Methods 93 (2016) 64–71

Contents lists available at ScienceDirect

Methods

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

Template-based identification of protein–protein interfaces using *e*FindSite^{PPI}

Surabhi Maheshwari [a], Michal Brylinski [a,b,*]

[a] Department of Biological Sciences, Louisiana State University, Baton Rouge, LA 70803, USA
[b] Center for Computation & Technology, Louisiana State University, Baton Rouge, LA 70803, USA

ARTICLE INFO

Article history:
Received 31 March 2015
Received in revised form 12 July 2015
Accepted 29 July 2015
Available online 30 July 2015

Keywords:
Protein–protein interactions
Protein interface prediction
Interfacial residues
Protein models
*e*FindSite^{PPI}

ABSTRACT

Protein–protein interactions orchestrate virtually all cellular processes, therefore, their exhaustive exploration is essential for the comprehensive understanding of cellular networks. A reliable identification of interfacial residues is vital not only to infer the function of individual proteins and their assembly into biological complexes, but also to elucidate the molecular and physicochemical basis of interactions between proteins. With the exponential growth of protein sequence data, computational approaches for detecting protein interface sites have drawn an increased interest. In this communication, we discuss the major features of *e*FindSite^{PPI}, a recently developed template-based method for interface residue prediction available at http://brylinski.cct.lsu.edu/efindsiteppi. We describe the requirements and installation procedures for the stand-alone version, and explain the content and format of output data. Furthermore, the functionality of the *e*FindSite^{PPI} web application that is designed to provide a simple and convenient access for the scientific community is presented with illustrative examples. Finally, we discuss common problems encountered in predicting protein interfaces and set forth directions for the future development of *e*FindSite^{PPI}.

1. Introduction

A wide range of biological processes are regulated by specific protein–protein interactions (PPIs) [1]. Anomalous interactions between endogenous proteins may severely disrupt cellular homeostasis, resulting in many disease states including cancer [2], Huntington's disease [3], cystic fibrosis [4], Alzheimer's disease [5] and cardiovascular disease [6]. Furthermore, interactions between host and pathogen proteins are essential components of viral and bacterial infections [7,8]. On that account, modulating PPIs is an important therapeutic strategy [9,10] with significant efforts devoted to identify, characterize and target PPIs involved in pathological states. A number of experimental techniques to detect PPIs have been developed, which can be broadly divided into *in vitro* (e.g., tandem affinity purification, affinity chromatography, co-immunoprecipitation, and protein-fragment complementation assays) and *in vivo* (e.g., yeast two-hybrid and synthetic lethality) methods [11]. Nonetheless, many of these approaches have important limitations including the high costs and long times of experiments, noisy data sets, and often high false positive and negative rates [12]. Because of the large diversity of PPIs, the successful design of novel therapeutics may also require atomic-level details on pharmacologically relevant complexes. Therefore, computational methods are increasingly becoming important to help and guide site-specific mutagenesis experiments [13–15] and to support the reconstruction of across-proteome protein interaction networks [16]. A diverse collection of algorithms are currently available to complement experimental efforts in studying PPIs, including homology-based approaches that employ sequence [17] and structure alignments [18], phylogenetic profiling [19], co-evolution methods [20], domain-pair exclusion analysis [21], interface residue prediction [22], and macromolecular docking [23].

The knowledge of even approximate locations of protein binding sites has a broad range of applications supporting further experimental and computational studies such as site-directed mutagenesis and complex structure assembly. Consequently, the prediction of interfacial sites and residues is a progressing area of research with a significant number of methods developed to date [24]. To improve the state-of-the-art in PPI interface prediction, particularly using computer-generated protein models, we recently developed *e*FindSite^{PPI}, an evolution/structure-based

* Corresponding author at: Department of Biological Sciences, Louisiana State University, Baton Rouge, LA 70803, USA.
E-mail address: michal@brylinski.org (M. Brylinski).

http://dx.doi.org/10.1016/j.ymeth.2015.07.017
1046-2023/

method for the identification of PPI residues from weakly homologous template structures [25]. *e*FindSite^{PPI} integrates sensitive meta-threading techniques with structure alignments and machine learning to locate putative interfacial sites in target proteins. A comprehensive comparative analysis of the performance of *e*FindSite^{PPI} with ProMate [26], PredUS [27], cons-PPISP [28], WHISCY [29], PriSE [30] and PINUP [31] demonstrated that *e*FindSite^{PPI} is highly accurate and outperforms many other interface prediction methods using not only experimental structures, but also computer-generated protein models [25,32]. We found that although structure-based prediction algorithms perform better than sequence-based methods, their accuracy strongly depends on the quality of query protein structures. However, in contrast to other structure-based algorithms, *e*FindSite^{PPI} tolerates small and moderate distortions in the input target structures. Furthermore, we also showed that combining the outputs from various prediction methods typically outperforms the best single algorithm, therefore, consensus predictions by meta-predictors are likely to significantly improve the accuracy of interface residue prediction [32].

*e*FindSite^{PPI} is available as a web server and a stand-alone software package at http://brylinski.cct.lsu.edu/efindsiteppi. The web application provides the scientific community with a user-friendly interface for job submission as well as the interpretation of results and data download. The stand-alone package can be installed locally for high throughput computations. In this communication, we describe the major features of *e*FindSite^{PPI} and present a typical procedure for PPI interface prediction.

2. Calculation

A detailed description of the algorithm implemented in *e*FindSite^{PPI} as well as training and testing procedures are provided in the original paper [25]. A concise protocol is presented in Fig. 1. For a given query protein (Fig. 1A), *e*FindSite^{PPI} employs a collection of evolutionary and structurally related templates identified by meta-threading using *e*Thread to calculate several residue-level features. These features characterizing interfacial residues can be broadly classified into three categories: sequence-, structure- and residue-based (Fig. 1B). Specifically, each surface residue is assigned (1) a relative accessible area, (2) a generic interface propensity, (3) sequence entropy, (4) a position-specific interface propensity, and (5) the fraction of templates that have an equivalent residue at the protein–protein interface. It is well known that individual features cannot unambiguously distinguish between interfacial and non-interfacial residues [33]. To address this issue, *e*FindSite^{PPI} combines individual attributes using non-linear machine learning models, Support Vector Machines (SVM) [34] and a Naïve Bayes Classifier (NBC) [35] (Fig. 1C). Both classifiers assign each surface residue in the query protein with a calibrated probability of being at the protein–protein interface (Fig. 1D).

The performance of *e*FindSite^{PPI} was benchmarked on a dataset of 1905 proteins dimers [25]. We used three structural forms for each receptor, a crystal structure and computer-generated high- and moderate-quality models. Although the accuracy of *e*FindSite^{PPI} decreases from experimental to modeled structures, we demonstrated that it tolerates to some extent distortions in theoretical target structures [25]. Therefore, in many cases, interface residues can be fairly accurately inferred even from moderate-quality models. In a subsequent study, we carried out comparative benchmarks of *e*FindSite^{PPI} and nine other web servers against the same dataset of different quality target structures [32]. The results indicate that *e*FindSite^{PPI} is one of the best performing on-line services for PPI prediction, particularly using computer-generated structures.

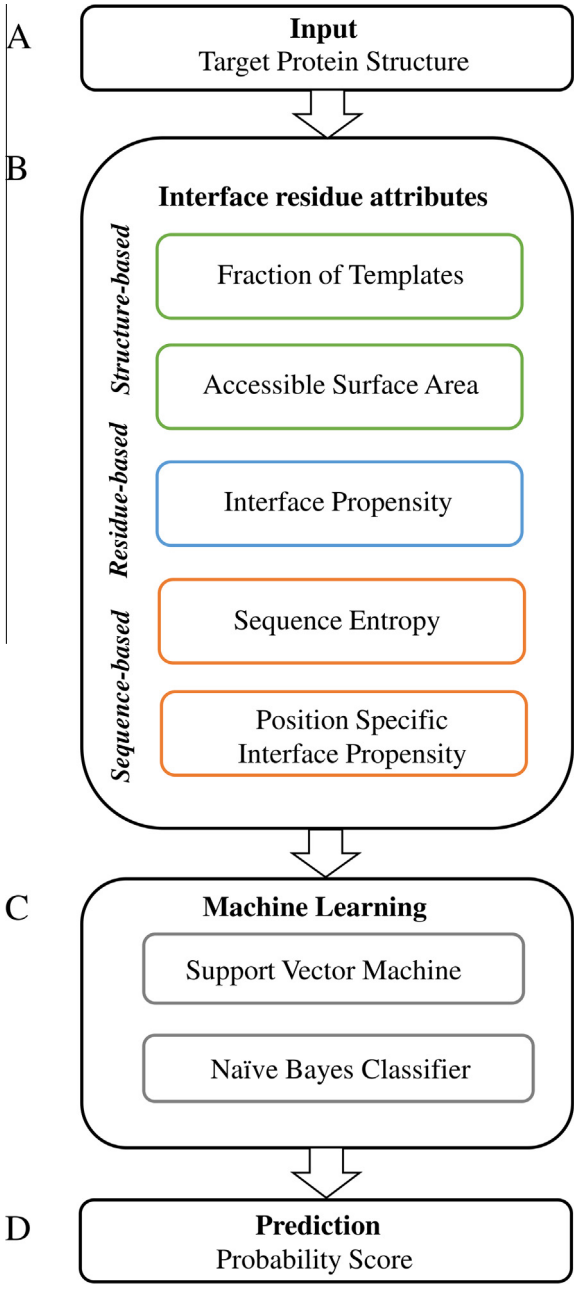

Fig. 1. General flowchart of *e*FindSite^{PPI}. The prediction process starts with the query structure (A). Using a set of dimer templates, individual surface residues in the query protein are assigned a series of structure- (green), residue- (blue) and sequence-based (orange) features (B). These feature vectors are subsequently passed to machine learning (C) that assigns query residues with the final interfacial probability scores (D).

3. Web server

The web application of *e*FindSite^{PPI} features a user interface comprising three major components, job submission, status check and results page. The webserver itself is implemented in PHP using Drupal, an open source content management system. Below, we briefly describe the major components of the *e*FindSite^{PPI} webserver.

3.1. Job submission

*e*FindSite^{PPI} has a simple and intuitive user interface. Using the "New submission" option, users can upload a three-dimension structure of the target protein in the Protein Data Bank (PDB) [36] format. In the absence of an experimentally determined structure of a target protein, users can upload a computer-generated model as well. In addition, the results of protein structure modeling using *e*Thread can be transferred directly to *e*FindSite^{PPI} using the "Use *e*Thread model" option. *e*FindSite^{PPI} webserver accepts proteins 50–600 residues long containing a single polypeptide chain.

3.2. Job processing

Once a job is submitted, the user is redirected to a webpage reporting the job status (queued, running, finished) and the automatically generated unique ticket number. The status page is automatically refreshed every hour until the job is completed. Moreover, users can check the status of their jobs anytime using the "Job tracking" box available from the right sidebar. Note that the simulation time of *e*FindSite^{PPI} depends on several factors including the length of the query protein, the number of templates, and the workload on our computer cluster. Typically, the results should be ready within 1–3 days.

Prior to the PPI prediction, the query structure undergoes a quick quality check. Because energy calculations and surface definition may be significantly affected by problems with atomic coordinates, the program ctrip from the Jackal modeling package [37] is used to reconstruct side chains and add missing atoms. Furthermore, if multiple configurations or rotamers for any residue are present in the PDB file, one configuration is selected by ctrip based on predetermined distance geometry constraints. As a result, a modified version of the submitted PDB file is generated and used as the actual input for *e*FindSite^{PPI}; this file is available for download from the result web page.

In addition to the PDB file containing the query structure, *e*FindSite^{PPI} requires two other input files, a query sequence profile and a list of structurally and evolutionary related templates. Users interested in the stand-alone version of *e*FindSite^{PPI} can obtain these files from most protein threading programs; the web server generates them automatically. For the webserver, sequence profiles are calculated using Sparks2 [38] and the list of templates is compiled using meta-threading by *e*Thread [39]. Sequence profiles for the stand-alone version of *e*FindSite^{PPI} can be calculated by PROFILpro [40], whereas the list of templates can be compiled using any protein threading program, e.g., HHpred [41], COMPASS [42], SP3 or Sparks2 [38]. For example, Fig. 2 shows Receiver Operating Characteristics (ROC) plots evaluating the performance of *e*FindSite^{PPI} using templates identified by several threading programs. The ROC plots clearly suggest that the performance of *e*FindSite^{PPI} using different threading methods is fairly comparable, thus users have the flexibility to pick any threading method of their choice.

3.3. Output

*e*FindSite^{PPI} reports a list of putative interface residues and interaction types. This data is available for download as three separate files, a PDB file containing the query structure after the quality check, a text file with the detailed information on the predicted interface residues, and a file that contains structure alignment between the query and template proteins. Moreover, a log file is displayed on the results page to reveal any errors encountered during the prediction process.

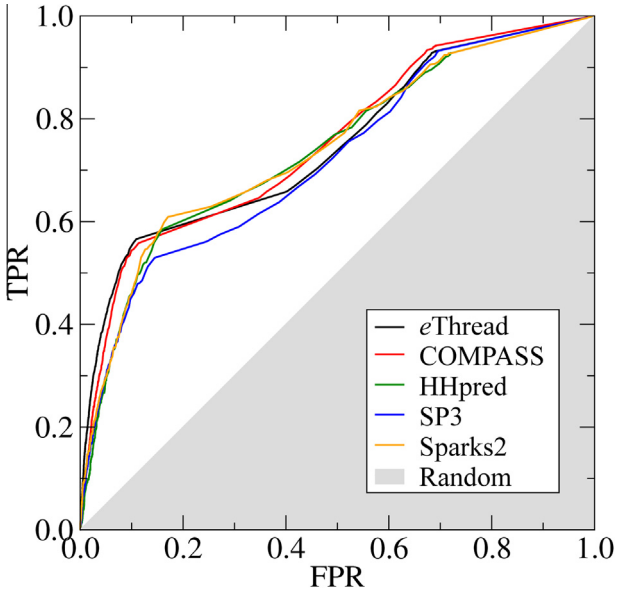

Fig. 2. ROC plots assessing the accuracy of interface residue prediction by *e*FindSite^{PPI}. The performance using templates identified by *e*Thread is compared to those using single threading methods, COMPASS, HHpred, SP3 and Sparks2. TPR – true positive rate, FPR – false positive rate; gray area corresponds to predictions no better than random.

3.3.1. Interface data

Putative interface residues are identified based on probability scores calculated using a set of structure and sequence features. The interface prediction file contains various scores assigned to all surface residues with those contributing to putative interfaces marked by asterisks. This file is divided into several sections shown in Fig. 3, which can be easily parsed using the following keywords:

`LIBRARY`: The version of the template library used by *e*FindSite^{PPI} (Fig. 3A).

`CONFDNC`: The prediction confidence, which can be high, medium or low (Fig. 3A). Note that representative benchmarking calculations show that these confidence estimates correlate with the prediction accuracy. The average Matthew's Correlation Coefficient (MCC) of interface residue prediction for high-, moderate- and low-confidence cases is 0.62, 0.39 and 0.13, respectively [25].

`RESIDUE`: This section provides the probability for each solvent-accessible residue in the query protein to be at the interface (Fig. 3A); seven columns contain the following information:

1. Confidently predicted interface residues are marked by `<*>`.
2. Surface residue index `(12)`.
3. Residue 3-letter code `(GLU)`.
4. Residue number `(18)`.
5. The fraction of templates with an interface residue at the structurally aligned position `(0.85714)`.
6. Probability score from SVM `(0.64595)`.
7. Probability score from NBC `(0.998953)`.

`TEMPLTE`: This section provides information on template proteins and their alignments to the query (Fig. 3B); seven columns contain the following information:

1. PDB-ID of the template `(1B7BA)`.
2. The number of residues in the template `(307)`.
3. TM-score to the query structure `(0.734)`.
4. Cα-RMSD in Å of the aligned region `(3.16)`.
5. The number of residues aligned by Fr-TM-align `(220)`.

A
```
LIBRARY oct2013
CONFDNC HIGH
RESIDUE  <*>   12 GLU   18  0.85714  0.64595  0.998953
```

B
```
TEMPLTE 1B7BA  307  0.734   3.16  220  0.252  0.139
TEMPLTE 1B8AA  438  0.757   2.87  282  0.301  0.191
```

C
```
ROTMTRX 1B7BA -16.7878719388   0.1752797462    0.5198313923    (cut)
ROTMTRX 1ASYA 122.0554573581  -0.8768106729   -0.4512099891    (cut)
ROTMTRX 1B8AA 28.3978494539   -0.5639774603   0.8170456866     (cut)
```

D
```
INTRCTN HBND <*> GLU    69      0.200000
INTRCTN SALT <*> GLU    28      0.625000
INTRCTN HYFB     ALA    147     0.062500
INTRCTN AROM <*> TYR    2       0.125000
```

E
```
>3SBXG 177 144 0.401 4.87 0.280
RWTVAVYCAAAP--THPE-----LLE--LAGAVGAAIAARGW-TLVWGG (cut)
.....:...    .::.     .::  :::::::::::::. ..:..   (cut)
-HMIILKLGG--SVITRKDSEEPAIDRDNLERIASEIGNASPSSLMIV- (cut)
*
```

Fig. 3. Prediction data included in the output files generated by *e*FindSite^{PPI}. (A) The version of *e*FindSite^{PPI} library, the prediction confidence, and the list of surface residues assigned various scores. (B) The list of template dimers used to predict interface residues and their global similarities to the query protein. (C) Translation vectors and rotation matrices to structurally align template proteins onto the query. (D) The list of putative interaction types for the predicted interface residues. (E) A sample structure alignment between a template and the query protein.

6. The global sequence identity to the target (0.252).
7. The sequence identity calculated over residues aligned by Fr-TM-align (0.139).

ROTMTRX: This section keeps record of a space-separated translation vector (3 values) and rotation matrix (9 values) that can be used to structurally align each template onto the query (Fig. 3C). The superposition of template atoms can be perform using the following transformations:

$$x_{sup} = value1 + value2 \times x_{lib} + value3 \times y_{lib} + value4 \times z_{lib}$$

$$y_{sup} = value5 + value6 \times x_{lib} + value7 \times y_{lib} + value8 \times z_{lib}$$

$$z_{sup} = value9 + value10 \times x_{lib} + value11 \times y_{lib} + value12 \times z_{lib}$$

where the original coordinates of template atoms in the *e*FindSite^{PPI} library are marked by a subscript *lib* and those superposed onto the query are marked by a subscript *sup*. *Value*1 to *value*12 correspond to numerical values in each column (from left to right) of the ROTMTRX section.

INTRCTN: Predicted residue interactions are listed in this section (Fig. 3D); five columns contain the following information:

1. The interaction type (HBND – hydrogen bond, SALT – salt bridge, HYFB – hydrophobic interaction, AROM – aromatic interaction).
2. Confidently predicted interactions are marked by <*>.
3. Residue 3-letter code (GLU).
4. Residue number (69).
5. Interaction probability score (0.200000).

3.3.2. Structure alignments

Template-to-target structure alignments constructed by Fr-TM-align are reported in a PIR-like format. Fig. 3E shows an example of a single alignment (the sequences are cut short for demonstration), where the first row provides several values separated by spaces:

1. PDB-ID of the template (3SBXG).
2. The template length (177).
3. The alignment length (144).
4. TM-score to the query (0.401).
5. Cα-RMSD in Å calculated over aligned residues (4.87).
6. The sequence identity over aligned residues (0.280).

The second and forth lines show the aligned sequences of the query and the template, respectively, whereas the third line highlights the aligned residue positions (.) and those residue pairs whose Cα atoms are aligned within a distance of 5Å (:). Asterisks (*) separate alignments for individual templates.

3.3.3. On-line visualization of the results

In addition to numerical results that can be downloaded from the results web page, the *e*FindSite^{PPI} web server features a graphical interface displaying the prediction data using a web browser. Fig. 4 shows a snapshot of the results web page generated for an example protein, glutathione isopentenyl phosphate kinase (PDB-ID: 3ll9, chain B) [43]. The first section (Fig. 4A) lists general information on the query, such as the user assigned target ID, automatically generated job ticket, the version of *e*FindSite^{PPI} template

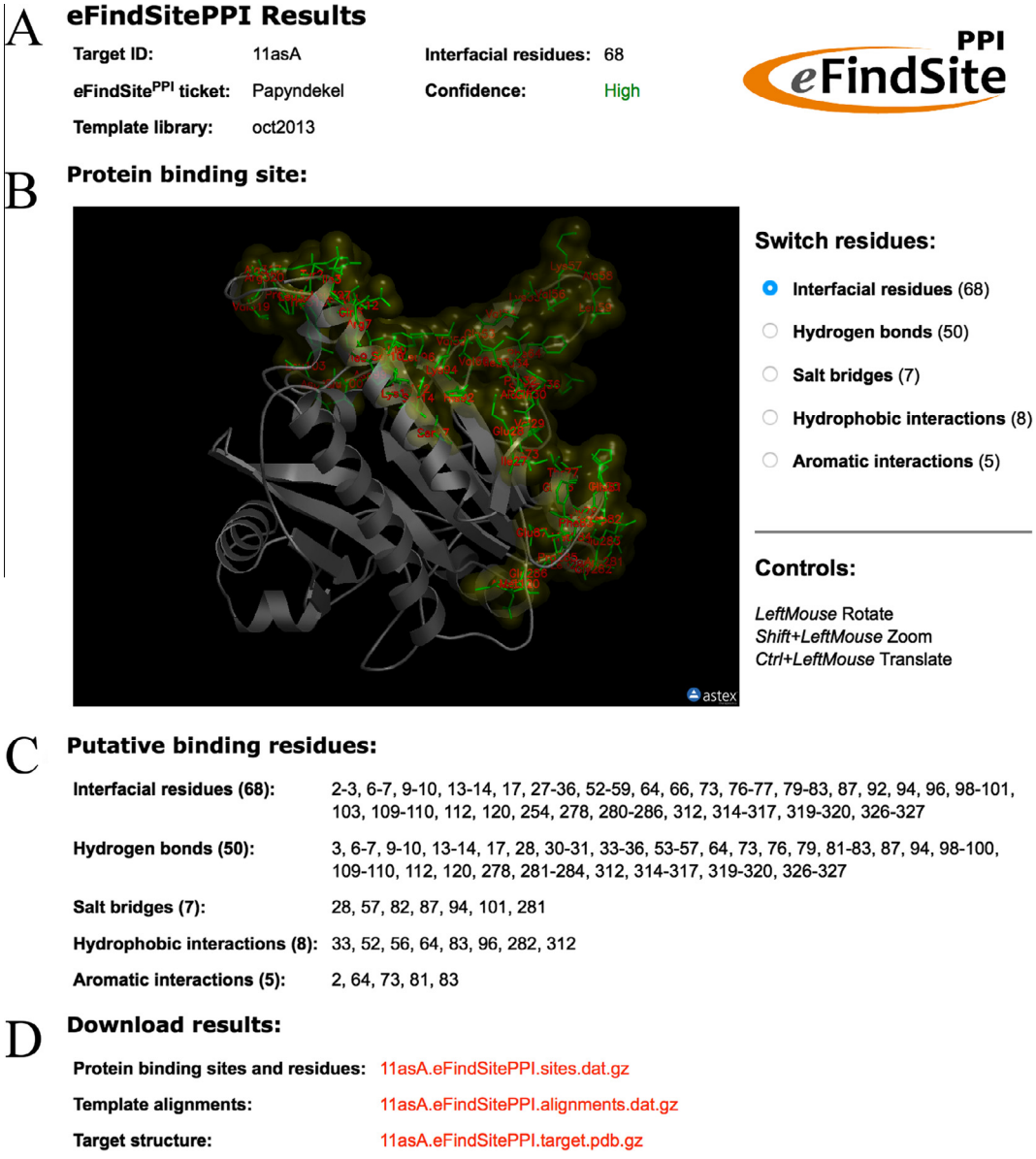

Fig. 4. Screenshot of the *e*FindSite[PPI] results web page. (A) Job information including a unique ticket that can be used to track the job and retrieve results, the version of *e*FindSite[PPI] library, the number of predicted interface residues, and the overall prediction confidence. (B) A molecular viewer showing putative interface residues and interactions mapped onto the query structure. (C) The list of putative binding residues and interactions. (D) Links to the downloadable files generated by the *e*FindSite[PPI] web server.

library, the number of identified interface residues, and the prediction confidence. Interface residues and specific molecular interactions are visualized using a Java applet, AstexViewer [44] (Fig. 4B). This section also includes a set of radio buttons corresponding to the predicted interaction types such as hydrogen bonds, salt bridges, hydrophobic interactions and aromatic contacts for putative interfacial residues. Once a particular interaction type is selected, the corresponding interface residues are highlighted as sticks with a transparent surface and labeled. This feature allows users to easily map prediction data to the submitted query structure. The lists of predicted interface residues and molecular interactions are reported below (Fig. 4C). Finally, the last section provides hyperlinks to numerical data that can be downloaded for a local analysis (Fig. 4D).

4. Standalone software package

Interface residue prediction across large protein datasets typically requires high-throughput computations, therefore, we offer a stand-alone version of *e*FindSite[PPI] that can be installed locally on any machine running Linux operating system. Below, we provide installation instructions and describe parameters that can be modified when using individual programs included in the *e*FindSite[PPI] stand-alone package.

4.1. Installation and requirements

*e*FindSite[PPI] software is implemented in Perl. Local installation breaks down into three steps. First, the following Perl modules

from the Comprehensive Perl Archive Network (CPAN) need to be installed: File::Temp, File::Slurp, File::Copy, Compress::Zlib, AI::NaiveBayes1, List::Util, Algorithm::NeedlemanWunsch, Benchmark, Cwd, and YAML. Next, users should obtain and install several programs that are free for academic and non-commercial use: LIBSVM [34] for machine learning, NACCESS [45] for the calculation of accessible surface area, Fr-TM-align [46] for structure alignments, and a protein threading program to compile a list of evolutionarily related templates and to construct sequence profiles for a query protein. Finally, users can download and install the latest version of *e*FindSite^{PPI} software and the template library.

4.2. Programs

*e*FindSite^{PPI} software distribution includes two programs, efindsiteppi and efindsiteppi_map. Each program can be executed without arguments to display help information and the list of available options; below is the full description of these programs.

4.2.1. efindsiteppi

This is the main program that predicts interfacial sites, residues, and interactions for a given query structure from protein threading data. efindsiteppi requires the following arguments:

- `s query_structure`, where `query_structure` is either experimental or computer-generated structure of the query protein in the PDB format.
- `t template_list`, where `template_list` is the list of template proteins complied by threading software.
- `e seq_prf`, where `seq_prf` is the query sequence profile also generated by threading software.
- `o output_name`, where `output_name` is used to save interface prediction results in two output files with different extensions.

In addition to these mandatory arguments, efindsiteppi offers several optional parameters to modify various prediction thresholds:

- `b seq_cut`, where `seq_cut` is a sequence identity threshold between the query and template proteins. Note that this argument should be used only for benchmarks. The default value is 1.0, which means that all identified templates will be used to predict interfacial residues.
- `m tm_score`, where `tm_score` is a TM-score threshold between the query and template proteins. The default value is 0.4, which means that only these template structures whose TM-score to query is $\geqslant$0.4 will be used.
- `x template_max`, where `template_max` is the maximum number of templates used in the prediction procedure; the default number if 1000.

efindsiteppi generates two output files, `output_name.sites.dat` containing detailed information on the predicted interface residues and interactions, the list of templates, rotation matrices to align templates onto the query, the prediction confidence, etc., and `output_name.alignments.dat` reporting structure alignments constructed by fr-TM-align.

4.2.2. efindsiteppi_map

This script prepares a mapping file linking threading templates to protein dimer templates used by *e*FindSite^{PPI}. Essentially, it offers a possibility to run *e*FindSite^{PPI} with any protein threading software. The mandatory arguments to efindsiteppi_map are:

- `t thread_lib`, where `thread_lib` is a complete threading library in FASTA format.
- `p efindsiteppi_lib`, where `efindsiteppi_lib` is the *e*FindSite^{PPI} library in FASTA format.
- `o output_file`, where `output_file` is the mapping file linking threading and *e*FindSite^{PPI} libraries.

Furthermore, an optional argument `–a proc_num` can be set to the desired number of processors to be used; by default, `proc_num` is set to 1. Note that in order to use efindsiteppi_map, users need to have NCBI BLAST installed with formatdb and blastall programs available from the default search path. efindsiteppi_map generates only one output file, which is a mapping file required by efindsiteppi.

5. Possible bottlenecks

*e*FindSite^{PPI} does not provide PPI predictions for all protein targets, therefore, we would like to make users aware of possible causes. The main reason is usually the absence of structurally related weakly homologous templates in their bound conformational state. Note that this is a general limitation of template-based approaches, however, we may expect the coverage of suitable targets to continuously expand given the exponential growth of structure databases. When suitable quaternary template structures cannot be identified for a query protein, alternative methods can be used. For example PrISE [30] is a recently developed interface prediction method that exploits local surface similarities by utilizing a repository of structural elements extracted from complexes in the PDB. Apart from the availability of evolutionarily and structurally related templates, the performance of *e*FindSite^{PPI} also depends of two other factors; the first is the quality of a query structure. Previous benchmarking calculations demonstrated that *e*FindSite^{PPI} to some extent tolerates distortions in modeled query structures, nonetheless, the accuracy of *e*FindSite^{PPI} is still better for experimentally determined structures compared to computer-generated models. The second factor is the type of association formed by the query protein. Similar to other prediction programs, e.g., ET [47] and iJET [48], the overall performance of *e*FindSite^{PPI} for homo-complexes is notably better than that for hetero-complexes. This is because homo-complexes often have a nearly perfect symmetric organization at the interface in contrast to smaller, asymmetric interfaces formed by hetero-complexes. Less frequently, *e*FindSite^{PPI} may give poor predictions for "promiscuous" proteins interacting with a diverse set of substrates at multiple interfaces. Such ambiguous cases are problematic in general for many template-based PPI prediction methods. Nevertheless, our benchmarks show that meaningful predictions can be obtained for the majority of cases.

6. Future work

*e*FindSite^{PPI} is an actively maintained project that undergoes regular updates and improvements extending its functionality. The current implementation employs a general template library that is used to predict PPI residues for both homo- and hetero-complexes. However, since hetero-dimers are largely underrepresented in the PDB, the template library is dominated by homo-dimer structures. In turn, this may decrease the performance of PPI residue prediction for hetero-dimer targets. To address this issue, future work includes the development of separate template libraries for homo- and hetero-dimers. We will use higher sequence identity thresholds to account for the large diversity of hetero-complexes observed in the PDB. Furthermore,

machine learning models in *e*FindSite^{PPI} will be re-trained separately for homo- and hetero-interfaces in order to improve the accuracy of PPI annotations for both types of assemblies. In addition to the prediction of interface residues and specific molecular interactions currently featured by *e*FindSite^{PPI}, we plan to cover other functional aspects as well. For instance, further annotations will include the identification of hot-spot residues, which are critical for the design of pharmaceuticals targeting protein–protein interfaces [49,50]. Finally, we are going to support a number of widely used protein threading/fold recognition programs for template identification, including HHpred [41], PSI-BLAST [51], RaptorX [52], and SparksX [53]. The prediction procedures in *e*FindSite^{PPI} will be customized allowing users to select threading software of their choice.

7. Availability

*e*FindSite^{PPI} is freely available to the scientific community at http://brylinski.cct.lsu.edu/efindsiteppi. Researchers interested in *e*FindSite^{PPI} can either use the web server or install the software locally for high-throughput computations. Moreover, the web site offers a manual that provides detailed installation instructions and includes illustrative examples and step-by-step tutorials. Large benchmarking datasets and the corresponding numerical results are also available for download to facilitate comparative studies with other prediction methods.

Acknowledgement

This study was supported by the Louisiana Board of Regents through the Board of Regents Support Fund [contract LEQSF(2012-15)-RD-A-05].

References

[1] J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G.F. Berriz, F.D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem, S. Milstein, J. Rosenberg, D.S. Goldberg, L.V. Zhang, S.L. Wong, G. Franklin, S. Li, J.S. Albala, J. Lim, C. Fraughton, E. Llamosas, S. Cevik, C. Bex, P. Lamesch, R.S. Sikorski, J. Vandenhaute, H.Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M.E. Cusick, D.E. Hill, F.P. Roth, M. Vidal, Towards a proteome-scale map of the human protein–protein interaction network, Nature 437 (7062) (2005) 1173–1178.
[2] A.L. Garner, K.D. Janda, Protein–protein interactions and cancer: targeting the central dogma, Curr. Top. Med. Chem. 11 (3) (2011) 258–280.
[3] S.-H. Li, X.-J. Li, Huntingtin–protein interactions and the pathogenesis of Huntington's disease, Trends Genet. 20 (3) (2004) 146–154.
[4] I. Devesa, G. Fernández-Ballester, A. Ferrer-Montiel, Targeting protein–protein interactions to rescue Δf508-cftr: a novel corrector approach to treat cystic fibrosis, EMBO Mol. Med. 5 (10) (2013) 1462–1464.
[5] M. Soler-López, A. Zanzoni, R. Lluís, U. Stelzl, P. Aloy, Interactome mapping suggests new mechanistic details underlying Alzheimer's disease, Genome Res. 21 (3) (2011) 364–376.
[6] L.C.Y. Lee, D.H. Maurice, G.S. Baillie, Targeting protein–protein interactions within the cyclic AMP signaling system as a therapeutic strategy for cardiovascular disease, Future Med. Chem. 5 (4) (2013) 451–464.
[7] P. Uetz, Y.-A. Dong, C. Zeretzke, C. Atzler, A. Baiker, B. Berger, S.V. Rajagopala, M. Roupelieva, D. Rose, E. Fossum, J. Haas, Herpesviral protein networks and their interaction with the human proteome, Science 311 (5758) (2006) 239–242.
[8] N. Simonis, J.-F. Rual, I. Lemmens, M. Boxus, T. Hirozane-Kishikawa, J.-S. Gatot, A. Dricot, T. Hao, D. Vertommen, S. Legros, S. Daakour, N. Klitgord, M. Martin, J.-F. Willaert, F. Dequiedt, V. Navratil, M.E. Cusick, A. Burny, C. Van Lint, D.E. Hill, J. Tavernier, R. Kettmann, M. Vidal, J.-C. Twizere, Host–pathogen interactome mapping for HTLV-1 and -2 retroviruses, Retrovirology 9 (2012) 26.
[9] J.A. Wells, C.L. McClendon, Reaching for high-hanging fruit in drug discovery at protein–protein interfaces, Nature 450 (7172) (2007) 1001–1009.
[10] H. Jubb, A.P. Higueruelo, A. Winter, T.L. Blundell, Structural biology and drug discovery for protein–protein interactions, Trends Pharmacol. Sci. 33 (5) (2012) 241–248.
[11] V.S. Rao, K. Srinivas, G.N. Sujini, G.N.S. Kumar, Protein–protein interaction detection: methods and analysis, Int. J. Proteomics 2014 (2014) 147648.
[12] S. Fields, High-throughput two-hybrid analysis. The promise and the peril, FEBS J. 272 (21) (2005) 5391–5399.
[13] M.E. Sowa, W. He, T.G. Wensel, O. Lichtarge, A regulator of G protein signaling interaction surface linked to effector specificity, Proc. Natl. Acad. Sci. USA 97 (4) (2000) 1483–1488.
[14] M.E. Sowa, W. He, K.C. Slep, M.A. Kercher, O. Lichtarge, T.G. Wensel, Prediction and confirmation of a site critical for effector regulation of RGS domain activity, Nat. Struct. Biol. 8 (3) (2001) 234–237.
[15] T. Kortemme, D.E. Kim, D. Baker, Computational alanine scanning of protein-protein interfaces, Sci. STKE 219 (2004) (2004) pl2.
[16] Q.C. Zhang, D. Petrey, R. Norel, B.H. Honig, Protein interface conservation across structure space, Proc. Natl. Acad. Sci. USA 107 (24) (2010) 10896–10901.
[17] Y. Murakami, K. Mizuguchi, Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites, Bioinformatics 26 (15) (2010) 1841–1848.
[18] Q.C. Zhang, D. Petrey, L. Deng, L. Qiang, Y. Shi, C.A. Thu, B. Bisikirska, C. Lefebvre, D. Accili, T. Hunter, T. Maniatis, A. Califano, B. Honig, Structure-based prediction of protein–protein interactions on a genome-wide scale, Nature 490 (7421) (2012) 556–560.
[19] M. Pellegrini, E.M. Marcotte, M.J. Thompson, D. Eisenberg, T.O. Yeates, Assigning protein functions by comparative genome analysis: protein phylogenetic profiles, Proc. Natl. Acad. Sci. USA 96 (8) (1999) 4285–4288.
[20] F. Pazos, A. Valencia, Similarity of phylogenetic trees as indicator of protein–protein interaction, Protein Eng. 14 (9) (2001) 609–614.
[21] B.A. Shoemaker, A.R. Panchenko, Deciphering protein–protein interactions. Part II. Computational methods to predict protein and domain interaction partners, PLoS Comput. Biol. 3 (4) (2007) e43.
[22] S. Maheshwari, M. Brylinski, Predicting protein interface residues using easily accessible on-line resources, Brief. Bioinform. (2015), http://dx.doi.org/10.1093/bib/bbv009.
[23] I.A. Vakser, Protein–protein docking: from interaction to interactome, Biophys. J. 107 (8) (2014) 1785–1793.
[24] I. Ezkurdia, L. Bartoli, P. Fariselli, R. Casadio, A. Valencia, M.L. Tress, Progress and challenges in predicting protein–protein interaction sites, Brief. Bioinform. 10 (3) (May 2009) 233–246.
[25] S. Maheshwari, M. Brylinski, Prediction of protein–protein interaction sites from weakly homologous template structures using meta-threading and machine learning, J. Mol. Recognit. 28 (1) (2015) 35–48.
[26] H. Neuvirth, R. Raz, G. Schreiber, ProMate: a structure based prediction program to identify the location of protein–protein binding sites, J. Mol. Biol. 338 (1) (2004) 181–199.
[27] Q.C. Zhang, L. Deng, M. Fisher, J. Guan, B. Honig, D. Petrey, PredUs: A web server for predicting protein interfaces using structural neighbors, Nucleic Acids Res. 39 (Suppl. 2) (2011).
[28] H. Chen, H.X. Zhou, Prediction of interface residues in protein–protein complexes by a consensus neural network method: Test against NMR data, Proteins Struct. Funct. Genet. 61 (1) (2005) 21–35.
[29] S.J. Vries, A.D.J. Dijk, A.M.J.J. Bonvin, WHISCY: what information does surface conservation yield?, Appl Data-Driven Docking 489 (2006) 479–489.
[30] R.A. Jordan, Y. El-Manzalawy, D. Dobbs, V. Honavar, Predicting protein–protein interface residues using local surface structural similarity, BMC Bioinformatics 13 (1) (2012) 41.
[31] S. Liang, C. Zhang, S. Liu, Y. Zhou, Protein binding site prediction using an empirical scoring function, Nucleic Acids Res. 34 (13) (2006) 3698–3707.
[32] S. Maheshwari, M. Brylinski, Prediction protein–protein interface using easily accessible on-line resources, Brief Bioinform (2015), http://dx.doi.org/10.1093/bib/bbv009.
[33] S. Jones, J.M. Thornton, Analysis of protein–protein interaction sites using surface patches, J. Mol. Biol. 272 (1) (1997) 121–132.
[34] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 1–39.
[35] H. Zhang, The optimality of Naive Bayes, Mach. Learn. 1 (2) (2004) 3.
[36] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank, Nucleic Acids Res. 28 (1) (2000) 235–242.
[37] Z. Xiang, B. Honig, Extending the accuracy limits of prediction for side-chain conformations, J. Mol. Biol. 311 (2) (2001) 421–430.
[38] H. Zhou, Y. Zhou, SPARKS 2 and SP3 servers in CASP6, Proteins 61 (Suppl. 7) (2005) 152–156.
[39] M. Brylinski, D. Lingam, EThread: a highly optimized machine learning-based approach to meta-threading and the modeling of protein tertiary structures, PLoS ONE 7 (11) (2012) e50200.
[40] J. Cheng, M.J. Sweredoski, P. Baldi, DOMpro: protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks, Data Min. Knowl. Disc. 13 (1) (2006) 1–10.
[41] J. Söding, A. Biegert, A.N. Lupas, The HHpred interactive server for protein homology detection and structure prediction, Nucleic Acids Res. 33 (Web Server issue) (2005) W244–W248.
[42] R. Sadreyev, N. Grishin, COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance, J. Mol. Biol. 326 (1) (2003) 317–336.
[43] M.F. Mabanglo, H.L. Schubert, M. Chen, C.P. Hill, C.D. Poulter, X-ray structures of isopentenyl phosphate kinase, ACS Chem. Biol. 5 (5) (2010) 517–527.
[44] M.J. Hartshorn, AstexViewer: a visualisation aid for structure-based drug design, J. Comput. Aided Mol. Des. 16 (12) (2002) 871–881.
[45] S. Hubbard, J. Thomson, NACCESS, Dep. Biochem. Mol. Biol. (1993).
[46] S.B. Pandit, J. Skolnick, Fr-TM-align: a new protein structural alignment method based on fragment alignments and the TM-score, BMC Bioinformatics 9 (2008) 531.

[47] O. Lichtarge, H.R. Bourne, F.E. Cohen, An evolutionary trace method defines binding surfaces common to protein families, J. Mol. Biol. 257 (2) (1996) 342–358.
[48] S. Engelen, L.a. Trojan, S. Sacquin-Mora, R. Lavery, A. Carbone, Joint evolutionary trees: a large-scale method to predict protein interfaces based on sequence sampling, PLoS Comput. Biol. 5 (1) (2009) e1000267.
[49] E. Cukuroglu, H.B. Engin, A. Gursoy, O. Keskin, Hot spots in protein–protein interfaces: towards drug discovery, Prog. Biophys. Mol. Biol. 116 (2–3) (2014) 165–173.
[50] B. Ma, R. Nussinov, Druggable orthosteric and allosteric hot spots to target protein–protein interactions, Curr. Pharm. Des. 20 (8) (2014) 1293–1301.
[51] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (17) (1997) 3389–3402.
[52] M. Källberg, G. Margaryan, S. Wang, J. Ma, J. Xu, RaptorX server: a resource for template-based protein structure modeling, Methods Mol. Biol. 1137 (2014) 17–27.
[53] Y. Yang, E. Faraggi, H. Zhao, Y. Zhou, Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates, Bioinformatics 27 (15) (2011) 2076–2082.