Persistent Path-Spectral Based Machine Learning for Protein-Ligand Binding Affinity Prediction

Ran Liu

BUAA & BIMSA

Outline

Motivation

Persisitent Path Spectral(PPS)

Protein-Ligand Binding Affinity Prediction

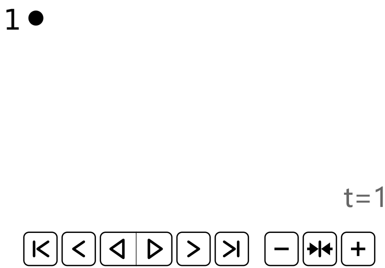
Motivation

Persisitent Path Spectral(PPS)

Protein-Ligand Binding Affinity Prediction

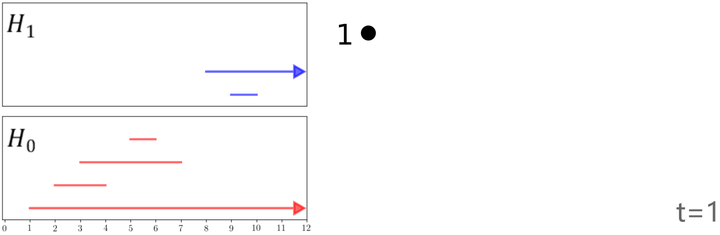
Motivation

- Persistent homology, a key theory for TDA, has been applied to numerous data science fields with many achievements. Its essence is to provide topological features to the data.



Motivation

- Persistent homology, a key theory for TDA, has been applied to numerous data science fields with many achievements. Its essence is to provide topological features to the data.

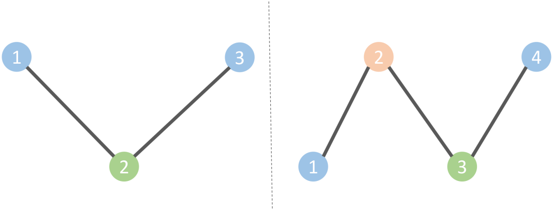


Motivation

- Recently, research Beyond TDA is being conducted.

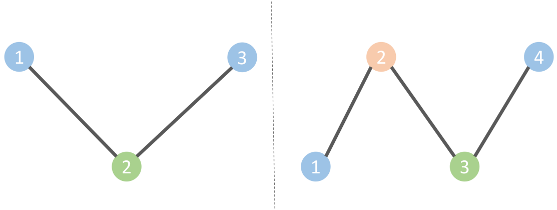
Motivation

- Recently, research Beyond TDA is being conducted.



Motivation

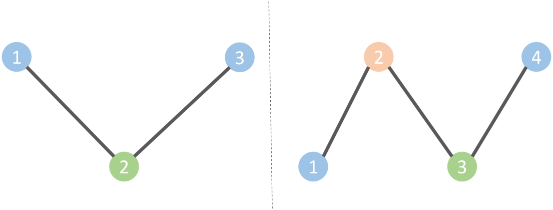
- Recently, research Beyond TDA is being conducted.



- The idea of hopping has been introducted on graph, and used to construct Laplacian matrix.

Motivation

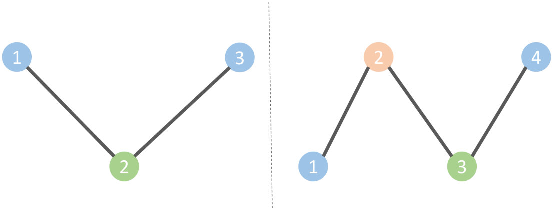
- Recently, research Beyond TDA is being conducted.



- The idea of hopping has been introducted on graph, and used to construct Laplacian matrix.
- We introduce the idea of hopping into the high-dimensional plate, combine it with the filtering process, consider specifically the Laplacian matrix, feed its spectral information into machine learning to obtain the Persistent Path Spectral(PPS) model, which can give a quantitative description of the data.

Motivation

- Recently, research Beyond TDA is being conducted.



- The idea of hopping has been introducted on graph, and used to construct Laplacian matrix.
- We introduce the idea of hopping into the high-dimensional plate, combine it with the filtering process, consider specifically the Laplacian matrix, feed its spectral information into machine learning to obtain the Persistent Path Spectral(PPS) model, which can give a quantitative description of the data.
- And test our model on issue of protein-ligand binding affinity prediction, PPS model can achieve competitive results.

Motivation

Persisitent Path Spectral(PPS)

Protein-Ligand Binding Affinity Prediction

Definition (Simplicial Complex)

An (abstract) simplicial complex C *is a pair* (V, C_V) *where* V *is a vertex set and* C *is a simplex set, such that every* $\sigma \in C_V$ *is a nonempty subset of vertex set , and every nonempty subset of* σ *is also* $\in C_V$*.*

Definition (Simplicial Complex)

An (abstract) simplicial complex C *is a pair* (V, C_V) *where* V *is a vertex set and* C *is a simplex set, such that every* $\sigma \in C_V$ *is a nonempty subset of vertex set , and every nonempty subset of* σ *is also* $\in C_V$.

Definition (n-simplex walk,path)

A series of n-simplices $\sigma_1^n, \sigma_2^n, ..., \sigma_l^n, \sigma_{l+1}^n$ *(not must diverse) is called an* ***n-simplex walk*** *from* σ_1^n *to* σ_{l+1}^n *while* σ_i^n *and* σ_{i+1}^n *share an (n+1)-simplex for each* $i = 1, 2, ..., l$.

Definition (Simplicial Complex)

An (abstract) simplicial complex C *is a pair* (V, C_V) *where* V *is a vertex set and* C *is a simplex set, such that every* $\sigma \in C_V$ *is a nonempty subset of vertex set , and every nonempty subset of* σ *is also* $\in C_V$.

Definition (n-simplex walk,path)

A series of n-simplices $\sigma_1^n, \sigma_2^n, ..., \sigma_l^n, \sigma_{l+1}^n$ *(not must diverse) is called an* ***n-simplex walk*** *from* σ_1^n *to* σ_{l+1}^n *while* σ_i^n *and* σ_{i+1}^n *share an (n+1)-simplex for each* $i = 1, 2, ..., l$.

Under another additional condition that these n-simplices are different from each other, this ***n-simplex walk*** *turns into an* ***n-simplex path****.*

Definition (shortest path)

Among all the n-simplex paths between σ_i^n and σ_j^n, the ones having the minimum number of (n+1)-simplexes are called ***the shortest n-simplex paths****(may more than one).*

Definition (shortest path)

Among all the n-simplex paths between σ_i^n and σ_j^n, the ones having the minimum number of (n+1)-simplexes are called ***the shortest n-simplex paths****(may more than one).*

Definition (path-distance)

The number of (n+1)-simplexes which a shortest n-simplex path between σ_i^n and σ_j^n passes is called the ***path-distance*** *between n-simplices σ_i^n and σ_j^n, and denoted by $d_{i,j}^n$.*

Definition (shortest path)

Among all the n-simplex paths between σ_i^n and σ_j^n, the ones having the minimum number of (n+1)-simplexes are called ***the shortest n-simplex paths****(may more than one).*

Definition (path-distance)

The number of (n+1)-simplexes which a shortest n-simplex path between σ_i^n and σ_j^n passes is called the ***path-distance*** *between n-simplices σ_i^n and σ_j^n, and denoted by $d_{i,j}^n$.*

Definition (k-hopping n-simplex walk,path)

An n-simplex walk $\sigma_1^n, \sigma_2^n, ..., \sigma_l^n, \sigma_{l+1}^n$ is called a ***k-hopping n-simplex walk*** *if the path-distance of σ_i^n and σ_{i+1}^n is k ($d_{i,i+1}^n = k$).*

Definition (shortest path)

*Among all the n-simplex paths between σ_i^n and σ_j^n, the ones having the minimum number of (n+1)-simplexes are called **the shortest n-simplex paths**(may more than one).*

Definition (path-distance)

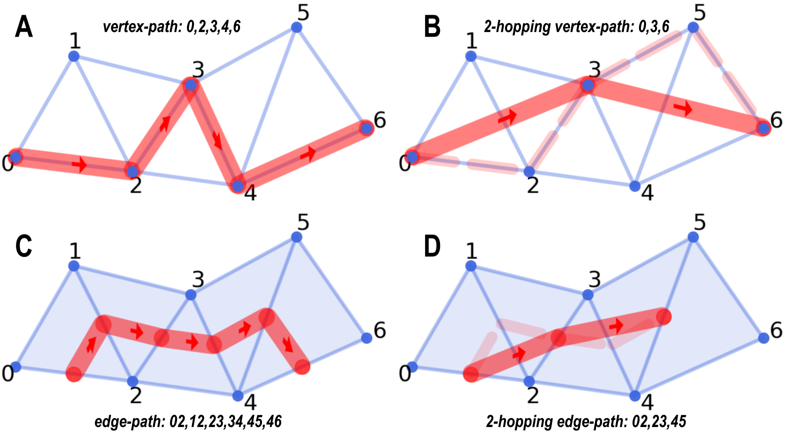
*The number of (n+1)-simplexes which a shortest n-simplex path between σ_i^n and σ_j^n passes is called the **path-distance** between n-simplices σ_i^n and σ_j^n, and denoted by $d_{i,j}^n$.*

Definition (k-hopping n-simplex walk,path)

*An n-simplex walk $\sigma_1^n, \sigma_2^n, ..., \sigma_l^n, \sigma_{l+1}^n$ is called a **k-hopping n-simplex walk** if the path-distance of σ_i^n and σ_{i+1}^n is k ($d_{i,i+1}^n = k$).*

*When these n-simplices are different from each other, this **k-hopping n-simplex walk** turns into a **k-hopping n-simplex path**.*

Example: hopping path of simplicial complex

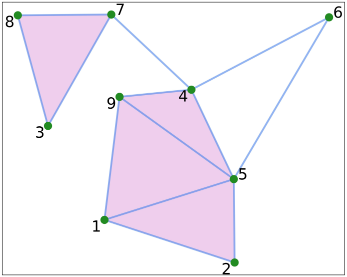

A
vertex-path: 0,2,3,4,6
B
2-hopping vertex-path: 0,3,6
C
D
edge-path: 02,12,23,34,45,46
2-hopping edge-path: 02,23,45
0
1
2
3
4
5
6

Definition (k-hopping n-simplex connected component)

Given a simplicial complex C, which can be represented by $\{C_n\}_{n\geq 0}$, here C_n is the collection of all n-simplices. For a subset of C_n, donated as X_n, if there is a k-hopping n-simplex walk visiting every n-simplices of X_n at lowest, the subset X_n is defined as a ***k-hopping n-simplex connected component*** *of C, which denoted by* ***(k,n) connected component*** *for simplicity.*

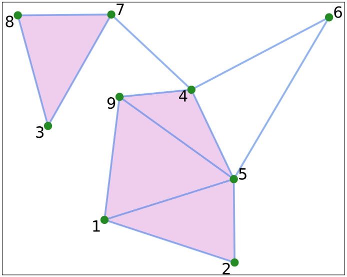
Definition (k-hopping n-simplex connected component)

Given a simplicial complex C, which can be represented by $\{C_n\}_{n\geq 0}$, here C_n is the collection of all n-simplices. For a subset of C_n, donated as X_n, if there is a k-hopping n-simplex walk visiting every n-simplices of X_n at lowest, the subset X_n is defined as a ***k-hopping n-simplex connected component*** *of C, which denoted by* ***(k,n) connected component*** *for simplicity.*



Definition (k-hopping n-simplex connected component)

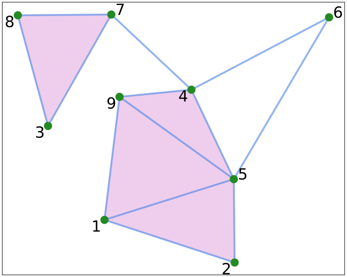
Given a simplicial complex C, which can be represented by $\{C_n\}_{n\geq 0}$, here C_n is the collection of all n-simplices. For a subset of C_n, donated as X_n, if there is a k-hopping n-simplex walk visiting every n-simplices of X_n at lowest, the subset X_n is defined as a ***k-hopping n-simplex connected component*** *of C, which denoted by* ***(k,n) connected component*** *for simplicity.*



- 2-hopping vertex walk: $\{v_1, v_6, v_7, v_5, v_7, v_9, v_2, v_4, v_8, v_4, v_3\}$
- 3-hopping vertex walk: $\{v_3, v_9, v_8, v_5, v_3, v_6\}$; $\{v_2, v_7, v_1\}$; $\{v_4\}$

Definition (k-hopping n-simplex connected component)

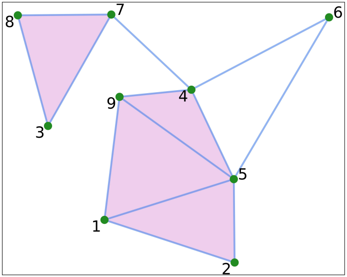
Given a simplicial complex C, which can be represented by $\{C_n\}_{n\geq 0}$, here C_n is the collection of all n-simplices. For a subset of C_n, donated as X_n, if there is a k-hopping n-simplex walk visiting every n-simplices of X_n at lowest, the subset X_n is defined as a ***k-hopping n-simplex connected component*** *of C, which denoted by* ***(k,n) connected component*** *for simplicity.*



- 2-hopping vertex walk: $\{v_1, v_6, v_7, v_5, v_7, v_9, v_2, v_4, v_8, v_4, v_3\}$
- 3-hopping vertex walk: $\{v_3, v_9, v_8, v_5, v_3, v_6\}$; $\{v_2, v_7, v_1\}$; $\{v_4\}$
- one (2,0) connected component, three (3,0) connected components

Definition (k-hopping n-simplex connected component)

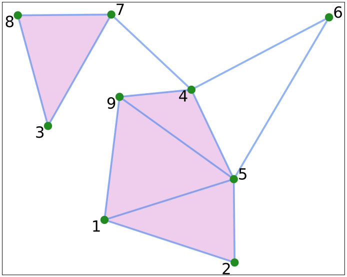
Given a simplicial complex C, which can be represented by $\{C_n\}_{n\geq 0}$, here C_n is the collection of all n-simplices. For a subset of C_n, donated as X_n, if there is a k-hopping n-simplex walk visiting every n-simplices of X_n at lowest, the subset X_n is defined as a ***k-hopping n-simplex connected component*** *of C, which denoted by* ***(k,n) connected component*** *for simplicity.*



- 2-hopping vertex walk: $\{v_1, v_6, v_7, v_5, v_7, v_9, v_2, v_4, v_8, v_4, v_3\}$
- 3-hopping vertex walk: $\{v_3, v_9, v_8, v_5, v_3, v_6\}$; $\{v_2, v_7, v_1\}$; $\{v_4\}$
- one (2,0) connected component, three (3,0) connected components
- 2-hopping edge walk: $\{[1,2], [5,9], [2,5], [1,9], [4,5], [1,5], [4,9]\}$
- 3-hopping edge walk: $\{[1,2], [4,5], [2,5], [4,9]\}$

Definition (k-hopping n-simplex connected component)

Given a simplicial complex C, which can be represented by $\{C_n\}_{n\geq 0}$, here C_n is the collection of all n-simplices. For a subset of C_n, donated as X_n, if there is a k-hopping n-simplex walk visiting every n-simplices of X_n at lowest, the subset X_n is defined as a ***k-hopping n-simplex connected component*** *of C, which denoted by* ***(k,n) connected component*** *for simplicity.*



- 2-hopping vertex walk: $\{v_1, v_6, v_7, v_5, v_7, v_9, v_2, v_4, v_8, v_4, v_3\}$
- 3-hopping vertex walk: $\{v_3, v_9, v_8, v_5, v_3, v_6\}$; $\{v_2, v_7, v_1\}$; $\{v_4\}$
- one (2,0) connected component, three (3,0) connected components
- 2-hopping edge walk: $\{[1,2], [5,9], [2,5], [1,9], [4,5], [1,5], [4,9]\}$
- 3-hopping edge walk: $\{[1,2], [4,5], [2,5], [4,9]\}$
- seven (2,1) connected components, ten (3,1) connected components

Definition (k-path degree)

k-path degree *$\delta_k(\sigma_i^n)$ of σ_i^n is defined as the count of n-simplices σ_j^n such that the path-distance between σ_i^n and σ_j^n is k.*

Definition (k-path degree)

k-path degree *$\delta_k(\sigma_i^n)$ of σ_i^n is defined as the count of n-simplices σ_j^n such that the path-distance between σ_i^n and σ_j^n is k.*

Definition ((k,n) path-Laplacian)

The ***k-path n-simplex Laplacian matrix*** *L_k^n of simplicial complex C is a $N(C_n)$ order square symmetric matrix whose entries is shown as follows, denoted by* ***(k,n) path-Laplacian****.*

$$L_k^n(C)(i,j) = \begin{cases} \delta_k(\sigma_i^n) & , \; i = j \\ -1 & , \; d_{i,j}^n = k \\ 0 & , \; otherwise \end{cases} \tag{1}$$

Definition (k-path degree)

k-path degree *$\delta_k(\sigma_i^n)$ of σ_i^n is defined as the count of n-simplices σ_j^n such that the path-distance between σ_i^n and σ_j^n is k.*

Definition ((k,n) path-Laplacian)

The ***k-path n-simplex Laplacian matrix*** *L_k^n of simplicial complex C is a $N(C_n)$ order square symmetric matrix whose entries is shown as follows, denoted by* ***(k,n) path-Laplacian****.*

$$L_k^n(C)(i,j) = \begin{cases} \delta_k(\sigma_i^n) & , \ i = j \\ -1 & , \ d_{i,j}^n = k \\ 0 & , \ otherwise \end{cases} \tag{1}$$

Theorem

The number of ***k-hopping n-simplex connected components*** *is the multiplicty of zero eigenvalue of* ***(k,n) path-Laplacian****.*

Example: Laplacian matrix of C_{20}

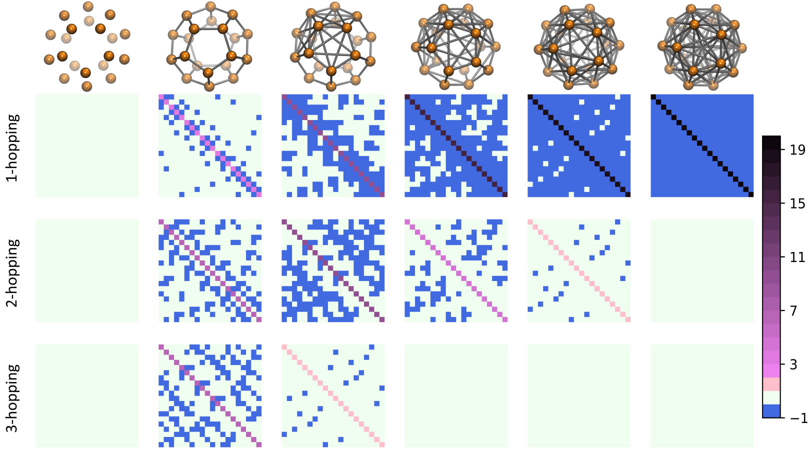
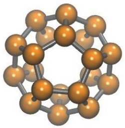
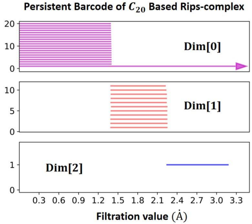
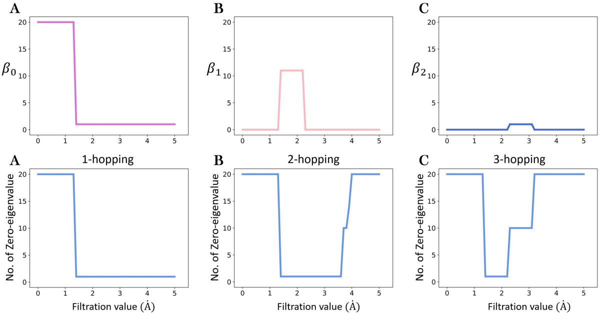


Figure: Vertex(0-simplex) path-Laplacian matrices with filtration values 1.0Å, 1.5Å, 2.3Å, 3.3Å, 3.7Å, 4.0Å.

Example: Persisitent feature of C_{20}







Definition (path spectral)

The eigenvalues and eigenvectors of the (k,n) path-Laplacian matrix is called the ***(k,n) path-spectral*** *of the simplicial complex.*

Definition (path spectral)

The eigenvalues and eigenvectors of the (k,n) path-Laplacian matrix is called the ***(k,n) path-spectral*** *of the simplicial complex.*

Assume we have a filtration of simplicial complexes, which is a sequence of nested simplicial complexes

$$O_1 \subset O_2 \subset ... \subset O_t$$

where O_i is a sub-complex of $O_{i+1}(0 < i < t)$. For each O_i, we consider its (k,n) path-Laplacian matrix $L_k^n(O_i)$, then we get a sequence of path-Laplacian matrixes for each pair (k,n)

$$L_k^n(O_1), L_k^n(O_2), ..., L_k^n(O_t)$$

Definition (path spectral)

*The eigenvalues and eigenvectors of the (k,n) path-Laplacian matrix is called the **(k,n) path-spectral** of the simplicial complex.*

Assume we have a filtration of simplicial complexes, which is a sequence of nested simplicial complexes

$$O_1 \subset O_2 \subset ... \subset O_t$$

where O_i is a sub-complex of $O_{i+1}(0 < i < t)$. For each O_i, we consider its (k, n) path-Laplacian matrix $L_k^n(O_i)$, then we get a sequence of path-Laplacian matrixes for each pair (k, n)

$$L_k^n(O_1), L_k^n(O_2), ..., L_k^n(O_t)$$

Definition (persistent path spectral)

*The persistence and variance of the path-spectral information through the sequence of path-Laplacian matrixes is called the **persistent path-spectral** of the sequence of simplicial complexes.*

Example: Persisitent path-spectral of C_{20}

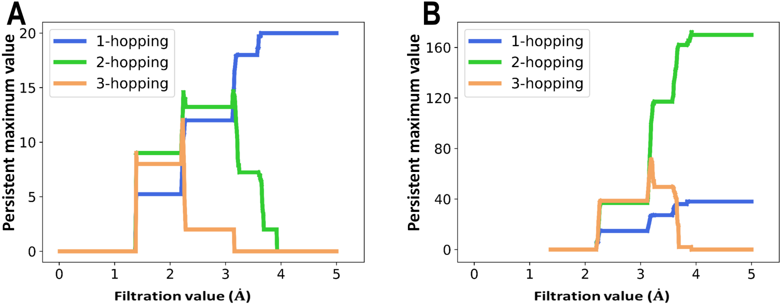


Figure: Persistent attribute curves from persistent path-spectral for C_{20}. Left is based on vertex, right is based on edge.

Motivation

Persisitent Path Spectral(PPS)

Protein-Ligand Binding Affinity Prediction

Protein-Ligand Complex and Affinity

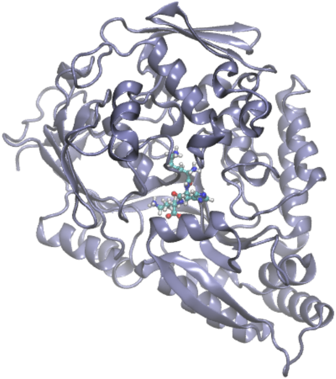


Figure: Protein-ligand complex ID: 1b3f.

- A protein-ligand complex is a complex of a protein bound with a ligand that is formed following molecular recognition between proteins that interact with each other or with various other molecules.

Protein-Ligand Complex and Affinity

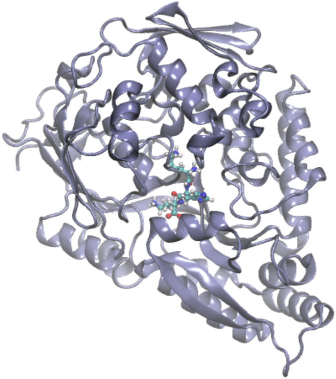


Figure: Protein-ligand complex ID: 1b3f.

- A protein-ligand complex is a complex of a protein bound with a ligand that is formed following molecular recognition between proteins that interact with each other or with various other molecules.
- The highest possible affinity from a protein towards the ligand, or target molecule, can be observed when the protein has a perfect mirror image of the shape of the target surface together with a charge distribution that complements perfectly the target surface.

Protein-ligand binding affinity prediction

- The task of predicting the interactions between compounds and proteins is the core and foundation of drug discovery, which consists of protein-ligand interaction, protein-ligand binding affinity, protein-ligand interaction sites and ligand bioactivity on proteins.

Protein-ligand binding affinity prediction

- The task of predicting the interactions between compounds and proteins is the core and foundation of drug discovery, which consists of protein-ligand interaction, protein-ligand binding affinity, protein-ligand interaction sites and ligand bioactivity on proteins.
- Protein-ligand interaction, also known as compound-protein interaction, is most reliably determined by in vitro experiments or biochips.

Protein-ligand binding affinity prediction

- The task of predicting the interactions between compounds and proteins is the core and foundation of drug discovery, which consists of protein-ligand interaction, protein-ligand binding affinity, protein-ligand interaction sites and ligand bioactivity on proteins.
- Protein-ligand interaction, also known as compound-protein interaction, is most reliably determined by in vitro experiments or biochips.
- However, this is extremely costly in the first screening of a compound, which requires a prohibitively enormous search space.

Protein-ligand binding affinity prediction

- The task of predicting the interactions between compounds and proteins is the core and foundation of drug discovery, which consists of protein-ligand interaction, protein-ligand binding affinity, protein-ligand interaction sites and ligand bioactivity on proteins.
- Protein-ligand interaction, also known as compound-protein interaction, is most reliably determined by in vitro experiments or biochips.
- However, this is extremely costly in the first screening of a compound, which requires a prohibitively enormous search space.
- To narrow the search space, there is an urgent need to develop more efficient computational approaches.

Computational approaches

The computational approaches for protein-ligand binding affinity pediction are usually called scoring functions (SFs), which can be generally divided into two groups.

Computational approaches

The computational approaches for protein-ligand binding affinity pediction are usually called scoring functions (SFs), which can be generally divided into two groups.

1. One is the classical methods which usually use linear functions to model the relationship between experimental data and features. Classical methods can be divided into three groups:
   - Physics-based(force-field based) methods
   - Empirical (regression-based) methods
   - Knowledge-based methods

Computational approaches

The computational approaches for protein-ligand binding affinity pediction are usually called scoring functions (SFs), which can be generally divided into two groups.

1. One is the classical methods which usually use linear functions to model the relationship between experimental data and features. Classical methods can be divided into three groups:
   - Physics-based(force-field based) methods
   - Empirical (regression-based) methods
   - Knowledge-based methods
2. The other is artificial intelligence (AI) based methods which can capture nonlinear relationship between features and experimental data. AI based models can be grouped into two categories:
   - Machine learning models
   - Deep learning models

Representation of Protein-Ligand Complex

- For the topological representation of a protein-ligand complex, its binding core region is extracted and characterized by element-specific representation.

Representation of Protein-Ligand Complex

- For the topological representation of a protein-ligand complex, its binding core region is extracted and characterized by element-specific representation.
- Euclidean distance and electrostatic distance functions are used to form the filtration of the representation.

Representation of Protein-Ligand Complex

- For the topological representation of a protein-ligand complex, its binding core region is extracted and characterized by element-specific representation.
- Euclidean distance and electrostatic distance functions are used to form the filtration of the representation.
- For each protein-ligand complex, 36 atom combinations are generated with protein atoms C, N, O, S and ligand atoms C, N, O, S, P, F, Cl, Br, I. And a filtered bipartite graph is constructed from every atom-combination where the distance is used as the filtration value for each edge.

Representation of Protein-Ligand Complex

- For the topological representation of a protein-ligand complex, its binding core region is extracted and characterized by element-specific representation.
- Euclidean distance and electrostatic distance functions are used to form the filtration of the representation.
- For each protein-ligand complex, 36 atom combinations are generated with protein atoms C, N, O, S and ligand atoms C, N, O, S, P, F, Cl, Br, I. And a filtered bipartite graph is constructed from every atom-combination where the distance is used as the filtration value for each edge.
- For electrostatic interactions, H atoms are also taken into consideration and a total of 50 atom combinations are generated from electrostatic interactions.

Featurization

- For the topological representation of a protein-ligand complex, use PPS to obtain feature, which can be combined with machine learning model.

Featurization

- For the topological representation of a protein-ligand complex, use PPS to obtain feature, which can be combined with machine learning model.
- For the distance-based PPS model, the filtration goes from 0Å to 10 Å with a step of 0.1 Å, and for the electrostatic-based PPS model, the filtration goes from 0 to 1 with a step of 0.02.

Featurization

- For the topological representation of a protein-ligand complex, use PPS to obtain feature, which can be combined with machine learning model.
- For the distance-based PPS model, the filtration goes from 0Å to 10 Å with a step of 0.1 Å, and for the electrostatic-based PPS model, the filtration goes from 0 to 1 with a step of 0.02.
- We use persistent median value curve and persistent mean value curve of the persistent spectral with hopping 1, 2 and 3 as the features.

Featurization

- For the topological representation of a protein-ligand complex, use PPS to obtain feature, which can be combined with machine learning model.
- For the distance-based PPS model, the filtration goes from 0Å to 10 Å with a step of 0.1 Å, and for the electrostatic-based PPS model, the filtration goes from 0 to 1 with a step of 0.02.
- We use persistent median value curve and persistent mean value curve of the persistent spectral with hopping 1, 2 and 3 as the features.
- The size of features based distance-model is $\mathbf{21600 = 36(atom\text{-}combinations) \times 100(persistence) \times 3(hopping) \times 2}$, the size of features based electrostatic function is $\mathbf{15000 = 50(atom\text{-}combinations) \times 50(persistence) \times 3(hopping) \times 2}$. Combined model's feature size is $\mathbf{36600 = 21600 + 15000}$.

Machine Learning

- As one of the most powerful algorithms in supervised learning, the gradient boosting tree (GBT) algorithm is a machine learning algorithm that combines decision tree and ensemble learning.

Machine Learning

- As one of the most powerful algorithms in supervised learning, the gradient boosting tree (GBT) algorithm is a machine learning algorithm that combines decision tree and ensemble learning.
- GBT is widely used and highly inclusive of feature inputs, and can achieve very robustness and generalization with stable loss functions.

Machine Learning

- As one of the most powerful algorithms in supervised learning, the gradient boosting tree (GBT) algorithm is a machine learning algorithm that combines decision tree and ensemble learning.
- GBT is widely used and highly inclusive of feature inputs, and can achieve very robustness and generalization with stable loss functions.
- The molecular descriptors obtained by PPS are used as feature inputs to GBT to obtain a machine learning model based on PPS, which denoted by PPS-ML.

Machine Learning

- As one of the most powerful algorithms in supervised learning, the gradient boosting tree (GBT) algorithm is a machine learning algorithm that combines decision tree and ensemble learning.
- GBT is widely used and highly inclusive of feature inputs, and can achieve very robustness and generalization with stable loss functions.
- The molecular descriptors obtained by PPS are used as feature inputs to GBT to obtain a machine learning model based on PPS, which denoted by PPS-ML.

No. of estimators	**Maximum features**	**Learning rate**	**Loss function**
40000	Square root	0.001	Least square
Minimum sample split	**Subsample size**	**Maximum depth**	**Repetition**
3	0.7	6	10

Table: Detailed parameters of GBT

Datasets

- The PDBbind database is a collection of the experimentally measured binding affinities exclusively for the protein-ligand complexes available in the Protein Data Bank(PDB).
- This type of knowledge is the much needed basis for many computational and statistical studies on molecular recognition.

Dataset	**Refined set**	**Training set**	**Test set (Core set)**
PBDbind-v2007	1300	1105	195
PDBbind-v2013	2959	2764	195
PDBbind-v2016	4057	3772	285

Table: Detailed information of the three PDBbind datasets, i.e., PDBbind-v2007, PDBbind-v2013, PDBbind-v2016.

Result

PDBbind-v2016	Dist	Charg	Dist+Charg
1-hopping	0.793(1.393)	0.808(1.359)	0.823(1.322)
2-hopping	0.792(1.392)	0.798(1.374)	0.810(1.347)
3-hopping	0.781(1.436)	0.800(1.375)	0.811(1.354)
(1,2,3)-hopping	0.829(1.287)	0.832(1.269)	0.843(1.248)

PDBbind-v2013	Dist	Charg	Dist+Charg
1-hopping	0.746(1.561)	0.760(1.534)	0.775(1.503)
2-hopping	0.753(1.535)	0.759(1.518)	0.767(1.497)
3-hopping	0.733(1.584)	0.725(1.606)	0.745(1.560)
(1,2,3)-hopping	0.778(1.478)	0.778(1.473)	0.791(1.444)

PDBbind-v2007	Dist	Charg	Dist+Charg
1-hopping	0.791(1.534)	0.800(1.509)	0.804(1.509)
2-hopping	0.793(1.500)	0.766(1.559)	0.791(1.497)
3-hopping	0.781(1.540)	0.776(1.547)	0.799(1.499)
(1,2,3)-hopping	0.818(1.142)	0.827(1.399)	0.827(1.399)

Table: PCCs and RMSEs of PPS-ML models on three datasets.

Result

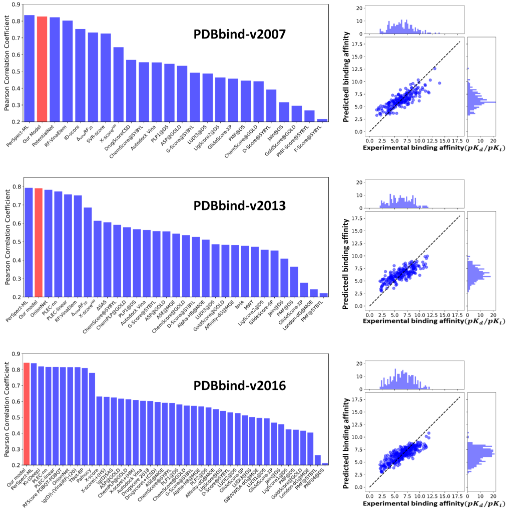


Figure: Performance of PPS-ML model on three datasets.

Result

- In our model, the feature size is 36600, which is much larger than the data size of three PDBbind datasets we used.
- We expanded the parameter step by 5 times for feature generation to do regression to alleviate overfitting problem.

Dataset	**Original size(36600)**	**Adjusted size(7320)**
PDBbind-v2016	0.843(1.248)	0.839(1.257)
PDBbind-v2013	0.791(1.444)	0.790(1.447)
PBDbind-v2007	0.827(1.399)	0.830(1.390)

Table: PCCs and RMSEs of PPS-ML model on three datasets based on different feature size.

Thank You!