Aplicación de minería de datos y modelamiento matemático en ingeniería de proteínas

Diseño e implementación de nuevas metodologías para el estudio de mutaciones

UNIVERSIDAD DE CHILE

David Medina Ortiz

Supervisor: Dr. Álvaro Olivera

Departamento de Ingeniería Química, Biotecnología y Materiales
Universidad de Chile

Este trabajo es para obtener el grado de
Dr. en Ciencias de la Ingeniería

June 2019

Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ...

Tabla de contenidos

Lista de figuras

Lista de tablas

Chapter 1

Aplicaciones de la minería de datos en ingeniería de proteínas

Chapter 2

Modelos predictivos asociados a mutaciones puntuales en proteínas

El análisis del efecto de mutaciones puntuales en proteínas, es una de las problemáticas más estudiadas en los últimos años. Los estudios se enfocan principalmente, en la evaluación de cambios en la estabilidad de la proteína mediante la variación de energía libre que la mutación provoca [87, 74, 85, 75].

Diferentes modelos predictivos han sido desarrollados para poder predecir cambios de energía libre, en base a algoritmos de aprendizaje supervisado o mediante técnicas de minería de datos, y así, determinar el efecto de la mutación en set de proteínas de interés [81, 22, 16, 54, 91, 44, 21]. No obstante, en casos más específicos, se han desarrollado modelos para proteínas independientes, con el fin de asociar la mutación a un rasgo clínico, particularmente, enfocado a casos de cáncer [46, 40], cambios en termo estabilidad [90], propiedades geométricas [8], entre las principales.

Sin importar el uso o la respuesta de los modelos, es necesario construir set de datos con ejemplos etiquetados, es decir, cuya respuesta sea conocida para poder entrenar modelos basados en algoritmos de aprendizaje supervisado y así evaluar su desempeño. Los enfoques principales al desarrollo de descriptores se basan en propiedades fisicoquímicas y termodinámicas, así como también, el ambiente bajo el cual se encuentra la mutación [21], ya sea a partir de la información estructural o sólo considerando la secuencia lineal. Sin embargo, no son considerados, los componentes asociados a conceptos filogenéticos y la propensión a cambios de dicha mutación generando un gap entre ambos puntos de vista [73].

Dado a los modelos existentes y en vista a la necesidad de generar nuevos sistemas de predicción para mutaciones puntuales en proteínas, en respuesta al aumento considerable de reportes en los últimos años, se propone una nueva metodología para el diseño e implementación de modelos predictivos en mutaciones puntuales de proteínas.

Las mutaciones son descritas desde los puntos de vista estructural, termodinámico y filogenético. El desarrollo de los predictores es inspirados en el concepto de Meta Learning y es apoyado con técnicas estadísticas, tanto para la selección de modelos como para la evaluación de medidas de desempeño, entregando como resultado, un conjunto de modelos para las mutaciones puntuales reportadas unificados en un único meta modelo.

Esta metodología será aplicada para generar estimadores en diferentes proteínas con mutaciones reportadas con respuesta conocida, como por ejemplo: evaluando las diferencias de energía libre que provoca la mutación y clasificaciones para evaluar si la sustitución de residuos aumenta o disminuye la estabilidad. A su vez, se implementarán modelos de clasificación para determinar la propensión clínica en un conjunto de mutaciones conocidas relacionados con el gen *p*VHL, responsable de la enfermedad von Hippel Lindau, con el fin de exponer la versatilidad de la metodología.

A continuación, se describen los principales conceptos relacionados a minería de datos y aprendizaje supervisado, seguido de algunas herramientas computacionales para el análisis de mutaciones y su relevancia en la de estabilidad de una proteína, continuando con la metodología propuesta, la caracterización de los diferentes set de datos a utilizar y resultados parciales obtenidos al aplicar esta metodología.

2.1 Minería de datos

Minería de datos es el proceso de descubrimiento de patrones en set de datos, involucrando métodos asociados a Machine Learning, Estadísticas y sistemas de bases de datos. [48]. La minería de datos es un subcampo interdisciplinario de la informática, el cual tiene por objetivo general extraer información (a través de métodos inteligentes) de un conjunto de datos y transformar la información en una estructura comprensible para su uso posterior. [38, 34]. La minería de datos es el paso de análisis del proceso de *descubrimiento de conocimiento en bases de datos*, o KDD. [37]. Además del análisis en bruto de los datos, también incluye aspectos de manipulación de bases de datos y pre procesamiento de datos, evaluaciones de modelo e inferencia, métricas de interés, consideraciones de complejidad, post procesamiento de estructuras descubiertas, visualización y actualización de la información [11].

En la Figura 2.1, se exponen las principales ramas que componen la minería de datos y los diferentes procesos que se asocian a dichas ramas.

Son tres las principales áreas que abarca la minería de datos: Estadística, Inteligencia Artificial y Manipulación de sistemas de información. Por otro lado, son distintos procesos los que interactúan entre estas ramas, tales como: Modelamiento Matemático, reconocimiento de patrones, Sistemas de almacenamiento persistente y machine learning [48].

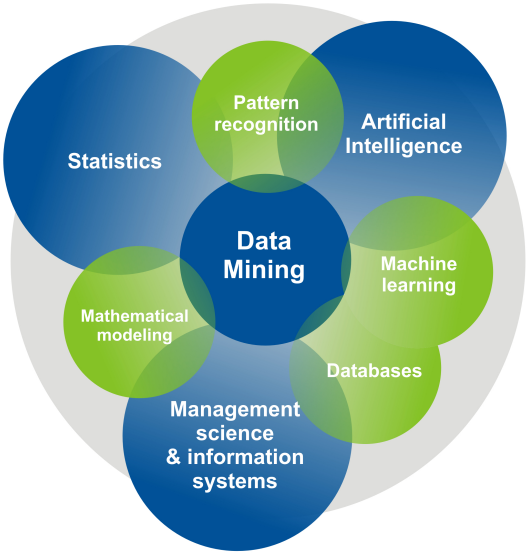


Fig. 2.1 Componentes en la minería de datos

Cada área en particular tiene un objetivo general y diversos objetivos específicos. Sin embargo, estas áreas interactúan entre sí, con el fin de poder extraer patrones de información que generen conocimientos a partir de la data de procesada [11].

La minería de datos se utiliza en diferentes campos, tales como: genética y genómica [56, 84], ingeniería de proteínas [47, 58, 59], comercio y negocios [50], sistemas de tránsito [61], optimizaciones en procesos industriales [28, 43, 14], reconocimiento de patrones [51, 36], rasgos cuantificables en enfermedades [96, 72, 32] y más recientemente en áreas de dinámicas moleculares [25, 95] y parámetros para la generación de pipe lines automatizados de simulaciones cuánticas en sistemas químicos [64, 30, 83].

2.2 Aprendizaje de Máquinas

Aprendizaje de Máquina, es una rama de la inteligencia artificial que tiene por objetivo el desarrollo de técnicas que permitan a los computadores aprender, es decir, generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos [67]. Aplicándose en diferentes campos de investigación: motores de búsqueda [29], diagnósticos médicos [27, 2], detección de fraude en el uso de tarjetas de crédito, bioinformática [55], reconocimiento de patrones en imágenes [35] y textos [71, 6], etc.

Los algoritmos de aprendizaje pueden clasificarse en dos grandes grupos [67]:

- **Supervisados**: se cumple un rol de predicción, clasificación, asignación, etc. a un conjunto de elementos con características similares, por lo que los datos de entrada son conocidos.

- **No Supervisados**: su objetivo es agrupar en conjuntos con características similares los elementos de entrada dado los valores de estos atributos, en base a la asociación de patrones característicos que representen sus comportamientos.

A continuación se describen en forma general, los algoritmos de aprendizaje supervisados utilizados para el desarrollo de la metodología, explicando los conceptos bajo los que se basan y cómo estos entrenan y se emplean para predecir o clasificar nuevos ejemplos.

2.2.1 Algoritmos de aprendizaje supervisado

Existen diferentes algoritmos de aprendizaje supervisado, los cuales pueden ser asociados a la clasificación de un elemento o la predicción de valores, dependiendo el tipo de respuesta existente en el conjunto de datos a estudiar. En el caso de respuestas con distribución continua, se trabajan con algoritmos de regresión, mientras que si la respuesta es binaria o multiclase y es representada por variables categóricas, los algoritmos se basan en clasificadores [67].

A su vez, también se pueden dividir con respecto a la forma en que se trata el problema, existiendo algoritmos basados en cálculos de distancia entre ejemplos (K-Vecinos Cercanos), otros que consideran transformaciones vectoriales y aplicaciones de funciones de kernel (Máquina Soporte de Vectores), así como también el uso de las características como entorno espacial de decisión (Árboles y métodos de ensamble) y aquellos que utilizan redes neuronales y trabajan en torno a cajas negras, o métodos basados en regresiones lineales, sólo aplicados a modelos predictivos de variables continuas.

Cada uno de estos algoritmos es descrito a continuación, enfocándose tanto en el componente matemático asociado, así como también en las ventajas y usos posibles que estos puedan tener, con respecto al conjunto de datos a trabajar.

2.2.2 K-Vecinos Cercanos

Algoritmo de aprendizaje supervisado, el cual tiene por objetivo asociar un elemento a una clase en particular, dada la información de ejemplos de entrada que tengan asociadas características particulares, que puedan declararse como *vecinos* del nuevo ejemplo a clasificar, siendo **k** el número de vecinos que se está dispuesto a utilizar para aplicar la clasificación [53]. La mejor elección de **k** depende fundamentalmente de los datos; generalmente, valores

grandes de **k** reducen el efecto de ruido en la clasificación, pero crean límites entre clases parecidas.

Con el fin de evaluar la cercanía de los ejemplos existentes contra el nuevo ejemplo a clasificar es necesario asociar ciertas medidas de distancia que permitan cuantificar esta cartacterística, para así poder comparar esta distancia y evaluar la cercanía para asociarle una clase a este nuevo ejemplo [33]. La distancia a emplear para evaluar la cercanía puede ser: Euclidiana [31], Manhattan [77], coseno [60], Mahalanobis [62], entre las principales, las cuales son expuestas de manera general en la Tabla 2.1.

Distancia	**Fórmula**	**Descripción**
Euclideana	$D_{(X,Y)} = \sqrt{\sum_{i=1}^{l}(X_i - Y_j)^2}$	Se basa en una recta entre dos puntos
Coseno	$D_{(X,Y)} = \arccos(\frac{X^T Y}{\|\|X\|\| \ \|\|Y\|\|})$	Se basa en vectores y en el coseno del ángulo que forman
Manhattan	$D_{(X,Y)} = \sum_{i=1} n\|X_i - Y_i\|$	Distancia en forma de zig-zag
Mahalanobis	$D_{(X,Y)} = \sqrt{(X-Y)^T S^{-1}(X-Y)}$	Considera las correlaciones entre las variables de estudio

Table 2.1 Resumen tipos de distancias utilizadas en procesos de comparación de ejemplos

K-Nearest Neighbors (KNN por su descripción en inglés), presenta algunos problemas, tales como: posibles errores al existir más de un elemento de distinta clase cercano al nuevo ejemplo a clasificar. Sin embargo, dicho error estimado es reducido [53].

Existen dos variaciones para la aplicación de KNN: aplicación basada en las distancias y aplicación basada en radios con respecto a puntos, la primera es mayormente usada. No obstante, en el caso de que los puntos no se encuentren uniformemente distribuidos es una mejor opción usar la segunda alternativa, siendo muy eficaz en problemas conocidos como *la maldición de la dimensionalidad*.

KNN utiliza el componente de peso [89], es decir, valores asignados a puntos específicos para determinar si un elemento a clasificar es de una clase o no, normalmente se utilizan pesos uniformes, sin embargo, es posible asignar valores de tal manera que al momento de realizar la votación puntos más cercanos en base a distancias presenten más peso que otros.

Se han implementando diversos algoritmos a la hora de aplicar la técnica de KNN, los cuales tienen relación con el coste computacional que presentan, dentro de estos se encuentran: Brute Force, K-D Tree y Ball Tree [76].

Este algoritmo de aprendizaje supervisado, puede ser utilizado tanto para el entrenamiento de modelos de clasificación (respuestas categóricas) y de regresión (respuestas continuas).

2.2.3 Naive Bayes

Naive Bayes es un conjunto de algoritmos de aprendizaje supervisados basados en la aplicación del teorema de Bayes con la suposición "ingenua" de independencia entre cada par de características [97]. Dada una variable de clase y y un vector de característica dependientes de la forma $x_1, ..., x_n$, el teorema de Bayes establece la siguiente relación:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n|y)}{P(x_1, \ldots, x_n)}$$

Utilizando la suposición ingenua de independencia de características, se tiene que:

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y)$$

Para todo i, esta relación se simplifica a:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \ldots, x_n)}$$

Dado que $P(x_1, \ldots, x_n)$ es constante dada la entrada, se puede utilizar la siguiente regla de clasificación:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

A pesar de sus supuestos aparentemente simplificados, los clasificadores de Naive Bayes han funcionado bastante bien en muchas situaciones del mundo real, la famosa clasificación de documentos y el filtrado de spam son ejemplos de ello [57, 26, 66]. Requieren una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios. Pueden ser extremadamente rápido en comparación con métodos más sofisticados. El desacoplamiento de las distribuciones de las características condicionales de clase significa que cada distribución se puede estimar de forma independiente como una distribución unidimensional. Esto a su vez ayuda a aliviar los problemas derivados de la dimensionalidad.

Existen distintos tipos de clasificadores de Naive Bayes, diferenciándose entre sí en la función de distribución de probabilidad que utilizan [66, 52, 63], dentro de los que se encuentran:

- **Gaussian Naive Bayes.**

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

- **Multinomial Naive Bayes.**

  La distribución se parametriza por el vector $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ para cada clase y, donde n es el número de características y θ_{y1} es la probabilidad $P(x_i \mid y)$ de que la característica i aparezca en una muestra que pertence a la clase y.

  Cada θ_y es estimado por:

$$\hat{\theta}_{yi} = \frac{N_{yi}+\alpha}{N_y+\alpha n}$$

  Donde $N_{yi} = \sum_{x \in T} x_i$ es el número de veces que aparece la característica i en la muestra de clase y en el set de entrenamiento T y $N_y = \sum_{i=1}^{|T|} N_{yi}$ representa el total de todas las características para la clase.

- **Bernoulli Naive Bayes.**

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

2.2.4 Árboles de Decisión

Se define árbol de decisión como un modelo de predicción utilizado en el ámbito de la inteligencia artificial, en el cual, dado un conjunto de datos se fabrican diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema. Aprendizaje basado en árboles de decisión es un método comúnmente utilizado en la minería de datos, cuyo objetivo consiste en desarrollar un modelo de predicción para el valor de una variable de destino en función de diversas variables de entrada [41].

El aprendizaje basado en árboles de decisión utiliza un árbol como un modelo predictivo que mapea las observaciones de las características que presenta un elemento. En estas estructuras de árbol, las hojas representan etiquetas de conjuntos ya clasificados, los nodos, a su vez, nombres o identificadores de los atributos y las ramas representan posibles valores para dichos atributos, a modo de ejemplo, se expone en la Figura 2.2, una representación de un posible árbol, el cual fue desarrollado para entrenar modelos de clasificación utilizando el set de datos iris [39].

Un árbol de decisión es una representación simple para clasificar ejemplos, el aprendizaje basado en esta metodología es una de las técnicas más eficientes para la clasificación supervisada. Donde cada ejemplo consta de atributos con valores discretos dentro de un dominio de conjunto finito, y existe un sólo término final denominado clasificación. En

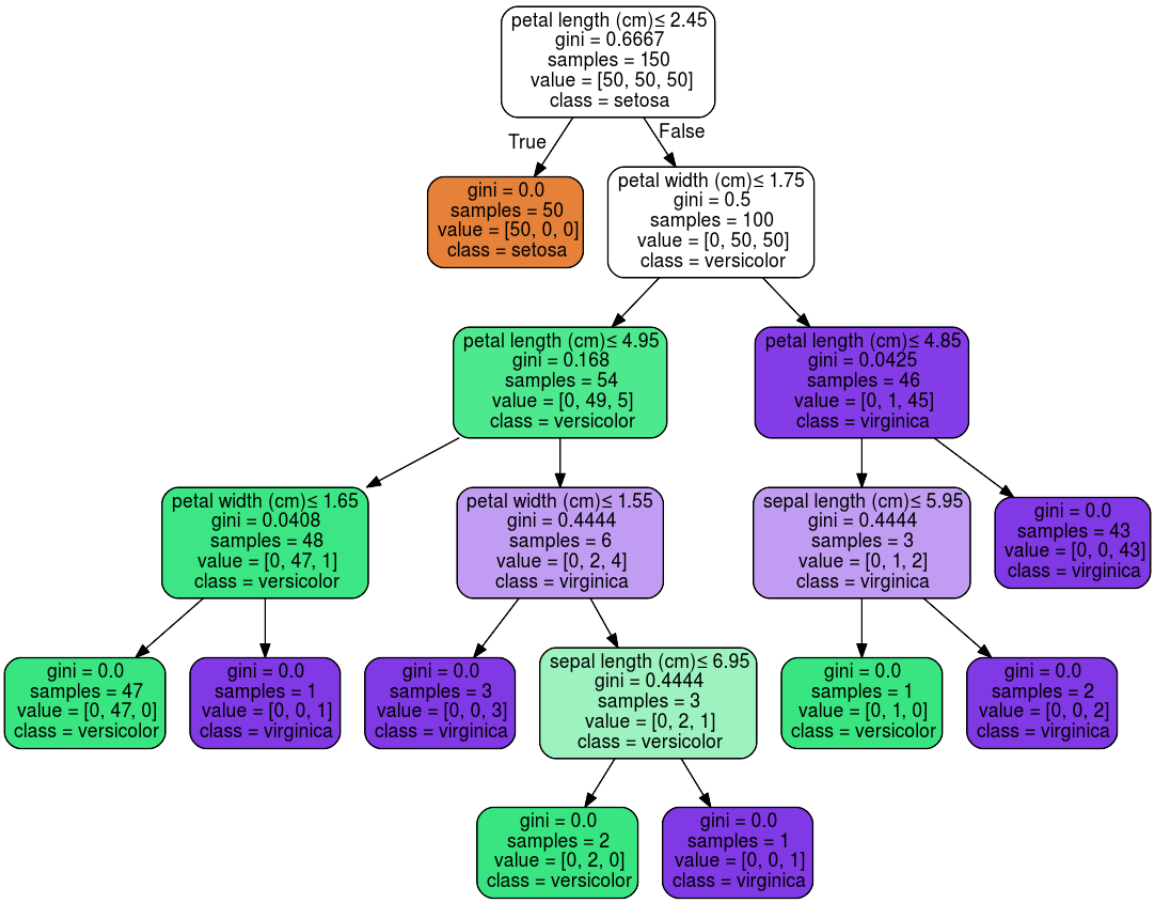


Fig. 2.2 Estructura de árbol para modelo de clasificación de set de datos iris.

un árbol de decisión, cada elemento del dominio de la clasificación se llama clase, cada nodo interno (no hoja) está etiquetado con una función de entrada, las ramas procedentes de un nodo etiquetado con una característica están asociados con cada uno de los posibles valores de la característica. Cada hoja del árbol se marca con una clase o una distribución de probabilidad sobre las clases [12].

Un árbol puede ser entrenado mediante el fraccionamiento del conjunto inicial en subconjuntos basados en una prueba de valor de atributo. Este proceso se repite en cada subconjunto derivado de una manera recursiva llamada particionamiento recursivo. La recursividad termina cuando el subconjunto en un nodo tienen todos el mismo valor de la variable objetivo, o cuando la partición ya no agrega valor a las predicciones.

Para cada división, es necesario el uso de una función que entregue una medida de impureza en cada división, esto con el objetivo de seleccionar la mejor partición para un atributo dado, la elección de dicho atributo se basa en el objetivo de separar de mejor manera los ejemplos.

La selección de los atributos se basa en qué atributo al momento de clasificar genera nodos más puros, para ello se utiliza una función de ganancia de información, la cual representa la ganancia obtenida a partir de una división de los ejemplos de entrenamiento [15].

Existen diferentes tipos de algoritmos basados en árboles de decisión, los cuales se exponen brevemente en la Tabla resumen .

Una definición matemática, tanto del proceso de clasificación o regresión y cómo son los criterios de selección de atributos es expuesta a continuación.

Sea $x_i \in R^n$ los vectores de entrenamiento del conjunto de datos y sea $y \in R^l$ el vector de respuestas asociadas a cada ejemplo, un árbol de decisión divide el espacio de forma recursiva, de manera que las muestras con las mismas etiquetas se agrupan.

Cada nodo m puede ser representado por Q y sea $\theta = (j, t_m)$ la división candidata para un atributo j y un umbral t_m, se definen las particiones $Q_{left}(\theta)$ y $Q_{rigth}(\theta)$ tal que:

$$\begin{aligned} Q_{left}(\theta) &= (x,y)|x_j <= t_m \\ Q_{right}(\theta) &= Q \setminus Q_{left}(\theta) \end{aligned} \tag{2.1}$$

Asociado a las divisiones, se tiene que, cada nodo generado se mide con respecto a la impureza de éste, la cual puede ser representada por una función $H()$ y a la ganancia de información que genera la división $G(Q,\theta)$, la cual se estima como:

$$G(Q,\theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)$$

Los descriptores se seleccionan con respecto a aquel que minimice la impureza de los nodos:

$$\theta^* = \text{argmin}_\theta \, G(Q,\theta)$$

Finalmente, se tiene que para los subconjuntos $Q_{left}(\theta^*)$ y $Q_{rigth}(\theta^*)$ la profundidad máxima se alcanza si:

- $N_m < min_{samples}$
- $N_m = 1$

Con N_m representando el número de nodos m.

Los criterios de clasificación se basan en la proporción de las clases según sus observaciones y en la función de impureza que es posible utilizar.

Si el vector de respuestas, es asociado a variables categóricas y toma valores entre $0, 1, \cdots, k-1$ para un nodo m, representando una región R_m con N_m ejemplos, se tiene que la proporción de observaciones de clase k en un nodo m puede definirse como:

$$p_{mk} = 1/N_m \sum_{x_i \in R_m} I(y_i = k)$$

Con respecto a las diferentes funciones de impureza $H()$ que pueden ser utilizadas se tienen las siguientes, descritas en la Tabla 2.2.

Función	Fórmula
Gini	$H(X_m) = \sum_k p_{mk}(1 - p_{mk})$
Entropía	$H(X_m) = -\sum_k p_{mk} \log(p_{mk})$
Misclassification	$H(X_m) = 1 - \max(p_{mk})$

Table 2.2 Tipos de medidas de impureza que pueden ser utilizadas en árboles de decisión para modelos de clasificación.

Siendo X_m datos de entrenamiento en el nodo m.

A la hora de entrenar modelos de regresión, es decir, con respuestas con distribución continua, se tiene que para el nodo m, el cual representa una región R_m con observaciones N_m, los criterios comunes para minimizar errores en futuras divisiones son el Error cuadrático medio y el Error absoluto medio, quienes minimizan el error tipo II y el error tipo I, respectivamente.

Estos se pueden definir como:

- **Error cuadrático medio**:

$$\begin{aligned} \bar{y}_m &= \frac{1}{N_m} \sum_{i \in N_m} y_i \\ H(X_m) &= \frac{1}{N_m} \sum_{i \in N_m} (y_i - \bar{y}_m)^2 \end{aligned} \tag{2.2}$$

- **Error absoluto medio**:

$$\begin{aligned} \bar{y}_m &= \frac{1}{N_m} \sum_{i \in N_m} y_i \\ H(X_m) &= \frac{1}{N_m} \sum_{i \in N_m} |y_i - \bar{y}_m| \end{aligned} \tag{2.3}$$

2.3 Herramientas computacionales asociadas a evaluación de mutaciones

Las herramientas computacionales asociadas a la evaluación de mutaciones puntuales se centran principalmente en el análisis de cómo ésta afecta a la estabilidad o la predicción de energía libre asociada a los residuos involucrados en la mutación. Sin embargo, a pesar de que el objetivo es el mismo, se enfocan en diferentes puntos de vista para abordar la problemática, tanto a nivel de entrenamiento de modelos, cómo manipulación de set de datos, así como las técnicas utilizadas para la predicción de los cambios de energía libre.

A continuación, se exponen algunas herramientas relacionadas con el estudio de estabilidad de proteínas, las cuales se aplicarán como métodos de comparación para los resultados obtenidos aplicando la metodología propuesta.

2.3.1 FoldX

FoldX es una herramienta computacional, que implementa un campo de fuerza empírico, desarrollado para la evaluación eficiente del efecto de las mutaciones sobre la estabilidad, el plegamiento y la dinámica de las proteínas y los ácidos nucleicos [87]. Se basa principalmente en el cálculo de energía libre a partir de estructuras 3D de macromoléculas. Sin embargo, permite además, estimar las posiciones de los protones y los puentes de hidrógeno.

La energía libre, es calculada utilizando la siguiente expresión matemática de aportes energéticos:

$\Delta G = Wvdw \cdot \Delta Gvdw + WsolvH \cdot \Delta GsolvH + WsolvP \cdot \Delta GsolvP + \Delta Gwb + \Delta Ghbond + \Delta Gel + \Delta GKon + Wmc \cdot T \cdot \Delta Smc + Wsc \cdot T \cdot \Delta Ssc$

Los componentes se definen a continuación.

- $\Delta Gvdw$ es la suma de las contribuciones de van der Waals de todos los átomo con respecto a la interacción con el solvente.
- $\Delta GsolvH$ y $\Delta GsolvP$ son las diferencias en energía de solvatación para grupos apolares y polares respectivamente, cuando estos cambian desde el estado no plegado a plegado.
- $\Delta Ghbond$ es la diferencia de energía libre entre la formación de un enlace de hidrógeno intra-molecular y uno inter-molecular.
- ΔGwb es la energía libre de estabilización adicional proporcionada por una molécula de agua que hace más de un enlace de hidrógeno a la proteína que no se puede tener en cuenta con aproximaciones de solventes no explícitas [79].

- ΔGel es la contribución electrostática de los grupos cargados, incluyendo las hélices dipolo.
- ΔSmc es el costo de la entropía de fijar el back-bone en el estado plegado; este término depende de la tendencia intrínseca de un aminoácido particular a adoptar ciertos ángulos diedros [69, 68].
- ΔSsc es el costo de entropía de fijar una cadena lateral en una conformación particular [1].
- Si la estimación se desarrolla sobre proteínas oligoméricas o complejos de proteína, se adicionan dos términos a la contribución energética: $\Delta Gkon$ que refleja el efecto de las interacciones electrostáticas en la constante de asociación *kon* (esto se aplica solo a las energías de enlace de la subunidad) [92] y ΔStr que es la pérdida de entropía traslacional y rotacional que se deriva de la formación del complejo. Este último término se cancela cuando observamos el efecto de mutaciones puntuales en complejos.
- Los valores de energía de $\Delta Gvdw$, $\Delta GsolvH$, $\Delta GsolvP$ y $\Delta Ghbond$ atribuidos a cada tipo de átomo se han derivado de un conjunto de datos experimentales, y ΔSmc y ΔSmc han sido considerados desde estimaciones teóricas.
- Los términos $Wvdw$, $WsolvH$, $WsolvP$, Wmc y Wsc corresponden a los factores de ponderación aplicados a los términos de energía bruta. Todos son 1, excepto por la contribución de van der Waals que es de 0.33 (las contribuciones de van der Waals se derivan de la transferencia de energía de vapor a agua, mientras que en la proteína vamos de solvente a proteína).

Como entrada principal, recibe una estructura PDB[1] y dentro de los resultados más relevantes, se encuentran los cálculos de energía libre.

Ha sido usada en diferentes investigaciones, incluyendo el análisis de mutaciones puntuales aplicados a ingeniería de proteínas [19, 5], evaluación de mutaciones en genomas [86], análisis de termoestabilidad [18, 49], interacciones proteínas-DNA [70], entre las principales, razón por la cual, es considerada como una herramienta común a la hora de la evaluación de mutaciones y un referente para comparar resultados de nuevas herramientas o métodos computacionales.

[1]Protein Data Bank. Formato para la exposición de macromoléculas u estructuras en torno a coordenadas espaciales, obtenidas desde una cristalografía de rayos X, resnonancia magnética nuclear, o a través de modelos computacionales.

2.3.2 I-Mutant

I-Mutant, es una familia de software basados en algoritmos de aprendizaje supervisado para la predicción automática de estabilidad de proteínas ante cambios de residuos o sustituciones expresadas en mutaciones puntuales [20], las cuales se reflejan en los cambios de energía libre. Emplea como algoritmo para entrenamiento de modelos, Support Vector Machine (SVM), permitiendo la evaluación de las mutaciones desde la secuencia lineal de proteína o a su vez desde la estructura 3D en formato PDB.

El método fue entrenado y testeado desde la base de datos ProTherm [9], la cual representa el mayor repositorio de experimentos termodinámicos con respuestas en energía libre basados en cambios de estabilidad de la proteína para mutaciones, en diferentes condiciones.

Actualmente I-Mutant permite la clasificación de la estabilidad de la mutación y a su vez facilita la predicción de los cambios de energía libre $\Delta\Delta G$. Las medidas de desempeño se diferencian dependiendo del uso de I-Mutant y del tipo de set de datos. Si se trabaja con datos de secuencias lineales presenta una accuracy de un 77% y un coeficiente de relación de un 0.62 con un error asociado de 1.45 kcal/mol. Para el caso en que los datos procedan de información estructural, los desempeños mejoran de manera no significativa, obteniendo una accuracy de un 80% y un coeficiente de relación de un 0.71 con un error asociado de 1.30 kcal/mol.

Si bien, es uno de los métodos más utilizados, el hecho de utilizar Support Vector Machine como algoritmo de aprendizaje supervisado para el entrenamiento de modelos, es un punto limitante a la hora de utilizar set de datos altamente no lineales, dado a que el algoritmo sólo traza hiperplanos y transforma los elementos aplicando funciones de kernel, con el fin de maximizar la varianza. Esto podría provocar sobre ajuste o generar bajos desempeños.

2.3.3 CUPSAT

CUPSAT (Cologne University Protein Stability Analysis Tool) es una herramienta web para analizar y predecir la estabilidad de la proteína frente a cambios o sustituciones puntuales de residuos. La herramienta utiliza información estructural específica de los átomos participantes en la mutación, tales como: ángulos de torsión y potenciales de energía, con el fin de predecir los cambios en diferencia de energía que representa la sustitución, expresados en forma de $\Delta\Delta G$ [75].

Como requisitos para su uso, es necesario la estructura en formato PDB y la posición del residuo a ser mutado. Como resultado, entrega información sobre el sitio de la mutación, principalmente accesibilidad al solvente, estructura secundaria y ángulos de torsión. Además, entrega información detallada sobre las 19 posibles mutaciones para el residuo objetivo.

La herramienta fue testeada utilizado 1538 mutaciones desde denaturaciones térmicas y 1603 denaturaciones aplicando técnicas químicas. Presentado un desempeño mayor al 80% de accuracy.

Esta herramienta no aplica algoritmos de aprendizaje supervisado y se basa principalmente en el uso de técnicas asociadas a bioinformática estructural, para poder evaluar los cambios producidos por las sustituciones, además, permite analizar un espectro amplio de elementos dado a que, facilita el análisis de las 19 posibilidades de residuos, permitiendo hacer una evaluación de cuáles podrían ser sustituciones favorables y cuales no. El desempeño se obtiene al comparar los valores estimados por la herramienta contra resultados reportados en base de datos ProTherm [9].

2.3.4 MultiMutate

2.3.5 Herramientas necesarias para la caracterización de los set de datos

Adicional a las herramientas expuestas, se hace una descripción breve de SDM [74] y MOSST [73], las cuales serán utilizadas a lo largo de la metodología con el fin de poder caracterizar las mutaciones desde los puntos de vista termodinámico (aplicando SDM) y filogenético (por medio de MOSST).

SDM

MOSST

2.4 Hipótesis

En base a las herramientas existentes y en vista del aumento considerable de datos asociados a mutaciones en proteínas y el conocimiento de las respuestas que éstas generan, se evidencia la necesidad del desarrollo de herramientas computacionales o nuevos modelos de clasificación o regresión que faciliten el entrenamiento de proteínas singulares y la evaluación de sus mutaciones puntuales, con el fin de poder evaluar nuevos ejemplos y cuáles serían los efectos de estos, sin tener que recurrir en grandes costos económicos y tiempos de espera.

Dado esto se propone la siguiente hipótesis.

Es posible utilizar técnicas de Meta Learning y algoritmos de aprendizaje supervisado para la generación de modelos de clasificación o regresión de mutaciones puntuales descritas

a partir de sus propiedades termodinámicas y filogenéticas?

Además de la hipótesis central surgen interrogantes como.

- Es posible utilizar estos nuevos modelos como herramientas para diagnóstico médico?
- Cómo se evalúan la robustez y la generalización de estos modelos, serán capaces de adaptarse a nuevos ejemplos?
- Es factible el desarrollo de una herramienta computacional que permita entrenar diferentes set de datos y que facilite la clasificación de nuevos ejemplos?

2.5 Objetivos

En base a la hipótesis planteada y a las preguntas adicionales expuestas, se exponen a continuación el objetivo general y los objetivos específicos.

2.5.1 Objetivo general

Diseñar e implementar estrategias inspiradas en Meta Learning para la implementación de modelos de clasificación y regresión asociado a mutaciones puntuales en proteínas de interés basados en descriptores termodinámicos, estructurales y filogenéticos.

2.5.2 Objetivos específicos

Dentro de los objetivos específicos se encuentran los siguientes.

1. Preparar y describir, por medio de propiedades termodinámicas, estructurales y filogenéticas, set de datos de mutaciones puntuales de proteínas con respuesta conocida expuestos en bibliografía o bases de datos reconocidas.
2. Implementar y evaluar metodología de meta learning para el diseño de meta modelos de clasificación y regresión de mutaciones puntuales aplicados a set de datos de proteínas generadas.
3. Diseñar e implementar herramienta computacional que permita el entrenamiento de set de datos y el uso de meta modelos para la evaluación de nuevos ejemplos.
4. Testear y evaluar comportamiento de la herramienta y los meta modelos en base a sistemas de datos que involucren mutaciones en proteínas con respuesta conocida.

5. Implementar modelos de clasificación para la relevancia clínica de mutaciones puntuales en proteína pVHL, asociada a la enfermedad von Hippel Lindau.

2.6 Metodología propuesta

Con el fin de poder responder a la hipótesis planteada y dar solución a los objetivos impuestos, se propone una metodología general, en la cual se consideran diferentes estrategias, implementaciones y evaluación de modelos. A continuación se explica la metodología propuesta y los componentes principales de ésta.

2.6.1 Preparación de set de datos

La preparación del set de datos consiste en obtener data para poder entrenar los modelos predictivos, la data se asocia a información de mutaciones en proteínas y la respuesta que ésta genera. En la Figura 2.3 se expone un esquema general con los pasos desarrollados para la preparación del set de datos.

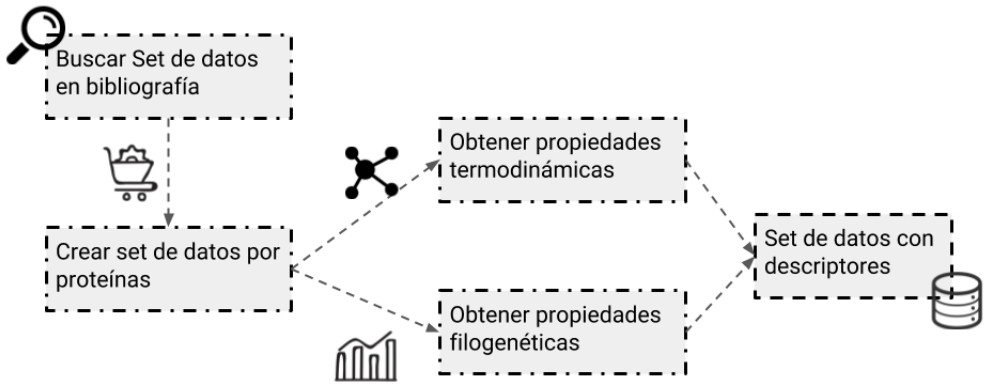


Fig. 2.3 Esquema representativo asociado al proceso de generación de set de datos de mutaciones puntuales en proteínas.

Tal como se expone en la Figura 2.3, los set de datos se buscan en la bibliografía, a partir de modelos desarrollados previamente, bases de datos en la literatura, etc. El objetivo fundamental, es encontrar proteínas con mutaciones puntuales cuyo efecto sea conocido,

dicha respuesta puede ser categórica, es decir, asociada al diseño e implementación de modelos de clasificación o continua y se aplica para modelos de regresión.

En una segunda instancia, a partir de la data recolectada ésta se procesa con el fin de poder obtener set de datos de proteínas individuales con una cantidad de ejemplos considerables que permitan el diseño de modelos válidos, para ello, fueron implementados scripts bajo el lenguaje de programación Python con el fin de recuperar las proteínas, obtener la información y generar la data de manera individual, además, eliminar ejemplos ambiguos. Es decir, filas con los mismos valores pero cuya columna de respuesta fuese diferente.

A partir de esto se forman n set de datos asociados a n proteínas, cada uno con m ejemplos y cuyos descriptores consisten en el residuo original, posición en proteína, residuo mutado y la respuesta asociada. El desbalance de clases se analiza con respecto a las posibles categorías existentes en la respuesta y el porcentaje de representatividad que éstas poseen en la muestra. Se considera que el set de datos exhibe este comportamiento cuando presentan las características expuestas en la sección **CITAR PARTE ANTERIOR**. En este caso, los ejemplos se tratan con SMOTE (Synthetic Minority Oversampling Technique) [24]

Posteriormente se aplican las herramientas SDM [74] y MOSST [73] con el fin de obtener los descriptores asociados a las propiedades termodinámicas y filogenéticas. Para ello, scripts Python son desarrollados para consumir los servicios de dichas herramientas y registrar los resultados obtenidos, formando así, set de datos con los descriptores planteados en los objetivos iniciales. Un punto importante a destacar, es que el uso de SDM implica que las proteínas a trabajar, deben presentar una estructura 3D reportada en el Protein Data Bank [10] o al menos poseer un modelo representativo y validado. Esto es debido a que se utilizan informaciones de coordenadas para la estimación del efecto de la mutación, minimizaciones energéticas y estabilización de la mutante.

Ya con los descriptores formados, las características asociadas a variables categóricas son codificadas. Si la totalidad de posibles categorías supera el 20% del total de características en el set de datos, se aplica Ordinal Encoder, en caso contrario, One Hot Encoder [76]. Ordinal Encoder consiste en la transformación de variables categóricas en arreglos de números enteros con valores desde $0, ..., n-1$ para n posibles categorías. Por otro lado, One Hot Encoder, consiste en agregar tantas columnas como posibles categorías existan en el set de datos completadas mediante binarización de elementos (0 si la característica no se presencia, 1 en caso contrario).

Es importante mencionar que las respuestas asociadas a las mutaciones pueden ser del tipo continuo o categórico, lo cual implica que tanto los modelos como las métricas varían. No obstante, se aplica la metodología indistintamente, con el fin de demostrar la robustez del método y la eficacia de éste sin importar el tipo de modelo que se éste entrenando.

2.6.2 Implementación de meta modelos de clasificación/regresión

La implementación de meta modelos consiste en la obtención de un grupo de estimadores que en conjunto, permiten clasificar o predecir nuevos ejemplos. Para ello, se diseña e implementa una metodología inspirada en Sistemas de Meta Learning y aplicando técnicas estadísticas para la evaluación del desempeño y el uso del meta modelo con nuevos ejemplos.

En la Figura 2.4, se exponen las etapas asociadas a la implementación de meta modelos, contemplando desde la fase de entrenamiento de los modelos hasta la unión en meta clasificadores (Paper en redacción). Cada una de las etapas contempla un conjunto de scripts implementados en lenguaje de programación Python y empleando la librería Scikit-Learn para el entrenamiento y evaluación de los clasificadores o predictores [76], así como Numpy para el uso de módulos estadísticos [94].

Fig. 2.4 Esquema representativo asociado al proceso de creación de meta modelos utilizando Meta Learning System Tools.

Tal como se observa en la Figura 2.4, es posible identificar etapas claves en el proceso: Exploración de modelos, Selección y Generación de los meta clasificadores/predictores, junto con su evaluación. Cada una de estas etapas se exponen a continuación.

Exploración de modelos

La exploración de modelos o estimadores, se basa en la aplicación de diferentes algoritmos de aprendizaje supervisado con variaciones en sus parámetros de configuración inicial. La utilización de los algoritmos, depende principalmente del tipo de respuesta que presente el

set de datos, es decir, si es continua o categórica. No obstante, a modo resumen, en la Tabla 2.3 se exponen los algoritmos utilizados, el caso en el que se usan y los parámetros que se varían junto con el total de iteraciones posibles para cada elemento:

Algoritmos y parámetros empleados en la etapa de Exploración en MLSTools					
#	**Algoritmo**	**Tipo**	**Parámetros**	**Uso**	**Iteraciones**
1.	Adaboost	Ensamble	Algoritmo Número estimadores	Clasificación y Regresión	16
2.	Bagging	Ensamble	Bootstrap Número estimadores	Clasificación y Regresión	16
3.	Bernoulli Naive Bayes	Probabilístico	Default	Clasificación	1
4.	Decision Tree	Características	Criterio división Función de impureza	Clasificación y Regresión	4
5.	Gaussian Naive Bayes	Ensamble	Default	Clasificación y Regresión	1
6.	Gradient Tree Boosting	Ensamble	Función de pérdida Número estimadores	Clasificación y Regresión	16
7.	k-Nearest Neighbors	Distancias	Número Vecinos Algoritmo Métrica distanciaPesos	Clasificación y Regresión	160
9.	Nu Support Vector Machine	Kernel	Kernel Nu Grado polinomio	Clasificación y Regresión	240
10.	Random Forest	Ensamble	Número estimadores Función de impureza Bootstrap	Clasificación y Regresión	32
11.	Support Vector Machine	Kernel	Kernel C Grado polinómio	Clasificación y Regresión	240
				Total Iteraciones	**726**

Table 2.3

Como se observa en la Tabla 2.3, son sobre 720 modelos los que se generan y a partir de ellos se obtiene distribuciones de medidas de desempeño que permiten evaluarlos. En el caso de modelos de regresión se utilizan los coeficientes de Pearson, Spearman, Kendall τ y R^2,

mientras que para modelos de clasificación, se consideran la Precisión, Exactitud, Recall y F1.

Finalmente, esta etapa entrega set de modelos entrenados y evaluados según las métricas de interés, se destaca que cada modelo es validado a través del proceso de validación cruzada, con el fin de poder disminuir posibles sobreajustes. El valor de k asociado a las subdivisiones a realizar varía con respecto a la cantidad de ejemplos que presente el set de datos, es decir, sea n la cantidad de ejemplos en la muestra, si $n <= 20$ se tiene que $k = n$ implicando el uso de Leave one out, si $n > 20$ y $n <= 50$ se considera un valor de $k = 3$, si $n > 50$ y $n <= 100$ $k = 5$, por último, si $n > 100$ se tiene un valor de $k = 10$.

Selección de modelos

Cada distribución de medida de desempeño perteneciente a los modelos entrenados en la fase de Exploración, se somete a test estadísticos basados en Z-score [76] que permite seleccionar los modelos cuyas métricas representen outliers positivos dentro de la distribución.

El algoritmo general, utilizado para el desarrollo de esta selección es como se expone en el algoritmo 1, para el cual se detallan los pasos simplificados que permiten obtener un conjunto de modelos entrenados y que representan los valores más altos dentro de su distribución. Es importante mencionar, que se obtiene un conjunto M' con los modelos, considerando como punto de selecciones los valores evaluados con respecto a la desviación estándar, considerando los umbrales 3 σ, 2 σ y 1.5 σ por sobre la media, si ningún factor se cumple, sólo se considera el valor máximo en la distribución.

Es importante mencionar, que cada distribución puede permitir la selección de distintos modelos, lo cual implica que un mismo modelo pueda ser seleccionado en diferentes medidas, razón por la cual, a la hora de obtener el conjunto de modelos M' se remueven aquellos elementos que se encuentran repetidos. Siendo estos, sólo los modelos que presenten igualdad tanto en el algoritmo como en sus parámetros de configuración inicial.

Algoritmo 1 Algoritmo de selección de modelos

Entrada: Conjunto M con modelos entrenados y sus medidas de desempeño, Lista L con medidas de desempeño.

Salida: Conjunto M' con modelos seleccionados.

para i en L **hacer**
  Calcular media μ, desviación estándar σ en distribución M_i
  para $x \in M_i$ **hacer**
    si $x \geq \mu + 3 * \sigma$ **entonces**
      Agregar x a M'
    fin si
  fin para
  si largo $M' = 0$ **entonces**
    para $x \in M_i$ **hacer**
      si $x \geq \mu + 2 * \sigma$ **entonces**
        Agregar x a M'
      fin si
    fin para
    si largo $M' = 0$ **entonces**
      para $x \in M_i$ **hacer**
        si $x \geq \mu + 1.5 * \sigma$ **entonces**
          Agregar x a M'
        fin si
      fin para
      si largo $M' = 0$ **entonces**
        para $x \in M_i$ **hacer**
          si $x = MAXM_i$ **entonces**
            Agregar x a M'
          fin si
        fin para
      fin si
    fin si
  fin si
fin para
devolver D sin valores extremos

Generación y evaluación de meta modelos

A partir del conjunto de modelos M', el cual representa los estimadores seleccionados cuyas medidas de desempeño son las más altas en sus distribuciones correspondientes, se generan meta modelos, es decir, estimadores compuestos de diversas unidades, los cuales en conjunto entregan una respuesta, ya sea por ponderación o votación. El proceso general para la generación de los meta modelos, es descrito a continuación.

En una primera instancia, los modelos son nuevamente entrenados y se comparan las nuevas medidas de desempeño con las obtenidas previamente, en caso de que exista una diferencia mayor al 20%, en cualquiera de sus métricas, el modelo se remueve del conjunto M'. La razón fundamental de esto, es debido a que se espera desarrollar modelos robustos cuyas evaluaciones no presenten variaciones significativas y que realmente no alteren sus predicciones ante nuevos ejemplos, razón por la cual, se aplica nuevamente validación cruzada para validar los modelos.

Con el fin de evaluar el desempeño de los meta modelos, nuevas medidas se generan a partir de la información resultante de los modelos individuales. No obstante, la forma en la que se obtienen varían dependiendo del tipo de respuesta que se debe entregar.

Si la respuesta es continua, se obtiene los valores de predicción de cada modelo y se promedian, para luego aplicar las métricas estándar (Coeficiente de Pearson, Kendall τ, Spearman y R^2) sobre estos valores promediados y los reales. Expresado matemáticamente:

Sea M' la cantidad de elementos en el meta modelo, n la cantidad de ejemplos en el set de datos y sea Y el vector de respuestas reales de tamaño n. Para cada $M'_i \in M'$ se obtiene un vector Y_i que representa los valores de predicción entregados por el modelo M'_i. A partir de cada Y_i se genera una matriz de predicciones $P(mxn)$ donde m representa la cantidad de modelos en M'. Finalmente, se obtiene un vector Y' de tamaño n, el cual se compone de la media de cada columna en la matriz P, es decir, para el ejemplo i se obtienen m predicciones, las cuales son promediadas, formando el valor $Y'_i \in Y'$. Vector el cual, se utiliza para obtener las métricas de desempeño.

Para el caso en que la respuesta sea categórica, es decir, los modelos son del tipo clasificación, se obtiene la respuesta de cada modelo individual y se selecciona una única categoría, correspondiente a aquella que presente una mayor probabilidad de ocurrencia dada la distribución de elementos y considerando para ello las probabilidades iniciales de cada categoría en el set de datos de estudio. De esta forma, se obtiene un vector respuesta con la clasificación de cada ejemplo cuyo valor corresponde al evento más probable a ocurrir, este vector se compara con el set de respuestas reales y se aplican las métricas de interés para clasificadores.

2.6.3 Cómo usar los meta modelos para la clasificación de nuevos ejemplos?

Nuevos ejemplos pueden ser clasificados o predecir su respuesta, dependiendo sea el caso, a partir de los meta modelos desarrollados. En el caso de estimadores basados en variables continuas, los nuevos ejemplos se someten a cada uno de los modelos individuales pertenecientes al sistema, los cuales generan una respuesta individual, a partir de dichas respuestas, se genera un intervalo de confianza con un nivel de significancia $\alpha = 0.01$ donde existe una mayor probabilidad de que se encuentre el valor real de la predicción dado los valores del entrenamiento. Para ejemplos que impliquen clasificación, se obtiene la respuesta de cada modelo individual y se evalúa la probabilidad de ocurrencia de cada categoría, entregando así, la respuesta condicionada por una probabilidad de ocurrencia del evento.

2.6.4 Uso de meta modelos en sistemas de proteínas

El objetivo principal de esta metodología, radica en el hecho de crear una herramienta que permita implementar modelos basados en algoritmos de aprendizaje supervisado para set de datos de mutaciones puntuales o variantes para una misma proteína.

Un flujo general del uso de la herramienta, se expone en la Figura 2.5.

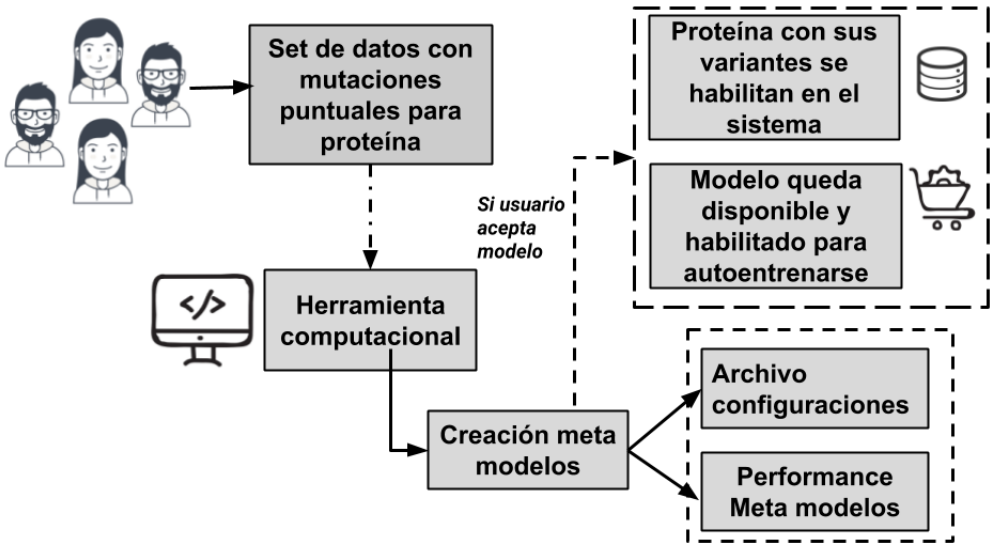


Fig. 2.5 Esquema representativo de flujo asociado a la herramienta de generación de meta modelos para mutaciones puntuales en proteínas de interés.

La idea general, consiste en que usuarios de la herramienta, puedan entrenar sus propios modelos de clasificación o regresión, basados en la metodología expuesta en los pasos

anteriores mediante el uso de Meta Learning Sytem Tools (Paper en Redacción). Para ello, los usuarios deben entregar sus set de datos con la información necesaria para ser procesada: cadena, residuo original, posición, residuo mutado y respuesta o efecto de la mutación, además del archivo PDB a ser procesado. La herramienta, aplica los pasos expuestos en la metodología de este capítulo generando un meta modelo basado en algoritmos de aprendizaje supervisado y las medidas de desempeño que permiten evaluar el modelo obtenido. Si el usuario acepta la metodología y por medio de un consentimiento informado, permite la publicación de los datos, el sistema habilita el acceso tanto a los meta modelos como a los set de datos y los agrega a la lista de procesos de modelos auto entrenables. Esto último, implica que ante la adición de nuevos ejemplos al set de datos, el sistema actualiza los modelos y las medidas de desempeño, aplicando la metodología expuesta, así, constantemente mantiene la actualización de la información y permite mantener en constante crecimiento los datos que contemplan el desarrollo de los modelos.

2.7 Análisis y evaluación de los set de datos a utilizar

A continuación, se exponen los resultados obtenidos hasta el momento y discusiones a cerca del proceso generado, contemplando desde la etapa inicial asociada a la búsqueda de set de datos de mutaciones en proteínas, la preparación de estos para la aplicación de la metodología desarrollada y la evaluación del funcionamiento de estos con el fin de generar meta modelos asociados a sistemas de mutaciones en proteínas y permitir el desarrollo de herramientas computacionales que faciliten dichas acciones. Adicional a esto, se expone un caso de estudio donde se emplea esta metodología para el desarrollo de modelos de clasificación de relevancia clínica de mutaciones asociadas a la enfermedad de von Hippel-Lindau, exponiendo las necesidades de adición de información en set de datos altamente no lineales.

2.7.1 Set de datos utilizados

En el presente apartado se describen las características básicas de los set de datos trabajados, así como también, qué representan las proteínas bajo las cuales se están desarrollando los modelos de estimadores.

Descripción general

Los set de datos utilizados, tanto para la formación de los inputs asociados al sistema, así como también la validación de respuesta correspondiente a la mutación que estos tienen,

fueron extraídos desde distintas bases de datos de mutaciones en proteínas de estudios relacionados a los cambios que provoca la sustitución del residuo inicial, ya sea a nivel de cambios energéticos o estabilidad de la proteína.

11 set de datos con respuesta continua fueron obtenidos. Cada set de datos contemplaba como elemento a predecir, las diferencias de energía libre de Gibbs, entre los residuos originales y mutados. Las mutaciones fueron seleccionadas desdes diversos estudios en los cuales se reportaron, centrándose en [93, 88, 78, 4, Bordner and Abagyan].

Adicional a los set de datos con respuesta continua, 8 conjuntos de elementos asociados a tareas de clasificación fueron obtenidos desde diversos estudios reportados a la actualidad [7, 17, 23, 82, 21, 80, 54, 65, 45].

De tal manera, se generó un total de 19 conjuntos de set de datos, con respuesta categórica y continua, los cuales se asocian a proteínas independientes, usadas para la evaluación de las metodologías planteadas. Estas 19 proteínas junto con su descripción, se exponen en la Tabla 2.4.

Resumen set de datos de proteínas y sus características				
#	**Código PDB**	**Tipo**	**Ejemplos**	**Descripción**
1.	1A22	Regresión	132	Human growth hormone bound to single receptor
2.	1CH0	Regresión	191	Crystal and molecular structures of the complex of alpha-*Chymotrypsin with its inhibitor Turkey Ovomucoid third domain
3.	1DKT	Regresión	119	CKSHS1: Human cyclin dependent kinase subunit, type 1 complex with metavanadate
4.	1FKJ	Regresión	219	Atomic structure of FKBP12-FK506, an immunophilin immunosupressant complex
5.	1FTG	Regresión	203	Structure of apoflavodoxin: closure of a Tyr/Trp aromatic gate leads to a compact fold
6.	1PPF	Regresión	190	X-Ray crystal structure of the complex of human leukocyte elastase and the third domain of the Turkey ovomucoid inhibitor

7.	1RX4	Regresión	556	Dihydrofolate reductase (E.C.1.5.1.3) complexed with 5,10-Dideazatetrahydrofolate and 2'-Monophosphadenosine 5'-Diphosphoribose
8.	1WQ5	Regresión	239	Crystal structure of tryptophan synthase alpha-subunit from Escherichia coli
9.	2AFG	Regresión	134	Human acidic fibroblast growth factor
10.	3SGB	Regresión	191	Structure of the complex of Streptomyces Griseus protease B and the Third domain of the Turkey ovomucoid inhibitor
11.	5AZU	Regresión	203	Crystal structure analysis of oxidize Pseudomonas Aeruginoa Azurin at PH 5.5 and PH 9.0. A PH-induced conformational Transition involves a peptide bond flip
12.	1BN1	Clasificación	1802	Carbonic anhydrase II inhibitor
13.	1BVC	Clasificación	561	Structure of a Biliverdin Apomyoglobin complex
14.	1LZ1	Clasificación	848	Human Lysozyme. Analysis of Non-Bonded and Hydrogen-Bond interactions
15.	1STN	Clasificación	2193	The crystal structure of Staphylococcal Nuclease
16.	1VQB	Clasificación	820	Gene V Protein (Single-Stranded DNA Binding Protein)
17.	2CI2	Clasificación	741	Crystal and molecular structure of the Serine proteinase inhibitor CI-2 from Barley seeds
18.	2LZM	Clasificación	2336	Structure of Baceriophage T4 Lysosyme
19.	2RN2	Clasificación	712	Structural details of ribonuclease H from Escherichia Coli

Table 2.4 Resumen de proteínas utilizadas para el desarrollo de meta modelos basados en metodología Meta Learning System propuesta durante este capítulo.

Cada una de las proteínas presentan diferentes características y funcionalidades, algunas facilitan la unión a DNA, mientras que otras presentan propiedades enzimáticas, por otro lado, existen enzimas que representan inhibidores, entre las principales. Esto es interesante a la hora de evaluar el poder que presenta la metodología con respecto al análisis de diferentes

proteínas, estructuras y complejos, ya que se presenta una gran variedad en cuanto a forma y funcionalidad de éstas, lo que implica que el sistema no se limita por cierto tipo de estructuras o complejos.

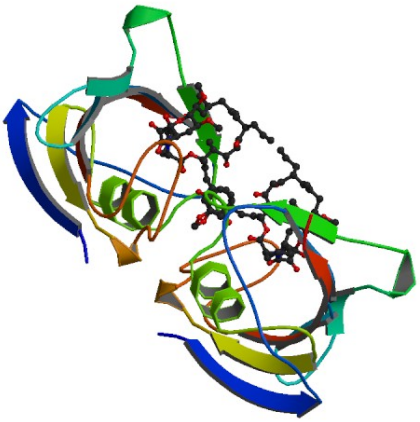
A modo de ilustrar las diferencias estructurales de las proteínas en estudio, en la Figura 2.6 se exponen algunas de las estructuras asociadas a las proteínas utilizadas para desarrollar modelos de clasificación o regresión.

Las mutaciones fueron recolectadas desde diferentes set de datos, por lo que, en caso de información ambigua, es decir, una misma mutación con diferentes respuestas, no fueron consideradas. Por otro lado, debido a que para la aplicación de la herramienta SDM se necesitaba la cadena a la cual pertenece en residuo, scripts desarrollados en Python y utilizando la librería BioPython, permitieron procesar los archivos asociados a las estructuras de las proteínas, identificadas desde el Protein Data Bank (PDB) [3]. Descartando aquellas mutaciones reportadas en las que no se encontró la cadena, obteniendo como resultante la cantidad de mutaciones reportadas para cada proteína expuestas en la Tabla 2.4.

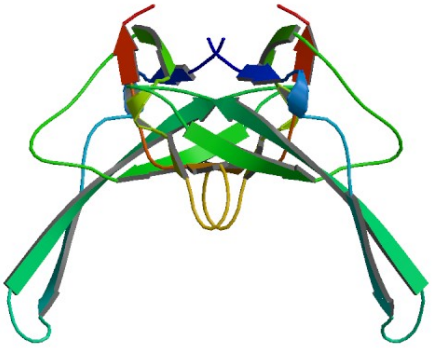
Evaluación del desbalance de clases y distribución de respuestas continuas

El desbalance de clases se evaluó en aquellos set de datos con respuesta categórica, lo cual contempla, tres posibles casos: Neutral, Estable, No Estable, esto es debido a que todos los estudios donde se evalúan mutaciones, normalmente se analizan cambios que alteren la estabilidad de la proteína. En la Figura 2.7, se aprecia a modo ejemplo dos set datos y su distribución de categorías para la variable respuesta.

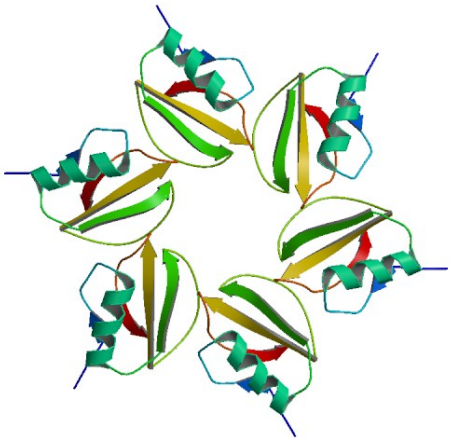
En las 8 proteínas en estudio para modelos de clasificación, la distribución de las categorías es similar a lo expuesta en la Figura 2.7 para todas ellas, donde cerca del 50% corresponden a mutaciones que afectan positivamente a la estabilidad, mientras que mutaciones que provocan cambios negativos o no generan diferencias, se encuentran en proporciones similares, ambas cercanas al 25%. Si bien las proporciones son dispares, para este caso, se considera un desbalance como un elemento que representa menos de un 5% del total de la muestra, además, dado a que la cantidad de ejemplos son elevadas, un 20% o un 25% implica cerca de 200 mutaciones, en promedio, que cumplen dicha característica. Además, el hecho de que exista una cantidad inferior de mutaciones no benéficas a la estabilidad viene dada a la dificultad de encontrar y reportar mutaciones que afecten negativamente a la para una proteína dado a la propensión filogenética [73] que estos ocurran, lo cual se ve reflejado en las diferencias asociadas a cambios positivos dentro del set de mutaciones. No obstante, si bien el hecho de que la propensión filogenética indique que el cambio tiende a mejorar estabilidad, diseñar mutantes con mejoras en propiedades de interés, es un problema latente



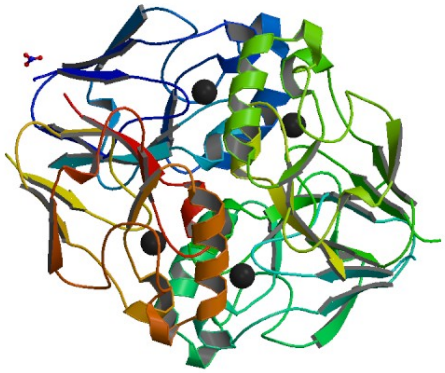
(a) 1FKJ



(b) 1VQB

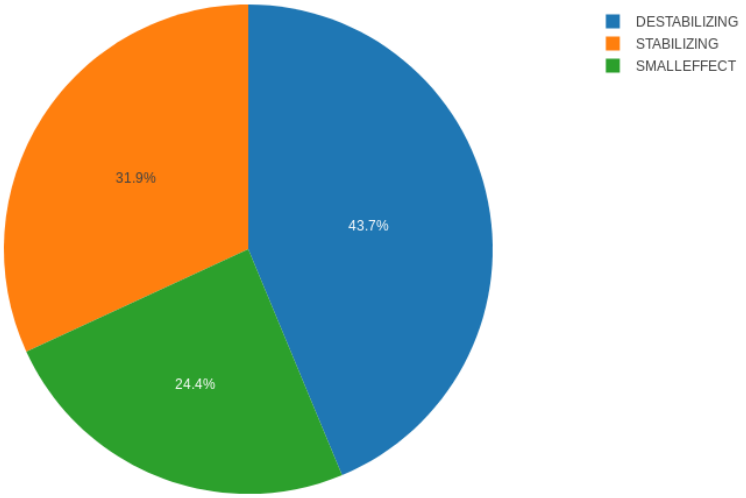


(c) 2CI2

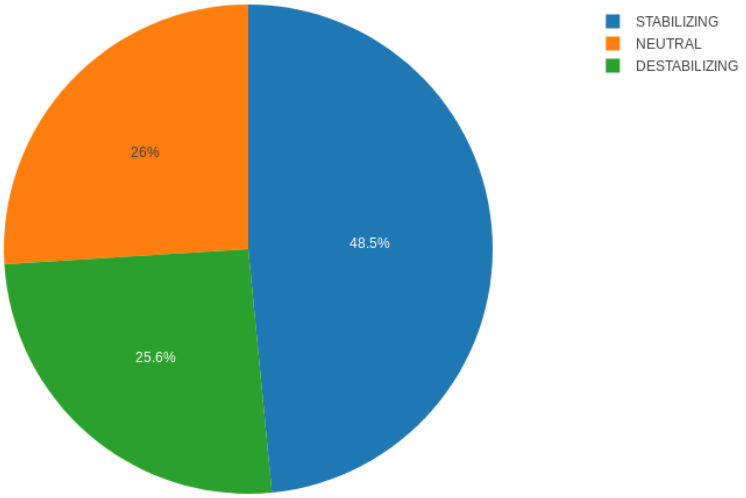


(d) 5AZU

Fig. 2.6 Representación de estructuras de proteínas ejemplos utilizadas para el desarrollo de meta modelos de clasificación.



(a) 1STN



(b) 2RN2

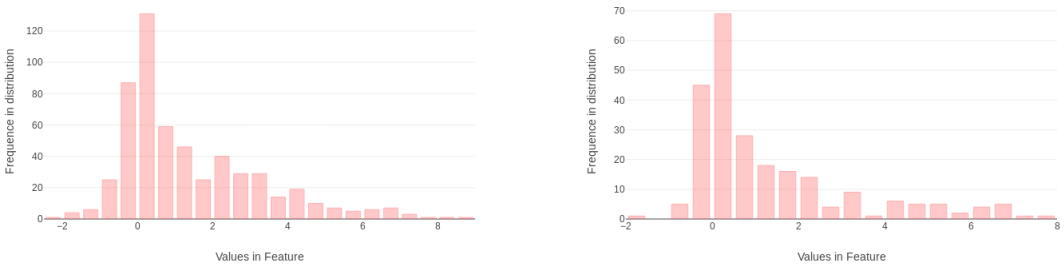
Fig. 2.7 Evaluación del desbalance de clases en proteínas ejemplo.

en la actualidad, de alto costo económico y computacional y con una gran demanda desde diferentes áreas del conocimiento.|

En los set de datos para el desarrollo de modelos de regresión, se evaluó la distribución de la respuesta, en este caso, valores de $\Delta\Delta\, G$ asociado a diferencias de energía libre producidas entre el residuo mutado y el original, tal que: $\Delta\, Res_{mut} - \Delta\, Res_{wild} = \Delta\Delta\, G$.

Las distribuciones se evaluaron utilizando el test de Shapiro, con el fin de determinar si la distribución se comportaba como una normal. Para todas las proteínas estudiadas, en los 11 set de datos, las respuestas presentaron distribución normal, con valores de Shapiro sobre 0.8 y un p-value ≤ 0.01, lo cual indica una alta confianza estadística en los resultados presentados por dicho test.

Una visualización de las distribuciones puede generarse a partir del desarrollo de histogramas, los cuales, a modo de ejemplo se expone en la Figura 2.8.



(a) Histograma para respuesta continua en 1RX4 (b) Histograma para respuesta continua en 1WQ5

Fig. 2.8 Evaluación de la distribución de respuesta continua en set de datos de proteínas.

El análisis de estas características es relevante a la hora de diseñar modelos de clasificación o regresión, debido a que si existe una tendencia por una clase condiciona al clasificador a *"aprender en base a la mayoría"*, por lo que puede aumentar los errores en cuanto a falsos positivos, dado a que, no se tiene la cantidad de ejemplos suficientes para una clase que permitan al modelo capturar las posibles variaciones asociadas a ésta.

Dado a los análisis de evaluación de representatividad de categorías en el set de datos y distribución de respuestas continuas, se expone que los set de datos seleccionados no presentan desbalance significativo para el caso de desarrollo de modelos de clasificación y a su vez, todas las respuestas asociadas a cambios en la energía libre para modelos de regresión, presentan distribución normal. Razón por la cual, es factible el desarrollo de modelos asociados a las respuestas presenten en los set de datos seleccionados. No obstante, sólo se ha considerado el problema del desbalance y la evaluación de distribución en la respuesta continua, una vez caracterizado los set de datos a partir de las propiedades fisicoquímicas y

termodinámicas, se analizarán las características y cómo éstas condicionan la clasificación o la predicción de cambios energéticos.

Chapter 3

Digitalización de secuencias lineales de proteínas aplicadas al reconocimiento de patrones y modelos predictivos

Desarrollar modelos predictivos basados en algoritmos de aprendizaje supervisado, o, la identificación de patrones aplicando técnicas de clustering, son tareas muy relevantes a la hora de trabajar con secuencias de proteínas, ya sea para identificar grupos con características comunes o entrenar modelos predictivos de respuestas de interés. En ambos casos, se requiere el uso de conjuntos de datos altamente informativos y con características numéricas para poder utilizar los métodos implementados en las librerías actuales [76].

Diferentes metodologías se han implementado, para manipular las variables categóricas en set de datos y lograr su codificación numérica. Enfoques basados en adición de columnas según las categorías o simple transformación empleando representaciones en conjuntos naturales suelen ser utilizados. No obstante, generan bastante discusión sobre las nuevas representaciones y a su vez, el hecho de aumentar el número de columnas, conlleva a incrementar las dimensiones del conjunto de datos, provocando efectos en los desempeños de los algoritmos.

Particularmente, en secuencias de proteínas, se han utilizado las frecuencias de los residuos para codificarlos, la cual, pese a su simplicidad, ha resultado ser efectiva en diferentes casos de uso. No obstante, este tipo de codificación, no permite explorar el ambiente bajo el cual se encuentran los residuos y tampoco considera el efecto de propiedades fisicoquímicas ni termodinámicas.

En diferentes estudios, los residuos se describen a partir de sus propiedades fisicoquímicas y adicional a ello, se emplea información que permite describir el ambiente del residuo a caracterizar, empleando binarizaciones que describen los residuos cercanos, ya sea por medio

del uso de un rango espacial, utilizando modelos o estructuras tridimensionales en donde se representan las coordenadas espaciales de los residuos, o empleando un rango lineal en secuencias lineales.

Otros estudios, se han basado en adicionar información filogenética a las descripciones de los residuos. Sin embargo, estos, sólo se basan en el análisis de mutaciones y permiten evaluar la propensión de que dicho cambio ocurra. No obstante, su uso ha permitido obtener resultados satisfactorios a la hora de entrenar modelos predictivos.

Un enfoque basado en las propiedades fisicoquímicas en combinación con la aplicación de transformaciones de Fourier, ha permitido demostrar que ciertos residuos permiten entregar las características asociadas a la propiedad en estudio, además, facilita comprender el aporte del ambiente sobre estos y representa una forma de estudio novedosa para el uso de información de secuencias lineales. Siendo una metodología ampliamente utilizada para identificar residuos que aporten a la propiedad, por medio de peacks de frecuencias de señales.

A pesar de ser una metodología interesante a la hora de estudiar secuencias lineales, exhiben problemas notorios sobre la selección de las propiedades relevantes a analizar, ya que, existe un número considerablemente alto de propiedades posibles a utilizar y es factible que diferentes familias de proteínas exhiban comportamientos notoriamente no similares y diverjan en cuanto a las propiedades que puedan ser representativas, inclusive, a la hora de estudiar mutaciones en una misma proteína puede que no sólo una propiedad permita su caracterización, si no, que un conjunto pequeño de éstas.

En el presente capítulo, se exponen en detalle, diferentes formas de representar secuencias lineales de proteínas, seguido a su vez del planteamiento del uso de transformadas de Fourier para la digitalización de propiedades fisicoquímicas y cómo es posible utilizar éstas para la identificación de patrones en secuencias lineales o el desarrollo de modelos de clasificación/regresión y la exposición de casos de uso en diferentes proteínas de interés.

3.1 Metodologías asociadas a la codificación de variables categóricas

Diferentes metodologías existen para poder codificar variables categóricas, a su vez, para set de datos de proteínas con secuencias lineales, es factible utilizar sus propiedades fisicoquímicas o frecuencias de residuos. Cada una de estas metodologías se expone a continuación.

3.1.1 One Hot encoder

One Hot encoder, es una de las técnicas más utilizadas a la hora de codificar variables categóricas y se basa principalmente en la adición de columnas con respecto a las categorías existentes en un conjunto de datos.

Dado el vector x de tamaño n con m categorías, por definición, One Hot encoder agrega al conjunto de datos m columnas, tal que, por cada categoría se adiciona una nueva columna al set de datos. Las nuevas columnas se completan con una binarización de los elementos, indicando si el elemento x_i posee la categoría m_j con un valor 1 y en caso contrario 0. Es posible expresar esto como se expone a continuación.

Sea x vector de m categorías de dimensiones $n \times 1$ representado por

$$x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

Su codificación mediante One Hot Encoder corresponde a

$$x'(x) = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Esta metodología, si bien es altamente usada, implica que, a medida que aumentan la cantidad de categorías incrementan el número de columnas a agregar en el set de datos, es decir, si se tienen m categorías se adicionan m columnas. Esto último puede provocar que los set de datos se afecten por problemas relacionados con la *maldición de dimensionalidad*, ya que, a medida que aumentan los descriptores, aumenta la probabilidad de que estos no sean informativos, provocando una adición de información innecesaria y que perjudicaría el rendimiento de los algoritmos de aprendizajes supervisado y no supervisado.

3.1.2 Ordinal encoder

Ordinal encoder es una simplicación de One Hot encoder, ya que simplemente codifica las categorías con números en el conjunto $[0, m-1]$. Es decir, sea el vector x de tamaño n con m categorías y sea M el espacio de las posibles categorías con $M = [m_1, \cdots, m_m]$, y cuya codificación implica el vector $M' = [0, \cdots, m-1]$. $\forall$ elemento que $\in$ a x se obtiene

su codificación a partir del elemento $M'(M(m_i))$ que corresponde a la codificación de la categoría en el espacio M.

Si bien, esta es una técnica ampliamente utilizada, es cuestionable con respecto al orden en que trata las categorías, la representación de la información y el mantenimiento del significado de la data. Razón por la cual, se usa sólo en los casos en que la adición de múltiples descriptores empleando One Hot Encoder sea perjudicial a la hora de implementar modelos de clasificación o regresión, e inclusive, en la búsqueda de patrones.

3.1.3 Frecuencias de residuos

Una secuencia lineal de proteína, corresponde a un vector v de tamaño n donde cada elemento corresponde a un residuo que pertenece a la secuencia. El uso de esta información para alimentar modelos de clasificación o regresión conlleva la codificación de sus elementos. Sin embargo, a la hora de utilizar las codificaciones basadas en One Hot Encoder, el conjunto de datos no queda estándar en cuanto a sus dimensiones, ya que el largo de las secuencias puede variar y a su vez, el número de columnas a agregar corresponde a $nx20$ dado a que son n residuos y el espacio muestral M es de tamaño 20 lo que genera un aumento considerable en la cantidad de dimensiones.

Con el fin de poder representar las secuencias lineales de proteínas, se idearon metodologías que consideran la frecuencia de aparición de los residuos en la secuencia, de tal manera de poder codificarla en un vector de tamaño 20, donde cada elemento representa el número de incidencias del residuo dividido por el largo del vector. Así, cada elemento se encuentra en un rango $[0,1]$ donde 0 indica no incidencia del residuo y 1, incidencia total.

Expresado de forma matemática, sea s una secuencia lineal de proteínas con r residuos, su codificación se basa en la frecuencia de aparición del residuo en s, tal que, sea R el espacio de los posibles residuos r en s, se estima para cada $r_j \in R$ su frecuencia:

$$f(r_j) = \frac{cont(r_j)\ if\ (r_i == r_j)}{n}$$

Finalmente, se tiene que cada residuo r_i se representa en su valor de frecuencia $f(r_i)$, generando un set de datos de tamaños $sx20$ con s secuencias representadas por un vector de tamaño 20.

Esta es una de las representaciones más utilizadas y más simples a la hora de codificar secuencias lineales de proteínas. Sin embargo, presenta diferentes problemas tales como:

- No considera información sobre los residuos asociados a propiedades fisicoquímicas, esto complica el hecho de representar un set de datos de secuencias o mutaciones, ya

que no representa la realidad y sólo expone el comportamiento de las frecuencias de residuos, favoreciendo a aquellos con una mayor incidencia en sus elementos.

- La codificación por frecuencias es utilizada como un primer acercamiento a la representación del problema y principalmente perjudica a los modelos ya que puede generar atributos no informativos, como lo son los residuos sin incidencia, esto conlleva a modelos sobre ajustados y a creación de set de datos no informativos.

- No evalúa elementos relevantes a la caracterización de residuos claves, ambiente bajo el cual ocurren mutaciones o componentes adicionales que facilitarían una mayor comprensión del problema, ya que, sólo conocer las incidencias, proporciona un conocimiento sobre la moda y cuáles son los residuos más relevantes. No obstante, sólo permite inferir características, relacionadas a estos.

El uso de las frecuencias de residuos, es una de las primeras aproximaciones a la codificación de secuencias lineales de proteínas. No obstante, en todos los casos donde han sido utilizadas, se agrega información adicional, que permite comprender diferentes comportamientos y evalúa ciertas propiedades del entorno, razones por las cuales, se recomiendan utilizarlas en conjunto con otros descriptores.

3.1.4 Uso de propiedades fisicoquímicas

El uso de propiedades fisicoquímicas para describir un residuo es ampliamente empleado en la generación de descriptores para conjuntos de datos en ingeniería de proteínas. Diversos enfoques y modelos han sido construidos o entrenados, contemplando información asociada a componentes termodinámicos del residuo, en particular, a la hora de describir residuos para evaluar cambios en la energía libre, relacionados a efectos en la estabilidad de una proteína.

Se han reportado cerca de 570 propiedades fisicoquímicas que pueden ser utilizadas para describir un residuo en una secuencia lineal de proteínas. A su vez, es posible caracterizar estos residuos empleando un conjunto de propiedades estructurales, termodinámicas e inclusive filogenéticas. Es decir, diferentes puntos de vista que permitan describir los residuos pertenecientes a una secuencia. Sin embargo, el hecho de seleccionar qué descriptores son relevantes y cuáles no, radica en un problema de evaluación de características, el cual es común, en el área de la minería de datos.

Dado al gran conjunto de propiedades existentes y a la diversidad de descriptores que pueden ser utilizados para un conjunto de secuencias lineales de proteínas, es necesaria una selección correcta de las características, las cuales permitan formar set de datos informativos y con una correlación mínima entre sus elementos.

Contemplando esta problemática, técnicas de reducción de dimensionalidad o análisis de características son las más utilizadas a la hora de seleccionar los descriptores más informativos para un conjunto de datos. No obstante, en ocasiones, el conocimiento sobre el problema es un factor relevante a considerar.

Dada la relevancia de la selección de descriptores correctos para poder caracterizar y codificar secuencias lineales a partir de propiedades, se describen a continuación algunas técnicas de reducción de dimensionalidad y análisis de características que pueden ser empleadas para dar solución a esta problemática.

Técnicas de reducción de dimensionalidad

Diferentes técnicas para el análisis de características y reducción de dimensionalidad han sido implementadas en el campo de minería de datos, con el fin de poder permitir la selección de descriptores informativos y sin contemplar conjuntos de datos altamente dimensionales. Dentro de las principales destacan: Análisis de correlación, mutual information, evaluaciones espaciales con respecto al entrenamiento de modelos empleando Random Forest, Análisis de componentes principales (PCA) y sus variantes como métodos lineales y reducción de dimensionalidad empleando métodos no lineales.

Análisis de correlación

Mutual information

Análisis espaciales de características

Métodos de reducción de dimensionalidad lineales

Métodos de reducción de dimensionalidad no lineales

3.1.5 Codificación de residuos con adición de información de su entorno

Adicional a las técnicas explicadas previamente con respecto a las codificaciones existentes, en algunos casos, no sólo basta con una única codificación del residuo, si no, que es relevante adicionar información que puede ser importante para describir los residuos. Normalmente, junto con las codificaciones basadas en propiedades fisicoquímicas, se emplean técnicas que permitan describir el ambiente bajo el cual se encuentre el residuo.

En la gran mayoría de los casos, se adiciona información de los residuos cercanos al residuo de interés, esto depende del tipo de datos bajo el cual se esté trabajando, es decir, si son secuencias lineales o son estructuras de proteínas en formato PDB.

Para el caso de que sean secuencias lineales, sea s secuencia de residuos de tamaño n y sea r_i el residuo de interés a evaluar su ambiente. Se crea una ventana de tamaño n' que contempla la cantidad de residuos r_j cercanos al residuo r_i, de tal manera que se crea un nuevo sub conjunto s' de datos de tamaño $2n'$ con n' residuos a la izquierda y n' a la derecha. El cual normalmente es codificado empleando binarización de elementos, así, en algunas ocasiones, a cada residuo, se le adicionan 20 descriptores que permiten indicar la ausencia o presencia de residuos cercanos a su entorno y el cual se completa con el conjunto de residuos s'.

Cuando se manejan estructuras de proteínas en formato PDB, la codificación y la evaluación del ambiente es similar. Sin embargo, en vez de utilizar una ventana de tamaño n' se utiliza un radio espacial de valor x para el cual, se toma el residuo y se estiman las distancias de los elementos cercanos, ya sea entorno a los carbonos α o a otros elementos. Esto, a diferencia de las secuencias lineales, permite adicionar información sobre las propiedades de distancia, ángulos y conformación de estabilidad por interacciones electrostáticas débiles que pueden generarse a partir de la proximidad de los elementos. No obstante, es una inferencia de su uso y se requieren de diferentes tipos de elementos que permitan caracterizar los eventos asociados al ambiente estructural asociado al residuo.

Actualmente, el uso de codificaciones mediante propiedades fisicoquímicas y el empleo de información adicional basada en descriptores de ambientes, es una de las metodologías más utilizadas a la hora de generar set de datos relacionados a mutaciones. Sin embargo, debido a que sólo se considera distancia, la binarización de los elementos no se ve afectada por sustituciones en residuos lejanos al lugar de ocurrencia, lo que denota la necesidad de idear metodologías que permitan contemplar el aporte completo de residuos a la caracterización de propiedades y cómo sustituciones puntuales afectan enormemente a residuos de interés. Una de las formas en las que se ha intentado dar solución a esta problemática, es modelar las propiedades fisicoquímicas de los residuos de las secuencias, a partir del uso de transformaciones de Fourier y en particular, empleando algoritmos relacionados a dichos conceptos, que aprovechen las ventajas referidas a la manipulación de espacios de frecuencias por sobre elementos temporales.

3.2 Transformaciones de Fourier

3.2.1 Uso de Transformadas de Fourier en digitalización de propiedades fisicoquímicas

3.3 Hipótesis

3.4 Objetivos

3.5 Metodología

Chapter 4

Filogenética, propiedades fisicoquímicas y minería de datos aplicadas al diseño de mutaciones en secuencias de proteínas

Chapter 5

Modelamiento matemático discreto aplicado al estudio de estructuras de proteínas.

Chapter 6

Reconocimiento de patrones y extracción de información en sistemas complejos multi-dimensionales

Chapter 7

Un caso de estudio completo: Aplicación de técnicas de minería de datos y reconocimiento de patrones para modelar el sistema de interacción antígeno anticuerpo

Referencias

[1] Abagyan, R. and Totrov, M. (1994). Biased probability monte carlo conformational searches and electrostatic calculations for peptides and proteins. *Journal of molecular biology*, 235(3):983–1002.

[2] Abdelaziz, A., Elhoseny, M., Salama, A. S., and Riad, A. (2018). A machine learning model for improving healthcare services on cloud computing environment. *Measurement*, 119:117 – 128.

[3] Abola, E. E., Bernstein, F. C., and Koetzle, T. F. (1984). The protein data bank. In *Neutrons in Biology*, pages 441–441. Springer.

[4] Alexov, E., Zhang, J., Wang, L., Zhenirovskyy, M., Gao, Y., and Zhang, Z. (2012). Predicting folding free energy changes upon single point mutations. *Bioinformatics*, 28(5):664–671.

[5] Alibés, A., Nadra, A. D., De Masi, F., Bulyk, M. L., Serrano, L., and Stricher, F. (2010). Using protein design algorithms to understand the molecular basis of disease caused by protein-dna interactions: the pax6 example. *Nucleic Acids Research*, 38(21):7422–7431.

[6] Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics.

[7] Ancien, F., Pucci, F., Godfroid, M., and Rooman, M. (2018). Prediction and interpretation of deleterious coding variants in terms of protein structural stability. *Scientific reports*, 8(1):4480.

[8] Barenboim, M., Masso, M., Vaisman, I. I., and Jamison, D. C. (2008). Statistical geometry based prediction of nonsynonymous snp functional effects using random forest and neuro-fuzzy classifiers. *Proteins: Structure, Function, and Bioinformatics*, 71(4):1930–1939.

[9] Bava, K. A., Gromiha, M. M., Uedaira, H., Kitajima, K., and Sarai, A. (2004). Protherm, version 4.0: thermodynamic database for proteins and mutants. *Nucleic Acids Research*, 32(suppl_1):D120–D121.

[10] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, 28(1):235–242.

[11] Berry, M. J. and Linoff, G. S. (2004). *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons.

[12] Bhargava, N., Sharma, G., Bhargava, R., and Mathuria, M. (2013). Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6).

[Bordner and Abagyan] Bordner, A. J. and Abagyan, R. A. Large-scale prediction of protein geometry and stability changes for arbitrary single point mutations. *Proteins: Structure, Function, and Bioinformatics*, 57(2):400–413.

[14] Braha, D. and Shmilovici, A. (2002). Data mining for improving a cleaning process in the semiconductor industry. *IEEE Transactions on Semiconductor Manufacturing*, 15(1):91–101.

[15] Breiman, L. (2017). *Classification and regression trees*. Routledge.

[16] Broom, A., Jacobi, Z., Trainor, K., and Meiering, E. M. (2017a). Computational tools help improve protein stability but with a solubility tradeoff. *J Biol Chem*, 292(35):14349–14361. 28710274[pmid].

[17] Broom, A., Jacobi, Z., Trainor, K., and Meiering, E. M. (2017b). Computational tools help improve protein stability but with a solubility tradeoff. *Journal of Biological Chemistry*, 292(35):14349–14361.

[18] Buß, O., Muller, D., Jager, S., Rudat, J., and Rabe, K. S. (2018). Improvement in the thermostability of a b-amino acid converting o-transaminase by using foldx. *ChemBioChem*, 19(4):379–387.

[19] Buß, O., Rudat, J., and Ochsenreither, K. (2018). Foldx as protein engineering tool: Better than random based approaches? *Computational and Structural Biotechnology Journal*, 16:25 – 33.

[20] Capriotti, E., Fariselli, P., and Casadio, R. (2005a). I-mutant2. 0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic acids research*, 33(suppl_2):W306–W310.

[21] Capriotti, E., Fariselli, P., and Casadio, R. (2005b). I-mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Res*, 33(Web Server issue):W306–W310. 15980478[pmid].

[22] Capriotti, E., Fariselli, P., Rossi, I., and Casadio, R. (2008a). A three-state prediction of single point mutations on protein stability changes. *BMC Bioinformatics*, 9(2):S6.

[23] Capriotti, E., Fariselli, P., Rossi, I., and Casadio, R. (2008b). A three-state prediction of single point mutations on protein stability changes. *BMC bioinformatics*, 9(2):S6.

[24] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

[25] Chen, C., Lu, Z., and Ciucci, F. (2017). Data mining of molecular dynamics data reveals li diffusion characteristics in garnet li7la3zr2o12. *Scientific Reports*, 7:40769 EP –. Article.

[26] Chen, J., Huang, H., Tian, S., and Qu, Y. (2009). Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, 36(3, Part 1):5432–5435.

[27] Chen, M., Hao, Y., Hwang, K., Wang, L., and Wang, L. (2017). Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879.

[28] Chien, C.-F. and Chen, L.-F. (2008). Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. *Expert Systems with Applications*, 34(1):280–290.

[29] Cooley, R., Mobasher, B., Srivastava, J., et al. (1997). Web mining: Information and pattern discovery on the world wide web. In *ictai*, volume 97, pages 558–567.

[30] Curtarolo, S., Morgan, D., Persson, K., Rodgers, J., and Ceder, G. (2003). Predicting crystal structures with data mining of quantum calculations. *Phys. Rev. Lett.*, 91:135503.

[31] Danielsson, P.-E. (1980). Euclidean distance mapping. *Computer Graphics and Image Processing*, 14(3):227 – 248.

[32] Duan, L., Street, W. N., and Xu, E. (2011). Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems*, 5(2):169–181.

[33] Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4):325–327.

[34] Dunham, M. H. (2006). *Data mining: Introductory and advanced topics*. Pearson Education India.

[35] Duygulu, P., Barnard, K., de Freitas, J. F. G., and Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In Heyden, A., Sparr, G., Nielsen, M., and Johansen, P., editors, *Computer Vision — ECCV 2002*, pages 97–112, Berlin, Heidelberg. Springer Berlin Heidelberg.

[36] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37.

[37] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996b). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34.

[38] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996c). Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88.

[39] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

[40] Forbes, S. A., Bindal, N., Bamford, S., Cole, C., Kok, C. Y., Beare, D., Jia, M., Shepherd, R., Leung, K., Menzies, A., Teague, J. W., Campbell, P. J., Stratton, M. R., and Futreal, P. A. (2010). Cosmic: mining complete cancer genomes in the catalogue of somatic mutations in cancer. *Nucleic Acids Research*, 39(suppl_1):D945–D950.

[41] Freund, Y. and Mason, L. (1999). The alternating decision tree learning algorithm. In *icml*, volume 99, pages 124–133.

[42] Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

[43] Ge, Z., Song, Z., Ding, S. X., and Huang, B. (2017). Data mining and analytics in the process industry: The role of machine learning. *IEEE Access*, 5:20590–20616.

[44] Getov, I., Petukh, M., and Alexov, E. (2016a). Saafec: Predicting the effect of single point mutations on protein folding free energy using a knowledge-modified mm/pbsa approach. *Int J Mol Sci*, 17(4):512–512. 27070572[pmid].

[45] Getov, I., Petukh, M., and Alexov, E. (2016b). Saafec: predicting the effect of single point mutations on protein folding free energy using a knowledge-modified mm/pbsa approach. *International journal of molecular sciences*, 17(4):512.

[46] Gossage, L., Pires, D., Olivera-Nappa, A., A. Asenjo, J., Bycroft, M., Blundell, T., and Eisen, T. (2014). An integrated computational approach can classify vhl missense mutations according to risk of clear cell renal carcinoma. *Human molecular genetics*, 23.

[47] Han, J. and Gao, J. (2009). Research challenges for data mining in science and engineering. *Next Generation of Data Mining*, pages 1–18.

[48] Hand, D. J. (2006). Data mining. *Encyclopedia of Environmetrics*, 2.

[49] Heselpoth, R. D., Yin, Y., Moult, J., and Nelson, D. C. (2015). Increasing the stability of the bacteriophage endolysin PlyC using rationale-based FoldX computational modeling. *Protein Engineering, Design and Selection*, 28(4):85–92.

[50] Hofmann, M. and Klinkenberg, R. (2013). *RapidMiner: Data mining use cases and business analytics applications*. CRC Press.

[51] Jain, A. K., Dubes, R. C., et al. (1988). *Algorithms for clustering data*, volume 6. Prentice hall Englewood Cliffs.

[52] John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc.

[53] Keller, J. M., Gray, M. R., and Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(4):580–585.

[54] Khan, S. and Vihinen, M. (2010). Performance of protein stability predictors. *Human Mutation*, 31(6):675–684.

[55] Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A., and Robles, V. (2006). Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112.

[56] Lee, J. K., Williams, P. D., and Cheon, S. (2008). Data mining in genomics. *Clinics in Laboratory Medicine*, 28(1):145–166.

[57] Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In Nédellec, C. and Rouveirol, C., editors, *Machine Learning: ECML-98*, pages 4–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

[58] Li, M., Wang, J., and Chen, J. (2008). A fast agglomerate algorithm for mining functional modules in protein interaction networks. In *2008 International Conference on BioMedical Engineering and Informatics*, volume 1, pages 3–7.

[59] Li, M., Wang, J., and Chen, J. (2008). A fast agglomerate algorithm for mining functional modules in protein interaction networks. In *2008 International conference on biomedical engineering and informatics*, volume 1, pages 3–7. IEEE.

[60] Liao, H. and Xu, Z. (2015). Approaches to manage hesitant fuzzy linguistic information based on the cosine distance and similarity measures for hfltss and their application in qualitative decision making. *Expert Systems with Applications*, 42(12):5328 – 5336.

[61] Ma, X., Wu, Y.-J., Wang, Y., Chen, F., and Liu, J. (2013). Mining smart card data for transit riders' travel patterns. *Transportation Research Part C: Emerging Technologies*, 36:1–12.

[62] Maesschalck, R. D., Jouan-Rimbaud, D., and Massart, D. (2000). The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1 – 18.

[63] Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

[64] Mao, L., Wang, Y., Liu, Y., and Hu, X. (2004). Molecular determinants for atp-binding in proteins: A data mining and quantum chemical analysis. *Journal of Molecular Biology*, 336(3):787 – 807.

[65] Masso, M. and Vaisman, I. I. (2008). Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis. *Bioinformatics*, 24(18):2002–2009.

[66] Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69. Mountain View, CA.

[67] Michie, D., Spiegelhalter, D. J., Taylor, C., et al. (1994). Machine learning. *Neural and Statistical Classification*, 13.

[68] Muñoz, V., Blanco, F. J., and Serrano, L. (1995). The hydrophobic-staple motif and a role for loop-residues in α-helix stability and protein folding. *Nature structural biology*, 2(5):380.

[69] Muñoz, V. and Serrano, L. (1996). Local versus nonlocal interactions in protein folding and stability–an experimentalist's point of view. *Folding and Design*, 1(4):R71–R77.

[70] Nadra, A. D., Serrano, L., and Alibés, A. (2011). Chapter one - dna-binding specificity prediction with foldx. In Voigt, C., editor, *Synthetic Biology, Part B*, volume 498 of *Methods in Enzymology*, pages 3 – 18. Academic Press.

[71] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

[72] Obenshain, M. K. (2004). Application of data mining techniques to healthcare data. *Infection Control &; Hospital Epidemiology*, 25(8):690–695.

[73] Olivera-Nappa, A., Andrews, B. A., and Asenjo, J. A. (2011). Mutagenesis objective search and selection tool (mosst): an algorithm to predict structure-function related mutations in proteins. *BMC Bioinformatics*, 12(1):122.

[74] Pandurangan, A. P., Ochoa-Montaño, B., Ascher, D. B., and Blundell, T. L. (2017). Sdm: a server for predicting effects of mutations on protein stability. *Nucleic Acids Res*, 45(W1):W229–W235. 28525590[pmid].

[75] Parthiban, V., Gromiha, M. M., and Schomburg, D. (2006). Cupsat: prediction of protein stability upon point mutations. *Nucleic Acids Res*, 34(Web Server issue):W239–W242. 16845001[pmid].

[76] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[77] Perlibakas, V. (2004). Distance measures for pca-based face recognition. *Pattern Recognition Letters*, 25(6):711 – 724.

[78] Petukh, M., Dai, L., and Alexov, E. (2016). Saambe: webserver to predict the charge of binding free energy caused by amino acids mutations. *International journal of molecular sciences*, 17(4):547.

[79] Petukhov, M., Cregut, D., Soares, C. M., and Serrano, L. (1999). Local water bridges and protein conformational stability. *Protein Science*, 8(10):1982–1989.

[80] Potapov, V., Cohen, M., and Schreiber, G. (2009). Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Engineering, Design and Selection*, 22(9):553–560.

[81] Quan, L., Lv, Q., and Zhang, Y. (2016a). Strum: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics*, 32(19):2936–2946. 27318206[pmid].

[82] Quan, L., Lv, Q., and Zhang, Y. (2016b). Strum: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics*, 32(19):2936–2946.

[83] Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. (2015). Big data meets quantum chemistry approximations: The d-machine learning approach. *Journal of Chemical Theory and Computation*, 11(5):2087–2096.

[84] Rebhan, M., Chalifa-Caspi, V., Prilusky, J., and Lancet, D. (1998). Genecards: a novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics*, 14(8):656–664.

[85] Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D. (2004). Protein structure prediction using rosetta. In *Methods in enzymology*, volume 383, pages 66–93. Elsevier.

[86] Sánchez, I. E., Beltrao, P., Stricher, F., Schymkowitz, J., Ferkinghoff-Borg, J., Rousseau, F., and Serrano, L. (2008). Genome-wide prediction of sh2 domain targets using structural information and the foldx algorithm. *PLOS Computational Biology*, 4(4):e1000052.

[87] Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., and Serrano, L. (2005). The foldx web server: an online force field. *Nucleic Acids Res*, 33(Web Server issue):W382–W388. 15980494[pmid].

[88] Sun, L., Li, L., Peng, Y., Jia, Z., and Alexov, E. (2017). Predicting protein-dna binding free energy change upon missense mutations using modified mm/pbsa approach: Sampdi webserver. *Bioinformatics*, 34(5):779–786.

[89] Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4):667 – 671.

[90] Tian, J., Wu, N., Chu, X., and Fan, Y. (2010). Predicting changes in protein thermostability brought about by single- or multi-site mutations. *BMC Bioinformatics*, 11(1):370.

[91] Vaisman, I. I. and Masso, M. (2008). Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis. *Bioinformatics*, 24(18):2002–2009.

[92] Vijayakumar, M., Wong, K.-Y., Schreiber, G., Fersht, A. R., Szabo, A., and Zhou, H.-X. (1998). Electrostatic enhancement of diffusion-controlled protein-protein association: comparison of theory and experiment on barnase and barstar. *Journal of molecular biology*, 278(5):1015–1024.

[93] Wainreb, G., Ashkenazy, H., Wolf, L., Ben-Tal, N., and Dehouck, Y. (2011). Protein stability: a single recorded mutation aids in predicting the effects of other mutations in the same amino acid site. *Bioinformatics*, 27(23):3286–3292.

[94] Walt, S. v. d., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30.

[95] Yang, H., Parthasarathy, S., and Mehta, S. (2005). A generalized framework for mining spatio-temporal patterns in scientific data. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 716–721, New York, NY, USA. ACM.

[96] Yoo, I., Alafaireet, P., Marinov, M., Pena-Hernandez, K., Gopidi, R., Chang, J.-F., and Hua, L. (2012). Data mining in healthcare and biomedicine: A survey of the literature. *Journal of Medical Systems*, 36(4):2431–2448.

[97] Zhang, H. (2004). The optimality of naive bayes. *AA*, 1(2):3.