Journal of
CHEMICAL INFORMATION
AND MODELING

Article

pubs.acs.org/jcim

PyDPI: Freely Available Python Package for Chemoinformatics, Bioinformatics, and Chemogenomics Studies

Dong-Sheng Cao,*,[†] Yi-Zeng Liang,[‡] Jun Yan,[‡] Gui-Shan Tan,[†] Qing-Song Xu,[§] and Shao Liu[⊥]

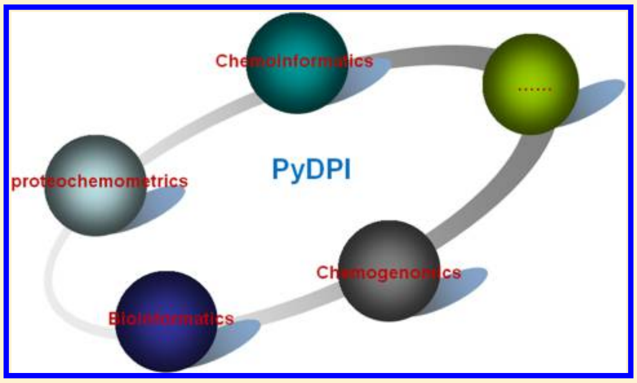
[†]School of Pharmaceutical Sciences, Central South University, Changsha 410013, P.R. China

[‡]Research Center of Modernization of Traditional Chinese Medicines and [§]School of Mathematics and Statistics, Central South University, Changsha 410083, P.R. China

[⊥]Xiangya Hospital, Central South University, Changsha 410008, P.R. China

Supporting Information

ABSTRACT: The rapidly increasing amount of publicly available data in biology and chemistry enables researchers to revisit interaction problems by systematic integration and analysis of heterogeneous data. Herein, we developed a comprehensive python package to emphasize the integration of chemoinformatics and bioinformatics into a molecular informatics platform for drug discovery. PyDPI (drug−protein interaction with Python) is a powerful python toolkit for computing commonly used structural and physicochemical features of proteins and peptides from amino acid sequences, molecular descriptors of drug molecules from their topology, and protein−protein interaction and protein−ligand interaction descriptors. It computes 6 protein feature groups composed of 14 features that include 52 descriptor types and 9890 descriptors, 9 drug feature groups composed of 13 descriptor types that include 615 descriptors. In addition, it provides seven types of molecular fingerprint systems for drug molecules, including topological fingerprints, electro-topological state (E-state) fingerprints, MACCS keys, FP4 keys, atom pair fingerprints, topological torsion fingerprints, and Morgan/circular fingerprints. By combining different types of descriptors from drugs and proteins in different ways, interaction descriptors representing protein−protein or drug−protein interactions could be conveniently generated. These computed descriptors can be widely used in various fields relevant to chemoinformatics, bioinformatics, and chemogenomics. PyDPI is freely available via https://sourceforge.net/projects/pydpicao/.



INTRODUCTION

The emergence of molecular medicine and the elucidation of the human genome in 2001 provided more opportunity to discover new unknown target proteins of drugs.[1,2] Now almost all members of a target family are visible and accessible at the DNA sequence level. Targets and ligands associated with a molecular target are no longer viewed as singular objects having no interrelationship, and the systematic exploration of selected target families appears to be a promising way to speed up and further industrialize target-based drug discovery, especially in the target identification and lead finding processes.[3,4] The development of high-throughput experimental technology greatly promotes the accessibility of protein−protein interaction and protein−ligand interaction data.[5] Several freely available databases that focus on protein−protein and drug−target relations are also emerging in the public sector, such as the Bimolecular Interaction Network Database (BIND),[6] the Database of Interaction Proteins (DIP),[7] STITCH,[8] the Human Protein References Database (HPRD),[9] the Therapeutic Target Database (TTD),[10] DrugBank,[11] ChEMBL,[12] Kyoto Encyclopedia of Genes and Genomes (KEGG),[13] BindingDB,[14] PDSP database, SuperTarget and Matador,[15] and so on. The accumulated content of these databases constitutes a gold standard. The rapidly increasing amount of publicly available data in biology and chemistry enables researchers to revisit interaction problems by systematic integration and analysis of these heterogeneous data.

Investigation of interactions is a complex molecular recognition process, which is not only related to the bioinformatics projects that aim at a systematic analysis of the structure and function of proteins that scales to the genome level, but also to the chemoinformatics projects that are devoted to the analysis of structure and biological activity of drug candidates. More importantly, systematic investigation of generated knowledge in both the biological and chemical knowledge spaces is useful in simultaneously identifying both new targets and their potential ligands.[16,17] However, the chemoinformatics and bioinformatics worlds have evolved more or less independently. It is first necessary to establish an integrated molecular informatics platform that links the chemical and biological knowledge spaces.

Received: February 25, 2013
Published: September 18, 2013

© 2013 American Chemical Society

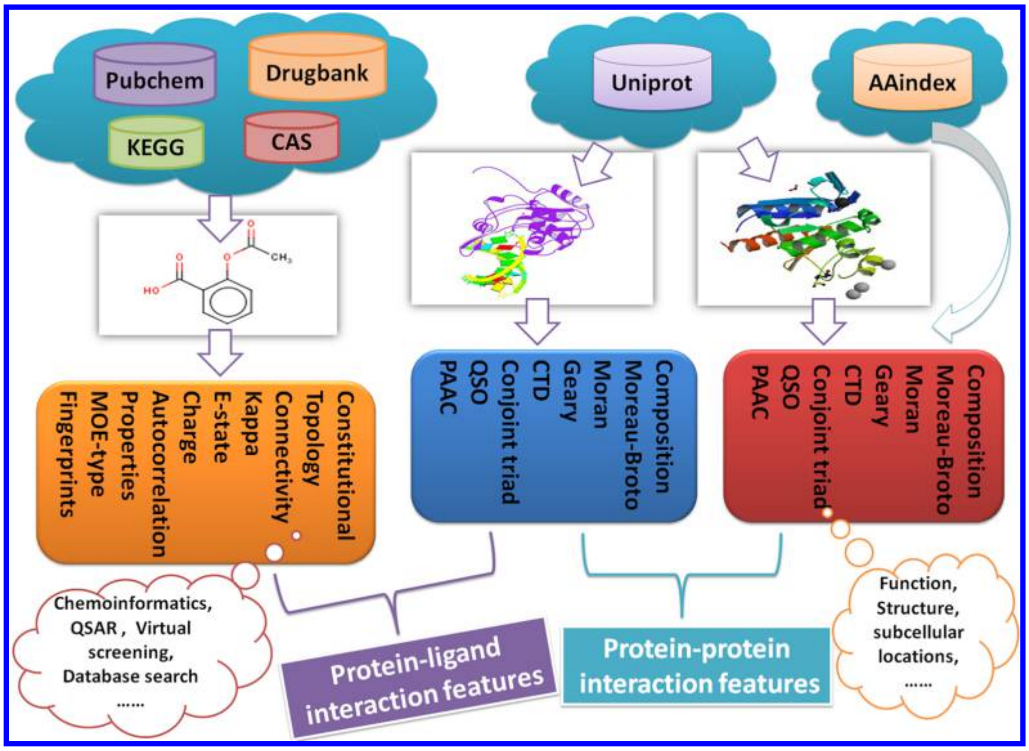


Figure 1. Schematic illustration of the PyDPI pipeline.

To develop a powerful method for studying and predicting interactions, one of the most important things to consider is how to characterize proteins and ligands in interactions and their interaction relations by a descriptor. In the field of bioinformatics, sequence-derived structural and physicochemical features have been widely used in the development of machine learning models for predicting protein structural and functional classes,[18,19] protein−protein interactions,[20] subcellular locations, and peptides of specific properties.[21] These features are highly useful for representing and distinguishing proteins or peptides of different structural, functional, and interaction profiles. Currently, these structural and physicochemical features of proteins and peptides are also routinely used to characterize target proteins in drug−target pairs and predict new drug−target associations, following the spirit of chemogenomics.[22−24] Several programs for computing protein structural and physicochemical features have been developed such as PROFEAT, BioJava, PseAAC, etc. However, they are not comprehensive and are limited only to a certain kind of features. Additionally, these are not freely and easily accessible. Of these, BioJava[25] and PseAAC[26] rarely compute several protein descriptors. PROFEAT uses a web-based engine to calculate many protein features; but, its performance is usually limited due to network reasons, and its applications are not convenient due to complicated submission steps and obscure descriptor tags.[27] In the field of chemoinformatics, molecular descriptors for small molecules have frequently been used in QSAR/SAR,[28] virtual screening,[29] database search,[30] drug ADME/T prediction,[31] and other drug discovery processes. These descriptors capture and magnify distinct aspects of molecular topology in order to investigate how molecular structure affects molecular property. Several open sources and commercial software have been developed, including Dragon, CODESSA, Chemistry Development Kit (CDK),[32] Molconn-Z, OpenBabel,[33] Cinfony,[34] Indigo, JOELib, Avogadro, and RDKit. Although a number of tools, which are either open sources or commercial software, have been developed and widely used in two fields, their applications only focus on the analysis of either small molecules or proteins. To the best of our knowledge, there is currently no open source code or tools available for the integration and analysis of the increasingly popular interaction problems.[35]

Here, we describe a comprehensive molecular representation tool, called drug−protein interaction with Python (PyDPI) to emphasize the integration of chemoinformatics and bioinformatics into a chemogenomics platform for drug discovery. PyDPI mainly focuses on the study of molecular representation techniques for not only small molecules and proteins but also interactions of protein−protein and protein−ligand. PyDPI can calculate a large number of structural and physicochemical features of proteins and peptides from amino acid sequences, molecular descriptors of drug molecules from their topology, and protein−protein interaction and drug−protein interaction descriptors. Additionally, it can also compute protein descriptors based on user-defined properties, which are easily accessible from the AAindex database. To easily use the PyDPI utilities and functionalities, we provide a uniform interface in PyDPI to perform data analysis. To introduce and describe PyDPI's utility and application, we used enzyme−drug interaction data as an example to show that PyDPI can be used as an integral part of an analytical pipeline. Our computational algorithms are extensively tested, and the computed features have also been used in a number of published studies for predicting drug activity and ADME/T properties, proteins of functional classes, protein−protein interactions, and protein−ligand interactions.

■ MATERIAL AND METHODS

PyDPI Package. PyDPI is a python package, which is licensed under the General Public License (GPL) and is freely available through the google code web. By developing this package and conforming to the strict guidelines for the user, we are able to utilize and incorporate the corresponding modules to assist in representing various molecular objects. PyDPI for the operation of small molecules builds on RDKit OpenBabel, and Pybel. A schematic overview of PyDPI is given in Figure 1. The 70 page PyDPI manual and 35 page user guide are provided in the Supporting Information as Supporting Material 1 and 2.

PyDPI is a powerful open source package for the extraction of features of complex molecular data. The PyDPI package contains several functions and modules manipulating proteins/peptides and small molecules. According to the difference of processing objects, they are separately included in two directories: drug and protein. Each group of molecular features is separately provided in an individual module, in which a python function corresponds to the computation of a molecular feature. There are six modules corresponding to the calculation of protein/peptide descriptors from six feature groups (i.e., AAComposition, Autocorrelation, CTD, ConjointTriad, QuasiSequenceOrder, PseudoAAC). Likewise, 14 modules corresponding to the calculation of small molecules for 12 feature groups (i.e., constitution, topology, connectivity, estate, kappa, basak, bcut, moreaubroto, moran, geary, charge, molproperty, moe, fingerprint) exist. The detailed instructions for modules and built-in methods are provided in the form of HTML in PyDPI (see Supporting Information Supporting Material 3). The user could use the corresponding function to calculate the molecular feature as needed. However, to conveniently calculate molecular features, PyDPI provides three simple and easy-to-use modules (i.e., pypro, pydrug, pydpi), which are used for easily manipulating protein/peptide molecules, small molecules, and interactions, respectively. The pypro module covers all methods used for computing descriptors of proteins/peptides. Additionally, it also contains a PyPro class, which encapsulates all operations for proteins/peptides. By importing the pypro module, there are two means to compute these structural and physicochemical features from protein or peptide sequences. One is to use the built-in functions. One could import related functions to compute these features as needed. The other is to call the PyPro class. One could construct a PyPro object with a protein sequence input, and then call corresponding methods to calculate these features. Likewise, the pydrug module includes a PyDrug class, which encapsulates all methods used for computing descriptors of small molecules. The pydpi module comprises two classes (PyDPI, PyPPI) that could be used for calculating protein−ligand interaction descriptors and protein−protein interaction descriptors, respectively. According to the applications to different fields such as chemoinformatics, bioinformatics, and chemogenomics, the user could apply related modules to perform corresponding operations. For instance, the researchers interested in performing similarity searching in the database could use the fingerprint functionality in pydrug. The researchers interested in investigating the molecular interaction space could use PyDPI or PyPPI class in pydpi.

In addition to main functionalities mentioned above, PyDPI can also provide a number of supplementary functionalities to facilitate the computation of molecular features. To obtain protein sequences easily, PyDPI provides a GetProteinFromUniprot module and a getpdb module, with which the user could easily get protein sequences from the Uniprot Web site and RCSB PDB Web site by providing IDs or a file containing IDs. A check module is also provided to ensure that the input for subsequent calculation is reliable. To facilitate the accessibility of the property or distance matrix of amino acids, PyDPI provides an AAIndex module, which helps the user automatically download the needed property from the AAindex database.[36] The output from the AAIndex module could be directly used as the user-defined property to calculate the above-mentioned descriptors, greatly enlarging the applications to our calculated features. Drug or small molecules could be directly downloaded from five databases (KEGG, CAS, PubChem, DrugBank, ChEMBL) by providing corresponding IDs. Furthermore, eight types of similarity measures could be calculated in the fingerprint module to compare the similarity between two molecules by using any molecular fingerprint system.

The PyDPI implementation of each of these algorithms was extensively tested by using a number of test proteins and drug molecules. The computed descriptor values were also compared to the known values for these molecules from different software tools to ensure that the computation is accurate. For drug descriptors, we compared our calculated descriptors with those from Dragon, MOE (Molecular Operating Environment from Chemical Computing Group), or MODEL (Molecular Descriptor Lab). If our calculated descriptor is identical to those from these tools, we will confirm that this descriptor is correctly coded. Protein descriptors can be compared with those from PROFEAT (Protein Feature Server) or the PseAAC server (http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/). Similarly, if our calculated descriptor is identical to those from PROFEAT and PseAAC, we will conform that this protein descriptor is correctly calculated.

Protein or Peptide Descriptors From Amino Acid Sequences. A list of features for proteins and peptides covered by PyDPI is summarized in Table 1. These features can be divided into six groups, each of which has been independently used for predicting protein- and peptide-related problems by using machine learning methods. Of these, each group corresponds to an individual python module. More detailed description and references can be found in the PyDPI manual (Supporting Information Supporting Material 1).

The first group includes three features: amino acid composition, dipeptide composition, and tripeptide composition, with 3 descriptors and 8420 descriptor values. These descriptors represent the fraction of each amino acid type, dipeptide type, and tripeptide type in a protein sequence. These simplistic descriptors have been used to predict protein fold and structural classes,[37] functional classes,[38] and subcellular locations[39] at accuracy levels of 72−95%, 83−97%, and 79−91%, respectively. All functionalities used for computing these three features are all included in the AAComposition module.

The second group consists of three different autocorrelation features: normalized Moreau−Broto autocorrelation, Moran autocorrelation, and Geary autocorrelation. The autocorrelation features describe the level of correlation between two protein or peptide sequences in terms of their specific structural or physicochemical property. In the default settings of PyDPI, there are eight amino acid properties used for deriving these autocorrelation descriptors. The first is hydrophobicity scale derived from the bulk hydrophobic character for 20 types of amino acids in 60 protein structures. The second is the relative mutability obtained by multiplying the number of observed mutations by the frequency of occurrence of the individual amino acids. The third is the free energy of amino acid solution

Table 1. List of PyDPI Computed Features for Protein Sequences

feature group	features	number of descriptors
amino acid composition	amino acid composition	20
	dipeptide composition	400
	tripeptide composition	8000
autocorrelation	normalized Moreau−Broto autocorrelation	240[a]
	Moran autocorrelation	240[a]
	Geary autocorrelation	240[a]
CTD	composition	21
	transition	21
	distribution	105
conjoint triad	conjoint triad features	343
quasi-sequence order	sequence order coupling number	60
	quasi-sequence order descriptors	100
pseudo amino acid composition	pseudo amino acid composition	50[b]
	pseudo amino acid composition	50[c]

[a]The number depends on the choice of the number of properties of amino acid and the choice of the maximum values of the lag. The default is use eight types of properties and lag = 30. [b]The number depends on the choice of the number of the set of amino acid properties and the choice of the λ value. The default is use three types of properties proposed by Chou et al.[48] and λ = 30. [c]The number depends on the choice of the λ value. The default is λ = 15.

in water measured by Charton.[77] The fourth is the polarizability parameter computed from the group molar refractivity values originally provided by Charton.[77] The fifth is the residue accessible surface areas taken from average values from folded proteins. The sixth is the steric parameters derived from the van der Waals radius of amino acid side-chain atoms. The seventh is the amino acid residue volumes measured by Bigelow.[78] The eighth is the average flexibility index derived from the statistical average of B-factors of each type of amino acids in the available protein X-ray crystallographic structures. Thus, three autocorrelation features are computed, each having 8 descriptor types and 8 × 30 = 240 descriptors. Autocorrelation descriptors have been used for predicting transmembrane protein types,[40] protein helix contents,[41] and protein secondary structural contents[42] at accuracy levels of 82−94%, 85%, and 91−94%, respectively. Apart from these descriptors, we can also compute previous descriptors based on user-defined properties, which are easily accessible from the AAindex database. All functionalities used for computing these descriptors are included in the Autocorrelation module.

The third group contains three feature sets: composition (C), transition (T), and distribution (D), with a total of 3(C) + 3(T) + 5 × 3(D) = 21 descriptor types, and 147 descriptors. They represent the amino acid distribution pattern of a specific structural or physicochemical property along a protein or peptide sequence.[43] Seven types of physicochemical properties have been used for calculating these features, including hydrophobicity, polarity, charge, polarizibility, normalized van der Waals volume, secondary structures, and solvent accessibility (see Table 1 in the PyDPI manual). C is the number of amino acids of a particular property (e.g., hydrophobicity) divided by the total number of amino acids in a protein sequence. T characterizes the percent frequency with which amino acids of a particular property is followed by amino acids of a different property. D measures the chain length within which the first, 25%, 50%, 75%, and 100% of the amino acids of a particular property are located, respectively. A detailed step and example for calculating CTD descriptors can be found in the PyDPI manual (see Figure 2 in the PyDPI manual). These CTD features have been widely used for predicting protein folds,[43] protein−protein interactions,[44] and protein functional families[45] at accuracy levels of 74−100%, 77−81%, and 67−99%, respectively. All functionalities used for computing CTD descriptors are included in the CTD module.

The fourth group, conjoint triad descriptors, proposed by Shen et al., was originally designed to represent protein−protein interactions.[20] These conjoint triad features abstract the features of protein pairs based on the classification of amino acids. Twenty amino acids were clustered into several classes according to their dipoles and volumes of side chains, (see Table 2 in the PyDPI manual). Herein, the dipoles and volumes of side chains of amino acids, reflecting electrostatic and hydrophobic interactions, were calculated, respectively, by using the density-functional theory method B3LYP/6-31G* and molecular modeling approach. The reason for dividing amino acids into seven groups is that amino acids within the same class likely involve synonymous mutations because of their similar characteristics. The conjoint triad features consider the properties of one amino acid and its neighboring ones and regard any three continuous amino acids as a unit. Thus, the triads can be differentiated according to the classes of amino acids, i.e., triads composed by three amino acids belonging to the same classes could be treated identically. For amino acids that have been catalogued into seven classes, we can finally construct a 7 × 7 × 7 = 343-dimensional vector, each dimension of which records the frequency of each triad appearing in the protein sequence. For detailed information on how to calculate these features, please refer to the PyDPI manual. Applying the conjoint triad features to the prediction of protein−protein interactions, the support vector machine based on S-kernel function obtained an average prediction accuracy of 83.90% on test sets. All functionalities used for computing the conjoint triad features are included in the ConjointTriad module.

The fifth group includes two sequence-order feature sets: one is sequence-order-coupling number with 2 descriptor types and 60 descriptors and the other is quasi-sequence-order with 2 descriptor types and 100 descriptors. These features are derived from both the Schneider−Wrede physicochemical distance matrix and Grantham chemical distance matrix. The sequence-order features can be used for representing amino acid distribution patterns of a specific physicochemical property along a protein or peptide sequence, which have been used for predicting protein subcellular locations[46,47] at accuracy levels of 72.5−88.9%. All functionalities used for computing sequence-order features are included in the QuasiSequenceOrder module.

The sixth group contains two types of pseudoamino acid compositions (PseAAC): type I PseAAC[48]with 50 descriptors and type II PseAAC (i.e., amphiphilic PseAAC)[49] with 50 descriptors. In simple amino acid composition, all the sequence-order effects are missing. To avoid losing the sequence-order information completely, the concept of PseAAC, developed by Chou,[48] was mainly used to reflect the composition of amino acids and the sequence-order information (at least partially) through a set of correlation factors. PseAAC has been frequently used in improving the prediction quality for subcellular location of proteins and their other attributes. All functionalities used for computing PseAAC features are included in the pseudoAAC module.

Molecular Descriptors From Molecular Topology. Twelve groups of molecular descriptors are calculated to represent drug molecules in PyDPI. A detailed list of descriptors for drug molecules covered by PyDPI is summarized in Table 2. These descriptors capture and magnify distinct aspects of chemical structures. The usefulness of molecular descriptors in the representation of molecular information is reflected in their widespread adoption and use across a broad range of applications and methodologies, as reported in a large number of published articles.[50−54] These 12 groups of descriptors are provided in 14 individual modules, respectively.

Table 2. List of PyDPI Computed Features for Small Molecules

feature group	features	number of descriptors
constitution	molecular constitutional descriptors	30
topology	topological descriptors	25
connectivity	molecular connectivity indices	44
E-state	E-state descriptors	237
kappa	kappa shape descriptors	7
Burden	Burden descriptors	64
information	topological information descriptors	21
autocorrelation	Moreau−Broto autocorrelation	32
	Moran autocorrelation	32
	Geary autocorrelation	32
charge	charge descriptors	25
property	molecular property	6
MOE-type	MOE-type descriptors	60
fingerprints	topological fingerprints	2048
	MACCS keys	166
	FP4 keys	307
	E-state fingerprints	79
	atom pairs fingerprints	
	topological torsions	
	Morgan fingerprints	

Constitutional descriptors consist of 30 descriptors, which are mainly used for characterizing the composition of chemical element type and chemical bond type, path length, hydrogen bond acceptor, and donator in the constitution module. Topology descriptors are those invariants calculated from molecular topological structure, which have been successfully used for predicting molecular physicochemical properties, such as boiling point and retention index. In the topology module, 25 commonly used topological descriptors like Weiner index, Balaban index, Harary index, and Schultz index are computed. Molecular connectivity indices consist of 44 descriptors that reflect simple molecular connectivity and valence connectivity for different path orders, cycle, or cluster size. They are among the most popular indices and are calculated from the vertex degree of the atoms in the H-depleted molecular graph. The connectivity module is responsible for the calculation of all connectivity descriptors. Kappa shape indices are computed through the kappa module, each of which represents a particular shape attribute of the molecule, such as molecular flexibility, molecular steric effect, molecular symmetry, etc. Burden descriptors are originally proposed to address searching for chemical similarity/diversity on large data sets. However, they have been widely applied to various toxicity problems. PyDPI calculates four different Burden matrices weighted by atomic masses, atomic van der Waals volumes, atomic Sanderson electronegatives, and atomic polarizabilities. The first eight highest and the first eight lowest eigenvalues from different Burden matrices are calculated as descriptors. Topological information indices, proposed by Basak, represent the difference of chemical environment of different atoms in a molecule, which have been widely used in various QSAR/QSPR studies. Twenty-one information descriptors could be calculated by PyDPI. Seventy-nine atom-type E-state indices were proposed in the estate module as molecular descriptors encoding topological and electronic information related to particular atom types in the molecule. E-state indices are especially useful in the prediction of drug ADME/T. In addition, the maximum and minimum of E-state values of 79 atom types are also calculated as molecular descriptors in PyDPI. Six commonly used molecular properties are directly used in the molproperty module for representing the molecule, including molar refractivity, LogP based on Crippen method and its square, topological polarity surface area, unsaturation index, and hydrophilic index. Twenty-five charge descriptors are computed based on Gasteiger−Marseli partial charges in the charge module, which describe electronic aspects both of the whole molecule and of particular regions, such as atoms, bonds, and molecular fragments. Electrical charges in the molecule are the driving force of electrostatic interactions, and it is well-known that local electron densities or charges play a fundamental role in many chemical reactions, physicochemical properties, and receptor−ligand binding. Three types of autocorrelation descriptors (i.e., Moreau−Broto, Moran, and Geary) are computed in the three individual modules: moreaubroto, moran, and geary. Four carbon-scaled atomic properties are used to calculate these descriptors, including atomic mass, atomic van der Waals volume, atomic Sanderson electronegativity, and atomic polarizability. Sixty MOE-type descriptors can be computed from connection table information based on atomic contributions to van der Waals surface area, LogP, molar refractivity, partial charge, and E-state value. These descriptors have been frequently applied to the construction of QSAR models for boiling point, vapor pressure, thrombin/factor Xa activity, blood−brain barrier permeability, and compound classification. All functionalities used for computing MOE-type descriptors are included in the moe module.

Molecular Fingerprints. Another striking feature we developed into PyDPI is the computation of a number of molecular fingerprints. Molecular fingerprints are string representations of chemical structures, which consist of bins, each bin being a substructure descriptor associated with a specific molecular feature. Seven types of molecular fingerprints are provided in PyDPI, including topological fingerprints, E-state fingerprints, MACCS keys, FP4 keys, atom pairs fingerprints, topological torsion fingerprints, and Morgan/circular fingerprints. All functionalities used for computing molecular fingerprints are included in the fingerprint module. The usefulness of these molecular fingerprints covered by PyDPI for representing structure features of small molecules have been sufficiently demonstrated by a number of published studies of the development of machine learning classification systems in QSAR/SAR, drug ADME/T prediction,[55] similarity searching, clustering, ranking, and classification.

The most commonly used class of fingerprints are Daylight-type topological fingerprints, which are proposed through Daylight Chemical Information Systems. This type of topological fingerprints use features based upon the presence of paths of varying lengths containing specific atom types. This generates a sparse binary vector, which is commonly folded to a bitset of specified size (e.g., 1024 or 2048) to reduce its size for ease of manipulation. Daylight-type topological fingerprints are mainly designed for substructure and similarity searching.

The next three fingerprints (E-state fingerprints, FP4 fingerprints, and MACCS fingerprints) are strictly some structural keys which define the presence or absence of predefined structural fragments, encoding this information in an array of binary values. These structural keys greatly rely on the use of a predefined fragment dictionary during practical applications. E-state fingerprints contain 79 substructure patterns, defined by Kier and Hall. The dictionary of FP4 fingerprints contains 307 mostly common substructure patterns. It is originally written in an attempt to represent the classification of organic compounds from the viewpoint of an organic chemist. The MACCS fingerprints use a dictionary of MDL keys, which contains a set of 166 mostly common substructure features. These are referred to as the MDL public MACCS keys. These three types of fingerprints have shown a high prediction quality in classifying drug ADME/T properties.[55−57]

An atom pair can be defined as a substructure composed of two non-hydrogen atoms and an interatomic separation as follows: ⟨atom 1 description⟩ − ⟨separation⟩ − ⟨atom 2 description⟩. The two atoms in an atom pair need not be directly connected. The ⟨separation⟩ tells how far apart they are, measured as the number of atoms in the shortest bond-by-bond path that contains both atoms 1 and 2. The ⟨description⟩ of each atom tells its chemical type, the number of non-hydrogen atoms attached to it, and the number of bonding π electrons that it bears. Only the common atom types C, O, N, S, F, Cl, Br, I, P, Si, B, Se, and generic symbol "Y" (atoms of any other type) is used. The atom pair fingerprints can capture possible long-range correlations between atoms in active molecules.[58]

Topological torsion can be defined as a linear sequence of four consecutively bonded non-hydrogen atoms, each described by its atomic type, the number of non-hydrogen branches attached to it, and its number of π electron pairs.[59] Schematically, the topological torsion can be illustrated as follows: ⟨NPI-TYPE-NBR⟩ − ⟨NPI-TYPE-NBR⟩ − ⟨NPI-TYPE-NBR⟩ − ⟨NPI-TYPE-NBR⟩, where NPI indicates the number of π electrons on each atom, TYPE indicates the atomic species, and NBR is the number of non-hydrogen branches. The topological torsion fingerprints were inspired by the fact that the torsion angle (defined by four consecutively bonded atoms) is the minimal structural unit in terms of which the conformation of a molecule can be completely described. Unlike atom pair fingerprints, topological torsion fingerprints are a type of short-range descriptors.

Each Morgan feature represents a circular substructure around a center atom.[60] The algorithm starts with the initial atom identifier of the center atom and grows a circular substructure around this atom throughout a defined number of iterations (i.e., search depth). For each round, the current extended version of the feature is added to the final set of features. Morgan fingerprints are closely related to the extended-connectivity fingerprints. In PyDPI, we could compute different extended-connectivity fingerprints by assigning the search depth.

Protein−Protein Interaction Descriptors. Let $\mathbf{F}_\mathrm{a} = \{\mathbf{F}_\mathrm{a}(\mathrm{i}), i = 1, 2, ..., p\}$ and $\mathbf{F}_\mathrm{b} = \{\mathbf{F}_\mathrm{b}(i), i = 1, 2, ..., p\}$ are the two descriptor vectors for interaction protein A and protein B, respectively. There are three methods to construct the interaction descriptor vector $\mathbf{F}$ for A and B:

(1) Two vectors $\mathbf{F}_\mathrm{ab}$ and $\mathbf{F}_\mathrm{ba}$ with dimension of $2p$ are constructed: $\mathbf{F}_\mathrm{ab} = (\mathbf{F}_\mathrm{a}, \mathbf{F}_\mathrm{b})$ for interaction between protein A and protein B and $\mathbf{F}_\mathrm{ba} = (\mathbf{F}_\mathrm{b}, \mathbf{F}_\mathrm{a})$ for interaction between protein B and protein A.

(2) One vector $\mathbf{F}$ with dimension of $2p$ is constructed: $\mathbf{F} = \{\mathbf{F}_\mathrm{a}(i) + \mathbf{F}_\mathrm{b}(i), \mathbf{F}_\mathrm{a}(i) \times \mathbf{F}_\mathrm{b}(i), i = 1, 2, ..., p\}$.

(3) One vector $\mathbf{F}$ with dimension of p^2 is constructed by the tensor product: $\mathbf{F} = \{\mathbf{F}(k) = \mathbf{F}_\mathrm{a}(i) \times \mathbf{F}_\mathrm{b}(j), i = 1, 2, ..., p, j = 1, 2, ..., p, k = (i - 1) \times p + j\}$.

Protein−Ligand Interaction Descriptors. There are two methods for construction of descriptor vector $\mathbf{F}$ for protein−ligand interaction from the protein descriptor vector $\mathbf{F}_\mathrm{t}$ $(\mathbf{F}_\mathrm{t}(i), i = 1, 2, ..., p_\mathrm{t})$ and ligand descriptor vector $\mathbf{F}_\mathrm{d}$ $(\mathbf{F}_\mathrm{d}(i), i = 1, 2, ..., p_\mathrm{d})$:

(1) One vector $\mathbf{V}$ with dimension of $p_\mathrm{t} + p_\mathrm{d}$ are constructed: $\mathbf{F} = (\mathbf{F}_\mathrm{t}, \mathbf{F}_\mathrm{d})$ for interaction between protein T and ligand D.

(2) One vector $\mathbf{V}$ with dimension of $p_\mathrm{t} \times p_\mathrm{d}$ is constructed by the tensor product: $\mathbf{F} = \{\mathbf{F}(k) = \mathbf{F}_\mathrm{t}(i) \times \mathbf{F}_\mathrm{d}(j), i = 1, 2, ..., p_\mathrm{t}, j = 1, 2, ..., p_\mathrm{d}, k = (i - 1) \times p_\mathrm{t} + j\}$.

Data. To illustrate the application of PyDPI, enzyme−drug interaction data previously used by Yamanishi et al. was chosen as an example.[61] This data set is composed of 445 known drugs and 664 target proteins, with 2926 associated drug-target interactions. Generally speaking, drug−target interaction networks can be conveniently modeled as a bipartite graph, where the nodes are target proteins or drug molecules and edges represent drug−target interactions. Initially, the graph only contains some edges that describe the real drug−target interactions determined by experiments. In our study, all 2926 drug−target interaction pairs are used as the positive examples. The corresponding negative examples were derived from the above positive examples as follows: (1) separate the pairs in positive samples into single drugs and proteins; (2) recouple these singles into pairs in a way that none of them occur in the positive data set. However, for a completely connected bipartite graph, 445 × 664 = 295 480 connections (i.e., interaction pairs) exist. Therefore, the possible number of negative samples (e.g., 295 480 − 2926 = 292 554) is significantly larger than the number of positive samples (e.g., 2926). To overcome the bias caused by unbalanced problems, we randomly picked the negative pairs formed above until they reached the number of positive pairs. Fifty of the constructed negative data sets are randomly generated to demonstrate the stability of predictive results (see the Supporting Information Supporting Material 4).

Random Forest. Random forest (RF), developed by Bremain and Culter, is capable of describing the relationship between independent and dependent variables with high flexibility and sufficient accuracy. RF has been successfully applied in many biological contexts: cancer tissue classification, protein domain classification, nucleosome positioning, etc. An extended depiction and study of theory on RF can be found on the Web site of Bremain[79] or the papers of Svetnik et al.[80] The RF algorithm grows a collection, called a forest, of the unpruned classification trees and uses these for classifying a data point into one of the classes. Two types of randomness, bootstrap sampling of samples and random selection of input features, are used in the algorithm to make sure that the classification trees grown in the forest are dissimilar and uncorrelated from each other. A forest is grown by using *ntree* bootstrapped samples, each of size *N* randomly drawn from the original data of *N* training samples with replacement. The first type of randomization helps to build an ensemble of trees and to increase diversity among the trees. In each bootstrap sample, about two-thirds of the original training samples are used to grow a classification tree. About one-third of the samples are left, called out of bag (OOB) samples. These samples are used to obtain unbiased estimates of correct classification rates and feature importance measure. The second type of randomness is used during building each tree. For each node of a tree, the RF algorithm randomly selects *mtry* features and uses only them to determine the best possible split using the *Gini* index as the splitting criterion. Predictions for test data are carried out either by the majority vote of classification trees or are based on a threshold selected by the user. The number of trees (*ntree*) to be grown is chosen appropriately to achieve low error rate of convergence. Finally, RF can produce scores or probability outputs that serve to rank predictions according to confidence and have a useful probabilistic interpretation.

■ RESULTS

To construct a predictive model, we first use our developed PyDPI to represent each drug−target pair. To calculate the descriptors of a drug−target pair, two inputs are required: a protein sequence and a drug structure. However, in view of the supply of molecular IDs in a large interaction data set, PyDPI provides separate download modules to help the user obtain the required molecular structure. Thus, we could automatically download all 445 drugs and 664 target proteins from KEGG according to their corresponding IDs by PyDPI. All target proteins are then saved in an independent .txt file, and each protein is represented by its amino acid sequence in a line. All drug molecules are saved as .sdf file format and, then, are further transformed as .smi file format so as to conveniently cope with them. By PyDPI, all previous steps are automatic and easy. The use of PyDPI greatly facilitates the building of predictive models for protein−ligand interactions.

Once we prepare these two structure files for proteins and drugs, we can conveniently compute the structural and physicochemical features from enzyme sequences and various molecular descriptors from drug structures, respectively. These computed features from enzymes and drugs can finally be used to calculate the drug−target interaction features according to different interaction representation methods. Investigation of interactions between proteins and ligands is a complex molecular recognition process, which is influenced or dominated by different types of structural factors from ligands and proteins, such as, for drugs, molecular constitution, topology, shape, charge, substructures, and so on; for proteins, shape of the binding site, amino acid composition, secondary structure, hydrophobicity, polarity, polarizability, and interaction energy, and so on. Large-scale study of interactions between proteins and ligands thereby need to consider as many influence factors as possible. Due to complex molecular recognition mechanisms between enzymes and drugs, a complete set of features was used to characterize the enzymes and drugs. Here, for enzyme sequences, 667 structural and physicochemical features were computed, including 20 amino acid composition features, 240 Geary autocorrelation features, 147 CTD features, 60 sequence order coupling numbers, 100 quasi-sequence order descriptors, and 100 PseAAC descriptors. For drug molecules, we computed 696 descriptors, including 30 constitution descriptors, 25 topological descriptors, 44 connectivity indices, 237 E-state descriptors, 7 kappa shape features, 96 autocorrelation descriptors (Moreau−Broto, Moran, Geary), 25 charge descriptors, 6 molecular properties, 60 MOE-type descriptors, and 166 MACCS structural keys. Finally, these features from enzymes and drugs were combined to generate 667 + 696 = 1363 interaction descriptors, as described above. All calculations for protein features, drug features, and interaction features are very simple when using PyDPI.

We employed the random forest (RF) algorithm to construct our predictive model because of its excellent reputation among the bioinformatics communities.[62,63] The Random Forest package in R, developed by Bremain and Culter, was used to build the RF prediction models (The randomForest package is freely available at http://cran.r-project.org/web/packages/randomForest/index.html). As in other multivariate statistical models, the performance of RF for classification depends on the combination of several parameters. In general, RF involves two parameters: the number of randomly selected variables (*mtry*) and the number of trees grown (*ntree*), which need to be further optimized. *mtry* is a regularization parameter that controls the trade-off between accuracy of individual trees and diversity of individual trees, ranging from 1 to the total number of the variables (p). A commendatory value for *mtry* in classification problems is usually taken to be the square root of the total number of variables ($\sqrt{p}$). To achieve the better performance, we screen *mtry* values ranging from 20 to 150 with a step of 5. Normally, the performance of a classification model might be severely affected if some irrelevant features are not removed prior to the model training. However, it has been shown that the feature selection is not quite necessary in the RF model, as the OOB error is used for estimating feature importance. We also select an appropriate number of trees to be grown to achieve a low error rate of convergence. We finally found that ensemble of 600 decision trees (*ntree* = 600) can generally obtain a steady and low error rate. The optimal model was determined using OOB error estimate as an objective function.

To evaluate classification performance, we first used a 5-fold cross validation method. Initially, the whole data set to be classified was randomly partitioned into five subsets. One subset was reserved as a validation data set, and the classifier was trained in the remaining four subsets. The constructed classifier was then used to predict the reserved validation data set to obtain its accuracy. The process was repeated five times so that every drug−target association was classified. Between there is a trade-off between sensitivity and specificity, we measure the quality of the classifier by calculating the area under the receiver operator characteristic (ROC) curve (auROC). An ROC curve shows the false-positive rate along the x-axis and the true-positive rate along the y-axis, as the classification threshold varies for declaring a prediction to be a real site. A model with no predictive ability would yield the diagonal line. The closer the auROC is to 1, the greater the predictive ability of the model is.

With the best parameter set in RF, the RF can successfully distinguish the drug−target interactions with auROC = 0.967 ±

0.011 and prediction accuracy of 89.76% ± 1.54%. The sensitivity and specificity of the RF model is 91.13% ± 1.43% and 88.34% ± 2.16%, respectively. The ROC curve reveals a false positive rate of 16% at a sensitivity of 90%.This is significantly better than the false positive rate of 90% from random predictions at this sensitivity (p-value < 10^{-37}). It can be seen from Figure 2 that the ROC curves of 50 models are very robust, suggesting a real model predictivity. The results are also comparative to or even better than those from other published studies.[64,65]

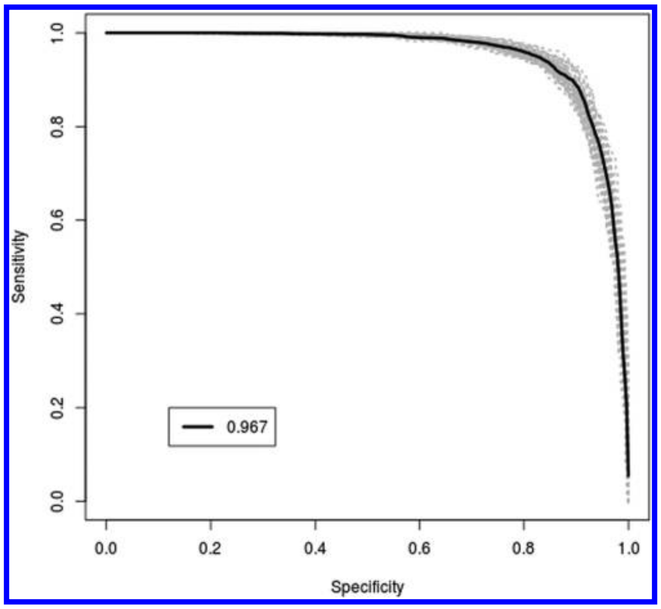


Figure 2. Receiver operator characteristics curves on 50 5-fold cross-validation data sets. The black line indicates the average value of 50 receiver operator characteristic curves.

To further demonstrate the prediction capability of our models, our models should also be validated by predicting the labels of other interactions not used in the training set, but whose labels are known (i.e., independent test set). The major difference between cross-validation and independent test is that the chemicals selected in the latter case are in a sense random. This provides a more robust evaluation of the model's prediction capability for untested interactions than cross-validation. Thus, the total data is further split into the training set of 4400 interactions and the independent test set of 1452 interactions randomly. The training set is used to reconstruct RF model, and then, the independent test set is used for assessing the predictive performance of RF. In order to demonstrate the stability and reliability of our prediction, this process is repeated ten times to avoid the influence of random partition (see the Supporting Information Supporting Material 5). The prediction results show that the RF model obtained prediction accuracy of 87.48% ± 2.47%, sensitivity of 90.16% ± 1.86%, and specificity of 85.86% ± 2.54%, respectively. These results are consistent with those from 5-fold cross validation. Together with the results by 5-fold cross validation, these results by the independent test set strongly indicate that our RF model is very predictive and robust.

■ DISCUSSION

Considering the amazing rate at which data are accumulated in the fields of chemistry and biology, new tools that process and interpret large and complex data are increasingly important. However, to our knowledge, no open source or freely available tool exists to perform these functions. The proposed toolkit makes a step in this direction providing a way to fully integrate information from chemical space and biology space into a pharmacologic space. PyDPI is a powerful open source package for the extraction of features of complex molecular data. After representation, different statistical learning tools can be applied for further analysis and visualization of the data. The case study shows how PyDPI was used to describe drug−target pairs and establish a model in a routing way. The application domain of PyDPI is not limited to the interaction data. It can be applied to a broad range of scientific fields such as QSAR/SAR, database search, diversity analysis, virtual screening, protein function/structure/family classification, subcellular locations, protein−protein interactions, protein−ligand interactions, chemogenoics, and proteochemometrics. We expect that PyDPI will better assist pharmacologists and biologists in characterizing, analyzing, and comparing complex molecular objects.

The current version of PyDPI has a number of strengths that make them useful for a wide variety of applications in computational biology. The usefulness of the features covered by PyDPI has been extensively tested by a number of published studies of the development of statistical learning algorithms for predicting protein structural and functional classes, protein−protein interactions, subcellular locations, and peptides of specific properties. Several web-based servers have been established to perform these tasks such as SVM-Prot[45] and Cell-Ploc.[21] The similarity principle is prominent in medicinal chemistry, although it is well-known as the similarity paradox, i.e., those very minor changes in chemical structure can result in total loss of activity. On the basis of different similarities, various molecular fingerprint systems were used for identifying novel drug targets.[66] Campillos et al.[67] proposed a novel method to identify new targets based on the similarity of side effects by Daylight-type topological fingerprints. A method to predict protein targets based on chemical similarity of their ligands was proposed by Keiser et al. by Daylight-type topological fingerprints and extended-connectivity fingerprints.[68−70] A number of studies have been performed on the modeling of the interaction of GPCR with a diverse set of ligands using a proteochemometrics approach, which aims at finding an empirical relation that describes the interaction activities of the biopolymer−molecule pairs as accurately as possible, based on a unified description of the physicochemical properties of the primary amino acid sequences of proteins, and the description of the physicochemical properties of the ligands that may interact with the proteins.[71−73] The results showed that building accurate, robust, and interpretable models for predicting the affinity data is totally possible, provided that suitable representations for proteins and ligands are used.[74,75] Moreover, a further analysis showed that the model quality greatly depended on the sequence homology of proteins, and the model was very predictive only for proteins that had similar counterparts remaining in the model.[76]

The main advantages of our proposed approach are summarized as follows: (1) PyDPI contains a selection of molecular features to analyze, classify, and compare complex molecular objects. They facilitate the exploitation of machine learning techniques to drive hypothesis from complex protein/peptide data sets, small molecule data sets, and interaction data sets. (2) Due to the application of dictionary form in PyDPI, the meaning of each descriptor is clear and explicit. This helps the researcher to interpret the model. (3) The PyDPI package can be easily applied to the construction of web-based servers that are used for solving various chemical and biological problems such as database searching, protein function/classes/family prediction, interaction prediction, and drug ADME/T prediction, etc. (4) PyDPI provides various interfaces to several popular databases such as KEGG, PubChem, Drugbank, Uniprot, etc., greatly facilitating the accessibility of molecular structure.

Owing to the modular structure of PyDPI, extensions or new functionalities can be implemented easily without complex and time-consuming alterations of the source code. In future work, we plan to apply the integrated features on various biological research questions and to extend the range of functions with new promising descriptors for the coming versions of PyDPI.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

Supporting Material 1: user guide for PyDPI 1.0. Supporting Material 2: molecular descriptors guide. Supporting Material 3: HTML files. Supporting Material 4: negative data sets. Supporting Material 5: further validation data. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*Tel.: +86 731 8883 0824. Fax: +86 731 8883 0824. E-mail: oriental-cds@163.com.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Dr. Débora Mendes for careful linguistic editing and Dr. L. X. Zhang from the Chinese Academy of Sciences for useful suggestions and discussions. We also thank Mr. Yamanishi from Kyoto University for providing the example dataset available in supporting materials. This work is financially supported by the National Nature Foundation Committee of P.R. China (Grant nos. 21075138, 21275164, 20975115, and 11271374). The studies meet with the approval of the university's review board.

■ REFERENCES

(1) Kanehisa, M.; Bork, P. Bioinformatics in the post-sequence era. *Nat. Genet.* **2003**, *33*, 305−310.

(2) Rask-Andersen, M.; Almen, M. S.; Schioth, H. B. Trends in the exploitation of novel drug targets. *Nat. Rev. Drug Discovery* **2011**, *10*, 579−590.

(3) Ashburn, T. T.; Thor, K. B. Drug repositioning: identifying and developing new uses for existing drugs. *Nat. Rev. Drug Discovery* **2004**, *3*, 673−683.

(4) Jenkins, J. L.; Bender, A.; Davies, J. W. In silico target fishing: Predicting biological targets from chemical structure. *Drug Discovery Today* **2006**, *3*, 413−421.

(5) Kuhn, M.; Campillos, M.; Gonzolez, P.; Jensen, L. J.; Bork, P. Large-scale prediction of drug-target relationships. *FEBS Lett.* **2008**, *582*, 1283−1290.

(6) Bader, G. D.; Betel, D.; Hogue, C. W. V. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res.* **2003**, *31*, 248−250.

(7) Xenarios, I.; Salwanski, L.; Duan, X. J.; Higney, P.; Kim, S.-M.; Eisenberg, D. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **2002**, *30*, 303−305.

(8) Jensen, L. J.; Kuhn, M.; Stark, M.; Chaffron, S.; Creevey, C.; Muller, J.; Doerks, T.; Julien, P.; Roth, A.; Simonovic, M.; Bork, P.; von Mering, C. STRING 8- A global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.* **2009**, *37* (suppl 1), D412−D416.

(9) Keshava Prasad, T. S.; Goel, R.; Kandasamy, K.; Keerthikumar, S.; Kumar, S.; Mathivanan, S.; Telikicherla, D.; Raju, R.; Shafreen, B.; Venugopal, A.; Balakrishnan, L.; Marimuthu, A.; Banerjee, S.; Somanathan, D. S.; Sebastian, A.; Rani, S.; Ray, S.; Harrys Kishore, C. J.; Kanth, S.; Ahmed, M.; Kashyap, M. K.; Mohmood, R.; Ramachandra, Y. L.; Krishna, V.; Rahiman, B. A.; Mohan, S.; Ranganathan, P.; Ramabadran, S.; Chaerkady, R.; Pandey, A. Human Protein Reference Database—2009 update. *Nucleic Acids Res.* **2009**, *37* (suppl 1), D767−D772.

(10) Chen, X.; Ji, Z. L.; Chen, Y. Z. TTD: Therapeutic Target Database. *Nucleic Acids Res.* **2002**, *30*, 412−415.

(11) Wishart, D. S.; Knox, C.; Guo, A. C.; Cheng, D.; Shrivastava, S.; Tzur, D.; Gautam, B.; Hassanali, M. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res.* **2008**, *36* (suppl 1), D901−D906.

(12) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.

(13) Ogata, H.; Goto, S.; Sato, K.; Fujibuchi, W.; Bono, H.; Kanehisa, M. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **1999**, *27*, 29−34.

(14) Liu, T.; Lin, Y.; Wen, X.; Jorissen, R. N.; Gilson, M. K. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res.* **2007**, *35* (suppl 1), D198−D201.

(15) Gunther, S.; Kuhn, M.; Dunkel, M.; Campillos, M.; Senger, C.; Petsalaki, E.; Ahmed, J.; Urdiales, E. G.; Gewiess, A.; Jensen, L. J.; Schneider, R.; Skoblo, R.; Russell, R. B.; Bourne, P. E.; Bork, P.; Preissner, R. SuperTarget and Matador: resources for exploring drug-target relationships. *Nucleic Acids Res.* **2008**, *36* (suppl1), D919−D922.

(16) Bredel, M.; Jacoby, E. Chemogenomics: an emerging strategy for rapid target and drug discovery. *Nat. Rev. Genet.* **2004**, *5*, 262−275.

(17) Harris, C. J.; Stevens, A. P. Chemogenomics: structuring the drug discovery process to gene families. *Drug Discovery Today* **2006**, *11*, 880−888.

(18) Huang, J.-H.; Cao, D.-S.; Yan, J.; Xu, Q.-S.; Hu, Q.-N.; Liang, Y.-Z. Using core hydrophobicity to identify phosphorylation sites of human G protein-coupled receptors. *Biochimie* **2012**, *94*, 1697−1704.

(19) Chou, P. Y.; Fasman, G. D. Prediction of the Secondary Structure of Proteins from their Amino Acid Sequence. In *Advances in Enzymology Related Areas of Molecular Biology*; John Wiley & Sons, Inc.: Hoboken, NJ, 2006; pp 45−148.

(20) Shen, J.; Zhang, J.; Luo, X.; Zhu, W.; Yu, K.; Chen, K.; Li, Y.; Jiang, H. Predicting protein-protein interactions based only on sequences information. *Proc. Natl. Acad. Sci. U.S.A.* **2007**, *104*, 4337−4341.

(21) Chou, K.-C.; Shen, H.-B. Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms. *Nat. Protocols* **2008**, *3*, 153−162.

(22) Cao, D.-S.; Liu, S.; Xu, Q.-S.; Lu, H.-M.; Huang, J.-H.; Hu, Q.-N.; Liang, Y.-Z. Large-scale prediction of drug-target interactions using protein sequences and drug topological structures. *Anal. Chim. Acta* **2012**, *752*, 1−10.

(23) Yu, H.; Chen, J.; Xu, X.; Li, Y.; Zhao, H.; Fang, Y.; Li, X.; Zhou, W.; Wang, W.; Wang, Y. A Systematic Prediction of Multiple Drug-Target Interactions from Chemical, Genomic, and Pharmacological Data. *PLoS ONE* **2012**, *7*, e37608.

(24) He, Z.; Zhang, J.; Shi, X.-H.; Hu, L.-L.; Kong, X.; Cai, Y.-D.; Chou, K.-C. Predicting Drug-Target Interaction Networks Based on Functional Groups and Biological Features. *PLoS ONE* **2010**, *5*, e9603.

(25) Holland, R. C. G.; Down, T. A.; Pocock, M.; Prlić, A.; Huen, D.; James, K.; Foisy, S.; Dräger, A.; Yates, A.; Heuer, M.; Schreiber, M.

J. BioJava: an Open-Source Framework for Bioinforamtics. *Bioinformatics* **2008**, *24*, 2096−2097.

(26) Shen, H.-B.; Chou, K.-C. PseAAC: A flexible web server for generating various kinds of protein pseudo amino acid composition. *Anal. Biochem.* **2008**, *373*, 386−388.

(27) Li, Z. R.; Lin, H. H.; Han, L. Y.; Jiang, L.; Chen, X.; Chen, Y. Z. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res.* **2006**, *34* (Web Server issue), W32−37.

(28) Williams, C.; Labute, P.; Bajorath, J. Binary Quantitative Structure activity Relationship (QSAR) Analysis of Estrogen Receptor Ligands. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 164−168.

(29) Willett, P. Similarity-based virtual screening using 2D fingerprints. *Drug Discovery Today* **2006**, *11*, 1046−1053.

(30) Hoffman, B. T.; Kopajtic, T.; Katz, J. L.; Newman, A. H. 2D QSAR Modeling and Preliminary Database Searching for Dopamine Transporter Inhibitors Using Genetic Algorithm Variable Selection of Molconn Z Descriptors. *J. Med. Chem.* **2000**, *43*, 4151−4159.

(31) van de Waterbeemd, H.; Gifford, E. ADMET in silico modelling: towards prediction paradise? *Nat. Rev. Drug Discov.* **2003**, *2*, 192−204.

(32) Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493−500.

(33) O'Boyle, N.; Banck, M.; James, C.; Morley, C.; Vandermeersch, T.; Hutchison, G. Open Babel: An open chemical toolbox. *J. Cheminf.* **2011**, *3*, 1−14.

(34) O'Boyle, N.; Hutchison, G. Cinfony - combining Open Source cheminformatics toolkits behind a common interface. *Chem. Cent. J.* **2008**, *2*, 24.

(35) Mestres, J. Computational chemogenomics approaches to systematic knowledge-based drug discovery. *Curr. Opin. Drug Discovery* **2004**, *7*, 304−313.

(36) Kawashima, S.; Kanehisa, M. AAindex: Amino Acid index database. *Nucleic Acids Res.* **1999**, *27*, 368−369.

(37) Reczko, M.; Bohr, H. The DEF data base of sequence based protein fold class predictions. *Nucleic Acids Res.* **1994**, *22*, 3616−3619.

(38) Bhasin, M.; Raghava, G. P. S. Classification of Nuclear Receptors Based on Amino Acid Composition and Dipeptide Composition. *J. Biol. Chem.* **2004**, *279*, 23262−23266.

(39) Chou, K.-C.; Cai, Y.-D. Using Functional Domain Composition and Support Vector Machines for Prediction of Protein Subcellular Location. *J. Biol. Chem.* **2002**, *277*, 45765−45769.

(40) Feng, Z. P.; Zhang, C. T. Prediction of Membrane Protein Types Based on the Hydrophobic Index of Amino Acids. *J. Protein Chem.* **2000**, *19*, 269−275.

(41) Horne, D. S. Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. *Biopolymers* **1988**, *27*, 451−477.

(42) Lin, Z.; Pan, X. M. Accurate Prediction of Protein Secondary Structural Content. *J. Protein Chem.* **2001**, *20*, 217−220.

(43) Dubchak, I.; Muchnik, I.; Holbrook, S. R.; Kim, S. H. Prediction of protein folding class using global description of amino acid sequence. *Proc. Natl. Acad. Sci. U.S.A.* **1995**, *92*, 8700−8704.

(44) Bock, J. R.; Gough, D. A. Predicting protein-protein interactions from primary structure. *Bioinformatics* **2001**, *17*, 455−460.

(45) Cai, C. Z.; Han, L. Y.; Ji, Z. L.; Chen, X.; Chen, Y. Z. SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Res.* **2003**, *31*, 3692−3697.

(46) Chou, K.-C. Prediction of Protein Subcellular Locations by Incorporating Quasi-Sequence-Order Effect. *Biochem. Bioph. Res. Co.* **2000**, *278*, 477−483.

(47) Chou, K.-C.; Cai, Y.-D. Prediction of protein subcellular locations by GO-FunD-PseAA predictor. *Biochem. Bioph. Res. Co.* **2004**, *320*, 1236−1239.

(48) Chou, K.-C. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins* **2001**, *43*, 246−255.

(49) Chou, K.-C. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics* **2005**, *21*, 10−19.

(50) Cao, D.-S.; Xu, Q.-S.; Liang, Y.-Z.; Chen, X.; Li, H.-D. Prediction of aqueous solubility of druglike organic compounds using partial least squares, back-propagation network and support vector machine. *J. Chemom.* **2010**, *24*, 584−595.

(51) Xue, C. X.; Zhang, R. S.; Liu, M. C.; Hu, Z. D.; Fan, B. T. Study of the Quantitative Structure-Mobility Relationship of Carboxylic Acids in Capillary Electrophoresis Based on Support Vector Machines. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 950−957.

(52) Hou, T.; Wang, J.; Zhang, W.; Xu, X. ADME Evaluation in Drug Discovery. 7. Prediction of Oral Absorption by Correlation and Classification. *J. Chem Inf. Model.* **2007**, *47*, 208−218.

(53) Krovat, E. M.; Fruhwirth, K. H.; Langer, T. Pharmacophore Identification, in Silico Screening, and Virtual Library Design for Inhibitors of the Human Factor Xa. *J. Chem Inf. Model.* **2005**, *45*, 146−159.

(54) Gunturi, S. B.; Narayanan, R. In silico ADME modeling 3: Computational models to predict human intestinal absorption using sphere exclusion and kNN QSAR methods. *QSAR Comb. Sci.* **2007**, *26*, 653−668.

(55) Cao, D.-S.; Hu, Q.-N.; Xu, Q.-S.; Yang, Y.-N.; Zhao, J.-C.; Lu, H.-M.; Zhang, L.-X.; Liang, Y.-Z. In silico classification of human maximum recommended daily dose based on modified random forest and substructure fingerprint. *Anal. Chim. Acta* **2011**, *692*, 50−56.

(56) Shen, J.; Cheng, F.; Xu, Y.; Li, W.; Tang, Y. Estimation of ADME Properties with Substructure Pattern Recognition. *J. Chem. Inf. Model.* **2010**, *50*, 1034−1041.

(57) Cao, D.-S.; Yang, Y.-N.; Zhao, J.-C.; Yan, J.; Liu, S.; Hu, Q.-N.; Xu, Q.-S.; Liang, Y.-Z. Computer-aided prediction of toxicity with substructure pattern and random forest. *J. Chemom.* **2012**, *26*, 7−15.

(58) Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom pairs as molecular features in structure-activity studies: definition and applications. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 64−73.

(59) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 82−85.

(60) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(61) Yamanishi, Y.; Araki, M.; Gutteridge, A.; Honda, W.; Kanehisa, M. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* **2008**, *24*, i232−i240.

(62) Breiman, L. Random forests. *Machine Learn.* **2001**, *45*, 5−32.

(63) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947−1958.

(64) Yamanishi, Y. Supervised bipartite graph inference. *Proceedings of the Conference on Advances in Neural Information and Processing System*, Vancouver, British Columbia, Canada, December 8−11, 2008.

(65) Yamanishi, Y.; Kotera, M.; Kanehisa, M.; Goto, S. Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics* **2010**, *26*, i246−i254.

(66) Schuffenhauer, A.; Floersheim, P.; Acklin, P.; Jacoby, E. Similarity Metrics for Ligands Reflecting the Similarity of the Target Proteins. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 391−405.

(67) Campillos, M.; Kuhn, M.; Gavin, A. C.; Jensen, L. J.; Bork, P. Drug target identification using side-effect similarity. *Science* **2008**, *321*, 263−266.

(68) Keiser, M. J.; Roth, B. L.; Armbruster, B. N.; Ernsberger, P.; Irwin, J. J.; Shoichet, B. K. Relating protein pharmacology by ligand chemistry. *Nat. Biotechnol.* **2007**, *25*, 197−206.

(69) Keiser, M. J.; Setola, V.; Irwin, J. J.; Laggner, C.; Abbas, A. I.; Hufeisen, S. J.; Jensen, N. H.; Kuijer, M. B.; Matos, R. C.; Tran, T. B.; Whaley, R.; Glennon, R. A.; Hert, J.; Thomas, K. L. H.; Edwards, D. D.; Shoichet, B. K.; Roth, B. L. Predicting new molecular targets for known drugs. *Nature* **2009**, *462*, 175−181.

(70) Lounkine, E.; Keiser, M. J.; Whitebread, S.; Mikhailov, D.; Hamon, J.; Jenkins, J. L.; Lavan, P.; Weber, E.; Doak, A. K.; Cote, S.; Shoichet, B. K.; Urban, L. Large-scale prediction and testing of drug activity on side-effect targets. *Nature* **2012**, *486*, 361−367.

(71) Wikberg, J.; Lapinsh, M.; Prusis, P. Proteochemometrics: A tool for modelling the molecular interaction space. In *Chemogenomics in Drug Discovery—A Medicinal Chemistry Perspective*, first ed.; Kubinyi, H., Müller, G.; Wiley-VCH Verlag GmbH & Co. KGaA: New York, 2004; pp 289−309.

(72) Lapinsh, M.; Prusis, P.; Lundstedt, T.; Wikberg, J. E. S. Proteochemometrics Modeling of the Interaction of Amine G-Protein Coupled Receptors with a Diverse Set of Ligands. *Mol. Pharmacol.* **2002**, *61*, 1465−1475.

(73) van Westen, G. J. P.; Wegner, J. K.; Geluykens, P.; Kwanten, L.; Vereycken, I.; Peeters, A.; Ijzerman, A. P.; van Vlijmen, H. W. T.; Bender, A. Which Compound to Select in Lead Optimization? Prospectively Validated Proteochemometric Models Guide Preclinical Development. *PLoS ONE* **2011**, *6*, e27518.

(74) Lapinsh, M.; Prusis, P.; Uhlun, S.; Wikberg, J. E. S. Improved approach for proteochemometrics modeling: application to organic compound-amine G protein-coupled receptor interactions. *Bioinformatics* **2005**, *21*, 4289−4296.

(75) Lapins, M.; Eklund, M.; Spjuth, O.; Prusis, P.; Wikberg, J. Proteochemometric modeling of HIV protease susceptibility. *BMC Bioinformatics* **2008**, *9*, 181.

(76) Lapinsh, M.; Prusis, P.; Mutule, I.; Mutulis, I.; Wikberg, J. QSAR and proteochemometric analysis of the interaction of a series of organic compounds with melanocortin receptor subtypes. *J. Med. Chem.* **2003**, *46*, 2572−2579.

(77) Charton, M.; Charton, B. I. The structural dependence of amino acid hydrophobicity parameters. *J. Theor. Biol.* **1982**, *99*, 629−644.

(78) Bigelow, C. C. On the average hydrophobicity of proteins and the relation between it and protein structure. *J. Theor. Biol.* **1967**, *16*, 187−211.

(79) http://www.stat.berkeley.edu/~breiman/RandomForests/ (accessed Apr 12, 2012).

(80) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (6), 1947−1958.