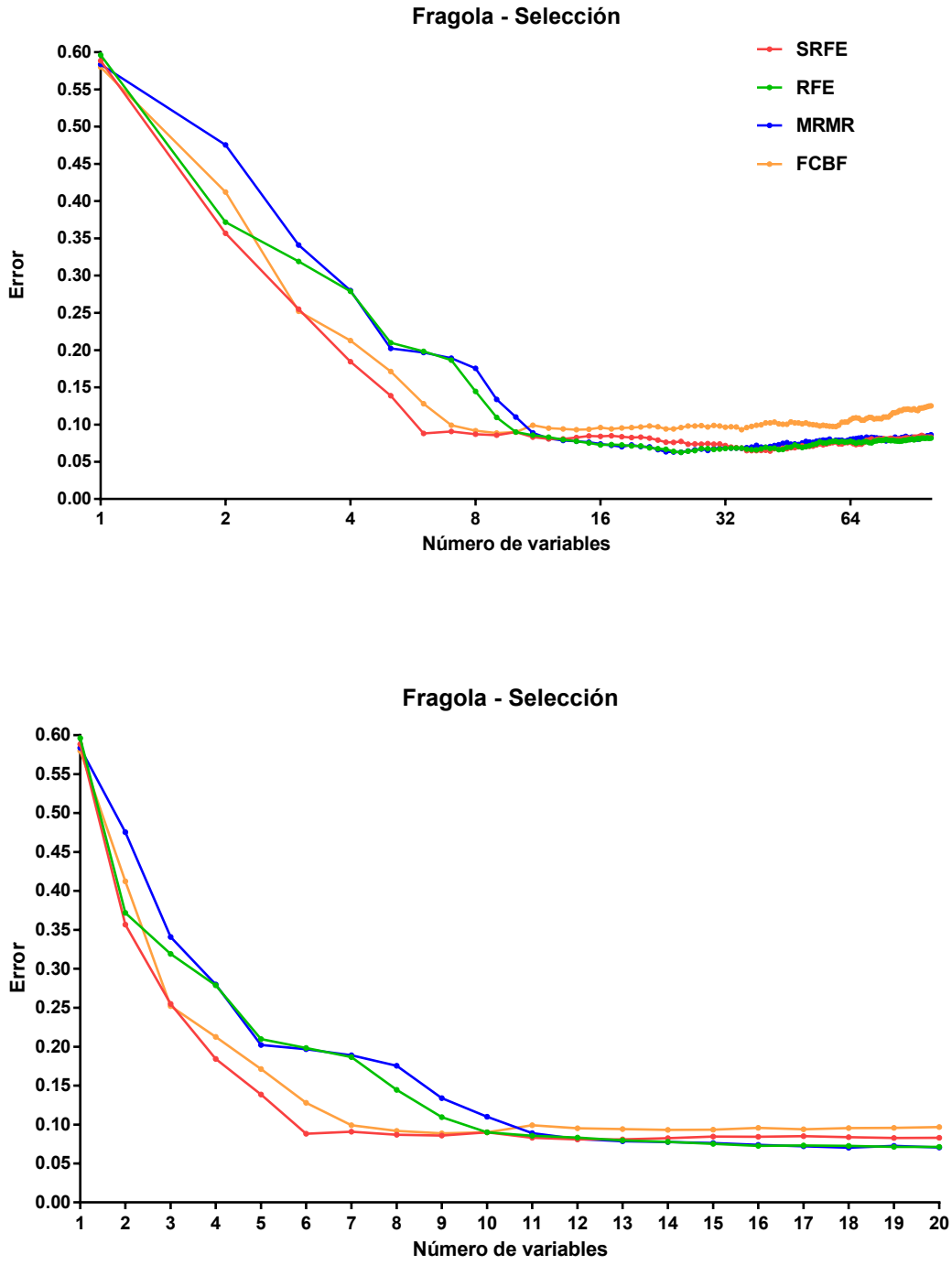


Figura 19: Fragola: nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

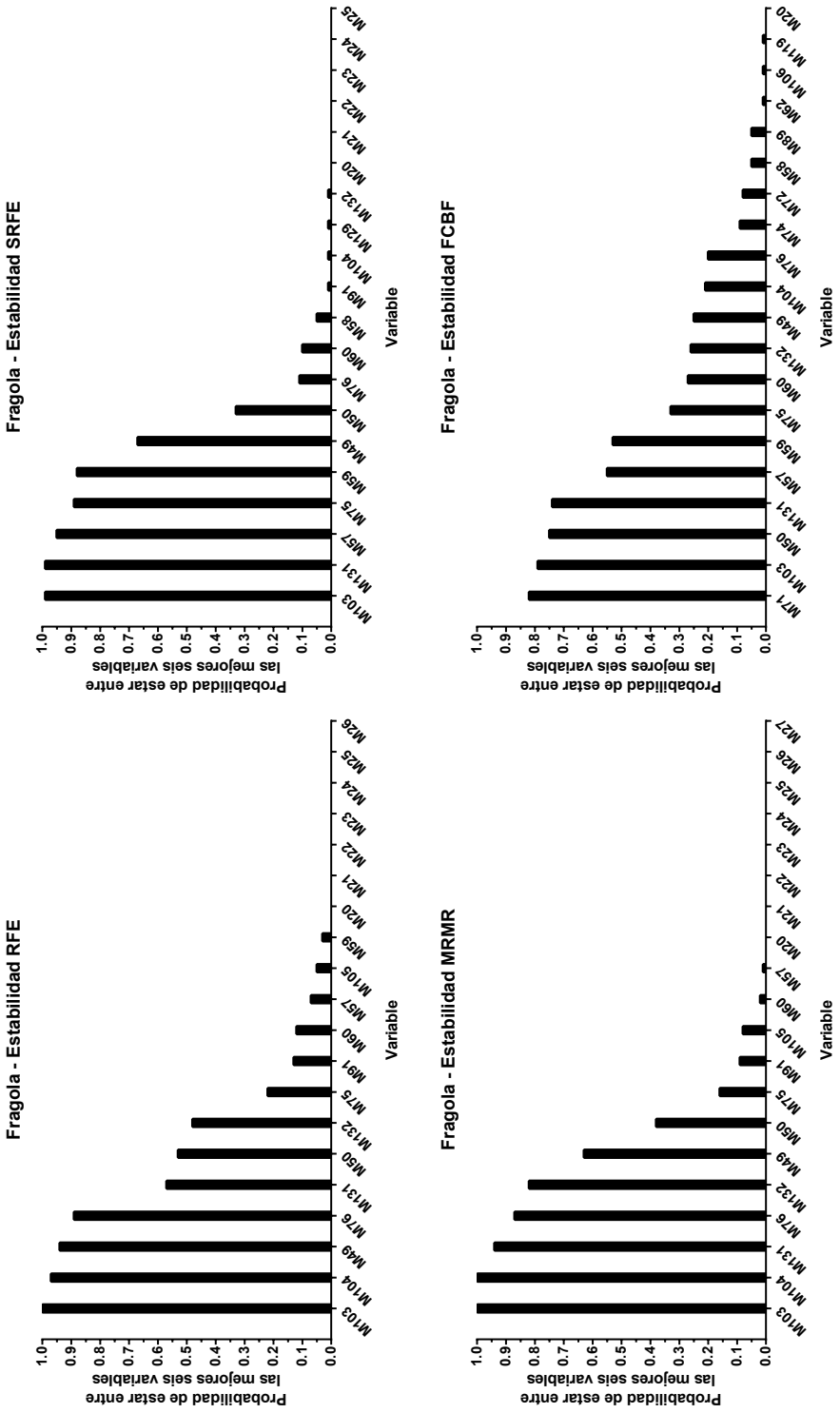


Figura 20: Fragola: estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras seis.

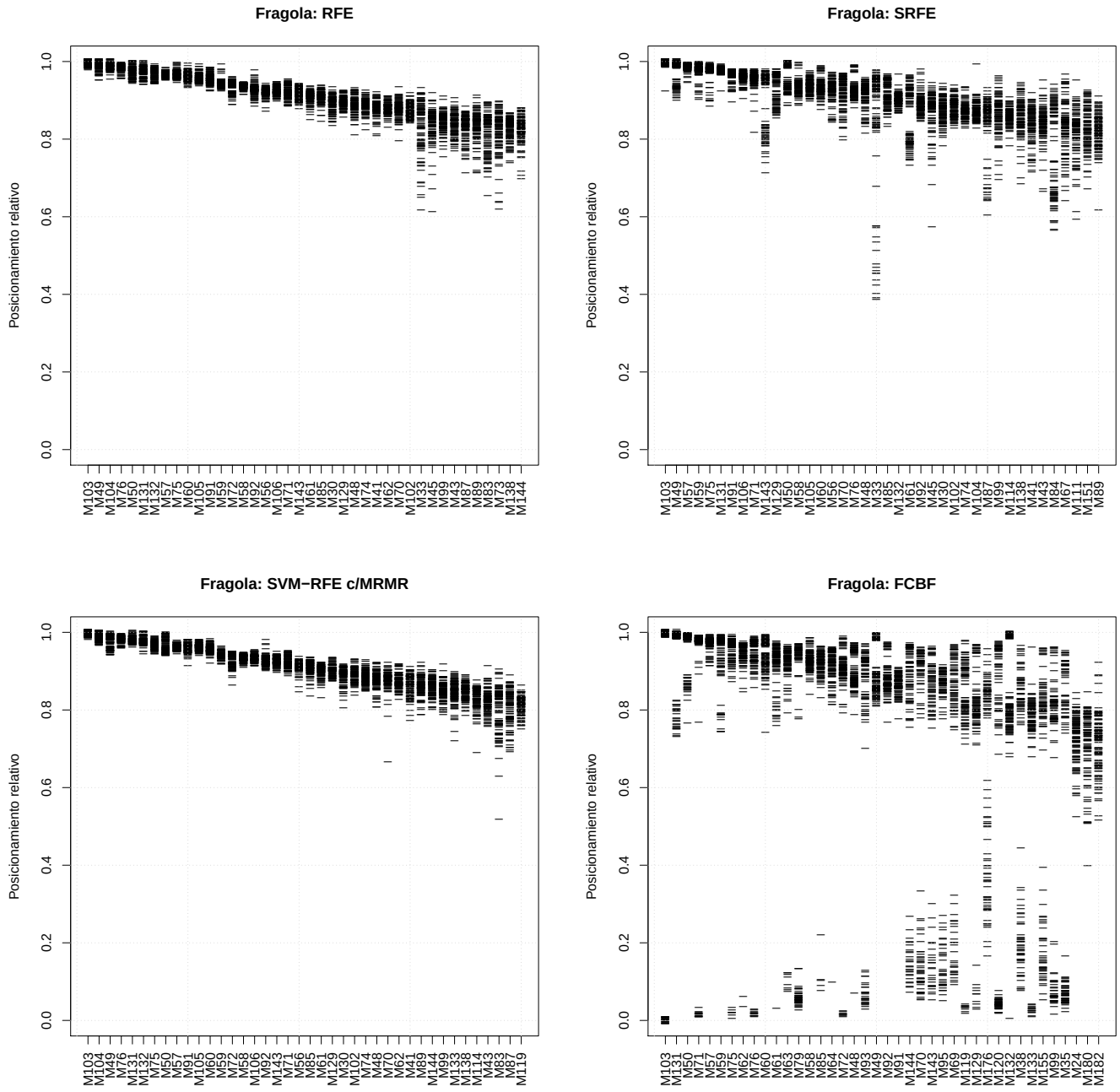


Figura 21: Fragola: distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

4.2.2. Colorimetría

Estos datasets contienen mediciones realizadas en tiras reactivas para análisis de orina en laboratorios clínicos. Las variables corresponden a distintas mediciones que caracterizan el color del patch reactivo bajo análisis, incluyendo distintas iluminaciones y sistemas de color para la lectura. Si bien no son problemas anchos, se sabe de antemano que contienen numerosas variables correlacionadas y pueden aportar a las conclusiones sobre la efectividad del método SRFE.

BIL

Este dataset corresponde a mediciones de bilirrubina en 460 muestras de orina, discriminadas en cuatro clases, con 36 variables medidas. La Figura 22 muestra el heatmap correspondiente, donde se aprecian varios grupos de variables con una alta correlación.

Los resultados muestran que SRFE es el método que genera el mejor modelo de datos con una cantidad óptima de cuatro variables frente a cinco y siete de MRMR y RFE, respectivamente, con menor error en general (Figura 23). FCBF falla en la obtención de un modelo útil sobre estos datos.

En cuestión de estabilidad (Figuras 24 y 25), SRFE y RFE obtienen las soluciones más estables seguidos muy de cerca por MRMR y, en último lugar, por FCBF. El análisis de correlación de las variables muestra como las selecciones de RFE y MRMR son altamente redundantes y, en consecuencia, de peor calidad que las del método SRFE. Si bien las variables de tipo "blanco" tienen un cierto nivel de correlación, esto es debido a que la luz blanca contiene a todos los colores y, es por ende, un resultado normal (de hecho, el resultado óptimo elegido en la experiencia real en los laboratorios). De todas maneras, la redundancia es notablemente menor que en las selecciones de "rojo" de los otros dos métodos, donde las variables son idénticas.

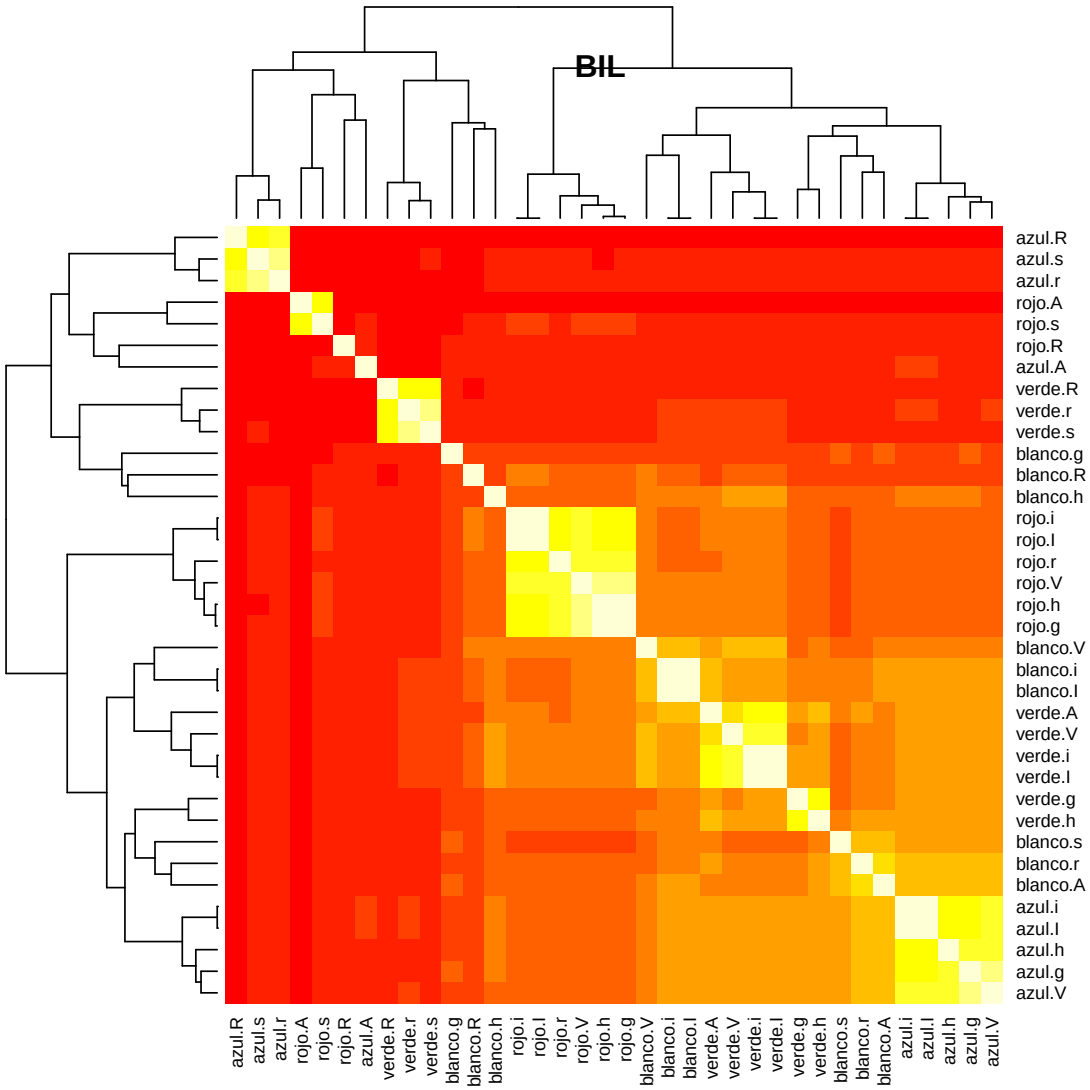


Figura 22: BIL: mapa de calor que muestra la correlación entre las variables del dataset. Se pueden distinguir varios grupos de distinto tamaño con una alta correlación interna.

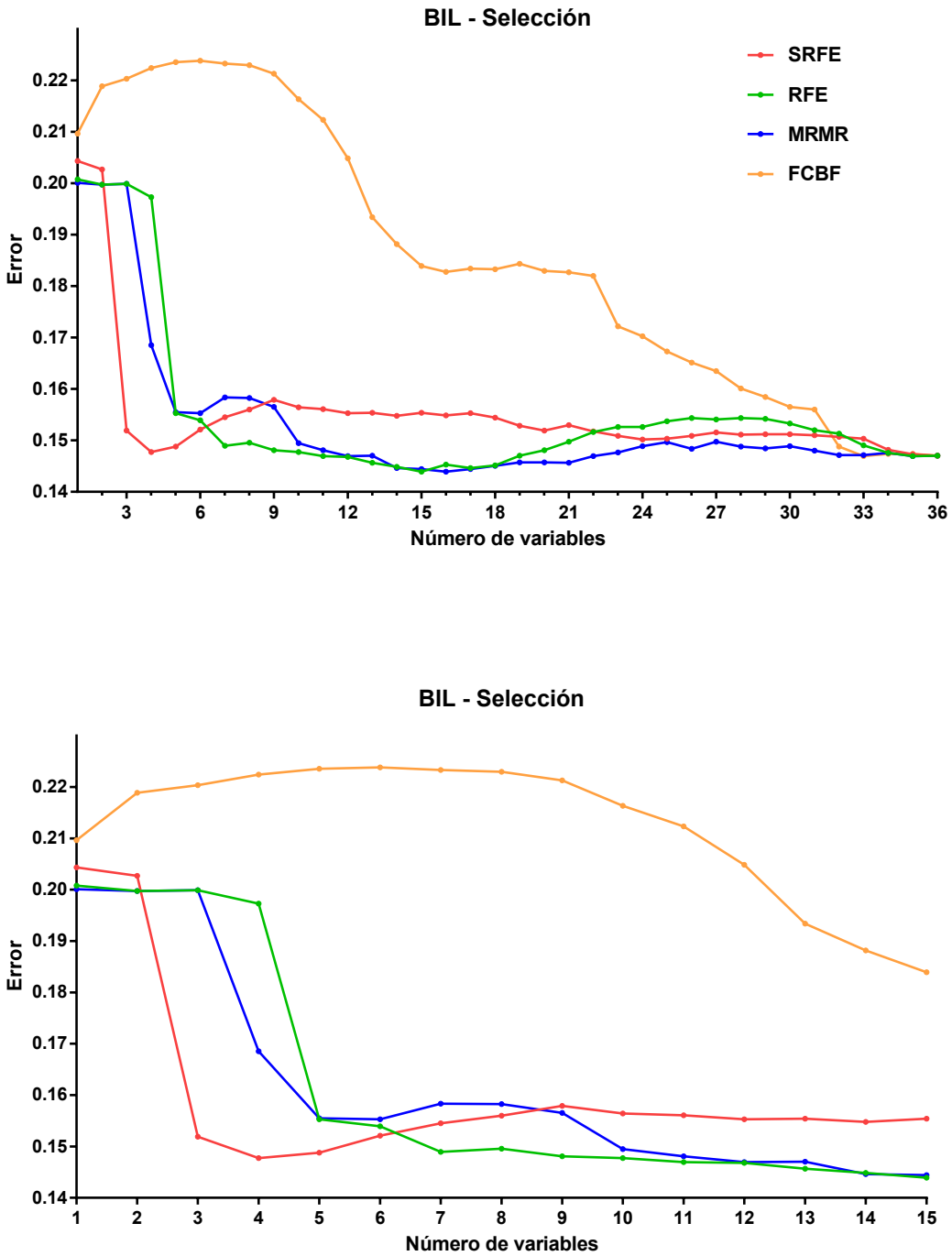


Figura 23: BIL: nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

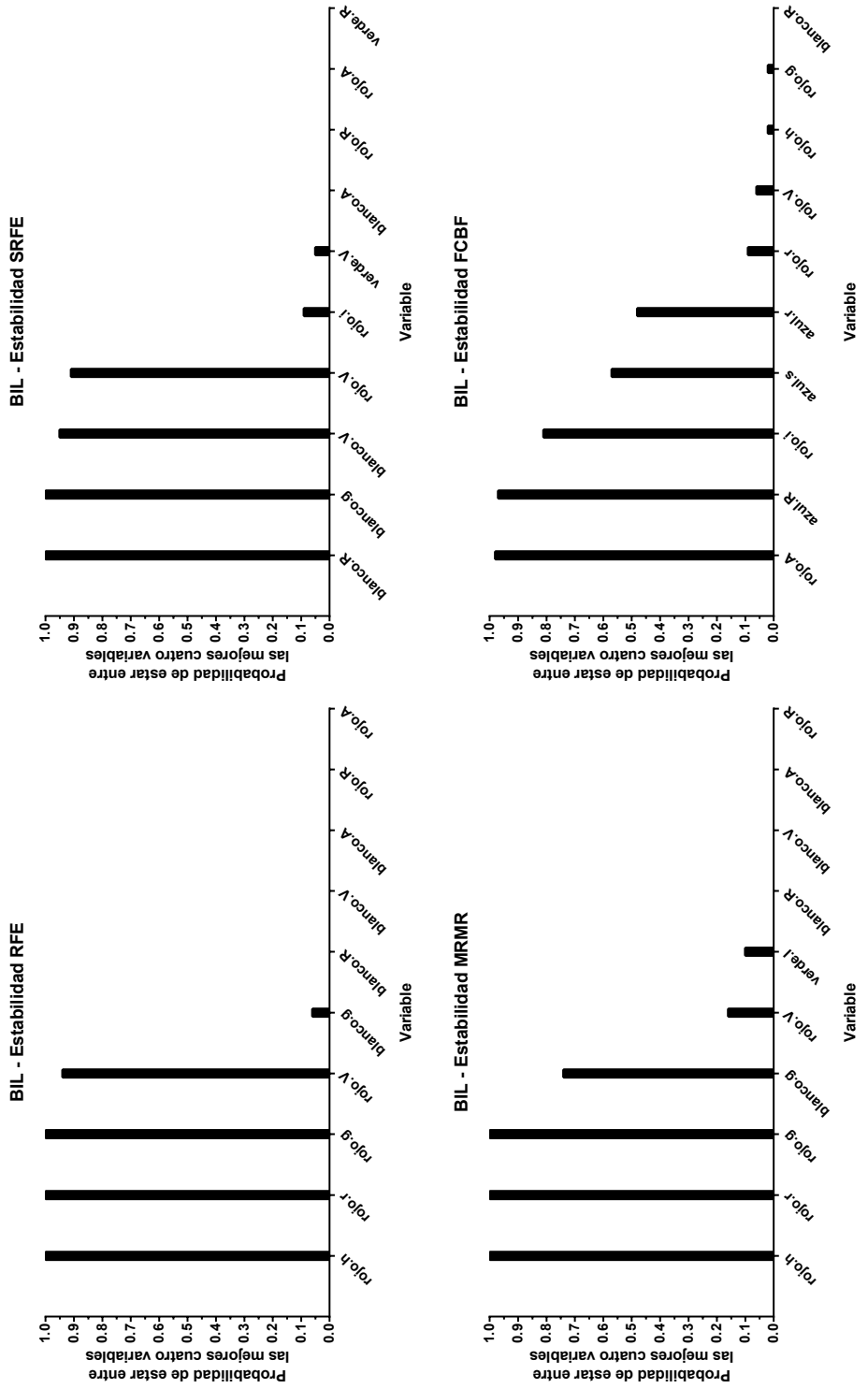


Figura 24: BIL: estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras cuatro.

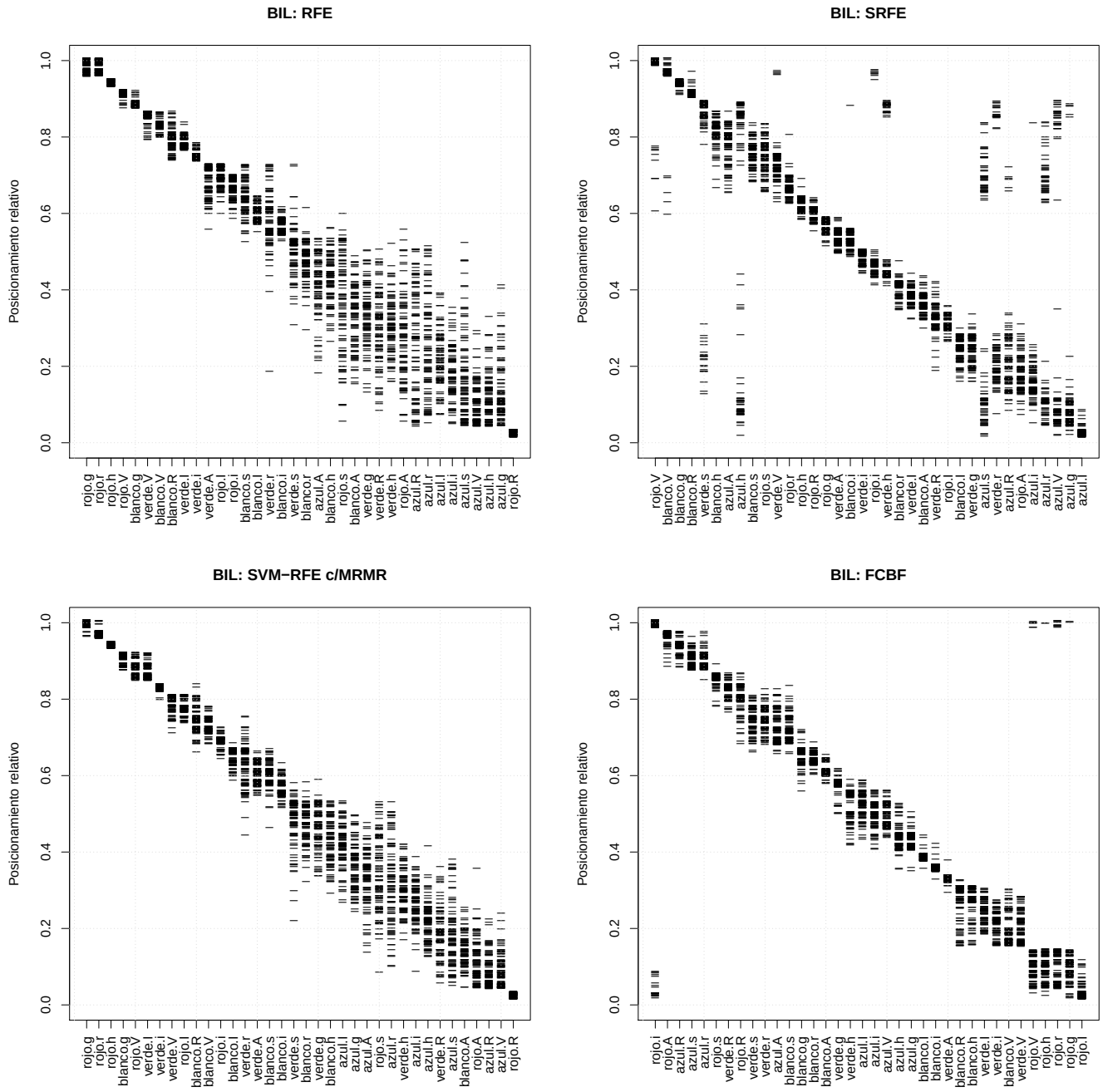


Figura 25: BIL: distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

Ph

Este dataset corresponde a determinaciones del Ph de la orina en tiras reactivas. Hay 440 muestras de 36 variables y seis clases. La Figura 26 muestra las altas correlaciones entre las variables.

Los resultados de este experimento, Figura 27, muestran modelos óptimos para los *wrappers* del mismo nivel de error, si bien SRFE lo logra con dos variables menos (tres en lugar de cinco). FCBF falla en la obtención de un modelo de datos comparable con resto.

En cuestión de estabilidad, la selección es clara y estable para los *wrappers*, pero el análisis de correlación revela gran redundancia en los modelos de RFE y, en particular, de MRMR (Figuras 28 y 29).

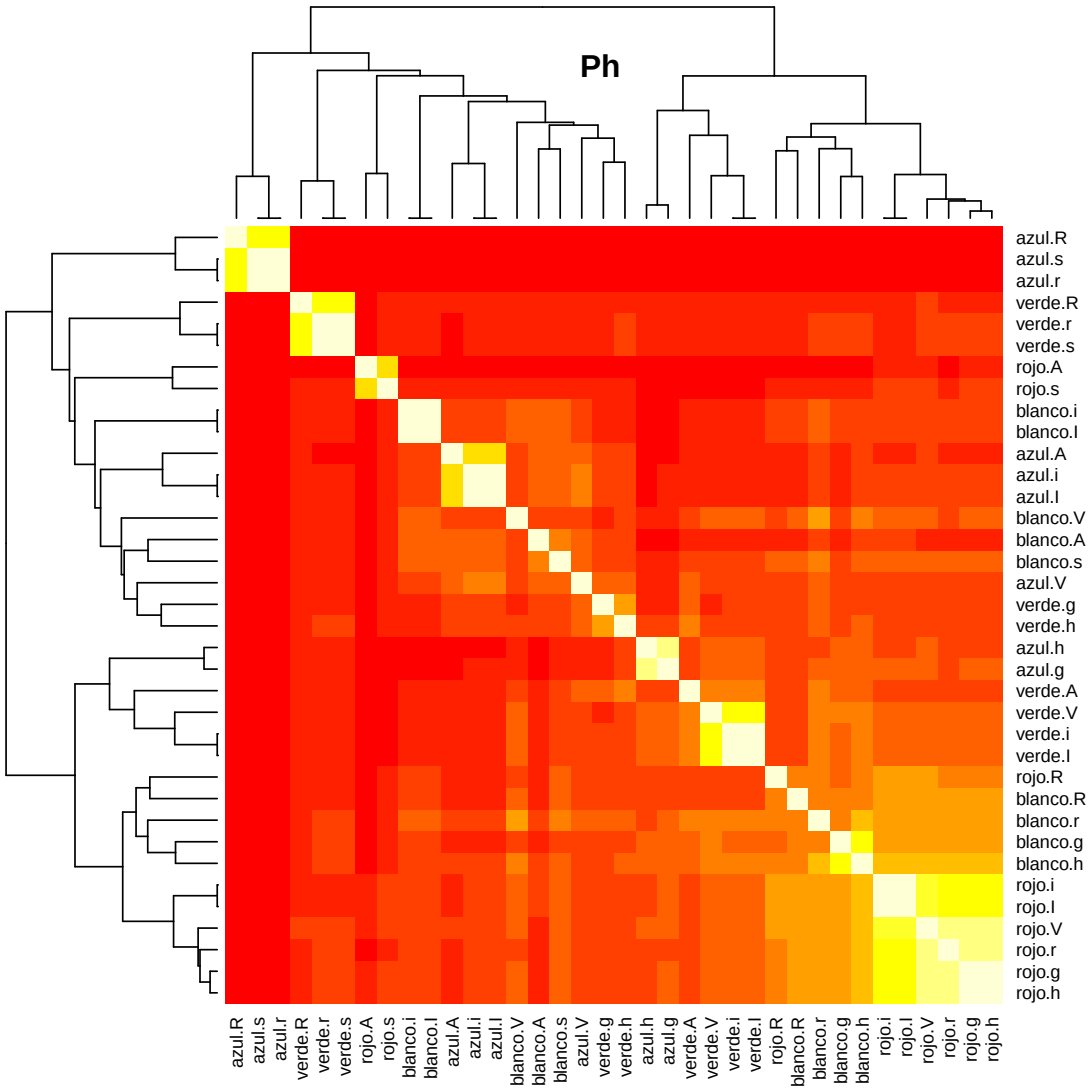


Figura 26: Ph: mapa de calor que muestra la correlación entre las variables del dataset. Se pueden distinguir varios grupos de distinto tamaño con una alta correlación interna.

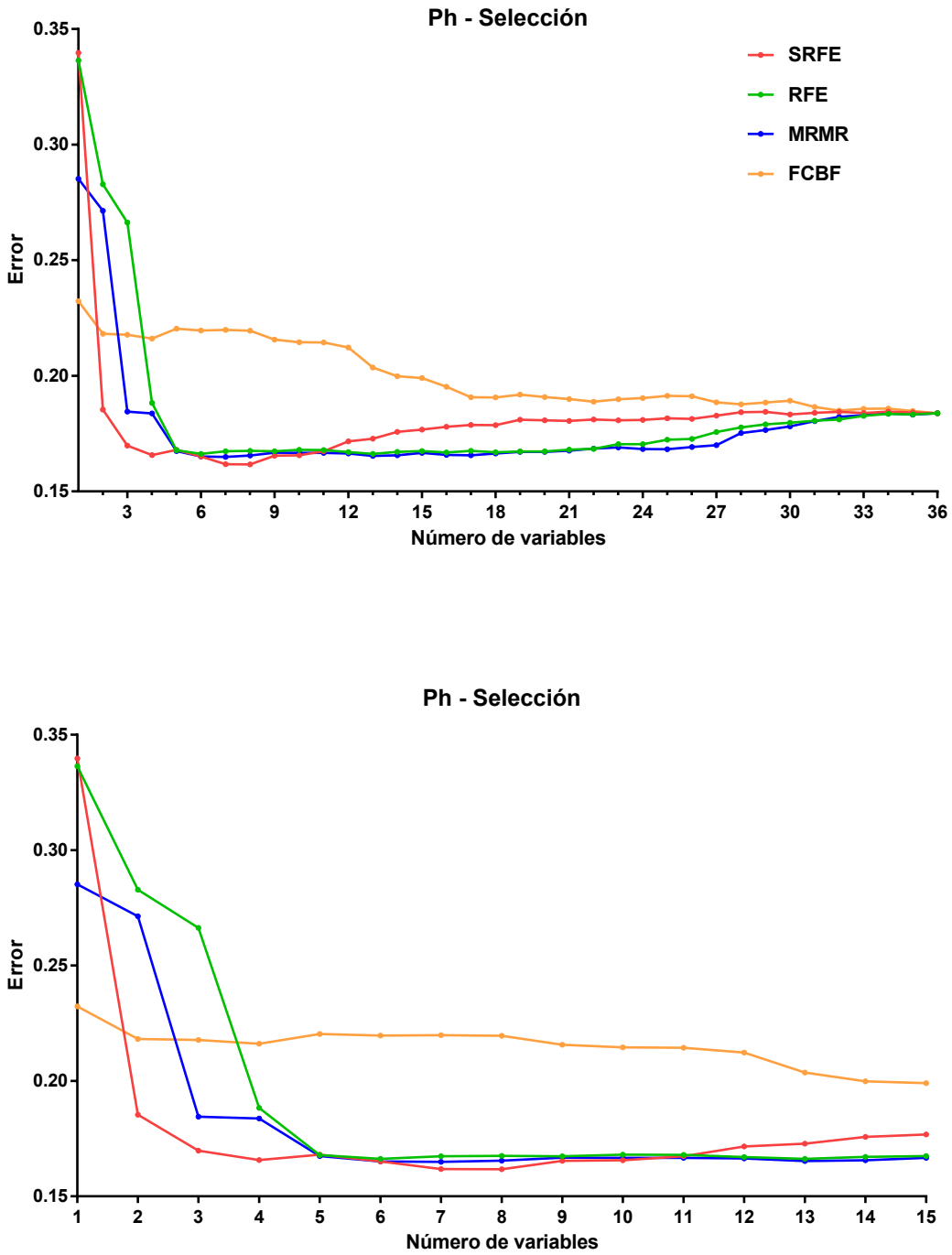


Figura 27: Ph: nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

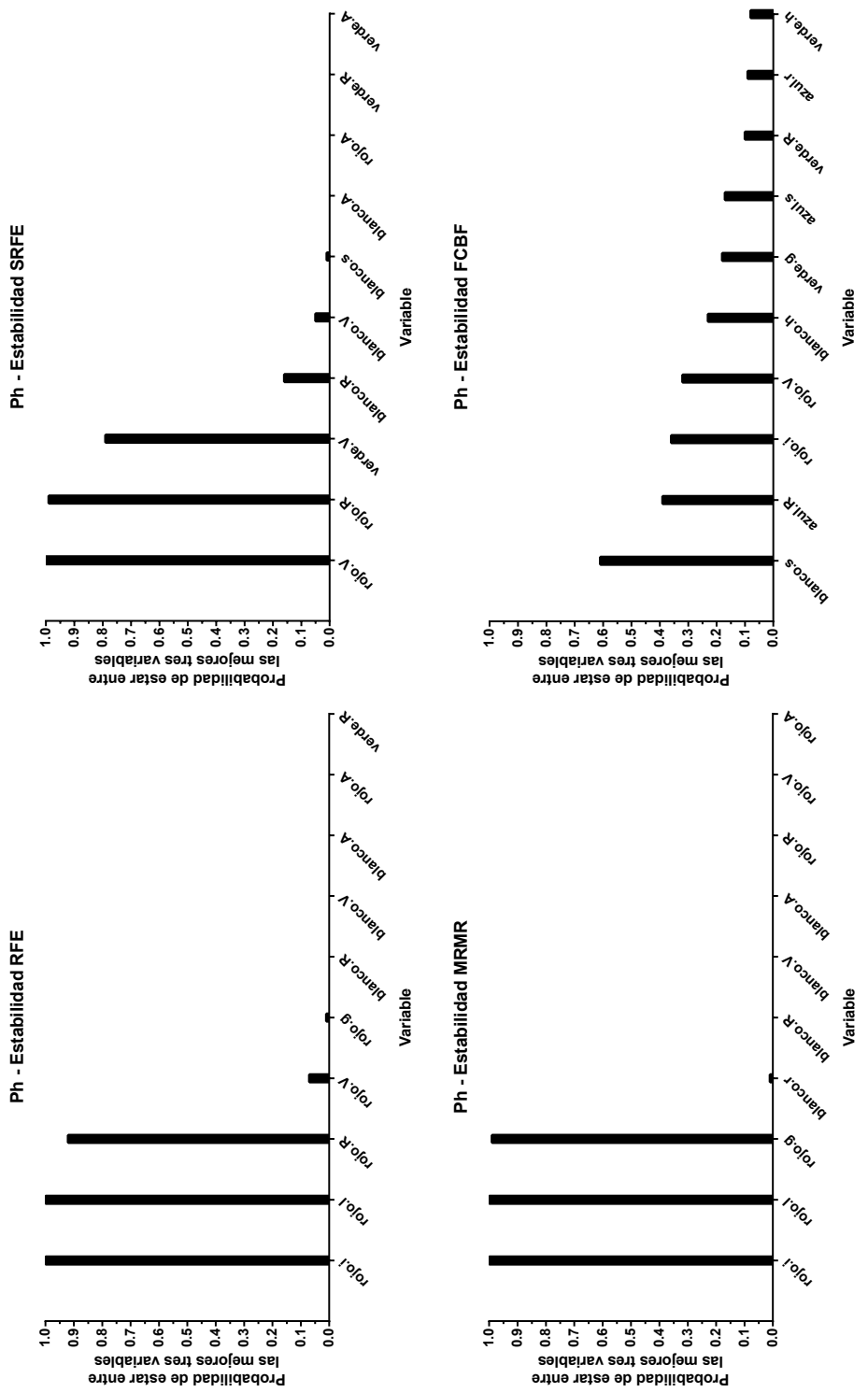


Figura 28: Ph: estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras tres.

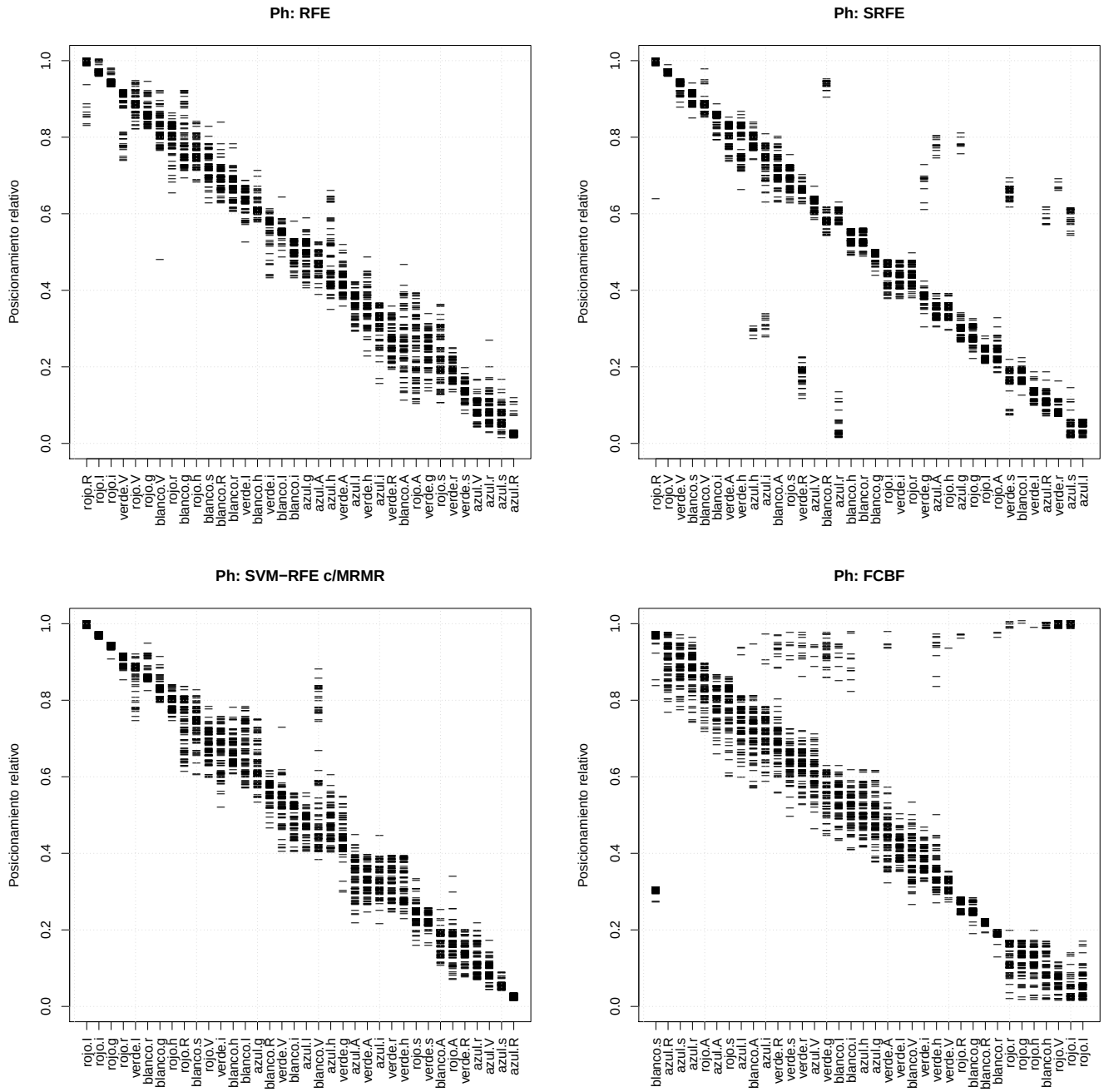


Figura 29: Ph: distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

SG

Este dataset corresponde a determinaciones de la densidad (SG viene de *Specific Gravity*) de la orina en tiras reactivas. Hay 521 muestras de 36 variables y siete clases. Las altas correlaciones, esperables en este tipo de datos, se muestran en la Figura 30.

Este experimento tiene un resultado particular. Si bien es cierto que el método de MRMR, seguido por RFE, puede generar un modelo de datos con un error menor (Figura 31), se requiere un tercio de las variables originales para una mejora del 4 % en un resultado del orden del 30 % de error. Dada la naturaleza de la solución[2], y siendo estos experimentos originalmente casos de *screening*, se decidió utilizar un modelo dado por tres variables, que corresponde a un mínimo local para dos métodos.

En cuestión de estabilidad, como se muestra en las Figuras 32 y 33, sólo el método de filtro muestra una mala selección. RFE es muy estable, aunque dos de las variables seleccionadas, $rojo.R$ y $rojo.I$, son casi idénticas en términos de correlación. SRFE ofrece una solución de menor error que RFE y MRMR, aunque la selección en este caso está bien definida pero es menos estable que la que aporta MRMR.

[2]Los resultados de SG en las tiras reactivas se encuentran en el intervalo de colores comprendido entre el azul y el amarillo. Como el amarillo está formado por las luces roja y verde, se puede precisar el color resultante midiendo cualquiera de los canales disponibles.

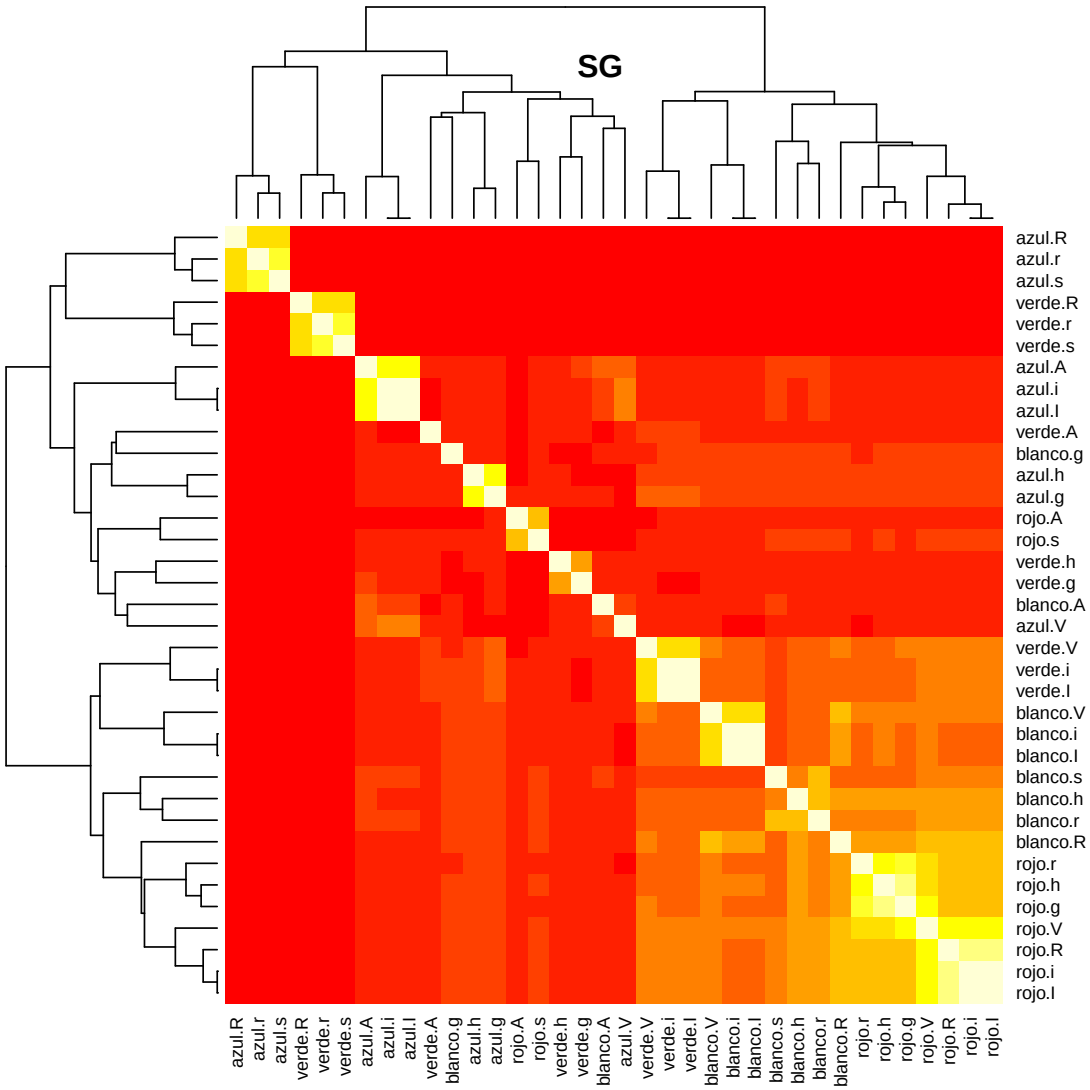


Figura 30: SG: mapa de calor que muestra la correlación entre las variables del dataset. Se pueden distinguir varios grupos de distinto tamaño con una alta correlación interna.

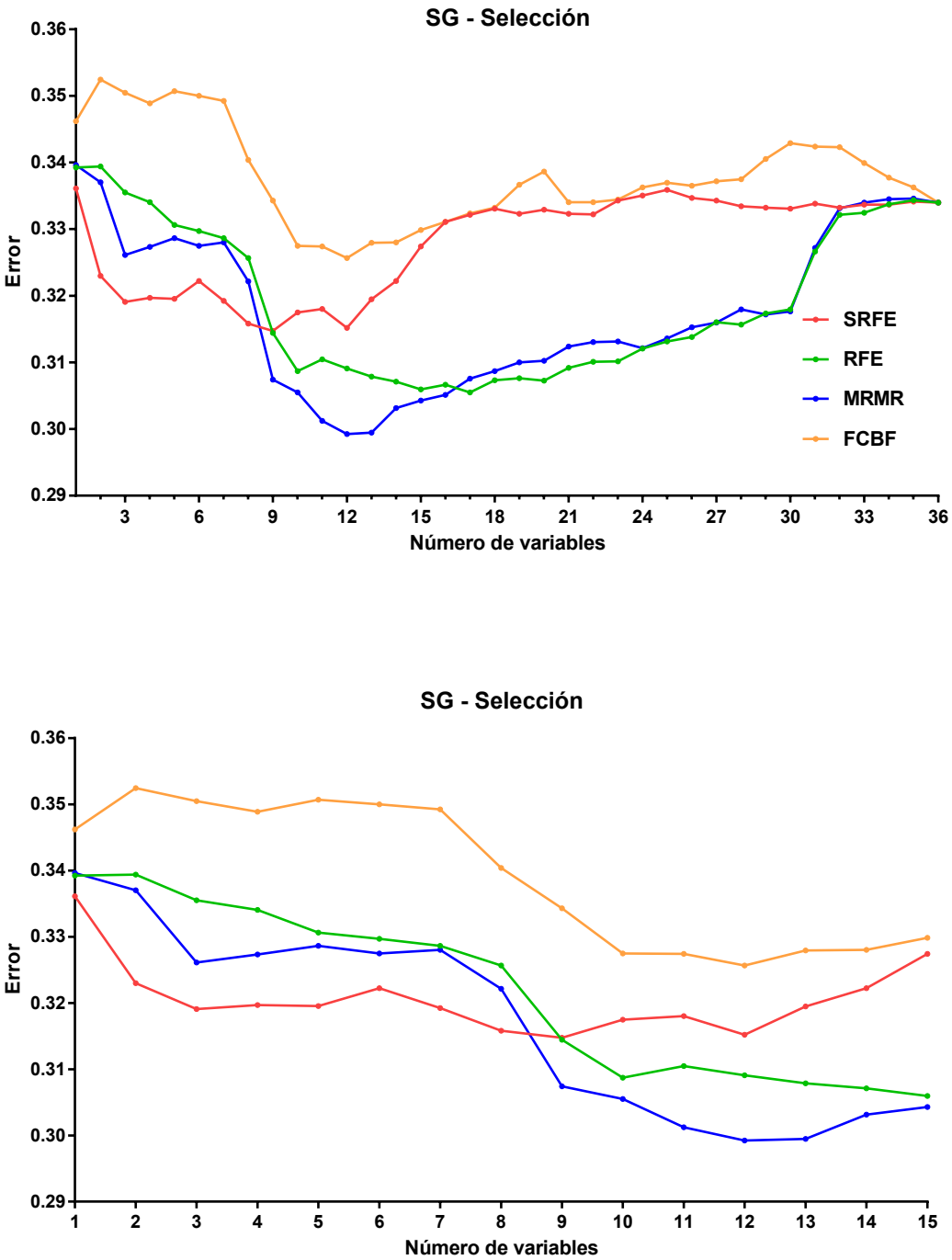


Figura 31: SG: nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

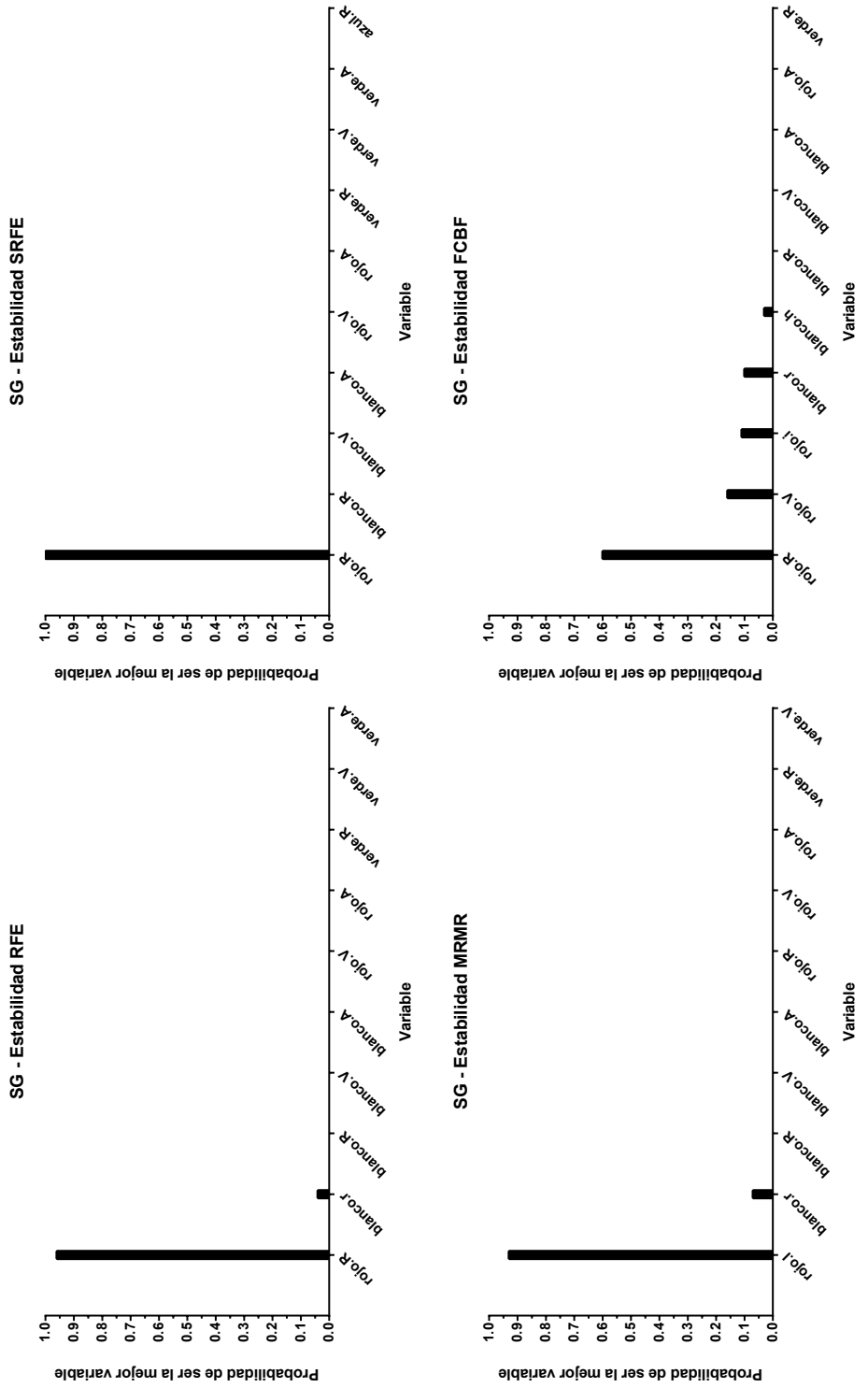


Figura 32: SG: estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras tres.

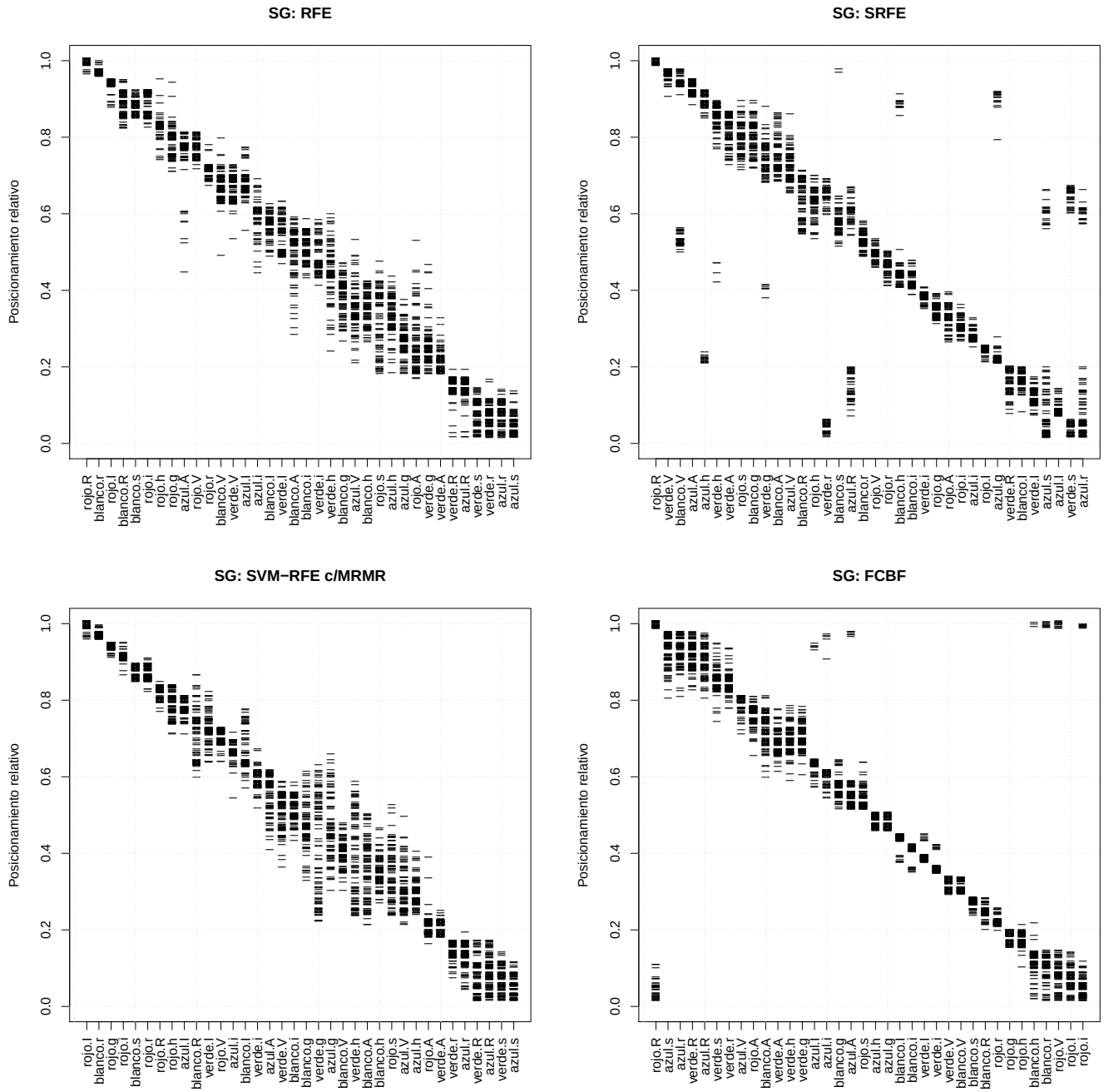


Figura 33: SG: distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

URO

El último dataset corresponde a determinaciones del urobilinógeno de la orina en tiras reactivas. Hay 465 muestras de 36 variables y seis clases. Como siempre en estos datos, las altas correlaciones entre variables se pueden ver en el heatmap de la Figura 34.

Los resultados de este experimento muestran a SRFE como el único método capaz de generar un buen modelo a partir de los datos (Figura 35). Con una cantidad óptima de cuatro variables obtiene un error promedio del 10 %, mientras que los demás métodos necesitan de al menos quince variables para acercarse a ese valor.

En cuestión de estabilidad, nuevamente es SRFE el método con mayor estabilidad en la selección. RFE y MRMR comienzan a confundir variables en el cuarto y tercer puesto, respectivamente (Figura 36). Tras el análisis de correlación, se observa la independencia de las variables elegidas por SRFE, mientras que el resto de los métodos elige todas variables del mismo cluster de tipo "verde".

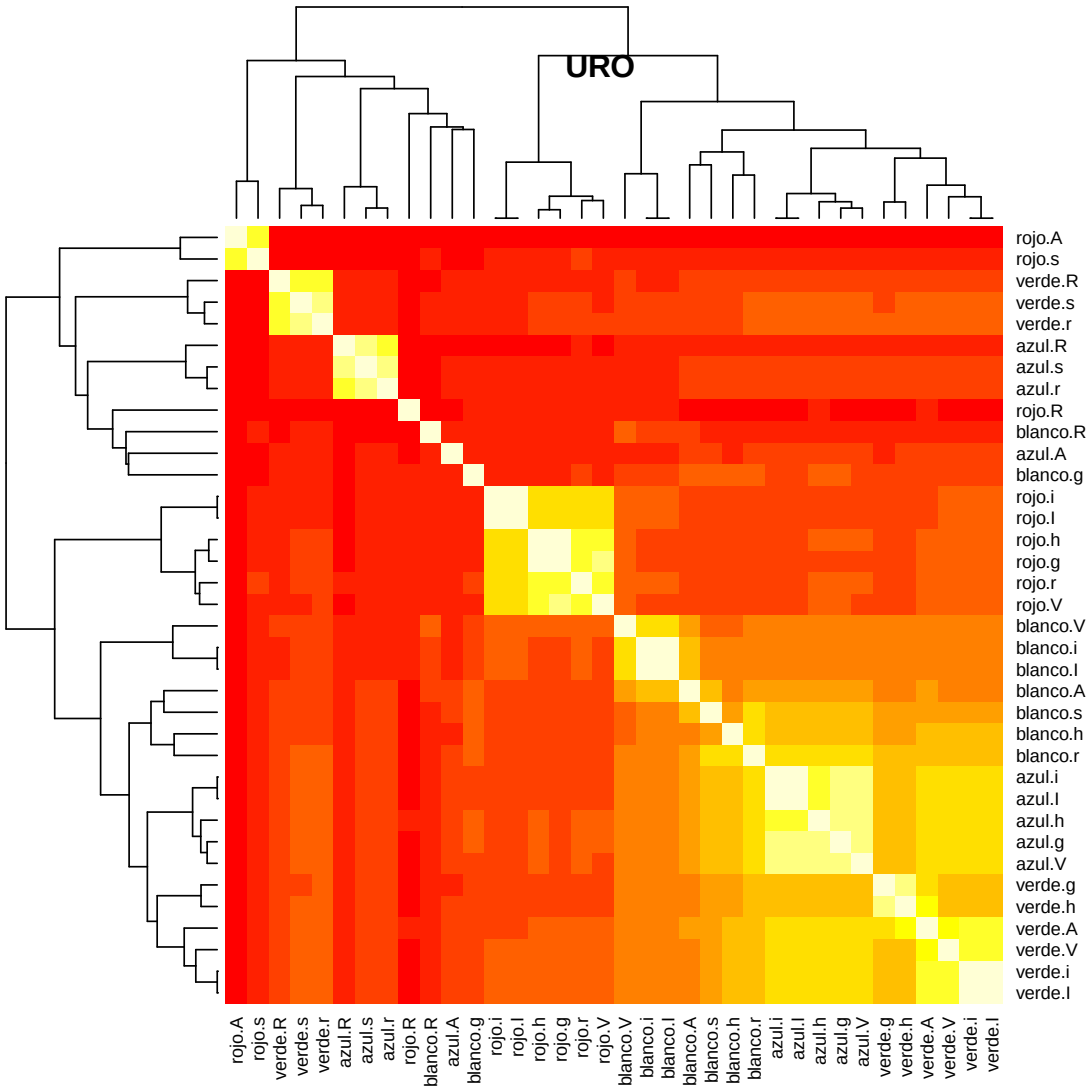
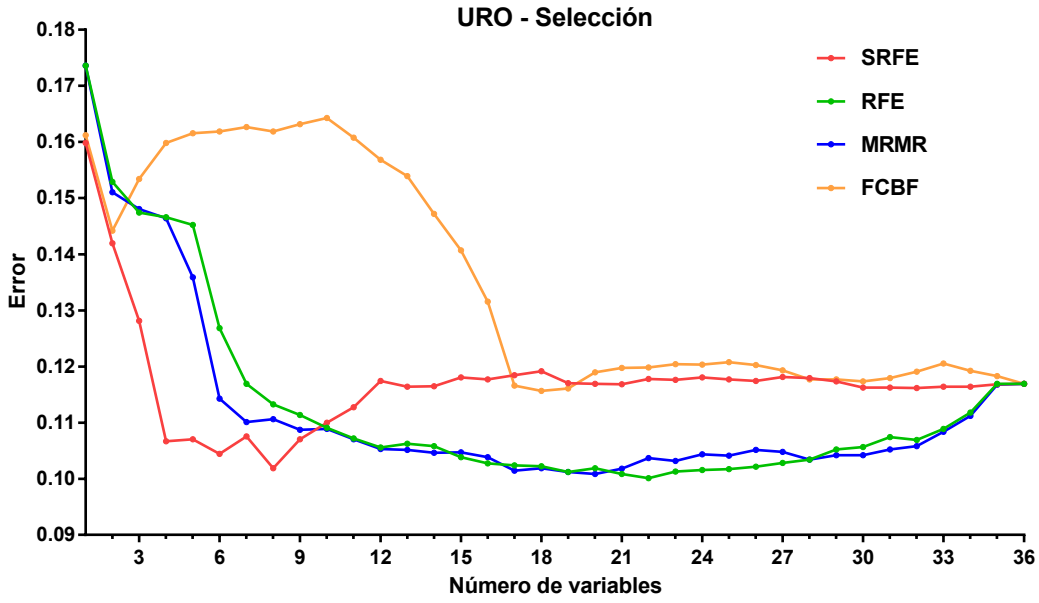


Figura 34: URO: mapa de calor que muestra la correlación entre las variables del dataset. Se pueden distinguir varios grupos de distinto tamaño con una alta correlación interna.



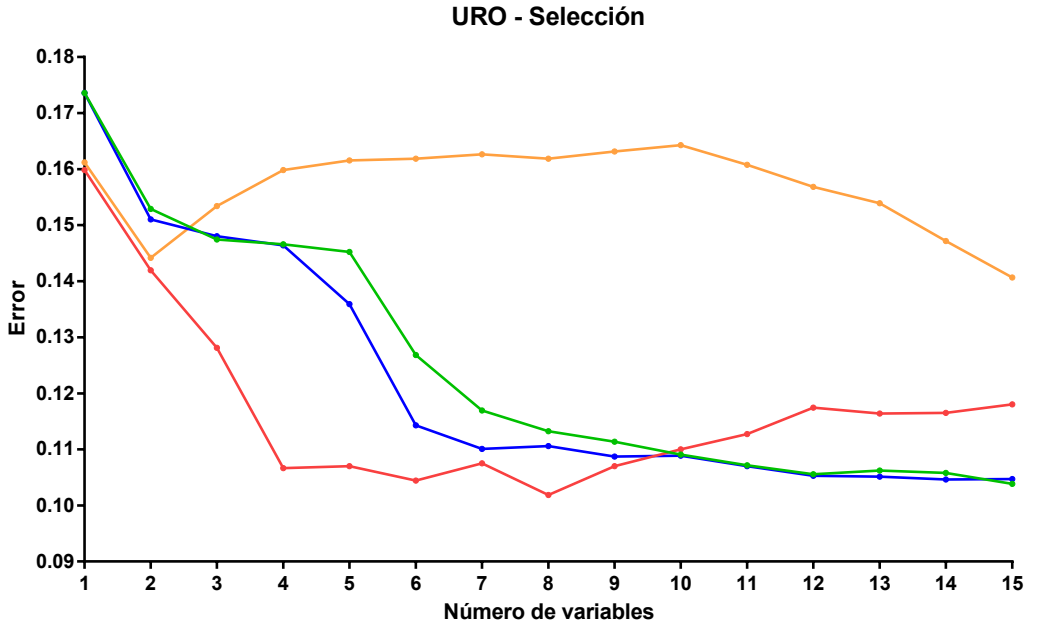


Figura 35: URO: nivel de error en función del número de variables seleccionadas. El panel superior muestra la figura completa mientras que el inferior muestra en detalle el comienzo de la figura.

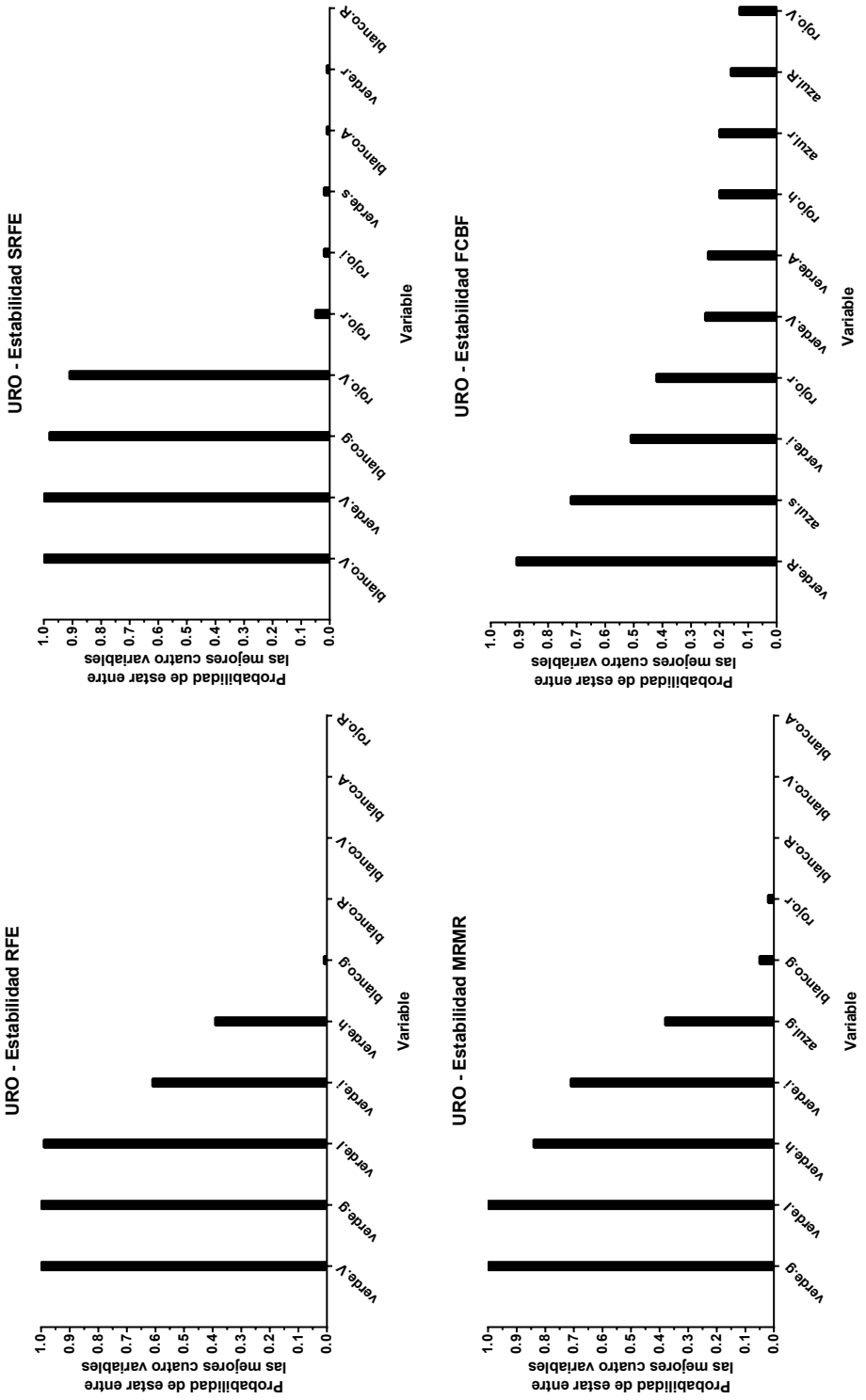


Figura 36: URO: estabilidad de las soluciones. Para cada método se muestra la probabilidad de cada de variable de estar seleccionada entre las primeras cuatro.

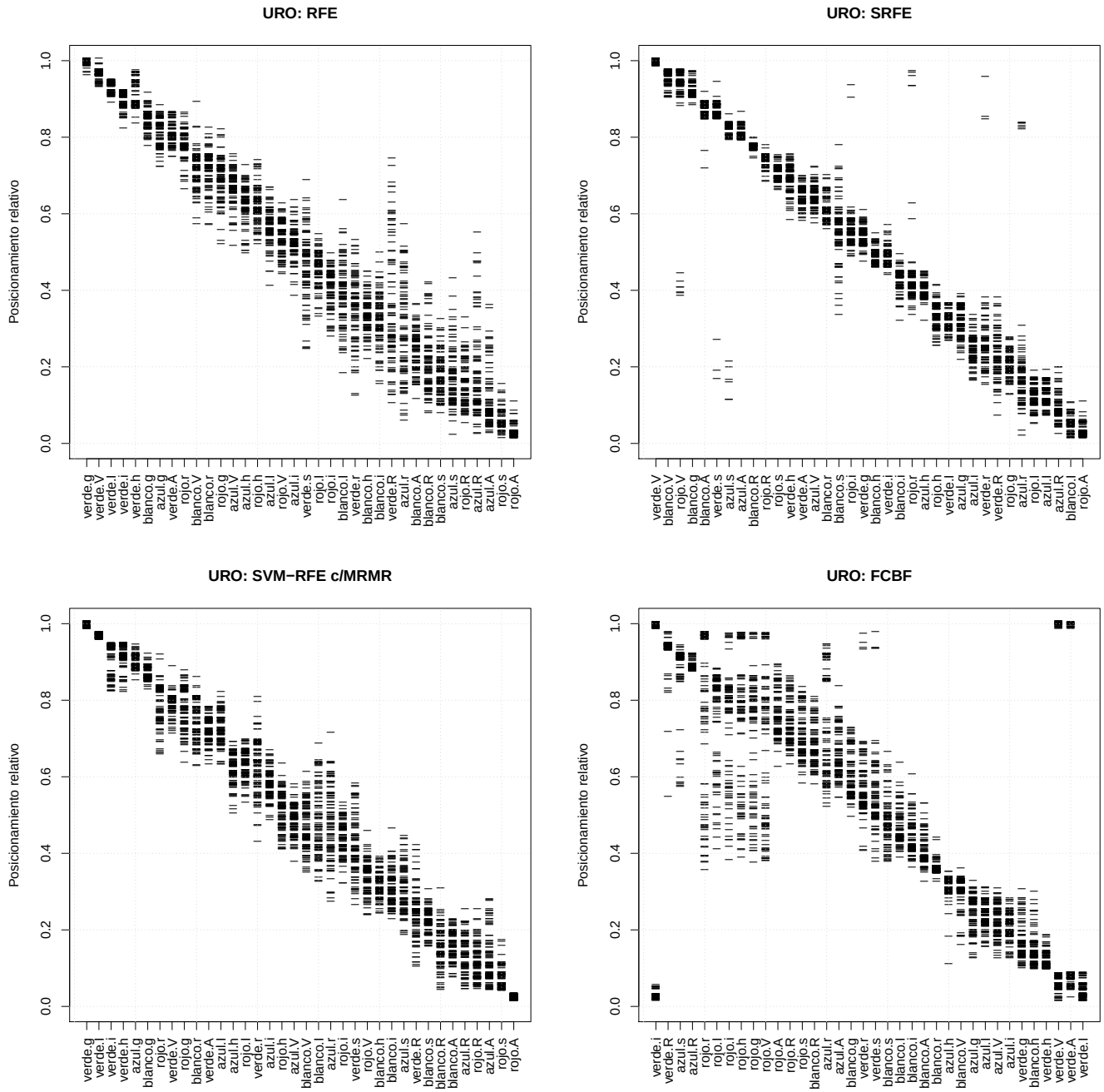


Figura 37: URO: Distribución de posiciones que toma cada variable en el ranking realizado por cada uno de los métodos comparados.

5. Conclusiones y trabajos futuros

En esta tesina se diseñó, desarrolló e implementó el método de selección de variables SRFE, el cual apunta a producir resultados estables y no redundantes. El método combina el ranking producido por el eficiente algoritmo RFE con una penalización que se apoya en la información mutua entre las variables. El nuevo método tiene en cuenta el problema de la estabilidad de las soluciones ante experimentos repetidos y trata de resolver el problema eligiendo, siempre que es posible, la misma variable entre un grupo de variables correlacionadas.

Al incorporar una penalización a las variables correlacionadas se esperaba obtener soluciones con un error global mayor que otros métodos que apuntan directamente a buscar el mínimo error de modelado, como RFE. Sin embargo, la hipótesis inicial resultó refutada, ya que en todos los experimentos realizados se destacó SRFE al obtener siempre mejores resultados, tanto a nivel de error como de estabilidad.

En primer término, se mostró con un dataset artificial cómo el nuevo método resuelve de forma eficiente y estable un problema con muchas variables correlacionadas y ruidosas, donde la clase depende de múltiples variables. Dicho problema artificial deja en evidencia las limitaciones de los métodos precedentes.

En segunda instancia, se analizaron datasets reales de problemas de espectrometría de masa y de colorimetría. Los datasets tenían distintas relaciones muestra/variable, pero siempre incorporaban un alto número de variables redundantes. En la gran mayoría de los casos, el método SRFE mostró mayor precisión y estabilidad en relación con los métodos comparados. Esto reafirma lo previamente observado en el caso artificial.

Se puede concluir que el método SRFE produce una selección de variables

independientes y a la vez más eficientes para el modelado, superando en particular la búsqueda exclusiva de la minimización de error del método RFE y a la penalización global que aplica MRMR.

Se plantean como trabajos futuros el análisis para la selección del argumento de balance β, así como de los argumentos de corte T_p y T_c que determinan el tamaño de los clusters de correlación. Es necesario también desarrollar una forma efectiva de graficar los grupos de variables que fueron penalizadas por el algoritmo para ayudarnos a entender sus relaciones.

Referencias

[1] John Robert Anderson, Ryszard Spencer Michalski, Ryszard Stanisław Michalski, Thomas Michael Mitchell, et al. *Machine learning: An artificial intelligence approach*, volume 2. Morgan Kaufmann, 1986.

[2] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[3] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[4] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[5] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[6] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[7] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[8] Piyushkumar A Mundra and Jagath C Rajapakse. Svm-rfe with mrmr filter for gene selection. *NanoBioscience, IEEE Transactions on*, 9(1):31–37, 2010.

[9] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, 2005.

[10] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[11] Bernhard Schölkopf and Alexander J Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond (adaptive computation and machine learning). 2001.

[12] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.