A DATABASE USED FOR AUTOMATION OF MOLECULAR PROPERTIES USING MACHINE DEEP-LEARNING APPROACH

MERJEM HOXHA[1], BLERTA RAHMANI[2], HIQMET KAMBERAJ[3]

[1]Department of Computer Engineering, Faculty of Engineering, International Balkan University, Skopje, Republic of North Macedonia

[2]Department of Physics, Faculty of Mathematics and Natural Sciences, State University of Tetova, Tetova, Republic of North Macedonia

[3]Advanced Computing Research Center, University of New York Tirana, Tirana, Albania

e-mail: h.kamberaj@gmail.com

Abstract

To provide accurate predictions of different thermodynamic properties of the (bio)molecular systems, such as free energy of hydration, pKa, binding energy, or quantum mechanical properties, is challenging and computationally time-consuming. Besides, the consistency among all other molecular systems already optimized remains still challenging and perhaps tricky to automate. Very recently, machine learning approaches are introduced to automate and predict different properties. Although these methods are very accurate to interpolate among the training dataset, they suffer for not being able to extrapolate outside the training dataset. Therefore, in practice, the increase of diversity among the molecules included in the training dataset is required to guarantee good predictions. On the other hand, that would require building up extensive and quickly to manage databases of (bio)molecular systems with known properties either experimentally or by quantum mechanical calculations. This study aims to introduce a database management system of a large dataset that can be used to automate the prediction of (bio)molecular properties using a new machine learning approach. Also, we aim to discuss the importance of the choice of the chemical space of the training dataset, so that the experience gained by the training to incorporate all the laws of physics adequately.

Key words: Machine Learning, feature descriptors, database, molecular properties, artificial neural networks, bootstrapping.

Introduction

Recently, the neural network method has seen a broad range of applications in molecular modeling. (Mater & Coote, 2019) In Ref. (Lubbers, *et al.*, 2018), a hierarchical interacting particle neural network approach is introduced using quantum models to predict molecular properties. In this approach, different hierarchical regularization terms have been added to improve the convergence of the optimized parameters. While in Ref. (Gastegger, *et al.*, 2018), machine learning like-potentials are used to predict molecular properties, such as enthalpies or potential energies. The degree to which the general features included in characterizing the chemical space of molecules to improve the predictions of these models is also discussed in

Refs. (Goh, *et al*., 2018; Collins, *et al*., 2018). Tuckerman and co-workers (Schneider, *et al*., 2017) used a stochastic neural network technique to fit high-dimensional free energy surfaces characterized by the reduced subspace of collective coordinates. While very recently (Kamath, *et al*., 2018), a comparison study has been performed between the neural network approach and Gaussian process regression to fit the potential energy surfaces. One of the recognized problems in using machine learning approaches in prediction free energy surfaces is the inaccurate representation of general features of the surface topology by the training data. To improve on this, a combination of metadynamics molecular dynamics with neural network chemical models has also proposed (Herr, *et al*., 2018). It is worth noting that in the prediction of free energy surfaces, an accurate representation of the reduced subspace can be crucial. For that, Wehmeyer & Noè (Wehmeyer & Noe, 2018) have used the time-lagged auto-encoder to determine essential degrees of freedom of dynamical data.

Machine learning approaches have also been in the field of drug-design, for instance, in predicting drug-target interactions (Chen, *et al*., 2018), and it is a promising approach. In particular, the method is used in combination with molecular dynamics to predict the ligand-binding mechanism to purine nucleoside phosphorylase (Decherchi, *et al.,* 2015), and it accurately identifies the mechanism of drug-target binding modes.

In this study, we employed a novel method for prediction of (macro)molecular properties using a swarm artificial neural network method as a machine learning approach. In this method, a (macro)molecular structure is represented by a so-called *description vector*, which then is used as input in a so-called *bootstrapping swarm artificial neural network* (BSANN) for training the neural network. (Kamberaj, 2020) We aim to develop an efficient approach for performing the training of an artificial neural network using either experimental or quantum mechanics data. In particular, we created different user-friendly online accessible databases of well-selected experimental (or quantum mechanics) results that can be used as proof of the concepts. Furthermore, with the optimized artificial neural network using the training data served as input for BSANN, we can predict properties and their statistical errors of new molecules using the plugins provided from that web-service.

There are four databases accessible using the web-based service. A database of 642 small organic molecules with known experimental hydration free energies (Mobley, 2013) is presented, which is well-studied in Ref. (Riquelme, *et al.*, 2018); the database of 1475 experimental pKa values of ionizable groups in 192 proteins (including 153 wild-type proteins and 39 mutant proteins) (Thurlkill, *et al*., 2006; Pace, *et al*., 2009; Click & Kaminski, 2009; Pahari, *et al*., 2019); the database of 2693 mutants in 14 proteins with given values of experimental values of changes in the Gibbs free energy (Gromiha, *et al*., 1999; Bava, *et al*., 2004); and a database of

7101 quantum mechanics heat of formation calculations with the Perdew-Burke-Ernzerhof hybrid functional (PBE0). (Rupp, et al., 2012; Collins, et al., 2018)

All the data are prepared and optimized in advance using the AMBER force field (Wang, et al., 2004) in CHARMM macromolecular computer simulation program. (Brooks, et al., 2009) The BSANN is the code for performing the optimization and prediction written in Python computer programming language. The descriptor vectors of the small molecules are based on the Coulomb matrix and the sum over bond properties, and for the macromolecular systems, they take into account the chemical-physical fingerprints of the region in the vicinity of each amino acid. Note that the application of the BSANN algorithm for the datasets introduced in this study shall be published later in (Rahmani & Kamberaj, 2020).

Materials and methods

Artificial Neural Network

Machine Learning (ML) approach provides a potential method to predict the properties of a system using decision-making algorithms, based on some predefined features characterizing these properties of the system. There exist different ML methods used to predict missing data or discover new patterns during the data mining process. (McCulloch & Pitts, 1943) The neural networks method considers a large training dataset, and then it tries to construct a system, which is made up of rules for recognizing the patterns within the training data set by a learning process. In general, for an ANN with K hidden layers (see also Figure 1), the output Y_i is defined as

$$Y_i = f\left(\underbrace{\sum_{l_K=1}^{L_K} f\left(\underbrace{\sum_{l_{K-1}=1}^{L_{K-1}} f\left(\underbrace{\cdots f\left(\underbrace{\sum_{l_2=1}^{L_2} f\left(\underbrace{\sum_{l_1=1}^{L_1} f\left(\underbrace{\sum_{j=1}^{n} X_j W_{jl_1} + b_{l_1}}_{Input\ Layer}\right) W_{l_1 l_2} + b_{l_2}}_{1st\ hidden\ layer}\right) \,..}_{2nd\ hidden\ layer}\right) \cdots}_{\cdots}\right) W_{l_{K-1} l_K} + b_{l_K}}_{(K-1)th\ hidden\ layer}\right) W_{l_K i} + b_i}_{Kth\ hidden\ layer}\right) \tag{1}$$

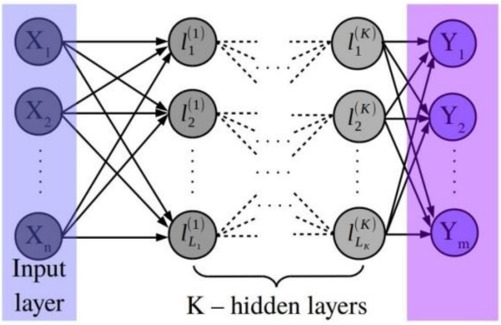

Figure 1. Illustration diagram of an artificial neural network (ANN). It is characterized by an input vector of dimension n, K hidden layers of neurons each, and an output vector of dimension m, as published in Reference. (Kamberaj, 2020; Rahmani & Kamberaj, 2020)

Here, $\boldsymbol{W}$ and $\boldsymbol{b}$ are considered as free parameters, which need to be optimized for a given training data used as inputs and given outputs, which are known. To optimize these parameters, the so-called loss function is minimized using the Gradient Descent method (Qian, 1999):

$$S(\boldsymbol{W},\boldsymbol{b}) = \sum_{i=1}^{m} (Y_i^0 - Y_i)^2 \qquad (2)$$

where $\boldsymbol{Y}^0$ represents the actual output vector. For that, the gradients of $S(\boldsymbol{W},\boldsymbol{b})$ with respect to $\boldsymbol{W}$ and $\boldsymbol{b}$ are calculated (Qian, 1999):

$$\Delta\boldsymbol{W} = -\left(\frac{\partial S(\boldsymbol{W},\boldsymbol{b})}{\partial \boldsymbol{W}}\right)_{\boldsymbol{b}}$$
$$\Delta\boldsymbol{b} = -\left(\frac{\partial S(\boldsymbol{W},\boldsymbol{b})}{\partial \boldsymbol{b}}\right)_{\boldsymbol{W}} \qquad (3)$$

To avoid over-fitting of ANN, the following regularization terms have been introduced in literature: (Srivastava, et al., 2014)

$$\Delta\boldsymbol{W}' = \gamma_w(\Delta\boldsymbol{W} + \gamma_1\boldsymbol{W})$$
$$\Delta\boldsymbol{b}' = \gamma_w(\Delta\boldsymbol{b} + \gamma_1\boldsymbol{b}) \qquad (4)$$

where γ_w is called learning rate for the gradient and γ_1 is called the regulation strength.

Usually, the Gradient Descent method often converges to a local minimum, and hence it provides a local optimization to the problem. To avoid that, a new Bootstrapping Swarm Artificial Neural Network method is proposed in the literature (Kamberaj, 2020), which is introduced in the following. The applications of the BSANN algorithm for the current databases shall be shown in Ref. (Rahmani & Kamberaj, 2020).

Bootstrapping Swarm Artificial Neural Network

In the following, we are presenting the BSANN algorithm, which shall be shown in Ref. (Rahmani & Kamberaj, 2020).

The standard ANN method deals with random numbers, which are used to initialize the parameters $\boldsymbol{W}$ and $\boldsymbol{b}$; therefore, the optimal solution of the problem will be different for different runs. In particular, we can say that there exists an uncertainty in the calculation of the optimal solution (i.e., in determining $\boldsymbol{W}$ and $\boldsymbol{b}$.) To calculate these uncertainties in the estimation of the optimal parameters, $\boldsymbol{W}$ and $\boldsymbol{b}$, we introduce a new approach, namely bootstrapping artificial neural network based on the method proposed by Gerhard Paass (Paass, 1993), or similar methods (Zhou, et al., 2019). In this approach, M copies of the same neural network are run independently using different input vectors. Here, we implement that at regular intervals, to swap optimal parameters (i.e., $\boldsymbol{W}$ and $\boldsymbol{b}$) between the two neighboring neural networks, which is equivalent to increasing the dimensionality of the problem by one; that is, if the dimensionality in each of the replicas is $d = K \times L$, then the dimensionality of the bootstrapping artificial neural network based on the method is $d + 1$. Figure 2 shows the layout of this configuration.

Furthermore, to achieve a good sampling of the phase space extended by the vectors $\boldsymbol{W}$ and $\boldsymbol{b}$, we introduce two other regularization terms similar to the swarm-particle sampling approach. First, we define two vectors for each neural network, namely $\boldsymbol{W}_n^{Lbest}$ and $\boldsymbol{b}_n^{Lbest}$, which represent the best local optimal parameters for each neural network n. In addition, we also define $\boldsymbol{W}^{Gbest}$ and $\boldsymbol{b}^{Gbest}$, which represent the global best optimal parameters among all neural networks.

Then, the expressions for the gradients are modified by introducing these two regularization terms as the following:

$$\Delta\boldsymbol{W}_n'' = \gamma_w\big(\Delta\boldsymbol{W}_n + \gamma_1\boldsymbol{W}_n - \gamma_2 U(0,1)(\boldsymbol{W}_n - \boldsymbol{W}_n^{Lbest}) - \gamma_3 U(0,1)(\boldsymbol{W}_n - \boldsymbol{W}^{Gbest})\big)$$
$$\Delta\boldsymbol{b}'' = \gamma_w\big(\Delta\boldsymbol{b}_n + \gamma_1\boldsymbol{b}_n - \gamma_2 U(0,1)(\boldsymbol{b}_n - \boldsymbol{b}_n^{Lbest}) - \gamma_3 U(0,1)(\boldsymbol{b}_n - \boldsymbol{b}^{Gbest})\big)$$

for each neural network configuration n, $n = 1, 2, \cdots, M$. Here, $U(0,1)$ is a random number between zero and one, and γ_2 and γ_3 represent the strength of biases toward the local best optimal parameters and global best optimal parameters, respectively. The first term indicates the individual knowledge of each neural network, and the second bias term, the collective expertise among the neural networks. This method is called Bootstrapping Swarm Artificial Neural Network. (Kamberaj, 2020), and it shall be applied to the current databases in Ref. (Rahmani & Kamberaj, 2020). Then, the weights, $\boldsymbol{W}_n$, and biases, $\boldsymbol{b}_n$, for each neural network n are updated at each iteration step according to:

$$\boldsymbol{W}_n^{new} = \boldsymbol{W}_n^{old} + \Delta\boldsymbol{W}_n''$$
$$\boldsymbol{b}_n^{new} = \boldsymbol{b}_n^{old} + \Delta\boldsymbol{b}_n''$$

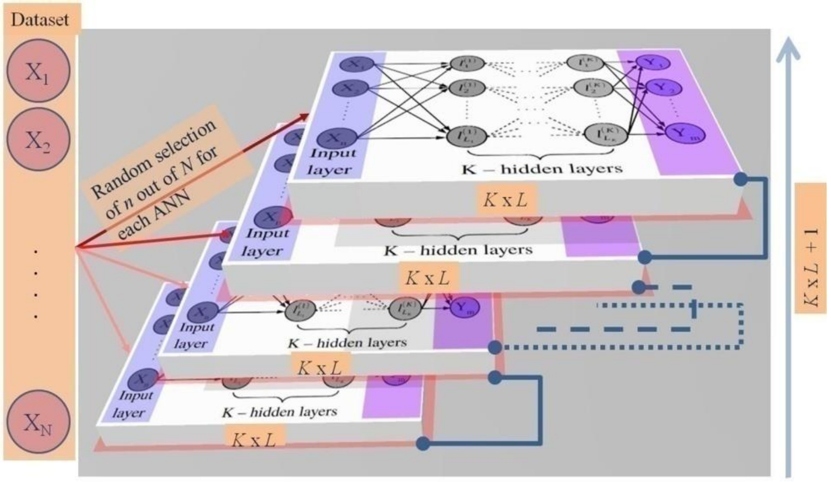

Figure 2. The layout of the Bootstrapping Swarm Artificial Neural Network (BSANN) as adopted by (Kamberaj, 2020). It is characterized by M different input vectors each of dimension n, K hidden layers of $l_{L_1}^{(1)}, l_{L_2}^{(2)}, \cdots, l_{L_K}^{(K)}$ neurons each, and M different output vectors each of dimension m. Every two neighboring neural networks communicate regularly with each other by swapping the optimized parameters. A test of the BSANN algorithm shall be published elsewhere (Rahmani & Kamberaj, 2020).

(Macro) molecular Feature Description

To construct data-driven models, such as in the ML approach, we will need to specify a list of physical and chemical properties of the input (macro)molecule that contain necessary information about the system. Here, the input data will be presented by a vector of length N, called $\boldsymbol{X}$. That process is called *feature description*, and the input data are called *feature descriptors*. We are describing in detail the algorithm for determining the feature descriptor vector of molecules in the following; the results of the applications to the databases introduced in this study shall be published in (Rahmani & Kamberaj, 2020).

Often, a so-called *Simplified Molecular Input Line Entry System* (SMILES) is used to represent a small molecule as a string of letters (Weininger, 1988). In such a case, the atoms could be encoded by a single integer number, such as H=1, C=2, N=3, and so on, or by the nuclear charge Z, such as H=1, C=6, N=7, and so on. (Unke & Meuwly, 2018) Therefore, it creates an (unnecessary) relationship between the input data, namely H<C<N, which could influence on the network performance. Other encoding models are also suggested, for instance, representing each atom of the input molecule by the following fingerprint: H=[1 0 0 …], C=[0 1 0 …], N=[0 0 1 …], and so on. (Unke & Meuwly, 2018) However, these fingerprints do also have drawbacks because the dimensions of the encoding vector depend on the number of atoms in the structure and may vary from molecule to molecule;

moreover, based on this model, the atoms belonging to the same group in the periodic table of elements do not behave the same.

In this study, we used the so-called *Coulomb matrix*, $\boldsymbol{C}$, to encode the molecular features, which contains both the geometrical information of the three-dimensional structure and the atom type in the molecule. (Rupp, et al., 2012) For any two atoms i and j in a given input molecule, the matrix element C_{ij} is as follows:

$$C_{ij} = \begin{cases} \dfrac{Z_i^{2.4}}{2} & i = j \\ \dfrac{Z_i Z_j}{r_{ij}} & i \neq j \end{cases} \qquad (7)$$

where Z_i is the atomic number of the i-th atom and r_{ij} is the distance between the atoms i and j. The fingerprint represented by the Coulomb matrix, $\boldsymbol{C}$, has some advantages, such as it takes into account the three-dimensional molecular structure, and it is invariant under rotational and translation of the structure. To calculate $\boldsymbol{C}$ for a given molecular structure, we need the nuclear charges for each atom and the Cartesian coordinates of the atomic positions taken from the equilibrium geometry. However, note that $\boldsymbol{C}$ is not invariant under the permutations of the atom order in a molecule. Therefore, the spectrum of eigenvalues of matrix $\boldsymbol{C}$ can be used as a fingerprint of the molecule, since they are invariant under both rotation/translation and permutations of the rows and columns.

A second feature descriptor that we used in this study is the so-called *sum over bonds*, which is a numerical descriptor representing the vector of bond types present in a molecule, similar to (Yang, et al., 2019). If N_b is the number of unique bonds in the dataset of the compounds studied, then a vector with dimensions N_b is constructed for each molecule with entry either zeros or the integers giving the frequency of appearance for each bond type in molecular structure. This fingerprint descriptor vector has the same length within the dataset. Then, the vector descriptor of the sum over bonds is concatenated at the end of the Coulomb matrix descriptor.

To construct the input descriptor vector for a macromolecule, we introduced the following model. For example, suppose we would like to calculate the change on the Gibbs free energy upon the mutations (either single or multiple mutations) or perform pKa calculations for a selected residue in a protein. We label each residue or nucleotide of the input sequence with an ID from 1 to 24. That is, we form a descriptor vector with length $N_1 = 24$, $\boldsymbol{X}_1$, which is a vector of zeros and ones defined as the following:

$$\begin{array}{lccccccl} & & \text{VAL} & & \cdots & \text{THR} & \cdots & \leftarrow \text{mutation point} \\ & & \downarrow & & \cdots & \downarrow & \cdots & \\ \mathrm{X}_1 = & 0 & 1 & 0 & \cdots & 1 & \cdots & \leftarrow \text{descriptor vector} \\ & \uparrow & \uparrow & \uparrow & \cdots & \uparrow & \cdots & \\ & \text{ALA} & \text{VAL} & \text{LEU} & \cdots & \text{THR} & \cdots & \leftarrow \text{A. A. dictionary} \end{array} \quad (8)$$

where *A. A. dictionary* represents the dictionary of all amino acids. In addition, to characterize the environment around any mutation point, we determine another descriptor vector, namely$\boldsymbol{X}_2$ with length $N_2 = 24$, which is defined as the following. For each mutation point amino acid i, we determine the nearest neighbor amino acids $k = i_1, i_2, \cdots, i_{n.n.}$, based, for example, on the center of mass distance. Then, the *j*th element $X_j^{(2)}$ of the vector $\boldsymbol{X}_2$ is defined as a modified `Coulombic matrix`:

$$X_j^{(2)} = \sum_i \sum_{k=i_1}^{i_{n.n.}} \begin{matrix} \frac{1}{r_{ik}} & k = j \\ 0 & k \neq j \end{matrix} \qquad (9)$$

In Eq. (9), the first sum runs over all point mutation amino acids, and the second sum runs over all nearest neighbors of amino acids *i*. Here, r_{ik} denotes the center-to-center distance between the two amino acids. To take into account the polarity of the amino acids, we introduce a binary vector of dimension $N_p = 3$, such that

$$\boldsymbol{X}_p^{(1)} = [1\ 0\ 0], \boldsymbol{X}_p^{(2)} = [0\ 1\ 0], \boldsymbol{X}_p^{(3)} = [0\ 0\ 1] \qquad (10)$$

where $\boldsymbol{X}_p^{(1)}$ represents a non-polar amino acid, $\boldsymbol{X}_p^{(2)}$ represents an uncharged polar amino acid, and $\boldsymbol{X}_p^{(3)}$ represents a charged polar amino acid. In addition, we also added another component to the net vector, which is a real value representing the percentage of the buried part of the amino acids $(\%SASA\}_{buried})$, which is defined as the ratio of the buried surface with the solvent accessible surface area of the amino acid in the protein structure, and it is represented by the vector $\boldsymbol{X}_4$. Note that vector $\boldsymbol{X}_4$ can also include other properties, such as the temperature, concentration of the salt, and pH value of the experiment; therefore, we can write:

$$X_4 = [\%SASA_{buried}\ T\ c\ pH\ \ldots] \qquad (11)$$

In Eq. (11), T, c, and pH are the temperature (in kelvin), concentration (in molar) and pH, respectively.

To determine the descriptor vector of the macromolecule, such as protein, we concatenate the vectors $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_p^{(i)}$, and $\boldsymbol{X}_4$ into a net descriptor vector $\boldsymbol{X}$ with length $N = 55$. Note that in the expression given for $X_j^{(2)}$, other properties can also be encoded. For example, we can encode the dielectric properties in the vicinity of each amino acid in the structure by modifying it as follows:

$$X_j^{(2)} = \sum_i \sum_{k=i_1}^{i_{n.n.}} \begin{cases} \dfrac{1}{\varepsilon_{ik} r_{ik}} & k = j \\ 0 & k \neq j \end{cases} \qquad (12)$$

where ε_{ik} is the dielectric constant of the environment in the vicinity of the mutated amino acid i, which can be taken a simple distant dependent dielectric constant between the amino acid i and its nearest neighbor k: $\varepsilon_{ik} = D\, r_{ik}$, where D is a constant or even other complicated distance dependence function. (Mehler, 1996; Mehler & Guarnieri, 1999) However, in this work, a different, more complicated distance dependent dielectric constant is considered, such as the sigmoidal function (Mehler, 1996; Mehler & Guarnieri, 1999):

$$\varepsilon(r) = \frac{\varepsilon_w + D_0}{1 + k \exp(-\kappa(\varepsilon_w + D_0) r)} - D_0 \qquad (13)$$

where r is the distance between two amino acids, $\varepsilon_w = 80$ is the dielectric constant of water,

$$D_0 = 8, \kappa = 0.5/(\varepsilon_w + D_0), \ k = (\varepsilon_w - \varepsilon_p)/(D_0 + \varepsilon_p) \qquad (14)$$

with $\varepsilon_p = 2$ being the dielectric constant of protein. A plot of the $\varepsilon(r)$ versus the distance r is presented in Figure 3 for both simple functions of the distance of dielectric constant and sigmoidal distance dependence function of the dielectric constant. Here, the sigmoidal function gives a smooth variation of the dielectric constant screening the electrostatic interactions from 2 (which is the dielectric constant of the internal protein) to 80 (which is the dielectric constant of bulk water, as shown in Figure 3.

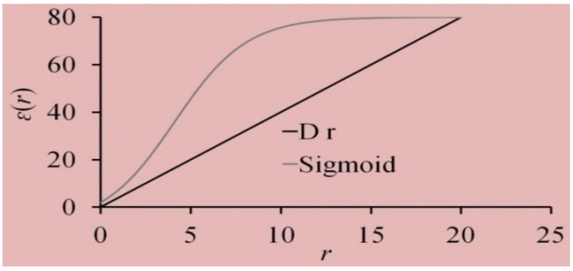

Figure 3. Dielectric constants as a function of the distance r between the amino acids for the two cases: $\varepsilon = Dr$ with $D = 8$, and dielectric constant $\varepsilon(r)$ as explained in the text for $\varepsilon_w = 80$ is the dielectric constant of water, $D_0 = 8$,

$\kappa = 0.5/(\varepsilon_w + D_0), k = (\varepsilon_w - \varepsilon_p)/(D_0 + \varepsilon_p)$ with $\varepsilon_p = 2$ being the dielectric constant of protein.

Note that these fingerprints of the structures are rotation and translation invariant. Furthermore, as a sequence of the amino acids in a macromolecular structure, the protein data bank, RCSB PDB (Berman, et al., 2000), can be used that is unique. Therefore, the descriptor vector $\boldsymbol{X}$ is an exclusive representation of a macromolecule in a dataset. Besides, the descriptor vector $\boldsymbol{X}$ has the same length for any set of the macromolecules used as input.

It is important to note that if the chemical sample space of the input descriptor vector becomes quite large, then the so-called *principal components analysis* (Karhunen, 1947) can be performed to reduce the degrees of freedom.

Web-based Services

Structure

Figure 4 shows a flowchart of the offered web-based services. It consists of several databases using different experimental or quantum mechanics data. In particular, it includes the hydration free energies of molecules database (Mobley, 2013); the heat of formation (or standard enthalpy of formation) of small molecules using quantum mechanics calculations with the Perdew-Burke-Ernzerhof hybrid functional (PBE0). (Rupp, et al., 2012; Collins, et al., 2018); pKa values experimentally calculated for different amino acids in various proteins (Thurlkill, et al., 2006; Click & Kaminski, 2009; Pace, et al., 2009; Pahari, et al., 2019); the experimental changes on the Gibbs free energies of the mutant proteins (Gromiha, et al., 1999; Bava, et al., 2004).

The *clients* are the personal computers (PCs), where the users with a login account will be able to access the first layer of the provided web-based services from the central computer, namely the *server*. With that first session login, the users will be allowed to access and manipulate the data from different databases. In particular, the users can read the data from each database in a tabular form, which then can be copied and pasted on the local computer. Besides, the users can see some statistics about the data included in each dataset, allowing for a judgment of the distribution of the data. Also, the users are permitted using the forms provided on the web to upload new data in a particular database. These data will initially be marked as ``not checked``, but after the chief administrator of the web-based services verifies the authenticity of the information, the data is added to the existing database. That can allow the databases to increase the amount of information in the future.

Using the data for each dataset, the chief administrator, frequently, performs the optimization of the artificial neural network parameter using an exhaustive machine deep-learning approach by employing the BSANN python code, internally in the server. Note that only specific users are also allowed to perform the optimization on the server using individual Login information. Then, the second session of login, for particular users, is allowed to use other services of our tools. In particular, those specific users can use web-based services online to design and optimize novel molecules. Then, using the optimized artificial neural network parameters with the training data, they can predict different properties provided for these molecules. It is interesting to note that the external plugins use graphical interfaces, making the services very user-friendly.

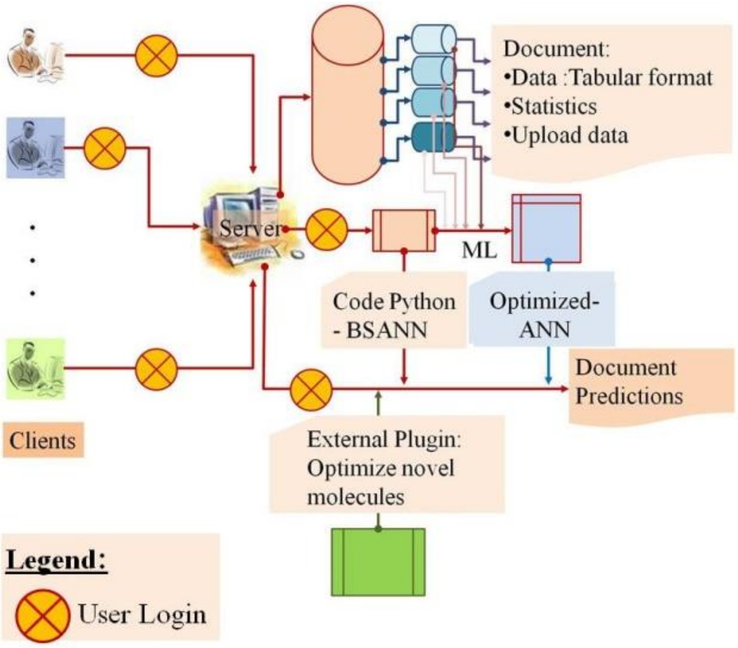

Figure 4. The flowchart of the web-based services.

Datasets

There are four databases served using that web-based service. The first database contains 642 small organic molecules, for which we know the experimental hydration free energies (Mobley, 2013), which has also been subject to our previous studies (Izairi & Kamberaj, 2017). The second database contains 1475 experimental pKa values of ionizable groups in 192 proteins, both wild type (153 proteins) and mutated (39 proteins) (Thurlkill, et al., 2006; Click & Kaminski, 2009; Pace, et al., 2009; Pahari, et al., 2019). The third database has 2693 experimental values of the Gibbs free energy changes in 14 mutant proteins (Gromiha, et al., 1999; Bava, et al., 2004). The last database has 7101 quantum mechanics heat of formation calculations (Collins, et al., 2018) (and the references therein), the so-called *QM7*, which is a subset of the so-called *GDB13* molecules, optimized at the quantum mechanics level with the Perdew-Burke-Ernzerhof hybrid functional (PBE0). (Rupp, et al., 2012) In Figure 5, we show the distribution of the average experimental pKa for each residue in the wild-type and mutated proteins of the database.

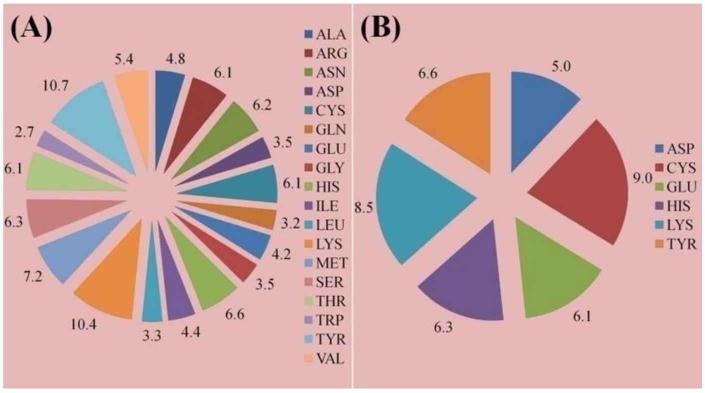

Figure 5. Average experimental pKa for each residue in the wild-type (A) and (B) mutants proteins of the database.

In Figure 6, we show the distribution of the percentage of each type of mutation in the database for which we know either $\Delta\Delta G$ or $\Delta\Delta G^{H_2O}$, namely 1: single mutation; 2: double mutations, and so on, up to 6: six mutations.

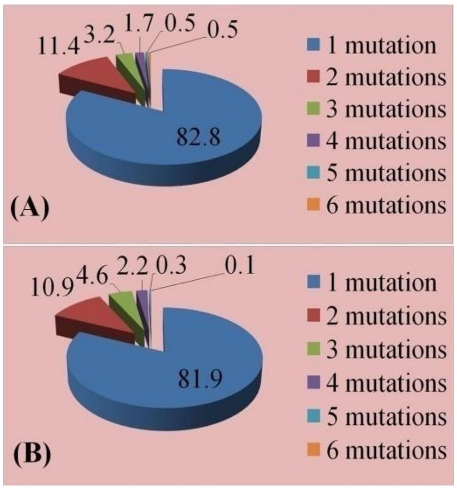

Figure 6. Percentage of each type of mutation in the database for which we know either $\Delta\Delta G$ or $\Delta\Delta G^{H_2O}$ (B). 1: single mutation; 2: double mutations, and so on, 6: six mutations.

All the data are prepared and optimized in advance using the AMBER force field (Wang, et al., 2004) in CHARMM macromolecular computer simulation program. (Brooks, et al., 2009) The BSANN code performing the optimization and prediction is in Python computer programming language. The descriptor vectors of the small molecules are based on the Coulomb matrix and the sum over bond properties, and for the macromolecular systems, they take into account the chemical-physical fingerprints of the region in the vicinity of each amino acid.

Software implementing the methods discussed in this study is free for download from the website given below. Also, from the same site, the database of all molecular structure and topology files, used in our calculations prepared using general AMBER force field, can be accessed via the web-based service: *https://hkamberajibu.wikidot.com/machine-learning*

Topological Data Analysis

It has been argued elsewhere (Kamberaj, 2020) that the diversity of the feature descriptors of the compound database is essential to increase the range of the test data that can be predicted since the machine learning methodology works very well in interpolating the new data points, but suffers on extrapolating new data outside the range covered by the training dataset. Therefore, one of the critical future developments of the automated machine learning methodologies is the choice of the training dataset and the feature descriptors of the chemical compounds. In Ref. (Kamberaj, 2020), a topological data analysis tool is discussed to analyze the feature descriptors of the molecules, which is introduced in the following. The topological data analysis (TDA) is a field dealing with the topology of the data to understand and analyze large and complex datasets. (Carlsson, 2009; Edelsbrunner & Harer, 2010) Here, we are investigating the dataset represented by a vector of feature descriptors of length N and each data point has a dimension D:

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$$
$$\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{iD}\} \qquad (15)$$

For example, N may represent the number of molecules in the dataset and D number of specific features for each molecule. Moreover, as in Ref. (Kamberaj, 2020), we assume that the data of the dataset are hidden in a ``black box", for example, a *database*, and also, they are about to be used by a machine learning, which is another ``black box". In such a situation, knowing about the topology of the data (*e.g.*, the sparsity of the data points) is of great interest. Note that the TDA is applicable even when the user has access to the data; that is, the structure of the molecules of the dataset is known as a priory. In such a situation, the TDA can be applied to determine the topology of the key feature descriptors for each molecule.

Note that the TDA is employed to reveal the intrinsic persistent features of the DNA and RNA. (Xia, et al., 2015; Mamuye, et al., 2016) Therefore, the construction of the topological spaces upon the input data of a machine learning approach can be applied for each dimension separately, namely to the time series of the form $X = \{x_{id}, x_{2d}, \cdots, x_{Nd}\}$, or for each molecular structure, namely $X = \{x_{k1}, x_{k2}, \cdots, x_{kD}\}$. But, it can also apply to both dimensions at the same time, for instance, by constructing the input data in the form of the following time series obtained by aligning feature descriptors of the molecular structures in one dimension:

$$X = \{x_{11}, \cdots, x_{1D}, x_{21}, \cdots, x_{2D}, \cdots, x_{N1}, \cdots, x_{ND}\} \qquad (16)$$

In that case, the input vector of the feature descriptors is a time series of length $N_{train} = ND$. Then, to determine the topological space for this dataset, we first define a distance $\sigma > 0$. The Vietoris-Rips simplicial complex $R(X, \sigma)$ or simply Rips complex for each $k = 1, 2, \cdots$ as a k-simplex of vertices $X_i^k = \{x_{i_1}, x_{i_2}, \cdots, x_{i_k}\}$ such that they satisfy the condition that the mutual distances between any pair of the vertices are less than σ:

$$d(x_{i_k}, x_{i_1}) \leq \sigma \quad \forall x_{i_k}, x_{i_1} \in X_i^k \qquad (17)$$

In other words, a k-simplex is part of a $R(X, \sigma)$ for every set of k data points that are distinct from each other at a resolution σ, and hence the Rips complexes form a filtration of the data from the dataset at a resolution σ. That is, for any two values of the resolution σ and σ' such that $\sigma < \sigma'$, then

$$R(X, \sigma) \subseteq R(X, \sigma') \qquad (18)$$

where $\subseteq$ denotes the subset. All the vertices of a k-simplex can be connected in a two-dimensional space by undirected edges forming a graph, which can have different two-dimensional shapes. Figure 7 illustrates how to build simplicial complexes using a set of point cloud data by increasing the resolution value σ.

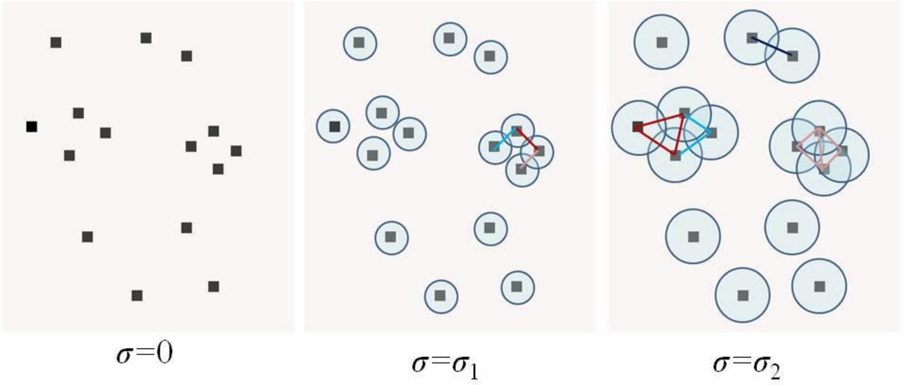

Figure 7. An illustration of building the simplicial complexes by increasing the resolution value σ. (Kamberaj, 2020)

The k-simplex dataset points form a loop that is called a *hole*. By increasing the resolution σ, the shapes grow, and some of the holes die, and some new holes are born. This process is the so-called σ loop expansion. The interval between birth and death of a hole is called *persistence interval*, indicating whether a hole is structurally relevant or just a noise into the data.

Persistent homology (PH) is an essential tool of TDA, which aims to construct a topological space gradually upon the input dataset, which is done by growing shapes based on the input data. Persistent homology measures, in this way, the persistence interval of the topological space. The features will be identified as persistent if, after the last iteration, they are still present.

This procedure is analog to systematic coarse-graining and is of crucial importance for any attempt at capturing natural feature descriptors in terms of a few relevant degrees of freedom, and thus they form the essential philosophical basis of a dataset for the machine learning approaches, as shall be shown by applying the BSANN algorithm to the current databases. (Rahmani & Kamberaj, 2020)

Data representation

To characterize the stability of the data, we adopted the mathematical model of data representation from Ref. (Bergomi, et al., 2019). According to this model, the data are represented as a sets $\mathbb{Y}$ of bounded real functions $F_{\mathbb{Y}}$:

$$F_{\mathbb{Y}}: \mathbb{X} \rightarrow \mathbb{R} \qquad (19)$$

where the set $\mathbb{X}$ represents the space where the measurements take place, and $\mathbb{Y}$ is the set of the measurements (outputs of an ANN), such as

$$\begin{gathered} Y_i \equiv F_i = F_{\mathbb{Y}}(X_i) \\ Y_i \in \mathbb{R},\ \mathrm{X_i} \in \mathbb{X} \end{gathered} \qquad (20)$$

for $i = 1,2, \cdots N_{dataset}$. That is, for example, a pKa value can be considered as a function $F_{\mathbb{Y}}$ from the real values of the feature descriptor vectors in $\mathbb{X}$ to the real numbers in $\mathbb{R}$. The structure of $\mathbb{X}$ is that of a topological data space characterized by the pseudo-metric $D_{\mathbb{X}}$, which distinguishes any two points

X_1 and X_2 in $\mathbb{X}$ only if they represent different measurements: (Bergomi, et al., 2019)

$$D_{\mathbb{X}}(X_1, X_2) = \sup_{\mathrm{F}_{\mathbb{X}} \in \mathbb{Y}} \mid F_1 - F_2 \mid \qquad (21)$$

where $F_1 = F_{\mathbb{Y}}(X_1)$ and $F_2 = F_{\mathbb{Y}}(X_2)$. In Eq. (21), the pseudo-metric $D_{\mathbb{X}}(X_1, X_2)$ is the distance, which does not obey to the property that $D_{\mathbb{X}}(X_1, X_2) = 0 \rightarrow X_1 = X_2$. (Bergomi, et al., 2019) Furthermore, it is assumed that $F_{\mathbb{Y}}$ is compact with respect to the topology of the measurements data induced by the distance $D_{\mathbb{Y}}(F_1, F_2)$:

$$D_{\mathbb{Y}}(F_1, F_2) \coloneqq \| F_1 - F_2 \|_{\infty} \qquad (22)$$

Moreover, if $\mathbb{Y}$ is compact and $\mathbb{X}$ is complete, then $\mathbb{X}$ is also compact. In other words, we are assuming that the topology spaces of $\mathbb{Y}$ and $\mathbb{X}$ as the subsets of the Euclidean space are closed (that is, they contain the limit points), and they are bounded (that is, all their points lie within some fixed distance of each other).

Consider a bijection function g, $g\colon \mathbb{X} \to \mathbb{X}$, such that and hence g is an isometry.

$$F_{\mathbb{Y}} \circ g \in \mathbb{Y}, F_{\mathbb{Y}} \circ g^{-1} \in \mathbb{Y}, \qquad \forall F_{\mathbb{Y}} \in \mathbb{Y} \tag{23}$$

The set of all these functions is said to be $\mathbb{Y}$-preserving homeomorphisms with respect to $D_{\mathbb{X}}$, denoted as $H_{\mathbb{Y}}(\mathbb{X})$. (Bergomi, et al., 2019) We, further, consider a subgroup G of the group $H_{\mathbb{Y}}(\mathbb{X})$, which represents the set of transformations g on the data preserving the equivariance. It has been proven (Bergomi, *et al.*, 2019) that $G$is a topology group with a pseudo-metric defined as:

$$D_G(g_1, g_2) \coloneqq \sup_{\mathrm{F}_{\mathbb{Y}} \in \mathbb{Y}} D_{\mathbb{Y}}(F_{\mathbb{Y}} \circ g_1, F_{\mathbb{Y}} \circ g_2) \tag{24}$$

for every $g_1, g_2 \in G$. Here, it is assumed that the action of G on $\mathbb{Y}$ is continuous. (Bergomi, et al., 2019) By definition, D_G gives the distance between two homeomorphisms determined as difference of their actions on the measurements $\mathbb{Y}$. Here, the pair $(\mathbb{Y}, G)$ is called a perception pair, and G is such that if G is complete, then it also is compact with respect to D_G. (Bergomi, et al., 2019)

On the space $\mathbb{Y}$, we can define the so-called *pseudo-distance* (Bergomi, et al., 2019) (and the references therein), $d_G\colon \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}$ as:

$$d_G(F_1, F_2) = \inf_{\mathrm{g} \in \mathrm{G}} D_{\mathbb{Y}}(F_1, F_2 \circ g) \tag{25}$$

which is associated with the group G acting on the measurements $\mathbb{Y}$. Note that if G is the identity function, that is $G(x) = x$, then $d_G = D_{\mathbb{Y}}$. Furthermore, for any $F_1, F_2 \in \mathbb{Y}$ and G_1, G_2 subgroups of $H_{\mathbb{Y}}(\mathbb{X})$, such that $G_1 \subseteq\ G_2$, then (Bergomi, et al., 2019):

$$d_{H_{\mathbb{Y}}(\mathbb{X})}(F_1, F_2) \leq d_{G_2}(F_1, F_2) \leq d_{G_1}(F_1, F_2) \leq D_{\mathbb{Y}}(F_1, F_2) \tag{26}$$

Moreover, the operators acting on the data are defined as in Ref. (Bergomi, et al., 2019). For that, $\mathbb{Y}_G$ and $(\mathbb{Z}, \mathbb{H})$ are two perception pairs with a fixed homomorphism $\mathbb{T}\colon G \to \mathbb{H}$, and $\mathbb{F}$ a perception map $\mathbb{F}\colon \mathbb{Y} \to \mathbb{Z}$, which is non-expansive, that is,

$$D_{\mathbb{Z}}\big(\mathbb{F}(F_1), \mathbb{F}(F_2)\big) \leq D_{\mathbb{Y}}(F_1, F_2), \qquad \forall F_1, F_2 \in \mathbb{Y} \tag{27}$$

According to (Bergomi, et al., 2019), a set of $\mathcal{F}^{all} = \{\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_n\}$ functions can be introduced of mapping of the perceptions pairs $(\mathbb{Y}, G)$ and $(\mathbb{Z}, \mathbb{H})$ related to the transformation $\mathbb{T}\colon G \to \mathbb{H}$:

$$\mathcal{F}^{all} \coloneqq (\mathbb{Y}, G) \to (\mathbb{Z}, \mathbb{H}) \tag{28}$$

It has been shown that (Bergomi, et al., 2019) if $\mathbb{Y}$ and $\mathbb{Z}$ are compact concerning $D_{\mathbb{Y}}$and $D_{\mathbb{Z}}$, respectively, then $\mathcal{F}^{all}$also is compact concerning the pseudo-metric:

$$D(\mathcal{F}_1,\mathcal{F}_2) := \sup_{F_{\mathbb{Y}}\in\mathbb{Y}} D_{\mathbb{Z}}\left(\mathcal{F}_i(F_{\mathbb{Y}}),\mathcal{F}_j(F_{\mathbb{Y}})\right), \qquad \forall\mathcal{F}_i,\mathcal{F}_j \in \mathcal{F}^{all} \tag{29}$$

Furthermore, to compare the data under the action of the set $\mathbb{F}$, for every $F_1, F_2 \in \mathbb{Y}$, the following pseudo-metric can be determined $[41]$:

$$D_{\mathcal{F}^{all},\mathbb{Y}}(F_1,F_2) := \sup_{\mathcal{F}\in\mathcal{F}^{all}} \| \mathcal{F}(F_1) - \mathcal{F}(F_2) \|_{\infty} \tag{30}$$

To simplify the computations, the information of every pair (X,Y), $\forall X \in \mathbb{X}$ and $\forall Y \in \mathbb{Y}$ is encoded as a persistence diagram, that is, every function $Y: X \to \mathcal{R}$ is encoded with a persistence diagram $\mathbb{D}_Y^k$ for k being a natural number. Here, $\mathbb{D}_Y^k$ is a collection of discrete points in a plane. A persistence diagram can then presented by the persistent Betti numbers, namely $r_k(Y)$. (Bergomi, et al., 2019) The matching distance δ_{match}, between the persistence diagrams, equals the metric distance d_{match} between the persistent Betti numbers:

$$\delta_{match}\left(\mathbb{D}_{Y_1}^k,\mathbb{D}_{Y_2}^k\right) = d_{match}\left(r_k(Y_1),r_k(Y_2)\right) \leq \| Y_1 - Y_2 \| \tag{31}$$

for every $Y_1,Y_2 \in \mathbb{Y}$.

Therefore, d_{match} can be used as a measure of the distance between the two measurements $F(Y_1)$ and $F(Y_2)$ data points in $\mathbb{R}$.

If one considers a non-empty set $\mathbb{f}$such that $\mathbb{f} \subseteq \mathcal{F}^{all}$then $\forall k$ fixed a pseudo-metric is defined as (Bergomi, et al., 2019)

$$\mathbb{D}_{match}^{\mathbb{f},k}(Y_1,Y_2) := \sup_{F\in\mathbb{f}} d_{match}\left(r_k\left(F(Y_1)\right),r_k\left(F(Y_2)\right)\right), \forall Y_1,Y_2 \in \mathbb{Y} \tag{32}$$

It is shown elsewhere (Bergomi, et al., 2019) that for a non-empty subset $\mathbb{f}$of $\mathcal{F}^{all}$and $\forall q > 0$ there exists a finite subset of $\mathbb{f}$, namely $\mathbb{f}^*$ such that

$$| \mathbb{D}_{match}^{\mathbb{f}^*,k}(Y_1,Y_2) - \mathbb{D}_{match}^{\mathbb{f},k}(Y_1,Y_2) | \leq q, \qquad \forall Y_1,Y_2 \in \mathbb{Y} \tag{33}$$

To sample the training data, we follow the procedure shown in Ref. (Bergomi, et al., 2019). For that, we denote by $\mathcal{Y} = \{Y_1,Y_2,\dots,Y_n\}$a dataset, represented by the following index mapping:

$$I: \mathcal{Y} \to \mathcal{J} \tag{34}$$

where $\mathcal{J} \subseteq N$ is a subset of the set of natural numbers. Furthermore, it is assumed that

$$\mathcal{Y} = \bigcup_{i\in\mathcal{J}} \mathcal{Y}_i \tag{35}$$

where $\mathcal{Y}_i$ are disjoint subsets of $\mathcal{Y}$. Furthermore, we denote by $\hat{\mathcal{F}}$ the set of operators acting on the samples $\mathcal{Y}_i$, $\forall i \in \mathcal{I}$. Then, we randomly chose N operators $\hat{F}_k$ (for $k = 1,2,\cdots,N$) from the set $\hat{\mathcal{F}}$, and denote this by the set $S = \{\hat{F}_1, \hat{F}_2, \cdots, \hat{F}_N\}$, which are such that for every $\hat{F} \in S$:

$$s_l(\hat{F}) = \max_{\mathrm{Y}_i^l, \mathrm{Y}_j^l} d_{match}\left(r_k\left(\hat{F}(Y_i^l)\right), r_k\left(\hat{F}(Y_j^l)\right)\right) \tag{36}$$

Here, we fix $k = 1$, as in Ref. (Bergomi, et al., 2019). Then, an operator $\hat{F}$ is chosen when $s_l(\hat{F})$ satisfies the criteria

$$s_l(\hat{F}) \leq e, \quad \forall l \tag{37}$$

That will create a subset $\mathcal{S} \subseteq S$, and the above criteria do not necessarily provide maximum diversity among the operators $\hat{F}$. Therefore, for any class l, the distance between two operators $\hat{F}_1$ and $\hat{F}_2$ is calculated as:

$$\Delta^l(\hat{F}_1, \hat{F}_2) \coloneqq \max_{\mathrm{Y}_i^l} d_{match}\left(r_k\left(\hat{F}_1(Y_i^l)\right), r_k\left(\hat{F}_2(Y_j^l)\right)\right) \tag{38}$$

Then, for every l, the pairs $(\hat{F}_1, \hat{F}_2)$ are sorted in ascending order of $\Delta^l(\hat{F}_1, \hat{F}_2)$, as suggested elsewhere. (Bergomi, et al., 2019) Then, only one operator is chosen for every pair that has a value of $\Delta^l(\hat{F}_1, \hat{F}_2) \leq t$, where t is a fixed threshold.

Feature descriptor vectors in higher dimensions

Note that in our discussion above, the feature descriptor vectors for each compound are considered time-invariant, and thus they represent only two- and three-dimensional feature descriptors of the (macro)molecules. However, higher feature descriptor vectors can also be constructed, such as four-dimensional feature descriptor vectors, by including the time as the fourth dimension. In that case, to build the three-dimension part of the feature descriptor vectors, different conformations of the compounds can be taken into considerations, for example, as generated from the molecular dynamics simulations.

In that case, the three-dimensional configurations of the compounds produced from the simulations can be mapped in a three-dimensional grid, where the centers of the grid points will represent the average positions of each atom obtained from its fluctuations after the configurations are aligned to remove the overall translation and rotation motion of the compounds.

Therefore, the feature descriptor vectors derived from these average structures mapped in a three-dimension grid are translation and rotation invariant. A review of such higher-dimensional descriptor vectors is discussed in Ref. (Peter, *et al.*, 2019).

Conclusions

In this study, we presented a web-based service for automation of (macro)molecular properties predictions using a new algorithm integrated into a machine learning approach. That web-service is made up of four different databases of both molecular and macromolecular systems properties. Each database has a user- friendly interface that provides the possibility to upload information into the database, which then is verified by the chief administrator of the service. Besides, the clients can perform statistics on the web related to each database, obtaining in this way, the information contained in each database in a tabular or graph format.

Also, our web-based service provides other tools and plugins for prediction of the properties of the new (macro)molecular systems using a newly developed deep-learning approach based on the bootstrapping swarm artificial neural network. Furthermore, we showed, in this study, how to create an input descriptor vector for the artificial neural network for both small molecules and macromolecular systems. The descriptor features included both the two-dimensional (macro)molecular fingerprints and the three-dimensional structure of the systems. Moreover, we shall present a statistical approach of how to estimate the bootstrapping confidence interval of the error. (Rahmani & Kamberaj, 2020)

The application of that new algorithm on our data indicated that the topological spaces of molecular properties description vector on the relevance or irrelevance of perturbations in the data analysis are crucial; those results shall be published in Ref. (Rahmani & Kamberaj, 2020). Furthermore, we envision that the persistence homology can be considered as necessary as the renormalization group theory in statistical physics when applied to equilibrium phenomena in understanding the relevant or irrelevant interactions. In this analogy, the resolution scaling factor on the topological data analysis can be considered similar to the characteristic correlation length scale that determines the judgment of the strong interactions and correlations renormalization group theory.

Acknowledgments

The authors (M.H. and H.K.) thank the International Balkan University for the support.

References

Bava, KA, Gromiha, MM, Uedaira, H, Kitajima, K & Sara, A (2004): ProTherm, version 4.0: thermodynamic database for protein and mutants, Nucleic Acids Research 32: D120-D121

Bergomi, MG, Frosini, P, Giorgi, D & Quercioli, N (2019): Towards a topological-geometrical theory of group equivariant non-expansive operators for data analysis and machine learning, Nature Machine Intelligence 1: 423-433

Berman, HM, Westbrook, J, Feng, Z, Gilliland, G, Bhat, TN, Weissig, H, Shindyalov, IN & Bourne, PE, (2000): The Protein Data Bank, Nucleic Acids

Research 28: 235-242

Brooks, B. R. and Brooks, C. L. and MacKerell, A. D. and Nilsson, L. and Petrella, R. J. and Roux, B. and Won, Y. and Archontis, G. and Bartels, C. and Boresch, S. and Caflisch, A. and Caves, L. and Cui, Q. and Dinner, A. R. and Feig, M. and Fischer, S. and Gao, J. and Hodoscek, M. and Im, W. and Kuczera, K. and Lazaridis, T. and Ma, J. and Ovchinnikov, V. and Paci, E. and Pastor, R. W. and Post, C. B. and Pu, J. Z. and Schaefer, M. and Tidor, B. and Venable, R. M. and Woodcock, H. L. and Wu, X. and Yang, W. and York, D. M. and Karplus, M. (2009): CHARMM: The biomolecular simulation program, J. Comput. Chem. 30: 1545-1614

Carlsson, G (2009): Topology and data, Bull. Amer. Math. Soc. 46: 255

Chen, R, Liu, X, Jin, S, Lin, J & Liu, J (2018): Machine learning for drug-target interaction prediction, Molecules 23: 2208-2215

Click, TH & Kaminski, GA (2009): Reproducing basic pKa values for turkey ovomucoid third domain using a polarizable force field, J. Phys. Chem. B 113: 7844-7850

Collins, CR, Gordon, GJ, von Lilienfeld, OA & Yaron, DJ (2018): Constant size descriptors for accurate machine learning models of molecular properties, J. Chem. Phys. 148: 241718-11

Decherchi, S, Berteotti, A, Bottegoni, G, Rocchia, W & Cavalli, A (2015): The ligand-binding mechanism to purine nucleoside phosphorylase elucidated via molecular dynamics and machine learning, Nat. Communic. 6: 1-10

Edelsbrunner, H & Harer, J (2010): Computational Topology: An introduction, Amer. Math. Soc

Gastegger, M, Schwiedrzik, L, Bittermann, M, Berzsenyi, F & Marquetand, P (2018): wACSF-Weighted atom-centered symmetry functions as descriptors in machine learning potentials, J. Chem. Phys. 148: 241709-11

Goh, GB, Siegel, C, Vishnu, A, Hodas, N & Baker, N (2018): How Much Chemistry Does a Deep Neural Network Need to Know to Make Accurate Predictions?, 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)

Gromiha, MM, An, J, Kono, H, Oobatake, M, Uedaira, H, Kitajima, K & Sarai, A (1999): ProTherm: thermodynamic database for protein and mutants, Nucleic Acids Research 27: 286-288

Herr, JE, Yao, K, McIntyre, R, Toth, DW & Parkhill, J (2018): Metadynamics for training neural network model chemistries: A competitive assessment, J. Chem. Phys.148: 241710-9

Izairi, R & Kamberaj, H (2017): Comparison Study of Polar and Nonpolar Contributions to Solvation Free Energy, J. Chem. Inf. Model. 57: 2539-2553

Kamath, A, Vargas-Hernandez, RA, Krems, RV, Carrington, TJ & Manzhos, S (2018): Neural networks vs. Gaussian process regression for representing potential energy surface, J. Chem. Phys. 148: 241702-7

Kamberaj, H (2020): Molecular Dynamics Simulations in Statistical Physics. Theory and Applications, 1st edn, Springer

Karhunen, K (1947): Uber Lineare Methoden in der Wahrscheinlichkeitsrechnung, Ann. Acad. Sci. Fenn. A1 37: 1-79

Lubbers, N, Smith, JS & Barros, K (2018): Hierarchical modelling of molecular energies using a deep neural network, J. Chem. Phys. 148: 241715-8

Mamuye, AL, Rucco, M, Tesei, L & Merelli, E (2016): Persistent homology analysis of RNA, Mol. Based Math. Biol. 4: 14-25

Mater, AC & Coote, ML (2019): Deep-learning in chemistry, J. Chem. Inf. Model. 59: 2545-2559

McCulloch, WS & Pitts, WH (1943): A logical calculus of the ideas immanent in neural nets, Bull. Math. Biophys.5: 115-133

Mehler, EL (1996): In Molecular Electrostatic Potential: Concepts and Applications, Murray; Elsevier Science, Amsterdam

Mehler, EL & Guarnieri, F (1999): A self-consistent, micro-environment modulated screened Coulomb potential approximation to calculate pH-dependent electrostatic effects in proteins, Biophys. J. 77: 3-22

Paass, G (1993): In Advances in Neural Information Processing Systems 5, Morgan-Kaufmann

Pace, CN, Grimsley, GR & Scholtz, JM (2009): Protein ionizable groups: pK values and their contribution to protein stability and solubility, J. Biol. Chem. 284: 13285-13289.

Pahari, S, Sun, L & Alexov, E (2019): PKAD: a database of experimentally measured pKa values of ionizable groups in proteins, Database 1-7.

Peter, SC, Dhanjal, JK, Malik, V, Radhakrishnan, N, Jayakanthan, M & Sundar, D (2019): Encyclopedia of Bioinformatics and Computational Biology, Academic Press, Oxford

Qian, N (1999): On the momentum term in gradient descent learning algorithms, Neural Networks 12: 145-151

Rahmani, B. & Kamberaj, H. (2020): Automation of (Macro)molecular Properties Using Bootstrapping Swarm Artificial Neural Network Method: A Machine Learning Approach. ACS Omega, Submitted

Riquelme, M, Lara, A, Mobley, DL, Verstraelen, T, Matamala, AR & V.-Martinez, E (2018): Hydration free energies in the FreeSolv database calculated with polarized iterative Hirshfeld charges, J. Chem. Inf. Model. 58: 1779-1797

Rupp, M, Tkatchenko, A, Muller, KR & von Lilienfeld, OA (2012): Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning, Phys. Rev. Lett. 108: 058301-5

Schneider, E, Dai, L, Topper, RQ, Drechsel-Gau, C & Tuckerman, ME (2017): Stochastic Neural Network Approach for Learning High-Dimensional Free Energy Surfaces, Phys. Rev. Lett. 119: 150601-5

Srivastava, N, Hinton, GE, Krizhevsky, A, Sutskever, I & Salakhutdinov, R (2014): Dropout: A Simple Way to Prevent Neural Networks from Overfitting, J. Mach. Learn. Res. 15: 1929-1958

Tauber, UC (2011): Renormalization Group: Applications in Statistical Physics, Nuclear Physics B Proceedings Supplement 00: 1-28

Thurlkill, RL, Grimsley, GR, Scholtz, JM & Pace, CN (2006): pK values of the ionizable groups of proteins, Protein Science 15: 1214-1218

Unke, OT & Meuwly, M (2018): A reactive, scalable, and transferable model for molecular energies from a neural network approach based on local information, J. Chem. Phys. 148: 241708-15

Wang, J, Wolf, RM, Caldwell, JW, Kollman, PA & Case, DA (2004): Development and testing of a general amber force field, Journal of Computational Chemistry 25: 1157-1174

Wehmeyer, C & Noe, F (2018): Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics, J. Chem. Phys. 148: 241703-9

Weininger, D (1988): SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules, J. Chem. Inf. Comput. Sci. 28: 31-36

Xia, K, Zhao, Z & Wei, GW (2015): Multiresolution Persistent Homology for Excessively Large Biomolecular Datasets, J. Chem. Phys. 143: 134103-12

Yang, K, Swanson, K, Jin, W, Coley, C, Eiden, P, Gao, H, G.-Perez, A, Hopper, T, Kelley, B, Mathea, M, Palmer, A, Settels, V, and K. Jensen, TJ & Barzilay, R (2019): Analyzing Learned Molecular Representations for Property Prediction, J. Chem. Inf. Model. 59: 3370-3388

Zhou, Z, Kearnes, S, Li, L, Zare, RN & Riley, P (2019): Optimization of molecules via deep-reinforcement learning, Scientific Reports 9: 10752-10