nature methods

ARTICLES

https://doi.org/10.1038/s41592-019-0666-6

Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning

P. Gainza[1], F. Sverrisson[1], F. Monti[2,3], E. Rodolà[4], D. Boscaini[5], M. M. Bronstein[2,3,6] and B. E. Correia[1*]

Predicting interactions between proteins and other biomolecules solely based on structure remains a challenge in biology. A high-level representation of protein structure, the molecular surface, displays patterns of chemical and geometric features that fingerprint a protein's modes of interactions with other biomolecules. We hypothesize that proteins participating in similar interactions may share common fingerprints, independent of their evolutionary history. Fingerprints may be difficult to grasp by visual analysis but could be learned from large-scale datasets. We present MaSIF (molecular surface interaction fingerprinting), a conceptual framework based on a geometric deep learning method to capture fingerprints that are important for specific biomolecular interactions. We showcase MaSIF with three prediction challenges: protein pocket-ligand prediction, protein–protein interaction site prediction and ultrafast scanning of protein surfaces for prediction of protein–protein complexes. We anticipate that our conceptual framework will lead to improvements in our understanding of protein function and design.

Interactions between proteins and other biomolecules are the basis of protein function in most biological processes. Predicting these interactions purely from structure remains one of the most important challenges in structural biology[1–4]. Many programs effectively predict these interactions by exploiting evolutionary signatures in protein sequence and structure[5–7], yet these approaches require the knowledge of homologous proteins. The molecular surface[8] is a higher-level representation of protein structure that models a protein as a continuous shape with geometric and chemical features. We propose that molecular surfaces are fingerprinted with patterns of chemical and geometric features that reveal information about the protein's interactions with other biomolecules. Our central hypothesis is that proteins with no sequence homology that undergo similar biomolecular interactions may display similar patterns, which are difficult to grasp by visual analysis but could be learned from large-scale datasets. Here, we present MaSIF (molecular surface interaction fingerprinting), a general geometric deep learning[9] method to recognize and decipher patterns on protein surfaces, without explicit consideration of the underlying protein sequence or structural fold.

The molecular surface representation describing protein structure (Fig. 1a) has long been used for many tasks involving protein interactions[10,11], and has been the preferred structural description to study protein–solvent electrostatic interactions[12]. More recently, several methods have captured molecular surface patterns with functional relevance, such as three-dimensional (3D) Zernike descriptors[13–16] and geometric invariant fingerprint (GIF) descriptors[17]. These approaches proposed 'handcrafted' descriptors, manually optimized vectors that describe protein surface features. The scope of these approaches is limited as it is hard to determine a priori the right set of features for a given prediction task.

Geometric deep learning[9] is a nascent field extending successful image-based deep neural network architectures, such as convolutional neural networks (CNNs)[18], to geometric data such as surfaces, where these techniques have been shown to substantially outperform handcrafted feature extraction[19,20]. MaSIF exploits geometric deep learning to learn interaction fingerprints in protein molecular surfaces. The molecular surface data is described in geodesic space, meaning that the distance between two points corresponds to the distance of 'walking' between the points along the surface. In highly irregular protein surfaces (for example, with deep pockets), geodesic distances can be much larger than Euclidean distances (Supplementary Fig. 1). First, MaSIF decomposes a surface into overlapping radial patches with a fixed geodesic radius (Fig. 1a,b). Each point within a patch is assigned an array of geometric and chemical input features (Fig. 1b). The input features (chemistry and geometry) are not learned, they are precomputed properties from the molecular surface. MaSIF then learns to embed the surface patch's input features into a numerical vector descriptor (Fig. 1d). Each descriptor is further processed with application-dependent neural network layers. The networks are trained end-to-end, meaning that the intermediate patch descriptors are not universal but rather optimized toward particular tasks.

We showcase MaSIF with three proof-of-concept applications (Fig. 1e): (1) ligand pocket similarity comparison (MaSIF-ligand); (2) protein–protein interaction (PPI) site prediction in protein surfaces (MaSIF-site) and (3) ultrafast scanning of surfaces, where we exploit surface fingerprints to predict the structural configuration of protein–protein complexes (MaSIF-search). Our conceptual framework will be useful for biologists that search for similar interaction fingerprints between proteins with no shared evolutionary ancestry. Crucially, MaSIF represents a departure from learning on Euclidean structural representation and may enable the recognition of important structural features for protein function and design.

MaSIF: a general framework to learn protein surface fingerprints

The MaSIF conceptual framework is shown in Fig. 1 and described in the Methods section. Briefly, from a protein structure we

[1]Institute of Bioengineering, École Polytechnique Fédérale de Lausanne and Swiss Institute of Bioinformatics, Lausanne, Switzerland. [2]Institute of Computational Science, Faculty of Informatics, USI, Lugano, Switzerland. [3]Twitter, London, UK. [4]Department of Computer Science, Sapienza University of Rome, Rome, Italy. [5]Technologies of Vision Unit, Fondazione Bruno Kessler, Trento, Italy. [6]Department of Computing, Imperial College London, London, UK. *e-mail: bruno.correia@epfl.ch

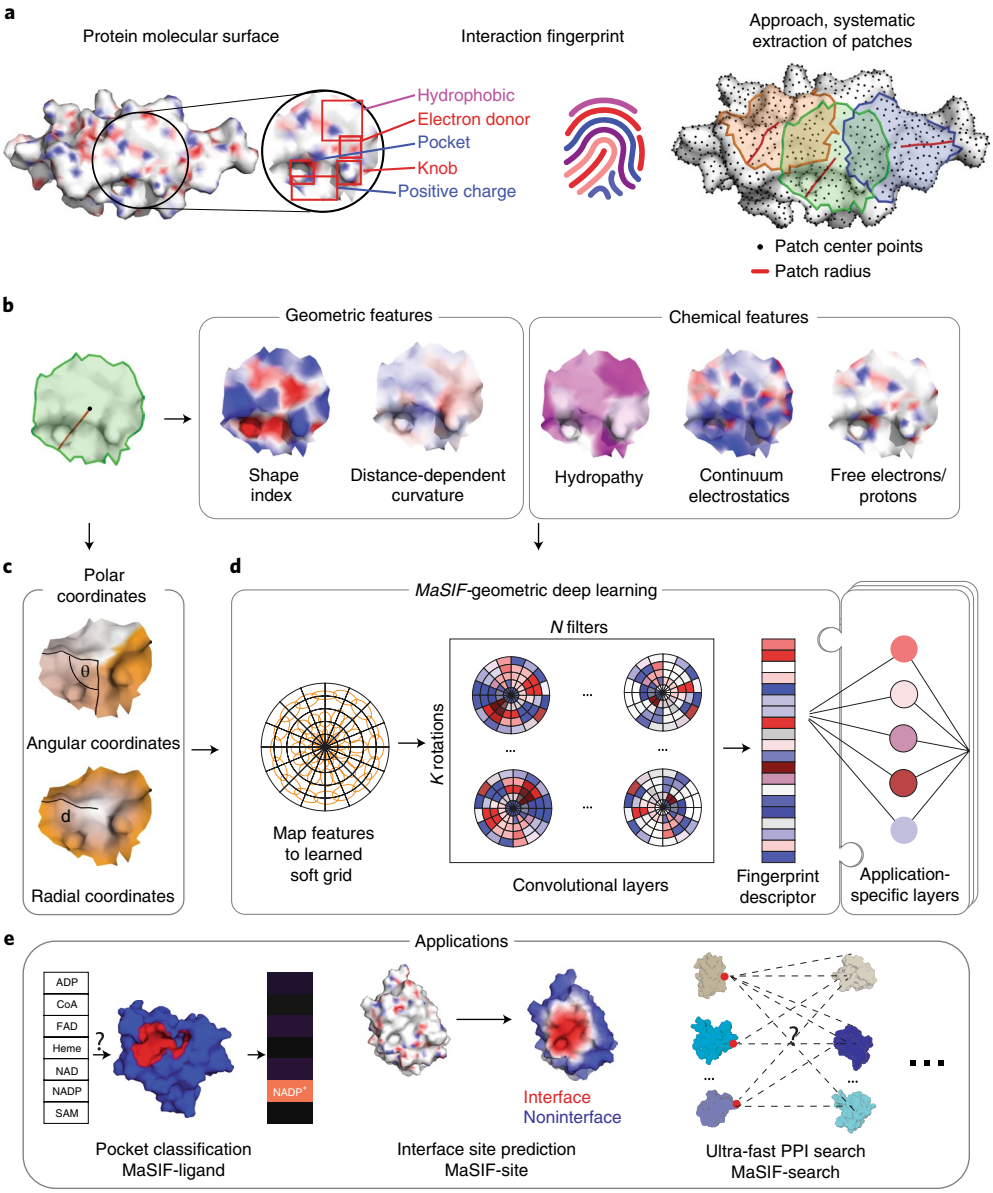

Fig. 1 | Overview of the MaSIF conceptual framework, implementation and applications. a, Left, conceptual representation of a protein surface engraved with an interaction fingerprint, surface features that may reveal their potential biomolecular interactions. Right, surface segmentation into overlapping radial patches of a fixed geodesic radius used in MaSIF. **b**, The patches comprise geometric and chemical features mapped on the protein surface. **c**, Polar geodesic coordinates used to map the position of the features within the patch. **d**, MaSIF uses geometric deep learning tools to apply CNNs to the data. Fingerprint descriptors are computed for each patch using application-specific neural network architectures, which contain reusable building blocks (geodesic convolutional layers). **e**, MaSIF is generalizable and applicable to multiple prediction tasks: a selected few are showcased in this paper.

compute a discretized molecular surface (solvent excluded surface)[21] and assign geometric and chemical features to every point (vertex) in the mesh (Fig. 1a,b). Around each vertex of the mesh, we extract a patch with geodesic radius of $r = 9$ Å or $r = 12$ Å (Fig. 1b). The choice of patch radius is application-dependent, in architectures with multiple geodesic convolutional layers we use a smaller patch size due to memory limitations (see Methods). For each vertex within the patch, we compute two geometric features (shape index[22] and distance-dependent curvature[17]) and three chemical features (hydropathy index[23], continuum electrostatics[24] and the location of free electrons and proton donors[25]). The vertices within a patch are assigned geodesic polar coordinates (Fig. 1c): the radial coordinate, representing the geodesic distance to the center of the patch and the angular coordinate, computed with respect to a random direction from the center of the patch, as the patch lacks a canonical orientation. The geometric structure of the surface (for example, the 'depth' of a pocket within the surface) are implicitly described through the geometric features (shape index and distance-dependent curvature) and the geodesic polar coordinates.

MaSIF applies a geometric deep neural network to these input features using the polar coordinates to spatially localize features. The neural network consists of one or more layers applied sequentially; a key component of the architecture is the geodesic convolution, generalizing the classical convolution to surfaces and implemented as an operation on local patches[20]. In the polar coordinates, we construct a system of Gaussian kernels defined in a local geodesic polar system for which the parameters are learnable.

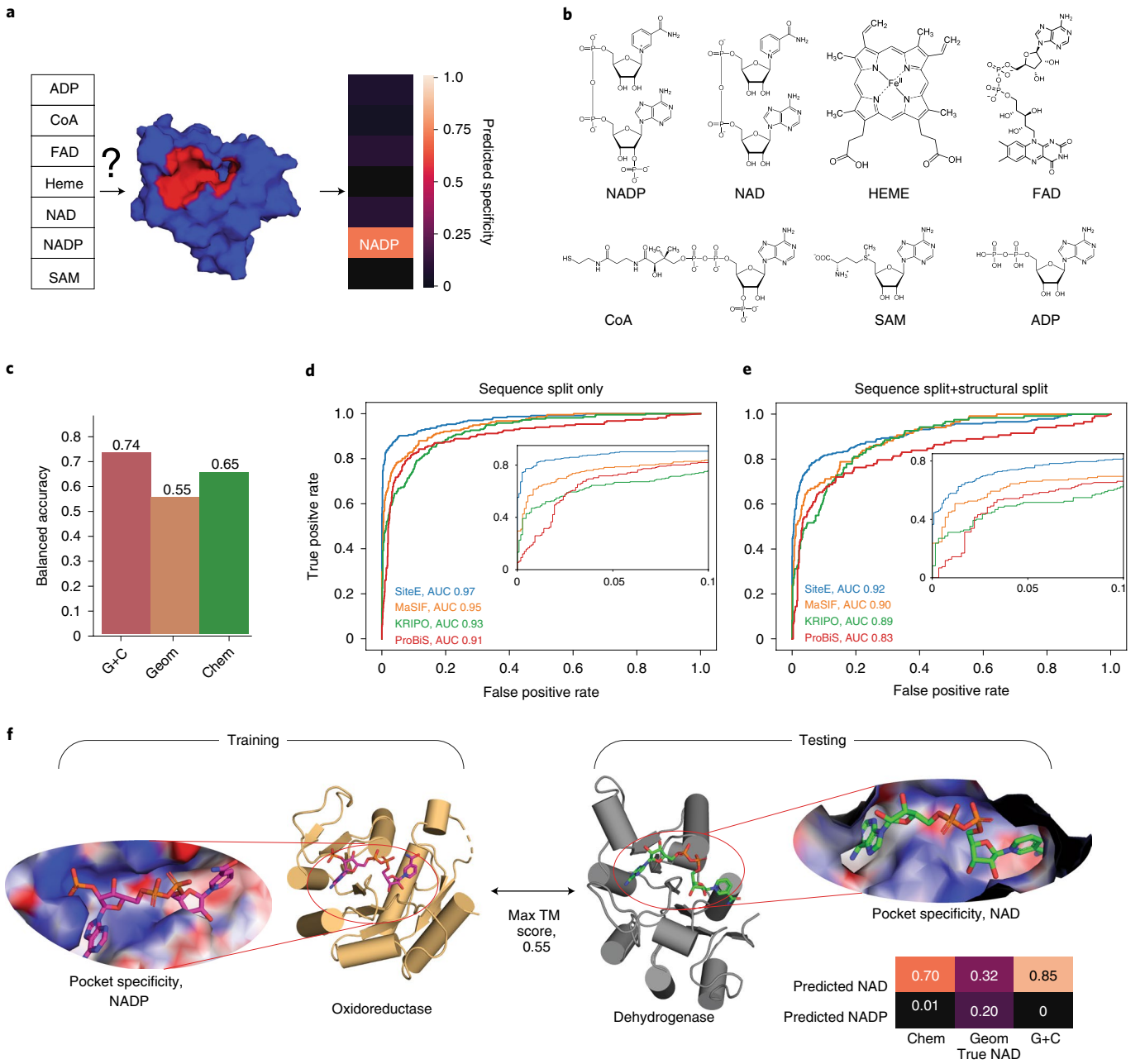

Fig. 2 | Classification of ligand-binding sites using MaSIF-ligand. a, Schematic representation of the prediction task. The neural network receives a protein pocket as input and classifies it into seven categories to reflect the predicted binding preference. **b**, Structures of the seven cofactors that bind proteins considered for the prediction task. **c**, Balanced accuracy of the prediction of the specificity of binding sites using all features (G+C, geometry and chemistry), only geometric features (Geom) or only chemical features (Chem). **d**, ROC curves for comparative benchmarks for pocket classification using the full training and testing sets (excluding HEME, total number of pockets in testing set was 216). **e**, ROC curves for comparative benchmarks using a strict structural split of the pockets between the training and test sets (template modeling (TM) score <0.5, total number of pockets in testing set was 121). **f**, Specific example on a protein fold that recognizes two similar ligands and yet is correctly predicted. A bacterial dehydrogenase in the test set binds to NAD (PDB ID 2O4C)[30], while its closest structural homolog in the training set corresponds to a mammalian oxidoreductase (PDB ID 2YJZ), which binds to NADP[31].

The learnable Gaussian kernels locally average the vertex-wise patch features (acting as soft pixels) and produce an output of fixed dimension, which is correlated with a set of learnable filters[19]. We refer to this family of learnable Gaussian kernels as a learned soft polar grid (see Methods).

A convolutional layer with a set of filters is then applied to the output of the soft polar grid layer. Note that since the angular coordinates were computed with respect to a random direction, it becomes essential to compute information that is invariant to different directions (rotation invariance, Fig. 1d). To this end, we perform K rotations on the patch and compute the maximum over all rotations[20], producing the geodesic convolution output for the patch location. The procedure is repeated for different patch locations similar to a sliding window operation on images, producing the surface fingerprint descriptor at each point in the form of a vector that embeds information about the surface patterns of the center point and its neighborhood. The learning procedure consists of minimizing the parameter set of the local kernels and filter weights with respect to

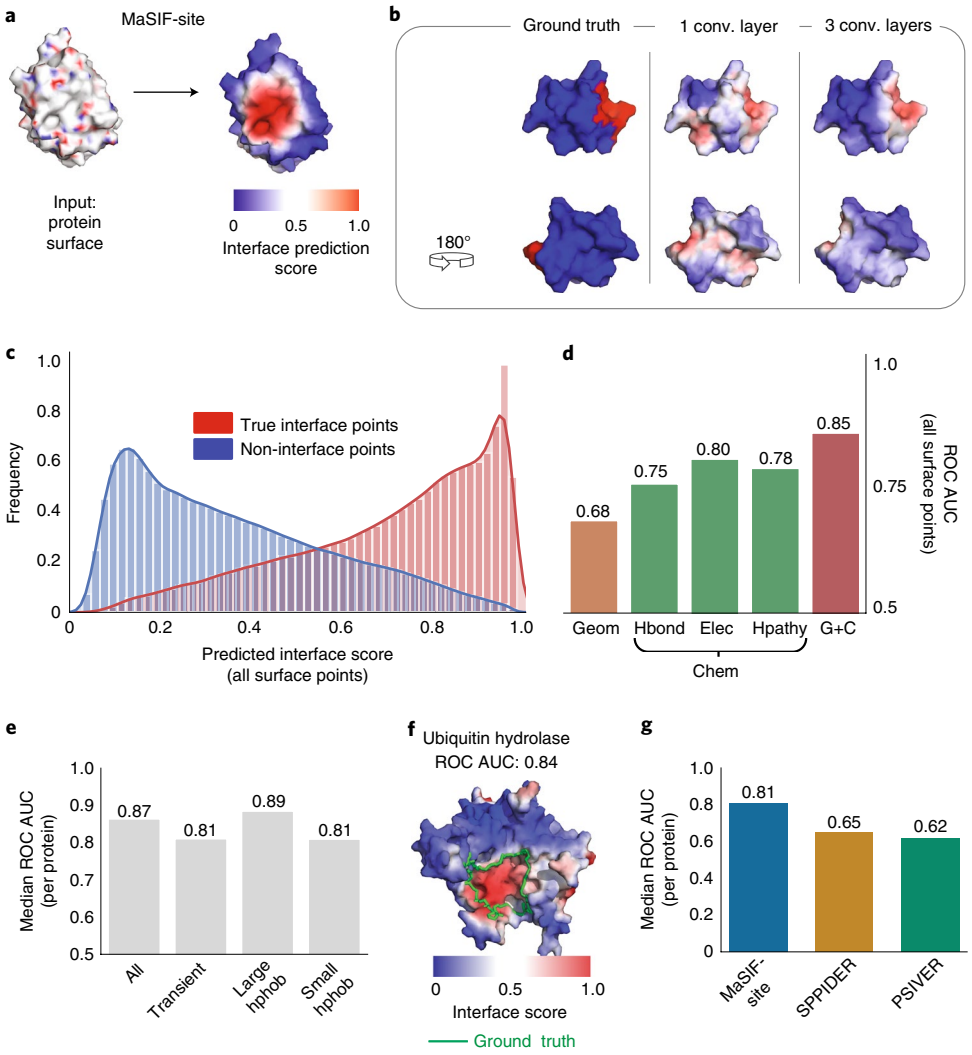

Fig. 3 | Prediction of surface patches involved in PPIs. a, Schematic representation of the interface site prediction workflow. The MaSIF-site receives as input a protein surface with a descriptor vector and outputs a surface score that reflects the predicted interface propensity (red for high interface propensity, blue for low propensity). **b**, Visual comparison between MaSIF-site with a network with one convolutional (conv.) layer versus three convolutional layers. **c**, Distribution of predicted scores for true positives (red) versus true negatives (blue) for a network trained with all features. The ROC AUC values were computed based on the surface points in the proteins of the test set. **d**, ROC AUC scores for ablation studies with networks trained with different subsets of features: only geometric (Geom), only the location of free electrons/proton donors (hbond), Poisson–Boltzmann electrostatics (elec), the hydropathy index (hpathy) and all features (G+C) (surface points, no. of positives = 218,246, no. of negatives = 1,973,624). **e**, Left, median ROC AUCs (per protein) for selected subsets of proteins. All, full test set containing all proteins (361 proteins); Transient, proteins forming known transient interactions (59 proteins); Large hphob, protein complexes with interfaces composed of mostly hydrophobic residues (74 proteins) and Small hphob, protein complexes with small hydrophobic interfaces (74 proteins). **f**, An illustrative example of a protein with a ROC AUC close to the median of 0.84, which is close to the median of MaSIF-site. **g**, Comparison of MaSIF-site with the SPPIDER and PSIVER predictor for a set of 53 single-chain transient interactions. Results are shown as the median ROC AUC per protein, evaluated on a per-residue basis for comparison with the other predictors.

the application-specific training data and cost function. Therefore, the parameter set is specific to each application presented here.

With this framework we created descriptors for surface patches that can be further processed in neural network architectures. Next, we will present various ways to leverage them to identify interaction fingerprints on protein surfaces.

Results

Molecular surface fingerprinting to classify ligand-binding pockets. Interactions between proteins and metabolites play a fundamental role in cellular homeostasis, yet our knowledge of these interactions is extremely limited[26]. We propose that the interaction fingerprints in protein surfaces hold information to decipher the metabolite-binding preference of protein pockets. To test this hypothesis, we developed MaSIF-ligand, a classifier to predict the metabolite-binding preference of a pocket from surface features (Fig. 2a). For this proof-of-concept we used seven cofactors: ADP, NAD, NADP, FAD, *S*-adenosyl methionine (SAM), coenzyme A (CoA) and heme, metabolites with large structural datasets available (Fig. 2b).

We trained MaSIF-ligand on a large set of cofactor-binding proteins using their holo structures, where sequences and structures were clustered to remove redundancy from the training and test sets. The balanced accuracy on an independent test set was used to gauge the classification power of MaSIF-ligand. We first trained MaSIF-ligand with all features (geometry and chemistry) and obtained a balanced accuracy of 0.73 (Fig. 2c) (expected random accuracy, 0.14). To investigate the importance of the features, we

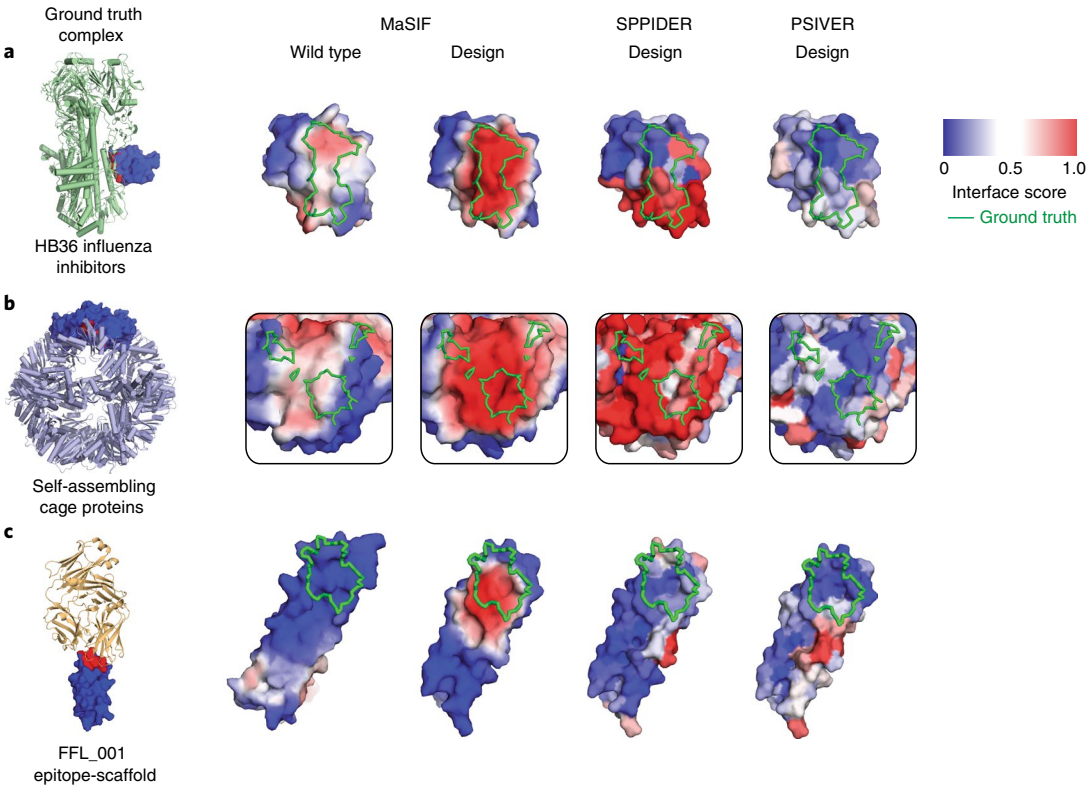

Fig. 4 | Prediction of PPI sites on a set of computationally designed proteins. a, Designed HB36 influenza inhibitors (PDB ID 3R2X) versus the wild-type scaffold protein (PDB ID 1U84). **b**, Designed self-assembling nanocage protein (PDB ID 3VCD) versus the wild-type scaffold (PDB ID 3N79). **c**, Designed respiratory syncytial virus epitope-scaffold (PDB ID 4JLR) versus wild-type scaffold (PDB ID 1ISE). MaSIF-site was tested on a set of de novo computationally designed proteins involved in PPIs, where the prediction on the designed binders was compared to the corresponding native proteins. For comparison, predictions with SPPIDER and PSIVER were generated for the designed proteins (right).

limited the set to geometric or chemical features that reduced the balanced accuracy to 0.55 and 0.65, respectively (Fig. 2c).

Next, we compared MaSIF-ligand with three other programs, ProBiS[27], KRIPO[28] and SiteEngine[10], which exploit structural features for pocket classification, and showed top-tier performance in a recent comprehensive benchmark[29]. To compare the different methods, we use the receiver operator characteristic area under the curve (ROC AUC). In our datasets, SiteEngine is the top performer among these tools, while MaSIF-ligand achieves a better performance than KRIPO and ProBiS (Fig. 2d). Both SiteEngine and MaSIF-ligand identify physicochemical and geometric similarities in molecular surfaces. However, SiteEngine is based on explicit alignments of pockets using pseudo-representations of the molecular surface, which results in a much higher runtime. It is therefore remarkable that MaSIF-ligand can achieve similar performances despite embedding the 3D space into fingerprint descriptors.

To analyze the MaSIF-ligand predictions in detail, we generated a confusion matrix with all features (Supplementary Fig. 2a). We observe variable performances across ligands, perhaps not surprisingly in the case of HEME (accuracy of 94%) given the chemically dissimilarity to the other cofactors. More challenging is the distinction between similar ligands, namely in the analysis of the confusion data between two highly similar cofactors: SAM versus ADP and NADP versus NAD. In both cases, the geometric features are not sufficient and are mainly the chemical features that contribute to the correct predictions (Supplementary Fig. 2a,b). The capacity of MaSIF-ligand to distinguish the features from very similar cofactors is remarkable, especially for NADP versus NAD that differ by a single phosphate group on the adenosine moiety. To understand these successful predictions, we analyzed the pocket features of an NAD-binding bacterial dehydrogenase[30] in our test set and its closest structural homolog in the training set, a mammalian oxidoreductase that binds to NADP (Fig. 2f)[31]. We analyzed the regions of the pockets giving the neural network the highest discrimination score between NAD versus NADP, and mapped this score on the pocket surface (see Methods) (Supplementary Fig. 2c). The largest discrimination scores arise from patches centered around the additional NADP phosphate in the oxidoreductase:NADP pocket, while in the dehydrogenase:NAD pocket, the adenine moiety region, where NAD and NADP differ, is crucial to correctly classify the pocket. The prediction probabilities for the dehydrogenase:NAD pocket are dependent on the chemical features (Fig. 2f, right), further confirmed by the Poisson–Boltzmann electrostatics showing that the oxidoreductase:NADP pocket (Fig. 2f, left) has a stronger positive charge distribution, consistent with its binding to the more negatively charged NADP.

Despite the lack of global sequence homology and structural similarity of the pockets in the test and training sets, MaSIF-ligand can decipher the surface interaction fingerprints to determine the binding preference of each pocket. As illustrated by the NAD/NADP example MaSIF-ligand can infer the correct cofactor in two proteins with the same fold based purely on surface features, without explicit consideration of the underlying amino acids or sequence-based signatures.

Overall, the interaction fingerprints in protein surfaces could be an additional source of information available to biologists to infer important protein:ligand interactions.

Predicting protein binding sites based on interaction fingerprints. Inspired by previous work on PPI site prediction[32–34], we

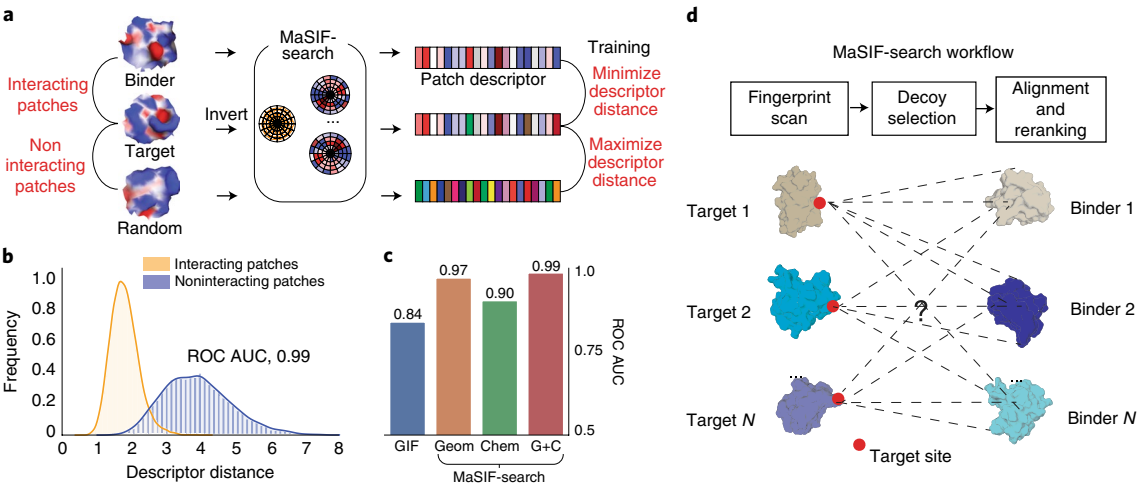

Fig. 5 | Prediction of PPIs based on surface fingerprints. a, Overview of the MaSIF-search neural network optimization (siamese architecture) to output fingerprint descriptors, such that the descriptors of interacting patches are similar, while those of noninteracting patches are dissimilar. The features of the target patch (with the exception of the hydropathy features) are inverted to enable the minimization of the fingerprint distance. **b**, Distribution of fingerprint distances showing interacting (yellow) and noninteracting (blue) patches for the test set (13,338 positive pairs and 13,338 negative pairs). MaSIF-search was trained and tested on both geometric and chemical features. **c**, Comparison of the performance between different fingerprint features shown in ROC AUC (13,338 positive pairs and 13,338 negative pairs from test set). GIF, ROC AUC for GIF fingerprint descriptors[17] Geom, MaSIF-search trained with only geometric features; Chem, MaSIF-search only with chemical features and G+C, geometry and chemistry features. **d**, Top, schematic of MaSIF-search workflow showing the three stages of the protocol and bottom, MaSIF-search benchmarking by performing a large-scale docking of *N* binder proteins to *N* known targets with site information, results of which are shown in Tables 1 and 2.

developed MaSIF-site, a classifier that receives a protein surface as input and outputs a predicted score for each surface vertex on the likelihood of being involved in a PPI (Fig. 3a).

MaSIF-site was trained and tested on a large dataset of protein structures that were cocrystallized in the holo state and separated into monomeric subunits. The training and testing sets were split based on sequence and structure (see Methods). This task greatly leverages the potential of deep learning approaches, since multiple layers yield superior predictions (Fig. 3b). Using one geodesic convolutional layer MaSIF-site's ROC AUC reaches 0.77 (Fig. 3b and Supplementary Fig. 3), while three layers boost the ROC AUC to 0.86, computed over all the surface points of the test set proteins.

A strong separation between the predicted true and false interfaces is observed (Fig. 3c). A feature ablation study showed that the Poisson–Boltzmann continuum electrostatics reached the highest performance (ROC AUC = 0.80) of all single features (Fig. 3d), suggesting an important contribution of electrostatics on the identification of PPI sites.

Surfaces involved in PPIs can be classified according to biophysical (for example, obligate versus transient) and structural/chemical (for example, large versus small, hydrophobic versus polar and so on) properties, we asked whether MaSIF-site had a biased performance for a particular type of surface (Fig. 3e). These predictions were reported in median ROC AUC per protein providing a better assessment of the performance for each query protein. The prediction accuracy for the whole dataset reached a median ROC AUC of 0.87 per protein, while for a subset of transient interactions the ROC AUC was 0.81. Proteins with large hydrophobic interfaces had a better performance (ROC AUC = 0.89) than those with the smallest hydrophobic surfaces (ROC AUC = 0.81). The median ROC AUC value is illustrated with the example of ubiquitin hydrolase (ROC AUC = 0.84), close to the median of the whole dataset (Fig. 3f).

We compared MaSIF-site to top performing predictors[35] SPPIDER[33] and PSIVER[36], in a subset of transient interactions that are likely among the most challenging test cases. MaSIF-site reaches the highest performance, median ROC AUC per protein of 0.81, while SPPIDER and PSIVER reach 0.65 and 0.62, respectively (Fig. 3g). The distribution of ROC AUCs per protein for each method is shown in Supplementary Fig. 4b. We further illustrate MaSIF's superior performances relative to SPPIDER in four randomly chosen proteins from the transient test set (Supplementary Fig. 4c).

Although evolutionary information can be crucial to predict protein interaction sites[36], in some cases such evolutionary history is sparse or completely absent. These extreme cases include computationally designed PPIs, whose interfaces were rationally designed in protein scaffolds. We used MaSIF-site to predict three such designed interfaces that have been experimentally validated: an influenza inhibitor[37] (Fig. 4a), a homo-oligomeric cage protein[38] (Fig. 4b) and an epitope-scaffold used as an immunogen[39] (Fig. 4c). The designs were based on wild-type scaffold proteins with no binding activity, and in each case, we compared their interface score with that of the noninteracting wild type. MaSIF-site clearly labels the interfaces of the designs, in contrast with SPPIDER and PSIVER's predictions. Overall, MaSIF-site may help to identify the sites of interactions with other proteins for PPI validation, paratope/epitope prediction or small molecule binding sites, for cases where evolutionary or experimental information may not be available.

Ultrafast scanning of interaction fingerprints for prediction of protein–protein complexes. As a last example of MaSIF's generality, we show the embedding of fingerprints as vectorized descriptors to predict specific interactions between proteins. This embedding, inspired by earlier work on GIF descriptors[17], is attractive because, once the descriptors are precomputed, nearest-neighbor techniques can scan billions of descriptors per second[40]. The gain in computational cost at runtime enables broad structural searches across large databases, moving away from the model of one binder versus one target, typical of docking programs, to one of many binders versus many targets. This is important for tasks such as protein design, where docking tools are used to search for structural templates to use as starting points for the design of new PPIs or ligand-binding proteins[37,41]. Thus, we introduce MaSIF-search, a method to quickly search protein binding partners based on surface fingerprints. MaSIF-search is then complemented with

Table 1 | Results for large-scale docking benchmark benchmark for PatchDock, MaSIF-search (with multiple numbers of decoys), ZDock and ZDock+ZRank2 on bound (holo) complexes

Method	Number of solved complexes in the top			Time (min)
	100	10	1	
MaSIF-search decoys = 100	37	36	30	4
MaSIF-search decoys = 2,000	67	56	43	39
PatchDock	43	32	21	2,743
ZDock	58	36	18	134,934
ZDock+ZRank2 decoys = 200,000	77	63	45	159,902

No. of solved complexes in the top, number of target–binder complexes within 5 Å iRMSD found in the top 100, top ten or top one (for holo cases) or top 1,000, top 100 and top ten (for apo cases). Time (min), CPU time in minutes for each program, which excludes precomputation time for MaSIF-search.

Table 2 | Results for large-scale docking benchmark benchmark for PatchDock, MaSIF-search (with multiple numbers of decoys), ZDock and ZDock+ZRank2 on unbound (apo) complexes

Method	Number of solved complexes in the top			Time (min)
	1,000	100	10	
MaSIF-search decoys = 2,000	17	7	2	16
PatchDock	11	4	1	560
ZDOCK	17	13	5	13,174
ZDock+ZRank2 decoys = 80,000	23	12	5	16,866

surface alignment and reranking stages to generate docked complexes with improved quality.

MaSIF-search learns patterns in interacting pairs of surface patches. PPIs occur through surface patches with some degree of complementary geometric and chemical features. To formalize this observation, MaSIF-search inverts the numerical features of one protein partner (multiplied by −1), with the exception of hydropathy. Although the models of complementarity are not perfect the network may be able to learn different levels of complementarity. After performing the inversion on one patch, the Euclidean distance between the fingerprint descriptors of two complementary surface patches should be close to zero. Within this framework, MaSIF-search will produce similar descriptors for pairs of interacting patches (low Euclidean distances between fingerprint descriptors), and dissimilar descriptors for noninteracting patches (larger Euclidean distances between fingerprint descriptors) (Fig. 5a). Thus, identifying potential binding partners is reduced to a comparison of numerical vectors.

To test this concept, we assembled a database with >100,000 pairs of interacting protein surface patches with high shape complementarity, as well as a set of randomly chosen surface patches, to be used as noninteracting patches. A trio of protein surface patches with the labels, binder, target and random patches were fed into the MaSIF-search network (Fig. 5a). The neural network is trained to simultaneously minimize the Euclidean distance between the fingerprint descriptors of binders versus targets, while maximizing the Euclidean distance between targets versus random, commonly referred to as a Siamese architecture in the machine learning literature[42] (see Methods).

Performance on the test set shows that the descriptor Euclidean distances for interacting surface patches is much lower than that of noninteracting patches, resulting in a ROC AUC of 0.99 (Fig. 5b). Our method is directly comparable to the previously proposed handcrafted GIF descriptors[17], which were proposed for a similar application: screening functional protein surfaces. Tested on our test set, GIF descriptors show a ROC AUC of 0.84, substantially lower than MaSIF-search (Fig. 5c). Testing MaSIF-search using only chemical or geometric features, we obtained ROC AUCs of 0.90 and 0.97, respectively. It is remarkable that chemical features alone can provide such a high discriminative power, the improvement from 0.97 to 0.99 is substantial, as if we interpret ROC AUC as error probability, it translates to reducing the number of mistakes from 3/100 to 1/100. We next investigated whether inverting the numerical features of the target patch is essential for MaSIF-search. Doing so results in faster learning and in gains in performance in a network trained with all features (ROC AUC of 0.97 with no inversion versus 0.99 with inversion, Supplementary Fig. 5). Finally, we observed that MaSIF-search and GIF descriptors, have superior performance on high shape complementarity patches, as training/testing on interacting patches with lower shape complementarity results in lower performance (Supplementary Fig. 5).

Next, we used MaSIF-search to predict the structure of known protein–protein complexes. Ideally, one would be able to predict whether two proteins interact simply by comparing their respective fingerprints, avoiding a time-consuming, systematic exploration of the 3D docking space. We find that fingerprint descriptors can provide an initial and fast evaluation of candidate binding partners. However, a better performance can be achieved by including a subsequent stage where candidate patches (referred to as decoys) selected by the Euclidean fingerprint distance of the patches center points to the target patch are rescored using fingerprints of neighboring points in the patch. Specifically, the MaSIF-search workflow entails two stages (Fig. 5d): (1) scanning a large database of descriptors of potential binders and selecting the top decoys by descriptor similarity and (2) three-dimensional alignment of the complexes exploiting fingerprint descriptors of multiple points within the patch, coupled to a reranking of the predictions with a separate neural network (see Methods and Supplementary Fig. 6). The first stage is performed extremely quickly; consequently, MaSIF-search runtime performance is dominated by the second stage whose complexity depends linearly on the number of decoys used. The tradeoff lies between increasing the number of decoys to improve accuracy, but slow down the overall runtime.

To benchmark MaSIF-search we simulated a scenario where the binding site of a target protein is known, and one attempts to recapitulate the true binder of a protein among many other binders. Specifically, we benchmarked MaSIF-search in 100 bound protein complexes randomly selected from our testing set (disjoint from the training set). For each complex, we first selected the center of the interface in the target protein (see Methods), and then attempted to recover the bound complex within the 100 binder proteins comprising the test set (Fig. 5d). A successful prediction means that a predicted complex with an interface root mean square deviation (iRMSD) of less than 5 Å relative to the known complex is found in a shortlist of the top 100, top ten or top one results. For comparison, we performed the same task using PatchDock[11], ZDock[43,44] and ZDock in combination with the scoring application ZRank2 (ref. [45]) (ZDock+ZRank2). For each program we compared our runtime performance and number of recovered complexes (Table 1). Among the baseline tools, PatchDock showed the fastest performance, while ZDock+ZRank2 showed the best performance. MaSIF-search with only 100 decoys per target shows performances similar to PatchDock, but the entire benchmark is performed in just four central processing unit (CPU) minutes, compared to 2,743 CPU minutes for PatchDock. If we expand MaSIF-search's decoys

to 2,000, it achieves similar performances to ZDock+ZRank2 with much faster runtimes (~4,000-fold).

Even though we trained only on cocrystallized protein complexes, we also tested our method in a benchmark set of 40 proteins crystallized in the unbound (apo) state. Since unbound docking is substantially more challenging, we changed the success criteria to finding the correct complex within the top 1,000, top 100 and top ten, for all methods (Table 2). Here the performance of all tools deteriorates, with slightly better accuracy for ZDock and ZDock+ZRank2. Although MaSIF-search can recover many of the complexes in the top 1,000 results, the scoring neural network, which was trained on holo structures, does not rank these into the top ten. These results point to the need for training MaSIF on apo structures, perhaps by augmenting datasets with simulated unbound states.

In the previous docking comparison, we provided the site of the interface as input; however, when the target site is unknown, a combination of MaSIF-site and MaSIF-search to predict protein complexes is an attractive possibility. To provide a specific example, we selected the protein complex PD1:PD-L1 (ref. [46]) (Protein Data Bank (PDB) ID 4ZQK) as a test case. We first used MaSIF-site for binding site prediction in the uncomplexed PD-L1 from the cocrystal structure, followed by MaSIF-search to scan a database of ~11,000 query structures (52 million surface fingerprint descriptors) to find putative binders of the predicted binding site in PD-L1 (this protocol is shown in Supplementary Fig. 7). The ground truth binder, PD1 was included among the 11,000 structures and PD1:PD-L1 related complexes were excluded from the training set. Our combined approach identified the mouse version of PD1 bound to human PD-L1 as the best binder (ranked no. 1, no. 3, no. 4), and the ground truth human PD1 binder (ranked no. 8) in 26 min. Performing vast searches using traditional docking tools is prohibitively expensive. In summary, MaSIF-search identifies patterns that drive PPIs that are embedded in a space amenable for fast searches.

Discussion

The molecular surface representation describes the features of a protein that contact other biomolecules, while abstracting the underlying protein sequence. This abstraction allows MaSIF to learn patterns that are independent of a protein's evolutionary history. Crucially, our general approach to learning surface fingerprints may enable a more complete understanding of protein function. This may prove critical in fields of protein science that have been shifting away from naturally evolved proteins. We foresee that MaSIF will be especially important for de novo protein design[47] applications, where the design of new biomolecular interactions remains a fundamentally unsolved problem, despite notable advances[37]. In the future, protein design programs such as Osprey[48] and Rosetta[49] may become fingerprint-aware, optimizing the sequence of de novo-designed proteins to display molecular surface patterns necessary to perform a functional task.

The proof-of-concept applications presented here meant to showcase MaSIF's generality and the concept of learning from surface features. Despite their early stage development, these methods can be useful to the wide community focused on understanding structure–function relationships. Such applications may entail the characterization of large-scale ligand–protein interaction networks (MaSIF-ligand), identification of 'surface hot-spots' that may be more easily targeted for the design of new biologics for therapeutic purposes (MaSIF-site). MaSIF-search could be coupled to experimental methods to identify binding partners for proteins, or it could be used to find potential engaging partners to use as starting points for protein design[37,41]. Moreover, all these methods could benefit from sequence evolutionary data to improve their predictive capabilities.

Collectively, we present a conceptual framework to decipher interaction fingerprints, leveraging the representation of protein structures as molecular surfaces, together with powerful data-driven learning techniques. The availability of our data and code will allow researchers to apply our framework to new problems. Our current applications show important technical advantages with great potential for further development and considerable impact on the fundamental study of protein structure and function, as well as for the design of new proteins and protein-based therapies.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-019-0666-6.

Received: 11 April 2019; Accepted: 28 October 2019;
Published online: 09 December 2019

References

1. Donald, B. R. *Algorithms in Structural Molecular Biology* (MIT Press, 2011).
2. Zhang, Q. C. et al. Structure-based prediction of protein–protein interactions on a genome-wide scale. *Nature* **490**, 556–560 (2012).
3. Hermann, J. C. et al. Structure-based activity prediction for an enzyme of unknown function. *Nature* **448**, 775–779 (2007).
4. Kortemme, T. et al. Computational redesign of protein–protein interaction specificity. *Nat. Struct. Mol. Biol.* **11**, 371–379 (2004).
5. Yang, J. et al. The I-TASSER Suite: Protein Structure and Function Prediction. *Nat. Methods* **12**, 7–8 (2015).
6. Planas-Iglesias, J. et al. Understanding protein–protein interactions using local structural features. *J. Mol. Biol.* **425**, 1210–1224 (2013).
7. Cong, Q., Anishchenko, I., Ovchinnikov, S. & Baker, D. Protein interaction networks revealed by proteome coevolution. *Science* **365**, 185–189 (2019).
8. Richards, F. M. Areas, volumes, packing, and protein structure. *Annu. Rev. Biophysics Bioeng.* **6**, 151–176 (2003).
9. Bronstein, M.M., Bruna, J., Lecun, Y., Szlam, A. & Vandergheynst, P. Geometric Deep Learning: Going Beyond Euclidean Data. *IEEE Signal Processing Magazine* **34**, https://doi.org/10.1109/MSP.2017.2693418 (2017).
10. Shulman-Peleg, A., Nussinov, R. & Wolfson, H. J. Recognition of functional sites in protein structures. *J. Mol. Biol.* **339**, 607–633 (2004).
11. Duhovny, D., Nussinov, R. & Wolfson, H.J. Efficient unbound docking of Rigid molecules. in *Proc. International Workshop on Algorithms in Bioinformatics* (eds., Guigó, R. and Gusfield, D.) **2452**, 185–200 (Springer, 2002); https://doi.org/10.1007/3-540-45784-4_14
12. Sharp, K. Electrostatic interactions in macromolecules: theory and applications. *Annu. Rev. Biophys. Biomol. Struct.* **19**, 301–332 (1990).
13. Daberdaku, S. & Ferrari, C. Antibody interface prediction with 3D Zernike descriptors and SVM. *Bioinformatics* **35**, 1870–1876 (2019).
14. Kihara, D., Sael, L., Chikhi, R. & Esquivel-Rodriguez, J. Molecular surface representation using 3D Zernike descriptors for protein shape comparison and docking. *Curr. Protein Pept. Sci.* **12**, 520–530 (2011).
15. Zhu, X., Xiong, Y. & Kihara, D. Large-scale binding ligand prediction by improved patch-based method Patch-Surfer2.0. *Bioinformatics* **31**, 707–713 (2015).
16. Venkatraman, V., Yang, Y. D., Sael, L. & Kihara, D. Protein–protein docking using region-based 3D Zernike descriptors. *BMC Bioinformatics* **10**, 407 (2009).
17. Yin, S., Proctor, E. A., Lugovskoy, A. A. & Dokholyan, N. V. Fast screening of protein surfaces using geometric invariant fingerprints. *Proc. Natl Acad. Sci. USA* **106**, 16622–16626 (2009).
18. Krizhevsky, A., Sutskever, I. & Hinton, G. Imagenet classification with deep convolutional neural networks. in *Advances in Neural Information Processing Systems* 1097–1105 (eds., F. Pereira, C.J.C. Burges, L. Bottou and K.Q. Weinberger) Curran Associates, Inc. (2012).
19. Monti, F. et al. Geometric deep learning on graphs and manifolds using mixture model CNNs. in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 5425–5434 (eds., R. Chellappa, Z. Zhang, and A. Hoogs) (2017).
20. Masci, J., Boscaini, D., Bronstein, M. M. & Vandergheynst, P. Geodesic convolutional neural networks on Riemannian manifolds. In *Proc. IEEE International Conference on Computer Vision* 832–840 (eds., R. Bajcsy, G. Hager, and Y. Ma) (2015).
21. Sanner, M. F., Olson, A. J. & Spehner, J. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers* **38**, 305–320 (1996).
22. Koenderink, J. J. & van Doorn, A. J. Surface shape and curvature scales. *Image Vis. Comput.* **10**, 557–564 (1992).
23. Kyte, J. & Doolittle, R. F. A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.* **157**, 105–132 (1982).

24. Jurrus, E. et al. Improvements to the APBS biomolecular solvation software suite. *Protein Sci.* **27**, 112–128 (2018).
25. Kortemme, T., Morozov, A. V. & Baker, D. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein–protein complexes. *J. Mol. Biol.* **326**, 1239–1259 (2003).
26. Chubukov, V., Gerosa, L., Kochanowski, K. & Sauer, U. Coordination of microbial metabolism. *Nat. Rev. Microbiol.* **12**, 327–340 (2014).
27. Konc, J. et al. ProBiS-CHARMMing: web interface for prediction and optimization of ligands in protein binding sites. *J. Chem. Inf. Modeling* **55**, 2308–2314 (2015).
28. Ritschel, T., Schirris, T. J. & Russel, F. G. KRIPO—a structure-based pharmacophores approach explains polypharmacological effects. *J. Cheminform.* **6**(Suppl 1): O26. https://doi.org/10.1186/1758-2946-6-S1-O26 (2014).
29. Ehrt, C., Brinkjost, T. & Koch, O. A benchmark driven guide to binding site comparison: An exhaustive evaluation using tailor-made data sets(ProSPECCTs). *PLoS Comput. Biol.* **14**(11), e1006483 (2018).
30. Ha, J. Y. et al. Crystal structure of d-erythronate-4-phosphate dehydrogenase complexed with NAD. *J. Mol. Biol.* **366**, 1294–1304 (2007).
31. Gauss, G. H., Kleven, M. D., Sendamarai, A. K., Fleming, M. D. & Lawrence, C. M. The crystal structure of six-transmembrane epithelial antigen of the prostate 4 (Steap4), a ferri/cuprireductase, suggests a novel interdomain flavin-binding site. *J. Biol. Chem.* **288**, 20668–20682 (2013).
32. Jones, S. & Thornton, J. M. Prediction of protein–protein interaction sites using patch analysis. *J. Mol. Biol.* **272**, 133–143 (1997).
33. Porollo, A. & Meller, J. Prediction-based fingerprints of protein–protein interactions. *Proteins* **66**, 630–645 (2007).
34. Northey, T. C., BarešiÄ, A. & Martin, A. C. R. IntPred: a structure-based predictor of protein–protein interaction sites. *Bioinformatics* **34**, 223–229 (2018).
35. Xue, L. C., Dobbs, D., Bonvin, A. M. J. J. & Honavar, V. Computational prediction of protein interfaces: a review of data driven methods. *FEBS Lett.* **589**, 3516–3526 (2015).
36. Murakami, Y. & Mizuguchi, K. Applying the naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. *Bioinformatics* **26**, 1841–1848 (2010).
37. Fleishman, S. J. et al. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science* **332**, 816–821 (2011).
38. King, N. P. et al. Computational design of self-assembling protein nanomaterials with atomic level accuracy. *Science* **336**, 1171–1174 (2012).
39. Correia, B. E. et al. Proof of principle for epitope-focused vaccine design. *Nature* **507**, 201–206 (2014).
40. Muja, M. & Lowe, D. G. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**, 2227–2240 (2014).
41. Greisen, P. J. et al. Computational design of environmental sensors for the potent opioid fentanyl. *eLife* **6**, 1–23 (2017).
42. Chopra, S., Hadsell, R. & LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. in *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **1**, 539–546 (eds., M. Hebert and D. Kriegman) IEEE (2005).
43. Pierce, B. G., Hourai, Y. & Weng, Z. Accelerating protein docking in ZDOCK using an advanced 3D convolution library. *PLoS ONE* **6**, e24657 (2011).
44. Lensink, M. F., Velankar, S. & Wodak, S. J. Modeling protein–protein and protein–peptide complexes: CAPRI 6th edition. *Proteins* **85**, 359–377 (2017).
45. Pierce, B. & Weng, Z. A combination of rescoring and refinement significantly improves protein docking performance. *Proteins* **72**, 270–279 (2008).
46. Zak, K. M. et al. Structure of the complex of human programmed death 1, PD-1, and its ligand PD-L1. *Structure* **23**, 2341–2348 (2015).
47. Huang, P. S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320–327 (2016).
48. Hallen, M. A. et al. OSPREY 3.0: Open-source protein redesign for you, with powerful new features. *J. Computational Chem.* **39**, 2494–2507 (2018).
49. Leaver-Fay, A. et al. in *Methods in Enzymology* (eds Johnson, M. J. & Brand, L.) 545–574 (Elsevier, 2010); https://doi.org/10.1016/b978-0-12-381270-4.00019-6

Methods

Computation of molecular surfaces. All proteins in the datasets were protonated using Reduce[50], and triangulated using the MSMS program[21] with a density of 3.0 and a water probe radius of 1.5 Å. Protein meshes were then downsampled and regularized to a resolution of 1.0 Å using pymesh[51]. Geometric and chemical features were computed directly on the protein mesh, with the exception of the distance-dependent curvature, which was computed on each patch according to the surface normals of the vertices in the patch.

Decomposition of proteins into overlapping radial patches and computation of features. For each point in the discretized protein surface mesh, a radial patch of geodesic radius of 9 or 12 Å (application-dependent) was extracted to perform an analysis of the surface features of the patch. The choice of radius was empirical, mainly driven by performance and memory constraints. For MaSIF-search we chose 12 Å because we found this to be a good value to cover the buried surface area of many PPIs. This patch size was reused for MaSIF-ligand. A patch of 9 Å was selected for MaSIF-site because the smaller patch allowed us to do multiple convolutional layers within our available memory resources, which we found critical for this application. In the absence of memory constraints, a patch larger than 12 Å would be ideal, as MaSIF's geometric deep learning architecture is capable of assigning different weights to different geodesically clustered kernels.

The following features were included in each patch.

Shape index. The shape index describes the shape around each point on the surface, with respect to the local curvature[17]. Values range from −1 (highly concave) to +1 (highly convex). It is defined with respect to the principal curvatures κ_1,κ_2, $\kappa_1 \geq \kappa_2$ as:

$$\frac{2}{\pi}\tan^{-1}\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2}$$

Distance-dependent curvature. For every vertex within an extracted patch, the distance-dependent curvature computes a value in the range [−0.7, 0.7] that describes the relationship between the distance to the center and the surface normals of each point and the center point. Details of this feature are described in ref. [17]. While the principal curvature component describes the shape around each vertex in the full protein, we found that it is also informative to compute the curvature within each patch, using the center of the patch as a reference.

Poisson–Boltzmann continuum electrostatics. PDB2PQR[52] was used to prepare protein files for electrostatic calculations and APBS[53] (v.1.5) was used to compute Poisson–Boltzmann electrostatics for each protein. The corresponding charge at each vertex of the meshed surface was assigned using Multivalue, provided within the APBS[53] suite. Charge values above +30 and below −30 were capped at those values and then values were normalized between −1 and 1.

Free electrons and proton donors. The location of free electrons and potential hydrogen bond donors in the molecular surface was computed using a hydrogen bond potential[25] as a reference. Vertices in the molecular surface whose closest atom is a polar hydrogen, a nitrogen or an oxygen were considered potential donors or acceptors in hydrogen bonds. Then, a value from a Gaussian distribution was assigned to each vertex depending on the orientation between the heavy atoms[25]. These values range from −1 (optimal position for a hydrogen bond acceptor) to +1 (optimal position for a hydrogen bond donor).

Hydropathy. Each vertex was assigned a hydropathy scalar value according to the Kyte and Doolittle[23] scale of the amino acid identity of the atom closest to the vertex. These values, in original scale ranged between −4.5 (hydrophilic) to +4.5 (most hydrophobic) and were then normalized to be between −1 and 1.

Computation of geodesic polar coordinates. Once surface patches are extracted from a protein, MaSIF uses a geodesic polar coordinate system to map the position of vertices in radial (that is, geodesic distance from the center) and angular coordinates (that is, angle with respect to a random directions) with respect to the center of the patch (Fig. 1c). These coordinates add information on the spatial relationship between features to the learning method.

Geodesic distances. On a continuous surface, a geodesic is the shortest path (curve) connecting two points when 'walking' over the surface; geodesic distance between two points is the length of a geodesic between them. On a mesh (the discretization of the continuous molecular surface we use in our implementation), a geodesic is the shortest polyline between two vertices, traversing triangular faces. On a graph, a geodesic is a collection of adjacent graph edges connecting two vertices. The computation of geodesics on meshes can be computed exactly or approximated using fast-marching methods. For computational efficiency, we used graph geodesics with weighted edges (corresponding to the Euclidean distance between the vertices), computed using the Dijkstra algorithm, as an approximation to the true geodesic. Since the molecular surfaces were regularly meshed, we found this to be an accurate compromise.

Radial coordinates. Describe the geodesic distance of a point to the center of the patch. Due to its speed, we used the Dijkstra algorithm implemented in MATLAB to compute an approximation of the true geodesic distance. Thus, in our implementation the geodesic distance is the sum of the edge lengths that connect the nodes defined on the surface mesh graph.

Angular coordinates. A classical multidimensional scaling algorithm[54] implemented in MATLAB was used to flatten patches into the plane based on the Dijkstra approximation to pairwise geodesic distances between all vertices. As molecular surface patches have no canonical orientation, a random direction in the computed plane was chosen as a reference and the angle of each vertex to this reference in the plane was set as the angular coordinate.

Geometric deep learning on a learned soft polar grid. Geometric deep learning allows us to apply successful image-based deep neural network architectures, such as CNNs[18], to geometric data such as surfaces. Traditional CNNs used in image analysis can be thought of as running a sliding window through the image; at each position of the window, a patch of pixels is extracted. Each pixel is then multiplied by a respective learnable filter value and the results summed up. On protein molecular surfaces, we do not have a regular grid, hence we replace it with a system of Gaussian kernels defined in a local geodesic polar system of coordinates that act as 'soft pixels'. The parameters of the Gaussians are learnable on their own[17]. Thus, we refer to this system of Gaussian kernels as a learned soft polar grid.

Our learned polar grid contains θ angular bins, and ρ polar bins, for a total of $J=\rho\theta$ bins. For each vertex in the discretized molecular surface x, with neighbors $N(x)$ and each vertex $y \in N(x)$, we define the coordinates $u(x,y)$, the radial and angular coordinates of y with respect to x. The mapping of each grid cell j for feature vector f and the patch centered at x, $D_j(x)f$, is defined as:

$$D_j(x)f = \sum_{y\in N(x)} w_j(u(x,y))f(y), j = 1,\ldots,J,$$

where w_j is a weight function and $f(y)$ are the features at vertex y.

Rotation invariance. Rotation invariance is handled in the neural network by performing θ rotations of the input patch and performing a max-pool operation on the output[18].

MaSIF-ligand: ligand site prediction and classification. *Dataset.* Proteins that bind to the selected cofactors were downloaded from the PDB and their biomolecular assemblies were built using SBI[55]. Details of pocket selection and clustering by sequence are presented in Supplementary Note 1.

Neural network architecture, cost function and training optimization. The training step and network architecture was as follows: 32 patches were randomly sampled from a single binding pocket. Each patch was used as input in a network and mapped to a learned soft grid with 16 angular bins and five radial bins. Each feature type (two geometric and three chemical features) was run through a separate neural network channel, where the learned soft grid layer was followed by a convolutional layer with 80 filters, an angular max pooling layer with 16 rotations, a rectified linear and a fully connected layer. A fully connected layer then combined the output from each channel and output to an 80-dimensional fingerprint. The resulting 32 fingerprints were multiplied together to generate an 80×80 covariance matrix. The architecture for this network is shown in Supplementary Fig. 9. The covariance matrix was flattened and fed first to a 64-unit, fully connected layer with rectified linear activation, and then to a seven-unit, fully connected layer with linear activation, followed by a softmax cross-entropy loss. The network was trained for 20,000 iterations (rather than epochs) with the Adam optimizer with a learning rate of 1×10^{-4}. The validation error was evaluated every epoch and the best network was selected based on this value. The initial choice of randomly sampling 32 patches in the pocket was made for three reasons: (1) each patch covers a 12 Å radius and, thus, 32 patches are likely to cover the surface from the entire pocket; (2) the number is low enough so that all ligand types are in contact with at least these many patch centers and (3) due to memory restrictions, since a larger number of patches exceeds our graphics processing unit (GPU) memory capabilities. To obtain more stable predictions, each pocket was sampled 100 times and the resulting 100 predictions were averaged to obtain the final prediction.

Visualization of relevant patches for NADP/NAD discrimination. For the discrimination in Supplementary Fig. 2, see Supplementary Note 2.

Comparisons to SiteEngine, ProBIS and KRIPO. See Supplementary Note 3.

MaSIF-site: protein interaction site prediction. *Datasets.* PPI pairs were taken from the PRISM list of nonredundant proteins, the ZDock benchmark, PDBBind and SabDab[56–59]. Sequence splits were performed using CD-HIT[60] and structural splits were performed using TM-align[61]. Details on the sequence and structural split are described in Supplementary Note 4.

Definition of interface points in a protein surface. We defined the ground truth interface as the region of the surface that becomes inaccessible to solvent molecules on complex formation. This was done by computing the surfaces of the complexes and the unbound partners. Surface regions in the individual partners that have no corresponding surface in the bound complex were then defined as the ground truth interface. Surface regions that become solvent inaccessible on complex formation were defined as the ground truth interface.

Neural network, cost function and training optimization. A neural network with three convolutional layers was used for this application. A diagram of the architecture is shown in Supplementary Fig. 10. The network received as input a full protein decomposed into overlapping surface patches with a radius of 9.0 Å. The smaller patch radius was selected because it reduced memory requirements, thus allowing more convolutional layers. The patches are mapped onto learned grids with three radial bins and four angular bins. The output of the network is an interface score between 0 and 1 for each patch center point. During training, the batch size consisted of a single protein, and the network was optimized using an Adam optimizer[62] on a sigmoid cross-entropy loss function. As the number of noninterface points is usually much larger than the number of interface points, a random subset of noninterface points was selected to train on an equal number of positive and negative samples. Training of the neural network was performed during 40 'wall-clock' hours, after which the job was automatically killed. These 40 h allowed for 43 epochs, whereas in each epoch all proteins in the training set were fed to the network. The best model was saved whenever the validation set's ROC AUC improved over that of a previous model. The last saved model occurred at epoch 42, which indicates that the neural network could have continued learning beyond the 40 allotted hours.

Comparisons to PSIVER and SPPIDER. See Supplementary Note 5.

MaSIF-search: prediction of PPIs based on surface fingerprints. *Datasets.* Details on the dataset and split are presented in Supplementary Note 6.

Selection of interacting and noninteracting patches. For each PPI, all pairs of surface patch centers belonging to distinct proteins and within 1.0 Å of each other were considered further. A radial shape complementarity score was computed for the pair as follows: (1) the shape complementarity of each point in the patch to the neighboring patch was computed; (2) points within 12 Å of the center were divided into ten concentric radial bins, in increments of 1.2 Å; the shape complementarity of the bin was computed as the 25th percentile of the points in the bin and (3) the radial shape complementarity S of the patch was computed as the median across all bins. The neural network for Fig. 5 was computed with interacting patches with a value of $S > 0.5$, while different ranges of S ($-1 < S < 0.1$ for very low complementarity, $0.1 < S < 0.3$ for low complementarity and $0.3 < S < 1.0$ for high complementarity) were also used to train and test (Supplementary Fig. 5). Noninteracting pairs were selected by pairing a truly interacting patch with a randomly chosen one from any other protein in the set.

Neural network architecture, cost function and training optimization. The MaSIF-search neural network receives the features of one patch (which may be inverted for the binding partner) as input and then outputs a vectorized descriptor. The architecture for this network is shown in Supplementary Fig. 11. During training and testing, a binder, a target and a random patch are input into the network, such that the binder and target are known interacting pairs and the target and random are assumed to be noninteracting. The features for the target are inverted (multiplied by −1), with the exception of the hydropathy index. A total of 85,652 true interacting pairs and 85,652 noninteracting pairs were chosen for training/validation, while 12,678 true interacting and 12,678 noninteracting pairs were chosen for testing. The network was trained to minimize the Euclidean distance between the fingerprint descriptors of binder and target, and maximize the distance between the descriptors of target and random. Each patch was input to a network and mapped to a learned soft grid with 16 angular and five radial bins. Each feature type (two geometric and three chemical features) was ran through a separate neural network channel, where the learned soft grid layer was followed by a convolutional layer with 80 filters, an angular max pooling layer with 16 rotations[20] and a rectified linear unit. A fully connected layer then combined the output from each channel, and output an 80-dimensional fingerprint. The optimization process during training, using an Adam optimizer[62], consists of minimizing the d-prime cost function[63]:

$$f(x) = \sigma_t + \sigma_f + \mu_t + \max\{0, M - \mu_f\},$$

where μ_t and μ_f are the median distance for true and noninteracting pairs, respectively, while σ_t and σ_f are the standard deviation for true and false interacting pairs. The neural network was trained with batches consisting of eight binder, eight target and eight random patches. In each batch the true interacting pairs and the random patch were randomly selected. The network was trained for 40 'wall-clock' hours, and killed after 40 h, which allowed for 335,000 iterations. The validation sets were evaluated after every 1,000 iterations. The best neural network model was determined as the one where the ROC AUC on the validation set achieved a maximum, which was reached after 260,000 iterations.

Structural alignment and rescoring. A second-stage alignment and scoring method generates the complexes based on the identified fingerprints. The top decoy patches with the shortest fingerprint descriptor distance to the target patch are selected as a shortlist of potential binding partners. Each binder patch is then aligned using the RANSAC algorithm implemented in Open3D[64] (Supplementary Fig. 6). Briefly, RANSAC selects three random points from the binder patch and uses the computed descriptors to find the closest points in the target patch by descriptor distance. Using these three newly found correspondences, RANSAC attempts to align the source patch to the target patch. RANSAC iterates 2,000 times and selects the transformation with the highest number of points within 1.0 Å between binder and target. Following RANSAC, an additional algorithm, the iterative closest point algorithm, as implemented in Open3D optimizes the alignment. After RANSAC completes, the transformation is rescored with a separate neural network. To optimize speed, the extracted patches were reduced to 9 Å.

Neural network for scoring aligned patches. To discriminate true alignments we trained a separate neural network to score binder patches after the alignment step (Supplementary Fig. 6). Once a patch alignment has been made, the nearest neighbor on the binder in 3D space to each point in the target is searched, establishing correspondences (Supplementary Fig. 6b). Then, the input to the neural network is the 3D Euclidean distance, the MaSIF-search fingerprint distance and the product of the normals between correspondences. The output is a predicted score on the alignments. To train this neural network we generated thousands of true and false alignments in the MaSIF-search training set. For each target structure we used one true alignment (defined as the true binder aligned within 5 Å iRMSD accuracy) and 200 false alignments (either sourced from a different protein from the true binder, or from the same protein but with over 5 Å iRMSD). iRMSD was defined as the RMSD of the Cα atoms of the binder that were less than 10 Å away from any of the Cα atoms of the target. For each point in an aligned patch we found its nearest neighbor (in 3D space, after alignment) on the target patch; for each pair of (binder, target) points we measured MaSIF-search fingerprint descriptor distance; the Euclidean distance in 3D space and dot products between their normals. The input features to our network were: 1/(descriptor distance), 1/(Euclidean distance) and the dot product of the normals. Each aligned patch was limited to 200 points, if the size of the aligned patch was greater than 200 points it was randomly sampled and if it was lower than 200 points it was zero-padded. Thus, the input to the network is a matrix of size 200,3 (200 point pairs with three features per pair). The network architecture was as follows: series of one-dimensional convolutional layers of dimensionalities 8, 16, 32, 64, 128, 256 with all these layers having a kernel size and stride of one; this was followed by a global average pooling layer and then a series of fully connected layers of dimensionality 128, 64, 32, 16, 8, 4, 2; alignments were labeled as positives or negatives and a cross-entropy loss was used, the negative class was weighted with 1/200. The Adam optimizer was used with a learning rate of 1×10^{-4}. From the training set, 10% of alignments were used as a validation set, the network was trained for 50 epochs with a batch size of 32. The best model was selected based on the lowest validation loss.

PPI search docking benchmark. See Supplementary Note 7.

Comparisons to GIF Descriptors, PatchDock11, Zdock and ZRank2. See Supplementary Note 8.

PD-L1 benchmark. See Supplementary Note 9.

Precomputation and neural network running times. The precomputing time of the PDB files to generate surfaces with features and runtime for MaSIF-search and MaSIF-site neural networks is dependent on the protein size, and is thus plotted in Supplementary Fig. 12. For example, a 125 amino acid protein is processed in 99.4 s accounting CPU, System and GPU times. GPU times were measured using 'wall-clock' time, since standard UNIX time tools do not account for GPU processing time. All times were measured on an Intel(R) Xeon(R) CPU E5-2650 v.2 at 2.60 GHz, and an NVIDIA Tesla K40 GPU running Red Hat Enterprise Linux 7.4. PDB files precomputations were performed on CPUs, while neural network calculations were performed on GPUs.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The bound PDBs in the training/testing set and the computed surfaces with chemical features are available at Zenodo with https://doi.org/10.5281/zenodo.2625420. The unbound PDBs in the test set are provided in the github repository. All scripts to generate the datasets are available at https://github.com/lpdi-epfl/masif.

Code availability

All code was implemented in Python and MATLAB. Neural networks were implemented using TensorFlow[65]. Both the code and scripts to reproduce the experiments of this paper are available at https://github.com/lpdi-epfl/masif[66]. The github repository also provides a PyMOL[67] plugin for the visualization of feature-rich molecular surfaces, used for the figures in this paper. All source code is provided under an Apache 2.0 permissive free software license.

References

50. Word, J. M., Lovell, S. C., Richardson, J. S. & Richardson, D. C. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *J. Mol. Biol.* **285**, 1735–1747 (1999).
51. Zhou, Q. PyMesh—Geometry Processing Library for Python. Software available for download at https://github.com/PyMesh/PyMesh (2019).
52. Dolinsky, T. J. et al. PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Res.* **35** (suppl. 2), W522–W525 (2007).
53. Baker, N. A., Sept, D., Joseph, S., Holst, M. J. & McCammon, J. A. Electrostatics of nanosystems: application to microtubules and the ribosome. *Proc. Natl Acad. Sci. USA* **98**, 10037–10041 (2001).
54. O'Connell, A. A., Borg, I. & Groenen, P. Modern multidimensional scaling: theory and applications. *J. Am. Stat. Assoc.* **94**, 338–339 (2006).
55. Bonet Martínez, J. *Exploiting Protein Fragments in Protein Modelling and Function Prediction* (Univ. Pompeu Fabra, 2015).
56. Baspinar, A. et al. PRISM: a web server and repository for prediction of protein–protein interactions and modeling their 3D complexes. *Nucleic Acids Res.* **42**, W285–W289 (2014).
57. Liu, Z. et al. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* **31**, 405–412 (2015).
58. Dunbar, J. et al. SAbDab: the structural antibody database. *Nucleic Acids Res.* **42**, D1140–D1146 (2013).
59. Vreven, T. et al. Updates to the integrated protein–protein interaction benchmarks: docking Benchmark version 5 and Affinity Benchmark version 2. *J. Mol. Biol.* **427**, 3031–3041 (2015).
60. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152 (2012).
61. Zhang, Y. & Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **33**, 2302–2309 (2005).
62. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Presented at *International Conference on Learning Representations (ICLR)* https://arxiv.org/abs/1412.6980 (2015).
63. Svoboda, J., Masci, J. & Bronstein, M. M. Palmprint recognition via discriminative index learning. In *Proc. International Conference on Pattern Recognition* 4232–4237 (eds. P. Gomez, S. Velastin) (2017); https://doi.org/10.1109/ICPR.2016.7900298
64. Zhou, Q.-Y., Park, J. & Koltun, V. Open3D: a modern library for 3D data processing. Technical report, available at: https://arxiv.org/abs/1801.09847 (2018).
65. Abadi, M. et al. TensorFlow: a system for large-scale machine learning. in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* 265–283 (eds., K. Keeton, T. Roscoe) (2016).
66. Pablo Gainza & Freyr S. *LPDI-EPFL/masif: MaSIF Paper Software Release* (Zenodo, 2019); https://doi.org/10.5281/zenodo.3519996
67. The PyMOL Molecular Graphics System v.1.8 (Schrödinger LLC, 2015).

Acknowledgements

We thank J. Bonet for helpful comments and J. Bonet, S.S. Vollers, P. de los Rios, S. Fleishman and A. Baptista for critical feedback on the manuscript. This work was funded by generous grants from the European Research Council (Starting grant no. 716058 to B.E.C. and Consolidator grant no. 724228 to M.M.B.). B.E.C. is also supported by the Swiss National Science Foundation (grants 31003A_163139 and 310030_188744) and the Biltema Foundation. P.G. is sponsored by an EPFL-Fellows grant funded by an H2020 Marie Sklodowska-Curie action and by the NCCR in Molecular Systems Engineering. F.S. is supported by a PhD fellowship from the Swiss Data Science Center. M.B. is partially supported by the Royal Academy Wolfson Research Merit Award, Google Faculty Research Awards. MaSIF's computations have been performed using the facilities of the Scientific IT and Application Support Center of EPFL.

Author contributions

P.G., F.S., F.M., M.M.B. and B.E.C designed the overall method and approach. M.M.B. and B.E.C supervised the research. P.G., F.M. and F.S. developed the base MaSIF method. P.G. designed and implemented MaSIF-site and MaSIF-search. F.S. designed and implemented MaSIF-ligand. F.S. and P.G. developed MaSIF-search's second-stage alignment algorithm. F.S. and P.G. developed the second-stage scoring neural network. P.G., F.S., M.M.B. and B.E.C. analyzed the data. E.R. and D.B. assisted in the design and development of these methods. P.G., F.S., M.M.B. and B.E.C wrote the manuscript. All authors read and commented the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-019-0666-6.

Correspondence and requests for materials should be addressed to B.E.C.

Peer review information Arunima Singh and Allison Doerr were the primary editors on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

nature research

Corresponding author(s): Bruno Correia

Last updated by author(s): Apr 19, 2019

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Code to reproduce the experiments and all the datasets used are available at https://github.com/lpdi-epfl/masif. The PDBs used and their corresponding surface files are available at: https://zenodo.org/record/2625420#.XLmvnJMzaL8
Data analysis	Software developed for data analysis is available at https://github.com/lpdi-epfl/masif.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The datasets generated during and/or analysed during the current study are available in https://github.com/lpdi-epfl/masif and the pdbs/surface files are available at: https://zenodo.org/record/2625420#.XLmvnJMzaL8

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	For our study we attempted to compile the largest datasets possible form structural data publicly available
Data exclusions	We excluded protein structures that could not be processed by our software.
Replication	Analysis and results were repeated multiple times.
Randomization	NA
Blinding	NA

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging