# EnzyNet: enzyme classification using 3D convolutional neural networks on spatial representation

Afshine Amidi[1,2], Shervine Amidi[2], Dimitrios Vlachakis[3], Vasileios Megalooikonomou[3], Nikos Paragios[2] and Evangelia I. Zacharaki[2,3]

[1] Massachusetts Institute of Technology, Cambridge, MA, USA
[2] Center for Visual Computing, Department of Applied Mathematics, Ecole Centrale de Paris (CentraleSupélec), Châtenay-Malabry, France
[3] MDAKM Group, Department of Computer Engineering and Informatics, University of Patras, Patras, Greece

## ABSTRACT

During the past decade, with the significant progress of computational power as well as ever-rising data availability, deep learning techniques became increasingly popular due to their excellent performance on computer vision problems.
The size of the Protein Data Bank (PDB) has increased more than 15-fold since 1999, which enabled the expansion of models that aim at predicting enzymatic function via their amino acid composition. Amino acid sequence, however, is less conserved in nature than protein structure and therefore considered a less reliable predictor of protein function. This paper presents EnzyNet, a novel 3D convolutional neural networks classifier that predicts the Enzyme Commission number of enzymes based only on their voxel-based spatial structure. The spatial distribution of biochemical properties was also examined as complementary information. The two-layer architecture was investigated on a large dataset of 63,558 enzymes from the PDB and achieved an accuracy of 78.4% by exploiting only the binary representation of the protein shape. Code and datasets are available at https://github.com/shervinea/enzynet.

Submitted 25 July 2017
Accepted 21 April 2018
Published 4 May 2018

Corresponding author
Shervine Amidi,
shervine.amidi@student.ecp.fr

Academic editor
Goo Jun

Additional Information and Declarations can be found on page 15

DOI 10.7717/peerj.4750

**Subjects** Bioinformatics, Computational Biology, Genomics, Computational Science
**Keywords** Deep learning, 3D convolutional neural networks, EnzyNet, Enzyme classification

## INTRODUCTION

The exponential growth of the number of enzymes registered in the Protein Data Bank (PDB) urges a need to propose a fast and reliable procedure to classify every new entry into one of the six standardized enzyme classes denoted by the enzyme classification number (EC): Oxidoreductase (EC1), Transferase (EC2), Hydrolase (EC3), Lyase (EC4), Isomerase (EC5), and Ligase (EC6). Previous work, such as by *Kumar & Choudhary (2012)*, included the use of traditional machine learning techniques requiring a tedious and crucial feature extraction step based on amino acid sequence alignment or using structural descriptors without relying on sequence alignment (*Dobson & Doig, 2005*). A summary of the main alignment-free methods for EC can be found in *Amidi et al. (2016)*,

**How to cite this article** Amidi et al. (2018), EnzyNet: enzyme classification using 3D convolutional neural networks on spatial representation. **PeerJ 6:e4750; DOI 10.7717/peerj.4750**

while a systematic review on the utility and inference of various machine learning techniques for functional characterization is presented in *Sharma & Garg (2014)* and in *Yadav & Tiwari (2015)*. Artificial neural networks (ANNs) in particular have often been used in the past as predictive tools in enzyme activity and catalytic studies. *Baskin, Palyulin & Zefirov (2008)* reviews several aspects in building quantitative structure-activity relationship (QSAR) models using ANNs, while *Szaleniec (2012)* focuses on the application of ANNs in QSAR modeling in enzyme reactivity prediction and relevant methodological issues that arise. In a recent review from *Li, Zhang & Liu (2017)*, the effectiveness of ANNs for catalysis prediction and design of new catalysts is discussed from the perspective of both experiment and theory and common challenges are presented.

A limitation in standard machine learning approaches is that the features have to be predefined, the appropriate choice of features affects the prediction accuracy and there is limited flexibility for model changes or updates (all preprocessing steps have to be repeated). These drawbacks are overcome by deep learning techniques that perform a seamless feature extraction from the input using conventional gradient-based methods or propagation of activation differences if the interpretability of features is essential (*Shrikumar, Greenside & Kundaje, 2017*). With the common availability of data and an ever-increasing computing power, deep learning approaches, such as convolutional neural networks (CNNs), proved to be very efficient and outperformed traditional approaches. Through their successive use of convolutional filters, pooling operations, and fully-connected layers, they mimic how the brain works as they lead the resulting network to focus on features that are crucial in solving supervised tasks. Thanks to their potential to leverage the information contained in considerable amounts of input data, they appear as a way to automatically construct features that we would have had to handcraft ourselves otherwise.

The popularity of deep learning applications in the field of computational biology, bioinformatics, and medical informatics has drastically increased the last years (*Angermueller et al., 2016*; *Jones et al., 2017*; *Min, Lee & Yoon, 2017*). While deep networks, such as sparse auto-encoders, recurrent neural networks (RNN) and long short term memory (LSTM) cells have been exploited for protein structure prediction (*Lyons et al., 2014*; *Heffernan et al., 2015*; *Spencer, Eickholt & Cheng, 2015*; *Nguyen, Shang & Xu, 2014*; *Baldi et al., 2000*; *Sonderby & Winther, 2014*; *Di Lena, Nagata & Baldi, 2012*) and protein classification (*Asgari & Mofrad, 2015*; *Hochreiter, Heusel & Obermayer, 2007*; *Sonderby et al., 2015*), convolutional architectures are not very common and have been used mainly for gene expression regulation (*Alipanahi et al., 2015*; *Lanchantin et al., 2016*; *Zeng et al., 2016*; *Kelley, Snoek & Rinn, 2015*; *Zhou & Troyanskaya, 2015*) and DNA/RNA binding (*Jones et al., 2017*). CNNs have been applied for the prediction of protein properties by *Lin, Lanchantin & Qi (2016)* and in combination with RNNs comprised of LSTM cells by *Li, Zhang & Liu (2017)* for enzyme function prediction based on sequence information. The closest to our work is a 2D CNN ensemble proposed by *Zacharaki (2017)* for EC which exploits mainly, but not purely, the protein structure. It achieved 90.1% accuracy on a benchmark of 44,661 enzymes from the PDB database. The protein structure was represented by multiple 2D feature maps characterizing the backbone conformation (torsion angles) and the (pairwise) amino acids distances. The results showed that the purely structural features (torsion angles) had limited contribution. This may be related to their global nature coming from the 2D representation, which fails to characterize the local 3D shape.

Recently, architectures directly dealing with 3D structures were tested on various datasets. *Maturana & Scherer (2015)* extended the use of CNNs from images to volumes with a 3D CNN approach called VoxNet. They showed that this strategy could be effective in solving tasks that lie in the 3D space by benchmarking it on traditional datasets (from domains such as LiDAR, RGBD, CAD) and subsequently outperforming traditional approaches. The 3D CNN uses as input volumetric data (of size 32 × 32 × 32) containing occupancy information and targets the problem of supervised classification. Other works such as the one of *Hegde & Zadeh (2016)* also take advantage of a 3D representation of the data towards classification. In their work, the authors present two different representations of it: one via voxels, and the other one via projected 2D pixel images. Their ensemble of networks slightly outperforms VoxNet on the ModelNet10 dataset and comes at a higher computational cost.

# MATERIALS AND METHODS

## Dataset

The dataset used in this study has been retrieved from the RCSB PDB (http://www.rcsb.org), which contained 63,558 enzymes as of mid-March of 2017. It has been randomly split in training and testing set with the proportions 80/20%. Also, 20% of the training set has been put aside for validation and was later used for model selection. The details of each of those sets have been summed up in Table 1.

## Representation space and occupancy grid

It has been established (*Illergard, Ardell & Elofsson, 2009*) that structure is far more conserved than sequence in nature. Since evolutionary relationships have been recognized as a confounding factor for EC, choosing an approach that focuses on spatial representation will be particularly robust in that respect. For this reason, we aim at building a model that seamlessly extracts relevant shape features from raw 3D structures. Accordingly, enzymes are represented as a binary volumetric shape with volume elements (voxels) fitted in a cube $V$ of a fixed grid size $l$ with respect to the three dimensions. Continuity between the voxels is achieved by nearest neighbor interpolation, such that for $(i, j, k) \in [\![0,l\!-\!1]\!]^3$ a voxel of vertices

$$(i + \delta x,\ j + \delta y,\ k + \delta z) \quad | \quad (\delta x, \delta y, \delta z) \in \{0, 1\}^3$$

takes the value 1 if the backbone of the enzyme passes through the voxel, and 0 otherwise. Instead of extracting features from this 3D structure and analyzing the calculated multi-channel 2D feature maps (as performed in *Zacharaki, 2017*), we introduce this structure to the CNN, which optimizes its internal parameters (weights and biases of convolutional

Table 1 **Structure of the dataset.**

| EC1 | EC2 | EC3 | EC4 | EC5 | EC6 | Total | |
|---|---|---|---|---|---|---|---|
| 7,096 | 12,081 | 15,290 | 2,875 | 1,703 | 1,632 | 40,677 | Training |
| 1,775 | 2,935 | 3,809 | 743 | 488 | 419 | 10,169 | Validation |
| 2,323 | 3,717 | 4,762 | 858 | 571 | 481 | 12,712 | Testing |
| 11,194 | 18,733 | 23,861 | 4,476 | 2,762 | 2,532 | 63,558 | Total |

filters) using the backpropagation algorithm (*Rumelhart, Hinton & Williams, 1986*). The output of the convolutional filters comprises the feature maps that are selected by the method as best performing for the specific classification task.

To construct this shape representation, some preprocessing steps are necessary. First, protein structure is mapped to a grid of a predefined resolution. The selection of grid resolution determines the level of complexity/scale retained for the enzymatic structures. A full resolution is not preferred due to high data dimensionality and because fine local details are less relevant in characterizing enzymes' chemical reactions. Thus, in order to avoid to get trapped into local minima, side chains are ignored and enzymes are represented exclusively through their "backbone" atoms that are carbon, nitrogen, and calcium.

Additionally, we note that enzymes do not possess any absolute spatial orientation. Unlike objects such as chairs or boats that appear usually with a specific orientation, proteins can have any orientation in 3D conformational space, thus the Cartesian coordinates defined by the model stored in PDB represent only a frozen in space and time snapshot of an overall highly dynamic structural diversity. The orientation is irrelevant to the properties of the protein. This observation underlines the need of either a rotation invariant representation or of a convention that makes output structures comparable one to another based on the definition of an intrinsic coordinate system. We define as origin of this intrinsic coordinate system the consensus barycenter of the protein as it is defined by taking into account only the four atoms of the backbone for each residue, and as axes the principal directions of each enzyme calculated by principal component analysis. Each structure is rotated around its center and the three principal directions of the enzyme aligned with the three axes of the Cartesian coordinate system. Instead of defining a common reference frame and aligning the objects before building the prediction model, other works (*Brock et al., 2016*; *Hegde & Zadeh, 2016*; *Maturana & Scherer, 2015*) applied rotations around relevant axes for data augmentation. We decided to spatially normalize the data instead of arbitrarily augmenting them (by random rotations) because the number of samples is already big enough and an adequate sampling of all possible orientations would lead to an extremely large dataset difficult to handle. However, similarly to these works the definition of orientation includes uncertainties about the direction (left-right, bottom-up), which we tackled also by data augmentation.

Another critical part of the process is to determine *how* the enzymes should be fit in their volumes, as those can be of all types of shapes and sizes. Should we scale enzymes

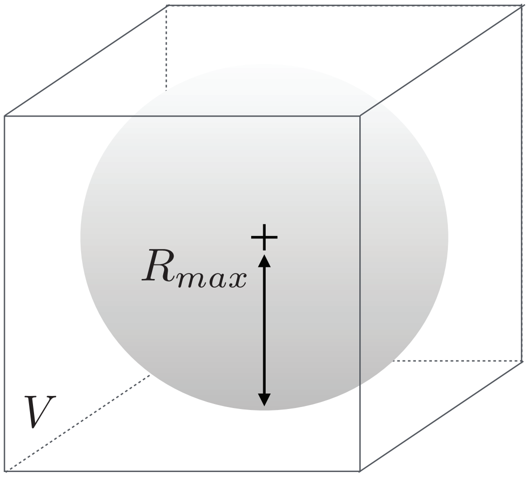

**Figure 1** Illustration of the meaning of $R_{\text{max}}$ with respect to volume $V$.
Full-size DOI: 10.7717/peerj.4750/fig-1

separately to make them fit in their respective volumes? Or, on the contrary, should we scale all enzymes in a uniform manner? We choose to select the second option because of two reasons. First, doing otherwise would lead enzymes to be represented at different resolutions. Second, biological considerations invite us to make the convolutional network aware of the size difference between samples, as those may be an implicit feature regarding class determination. We already know that our source files provide the coordinates of the enzymes at a same scale. After all, proteins are comprised of various combinations of equally sized amino acids. This scaling issue is therefore equivalent to determining a maximum radius $R_{\text{max}}$ so that the atom occupancy information contained in the sphere centered on the barycenter of the enzyme and of radius $R_{\text{max}}$ fits into $V$. This situation is illustrated in Fig. 1.

$R_{\text{max}}$ has to be large enough so that a sufficient number of enzymes fit the most of their volume inside $V$. Conversely, it also has to be small enough so that most enzymes are represented at a satisfactory resolution.

As a result, a homothetic transformation with center $S$ and ratio $\lambda$ defined by

$$S \text{ center of } V \text{ and } \lambda = \left\lfloor \frac{l}{2} - 1 \right\rfloor \times \frac{1}{R_{\text{max}}} \tag{1}$$

is performed on all enzymes to scale them to the desired size.

When $l$ is low, the grid is coarse enough so that the voxels of the structure have a contiguous shape. Conversely, big volumes tend to separate voxels, which engender "holes." In that case, consecutive backbone atoms $(\overrightarrow{A_i}, \overrightarrow{A_{i+1}})$ are interpolated by $p$ regularly spaced new points computed by

$$\frac{(p-k+1) \times \overrightarrow{A_i} + k \times \overrightarrow{A_{i+1}}}{p+1} \tag{2}$$

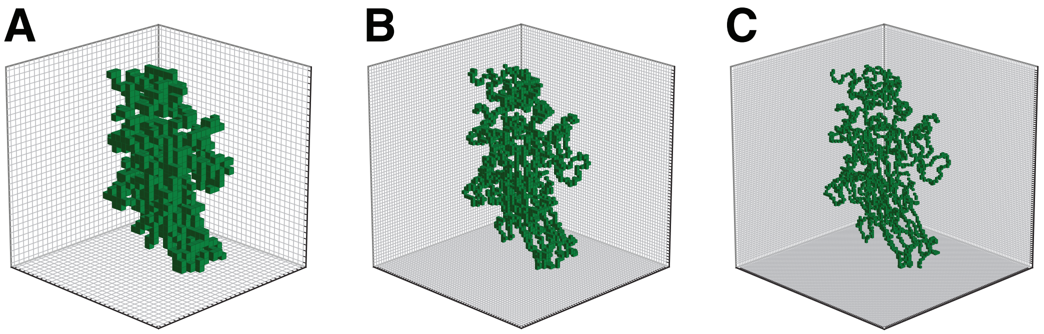

**Figure 2** **Illustration of enzyme 2Q3Z for grid sizes $l$ = 32 (A), $l$ = 64 (B), and $l$ = 96 (C).**
Full-size DOI: 10.7717/peerj.4750/fig-2

where $k$ varies from 1 to $p$. The latter is determined empirically beforehand on enzymes of the training set.

A last preprocessing step is to remove potential outliers from the volume. This is done by eliminating voxels that do not have any immediate neighbor.

The illustration of the output volume obtained for different grid sizes has been provided in Fig. 2.

## Data augmentation

As noted earlier, differences in spatial orientation had to be considered so that volumes can be comparable one to another.

Keeping the orientation constant, we perform data augmentation by applying transformations that preserve the principal components along the three axes, i.e., flips and combination of flips. The number of possible transformations applicable to each protein is $2^3$−1.

During training, all samples go through the process described in Algorithm 1.

## Architecture

We considered shallow architectures in order to develop a framework that can be trained with common computational means. A grid search of configurations has been conducted on the training set and led us to consider the two-layer architecture presented in Fig. 3: input volumes of size 32 × 32 × 32, containing the structural coordinates of the enzymes encoded in voxels, first go through a convolutional layer of 32 filters of size 9 × 9 × 9 with stride 2. Then, a second convolutional layer of 64 filters of size 5 × 5 × 5 with stride 1 is used, followed by a max-pooling layer of size 2 × 2 × 2 with stride 2. Finally, there are two fully-connected layers of 128 and six (the number of classes) hidden units respectively, concluded by a softmax layer that outputs class probabilities corresponding to the six first-level EC numbers.

Several components of the VoxNet architecture depicted by *Maturana & Scherer (2015)* have also been investigated in our network. Leaky ReLU with parameter $\alpha = 0.1$ is used as activation layer after each convolutional layer. The L2 regularization technique of strength

**Algorithm 1** **Summary of the preprocessing steps done to each enzyme at training time**

```
Data:   $N$ training enzymes, grid size of $l$, homothetic transformation ratio $\lambda$, $p$ interpolations,
        probability $p_{\text{flip}}$ to flip with respect to each axis.
Input:  Raw coordinates contained in PDB files
Output: Volumes of binary voxels representing backbone atoms occupancy
foreach of the $N$ enzymes of the training set do
    Step 1: structural information extraction
    Extract coordinates of backbone atoms from its PDB file
    Step 2: holes completion
    Interpolate consecutive backbone atoms by $p$ new points
    Step 3: size adjustment
    Center barycenter $S$ of the coordinates on $(0, 0, 0)$
    Homothetic transformation of each point with center $S$ and ratio $\lambda$
    Step 4: enzyme orientation
    Principal component analysis (PCA) transformation
    Step 5: random augmentation
    if True with probability $p_{\text{flip}}$ then
        Flip coordinates with respect to the origin along $x$—axis
    if True with probability $p_{\text{flip}}$ then
        Flip coordinates with respect to the origin along $y$—axis
    if True with probability $p_{\text{flip}}$ then
        Flip coordinates with respect to the origin along $z$—axis
    Step 6: voxelization
    Center barycenter $S$ of the coordinates on $\left(\frac{l}{2}, \frac{l}{2}, \frac{l}{2}\right)$
    Transform coordinate points into binary voxels
```

$\lambda = 0.001$ is applied on the network's layers. Overfitting is also tackled using dropout throughout the network.

In summary, this model contains 804,614 distinct parameters (including biases), which is approximately 13% less than for the VoxNet architecture.

Additionally, the Adam optimizer introduced by *Kingma & Ba (2015)* has been chosen for its computational efficiency and low need of hyper-parameter tuning.

The categorical cross entropy is used as loss function, since the latter can be adapted to multiclass classification. Two approaches are considered for computing this loss. The first one assesses misclassification error irrespectively of individual class sizes, whereas the second one takes class imbalance into account and uses customized penalization weights for each class. In this second approach, the aim is to compensate the lack of training samples in under-represented classes by providing a greater penalization to the loss in the event of misclassification than for larger classes.

The cross entropy loss is given by

$$\mathcal{L} = -\sum_{x \in \text{set}} \sum_{i=1}^{6} w_i \cdot \delta_{x,i} \cdot \log(\hat{p}_{x,i}) \tag{3}$$

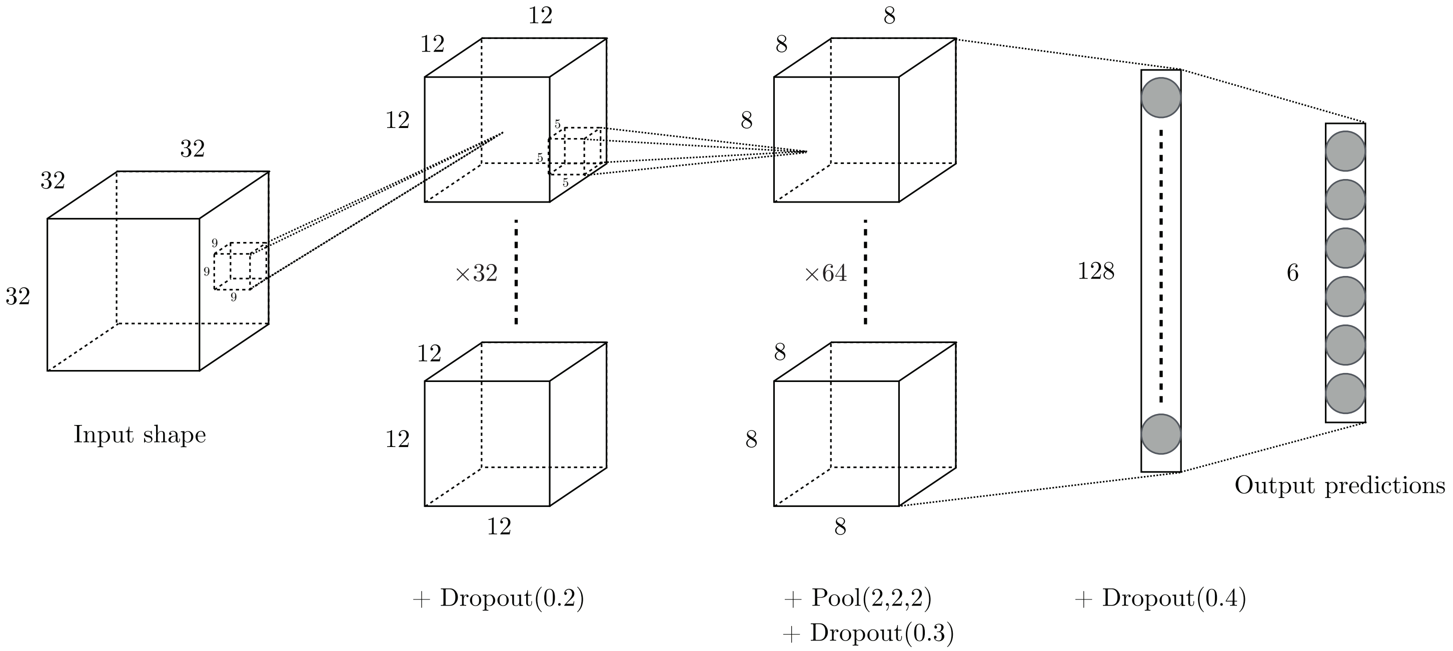

**Figure 3** **Drawing of the architecture selected for our experiments.** Full-size DOI: 10.7717/peerj.4750/fig-3

where $\hat{p}_{x,i}$ is the predicted probability of enzyme $x$ belonging to class EC$i$, $\delta_{x,i}$ the quantity equal to 1 only if enzyme $x$ belongs to class EC$i$, and $w_i$ the weight associated to class EC$i$. For the first approach, all weights $w_i$ are taken equal to 1. For the weighted approach, we chose weights according to the formula

$$w_i = \frac{\max\limits_{j \in [\![1;6]\!]} \#(\text{EC}j \text{ training enzymes})}{\#(\text{EC}i \text{ training enzymes})} \tag{4}$$

which increases the contribution of the under-represented classes by an amount inversely proportional to their size and in respect to the largest class.

## Metrics

Among the various multi-class metrics that have been studied by *Sokolova & Lapalme (2009)*, we selected the most representative ones to assess the model's performance. The metrics are based on the confusion matrix whose elements $C(i, j)$ with $i, j \in [\![1, 6]\!]$ indicate the number of enzymes that belong to class EC$i$ and are predicted as belonging to class EC$j$. They include:

- *Accuracy* which captures the average per-class effectiveness of the classifier:

$$\text{Accuracy} = \frac{\sum\limits_{i=1}^{6} C(i, i)}{\sum\limits_{i,j=1}^{6} C(i, j)}$$

- *Precision, recall* and *F*1 score which are calculated per class EC*i*:

$$\text{Precision}_{\text{EC}i} = \frac{C(i,i)}{\sum_{j=1}^{6} C(j,i)} \qquad \text{Recall}_{\text{EC}i} = \frac{C(i,i)}{\sum_{j=1}^{6} C(i,j)}$$

$$F1_{\text{EC}i} = 2 \times \frac{\text{Precision}_{\text{EC}i} \times \text{Recall}_{\text{EC}i}}{\text{Precision}_{\text{EC}i} + \text{Recall}_{\text{EC}i}}$$

  For each class, precision gives us an idea of the proportion of correctly classified enzymes among enzymes that have been classified in that class, while recall highlights the proportion of correctly classified enzymes among enzymes that actually belong to that class.

- *Macro precision, recall* and *F*1 score which express average performance over the six enzyme classes:

$$\text{Precision}_M = \frac{1}{6}\sum_{i=1}^{6} \text{Precision}_{\text{EC}i} \qquad \text{Recall}_M = \frac{1}{6}\sum_{i=1}^{6} \text{Recall}_{\text{EC}i}$$

$$F1_M = 2 \times \frac{\text{Precision}_M \times \text{Recall}_M}{\text{Precision}_M + \text{Recall}_M}$$

## Final decision rule

At testing time, several approaches are considered for determining an enzyme's final class.

- The "None" strategy consists of making a prediction based on the model of the 3D shape without any transformation.
- In the "flips" approach, the $2^3$–1 possible combinations of flipped volumes are generated and introduced to the classifier. The final class prediction $\hat{i}$ is either:
  - the class of maximum total probability (*probability*-based decision), determined by

    $$\hat{i} = \arg \max_{i \in [\![1,6]\!]} \sum_{j=1}^{n} \hat{p}_{i,j}$$

    with $n$ the total number of flips used during inference and $\hat{p}_{i,j}$ the probability to belong to class $i$ during flip $j$
  - or the class selected by majority voting (*class*-based decision), computed through the formula

    $$\hat{i} = \arg \max_{i \in [\![1,6]\!]} \sum_{j=1}^{n} \hat{\delta}_{i,j}$$

    where $\hat{\delta}_{i,j}$ equals 1 if at flip $j$, the enzyme is predicted to belong to class $i$, 0 otherwise.

- The "weighted flips" (W. flips) strategy is also based on fusion of decisions produced for each flipped volume, but this one weights each decision by a different coefficient, such

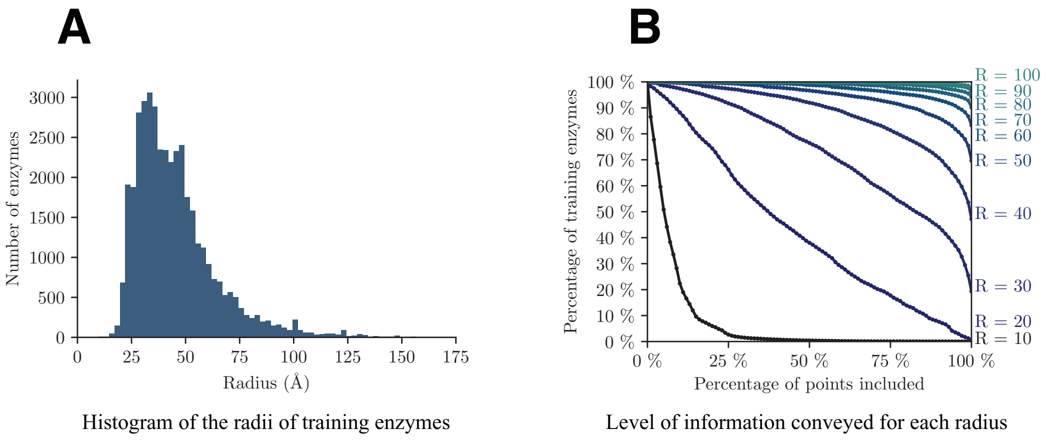

**Figure 4 Analysis of the distribution of the radii among training enzymes (A) and corresponding level of information conveyed (B).** Full-size DOI: 10.7717/peerj.4750/fig-4

as $\frac{1}{\delta_x+\delta_y+\delta_z+1}$, which highlights transformations with the least number of flips. Just like the previous approach, the final result can be determined from either a probability- or class-based viewpoint.

## Hyperparameter selection

### *Radius $R_{max}$ and interpolation parameter p*

A crucial point in the presented approach is to make a cogent selection of $R_{\max}$. As previously discussed, this parameter controls the trade-off between the level of information ($l$) retained in each volume and the resolution with which they are conveyed ($R_{\max}$). Figure 4 shows the analysis of the dataset from these two perspectives. The graph on the left helps us assess the minimum radius for which a decent amount of enzymes will be totally included in the volume, whereas the graph on the right highlights the quantity of information retained by each radius.

In details, from Fig. 4A we can see that the majority of enzymes can fit in a sphere with radius between 25 and 75 Å, with a peak around 35 Å. In Fig. 4B, each point $(x, y)$ on a curve of radius $R$ shows the percentage $y$ of training enzymes that have at least $x$ points included within radius $R$ around their respective enzyme barycenter. Based on both graphs, we select a value of $R_{\max} = 40$, that is big enough to capture more than half of enzymes in $V$, but small enough so that the smallest enzymes have radii of at least half of $R_{\max}$.

Empirical observations show that a grid of $l = 32$ does not require any atom interpolation. Therefore, $p$ is set to zero for our computations. For denser grids, e.g., with grid size $l = 64$ or 96, appropriate $p$ values were $p = 5$ and $p = 9$, respectively.

### *Random selection of transformations and samples*

Regarding data augmentation, the probability of enzyme flip along each axis has been set to 2/10. That way for each enzyme, higher numbers of flips have a lower probability of happening. Also for each pass, approximately half randomly selected enzymes are to be

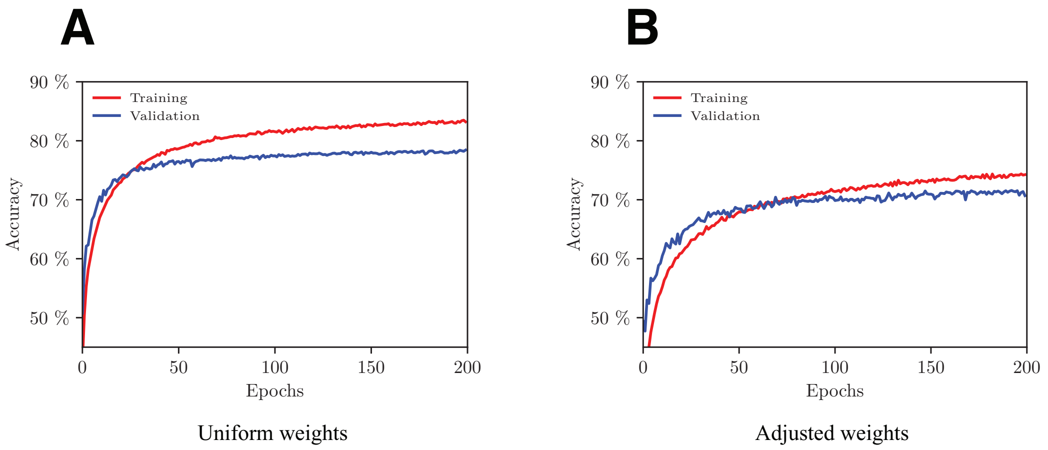

**Figure 5 Evolution of the network's performance during the training process with uniform (A) and adjusted (B) weights.** Full-size DOI: 10.7717/peerj.4750/fig-5

augmented by flips (or a combination of flips). This process is useful, as it will help us obtaining a robust classifier.

## RESULTS

We trained the model with and without weights adjustment in the loss function using a fivefold cross-validation scheme. The evolution of performance on one of the folds with increasing number of epochs is shown in Fig. 5. With the adopted configuration of hyperparameters, the model converges after 200 epochs.

Cross-validation results associated with our experiments are shown in Table 2. For each fold, both networks with and without class weights (respectively called "adapted" and "uniform" models) have been trained on a training set of 50,846 samples and tested on the remaining 12,712 samples. The cross-validation performance of each model has been evaluated using each of the five inference schemes described earlier and with both micro- and macro-level metrics.

Overall, the model that performs best in terms of both accuracy (77.6%) and macro *F*1 (73.7%) is the one with uniform weights using no data augmentation.

Precision per class on under-represented classes (EC4, 5, 6) is far better on uniformly weighted models compared to adapted models, with best scores being 97.2%, 97.3% and 97.7%, versus 83.6%, 59.1% and 35.1% respectively. Over-represented classes (EC2, 3) have roughly the same performance in those two types of model.

It is worth noting that by augmenting the data, the precision per class is noticeably improved on under-represented classes (EC4, 5, 6) in both uniform and adapted models. In fact, in the uniformly weighted framework, the best data augmented models achieve 97.2%, 97.3%, and 97.7% versus 92.7%, 88.8%, and 84.9% for the non-augmented one for EC4, 5, 6 respectively. An interpretation to this fact is that the configurations produced with flips are good at imitating plausible configurations that hold similar EC properties. Similarly, using data augmentation during inference for the adapted framework achieves better performance than when it is not used, with 83.6%, 59.1%, and 35.1% versus 69.6%, 46.2%, and 32.4% for EC4, 5, 6 respectively.

Table 2 **Fivefold cross-validation results.**

| **Weights** | | | **Uniform** | | | | | **Adapted** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Decision** | | | **None** | **Probabilities** | | **Classes** | | **None** | **Probabilities** | | **Classes** | |
| **Augmentation** | | | | **Flips** | **W. flips** | **Flips** | **W. flips** | | **Flips** | **W. flips** | **Flips** | **W. flips** |
| Accuracy | | | **0.776** ± 0.002 | 0.752 ± 0.004 | 0.771 ± 0.002 | 0.748 ± 0.002 | 0.765 ± 0.002 | 0.711 ± 0.006 | 0.719 ± 0.007 | 0.733 ± 0.004 | 0.710 ± 0.007 | 0.722 ± 0.008 |
| Precision | EC | 1 | 0.841 ± 0.019 | 0.900 ± 0.021 | **0.905** ± 0.022 | 0.859 ± 0.026 | 0.884 ± 0.022 | 0.752 ± 0.026 | 0.786 ± 0.036 | 0.796 ± 0.035 | 0.717 ± 0.036 | 0.769 ± 0.029 |
| | | 2 | 0.761 ± 0.021 | 0.753 ± 0.040 | 0.768 ± 0.035 | 0.725 ± 0.035 | 0.758 ± 0.032 | 0.784 ± 0.015 | 0.784 ± 0.024 | **0.799** ± 0.026 | 0.760 ± 0.026 | 0.789 ± 0.025 |
| | | 3 | 0.740 ± 0.017 | 0.686 ± 0.026 | 0.706 ± 0.023 | 0.702 ± 0.025 | 0.706 ± 0.022 | **0.774** ± 0.020 | 0.735 ± 0.030 | 0.750 ± 0.030 | 0.743 ± 0.030 | 0.752 ± 0.028 |
| | | 4 | 0.927 ± 0.018 | **0.972** ± 0.008 | **0.972** ± 0.011 | 0.970 ± 0.012 | 0.964 ± 0.007 | 0.696 ± 0.032 | 0.836 ± 0.042 | 0.833 ± 0.042 | 0.835 ± 0.036 | 0.792 ± 0.037 |
| | | 5 | 0.888 ± 0.013 | 0.967 ± 0.012 | **0.973** ± 0.010 | 0.971 ± 0.015 | 0.954 ± 0.010 | 0.462 ± 0.046 | 0.584 ± 0.070 | 0.589 ± 0.060 | 0.591 ± 0.074 | 0.550 ± 0.067 |
| | | 6 | 0.849 ± 0.037 | **0.977** ± 0.029 | 0.971 ± 0.031 | **0.977** ± 0.024 | 0.953 ± 0.038 | 0.324 ± 0.040 | 0.330 ± 0.052 | 0.347 ± 0.049 | 0.351 ± 0.049 | 0.339 ± 0.051 |
| | | Macro | 0.834 ± 0.011 | 0.876 ± 0.007 | **0.883** ± 0.009 | 0.867 ± 0.007 | 0.870 ± 0.009 | 0.632 ± 0.006 | 0.676 ± 0.008 | 0.685 ± 0.006 | 0.666 ± 0.010 | 0.665 ± 0.009 |
| Recall | EC | 1 | **0.747** ± 0.013 | 0.686 ± 0.021 | 0.715 ± 0.018 | 0.706 ± 0.018 | 0.718 ± 0.018 | 0.746 ± 0.008 | 0.722 ± 0.018 | 0.745 ± 0.012 | 0.735 ± 0.014 | 0.739 ± 0.009 |
| | | 2 | 0.784 ± 0.021 | 0.766 ± 0.033 | **0.785** ± 0.028 | 0.783 ± 0.028 | 0.777 ± 0.026 | 0.657 ± 0.026 | 0.646 ± 0.040 | 0.665 ± 0.034 | 0.650 ± 0.035 | 0.656 ± 0.032 |
| | | 3 | 0.880 ± 0.015 | 0.907 ± 0.021 | **0.909** ± 0.019 | 0.881 ± 0.023 | 0.900 ± 0.019 | 0.753 ± 0.018 | 0.790 ± 0.031 | 0.800 ± 0.029 | 0.766 ± 0.032 | 0.781 ± 0.028 |
| | | 4 | 0.645 ± 0.008 | 0.541 ± 0.006 | 0.583 ± 0.009 | 0.514 ± 0.007 | 0.583 ± 0.008 | **0.717** ± 0.019 | 0.698 ± 0.019 | 0.711 ± 0.016 | 0.678 ± 0.015 | 0.708 ± 0.021 |
| | | 5 | 0.525 ± 0.016 | 0.416 ± 0.017 | 0.462 ± 0.012 | 0.402 ± 0.017 | 0.458 ± 0.009 | 0.663 ± 0.019 | 0.663 ± 0.021 | **0.667** ± 0.020 | 0.643 ± 0.015 | 0.665 ± 0.019 |
| | | 6 | 0.382 ± 0.020 | 0.233 ± 0.022 | 0.283 ± 0.019 | 0.203 ± 0.017 | 0.284 ± 0.018 | 0.638 ± 0.037 | 0.664 ± 0.041 | **0.665** ± 0.047 | 0.633 ± 0.036 | 0.655 ± 0.043 |
| | | Macro | 0.660 ± 0.003 | 0.592 ± 0.007 | 0.623 ± 0.005 | 0.582 ± 0.007 | 0.620 ± 0.005 | 0.695 ± 0.007 | 0.697 ± 0.009 | **0.709** ± 0.008 | 0.684 ± 0.008 | 0.701 ± 0.010 |
| F1 | EC | 1 | 0.791 ± 0.007 | 0.778 ± 0.008 | **0.799** ± 0.008 | 0.774 ± 0.006 | 0.792 ± 0.005 | 0.749 ± 0.011 | 0.752 ± 0.011 | 0.769 ± 0.013 | 0.725 ± 0.014 | 0.754 ± 0.010 |
| | | 2 | 0.772 ± 0.007 | 0.758 ± 0.005 | **0.775** ± 0.005 | 0.752 ± 0.007 | 0.766 ± 0.006 | 0.712 ± 0.010 | 0.707 ± 0.016 | 0.724 ± 0.013 | 0.700 ± 0.013 | 0.715 ± 0.013 |
| | | 3 | **0.803** ± 0.005 | 0.781 ± 0.009 | 0.795 ± 0.008 | 0.781 ± 0.007 | 0.791 ± 0.007 | 0.763 ± 0.007 | 0.760 ± 0.006 | 0.773 ± 0.005 | 0.753 ± 0.008 | 0.766 ± 0.007 |
| | | 4 | 0.761 ± 0.007 | 0.695 ± 0.005 | 0.728 ± 0.007 | 0.672 ± 0.005 | 0.726 ± 0.005 | 0.706 ± 0.011 | 0.761 ± 0.015 | **0.767** ± 0.014 | 0.748 ± 0.012 | 0.747 ± 0.016 |
| | | 5 | **0.659** ± 0.015 | 0.582 ± 0.018 | 0.625 ± 0.013 | 0.569 ± 0.019 | 0.619 ± 0.010 | 0.543 ± 0.032 | 0.618 ± 0.037 | 0.624 ± 0.032 | 0.613 ± 0.043 | 0.600 ± 0.043 |
| | | 6 | **0.526** ± 0.021 | 0.376 ± 0.029 | 0.438 ± 0.023 | 0.336 ± 0.023 | 0.438 ± 0.022 | 0.428 ± 0.031 | 0.437 ± 0.042 | 0.452 ± 0.034 | 0.449 ± 0.036 | 0.444 ± 0.041 |
| | | Macro | **0.737** ± 0.003 | 0.706 ± 0.004 | 0.730 ± 0.003 | 0.696 ± 0.003 | 0.724 ± 0.002 | 0.662 ± 0.006 | 0.687 ± 0.007 | 0.697 ± 0.004 | 0.675 ± 0.007 | 0.683 ± 0.008 |

**Note:**
Best performance for each indicator is shown in bold.

Recall per class on over-represented classes (EC2, 3) are best with uniformly weighted models (78.5% and 90.9% respectively). It is worth noting that the adapted model outperforms the uniform one on under-represented classes (EC4, 5, 6) with 71.7%, 66.7%, 66.5% recall per class respectively, compared to 64.5%, 52.5%, and 38.2% respectively on uniform models. This drastic performance increase was once again expected, as we know that the adapted-weights network was heavily penalized for its errors on small classes during the training process.

Uniformly weighted models perform best in terms of *F*1 scores, ranging from 52.6% for the smallest class (EC6) to 79.9% for EC1, with a macro *F*1 of 73.7%.

### Computation time

Our architecture was implemented on Python 3 using Keras (*Chollet, 2015*) on top of a GPU-enabled version of TensorFlow (*Abadi et al., 2015*). Enzyme information has been extracted using the open-source module BioPython, a fast and easy-to-use tool presented by *Cock et al. (2009)*. The complete training of our model on 50,846 samples took about 5 h on an Intel i7 6700K machine with 32 GB of RAM and a GTX 1080 graphics card. During the training phase, the code computes batches in parallel and stores them in a queue in real-time, so that the main computational burden comes from neural network calculations and not the representation process. The average prediction time of the function of a single enzyme was about 6 ms without flips, or 50 ms with flips.

## DISCUSSION

The general trend is that uniformly weighted models perform well in terms of macro accuracy, macro precision as well as macro *F*1 while adapted models are slightly better in macro recall. More particularly, on under-represented classes (EC4, 5, 6), models perform better in terms of precision per class when using flip data augmentation, which means that using data augmentation increases reliability on predictions. This can be explained by the fact that the classifier has more examples at hand, which makes its predictions more robust. Interestingly, on under-represented classes, uniform models have the highest precision per class and the lowest recall per class while adapted models are exactly the opposite: they have the lowest precision per class and the highest recall per class. The interpretation of this clear trend is that enzymes coming from under-represented classes are not always recognized by uniform classifiers as they are biased towards enzymes from over-represented class because the classifier does not correct for class imbalance. On the contrary, enzymes from under-represented classes are well recognized by adapted classifiers as they account for class imbalance, but this comes at a price: by predicting more enzymes as being in those under-represented classes (false positives), the classifier tends to make a lot more mistakes which leads to a low precision per class.

### Further improvements of the method

In parallel, we investigated possible improvements in the architecture. We introduced batch normalization (*Ioffe & Szegedy, 2015*) at several positions of the network and repeated the same experiments as before. Those were placed after every activation layer,

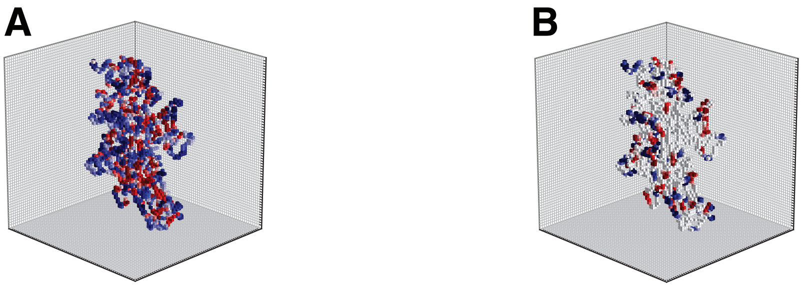

Figure 6 **Illustration of the hydropathy (A) and charge (B) attributes incorporated into the shape model for enzyme 2Q3Z.** Each attribute is illustrated in color for better visualization, but it actually corresponds to a single channel. Full-size DOI: 10.7717/peerj.4750/fig-6

as suggested by *Mishkin, Sergievskiy & Matas (2017)*. Also, Leaky ReLU activation layers were replaced by PReLU (*He et al., 2015*) layers which enable the network to learn adaptively the Leaky ReLU's best $\alpha$ parameter. Although these changes led the network to converge faster to a stable optimum, the final performance increase was of an order of magnitude of only one percent and came at a higher computational cost.

In the following, we considered the transition from a binary representation capturing only shape information to "gray-level" representation capturing also information content. Several biological indicators such as the hydropathy index (*Kyte & Doolittle, 1982*), or isoelectric points (*Wade, 2002*) can be used to better describe local properties of amino acids that are the building blocks of the protein structure. From the perspective of computational analysis, these attributes can be incorporated into the representation model by attaching to the shape also appearance information. Figure 6 illustrates this idea showing the volumetric representation of two different attributes (hydropathy on the left and charge on the right). The different attributes, if up to three, can also be merged into a single structure visualized in color (RGB) scale. Different approaches to handle multi-channel volumetric data (images) by CNNs have recently been presented in works such as those of *Zhang et al. (2017)*, *Cao et al. (2016)*, *Kamnitsas et al. (2017)* and *Nie et al. (2016)*. The literature however on the use of deep networks for 3D shapes with multi-channel appearance is limited. As preliminary analysis, we applied EnzyNet without modification on the architecture, but with re-training for the adjustments of the weights of the convolutional kernels. We introduced to the network either a single attribute (as a binary 3D image or gray-level 3D image), such as shape, hydropathy and isoelectric points, or a combination of these different attributes. Two other ways of information combination were examined: either each attribute was introduced as a different channel in the CNN, or the outputs of the single channel networks were combined through a fusion rule. However, all methods showed an increase of the order of magnitude of one percent. Further investigation will be required to appropriately harness this added information.

Furthermore, in the current study a relatively small grid size was selected due to computational limitations. However, as can be seen in Fig. 2 the details of the spatial structure of proteins can be better differentiated for higher values of *l*. The next step of this work would be to adapt the current architecture accordingly into a similar network that processes higher-resolution volumes. This could enable the network to capture more subtle features and potentially boost the performance of the classifier. On the other hand, the model performance using a finer representation might be more sensitive to possible inaccuracies in protein structure prediction. Initial tests showed that the appearance of artifacts in the location of atoms at $l = 64$ or 96 had a negligible impact on accuracy.

*Wei et al. (2015)* present a method that enables 3D CNN to deal with multi-label classification problems. This approach is interesting for us as it allows the extension of our method to the classification of multi-label enzymes. The obtained performance could subsequently be compared to previous work *Amidi et al. (2017)*.

Other approaches based on the 3D representation of enzymes are also possible. *Brock et al. (2016)* performed very well on the ModelNet dataset using generative and discriminative modeling. Their voxel-based autoencoder is helpful for assessing the key features that are correctly learned from the 3D shapes. We could elaborate this information in order to identify significant features in a pre-training phase aiming to obtain better prediction performance.

## ADDITIONAL INFORMATION AND DECLARATIONS

### Funding

The authors received no funding for this work.

### Competing Interests

The authors declare that they have no competing interests.

### Author Contributions

- Afshine Amidi conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, authored or reviewed drafts of the paper, approved the final draft.
- Shervine Amidi conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, authored or reviewed drafts of the paper, approved the final draft.
- Dimitrios Vlachakis conceived and designed the experiments, analyzed the data, contributed reagents/materials/analysis tools, authored or reviewed drafts of the paper, approved the final draft.
- Vasileios Megalooikonomou conceived and designed the experiments, authored or reviewed drafts of the paper, approved the final draft.
- Nikos Paragios conceived and designed the experiments, authored or reviewed drafts of the paper, approved the final draft.
- Evangelia I. Zacharaki conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the paper, approved the final draft.

### Data Availability

The following information was supplied regarding data availability:

GitHub: https://github.com/shervinea/enzynet.

## REFERENCES

**Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. 2015.** TensorFlow: large-scale machine learning on heterogeneous systems. *Available at http://tensorflow.org*.

**Alipanahi B, Delong A, Weirauch MT, Frey BJ. 2015.** Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* **33(8)**:831–838 DOI 10.1038/nbt.3300.

**Amidi A, Amidi S, Vlachakis D, Paragios N, Zacharaki EI. 2016.** *A Machine Learning Methodology for Enzyme Functional Classification Combining Structural and Protein Sequence Descriptors*. Cham: Springer International Publishing, 728–738.

**Amidi S, Amidi A, Vlachakis D, Paragios N, Zacharaki EI. 2017.** Automatic single- and multi-label enzymatic function prediction by machine learning. *PeerJ* **5**:e3095 DOI 10.7717/peerj.3095.

**Angermueller C, Pärnamaa T, Parts L, Stegle O. 2016.** Deep learning for computational biology. *Molecular Systems Biology* **12(7)**:878 DOI 10.15252/msb.20156651.

**Asgari E, Mofrad MRK. 2015.** Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE* **10(11)**:e0141287 DOI 10.1371/journal.pone.0141287.

**Baldi P, Pollastri G, Andersen C, Brunak S. 2000.** Matching protein beta-sheet partners by feedforward and recurrent neural networks. In: *Proceedings, International Conference on Intelligent Systems for Molecular Biology, Menlo Park, CA*, 25–36.

**Baskin II, Palyulin VA, Zefirov NS. 2008.** Neural networks in building QSAR models. In: Livingstone DJ, ed. *Methods in Molecular Biology*. Vol. 458. Totowa: Humana Press, 137–158.

**Brock A, Lim T, Ritchie J, Weston N. 2016.** Generative and discriminative voxel modeling with convolutional neural networks. *Available at http://arxiv.org/abs/1608.04236*.

**Cao L, Liu Z, He X, Cao Y, Li K. 2016.** Mental disease feature extraction with MRI by 3D convolutional neural network with multi-channel input. In: *2016 IEEE International Conference on Smart Cloud, SmartCloud 2016*. November 18–20, 2016. New York, 224–227.

**Chollet F. 2015.** Keras. *Available at https://github.com/fchollet/keras*.

**Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, de Hoon MJ. 2009.** Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25(11)**:1422–1423 DOI 10.1093/bioinformatics/btp163.

**Di Lena P, Nagata K, Baldi P. 2012.** Deep architectures for protein contact map prediction. *Bioinformatics* **28(19)**:2449–2457 DOI 10.1093/bioinformatics/bts475.

**Dobson PD, Doig AJ. 2005.** Predicting enzyme class from protein structure without alignments. *Journal of Molecular Biology* **345(1)**:187–199 DOI 10.1016/j.jmb.2004.10.024.

**He K, Zhang X, Ren S, Sun J. 2015.** Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: *International Conference on Conference Vision, Santiago, Chile*.

**Heffernan R, Paliwal K, Lyons J, Dehzangi IA, Sharma A, Wang J, Sattar A, Yang Y, Zhou Y. 2015.** Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific Reports* **5**(**1**):1–11 DOI 10.1038/srep11476.

**Hegde V, Zadeh R. 2016.** FusionNet: 3D object classification using multiple data representations. *Available at http://arxiv.org/abs/1607.05695v4*.

**Hochreiter S, Heusel M, Obermayer K. 2007.** Fast model-based protein homology detection without alignment. *Bioinformatics* **23**(**14**):1728–1736 DOI 10.1093/bioinformatics/btm247.

**Illergard K, Ardell DH, Elofsson A. 2009.** Structure is three to ten times more conserved than sequence—A study of structural response in protein cores. *Proteins: Structure, Function, and Bioinformatics* **77**(**3**):499–508 DOI 10.1002/prot.22458.

**Ioffe S, Szegedy C. 2015.** Batch normalization: accelerating deep network training by reducing internal covariate shift. *JMLR*, 37 *Available at http://arxiv.org/abs/1502.03167*.

**Jones W, Alasoo K, Fishman D, Parts L. 2017.** Computational biology: deep learning. *Emerging Topics in Life Sciences* **1**(**3**):257–274 DOI 10.1042/etls20160025.

**Kamnitsas K, Ledig C, Newcombe VFJ, Simpson JP, Kane AD, Menon DK, Rueckert D, Glocker B. 2017.** Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis* **36**:61–78 DOI 10.1016/j.media.2016.10.004.

**Kelley DR, Snoek J, Rinn J. 2015.** Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *bioRxiv Preprint* 028399.

**Kingma DP, Ba JL. 2015.** Adam: a method for stochastic optimization. In: *ICLR. Available at http://arxiv.org/abs/1412.6980v8*.

**Kumar C, Choudhary A. 2012.** A top-down approach to classify enzyme functional classes and sub-classes using random forest. *EURASIP Journal on Bioinformatics and System Biology* **2012**(**1**):1–14 DOI 10.1186/1687-4153-2012-1.

**Kyte J, Doolittle RF. 1982.** A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology* **157**(**1**):105–132 DOI 10.1016/0022-2836(82)90515-0.

**Lanchantin J, Singh R, Lin Z, Qi Y. 2016.** Deep motif: visualizing genomic sequence classifications. *CoRR, Available at http://arxiv.org/abs/1605.01133*.

**Li H, Zhang Z, Liu Z. 2017.** Application of artificial neural networks for catalysis: a review. *Catalysts* **7**(**10**):306 DOI 10.3390/catal7100306.

**Lin Z, Lanchantin J, Qi Y. 2016.** MUST-CNN: a multilayer shift-and-stitch deep convolutional architecture for sequence-based protein structure prediction. In: *Association for the Advancement of Artificial Intelligence, Phoenix, AZ*.

**Lyons J, Dehzangi A, Heffernan R, Sharma A, Paliwal K, Sattar A, Zhou Y, Yang Y. 2014.** Predicting backbone Cα angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network. *Journal of Computational Chemistry* **35**(**28**):2040–2046 DOI 10.1002/jcc.23718.

**Maturana D, Scherer S. 2015.** VoxNet: a 3D convolutional neural network for real-time object recognition. In: *IROS, Hamburg, Germany*.

**Min S, Lee B, Yoon S. 2017.** Deep learning in bioinformatics. *Briefings in Bioinformatics* **18**(**5**):851–869.

**Mishkin D, Sergievskiy N, Matas J. 2017.** Systematic evaluation of CNN advances on the imagenet. *Computer Vision and Image Understanding* **161**:11–19 DOI 10.1016/j.cviu.2017.05.007.

**Nguyen SP, Shang Y, Xu D. 2014.** Dl-pro: a novel deep learning method for protein model quality assessment. In: *International Joint Conference on Neural Networks, Beijing, China*, 2014, 2071–2078.

**Nie D, Zhang H, Adeli E, Liu L, Shen D. 2016.** 3D deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In: *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2016—19th International Conference, October 17–21, 2016.* Athens, 212–220.

**Rumelhart DE, Hinton GE, Williams RJ. 1986.** Learning representations by back-propagating errors. *Nature* **323(6088)**:533–536 DOI 10.1038/323533a0.

**Sharma M, Garg P. 2014.** Computational approaches for enzyme functional class prediction: a review. *Current Proteomics* **11(1)**:17–22 DOI 10.2174/157016461166140415225013.

**Shrikumar A, Greenside P, Kundaje A. 2017.** Learning important features through propagating activation differences. *CoRR. Available at http://arxiv.org/abs/1704.02685*.

**Sokolova M, Lapalme G. 2009.** A systematic analysis of performance measures for classification tasks. *Information Processing and Management* **45(4)**:427–437 DOI 10.1016/j.ipm.2009.03.002.

**Sonderby S, Winther O. 2014.** Protein secondary structure prediction with long short term memory networks. *Available at http://arxiv.org/abs/1412.7828*.

**Sonderby SK, Sonderby CK, Nielsen H, Winther O. 2015.** *Convolutional LSTM Networks for Subcellular Localization of Proteins.* Cham: Springer International Publishing, 68–80.

**Spencer M, Eickholt J, Cheng J. 2015.** A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **12(1)**:103–112 DOI 10.1109/tcbb.2014.2343960.

**Szaleniec M. 2012.** Prediction of enzyme activity with neural network models based on electronic and geometrical features of substrates. *Pharmacological Reports* **64(4)**:761–781 DOI 10.1016/s1734-1140(12)70873-3.

**Wade LG. 2002.** *Organic Chemistry.* Upper Saddle River: Prentice Hall.

**Wei Y, Xia W, Huang J, Ni B, Dong J, Zhao Y, Yan S. 2015.** HCP: a flexible CNN framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38(9)**:1901–1907 DOI 10.1109/tpami.2015.2491929.

**Yadav S, Tiwari A. 2015.** Classification of enzymes using machine learning based approaches: a review. *Machine Learning and Applications* **2(3/4)**:30–49 DOI 10.5121/mlaij.2015.2404.

**Zacharaki EI. 2017.** Prediction of protein function using a deep convolutional neural network ensemble. *PeerJ* **3**:e124.

**Zeng H, Edwards MD, Liu G, Gifford DK. 2016.** Convolutional neural network architectures for predicting DNA—protein binding. *Bioinformatics* **32(12)**:i121–i127 DOI 10.1093/bioinformatics/btw255.

**Zhang W, Han L, Sun J, Guo H, Dai J. 2017.** Application of multi-channel 3D-cube successive convolution network for convective storm nowcasting. *CoRR, Available at http://arxiv.org/abs/1702.04517*.

**Zhou J, Troyanskaya OG. 2015.** Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods* **12(10)**:931–934 DOI 10.1038/nmeth.3547.