Methods 93 (2016) 41–50

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

PatchSurfers: Two methods for local molecular property-based binding ligand prediction

Woong-Hee Shin[a], Mark Gregory Bures[b], Daisuke Kihara[a,c,*]

[a] Department of Biological Sciences, Purdue University, West Lafayette, IN 47907, USA
[b] Discovery Chemistry Research and Technologies, Eli Lilly and Company, Indianapolis, IN 46285, USA
[c] Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 2 August 2015
Received in revised form 27 September 2015
Accepted 28 September 2015
Available online 30 September 2015

Keywords:
Structure–function relationship
3D Zernike descriptor
Protein function prediction
Patch-Surfer
Ligand–protein interaction
Ligand binding pockets

ABSTRACT

Protein function prediction is an active area of research in computational biology. Function prediction can help biologists make hypotheses for characterization of genes and help interpret biological assays, and thus is a productive area for collaboration between experimental and computational biologists. Among various function prediction methods, predicting binding ligand molecules for a target protein is an important class because ligand binding events for a protein are usually closely intertwined with the proteins' biological function, and also because predicted binding ligands can often be directly tested by biochemical assays. Binding ligand prediction methods can be classified into two types: those which are based on protein–protein (or pocket–pocket) comparison, and those that compare a target pocket directly to ligands. Recently, our group proposed two computational binding ligand prediction methods, Patch-Surfer, which is a pocket–pocket comparison method, and PL-PatchSurfer, which compares a pocket to ligand molecules. The two programs apply surface patch-based descriptions to calculate similarity or complementarity between molecules. A surface patch is characterized by physicochemical properties such as shape, hydrophobicity, and electrostatic potentials. These properties on the surface are represented using three-dimensional Zernike descriptors (3DZD), which are based on a series expansion of a 3 dimensional function. Utilizing 3DZD for describing the physicochemical properties has two main advantages: (1) rotational invariance and (2) fast comparison. Here, we introduce Patch-Surfer and PL-PatchSurfer with an emphasis on PL-PatchSurfer, which is more recently developed. Illustrative examples of PL-PatchSurfer performance on binding ligand prediction as well as virtual drug screening are also provided.

1. Introduction

Understanding protein function is one of the central problems in modern biology, including molecular biology, genetics, biochemistry, and bioinformatics. Bioinformatics can make substantial contributions in elucidating function of proteins. Using various types of biological databases now available, computational methods can quickly make function prediction to a large number of query proteins. To date, a number of bioinformatics tools for function prediction have been proposed [1–3]. These existing approaches can be categorized based on types of information they use, which include sequence-based, genome-based, proteomics-based, pathway-based, and structure-based [2].

Sequence-based methods compare a query protein sequence to sequences of known function in a database. This is the most classical strategy of function prediction. Conventional methods, which are often called homology search methods [4–6], use the well-accepted concept of homology and transfer function from highly similar (and thus considered as homologous) sequences. The sensitivity of a search can be improved by employing a statistical algorithm, hidden Markov models [7,8]. Identifying short sequence patterns that are conserved at known functional sites supplements homology search and help in annotating protein sequences [9–11]. Recent sequence-based methods try to improve the prediction performance in terms of the accuracy and the coverage by using more elaborated algorithms [12–16].

Genome-based methods predict functional relationship of protein genes from conservation of gene orders in different genomes [17], domain fusion events [18], and the similarity of phylogenetic profile [19]. STRING [20] is a database that contains pre-computed

* Corresponding author at: Department of Biological Sciences, Purdue University, West Lafayette, IN 47907, USA.
E-mail address: dkihara@purdue.edu (D. Kihara).

http://dx.doi.org/10.1016/j.ymeth.2015.09.026
1046-2023/© 2015 Elsevier Inc. All rights reserved.

predicted functional relationship of proteins from genome information.

Proteomics-based methods predict protein function in the context of protein–protein interaction (PPI) or gene expression. Proteins exhibit their functions by interacting with their partner molecules. Therefore, their functions can be inferred from an interaction graph drawn by PPI network data [21,22]. Gene expression data can also be a source for protein function annotation, because functionally related proteins are expected to have correlated expression patterns [23,24].

Pathway-based methods find missing genes in pathways of an organism, which make holes in pathway assignment of genes. Examples of holes in pathways can be observed, for example, at the KEGG pathway database [25], which maps genes or an organism to known pathways by homology search. Candidates of missing genes are unannotated genes in a genome. PathoLogic [26] employs Bayesian approach to match the gaps of pathways and uncharacterized proteins, while Chen and Vitkup [27] fill holes in the pathway by integrating phylogenetic profile and local structures of metabolic networks.

In this article, we introduce two of our methods, Patch-Surfer [28,29] and PL-PatchSurfer [30,31], which predict biological function, more precisely, the binding ligand for a query protein structure. These methods belong to structure-based function prediction methods. In general, structure-based methods are further classified into global structure-based methods and local structure-based methods. The former type compares the global fold of proteins as a strategy for finding distantly related proteins, using the observation that structures are more conserved than sequences [32,33]. Predicted structures of query proteins can be also used in global-structure based function assignment to achieve larger coverage in a genome-scale function assignment [34]. FINDSITE [35] and GalaxySite [36] predict active sites in a query protein structure by global structure matching. The latter, local-structure-based methods, search known functional sites in the global structure of a query protein [37–39] or compare a potential ligand binding site in a query structure to known ligand binding pockets [28–31,40–44]. Ligand binding sites in a protein structure can be predicted by considering geometric or energetic features of known binding pockets, which usually are cavities in protein surface [45–50]. Predicting binding ligands for a protein forms an important and interesting class of protein function predictions because it can often be directly tested by biochemical assays, and because it can provide useful information for drug design [51] and polypharmacology [52–54]. Therefore, conversely, ligand screening methods used for drug development, often called virtual screening methods, such as AutoDock [55], DOCK [56], and GLIDE [57], or pharmacophore search, e.g. LigandScout [58–60], can be applied for binding ligand prediction.

Recently, our group proposed two binding ligand prediction methods, one that performs pocket-pocket comparison, named Patch-Surfer [28,29] and the other one that compares a pocket against ligand molecules, named PL-PatchSurfer [30,31]. Patch-Surfer was designed specifically for binding ligand prediction as a way of predicting protein function while PL-PatchSurfer was developed for structure-based drug virtual screening. The two methods represent molecular surface and physicochemical properties of the surface using three-dimensional Zernike descriptors (3DZD) [61,62], a descriptor that is based on a mathematical series expansion of a three dimensional (3D) function. 3DZD compactly represents molecular surface as a vector of coefficients of a series expansion in a rotationally invariant fashion, which makes it a faster program than others in the field. Here, we introduce Patch-Surfer and PL-PatchSurfer with an emphasis on PL-PatchSurfer, which is more recently developed. Illustrative examples of PL-PatchSurfer performance on binding ligand prediction as well as virtual drug screening are provided.

2. Methods

2.1. Three-dimensional Zernike descriptors (3DZD)

In this section, a brief introduction of 3DZD will be given. Details of 3DZD can be found on two papers [61,62]. 3DZD is a representation of three-dimensional function of Euclidean space using 3D Zernike polynomials [61,62]. 3D Zernike polynomials are shown in Eq. (1).

$$Z^n_{ml}(r,\theta,\varphi) = R_{nl}(r)Y^m_l(\theta,\varphi) \tag{1}$$

n, l, m are called as order, degree, and repetition, respectively. The three indices are integers that are subjected to $-l < m < l$, $0 \leqslant l \leqslant n$, and $(n - l)$ is even. $R_{nl}(r)$ is a radial function while $Y^m_l(\theta,\varphi)$ is a spherical harmonics.

To compute 3DZD of any properties, the values should be mapped on three-dimensional grid points. For example, shape of a molecule is represented as a binary function, which gives 1 for surface region and 0 otherwise. For representing the surface electrostatic potential, the electrostatic values are first mapped on the surface and 3DZD is computed separately for regions with positive values and those with negative values [63]. The grid points with values that are assigned are considered a three-dimensional function of Euclidean space, $f(\boldsymbol{x})$, where $\boldsymbol{x} = \{x, y, z\}$. The function is expanded as a series of terms of Zernike basis as Eq. (2).

$$\Omega^m_{nl} = \frac{3}{4\pi}\int_{|x|\leqslant 1} f(\boldsymbol{x})\overline{Z^m_{nl}(\boldsymbol{x})}\boldsymbol{dx} \tag{2}$$

Calculating Zernike moments, Ω^m_{nl}, taking the norm of the moments as Eq. (3), and collecting them yields $(2l + 1)$ dimensional vector, a rotationally invariant representation of three-dimensional functions.

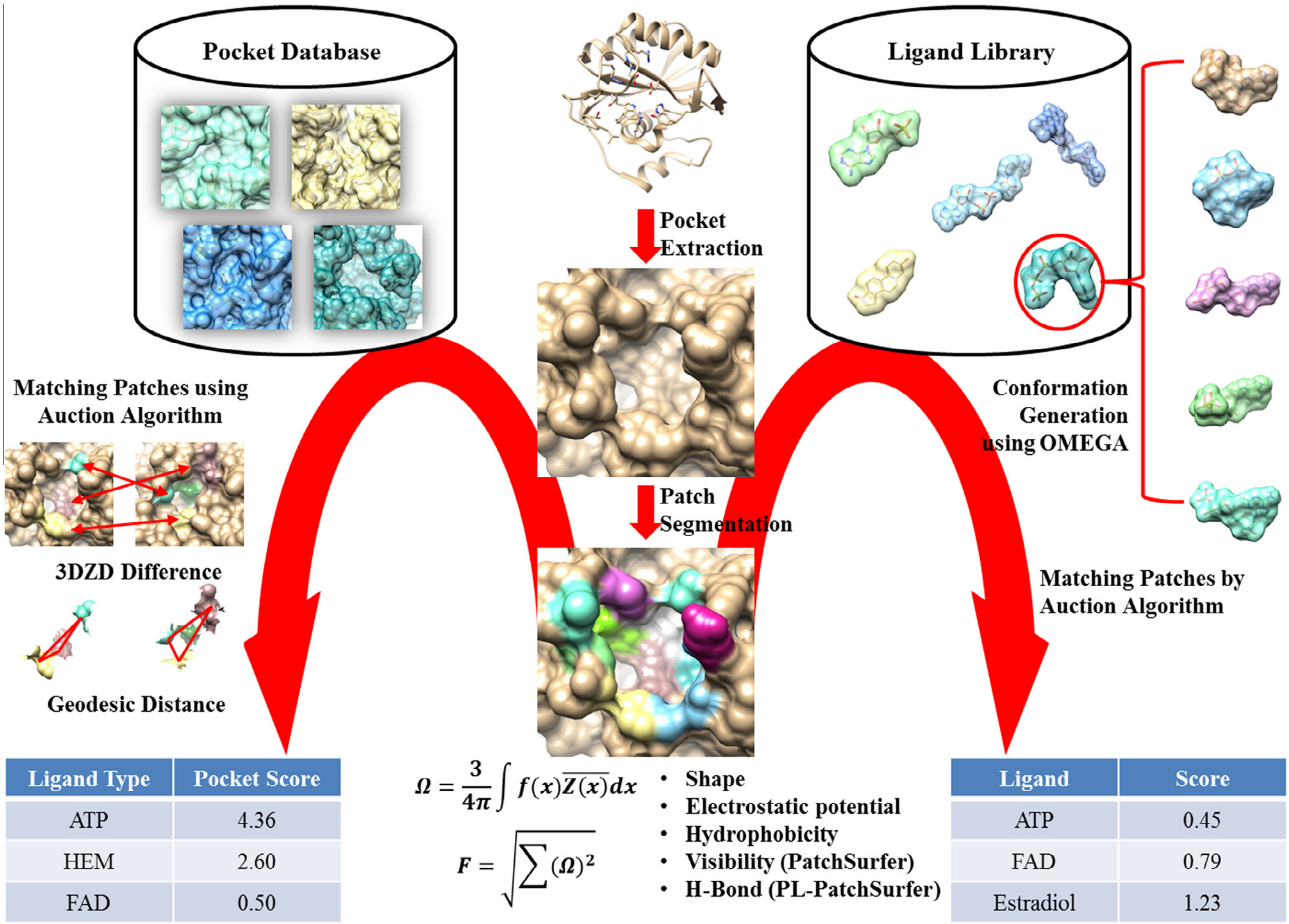
$$F_{nl} = \sqrt{\sum_{m=-l}^{m=l}(\Omega^m_{nl})^2} \tag{3}$$

The dimension of a 3DZD vector is determined by order n, which is related to the resolution of the representation. In Patch-Surfer and PL-PatchSurfer, introduced in following sections, $n = 15$ is used, yielding 72 invariants.

2.2. Patch-Surfer: a binding ligand prediction by patch-based pocket-pocket comparison

Patch-Surfer predicts a binding ligand of a given query protein by comparing a query pocket of the protein with known pockets in a pre-constructed pocket database. It was developed for structure-based protein function prediction by pocket comparison to characterize protein structures with unknown function. Pockets are segmented to a set of surface patches. Each patch is represented with 3DZD. The advantages of 3DZD and local patch are: (1) 3DZD comparison is fast and rotationally invariant and (2) local patch matching captures the local similarity of pockets, which is good for predicting docking of flexible molecules [28,29]. Below we outline the algorithm as illustrated in the left half of Fig. 1.

First, given a query protein with a potential ligand binding pocket, the surface of the pocket is generated using APBS [64], which constructs a molecular surface on a 3D grid and computes the electrostatic potential on the surface by solving the Poisson–Boltzmann equation. The pocket region of a query protein is defined by the atoms of the protein that are interacting with atoms


Pocket Database
Pocket Extraction
Ligand Library
Matching Patches using Auction Algorithm
Conformation Generation using OMEGA
Patch Segmentation
3DZD Difference
Matching Patches by Auction Algorithm
Geodesic Distance
$\Omega = \frac{3}{4\pi}\int f(x)\overline{Z(x)}dx$
$F = \sqrt{\sum(\Omega)^2}$
• Shape
• Electrostatic potential
• Hydrophobicity
• Visibility (PatchSurfer)
• H-Bond (PL-PatchSurfer)

Ligand Type	Pocket Score
ATP	4.36
HEM	2.60
FAD	0.50

Ligand	Score
ATP	0.45
FAD	0.79
Estradiol	1.23

Fig. 1. Flowchart of Patch-Surfer and PL-PatchSurfer.

of the ligand, if a ligand bound structure of the query protein is available. If the query structure is in *apo* form (without ligand) and binding sites are not known, a pocket can be predicted using various software [35–50]. After the surface is generated, hydrophobicity and concavity are also calculated and assigned on the surface. Then, the surface of the pocket is segmented into a number of overlapping patches. The segmentation is done as follows: (1) identifying the center of the ligand binding pocket; (2) casting rays from the center of the pocket to the protein surface to identify the boundary of the pocket; (3) distributing seed points on the pocket surface so that they are 3.0 Å away from each other; (4) generating a sphere with 5.0 Å radius centered at each seed point to define a patch; and (5) computing 3DZD for the four properties that characterize a patch: the shape, electrostatic potential, hydrophobicity, and the concavity.

To compute the similarity between a query pocket and a pocket in the database, a modified Auction algorithm [65] is implemented, which optimizes a target scoring function that consists of three parts: (1) similarity of matched patches taken from the two pockets, which is quantified by the Euclidean distance of 3DZDs; (2) the geodesic distance difference of matched patch pairs in the two pockets, to make sure that corresponding patches in each pocket locate in similar relative locations within each pocket; and (3) difference of approximate position of corresponding patches in the two pockets, which is represented as a histogram of geodesic distance between the patch and other patches of the pocket.

Once the scores for a query pocket to all pockets in the database are computed and ranked by the score, prediction of binding ligands for the pocket is made using the following Pocket_Score (Eq. (4)). The score of a ligand type *F* is calculated as:

$$\text{Pocket\_Score}(F) = \sum_{i=1}^{k}\omega_{l(i),F}\log\left(\frac{n}{i}\right)\frac{\sum_{i=1}^{k}\omega_{l(i),F}}{\sum_{i=1}^{n}\omega_{l(i),F}} \quad (4)$$

where *n* is the number of pockets in rank list, *k* is the number of top matches of similar ligands of *F* found by *k*-nearest neighbor classifier, *i* is the rank of the ligand, and $\omega_{l(i),F}$ is a two-dimensional similarity calculated by SIMCOMP [66].

The current pocket database used by Patch-Surfer contains 6547 pockets that bind to 2444 different ligand types. This database was constructed from protein-small-molecule database (PSMDB) [67,68]. Patch-Surfer is available as a webserver at http://kiharalab.org/patchsurfer2.0/.

2.2.1. Previous benchmark studies of Patch-Surfer

The performance of Patch-Surfer has been extensively tested since its initial development [28,29,69]. Firstly, it was compared with Pocket-Surfer [70], which uses 3DZD for describing the global shape of a pocket, and with four other similar mathematical series expansion, i.e. spherical harmonics, Legendre moments, 2DZD, and pseudo 2D Zernike descriptors [28]. Thus, the purpose of this benchmark was to evaluate the effect of using patch-representation over the global pocket representation by 3DZD (Pocket-Surfer) as well as evaluation of 3DZD relative to the similar choices of mathematical representations of a pocket. Patch-Surfer showed the highest performance among six programs when they were benchmarked on a dataset of 100 receptor pockets with nine ligand types [71]. Area Under Curve (AUC) values of Patch-Surfer, Pocket-Surfer, Legendre polynomial, pseudo Zernike, 2DZD, spherical harmonics are 0.81, 0.66, 0.53, 0.66, 0.66, 0.64, respectively when only pocket shape information was considered.

In addition, Patch-Surfer was further compared with four existing binding site comparison methods; eF-seek (which use graph matching) [72], SitesBase (geometric hashing) [73], PROSURFER (fingerprinting) [74], and XBSite2F (fingerprinting) [75]. These five methods were tested to 118 proteins bind with 18 different ligands [69]. The AUC value of Patch-Surfer is 0.86 while the other four methods show 0.49, 0.60, 0.57, and 0.55, respectively.

The latest systematic benchmark was done to compare the performance between Patch-Surfer2.0 [29] and APoc [40]. APoc searches similar binding pocket by structural alignment between query pocket and pockets in database. The benchmark set is composed of 348 binding sites with 15 different ligands. The average AUC values are 0.77 and 0.65 for Patch-Surfer and APoc, respectively.

To summarize, 3DZD, the mathematical base of the surface representation used in Patch-Surfer, performed better than the other similar moments, and moreover, the local patch representation was better than a global pocket shape representation employed in Pocket-Surfer. Moreover, Patch-Surfer showed best performance among other related binding site comparison methods.

2.3. PL-PatchSurfer: virtual screening and binding ligand prediction by patch-based pocket-ligand complementarity calculation

Recently, we have proposed a new protein–ligand screening method, called PL (Protein–Ligand)-PatchSurfer [30,31]. Instead of comparing a query pocket to known ligand binding pockets as Patch-Surfer performs, PL-PatchSurfer compares a pocket to ligand molecules and identifies the molecules that have high complementarity to the query pocket. Unlike Patch-Surfer, PL-PatchSurfer is originally designed for investigating ligand–protein interaction with an application of virtual screening of drug molecules. The flow of PL-PatchSurfer is illustrated on the right side in Fig. 1.

The algorithm of PL-PatchSurfer is similar to that of Patch-Surfer, except that alternative conformations of a ligand molecule (usually up to 50 conformations) are explicitly generated with OMEGA [76] to fully incorporate the ligand flexibility. In the case of Patch-Surfer, we thought representing a pocket with a combination of local patches is sufficient to consider flexibility of proteins [77], but in the case of ligands, a small bond rotation in a molecule can cause a large conformational change relative to the overall conformation of the molecule. After generating multiple conformations for a ligand, surfaces of all the generated ligand conformations are computed by APBS and physicochemical properties are assigned on the surfaces. Physicochemical properties represented with 3DZDs in PL-PatchSurfer are shape, electrostatic potential, hydrophobicity, and hydrogen bonding acceptors and donors. To measure complementarity between protein and ligand, the properties on ligand surface assigned opposite sign of its original value so that the complementarity of the properties can be computed as the similarity of 3DZDs. For example, if an electrostatic potential on a ligand surface point is −0.1, then it is converted to +0.1 while the sign is kept the same as the original value on the protein surface.

The scoring function of PL-PatchSurfer is composed of four parts. Three of which are the same as Patch-Surfer. The fourth term compares the size of a ligand and pocket, which is quantified as the number of patches that cover the ligand and the pocket. Since multiple conformations are generated for a ligand, a query pocket is matched to each conformation and the best score among them is taken as the final score between the pocket and ligand. Finally, ligands in a library are ranked by their scores.

2.3.1. Previous benchmark studies of PL-PatchSurfer

Since PL-PatchSurfer was originally developed for virtual screening of ligand molecules for a query binding pocket, it was originally tested [30] for its ability of finding active drugs among decoy molecules using the directory of useful decoys (DUD) dataset [78], a commonly used dataset in this field. The DUD dataset is composed of 40 traditional drug target proteins (e.g. kinase, nuclear receptors, and metalloenzymes) and their active compounds that are known to bind to the target, and decoy compounds that have similar chemical properties with active compounds. We used 25 targets out of 40, excluding 15 targets that have cofactors or metal ions because these additional molecules cannot be easily handled by APBS to compute electrostatic potential. In this benchmark, the initial version of PL-PatchSurfer [30], which only considers the shape and electrostatic potential to represent local patches, was tested. The performance of screening was compared with PharmDock [79], a protein pharmacophore-based docking program that has been compared with six state-of-the-art programs, DOCK [56], FlexX [80], Glide [57], ICM [81], Surflex [82], and PhDock [83], and shown to have a comparable performance with ICM and FlexX, and better performance than DOCK and PhDock [79]. Using the average Enrichment Factor (EF) at 1%, 10%, and 20% as the evaluation metrics, PL-PatchSurfer performed better than PharmDock at EF of 1% (8.6 and 6.9, respectively) and at 10% (2.5 and 2.2, respectively), while at 20% the two programs showed the same value (1.7).

PL-PatchSurfer was also applied to compute the similarity between ligands (ligand-to-ligand comparison) [31]. From the Jain set [84], which is composed of 22 proteins and their active and decoy compounds, up to 22 active compounds and 845 decoy compounds for each target protein were selected as a benchmark set. The performance of the program was compared with Global 3DZD [85], USR [86], and ROCS [87]. Although ROCS performs best, PL-PatchSurfer showed a unique feature to find diverse active compounds from a ligand library, which was different from the other global similarity search methods [31].

3. Benchmark setting

In this work we benchmarked PL-PatchSurfer in two settings. The first test was to evaluate its binding ligand prediction using datasets of ligand binding pockets. Next, we evaluated PL-PatchSurfer for virtual drug screening ability with an interest on its performance on *apo* form of target proteins. In what follows, first we explain the setting of the benchmark study and then report the results.

3.1. Datasets for binding ligand prediction

The binding ligand prediction was performed on two datasets. The first dataset is a compilation of 100 X-ray structures that bind either of nine ligand types. This dataset is called the Kahraman set [71]. The list of the PDB entries and the ligand structures are shown in Table 1A and Fig. 2(A). This set was manually curated using the following five criteria: (1) Structures should be determined with X-ray crystallography. (2) Binding sites of each ligand are not evolutionary related with each other and belong to different H-levels (homology-levels) in the CATH database [88]. (3) Modified, partial, or incorrectly labeled ligands are discarded. (4) Binding sites are only occupied by their cognate ligands. (5) Each ligand set has at least five members.

The second dataset was 36 protein structures that bind with either of 12 ligands (the Chikhi dataset) [70]. Three proteins are selected for each ligand type. The PDB entries are listed in Table 1B and the 2D structure of ligands is shown in Fig. 2(B). These

Table 1A
PDB ID list of Kahraman data set.

Ligand	PDB ID
AMP	12AS, 1AMU, 1C0A, 1CT9, 1JP4, 1KHT, 1QB8, 1TB7, 8GBP
ATP	1A0I, 1A49, 1AYL, 1B8A, 1DV2, 1DY3, 1E2Q, 1E8X, 1ESQ, 1GN8, 1KVK, 1O9T, 1RDQ, 1TID
FAD	1CQX, 1E8G, 1EVI, 1H69, 1HSK, 1JQI, 1JR8, 1K87, 1POX, 3GRS
FMN	1DNL, 1F6V, 1JA1, 1MVL, 1P4C, 1P4M
α-D-Glucose	1BDG, 1CQ1, 1K1W, 1NF5, 2GBP
HEME	1D0C, 1D7C, 1DK0, 1EQG, 1EW0, 1GWE, 1IQC, 1NAZ, 1NP4, 1PO5, 1PP9, 1QHU, 1QLA, 1QPA, 1SOX, 2CPO
NAD	1EJ2, 1HEX, 1IB0, 1JQ5, 1MEW, 1MI3, 1O04, 1OG3, 1QAX, 1RLZ, 1S7B, 1T2D, 1TOX, 2AF5, 2NPX
PO_4	1A6Q, 1B8O, 1BRW, 1CQJ, 1D1Q, 1DAK, 1E9G, 1EJD, 1EUC, 1EWC, 1FBT, 1GYP, 1H6L, 1HO5, 1L5W, 1L7M, 1LBY, 1LYV, 1QF5, 1TCO
Steroids	1E3R, 1FDS, 1J99, 1LHU, 1QKT

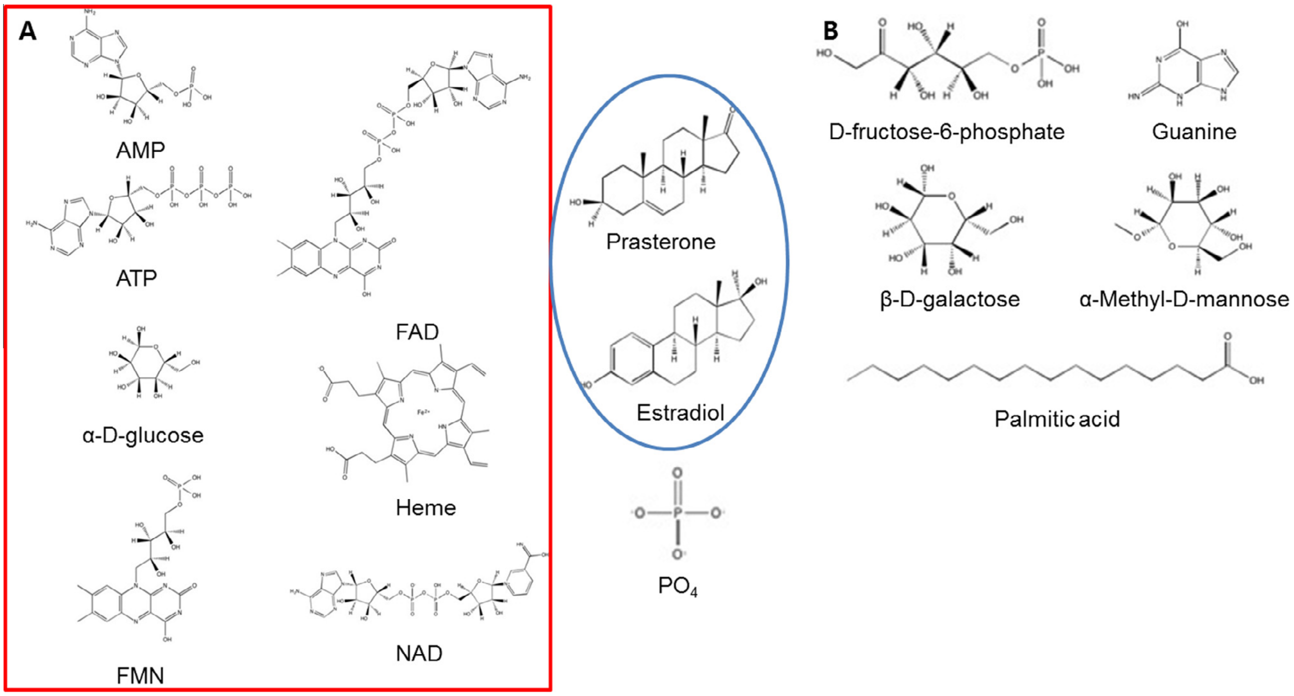


Fig. 2. 2D structures of ligands in (A) the Kahraman benchmark set and (B) the Chikhi set. The ligands in red box are common compounds of the both sets. Two ligands in blue circle are grouped as steroids.

two benchmark sets were chosen because they have been used in our earlier studies so that we can compare PL-PatchSurfer's performance with them [28,70].

The files of the datasets were prepared as follows: (1) protein structure files were downloaded from the PDB database [89]. (2) Multiple conformations of each ligand were generated from its SMILES string using OMEGA [76]. The maximum number of conformations was set to 50. The 'ewindow' option, which sets the condition for maximum energy of conformation, was set to 15 kcal/mol. The RMSD cutoffs between conformations were set to 0.5 Å, 0.8 Å, and 1.0 Å for ligands with zero to five, six to ten, and more than ten free torsion angles, respectively. The other options of OMEGA were kept same as default values.

Table 1B
PDB list of Chikhi set.

Ligand	PDB ID
AMP	1AMU, 1QB8, 8GPB
ATP	1A0I, 1KVK, 1O9T
FAD	1EVI, 1POX, 3GRS
FMN	1DNL, 1JA1, 1P4C
α-D-Glucose	1K1W, 1NF5, 2GBP
HEME	1QPA, 1SOX, 2CPO
NAD	1IB0, 1TOX, 2NPX
D-Fructose-6-phosphate	1UXR, 2BIF, 4PFK
β-D-Galactose	1XC6, 1Z45, 2GAL
Guanine	1XE7, 2PUC, 2PUF
α-Methyl-D-mannose	1LOB, 1MSA, 1MVQ
Palmitic acid	1MZM, 1PZ4, 1SZ7

3.2. Dataset for virtual screening

The performance of PL-PatchSurfer on virtual screening was evaluated on ten targets in the DUD set [78]. One of the foci of this benchmark is to investigate how well PL-PatchSurfer performs on *apo* structures of the targets. The list of the *apo* structures were found in a paper by Fan et al. [90]. We selected ten targets that have crystallized *holo* and *apo* structures; three from nuclear receptors (PPARγ, ERα (agonist), RXRα), three from kinases (CDK2, SRC, P38 MAP), one from serine proteases (thrombin), and three from other enzymes (AChE, NA, HIVPR). The receptor PDB ID is listed in Table 1C.

The binding site of a target was defined as a set of residues that have any heavy atom that is closer or equal to 5.0 Å to ligand heavy atoms. Cα-RMSD between an *apo* and a *holo* form of a binding pocket of a target was computed after aligning them with

Table 1C
PDB list of DUD set.

Protein	PDB ID
CDK2	1CKP, 1HCL
AChE	1EVE, 1QIH
PPARγ	1FM9, 1PRG
ERα (agonist)	1L2I, 2B23
NA	1A4G, 1NSB
P38 MAP	1KV2, 1P38
RXRα	1MVC, 1G1U
HIVPR	1HPX, 3PHV
SRC	2SRC, 1FMK
Thrombin	1BA8, 2AFQ

Underlined PDB ID is an *apo* form.

TM-align [91]. Three targets, RXRα, HIVPR, and SRC, have Cα-RMSD higher than 2.0 Å.

Holo receptor structures and the ligand library were kept as the same as the original DUD dataset, except that ratio of the number of active compounds and decoys was changed to 1:29 from 1:36 in order to make the computation time shorter. To do so, if the target has more than 3000 decoy molecules, all the actives and decoys were selected randomly so that the ratio of actives and decoys to achieve the 1:29 ratio. If not, only decoys were randomly removed. The *holo* structures were included in the DUD dataset while the *apo* structures were obtained from PDB. The multiple conformations of ligands were generated by OMEGA with the same parameter as described in Section 3.1. These datasets are made available at http://www.kiharalab.org/ps_ligandset/.

3.3. Performance metric used in the virtual screening benchmark

The performances of the methods were evaluated with Enrichment Factor (EF) and Area-Under Receiver Operating Characteristics Curve (AUC). These two values can be computed after compounds in the dataset, both actives and decoys, are ranked according to the score of a method to be evaluated. The perfect performance of selecting actives from decoys is achieved when all the actives are ranked higher than any decoys. EF at top *x*% subset is calculated as follows:

$$\mathrm{EF}_{x\%} = \frac{\mathrm{Actives}_{x\%}/\mathrm{Compounds}_{x\%}}{\mathrm{Actives}_{\mathrm{Library}}/\mathrm{Compounds}_{\mathrm{Library}}} \tag{5}$$

where $\mathrm{Actives}_{x\%}$ is the number of actives found in the top *x*% subset of compounds ranked by the score of the method, and $\mathrm{Compounds}_{x\%}$ represents the number of all the compounds within the top *x*% subset. In the similar way, $\mathrm{Actives}_{\mathrm{Library}}$ and $\mathrm{Compounds}_{\mathrm{Library}}$ are the number of actives and compounds that are contained in the whole library, respectively. $\mathrm{EF}_{x\%} = 1$ means that the retrieval of actives by a program is in the same level as a random selection. In this benchmark we examined EFs at 1%, 2%, and 5%.

Receiver operating characteristic (ROC) curve shows the performance of a virtual screening program by plotting actives found rate (true positive rate) against decoys found rate (false positive rate). As a program performs better, the curve approaches the upper-left corner. AUC is the area under the ROC curve. If a program successfully finds active compounds without finding any decoy compounds, AUC value becomes one, while the program totally fails to find active compounds then the value becomes zero, and a value close to 0.5 indicates that the retrieval is random.

Table 2
Success rates of binding ligand prediction methods benchmarked on Kahraman set.

	Top 1 prediction (%)	Top 3 prediction (%)
Kahraman set		
PL-PatchSurfer	48.2	82.7
PL-PatchSurfer (w/o H-bond)	44.7	80.7
PatchSurfer[a]	45.5	87.0
PocketSurfer[b]	36.1	81.5
Chikhi set		
PL-PatchSurfer	44.4	75.0
eF-Seek[b]	19.4	36.1
SitesBase[b]	32.2	60.0

[a] The success rates are taken from [28].
[b] The success rates are taken from [70].

4. Result and discussion

4.1. Binding ligand prediction

First, we tested PL-PatchSurfer for its performance on binding ligand prediction using the Kahraman set, which contains 100 binding pockets. Each pocket was considered as a query. The complementarities between a query pocket and ten ligands (nine ligand types, because prasterone and estradiol were considered as steroid) with multiple conformations are calculated. The ligand conformations were sorted according to PL-PatchSurfer score. The performance of a retrieval was evaluated by examining whether the top hit was the correct ligand type for the query pocket (top 1 accuracy) and also if the correct ligand was ranked within top three (top 3 accuracy). We used these metrics so that we can compare the performance of PL-PatchSurfer with our previous studies on Patch-Surfer [28] and Pocket-Surfer [70].

The summary of the results are given in Table 2. Overall, PL-PatchSurfer showed the best Top 1 prediction accuracy of 48.2% while Patch-Surfer performed better than PL-PatchSurfer when the best top 3 accuracy was considered. Examining results of each ligand type in Table 3, it is apparent that the larger overall average Top 1 accuracy by PL-PatchSurfer over Patch-Surfer comes mainly from better accuracy for ATP and NAD. For ATP, PL-PatchSurfer showed 78.6% and 100.0% for Top 1 and Top 3 accuracy, respectively, for which Patch-Surfer's accuracy showed 28.6% and 85.7%, respectively. For NAD, PL-PatchSurfer's Top 1 and Top 3 accuracy were 66.7% and 80.0%, while the corresponding values for Patch-Surfer were 6.7% and 80.0%, respectively. These differences might come from consideration of hydrogen bonding in PL-PatchSurfer. As mentioned in the previous section, PL-PatchSurfer additionally considers hydrogen-bonding acceptors and donors as one of the features of protein and ligand surfaces, which is not coded in Patch-Surfer. ATP has seven hydrogen bond donors and 18 hydrogen bond acceptors. Likewise, NAD has eight

Table 3
The success rate of binding ligand prediction on the Kahraman set.

	PL-PatchSurfer		PL-PatchSurfer (w/o H-bond)		PatchSurfer[a]	
	Top 1 (%)	Top 3 (%)	Top 1 (%)	Top 3 (%)	Top 1 (%)	Top 3 (%)
AMP	44.4	100.0	55.6	100.0	66.7	100.0
ATP	78.6	100.0	42.8	85.7	28.6	85.7
FAD	50.0	70.0	50.0	60.0	80.0	80.0
FMN	0.0	66.7	16.7	66.7	0.0	50.0
α-D-Glucose	40.0	60.0	40.0	60.0	40.0	100.0
HEME	68.8	87.5	68.8	93.8	87.5	100.0
NAD	66.7	80.0	13.3	80.0	6.7	80.0
PO_4	65.0	100.0	75.0	100.0	100.0	100.0
Steroids	20.0	80.0	40.0	80.0	0.0	80.0
Average	48.2	82.7	44.7	80.7	45.5	87.0

[a] Success rates were recalculated from Fig. 5 of [28].

Table 4
Virtual screening results on the ten DUD targets by the three programs. The ten targets are further classified into two groups based on Cα-RMSD of binding sites.

Program	Overall				RMSD < 2.0 Å				RMSD ⩾ 2.0 Å			
	$EF_{1\%}$	$EF_{2\%}$	$EF_{5\%}$	AUC	$EF_{1\%}$	$EF_{2\%}$	$EF_{5\%}$	AUC	$EF_{1\%}$	$EF_{2\%}$	$EF_{5\%}$	AUC
Holo receptor structures												
PL-PatchSurfer	14.3	9.5	5.3	0.578	12.0	7.8	4.4	0.550	19.7	13.5	7.5	0.643
AutoDockVina	7.1	6.7	5.4	0.689	5.5	5.5	4.3	0.650	10.8	9.4	8.2	0.780
DOCK6	13.6	9.3	5.5	0.578	11.5	8.1	5.2	0.604	18.4	12.1	6.2	0.516
Apo receptor structures												
PL-PatchSurfer	12.8	8.7	4.6	0.562	12.3	7.6	4.3	0.547	14.1	11.1	5.3	0.596
AutoDockVina	2.8	2.8	2.3	0.546	3.4	3.2	1.8	0.521	1.3	1.7	3.4	0.604
DOCK6	6.6	5.5	4.0	0.551	8.7	7.2	5.1	0.615	1.7	1.7	1.6	0.402

The values are averaged over the targets.

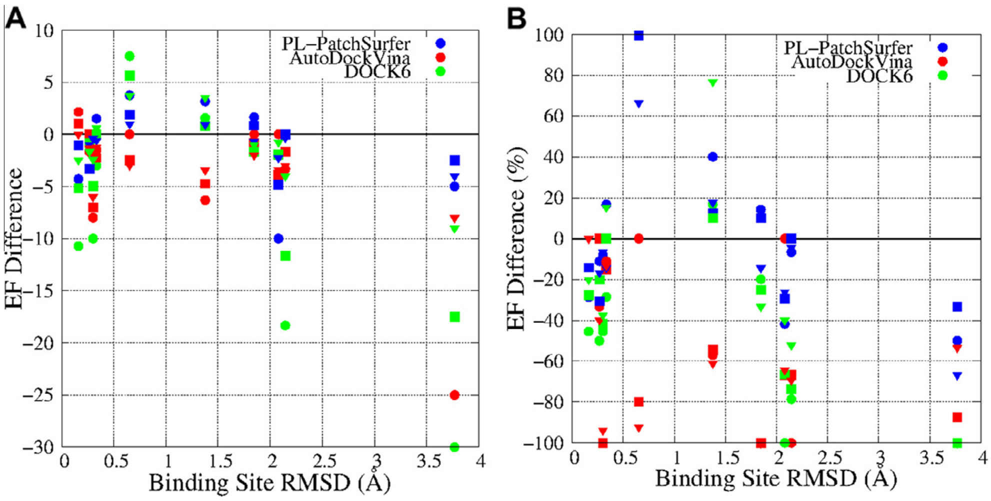


Fig. 3. Differences of enrichment factor for *holo* and *apo* forms. (A) Enrichment factor differences between the *holo* form and the *apo* form structure (*y*-axis, $EF_{apo} - EF_{holo}$) as a function of binding site Cα-RMSD (*x*-axis). (B) the ratio of enrichment factor difference relative to the enrichment factor fo the *holo* form (*y*-axis, $(EF_{apo} - EF_{holo})/EF_{holo}$) as a function of Cα-RMSD (*x*-axis). PL-PatchSurfer, AutoDockVina, and DOCK6 are colored in blue, red, and green, respectively. The circle, square, and triangle represent $EF_{1\%}$, $EF_{2\%}$, and $EF_{5\%}$.

hydrogen bond donors and 19 hydrogen bond acceptors. Except for NAD and ATP, the eight ligands have 3.8 hydrogen bond donors and 8.3 hydrogen bond acceptors on average. To further examine the effect of considering hydrogen bonding, we ran PL-PatchSurfer without the hydrogen bonding scoring term. The result is shown in the second row of Tables 2 and 3. Without considering hydrogen bonds, the Top 1 accuracy of PL-PatchSurfer dropped from 78.6% to 42.8% for ATP and 66.7% to 13.3% for NAD (Table 3), which reflected to the overall better accuracy of PL-PatchSurfer with the hydrogen term over the method without hydrogen bond term (weights for features of patches were trained without hydrogen bond term) (Table 2). However, Table 3 also shows that the accuracy of some other ligands decreased by adding the hydrogen bond feature, which implies that different ligands may have a different optimal setting of parameters.

In the second half of Table 2, we compared PL-PatchSurfer with two other existing programs, eF-Seek [72] and SitesBase [73]. PL-PatchSurfer showed the highest success rates in both Top 1 and Top 3 predictions among them.

To summarize, PL-PatchSurfer performed comparably if not better than Patch-Surfer in binding ligand prediction, although its original purpose is different, virtual screening. It also performed better than two other approaches for structure-based binding ligand comparison. Considering the complementary nature of PL-PatchSurfer and Patch-Surfer, in that they compare a query pocket to different types of structural data, it would be beneficial to use either one of them or combine their results depending on the situation of scenarios users have. This is discussed further in Conclusions.

4.2. Results of the virtual screening experiments

Next, we tested the performance of PL-PatchSurfer on *holo* and *apo* structures of ten targets in the DUD dataset in comparison with two state-of-the-art protein–ligand docking programs, AutoDockVina [55] and DOCK6 [56]. These two programs were run with their default parameters. The center coordinates of the binding pocket box for both programs were set to the geometric center of the bound ligand in the crystal structure of target proteins. Protonation states and atomic charges of all targets and ligands were kept as same as given in the DUD dataset. The results are summarized in Table 4.

For the *holo* structure (ligand bound structure) set (the upper half of Table 4), PL-PatchSurfer showed overall highest $EF_{1\%}$ and $EF_{2\%}$ values among three programs (the left columns). For $EF_{5\%}$, DOCK6 showed a slightly higher value of 5.5 over 5.3 by PL-PatchSurfer. In terms of AUC, AutoDockVina showed the highest performance. These results imply that PL-PatchSurfer finds active compounds in earlier ranks than the other two programs.

For the *apo* structure (ligand unbound structure) set (the bottom half in Table 4), overall PL-PatchSurfer gave the highest performance except for AUC (the left columns). All the values of all programs decreased from the *holo* set results as expected, because the *apo* structures are different from the ligand binding

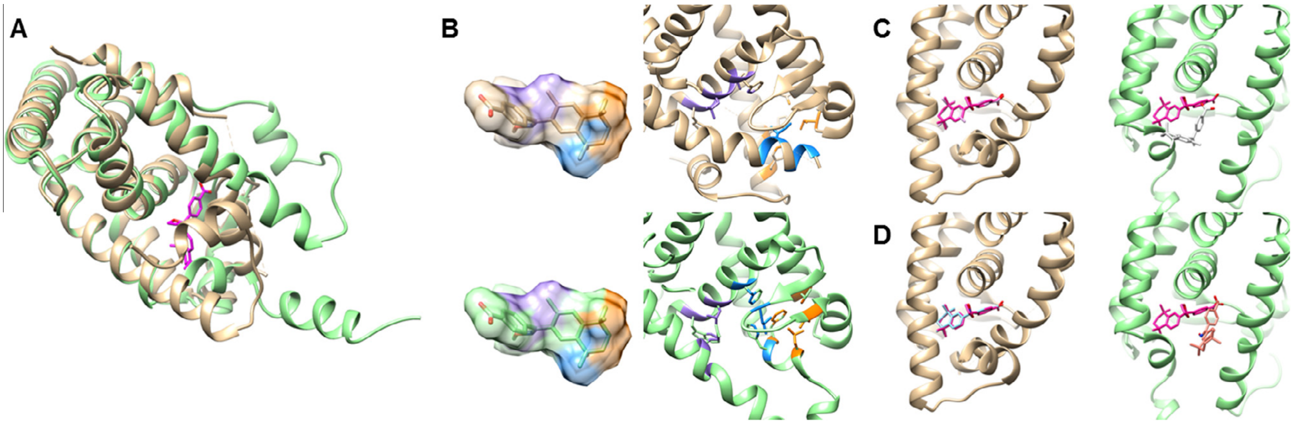


Fig. 4. Ligand docking of RXRα. (A) Superimposed *holo* (gold) and *apo* form (green) of RXRα. The cognate ligand is colored in magenta. (B) The top scored conformation of ZINC03834071, which is ranked 2nd (*holo* form) and 1st (*apo* form) by PL-PatchSurfer. Matched patch pairs of the pocket and the ligand are shown in the same colors. (C) The docked structure of an active compound, ZINC01539579, which was ranked the 1st (*holo* form, left, pink) and the 8th (*apo* form, right, white) by AutoDockVina. The bound structure of the compound is shown in magenta. (D) The docked structure of ZINC03834076, which was ranked the 1st (*holo* form, left, cyan) and the 84th (*apo* form, right, orange) by DOCK6.

form. It is interesting that for the two conventional docking programs, EF values decreased about 50% or more for $EF_{1\%}$ and $EF_{2\%}$ for EF, whereas PL-PatchSurfer EF values did not deteriorate significantly. To investigate the effect of the receptor structure change between *holo* and *apo* forms, we classified the targets based on the pocket Cα-RMSD. For targets with Cα-RMSD $\geqslant$ 2.0 Å (the right columns), the performances of AutoDockVina and DOCK6 decreased drastically. Their $EF_{1\%}$ and $EF_{2\%}$ values decreased by about 90% when compared with the *holo* form results. In contrast, PL-PatchSurfer results retained ~70% of those of the *holo* forms. The reason why PL-PatchSurfer was not severely affected by pocket structure change is probably because it uses the molecular surface description of local patches. This positive effect of the patch representation was also observed for Patch-Surfer in the previous works [28,29], where it was able to identify pockets of the same ligands that have largely different global shape.

The changes of enrichment factors of the individual targets are shown in Fig. 3. Fig. 3A shows the difference of enrichment factors ($\Delta EF = EF_{apo} - EF_{holo}$). As binding site Cα-RMSD becomes larger, ΔEF tends to largely decrease for AutoDockVina and DOCK6. However, PL-PatchSurfer seems not to be greatly affected by the structural change of the receptors, as the decrement of PL-PatchSurfer is lower than 10 in all the cases. In Fig. 3B, the ratio of the ΔEF relative to the enrichment factor of the *holo* forms (i.e. $\Delta EF/EF_{holo}$) were plotted. It is obvious also from this plot that PL-PatchSurfer is more tolerant to the structure change. Enrichment factors of PL-PatchSurfer did not decrease more than 70%, while those of the other two programs decreased by 100% in one or two targets. When the RMSD was smaller than 2.0 Å, PL-PatchSurfer somehow performed even better for the *apo* forms for a number of cases.

The largest structure variation occurs in RXRα, with a Cα-RMSD of 3.8 Å. Similar to other nuclear receptors, AF2-helix of RXRα changes its position by the state of the protein; *apo* form, agonist binding, and antagonist binding form [92]. The superimposition of the *holo* and the *apo* forms is shown in Fig. 4A. When the *holo* form was used as a target structure, $EF_{1\%}$ of PL-PatchSurfer, AutoDockVina, and DOCK6 were 10.0, 25.0, and 30.0, respectively. For the *apo* form, $EF_{1\%}$ of PL-PatchSurfer decreased to 5.0 while that of the other two programs deteriorated to 0.0, which means that actives were not retrieved within 1.0% at all. To find the reason for this performance difference between PL-PatchSurfer and the

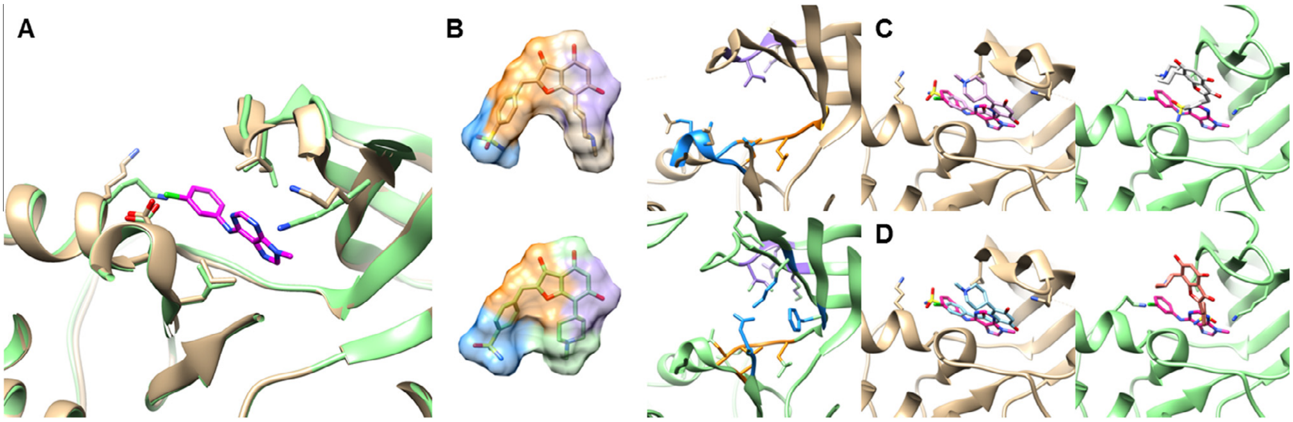


Fig. 5. Ligand docking for CDK2. (A) Superimposition of the *holo* (gold) and the *apo* (green) form of the binding sites of CDK2. The cognate ligand is colored in magenta. The three key residues for the interaction, ASP86, ILE11, and LEU134 [94] (from left to right) are shown. (B) The top scoring conformation of an active compound, ZINC03814433, which was ranked the 12th (*holo* form) and the 26th (*apo* form) by PL-PatchSurfer. Matched patch pairs of the pocket and the ligand are shown in the same colors. (C) Docked structure of ZINC03814433, which was ranked at the 23rd (*holo* form, left, pink) and the 429th (*apo* form, right, white) by AutoDockVina. The docked conformation of the compound is shown in magenta. (D) The docked structure of ZINC03814433, which was ranked the 10th (*holo* form, left, cyan) and the 60th (*apo* form, right, orange) by DOCK6.

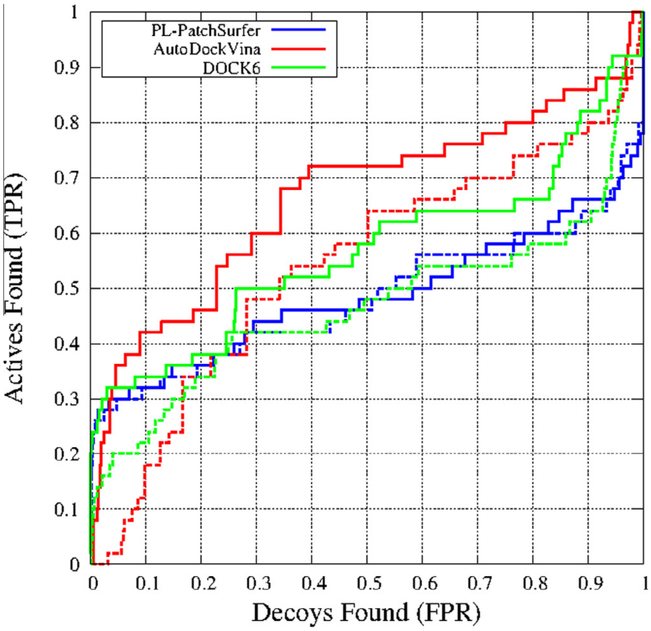


Fig. 6. ROC of CDK2 virtual screening. Blue, red, and green lines are for PL-PatchSurfer, AutoDockVina, and DOCK6, respectively. Solid lines are results when the *holo* structure was used while dashed lines are results for the *apo* structure.

other two programs, the highest ranked active compounds in the *holo* form were examined. When the *holo* structure was used as a target, both AutoDockVina (Fig. 4C, left panel) and DOCK6 (Fig. 4D, left panel) found the correct docked conformation of the ligand that were well-aligned with the cognate ligand. However, when the *apo* structure was used, the active ligands were located outside of the binding pocket due to the change of the pocket shape caused by the move of the AF-2 helix. These docked conformations cannot make hydrophobic contact with phenylalanines, which is important for ligand binding in the nucleus [93]. In comparison, in Fig. 4B we show the docking poses of the top scoring active compound of PL-PatchSurfer (ZINC03834071). It is shown that the identified interacting local regions between the receptor and the ligand shown in the same colors are almost the same between the *holo* (upper panel) and the *apo* form (lower panel). These interacting residues to the ligand are composed of aromatic residues and hydrophobic residues. Thus, unlike the other two docking programs, PL-PatchSurfer could find the similar corresponding docking poses of the ligand in the binding pocket.

Similar to RXRα, the performance of the two conventional docking programs significantly dropped for CDK2 when the *apo* form was used as the query. The binding site Cα-RMSD between the *holo* and *apo* form is only 0.3 Å; however, two lysines change their rotameric state when the ligand binds (Fig. 5A). When the *holo* form was used as a query, EF at 1% of PL-PatchSurfer, AutoDockVina, and DOCK6 were 24.0, 8.0, and 22.0, respectively. When the *apo* structure was used, the values were reduced to 22.0, 0.0, and 12.0 respectively. Although the conformational change between the *holo* and the *apo* form was small, $EF_{1\%}$ values of AutoDockVina and Dock6 dropped substantially while PL-PatchSurfer almost maintained performance. To understand this different performance of the programs for the *apo* form, we examined docking poses of ZINC03814433, the ligand that ranked within top 30 (which corresponds to the top 2% rank) for the *holo* form by all the three programs. PL-PatchSurfer found corresponding patch pairs in the binding pocket for ligand patches in similar positions for the both *holo* and *apo* structures (Fig. 5B). Purple and orange patches in the receptor structures are composed of hydrophobic residues, while blue patches correspond to polar residues (ASN for the *holo* form, ASP for the *apo* form). The other two programs (Fig. 5C and D) found a correct position of the ligand, where half of the ligand is overlapped with the cognate ligand binding position, for the *holo* form. However, when the *apo* structure was used, the two programs could not find the correct docking position of the ligand, but rather located the ligand outside of the binding pocket. The ROC curves of CDK2 for the three programs are shown in Fig. 6. In Fig. 6, solid lines represent the *holo* structure virtual screening results, while dashed lines show *apo* structure results. It is shown that the performance of AutoDockVina and DOCK6 were substantially lowered for early recognition for the *apo* form.

4.3. Computational time of PL-PatchSurfer

The average computational times to screen the library of each target are 2.7, 51.7, and 54.4 h for PL-PatchSurfer, DOCK6, and AutoDockVina, respectively. Thus, PL-PatchSurfer is ~20 times faster than the other two programs. The computational times were measured on a Linux machine with Intel i7-3820 3.60 GHz CPU and 64 GB RAM.

5. Conclusions

We introduced two binding ligand prediction programs, Patch-Surfer and PL-PatchSurfer. Both programs use 3DZD to describe the properties of surface patches of molecules. The advantage of 3DZD moments is that it allows fast comparison of surfaces, because it is a compact and rotationally invariant representation of surfaces. In addition, the programs also enjoy advantages from the local patch description, which is insensitive to subtle atomic position change of molecular surfaces, such as tautomeric shifts and conformational changes. PL-PatchSurfer performed in the similar level if not better to Patch-Surfer for the binding ligand predictions in the Kahraman dataset and performed better than the two state-of-the-art programs, AutoDockVina and Dock6, in the virtual screening benchmarks. The advantage of PL-PatchSurfer was evident over the two programs when *apo* forms of binding pockets were used as queries.

Although the algorithms of the two methods share a common architecture, they are fundamentally different in the reference data, against which a query pocket is compared. Patch-Surfer compares a query pocket against known ligand binding pockets, while the PL-PatchSurfer compares a pocket directly against 3D conformations of compounds in a library. These two strategies have both advantages and disadvantages: An advantage of the pocket-pocket comparison performed by Patch-Surfer is that generating conformations of compounds, which is known to be difficult and time consuming, is circumvented. Also, biological function (ligand binding) of a query protein can be easily inferred by similarity to known pockets. On the other hand, the types of ligands that can be predicted by Patch-Surfer are limited due to the limited availability of known crystal structures of binding pockets. On the other hand, PL-PatchSurfer has a larger coverage in the ligand space that can be predicted, because as long as the 2D structure of ligands are known, their 3D conformations can be generated and docked to a query pocket. However, a challenge in PL-PatchSurfer is the efficient and accurate sampling of conformational space of ligands. Knowing the differences and complementary nature of these two methods, they must be properly chosen or combined depending on the user's prediction scenarios.

Acknowledgements

The authors are thankful to Josh McGraw and Lyman Monroe for proofreading the manuscript. This work is supported by a grant from the Lilly Research Award Program. DK also acknowledges funding from the National Institute of General Medical Sciences of the National Institutes of Health (R01GM097528) and the National Science Foundation (IIS1319551, DBI1262189, IOS1127027).

References

[1] J.D. Watson, R.A. Laskowski, J.M. Thornton, Curr. Opin. Struct. Biol. 15 (2005) 275–284.
[2] T. Hawkins, D. Kihara, J. Bioinform. Comput. Biol. 5 (2007) 1–30.
[3] J. Konc, D. Janezic, Curr. Opin. Struct. Biol. 25 (2014) 34–39.
[4] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, J. Mol. Biol. 215 (1990) 403–410.
[5] W.R. Pearson, Methods Enzymol. 183 (1990) 63–98.
[6] W.R. Pearson, D.J. Lipman, Proc. Natl. Acad. Sci. U.S.A. 85 (1988) 2444–2448.
[7] A. Bateman, L. Coin, R. Durbin, R.D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E.L. Sonnhammer, D.J. Studholme, C. Yeats, S.R. Eddy, Nucleic Acids Res. 31 (2003) D138–D401.
[8] M. Remmer, A. Biegert, A. Hauser, J. Söding, Nat. Methods 9 (2011) 173–175.
[9] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, B.A. Cuche, E. de Castro, C. Lachaize, P. S. Langendijk-Genevaux, C.J.A. Sigrist, Nucleic Acids Res. 36 (2008) D245–D249.
[10] S. Hunter, P. Jones, A. Mitchell, R. Apweiler, T.K. Attwood, A. Bateman, T. Bernard, D. Binns, P. Bork, S. Burge, E. de Castro, P. Coggill, M. Corbett, U. Das, L. Daugherty, L. Duquenne, R.D. Finn, M. Fraser, J. Gough, D. Haft, N. Hulo, D. Kahn, E. Kelly, I. Letunic, D. Lonsdale, R. Lopez, M. Madera, J. Maslen, C. McAnulla, J. McDowall, C. McMenamin, H. Mi, P. Mutowo-Muellenet, N. Mulder, D. Natale, C. Orengo, S. Pesseat, M. Punta, A.F. Quinn, C. Rivoire, A. Sangrador-Vegas, J.D. Selengut, C.J.A. Sigrist, M. Scheremetjew, J. Tate, M. Thimmajanarthanan, P.D. Thomas, C.H. Wu, C. Yeats, S.-Y. Yong, Nucleic Acids Res. 40 (2012) D306–312.
[11] I. Letunic, R.R. Copley, S. Schmidt, F.D. Ciccarelli, T. Doerks, J. Schultz, C.P. Ponting, P. Bork, Nucleic Acids Res. 32 (2004) D142–D144.
[12] T. Hawkins, S. Luban, D. Kihara, Protein Sci. 15 (2006) 1550–1556.
[13] I.K. Khan, Q. Wei, D. Kihara, Bioinformatics 31 (2015) 271–272.
[14] M.A. Messih, M. Chitale, V.B. Bajic, D. Kihara, X. Gao, Bioinformatics 28 (2012) i444–i450.
[15] M.N. Wass, M.J. Sternberg, Bioinformatics 24 (2008) 798–806.
[16] F. Minneci, D. Piovesan, D. Cozzetto, D.T. Jones, PLoS ONE 8 (2013) e63754.
[17] D. Kihara, M. Kanehisa, Genome Res. 10 (2000) 731–743.
[18] I. Yanai, A. Derti, C. DeLisi, Proc. Natl. Acad. Sci. U.S.A. 98 (2001) 7940–7945.
[19] M. Pellegrini, E.M. Marcotte, M.J. Thompson, D. Eisenberg, T.O. Yeats, Proc. Natl. Acad. Sci. U.S.A. 96 (1999) 4285–4288.
[20] C. von Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, B. Snel, Nucleic Acids Res. 31 (2003) 258–261.
[21] B. Schwikowski, P. Uetz, S. Fields, Nat. Biotechnol. 18 (2000) 1257–1261.
[22] A. Vasquez, A. Flammini, A. Maritan, A. Vespignani, Nat. Biotechnol. 21 (2003) 697–700.
[23] B.R. Zeeberg, W. Feng, G. Wang, M.D. Wang, A.T. Fojo, M. Sunshine, S. Narasimhan, D.W. Kane, W.C. Reinhold, S. Lababidi, K.J. Bussey, J. Riss, J.C. Barnett, J.N. Weinstein, Genome Biol. 4 (2003) R28.
[24] K.D. Dahlquist, N. Salomonis, K. Vranizan, S.C. Lawlor, B.R. Conklin, Nat. Genet. 31 (2002) 19–20.
[25] M. Kanehisa, S. Goto, Y. Sato, M. Kawashima, M. Furumichi, M. Tanabe, Nucleic Acids Res. 42 (2014) D199–D205.
[26] M.L. Green, P.D. Karp, BMC Bioinform. 5 (2004) 76.
[27] L. Chen, D. Vitkup, Genome Biol. 7 (2006) R17.
[28] L. Sael, D. Kihara, Proteins 80 (2012) 1177–1195.
[29] X. Zhu, Y. Xiong, D. Kihara, Bioinformatics 31 (2015) 707–713.
[30] B. Hu, X. Zhu, Y. Xiong, M.G. Bures, D. Kihara, Int. J. Mol. Sci. 15 (2014) 15122–15145.
[31] W.-H. Shin, X. Zhu, M.G. Bures, D. Kihara, Molecules 20 (2015) 12841–12862.
[32] C. Chothia, A.M. Lesk, EMBO J. 5 (1986) 823–826.
[33] C.A. Wilson, J. Kreychman, M. Gerstein, J. Mol. Biol. 297 (2000) 233–249.
[34] D. Kihara, J. Skolnick, Proteins 55 (2004) 464–473.
[35] M. Brylinski, J. Skolnick, PLoS Comput. Biol. 5 (2009) e1000405.
[36] L. Heo, W.-H. Shin, M.S. Lee, C. Seok, Nucleic Acids Res. 42 (2014) W210–W214.
[37] C.T. Potter, G.J. Bartlett, J.M. Thornton, Nucleic Acids Res. 32 (2004) D129–133.
[38] M.P. Liang, D.R. Banatao, T.E. Klein, D.L. Brutlag, Nucleic Acids Res. 31 (2003) 3324–3327.
[39] K. Kinoshita, H. Nakamura, Bioinformatics 20 (2004) 1329–1330.
[40] M. Gao, J. Skolnick, Bioinformatics 29 (2013) 579–604.
[41] M. Brylinski, PLoS Comput. Biol. 10 (2014) e1003829.
[42] H.S. Lee, W. Im, J. Chem. Inf. Model. 52 (2012) 2784–2795.
[43] J. Ito, K. Ikeda, K. Yamada, K. Mizuguchi, K. Tomil, Nucleic Acids Res. 43 (2015) D392–398.
[44] A. Roy, J. Yang, Y. Zhang, Nucleic Acids Res. 40 (2012) W471–W477.
[45] A.T. Laurie, R.M. Jackson, Bioinformatics 21 (2005) 1908–1916.
[46] B. Huang, M. Schroeder, BMC Struct. Biol. 6 (2006) 19.
[47] B. Li, S. Turuvekere, M. Agrawal, D. La, K. Ramani, D. Kihara, Proteins 71 (2008) 670–683.
[48] K. Chen, M.J. Mizianty, J. Gao, L. Kurgan, Structure 19 (2011) 613–621.
[49] V. Le Guilloux, P. Schmidtke, P. Tuffery, BMC Bioinformatics 10 (2009) 168.
[50] M. Weisel, E. Proschak, G. Schneider, Chem. Cent. J. 1 (2007) 7.
[51] Y. Xiong, X. Zhu, D. Kihara, In silico Drug Discovery and Design Techniques, Future Science, London, UK, 2013. pp. 204–220.
[52] M.J. Keiser, B.L. Roth, B.N. Ambruster, P. Ernsberger, J.J. Irwin, B.K. Shoichet, Nat. Biotechnol. 25 (2007) 197–206.
[53] F. Milletti, A. Vulpetti, J. Chem. Inf. Model. 50 (2010) 1418–1431.
[54] A. Anighoro, J. Bojorath, G. Rastelli, J. Med. Chem. 57 (2014) 7874–7887.
[55] O. Trott, A.J. Olson, J. Comput. Chem. 31 (2010) 455–461.
[56] W.J. Allen, T.E. Balius, S. Mukherjee, S.R. Brozell, D.T. Moustakas, P.T. Lang, D.A. Case, I.D. Kuntz, R.C. Rizzo, J. Comput. Chem. 36 (2015) 1132–1156.
[57] R.A. Friesner, J.L. Banks, R.B. Murphy, T.A. Halgren, J.J. Klicic, D.T. Mainz, M.P. Repasky, E.H. Knoll, M. Shelley, J.K. Perry, D.E. Shaw, P. Francis, P.S. Shenkin, J. Med. Chem. 47 (2004) 1739–1749.
[58] G. Wolber, T. Langer, J. Chem. Inf. Model. 45 (2005) 160–169.
[59] S. Cross, M. Baroni, E. Carosati, P. Benedetti, S. Clementi, J. Chem. Inf. Model. 50 (2010) 1442–1450.
[60] E. Abrahamian, P.C. Fox, L. Naerum, I.R. Christensen, H. Thøgersen, R.D. Clark, J. Chem. Inf. Sci. 43 (2003) 458–468.
[61] N. Canterakis, Proceedings of 11th Scandinavian Conference on Image Analysis (1999) 85–93.
[62] M. Novotni, R. Klein, Proceedings of eighth ACM symposium on Solid modeling and applications (2003) 216–225.
[63] L. Sael, D. La, B. Li, R. Rustamov, D. Kihara, Proteins 73 (2008) 1–10.
[64] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, J.A. McCammon, Proc. Nat. Acad. Sci. U. S.A. 98 (2001) 10037–10041.
[65] L. Sael, D. Kihara, Int. J. Mol. Sci. 11 (2010) 5009–5026.
[66] M. Hattori, Y. Okuno, S. Goto, M. Kanehisa, J. Am. Chem. Soc. 125 (2003) 11853–11865.
[67] I. Wallach, R. Lilien, Bioinformatics 25 (2009) 615–620.
[68] L. Sael, D. Kihara, BMC Bioinformatics 13 (2012) S7.
[69] D. Kihara, L. Sael, R. Chikhi, J. Esquivel-Rodriguez, Curr. Protein Pept. Sci. 12 (2011) 520–530.
[70] R. Chikhi, L. Sael, D. Kihara, Proteins 78 (2010) 2007–2028.
[71] A. Kahraman, R.J. Morris, R.A. Laskowski, J.M. Thornton, J. Mol. Biol. 368 (2007) 283–301.
[72] K. Kinoshita, Y. Murakami, H. Nakamura, Nucleic Acids Res. 35 (2007) W398–W402.
[73] N.D. Gold, R.M. Jackson, J. Mol. Biol. 355 (2006) 1112–1124.
[74] R. Minai, Y. Matsuo, H. Onuki, H. Hirota, Proteins 72 (2008) 367–381.
[75] B. Xiong, J. Wu, D. Burk, M. Xue, H. Jiang, J. Shen, BMC Bioinformatics 11 (2010) 47.
[76] J. Kirchmair, G. Wolber, C. Laggner, T. Langer, J. Chem. Inf. Model. 46 (2006) 1848–1861.
[77] F. Gaudreault, M. Chartier, R. Najmanovich, Bioinformatics 28 (2012) i423–i430.
[78] N. Huang, B.K. Shoichet, J.J. Irwin, J. Med. Chem. 49 (2006) 6789–6801.
[79] B. Hu, M.A. Lill, J. Cheminform. 4 (2014) 14.
[80] B. Kramer, M. Rarey, T. Lengauer, Proteins 37 (1999) 228–241.
[81] R. Abagyan, M. Totrov, D. Kuznetsov, J. Comput. Chem. 15 (1994) 488–506.
[82] A. Jain, J. Med. Chem. 46 (2003) 499–511.
[83] D. Joseph-McCarthy, B.E. Thomas, M. Belmarsh, D. Moustakas, J.C. Alvarez, Proteins 51 (2003) 172–188.
[84] A.E. Cleves, A.N. Jain, J. Med. Chem. 49 (2006) 2921–2938.
[85] V. Venkatraman, P. Chakravarthy, D. Kihara, J. Cheminf. 1 (2009) 19.
[86] P.J. Ballester, W.G. Richards, J. Comput. Chem. 28 (2007) 1711–1723.
[87] P.C.D. Hawkins, A.G. Skillman, A. Nicholls, J. Med. Chem. 50 (2007) 74–82.
[88] I. Sillitoe, T.E. Lewis, A. Cuff, S. Das, P. Ashford, N.L. Dawson, N. Furnham, R.A. Laskowski, D. Lee, J.G. Lees, S. Lehtinen, R.A. Studer, J. Thornton, C.A. Orengo, Nucleic Acids Res. 43 (2015) D376–D381.
[89] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, Nucleic Acids Res. 28 (2000) 235–242.
[90] H. Fan, J.J. Irwin, B.M. Webb, G. Klebe, B.K. Shoichet, A. Sali, J. Chem. Inf. Model. 49 (2009) 2512–2527.
[91] Y. Zhang, J. Skolnick, Nucleic Acid Res. 33 (2005) 2302–2309.
[92] D.L. Bain, A.F. Heneghan, K.D. Connaghan-Jones, M.T. Miura, Annu. Rev. Physiol. 69 (2007) 201–220.
[93] A. Beautrait, A.S. Karaboga, M. Souchet, B. Maigret, Proteins 72 (2008) 873–882.
[94] Y. Jiang, J. Zou, C. Gui, J. Mol. Model. 11 (2005) 509–515.