Un primer estudio estadístico de la Certificación en la UACM

Carlos E. Martínez-Rodríguez*

27 de noviembre de 2022

Índice

*Departamento de Estadística, Universidad Autónoma de la Ciudad de México (UACM). Correo electrónico: carlos.martinez@uacm.edu.mx

1. Introducción y antecedentes

1.1. Artículo 1: Machine Learning in Enzyme Engineering

Título: Machine Learning in Enzyme Engineering, Stanislav Mazurenko, Zbynek Prokop, and Jiri Damborsky [1]

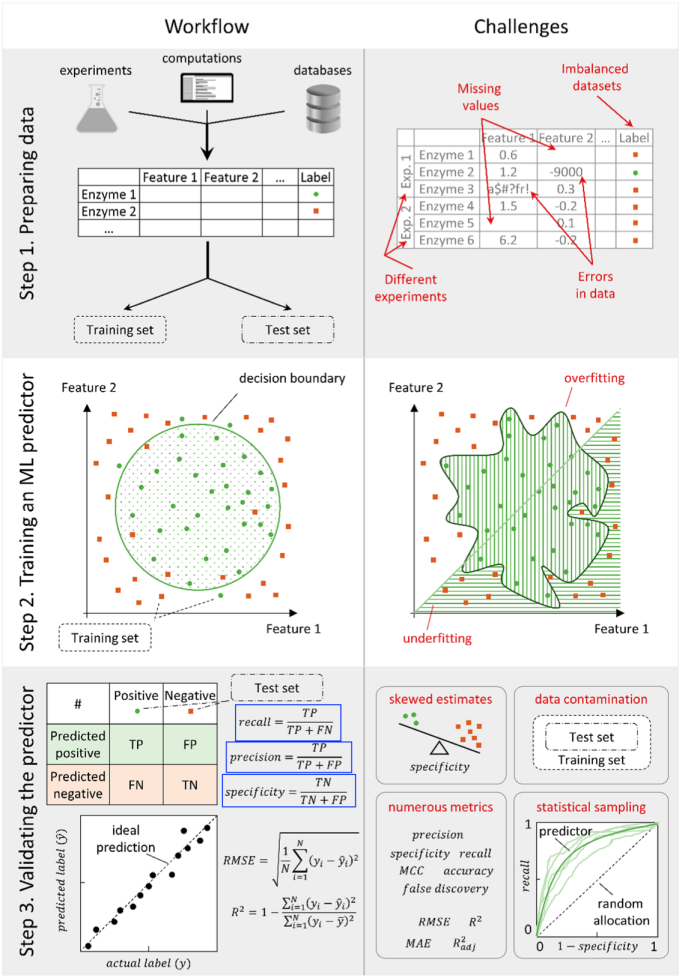
- Enzyme engineering is the process of customizing new biocatalysts with improved properties by altering their constituting sequences of amino acids.
- Multiple ML algorithms have already been applied to enzyme engineering. Some notable examples include random forests used to predict protein solubility [2], support vector machines [3, 4] and decision trees [5] to predict enzyme stability changes upon mutations, K-nearest-neighbor classifiers to predict enzyme function[6] and mechanisms,[7] and various scoring and clustering algorithms for rapid functional sequence annotation [8, 9]. The main attractiveness of ML in enzyme engineering stems from its generalizability: once it is trained on the known input, called a training set, an ML algorithm can potentially make predictions about new variants almost instantly.
- The aim of this Perspective is, therefore, to highlight recent advances in data collection and algorithm implementation for ML in enzyme engineering.

1.2. The essence of Machine Learning

La esencia de la mayoría de los algoritmos de Machine Learning (ML) es encontrar patrones en los datos disponibles, datos que consisten en varios descriptores o características, por ejemplo secuencias de encimas, sus estructuras secundarias y terciarias, substiruciones, etc. El nímero de características usualmente varían de decenas a miles lo que convierte el problema en uno de alta dimensión.

Los principales tipos de Machine Learning son: Aprendizaje Supervisado y Aprendizaja No-Supervisado. En el aprensizaje no supervisado el objetivo es disminuir la alta dimensionalidad de los datos en uno de menor dimensión, o el de encontrar clústers en los datos. En el aprendizaje supervisado varias propiedades objetivo tales como actividad o estabilidad de enzimas, y el objetivo es diseñar un predictor que regrese etiquetas para datos no vistos considerando sus descriptores, utilizando el conjunto de datos etiquetado como datos de entrenamiento.

Nota 1 *Step 1: the data are usually turned into a table format and split into the training and test parts. Any errors, biases, or imbalances will be translated to the predictor's performance and, hence, must be accounted for. Step 2: the predictor is trained on the training data set. For example, a decision boundary is derived that allows classifying future input based on whether data points are inside or outside the boundary. This is a balancing act between two extremes: explaining noise rather than fundamental dependencies (overfitting) or failure to account for complex dependencies in the data (underfitting). Step 3: the performance of the predictor is evaluated based on the test data set. For example, true and false positives and negatives and the associated measures are calculated or the root mean square error (RMSE)*



ht!

Figura 1: Schematic workflow of constructing an ML predictor and associated challenges.

is calculated for continuous labels. The random nature of the initial data split as well as data imbalances might skew the evaluation, and numerous metrics used for evaluation vary in their robustness to different data skews. Even partial inclusion of the test set at any stage of ML predictor training is called data contamination and usually invalidates the final evaluation.

La etapa que más tiempo consume es la de recolección de datos y su preparación para alimentar el algoritmo de ML, entonces los datos son introducidos en el subconjunto de entrenamiento, el resultado se utiliza para meujorar los parámetros del predictor de ML, mientras que el segundo se utiliza para la evaluación.

Nota 2

- *En problemas de clasificación con etiquetas binarias o etiquetas con una cantidad finita de opciones, la evaluación usualmente se realiza por medio de la matriz de confusión: el número de verdaderos/falsos positivos y negativos.*

	Positivo	*Negativo*
Predecido Positivo	*TP*	*FP*
Predicido Negativo	*FN*	*TN*

- *Para problemas de regresión con etiquetas de valores continuos usualmente se calcula la raíz del error cuadrático medio*

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}\right)^2} \tag{1}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}\right)^2}{\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2} \tag{2}$$

En cualquiera de los dos casos la evaluación final se lleva a cabo en el conjunto de prueba, el cuál es esencial dado que el último objetivo es obtener el predictor más general en los datos no utilizados para entrenar el algoritmo.

Nota 3 *Las siguientes métricas se utilizan para medir el rendimiento de un modelo en función de su capacidad para predecir correctamente las clases de un conjunto de datos.*

- ***Recall (Recall o Sensibilidad):*** *Conocido como sensibilidad o tasa positiva real, mide la capacidad de un modelo para identificar correctamente todos los ejemplos positivos en un conjunto de datos. Se calcula como el número de verdaderos positivos dividido por la suma de verdaderos positivos y falsos negativos:*

$$Recall = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Negativos} \tag{3}$$

Un recall alto significa que el modelo es bueno para detectar los casos positivos, minimizando los falsos negativos. Es importante en situaciones donde los falsos negativos son costosos o críticos.

- *Precision (Precisión): La precisión mide la capacidad de un modelo para predecir correctamente los casos positivos entre todas las predicciones positivas que realiza. Se calcula como el número de verdaderos positivos dividido por la suma de verdaderos positivos y falsos positivos:*

$$Precision = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Positivos} \tag{4}$$

  Una alta precisión significa que el modelo tiene una baja tasa de falsos positivos, es decir, que cuando predice una clase como positiva, es probable que sea correcta. La precisión es importante en situaciones en las que los falsos positivos son costosos o no deseados.

- *Specificity (Especificidad): La especificidad mide la capacidad de un modelo para predecir correctamente los casos negativos entre todas las predicciones negativas que realiza. También se conoce como tasa negativa real. Se calcula como el número de verdaderos negativos dividido por la suma de verdaderos negativos y falsos positivos:*

$$Specificity = \frac{Verdaderos\ Negativos}{Verdaderos\ Negativos + Falsos\ Positivos} \tag{5}$$

  Una alta especificidad indica que el modelo es bueno para identificar correctamente los casos negativos, minimizando los falsos positivos. Esto es importante en situaciones en las que los falsos positivos son costosos o problemáticos.

Estas métricas proporcionan una forma más completa de evaluar el rendimiento de un modelo de clasificación que simplemente mirar la precisión general.

En la ingeniería de proteínas, las similitudes en secuencias en ambos subconjuntos de datos deben ser tenidas en cuenta. Si alguna familia de proteínas está sobre representada en el conjunto de prueba, el predictor resultante puede resultar sesgado hacia la identificación de patrones válidos solamente para esta familia. Si algunas secuencias en el conjunto de prueba son muy cercanas al conjunto de entrenamiento, la evaluación final de desempeño dara resultados sobre optimistas.

En el paso 2 de entrenamiento, es posible ajustar el predictor o seleccionar de entre varios predictores, usualmente por medio de validación $k - fold$. En este caso los datos de entrenamiento se subdividen en K subconjuntos y el flujo de trabajo se repite K veces, con cada uno de ellos utilizaods para la evaluación de los $K - 1$ subconjuntos utilizados para entrenar. El reto principal en el paso 2 para cualquiern entrenamiento tipo ML supervisado es evitar el subajuste de los datos (sesgo alto) y el sobre ajuste (varianza grande).

La **subestimación** ocurre cuando un predictor falla en encontrar patrones incluso en los datos de entrenamiento (cuando un modelo lineal simple se utiliza para explicar dependencia dependencias no lineales en los datos). El **sobreajuste** ocurre cuando el desempeño de un predictor disminuye notablemente en los datos de prueba en comparación con los datos de prueba, debido al aprendizaje de demasiado detalle y ruido, en lugar de identificar patrones generales. Tanto el subajuste como el sobreajuste pueden ser debido a la insuficiente calidad de los datos: ruido excesivo, características faltantes o irrelevantes, sesgo en los datos, o

datos dispersos. También pueden ocurrir como consecuencia de una pobre aplicación del algoritmo: excesiva o insuficiente flexibilidad en la selección de los parámetros, protocolo de entrenamiento inapropiado, o contaminación de los datos de entrenamiento con el conjunto de datos de prueba.

1.3. Bases de datos relevantes a Ingeniería de Enzima

1.3.1. The State of the Art in Data Accumulation

Debido a que los algoritmos de ML se basan en los datos, la importancia de la calidad de los mismos utilizados para entrenamiento no puede ser subestimada.

Ejemplos de conjuntos de bases de datos, utilizadas en la ingeniería de enzimas, son secuencias de proteínas y estructuras de proteínas. La estabilidad y solubilidad de las proteínas son dos cualidades que han sido medidas por varias décadas y hasta la fecha. Tareas más desafiante es la anotar las propiedades catalíticas de las enzimas debido a la abundancia de tipos de reacciones, mecanismos, cofactores, amplios rangos de especificidades de sustratos, enancionselectividades y promiscuidades.

Las enzimas son catalizadores biológicos que facilitan una amplia variedad de reacciones químicas en los organismos vivos. Algunas razones por las cuales la anotación de sus propiedades catalíticas es complicada son:

- **Tipos de reacciones diversos:** Las enzimas pueden catalizar una amplia gama de reacciones químicas, incluyendo reacciones de oxidación-reducción, hidrólisis, condensación, isomerización y más. Cada tipo de reacción involucra mecanismos químicos y sustratos diferentes.

- **Mecanismos:** Incluso dentro de un solo tipo de reacción, las enzimas pueden emplear múltiples mecanismos. Comprender el mecanismo específico utilizado por una enzima requiere un conocimiento detallado de la estructura de la enzima y de su sitio activo.

- **Cofactores:** Muchas enzimas requieren cofactores, como iones metálicos o coenzimas, para catalizar reacciones de manera efectiva. Identificar los cofactores necesarios para cada enzima es esencial para la anotación.

- **Condiciones de reacción:** La actividad enzimática puede depender en gran medida de las condiciones ambientales, incluyendo la temperatura, el pH y la fuerza iónica. La anotación de las condiciones óptimas para la actividad enzimática es crucial.

- **Especificidades de sustratos:** Las enzimas pueden ser altamente específicas para ciertos sustratos, reconociéndolos con alta afinidad, mientras que otras son más promiscuas y pueden unirse a una variedad de sustratos. La caracterización de la especificidad de sustratos es compleja.

- **Enantioselectividades:** Algunas enzimas pueden discriminar entre enantiómeros (isómeros de imagen especular) de una molécula, catalizando reacciones con alta selectividad por un enantiómero. La anotación de esta propiedad implica comprender la estereoquímica.

- **Promiscuidades:** Las enzimas pueden exhibir actividades promiscuas, catalizando reacciones diferentes a su función principal. Detectar y caracterizar tales promiscuidades es complicado.

1.3.2. Current Challenges Related to Databases

Si la dependencia que se busca no se encuentra en los datos disponibles, no importa la cantidad de nuevos datos ayudarán a mejorar la calidad del predictor de ML. En el caso de la ingeniería de enzimas se espera que las funciones enzimáticas estén codificadas en las secuencias y así depender en las propiedades físico-químicas de los aminoácidos, de aquí que la cantidad y la calidad de los datos en las bases de datos sean de importancia para diseñar un predictor de ML.

La falta de estándares en los reportes resulta en pérdida de información o valores erróneos para algunos descriptores. A esto hay que agregar la falta de protocolos robustos en los análisis de datos, como por ejemplo aquellos utilizados para ajuste de curvas para determinar las temperaturas de fusión o constantes cinéticas. Otro factor es que los recientes avances vuelven obsoletos resultados previos. La curación manual ayuda mejorar la calidad de los datos, sin embargo no se encuentra excenta de errores de anotación de las funciones de las proteínas y errores de propagación a partir de resultados previamente refutados.

Este tipo de procedimientos, verificación manual, puede incluir la limpieza o formato de datos para que sean amigables con ML. Uno de los principios más populares es **FAIR**, por sus siglas en inglés, Findable, Accesible, Interoperables y Reutilizables, debería de facilitar a las computadoras para que de manera automática pueda encontrar y utilizar los datos. Para las enzimas la guía estándar para reportar datos de enzimas (STRENDA) debería de aumentar la calidad de los datos, especialmente en bases de datos (bdd) heterogéneas recopiladas de diversas fuentes.

El desarrollo de nuevos predictores de ML ha incrementado considerablemente la demanda de mejora de las bdd existentes, así como la generación de nuevas bdd uniformes y representativas de mayor calidad.

Existen varias nuevas técnicas emergentes tales como

i Secuenciación de nueva generación.

ii Clasificación de células activadas por fluorescencia.

iii Exploración mutacional profunda, y

iv Microfluidos

1.3.3. Emerging Methods for High-Throughput Data Collection

Avances tecnológicos hacia la miniaturización, automatización y paralelización han generado tecnologías eficientes de nuevos métodos de investigación experimental con capacidades incomparables. Secuenciación de nueva generación (NGS) ha revolucionado la investigación genómica, habilitado el acceso a datos moleculares fundamentales y revelado firmas genómicas y transcriptomicas [10, 11].

La capacidad de secuencias en el rango de gigabases por ejecución del instrumento permite secuencias el genoma humano en su totalidad en tan sólo un día.

Múltiples instrumentos comerciales de segunda generación disponibles ofrecen mayor capacidad y precisión. Métodos de tercera generación recientemente introducidos (lectura larga) que emplean secuenciación en tiempo real de moléculas [12] o secuenciación de nanoporos [13] resuelve las limitaciones de lecturas cortas, como el sesgo de GC o mapear elementos repetitivos

Mientas el avance de la tecnología de secuenciación proporciona una gran cantidad de secuencias de datos, para la mayoría de estas entradas, las anotaciones estructurales y anotaciones funcionales aún están perdidas.

Como siguiente paso, se está centrando en el desarrollo de nuevos métodos experimentales efectivos para recopilar información funcional y estructural.

La clasificación de células activadas por fluorescencia (FACS) es una tecnología ampliamente disponible que permite el cribado de hasta 108 variantes de enzimas por día. La FACS requiere que los sustratos fluorogénicos estén atrapados dentro o en la superficie de la célula para vincular el genotipo y el fenotipo. Alternativamente, se utiliza la clasificación de enzimas encapsuladas junto con su ADN codificador y un sustrato fluorogénico en perlas de hidrogel.

Cuando se combinan con la secuenciación de próxima generación, los ensayos de alto rendimiento representan una estrategia poderosa para analizar de manera integral las relaciones entre secuencia y función en las enzimas [14, 15]. Este enfoque, llamado escaneo mutacional profundo (DMS por sus siglas en inglés), vincula el genotipo con el fenotipo sin necesidad de procesos laboriosos que involucren la purificación y caracterización de proteínas. Durante el proceso, se sintetiza una gran biblioteca de secuencias mutantes, seguida de la selección de fenotipos expresados. Luego, la secuenciación de la biblioteca antes y después de la selección cuantifica la aptitud de cada mutante. De esta manera, el DMS proporciona un método rápido y sencillo para inferir los determinantes de la secuencia de la estabilidad y la función de las proteínas[14, 16, 17]. El DMS se ha utilizado como una estrategia experimental alternativa para la determinación de la estructura de las proteínas.

1.4. MACHINE LEARNING APPLICATIONS TO ENZYME ENGINEERING

- Despite being a relatively new field of study, machine Learning for enzyme engineering has already been applied for several challenging predictions. First consider predictors aimed at elucidating the structure function relationships crucial for enzymes on both sides:
    - predicting the structure for a known sequence, and
    - predicting the catalytic activity or substrate specificity for a known sequence-structure.
    - solubility and stability, from the point of view of amino acid substitutions,
- The protein structure prediction is one of the longest-standing challenges in biochemistry, as the number of resolved structures is dramatically lagging behind the number

of known sequences.

- Over 145000 structures have been released in the Protein Data Bank, but this is still nowhere near over 215 million publicly available protein sequences[18].
- Nevertheless, even despite a relatively small data set size in comparison to millions of data points usually available for this method, deep neural networks showed most the notable results in the latest biennial assessment of protein structure prediction methods, CASP13.

  Nota 4 *The AlphaFold network was trained on the PDB entries to predict the distances between C-beta atoms of residues using multiple sequence alignments[19] and received the highest score at the competition. Out of 124 targets, around two-thirds of AlphaFold predictions had a GDT_TS score above 50, which is indicative of a topologically correct structure [20].*

- Apart from predicting protein structures, predicting catalytic activities is another active field of research currently.
- Computational methods for the protein function prediction range from sequence-to structure-based and from gene-to genome- and interactome-based[21].
- Several initiatives similar to the CASP competition have already been proposed to address the functional annotation of enzymes, namely Enzyme Function Initiative (EFI), the Computational Bridges to Experiments initiative (COMBREX), and the Critical Assessment of Function Annotation community-driven experiment (CAFA).
- Successful attempts to apply ML to assign enzyme EC numbers using predicted 3D structures [22] or exploiting sequence similarities [23] have already been made.
- Recently, deep learning was also applied to predict EC numbers on the basis of a protein sequence using both sequence-length-dependent features, such as raw sequence one-hot encoding, and sequence-length-independent features, such as functional domain encoding [24].
- The former type of features introduced non uniformity in feature dimensionality, and the authors presented a framework to perform simultaneously dimensionality uniformization, feature selection, and classification model training.
- The large data sets of enzyme structures and activities accumulated to date already allow using deep learning in the engineering of catalytic activity.
- A more precise functional prediction is possible by restricting ML training to a particular family of enzymes, which comes at the cost of much smaller data sets available for training. This problem may be tackled by applying high-throughput data collection methods mentioned before. The authors of the recently released GT-predict [25] selected for their analysis the glycosyltransferase superfamily 1, a group of enzymes with highly diverse substrates.

Nota 5 *Data from the label-free mass spectroscopy-based assay of 91 substrates and 54 enzymes derived from the plant Arabidopsis thaliana were used for functional prediction. The authors trained sequence-based decision trees, systematically varying combinations of physicochemical properties, e.g. log P, molecular area, and number/type of nuclephilic groups, and structural information, e.g. scaffold type and functional groups. The resulting predictor was successfully tested on four individually selected gene sequences as well as two complete families of enzymes from four different organisms, which highlights the tremendous potential of training ML predictors on the newly acquired data from high-throughput data collection methods.*

- In their recent paper [29] Han and coauthors considered seven different binary and continuous ML algorithms: logistic regression, decision tree, support vector machines, Naive Bayes, conditional random forest, XGBoost, and artificial neural networks.
- The authors attempted to use generative adversarial networks to synthesize more data. This is a pair of two neural networks competing against each other: one learns to generate artifficial examples and the other to distinguish them from real data.
- Another point of view on protein solubility prediction is studying the effects of individual mutations. The recent successes in the application of deep mutational scanning to collect the data on protein solubility changes upon mutations[27] are likely to promote the development of more sophisticated ML-based protein solubility predictors in the nearest future.
- Predicting the effects of amino acid substitutions is not only limited to solubility: stability, substrate specificity, catalytic activity, and enantioselectivity can also be targeted if sufficient data are available.
- Protein stability predictors are perhaps those with the most abundant data sets of this type available for ML training
- The authors also presented a random forest classiffier trained using 1106 features from the following groups: experimental conditions, conservation and coevolution scores for mutated positions, amino acid substitutions and their physicochemical properties, neighborhood features for 11 positions before and after substitution sites, and thermodynamic sequence-based features extracted from ProtDCal [28].
- PON-tstab is a three-class predictor (stability increasing, decreasing, unchanged) and achieved the correct prediction ratio of around 0.5 versus the value 0.33 for a random predictor. This implies that, even with a data set of higher quality, predicting protein stability remains an extremely challenging task [30].
- Another intriguing application of ML in protein engineering is to design smart combinatorial libraries for directed protein evolution[31]. This has the potential to both reduce the experimental effort and improve the exploration of the sequence space by mutating multiple positions simultaneously. Moreover, it can approximate the empirical fitness landscape to suggest a refined set of variants for the next round of screening.

- Wu et al[32] used ML assisted directed evolution to engineer an enzyme for a new stereodivergent carbon-silicon bond formation. The authors selected the reaction of phenyldimethyl silane with ethyl 2-diazopropanoate catalyzed by a putative nitric oxide dioxygenase from Rhodothermus marinus. They tested a variety of ML algorithms such as linear and kernel models, shallow neural networks, and ensemble methods to improve the enzyme enantioselectivity.

1.4.1. Current Challenges Related to ML-Aided

- One of the main challenges in applications of ML to enzyme engineering stems from the intrinsic multidisciplinarity of the approach. Biochemists, molecular biologists, mathematicians, and computer scientists have to find a common language to clarify goals, carry out rigorous analysis and training, and avoid common pitfalls, wrongful usage of methods, and misinterpretations.

- The No Free Lunch theorem [33] claims that no single ML method is superior to others a priori [34];

- A thorough understanding of the data types to be used and problems to be solved is essential in the development of efficient predictors. The current shift toward new and more complex ML methods, namely aggregating several algorithms into hybrid meta-predictors, hyperparameter optimization with many training subcycles, feature learning, and the fusion of ML-based and classical bioinformatics tools in a single predictor, will further challenge the crosstalk between disciplines necessary for the development of efficient and robust predictors in enzyme engineering.

- With the continuous growth of ML applications in enzyme engineering, the need for robust comparison of various predictors is of growing importance. This comparison is mainly obstructed by the lack of both standardized protocols for comparison and new data sets for testing.

- The lack of benchmark data sets, discrepancies in the performance measurements used, inaccurate or insufficient disclosure in publications, and the difficulty in finding reviewers with sufficiently broad expertise [35] are among the most pressing issues.

- Researchers working on some applications with a long track record in bioinformatics, such as protein structure or function predictions, have already established several platforms that can be used for comparison of the ML predictors, i.e. CASP, CAFA, EFI, and COMBREX.

- Other applications have yet to see similar initiatives as at least three key ingredients are necessary:

  i a sufficiently large community of researchers working on development of such applications,

  ii a sufficient amount of new high-quality data being collected regularly, and

iii a leader that will take on responsibility and invest time and effort into coordinating this activity.

- Few papers go beyond simple ROC analysis: e.g., resample cross-validation to estimate its statistical significance, explore the reasons for weak predictions, and analyze learning curves. Why does a particular predictor have a better performance? What features are critical for the performance of a predictor on a global scale? What ranges for feature values and what parts of the feature space are most critical for a particular data point to be classiffied correctly?

1.5. Emerging Trends in ML-Based Methods for Enzyme Engineering

- With the accumulation of more data by virtue of the emerging high-throughput experimental methods, the development of benchmark data sets and unified performance measurements is only a matter of time.

- Recently, an intriguing algorithm based on semisupervised learning has been presented to allow benchmarking in five different prediction tasks related to protein engineering, including secondary structure, fluorescence landscape, and stability landscape predictions [36]

- As the data generation is streamlined, a data set from a single experiment is starting to have the size large enough for training ML algorithms to guide the design of future experiments, as was the case in the development of stereodivergent carbon silicon bond formation[32] and the application of Gaussian processes to the directed evolution of cytochromes[37].

- The increase in the available data will prompt more extensive use of deep neural networks. Sophisticated neural network architectures, such as recurrent or graph-based neural networks, simultaneous training of several types of predictors (multitasking), combining structurally different input data (multimodal design), ML-based modeling of data sets (generative models), and retraining predictors used in one area by new data from another area (transfer learning) have only recently been applied in genomics[38]

- Several attempts have recently been made to apply some of those advanced techniques to proteins: using generative models to create soluble and functional malate dehydrogenase variants[39] or predict mutational effects with high correlation with those actually observed in 42 high-throughput deep mutational scanning experiments[40]

- More data will also allow improving the existing methods, i.e. learning the optimal architecture of a predictor from the data (hyperparameter optimization)[41], smart aggregation of several predictions from multiple methods[42], and introducing robust conffidence scores for predictions[43]. In enzyme engineering, this new level of algorithmic complexity will further save time and resources wasted on validating misleading

predictions but will also require more sophisticated computer architecture, e.g. an increased use of parallel computing and stochastic training methods, which have already become standard techniques for the acceleration of deep neural network training.

- Another noticeable trend in ML is toward interpretable and explainable predictors[44]. Apart from the global importance of features for ML predictors, feature importance scores calculated for each input example[45, 46] may help explain why a particular prediction was made for each input data point.

- Interpretable algorithms can aid in smart biocatalyst design. For instance, instead of simply screening all the possible mutations with an ML-based tool to improve a target property, researchers can make use of designing variants on the basis of the structure of a predictor using adaptive sampling[47].

- Such an approach favors predictors whose parameters can provide such guidance: e.g., linear predictors over more flexible yet harder to interpret artifficial neural networks (Figure 3). Linear predictors allow analytical design on the basis of the coefficients [48] in contrast, sophisticated predictors are usually prone to pathological behavior, i.e. sudden misclassiffication after a slight and almost imperceptible perturbation of input[49].

- Another promising approach is to use interpretable architectures of predictors already at the design stage, e.g. the visible neural networks[50]. The design of such networks is guided by the knowledge of the underlying biological mechanism, e.g. the choice of layers and the connections between layers may mimic the hierarchical organization of transcriptional regulatory factors in the cell nucleus.

2. Artículo 2: A general model to predict small molecule substrates of enzymes based on machine and deep learning

2.1. About the existence techniques

Nota 6

- *Las enzimas han evolucionado de manera eficiente para catalizar eficientemente una o más reacciones químicas incrementando las tasas de reacción hasta más de un millón. Además la mayoría de las enzimas son promiscuas, es decir, se catalizan más, fisiológicamente irrelevantes o reacciones inofensivas [51, 52]*

- *Un mapeo exhaustivo de las relaciones enzima-sustrato desempeña un papel crucial en la investigación farmacéutica y bioingeniería, por ejemplo, para la producción de medicamentos, productos químicos, alimentos y biocombustibles [53, 54, 55].*

- *La base de datos UniProt contiene entradas de más de 36 millones de enzimas diferentes, más del* 99 % *de ellas carecen de anotaciones de alta calidad de las reacciones catalizadas. Se están realizando esfuerzos para desarrollar métodos de alto rendicmiento*

para la determinación experimentarl de relaciones entre enzimas y sustratos, esfuerzos que se encuentran en su etapa inicial[56, 57, 58].

- *Se han realizado esfuerzos por desarrollar metodos de alto rendimiento para determinar las relaciones experimentales de las relaciones enzima-sustratos. El objetivo en este trabajo es desarrollar un modelo de Machine Learning (ML) con la capacidad de predecir las relaciones enzima-sustratos a traves de todas las proteínas, por tanto contar con una herramienta que ayude a focalizar esfuerzos experimentales en pares de moléculas de pequeñas enzimas parece ser biológicamente relevantes.*

- *Los principales retos son que la representación numérica de caeda enzima que máximamente informativa para la predicción rio abajo, para ser lo más ampliamente posible, estas representaciones deben basarse de manera única en la secuencia primaria de las enzimas sin información adicional. Otro reto es que las bases de datos públicas de enzimas, solamente enlista instancias positivas, es decir, moleculas con las cuales las enzimas despliegan actividades medibes [59].*

2.2. About the ML techniques

Nota 7

- *Para entrenar un modelo de predicción, es necesario idear una estrategia automatizada para obtener instancias adecuadas de enzimas y pequeñas moléculas que sean negativas y no se unan. Los enfoques de aprendizaje automático existentes para predecir pares enzima-sustrato fueron desarrollados específicamente para familias de enzimas pequeñas para las cuales existen conjuntos de datos de entrenamiento inusualmente completos [59]-[64].*

- *Mou et al.[60] desarrollaron modelos para predecir los sustratos de las nitrilasas bacterianas, utilizando características de entrada basadas en las estructuras tridimensionales y sitios activos de las enzimas. Entrenaron varios modelos de aprendizaje automático basados en evidencia experimental para todas las posibles combinaciones de enzima y molécula pequeña dentro del alcance de predicción de los modelos (N = 240).*

- *Yang et al.[61] predijeron el alcance de sustratos de las glicosiltransferasas de plantas entre un conjunto predefinido de moléculas pequeñas. Entrenaron un modelo basado en árboles de decisión con un conjunto de datos que cubría casi todas las posibles combinaciones de enzimas y moléculas pequeñas relevantes.*

- *Pertusi et al.[59] entrenaron cuatro máquinas de vectores de soporte (SVM) diferentes, cada una para una enzima específica. Como características de entrada, sus modelos solo utilizan información sobre los sustratos (potenciales), así como no sustratos extraídos manualmente de la literatura; no se utilizó información explícita sobre las enzimas.*

- *Roettig et al.[62] y Chevrette et al.[63] predijeron los alcances de sustratos de familias de enzimas pequeñas, entrenando modelos de aprendizaje automático con información estructural relacionada con los sitios activos de las enzimas.*

- *Visani et al.[65] implementaron un modelo de aprendizaje automático general para predecir clases EC adecuadas para un sustrato dado. Para entrenar este modelo, se utilizaron todas las clases EC que no están asociadas con un sustrato específico como puntos de datos negativos, lo que resultó en una baja proporción promedio de positivos a negativos de 0.0032. Visani et al. no utilizaron información sobre las enzimas más allá de la clase EC como entrada del modelo, y por lo tanto, el modelo no puede distinguir entre diferentes enzimas asignadas a la misma clase EC.*

2.3. About the approuch of existing models

Nota 8

- *Todos estos modelos no pueden aplicarse a enzimas individuales o intentan predecir sustratos solo para una enzima o familia de enzimas.*

- *Aquellos modelos que hacen predicciones para enzimas específicas se basan en datos de entrenamiento experimentales muy densos, es decir, resultados experimentales para todas o casi todas las posibles combinaciones de enzima-sustrato.*

- *Para la gran mayoría de familias de enzimas, no está disponible un conjunto de datos de entrenamiento tan extenso.*

- *Hasta el momento, no ha habido intentos publicados de formular y entrenar un modelo general que pueda aplicarse para predecir sustratos para enzimas específicas en familias de enzimas ampliamente diferentes.*

- *Los modelos de aprendizaje profundo se han utilizado para predecir funciones de enzimas ya sea prediciendo su asignación a clases EC20–22, o prediciendo dominios funcionales dentro de la secuencia de proteínas[69]*

- *Predecir directamente sustratos específicos para enzimas representa un paso importante más allá de esos métodos previos y puede ayudar a predecir la función de la enzima de manera más específica y precisa.*

- *Los enfoques de vanguardia en este dominio son basados en características, es decir, se utilizan representaciones numéricas de la proteína y la molécula del sustrato como entrada para modelos de aprendizaje automático[71]-[75].*

- *Como descripciones numéricas de la molécula del sustrato, estos enfoques utilizan representaciones SMILES30, huellas digitales expertas[77] o huellas digitales creadas con redes neuronales gráficas[78, 79].*

- *Las proteínas suelen codificarse numéricamente mediante representaciones basadas en el aprendizaje profundo de las secuencias de aminoácidos[80]-[82].*

2.4. About the work in this article

Nota 9

- *In this work, we go beyond the current state-of-the-art by creating maximally informative protein representations, using a customized, task-specific version of the ESM-1b transformer model [80].*

- *The model contains an extra 1280-dimensional token, which was trained end-to-end to store enzyme-related information salient to the downstream prediction task.*
- *This general approach was first introduced for natural language processing37, but has not yet been applied to protein feature prediction.*
- *We created negative training examples using data augmentation, by randomly sampling small molecules similar to the substrates in experimentally confirmed enzyme-substrate pairs.*
- *we sampled all negative data points from a limited set of metabolites, the set of 1400 substrates that occur among all experimentally confirmed enzyme-substrate pairs of our dataset.*
- *Thus, we do not sample from the space of all possible alternative reactants similar to the true substrates, but only consider small molecules likely to occur in at least some biological cells.*
- *While many enzymes are rather promiscuous 2-4, it is likely that most of the potential secondary substrates are not contained in this restricted set for any given enzyme, and hence the chance of sampling false negative data points was likely small.*
- *We numerically represented all small molecules with task-specific fingerprints that we created with graph neural networks (GNNs)[84, 85, 86].*
- *A gradient-boosted decision tree model was trained on the combined protein and small molecule representations for a high-quality dataset with 18,000 very diverse, experimentally confirmed positive enzyme-substrate pairs.*
- *The resulting Enzyme Substrate Prediction model-ESP-achieves high prediction accuracy for those 1400 substrates that have been part of our training set and outperforms previously published enzyme family-specific prediction models.*

2.5. About the obtained results

Nota 10

- *A data set with experimentally confirmed enzyme-substrate pairs using the GO annotation database for Uniprot IDs[**?**] was created.*
- *For training the machine learning models, they extracted 18,351 enzyme-substrate pairs with experimental evidence for binding, comprised of 12,156 unique enzymes and 1379 unique metabolites.*
- *Also extracted 274,030 enzyme-substrate pairs with phylogenetically inferred evidence, i.e., these enzymes are evolutionarily closely related to enzymes associated with the same reactions. These guilt by associations assignments are much less reliable than direct experimental evidence, and we only used them during pre-training to create task-specific enzyme representations-numerical vectors aimed at capturing information relevant to the prediction task from the enzyme amino acid sequences.*

- *The validations demonstrate that using phylogenetically inferred functions for the construction of appropriate enzyme representations has a positive effect on the prediction of experimentally confirmed enzyme-substrate pairs.*
- *There is no systematic information on negative enzyme-small molecule pairs, i.e., pairs where the molecule is not a substrate of the enzyme.*
- *It was hypothesized that such negative data points could be created artificially through random sampling, which is a common strategy in classification tasks that lack negative training data [83].*
- *Only were considered small molecules included among the experimentally confirmed enzyme-substrate pairs in our dataset.*
- *Among such a limited and biased subset, enzymes are quite specific catalyst, and therefore most of the potential secondary substrates are not included for the majority of enzymes.*
- *It was assumed that the frequency of incorrectly created negative labels is sufficiently low o not adversely affect model performance. This assumption was confirmed by the high model accuracy on independent test data.*
- *To select putatively non-binding small molecules that are structurally similar to the known substrates, it was used a similarity score based on molecular fingerprints with values ranging from 0 (no similarity) to 1 (identity; see Methods, sampling negative data points).*
- *For every positive enzyme-substrate pair, it was sampled three molecules with similarity scores between 0.75 and 0.95 to the actual substrate of the enzyme, and used them to construct negative enzyme-molecule pairs.*
- *it was opted for creating more negative data points than positive data points, as this not only provided us with more data, but it also more closely reflects the true distribution of positive and negative data points compared to a balanced distribution.*
- *Our final dataset comprises 69,365 entries. We split this data into a training set (*80 %*) and a test set (*20 %*).*
- *In many machine learning domains, it is standard practice to split the data into training and test set completely at random. However, when dealing with protein sequences, this strategy often leads to test sets with amino acid sequences that are almost identical to those of proteins in the training set.*
- *It is common practice to split such datasets into training, validation, and test sets based on protein sequences similarities [***?***].*
- *It was made sure that no enzyme in the test set has a sequence identity higher than* 80 % *compared to any enzyme in the training set. To show that despite this sequence-based partitioning, enzymes from the training and test sets follow the same distribution, we*

used dimensionality reduction to map all enzymes to a two-dimensional subspace and plotted the corresponding data points.

- *To evaluate how well the final model performs for different levels of enzyme similarities, it was divided the test set further into three subsets with maximal sequence identities between* 0–40 %*,* 40–60 %*, and* 60–80 % *compared to all enzymes in the training set.*

- *For the numerical encoding, one classiffies bond types and calculates feature vectors with information about every atom (types, masses, valences, atomic numbers, atom charges, and number of attached hydrogen atoms)[77]. Afterwards, these identifiers are updated for a fixed number of steps by iteratively applying predefined functions to summarize aspects of neighboring atoms and bonds.*

- *After the iteration process, all identiffiers are converted into a single binary vector with structural information about the molecule. The number of iterations and the dimension of the fingerprint can be chosen freely, in this case they were set to the default values of 3 and 1024, respectively, also was created 512- and 2048-dimensional ECFPs, but these led to slightly inferior predictions. Using ECFPs can lead to identical representations for structurally very similar molecules, e.g., for some molecules that differ only by the length of a chain of carbon atoms. In our dataset, 182 out of 1379 different molecules shared an identical fingerprint with a structurally similar molecule.*

- *As an alternative to expert-crafted fingerprints such as ECFPs, neural networks can be used to learn how to map graph representations of small molecules to numerical vectors, such networks are referred to as graph neural networks (GNNs)[84, 85, 86]*

- *We trained a GNN for the binary task of predicting if a small molecule is a substrate for a given enzyme. While training for this task, the GNN is challenged to store all information about the small molecule that is relevant for solving the prediction task in a single numerical vector.*

- *After training, were extracted these 100-dimensional task-specific vectors for all small molecules in our dataset. It has been observed that pre-training GNNs for a related task can signiffcantly improve model performance[87, 88].*

- *Thus, first were pre-trained the GNN for the related task of predicting the Michaelis constants KM of enzyme-substrate pairs (see Methods, pre-training indeed improved prediction performance significantly. In contrast to ECFPs, GNN-generated fingerprints lead to much fewer cases of identical representations for different molecules.*

- *In the dataset, identical fingerprints occurred for 42 out of 1379 molecules training of $ESM-1b$,* 15 % *of the amino acids in a protein's sequence are randomly masked and the model is trained to predict the identity of the masked amino acids. This training procedure forces the model to store both local and global information about the protein sequence in one 1280-dimensional representation vector for each individual amino acid.*

- *In order to create a single fixed-length numerical representation of the whole protein, one typically calculates the element-wise mean across all amino acid representations[80, 81, 91]. it was refered to these protein representations as ESM-1b vectors*

  Nota 11 *Simply taking the element-wise mean results in information loss and does not consider the task for which the representations shall be used, which can lead to subpar performance. To overcome these issues, we created task-specific enzyme representations optimized for the prediction of enzyme-substrate pairs.*

  We slightly modified the architecture of the ESM-1b model, adding one additional 1280-dimensional token to represent the complete enzyme, intended to capture information salient to the downstream prediction task. The extra token is not adding input information to the model, but it allows an easier extraction of enzyme information from the trained model. This whole-enzyme representation was updated in the same way as the regular ESM-1b amino acid representations.

- *After a predefined number of update steps, the enzyme representation was concatenated with the small molecule ECFP-vector. The combined vector was used as the input for a fully connected neural network (FCNN), which was then trained end-to-end to predict whether the small molecule is a substrate for the enzyme.*

- *This approach facilitates the construction of a single, optimized, task-specific representation. The ESM-1b model contains many parameters and thus requires substantial training data. Therefore, in the pre-training that produces the task-specific enzyme representations, we added phylogenetically inferred evidence to our training set; this resulted in a total of 287,000 data points used for training the task-specific enzyme representation.*

- *After training, we used the network to extract the 1280-dimensional task-specific representations for all enzymes in our dataset, these representations are called ESM-1bts vectors.*

- *The ESM-1b model is a state-of-the-art transformer network[89], trained with 27 million proteins from the UniRef dataset[**?**] in a self-supervised fashion[80]. This model takes an amino acid sequence as its input and puts out a numerical representation of the sequence; these representations are often referred to as protein embeddings.*

- *we estimated prediction quality on our test set when using machine learning models with each of the four combinations of enzyme and small molecule representations. In each case, we concatenated one of the two 1280-dimensional enzyme representations with one of the two small molecule representations to create a single input vector for every enzyme- small molecule pair.*

- *We used these inputs to train gradient-boosted decision tree models[**?**] for the binary classification task of predicting whether the small molecule is a substrate for the enzyme.*

- *We performed hyperparameter optimizations for all four models, including the parameters learning rate, depth of trees, number of iterations, and regularization coefficients.*

- *For this, we performed a random grid search with a 5-fold cross-validation (CV) on the training set. To challenge the model to learn to predict the substrate scope of enzymes not included in the training data, we made sure that each enzyme occurred in only one of the five subsets used for cross-validation.*

- *To account for the higher number of negative compared to positive training data, we also included a weight parameter that lowered the influence of the negative data points.*

- *After hyperparameter optimization, the models were trained with the best set of hyperparameters on the whole training set and were validated on our independent test set, which had not been used for model training or hyperparameter selection. It is noteworthy that for some input combinations, the accuracies on the test set are higher than the accuracies achieved during cross-validation.*

- *This improved performance on the test set may result from the fact that before validation on the test set, models are trained with approximately 11,000 more samples than before each cross-validation; the number of training samples has a substantial influence on model performance.*

- *To compare the gradient boosting model to alternative machine learning models, we also trained a logistic regression model and a random forest model for the task of predicting enzyme-substrate pairs from the combined ESM-1bts and GNN vectors. However, these models performed worse compared to the gradient boosting model.*

- *To investigate the importance of the added token for the observed superior performance, we alternatively re-trained the ESM-1b without such an extra token.*

- *Good predictions even for unseen enzymes It appears likely that prediction quality is best for enzymes that are highly similar to enzymes in the training set, and decreases for enzymes that are increasingly dissimilar to the enzymes used for training.*

- *How strong is that dependence? To answer this question, we first calculated the maximal enzyme sequence identity compared to the enzymes in the training set for all 2291 enzymes in the test set. Next, we split the test set into three subgroups: data points with enzymes with a maximal sequence identity to training data between 0 and* 40 %*, between* 40 % *and* 60 % *and between* 60 % *and* 80 %*.*

- *It was shown model performance is highest for enzymes that are similar to proteins in the training set. Similarly, it appears likely that the model performs better when making predictions for small molecules that are also in the training set. To test this hypothesis, we divided the test set into data points with small molecules that occurred in the training set (N = 13,459) and those with small molecules that did not occur in the training set (N = 530).*

- *The ESP model does not perform well for data points with small molecules not present in the training set. When considering only enzyme-small molecules pairs with small molecules not represented in the training set and an enzyme sequence identity level of* 0–40 % *compared to the training data, ESP achieves an accuracy of* 71 %*, ROC-*

AUC score 0.59, and MCC 0.15. At an enzyme sequence identity level of 40–60 %*, accuracy improves to* 83 %*, with ROC-AUC score 0.78, and MCC 0.25 for unseen small molecules.*

- *For those test data points with small molecules not present in the training set, we wondered if a high similarity of the small molecule compared to at least one substrate in the training set leads to improved predictions, analogous to what we observed for enzymes with higher sequence identities.*
- *For each small molecules not present in the training set, we calculated the maximal pairwise similarity score compared to all substrates in the training set.*
- *we conclude that ESP only achieves high accuracies for new enzyme-small molecule pairs if the small molecule was present among the 1 400 substrates of our training set.*
- *How many training data points with identical substrates are needed to achieve good model performance? For every small molecule in the test set, we counted how many times the same molecule occurs as an experimentally confirmed substrate in the training set.*
- *Model performance increases with increased training set size The previous subsections suggest that a bigger training set with a more diverse set of enzymes and small molecules should lead to improved performance. However, using more data does not guarantee an improved model performance.*
- *To test how our model performs with different amounts of training data and to analyze if more data is expected to lead to higher generalizability, we trained the gradient boosting model with different training set sizes, ranging from* 30 % *to* 100 % *of the available training data.*
- *Internally, our trained classification model does not simply output the positive or negative class as a prediction. Mou et al.[60] trained four different machine learning models (logistic regression, random forest, gradient-boosted decision trees, and support vector machines) to predict substrates of bacterial nitrilases.*
- *For model training and validation, they used a dataset with all possible combinations of 12 enzymes and 20 small molecules (N = 240), randomly split into* 80 % *training data and* 20 % *test data. Instead, it outputs a prediction score between 0 and 1,Yang et al.15 published a decision tree-based model, GT-Predict, for predicting the substrate scope of glycosyltransferases of plants.All software was coded in Python57. We implemented and trained the neural networks using the deep learning library PyTorch58. We fitted the gradient boosting models using the library XGBoost50.*

Nota 12 *Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction*

- *Deep learning ha sido aplicado y mostrado un gran desempeño in el modelado de espacios químicos, expresión genética, parámetros relacionados a enzimas tales como afinidad enzimatica y números EC*

- *developed a deep learning approach (DLKcat) that uses substrate structures and protein sequences as inputs, and demonstrated its capability for the large-scale prediction of kcat val- ues for various organisms*
- *The deep learning approach DLKcat was developed by combin- ing a graph neural network (GNN) for substrates and a convolutional neural network (CNN) for proteins*

3. Articulo:Advances in Machine Learning for Directed Evolution

En este documento, se presentan algunas respuestas a preguntas y fragmentos de texto previamente proporcionados.

La evolución dirigida, que implica rondas iterativas de diversificación genética y cribado o selección fenotípica, ha surgido como una herramienta especialmente poderosa para moldear las propiedades de las proteínas en el laboratorio. La actividad específica, la estabilidad, el alcance de sustratos y la estereoselectividad de las enzimas se pueden optimizar utilizando esta técnica.

Las enzimas ofrecen soluciones a los problemas químicos más desafiantes de la vida. La capacidad de las enzimas para catalizar reacciones químicas de manera eficiente y selectiva las hace útiles no solo para los organismos anfitriones, sino también para una multitud de aplicaciones que los humanos han ideado. Como catalizadores verdes, económicos y eficientes, las enzimas han sido adoptadas por industrias que van desde la farmacéutica hasta productos de consumo, materiales, alimentos y combustibles, y se espera que su importancia siga creciendo [1–3].

La secuencia de una proteína codifica su función (*aptitud*), y la relación entre ambas se conceptualiza a menudo como una superficie en un espacio de alta dimensionalidad llamado el paisaje de aptitud de la proteína [6,7]. Nuevas proteínas se desarrollan buscando en este paisaje, comúnmente a través de un proceso de evolución dirigida [7]. La evolución dirigida avanza sometiendo una proteína que posee al menos una pequeña cantidad de la función deseada a rondas iterativas de mutagénesis y cribado, utilizando la mejor variante en cada ronda como punto de partida para la siguiente hasta que se logre el objetivo funcional (Figura 1A). A pesar de su éxito, la evolución dirigida depende de una extensa caracterización en el laboratorio, lo que constituye un cuello de botella para el desarrollo de muchas proteínas diseñadas, donde el cribado de más de unas pocas cientos o miles de variantes puede ser altamente intensivo en recursos.

Cuando se aplica a la evolución dirigida, el aprendizaje automático (ML, por sus siglas en inglés) hasta ahora se ha planteado en gran medida como un problema supervisado; es decir, dado un conjunto de secuencias de proteínas con etiquetas asociadas (por ejemplo, actividad catalítica, estabilidad, etc.), la tarea es aprender una función que pueda predecir la etiqueta de secuencias previamente no vistas (Figura 1B). Utilizando esta función, se pueden evaluar computacionalmente grandes cantidades de proteínas durante cada ciclo de evolución, lo que permite explorar mucho más a fondo el paisaje de aptitud de la proteína de lo que se podría lograr solo con cribado en el laboratorio.

Nos centramos en las formas en que los investigadores están aprovechando estrategias de aprendizaje no supervisado, es decir, estrategias que aprenden a partir de secuencias de proteínas no etiquetadas, para superar los desafíos relacionados con la recopilación de grandes conjuntos de datos de secuencias de proteínas y su función.

Comenzamos discutiendo contribuciones destacadas en el uso de secuencias de proteínas para reducir o eliminar la cantidad de datos de entrenamiento etiquetados necesarios en el aprendizaje automático supervisado. Luego destacamos trabajos que demuestran cómo los modelos entrenados solo con datos no etiquetados pueden utilizarse para generar nueva diversidad de secuencias con propiedades deseadas, así como para navegar en paisajes de aptitud de proteínas extremadamente grandes. Nuestro objetivo es hacer que esto sea accesible para la audiencia de la ingeniería de proteínas y, por lo tanto, evitamos una explicación extensa de las arquitecturas de los modelos, algoritmos y estrategias de aprendizaje que sustentan los ejemplos presentados.

Una estrategia de optimización de larga data para guiar la recopilación de datos costosos es el aprendizaje activo.

En este enfoque, un investigador entrena iterativamente un modelo con una pequeña cantidad de datos etiquados, y luego utiliza ese modelo para identificar nuevos puntos de datos para recopilar, que serían informativos y mejorarían el rendimiento del modelo. Los procesos gaussianos, que modelan su propia incertidumbre, se encuentran entre los modelos más populares para este enfoque y se han utilizado, por ejemplo, en la evolución dirigida de citocromos P450 más termoestables y variantes de channelrhodopsin para aplicaciones de optogenética.

El preentrenamiento no supervisado se basa en la suposición de que cada proteína secuenciada sigue un conjunto de reglas biofísicas y evolutivas que permiten que esa proteína sea producida y lleve a cabo una función biológica. Al entrenar modelos, que a menudo se adaptan de procesamiento de lenguaje natural (NLP) [22], en secuencias de proteínas no etiquetadas, se pueden aprender las restricciones de secuencia que resultan de estas reglas.

Los investigadores se han centrado en aumentar pequeños conjuntos de datos etiquetados con información extraída de grandes conjuntos de datos no etiquetados, una estrategia generalmente conocida como aprendizaje semi-supervisado. Cuando se aplica al campo de la ingeniería de proteínas, el aprendizaje semi-supervisado consta de una fase de aprendizaje no supervisado, a menudo denominada "preentrenamiento no supervisado.º "preentrenamiento auto-supervisado"debido a los procedimientos de entrenamiento de modelos específicos generalmente empleados, seguida de una fase de aprendizaje supervisado.

Una codificación de proteína es una representación vectorial de una secuencia de proteína necesaria para su uso en algoritmos de aprendizaje automático. Las codificaciones más simples resultan en una representación dispersa del espacio de secuencias, lo que proporciona información limitada sobre las relaciones entre secuencias y, por lo tanto, dificulta el aprendizaje [8,12]. Los embeddings de proteínas obtenidos de modelos no supervisados capturan la información aprendida durante el preentrenamiento y definen las relaciones entre las proteínas en el contexto de las restricciones de secuencia aprendidas: las secuencias similares se encontrarán más cerca unas de otras en el espacio de embeddings y, por ejemplo, se pueden inferir propiedades similares por un modelo supervisado posterior. De esta manera, los embeddings de proteínas aprendidos permiten que la información contenida en secuencias no etiquetadas se transfiera a una tarea supervisada posterior (Figura 2C-D), en principio

reduciendo la cantidad de datos etiquetados necesarios en comparación con estrategias de codificación menos informativas.

Aún queda mucho por explorar en el aprendizaje semi-supervisado en la ingeniería de proteínas. Por ejemplo, hasta ahora, las arquitecturas de modelos no supervisados utilizadas para el preentrenamiento se han adaptado principalmente de NLP. Si bien existen pruebas que sugieren que modelos de NLP más grandes entrenados en secuencias más diversas pueden mejorar los resultados de la ingeniería [23,30], también hay evidencia de que modelos mucho más pequeños con objetivos de aprendizaje más específicos para proteínas pueden lograr un rendimiento predictivo competitivo en tareas supervisadas posteriores [38]. Tampoco siempre está claro cuándo las estrategias semi-supervisadas serán superiores a las completamente supervisadas.

En medio de la creciente preocupación en la comunidad de procesamiento de lenguaje natural (NLP) acerca de los costos monetarios y energéticos de entrenar modelos de lenguaje grandes [40], el desarrollo adicional de modelos no supervisados más pequeños y la identificación de situaciones en las que el aprendizaje semi-supervisado es beneficioso son áreas importantes para investigaciones futuras.

Finalmente, también es importante señalar que, dada la inmensa magnitud del espacio de proteínas posible, el aprendizaje automático para la evolución dirigida siempre se realizará en un entorno de N relativamente bajo y nunca podrá enumerar por completo el espacio de proteínas posibles, siendo necesaria cierta cantidad de iteración. Con esta consideración, la pregunta de cómo combinar enfoques de preentrenamiento no supervisado con el aprendizaje activo se vuelve importante. Una estrategia recientemente descrita por Hie et al., que combina procesos gaussianos con embeddings de proteínas aprendidos, es un enfoque posible, al igual que una serie de algoritmos incipientes para la optimización en espacios combinatorios grandes [41–50]. En resumen, distinguir las mejores arquitecturas de modelos no supervisados y estrategias de iteración requerirá una extensa evaluación comparativa en conjuntos de datos recopilados para diferentes tareas de ingeniería de proteínas, como los proporcionados por Rao et al. [35].

Dado que las mutaciones con frecuencia llevan a la pérdida de la función, la capacidad de evitar de antemano variantes no funcionales ahorraría recursos de cribado y mejoraría significativamente la eficiencia de la evolución dirigida. Entre las aplicaciones más interesantes del aprendizaje no supervisado se encuentra la predicción "zero-shot", donde se utilizan modelos completamente no supervisados para predecir si una proteína funciona sin necesidad de un entrenamiento supervisado adicional con datos etiquetados [32,51,52]. Normalmente, esto se logra mediante el uso de un modelo generativo, que es un modelo entrenado con datos de secuencias de proteínas no etiquetados y que aprende una representación de la distribución de secuencias de proteínas permitidas (Figura 3A). Estos modelos se utilizan para consultar la probabilidad de que una nueva secuencia de proteína haya sido generada a partir de la distribución aprendida de secuencias subyacentes (Figura 3B). Si esta secuencia tiene una alta probabilidad de pertenecer a la distribución aprendida, entonces es más probable que sea una proteína funcional, y viceversa. En muchos aspectos, este enfoque es similar a la estrategia de puntuación de mutantes de proteínas basada en la conservación evolutiva, como el uso de matrices BLOSUM. Los modelos de aprendizaje automático son más capaces de capturar las interacciones epistáticas de orden superior que se cree que impregnan la evolución de las proteínas.

Mientras que entrenar un predictor de predicción "zero-shot.[en] alineamientos múltiples (MSAs) permite que un modelo generativo aprenda una representación rica de secuencias estrechamente relacionadas con un objetivo de ingeniería, si hay pocas secuencias homólogas al objetivo, entonces la distribución aprendida puede ser demasiado estrecha o dispersa para ser utilizada de manera confiable para la predicción "zero-shot". De hecho, DeepSequence, utilizado por Riesselman et al., tuvo dificultades cuando se aplicó a proteínas para las cuales se encontraron pocas secuencias homólogas. Debido a que los modelos entrenados en bases de datos globales pueden aprender una representación más general de secuencias de proteínas, pueden ser más efectivos en tales casos. Madani et al., por ejemplo, demostraron que un gran modelo de procesamiento de lenguaje natural (NLP) entrenado en cientos de millones de secuencias de diversas familias se podía utilizar como un predictor "zero-shot"sin necesidad de recopilar secuencias de proteínas estrechamente relacionadas con el objetivo [32]. Por supuesto, todos estos estudios asumen que la aptitud del objetivo de un experimento de evolución dirigida se correlaciona bien con la aptitud optimizada evolutivamente, pero esto no siempre será el caso.

Desde la perspectiva de la evolución dirigida, un uso ideal de los modelos generativos sería identificar variantes mejoradas entre un gran número de posibilidades. Desafortunadamente, debido a que la distribución aprendida de secuencias no modela explícitamente en qué medida una proteína puede ser apta, sino solo una sensación de similitud con secuencias en las que el modelo fue entrenado, no hay expectativa de que una secuencia seleccionada sea mejorada en aptitud. Sin embargo, las estrategias recientemente propuestas que acoplan un modelo predictivo, que puede identificar variantes aptas pero requiere una costosa predicción de la aptitud de todos los candidatos, con un modelo generativo, que puede proponer variantes funcionales a partir de grandes grupos de candidatos, combinan las fortalezas de ambos y potencialmente permiten la optimización en vastos paisajes de aptitud de proteínas sin una caracterización computacional extensa [42,48–50,59]. Aunque los detalles varían, el enfoque de alto nivel de tales métodos es primero utilizar el modelo generativo para proponer un conjunto de secuencias que el modelo predictivo evaluará. Luego, se utilizan las secuencias con la aptitud predicha más alta para actualizar el modelo generativo (y posiblemente el modelo predictivo) con el fin de proponer variantes de mayor aptitud. Al repetir este ciclo, el modelo generativo propone proteínas cada vez más aptas, optimizando así la aptitud de las proteínas. Hasta ahora, tales estrategias son principalmente teóricas y deberán ser validadas a fondo mediante experimentación en el laboratorio, aunque hay algunos ejemplos de aplicación exitosa a la ingeniería de sistemas biológicos.

Al mover las costosas pruebas experimentales in silico, el aprendizaje automático (ML) amplía en gran medida nuestra capacidad para explorar el espacio de secuencias de proteínas. Si bien hasta ahora, el ML se ha planteado principalmente como un problema supervisado cuando se aplica a la evolución dirigida, también ha habido una expansión significativa en las estrategias de ML no supervisadas. Estas aproximaciones no supervisadas se pueden utilizar para limitar o eliminar la caracterización experimental requerida de proteínas, ayudar en la navegación del espacio de secuencias combinatorias y generar nueva diversidad de secuencias de proteínas, todo lo cual puede mejorar la eficiencia de las campañas de evolución dirigida. Sin embargo, el ML para la evolución dirigida sigue siendo un campo relativamente joven con mucho espacio para avances continuos. En particular, la disminución continua en el costo y el tiempo de síntesis y secuenciación de genes, así como el aumento de la potencia

computacional, harán que la aplicación de métodos de ML en el laboratorio sea más factible y permitirán la expansión tanto de las bases de datos de secuencias como de secuencias-función. A medida que la disponibilidad de datos crezca, la colaboración continua y mejorada entre científicos de ML y ingenieros de proteínas será fundamental para desarrollar estrategias de ML experimentalmente viables que avancen en el campo y fomenten una adopción más generalizada de la tecnología.

4. Referencias

Referencias

[1] Mazurenko, S., Prokop, Z., and Damborsky, J. (2019). Machine learning in enzyme engineering. ACS Catalysis, 10(2), 1210-1223.

[2] Yang, Y.; Niroula, A.; Shen, B.; Vihinen, M. PON-Sol: Prediction of Effects of Amino Acid Substitutions on Protein Solubility. Bioinformatics 2016, 32, 2032!2034.

[3] Folkman, L.; Stantic, B.; Sattar, A.; Zhou, Y. EASE-MM: Sequence-Based Prediction of Mutation-Induced Stability Changes with Feature-Based Multiple Models. J. Mol. Biol. 2016, 428, 1394! 1405.

[4] Teng, S.; Srivastava, A. K.; Wang, L. Sequence Feature-Based Prediction of Protein Stability Changes upon Amino Acid Substitutions. BMC Genomics 2010, 11, S5.

[5] Huang, L.; Gromiha, M. M.; Ho, S. iPTREE-STAB: Interpretable Decision Tree Based Method for Predicting Protein Stability Changes Upon Mutations. Bioinformatics 2007, 23, 1292! 1293.

[6] Koskinen, P.; Toronen, P.; Nokso-Koivisto, J.; Holm, L. PANNZER: High-Throughput Functional Annotation of Uncharac- terized Proteins in an Error-Prone Environment. Bioinformatics 2015, 31, 1544!1552.

[7] De Ferrari, L.; Mitchell, J. B. From Sequence to Enzyme Mechanism Using Multi-Label Machine Learning. BMC Bioinf. 2014, 15, 150.

[8] Falda, M.; Toppo, S.; Pescarolo, A.; Lavezzo, E.; Di Camillo, B.; Facchinetti, A.; Cilia, E.; Velasco, R.; Fontana, P. Argot2: A Large Scale Function Prediction Tool Relying on Semantic Similarity of Weighted Gene Ontology Terms. BMC Bioinf. 2012, 13, S14.

[9] Cozzetto, D.; Buchan, D. W.; Bryson, K.; Jones, D. T. Protein Function Prediction by Massive Integration of Evolutionary Analyses and Multiple Data Sources. BMC Bioinf. 2013, 14, S1.

[10] Kulski, J. Next Generation Sequencing: Advances, Applications and Challenges; InTechOpen: London, 2016.

[11] Straiton, J.; Free, T.; Sawyer, A.; Martin, J. From Sanger Sequencing to Genome Databases and Beyond. BioTechniques 2019, 66, 60-63.

[12] Ardui, S.; Ameur, A.; Vermeesch, J. R.; Hestand, M. S. Single Molecule Real-Time (SMRT) Sequencing Comes of Age: Applications and Utilities for Medical Diagnostics. Nucleic Acids Res. 2018, 46, 2159-2168.

[13] Kono, N., and Arakawa, K. (2019). Nanopore sequencing: Review of potential applications in functional genomics. Development, growth and differentiation, 61(5), 316-326.

[14] Bunzel, H. A., Garrabou, X., Pott, M., and Hilvert, D. (2018). Speeding up enzyme discovery and engineering with ultrahigh-throughput methods. Current opinion in structural biology, 48, 149-156.

[15] Wrenbeck, E. E., Faber, M. S., and Whitehead, T. A. (2017). Deep sequencing methods for protein engineering and design. Current opinion in structural biology, 45, 36-44.

[16] Fowler, D. M., and Fields, S. (2014). Deep mutational scanning: a new style of protein science. Nature methods, 11(8), 801-807.

[17] Gupta, K., and Varadarajan, R. (2018). Insights into protein structure, stability and function from saturation mutagenesis. Current opinion in structural biology, 50, 117-125.

[18] UniProt Consortium. UniProt: A Worldwide Hub of Protein Knowledge. Nucleic Acids Res. 2018, 47, D506-D515.

[19] Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Zidek, A.; Nelson, A.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Jones, D. T.; Silver, D.; Kavukcuoglu, K.; Hassabis, D.; Senior, A. W. De Novo Structure Prediction with Deeplearning Based Scoring. In Thirteenth Critical Assessment of Techniques for Protein Structure Prediction Abstracts; 2018; pp 11-12.

[20] Kinch, L. N.; Shi, S.; Cheng, H.; Cong, Q.; Pei, J.; Mariani, V.; Schwede, T.; Grishin, N. V. CASP9 Target Classification. Proteins: Struct., Funct., Genet. 2011, 79, 21-36.

[21] Shehu, A.; Barbará, D.; Molloy, K. A Survey of ComputationalMethods for Protein Function Prediction. In Big Data Analytics in Genomics; Wong, K. C., Ed.; Springer: Cham, 2016; pp 225-298.

[22] Zhang, C.; Freddolino, P. L.; Zhang, Y. COFACTOR: Improved Protein Function Prediction by Combining Structure, Sequence and Protein-Protein Interaction Information. Nucleic Acids Res. 2017, 45, W291-W299.

[23] Kumar, N.; Skolnick, J. EFICAz2. 5: Application of a High-Precision Enzyme Function Predictor to 396 Proteomes. Bioinformatics 2012, 28, 2687-2688.

[24] Li, Y.; Wang, S.; Umarov, R.; Xie, B.; Fan, M.; Li, L.; Gao, X. DEEPre: Sequence-Based Enzyme EC Number Prediction by Deep Learning. Bioinformatics 2018, 34, 760-769.

[25] Yang, M.; Fehl, C.; Lees, K. V.; Lim, E. K.; Offen, W. A.; Davies, G. J.; Bowles, D. J.; Davidson, M. G.; Roberts, S. J.; Davis, B. G. Functional and Informatics Analysis Enables Glycosyltransferase Activity Prediction. Nat. Chem. Biol. 2018, 14, 1109-1117.

[26] Niwa, T.; Ying, B. W.; Saito, K.; Jin, W.; Takada, S.; Ueda, T.; Taguchi, H. Bimodal Protein Solubility Distribution Revealed by an Aggregation Analysis of the Entire Ensemble of Escherichia Coli Proteins. Proc. Natl. Acad. Sci. U. S. A. 2009, 106, 4201-4206.

[27] Klesmith, J. R.; Bacik, J. P.; Wrenbeck, E. E.; Michalczyk, R.; Whitehead, T. A. Trade-Offs Between Enzyme Fitness and Solubility Illuminated by Deep Mutational Scanning. Proc. Natl. Acad. Sci. U. S. A. 2017, 114, 2265-2270.

[28] Ruiz-Blanco, Y. B.; Paz, W.; Green, J.; Marrero-Ponce, Y. ProtDCal: A Program to Compute General-Purpose-Numerical Descriptors for Sequences and 3D-Structures of Proteins. BMC Bioinf. 2015, 16, 162.

[29] Han, X.; Wang, X.; Zhou, K. Develop Machine Learning-Based Regression Predictive Models for Engineering Protein Solubility. Bioinformatics 2019, 35, 4640-4646.

[30] Musil, M.; Konegger, H.; Hon, J.; Bednar, D.; Damborsky, J. Computational Design of Stable and Soluble Biocatalysts. ACS Catal. 2019, 9, 1033-1054.

[31] Li, G.; Dong, Y.; Reetz, M. T. Can Machine Learning Revolutionize Directed Evolution of Selective Enzymes? Adv. Synth. Catal. 2019, 361, 2377-2386.

[32] Wu, Z.; Kan, S. B. J.; Lewis, R. D.; Wittmann, B. J.; Arnold, F. H. Machine Learning-Assisted Directed Protein Evolution with Combinatorial Libraries. Proc. Natl. Acad. Sci. U. S. A. 2019, 116, 8852-8858.

[33] Wolpert, D. H.; Macready, W. G. No Free Lunch Theorems for Optimization. IEEE Trans. Evol. Comput. 1997, 1, 67-82.

[34] Wolpert, D. H. The Lack of a Priori Distinctions between Learning Algorithms. Neural Comput. 1996, 8, 1341-1390.

[35] Walsh, I.; Pollastri, G.; Tosatto, S. C. Correct Machine Learning on Protein Sequences: A Peer-Reviewing Perspective. Briefings Bioinf. 2016, 17, 831-840.

[36] Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y. S. Evaluating Protein Transfer Learning with TAPE. arXiv preprint arXiv:1906.08230, 2019.

[37] Romero, P. A.; Krause, A.; Arnold, F. H. Navigating the Protein Fitness Landscape with Gaussian Processes. Proc. Natl. Acad. Sci. U. S. A. 2013, 110, E193-E201

[38] Eraslan, G.; Avsec, Z; Gagneur, J.; Theis, F. J. Deep Learning: New Computational Modelling Techniques for Genomics. Nat. Rev. Genet. 2019, 20, 389-403.

[39] Repecka, D.; Jauniskis, V.; Karpus, L.; Rembeza, E.; Zrimec, J.; Poviloniene, S.; Rokaitis, I.; Laurynenas, A.; Abuajwa, W.; Savolainen, O.; Meskys, R.; Engqvist, M. K. M.; Zelezniak, A. Expanding Functional Protein Sequence Space Using Generative Adversarial Networks. bioRxiv 2019, DOI: 10.1101/789719.

[40] Riesselman, A. J.; Ingraham, J. B.; Marks, D. S. Deep Generative Models of Genetic Variation Capture the Effects of Mutations. Nat. Methods 2018, 15, 816-822.

[41] Thornton, C.; Hutter, F.; Hoos, H. H.; Leyton-Brown, K. Auto- WEKA: Combined Selection and Hyperparameter Optimization of Classiffication Algorithms. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining; 2013; pp 847-855.

[42] Polikar, R. Ensemble based systems in decision making. IEEE Circuits and systems magazine 2006, 6, 21-45.

[43] Gammerman, A.; Vovk, V. Hedging Predictions in Machine Learning. Comput. J. 2007, 50, 151-163.

[44] Samek, W.; Wiegand, T.; Müller, K. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. ITU Journal: ICT Discoveries 2017, 39-48.

[45] Shrikumar, A.; Greenside, P.; Kundaje, A. Learning Important Features through Propagating Activation differences. In Proceedings of the 34th International Conference on Machine Learning; 2017; Vol. 70, pp 3145-3153.

[46] Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classiffication Models and Saliency Maps. arXiv preprint arXiv:1312.6034 2013.

[47] Brookes, D. H.; Park, H.; Listgarten, J. Conditioning by Adaptive Sampling for Robust Design. In Proceedings of the 36th International Conference on Machine Learning; 2019; Vol. 97, pp 773-782.

[48] Ribeiro, M. T.; Singh, S.; Guestrin, C. "Why Should I Trust You?" Explaining the Predictions of Any Classiffier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016; pp 1135-1144.

[49] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. arXiv preprint arXiv:1312.6199 201

[50] Yu, M. K.; Ma, J.; Fisher, J.; Kreisberg, J. F.; Raphael, B. J.; Ideker, T. Visible Machine Learning for Biomedicine. Cell 2018, 173, 1562-1565.

[51] Copley, S. D. Shining a light on enzyme promiscuity. Curr. Opin. Struct. Biol. 47, 167–175 (2017).

[52] Nobeli, I., Favia, A. D. and Thornton, J. M. Protein promiscuity and its implications for biotechnology. Nat. Biotechnol. 27, 157–167 (2009)

[53] Adrio, J. L. and Demain, A. L. Microbial enzymes: tools for biotechnological processes. Biomolecules 4, 117–139 (2014).

[54] Wang, S. et al. Engineering a synthetic pathway for gentisate in pseudomonas chlororaphis p3. Front. Bioeng. Biotechnol. 8, 1588 (2021).

[55] Wu, M.-C., Law, B., Wilkinson, B. and micklefied, J. Bioengineering natural product biosynthetic pathways for therapeutic applications. Curr. Opin. Biotechnol. 23, 931–940 (2012)

[56] Rembeza, E., Boverio, A., Fraaije, M. W. and Engqvist, M. K. Discovery of two novel oxidases using a high-throughput activity screen. ChemBioChem 23, e202100510 (2022).

[57] Longwell, C. K., Labanieh, L. and Cochran, J. R. High-throughput screening technologies for enzyme engineering. Curr. Opin. Biotechnol. 48, 196–202 (2017).

[58] Black, G. W. et al. A high-throughput screening method for determining the substrate scope of nitrilases. Chem. Commun. 51, 2660–2662 (2015).

[59] Pertusi, D. A. et al. Predicting novel substrates for enzymes with minimal experimental effort with active learning. Metab. Eng. 44,171-181 (2017).

[60] Mou, Z. et al. Machine learning-based prediction of enzyme substrate scope: Application to bacterial nitrilases. Proteins Struct. Funct. Bioinf. 89, 336-347 (2021).

[61] Yang, M. et al. Functional and informatics analysis enables glycosyltransferase activity prediction. Nat. Chem. Biol. 14, 1109–1117 (2018).

[62] Rottig, M., Rausch, C. and Kohlbacher, O. Combining structure and sequence information allows automated prediction of substrate specificities within enzyme families. PLoS Comput. Biol. 6, e1000636 (2010).

[63] Chevrette, M. G., Aicheler, F., Kohlbacher, O., Currie, C. R. and Medema, M. H. Sandpuma: ensemble predictions of nonribosomal peptide chemistry reveal biosynthetic diversity across actinobacteria. Bioinformatics 33, 3202-3210 (2017).

[64] Goldman, S., Das, R., Yang, K. K. and Coley, C. W. Machine learning modeling of family wide enzyme-substrate specificity screens. PLoS Comput. Biol. 18, e1009853 (2022).

[65] Visani, G. M., Hughes, M. C. and Hassoun, S. Enzyme promiscuity prediction using hierarchy-informed multi-label classiffication Bioinformatics 37, 2017-2024 (2021).

[66] Ryu, J. Y., Kim, H. U. and Lee, S. Y. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. PNAS 116, 13996-14001 (2019).

[67] Li, Y. et al. DEEPre: sequence-based enzyme EC number prediction by deep learning. Bioinformatics 34, 760-769 (2017).

[68] Sanderson, T., Bileschi, M. L., Belanger, D. and Colwell, L. J. Proteinfer, deep neural networks for protein functional inference. eLife 12, e80942 (2023).

[69] Bileschi, M. L. et al. Using deep learning to annotate the protein universe. Nat. Biotechnol.https://doi.org/10.1038/s41587-021-01179-w (2022).

[70] Rembeza, E. and Engqvist, M. K. Experimental and computational investigation of enzyme functional annotations uncovers misannotation in the ec 1.1. 3.15 enzyme class. PLoS Comput. Biol. 17, e1009446 (2021).

[71] Ozturk, H., Ozgur, A. and Ozkirimli, E. Deepdta: deep drugtarget binding affinity prediction. Bioinformatics 34, i821-i829 (2018).

[72] Feng, Q., Dueva, E., Cherkasov, A. and Ester, M. Padme: A deep learning-based framework for drug-target interaction prediction. Preprint at https://doi.org/10.48550/arXiv.1807.09741 (2018).

[73] Karimi, M., Wu, D., Wang, Z. and Shen, Y. Deep affinity: interpretable deep learning of compound–protein affinity through UNIFIED recurrent and convolutional neural networks. Bioinformatics 35, 3329-3338 (2019).

[74] Kroll, A., Engqvist, M. K., Heckmann, D. and Lercher, M. J. Deep learning allows genome-scale prediction of michaelis constants from structural features. PLoS Biol. 19, e3001402 (2021).

[75] Li, F. et al. Deep learning-based k cat prediction enables improved enzyme-constrained model reconstruction. Nat. Catal. 5, 662-672 (2022).

[76] Weininger, D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. 28, 31-36 (1988).

[77] Rogers, D. and Hahn, M. Extended-connectivity fingerprints. J. Chem. Inf. Model. 50, 742-754 (2010).

[78] Zhou, J. et al. Graph neural networks: A review of methods and applications. AI Open 1, 57-81 (2020).

[79] Yang, K. et al. Analyzing learned molecular representations for property prediction. J. Chem. Inf. Model. 59, 3370-3388 (2019).

[80] Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. PNAS 118, e2016239118 (2021).

[81] Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. Nat. Methods. 16, 1315-1322 (2019).

[82] Xu, Y. et al. Deep dive into machine learning models for protein engineering. J. Chem. Inf. Model. 60, 2773–2790 (2020).

[83] Bekker, J. and Davis, J. Learning from positive and unlabeled data: A survey. Mach. Learn. 109, 719-760 (2020)

[84] Kearnes, S., McCloskey, K., Berndl, M., Pande, V. and Riley, P. Mole- cular graph convolutions: moving beyond !ngerprints. J. Comput. -Aided Mol. Des. 30, 595–608 (2016).

[85] Duvenaud, D. K. et al. Convolutional networks on graphs for learning molecular fingerprints. In Advances in Neural Information Processing Systems, 2224-2232 (2015).

[86] Zhou, J. et al. Graph neural networks: A review of methods and applications. AI Open 1, 57–81 (2020).

[87] Hu, W. et al. Strategies for pre-training graph neural networks. Preprint at https://doi.org/10.48550/arXiv.1905.12265 (2019).

[88] Capela, F., Nouchi, V., Van Deursen, R., Tetko, I. V. and Godin, G. Multitask learning on graph neural networks applied to molecular property predictions. Preprint at https://doi.org/10.48550/arXiv. 1910.13124 (2019).

[89] Vaswani, A. et al. Attention is all you need. In Advances in neural information processing systems, 5998–6008 (2017).

[90] Suzek, B. E. et al. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinfor- matics 31, 926–932 (2015).

[91] Elnaggar, A. et al. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. IEEE Trans. Pattern Anal. Mach. Intell. PP https://doi. org/10.1109/TPAMI.2021.3095381 (2021).

[92] Wittmann, B. J., Johnston, K. E., Wu, Z., and Arnold, F. H. (2021). Advances in machine learning for directed evolution. Current opinion in structural biology, 69, 11-18.