DISEÑO Y CREACIÓN DE UN CLASIFICADOR DE PÉPTIDOS ANTIBACTERIANOS UTILIZANDO TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES Y ALGORITMOS DE APRENDIZAJE

DIEGO FERNANDO COBA CRUZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES

BUCARAMANGA

2016

DISEÑO Y CREACIÓN DE UN CLASIFICADOR DE PÉPTIDOS ANTIBACTERIANOS UTILIZANDO TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES Y ALGORITMOS DE APRENDIZAJE

DIEGO FERNANDO COBA CRUZ

Trabajo de Grado para optar por el título de:

Ingeniero Electrónico

Director:

Daniel Alfonso Sierra Bueno

PhD Biomedical Engineering

Codirectora:

Nydia Paola Rondón Villarreal

MSc. Ingeniería de Sistemas e Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES

BUCARAMANGA

2016

AGRADECIMIENTOS

A mis padres por su apoyo incondicional y por haberlo dado todo para salir adelante sin importar las circunstancias. A mi director de proyecto Daniel Alfonso Sierra Bueno por su colaboración, paciencia y confianza en mí y finalmente a mi codirectora Paola Rondón Villarreal por su motivación, consejos, amistad y ayuda durante este proceso por el que hoy estoy un paso adelante en el camino que me lleva hacia mis metas.

TABLA DE CONTENIDO

pág

LISTA DE FIGURAS

pág

LISTA DE ANEXOS

pág

RESUMEN

TÍTULO:

DISEÑO Y CREACIÓN DE UN CLASIFICADOR DE PÉPTIDOS ANTIBACTERIANOS UTILIZANDO TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES Y ALGORITMOS DE APRENDIZAJE.[1]

AUTOR:

Diego Fernando Coba Cruz[2]

PALABRAS CLAVE:

Clasificador, péptido, Máquinas de soporte vectorial, Validación Cruzada, KNN, Potencial de interacción Ion Electrón, Señal discreta, Representación en frecuencia.

DESCRIPCIÓN:

Este informe presenta la descripción del diseño y la obtención de un clasificador de péptidos antimicrobianos que permite realizar una preselección de cadenas peptídicas, de manera que se reduzca el número de experimentos necesarios para encontrar alguna con actividad antibacteriana.

Se hace uso de métodos de aprendizaje supervisado para que el sistema *aprenda* a diferenciar una cadena con la propiedad deseada de una que no la posee. Se implementan los métodos *Máquinas de Soporte Vectorial* con kernel tanto lineal como de base radial y el algoritmo de *Los K Vecinos más cercanos* con pesos tanto uniformes como dependientes de la distancia. Se emplean la representación vectorial y la representación en frecuencia obtenida a partir de la representación como señal discreta, cada una tanto normalizada como sin normalizar. Para cada combinación de método de aprendizaje, parámetros libres y representación de datos, se valida el rendimiento mediante el método de *Validación Cruzada anidada* para finalmente tomar el clasificador con el mejor resultado. Este clasificador final es obtenido mediante el método de *Validación Cruzada.*

El software creado para tal fin se compone de módulos que permiten realizar cada una de las etapas. El resultado final consta de los clasificadores obtenidos, su estimación de rendimiento y el sistema con el que se obtienen.

[1] Proyecto de Grado

[2] Facultad de Ingenierías Físico – Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Daniel Alfonso Sierra Bueno. Codirectora: Nydia Paola Rondón Villarreal.

ABSTRACT

TITLE:

DESIGN AND CREATION OF AN ANTIBACTERIAL PEPTIDE CLASSIFIER USING DIGITAL SIGNAL PROCESSING TECHNICS AND SUPERVISED LEARNING MODELS.[3]

AUTOR:

Diego Fernando Coba Cruz[4]

KEYWORDS:

Classifier, Peptide, Support vector machine, K Fold cross validation, KNN, Electron ion interaction potential, Digital signal, Frequency Spectrum.

DESCRIPTION:

This paper presents the design and implementation of an antibacterial peptides classifier that allow the pre-selection of those sequences that possess antibacterial activity in order to reduce the quantity of experiments that should be performed to find a successful antibacterial peptide.

Supervised learning models were used in order to *teach* the system how to discriminate a peptide with certain property from other which hasn't it. Support vector machines with linear and radial basis functions, and k-nearest neighbors with uniform and distance dependent weights were implemented. Vector and frequency spectrum mathematical representations where used both normalized and no normalized forms. For each combination of learning model, learning model parameters and mathematical representation, the estimated assessment was determined through *Nested K Fold Cross Validation* process to finally take the one with the best performance which was obtained using *K Fold Cross Validation* process. Finally, a comparison between the different learning models and representations was made as conclusions of this work.

The software created for this purpose is made by modules that allow the development of each one of the process stages. The final result is composed of the obtained classifiers, their estimated assessment metrics file and the system through which were gotten.

[3] Graduation Work

[4] Faculty of Mechanical and Physical Engineering. School of Electrical Electronics and Telecommunications engineering. Director: Daniel Alfonso Sierra Bueno. Co-Director: Nydia Paola Rondón Villarreal.

INTRODUCCIÓN

La búsqueda de la cura a las diferentes enfermedades que afectan al ser humano ha sido siempre una de las principales tareas de la ciencia. Grandes avances se han logrado a través de la historia de la humanidad en la tarea de erradicar las enfermedades y en particular las infecciones bacterianas; sin embargo, en muchos casos estos descubrimientos resultan en soluciones temporales debido al fenómeno de la resistencia antibacteriana.

En la actualidad existen múltiples bacterias resistentes a casi todos los antibióticos y, por lo tanto, es necesario el desarrollo de nuevos agentes antibacterianos que permitan controlar las infecciones ocasionadas por este tipo de bacterias. Entre los factores más importantes de esta problemática se encuentra la falta de interés que tienen las compañías farmacéuticas en el desarrollo de este tipo de medicamentos, debido en gran parte, a las bajas tasas de retorno de la inversión[*].

Esto ha sido la motivación para que la IDSA (*Infectious Diseases Society of America*) creara la iniciativa 10x'20[*]. Esta iniciativa busca un compromiso global para crear una compañía de Investigación y Desarrollo de antibióticos capaz de producir diez nuevos antibióticos sistémicos para el año 2020. Por eso la IDSA trabaja de la mano con aquellas personas o entidades interesadas en este tema.

En la búsqueda de nuevas soluciones al problema de la resistencia bacteriana, en la comunidad científica ha surgido gran interés en los péptidos antimicrobianos. Estos compuestos químicos son cadenas de aminoácidos que cumplen con ciertas propiedades físico-químicas, que permiten su interacción con las membranas de los microorganismos patógenos ocasionando su muerte o inhibiendo su crecimiento. Por lo anterior, estos péptidos pueden ser utilizados como agentes antibacterianos.

Este reporte presenta la descripción detallada del proceso de obtención de un clasificador utilizado para determinar, con cierto porcentaje de especificidad y precisión, si un péptido presenta o no actividad antibacteriana.

[*] ISDA. "Antibiotic Development: The 10 x '20 Initiative", Disponible desde Internet en: <http://www.idsociety.org/10x20/>.

1. MARCO TEÓRICO

1.1 PÉPTIDOS ANTIMICROBIANOS

Son compuestos químicos similares a las proteínas, diferenciándose de éstas por ser más cortos (6 a 100 aminoácidos). Hacen parte del sistema inmune de insectos, plantas y animales. La mayoría de ellos son pequeños péptidos catiónicos que inhiben el crecimiento de los microbios y resultan tóxicos para los virus, bacterias y hongos. Estos péptidos también activan mecanismos de reacción inmunitaria adaptativa y modifican la respuesta inflamatoria local[*]. Sus características incluyen baja toxicidad, rápida acción, baja probabilidad de generar resistencia, rápida degradación, toxicología desconocida, y altos costos de producción. Debido a esta última característica se hace tan importante el diseño racional de péptidos.

1.2 CLASIFICACIÓN MEDIANTE APRENDIZAJE SUPERVISADO

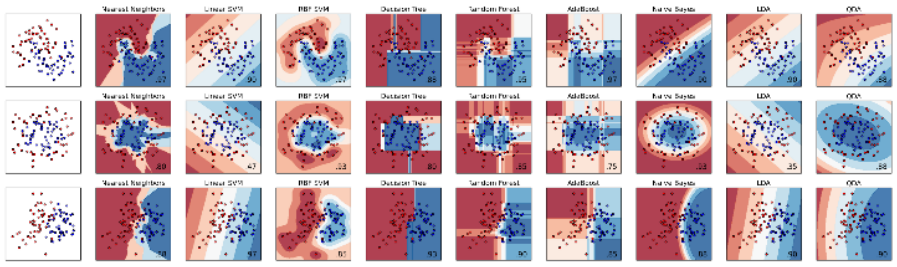
Un clasificador es un mecanismo capaz de asignar una clase a un objeto cuya clase inicialmente no está determinada, Un clasificador puede ser binario (asigna a un objeto una de dos clases) o multiclase. La clasificación es una de las principales aplicaciones del aprendizaje supervisado ya que permite al sistema *aprender* a diferenciar los elementos de una u otra clase a partir del conjunto de datos de entrenamiento.

La clasificación puede realizarse mediante diferentes algoritmos tales como

- Árboles de decisión
- K Vecinos más cercanos
- Gradiente descendiente estocástico
- Máquinas de soporte vectorial
- Redes neuronales

[*] CASTRILLÓN RIVERA Laura E, PALMA RAMOS Alejandro, DESGARENNES Carmen Padilla, "Péptidos antimicrobianos: antibióticos naturales de la piel", Dermatología Rev Mex 2007;51:57-67.

Figura 1. Ejemplos de clasificación mediante diferentes algoritmos. Fuente [9]



Fuente: SCIKIT LEARN, Supervised Learning, URL: http://scikit-learn.org/stable/supervised_learning.html #supervised-learning

El trabajo presentado en este artículo fue realizado utilizando Máquinas de soporte vectorial y el algoritmo de los K vecinos más cercanos.

1.3 MÁQUINAS DE SOPORTE VECTORIAL

Son algoritmos de clasificación optimizada pues se busca encontrar el hiperplano que separe a las clases de modo que su distancia hasta los puntos más cercanos de cada clase sea máxima. Dado que se basan en el aprendizaje supervisado es necesario tener un conjunto de datos de entrenamiento, de los cuales se conoce su clase a priori, y a partir de ellos se infiere un modelo matemático, capaz de clasificar un nuevo elemento de clase desconocida (Conjunto de prueba).

Matemáticamente, sea el conjunto de entrenamiento (x_1, y_1), (x_2, y_2) … (x_n, y_n) donde $x_i \in X$, $y_i \in \beta$ donde X representa el conjunto de los péptidos, $\beta = \{-1, 1\}$ indica la presencia o ausencia de actividad antibacteriana y n la cantidad de individuos.

Resolver el problema de clasificación implica encontrar el hiperplano con máximo margen de separación que divide a los vectores cuyo y_i=1 de aquellos cuyo y_i=-1. Tal hiperplano puede ser descrito por la ecuación

$$\boldsymbol{w}.\boldsymbol{x} + b = 0 \quad (1)$$

donde **w** es el vector normal al plano y "**.**" denota el producto punto vectorial.

Si los datos de entrenamiento pueden separarse linealmente, puede escogerse un par de hiperplanos que separen a las dos clases, de manera que en el espacio entre

ellos no quede ningún dato y que a la vez dicho espacio se maximice. Estos hiperplanos pueden ser descritos por las ecuaciones:

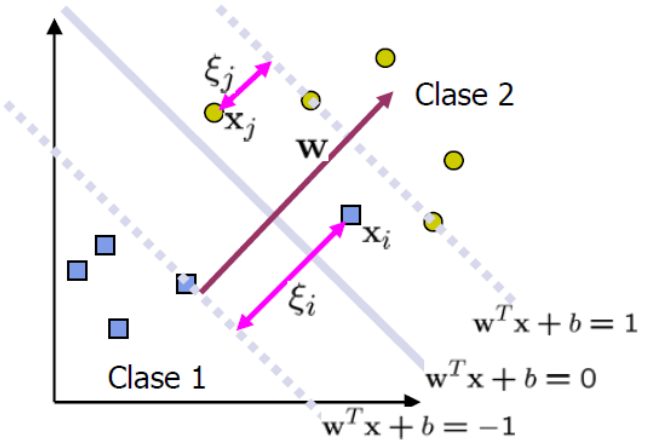
$\boldsymbol{w}.\boldsymbol{x} + b = 1$ (2)

y

$\boldsymbol{w}.\boldsymbol{x} + b = -1$ (3)

respectivamente.

Figura 2. Representación gráfica del problema de clasificación.



Fuente: CAMACHO URREA, Francy Liliana, “Sistema de clasificación de péptidos antibacterianos utilizando máquinas de soporte vectorial”, Trabajo de Grado. Universidad Industrial de Santander. 2012

La separación entre los hiperplanos está dada por la expresión $\frac{2}{||\boldsymbol{w}||}$. Además, tenemos que para los datos de la primera clase (y_i=1)

$\boldsymbol{w}.\boldsymbol{x} + b \geq 1$ (4)

mientras que para los de la segunda clase (y_i=-1)

$\boldsymbol{w}.\boldsymbol{x} + b \leq -1$ (6)

entonces

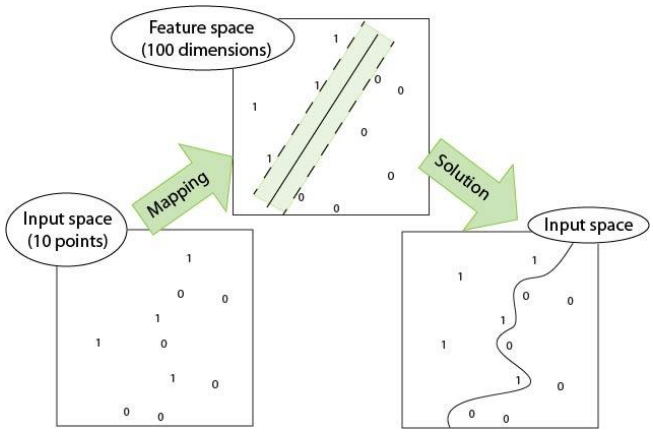
$y_i(\boldsymbol{w}.\boldsymbol{x} + b) \geq 1$ (7).

El problema se reduce a minimizar $||\boldsymbol{w}||$ lo que es equivalente a minimizar $\frac{||\boldsymbol{w}||^2}{2}$ sujeto a (7). El cambio de variable se realiza para poder aplicar los métodos de

optimización cuadrática.

El algoritmo original es un clasificador lineal, sin embargo, existe una manera de hacer clasificación no lineal. Esto se logra sustituyendo el producto punto por una función kernel de tipo no lineal lo que equivale a llevar los vectores de entrenamiento a un espacio diferente en el cual sea posible su separación lineal.

Figura 3. Efecto de la aplicación de una función kernel. Fuente [11]



Fuente: THORNTON, Chris. SVM-Illustration. URL: http://users.sussex.ac.uk/~christ/crs/ml/copied-pics/SVM-illustration.jpg

Para el caso del kernel de base radial o RBF la función que lo describe es:

$$K(x_i, x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0 \quad (8)$$

la cual tiene un parámetro libre.

Además, puede agregarse un margen de error para encontrar una solución aproximada en los casos donde no es posible encontrar una solución que separe completamente a las clases. Esto resulta en la inclusión de un segundo parámetro, llamado *soft margin parameter* C que es un parámetro de penalización y permite algunos errores de clasificación mientras los penaliza.

1.4 KNN

K - *Nearest Neighbors*, es un conjunto de técnicas tanto supervisadas como no supervisadas. Las técnicas supervisadas tienen aplicación en problemas de regresión y clasificación.

La idea básicamente consiste en encontrar un número predefinido (K) de las muestras más cercanas al individuo bajo análisis y predecir la etiqueta de éste. El

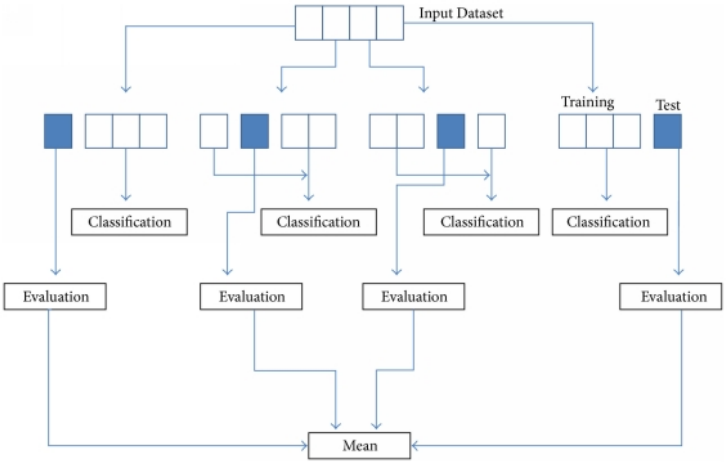
número K puede ser definido por el usuario o puede ser basado en la densidad local de puntos. La distancia puede ser cualquier métrica, siendo la distancia euclideana la más utilizada[*].

El algoritmo básico utiliza pesos uniformes, lo que implica que la etiqueta asignada al nuevo punto únicamente depende del "voto" de la mayoría de los vecinos más cercanos. Existe una variación del método original que asigna a cada vecino, un peso proporcional al inverso de la distancia al nuevo punto.

1.5 VALIDACIÓN CRUZADA DE N ITERACIONES

Cuando se hace uso de un algoritmo de clasificación que requiere de parámetros, como el parámetro K en KNN o los parámetros C y Gamma de las máquinas de soporte vectorial, se hace necesario un método para obtener el conjunto de parámetros que permita realizar la mejor clasificación para el conjunto de datos de trabajo.

Figura 4. Validación Cruzada de N = 4 iteraciones.



Fuente: RESEARCHGATE, K -fold cross-validation scheme, with K=4 and one classifier. - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/290441113_fig13_K-fold-cross-validation-scheme-with-K4-and-one-classifier.

[*] SCIKIT-LEARN, "Nearest Neighbors", Disponible desde Internet en: < http://scikit-learn.org/stable/modules/neighbors.html

Esto se logra mediante el método de validación cruzada el cual consiste en:

Separar la totalidad de los datos en N subconjuntos

Realizar el proceso de entrenamiento y pruebas tantas veces como lo indique el valor de N. En cada una de las iteraciones, se debe utilizar uno de los subconjuntos como conjunto de prueba y los restantes subconjuntos se unen para formar el conjunto de entrenamiento. Este procedimiento permite que todos los datos sean utilizados en el conjunto de prueba en una oportunidad.

De cada una de estas pruebas se obtienen sus medidas de rendimiento y a partir de estas se calcula el puntaje de selección, el cual finalmente es promediado para llegar a su valor final.

El proceso entero debe repetirse para cada combinación de parámetros posible de manera que se obtenga un puntaje de selección por cada una y así seleccionar aquella con la mejor puntuación, es decir, el conjunto de parámetros óptimos.

Finalmente, el clasificador definitivo se obtiene al entrenar la máquina de soporte con la totalidad de los datos como conjunto de entrenamiento y haciendo uso de los parámetros óptimos encontrados.

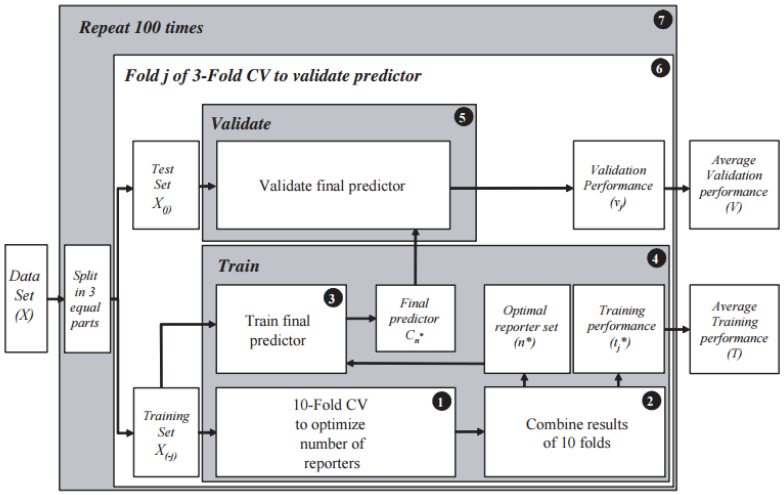
1.6 VALIDACIÓN CRUZADA ANIDADA

Mediante el método de validación cruzada descrito en el apartado anterior, el desempeño estimado del clasificador parece depender de la manera en la que son divididos los datos, por lo que se prefiere un proceso de validación dividido en el que el desempeño estimado de cada porción de datos se obtiene de manera independiente y es agregado a la estimación total[*].

Adicionalmente, el algoritmo sencillo no permite evaluar el rendimiento de la máquina de soporte obtenida con los parámetros óptimos sin incurrir en sesgo ya que no se dispone de una porción de los datos que no haya estado involucrada en su obtención y que pueda ser utilizada como conjunto de pruebas.

[*] WESSELS Lodewyk F. A., REINDERS Marcel J. T., HART Augustinus A. M., VEENMAN Cor J., DAI Hongyue, HE Yudong D. and VAN'T VEER Laura J., "A
protocol for building and evaluating predictors of disease state based on microarray data", Bioinformatics Vol 21

Figura 5. Diagrama de la validación cruzada anidada de N iteraciones.



Fuente: LODEWYK F. A. Wessels, MARCEL J. T. Reinders, AUGUSTINUS A. M. Hart, COR J. Veenman, HONGYUE Dai, YUDONG D. He and VAN'T VEER Laura J., "A protocol for building and evaluating predictors of disease state based on microarray data", Bioinformatics Vol 21.

El presente trabajo se realizó haciendo uso del algoritmo de validación cruzada anidada el cual consiste en:

- Dividir el conjunto de datos en N subconjuntos externos.
- Generar los N conjuntos externos de entrenamiento y pruebas.
- A partir de cada conjunto externo de entrenamiento realizar una nueva subdivisión.
- Construir nuevos conjuntos de entrenamiento y pruebas internos.
- Realizar entrenamiento, pruebas y obtención de medidas de rendimiento del proceso interno.
- Obtener los valores óptimos.
- En el proceso externo, entrenar utilizando los valores óptimos obtenidos en el proceso interno.
- Evaluar el rendimiento del clasificador utilizando el conjunto de pruebas externo y calculando las medidas de rendimiento.

- Obtener la media de las medidas de rendimiento obtenidas para cada una de las pruebas del proceso externo con el fin de hallar el valor estimado del rendimiento del clasificador final.

1.7 MEDIDAS DE RENDIMIENTO

Se hace uso de las siguientes medidas de rendimiento

- Sensibilidad: Proporción de Verdaderos positivos respecto a la cantidad de positivos en la muestra.
- Especificidad: Proporción de Verdaderos negativos respecto a la cantidad de negativos de la muestra.
- Precisión: Proporción de verdaderos (tanto positivos como negativos) respecto al total de elementos de la muestra.
- Tasa de Falso Positivo: Proporción de falsos positivos respecto al total de negativos de la muestra.
- Tasa de Falso Negativo: Proporción de falsos negativos respecto al total de positivos de la muestra.
- Coeficiente de correlación de Mattews.
- Puntaje de Selección: Se obtiene mediante la siguiente ecuación

$$\frac{[(Trs+Tre+0.5*Tra)*0.6+(Ts+Te+Ta*0.5)]}{4} \quad (9)$$

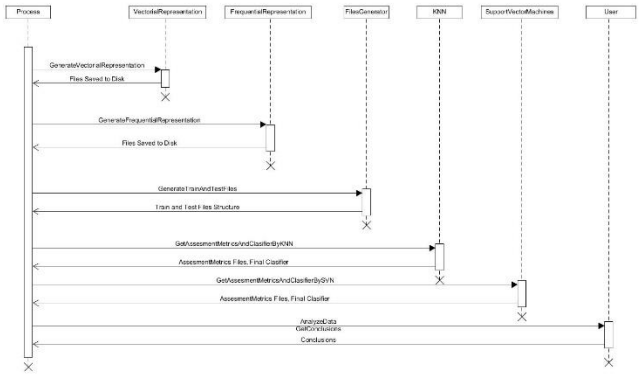
Donde

- Trs – Sensibilidad sobre el conjunto de entrenamiento.
- Tre – Especificidad sobre el conjunto de entrenamiento.
- Tra – Precisión sobre el conjunto de entrenamiento.
- Ts – Sensibilidad sobre el conjunto de pruebas.
- Te – Especificidad sobre el conjunto de pruebas.
- Ta – Precisión sobre el conjunto de pruebas.

2. DISEÑO Y OBTENCIÓN DEL CLASIFICADOR

El proceso general para la obtención del clasificador consiste en la ejecución de los diferentes módulos construidos según se muestra en el siguiente diagrama:

Figura 6. Proceso General de Obtención del Clasificador



Se parte del conjunto de datos el cual pasa por dos procesos de representación, uno como señal discreta y otro como vector, seguido de una etapa que genera para cada uno, los conjuntos de entrenamiento y pruebas. Posterior a ello se ejecuta la clasificación y se obtienen las medidas de rendimiento junto a los clasificadores finales para la variante elegida de cada algoritmo.

2.1 CONJUNTOS DE DATOS

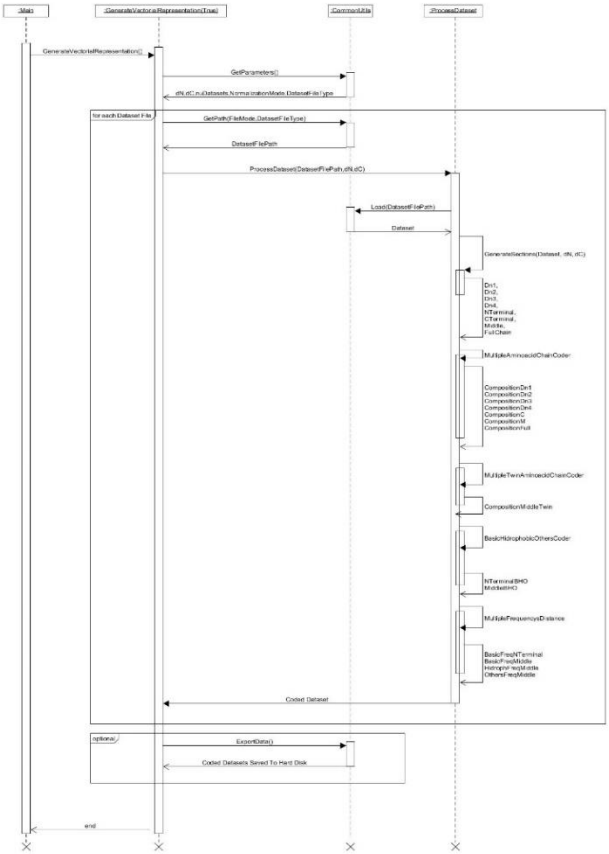
El punto de partida es la obtención de los conjuntos de datos de péptidos, los cuales son una recopilación de las bases de datos APD* y CAMP** del año 2015.

* APD Peptide's Database, 2015, Disponible desde Internet en: <http://aps.unmc.edu/AP/database/antiB.php>
** CAMP Peptide's Database, 2015, Disponible desde Internet en: <http://www.camp.bicnirrh.res.in/>

2.2 REPRESENTACIONES NUMÉRICAS

2.2.1 Vectorial. Fueron generadas a partir de las cadenas de aminoácidos. Se utilizó la representación vectorial* obtenida mediante el módulo Vectorial_Representation el cual básicamente divide cada cadena en tres porciones llamadas N Terminal, Middle y C Terminal, luego determina la frecuencia de aparición de cada aminoácido en cada una de estas porciones, así como la frecuencia de aparición de aminoácidos dobles (un mismo aminoácido dos veces consecutivas) y la frecuencia de la distancia entre aminoácidos de cada tipo (hidrofóbico, básico u otro)

Figura 7. Diagrama de Secuencia de la Representación Vectorial



* MATSUDA Setsuro, VERT Jean-Philippe, SAIGO Hiroto, UEDA Nobuhisa, TOH Hiroyuki, AKUTSU Andtatsuya, "A novel representation of protein sequences for prediction of subcellular location using support vector machines", ProteinScience (2005),14:2804–2813.

Se generaron tres variantes de la representación vectorial

- Sin Normalización
- Normalizado respecto a la clase
- Normalizado respecto al conjunto de datos.

2.2.2 Espectro en frecuencia. Se utilizó además la representación como espectro en frecuencia de señales discretas basadas en EIIP obtenida mediante el módulo Frequential_Representation tomando en cuenta únicamente la magnitud.
Se generaron dos variantes de la representación en frecuencia

- Normalizada
- Sin normalización

Figura 8. Obtención de representación basada en EIIP

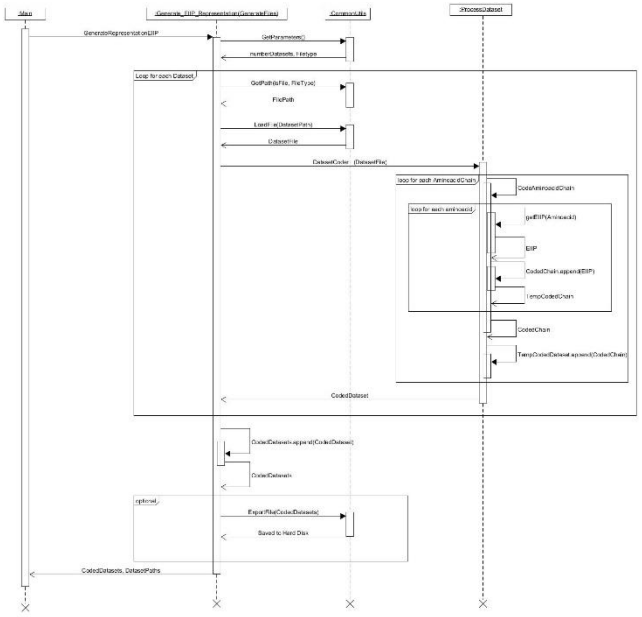
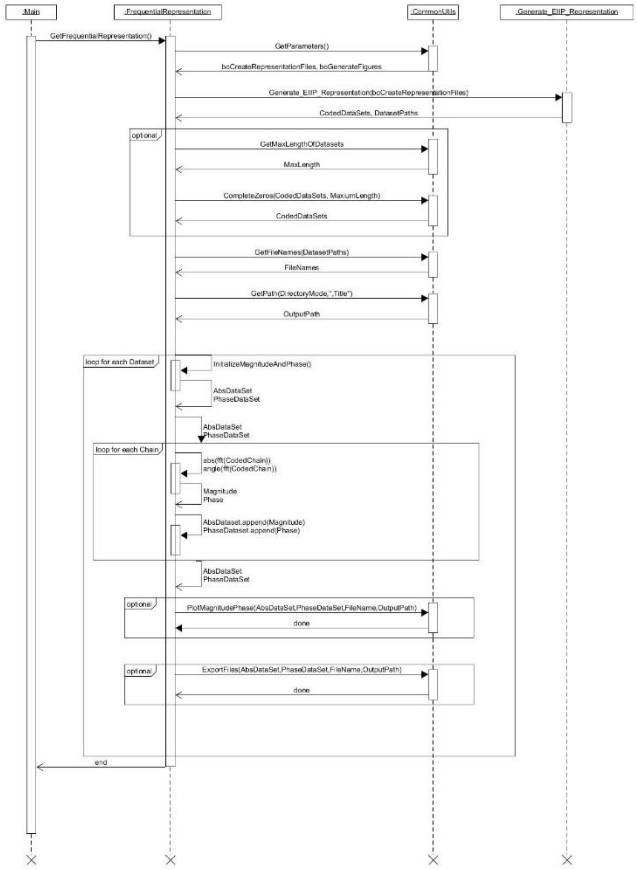


Figura 9. Representación en Frecuencia



2.3 SUBCONJUNTOS DE ENTRENAMIENTO Y PRUEBAS

Una vez obtenidas las representaciones, inició la obtención de los subconjuntos de entrenamiento y pruebas para el proceso de validación cruzada anidada, estos subconjuntos fueron generados mediante el módulo kFoldFilesGenerator. Se generó una única estructura de archivos para cada representación.

Figura 10. Estructura de archivos de entrenamiento y pruebas.

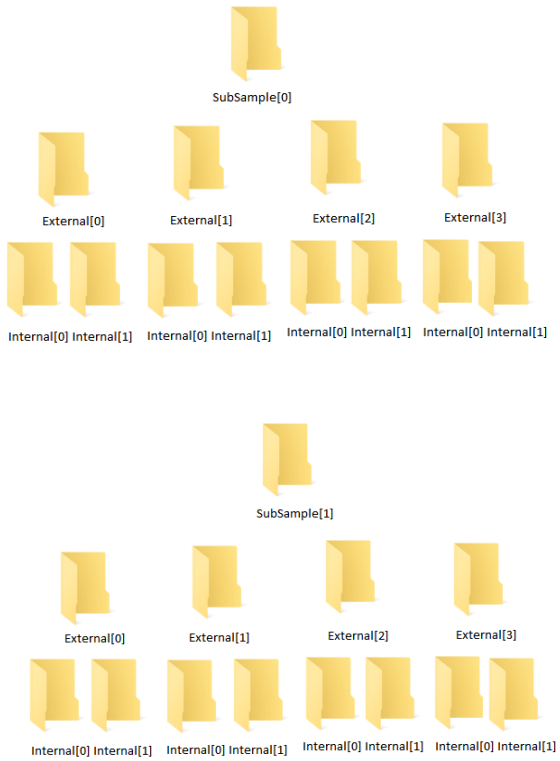
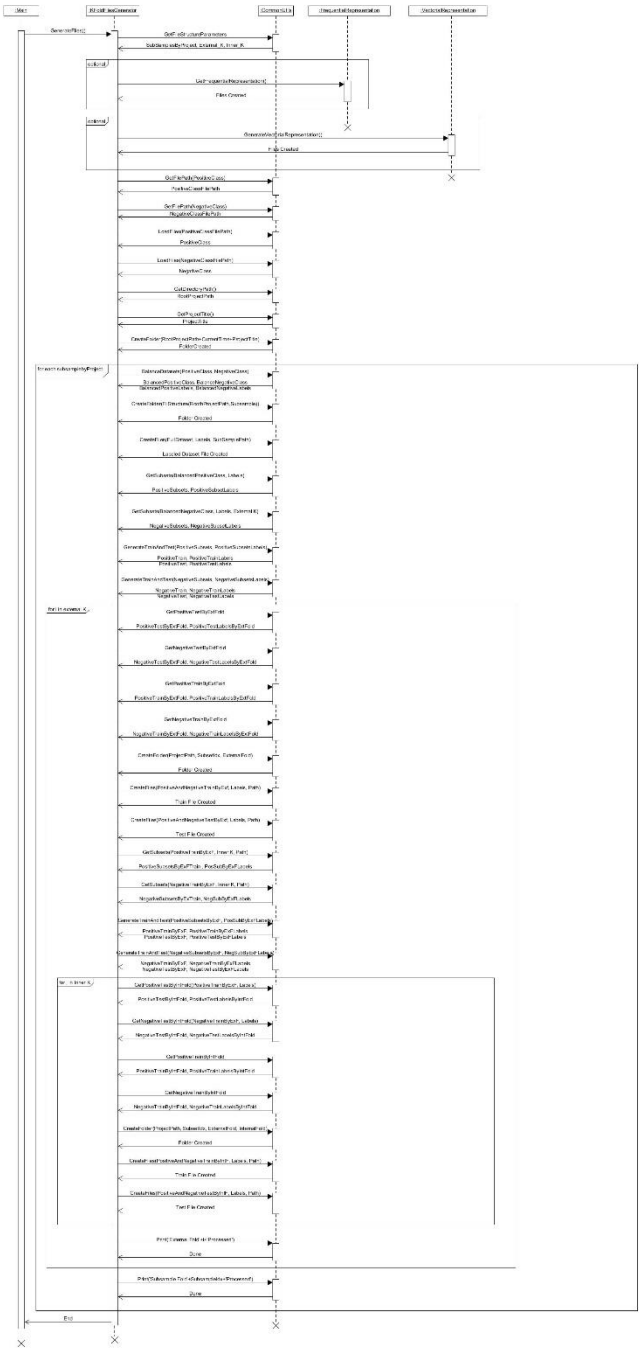


Figura 11. Generación de la estructura de archivos



2.4 ARCHIVOS DE MEDIDAS DE RENDIMIENTO

Los archivos de nombre *MeanAssesmentMetrics* corresponden a las medidas de rendimiento del nivel inferior promediadas, por ejemplo, el archivo *SVM-RBF MeanAssesmentMetrics* en la carpeta principal de la representación, se obtiene promediando los diferentes archivos del mismo nombre encontrados en las carpetas SubSample[i], el cual a su vez se obtiene a partir del promedio de los archivos del mismo nombre contenidos en las carpetas External[j]. Dentro de las carpetas External se encuentran además los archivos SVM-RBF AssesmentMetrics[Train] y AssesmentMetrics[Test] que corresponden a las medidas de rendimiento del clasificador obtenido y probado con los conjuntos de entrenamiento y pruebas respectivamente; se tiene también el archivo SVM-RBF MeanAssesmentMetrics, resultado de promediar las medidas de rendimiento encontradas en los SVM-RBF AssesmentMetrics[Test] y SVM-RBF AssesmentMetrics[Train] de las carpetas Internal[k], estos últimos contienen las medidas de rendimiento del clasificador generado en la iteración interna, para cada combinación de parámetros.

2.5 VALIDACIÓN CRUZADA ANIDADA

Luego de generados los archivos de entrenamiento y pruebas, se realizó el proceso de validación cruzada anidada para los siguientes algoritmos:

- Máquinas de soporte con Kernel Lineal
- Máquinas de soporte con Kernel RBF
- KNN con pesos uniformes
- KNN con pesos dependientes de la distancia.

Figura 12. Validación cruzada anidada mediante KNN

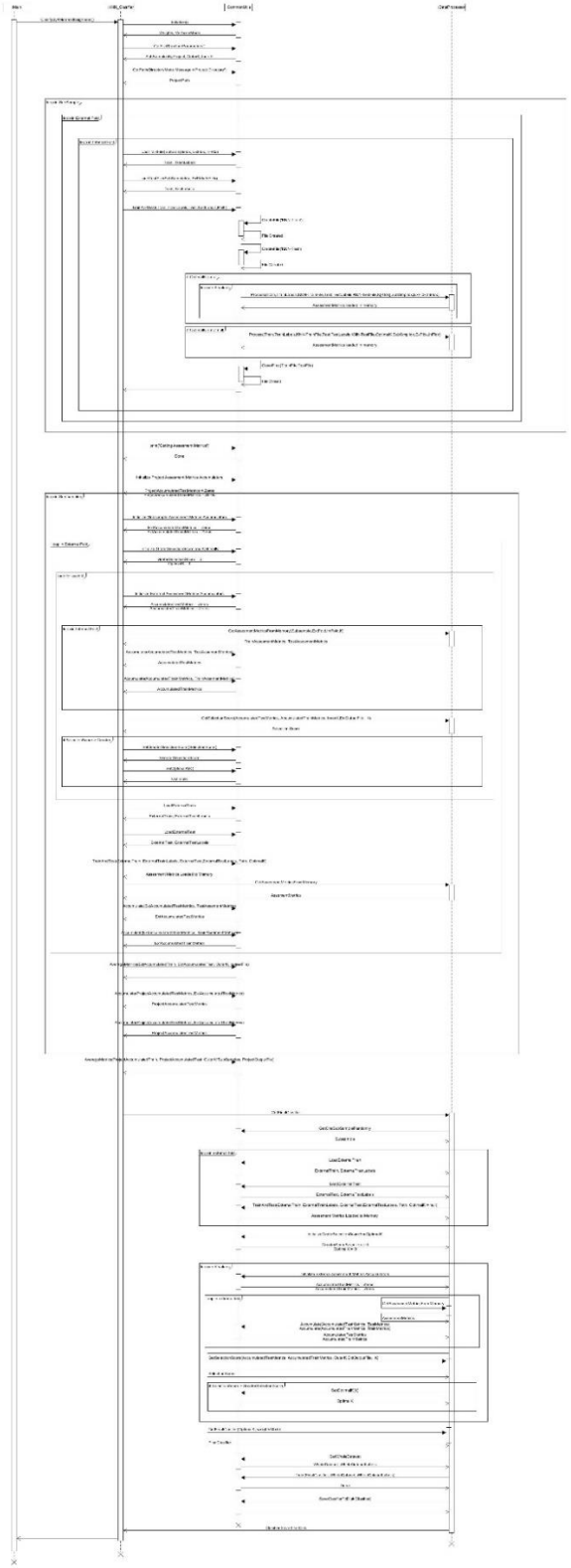
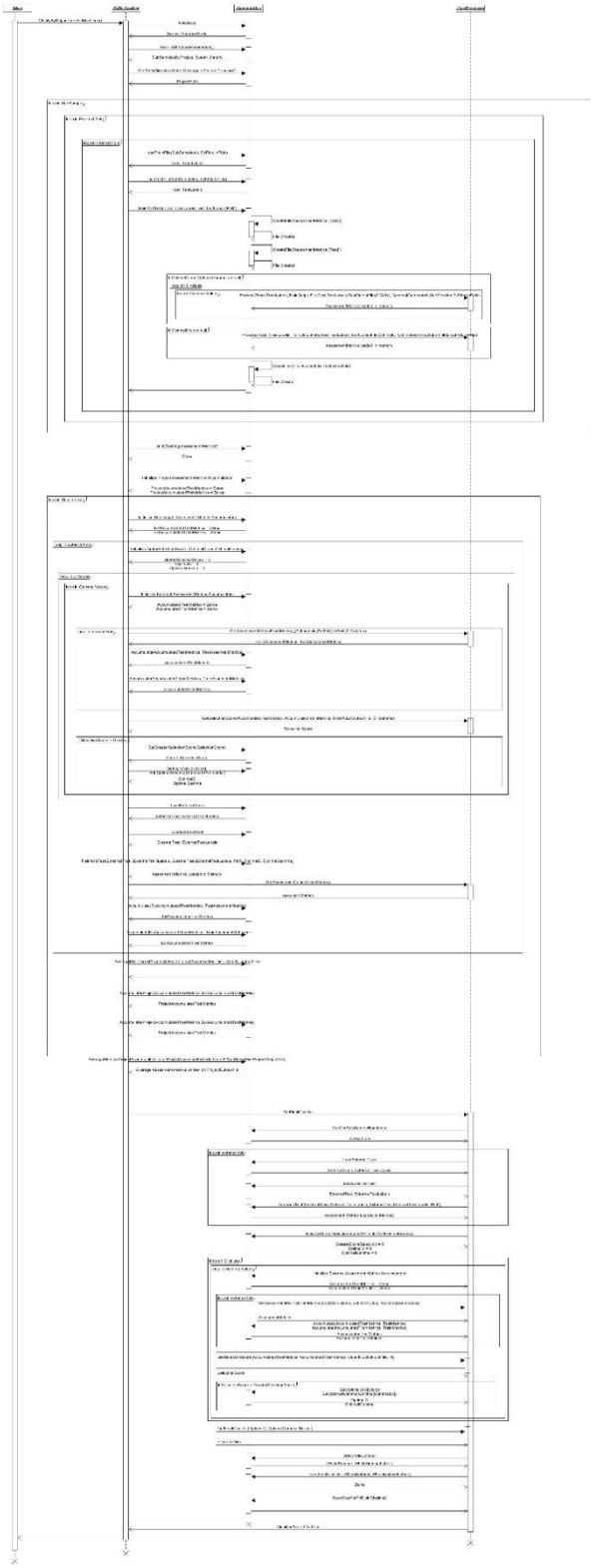


Figura 13. Validación cruzada anidada mediante SVN.



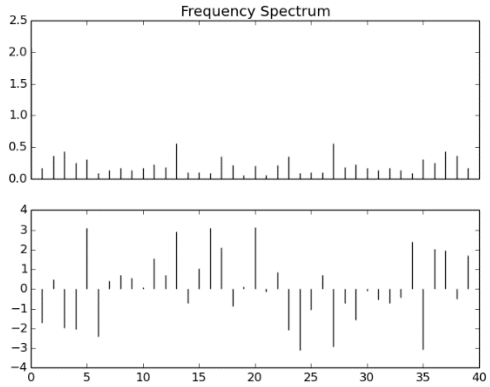
Esto con cada una de las representaciones obtenidas, lo que asciende a un total de 20 clasificadores.

En la estructura de archivos se encuentra el resultado final de la estimación de desempeño de cada clasificador junto con el clasificador generado (clasificador entrenado exportado como grupo de archivos), el cual puede ser importado para su uso en otras aplicaciones.

3. RESULTADOS

Entre los resultados obtenidos se encuentran las representaciones vectoriales (de 184 componentes) de los péptidos y las representaciones de los mismos como señales discretas. Adicionalmente, también se generaron las correspondientes representaciones en frecuencia, provenientes de aplicar el algoritmo DFT sobre las representaciones de los péptidos como señales discretas.

Figura 14. Espectro en frecuencia para la señal obtenida a partir la primera cadena del conjunto de datos abps, Magnitud (Arriba) – Fase (Abajo).



Se generó la estructura de archivos para entrenamiento y pruebas para cada representación.

Las medidas de rendimiento obtenidas para cada uno de los algoritmos y representaciones son las siguientes:

Figura 15 - Medidas de rendimiento obtenidas para cada combinación de representaciones y algoritmos.

SVM Sin Normalización

Lineal

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.833890701945	0.696369091483	0.303630908517	0.166109298055	0.765129896714	0.535562541829
Test	0.746410628019	0.621782608696	0.378217391304	0.253589371981	0.684096618357	0.372448665645

RBF

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	1.0	1.0	0.0	0.0	1.0	1.0
Test	0.953425120773	0.298425120773	0.701574879227	0.0465748792271	0.625925120773	0.336990065104

Figura 16 - Medidas de rendimiento obtenidas para cada combinación de representaciones y algoritmos (Continuación).

SVM Normalización Global

Lineal

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.78005222772	0.552459030775	0.447540969225	0.21994777228	0.666255629248	0.341889305083
Test	0.768589371981	0.540724637681	0.459275362319	0.231410628019	0.654657004831	0.31970330742

RBF

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.783732381186	0.595741247617	0.404258752383	0.216267618814	0.689736814402	0.386673263138
Test	0.765342995169	0.575797101449	0.424202898551	0.234657004831	0.670570048309	0.349490609775

SVM Normalización por Clase

Lineal

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.686661812355	0.753933571955	0.246066428045	0.313338187645	0.720297692155	0.441890247657
Test	0.669309178744	0.742869565217	0.257130434783	0.330690821256	0.706089371981	0.415422739365

RBF

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.765480927817	0.725465078195	0.274534921805	0.234519072183	0.745473003006	0.491470401448
Test	0.741618357488	0.71170531401	0.28829468599	0.258381642512	0.726661835749	0.455308481323

SVM Magnitud FFT EIIP – Sin normalización

Lineal

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.755408288592	0.555797978735	0.444202021265	0.244591711408	0.655603133663	0.317880461782
Test	0.70238647343	0.499603864734	0.500396135266	0.29761352657	0.600995169082	0.207549159897

RBF

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.964028143313	0.804963154016	0.195036845984	0.0359718566867	0.884495648665	0.779198473645
Test	0.745599033816	0.552951690821	0.447048309179	0.254400966184	0.649275362319	0.30696777365

SVM Magnitud FFT EIIP – Normalizada

Lineal

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.692927676598	0.447618872162	0.552381127838	0.307072323402	0.57027327438	0.14568639071
Test	0.689347826087	0.442463768116	0.557536231884	0.310652173913	0.565905797101	0.13712541089

RBF

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.692806728194	0.447812576762	0.552187423238	0.307193271806	0.570309652478	0.145746803468
Test	0.688913043478	0.442463768116	0.557536231884	0.311086956522	0.565688405797	0.136649911828

KNN Vectorial Sin Normalización

Uniforme

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	1.0	1.0	0.0	0.0	1.0	1.0
Test	0.67390821256	0.732202898551	0.267797101449	0.32609178744	0.703055555556	0.408330462624

Dependiente de la Distancia

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	1.0	1.0	0.0	0.0	1.0	1.0
Test	0.683690821256	0.714792270531	0.285207729469	0.316309178744	0.699241545894	0.400394010394

Figura 17 - Medidas de rendimiento obtenidas para cada combinación de representaciones y algoritmos (Continuación).

KNN Vectorial Normalización Global

Uniforme

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	1.0	1.0	0.0	0.0	1.0	1.0
Test	0.680188405797	0.735748792271	0.264251207729	0.319811594203	0.707968599034	0.418060149568

Dependiente de la Distancia

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	1.0	1.0	0.0	0.0	1.0	1.0
Test	0.682565217391	0.719632850242	0.280367149758	0.317434782609	0.701099033816	0.404058467944

KNN Vectorial Normalización por Clase

Uniforme

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	1.0	1.0	0.0	0.0	1.0	1.0
Test	0.59390821256	0.862487922705	0.137512077295	0.40609178744	0.728198067633	0.475500086052

Dependiente de la Distancia

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	1.0	1.0	0.0	0.0	1.0	1.0
Test	0.581903381643	0.864666666667	0.135333333333	0.418096618357	0.723285024155	0.467542345019

KNN Magnitud FFT EIIP – Sin normalización

Uniforme

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.999636920845	0.999540068545	0.00045993145477	0.000363079154531	0.999588494695	0.999177983082
Test	0.613057971014	0.677144927536	0.322855072464	0.386942028986	0.645101449275	0.292303835255

Dependiente de la Distancia

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.999201202466	0.999975786925	2.42130750605e-05	0.000798797534243	0.999588494695	0.999177983082
Test	0.630724637681	0.652589371981	0.347410628019	0.369275362319	0.641657004831	0.284970640859

KNN Magnitud FFT EIIP – Normalizada

Uniforme

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.99951585547	0.999297937795	0.000702062205374	0.000484144529834	0.999406896632	0.998815108344
Test	0.614391304348	0.662512077295	0.337487922705	0.385608695652	0.638451690821	0.278593141913

Dependiente de la Distancia

	Sensitivity	Specificity	False Positive Rate	False Negative Rate	Accuracy	Mattews Correlation Coefficient
Train	0.99888643249	0.999927360775	7.26392251816e-05	0.00111356751003	0.999406896632	0.998815574538
Test	0.638589371981	0.639415458937	0.360584541063	0.361410628019	0.639002415459	0.279832495573

Se obtuvieron los siguientes puntajes de selección sobre cada uno de los clasificadores generados.

Figura 18. Resumen de resultados por clasificador. Se resalta el clasificador con mejor desempeño estimado.

Representación	Método	Puntaje de Selección
Vectorial Sin Normalización		
	SVM Lineal	0.71448409774113
	SVM RBF	0.7662032
	KNN Uniforme	0.814409722
	KNN Distancia	0.812025966
Vectorial Normalización Global		
	SVM Lineal	0.659006489
	SVM RBF	0.677757586
	KNN Uniforme	0.817480374
	KNN Distancia	0.813186896
Vectorial Normalización por Clase		
	SVM Lineal	0.711417492
	SVM RBF	0.733716023
	KNN Uniforme	0.830123792
	KNN Distancia	0.82705314
Magnitud FFT EIIP – Sin normalización		
	SVM Lineal	0.621473156
	SVM RBF	0.73748297
	KNN Uniforme	0.778034091
	KNN Distancia	0.775881314
Magnitud FFT EIIP – Normalizada		
	SVM Lineal	0.567543601
	SVM RBF	0.567421373
	KNN Uniforme	0.773809893
	KNN Distancia	0.774154096

Se encontró que el puntaje de selección varía de manera similar para las diferentes representaciones.

Figura 19 Puntaje de Selección en función de C y Gamma para la representación en frecuencia.

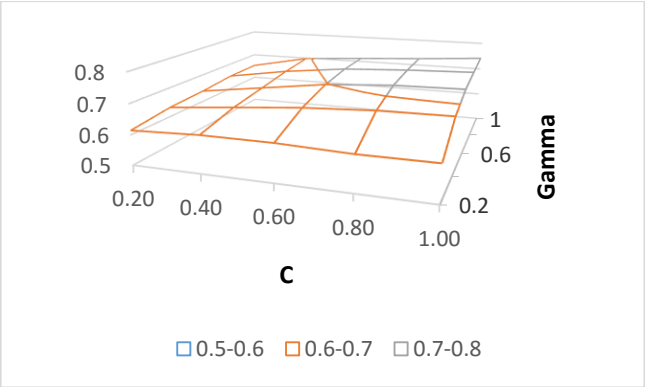


Figura 20. Puntaje de Selección en función de C y Gamma para la representación vectorial. Fuente: Autor.

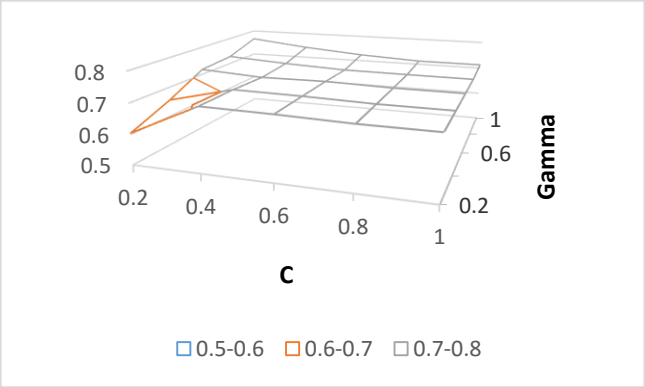


Figura 21. Variación del puntaje de selección en función de K para la representación Vectorial.

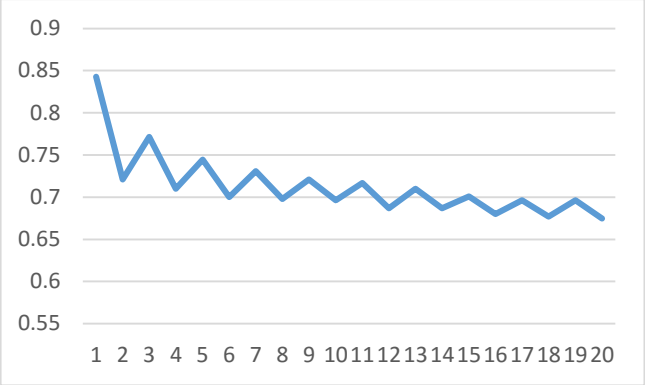


Figura 22. Variación del puntaje de selección en función de K para la representación Vectorial y la variante dependiente de la distancia.

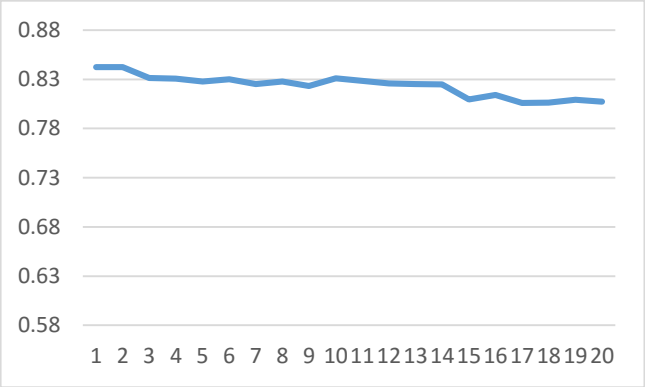


Figura 23. Variación del puntaje de selección en función de C para la representación en frecuencia.

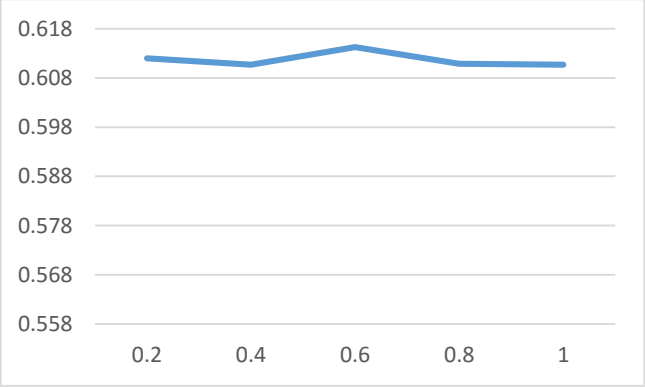
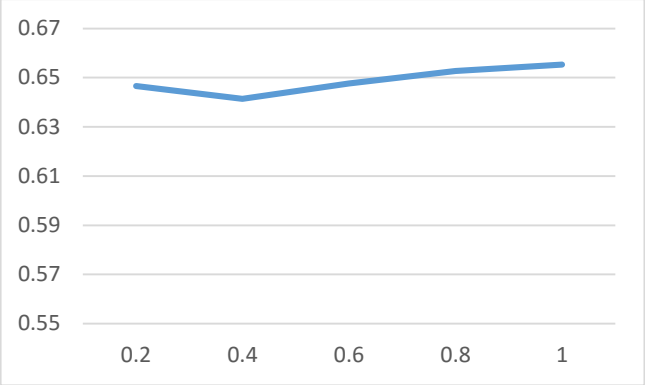


Figura 24. Variación del puntaje de selección en función de C para la representación vectorial.



4. CONCLUSIONES

Se crearon veinte clasificadores los cuales presentaron rendimiento entre 56.74% y 83.3%.

El clasificador con el mejor desempeño estimado según el puntaje de selección utilizado es el generado mediante el algoritmo KNN Uniforme para la representación vectorial normalizada por clase con un puntaje de selección de 83.3%.

El mejor clasificador generado fue el obtenido a partir de la representación vectorial sin normalización presentando un desempeño del 71.45% para el algoritmo lineal y un 76.62% para el algoritmo con kernel de base radial.

La representación en frecuencia mostró un mejor desempeño en su versión no normalizada obteniendo un puntaje de selección un 9.5% mayor para la máquina de soporte vectorial lineal, un 29.97% mayor para la máquina de soporte vectorial con kernel de base radial, un 0.54% mayor para el algoritmo KNN y un 0.22% mayor para el algoritmo KNN dependiente de la distancia.

El puntaje de selección en función del parámetro K muestra una tendencia negativa, siendo este comportamiento descrito mediante la ecuación $y = -0.0047x + 0.7622$ para el algoritmo básico $y = -0.0017x + 0.8415$ para la variante dependiente de la distancia.

El algoritmo KNN dependiente de la distancia mostró no disminuir su desempeño para valores pares del parámetro libre K a diferencia de la versión básica de este.

El puntaje de selección para la máquina de soporte vectorial lineal con representación en frecuencia en función del parámetro C está dado por la siguiente ecuación $y = -0.0002x + 0.6124$, mostrando una ligera tendencia negativa.

El puntaje de selección en función del parámetro C para la máquina de soporte vectorial lineal con representación vectorial está descrito mediante la ecuación $y = 0.0029x + 0.6402$, presentando una tendencia positiva.

La máquina de soporte vectorial con kernel de base radial mostró un mejor desempeño respecto a la máquina de soporte vectorial lineal siendo un 7.24% mayor para la representación vectorial sin normalización, un 2.84% mayor para la representación vectorial con normalización global, un 3.13% mayor para la

representación vectorial con normalización por clase, un 18.67% mayor para la representación en frecuencia sin normalización y un 0.02% menor para la representación en frecuencia normalizada.

El puntaje de selección en función de los parámetros C y Gamma, presenta una tendencia positiva en función de C para valores de Gamma de hasta 0.4 tanto con la representación vectorial como con la representación en frecuencia. Para valores de Gamma superiores a 0.4 la tendencia pasa a ser negativa para la representación vectorial mientras que mantiene una tendencia positiva para la representación en frecuencia sin normalización.

El puntaje de selección no muestra variación respecto a los parámetros C y Gamma para la representación en frecuencia normalizada.

Se observa que la representación en frecuencia presenta un desempeño hasta un 13.2% menor respecto a la representación vectorial para la máquina de soporte vectorial lineal, hasta un 3.7% menor para la máquina de soporte vectorial con kernel de base radial, hasta un 6.27% menor para el algoritmo KNN y hasta un 6.39% menor para el algoritmo KNN dependiente de la distancia; con lo cual, se concluye que es una representación útil y presenta un resultado satisfactorio dada su simplicidad.

Se entrega la totalidad de los clasificadores generados junto con el sistema que posibilita su obtención. Aunque no se cuenta con una interfaz gráfica de usuario, su uso es bastante sencillo, pues requiere solo de seleccionar archivos o carpetas y realizar configuraciones básicas.

El sistema construido para realizar la obtención de los clasificadores, puede ser utilizado en futuras investigaciones relacionadas. Su uso no se limita a datos de naturaleza genómica, puede utilizarse, en general, para obtener clasificadores destinados a cualquier aplicación.

5. RECOMENDACIONES

Como recomendaciones para trabajos futuros se plantea construir una interfaz gráfica de usuario que facilite aún más la tarea de generar los clasificadores. Esta podría contar con un archivo de configuraciones y una opción en la interfaz que permita editarlo. Se recomienda, además, en cuanto a clasificadores generados a partir de la representación en frecuencia, realizar el proceso conocido como feature forward selection que consiste en generar un listado de descriptores de acuerdo al rendimiento obtenido con cada uno de ellos. Para cada descriptor se crea un clasificador y posteriormente, el descriptor para el cual el clasificador obtuvo el mejor rendimiento, es seleccionado. El proceso de selección de descriptores continúa con los descriptores restantes hasta que el clasificador creado con todos los descriptores que han sido seleccionados, alcance el rendimiento deseado.

BIBLIOGRAFÍA

APD. Peptide's Database. [En línea]. 2015. [Citado 08-Sept-2016]. Disponible desde Internet en: http://aps.unmc.edu/AP/database/antiB.php

CAMACHO URREA, Francy Liliana, "Sistema de clasificación de péptidos antibacterianos utilizando máquinas de soporte vectorial", Trabajo de Grado. Universidad Industrial de Santander. 2012

CAMP. Peptide's Database. [En línea]. 2015. [Citado 08-Sept-2016]. Disponible desde Internet en: <http://www.camp.bicnirrh.res.in/>

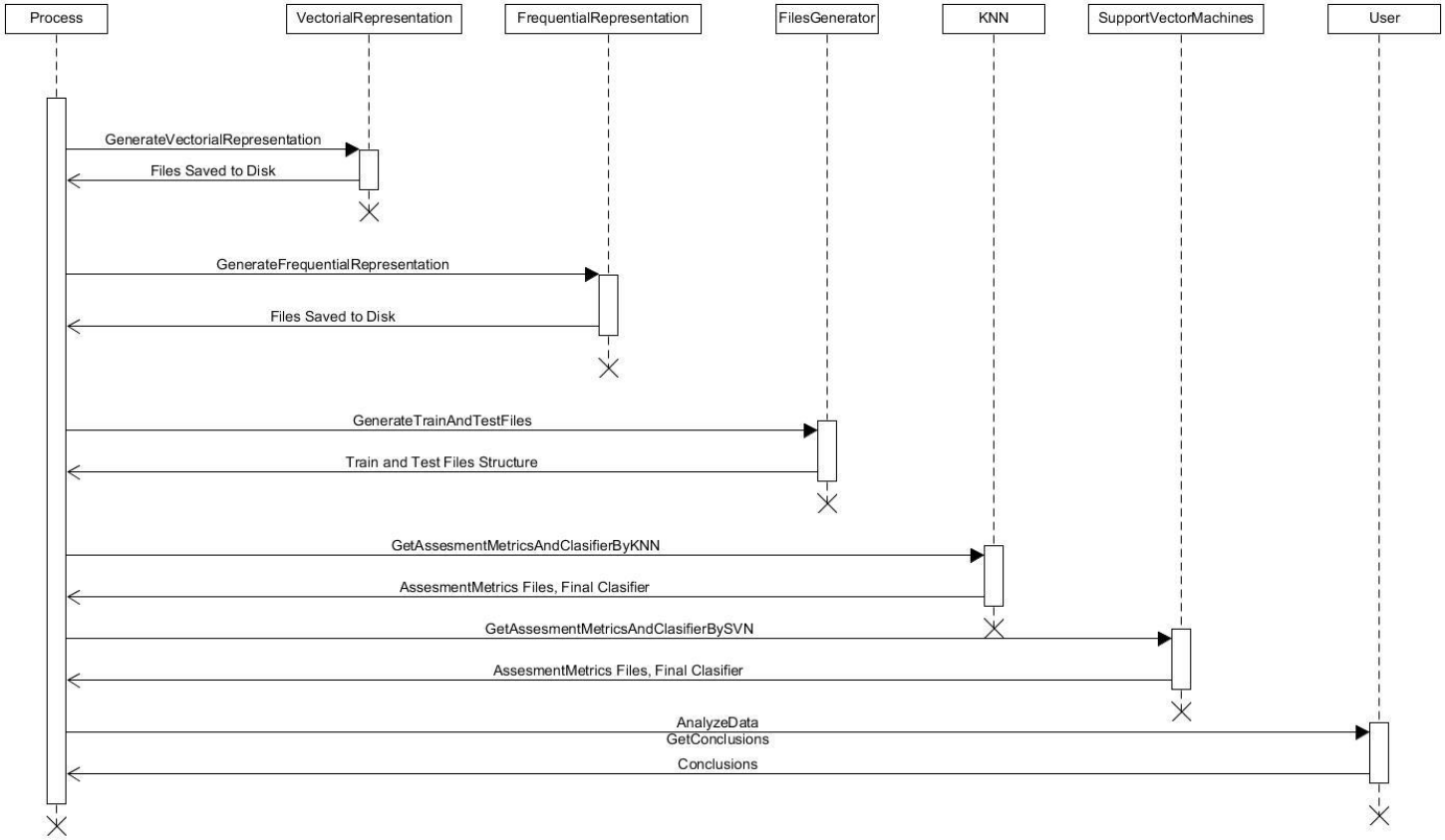
CASTRILLÓN RIVERA Laura E, PALMA RAMOS Alejandro, DESGARENNES Carmen Padilla, "Péptidos antimicrobianos: antibióticos naturales de la piel", Dermatología Rev Mex 2007;51:57-67.

ISDA. "Antibiotic Development: The 10 x '20 Initiative". [En línea]. 2015. [Citado 08-Sept-2016]. Disponible desde Internet en: <http://www.idsociety.org/10x20/>.

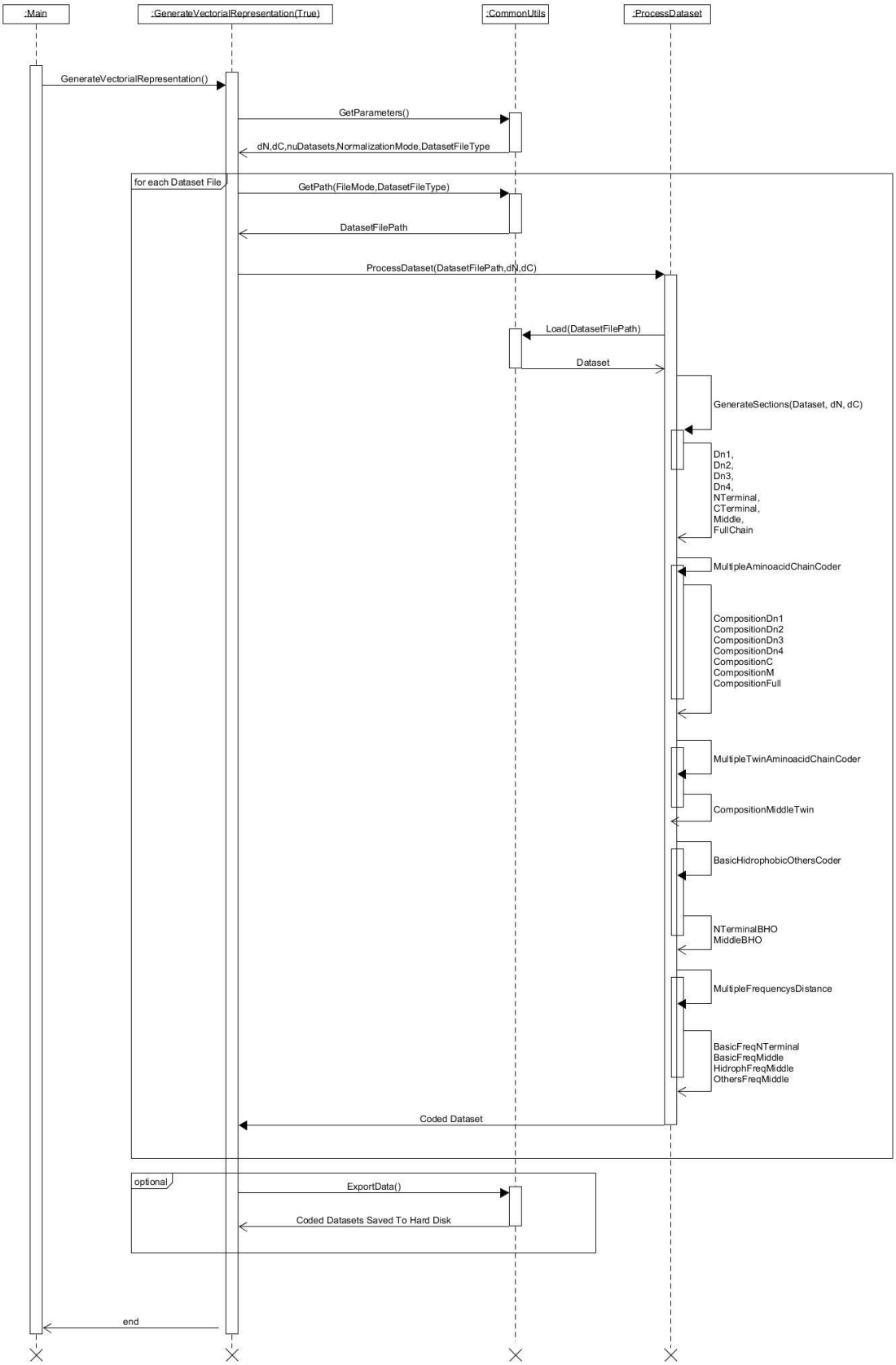
MATSUDA Setsuro, VERT Jean-Philippe, SAIGO Hiroto, UEDA Nobuhisa, TOH Hiroyuki, AKUTSU Andtatsuya, "A novel representation of protein sequences for prediction of subcellular location using support vector machines", ProteinScience (2005),14:2804–2813.

SCIKIT-LEARN, "Nearest Neighbors". [En línea]. 2015. [Citado 08-Sept-2016]. Disponible desde Internet en: < http://scikit-learn.org/stable/modules/neighbors.html

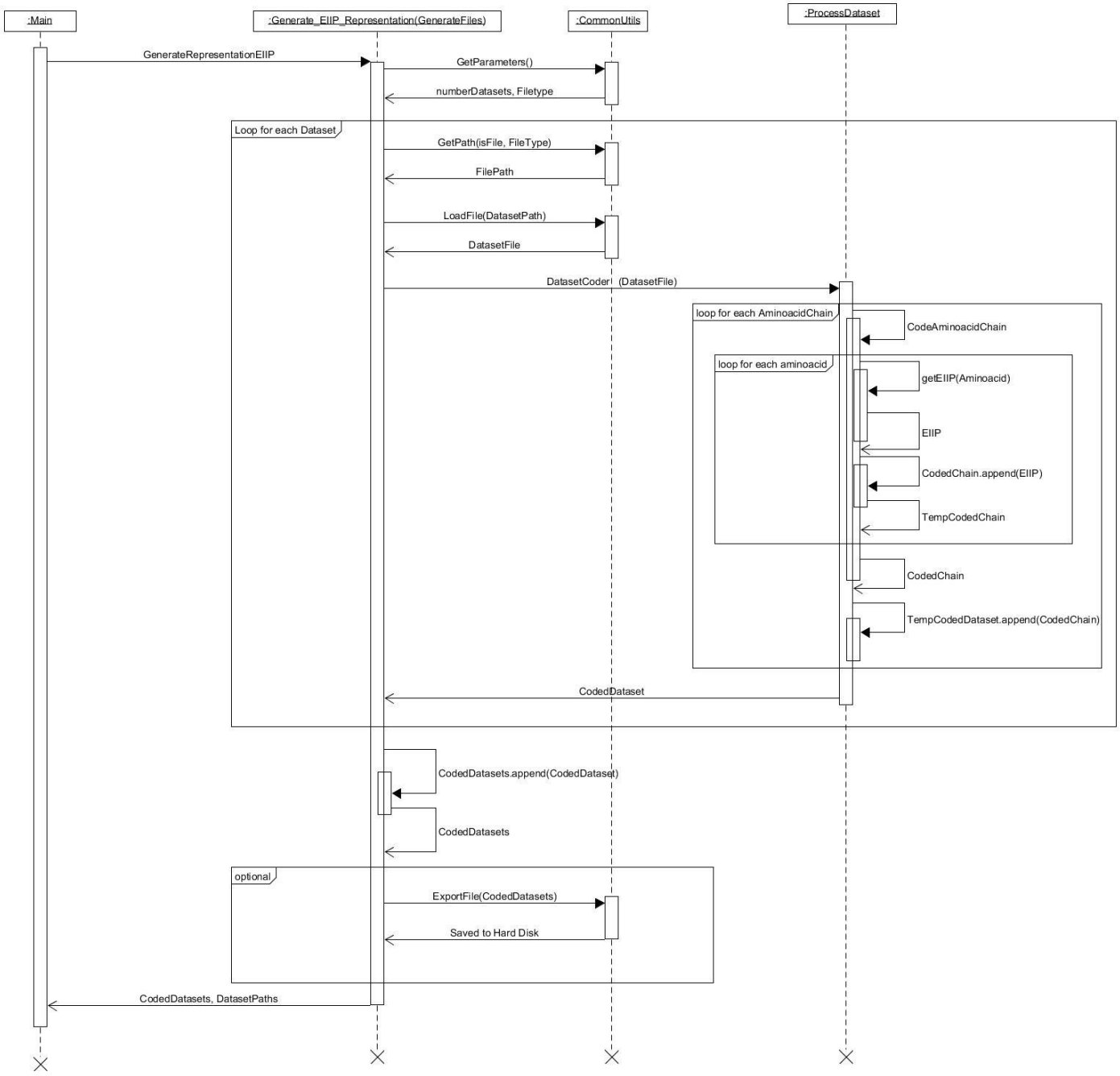
WESSELS Lodewyk F. A., REINDERS Marcel J. T., HART Augustinus A. M., VEENMAN Cor J., DAI Hongyue, HE Yudong D. and VAN'T VEER Laura J., "A protocol for building and evaluating predictors of disease state based on microarray data", Bioinformatics Vol 21



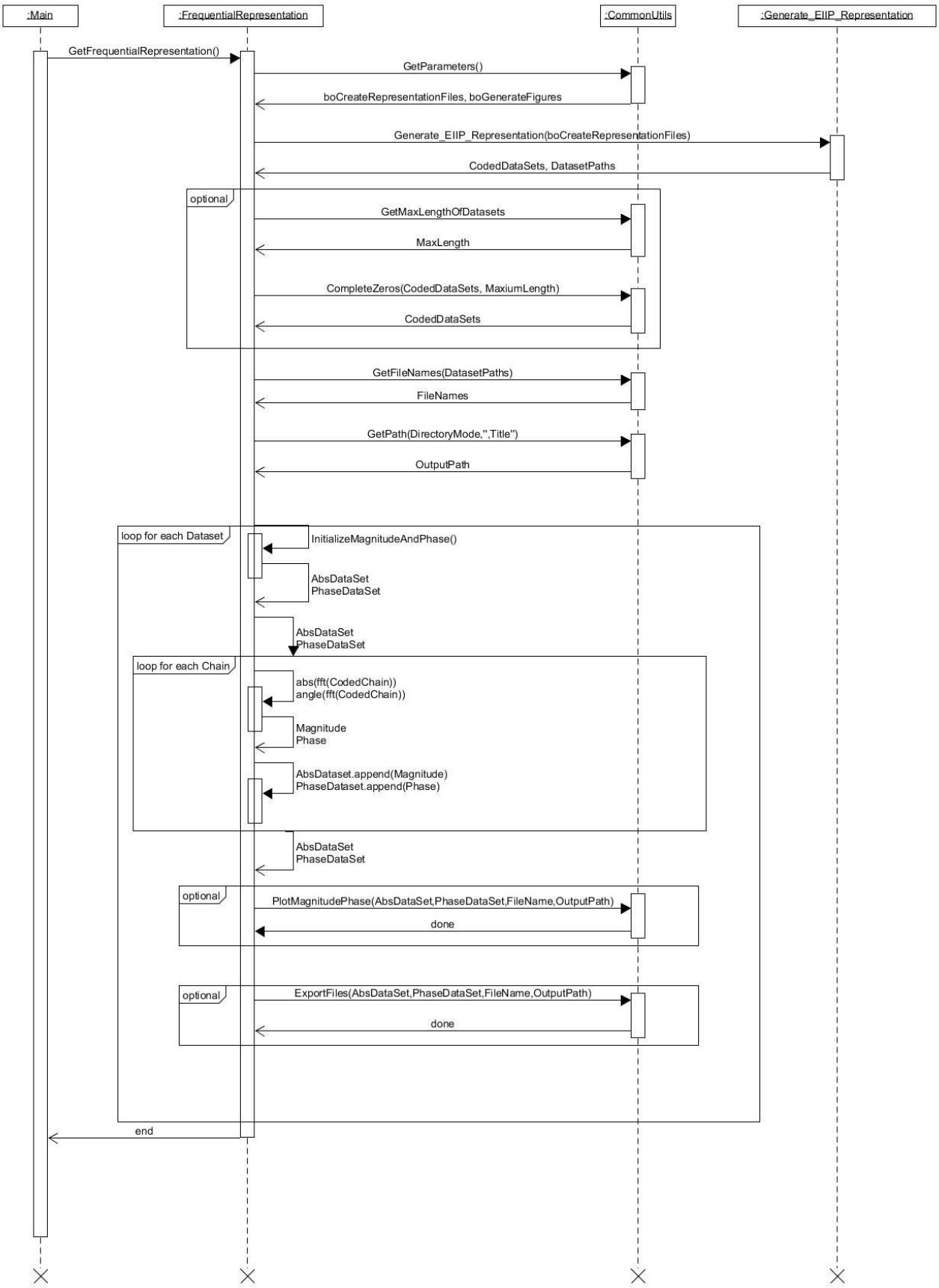
Anexo A. Proceso General de Obtención del Clasificador



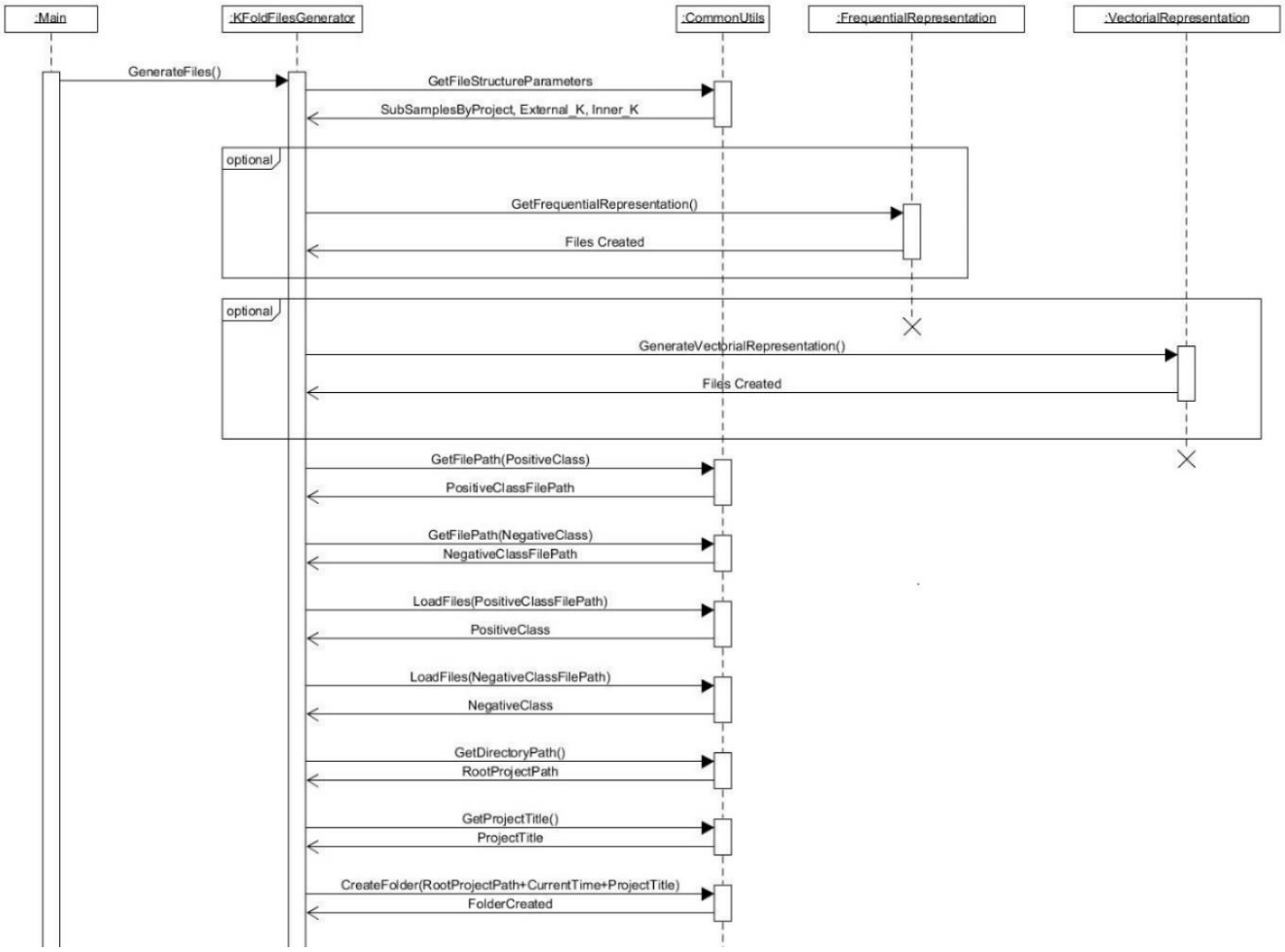
Anexo B. Diagrama de Secuencia de la Representación Vectorial



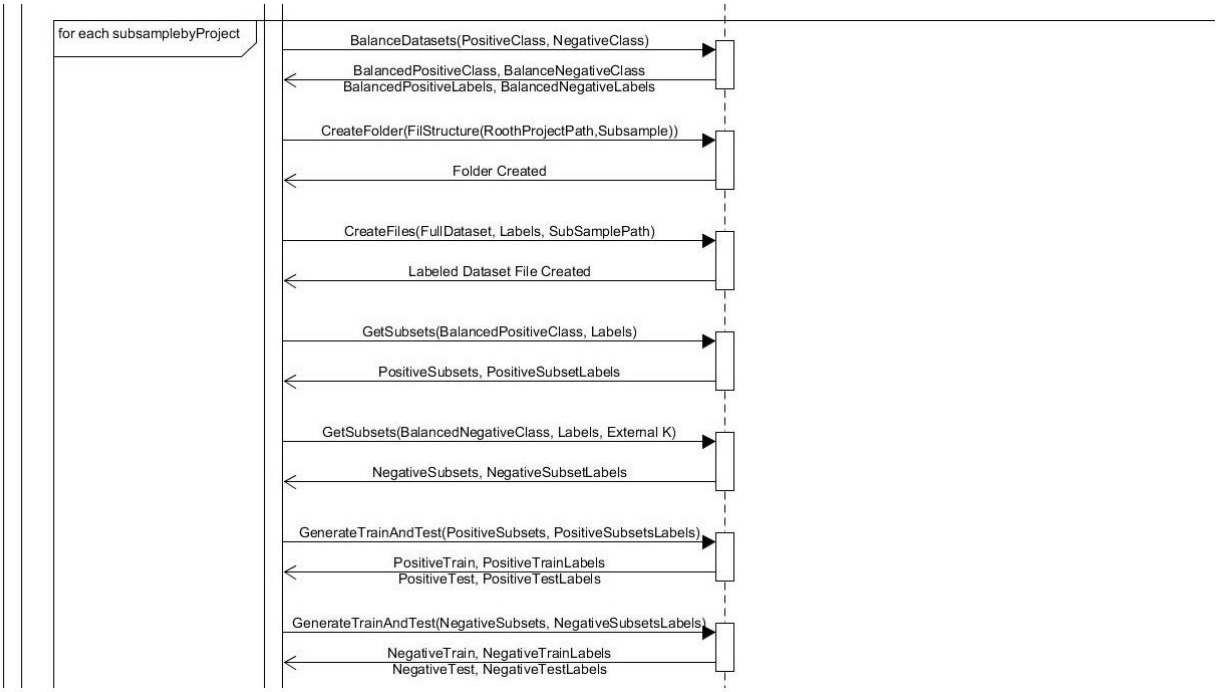
Anexo C. Obtención de representación basada en EIIP



Anexo D. Representación en Frecuencia



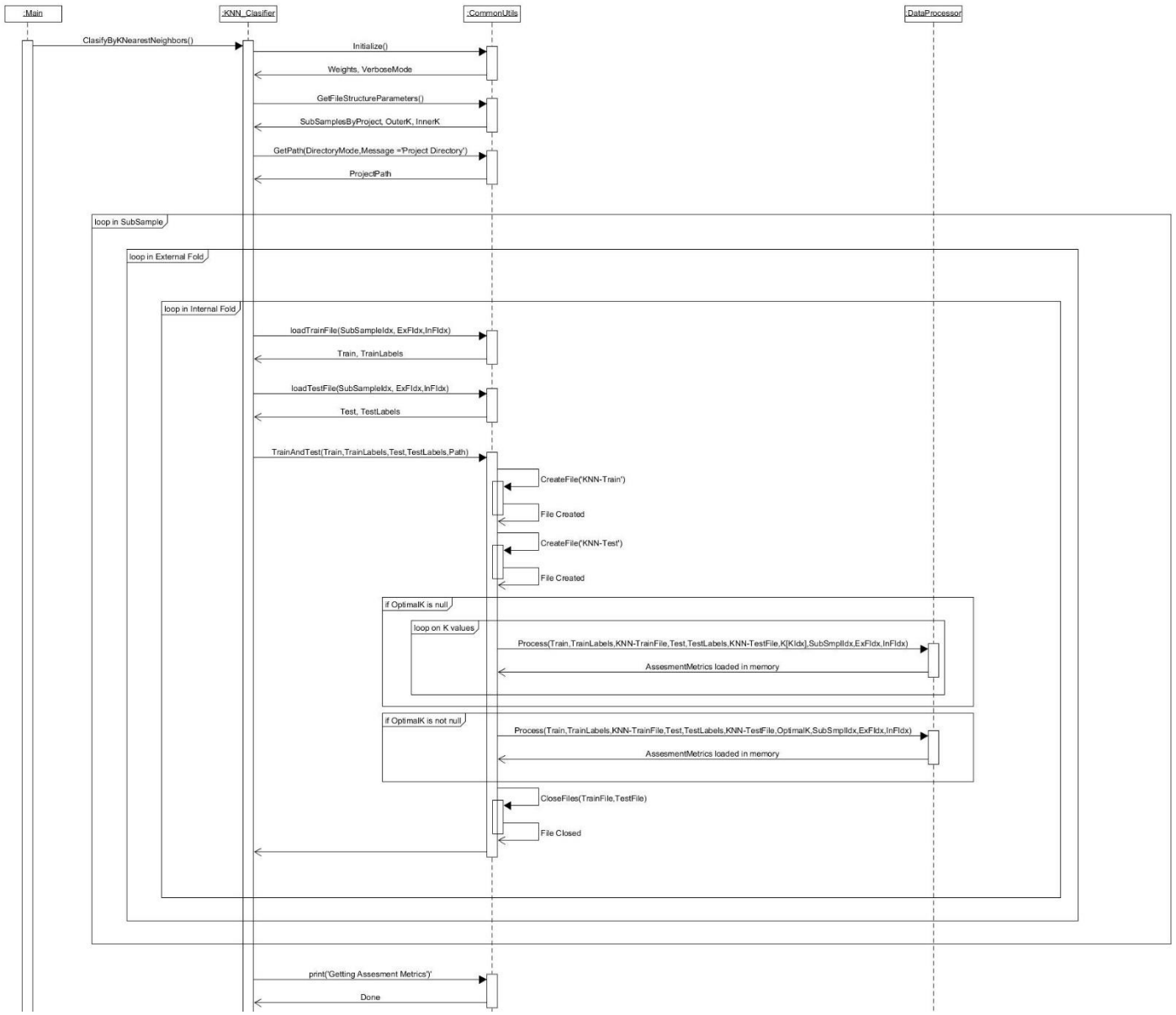
Anexo E. Generación de la estructura de archivos Parte 1



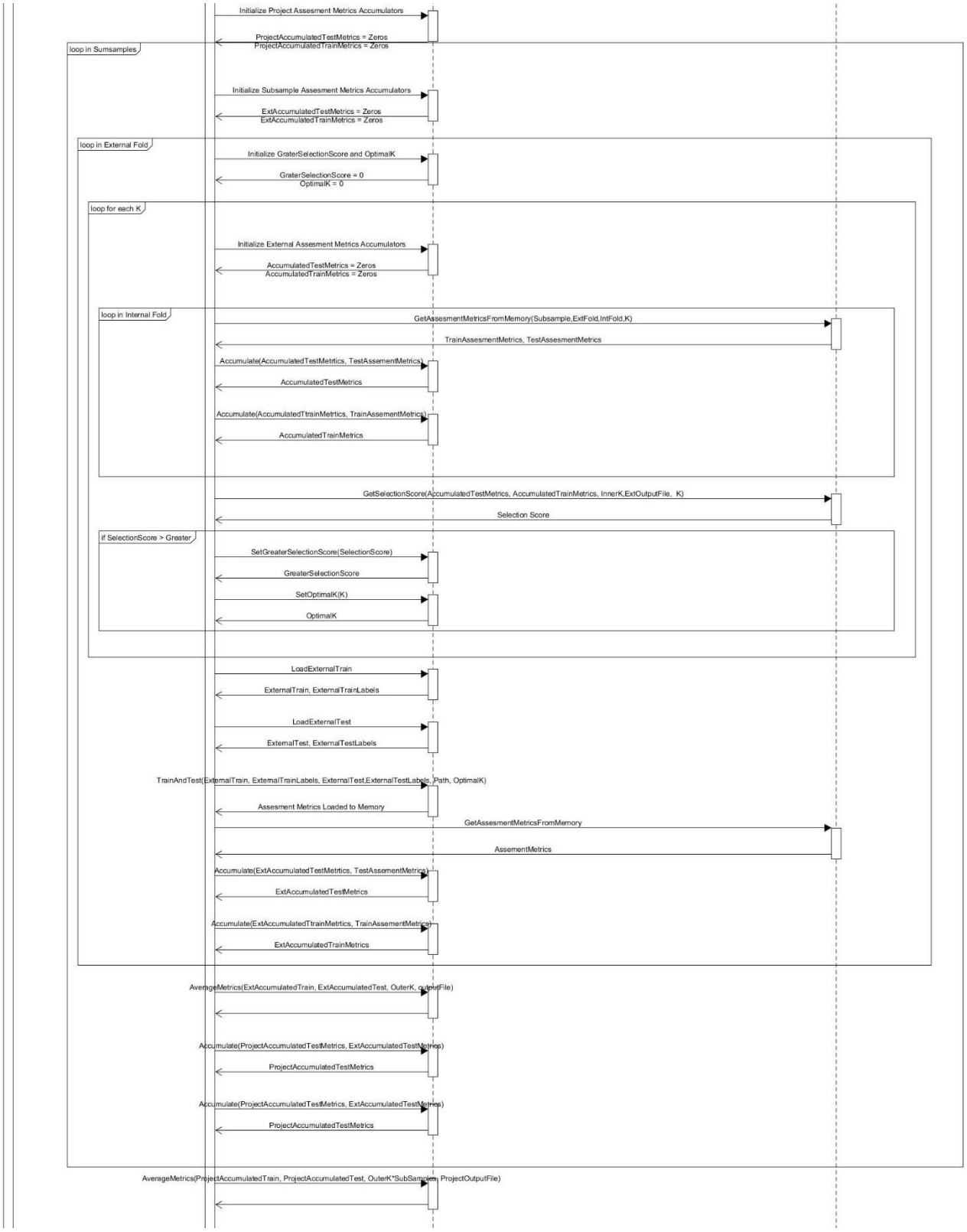
Anexo F. Generación de la estructura de archivos Parte 2

Anexo G. Generación de la estructura de archivos Parte 3

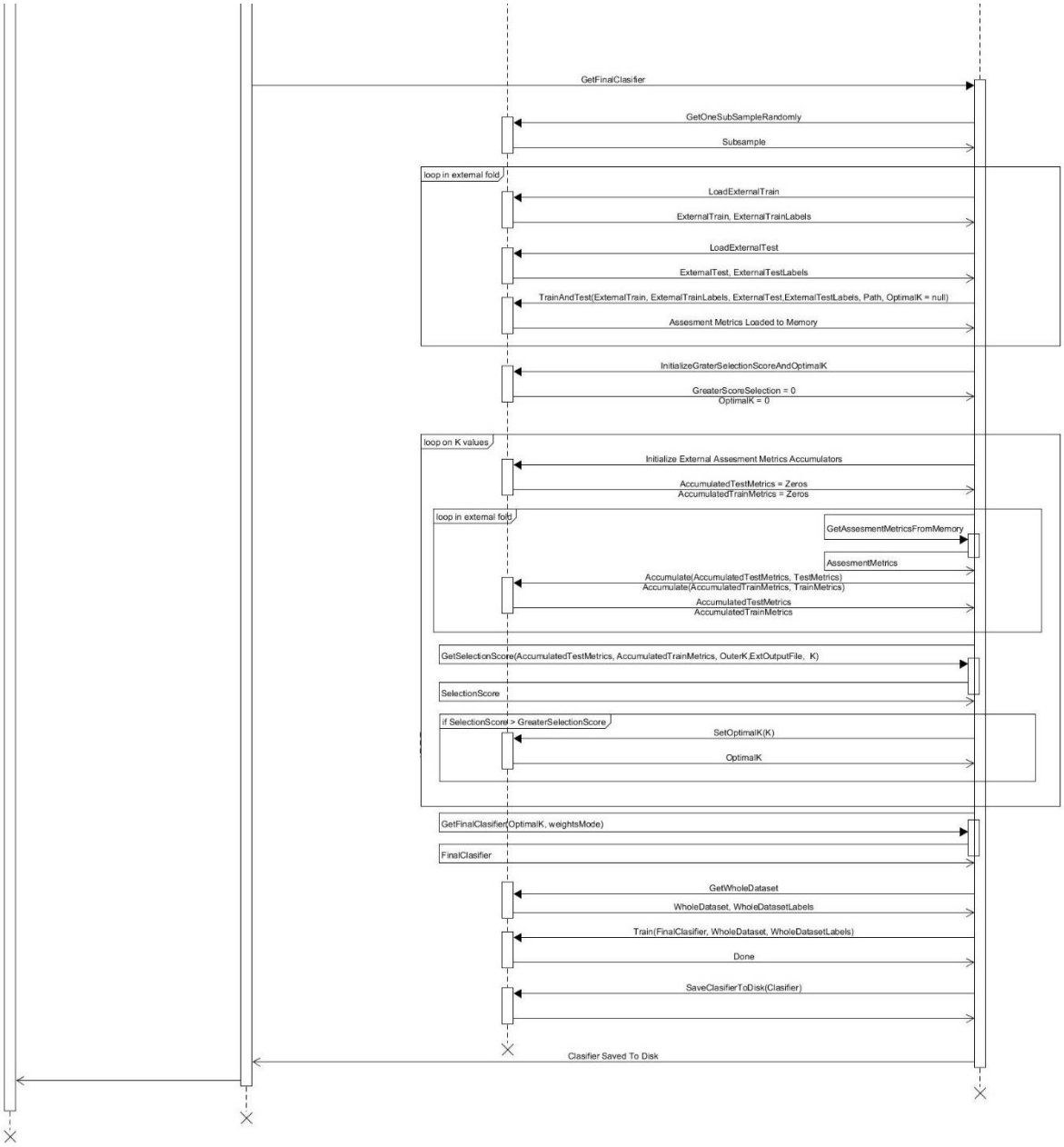
Anexo H. Generación de la estructura de archivos Parte 4



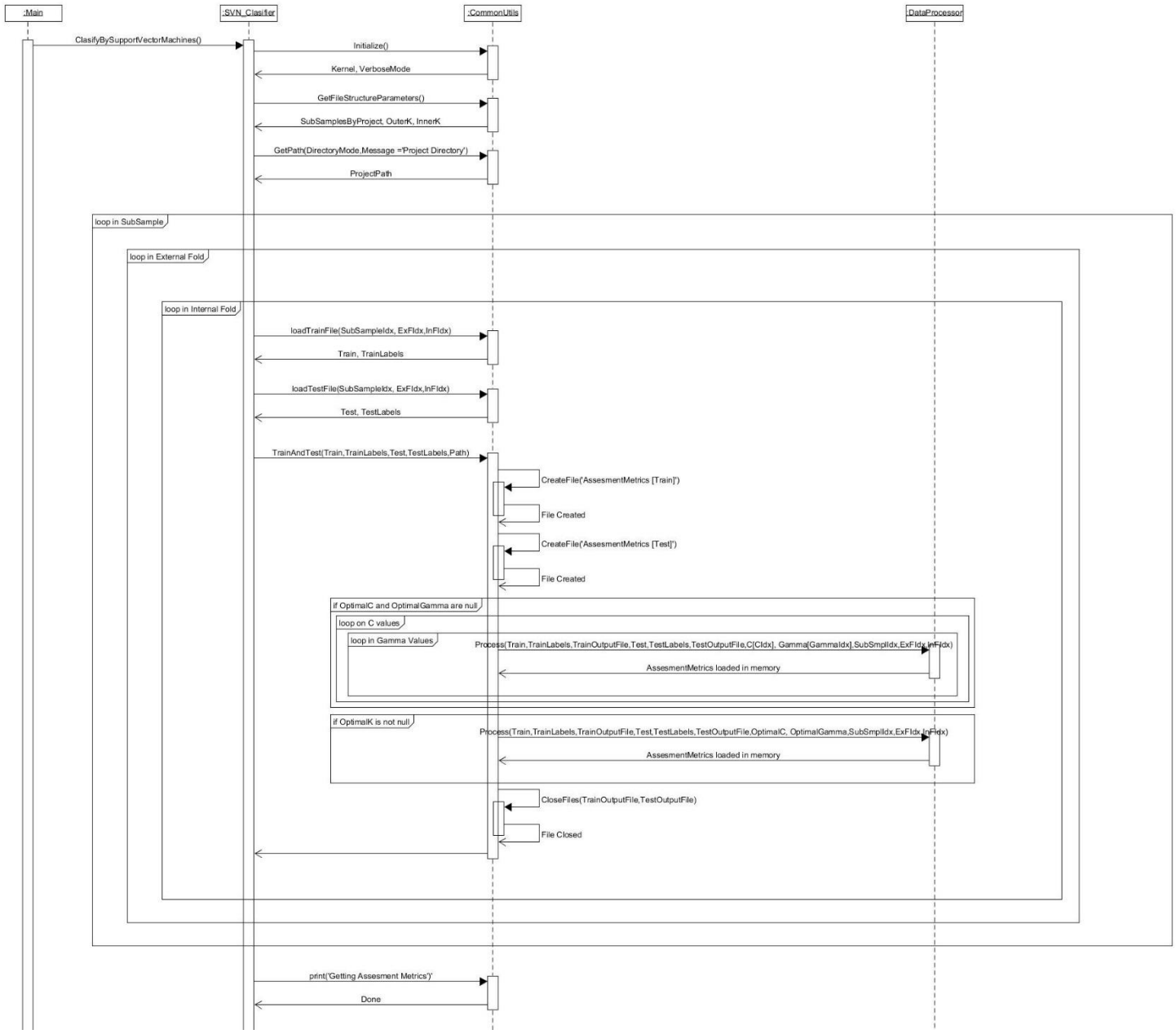
Anexo I. Validación cruzada anidada mediante KNN Parte 1



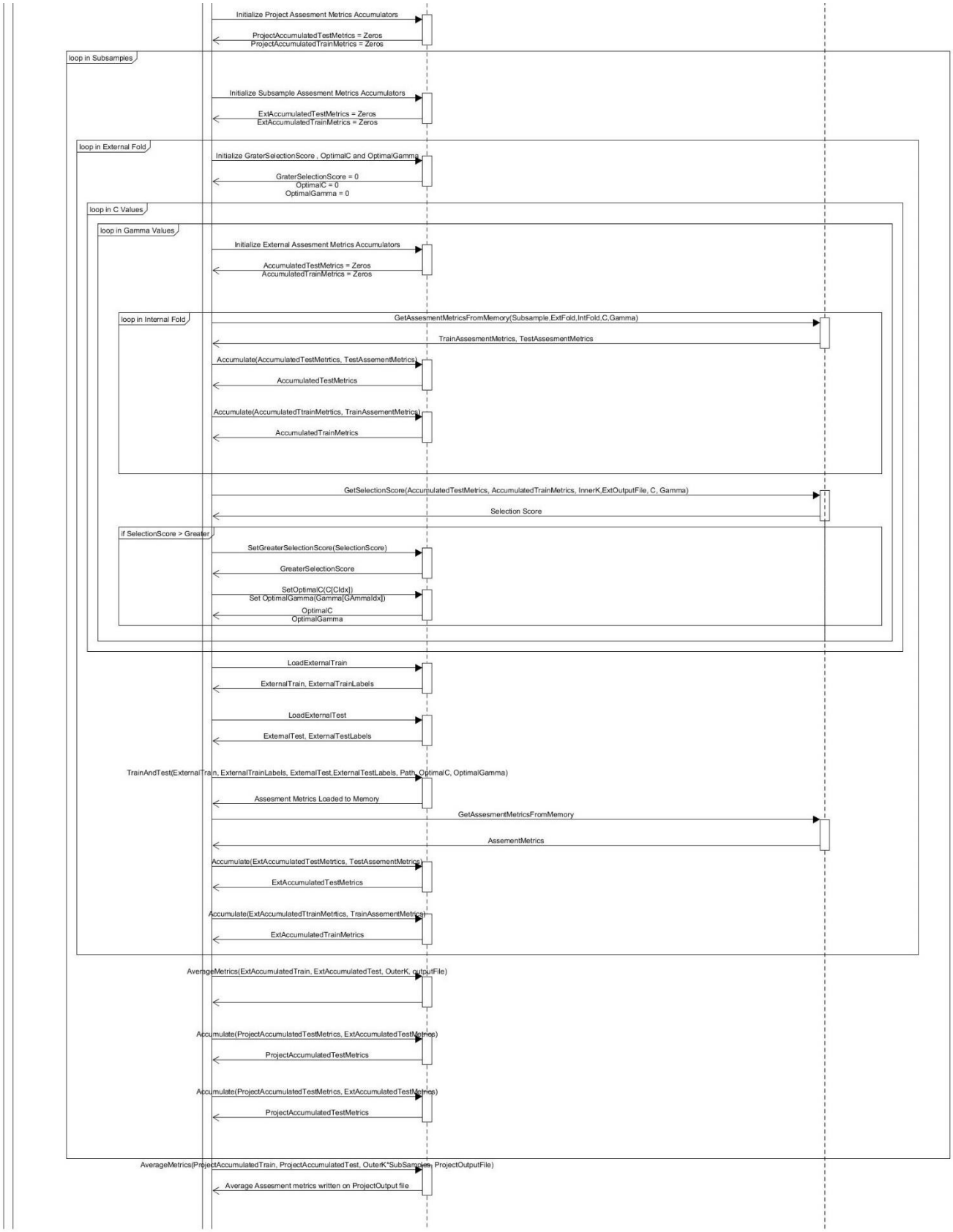
Anexo J. Validación cruzada anidada mediante KNN Parte 2



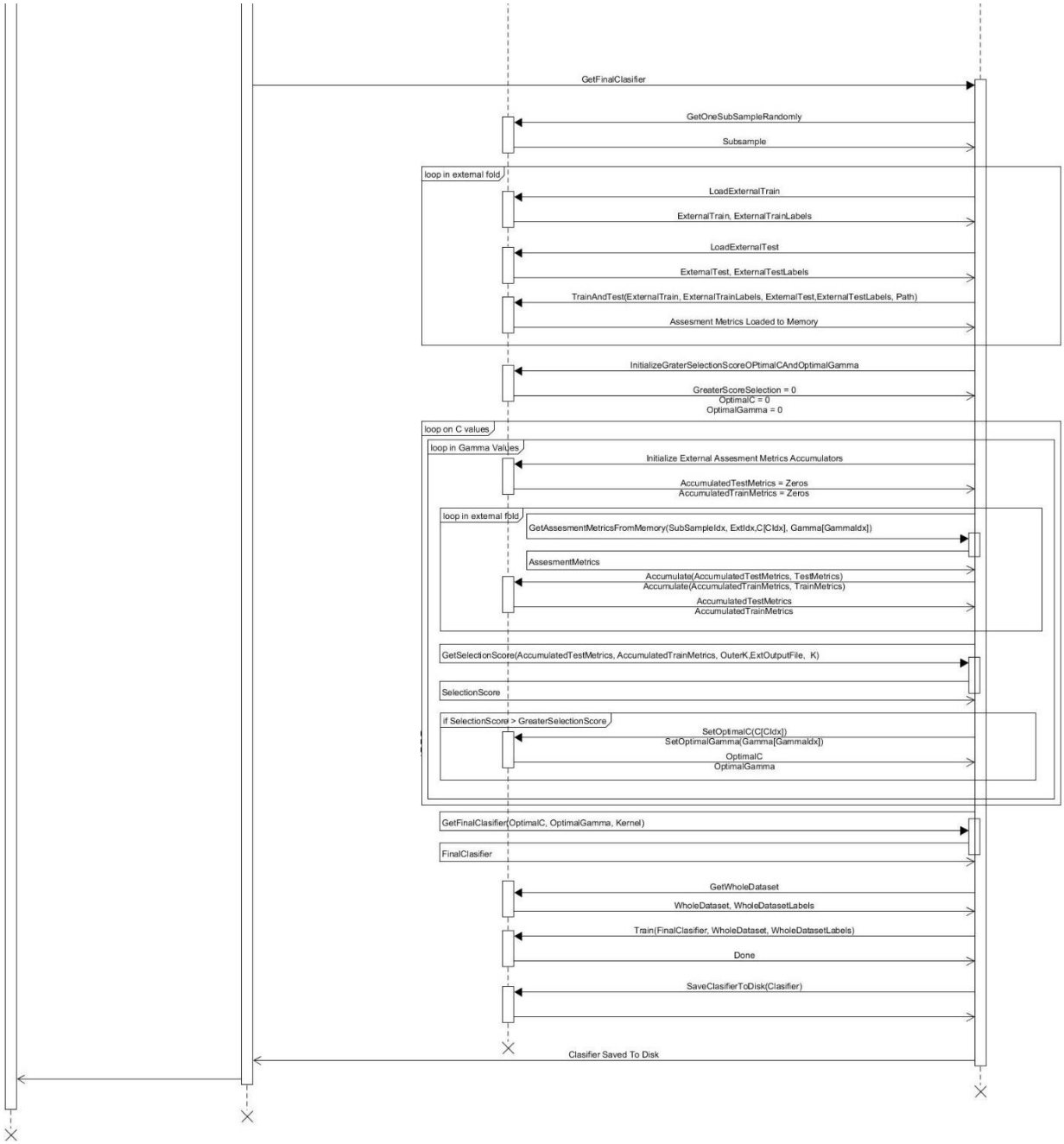
Anexo K. Validación cruzada anidada mediante KNN Parte 3



Anexo L. Validación cruzada anidada mediante SVN Parte 1



Anexo M. Validación cruzada anidada mediante SVN Parte 2



Anexo N. Validación cruzada anidada mediante SVN Parte 3