# DeepSurf: A surface-based deep learning approach for the prediction of ligand binding sites on proteins

Stelios K. Mylonas[1,*], Apostolos Axenopoulos[1,*], and Petros Daras[1]

[1]Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece.

[*]*Correspondence*: smylonas@iti.gr, axenop@iti.gr

## Abstract

The knowledge of potentially druggable binding sites on proteins is an important preliminary step towards the discovery of novel drugs. The computational prediction of such areas can be boosted by following the recent major advances in the deep learning field and by exploiting the increasing availability of proper data. In this paper, a novel computational method for the prediction of potential binding sites is proposed, called DeepSurf. DeepSurf combines a surface-based representation, where a number of 3D voxelized grids are placed on the protein's surface, with state-of-the-art deep learning architectures. After being trained on the large database of scPDB, DeepSurf demonstrates superior performance on two diverse testing datasets, by surpassing all its main deep learning-based competitors. The source code of the method along with trained models are freely available at https://github.com/stemylonas/DeepSurf.git.

## 1 Introduction

Structure-based drug discovery relies mostly on knowledge of potential binding sites of small compounds on protein structures. Computational binding site prediction (BSP) allows to predict in silico properties that would require much effort to establish experimentally and can enhance significantly the drug discovery process.

Through the years, a plethora of methods have been proposed for the structure-based BSP task, and, according to [1], they can be roughly separated in three categories: the geometry-based, the energy-based and the template-based ones. Geometry-based methods (ConCavity [2], Fpocket [3], CriticalFinder [4]) predict binding cavities based solely on the geometry of the molecular surface, while energy-based methods (FTSite [5], AutoSite [6], [7]) calculate interaction energies between protein atoms and chemical probes and attempt to locate energy minima on protein's surface. On the other hand, template-based methods (Findsite [8], LBias [9], LIBRA [10]) aim to extract binding sites on a protein by performing global or local structural alignment between this protein and a set of preexisting templates. Furthermore, consensus algorithms have been proposed that combine the results from numerous standalone methods (metaPocket2.0 [11], COACH [12]).

A new perspective on bioinformatics has been provided by the machine learning (ML) field. Machine learning techniques exploit the available amount of labeled data and, through the automated and iterative process of learning, manage to analyze and extract the underlying patterns that eventually correlate the data with their assigned label. Such methodologies have also been recently introduced to the structure-based BSP task [13],[14]. Specifically, Krivak and Hoksza proposed the P2Rank method [14],

which employs a random-forest (RF) classifier to predict ligandability score for points placed on the solvent accessible surface of a protein. A set of chemical and geometrical features are calculated on local spherical neighbourhoods around these points and operate later as input to the RF classifier. The points receiving the highest ligandability scores are spatially clustered to finally provide the predicted binding sites.

Over the last few years, the increasing availability of large amount of data has led to the development of a subfield in ML, namely the deep learning (DL) field. DL has surpassed by far more traditional ML methods in many scientific domains (computer vision, natural language processing, etc) and has been recently applied in a variety of structural bioinformatics tasks, such as virtual screening [15],[16], binding affinity prediction [17],[18] or protein structure prediction [19],[20]. DeepSite [21] was the first attempt to employ a DL architecture in structure-based BSP task, by using a rather shallow convolutional neural network (CNN) of 4 layers. DeepSite, like P2Rank, treats binding site prediction as a binary classification problem, where "binding" and "not binding" are the two considered classes. Their main difference is that DeepSite does not utilize any surface information but, on the contrary, operates on a 3D voxelized grid of the protein. For each voxel of the grid, a feature vector is computed based on the physicochemical properties of the neighboring protein atoms. Then, a sliding cuboid window of $16 \times 16 \times 16$ traverses the entire grid creating subgrids of features, which are then imported to the CNN. Each subgrid is finally assigned a ligandability score by the network. A very similar approach has been proposed in [22], where the main difference is the set of features employed. Two recently proposed methods, called Kalasanty [23] and FRSite [24], resemble to DeepSite in protein representation, since they also employ a 3D voxelization of the entire protein, but they differ on how they approach the BSP task. BSP is treated as an object-detection problem by FRSite and as a semantic segmentation problem by Kalasanty, where in both cases the desired object to be extracted is the corresponding binding site. In FRSite, a 3D version of Faster-RCNN is employed, while in Kalasanty a common segmentation architecture, called U-Net, is adapted to the needs of the specific task. According to the reported results in Kalasanty, this alternate representation has achieved higher accuracies than DeepSite.

Among the aforementioned methodologies, both DeepSite and Kalasanty exploit the inherent capabilities of deep learning architectures to learn from large databases and automatically extract features. However, the voxelized representation of the entire protein they adopt may have several limitations: It neglects any knowledge of the surface morphology, while this fixed structural discretization of the input space can lead to information loss. On the other hand, P2Rank tries to exploit the binding mechanics in a more efficient way, by employing a surface-based representation of the protein, which resembles more to the actual binding process. As stated in [14], focusing on grid points or atoms has led experimentally to significantly worse results than focusing on surface points. Nevertheless, the classification approach employed in P2Rank (using RF classifier) has limited generalisation capabilities comparing with approaches based on Deep Learning, which can benefit from much larger training datasets.

Inspired by the aforementioned approaches, our proposed DeepSurf method combines effectively the learning capabilities of advanced CNN architectures with the surface-based representation of the 3D protein structure. More specifically, DeepSurf employs a 3D-CNN architecture on localized 3D grids, which are appropriately oriented and placed on a set of selected surface points, as detailed in Section 2.1. This approach of localized grids resembles to P2Rank, in the sense that both methodologies consider a local neighborhood around each surface point. Their basic differences are that, instead of calculating handcrafted features, we employ a 3D-CNN for feature extraction, and that we consider a voxelized cuboid neighborhood, instead of a spherical one.

The main contributions of our approach are i) a new representation of the 3D protein surface is introduced, based on local voxel grids centred at sample points of the surface; ii) a novel residual network LDS-ResNet that has shown better performance than the baseline ResNet in image analysis tasks has been

**Algorithm 1** DeepSurf

**Input:** Protein structure

1: Create the solvent accessible surface of the protein
2: Reduce the set of surface points
3: **for** each point $P$ **do**
4: Compute normal vector $\boldsymbol{n}$ on $P$
5: Create local grid on $P$, aligned according to $\boldsymbol{n}$
6: Calculate grid features
7: Import grid to 3D-CNN and get ligandability score for $P$
8: **end for**
9: Discard points with score less than $T$
10: Cluster the remaining points
11: **for** each cluster **do**
12: Assign each cluster point to its closest protein atom
13: Form a binding site from these atoms
14: **end for**
15: Rank binding sites by average ligandability score

**Output:** Binding sites

extended in three dimensions to be applicable to volumetric data. The proposed method has been evaluated in binding site prediction using different benchmark datasets, demonstrating superior performance among state-of-the-art approaches.

# 2 Proposed method

## 2.1 DeepSurf

A short outline of our method is given in Algorithm 1. Firstly, the solvent accessible surface (SAS) of the protein is created in a triangular mesh format. The resulting mesh is usually too dense, with unnecessary redundancy of points, which can lead our algorithm to a severe computational burden. For this reason, we apply a subsequent "mesh simplification" step, where the total number of surface points is reduced by a factor of $f$ (e.g. $f = 10$). For this task, we employ the K-means clustering algorithm aiming to aggregate adjacent points into one cluster. If $n_p$ is the number of original surface points, the total number of created clusters is equal to $n_p/f$. As we can see, parameter $f$ controls the density of the points to be preserved and corresponds to the average number of points per cluster. Finally, from each cluster the closest point to the cluster center is kept.

One issue related to the voxelized representation of a protein is the lack of rotation invariance. Specifically, due to lack of symmetry, the employed 3D cuboid grids are always rotation-sensitive and strongly depend on the arbitrary placement of the axes. Most methods attempted to address this issue by augmenting the data with random rotations during training [15],[23]. On the other hand, P2Rank, as a non-voxelized method, bypassed this issue by utilizing symmetric spherical neighborhoods. We aim to alleviate this problem by aligning the local grids with the orientation of the normal vectors of the corresponding surface points. This alignment approach was inspired by a previous work [25], where local spherical regions on a protein surface were aligned according to the orientation of the normal vectors, in order to extract local shape descriptors. An illustration of this step is shown in Fig. 1. A local grid of size $16 \times 16 \times 16$ and resolution 1 Å is centered on surface point $P$ and is oriented such that the z-axis is always parallel to the normal vector $n$ on $P$, i.e. perpendicular to the surface. With this approach, the rotation issue is not eliminated, since random rotations are still applied during training. However, this selective initial placement of axes, instead of a random one, resulted to a more effective training and evaluation scheme.

After the proper localization and orientation of the grid, the next step is to calculate the necessary features that will form the 4D tensor which is then imported to the 3D-CNN. We adopt here the featurization scheme initially introduced by [17] and used also in Kalasanty [23], which consists of 18 chemical features calculated per protein atom. Each grid voxel receives the features of the atoms inside it. The formed 4D tensor is then imported to CNN and produces at the output a ligandability score for the specific surface point. Although our approach has been tested using specific deep neural network architectures, the proposed methodology is generic, meaning that any 3D-CNN architecture that receives as input

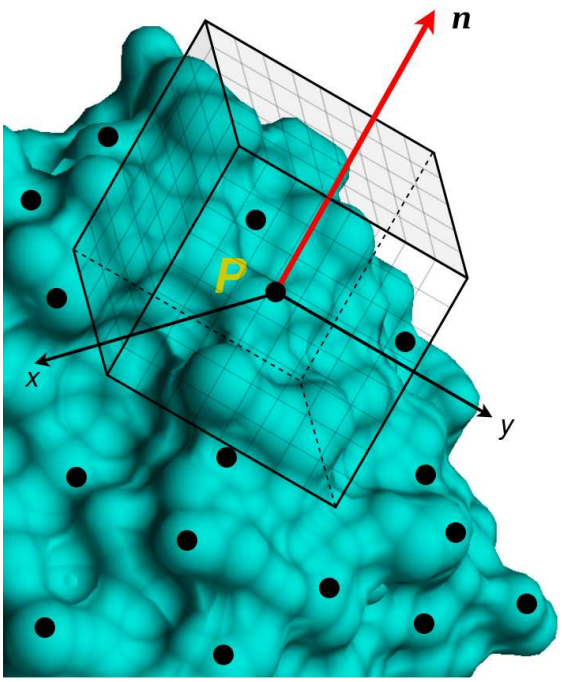


Figure 1: Illustration of 3D grid localized on surface point $P$ and aligned according to normal $n$.

a 4D tensor and returns as output a float value in range [0,1] can be used instead. The exact network architectures employed in our experiments are elaborated in the next subsection. After obtaining ligandability scores for all surface points, we need to extract distinct binding sites. Points with score less than $T$ are considered not reliable and are discarded, while the remaining ones are clustered using the mean-shift algorithm [26]. The main reason for selecting mean-shift instead of other clustering algorithms, is that with mean-shift we do not need to declare the number of clusters in advance. This property matches exactly to our case, since the exact number of binding sites is not known beforehand. Finally, the surface points from each cluster are assigned to their nearest protein atoms and form the desired binding sites.

## 2.2 Network architectures

As previously noted, the 3D-CNN in DeepSurf can be substituted by any 3D convolutional network of user's choice. In this work we adopted the ResNet [27] architecture, which belongs to the family of residual networks. The main attribute of ResNet is the existence of skip connections between adjacent layers, so as to avoid the vanishing gradient problem. The baseline residual block of 3D-ResNet is depicted in Fig. 2(a). ResNet is formed by stacking a number of these blocks. We employed here a 18-layer ResNet, with the exact structure being shown in the original work [27]. Considering the fact that we are employing 3D convolutions, the number of parameters in 3D-ResNet can be dramatically increased compared to 2D-ResNet. In the same work [27], the bottleneck architecture had also been presented, which allows more effective training of deeper ResNets with considerably less parameters per block. Recently, a novel residual network has been proposed, called LDS-ResNet [28], that has shown better performance than the baseline ResNet in computer vision tasks. Notably, LDS-ResNet acquired its best results when combined with a bottleneck architecture, which significantly surpassed all the non-bottleneck variants. In this work, we implemented a 3D variant of the bottleneck LDS-ResNet, with its main block depicted in Fig. 2(b). The difference to Fig. 2(a) is the addition of a second branch with an LDS module parallel to the original convolutional branch and the subsequent concatenation of these two branches. In the following subsection, the extension of the LDS module in three dimensions, which is proposed in this paper, is illustrated.

### 2.2.1 Bottleneck 3D-LDS-Resnet

LDS-ResNets were inspired by the Linear Dynamical Systems theory where a dynamical system is modeled through two time-evolving stochastic processes. The first process estimates a hidden state vector $h_t$ and the second one provides the observed output $y_t$ as a function of this hidden state. A similar approach was adopted in the LDS-module proposed by [28], with the exclusion of the time-evolution factor. The herein

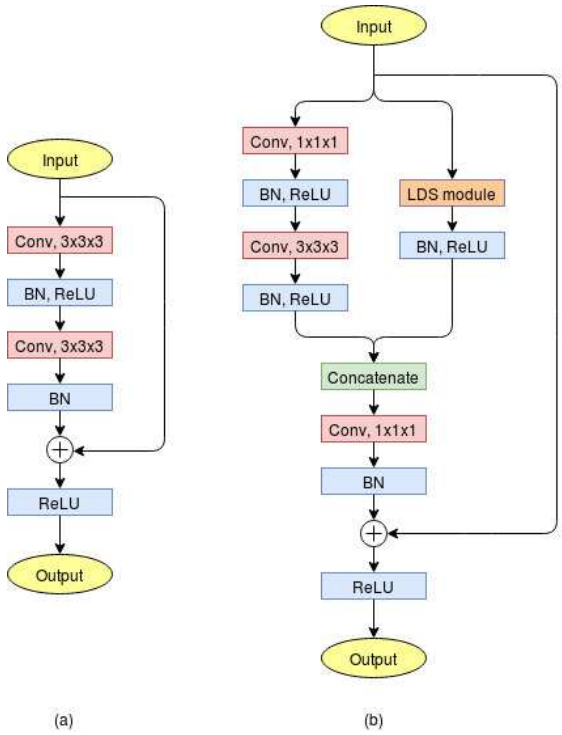


Figure 2: Baseline blocks for (a) original 3D-ResNet, (b) bottleneck 3D-LDS-ResNet

proposed 3D variant of this module is illustrated in Fig. 3. Let us assume the input $X$ to the module is a 4D tensor of size $h \times w \times d \times d_{in}$, where $h$, $w$, $d$ are the spatial dimensions and $d_{in}$ is the number of channels. For clarification reasons, the operation is presented in Fig. 3 just for one channel ($d_{in} = 1$). The LDS module operates iteratively over $X$ on 4D patches $X_t \in R^{n \times n \times n \times d_{in}}$ (in our experiments we used $n = 3$). The calculation of the LDS modules output $Y_t$ involves two main steps.

The first one simulates the hidden state calculation of the LDS theory. Each patch $X_t$ is unfolded to a 2D matrix $x_t \in R^{n^3 \times d_{in}}$ and the hidden state $h_t \in R^{n^3 \times d_{in}}$ is obtained by:

$$h_t = Ax_t \tag{1}$$

where $A \in R^{n^3 \times n^3}$ is the hidden state transition matrix. Its values are randomly initialized for each layer and subsequently optimized during training through backpropagation. Then, $h_t$ is folded back to $H_t \in R^{n \times n \times n \times d_{in}}$ and for every $t$ these subvolumes are stored successively without overlapping, resulting in the intermediate volume of $H \in R^{nh \times nw \times nd \times d_{in}}$.

The second step of the module performs the mapping from the hidden state $h_t$ to output $y_t$, as in original LDS theory. Specifically,

$$y_t = f(W, h_t) \tag{2}$$

where $f()$ is a non-linear function with learnable parameters $W$, and is implemented here by a convolutional operation. Volume $H$ is convolved with a set of $d_{out}$ filters $W \in R^{n \times n \times n \times d_{in}}$ with a stride $k \cdot n$ in each spatial dimension in order to align the filters $W$ with the regions corresponding to each of the patches $X_t$. Factor $k$ controls the downsampling rate of the specific building block. When $k = 1$, the output $Y$ of the LDS module is a $h \times w \times d \times d_{out}$ tensor, while $k > 1$ downsamples all spatial dimensions of the input tensor by a factor of $k$.

# 3 Materials

The demonstrated efficiency of deep neural networks on many research fields lies greatly on the exploitation of large amount of qualitative and properly labeled data that can be used for training. The largest and most suitable database currently available for the BSP task is the scPDB database [29], a continuously updated collection of ligandable binding sites of the Protein Data Bank. These binding sites are defined from complexes between a protein and a pharmacological ligand. One asset of scPDB is that, beyond the atom-based description of the protein and its ligand, it provides also their binding site, being thus suitable for a robust comparison and assessment of the examined methods. We utilized the 2017 release of the database which comprises 16034 entries corresponding to 4782 proteins with 17594 total binding samples. After removing some entries due to failure in reading or in feature extraction, the final dataset contains 15182 structures. For training and validation purposes, the remaining structures were split to

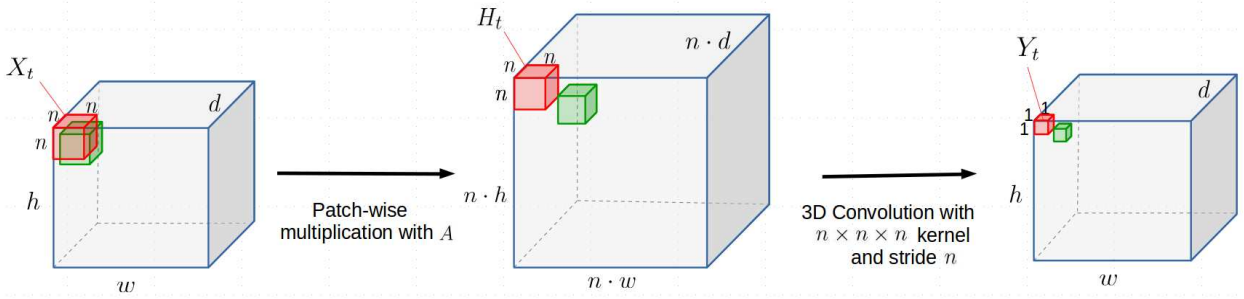


Figure 3: Graphical illustration of LDS-module's operation in three dimensions. For presentation reasons, only one input and output channel is considered ($d_{in} = d_{out} = 1$).

5 folds according to their Uniprot IDs, so as structures from the same protein should be included in the same fold. This separation ensures that the same protein pockets does not coexist in the training and testing set of a split, allowing a more robust and fair assessment.

For testing purposes, two different datasets were used, namely the COACH420 and HOLO4K. These datasets were employed in the evaluation of the P2Rank method [14] and the corresponding authors provided them freely. COACH420 has been derived from the COACH test set [12] and consists of 420 single chain structures containing a mix of drug targets and naturally occurring ligands. HOLO4K is a larger dataset (4009 structures) containing larger multi-chain structures and was initially utilized by [30].

# 4 Implementation

The solvent accessible surface (SAS) of proteins is calculated by the DMS software. Besides the molecular surface, DMS also returns the normal vectors at each surface point. Despite setting DMS to create surfaces with the lowest possible density, the returned set of points is still quite dense, with the average minimum distance between neighboring points being 0.7 Å. Parameter $f$, that controls the subsequent simplification process, should be set in a way that achieves a compromise between losing valuable surface information and avoiding excessive computational cost. In our case, we chose a value of $f = 10$, which raised the average minimum distance of the remaining surface points to 2.3 Å.

Prior to importing in DeepSurf, proteins should also be properly pre-processed. Specifically, water, ions and ligands are removed from the PDB structures, and the remaining structure is protonated, if needed. Before the final step of binding sites extraction (step 12 in Algorithm 1), hydrogen atoms are removed from the protein in order binding sites to maintain only heavy atoms.

As previously stated, BSP is treated here as a binary classification problem, where the two considered classes are the "binding" and "non binding" ones. Therefore, the samples used for training of the 3D-CNN should belong to one of these classes. For each protein of scPDB, surface points that are within 4 Å distance from any ligand atom are considered as binding samples, while all the rest ones are non-binding samples. In this case, the resulting dataset would be quite imbalanced, since the non-binding samples outnumber by far the binding ones.The class imbalance problem is a well-known problem in machine learning applications and a number of tactics have been proposed to tackle with it [31]. The most common tactic lies on the data level and consists of either undersampling the main class or oversampling the secondary one. Due to the required time efficiency during training, the former technique was herein fol-

lowed. For each protein, from the set of non-binding samples a number equal to the binding samples was randomly chosen in order to obtain a 50/50 balance between the two classes.

DeepSurf was implemented in Python and the Tensorflow framework was employed for the deep learning operations. As already shown in Fig. 3, the LDS module consists of two layers: a custom layer and a 3-D convolutional layer. The custom layer consists of the transition matrix $A$ calculation and the patch level multiplication (1). Since this is a patch-based iterative operation, like convolution, it can become extremely computationally heavy for larger input sizes. For this reason, the custom layer was implemented in CUDA to enable high level of parallelization and was, afterwards, integrated in Tensorflow. The source code of the method along with trained models are available at https://github.com/stemylonas/DeepSurf.git.

Regarding the training process, L2 regularization was applied on the weights of all convolutional layers ($\lambda = 10^{-4}$), while batch normalization was applied with its default parameters. All models were trained for 20 epochs, with batch size of 64 samples, and were optimized by the Adam optimizer [32] with a learning rate of $10^{-3}$.

# 5 Results and Discussion

The evaluation criteria used to assess the performance of the proposed method are the following:

- **DCC:** Success rate (%) when considering the distance between the predicted pocket center and the real binding site center. Distances less than 4 Å are successful.
- **DCA:** Success rate (%) when considering the distance between the predicted pocket center and the closest ligand atom. Distances less than 4 Å are successful.
- **OVR:** Overlapping criterion on the atom level defined as the intersection of the real and predicted binding sites divided by their union.

Table 1: Contribution evaluation of the core parts of DeepSurf. Results depicted are average cross-validation performances on scPDB dataset using the DCC criterion.

| | Top-n | Top-(n+2) |
|---|---|---|
| DeepSite's network | 62.1 | 64 |
| ResNet-18 (w/o align) | 66.8 | 69.1 |
| ResNet-18 (w align) | 68.1 | 70.4 |

The DCC and DCA metrics have been widely used in previous works to evaluate the localization quality of extracted binding sites by measuring their distance from either the annotated binding site or the corresponding ligand. On the other hand, OVR differs from the above distance-based metrics by considering also the shape of the binding sites, since it expresses a normalized spatial overlap between the predicted and the actual location of the binding pocket. In the following experiments, the DCC metric is used for evaluating the performance on scPDB, while the DCA and OVR metrics are employed for the comparative assessment on COACH420 and HOLO4K. In all cases, the top-n and top-(n+2) predicted pockets are considered, where $n$ is the number of ligands for the specific protein. Finally, the ligandability threshold $T$ is set to 0.9 in all experiments. The sensitivity of DeepSurf on selection of $T$ is examined more thoroughly in Section 5.3.

## 5.1 Cross-validation on scPDB

The first stage of our experiments consists of the 5-fold cross validation (CV) on scPDB. The goal of this experiment is twofold. Firstly, we would like to evaluate independently some fundamental steps of our method and, secondly, to test the behavior of DeepSurf with residual architectures of different size and type. As described in Section 3, the scPDB dataset was split to five folds and for each fold a different model was trained. Results depicted in both Tables 1 and 2 are average performances on these folds.

Initially, in order to estimate the separate contribution of some core parts of DeepSurf, we conducted two additional experiments. In the first one,

Table 2: Evaluation of DeepSurf with lightweight residual architectures. Results depicted are average cross-validation performances on scPDB dataset using the DCC criterion.

| | #params(M) | Top-n | Top-(n+2) |
|---|---|---|---|
| ResNet-18 | 33.1 | 68.1 | 70.4 |
| Bottleneck ResNet-18 | 1.7 | 66.8 | 69.4 |
| Bottleneck ResNet-34 | 2.9 | 67.6 | 69.9 |
| Bottleneck LDS-ResNet-18 | 3 | 68.3 | 70.8 |

we employed DeepSurf with the exact network architecture used in DeepSite [21] in order to demonstrate the added value of our proposed representation. In the second experiment, we examined the contribution of the surface grid alignment step by training our method without this feature. The obtained results from both experiments are depicted in Table 1, along with the performance obtained by DeepSurf using the ResNet-18 architecture. Although, in the original work of DeepSite, there is no exact reference to the obtained CV performance, we derived from the provided results that the average CV performance of DeepSite is about 50%. Here, DeepSurf when using the same network architecture achieves a clearly better performance of 62.1% proving thus the effectiveness of the surface-based representation in comparison to the strict voxelization of the entire protein used in DeepSite. When the smaller network of DeepSite is replaced by the larger ResNet-18 architecture, we observe that DeepSurf achieves a higher top-n prediction score of 68.1%. This finding indicates, as expected, that larger architectures are able to exploit more efficiently the large amount of training data and can provide much better generalization accuracies. When omitting the surface grid alignment step, the average performance drops by 1.3% in both top-n and top-(n+2) accuracies. This decrease is an indication that this feature, with no additive cost, can boost the overall performance of DeepSurf.

In the next experiment, we employed some alternative lightweight residual architectures in place of ResNet-18 in order to investigate whether lighter networks can achieve similar performances. The obtained results, along with the number of parameters of each network, are summarized in Table 2. As already stated in Section 4, the number of parameters in 3D-ResNets can be quite large due to the 3D convolutions, e.g. ResNet-18 has about 33 million parameters. When replacing the basic residual block with the respective bottleneck (Fig. 2), the resulted network has significantly less parameters (1.7 million), yet it leads to an expected drop in accuracy of 1-1.3%. The addition of LDS block (bottleneck LDS-ResNet-18), which was detailed in Section 2.2.1, led to a rise of 1.5% in accuracy comparing to bottleneck ResNet-18, but at the cost of approximately double parameters. For a fairer assessment of the contribution of the LDS block, we tested DeepSurf employed with a bottleneck ResNet with 34 layers, which has the same parameters as the bottleneck LDS-ResNet-18. We notice that the LDS variant achieved 0.7% higher top-n accuracy and 0.9% higher top-(n+2) accuracy than bottleneck Resnet-34, making it preferable as a lightweight architecture. Comparing now to the baseline ResNet-18 variant, although bottleneck LDS-ResNet-18 has more than 10 times fewer parameters, it achieves similar, if not better, CV performance than its competitor.

## 5.2 Comparison to other methods

After evaluating the individual features of our method through cross-validation, we perform comparison of DeepSurf to other competing in the BSP task deep learning methods that are publicly available. Specifically, we perform comparison to DeepSite [21], Jiang's method [22] and Kalasanty [23]. From the various architectures of DeepSurf tested in Section 5.1, we keep for comparison the baseline ResNet-18 and the lightweight bottleneck LDS-ResNet-18, which provided the highest accuracies.

Table 3: Performance comparison of DeepSurf with competing deep learning methods using the DCA criterion.

| | COACH420 | | HOLO4K | |
|---|---|---|---|---|
| | Top-n | Top-(n+2) | Top-n | Top-(n+2) |
| DeepSite [21] | 57.5 | 65.1 | 45.6 | 48.2 |
| Jiang *et al.* [22] | 55 | 58.7 | 38.2 | 41.5 |
| Kalasanty [23] | 68 | 70.4 | 32.1 | 32.3 |
| DeepSurf (ResNet-18) | 71.9 | 72.3 | 50.7 | 51.1 |
| DeepSurf (Bot-LDS-ResNet-18) | 71.3 | 72.9 | 50.4 | 50.9 |

Table 4: Performance comparison of DeepSurf and Kalasanty using the OVR criterion, computed only for correctly located binding sites ($DCA < 4$ Å).

| | COACH420 | HOLO4K |
|---|---|---|
| Kalasanty | 0.21 | 0.15 |
| DeepSurf (ResNet-18) | 0.29 | 0.17 |
| DeepSurf (Bot-LDS-Res-18) | 0.28 | 0.16 |

For testing purposes, two different datasets were utilized, namely the COACH420 and HOLO4K datasets (for more details see Section 3). In order to avoid data leakage, all proteins from COACH420 and HOLO4k were removed from scPDB, and the remaining dataset was used to train the two variants of DeepSurf. Although all of the competing methods have been trained on the same database (scPDB), any proteins common to our testing datasets have not been removed. This means that these methods have a slight advantage due to this specific data leakage. Provided DeepSite results are those obtained by [14]. In case that a method fails to produce any binding site, an adequately large value of DCA is assigned for each ligand of this protein ensuring that this solution will be regarded erroneous.

The obtained DCA performances are shown in Table 3. Regarding the three competing methods, we notice that Kalasanty surpasses clearly the others in COACH420, while DeepSite is by far superior in the most challenging dataset of HOLO4K. Nevertheless, DeepSurf clearly outperforms all competing methods in both datasets. Specifically, DeepSurf is superior to Kalasanty in COACH420 by 3.7% in top-n accuracy and 2% in top-(n+2), while in HOLO4K, DeepSurf outperforms DeepSite by 5% in top-n accuracy and 3% in top-(n+2). Among the two DeepSurf alternatives, ResNet-18 achieves less than 0.5% higher top-n accuracies in both datasets, while bottleneck LDS-ResNet-18 prevails in terms of top-(n+2) accuracy in COACH420. This indicates the computational and generalization effectiveness of the LDS-equipped network when applied to unknown structures, since it achieves similar results to ResNet-18 but with the benefit of more than 10 times fewer parameters.

For a more comprehensive comparison of the above methods, an overlapping criterion should also be applied that evaluates the shape of the extracted pockets. According to [33] and [34], binding sites are defined as the non-hydrogen atoms of a residue that are within 4 Å to a non-hydrogen atom of the ligand. Following this principle, we extracted binding sites for all proteins in COACH420 and HOLO4K and computed the OVR values only for the correctly located binding pockets ($DCA < 4$ Å) in each corresponding case. The obtained average values are presented in Table 4. Except from DeepSurf, Table 4 holds also the average OVR values obtained by Kalasanty, since it is the only competing method that, additionally to centers, returns explicitly the binding site atoms. As we can see, DeepSurf achieves higher overlapping values in both datasets, and especially in COACH420. Nevertheless, the attained values, mainly in HOLO4k, are relative small compared to the ideal score of 1, indicating that the extraction of properly shaped binding sites is still an open issue.

A more qualitative assessment of the competing

Table 5: Qualitative comparison of DeepSurf and the competing methods. The number of proteins where each method failed and the average number of predicted pockets are shown.

| | Number of failed proteins | | Average number of predicted pockets | |
|---|---|---|---|---|
| | COACH420 | HOLO4K | COACH420 (1.2) | HOLO4K (2.8) |
| DeepSite | 3 | 21 | 3.2 | 2.8 |
| Jiang *et al.* | 12 | 65 | 1.4 | 3.4 |
| Kalasanty | 16 | 475 | 1.1 | 1.2 |
| DeepSurf (ResNet-18) | 10 | 9 | 1.1 | 1.8 |
| DeepSurf (Bot-LDS-ResNet-18) | 11 | 24 | 1.1 | 1.7 |

methods is given in Table 5, which provides the average number of predicted pockets along with the number of proteins where each method failed to produce a result. As we can see, Kalasanty was unable to extract binding sites for a large number of proteins, even after adjusting its default parameters. For example, in the case of HOLO4K, no binding site returned for 475 out of 4009 proteins. Among the two DeepSurf variants, ResNet-18 appeared more robust, since it encountered the fewer failures in case of HOLO4K. Regarding the number of extracted binding sites, DeepSurf and Kalasanty have the tendency to return fewer pockets than DeepSite and Jiang's method in both datasets. In the case of COACH420, it is beneficial since both methods extract a number of pockets similar to the average number of true ones (1.2). On the other hand, in HOLO4K, DeepSurf, and especially Kalasanty, return on average fewer binding sites than the actual ones (2.8). This can explain the larger differences between top-n and top-(n+2) accuracies observed in the case of DeepSite and Jiang's method compared to the rest of the methods (see Table 3).

### 5.3 Sensitivity on ligandability threshold

A key parameter of DeepSurf is the ligandability threshold $T$ above which surface points are considered potential binders. In the above experiments, a high ligandability threshold of $T = 0.9$ was selected so as to preserve only the most reliably assigned points. This choice was based both on the observation that the employed 3D networks return a large number of high ligandability scores and on visual inspection of the extracted pockets. An illustrative example is given in Fig. 4, which displays the binding sites extracted by DeepSurf for structure 1lqdB with $T = 0.5$ and $T = 0.9$, respectively. Although, in both cases, the extracted pockets are considered successful due to low DCA values, we can observe that the extracted pocket in Fig. 4(a) is larger and expands to undesired areas (marked with black circles) away from the ligand. This is totally expected/reasonable, since a smaller value of $T$ leads to the preservation of more surface points before clustering and, subsequently, to the formation of larger binding sites. Furthermore, the influence of varying $T$ in the obtained results is examined quantitatively in Table 6, where the DCA values for both DeepSurf variants and for various ligandability thresholds are presented. As we can see, lower values of $T$ lead consistently to smaller performances in both datasets, especially when $T = 0.5$. Among the two DeepSurf variants, the lighter one seems to be more affected by variations of $T$ (2% drop in COACH420 and 3% in HOLO4K). From the aforementioned, it is concluded that DeepSurf exhibits its optimal performance when a high ligandability threshold is set.

## 6 Conclusion

In this paper, a novel method, called DeepSurf, was presented for predicting potential druggable sites on proteins. The identification of promising candidate

Table 6: Performance of DeepSurf using the DCA criterion for different values of $T$.

| | COACH420 | | | HOLO4K | | |
|---|---|---|---|---|---|---|
| Threshold $T$ | 0.5 | 0.7 | 0.9 | 0.5 | 0.7 | 0.9 |
| DeepSurf (ResNet-18) | 71.5 | 71.7 | 71.9 | 48.9 | 50.1 | 50.7 |
| DeepSurf (Bot-LDS-ResNet-18) | 69.4 | 71.3 | 71.3 | 47.3 | 49.2 | 50.4 |

areas for binding on a protein's surface plays an important role towards drug discovery. DeepSurf proposes a novel approach on this task by combining a surface representation of the protein with a set of local 3D voxelized grids placed on protein's surface. After computing appropriate chemical features, these grids are iteratively imported to a state-of-the-art 3D convolutional network and the resulted ligandability scores of each surface point are, finally, clustered to create the binding sites.

After comparing the proposed method with a set of competing deep learning methods in two diverse datasets, DeepSurf proved quite effective by outperforming all the competing methods in terms of both localization and overlapping accuracies. Despite its domination on overlapping accuracies, the attained values remained quite small proving that the extraction of ideally shaped binding sites still remains an open issue. Finally, a sensitivity analysis on ligandability threshold showed that DeepSurf needs to preserve only the most reliably assigned by the network points in order to acquire its maximum performance.

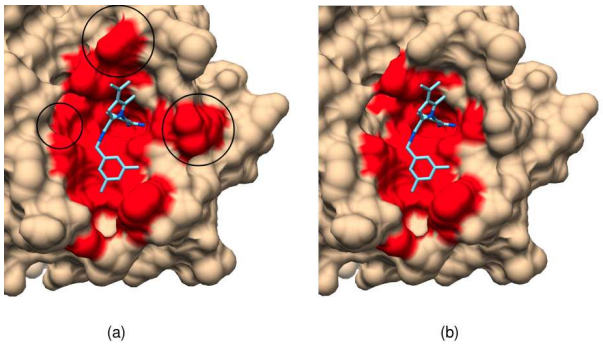


Figure 4: Binding site extraction example for structure '1lqdB' with ligandability threshold (a) $T = 0.5$ and (b) $T = 0.9$. Black circles point the areas where the two results differ.

## Funding

The work has been supported by the ATXN1-MED15 PPI project funded by the GSRT - Hellenic Foundation for Research and Innovation.

## References

[1] Gabriele Macari, Daniele Toti, and Fabio Polticelli. Computational methods and tools for binding site recognition between proteins and small molecules: from classical geometrical approaches to modern machine learning strategies. *Journal of Computer-Aided Molecular Design*, 33(10): 887–903, 2019.

[2] John A Capra, Roman A Laskowski, Janet M Thornton, Mona Singh, and Thomas A Funkhouser. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3d structure. *PLoS computational biology*, 5(12):e1000585, 2009.

[3] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, 10(1):168, 2009.

[4] Sérgio ED Dias, Quoc T Nguyen, Joaquim A Jorge, and Abel JP Gomes. Multi-gpu-based detection of protein cavities using critical points. *Future Generation Computer Systems*, 67:430–440, 2017.

[5] Chi-Ho Ngan, David R Hall, Brandon Zerbe,

Laurie E Grove, Dima Kozakov, and Sandor Vajda. Ftsite: high accuracy detection of ligand binding sites on unbound protein structures. *Bioinformatics*, 28(2):286–287, 2011.

[6] Pradeep Anand Ravindranath and Michel F Sanner. Autosite: an automated approach for pseudo-ligands predictionfrom ligand-binding sites identification to predicting key ligand atoms. *Bioinformatics*, 32(20):3142–3149, 2016.

[7] Hiroto Tsujikawa, Kenta Sato, Cao Wei, Gul Saad, Kazuya Sumikoshi, Shugo Nakamura, Tohru Terada, and Kentaro Shimizu. Development of a protein–ligand-binding site prediction method based on interaction energy and sequence conservation. *Journal of structural and functional genomics*, 17(2-3):39–49, 2016.

[8] Michal Brylinski and Jeffrey Skolnick. A threading-based method (findsite) for ligand-binding site prediction and functional annotation. *Proceedings of the National Academy of sciences*, 105(1):129–134, 2008.

[9] Howook Hwang, Fabian Dey, Donald Petrey, and Barry Honig. Structure-based prediction of ligand–protein interactions on a genome-wide scale. *Proceedings of the National Academy of Sciences*, 114(52):13685–13690, 2017.

[10] Daniele Toti, Le Viet Hung, Valentina Tortosa, Valentina Brandi, and Fabio Polticelli. Libra-wa: a web application for ligand binding site detection and protein function recognition. *Bioinformatics*, 34(5):878–880, 2017.

[11] Zengming Zhang, Yu Li, Biaoyang Lin, Michael Schroeder, and Bingding Huang. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics*, 27(15):2083–2088, 2011.

[12] Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

[13] Jhih-Wei Jian, Pavadai Elumalai, Thejkiran Pitti, Chih Yuan Wu, Keng-Chang Tsai, Jeng-Yih Chang, Hung-Pin Peng, and An-Suei Yang. Predicting ligand binding sites on protein surfaces by 3-dimensional probability density distributions of interacting atoms. *PloS one*, 11(8): e0160315, 2016.

[14] Radoslav Krivák and David Hoksza. P2rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10 (1):39, 2018.

[15] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein–ligand scoring with convolutional neural networks. *Journal of chemical information and modeling*, 57(4):942–957, 2017.

[16] Fergus Imrie, Anthony R Bradley, Mihaela van der Schaar, and Charlotte M Deane. Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data. *Journal of chemical information and modeling*, 58(11):2319–2330, 2018.

[17] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.

[18] José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2): 287–296, 2018.

[19] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6:18962, 2016.

[20] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green,

Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (casp13). *Proteins: Structure, Function, and Bioinformatics*, 87 (12):1141–1148, 2019.

[21] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S Rose, and Gianni De Fabritiis. Deepsite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

[22] Mingjian Jiang, Zhen Li, Yujie Bian, and Zhiqiang Wei. A novel protein descriptor for the prediction of drug binding sites. *BMC bioinformatics*, 20(1):1–13, 2019.

[23] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Detection of protein-ligand binding sites with 3d segmentation. *arXiv preprint arXiv:1904.06517*, 2019.

[24] Mingjian Jiang, Zhiqiang Wei, Shugang Zhang, Shuang Wang, Xiaofeng Wang, and Zhen Li. Frsite: Protein drug binding site prediction based on faster r–cnn. *Journal of Molecular Graphics and Modelling*, 93:107454, 2019.

[25] Apostolos Axenopoulos, Dimitrios Rafailidis, Georgios Papadopoulos, Elias N Houstis, and Petros Daras. Similarity search of flexible 3d molecules combining local and global shape descriptors. *IEEE/ACM transactions on computational biology and bioinformatics*, 13(5):954–970, 2015.

[26] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(5):603–619, 2002.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Anastasios Dimou, Dimitrios Ataloglou, Kosmas Dimitropoulos, Federico Alvarez, and Petros Daras. Lds-inspired residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[29] Jérémy Desaphy, Guillaume Bret, Didier Rognan, and Esther Kellenberger. sc-pdb: a 3d-database of ligandable binding sites10 years on. *Nucleic acids research*, 43(D1):D399–D404, 2014.

[30] Peter Schmidtke, Catherine Souaille, Frédéric Estienne, Nicolas Baurin, and Romano T Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of chemical information and modeling*, 50(12):2191–2200, 2010.

[31] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Nicholas M Luscombe, Roman A Laskowski, and Janet M Thornton. Amino acid–base interactions: a three-dimensional analysis of protein–dna interactions at an atomic level. *Nucleic acids research*, 29(13):2860–2874, 2001.

[34] Ke Chen, Marcin J Mizianty, Jianzhao Gao, and Lukasz Kurgan. A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. *Structure*, 19(5):613–621, 2011.