Bioinformatics, 31(12), 2015, 1945–1950
doi: 10.1093/bioinformatics/btv077
Advance Access Publication Date: 4 February 2015
Original Paper

Structural bioinformatics

Evolutionary profiles improve protein–protein interaction prediction from sequence

Tobias Hamp and Burkhard Rost*

Department of Informatics, Bioinformatics & Computational Biology, TUM (Technische Universität München)—I12, Boltzmannstr. 3, 85748 Garching/Munich, Germany

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on November 5, 2014; revised on January 15, 2015; accepted on February 2, 2015

Abstract

Motivation: Many methods predict the physical interaction between two proteins (protein-protein interactions; PPIs) from sequence alone. Their performance drops substantially for proteins not used for training.

Results: Here, we introduce a new approach to predict PPIs from sequence alone which is based on evolutionary profiles and profile-kernel support vector machines. It improved over the state-of-the-art, in particular for proteins that are sequence-dissimilar to proteins with known interaction partners. Filtering by gene expression data increased accuracy further for the few, most reliably predicted interactions (low recall). The overall improvement was so substantial that we compiled a list of the most reliably predicted PPIs in human. Our method makes a significant difference for biology because it improves most for the majority of proteins without experimental annotations.

Availability and implementation: Implementation and most reliably predicted human PPIs available at https://rostlab.org/owiki/index.php/Profppikernel.

Contact: rost@in.tum.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

PPIs: physical protein–protein interactions between different proteins. We define PPIs as interactions that bring two different proteins A and B directly into 'physical contact'. This 'molecular' perspective on PPIs differs from the view adopted by many 'users' of interaction data who look for associations. Again: for us the crucial aspect of an interaction is the direct physical contact. This molecular perspective is a crucial component of curated resources such as Hippie (Schaefer, *et al.*, 2012), as well as of our method.

Predictions and experimental evidence intertwined. PPIs are supported by an increasing amount of data, derived for example from sequences, structures, co-evolution, co-expression, domain co-occurrence, text-mining or subcellular co-localization. Many *in silico* tools use these data to enrich PPI networks and to predict new interactions (Lees *et al.*, 2011; Liu *et al.*, 2008; Mosca *et al.*, 2013). For instance, high-throughput data in integrative models such as Bayesian Networks improves predictions and blurs the line between *in vitro* and *in silico* (Jansen *et al.*, 2003). Predictions can be improved further by, e.g. developing better statistical models (Jansen *et al.*, 2003; Soong, 2009; Zhang *et al.*, 2012).

Protein sequences improve most integrative models, e.g. through homology-based inference and are highly predictive of PPIs on their own (Martin *et al.*, 2005; Pitre *et al.*, 2012). Sequences are also by far the most abundant data.

Homology-based inference assigns feature F to a protein A if another protein B is experimentally annotated with feature F and sequence-similar to A. This concept works well for Gene Ontology terms (Hamp *et al.*, 2013; Radivojac *et al.*, 2013) and can even outperform advanced predictions of subcellular localization (Goldberg *et al.*, 2014). For PPIs, however, it is substantially more challenging (Mika and Rost, 2006) and many advanced sequence-based PPI prediction methods have been developed.

New prediction method for difficult cases. Park and Marcotte recently introduced three classes of difficulty for predicting whether proteins A and B interact: *C1* if both A and B were in the dataset used to develop the prediction method (but not the PPI A–B itself), *C2* if this was the case for either A or B and *C3* if neither of the two was in the dataset (Park and Marcotte, 2012). Even today's best sequence-based methods perform significantly worse if A and B were not used for method training. Here, we introduce a new method that tackles this problem. It only uses features that are available for all proteins of known sequence and combines empirical rules with advanced machine learning protocols. We show that it slightly outperformed other methods for classes C1 and C2 and that it improved substantially for C3. A filter based on recent tissue-specific gene expression data further increased performance. Finally, by generalizing classes C1–C3, we could identify and predict all difficult query protein pairs in human.

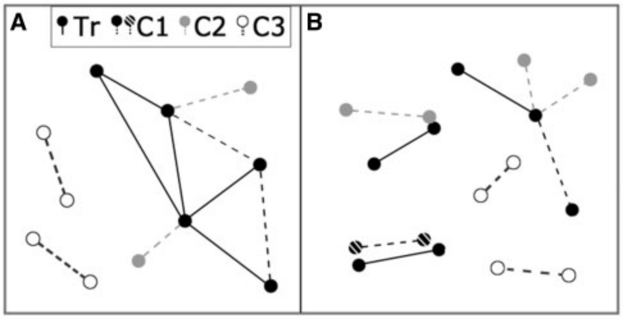

Fig. 1. Comparison of Park and Marcotte and our dataset. Nodes are proteins and edges are PPIs. Proteins closer in space are more sequence-similar. (**A**) Park and Marcotte redundancy reduced the dataset on the level of proteins. Training proteins ('Tr') can have many interaction partners and form networks. Each test interaction A–B (gray and/or dotted nodes/edges) was assigned to one of three classes ('C1–C3') based on whether A and/or B were in the training set. (**B**) In our datasets, there is no sequence similarity between training PPIs ('Tr') because both proteins of a training interaction A–B needed to be dissimilar to all other proteins. We also distinguished between three types of test pairs ('C1–C3'), but our classification was based on sequence similarity to the training proteins, not on whether the exact same protein has already occurred in the training set.

2 Methods

Park and Marcotte datasets. Park and Marcotte used all human and yeast PPIs from PINA v3/2010 (Wu *et al.*, 2009). They redundancy reduced both sets such that no two proteins had >40% pairwise sequence identity. For cross-validation, they divided each set into 10 partitions, using nine to train and one to test. Each test interaction between proteins A and B was assigned to one of the three classes of difficulty (C1–C3: Introduction). Non-interactions ('negatives') were sampled randomly from the respective proteins in each of the four PPI sets (one training, three testing). Datasets and cross-validation splits were publicly available, allowing us to perform exactly the same cross-validations as in Supplementary Table 2 of Park and Marcotte (2012).

New high-quality datasets. The Hippie database collects human PPIs with experimental annotations (Schaefer *et al.*, 2012). Reliability scores grade the interactions by considering, e.g. the number of publications or the type of experimental support. We picked the 10% top-scoring interactions from Hippie v1.2 (10/2011) to obtain a high-quality subset (dubbed *HumanHQ*). We applied the same procedure to Hippie v1.6 (11/2013) and used the difference between both sets of PPIs for testing (*HumanHQ_new*). High-quality yeast PPIs were available from the Database of Interacting Proteins (DIP) core set, which is a subset of the full DIP database and contains only the most reliable physical PPIs (Salwinski *et al.*, 2004). We used DIP v04/2014 (*YeastHQ)*, because slow growth prevented compiling a 'new' dataset for yeast.

Redundancy reduction. We redundancy reduced HumanHQ and YeastHQ such that no two PPIs were sequence-similar. Two PPIs were considered similar if at least one pair of their sequences had HVAL >20 (Rost, 1999; Sander and Schneider, 1991). This corresponds to ~40% pairwise sequence identity for 250 aligned residues. We refer to the non-redundant sets as *HumanHQ_nr* and *YeastHQ_nr*.

Cross-validations and test on new PPIs. Next, we split each non-redundant set into 10 parts, using nine for training and one for testing. By definition, all such test cases were in class C3, because neither protein of a query interaction A–B was sequence-similar to a protein in the training set. We obtained C2 test cases by going through the full redundant high-quality sets (HumanHQ or YeastHQ) and by taking all PPIs A–B for which either A or B was sequence-similar (HVAL >20) to the proteins in the training set. This set of C2 test cases was then internally redundancy reduced as described earlier for the full high-quality sets. Analogously, we created C1 test cases (both proteins A and B of a query A–B were sequence-similar to proteins in the training set). We repeated all this 10 times, so that each of the 10 splits of the full non-redundant set was the test split exactly once. In the end, there were 1825, 2046 and 842 PPIs in C1–C3 in the human cross-validation and 1636, 1663 and 746 PPIs in the yeast cross-validation.

The tests with HumanHQ_new were constructed similarly: as the training set (i.e. the old interactions), we used the full HumanHQ_nr. To obtain new C3 test PPIs, we first collected all interactions A–B in HumanHQ_new for which neither A nor B was sequence-similar to a protein in HumanHQ_nr. Then we redundancy reduced this new set internally exactly as HumanHQ. C2 and C1 were created in the same way, except that one (C2) or both (C1) proteins of an interaction in HumanHQ_new had to be sequence-similar to interactions in HumanHQ_nr. This resulted in 392, 580 and 218 new human test PPIs in classes C1–C3.

Negative interactions. Park and Marcotte made it clear that we need to distinguish between classes C1–C3. It is less clear, however, how to sample negatives for these classes. Here, we propose a solution that is especially suited for the prediction of the 'full interactome', arguably the most common and challenging application.

Our definition of classes C1–C3 only required two proteins to be similar to be in class C1 or C2, not identical as for Park and Marcotte (Fig. 1). Hence, given a positive training set (e.g. a training split in the cross-validation with HumanHQ), we could assign each possible protein pair in an organism to one of the classes C1–C3. This allowed us to measure how well a method separated interacting (positives) from non-interacting (negatives) pairs in each class: the positives were already given because each positive training set was associated with three test sets (C1–C3). We obtained the negatives by randomly sampling from all the pairs in the respective class, e.g. from all the C3 pairs in human (10 negatives for each positive PPI). Prediction performance for all C3 pairs was then estimated by measuring how the positive C3 test PPIs ranked among the negative C3 test PPIs, e.g. as a recall-precision curve.

Most PPI prediction methods also require training on negatives (non-interacting proteins). In supervised learning, training and test data should be sampled from the same population, which in our case is the same set of protein pairs. Hence, we sampled negatives for training in the same way as for testing, i.e. from all pairs in one of the classes C1–C3 (again: 10 negatives for each positive training PPI). Note that this still did not allow predicting PPIs simply by similarity between single proteins: in case of C3, e.g., both negative and positive test PPIs were sequence-similar to negative training PPIs, but sequence-dissimilar to positive training PPIs.

All proteins of each organism were provided by the EMBL-EBI Reference Proteomes (Dessimoz *et al.*, 2012). We removed short (<50 residues) and long (>5000) proteins. Additionally, we ascertained that no negative was listed as positive in the full Hippie 1.2 database (i.e. as a PPI with experimental evidence that we deemed insufficient for training).

Performance measures. All methods tested here could be re-trained with custom PPIs and provided a score for each prediction. Hence, we could calculate standard recall-precision curves. To minimize sampling noise, we followed a standard procedure and repeated each experiment 10 times from the start (Witten and Frank, 2005). Thus, we performed 10 times 10-fold cross-validations with HumanHQ and YeastHQ (including re-sampling of negatives) and also repeated the tests on new human PPIs 10 times, each time with a new sample of negatives. In the end, we always averaged over the 10 curves.

New method: profile interaction kernel. Our new method uses support vector machines (SVMs) which calculate hyperplanes that optimally separate data points (high dimensional numerical vectors) of one class from those of another class (Schölkopf and Smola, 2001). The evolutionary profile kernel designed by the Leslie group defines one such feature vector for each protein (Kuang *et al.*, 2005). Each element represents a sequence of k residues (k-mer). If $k=3$, e.g. the feature vector has $20^3=8000$ elements (all possible 3-mers taken from 20 amino acids). The value of an element is the number of times this k-mer is conserved in the evolutionary profile of the protein, i.e. how often the sum of amino acid substitution scores is below a user-defined threshold σ. Let us illustrate this for the 3-mer WTG. To test whether it is conserved at residue 37, we look up the frequency of W at residue 37 and convert it to a score by taking the negative logarithm. Then we do the same for T and G at positions 38 and 39, sum up the three scores and check whether the sum is smaller than σ. [Note that there can be more than one conserved k-mer per position: e.g. with $\sigma=\infty$, there are 20^k conserved k-mers for every position in a sequence of length n, and hence, $n \times 20^k$ in total]. The profile kernel is then defined as the dot product of two such vectors of k-mer counts. In the actual implementation, an efficient k-mer trie-based algorithm takes all evolutionary profiles at once and calculates all dot products in one traversal of the trie. Recently, we further accelerated this algorithm and made it easier to use (Hamp *et al.*, 2013).

In our new PPI feature space, each dimension represents a pair of k-mers. Continuing the example earlier: in addition to WTG in protein A, we now also look for conserved k-mer LGA in protein B and count how often WTG and LGA co-occurred in the interaction. This new feature space has $20^k \times 20^k$ dimensions. However, the dot product between two such feature vectors (i.e. two PPIs) only requires dot products in the 20^k-dimensional single-protein feature space (Supplementary S1) (Martin *et al.*, 2005). More formally: $K(I_{A-B},\ I_{A'-B'}) = \widehat{K}(A,A')\widehat{K}(B,B') + \widehat{K}(A,B')\widehat{K}(A',B)$, where K is the profile kernel in interaction space, I_{A-B} and $I_{A'-B'}$ are the PPIs A–B and A′-B′, respectively, and $\widehat{K}$ is the dot product in single-protein space, i.e. the original profile kernel.

Other sequence-based methods. We compared our method to *PIPE2* (Pitre *et al.*, 2012), *SigProd* (Martin *et al.*, 2005) and *AutoCorrelation* (Guo *et al.*, 2008), which performed well compared with other sequence-based PPI prediction methods in a recent assessment (Park and Marcotte, 2012). All implementations were freely available and provided the functionality to re-train the method with custom PPIs.

PIPE2. For each query PPI A–B, PIPE2 counts how often two particular 20-mers from A and B co-occur in training interactions. The result is stored in a matrix with all 20-mers of A as rows and all 20-mers of B as columns. Each cell gives the number of PPIs this pair of 20-mers has been observed in. The matching of 20-mers is inexact, i.e. not exactly the same 20-mer has to be found in a training protein A′ (B′) to be considered a hit for A (B). A PAM120 score above a certain threshold suffices. The matrix is smoothed by a sliding 3 × 3 window (central value replaced by the median over the window). A and B are predicted to interact if the average score in a 3 × 3 window of the new matrix exceeds a certain threshold.

SigProd. Proteins are encoded as vectors of ‘signatures’. A signature is a pair of 3-mers that have the same amino acid at position 2 and the same set of amino acids at positions 1 and 3 (e.g. GTW = WTG). Each element is the number of times this signature has been observed in the protein. A PPI is represented as a vector of signature co-occurrences, as explained for the profile-kernel. Thus, our approach can be seen as an enhancement of the signature PPI kernel by using protein profiles instead of sequences, extending the single-protein feature space (∼4000 signatures versus 3.2M k-mers with $k=5$) and introducing and optimizing critical parameters (k, σ and C).

AutoCorrelation. A sliding window extracts all pairs of residues in a protein that are x residues apart in sequence [e.g. residues (1,4), (2,5), … for $x=2$]. An amino acid index (e.g. polarity) encodes these pairs as 2D numerical vectors. A feature is defined as the correlation coefficient between the first and second element of these vectors. Doing this for all $x \in [0,1, \dots ,29]$ and seven different amino acid indices creates a total of $30 \times 7 = 210$ features. A PPI is encoded by concatenating the features of the two proteins (420 features). As proposed in (Park and Marcotte, 2012), we used random forests to make PPI predictions.

Optimization of free parameters. For our new method, we optimized the k-mer length (k) and the substitution score threshold (σ) empirically with a grid search on one split (90%) of the full HumanHQ_nr cross-validation (all test cases in C3). Possible values for k and σ were [4,5,6] and [4,5, … ,11]. This procedure should have prevented ‘leaks’ of test data into the training phase (Supplementary S2). To keep training times low, we re-used the best combination found ($k=5$, $\sigma=9$) in all other experiments (C1–C2 HumanHQ; C1–C3 YeastHQ, Park&Marcotte and HumanHQ_new). Leaks could be ruled out in these cases. In fact, here we might underestimate performance due to potentially suboptimal k and σ. For classification, we always used the Sequential Minimal Optimization (Platt, 1999) implementation in Weka (Hall *et al.*, 2009). This platform also allowed optimizing the complexity parameter C for every

SVM that we trained (internal 10-fold cross-validation; values for C were 10−2, −1, . . . , 2). In all optimizations, we chose the parameter combination that achieved the highest average precision up to a recall of 25% (step size of 5%). Evolutionary profiles were taken from PredictProtein (Yachdav *et al.*, 2014).

As mentioned earlier, all methods compared provided the functionality to re-train on custom PPIs. The developers of those methods had only defined the template, i.e. which parameters to optimize how during training. We did not change any method in this respect. Instead, we re-trained each method for every training-test set combination such that all methods had always used the same training set when predicting a test set. As we had optimized the parameters of our method only with older human C3 PPIs, all methods were equally 'blind' for all other test sets.

Table 1. Best AUCs for Park and Marcotte datasets

	Human			Yeast		
Method	C1	C2	C3	C1	C2	C3
PPI-PK	**87 ± 1**	**69 ± 1**	**67 ± 2**	87 ± 2	69 ± 2	**68 ± 2**
Old best	85 ± 1	64 ± 1	59 ± 2	85 ± 1	67 ± 1	59 ± 2

We evaluated our new method (PPI-PK) as in (Park and Marcotte, 2012). 'Old best' marks the best AUC achieved by previously published methods. C1: both proteins A and B of a query PPI A–B used to train, C2: only one used to train, C3: none used to train.

Expression data. We used recently published human gene expression data to filter protein pairs that are unlikely to interact because they are expressed in different tissues (Fagerberg *et al.*, 2014). The minimum expression level (measured in fragments per kilobase of transcript per million mapped reads) was set to 10, corresponding to 'high expression' according to (Fagerberg *et al.*, 2014). 6656 of the 20 249 proteins in the EBI reference proteome could not be mapped to tissues, either because they were not measured (1276) or because no expression level was high enough (5380). The filter worked by removing each protein pair A–B, if any of the two proteins could not be mapped to tissues or if there was no tissue in which both proteins A and B were expressed. A comparison of the filtered and unfiltered pairs in terms of house-keeping genes and over- and underrepresented protein functions revealed significant, but overall rather small changes (Supplementary S4.1). The key factors that determined whether a method could benefit from the filtering are discussed in Supplementary S4.3.

3 Results and discussion

New method improves sequence-based PPI prediction. Our new approach for predicting PPIs from sequence alone is based on

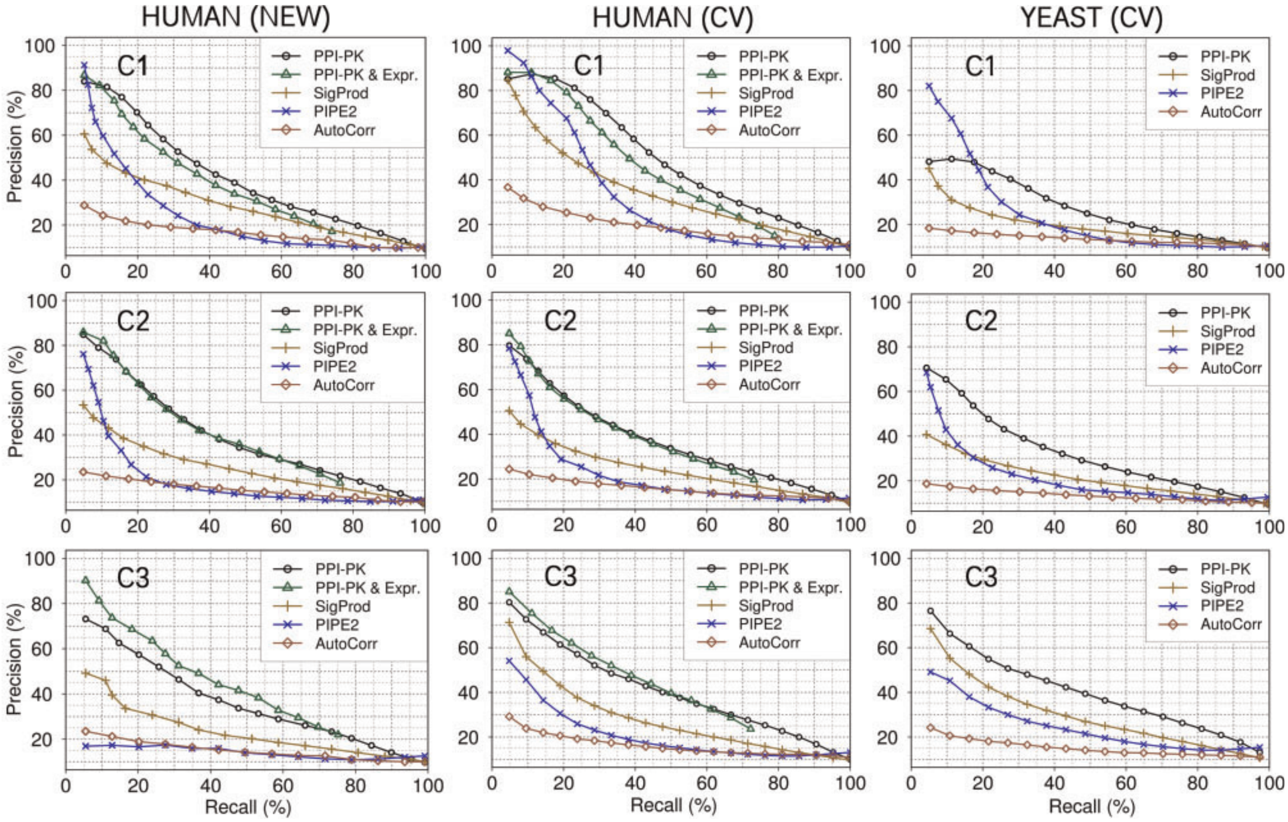

Fig. 2. Better predictions for high-quality PPIs. Our new method (PPI-PK) compared favorably to the three state-of-the-art sequence-based methods SigProd, PIPE2 and AutoCorrelation. All lines show standard recall-precision curves. Columns describe different data sets: leftmost [Human (new)]: test with new human interactions added after those used for development (Methods: *Human_new*); middle [Human (cross-validation)] and rightmost [Yeast (cross-validation)]: tests with the cross-validation sets (Methods: *HumanHQ_nr* and *YeastHQ_nr*). For each query interaction between proteins A and B, we distinguished three difficulty classes: C1 if both A and B (or homologs) were used for training (but not the interaction A–B); C2 if either A or B (or homologs) were used to train; C3 if neither of the two nor their homologs were used to train. 'PPI-PK & Expr' refers to results obtained after applying our baseline gene expression filter to the predictions by PPI-PK. Negatives were always ten times more frequent than positives.

evolutionary profiles and the profile-based kernel designed by the Leslie group (Kuang *et al.*, 2005). We compared it to state-of-the-art sequence-based methods using experimental PPIs from human and yeast. Only few known proteins have experimentally annotated PPIs and even fewer have reliable annotations. Hence, most pairs fall into classes C2 and C3. Prediction methods, however, were mainly tested on class C1 due to a flaw in the traditional cross-validation scheme. This was recently discovered with a refined cross-validation procedure (Park and Marcotte, 2012).

Better for Park and Marcotte data. Using exactly the same procedure and data introduced by Park and Marcotte, our new method improved over the state-of-the-art for yeast and human for all classes of difficulty (Table 1). The improvement was highest for interactions between proteins that were sequence-dissimilar to proteins in the training set (C3). For instance, our new method pushed the C3 area under the ROC curve (AUC) mark for human to 67 ± 2% compared with the previous best of 59 ± 2% (random = 50%), i.e. almost doubling the 'distance to random'. Confirming Park and Marcotte, our method also performed best if both proteins of a query interaction were used to train (but not the interaction itself).

Evaluation with most reliable PPIs. In another test, we only used highly reliable PPIs from Hippie and DIP and rigorously reduced redundancy (Methods; discussion and comparison to redundant datasets in Supplementary S3). Generalized classes C1–C3 only needed proteins to be similar, not identical, to put the corresponding PPIs into class C1 or C2 (Methods). As before, we compared our method to SigProd (Martin *et al.*, 2005), PIPE2 (Pitre *et al.*, 2012) and AutoCorrelation (Guo *et al.*, 2008), previously established as top sequence-based methods (Park and Marcotte, 2012).

Our new method compared favorably to others throughout the recall-precision curves (Fig. 2) except for very low recall in C1–C2 (PIPE2 better in C1 and almost on par in C2). PIPE2 seemed to perform quite well for a few C1 cases. SigProd (for human) and our new method (for human and yeast) overtook for higher recall. This realm of high recall is important for users who try to get many interactions from as few wet-lab experiments as possible. For C3, our method consistently improved greatly over both PIPE2 and SigProd. AutoCorrelation was not competitive although it slightly improved in C1–C2 with more redundancy in the training sets (Supplementary S3). Filtering out proteins not highly expressed in at least one common tissue increased precision for human C3 interactions, in particular for the test with new human interactions (HumanHQ_new). For the other classes, we observed minor improvements exclusively for low recall levels. We observed a similar trend when applying the filter to the other methods (Supplementary S4.2); possible reasons for this are discussed in Supplementary S4.3.

We repeated our analysis (shown in Fig. 2) with the least reliable PPIs added between 2011 and 2013 (Supplementary S5). All methods except AutoCorrelation performed substantially worse. This result justified our initial selection of only the most reliable PPIs.

Full human interactome prediction. The interaction partners of many human proteins remain unknown. Several *in silico* methods help annotating experimental data and designing new experiments. As the method introduced here performed better than those previously established as the state-of-the-art for sequence based PPI prediction, we made the best 200 000 of all human C2 and C3 predictions available online. First analyses indicate a large diversity of proteins and similar degree distributions as in the full Hippie database, except that large hubs (>50 interactions) are slightly more frequent among our predictions (Supplementary S6).

4 Conclusions

We introduced a new method that predicts physical protein–protein interactions from sequence alone by using evolutionary profiles through profile-kernel SVMs. It is optimized to predict pairs of proteins that come into close contact at some point in time, *not* to predict functional associations. In our hands, this method improved over the state-of-the-art in methods that exclusively rely on sequence information. The improvements were most substantially for proteins without significant sequence similarity to proteins with reliable experimental annotations. These by far outnumber the set of 'well characterized' proteins even for the best-studied model organisms. A simple filter removing protein pairs not expressed in the same tissue further improved performance. We provide downloadable implementations to re-train our method with custom PPIs, together with a list of the 200 000 most reliably predicted human proteins pairs that are sequence-dissimilar to known interactions.

Acknowledgements

Thanks to Yung Ki Park (HJKRI) and Shawn Martin (Sandia) for method implementations and very helpful hints; thanks to Tim Karl (TUM) for invaluable help with hardware and software; to Marlena Drabik (TUM) for administrative support. Thanks to the anonymous reviewers for the detailed help to improve work and presentation! Thanks also to Miguel Andrade (MDC Berlin), David Eisenberg (UCLA) and the whole Hippie and DIP teams for providing such excellent resources. Last, not least, thanks to Rolf Apweiler (UniProt, EBI, Hinxton), Amos Bairoch (CALIPHO, SIB, Geneva), Ioannis Xenarios (Swiss-Prot, SIB, Geneva) and their crews for maintaining excellent databases and to all experimentalists who enabled this analysis by publishing their data. This work was supported by a grant from the Alexander von Humboldt foundation through the German Ministry for Research and Education (BMBF: Bundesministerium fuer Bildung und Forschung).

Conflict of Interest: none declared

References

Dessimoz,C. *et al.* (2012) Toward community standards in the quest for orthologs. *Bioinformatics*, **28**, 900–904.

Fagerberg,L. *et al.* (2014) Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics. *Mol. Cell. Proteomics*, **13**, 397–406.

Goldberg,T. *et al.* (2014) LocTree3 prediction of localization. *Nucleic Acids Res.*, **42**, W350–W355.

Guo,Y. *et al.* (2008) Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Res.*, **36**, 3025–3030.

Hall,M. *et al.* (2009) The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, **11**, 10–18.

Hamp,T. *et al.* (2013) Homology-based inference sets the bar high for protein function prediction. *BMC Bioinformatics*, **14** (Suppl. 3), S7.

Jansen,R. *et al.* (2003) A bayesian networks approach for predicting protein–protein interactions from genomic data. *Science*, **302**, 449–453.

Kuang,R. *et al.* (2005) Profile-based string kernels for remote homology detection and motif extraction. *Journal of bioinformatics and computational biology*, **3**, 527–550.

Lees,J.G. *et al.* (2011) Systematic computational prediction of protein interaction networks. *Phys. Biol.*, **8**, 035008.

Liu,Y. *et al.* (2008) Protein interaction predictions from diverse sources. *Drug Discov. Today*, **13**, 409–416.

Martin,S. *et al.* (2005) Predicting protein–protein interactions using signature products. *Bioinformatics*, **21**, 218–226.

Mika,S. and Rost,B. (2006) Protein–protein interactions more conserved within species than across species. *PLoS Comput. Biol.*, **2**, e79.

Mosca,R. *et al.* (2013) Towards a detailed atlas of protein–protein interactions. *Curr. Opin. Struct. Biol.*, **23**, 929–940.

Park,Y. and Marcotte,E.M. (2012) Flaws in evaluation schemes for pair-input computational predictions. *Nat. Methods*, **9**, 1134–1136.

Pitre,S. *et al.* (2012) Short co-occurring polypeptide regions can predict global protein interaction maps. *Sci. Rep.*, **2**, 239.

Platt,J.C. (1999) Fast training of support vector machines using sequential minimal optimization. In, *Advances in kernel methods*. MIT Press, pp. 185–208.

Radivojac,P. *et al.* (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.

Rost,B. (1999) Twilight zone of protein sequence alignments. *Prot. Eng.*, **12**, 85–94.

Salwinski,L. *et al.* (2004) The database of interacting proteins: 2004 update. *Nucleic Acids Res.*, **32**, D449–D451.

Sander,C. and Schneider,R. (1991) Database of homology-derived structures and the structural meaning of sequence alignment. *Proteins*, **9**, 56–68.

Schaefer,M.H. *et al.* (2012) HIPPIE: integrating protein interaction networks with experiment based quality scores. *PLoS One*, **7**, e31826.

Schölkopf,B. and Smola,A.J. (2001) *Learning with Kernels Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA.

Soong,T.-T. (2009) Computational prediction of physical protein-protein interactions with novel microarray analysis and efficient data integration. *The Center for Computational Biology and Bioinformatics (C2B2)*, Columbia University, New York.

Witten,I.H. and Frank,E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. *(Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc, Burlington, Massachusetts.

Wu,J. *et al.* (2009) Integrated network analysis platform for protein–protein interactions. *Nat. Methods*, **6**, 75–77.

Yachdav,G. *et al.* (2014) PredictProtein—an open resource for online prediction of protein structural and functional features. *Nucleic Acids Res.*, **42**, W337–W343.

Zhang,Q.C. *et al.* (2012) Structure-based prediction of protein–protein interactions on a genome-wide scale. *Nature*, **490**, 556–560.