# Supplement 1
# Methods Supplement – Datasets and Computational Methods

## 1. Gold standard datasets for classification

Our computational method uses a supervised learning framework to predict the receptor interactome, and therefore requires a training or reference set (gold standard set).

A small number of interacting protein pairs have been experimentally validated in small scale experimental studies. This set served as a positive set for our learning task. The Human Protein Reference Database (HPRD [1,2]) contains 14608 pair-wise protein-protein interactions (excluding self-interactions). These pairs were retrieved through experts' critical reading of published literature. Among these, 2522 interactions contain at least one receptor protein. The list of receptors was retrieved from the HPMR [3]. We use these 2522 interactions as our gold standard positive set. Table S1.1 lists the number of proteins and number of pairs in each of these two sets.

**Table S1.1 Gold standard positive set.** The second column lists the number of known protein and protein pairs in which at least one of the two proteins is receptor [3]. The third column lists the total number of proteins and protein pairs in HPRD [1,2].

| | **Receptor Related (Gold Standard Positive)** | **HPRD Full (Physical Interactions)** |
|---|---|---|
| Number of Protein Pairs | 2522 | 14608 |
| Number of Proteins | 1455 | 5712 |

Unlike positive interactions, it is essentially not possible to rule out an interaction entirely. Considering the small fraction of interacting pairs in the total set of potential protein pairs, we use a random set of protein pairs excluding those known interacting pairs as the negative set instead. For the receptor related task, all protein pairs in the random negative set contain at least one receptor protein. Based on the histogram distribution of the number of interacting partners each receptor has in HPRD (data not shown), we estimated that roughly only 1 in ~1000 possible protein pairs is actually interacting. Thus, over 99.8% of our random data is indeed non-interacting, which is probably better than the accuracy of most training negative datasets.

Combining the positive and negative pair sets, a reference set (also called gold standard set) is constructed and used to train/test our learning methods.

## 2. Biological data sources for human protein interactions predictions

In previous evidence integration efforts in yeast and human, data related directly and indirectly to protein-protein interactions (PPI) were combined. Due to the low coverage of membrane receptors in the currently available high throughput human interaction datasets we could not use direct data as features. However, there are rich sources of biological data that might be indirectly related to membrane receptor PPIs. We collected feature attributes from eight different feature categories. Each of the collected data sets has its own representative form. For example, protein sequence is encoded in the form of a character string, which means the order of amino acids as they occur in a polypeptide chain. Gene expression data is usually a vector of expression values across multiple time points for a specific gene. To combine different forms of information, for each data set we determined a natural way to calculate the similarity between two proteins with respect to the evidence. Concatenating all these similarity values together resulted in a feature vector describing a receptor-protein pair. Biological insight was used to optimize the feature.

In the final optimized feature set, the data was encoded as follows:

- (a). Features 1-3: GO ontology. Three 'similarity' measures were derived from Gene Ontology (GO) [55], according to the proteins' positions in the three ontology hierarchies: biological process, molecular function and cellular component. For each candidate protein pair the feature describes how many times both proteins are in the same functional class of the GO slim level [55]. GO slims contain a subset of the terms in the full GO. They give a broad overview of the ontology content without the detail of the specific fine grained terms. For instance, there are 32 classes of cellular component in GO slim. If a candidate pair shares 2 of the 32 classes this pair is assigned the value 2 as its GO component feature.
- (b). Feature 4: Tissue distribution. To describe whether two proteins appear in the same human tissues or not, we counted the number of tissues in which both are expressed and used this number as the feature.
- (c). Features 5-21: Gene co-expression. Features were derived from sixteen expression sets (details in Supplement S1) downloaded from the NCBI Gene Expression Omnibus [56] database. Pearson’s correlation between two genes’ expression values (normalized first) are calculated and used as features.
- (d). Feature 22: Sequence. Protein sequence alignment score was used as another similarity feature. We used NCBI’s PSI-BLAST [57] to align the two sequences of each pair. All BLASTP hits with E-value less than or equal to 0.001 are used. The actual E-value was used as the feature.
- (e). Feature 23-26: Homologous interactions in yeast. Homologous PPIs were derived based on if a candidate pair’s homologous proteins bind each other in another species or not. The homology between human proteins and yeast proteins is based on the sequence alignment scores from PSI-BLAST [57]. The yeast PPI data sets used include interactions from the DIP database and four other predicted PPI data sets, including computationally predicted co-complex pairs and physical binding pairs, which are either predicted by SVM or RF classifiers [39].
- (f). Feature 27: Domain-domain interactions. These features were derived based on the hypergeometric distribution of domain-domain co-occurrences in receptor-protein pairs. The domain composition evidence of each human protein was downloaded from the HPRD [4]. For every interacting protein pair, each domain from protein A was connected to the domains in protein B. The frequency of these

domain pairs was determined for all interacting protein pairs, as well as all non-interacting pairs. The hypergeometric distribution was then used to determine which domain pairs are enriched in interacting protein pairs (HPRD pairs that are not in our gold standard positive set) compared to the non-interacting pairs (random pairs not in HPRD and also not in our gold-standard negative set). For a new candidate protein pair we used the smallest p-value from their related domain-domain pairs as features, the smaller the value the more significant the domain pair.

**Table S1.2 Feature set derived for pairwise protein-protein interaction prediction in human.** We collected a total of 27 features from 8 different data sources. The second column lists the name of the feature source. The third column lists the number of attributes from each source. The fourth column describes the value property. The fifth column presents the average percentage of pairs for which information is available using this feature source. The last column gives the references of each data source.

| Source Index | Feature Name | Num of Feature | Feature Property | Average Coverage | Ref-erence |
|---|---|---|---|---|---|
| 1 | GO Function | 1 | Non-negative Integer | 0.3908 | [8] |
| 2 | GO Component | 1 | Non-negative Integer | 0.3627 | [8] |
| 3 | GO Process | 1 | Non-negative Integer | 0.3756 | [8] |
| 4 | Co-Tissue | 1 | Non-negative Real | 0.5712 | [6] |
| 5 | Co-Gene Expression | 16 | Real between (-1, 1) | 0.3401 | [5] |
| 6 | BlastP E-value | 1 | Non-negative Real | 1 | [9] |
| 7 | Homologous protein interactions from Yeast | 5 | Non-negative Real | 1 | [9,10,11] |
| 8 | Domain-domain Interaction | 1 | Non-negative Real | 0.3769 | [12, 1,2] |

Most biological datasets are noisy and contain many missing values (Table S1.2). The coverage of the 8 groups ranges from 34% for gene expression to 57% for tissue feature and reaches 100% only for sequence based features. Concatenating all these features together gives us the feature vector describing a protein-protein pair.

**Table S1.3 Summary of the sixteen gene expression data sets we used.** All were retrieved from the GEO [5] database.

| **Sample Size** | **Set Summary** | **GDS No.** |
|---|---|---|
| 120 | Acute lymphoblastic leukemia treatment responses | GDS330 |
| 66 | B-cells and acute renal allograft rejection | GDS365 |
| 173 | Multiple myeloma and bone lesions | GDS531 |
| 75 | Smoking-induced changes in airway transcriptome | GDS534 |
| 158 | Large-scale analysis of the human transcriptome (HG-U133A) | GDS596 |
| 42 | T lymphocyte activation gene identification | GDS601 |
| 91 | Lung neuroendocrine tumor classification | GDS619 |
| 37 | Heart failure arising from different etiologies | GDS651 |
| 87 | Acute myeloid leukemia cell differentiation induced by various drugs | GDS715 |
| 60 | Estrogen positive breast cancer recurrence during tamoxifen therapy: whole tissue tumor | GDS806 |
| 60 | Estrogen positive breast cancer recurrence during tamoxifen therapy: microdissected tumor | GDS807 |
| 44 | Adult acute myeloid leukemia: bone marrow and peripheral blood expression profiles (SHCZ) | GDS842 |
| 49 | Adult acute myeloid leukemia: bone marrow and peripheral blood expression profiles (SHDJ) | GDS843 |
| 41 | Kidney transplant response to calcineurin inhibitor-free immunosuppression using sirolimus | GDS987 |
| 35 | Normal tissues of diverse types (SHBW) | GDS1085 |
| 38 | Normal tissues of diverse types (SHCN) | GDS1086 |

## 3. The random forest classifier

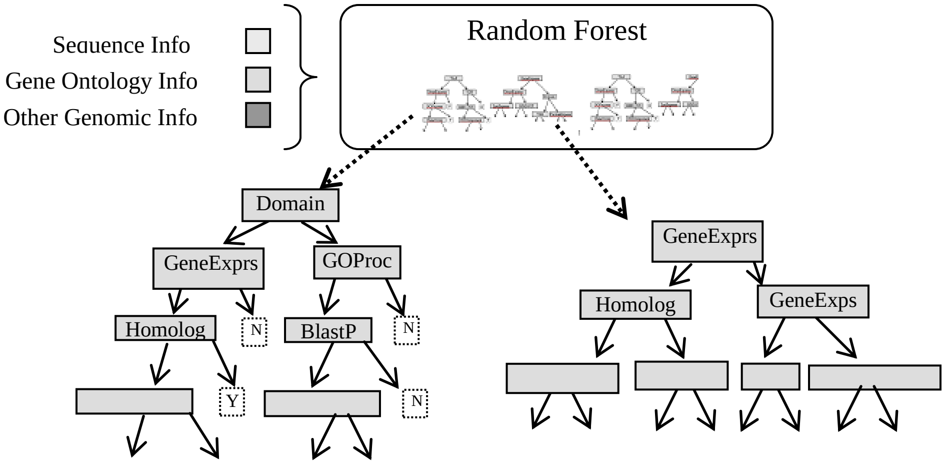

**Figure S1.1 Random forest classifier for PPI prediction.** To generate the random forest, we select for each tree a bootstrap sample of the training data. Next, for every node in these trees a random subset of the attributes is chosen and the attribute achieving the best division is selected. Once model trees are grown, protein pairs are propagated down and the 'votes' from all trees are used to compute interaction scores. [13]

The Random Forest (RF) [38] consists of a collection of independent decision trees. Decision trees are grown using a training set. At each node the algorithm searches for an attribute that best separates all instances in that node. If the attribute perfectly classifies all instances so that all instances in one of the two descendent nodes have the same label then this node becomes a terminal node with the appropriate label. Otherwise, the above process is repeated until all instances are at terminal nodes.

In the RF, each tree is grown on a bootstrap sample of the training set. For each node in the tree the split is chosen from a fixed number of features that are selected at random out of the total attributes. RF performs better than a single decision tree because RF can utilize randomization and redundant features. This is important if a pair has values for one redundant feature but not the other (many biological datasets are expected to be correlated and have noise and missing values). RF classifier used in this work was implemented by modifying the Berkeley Random Forest package [38]. Two hundred trees were grown for training. For the number of variables randomly selected at each node we used the default value that was equal to the square root of the feature dimension.

From various biological data sources, we construct an $M$-dimensional input feature vector $X$ for every pair of proteins. Given these vectors, the task of receptor interactome prediction can be presented as a binary classification problem. That is, given $X$ does this pair interact ($Y=1$) or not ($Y= -1$).

**Decision tree:** A decision tree is a binary tree with nodes corresponding to attributes in the input vectors. Tree nodes are used to determine how to propagate a given attribute set down the tree. Nodes can either be threshold nodes or categorical nodes. Decision trees also contain terminal (or leaf) nodes that are labeled as *-1* or *1*. In order to classify a protein pair as interacting or not, this pair is propagated down the tree and decision is made based on the terminal node that is reached. Decision trees are grown using a training set. At each node the algorithm searches for an attribute that best separates all instances in that node. If the attribute perfectly classifies all instances so that all instances in one of the two descendent nodes have the same label then this node becomes a terminal node with the appropriate label. Otherwise, the above process is repeated until all instances are at terminal nodes.

**Random forest:** The Random Forest (RF [13]) classifier is one of the most effective and widely used machine learning techniques. RF uses a collection of independent decision trees instead of one tree, where each tree grown on a bootstrap sample of the training set (this helps in avoiding overfitting). A number $m << M$ ($M$ is the total number of attributes) is specified, and for each node in the tree, the split is chosen from $m$ variables that are selected at random out of the total $M$ attributes. To classify a new example, put its feature vector down each of the trees in the forest. Each tree gives a classification, and the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). RF used in this paper was implemented by using the Berkeley Random Forest package [13]. One of the main reasons random forests perform better than a single decision tree is their ability to utilize redundant features and the independence of the different classifiers (trees) used.

**Feature importance estimation using Gini criterion:** The RF classifier uses a splitting function called the *Gini* index to determine which attribute to split on during the tree learning phase. The *Gini* index measures the level of impurity / inequality of the samples assigned to a node based on a split at its parent. Let $p$ represent the fraction of interacting pairs assigned to node $m$ and $1-p$ the fraction of the non interacting pairs. Then, the Gini index at node $m$ is defined as: $G_m = 2p(1-p)$. The purer a node is, the smaller the Gini value. Every time a split of a node is made using a certain feature attribute, the Gini value for the two descendant nodes is less than the parent node. The decrease in the sum of these Gini values (from parent to children) for each feature over all trees in the forest provides a simple and reliable estimate of the feature importance for this prediction task.

## 4. Performance evaluation

**Comparison to other classifiers:** We compared the RF classifier with three other popular classifiers: Naïve Bayes (NB), Logistic Regression (LR) and Support Vector Machine (SVM). (1). Naïve Bayes (NB) is a probabilistic classifier that uses the joint probabilities of features and categories to estimate the conditional probabilities of categories given feature evidence. The NB classifier was obtained from the WEKA machine learning [15] tool box using supervised discretization to process numeric attributes. (2). Logistic Regression (LR) is a generalized linear statistical model that can predict a discrete outcome from a set of variables that may be continuous, discrete, dichotomous, or a mixture of these types. The LR classifier was also obtained from the WEKA tool box and it uses a ridge estimator for building a multinomial LR model. (3). Support Vector Machines (SVM) is a popular learning approach for solving two-class pattern recognition problems. It is based on the structure risk minimization principle for which error-bound analysis has been theoretically motivated. We used the SVMLight tool box with linear kernel [12], an implementation of SVMs in C. (4). For RF, we grew 200 trees during each training procedure and used the square root of the feature dimension (default) as the number of attributes (*m*) to select from for each node.

**Training/testing procedures:** Performance comparisons were based on the following training and testing procedures. Parameter optimization was carried out in all cases using separate training and validation datasets. We randomly sampled a training set containing 80,000 protein pairs to learn the prediction model. Then we sampled a test set (another 80,000 pairs) from the remaining protein pairs, and used the trained model to evaluate the performance of the classifier. The above steps were repeated 12 times for each classifier and average values are reported. Based on the estimated ratio (1:1000 true to negative interactions) we have ~80 positive PPIs in each test set. For the training set, we down-sampled [18,19] the negative examples in a pre-processing step. We tested different ratios for training the classifiers. Regardless of the ratio of the training data, the ratio of the test data was always fixed at 1 to 1000. The best ratio for training turned out to be 1 to 100, which resulted in roughly ~800 positive examples in each training run. The down-sampling strategy addresses the problem of too few positive examples in the training set.

**Evaluation measures:** We used three well established measures to evaluate prediction performance: Prediction accuracy versus Sensitivity (also called Precision vs. Recall) curves and full or partial areas under Receiver Operator Characteristic (AUC) curves [16,17].

- Prediction accuracy vs. Sensitivity curve – This curve is also called Precision vs. Recall curve in information retrieval [16]. Prediction accuracy (Precision) refers to the fraction of interacting pairs predicted by the classifier that are truly interacting. Sensitivity (Recall) measures how many of the known pairs of interacting proteins have been identified by the learning model. The Prediction accuracy vs. Sensitivity (Precision vs. Recall curve) is then plotted for different cutoffs on the predicted score.
- AUC scores - Receiver Operator Characteristic (ROC) [17] curves plot the true positive rate against the false positive rate for different cut-off values of the predicted score. ROC curves therefore measure the trade-off between sensitivity and specificity. The area under the ROC curve (AUC) is commonly used as a summary measure of diagnostic accuracy. It can take values from 0.0 to 1.0. AUC values are interpreted as the probability that a randomly selected "event" will be regarded with greater suspicion (in terms of its continuous measurement) than a randomly selected "non-event". In some cases, rather than looking at the area under the entire ROC curve, it is more informative to only consider the area under a portion of the curve.
- Partial AUC scores - In PPI prediction, we are interested in performance of our models under conditions where the false positive rate is very low. Other false positive rates, even those that seem low such as FP = 0.1 are not meaningful for the task we

consider. For such a FP rate, a testing set of size 80,000 will yield roughly 8000 negative misclassified samples. Even if the true positive examples (about 80 examples) are all correctly classified (which is often impossible), the precision of this prediction is just 0.01. AUC *n* (where *n* is an integer) reports the percentage of recovered interactions up to n false positives.

**Investigating of different gold standard positive or different gold standard negative:** Please see Supplementary S2 for our comparisons.

## Bibliography

1. Peri S, Navarro JD, Amanchy R, Kristiansen TZ, et al. and Pandey A. Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Research. (2003). 13:2363-2371.
2. Mishra GR, Suresh M, Kumaran K, Kannabiran N, et al. and Pandey A. Human protein reference database--2006 update. Nucleic Acids Res 2006 Jan 1; 34(Database issue) D411-4.
3. Ben-Shlomo I, Yu Hsu S, Rauch R, Kowalski HW, Hsueh AJ., Signaling receptome: a genomic and evolutionary perspective of plasma membrane receptors involved in signal transduction. Sci STKE. 2003. 187: RE9
4. NCBI Taxonomy http://www.ncbi.nlm.nih.gov/Taxonomy (2005)
5. NCBI Gene Expression Omnibus (GEO) ftp://ftp.ncbi.nih.gov/pub/geo/data/gds/ (2005)
6. Joan U. Pontius and Lukas Wagner and and Gregory D. Schuler, UniGene: A Unified View of the Transcriptome. ftp://ftp.ncbi.nih.gov/repository/UniGene/Homo_sapiens/ (2006).
7. GDB: http://www.gdb.org/gdbreports/GeneticDiseases.html Genetic Disorders by Chromosome. (2006)
8. The Gene Ontology Consortium, Gene Ontology: tool for the unification of biology. Nature Genet. 2000; 25 25-29.
9. NCBI BLAST. http://www.ncbi.nlm.nih.gov/BLAST. (2005)
10. Xenarios I., Salwinski L., Duan XJ., Eisenberg D., et al. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Res. 2002: 30, (1):303-5.
11. Qi Y., Bar-Joseph Z., Klein-Seetharaman J., "Evaluation of different biological data and computational classification methods for use in protein interaction prediction", PROTEINS: Structure, Function, and Bioinformatics. 2006 May 15; 63 (3):490-500.
12. GeneMerge--post-genomic analysis, data mining, and hypothesis testing. Bioinformatics. 2003. 19(7):891-2.
13. Breiman L. Random Forests. Machine Learning, 2001: 45, 5-32.
14. Joachims T., Learning to Classify Text Using Support Vector Machines. Dissertation, Kluwer, 2002.
15. Witten IH, and Frank E, Data Mining: Practical machine learning tools with Java implementations, Morgan Kaufmann, San Francisco, 2000
16. Jones KS. Information retrieval experiment. London Butterworths 1981. p 213-255.
17. Flach, P., The Many Faces of ROC Analysis in Machine Learning, ICML-04 Tutorial. 2004. Notes available from http://www.cs.bris.ac.uk/flach/ ICML04tutorial/
18. Foster Probost, Machine learning from imbalanced data sets 101, Invited paper for the AAAI'2000 Workshop on Imbalanced Data Sets. 2000
19. Chao Chen, Andy Liaw & Leo Breiman, Using Random Forest to Learn Imbalanced Data, Technical Report, No.666, Department of Statistics, University of Berkely, July 2004