Universidad de las Ciencias Informáticas

Centro de Identificación y Seguridad Digital

Facultad 1

Predicción de resistencia a fármacos del VIH utilizando multiclasificadores

Tesis para optar por el título académico de

Máster en Informática Aplicada

Autor: **Lic. Joel Arencibia Ramírez**

Tutora: **Dra. C. Isis Bonet Cruz**

La Habana, octubre de 2011

DECLARACIÓN JURADA DE AUTORÍA

Declaro por este medio que yo Joel Arencibia Ramírez, con carné de identidad 78090314764, soy el autor principal del trabajo final de maestría “Predicción de resistencia a fármacos del VIH utilizando multiclasificadores”, desarrollada como parte de la Maestría en Informática Aplicada y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los 25 días del mes de octubre del año 2011.

_____

AGRADECIMIENTOS

...

DEDICATORIA

...

RESUMEN

Debido a la elevada capacidad de mutación del VIH, es importante determinar la resistencia que puede ofrecer la cepa del virus presente en un paciente, antes de administrarle un fármaco. Las pruebas para determinar esta resistencia son sumamente costosas o complejas, por lo que en los últimos años se han aplicado técnicas de Inteligencia Artificial para predecirla, pero sus porcentajes de predicción aún no satisfacen las necesidades en algunos casos. En este trabajo se muestra el uso de multiclasificadores para mejorar la predicción de resistencia de la proteína transcriptasa inversa, ante nueve inhibidores de la misma. Las bases de casos se construyeron utilizando la relación genotipo-fenotipo de varios mutantes del virus, disponible en la base de datos de Stanford. Para entrenar los clasificadores y multiclasificadores se utilizó una extensión de la plataforma de aprendizaje automatizado Weka. Se implementa una herramienta para transformar las bases, vistas como un problema de clasificación, a un formato .arff. Al entrenar los multiclasificadores, uno de los mayores problemas es seleccionar los clasificadores de base; para facilitar esta selección se diseñó e implementó una aplicación que permite el cálculo de diferentes medidas de diversidad, sugiriendo así aquellos clasificadores con resultados menos correlacionados entre sí. Se crearon, entrenaron y validaron varios modelos, alcanzando resultados significativamente superiores para *Multi-Expert by Hard Instances*, que supera el 97% de clasificaciones correctas en la mayoría de los casos. Los multiclasificadores entrenados, podrán utilizarse para predecir la resistencia a estos fármacos, partiendo de la información genética de nuevas cepas del virus.

Palabras clave:

clasificación, clasificadores, modelos de combinación de clasificadores, multiclasificadores, predicción de resistencia a fármacos

ABSTRACT

Considering the HIV high capacity to mutate, is important to check the resistance it can offer before giving a drug to a patient. The tests for checking this resistance are extremely expensive or complex; therefore some Artificial Intelligence techniques have been used for predicting that in the last years. Nevertheless the percentage of right predictions is not enough in some cases. This investigation shows the use of multiclassifiers for improving the prediction of resistance of the Reverse Transcriptase protein to nine inhibitors. The case bases were built by using the genotype-phenotype relationship of some virus mutants, available at Stanford database. An extension of WEKA platform was used for training the classifiers and multiclassifiers. A tool for transforming the bases, seen as a classification problem one, to "arff" format, was developed. When training the multiclassifiers, one of the biggest problems is to select the base classifiers. An application for calculating some diversity measures was designed and developed. So the classifiers with the lowest correlation results were easily identified. Some classification models were created, trained and validated, reaching the best results with *Multi-Expert by Hard Instances*, which exceed 97% of right classifications in most of the cases. The trained multiclassifiers could be used to predict HIV resistance to these nine drugs starting on the genetic information for new virus stumps.

Keywords:

classification, classifiers, ensembles, multiclassifiers, resistance to inhibitors prediction

Tabla de contenido

INTRODUCCIÓN

Los retrovirus están asociados a un amplio rango de enfermedades que incluyen tumores, síndromes de inmunodeficiencia y desórdenes neurológicos. Ellos afectan un gran número de especies. Uno de los que más afectan a la humanidad es el Virus de Inmunodeficiencia Humana (VIH). Cada año este virus causa más de 3 millones de muertes a pesar de los avances en el desarrollo de terapias para combatirlo [1].

La búsqueda de nuevos fármacos efectivos para el VIH es un gran reto para los científicos. Para poder diseñar un fármaco efectivo contra un virus como éste, el cual manifiesta alta capacidad de mutación, es necesario conocer la estructura tridimensional de todas las variantes mutantes de la proteína que se pretende atacar. El inmenso costo de la obtención de esta estructura hace que el porcentaje de secuencias que se tienen con esta información sea muy pequeño. Además de estos impedimentos, el nuevo fármaco, después de ser diseñado y sintetizado, requiere de un largo y costoso proceso de evaluación con pruebas toxicológicas y biomédicas antes de ser aprobado y salir al mercado. Por tales razones, se hace necesario continuar el estudio sobre cómo usar adecuadamente los fármacos ya existentes [2].

La mayoría de los fármacos aprobados para el tratamiento del VIH están diseñados para inhibir dos de las proteínas (enzimas) más importantes: la proteasa y la transcriptasa inversa. El fenómeno de resistencia a estos fármacos está asociado a la capacidad de mutar que presentan dichas proteínas. Cada variante mutante de estas enzimas contiene cambios de aminoácidos que pueden alteran la estructura tridimensional de las mismas. Como consecuencia de estos cambios, los fármacos no tienen acceso a los centros activos de las enzimas y no pueden inhibir sus funciones biológicas. Justamente, esto es lo que hace tan difícil la construcción de fármacos para este virus, ya que al aparecer una nueva variante del mismo, puede ser que ninguno de los fármacos diseñados anteriormente puedan acoplar en su estructura tridimensional para inhibir su función. Sin embargo, como se había dicho anteriormente, la construcción de nuevos fármacos es un proceso costoso en recursos y tiempo, por lo que el uso apropiado de los fármacos ya existente es de gran importancia. Es por esto que cuando se está en presencia de un nuevo mutante es necesario medir la resistencia que tendrá éste frente a cada fármaco para realizar una selección apropiada de los mismos.

Existen dos formas experimentales de estimar la susceptibilidad de una cepa del VIH ante un fármaco, una basada en el fenotipo y otra en genotipo. Las pruebas que se basan en el fenotipo dan una cuantificación de la sensibilidad al fármaco, generalmente expresado como la concentración requerida del fármaco para inhibir el virus. El cálculo de éstas sería ideal debido a que ofrecen, precisamente, el valor de concentración del fármaco que se necesita para inhibir el virus y, una vez conocido este valor, se analiza si es posible aplicar la concentración indicada a un humano, sin que peligre su vida. Sin embargo, las pruebas de fenotipo son muy costosas y se requiere de mucho tiempo para obtener los resultados. Por otro lado, las pruebas basadas en el genotipo están sustentadas en el análisis de las mutaciones asociadas con la resistencia, o sea, la secuencia del ADN que codifica para la enzima analizada es alineada con la secuencia correspondiente de una cepa viral tomada como referencia y se calcula la lista de posiciones mutadas. Éstas proporcionan los resultados en muy pocos días y son menos costosas; pero la interpretación de la resistencia partiendo de la información del genotipo es muy difícil, hoy en día todavía es objeto de investigación y a menudo requiere análisis de expertos.

La unión de estas pruebas de resistencia ha arrojado un gran cúmulo de información sobre este virus, y algunas de ellas se encuentran disponibles en bases de datos internacionales como "Los Álamos"[3] y "*Stanford HIV Resistance Database*"[4]. Sin embargo, el problema de relacionar el resultado de ensayos genotípicos y fenotípicos ha conducido a otros retos, pues esta combinación conlleva a interpretar el valor de fenotipo como una consecuencia de un gran número de posibles mutaciones. Tenga en cuenta que intentando minimizar el costo de los ensayos biológicos, definitivamente se está haciendo evidente un problema de una gran complejidad computacional. Para poder resolver estos problemas, especialmente el de predecir la resistencia del VIH ante inhibidores de algunas proteínas, a partir de la secuencia genómica del VIH, se ha interpretado como un problema de clasificación, donde se necesita predecir el fenotipo, partiendo de la información del genotipo. Buscando solucionar este problema se han aplicado una gran variedad de métodos estadísticos y de inteligencia artificial. Precisamente en este trabajo se pretende utilizar algunos métodos computaciones que ayuden a mejorar los resultados de predicción de la resistencia del virus del VIH, utilizando información de pares genotipo-fenotipo.

Los modelos computacionales usados hasta el momento para predecir la resistencia de fenotipo desde la información del genotipo, han ofrecido buenos resultados para algunos fármacos, pero

para otros, el porcentaje de predicción todavía es bajo. En estos problemas de análisis de secuencias una de las mayores dificultades es la diversidad de tamaño en las secuencias que se analizan, debido a las mutaciones que causan inserciones o eliminaciones de bases del ADN. Otras de las grandes dificultades son la dimensionalidad de los datos y la complejidad de la característica a predecir sólo a partir de la información de la secuencia. Típicamente en las proteínas como la transcriptasa inversa, que tiene alta variación en el tamaño de sus variantes mutacionales, por la cantidad de inserciones y eliminaciones que presenta, se utilizan métodos computacionales que analizan sólo la parte más importante de la secuencia, o sea aquella que está más cerca del centro activo.

Dentro de los métodos de inteligencia artificial que se han utilizado previamente se pueden destacar diferentes trabajos. Beerenwinkel utilizó varios métodos de aprendizaje supervisado para predecir la resistencia tanto de la proteasa como la transcriptasa inversa [5-7]. Murray propuso variantes de KNN y árboles de decisión para resolver este problema [8]. Actualmente se sigue investigando en el uso de técnicas de inteligencia artificial para mejorar los resultados alcanzados hasta el momento [9, 10].

En un trabajo de investigación previo a éste, se utilizaron modelos de redes neuronales recurrentes y multiclasificadores para predecir la resistencia de la proteasa del VIH, los resultados alcanzados fueron superiores a los obtenidos anteriormente por otros autores. Para representar las secuencias de la proteína se utilizó la energía de contacto, que es una medida que está relacionada con la estructura tridimensional. Se representó el problema como un problema de clasificación, a partir de establecer un corte para el fenotipo que permitió dividir las secuencias en resistentes y susceptibles [2].

Actualmente, debido a la cantidad de mutaciones del virus, algo que ha dado muchos resultados en pacientes es utilizar un coctel de fármacos para contrarrestar la enfermedad. A medida que al paciente se le administra mayor cantidad de medicamentos está más propenso a sufrir la cantidad de contraindicaciones que tienen los mismos. Es por esto y por todo lo explicando anteriormente sobre el costo de analizar las mutaciones, que se hace necesario un análisis de todos los fármacos que se poseen para atacar diferentes proteínas del virus.

Lo descrito antes evidencia la existencia de una **situación problémica** que se resume en:

- La búsqueda de nuevos fármacos para el tratamiento del VIH es un proceso costoso y las

numerosas pruebas a las que debe ser sometido el nuevo fármaco para permitir su uso, hacen que también tenga un alto costo de tiempo su salida al mercado, por lo que es de suma importancia probar la efectividad de fármacos existentes ante nuevas mutaciones del virus.

- El análisis de la resistencia ante un fármaco, de una determinada mutación del virus, se puede realizar mediante pruebas de fenotipo o de genotipo, que resultan o bien muy costosas o bien muy complejas, por lo que el uso de técnicas computacionales para ayudar a la predicción de la misma es de vital importancia.
- La transcriptasa inversa del VIH es una de las proteínas para las que se han desarrollado más inhibidores de su función, pero todavía no se logra un uso eficiente de éstos cuando aparecen nuevas mutaciones del virus.
- Teniendo en cuenta la información disponible en bases de datos internacionales, se han utilizado algunos modelos computacionales para predecir la resistencia ante fármacos; pero el porcentaje de predicción todavía no satisface las necesidades en algunos casos. De igual manera se han realizado trabajos anteriores con la proteasa, que han dado muy buenos resultados, pero que el simple uso de esos medicamentos no inhibe la función del virus, por lo que es necesario el estudio de resistencia para otras proteínas del mismo.

Teniendo en cuenta la situación problémica descrita, se plantea como **problema de investigación**: la efectividad de los métodos computacionales para la predicción de la resistencia de mutaciones del VIH ante inhibidores de la transcriptasa inversa no alcanza niveles adecuados en algunos casos.

Anteriormente se han desarrollados redes neuronales artificiales y modelos de combinación de clasificadores, que fueron utilizados en un trabajo previo, con buenos resultados en la predicción de resistencia de la proteína de la proteasa. En la presente investigación se define como **objeto de estudio** el proceso de predicción de la resistencia de mutaciones del VIH ante inhibidores de la transcriptasa inversa. El **campo de acción** se localiza en la Biología computacional, específicamente en el uso de multiclasificadores para predecir la resistencia del VIH ante inhibidores de la transcriptasa inversa.

Para dar solución al problema planteado se formula el siguiente ***objetivo general***: Seleccionar un modelo de combinación de clasificadores que mejore la predicción de la resistencia ante inhibidores de la transcriptasa inversa.

El objetivo general se desglosa en los siguientes ***objetivos específicos***:

- Caracterizar la utilización de métodos de inteligencia artificial para la predicción de la resistencia ante inhibidores de proteínas del VIH.
- Proponer un multiclasificador que mejore la predicción de la resistencia ante inhibidores de la transcriptasa inversa.
- Validar el multiclasificador propuesto mediante la comparación de sus resultados con los de clasificadores y otros multiclasificadores existentes.

Para dar cumplimiento a estos objetivos se trazan las siguientes ***tareas de investigación***:

1. Caracterizar la resistencia del VIH ante fármacos.
2. Identificar las principales técnicas de inteligencia artificial utilizadas en la predicción de resistencia ante fármacos del VIH y su implementación.
3. Caracterizar los modelos de combinación de clasificadores.
4. Modelar el problema de predicción de resistencia de la transcriptasa inversa del VIH y construir la base de casos.
5. Diseñar e implementar una herramienta para transformar una base de casos Stanford a una base de casos en el formato Weka.
6. Aplicar métodos de clasificación simples para predecir la resistencia del VIH ante inhibidores de proteínas.
7. Utilizar diferentes topologías de modelos de combinación de clasificadores para la predicción de resistencia del VIH.
8. Diseñar e implementar una herramienta para calcular las medidas de diversidad.
9. Seleccionar las topologías de modelos de combinación de clasificadores que mejoren los resultados para la predicción de resistencia del VIH.
10. Validar los resultados de la topología del multiclasificador propuesto, a partir de la comparación de los resultados de validación cruzada y mediante técnicas estadísticas.

Se plantea como ***hipótesis de investigación*** la siguiente: *"La correcta selección del método de combinación de clasificadores y de los clasificadores de base permite mejorar la predicción de la resistencia ante inhibidores de la transcriptasa inversa del VIH"*.

Para dar cumplimiento al objetivo propuesto se utilizaron los métodos y técnicas siguientes:

Métodos teóricos:

- **Hipotético-deductivo:** para la elaboración de la hipótesis central de la investigación y

para proponer nuevas líneas de trabajo a partir de los resultados obtenidos.

- **Análisis-síntesis:** permite descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la propuesta de solución.
- La **modelación**: en la estructuración del multiclasificador propuesto y de las herramientas para el cálculo de las medidas de diversidad y para la transformación de la base de casos Stanford al formato Weka.

Dentro de los **métodos empíricos** utilizados se encuentran el **análisis documental** empleado en la revisión bibliográfica que permitió caracterizar entre otros elementos, los esquemas de combinación de las salidas de clasificadores de base y el **método experimental** para probar los resultados que arrojan los multiclasificadores para distintas combinaciones de clasificadores de base y seleccionar el que ofrezca mejores resultados en cada caso.

Se utilizaron como **métodos de la estadística no paramétrica** el **Test de Friedman** y el **Test de Wilcoxon**. El Test de Friedman se utilizó para comparar las muestran dependientes, teniendo en cuenta que se utilizan las mismas bases de casos. El Test de Wilcoxon se utilizó para comparar muestras dependientes dos a dos.

El trabajo se estructura en introducción, tres capítulos, conclusiones, recomendaciones y un cuerpo de anexos. En el capítulo 1 se realiza un análisis del problema de la resistencia del VIH ante fármacos, se caracterizan algunos algoritmos basados en casos y su utilización para predecir la resistencia del VIH ante inhibidores de proteínas.

En el capítulo 2 se caracterizan los sistemas de combinación de clasificadores; además se explica la relación entre la selección de los clasificadores de base y las medidas de diversidad; así como el diseño e implementación de dos herramientas: una que facilita el cálculo de estas medidas y otra que permite la transformación de una base de casos Stanford, al formato exigido por la herramienta Weka.

En el tercer capítulo se muestran los resultados de la predicción de resistencia del VIH utilizando modelos de combinación de clasificadores, se explica cómo se modeló el problema de predicción de resistencia del VIH y cómo se construyó la base de casos. Posteriormente se exponen los resultados de la utilización de los modelos de clasificación simples para predecir la resistencia del VIH y de aplicar diferentes topologías de modelos de combinación de clasificadores para esta

tarea. Se muestra la utilización de las medidas de diversidad para seleccionar los clasificadores de base. Por último se justifica la selección del multiclasificador propuesto.

El aporte del trabajo se concreta en la propuesta de un multiclasificador que mejora la predicción de la resistencia ante cada uno de los inhibidores de la transcriptasa inversa estudiados. Además se proponen dos herramientas, una para el cálculo de algunas medidas de diversidad que facilita la selección de los clasificadores de base de una manera conveniente y otra para la transformación de la base Stanford a una base en formato Weka.

CAPÍTULO 1. ANÁLISIS DEL ESTADO DEL ARTE PARA PROBLEMAS DE CLASIFICACIÓN DE RESISTENCIA DEL VIH

En el presente capítulo se explica el papel de la proteína transcriptasa inversa en la infección del VIH, así como el fenómeno de la resistencia que ofrecen las diferentes cepas del virus del VIH a los fármacos utilizados para su tratamiento. Se describen las pruebas experimentales que se realizan con el objetivo de determinar la resistencia ante fármacos. Además se exponen las principales técnicas de inteligencia artificial que pueden utilizarse para resolver el problema de la determinación de resistencia del VIH ante fármacos y algunos ejemplos de su aplicación.

1.1. Resistencia del VIH ante fármacos

El Virus de Inmunodeficiencia Humana (VIH) es un retrovirus con alta capacidad de mutación que afecta considerablemente a la humanidad, a pesar de los avances en el desarrollo de terapias para combatirlo. La obtención de nuevos fármacos para el tratamiento del VIH es un proceso costoso y prolongado. Por estas razones se hace necesario el estudio de la resistencia a los fármacos ya existentes, con el objetivo de utilizar de manera adecuada cada uno o combinaciones de ellos, ante la aparición de nuevas mutaciones del virus.

1.1.1. Infección del VIH

Los virus son un complejo macromolecular formado por proteínas y ácidos nucleicos, en el cual las proteínas se agrupan formando una cápside que protege al material genético, teniendo una estructura que puede ser más o menos compleja en dependencia del virus. Existe una enorme variedad de estructuras genómicas entre las especies de virus que, como grupo biológico, contienen una diversidad genómica superior a la de los reinos de plantas, los animales o las bacterias. En general, el genoma de los virus puede clasificarse en monocatenario o bicatenario, de ARN o ADN, y que pueden utilizar o no la transcriptasa inversa para el proceso de replicación. En particular los retrovirus, nombre que reciben los virus que pertenecen a la familia *Retroviridae*, tienen una forma de replicación característica en el interior de las células huéspedes. Como otros grupos virales, los retrovirus contienen un núcleo constituido por ácido nucleico ARN, en lugar de contener ADN. Pero a diferencia de otros virus con ARN, cuando los retrovirus se replican en el interior de las células, lo hacen como genomas de ADN y esto es posible gracias a que poseen la enzima transcriptasa inversa [11].

En la actualidad, el Virus de Inmunodeficiencia Humano (VIH) es uno de los virus que más afecta la humanidad causando más de tres millones de muertes anuales. El VIH es un lentivirus de la familia *Retroviridae*. Fue descubierto y considerado como el agente de la naciente epidemia de Síndrome de Inmunodeficiencia Adquirido por el equipo de Montagnier en Francia en 1983 [12].

Un virión (organismo) del VIH tiene forma esférica y su diámetro mide aproximadamente 100 nanómetros. Está dotado de una envoltura (una membrana que originalmente pertenecía a la célula de donde el virus emergió) donde se encuentran 72 prolongaciones formadas por las glicoproteínas gp120 y gp41 que actúan en el momento de la unión del virus a la célula hospedadora. La capa intermedia es una cápside proteica que protege al material genético localizado en la capa interior (Ver Figura 1.1) [11].

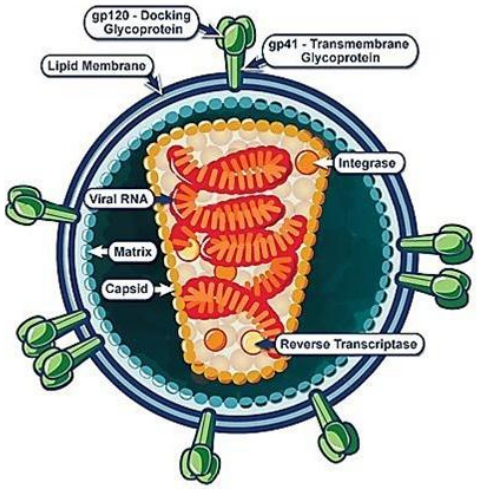

Figura 1.1 Estructura de un virión del VIH. Tomado de [13].

La cadena genética del VIH está constituida por un ARN de cadena simple compuesto por dos filamentos idénticos. El ARN contiene varios genes, cada uno de los cuales codifica las diversas proteínas que el VIH necesita para reproducirse, entre ellas la proteasa y la transcriptasa inversa.

1.1.2. Función de la transcriptasa inversa y sus inhibidores

Las proteínas son largas cadenas de aminoácidos (unos mil como promedio), siendo estas cadenas tridimensionales en plantas y animales, pero casi lineales en bacterias. Hay veinte aminoácidos distintos que se combinan para formar las proteínas, lo cual explica la gran cantidad, variedad y clases de proteínas existentes. Normalmente los virus son organismos que contienen varias proteínas con funciones distintas y los tratamientos antivirales se dirigen a algunas de ellas, cuya inhibición pueden evitar su función o reproducción.

En particular la transcriptasa inversa es una enzima de tipo ADN-polimerasa, que tiene como función sintetizar ADN de doble cadena utilizando como molde ARN, es decir, transcribe una sola cadena de ARN en una sola cadena de ADN. Cuando el VIH infecta una célula, hace una copia de su propio código genético dentro del ADN de la célula (Ver Figura 1.2). De esta manera la célula queda "programada" para crear nuevas copias del VIH. El virus infecta a los seres humanos gracias a esta enzima. Sin la transcriptasa inversa, el genoma viral no sería capaz de incorporarse en la célula huésped, a causa de la incapacidad de estos para replicarse [14]. Esta aseveración reafirma el hecho de que varios fármacos para el tratamiento del VIH estén dirigidos a inhibir las funciones de dicha enzima.

Los fármacos antirretrovirales son medicamentos para el tratamiento de la infección por retrovirus (en particular el VIH) que actúan en diferentes etapas de su ciclo vital. Dado que la vida media del virus es corta y las copias de ADN son muy variadas por la gran cantidad de errores de transcripción de RNA a DNA, existe una alta tasa de mutación. Las combinaciones de antirretrovirales actúan incrementando el número de obstáculos para la mutación, manteniendo bajo el número de copias virales. Los agentes antirretrovirales individualmente no suprimen la infección por VIH a largo plazo por lo cual deben usarse en combinaciones. Las combinaciones de tres o cuatro fármacos antirretrovirales se conoce como Terapia Antirretroviral de Gran Actividad o TARGA [15].

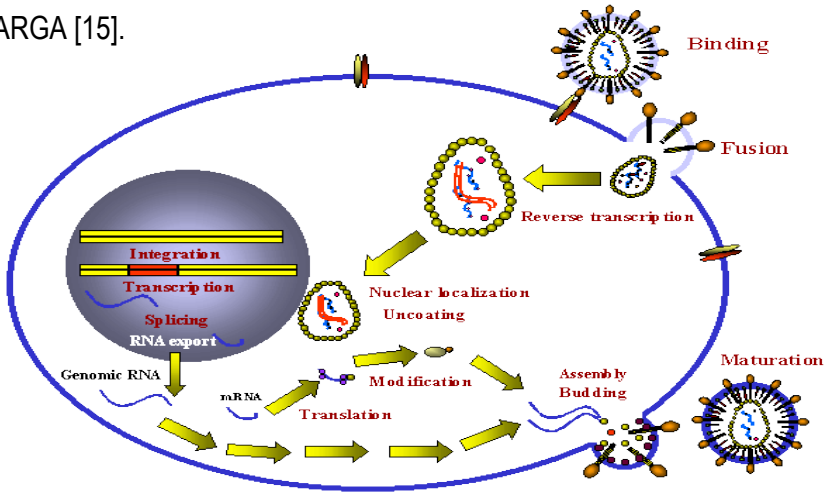

Figura 1.2 Ciclo de vida del VIH. Tomado de [16].

Para inhibir las funciones de la transcriptasa inversa existen dos tipos fundamentales de fármacos: los inhibidores nucleótidos o nucleósidos (*Nucleoside analog reverse-transcriptase inhibitors*, NRTIs) y los inhibidores no nucleósidos (*Non-nucleoside reverse-transcriptase inhibitors*, NNRTIs). Los NRTIs fueron los primeros fármacos aprobados para el tratamiento de la

infección por VIH. Estos causan la terminación prematura de la cadena de ADN bloqueando la transcriptasa inversa, lo que constituye un paso temprano en el ciclo de replicación viral cuando el ADN del virus se produce a partir del ARN viral en una célula recién infectada. La clase de los NRTIs incluye medicamentos como la zidovudina (AZT), didanosina (DDI), zalcitabina (DDC), estavudina (D4T), lamivudina (3TC), abacavir (ABC), emtricibina (FTC) y tenofovir (TDF). Aunque con un mecanismo ligeramente diferente, los NNRTIs también tienen como objetivo bloquear este paso en el proceso de replicación del virus. En la clase de los NNRTI se pueden encontrar fármacos como nevirapina (NVP), delavirdina (DLV) y efavirenz (EFV) [17].

La ya comentada capacidad de mutación de virus produce variantes mutantes de la transcriptasa inversa, que contienen cambios de aminoácidos, alterando la estructura tridimensional de las mismas. Como consecuencia de estos cambios los fármacos no tienen acceso a los centros activos de las enzimas y no pueden inhibir sus funciones biológicas, fenómeno conocido como resistencia.

1.1.3. Pruebas de Resistencia

Antes de suministrar un fármaco a un paciente es necesario saber qué variantes del virus posee en su organismo y en base a esto investigar cuál sería el fármaco, o la combinación de fármacos, más recomendado para su tratamiento.

Existen dos formas experimentales de estimar la susceptibilidad de una cepa del VIH ante un fármaco: las pruebas fenotípicas y las pruebas genotípicas. Las pruebas basadas en el fenotipo dan una cuantificación de la sensibilidad de la cepa del virus al inhibidor, generalmente expresada como la concentración requerida del fármaco para inhibir el virus. Las pruebas basadas en el genotipo están sustentadas en el análisis de las mutaciones asociadas con la resistencia, o sea, la secuencia del ADN que codifica para la enzima analizada es alineada con la secuencia correspondiente de una cepa viral tomada como referencia (secuencia salvaje) y se calcula la lista de posiciones mutadas [18].

Los ensayos genotípicos sólo proveen pistas hacia la resistencia a la droga, o sea, el genotipo intenta establecer la presencia o ausencia de mutaciones genéticas de una proteína del VIH que ha sido previamente asociada con la resistencia a la droga. Sin embargo, la interpretación de la resistencia a drogas partiendo solamente de la información del genotipo a menudo requiere análisis de expertos. Por otro lado las pruebas fenotípicas proveen una cuantificación directa de

la resistencia a la droga, pero requieren un gasto económico muy alto [18].

Las pruebas genotípicas son menos costosas pero todavía cuestan entre $300 y $600. Están más ampliamente disponibles y los resultados por lo general regresan en una o dos semanas. Sin embargo, estas pruebas también pueden requerir que sus resultados sean interpretados por un experto. Obtener los resultados de las pruebas fenotípicas lleva tiempo puesto que se realizan con cada medicamento, lo que además implica que sea más costoso. Aunque los resultados son más fáciles de interpretar puede tardar varias semanas en obtenerse. Puesto que estas pruebas no son utilizadas de manera tan rutinaria como las genotípicas, podría ser difícil encontrar un laboratorio que las realice [19].

La unión de estas pruebas de resistencia han arrojado un gran cúmulo de información sobre este virus, la cual se encuentra disponible en bases de datos internacionales como "Los Álamos"[3] y "*Stanford HIV Resistance Database*"[4]. En el presente trabajo se han seleccionado los datos de esta última base como punto de partida para la construcción de la base de conocimiento que será utilizada para el entrenamiento y validación de los multiclasificadores propuestos. Esta base tiene información referente a nueve de los fármacos más conocidos desarrollados para inhibir la transcriptasa inversa: AZT, DDI, D4T, 3TC, ABC, TDF, NVP, DLV y EFV.

1.2. Métodos de Clasificación

Los problemas pueden dividirse, teniendo en cuenta si tenemos conocimiento de la función o hipótesis que se desea predecir, en problemas supervisados y no supervisados. Los problemas supervisados son aquellos donde se tiene información de la hipótesis y los no supervisados donde no se tiene información de la misma. Cuando la función o hipótesis a predecir es continua, los algoritmos relacionados con los problemas supervisados son conocidos como métodos de regresión y cuando la hipótesis es discreta se conocen como métodos de clasificación o clasificadores. Este trabajo se centra en estos últimos.

La información que manejan los clasificadores se encuentra almacenada en una base de conocimiento, que va a estar conformada por un conjunto de objetos, casos o instancias. Estos objetos pueden ser representados como vectores $X=(x_1,x_2,...,x_N)$, donde cada x_i es un atributo o rasgo que lo caracteriza y N es la cantidad de atributos. Luego, una base de conocimiento es un conjunto de objetos etiquetados, donde cada uno tiene la clase correspondiente. Es decir, a cada vector $(x_1, x_2,..., x_N)$ se le hace corresponder una clase $c_j \in \Omega$, donde $\Omega=\{c_1,c_2,...,c_k\}$ y k es la

cantidad de clases del problema. En [2] se define formalmente un clasificador como una función objetivo: $F : \Re^n \rightarrow \Omega$, que a cada *N*-uplo de atributos *X* asocia la clase a la cual debe pertenecer.

La diferencia de los clasificadores está en la manera de representar *F* y de buscar el espacio de todas las posibles hipótesis. A la etapa de construcción del clasificador en la que éste aprende la relación entre los objetos y la clase, se le llama *entrenamiento*. Después de representado el problema, seleccionado el modelo de clasificador y entrenado el mismo, sólo restaría comprobar que el clasificador tiene potencialidad de generalización. Para ello se utilizan casos nuevos, cuya clase es conocida, pero que no fueron usados en el entrenamiento del mismo, para verificar su efectividad. A esto se le denomina fase de *prueba*, *validación o evaluación del clasificador*.

A continuación se describen algunos de los métodos de clasificación más usados: *k*-vecinos más cercanos, árboles de decisión, redes bayesianas, redes neuronales artificiales y máquinas de soporte vectorial.

1.2.1. Algoritmos Basados en Casos

Todos los métodos de clasificación utilizan razonamiento basado en casos, que permite resolver problemas nuevos a partir de experiencias viejas; no obstante, existe un grupo de ellos que se conocen como algoritmos basados en casos, cuyo entrenamiento es simplemente el almacenamiento de los casos y no necesitan crear reglas, ni árboles, ni ajustar parámetros. El tipo de aprendizaje que utilizan estos algoritmos es conocido como perezoso.

Estos algoritmos necesitan de la definición de una medida de distancia para comparar cada nueva instancia con las de la base de conocimientos. Luego, en la determinación de la clase a la que pertenece una nueva instancia X_p, puede buscarse la instancia X_q de la base que esté más cercana a ella, según la medida de distancia, para asignarle la clase de X_q a la instancia X_p. Éste se conoce como método del vecino más cercano.

A menudo se usa más de una instancia cercana y la clase mayoritaria es la asignada al nuevo caso. Si en lugar de usar el caso más cercano utilizamos los *k* casos más similares, entonces hablamos de *k*-vecinos más cercanos y la clase asignada a la nueva instancia será la más común entre las *k* instancias más cercanas encontradas en la base de casos. Este algoritmo se conoce como *k*NN por sus siglas en Inglés(*k* *NearestNeighbors*) [20, 21].

1.2.2. Árboles de Decisión

El aprendizaje usando árboles de decisión es un método para aproximar funciones de valores

discretos. Los árboles de decisión pueden también ser representados como conjuntos de reglas if-then (si-entonces).

Un árbol de decisión es un grafo acíclico donde cada nodo especifica una prueba de algún rasgo y cada arco que sale del nodo corresponde a alguno de los valores posibles del rasgo que representa ese nodo. Estos árboles clasifican casos no conocidos comenzando por la raíz del árbol, probando el rasgo especificado en el nodo y en dependencia del valor de ese rasgo se mueve al próximo nodo. Este proceso es repetido entonces hasta alcanzar un nodo hoja o terminal que define la clasificación [21].

Un árbol de decisión clasifica las instancias ordenándolas de la raíz a las hojas. Cada nodo interior del árbol especifica una prueba de algún atributo y las hojas son las clases en las cuales se clasifican las instancias, cada rama descendiente de un nodo interior corresponde a un valor posible del atributo probado en ese nodo. Un árbol de decisión representa una disyunción de conjunciones sobre los valores de los atributos. Cada rama, de la raíz a un nodo hoja, corresponde a una conjunción de atributos y el árbol en sí, a una disyunción de estas conjunciones [22].

El algoritmo clásico para la construcción de un árbol de decisión es el ID3[23]; pero el mismo presenta algunas limitaciones como la posible aparición de una sobreestimación en el caso de una cantidad de casos pequeña o el hecho de que no maneja atributos continuos, solo discretos. El mismo autor, en [24], propone el algoritmo C4.5 como variante para resolver estas limitaciones, que usa puntos de corte e introduce varias medidas para evitar el sobreentrenamiento, en particular los criterios de parada de la división y de poda del árbol. Este algoritmo se basa en la utilización del criterio razón de ganancia. De esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección. Además el algoritmo C4.5 incorpora una poda del árbol de clasificación una vez que este ha sido inducido. La poda está basada en la aplicación de una prueba de hipótesis que trata de responder a la pregunta de si merece la pena expandir o no una determinada rama.

Hoy en día existen muchos otros algoritmos e implementaciones, de construcción de árboles de decisión; CHAID (*Chi-square Automatic Interaction Detector*), *Exhaustive* CHAID, QUEST, CRT; y también existen muchos otros criterios de parada o poda. Ellos han sido utilizados con bastante eficiencia en la solución de problemas bioinformáticos [25].

1.2.3. Redes Bayesianas

Una red bayesiana es un modelo gráfico probabilístico que representa un conjunto de variables y sus dependencias probabilísticas. Las redes Bayesianas permiten declarar supuestos de independencia condicionales que son aplicados a subconjuntos de variables. Son representadas por un gráfico acíclico, donde cada variable es representada por un nodo de la red, y de ella se especifican dos tipos de información, la estructura de dependencias condicionales que son los arcos de la red y las distribuciones de probabilidad correspondientes. Una red Bayesiana puede calcular la distribución de probabilidad para cualquier subconjunto de variables de la red dado los valores o distribuciones de las variables restantes [21].

Existen algoritmos de "propagación de evidencias" que facilitan calcular la probabilidad de una conclusión, sobre cualquier variable a partir de ciertas evidencias. Cuando no se conocen todos los valores de las variables en el conjunto de entrenamiento, el aprendizaje con una red bayesiana puede ser más difícil. Este problema es análogo a la búsqueda de pesos en una red neuronal. Se han propuesto varios algoritmos de entrenamiento para estos casos como el llamado K2, que usa un algoritmo de ascenso de colinas (*hill climbing*) restringido por un orden sobre las variables [26]. En [21, 27] se describen otros algoritmos que se han desarrollado con este propósito como por ejemplo *tree augmented Naive Bayes.*

Estas redes pueden ser usadas para inferir un valor objetivo dado los valores observados de otras variables, y recíprocamente, inferir el valor probable de una variable, a partir de la evidencia de otras y/o del valor objetivo. Ello constituye el principal mérito de las redes bayesianas. Por otro lado, este tipo de clasificador no es muy sensible a los cambios de sus parámetros, ya que se basa en información de toda la base, lo cual hace que pequeños cambios en la base no sean necesariamente significativos.

1.2.4. Máquinas de Soporte Vectorial

Máquina de soporte vectorial o máquina de vectores de soporte (*Support Vector Machine,* SVM) es una técnica de aprendizaje supervisado que se desarrolló en los últimos años pero muy rápidamente, partiendo de la teoría de aprendizaje estadístico y basada en el principio de minimización de riesgo estructural. Concretamente, fundamenta las decisiones de clasificación, no basadas en todo el conjunto de datos, sino en un número finito y reducido de casos, que constituyen los "vectores soporte". Se ha usado tanto para clasificación (aprendizaje supervisado

con función objetivo discreta), como para regresión (aprendizaje supervisado con función objetivo continua). Puede dividirse en SVM lineal y no lineal, basado este último en diferentes funciones núcleo (*kernel*) [2].

En el caso del SVM lineal, éste construye un hiperplano *n*-dimensional de separación en el espacio y selecciona el hiperplano, de tal forma que la distancia desde los ejemplos más cercanos al hiperplano sea máxima [28]. En el caso de la clasificación no lineal la idea es similar, excepto que se realiza una transformación no lineal del conjunto de entrenamiento, o sea, el conjunto de puntos originales es remplazado por los obtenidos con una función núcleo, de forma que se fije el hiperplano en el espacio de rasgos transformados. Hay que tener en cuenta que para que una función pueda ser considerada función núcleo es necesario ante todo que sea simétrica y semidefinida positiva [2].

Si la función núcleo usada es una de las denominadas gaussiana de base radial (g*aussian radial basis*), el espacio de rasgos correspondiente es un espacio de *Hilbert* de dimensión finita. Los clasificadores de margen máximo son bien regulados, así la dimensión infinita no arruina los resultados. Algunas de las funciones núcleo más comúnmente usadas son [2]:

Polinomial:

$$k(x, x') = \langle x \cdot x' \rangle^d$$

Gaussiana de base radial:

$$k(x, x') = \exp\left(\frac{\|x - x'\|}{2\sigma^2}\right)$$

1.2.5. Redes Neuronales Artificiales

Las redes neuronales son herramientas matemáticas para la modelación de problemas, que permiten obtener las relaciones funcionales subyacentes entre los datos involucrados en problemas de clasificación, reconocimiento de patrones, regresiones, etc. Son consideradas excelentes aproximadores de funciones esencialmente no lineales, siendo capaces de aprender las características relevantes de un conjunto de datos, para luego reproducirlas en entornos ruidosos o incompletos [29].

Una red neuronal puede definirse como un conjunto de unidades computacionales llamadas neuronas, interconectadas por medio de arcos pesados a manera de grafo dirigido. El objetivo de tal red, que puede verse como una caja negra, es calcular una salida Y a partir de una información X recibida con anterioridad. Usualmente las redes reciben la información proveniente del exterior mediante un conjunto de neuronas de entrada y cuentan con un conjunto distinto, llamado neuronas de salida, para ofrecer los resultados. El resto de las neuronas se organizan en

capas ocultas. Se concibe el cálculo general de la red a partir de la información que es procesada por cada una de sus neuronas en forma independiente. Cada una de ellas puede recibir información de las restantes y calcular su propia salida a partir de dicha entrada y de su estado actual, transitando eventualmente hacia un nuevo estado. Por lo general, el flujo de cálculo de la red avanza progresivamente desde las neuronas de entrada hacia las neuronas de salida, en un proceso en el que cada una de las neuronas ocultas va activándose progresivamente según el esquema de conexión particular de cada red [30].

Una red neuronal puede ser caracterizada por el modelo de la neurona, el esquema de conexión que presentan sus neuronas, o sea su topología, y el algoritmo de aprendizaje empleado para adaptar su función de cómputo a las necesidades del problema particular. Se ha producido una amplia variedad y clasificación de topologías de redes neuronales, que pueden agruparse en dos grandes grupos: las redes multicapa de alimentación hacia delante (*Feed-Forward Neuronal Networks*, FFN) y las redes neuronales recurrentes (*Recurrent Neuronal Networks*, RNN) [2].

Los algoritmos de entrenamiento constituyen métodos que se aplican sobre los modelos de red para ajustar sus pesos y obtener un comportamiento determinado. Con frecuencia los algoritmos de entrenamiento son caracterizados por la clase de topologías sobre las que se aplica, los tipos de parámetros libres que afecta (pesos de las conexiones entre neuronas, parámetros del algoritmo de entrenamiento, la topología misma de la red, etc.) y la regla de modificación de los mismos. Existe una amplia variedad de algoritmos de entrenamiento disponibles, y generalmente se clasifican en supervisados o no supervisados.

Las redes *Multilayer Perceptron* (MLP) constituyen un ejemplo genérico de las redes FFN. Frecuentemente se encuentran formadas por un conjunto de capas de neuronas ordenadas secuencialmente por: una capa de entrada, un conjunto de capas intermedias denominadas capas ocultas y una capa de salida.

El modelo precursor de las redes MLP fue el *Perceptron* de Rosemblatt. Esta red no tenía capas ocultas por lo que la principal diferencia entre ella y el MLP reside en la potencia de cómputo: el *Perceptron* sólo es capaz de separar linealmente (por un hiperplano) las entradas; mientras que el MLP, usando neuronas ocultas con funciones no lineales, es capaz de aproximar cualquier tipo de función continua y brindar excelentes resultados en las tareas de clasificación [29].

Entre las técnicas de entrenamiento supervisado más difundidas se encuentran aquellas que

basan su funcionamiento en el método del gradiente descendente. Entre ellas, el algoritmo de propagación del error hacia atrás (*Backpropagation*, BP), es el de más amplio uso, aplicado a redes con conexiones hacia adelante. Este algoritmo aplica la técnica gradiente descendente para la minimización del error de funcionamiento de la red [31].

Las redes neuronales recurrentes poseen una diferencia notable con la clase anterior: admiten conexiones hacia atrás, o sea, pueden formar ciclos en el grafo que describe sus conexiones. Estas conexiones hacia atrás, llamadas de retroalimentación, son las que permiten que la red sea capaz de guardar una memoria de los estados anteriores para su uso en el cálculo de las salidas del estado actual, o sea, mantener una especie de memoria de los procesamientos pasados; ésta es la característica esencial que convierte a este tipo de redes en una herramienta de amplio uso en tareas de reproducción de señales y análisis de secuencias, donde se reflejan relaciones causales en el tiempo y el espacio respectivamente. De manera general son aplicables a problemas reales que reflejan relaciones dinámicas y estructurales de orden complejo [32, 33].

1.2.6. Aplicación de los métodos de clasificación para predecir la resistencia del VIH ante inhibidores de proteínas

En la literatura se proponen diversos modelos de aprendizaje automatizado para la predicción de resistencia a partir del genotipo. Por ejemplo, en [8] se utilizan árboles de decisión y *k*NN; en [5] se muestra el uso de SVM; en [34] usan redes neuronales artificiales. Además, en [35] se utilizan redes neuronales artificiales en la predicción de resistencia de la Proteasa ante el fármaco lopinavir (LPV).

Por otro lado, en [36] se analiza el uso de árboles de decisión, regresión lineal, análisis discriminante, máquinas de soporte vectorial y redes neuronales en la predicción de resistencia con una selección de los mejores casos de cada base de fármacos disponible en *Stanford HIV Resistance Database*. Por su parte, Cao [37] y Rabinowitz [38] realizan una comparación de varios de estos métodos en la predicción de la resistencia al VIH, resultando para algunos fármacos un clasificador mejor y para otros, otro.

En [2] se proponen dos métodos de clasificación para predecir la resistencia a siete inhibidores de la proteasa a partir de la información disponible en *Stanford HIV Resistance Database*, que parten del uso de las energías de contacto asociadas a cada aminoácido y consideran como clase la resistencia o susceptibilidad ante fármacos. El primero está basado en redes neuronales

recurrentes bidireccionales con un módulo que combina las salidas de los diferentes tiempos en la red. El segundo combina varios modelos de clasificación a partir de un metaclasificador, obteniendo resultados satisfactorios para los inhibidores estudiados.

Otros resultados más recientes, muestran el uso de diferentes tipos de clasificadores en la predicción de resistencia a fármacos del VIH y cómo todavía sigue siendo un reto mejorar los resultados de predicción. Algunos ejemplos de estos se muestran a continuación. En [39] se propone un modelo basado en reglas para la predicción de ocho inhibidores de la transcriptasa inversa, en este enfoque se utiliza el método de selección de atributos de Monte Carlo. En [10] se analizan 6 inhibidores de la transcriptasa inversa: ABC, 3TC, D4T, AZT, TDF y NVP, utilizando árboles de decisión y se obtienen redes de interdependencia entre algunos rasgos físico-químicos de la enzima, que influyen en la resistencia. En este trabajo también se utiliza el método de selección de atributos de Monte Carlo. En [9] se utiliza SVM en la predicción de resistencia al NVP, obteniendo un 90% de efectividad.

1.2.7. Evaluación en la clasificación

Con el objetivo de validar la efectividad de un clasificador y dar una idea de cuan efectivo es el mismo en la solución de un problema determinado al que se quiere aplicar, han surgido varias medidas para evaluar la clasificación y comparar los modelos empleados para dicho problema. Las medidas más conocidas para evaluar la clasificación están basadas en la matriz de confusión que se obtiene cuando se prueba el clasificador en un conjunto de datos que no intervienen en el entrenamiento. En un problema de dos clases, donde C1 es la clase negativa y C2 la clase positiva, la matriz de confusión obtenida sería la que se muestra a continuación:

	Clase obtenida	
Clase Real	C_1	C_2
C_1	TN	FP
C_2	FN	TP

Tabla 1.1 Matriz de confusión para un problema de clasificación con dos clases

TP y TN son la cantidad de elementos bien clasificados de la clase positiva y negativa, respectivamente. FP y FN son la cantidad de elementos negativos y positivos mal clasificados, respectivamente. Basadas en estos cuatro valores, se calculan diferentes medidas, tales como el error, la exactitud (*accuracy*), la razón de TP (*TP rate*) o sensitividad, la razón de FP (*FP rate*), la precisión y especificidad, que se dan por las expresiones siguientes:

$$Error = \frac{FP + FN}{TP + TN + FP + FN} \qquad Exactitud = 1 - Error$$

$$Razón\ de\ TP = Sensitividad = \frac{TP}{TP + FN} \qquad Razón\ de\ FP = \frac{FP}{FP + TN}$$

$$Precisión = \frac{TP}{TP + FP} \qquad Especificidad = \frac{TN}{TN + FP}$$

Otra forma de evaluar el rendimiento de un clasificador es por el análisis de la llamada *Receiver Operator Curve* (ROC) [40]. En esta curva es representado el valor de razón de TP vs. la razón de FP, mediante la variación del umbral de decisión. Se denomina umbral de decisión a aquel que decide si una instancia *x*, a partir del vector de salida del clasificador, pertenece o no a cada una de las clases. Usualmente, en el caso de dos clases se toma como umbral por defecto 0.5; pero esto no es siempre lo más conveniente. Se usa el área bajo esta curva (*Area Under the Curve*, AUC) como un indicador de la calidad del clasificador. En tanto dicha área esté más cercana a 1, el comportamiento del clasificador está más cercano al clasificador perfecto (aquel que lograría 100% de TP con un 0% de FP).

Estas medidas no son suficientes, por sí solas, para evaluar un clasificador. La manera en que se dividen los datos en conjunto de entrenamiento y prueba es también muy importante. Existen varias técnicas para hacer esto, la más vieja y simple se basa en entrenar y probar el clasificador con la misma base de conocimiento. Este método puede traer como consecuencia un sobre-aprendizaje del clasificador, o sea, que el clasificador más que generalizar el conocimiento de los datos, aprenda estos "de memoria", pues no se tiene en cuenta cómo reacciona el modelo ante casos que no ha visto antes. Otro método usado es el método *H* (*Hold-out*), que divide la base a la mitad, una mitad para entrenamiento y otra para prueba. Una versión de éste es el *data shuffle* que realiza *n* veces el método *H* y promedia los resultados. Este método no es efectivo para bases pequeñas, pues los ejemplos pudieran no ser representativos si se diera el caso de que los casos no fueran divididos convenientemente. El *método de validación cruzada con k subconjuntos* (*k-fold cross-validation*) es uno de los más usados, este método se basa en dividir la base en *k* segmentos y realizar *k* procesos de entrenamientos y pruebas, de forma que el proceso *i* toma el segmento *i* para prueba y el resto para entrenamiento. Tiene la ventaja que todos los ejemplos de la base de casos son, eventualmente usados para ambos procesos, entrenamiento y prueba. Como inconveniente podemos señalar que en grandes volúmenes de

datos la validación se haría muy demorada. El método *boostrap* se basa en la generación de *n* conjuntos de cardinalidad *l* desde el conjunto de datos original, con reemplazo [41].

Actualmente se usa también en vez de dos conjuntos de datos, tres: uno para entrenamiento, uno para prueba y un tercero para validación. Este último se usa como seudoentrenamiento, de tal manera que el proceso de entrenamiento se detiene cuando comience a decrecer el rendimiento sobre el conjunto de validación, aunque continúe aumentando sobre el conjunto de entrenamiento [42]. Este método es muy útil para evitar el sobreentrenamiento. También se usa para ajustar parámetros y seleccionar un modelo apropiado. Tiene como desventaja que necesita un conjunto de datos muy grande.

La mejor forma de organizar el experimento en entrenamiento y prueba realmente depende de las características de la base de conocimientos. En nuestra investigación, se utilizará el método de validación cruzada con *k*-subconjuntos, teniendo en cuenta que el tamaño de las bases no es muy grande y que este método nos permite utilizar cada caso tanto para entrenamiento como para prueba. Por otro lado, como medida de eficiencia utilizaremos la exactitud.

1.3. La plataforma para aprendizaje automatizado: Weka

La experiencia ha demostrado que no existe un esquema de aprendizaje apropiado para todo tipo de problemas. La existencia de un clasificador por excelencia, es una fantasía. Para cada problema es importante la selección del clasificador más apropiado, de manera experimental. En este sentido, puede resultar de gran importancia el uso de la plataforma Weka. La plataforma Weka, cuyo nombre proviene de *Waikato Environment for Knowledge Analysis,* es un paquete de software libre, desarrollado en la Universidad de Waikato, Nueva Zelanda. Está desarrollado en el lenguaje de programación Java y se distribuye bajo los términos de la licencia GNU. Corre prácticamente sobre cualquier plataforma y ha sido probado en los sistemas operativos Linux, Windows y Macintosh [27].

Weka incluye métodos para diferentes problemas de minería de datos: regresión, clasificación, clusterización. De este modo, implementa diferentes algoritmos de aprendizaje automatizado, incluyendo todos los analizados en el presente capítulo. El hecho de que provee una interfaz uniforme para acceder a los mismos, permite que sus usuarios puedan comparar los resultados de aplicar diferentes métodos a un problema en cuestión y seleccionar aquellos que sean más apropiados para dicho problema.

Debido a su interfaz gráfica, Weka permite fácilmente el preprocesamiento de los datos, introducirlos en un esquema de aprendizaje y analizar el resultado de la clasificación y su eficiencia. El Explorador (*Explorer*) posibilita leer el conjunto de datos de un archivo de extensión ".arff" y luego va forzando al usuario a realizar las acciones en el orden correspondiente, para seleccionar el método a aplicar, visualizar los resultados, etc. Como parte de su menú para problemas de clasificación, permite utilizar diferentes formas de evaluación, como el método de validación cruzada con *k*-subconjuntos, descrito en el epígrafe anterior. Por su parte, el Experimentador (*Experimenter*) permite el diseño de pruebas con mayor grado de complejidad. Por lo general, la mayor parte de los usuarios prefiere el uso del Explorador, al menos cuando trabaja por primera vez con la plataforma Weka [27].

La ventaja de ser de código abierto, dota a Weka de la posibilidad de ser extensible. De este modo, su desarrollo ha dejado de concentrarse en el lugar donde fue creado. Su estructura bien organizada en paquetes, facilita el trabajo de aquellos desarrolladores que deseen agregar nuevos modelos y algoritmos a la plataforma, o modificar los existentes, de manera ordenada. En [2] aparece, de manera detallada, la metodología para adicionar un nuevo clasificador a Weka, de manera que se garantice una total compatibilidad con la herramienta y su correcto funcionamiento.

Los archivos de las bases de casos que utiliza Weka tienen extensión ".arff" y deben cumplir con la estructura mostrada en la figura 1.3, de no cumplirse con ella, se notifica la incompatibilidad.

Encabezado	**@RELATION** <nombre de la base de casos>
Declaración de atributos	**@ATTRIBUTE** <nombre atributo$_1$> <TIPO$_1$> … **@ATTRIBUTE** <nombre atributo$_n$> <TIPO$_n$>
Datos	**@DATA** <valor$_{1,1}$>,…,<valor$_{1,n}$> … <valor$_{m,1}$>,…,<valor$_{m,n}$>

Figura 1.3 Estructura de las bases de casos que utiliza Weka.

La estructura definida por Weka no constituye un estándar, en ocasiones es necesario utilizar bases de casos que no cumplen con ella, lo cual exigirá transformar la base de casos original.

Como se dijo antes, Weka implementa todos los algoritmos de clasificación analizados previamente en este capítulo. Para utilizar uno de ellos, una vez seleccionada la base de casos

en la ficha de preprocesamiento, debe seleccionarse el clasificador en cuestión en la ficha de clasificación. Los clasificadores se encuentran agrupados en carpetas según su tipo, por ejemplo, los clasificadores basados en árboles de decisión podrán ser encontrados dentro de la carpeta *trees*. La tabla 1.2 muestra una relación de los nombres con los que aparecen, en el explorador de Weka, los algoritmos de clasificación analizados en el presente capítulo, así como la carpeta en la que se encuentra.

Clasificador	Nombre en Weka	Carpeta
k-vecinos más cercanos (kNN)	IBk	lazy
Árboles de Decisión	J48	trees
Redes Bayesianas	BayesNet	bayes
Máquinas de Soporte Vectorial (SVM)	SMO	functions
Multilayer Perceptron (MLP)	MultilayerPerceptron	functions

Tabla 1.2 Nombres con los que aparece cada clasificador en Weka.

Consideraciones finales del capítulo

La alta capacidad de mutación del VIH hace que aparezca resistencia a los fármacos antirretrovirales utilizados. Las pruebas existentes para determinar la resistencia de las mutaciones son costosas y complejas, de ahí la importancia de que en los últimos años, se hayan utilizado diferentes técnicas de inteligencia artificial para predecir la resistencia al VIH.

Los datos obtenidos en las pruebas realizadas se encuentran en bases de datos internacionales como "*Stanford HIV Resistance Database*", a partir de ella se han realizado estudios para predecir la resistencia de la proteasa, por lo que se pudiera utilizar para predecir la resistencia a los inhibidores de la transcriptasa inversa.

En trabajos consultados en los que se aplican diferentes clasificadores a la predicción de la resistencia a fármacos de las enzimas del VIH, los resultados no han sido satisfactorios. En el caso de la proteasa el método que mejores resultados arrojó es un modelo que combina clasificadores, por lo que esta investigación se centra en la utilización de multiclasificadores para la predicción de la resistencia a inhibidores de la enzima transcriptasa inversa.

La plataforma de aprendizaje automatizado Weka, utilizada en diferentes estudios, implementa diferentes modelos de clasificación que pueden ser utilizados en la predicción de resistencia a fármacos.

CAPÍTULO 2. SISTEMAS DE COMBINACIÓN DE CLASIFICADORES

En la última década se ha desarrollado la tendencia de combinar varios clasificadores con el propósito de mejorar la precisión de los resultados. Ante la interrogante de si se justifica la combinación de clasificadores, algunos autores han respondido desde diferentes puntos de vista. Por ejemplo, Witten y Frank plantean que cuando las personas sabias toman decisiones críticas, tienen en cuenta la opinión de varios expertos en lugar de confiar en su propio juicio o la de un solo consejero [27]. Polikar explica que una idea semejante a ésta constituye la base de estos algoritmos que también son conocidos como multiclasificadores: utilizar varios expertos (clasificadores) y combinar sus diferentes salidas en aras de lograr un mejor rendimiento [43]. Ho[44] plantea que en lugar de buscar el mejor conjunto de rasgos y el mejor clasificador, ahora pudiera buscarse el mejor conjunto de clasificadores y la mejor manera de combinarlos. Por su parte, Dietterich [45] sugiere tres tipos de razones por las cuales un sistema multiclasificador puede ser mejor que un clasificador simple.

En [2] se plantea que existen varios algoritmos desarrollados para construir multiclasificadores, pero que en esencia todos estos métodos tienen dos partes importantes: selección de los clasificadores de base y elección de la forma de combinar las salidas. Precisamente, en el presente capítulo se expone en qué consiste la selección de clasificadores de base y la forma de combinar las salidas, así como las características principales de algunos modelos de multiclasificadores existentes. Además, se explica el diseño e implementación de dos herramientas: una que permite la transformación de las bases de casos a utilizar en la validación de los multiclasificadores y otra que agiliza el cálculo de algunas medidas de diversidad, lo que facilita a selección de clasificadores de base.

2.1. Selección de Clasificadores de Base y Diversidad

Como se había explicado, el funcionamiento de un multiclasificador se basa en dos etapas fundamentales: el uso de varios clasificadores, que se conocen como clasificadores de base, y la combinación de sus diferentes salidas. Si para un problema en particular, se encontrara un clasificador perfecto, que no cometiera errores, entonces no sería necesario utilizar un multiclasificador. Si por el contrario, el clasificador comete errores, entonces lo que se busca es

complementarlo con otros clasificadores que cometan errores en casos diferentes [42]. Teniendo en cuenta esto, puede inferirse que para seleccionar los clasificadores de base, la combinación más efectiva no tiene por qué ser aquella conformada por los clasificadores con los que mejores resultados se obtiene por separado, sino que debe buscarse la manera en que los mismos se complementen, o sea, que no se equivoquen al clasificar los mismos casos. En este sentido resulta de vital importancia la diversidad de los clasificadores de base.

2.1.1. Clasificadores de Base y diversidad

La selección de los clasificadores de base es fundamental en la construcción de un multiclasificador. Al realizar la selección, es imprescindible que se logre diversidad en los mismos si se desea mejorar los resultados individuales de los clasificadores. La diversidad da una medida de la correlación que existe entre los resultados de los diferentes clasificadores. Existen diferentes maneras de lograr la diversidad, lo cual está asociado, en muchos casos, a la topología del multiclasificador construido.

Una manera de lograr diversidad consiste en variar algún parámetro del modelo de clasificación. Por ejemplo, *Bagging* y *Boosting*, modelos de combinación de clasificadores que serán caracterizados en este capítulo, lo hacen variando el conjunto de entrenamiento, lo que les permite utilizar un único modelo de clasificación, el cual es entrenado con diferentes subconjuntos de casos; el primero selecciona los subconjuntos aleatorios y el segundo los va seleccionando iterativamente en dependencia del resultado de la iteración anterior.

Algunos paradigmas de combinar clasificadores usan el mismo modelo de clasificación, pero no existe evidencia de que esa estrategia sea mejor que el uso de diferentes modelos [42]. Teniendo en cuenta esto, se han desarrollado otros modelos de combinación de clasificadores que logran la diversidad utilizando diferentes modelos de clasificación, los clasificadores de base, que se entrenan utilizando la misma base de casos, entre los que se encuentran *Stacking* y *Multi Expert by Hard Instances*, que también serán analizados en el presente capítulo.

En el caso en que se utilizan distintos modelos de clasificación, una selección al azar de diferentes clasificadores de base, no garantiza la diversidad, pues se hace necesario tener una idea de la correlación que existe entre los resultados de cada uno. Este análisis evitaría, por ejemplo, seleccionar un conjunto de clasificadores de base que se equivoquen al clasificar los

mismos casos, pues de darse esta situación, el multiclasificador nunca lograría clasificar esos casos de manera correcta.

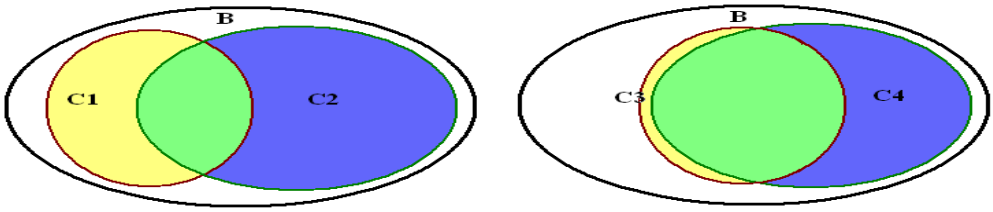

Figura 2.1 Resultado de aplicar dos pares de clasificadores a una base de casos.

Para una mejor comprensión de la diversidad, se plantea la siguiente situación hipotética: Supongamos que se desea combinar dos clasificadores en un determinado multiclasificador. Para ello se cuenta con los clasificadores Q1, Q2, Q3 y Q4; que se desean combinar dos a dos: Q1-Q2 o Q3-Q4. Una vez entrenados los clasificadores, se obtienen los resultados de la clasificación para una misma base de casos. La figura 2.1 muestra un conjunto B que representa la base de casos; los conjuntos C1, C2, C3 y C4 representan los casos que fueron clasificados de manera correcta por cada clasificador, respectivamente. Note que el espacio que queda sin sombrear, representa los casos que fueron clasificados de manera incorrecta por ambos clasificadores. En este caso podemos decir que, aunque la cantidad de casos bien clasificados por cada par de clasificadores es similar en ambos casos, para la variante Q1-Q2 se evidencia una mayor diversidad que para la variante Q3-Q4, pues la cantidad de casos que no fueron clasificados de manera correcta por ninguno de los clasificadores es menor en el primer caso.

2.1.2. Medidas de diversidad

Como ya se había expuesto, la diversidad es un factor importante a tener en cuenta para garantizar que un multiclasificador obtenga resultados satisfactorios. Para los modelos de combinación de clasificadores que utilizan más de un clasificador de base, es necesario, a la hora de seleccionar el conjunto de clasificadores de base, valorar la diversidad de los mismos. Con este propósito, algunos autores [42, 46, 47] han definido medidas que pueden utilizarse para estimar cuan diversos son los clasificadores. A continuación se relacionan algunas medidas de diversidad, descritas en [42] .

Las medidas expuestas a continuación, analizan un par de clasificadores en cada momento, por lo que se conocen como medidas en forma de pares. Las mismas se basan en los resultados

obtenidos por cada clasificador, al clasificar un caso, que pueden ser 0 (clasificó el caso de manera incorrecta) o 1 (clasificó el caso de manera correcta). Por ejemplo, para dos clasificadores (*Ci, Cj*), podemos recoger los resultados obtenidos para un caso, de la manera siguiente:

	Cj correcto (1)	Cj incorrecto (0)
Ci correcto (1)	*a*	*b*
Ci incorrecto (0)	*c*	*d*
a + b + c + d = 1		

Tabla 2.1 Relación de resultados de dos clasificadores con un caso

Note que para un caso, sólo puede ocurrir que: ambos clasificadores lo clasificaron de manera correcta, ambos clasificadores lo hicieron de manera incorrecta, o uno lo hizo de manera correcta y el otro de manera incorrecta. Luego, sólo uno de los valores *a*, *b*, *c* o *d* valdrá 1, el resto tendrá valor 0. Pero un caso no es suficiente para analizar la diversidad. Si sumamos los resultados obtenidos al clasificar cada caso de una base de tamaño *N*, con el par de clasificadores (*Ci, Cj*), haciendo que $A= a_1 + a_2 + \ldots + a_N$ y de manera análoga obtenemos los valores de *B*, *C* y *D*; podemos consolidar los resultados de ambos clasificadores al clasificar los *N* casos de la base, de la manera siguiente:

	C_j correcto (1)	C_j incorrecto (0)
C_i correcto (1)	*A*	*B*
C_i incorrecto (0)	*C*	*D*
A + B + C+ D = N		

Tabla 2.2 Relación de resultados de dos clasificadores con toda la base

De esta manera, El valor de *A* representa todos los casos que fueron clasificados de manera correcta por ambos clasificadores, el valor de *B*, todos los casos que fueron clasificados de manera correcta por C_i y de manera incorrecta por C_j; y así sucesivamente. Estos valores son utilizados en el cálculo de las medidas de diversidad.

Medida de desacuerdo

La medida de desacuerdo (*Disagreement Measure*) es la más intuitiva de las medidas entre un par de clasificadores, muestra la cantidad de casos en los que los dos clasificadores no están de acuerdo; por lo que si se divide el valor de *D* entre la cantidad de casos de la base, se obtiene la

probabilidad de que ambos clasificadores discrepen en sus predicciones.

$$D(C_i, C_j) = B + C \qquad \text{Ecuación 2.1}$$

Mientras mayor sea el valor de *D*, mayor será la diversidad del par de clasificadores analizado.

Medida de doble fallo

Otra medida también muy intuitiva, se conoce como medida de doble fallo (*Double-Fault Measure*) y considera la cantidad de casos en que se equivocan los dos clasificadores al mismo tiempo. Esta medida se basa en el hecho de que es más importante conocer cuándo los dos clasificadores se equivocan que cuándo los dos clasificadores clasifican de manera correcta.

$$DF(C_i, C_j) = D \qquad \text{Ecuación 2.2}$$

Mientras menor sea el valor de *DF*, mayor será la diversidad del par de clasificadores analizado.

Coeficiente de correlación ρ

Otra de las medidas de diversidad es el coeficiente de correlación (*Correlation*), el cual, como indica su nombre, da una muestra de la correlación que existe entre las salidas de ambos clasificadores y se calcula como sigue

$$\rho(C_i, C_j) = \frac{A \cdot D - B \cdot C}{\sqrt{(A + B) \cdot (C + D) \cdot (A + C) \cdot (B + D)}} \ , -1 \leq \rho \leq 1 \qquad \text{Ecuación 2.3}$$

Mientras menor sea el valor de ρ, mayor será la diversidad del par de clasificadores analizado.

El estadístico Q

El estadístico Q (Q *Statistics*) es otra de las medidas para pares de clasificadores. Aquellos clasificadores que tienden a clasificar de manera correcta el mismo conjunto de casos tienen un valor positivo de Q. ρ y Q tienen el mismo signo y se puede probar que $|\rho| \leq |Q|$.

$$Q(C_i, C_j) = \frac{A \cdot D - B \cdot C}{A \cdot D + B \cdot C} \ , -1 \leq Q \leq 1 \qquad \text{Ecuación 2.4}$$

Mientras menor sea el valor de Q, mayor será la diversidad del par de clasificadores analizado.

Coeficiente de acuerdo interclasificador

El coeficiente de acuerdo interclasificador (*Interrater Agreement*) es una medida que muestra el nivel de acuerdo entre los clasificadores

$$k(C_i, C_j) = \frac{2 \cdot (A \cdot C - B \cdot D)}{(A + B) \cdot (C + D) + (A + C) \cdot (B + D)}$$ Ecuación 2.5

Mientras menor sea el valor de k, mayor será el desacuerdo entre el par de clasificadores analizado y por tanto, mayor será la diversidad.

En todos los casos se muestra una fórmula relativamente sencilla para calcular la medida de diversidad entre dos clasificadores. Para un conjunto de clasificadores de tamaño L>2, deben analizarse los clasificadores de dos en dos y promediarse los resultados obtenidos al procesar los L(L – 1)/2 pares de clasificadores que se pueden formar, para obtener un único resultado de la medida.

Del mismo modo que no podemos decir que un clasificador sea mejor que otro de manera general, tampoco podemos decir que lo sea una medida de diversidad u otra. El diseño de nuevas medidas de diversidad es un área activa de investigación. La misma autora define diferentes medidas de diversidad en otros trabajos [42, 46, 48]. El presente trabajo se ha limitado al uso de las cinco medidas de diversidad en forma de pares, relacionadas en este epígrafe, porque las mismas resultan bastante intuitivas y fáciles de aplicar.

2.2. Combinación de salidas de los clasificadores de base

Otro elemento importante en el funcionamiento de un multiclasificador, además de los clasificadores de base, es la forma en que se deben combinar las salidas de éstos para obtener la salida del modelo de combinación de clasificadores. La figura 2.2 muestra un esquema de combinación abstracto en el que la salida del multiclasificador corresponde a una función de decisión D: $\Re^n \rightarrow \Omega$, que asigna al vector de entrada $x \in \Re^n$, una de las k clases pertenecientes al conjunto $\Omega = \{c_1, c_2, \ldots, c_k\}$, para lo cual usa una regla de combinación que procesa las salidas $S^{(i)} \in \Omega$, correspondientes a los N clasificadores que utiliza el modelo y decide la clase $c_j \in \Omega$ que asignará a la entrada x [49].

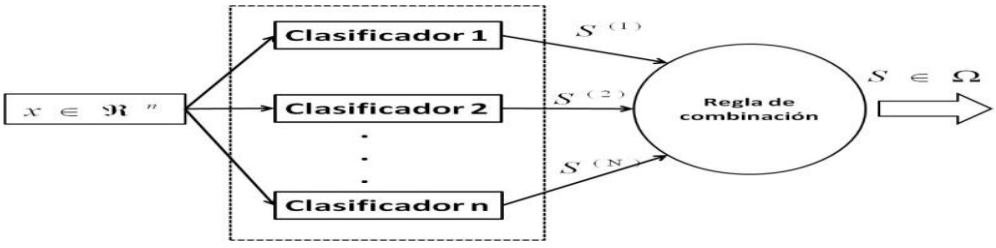

Figura 2.2 Esquema de un método de combinación de nivel abstracto. Tomado de [49]

Existen diferentes maneras de combinar la salida de los clasificadores pero en su mayoría pueden agruparse en dos estrategias fundamentales de combinación: selección y fusión [42]. Aunque esta misma autora no descarta la existencia de esquemas que las mezclan, en este trabajo se caracterizan únicamente estas estrategias "puras".

2.2.1. Combinación por selección

La selección es la simple elección del "mejor" clasificador para una instancia determinada [2]. La combinación por selección presupone que cada miembro del multiclasificador conoce bien una parte del conjunto de casos y que existe la forma de determinar el mejor clasificador para una entrada *x*. Así, la decisión de ese clasificador es aceptada como resultado para la entrada *x*. La combinación por selección no atrae tanto la atención como la fusión; pero esto pudiera cambiar en el futuro pues probablemente la primera estrategia, si se realiza un buen entrenamiento, sea mejor que la segunda [42].

2.2.2. Combinación por fusión

La combinación por fusión presupone que cada miembro del multiclasificador tiene conocimiento de todo el conjunto de entrenamiento. En este enfoque se combina de alguna manera la salida de todos los clasificadores, como por ejemplo el promedio o el voto mayoritario [42]. La fusión se basa en combinar, mediante alguna función, las salidas de los diferentes clasificadores [2].

Una de las formas más comunes de fusión es el voto mayoritario. En este caso, cada clasificador vota por una clase, de acuerdo con su resultado y finalmente se selecciona la clase con mayor cantidad de votos. En caso de más de dos clases, el resultado se daría por mayoría y cuando sólo existen dos clases, coincide efectivamente con la mayoría absoluta (50 % de votos más 1). Tiene el inconveniente de que los empates se resuelven de manera arbitraria [42].

Teniendo en cuenta que no todos los clasificadores en el multiclasificador tienen la misma precisión, es razonable que se intente dar al clasificador con mejores resultados, mayor poder en la toma de decisión. De este modo surge la variante del voto mayoritario ponderado, donde a cada clasificador se asocia un peso, que puede estar dado, por ejemplo, por el error de clasificación obtenido por él sobre el conjunto de entrenamiento. La clase de mayor peso obtenida, es la que se da como resultado final del clasificador. Las figuras 2.3 y 2.4 muestran el

funcionamiento de ambas formas de combinar las salidas, respectivamente.

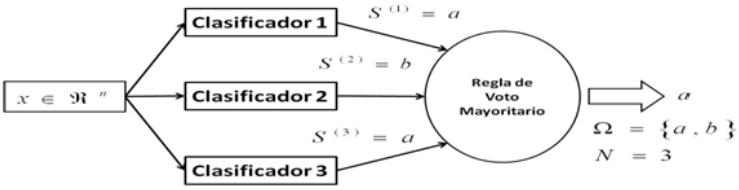

Figura 2.3 Ejemplo de combinación por voto mayoritario. Tomado de [49]

En el caso de la figura 2.3, dos clasificadores dieron como salida la clase "a" y un clasificador dio como salida la clase "b", por lo que el resultado de combinar estas salidas es la clase "a", que fue la que más votos recibió. Por el contrario, en el caso de la figura 2.4, aunque sólo un clasificador dio como salida la clase "b", el peso asociado a dicho clasificador es mayor que la suma de los pesos de los dos clasificadores que dieron como salida la clase "a", por lo que el resultado de combinar las salidas, en este caso, es la clase "b".

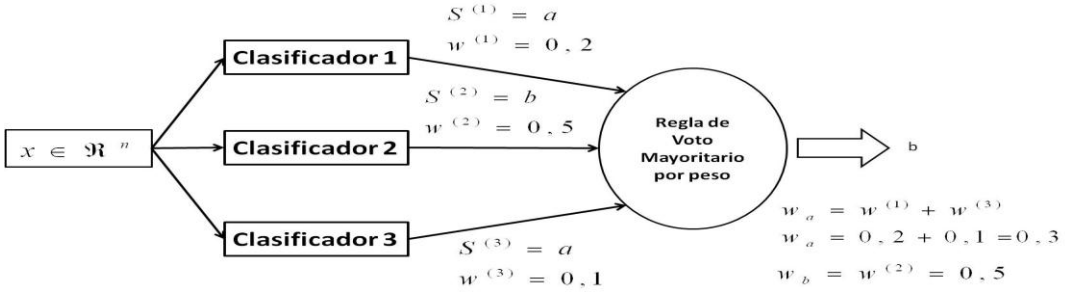

Figura 2.4 Ejemplo de combinación por voto mayoritario ponderado. Tomado de [49]

2.3. Modelos de combinación de clasificadores

Maudes [50] define un multiclasificador como un conjunto de clasificadores cuyas predicciones individuales se combinan de alguna forma para así obtener una predicción final conjunta. El modelo de voto de clasificadores[51] introducido por Kittler en 1998 e identificado como *Vote* en Weka es quizás uno de los más fieles a este precepto, ya que utiliza diferentes modelos de clasificación como clasificadores de base que van a actuar de manera independiente. Estos clasificadores tienen como salida vectores de distribución de probabilidad para cada una de las clases y *Vote* puede combinar estas de diferentes maneras: buscando la clase de máxima o mínima probabilidad, de probabilidad promedio o por voto mayoritario.

Para explicar la idea de este modelo, consideremos que el mismo cuenta con *L* clasificadores de base: ($C_1, C_2, C_3, \ldots, C_L$) y que permite manejar un problema de clasificación de *m* clases:

($w_1, w_2, w_3, \ldots, w_m$). Un clasificador C_j tiene para cada clase w_i un estimado de probabilidad $P(C_j, w_i)$ y da como salida un vector de distribución de probabilidades para cada clase, cumpliendo que:

$$\sum_{i=1}^{m} P(C_j, w_i) = 1 \qquad \text{Ecuación 2.5}$$

Luego, para clasificar un caso x, los vectores de probabilidad obtenidos por cada clasificador pueden representarse en la matriz DP (Figura 2.5), a partir de la cual se pueden definir las diferentes combinaciones de salida, por ejemplo:

$$Promedio_{w_i} = \frac{\sum_{j=1}^{L} DP(j,i)}{L} \qquad \text{Ecuación 2.6}$$

Luego, la función de máxima probabilidad se calcularía como muestra la Ecuación 2.7, dando como resultado la clase que mayo promedio de probabilidad posee

$$Máximo_{w_i} = \frac{V(w_i)}{\sum_{i=1}^{m} V(w_i)}, \quad donde \quad V(w_i) = max_{j=1}^{L} DP(j,i) \qquad \text{Ecuación 2.7}$$

De esta misma manera, se calculan las otras funciones a partir de estos resultados obtenidos por los clasificadores de base.

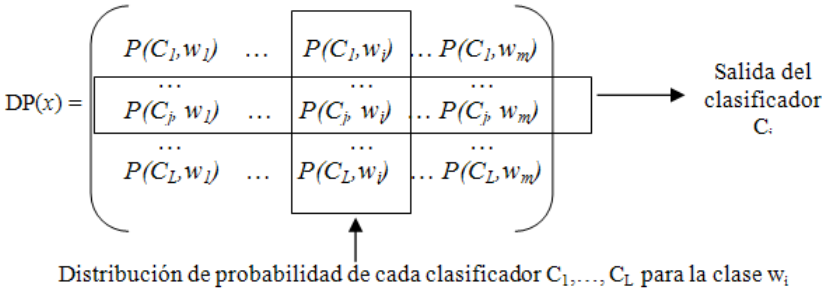

Figura 2.5 Matriz de distribución de probabilidades de *L* clasificadores para *m* clases.

No todos los modelos de multiclasificadores se asemejan a *Vote* en cuanto al tratamiento que dan a sus clasificadores de base y la forma en que combinan las salidas. A continuación se caracterizan tres de los multiclasificadores más populares: *Bagging*, *Boosting* y *Stacking*, así como el modelo *Multi-Expert by Hard Instances*, por la importancia que tiene para esta investigación, ya que fue utilizado en [2] para la predicción de resistencia a la proteasa, con buenos resultados.

2.3.1. Bagging

Bagging es uno de los primeros multiclasificadores. Fue introducido en 1996 como acrónimo de ***B**oostrap **agg**regat**ing*** [52]. El método construye *N* clasificadores base, cada uno de ellos utilizando el mismo algoritmo, pero utilizando distintos conjuntos de entrenamiento. Cada conjunto de entrenamiento se obtiene a partir de un remuestreo con reemplazamiento (*bootstrap*) de un determinado porcentaje de instancias del conjunto de entrenamiento original, que es lo que da nombre al método. El multiclasificador, una vez construido, hace sus predicciones a partir de las predicciones de sus clasificadores de base, de forma que la clase con más votos es la que finalmente es tomada como resultado [50].

El algoritmo que utiliza *Bagging* puede ser especificado como sigue [27, 42]:

Fase de entrenamiento:

- Inicializar conjunto de clasificadores D = Ø y *L* = número de veces que se entrenará el modelo de clasificación utilizado.
- Para k = 1 hasta *L* repetir:
  - Tomar un subconjunto con reemplazo S_k del conjunto de entrenamiento.
  - Construir el clasificador D_k a partir del entrenamiento, utilizando el subconjunto S_k, del modelo de clasificación utilizado.
  - Adicionar D_k al conjunto de clasificadores.

Fase de clasificación:

- Para k = 1 hasta L repetir:
  - Predecir utilizando el modelo D_k la clase a la que pertenece la entrada x.
- Retornar como clase para x aquella con más número de votos, entre todas las obtenidas por los modelos D_k, $1 \leq k \leq L$

Bagging sólo utiliza un modelo de clasificador y su funcionamiento se basa en crear diferentes conjuntos de entrenamiento, extraídos del conjunto inicial de manera aleatoria y con reemplazo, con los cuales se entrenan varios clasificadores del mismo modelo. El resultado de entrenar cada vez el modelo de clasificación utilizado, será tomado como un nuevo clasificador, asegurando así la diversidad. La combinación de las salidas de los clasificadores resultantes, se realiza con la técnica de voto mayoritario.

Como ventaja de este modelo se tiene que tanto en la fase de entrenamiento como de operación, es paralelizable, o sea los *L* clasificadores a utilizar pueden entrenarse o explotarse en procesadores diferentes si así se desea. Una desventaja del mismo es que se requiere que el clasificador de base que utilice sea inestable, en el que pequeños cambios en el conjunto de

entrenamiento se revierta en grandes cambios en la salida del clasificador; de lo contrario se contaría con un conjunto de clasificadores casi similares que es muy improbable que mejore los resultados del clasificador por sí solo [27, 42]. Este multiclasificador se encuentra incluido en Weka bajo el nombre de *Bagging*, dentro de la carpeta *meta*.

2.3.2. Boosting

Boosting fue creado en 1990 inspirado en el algoritmo *Hedge(β)* [53]. Del mismo modo que *Bagging*, utiliza un único modelo de clasificación, que es entrenado utilizando muestras con reemplazo de la base de entrenamiento y al igual que *Bagging* utiliza voto mayoritario para combinar las salidas de los diferentes modelos. La diferencia entre ambos modelos radica en que *Boosting* es iterativo. Mientras que en el primero, cada modelo se construye por separado, en *Boosting* cada modelo es influenciado por aquellos que fueron construidos previamente [27].

Boosting es similar a *Bagging* en el método de crear bases de entrenamiento aleatorias con reemplazo, a partir de la base original y un único modelo de clasificación para obtener los diferentes clasificadores de base. Sin embargo, este algoritmo se realiza de manera secuencial, ya que los clasificadores se van entrenando uno detrás del otro, utilizando información del anterior. En 1997 se crea una variante de *Boosting* denominada *AdaBoost,* cuyo nombre proviene de ***Ada****ptative* ***Boost****ing* [54]. Varios autores [2, 27, 42, 50] consideran ésta como la versión más utilizada y la misma se ha dividido en *AdaBoost.M1* y *AdaBoost.R*, para clasificación utilizando múltiples clases y problemas de regresión, respectivamente. *AdaBoost* se basa en dos ideas fundamentales [50]:

1. En entrenamiento, construir iterativamente los clasificadores de base, de manera que el clasificador de base actual dé más importancia a las instancias del conjunto de entrenamiento mal clasificadas por el clasificador de base de la iteración anterior.
2. En clasificación, hacer la predicción en base a un esquema de votación ponderado, de forma que aquellos clasificadores de base con un mayor acierto sobre el conjunto de entrenamiento, tengan un mayor peso en la votación.

Este multiclasificador se encuentra incluido en Weka bajo el nombre de *AdaBoostM1*, dentro de la carpeta *meta*.

El algoritmo que utiliza *AdaBoost.M1* puede ser especificado como sigue [27, 42]:

Fase de entrenamiento:

- Inicializar conjunto de clasificadores D = Ø y L = número de veces que se entrenará el modelo de clasificación utilizado. Inicializar cada instancia con el mismo peso.
- Para k = 1 hasta L repetir:
    - Tomar una muestra con reemplazo S_k del conjunto de entrenamiento.
    - Construir el clasificador D_k a partir del entrenamiento, utilizando la muestra S_k, del modelo de clasificación utilizado.
    - Calcular el error e_k del modelo sobre la muestra ponderada S_k
    - Si $e_k = 0$ o $e_k \geq 0{,}5$ Desechar D_k y continuar con el resto de las muestras
    - Para cada instancia en S_k
        - Si la instancia fue clasificada correctamente, multiplicar su peso por $e_k/(1-e_k)$
    - Normalizar los pesos para todas las instancias.

Fase de clasificación:

- Asignar peso 0 a todas las clases
- Para k = 1 hasta L' (L' ≤ L) repetir:
    - Predecir utilizando el modelo D_k la clase a la que pertenece la entrada x.
    - Adicionar log $((1-e_k) / e_k)$ al peso de la clase que predijo D_k
- Retornar como clase para x aquella que tiene mayor peso

2.3.3. Stacking

Este modelo fue introducido en 1992 por Wolpert [55] bajo el nombre de *Stacked Generalization (Stacking,* para abreviar). Del mismo modo que *Vote* y a diferencia de *Bagging* y *Boosting*, *Stacking* utiliza diferentes modelos de clasificación como clasificadores de base. Una característica que lo diferencia de estos tres modelos es la manera en que combina las salidas de los clasificadores, pues para ello introduce el concepto de metaclasificador, el cual suplirá el uso de, por ejemplo, voto mayoritario. Este metaclasificador tiene el objetivo de aprender qué salidas de los clasificadores de base son las más confiables y así descubrir la mejor manera de combinar las mismas [27].

Stacking impone cierto nivel de jerarquización a sus clasificadores miembros. El nivel inferior, conformado por los clasificadores de base y que toma las entradas directamente del conjunto de datos original, se denomina nivel 0 y el nivel superior, que contiene el metaclasificador y toma las entradas del nivel inferior, se denomina nivel 1 [55]. Es importante destacar que los diferentes clasificadores del nivel 0 tendrán la misma entrada, vectores que representan el conjunto de rasgos de la base de casos y el metaclasificador del nivel 1 recibirá como entrada los vectores de probabilidades, que dieron como salida los clasificadores del nivel anterior (Ver figura 2.6).

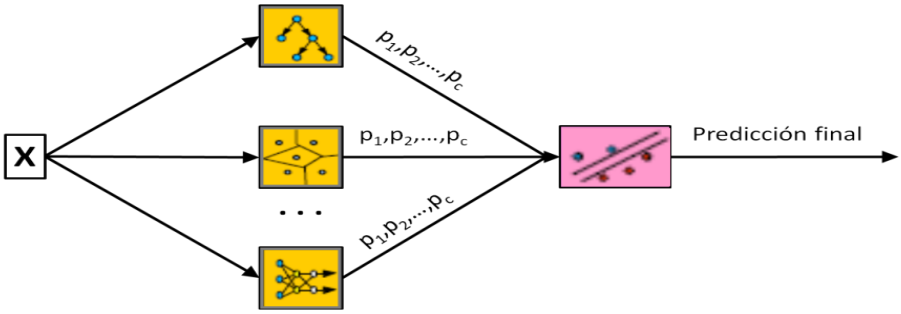

Figura 2.6. Predicción de la clase de una entrada x, utilizando *Stacking*.

El algoritmo de *Stacking* puede ser especificado como sigue [50]:

Fase de entrenamiento:

- Agrupar los casos en N_f particiones disjuntas aleatorias (Como para una validación cruzada con N_f subconjuntos).
- Para cada clasificador de nivel 0 (B clasificadores de base)
  - Para cada una de las N_f particiones de los casos
  - Entrenar una nueva versión del clasificador utilizando las Nf – 1 particiones restantes
  - Obtener para cada caso en la partición que no se usó en el entrenamiento, un vector de estimaciones de probabilidad para cada una de las C clases: $(p_1, p_2, \ldots, p_C)$
- Almacenar los vectores de estimación de probabilidad obtenidos para cada caso, con el clasificador actual:

	Clasificador i
Caso	Probabilidades
X_1	$(p_1, p_2, \ldots, p_C)$
…	…
X_N	$(p_1, p_2, \ldots, p_C)$

- Almacenar los vectores de probabilidades de cada clasificador de base, para cada caso y asociar a cada caso su clase original, para formar una nueva base de casos, según muestra la tabla siguiente:

	Clasificador i	…	Clasificador b	
Caso	Probabilidades	…	Probabilidades	Clase original
X_1	$(p_1, p_2, \ldots, p_C)$	…	$(p_1, p_2, \ldots, p_C)$	a_1
…	…	…	…	…
X_N	$(p_1, p_2, \ldots, p_C)$	…	$(p_1, p_2, \ldots, p_C)$	a_N

- Entrenar el metaclasificador de nivel 1 utilizando los datos almacenados en la tabla anterior
- Desechar todas las versiones de los clasificadores de base entrenadas anteriormente
- Entrenar cada uno de los B clasificadores de base utilizando todos los casos

Fase de clasificación:

- Para cada uno de los B clasificadores de base
  - Obtener el vector de estimación de probabilidades: $P_i(p_1, p_2, \ldots, p_C)$ para la entrada X
- Pasar como entrada al metaclasificador todos los vectores P_i $(1 \leq i \leq B)$ y obtener la predicción final de la clase

Este multiclasificador se encuentra incluido en Weka bajo el nombre de *Stacking*, dentro de la carpeta meta.

2.3.4. Muti-Expert by Hard Instances

Multi-Expert by Hard Instances, MEHI para abreviar, fue introducido en 2008 por Bonet [2] y además de sus facilidades de configuración para transformar el modelo, que serán explicadas en este epígrafe, despierta más su interés el hecho de que fue usado por la misma autora para predecir resistencia a fármacos de la enzima proteasa, del VIH. MEHI es similar a *Stacking* por el hecho de que utiliza diferentes modelos de clasificación como clasificadores de base y además utiliza un metaclasificador que aprende los resultados de los clasificadores de base. En su funcionamiento se apoya en conjuntos de casos bien clasificados y para combinar las salidas de los distintos clasificadores puede usar selección o fusión[56].

MEHI basa su funcionamiento en la identificación de casos duros, que son aquellos que fueron bien clasificados por algún clasificador de base. Para *C* clasificadores de base, separará los casos que fueron bien clasificados por cada uno de los clasificadores de base, conformando *C*+1 grupos de casos, uno por cada clasificador de base y un grupo adicional que contendrá aquellos casos que fueron bien clasificados por todos los clasificadores. La decisión de si un caso fue bien clasificado o no, depende de que la probabilidad obtenida como salida del clasificador para ese caso, supere determinado umbral, que para problemas de dos clases, pudiera tomarse 0,5 como punto de corte.

La figura 2.7 muestra un ejemplo de este modelo para 3 clasificadores de base, en el que se han representado los conjuntos de casos bien clasificados por cada clasificador de manera concentrada, lo que permite apreciar la existencia de casos que fueron bien clasificados por los tres clasificadores de base, algunos por dos clasificadores de base e incluso que algunos que no fueron bien clasificados por ningún clasificador de base.

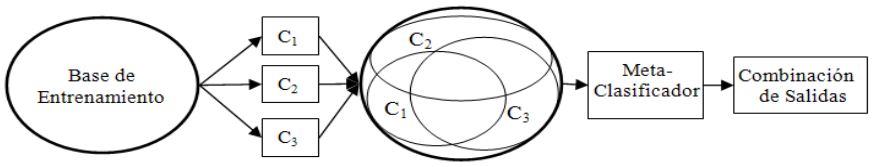

Figura 2.7 Modelo de MEHI, basado en conjuntos de instancias bien clasificadas. Tomado de [2]

En el modelo hay dos grupos de casos interesantes a tener en cuenta. Aquellos que no fueron

bien clasificados por ningún clasificador, serán excluidos del modelo y los que fueron bien clasificados por todos los clasificadores serán incluidos en un grupo aparte, como se había dicho. Si este grupo se hace muy grande, esto pudiera provocar que los grupos asociados a algunos clasificadores contengan un conjunto de casos muy pequeño, dificultando el aprendizaje. Para resolver este problema, se ha definido un umbral basado en la probabilidad de salida de los clasificadores, que permitirá decidir si un caso será incluido en el grupo que representa los bien clasificados por ese clasificador o los bien clasificados por todos los clasificadores.

Después de entrenados los clasificadores de base, se procede a entrenar el metaclasificador, para lo que se conformará una nueva base de casos. A esta nueva base se adicionarán los casos que conforman cada uno de los grupos de casos duros asociados a los clasificadores de base. Cada caso será descrito por los mismos rasgos descriptivos que tenía en la base original, sólo el rasgo objetivo cambiará, ahora se establecerá como clase para cada caso, el grupo de casos duros al cual pertenece. Note que pudieran existir casos repetidos que solo difieran en la clase asociada, teniendo en cuenta que algunos casos pueden pertenecer a más de un grupo.

El metaclasificador tendrá como salida un vector de probabilidades en correspondencia con los grupos que se crearon. Para el módulo de combinación de salidas se proponen dos formas de combinarlas, a seleccionar alternativamente cuando se cree el multiclasificador: una basada en selección y otra basada en fusión. La combinación por fusión se basa en ponderar las salidas de los clasificadores con la salida del metaclasificador. La combinación por selección es más simple, consiste en seleccionar el clasificador de mayor probabilidad asignada atendiendo al vector de salida del metaclasificador. El algoritmo de MEHI puede ser especificado como sigue [2]:

Fase de entrenamiento:

- Entrenar los clasificadores de base
- Formación de los conjuntos de casos bien clasificados
- Detección del conjunto intersección
- Formación de la nueva base de entrenamiento para el metaclasificador
    - Para cada caso de la base original,
        - Para cada clasificador C_i,
            - Si el caso pertenece a la diferencia del conjunto de bien clasificados y el conjunto intersección, añadir a la base y asociar como clase *i*
            - Si el caso pertenece al conjunto intersección y la probabilidad está por encima del umbral establecido, añadir a la base y asociar como clase *i*
        - Si el caso pertenece al conjunto intersección y no fue añadido con ningún clasificador, añadir a la base y asociar como clase C.
- Entrenar el metaclasificador con la nueva base creada

Fase de clasificación:

- Clasificar utilizando el metaclasificador para obtener el vector de probabilidades asociadas a cada clase (clasificador).
- Según el método de combinación de salida:
    - Si selección, clasificar utilizando el clasificador asociado a la mayor probabilidad obtenida, devolver el vector obtenido como salida.
        - Si esta probabilidad está asociada a la última clase (C+1) proceder igual que en fusión.
    - Si fusión, clasificar utilizando todos los clasificadores de base y pesar los vectores de salida con el obtenido por el metaclasificador. Devolver como salida el vector obtenido de esta suma pesada, normalizada.

Este multiclasificador se encuentra incluido en Weka bajo el nombre de *MultiClassifierByMetaClassifier* dentro de la carpeta *meta*.

2.4. Herramientas para el preprocesamiento de datos en problemas de clasificación

La utilización de Weka en la solución de problemas de clasificación, requiere que en ocasiones se realice un preprocesamiento de la información con que se cuenta. Ejemplo de lo anterior es que para la selección de los clasificadores de base que conforman un multiclasificador es necesario conocer la diversidad de los mismos, para lo cual se utilizan algunas medidas que no se encuentran implementadas en Weka. Por otro lado cuando se describió dicha herramienta en el capítulo 1 se explicó la necesidad de que las bases de casos a utilizar respondieran a la estructura definida por Weka. En el presente epígrafe se describen dos herramientas orientadas a resolver ambas problemáticas. Para formalizar el análisis y diseño se utilizó la metodología de desarrollo de software SXP [57], una metodología ágil que tiene como características no generar gran cantidad de información y la integración del cliente con el equipo de desarrollo.

Los elementos fundamentales del análisis, diseño e implementación de ambas herramientas se encuentran en el portafolio digital del Departamento de Biometría del CISED.

2.4.1. Herramienta para el cálculo de las medidas de diversidad

El cálculo de las medidas de diversidad se torna complejo si tenemos en cuenta que deben analizarse los resultados de los clasificadores individuales utilizados para cada una de las bases procesadas y en cada caso calcular las medidas para todos los grupos de clasificadores posibles a formar, cuya cantidad depende de la topología seleccionada. A partir de esta situación se

decidió diseñar una aplicación que facilite el cálculo de estas medidas, la misma debe cumplir con:

- Facilidad para importar los resultados generados por Weka como salida del procesamiento de cada base.
- Calcular los valores de A, B, C y D descritos en la tabla 2.2, referidos a la coincidencia o no en el resultado de la clasificación de un par de clasificadores.
- Obtener un reporte de los grupos de clasificadores con mejor diversidad, según las medidas definidas.
- Garantizar la escalabilidad de la aplicación permitiendo la inclusión de nuevas medidas de diversidad, en el momento que sea necesario.

2.4.2. Herramienta para la transformación de bases de casos al formato Weka

Como se mencionó en el epígrafe 1.3 la estructura definida por Weka para los archivos que contienen la base de casos no constituye un estándar. En algunos de los trabajos consultados se utilizan otras bases de casos que no cumplen con los requerimientos de Weka; ejemplo de ello son los trabajos de [2, 11] donde se transforma la base de casos Stanford. La presente investigación prevé la utilización de esta base de casos, al tomar sus principales fundamentos de [2]. A partir de esta situación se decidió diseñar una herramienta que facilite la transformación de una base de casos al formato utilizado por Weka, la misma debe cumplir con:

- Permitir importar una base de casos con un formato específico.
- Realizar la transformación de la base de casos importada brindando la posibilidad de modificar sus casos.
- Exportar la información existente de acuerdo al formato definido por Weka.
- Garantizar la escalabilidad de la aplicación permitiendo la inclusión de nuevas transformaciones en el momento que sea necesario.

Conclusiones del capítulo

El desempeño de los modelos de combinación de clasificadores supera al de los clasificadores individuales en la mayoría de los casos estudiados.

Para construir un multiclasificador deben ser tenidos en cuenta la selección de los clasificadores de base y la combinación de las salidas de los mismos. La selección de los clasificadores de

base se debe realizar sobre la base de garantizar la diversidad, esto garantiza que sus resultados mejoren los de los clasificadores individuales. La combinación de las salidas de los clasificadores de base puede realizarse utilizando una de las dos estrategias principales: selección o fusión.

Bagging, Boosting, Stacking y MEHl son algunos de los modelos de combinación de clasificadores, se diferencian por la manera de garantizar la diversidad y de combinar las salidas de los clasificadores de base. Los cuatro se encuentran implementados en la herramienta Weka, por lo que pueden se pueden utilizar en la solución de problemas de clasificación.

Se diseñó e implementó una herramienta que permite el cálculo de las medidas de diversidad, permitiendo además obtener reportes de comparación entre ellas, lo cual facilita la selección de los clasificadores de base.

Se diseñó e implementó una herramienta que permite transformar una base de casos *Stanford* en el formato exigido por Weka, la transformación incluye la sustitución de los aminoácidos presentes en la secuencia, por sus energías de activación.

CAPÍTULO 3. PREDICCIÓN DE RESISTENCIA DEL VIH UTILIZANDO MODELOS DE COMBINACIÓN DE CLASIFICADORES

Como se explica en el capítulo 1, cuando se enfrenta un problema de clasificación se debe primeramente representar el problema, luego seleccionar el modelo de clasificador, entrenarlo y por último validar o evaluar el mismo.

En el presente capítulo se modela el problema de la predicción de resistencia a fármacos, de la enzima transcriptasa inversa, proceso en el que resulta fundamental la construcción de la base de casos a partir de la información obtenida de la base de datos de *Stanford* para nueve fármacos que inhiben a dicha proteína. Como una primera parte de la investigación, se muestra el uso de modelos de clasificación convencionales y los resultados obtenidos al evaluar el funcionamiento de los mismos para predecir la resistencia. A continuación se describen los modelos de combinación de clasificadores utilizados, así como el uso de algunas medidas de diversidad para seleccionar los clasificadores de base a combinar en cada caso. Posteriormente se muestran los resultados obtenidos al evaluar el funcionamiento de los mismos en la predicción de resistencia y se propone la mejor variante a utilizar en cada caso. Además, se validan las herramientas para el cálculo de medidas de diversidad y la transformación de bases.

3.1. Modelación del problema de predicción de resistencia del VIH y construcción de la base de casos

El problema a resolver se resume en: predecir la resistencia de mutaciones de la proteína transcriptasa reversa del virus del VIH ante nueve fármacos inhibidores de su función. Para esta tarea se cuenta con igual número de bases de antirretrovirales extraídas de la base de datos de *Stanford* [4]. Estaríamos en presencia entonces de nueve problemas de clasificación, uno por cada fármaco que se analiza.

La base de datos de *Stanford* muestra una gran cantidad de casos, teniendo en cuenta el poder de mutación de la enzima. Cada caso en la misma está descrito por el genotipo y el fenotipo. El genotipo está representado por el listado de las 240 posiciones mutadas con los aminoácidos que fueron cambiados respecto a una secuencia de referencia: HXB2, también conocida como secuencia salvaje (*wild type*). El fenotipo está representado por un factor de resistencia basado

en la concentración que se necesita del fármaco para inhibir el virus a un 50%, conocido como IC50. Se tiene el fenotipo, de diferentes mutaciones virales, asociado a nueve inhibidores de la transcriptasa inversa: AZT, DDI, D4T, 3TC, ABC, TDF, NVP, DLV y EFV.

Para construir la base de casos, a partir de la base de datos citada, primeramente se debe reconstruir cada secuencia reportada en la base. En la secuencia se señala, en cada posición, si hubo mutación, por el aminoácido que mutó, y si no hubo mutación se simboliza con (-), significando que se mantiene el aminoácido que tiene la secuencia salvaje en esa posición. Por lo que, a la hora de construir la secuencia, las posiciones que contienen un determinado aminoácido se mantendrán y aquellas posiciones marcadas como consenso (-) se van a sustituir por el aminoácido que se encuentra en esa misma posición en la secuencia de referencia HXB2. La parte superior de la figura 3.1 muestra este proceso. Los rasgos P5 y P11 se corresponden con posiciones que se mantienen tal como aparecían en la base de datos y el caso P9, con una posición de consenso, en la que se toma el aminoácido que aparece en la secuencia de referencia. Aquellas secuencias en las que aparecen posiciones no secuenciadas (.) u otro tipo de alteraciones, no serán incluidos en la base de casos.

La clase correspondiente a cada caso está basada en la razón de resistencia con respecto a la secuencia de referencia. En [4] se define un corte (*cut-off*) que casualmente es el mismo para todos los fármacos utilizados: 3.5. La misma se codificará con valor 0 (susceptible) si su razón de resistencia es menor que 3.5 y como 1 (resistente) si es mayor o igual que el corte. Este valor se adicionará como un nuevo atributo al caso. (Ver figura 3.1)

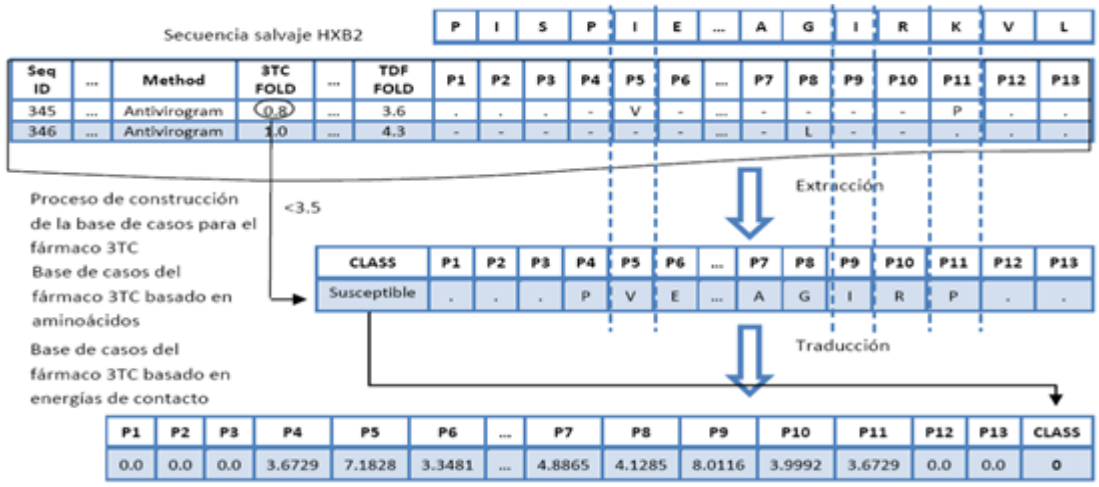

Figura 3.1 Transformación de la base de datos de Stanford a la base de casos del problema [11]

Hasta este momento se cuenta con la información de experiencia que se necesita para el problema, o sea, una base con un conjunto de secuencias mutadas de la proteasa del VIH a la que se le hace corresponder la resistencia del fármaco que se analice en cada caso. Como la clase fue codificada de manera binaria (susceptible o resistente), el problema se convierte en un problema de clasificación de dos clases.

En [2] se propone sustituir, en las secuencias, los aminoácidos por las energías de contacto, ya que esta representación expresa mejor las influencias entre las posiciones de la enzima en su estructura tridimensional. Teniendo en cuenta la validación del uso de energías de contacto, realizada en dicha investigación, se transformará la base de casos sustituyendo cada aminoácido por la energía de contacto correspondiente al mismo (Ver anexo 1), proceso que se muestra en la parte inferior de la figura 3.1.Para agilizar este proceso de transformación de las bases originales de *Stanford* a la base de casos deseada, se utilizó la aplicación descrita en el epígrafe 2.4.2.

3.2. Uso de modelos de clasificación para predecir la resistencia del VIH

En primer capítulo se caracterizaron algunos de los métodos de clasificación más usados: *k*-vecinos más cercanos, árboles de decisión, redes bayesianas, redes neuronales artificiales y máquinas de soporte vectorial. Estos han sido utilizados para predecir la resistencia a inhibidores de las funciones de proteínas del VIH, con resultados conservadores en la mayoría de los casos.

El hecho de que se haya construido la base de casos utilizando energías de activación en lugar de aminoácidos, para representar las secuencias de mutaciones de la enzima transcriptasa inversa, conduce a comprobar inicialmente cómo se comporta el problema de predicción de resistencia a fármacos en estas condiciones, utilizando los modelos de clasificación más sencillos. Es por ello que como parte de esta investigación se ha experimentado inicialmente con los modelos de clasificación que se muestran en la tabla 3.1. En la misma se muestran los diferentes valores de *k* utilizados con kNN y las diferentes cantidades de neuronas en la capa oculta (cantN) en el caso de MLP, lo cual significa que se usaron diez modelos de clasificación.

Clasificador	**Nombre en Weka**	**Variantes**
k-vecinos más cercanos (kNN)	IBk	*k*=1,2,3,8
Árboles de Decisión	J48	
Redes Bayesianas	BayesNet	
Máquinas de Soporte Vectorial (SVM)	SMO	
Multilayer Perceptron (MLP)	MultilayerPerceptron	cantN = 3,5,7

Tabla 3.1 Clasificadores utilizados para predecir resistencia a fármacos.

En la tabla 3.1 se muestran, además, los nombres por lo que se conocen estos métodos en la plataforma Weka, que será como serán nombrados a partir de ahora, la mayoría de ellos, cada vez que se muestren los resultados de los mismos.

3.2.1. Resultados de los clasificadores

Teniendo en cuenta que los clasificadores a utilizar se encuentran implementados en la herramienta Weka, la solución al problema está en cargar las bases asociadas a cada fármaco, entrenar los diferentes modelos y comparar los resultados de cada uno. A modo de prueba, para obtener el resultado de los diferentes clasificadores, se utilizó el método de validación cruzada con k subconjuntos (en este caso k=10).

En el anexo 2 se recogen los resultados de utilizar los diez modelos de clasificación relacionados en el epígrafe anterior para predecir la resistencia a los nueve fármacos analizados. Para todos los casos se muestra el porciento de clasificaciones correctas, teniendo en cuenta la definición realizada en 1.2.7 de utilizar la exactitud como medida de eficiencia.

Los resultados obtenidos para las bases D4T, DDI y TDF se muestran en la figura 3.2. Para estos fármacos se obtienen resultados similares para las tres topologías de MLP utilizadas, todos superiores a un 99% de clasificaciones correctas. Con el resto de los clasificadores se obtienen resultados inferiores en al menos 3 puntos porcentuales, comportándose de peor manera en el caso de la base correspondiente a TDF, cuyos resultados están incluso por debajo del 90%.

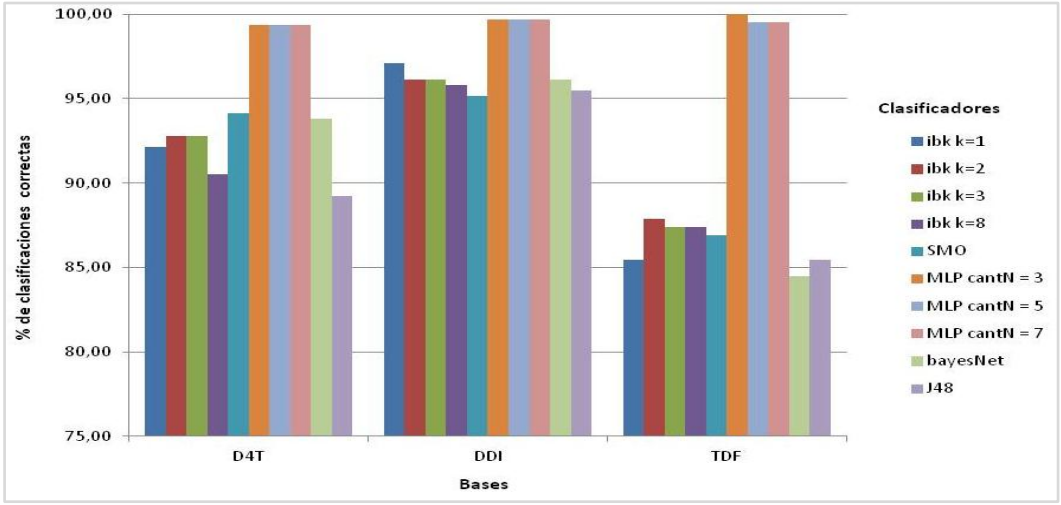

Figura 3.2 Resultados de predicción de resistencia a los fármacos D4T, DDi y TDF utilizando diferentes clasificadores

La figura 3.3 muestra los resultados obtenidos para las bases NVP, ABC y AZT. En este caso los mejores resultados se logran, en los tres casos, con un MLP de 7 neuronas en la capa oculta. Para las otras dos topologías de MLP se obtienen resultados bastante favorables también, aunque inferiores en alrededor de 2 puntos porcentuales. En el caso específico de D4T se obtienen resultados cercanos a los de MLP para la red bayesiana y el árbol de decisión.

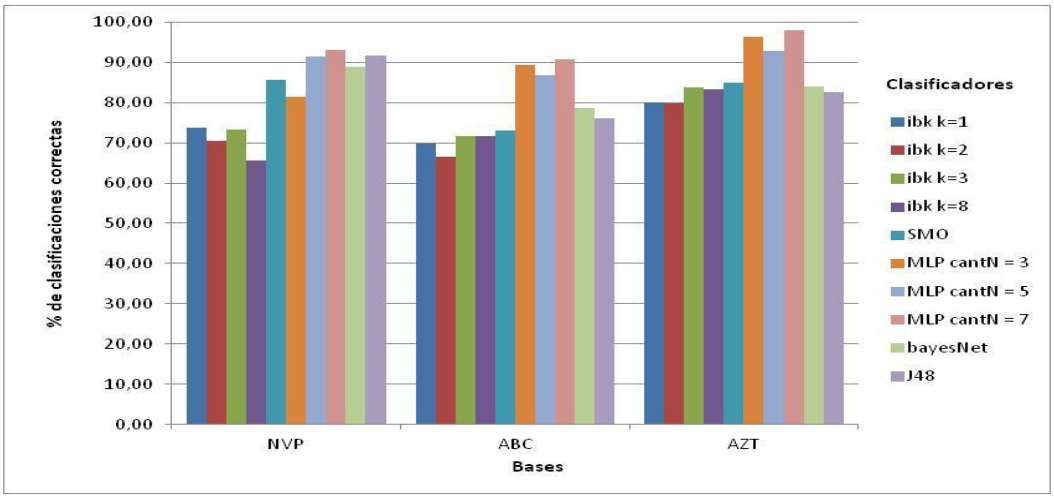

Figura 3.3 Resultados de predicción de resistencia a los fármacos NVP, ABC y AZT utilizando diferentes clasificadores

En el caso de las bases DLV, EFV y 3TC, se obtuvieron los resultados que se muestran en la figura 3.4. Para la base DLV, se obtuvieron resultados similares con las tres topologías de MLP utilizadas, cercanos al 96% de clasificaciones correctas. EFV, por su parte, fue calificada con mayor efectividad por un MLP con 3 neuronas en la capa oculta, con 94% de clasificaciones correctas, siendo las restantes inferiores al 90%. En el caso de 3TC, el árbol de decisión fue el único clasificador en superar el 90% de clasificaciones correctas, llegando a penas a un 92%.

Las pruebas realizadas utilizando modelos de clasificación simples, no arrojan resultados lo suficientemente favorables en la mayoría de los casos. Solo para tres bases se supera el 99% de clasificaciones correctas: D4T, DDI y TDF; con una base se logra alcanzar el 98%: AZT y para las cinco bases restantes, el porciento de clasificación oscila entre el 90 y 96%. Aunque no es un porciento malo de clasificación, en aras de obtener mejores predicciones, se realizaron pruebas utilizando modelos de combinación de clasificadores, las cuales se describen el próximo epígrafe.

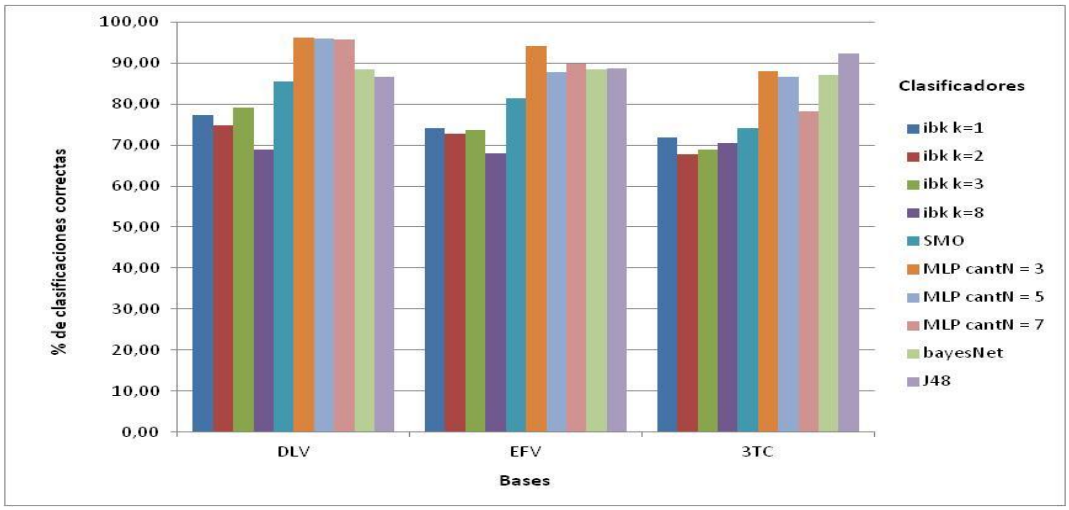

Figura 3.4 Resultados de predicción de resistencia a los fármacos DLV, EFV y 3TC utilizando diferentes clasificadores

3.3. Topologías de los modelos de combinación de clasificadores para la predicción de resistencia del VIH

Tal como se explica en 2.3, los modelos *Bagging* y *Boosting* utilizan un solo modelo de clasificación para conformar sus clasificadores de base y garantizan la diversidad entrenando varias veces el clasificador utilizado mediante la selección aleatoria de muestras con reemplazo de la base de entrenamiento. *Stacking* y MEHI, por su parte, garantizan la diversidad mediante el uso de diferentes clasificadores de base y para combinar las salidas, ambos utilizan un metaclasificador que aprende la relación entre las salidas de los clasificadores de base, con la diferencia de que en el caso de MEHI se utilizan conjuntos de casos duros (bien clasificados) en el entrenamiento del metaclasificador.

Bagging, *Boosting* y *Stacking* son reconocidos entre los multiclasificadores más populares; *MEHI,* propuesto en [2], mostró su efectividad en la predicción de resistencia a inhibidores de la enzima proteasa. Estos cuatro multiclasificadores se encuentran implementados en la herramienta Weka, lo cual facilita su uso en la presente investigación, pues del mismo modo que en el caso de los clasificadores simples, los experimentos consisten básicamente en cargar con dicha herramienta las bases asociadas a cada fármaco, entrenar los diferentes modelos y comparar los resultados de cada uno.

Los modelos de clasificación simples, cuyo comportamiento como clasificadores individuales se analizó en el epígrafe anterior, fueron tenidos en cuenta para formar parte de los clasificadores

de base para cada uno de los cuatro multiclasificadores utilizados. En el caso de *Bagging* y *Boosting* que sólo utilizan un clasificador de base, se conformaron los modelos combinados, asignando como clasificadores de base, cada uno de los diez modelos simples relacionados en el epígrafe anterior, para ambos multiclasificadores. Los diez modelos de *Bagging* y los diez de *Boosting* obtenidos fueron utilizados para predecir la resistencia a los nueve fármacos en estudio.

La topología de *Stacking* y *MEHI* permite el uso de múltiples clasificadores de base. Usar tres de ellos sería una buena opción en la gran mayoría de los casos, de donde surge la pregunta: ¿cuáles? Si se experimentara con todos los grupos de tres clasificadores de base que se pueden formar sobre el conjunto de diez clasificadores individuales con el que se cuenta, deberían comprobarse los resultados de C = 120 modelos de clasificación (ver ecuación 3.1) para cada uno de los dos multiclasificadores.

$$C = \binom{10}{3} = 120 \qquad \text{Ecuación 3.1}$$

Estas cifras complejizarían la investigación, teniendo en cuenta el tiempo requerido para entrenar cada clasificador de base y cada modelo de combinación de éstos, por lo que se impone un análisis sobre qué clasificadores de base debemos utilizar en cada caso. La selección de los clasificadores de base utilizados para estos multiclasificadores se explica en el siguiente acápite.

3.3.1. Uso de medidas de diversidad

En los experimentos realizados con los modelos simples que pudieran utilizarse como clasificadores de base, se obtuvieron los resultados de éstos para cada una de los fármacos. Una solución sencilla al problema de seleccionar los clasificadores de base adecuados, pudiera conducirnos a la opción equivocada: apostar por aquellos que tuvieron mejores resultados como clasificadores individuales.Esta variante puede ser responsable de no obtener los mejores resultados en muchos casos, pues según se explica en el capítulo 2, para lograr buenos resultados en un modelo de combinación de clasificadores es necesario garantizar la diversidad de los clasificadores de base.

El epígrafe 2.1 describe cinco medidas de diversidad cuyo cálculo permite estimar la diversidad para dos pares de clasificadores. Las cinco se corresponden con medidas en forma de pares, pues analizan un par de clasificadores en cada momento. El problema está en determinar, para cada base, cuáles son las combinaciones de tres clasificadores que dan por resultado las medidas de diversidad más convenientes, teniendo en cuenta que para la medida de desacuerdo,

los valores más grandes significan mayor diversidad, pero para el resto de las medidas, la diversidad está asociada a un menor valor de la medida.

Estas medidas se basan en un conteo de casos bien y mal clasificados, por lo que se deben procesar los resultados obtenidos previamente de los experimentos realizados para cada una de las bases con los clasificadores individuales. Este conteo implica identificar qué casos fueron clasificados de manera correcta por ambos clasificadores, por sólo uno de ellos o por ninguno de los dos. Como las medidas están definidas para pares de clasificadores, al analizar un grupo de tres de ellos, deben aplicarse las medidas dos a dos y promediar su resultado. Para decidir cuáles son los grupos de tres clasificadores con mayor diversidad, se realizó para cada base el siguiente proceso:

- Aplicar las cinco medidas de diversidad a todos los grupos de tres clasificadores que se pueden formar.
- Ordenar los grupos de clasificadores en orden decreciente de la diversidad, según el valor obtenido para la medida.
- Seleccionar aquellos grupos que obtuvieron mejores posiciones y se repitieron en un mayor número de medidas.

El cálculo de las medidas de diversidad así como los reportes de aquellas combinaciones de clasificadores que resulten más convenientes en cada caso, se torna demasiado engorroso, si tenemos en cuenta que deben procesarse los resultados de los diez clasificadores individuales para cada una de las nueve bases y en cada caso calcular las medidas para todos los grupos de tres clasificadores posibles a formar. Para agilizar este proceso, se utilizó la herramienta descrita en el epígrafe 2.4.1. En las tablas A.3.3 a la A.3.5 del anexo 3, se puede apreciar la relación de los clasificadores de base con mayor diversidad para cada base, ya que estos fueron incluidos en los modelos de combinación de clasificadores que se utilizaron para predecir la resistencia a los diferentes fármacos.

3.3.2. Resultados de los multiclasificadores

Del mismo modo que con los clasificadores simples, se utilizó la herramienta Weka para entrenar y validar los modelos de combinación de clasificadores propuestos. Los resultados obtenidos para cada uno de ellos en cada una de las bases, se encuentran en el anexo 3. De manera general, con los multiclasificadores se han mejorado notablemente los resultados obtenidos por

los clasificadores simples. A continuación se valoran algunos de estos resultados.

La figura 3.5 muestra una comparación entre los resultados de los multiclasificadores *Bagging* y *Boosting* respecto al resultado del clasificador individual que usa cada uno como clasificador de base (CB en la gráfica). En la misma se puede apreciar que los multiclasificadores tienen un comportamiento similar al clasificador individual, para todas las bases, o peor aún, para algunas bases los resultados del clasificador individual se encuentran por encima de los multiclasificadores.

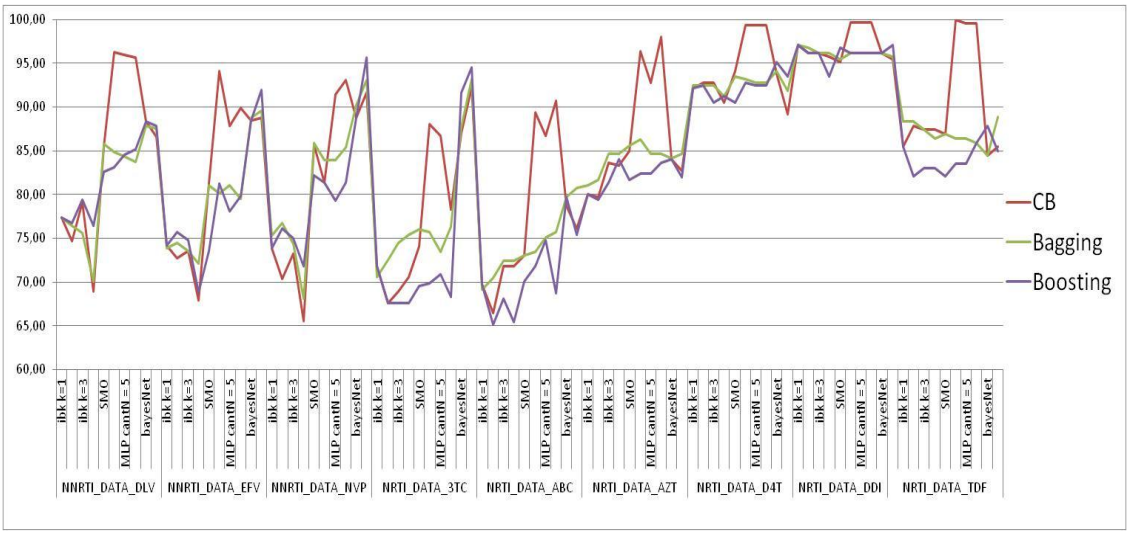

Figura 3.5 Resultados de predicción de resistencia utilizando los clasificadores individuales y los multiclasificadores *Bagging* y *Boosting*

Afortunadamente, los resultados de MEHI superan en la mayoría de los casos a los resultados individuales de los clasificadores de base que utiliza. La figura 3.6 muestra una comparación de los resultados de MEHI contra el máximo de sus clasificadores de base, o sea, el resultado del clasificador de base que mejor se comportó de manera individual. Cada punto de la gráfica tiene por abscisa, el valor del máximo resultado de los clasificadores de base y por ordenada, el resultado de MEHI. Para una mejor apreciación de los resultados se ha trazado la recta $y=x$. Luego, los puntos cercanos a la recta se corresponden con resultados similares entre MEHI y su mejor clasificador de base; los puntos por debajo de la recta corresponden a casos en los que MEHI quedó por debajo del mejor clasificador de base; los puntos sobre la recta representan aquellos casos en los que MEHI superó a sus tres clasificadores de base, lo cual ocurre en la mayoría de los casos.

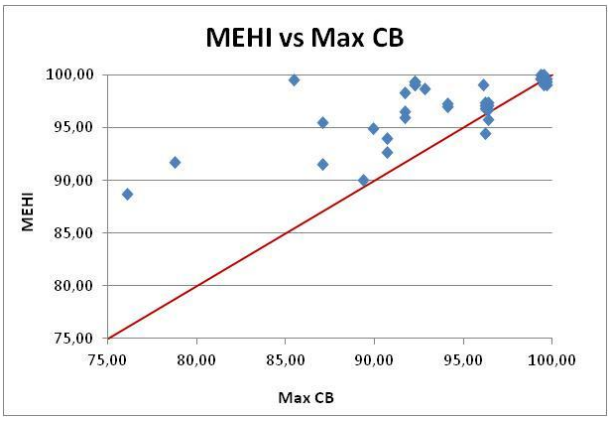

Figura 3.6 Correlación de resultados de predicción de resistencia entre el multiclasificador MEHI y el máximo de sus clasificadores de base.

Teniendo en cuenta que *Bagging* y *Boosting* muestran resultados similares y en algunos casos peores que los clasificadores de base y que MEHI supera a los clasificadores de base en la mayoría de los casos, entonces podemos concluir que MEHI supera también a estos dos multiclasificadores.

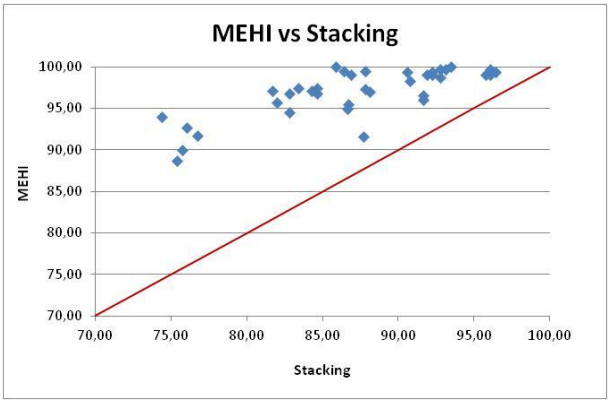

Figura 3.7 Correlación de resultados de predicción de resistencia entre los multiclasificadores MEHI y *Stacking*.

Sólo quedaría comparar los resultados de MEHI con los del multiclasificador *Stacking*. Haciendo un análisis similar al anterior, la figura 3.7 representa una comparación entre los resultados de MEHI y *Stacking*, en la que se aprecia claramente la superioridad del primero respecto al segundo en todos los casos experimentados.

Las comparaciones realizadas demuestran la superioridad de MEHI sobre el resto de los multiclasificadores; pero los excelentes resultados de este multiclasificador no se obtienen con una misma combinación de clasificadores de base para los nueve inhibidores de la transcriptasa

inversa. Este resultado era esperado, debido a que los fármacos analizados tienen forma diferente de interactuar con el centro activo de la transcriptasa, por lo que es lógico que los métodos que logren encontrar relación entre el fármaco y las secuencias de mutaciones no sean necesariamente los mismos. De esta manera, aunque se utilice el mismo tipo de multiclasificador, los clasificadores de base que utiliza este no son los mismos para cada una de las bases, que representan inhibidores diferentes. La tabla 3.2 muestra las configuraciones de MEHI para las que se logró el mejor resultado para cada base.

Base de casos	Clasificador de $Base_1$	Clasificador de $Base_2$	Clasificador de $Base_3$
TDF, 3TC	Ibk, k=1	J48	bayesNet
NVP, AZT , EFV	Ibk, k=2	J48	MLP, CantN=3
D4T	Ibk, k=8	bayesNet	MLP, CantN=5
DDI	Ibk, k=3	J48	MLP, CantN=3
ABC	MLP, CantN=3	J48	MLP, CantN=7
DLV	Ibk, k=2	bayesNet	MLP, CantN=3

Tabla 3.2 Configuraciones de MEHI con mejores resultados para cada base.

Para realizar una comparación más confiable que esta simple comparación visual, se realizaron pruebas estadísticas cuyos resultados se describen en el próximo epígrafe.

3.4. Análisis estadístico de los resultados

Las pruebas estadísticas de validación se concentran en la exactitud de los resultados de la clasificación, y como no existe una suficiente cantidad de bases a comparar, son analizadas de forma no paramétrica. Para salvar las posibles insuficiencias por el tamaño de las muestras pequeñas, y aprovechando el desempeño de las computadores disponibles, se calculan las significaciones no de manera asintótica, sino de forma exacta [58]. Para realizar las pruebas fue utilizado el paquete estadístico SPSS.

Primeramente se comparan los clasificadores individuales (CI), *Bagging* y *Boosting*. La figura 3.8 muestra la salida de SPSS para la prueba de Wilcoxon. Según los resultados hay diferencias significativas entre los resultados de *Boosting* y CI, siendo los resultados de estos últimos mejores que los de *Boosting*. Hay diferencias significativas entre *Boosting* y *Bagging*, teniendo mejores resultados *Bagging*, como lo muestran los rangos medios y la tabla de Wilcoxon de comparación de resultados; pero entre *Bagging* y CI no hay diferencias significativas, aunque los rangos medios de *Bagging* son superiores; es por ello que se comparan los resultados de

Bagging con *Stacking* y MEHI.

Ranks

		N	Mean Rank	Sum of Ranks
Bagging - CI	Negative Ranks	36[a]	53,92	1941,00
	Positive Ranks	45[b]	30,67	1380,00
	Ties	9[c]		
	Total	90		
Boosting - CI	Negative Ranks	50[d]	42,51	2125,50
	Positive Ranks	22[e]	22,84	502,50
	Ties	18[f]		
	Total	90		
Boosting - Bagging	Negative Ranks	54[g]	45,52	2458,00
	Positive Ranks	25[h]	28,08	702,00
	Ties	11[i]		
	Total	90		

a. Bagging < CI
b. Bagging > CI
c. Bagging = CI
d. Boosting < CI
e. Boosting > CI
f. Boosting = CI
g. Boosting < Bagging
h. Boosting > Bagging
i. Boosting = Bagging

Test Statistics[b]

	Bagging - CI	Boosting - CI	Boosting - Bagging
Z	-1,321[a]	-4,554[a]	-4,291[a]
Asymp. Sig. (2-tailed)	,187	,000	,000
Exact Sig. (2-tailed)	**,190**	**,000**	**,000**
Exact Sig. (1-tailed)	,095	,000	,000
Point Probability	,000	,000	,000

a. Based on positive ranks.
b. Wilcoxon Signed Ranks Test

Figura 3.8 Test de Wilcoxon aplicado a los resultados de los clasificadores individuales, *Bagging* y *Boosting*

Ranks

		N	Mean Rank	Sum of Ranks
MEHI - Stacking	Negative Ranks	0[a]	,00	,00
	Positive Ranks	37[b]	19,00	703,00
	Ties	0[c]		
	Total	37		

a. MEHI < Stacking b. MEHI > Stacking c. MEHI = Stacking

Test Statistics[b]

	MEHI - Stacking
Z	-5,303[a]
Asymp. Sig. (2-tailed)	,000
Exact Sig. (2-tailed)	**,000**
Exact Sig. (1-tailed)	,000
Point Probability	,000

a. Based on negative ranks.
b. Wilcoxon Signed Ranks Test

Figura 3.9 Test de Wilcoxon aplicado a los resultados de *Stacking* y MEHI

La figura 3.9 muestra los resultados de comparar *Stacking* con MEHI, los cuales son muy positivos porque existe diferencia significativa y MEHI supera en el 100% de los casos a Stacking. La salida del SPSS de la comparación entre *Stacking* y *Bagging* se puede apreciar en la figura 3.10. Aquí se observa que no hay diferencias significativas entre *Stacking* y los

diferentes modelos de *Bagging*, con los diferentes CI como clasificadores base; sólo hay diferencias con *Bagging* utilizando *k*NN como clasificador de base, y en este caso es a favor de *Stacking*. Para realizar una comparación más justa, se comparó *Stacking* contra el mejor resultado de *Bagging* para cada base y en este caso sí da diferencias significativas a favor de *Bagging*. Es por esto que se procede a comparar el mejor de *Bagging* contra MEHI.

Ranks

		N	Mean Rank	Sum of Ranks
Stacking - Bagging_IBK	Negative Ranks	2[a]	1,50	3,00
	Positive Ranks	6[b]	5,50	33,00
	Ties	1[c]		
	Total	9		
Stacking - Bagging_SMO	Negative Ranks	2[d]	3,50	7,00
	Positive Ranks	6[e]	4,83	29,00
	Ties	1[f]		
	Total	9		
Stacking - Bagging_MLP	Negative Ranks	2[g]	4,50	9,00
	Positive Ranks	7[h]	5,14	36,00
	Ties	0[i]		
	Total	9		
Stacking - Bagging_ Bayes	Negative Ranks	4[j]	5,25	21,00
	Positive Ranks	5[k]	4,80	24,00
	Ties	0[l]		
	Total	9		
Stacking - Bagging_J48	Negative Ranks	6[m]	4,83	29,00
	Positive Ranks	2[n]	3,50	7,00
	Ties	1[o]		
	Total	9		
Stacking - Max_Bagging	Negative Ranks	9[p]	5,00	45,00
	Positive Ranks	0[q]	,00	,00
	Ties	0[r]		
	Total	9		

a. Stacking < Bagging_IBK
b. Stacking > Bagging_IBK
c. Stacking = Bagging_IBK
d. Stacking < Bagging_SMO
e. Stacking > Bagging_SMO
f. Stacking = Bagging_SMO
g. Stacking < Bagging_MLF
h. Stacking > Bagging_MLF
i. Stacking = Bagging_MLP
j. Stacking < Bagging_Bay
k. Stacking > Bagging_Bay
l. Stacking = Bagging_Bay
m. Stacking < Bagging_J48
n. Stacking > Bagging_J48
o. Stacking = Bagging_J48
p. Stacking < Max_Bagging
q. Stacking > Max_Bagging
r. Stacking = Max_Bagging

Test Statistics[c]

	Stacking - Bagging_IBK	Stacking - Bagging_SMO	Stacking - Bagging_MLP	Stacking - Bagging_ Bayes	Stacking - Bagging_J48	Stacking - Max_Bagging
Z	-2,100[a]	-1,540[a]	-1,599[a]	-,178[a]	-1,542[b]	-2,668[b]
Asymp. Sig. (2-tailed)	,036	,123	,110	,859	,123	,008
Exact Sig. (2-tailed)	**,039**	,148	,129	,910	,133	**,004**
Exact Sig. (1-tailed)	,020	,074	,064	,455	,066	,002
Point Probability	,008	,020	,016	,045	,008	,002

a. Based on negative ranks.
b. Based on positive ranks.
c. Wilcoxon Signed Ranks Test

Figura 3.10 Test de Wilcoxon aplicado a los resultados de *Stacking* y *Bagging*

La figura 3.11 muestra los resultados de comparar el mejor resultado de *Bagging* con MEHI. Aquí se obtienen diferencias significativas a favor de MEHI, como se muestra en las medias, por lo que se puede concluir que existen diferencias significativas entre los resultados obtenidos por MEHI y

el resto de los multiclasificadores, así como de los clasificadores simples, corroborando la superioridad de MEHI para este problema.

Ranks

		N	Mean Rank	Sum of Ranks
MEHI - Max_Bagging	Negative Ranks	0[a]	,00	,00
	Positive Ranks	9[b]	5,00	45,00
	Ties	0[c]		
	Total	9		

a. MEHI < Max_Bagging
b. MEHI > Max_Bagging
c. MEHI = Max_Bagging

Test Statistics[b]

	MEHI - Max_ Bagging
Z	-2,666[a]
Asymp. Sig. (2-tailed)	,008
Exact Sig. (2-tailed)	**,004**
Exact Sig. (1-tailed)	,002
Point Probability	,002

a. Based on negative ranks.
b. Wilcoxon Signed Ranks Test

Figura 3.11 Test de Wilcoxon aplicado a los resultados de *MEHI* y el mejor resultado de *Bagging*

3.5. Validación de las herramientas para el preprocesamiento de datos en problemas de clasificación

Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente [59]. En el presente epígrafe se describen los resultados de las pruebas realizadas a las dos herramientas desarrolladas.

Se desarrollaron pruebas de sistema, evaluando tanto la funcionalidad (pruebas funcionales) como los requisitos no funcionales, para estos últimos los requisitos que más se tuvo en cuenta fueron: el rendimiento, a partir de la importancia del tratamiento de ficheros con volúmenes importantes de datos y la fiabilidad en la transformación de las bases de casos.

Durante las pruebas funcionales se desarrollaron casos de pruebas basados en las Historias de Usuarios (HU) y se utilizaron tanto los métodos de caja negra como de caja blanca. Los casos de prueba correspondientes a las HU de la herramienta para el cálculo de las medidas de diversidad, se relacionan en la documentación generada por el desarrollo del proyecto que se encuentra en el portafolio digital del Departamento de Biometría del CISED, del mismo modo pueden consultarse los correspondientes a la herramienta para la transformación de bases.

Durante las pruebas no funcionales se utilizaron listas de chequeo. En ambos tipos de prueba se

describieron las No Conformidades (NC) detectadas.

3.5.1. Resultados de las pruebas realizadas a la herramienta para el cálculo de las medidas de diversidad

En la primera iteración, respecto a las pruebas funcionales, se detectaron las NC que a continuación se describen:

Código	Descripción
CD001	El sistema no verifica la estructura del formato a cargar y cuando se intenta abrir el fichero el sistema muestra un error de lectura de fichero
CD001	Al abrir un archivo correspondiente a una base de casos diferente a la que se había cargado antes se adicionan los nuevos clasificadores al final del menú desplegable.
CD002	Al eliminar un clasificador se eliminan sus datos del área de trabajo pero no se adiciona al menú de clasificadores.
CD003	Cuando se seleccionan dos clasificadores por primera vez no se habilita la opción “Calcular medidas de diversidad”.

En la segunda iteración se realizaron pruebas de regresión, verificando que las NC de la iteración anterior se resolvieron y no se introdujeron nuevas.

3.5.2. Resultados de las pruebas realizadas a la herramienta para la transformación de las bases

En la primera iteración, respecto a las pruebas funcionales, se detectaron las NC que a continuación se describen:

Código	Descripción
TB002	Cuando se transforma la base de casos en el área de trabajo se mantiene la base de casos original y no se muestra la base transformada.
TB003	La base de casos es exportada a la carpeta de la aplicación y no a la carpeta seleccionada por el usuario.

Se realizó el mismo procedimiento que en el caso anterior. Durante la segunda iteración no se detectaron nuevas NC.

Conclusiones del capítulo

El problema de predecir la resistencia de mutaciones de la proteína transcriptasa reversa del virus del VIH ante nueve fármacos inhibidores de su función, se modeló como un problema de clasificación, utilizando el genotipo para representar los rasgos del problema, siendo cada aminoácido de la secuencia un rasgo, y el fenotipo para representar las clases: resistente o

susceptible.

Las bases de casos utilizadas en el entrenamiento y validación de los modelos de clasificación, se obtuvieron a partir de la información disponible en la base de datos *Stanford*. Estas bases de casos se transformaron utilizando una herramienta que permite sustituir cada aminoácido por su energía de activación y luego exportarla al formato utilizado por Weka.

Se comprobó la efectividad de diez modelos de clasificación simples en la solución del problema planteado, basados en los clasificadores IBk, J48, BayesNet, SMO y MLP. Este último fue el de mejores resultados en la mayoría de las bases; aunque no fueron lo suficientemente adecuados en varios casos.

El cálculo de las medidas de diversidad se realizó mediante una herramienta desarrollada para este fin; ello facilitó la selección de los clasificadores de base a utilizar en los modelos de combinación de clasificadores: *Stacking* y MEHI. En el caso de los multiclasificadores *Bagging* y *Boosting*, cuya topología sólo utiliza un clasificador de base, se utilizaron cada uno de los diez clasificadores simples que fueron usados antes como clasificadores individuales.

Se compararon los resultados de los muticlasificadores contra los clasificadores individuales para solucionar el problema planteado, concluyendo que *Bagging* y *Boosting* tienen un comportamiento similar y en ocasiones inferior a los clasificadores simples; MEHI, por su parte, tiene una efectividad superior al mejor de sus clasificadores de base en la mayoría de los casos y además supera a *Stacking* en la totalidad de los casos.

Se propone como multiclasificador que mejora los resultados de predicción a MEHI, pero con la especificidad de variar los clasificadores de base según sea el inhibidor a analizar; teniendo en cuenta que su efectividad no se logra con un único modelo, sino que los clasificadores de base difieren para algunos fármacos. Los métodos estadísticos utilizados (Test de Friedman y Wilcoxon) demuestran la superioridad del multiclasificador MEHI en la solución al problema presentado.

CONCLUSIONES

En este trabajo se realiza un estudio y caracterización de los métodos de inteligencia artificial para clasificación y su utilización para predecir la resistencia del VIH. El uso de técnicas computacionales para este problema es de gran ayuda, ya que las pruebas de resistencia son muy costosas y complejas. En trabajos previos se han obtenido buenos resultados con el modelo de multiclasificador MEHI, para bases relacionadas con inhibidores de la proteasa. En este trabajo se prueban este y otros multiclasificadores para bases relacionadas con inhibidores de la transcriptasa inversa.

Se diseñaron e implementaron herramientas para la transformación de la base de datos Stanford a formato "arff" y el cálculo de las medidas de diversidad, que resultaron de gran utilidad en los procesos de construcción de la base de casos y la selección de los clasificadores de base, respectivamente.

Se realizaron experimentos, en la plataforma Weka, que evidenciaron que los resultados de los multiclasificadores *Stacking* y *MEHI* superan a los de los clasificadores individuales: IBk, J48, BayesNet, SMO y MLP, teniendo *Bagging y Boosting* un comportamiento similar a estos últimos. Se comprobó que los resultados de MEHI superan significativamente los resultados de Stacking, Bagging, Boosting y los clasificadores individuales, obteniéndose resultados de predicción superiores a los 94% en todas bases.

Los resultados de los diferentes modelos de clasificación fueron validados mediante el método de validación cruzada. Las pruebas estadísticas de Friedman y Wilcoxon respaldan la selección de MEHI como el modelo más conveniente para resolver la predicción de resistencia del VIH, utilizando un conjunto de clasificadores de base diferente en correspondencia con el inhibidor a analizar.

RECOMENDACIONES

1. Valorar la efectividad de la solución propuesta utilizando información sobre nuevos pares genotipo-fenotipo que no se encuentren en la base Stanford.

2. Desarrollar una herramienta que facilite la toma de decisiones sobre la combinación de fármacos para el tratamiento del VIH a partir de la predicción de resistencia de la transcriptasa inversa a cada uno de ellos, y los resultados ya obtenidos para la proteasa, utilizando los modelos propuestos.

3. Determinar otros multiclasificadores que mejoren los resultados para la base ABC con la que MEHI obtuvo sus más bajos resultados.

4. Utilizar el multiclasificar MEHI en problemas de clasificación de otras proteínas del VIH y secuencias genómicas de otros virus.

REFERENCIAS BIBLIOGRÁFICAS

[1] A. Kontijevskis*, et al.*, "A look inside HIV resistance through retroviral protease interaction maps," *PLOS Computational Biology,* vol. 3, pp. 424-435, 2007.

[2] I. Bonet, "Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial," Tesis de doctorado, Departamento de Ciencias de la Computación, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, 2008.

[3] (2005). *HIV Databases*. Available: http://www.hiv.lanl.gov

[4] (2011). *HIV Drug Resistance Database*. Available: http://hivdb.stanford.edu//cgi-bin/GenoPhenoDS.cgi

[5] N. Beerenwinkel*, et al.*, "Geno2pheno: estimating phenotypic drug resistance from HIV-1 genotypes," *Nucl. Acids Res.,* vol. 31, pp. 3850-3855, July 1, 2003 2003.

[6] N. Beerenwinkel*, et al.*, "Diversity and complexity of HIV-1 drug resistance: A bioinformatics approach to predicting phenotype from genotype," *PNAS,* vol. 99, pp. 8271-8276, Jun 11 2002.

[7] N. Beerenwinkel*, et al.*, "Computational methods for the design of effective therapies against drug resistant HIV strains," *Bioinformatics,* vol. 21, pp. 3943-3950, Nov 1 2005.

[8] R. J. Murray, "Predicting Human Immunodeficiency Virus Type 1 Drug Resistance from Genotype Using Machine Learning," MSc Thesis, University of Edinburgh, 2004.

[9] V. L. Ravich*, et al.*, "A combined sequence-structure approach for predicting resistance to the non-nucleoside HIV-1 reverse transcriptase inhibitor Nevirapine," *Biophysical Chemistry,* vol. 153, pp. 168-172, 2011.

[10] M. D. Marcin Kierczak, Jacek Koronacki, Jan Komorowski, "Computational Analysis of Molecular Interaction Networks Underlying Change of HIV-1 Resistance to Selected Reverse Transcriptase Inhibitors," *Bioinformatics and Biology Insights,* vol. 4, pp. 137-146, 2010.

[11] I. Grau, "Aprendizaje de Redes Neuronales Recurrentes con instancias de longitud variable. Aplicaciones a la resistencia antiviral del VIH.," Trabajo de Diploma, Universidad Central "Martha Abreu" de Las Villas, Santa Clara, 2011.

[12] J. A. Smith and R. Daniel, "Following the path of the virus: the exploitation of host DNA repair mechanisms by retroviruses.," *ACS Chem Biol,* vol. 4, pp. 217-226, 2006.

[13] NIAID. (2009). *Biology of HIV*. Available: http://www.niaid.nih.gov/topics/hivaids/understanding/biology/Pages/biology.aspx

[14] G. Kaiser. (2008, 2011). The Life Cycle of HIV. *Community College of Baltimore Country*. Available: http://student.ccbcmd.edu/courses/bio141/lecguide/unit3/viruses/hivlc.html
[15] H. Navarro. (2011, septiembre, 2011). TRATAMIENTO DE LA INFECCIÓN POR VIH. ATENCIÓN FARMACÉUTICA. Available: http//:www.academiadefarmaciadearagon.es/docs/Documentos/Documento36.pdf
[16] R. Hunt. (2009, HUMAN IMMUNODEFICIENCY VIRUS. ANTI-HIV CHEMOTHERAPY . APPENDIX III. Available: http://pathmicro.med.sc.edu/lecture/hiv14a.htm
[17] M. A. THOMPSON*, et al.*, "Antiretroviral Treatment of Adult HIV Infection.," presented at the Recommendations of the International AIDS Society–USA Panel., 2010.
[18] I. Bonet*, et al.*, "Predicting Human Immunodeficiency Virus Drug Resistance Using Recurrent Neural Networks," in *IWINAC 2007, LNCS 4527*, J. Mira and J. R. Alvarez, Eds., ed: Springer-Verlag Berlin Heidelberg, 2007, pp. 234-243.
[19] (2011, La resistencia del VIH y las pruebas de resistencia. *PROJECT INFORM*. Available: http://www.projectinform.org/publications/resistence_sp/
[20] D. Aha and D. Kibler, "Instance-based learning algorithms," *Machine Learning,* vol. 6, pp. 37-66, 1991.
[21] T. M. Mitchell, *Machine Learning*: McGraw-Hill Science/Engineering/Math; (March 1, 1997), 1997.
[22] A. Guerra-Hernández. (2010, agosto, 2011). Inducción de Arboles de Decisión. Available: http://www.uv.mx/aguerra/documents/2010-prolog-slides-12.pdf
[23] J. R. Quinlan, "Induction of decision trees," *Machine Learning,* vol. 1, pp. 81-106, 1986.
[24] J. R. Quinlan, *C4.5: Programs for Machine Learning.* San Mateo,CA: Morgan Kaufmann, 1993.
[25] R. Grau*, et al.*, "Boolean Algebraic Structures of the Genetic Code: Possibilities of Applications," in *KDECB 2006, LNBI 4366*, K. Tuyls, Ed., ed: Springer-Verlag Berlin Heidelberg, 2006, pp. 10-21.
[26] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning,* vol. 9, pp. 309-347, 1992.
[27] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2 ed. San Francisco: Diane Cerra, 2005.
[28] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[29] B. Hammer and T. Villmann, "Mathematical Aspects of Neural Networks," in *European Symposium on Artificial Neural Networks 2003*, 2003, pp. 59-72.
[30] J. R. Hilera and V. J. Martínez, *Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*. Madrid / Mexico: RA-MA / Addison Wesley Iberoamericana, 1995.
[31] E. Trentin. (2008, septiembre, 2011). Multilayer Perceptron (MLP): the Backpropagation (BP) Algorithm. Available: http://www.uow.edu.au/~markus/teaching/CSCI323/Lecture_MLP.pdf
[32] B. Pearlmutter, "Dynamic Recurrent Neural Networks," *DARPA Research,* 1990.
[33] P. Baldi, *New Machine Learning Methods for the Prediction of Protein Topologies*: In: P. Frasconi and R. Shamir (eds.) Artificial Intelligence and Heuristic Methods for Bioinformatics, IOS Press, 2002.
[34] S. Draghici and R. B. Potter, "Predicting HIV drug resistance with neural networks," *Bioinformatics,* vol. 19, pp. 98-107, January 1, 2003 2003.
[35] D. C. Wang and B. Larder, "Enhanced prediction of lopinavir resistance from genotype by use of artificial neural networks," *Journal Of Infectious Diseases,* vol. 188, pp. 653-660, Sep 1 2003.
[36] S.-Y. Rhee*, et al.*, "Genotypic predictors of human immunodeficiency virus type 1 drug resistance," *Proceedings of the National Academy of Sciences,* vol. 103, pp. 17355-17360, 8 November 14, 2006 2006.
[37] Z. W. Cao*, et al.*, "Computer prediction of drug resistance mutations in proteins," *Drug Discovery Today,* vol. 10, pp. 521-529, Apr 1 2005.
[38] M. Rabinowitz*, et al.*, "Accurate prediction of HIV-1 drug response from the reverse transcriptase and protease amino acid sequences using sparse models created by convex optimization," *Bioinformatics,* vol. 22, pp. 541-549, March 1, 2006 2006.
[39] K. G. Marcin Kierczak, Michał Dramiński, Jacek Koronacki, W. Rudnicki, Jan Komorowski, "A Rough Set-Based Model of HIV-1 Reverse Transcriptase Resistome," *Bioinformatics and Biology Insights,* 2009.
[40] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions," 1997, pp. 43-48.
[41] B. Efron and R. J. Tibshirani, "Improvements on cross-validation: the .632+ bootstrap method," vol. 92, ed, 1997, pp. 548 - 560.
[42] L. I. Kuncheva, *Combining Pattern Classifiers, Methods and Algorithms*. New York, NY: Wiley Interscience, 2004.
[43] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine,* vol. 6, pp. 21-44, 2006.
[44] T. K. Ho*, et al.*, "Decision combination in multiple classifier systems," *IEEE Trans. on Pattern Analy. Machine Intel.,* vol. 16, pp. 66-75, 1994.

[45] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. vol. 1857, ed Berlin: Springer-Verlag Berlin, 2000, pp. 1-15.
[46] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning,* vol. 51, pp. 181-207, 2003.
[47] G. Brown*, et al.*, "Diversity creation methods: a survey and categorisation," *Information Fusion,* vol. 6, pp. 5-20, 2005.
[48] S. T. Hadjitodorov*, et al.*, "Moderate diversity for better cluster ensembles," *Information Fusion,* vol. 7, pp. 264-275, 2006.
[49] S. Segrera and M. N. Moreno. (2006, 2011). Multiclasificadores: Métodos y Arquitecturas. Available: http://olmo.usal.es/publicacion/download_documento?id=3&locale=es
[50] J. Maudes, "Combinación de Clasificadores: Construcción de Características e Incremento de la Diversidad," Tesis de doctorado, Universidad de Burgos, 2010.
[51] J. Kittler*, et al.*, "On combining classifiers.," *IEEE Transactions on Pattern Analysis and Machine Intelligence,,* vol. 20, pp. 226-239, 1998.
[52] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, pp. 123-140, 1996.
[53] R. E. Schapire, "The strength of weak learnability," *Machine Learning,* vol. 5, pp. 197-227, 1990.
[54] Y. Freund and R. E. Schapire, "Decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences,* vol. 55, pp. 119–139, 1997.
[55] D. Wolpert, "Stacked generalization," *Neural Networks,* vol. 5, pp. 241-259, 1992.
[56] I. Bonet*, et al.*, "Ensemble of Classifiers Based on Hard Instances Pattern Recognition." vol. 6718, J. Martínez-Trinidad*, et al.*, Eds., ed: Springer Berlin / Heidelberg, 2011, pp. 67-74.
[57] A. S. Ayra and A. D. R. Reyes, "Propuesta de técnicas de estimación y métricas para la metodología ágil SXP," Trabajo de Diploma, Universidad de las Ciencias Informáticas, La Habana, 2009.
[58] R. Grau, *Estadística aplicada con ayuda de paquetes de software*. Guadalajara, Jalisco, México: Editorial Universitaria, 1994.
[59] "IEEE Standard Glossary of Software Engineering Terminology," *IEEE Std 610.12-1990,* p. 1, 1990.

ANEXOS

Anexo 1. Tabla de energías de contacto de los aminoácidos

Aminoácido	eir 1996	qi, 1996	**Energías** 1996 -0,6*qi*eir/2 kcal/mol
A	-2.57	6.334	4.883514
C	-3.57	6.646	7.117866
D	-1.84	6.487	3.580824
E	-1.79	6.235	3.348195
F	-4.76	5.870	8.382360
G	-2.19	6.284	4.128588
H	-2.56	6.241	4.793088
I	-4.42	6.042	8.011692
K	-1.52	6.569	2.995464
L	-4.81	6.087	8.783541
M	-3.92	6.137	7.217112
N	-1.92	6.574	3.786624
P	-2.09	5.858	3.672966
Q	-2.00	6.469	3.881400
R	-2.11	6.318	3.999294
S	-1.98	6.582	3.909708
T	-2.29	6.486	4.455882
V	-3.89	6.155	7.182885
W	-3.81	5.793	6.621399
Y	-3.41	6.037	6.175851

Tabla A.1.1 Energías de contacto.

Anexo 2. Resultados de predicción de resistencia a nueve fármacos del VIH con diferentes clasificadores simples

Bases ->	D4T	DDI	TDF	NVP	ABC	AZT	DLV	EFV	3TC
Clasificador	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta
ibk k=1	92,1569	97,0779	85,4369	73,8506	69,7674	80,0654	77,3256	74,1840	71,8447
ibk k=2	92,8105	96,1039	87,8641	70,4023	66,4452	79,7386	74,7093	72,7003	67,6375
ibk k=3	92,8105	96,1039	87,3786	73,2759	71,7608	83,6601	79,0698	73,5905	68,9320
ibk k=8	90,5229	95,7792	87,3786	65,5172	71,7608	83,3333	68,8953	67,9525	70,5502
SMO	94,1176	95,1299	86,8932	85,6322	73,0897	84,9673	85,4651	81,3056	74,1100
MLP cantN = 3	99,3464	99,6753	100,0000	81,3218	89,3688	96,4052	96,2209	94,0653	88,0259
MLP cantN = 5	99,3464	99,6753	99,5146	91,3793	86,7110	92,8105	95,9302	87,8338	86,7314
MLP cantN = 7	99,3464	99,6753	99,5146	93,1034	90,6977	98,0392	95,6395	89,9110	78,3172
bayesNet	93,7908	96,1039	84,4660	88,7931	78,7375	83,9869	88,3721	88,4273	87,0550
J48	89,2157	95,4545	85,4369	91,6667	76,0797	82,6797	86,6279	88,7240	92,2330

Tabla A.2.1 Resultados obtenidos para cada base con los clasificadores individuales.

Anexo 3. Resultados de predicción de resistencia a nueve fármacos del VIH con diferentes topologías de multiclasificadores

Bases ->		D4T	DDI	TDF	NVP	ABC	AZT	DLV	EFV	3TC
Multiclasificador	Clasificador de base	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta
Bagging	ibk k=1	92,4837	97,0779	88,3495	75,2874	69,1030	81,0458	77,3256	73,8872	70,5502
	ibk k=2	92,4837	96,7532	88,3495	76,7241	70,4319	81,6993	76,4535	74,4807	72,4919
	ibk k=3	92,4837	96,1039	87,3786	74,4253	72,4252	84,6405	75,5814	73,5905	74,4337
	ibk k=8	91,1765	96,1039	86,4078	68,1034	72,4252	84,6405	70,0581	72,1068	75,4045
	SMO	93,4641	95,4545	86,8932	85,9195	73,0897	85,6209	85,7558	81,0089	76,0518
	MLP cantN = 3	93,1373	96,1039	86,4078	83,9080	73,4219	86,2745	84,8837	80,1187	75,7282
	MLP cantN = 5	92,8105	96,1039	86,4078	83,9080	75,0831	84,6405	84,3023	81,0089	73,4628
	MLP cantN = 7	92,8105	96,1039	85,9223	85,3448	75,7475	84,6405	83,7209	79,5252	76,3754
	bayesNet	94,1176	96,1039	84,4660	90,2299	79,7342	84,1370	87,7907	88,7240	87,7023
	J48	91,8301	95,7792	88,8350	93,1034	80,7309	84,6405	87,5000	89,6142	93,2039

Tabla A.3.1 Resultados obtenidos para cada base con el multiclasificador Bagging.

Bases ->		D4T	DDI	TDF	NVP	ABC	AZT	DLV	EFV	3TC
Multiclasificador	Clasificador de base	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta	% clasificación correcta
Boosting	ibk k=1	92,1569	97,0779	85,4369	73,8506	69,7674	80,0654	77,3256	74,1840	71,8447
	ibk k=2	92,4837	96,1039	82,0388	76,1494	65,1163	79,4118	76,7442	75,6677	67,6375
	ibk k=3	90,5229	96,1039	83,0097	75,0000	68,1063	81,3725	79,3605	74,7774	67,6375
	ibk k=8	91,1765	93,5065	83,0097	71,8391	65,4485	83,9869	76,4535	68,8427	67,6375
	SMO	90,5229	96,7532	82,0388	82,1839	70,0997	81,6993	82,5581	73,5905	69,5793
	MLP cantN = 3	92,8105	96,1039	83,4951	81,3218	71,7608	82,3529	83,1395	81,3056	69,9029
	MLP cantN = 5	92,4837	96,1039	83,4951	79,3103	74,7508	82,3529	84,5930	78,0415	70,8738
	MLP cantN = 7	92,4837	96,1039	85,9223	81,3218	68,7708	83,6601	85,1744	79,8220	68,2848
	bayesNet	95,0980	96,1039	87,8641	88,7931	79,7342	83,9869	88,3721	88,4273	91,5858
	J48	93,4641	97,0779	84,9515	95,6897	75,4153	82,0261	87,7907	91,9881	94,4984

Tabla A.3.2 Resultados obtenidos para cada base con el multiclasificador Boosting.

←Bases	Multi-clasificador	Clasificadores de base y meta-clasificador	% clasificación correcta	Multi-clasificador	Clasificadores de base y meta-clasificador	% clasificación correcta
D4T	Stacking	ibk8, bayes, mlp5 meta mlp3	93,4641	MEHI	ibk8, bayes, mlp5 meta mlp3	100,0000
		ibk8, bayes, mlp3 meta mlp3	93,1373		ibk8, bayes, mlp3 meta mlp3	99,6732
		j48, bayes, mlp3 meta mlp3	92,8105		j48, bayes, mlp3 meta mlp3	99,6732
		ik2, ibk8, bayes meta mlp3	92,8105		ik2, ibk8, bayes meta mlp3	98,6928
DDI		ibk3, smo, j48, meta mlp3	96,1039		ibk3, smo, j48, meta mlp3	99,0260
		ibk3, smo, mlp3, meta mlp3	95,7792		ibk3, smo, mlp3, meta mlp3	99,0260
		ibk3, j48, mlp3 meta mlp3	96,1039		ibk3, j48, mlp3 meta mlp3	99,6753
		j48, bayes, mlp3 meta mlp3	96,4286		j48, bayes, mlp3 meta mlp3	99,3506
TDF		smo,j48,mlp7 meta mlp3	86,8932		smo,j48,mlp7 meta mlp3	99,0291
		ibk2, smo, mlp7 meta mlp3	87,8641		ibk2, smo, mlp7 meta mlp3	99,5146
		ibk1, j48, bayes meta mlp3	86,4078		ibk1, j48, bayes meta mlp3	99,5146
		j48, bayes, mlp7 meta mlp3	85,9223		j48, bayes, mlp7 meta mlp3	100,0000

Tabla A.3.3 Resultados obtenidos con los multiclasificadores Stacking y MEHI para las bases TDF, DDI y D4T.

←Bases	Multi-clasificador	Clasificadores de base y meta-clasificador	% clasificación correcta	Multi-clasificador	Clasificadores de base y meta-clasificador	% clasificación correcta
NVP	Stacking	ibk2, j48, mlp3 meta mlp3	90,8046	MEHI	ibk2, j48, mlp3 meta mlp3	98,2759
		ibk1, smo, j48 meta mlp3	91,6667		ibk1, smo, j48 meta mlp3	96,5517
		smo,j48,mlp5 meta mlp3	91,6667		smo,j48,mlp5 meta mlp3	95,9770
ABC		ibk3, j48, mlp3, meta mlp 3	75,7475		ibk3, j48, mlp3, meta mlp 3	90,0332
		mlp3, j48, mlp7, meta mlp7	74,4186		mlp3, j48, mlp7, meta mlp7	94,0199
		ibk3, bayes, mlp7 meta mlp5	76,0797		ibk3, bayes, mlp7 meta mlp5	92,6910
		ibk3, smo, j48, meta mlp3	75,4153		ibk3, smo, j48, meta mlp3	88,7043
		ibk3, j48, bayes, meta mlp3	76,7442		ibk3, j48, bayes, meta mlp3	91,6944
AZT		ibk2, bayes, mlp3, meta mlp3	84,6405		ibk2, bayes, mlp3, meta mlp3	96,7320
		ibk2, bayes, mlp3, meta smo	82,0261		ibk2, bayes, mlp3, meta smo	95,7516
		ibk2, j48, mlp3, meta mlp3	84,6405		ibk2, j48, mlp3, meta mlp3	97,3856
		ibk2, j48, mlp3, meta smo	81,6993		ibk2, j48, mlp3, meta smo	97,0588

Tabla A.3.4 Resultados obtenidos con los multiclasificadores Stacking y MEHI para las bases AZT, ABC y NVP.

←Bases	Multi-clasificador	Clasificadores de base y meta-clasificador	% clasificación correcta	Multi-clasificador	Clasificadores de base y meta-clasificador	% clasificación correcta
DLV	Stacking	ibk2, ibk8, mlp3 meta mlp3	82,8488	MEHI	ibk2, ibk8, mlp3 meta mlp3	96,8023
		ibk2, j48, mlp3 meta mlp3	84,3023		ibk2, j48, mlp3 meta mlp3	97,0930
		ibk2, bayes, mlp3 meta mlp3	83,4302		ibk2, bayes, mlp3 meta mlp3	97,3837
		ibk2, smo, mlp3 meta mlp3	82,8488		ibk2, smo, mlp3 meta mlp3	94,4767
EFV		ibk2, bayes, mlp3 meta mlp3	88,1306		ibk2, bayes, mlp3 meta mlp3	97,0326
		ibk2, j48, mlp3 meta mlp3	87,8338		ibk2, j48, mlp3 meta mlp3	97,3294
		ibk3, bayes, mlp7 meta mlp3	86,6469		ibk3, bayes, mlp7 meta mlp3	94,9555
3TC		ibk3, bayes, mlp7, meta mlp 3	87,7023		ibk3, bayes, mlp7, meta mlp 3	91,5858
		ibk3, bayes, mlp7, meta smo	86,7314		ibk3, bayes, mlp7, meta smo	95,4693
		ibk3, j48, mlp5, meta mlp 3	91,9094		ibk3, j48, mlp5, meta mlp 3	99,0291
		ibk3, j48, mlp5, meta smo	92,2330		ibk3, j48, mlp5, meta smo	99,0291
		ibk1, j48, bayes, meta mlp 3	90,6149		ibk1, j48, bayes, meta mlp 3	99,3528
		ibk1, j48, bayes, meta smo	92,2330		ibk1, j48, bayes, meta smo	99,3528

Tabla A.3.5 Resultados obtenidos con los multiclasificadores Stacking y MEHI para las bases 3TC, EFV y DLV