Chapter 7
A Review of Feature Reduction Methods for QSAR-Based Toxicity Prediction

Gabriel Idakwo, Joseph Luttrell IV, Minjun Chen, Huixiao Hong, Ping Gong and Chaoyang Zhang

Abstract Thousands of molecular descriptors (1D to 4D) can be generated and used as features to model quantitative structure–activity or toxicity relationship (QSAR or QSTR) for chemical toxicity prediction. This often results in models that suffer from the "curse of dimensionality", a problem that can occur in machine learning practice when too many features are employed to train a model. Here we discuss different methods of eliminating redundant and irrelevant features to enhance prediction performance, increase interpretability, and reduce computational complexity. Several feature selection and extraction methods are summarized along with their strengths and shortcomings. We also highlight some commonly overlooked challenges such as algorithm instability and selection bias while offering possible solutions.

Keywords Molecular descriptors · Feature selection · Feature extraction · Toxicity prediction · Machine learning · Quantitative Structure–Activity or toxicity relationship (QSAR or QSTR) · Prediction accuracy

G. Idakwo · J. Luttrell IV · C. Zhang (✉)
School of Computing Sciences and Computer Engineering,
University of Southern Mississippi, Hattiesburg, MS, USA
e-mail: Chaoyang.Zhang@usm.edu

G. Idakwo
e-mail: Gabriel.Idakwo@usm.edu

J. Luttrell IV
e-mail: Joseph.Luttrell@usm.edu

M. Chen · H. Hong
Division of Bioinformatics and Biostatistics, National Center for Toxicological Research,
US Food and Drug Administration, Jefferson, AR, USA
e-mail: Minjun.Chen@fda.hhs.gov

H. Hong
e-mail: Huixiao.Hong@fda.hhs.gov

P. Gong
Environmental Laboratory, US Army Engineer Research and Development Center,
Vicksburg, MS, USA
e-mail: Ping.Gong@usace.army.mil

H. Hong (ed.), *Advances in Computational Toxicology*, Challenges and Advances
in Computational Chemistry and Physics 30,
https://doi.org/10.1007/978-3-030-16443-0_7

Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
ACO	Ant colony optimization
ECFP	Extended connectivity fingerprints
GA	Genetic algorithm
KPCA	Kernel principal component analysis
LASSO	Least absolute shrinkage and selection operator
LDA	Linear discriminant analysis
LOOCV	Leave-one-out cross-validation
MACCS	Molecular access system
MDS	Multi-dimensional scaling
PCA	Principal component analysis
PSO	Particle swarm optimization
QSAR	Quantitative structure–activity relationship
QSTR	Quantitative structure–toxicity relationship
RFE	Recursive feature elimination
SA	Simulated annealing
SAR	Structure–activity relationship
SFFS	Sequential floating forward selection
SFS	Sequential forward selection
STR	Structure–toxicity relationship
SVM	Support vector machine
Tox21	Toxicology in the twenty-first century
t-SNE	t-Distributed stochastic neighbor embedding

7.1 Introduction

The limitations of in vivo and in vitro approaches for determination of the biological activity of chemicals have fostered the development of in silico approaches [1]. In silico predictive toxicology is designed to complement experimental efforts with a view toward improving the quality of toxicity predictions for safety assessment while decreasing the associated time, cost, and ethical conflicts (animal testing) [2–4]. Methodology for in silico predictive toxicology has been dominated by (quantitative) structure–activity or toxicity relationship [(Q)SAR or (Q)STR] (hereafter called SAR). Traditional SAR models describe a relationship between the chemical structure of molecules (numerically encoded as molecular descriptors) and their activity against a specific biological target [1]. This is achieved by establishing a trend in the molecular descriptor space that links to a biological activity. Thus, all SAR models are developed on the assumption of a similarity principle. That is, molecules with similar structures (and descriptors, consequently) will have similar biological activity [4, 5]. A SAR model to predict toxicity (T) is given in Eq. (1)

$$T = g\left(D_f\right) \tag{1}$$

where $\left(D_f\right)$ represents the feature space of molecular descriptors as chemical properties and g is a function that relates T to $\left(D_f\right)$ [2]. The accuracy of the model or function g has been shown to depend on the most representative set of molecular descriptors that will encode the useful properties of the molecules for prediction.

Molecular descriptors, being numerical features extracted from molecular structures, are the most common variables used for SAR-based toxicity prediction modeling [6]. The information encoded by descriptors depends on the molecular representation or "dimensionality" of the compound as well as the algorithm used to calculate the descriptors [7]. One-dimensional (1D) descriptors are scalars encoding physiochemical properties (molecular weight, *logP*) and constitutional parameters, such as number of atoms, bond count, atom type, ring count, and fragment counts. 1D descriptors are insensitive to the topology of the molecule and tend to be similar for distinct compounds. As a result, they are often used in combination with other descriptors. Two-dimensional (2D) descriptors are more frequently used for chemical space description. 2D descriptors, including topological indices and structural fragments, are calculated from the connection table (chemical graph) representation of a molecule. They are not only independent of the conformation of the molecule but also graph invariant (not sensitive to altering the number of graph nodes). Three-dimensional (3D) descriptors provide a more complete characterization of molecular structures. 3D descriptors require conformational searching and can discriminate between isomers; this comes at the price of being computationally expensive. The ability to discriminate between isomers can translate to less redundant features. Examples of 3D descriptors include geometric, electrostatic, quantum chemical, and WHIM & GETAWAY. Four-dimensional (4D) descriptors are much like 3D descriptors that evaluate multiple structural conformations simultaneously. Fingerprints are another form of molecular descriptors [7–9]. Commonly used fingerprints include the Molecular ACCess System (MACCS) [10] substructure fingerprints, PubChem [11], and extended-connectivity fingerprints (ECFP) [12]. These fingerprints and 2D descriptors were widely used in the Tox21 data challenge [13] where the winning submissions used over 2500 predefined features covering a wide range of data from topological and physical properties to fingerprints [14].

As shown above, the chemical structures used in SAR modeling are characterized by many molecular descriptors. It is common to generate thousands of descriptors for a single molecule [14]. It is well known that the accuracy of predictive models is not positively correlated to the dimensionality of the data, as overfitting tends to become an issue [15–17]. High-dimensional spaces are prone to include irrelevant and noisy features [18]. SARs developed using such features tend to focus on the peculiarities of molecules and fail to be generalizable [19]. In the chemical space for a given library, each descriptor adds a dimension to the n-dimensional chemical space. Every molecule in the library is assigned a coordinate depending on its values for all the descriptors. A reduction in the dimensionality of the chemical space correlates with an increasing similarity between molecules. This is important because the underlying assumption in SAR modeling posits that molecules with similar structures should have similar activity [20, 21]. Thus, one of the most important tasks prior to modeling is dimension reduction focused on keeping the most important and relevant descriptors with the maximum amount of biologically meaningful information required for predicting the desired toxicity end point. Shen et al. [13] demonstrated the usefulness of feature selection for toxicity prediction, particularly for interpreting the role of the features. By reducing the feature space, they were able to pinpoint *MolRef* and *AlogP* as the most important descriptors for predicting the toxicity of aromatic compounds.

In simple terms, dimensionality reduction is considered desirable for activity prediction modeling for the following reasons [22]:

(i) Employing fewer descriptors means that the model can focus on important information for establishing a relationship, thus improving prediction accuracy and reducing overfitting (Models with many features enjoy more discriminating power during training but are often not generalizable).

(ii) As the number of features decreases, interpretability of certain models increases.

(iii) Computational costs reduce significantly as the complexity of many learning algorithms is greater than linear [19, 23].

(iv) Elimination of irrelevant descriptors can help remove activity cliffs [7].

(v) Machine learning algorithms are statistical in nature; hence, they suffer from the "curse of dimensionality", which is common with optimization problems as described by Bellman [24].

As the dimensionality increases, the amount of data needed to develop generalizable models increases exponentially [25, 26]. SAR data rarely have an abundance of labeled molecules and, as such, the final model and resulting toxicity prediction will benefit from a reduction in dimension as a smaller dimension means fewer samples will be required during training. The optimal subset of a feature space is one which has the least number of dimensions yet offers the best learning accuracy [26]. Two techniques used to alleviate the challenges of high dimension in SAR datasets include feature selection and feature extraction.

In this review, we discuss different methods for both feature selection and feature extraction techniques, as well as their applications in SAR modeling. In the next two sections, we discuss feature selection and feature extraction methods consecutively. In the last section, we highlight important aspects that must be considered while attempting feature space reduction, such as the stability and validation of the methods.

7.2 Feature Selection

Feature selection works by selecting a subset of features from the original feature set and removing irrelevant features without altering the original representation of the data, on the basis of certain relevance criteria [18, 26–28]. The physical meanings of the features are retained.

Mathematically, considering a descriptor space $X = \{x_i, i = 1 \ldots n\}$, find a subset Y_k (with $k < n$) that maximizes an objective function $J(X)$ for the probability P that a compound is correctly predicted as active or inactive using Eq. (2).

$$Y_k = \left\{x_{(1),} x_{(2),} \ldots, x_{(k)}\right\} = \text{argmax}_{Y_k \subseteq X} J(Y_k) \tag{2}$$

Thus, the ultimate goal of feature selection is to define a subset of Y_k relevant descriptors (obtained from an initial set of X descriptors) which holds the most useful molecular structure information for learning the underlying pattern present in the data.

One pronounced benefit of feature selection is that it can be used to avoid overfitting. Models with high dimension offer many degrees of freedom and tend to learn random patterns and noise instead of important underlying patterns between descriptors and the target end point [29, 30]. Many feature selection algorithms have been documented. Broadly, these algorithms can be grouped into the following three categories depending on the availability of class labels for the training set: supervised [22, 25, 28, 31], semi-supervised [18, 32], and unsupervised [18, 33]. The choice of an appropriate method is dependent on the learning algorithm to be employed and the data to be used [34]. The focus of this review is on supervised feature selection methods. Supervised feature selection requires that the entire training dataset be labeled. Feature selection is achieved by eliminating descriptors that have a low correlation with the toxicity end point to be predicted [28]. Feature selection methods applied to supervised tasks can be classified into filter, wrapper, and embedded methods [28]. We discuss each of these methods and further describe Hybrid [35, 36] and Ensemble [37–39] methods, which are a blend of the earlier listed methods. These methods are illustrated in Fig. 7.1.

7.2.1 *Filter*

Filter methods evaluate the relevance of a feature based on its intrinsic properties and are completely independent of the learning algorithm [18, 27, 28, 40]. The majority of filter methods are univariate, where each feature is considered independently of the feature space. Multivariate methods, such as correlation-based scores and paired -scores, have also been used to assess the relevance of feature pairs and how well they synergize to enhance prediction of the desired end point [41]. Filter methods are computationally efficient and fast in comparison with wrapper methods. Their lack

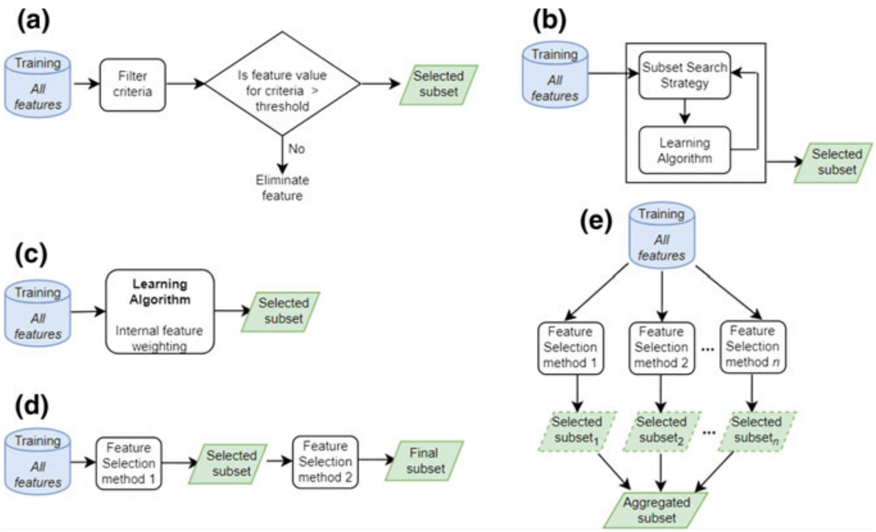

Fig. 7.1 An illustration of different feature selection methods: **a** Filter **b** Wrapper **c** Embedded **d** Hybrid **e** Ensemble

of dependence on any learning algorithm means that the features they select can be used with almost any learning algorithm. However, this independence often results in varied performance from these different learning algorithms [28]. Statistical methods make the assumption that the data they are applied on are normally distributed [40]. By not taking the learning algorithm into consideration, filter methods also turn a blind eye to the heuristics and biases of these algorithms, which may impair their predictive abilities [25].

Filter methods use feature ranking and filtering techniques as the basis for selection. Features are first evaluated and ranked based on a criterion. Then, a threshold is used to select all features above the mark that are considered to be relevant for predicting the end point [18, 28, 41], as shown in Fig. 7.1a. The elimination of low-variance and highly correlated descriptors is a common filtering technique applied to SAR datasets [14, 23, 42]. Several criteria have been employed for filtering descriptors, including variance score [32], correlation coefficient [25, 34], fisher [28, 43], and information gain [44].

7.2.2 *Wrapper*

Wrapper methods use learning algorithms to evaluate the relevance of a feature, where the learning algorithm's error rate or accuracy is treated as the objective function/criterion for evaluating a feature. A wrapper method begins by selecting a subset of the features heuristically or sequentially, and then a learning algorithm of choice is used to evaluate this subset. This process of subset generation and testing is repeated until the desired objective function is achieved [27, 28] (Fig. 7.1b). Wrappers tend to perform better than filters in selecting features since they consider feature dependencies and directly incorporate the specific biases and heuristics of the learning algorithm into the selection process. However, this implies that the selected features are unlikely to be optimal for any other classifiers [18].

The size of search space for m features is $O(2^m)$ [28]. Since evaluating the subsets of such a search space is considered an NP-hard problem, the computational inefficiency of wrappers becomes evident when using larger datasets. However, search algorithms have been proposed for selecting optimal subsets of the feature space. Broadly, we consider two groups of search strategies for wrappers: sequential and heuristic selection algorithms [25].

7.2.2.1 Sequential Selection Algorithms

Sequential selection can be achieved in two ways: forward selection and backward elimination. Sequential forward selection (SFS) begins with an empty set of features, and features are progressively incorporated into larger and larger subsets (one at a time) until no further improvement is recorded in the evaluation criterion. A backward elimination algorithm begins with the full set of features and iteratively eliminates the least relevant features [28].

The sequential floating forward selection (SFFS) [45, 46] algorithm has been suggested as an improvement over SFS because it includes flexible backtracking capabilities. Similar to SFS, SFFS adds one feature at a time as determined by the objective function. Meanwhile, it backtracks by eliminating one feature at a time from the initial subset, followed by an evaluation. If an improvement is noticed in the objective function, it leaves that feature out and moves on to add a new feature. This process goes on iteratively until the desired goal is met with the fewest number of features.

7.2.2.2 Heuristic Selection Algorithms

Heuristic search algorithms evaluate different subsets to optimize the objective function. Subsets can be generated by evaluating a search space or by generating solutions to the optimization problem, with the learning algorithm's performance being the objective function [25]. Simulated annealing (SA) [47] and genetic algorithms (GA) [48], two widely used heuristic algorithms, find a subset of features for wrappers. A hybrid of these methods has also been suggested [49]. In GA, the chromosome bits indicate if a feature should be included or not. SA, a stochastic algorithm, solves for the global minimum of a function by improving the initial solution repeatedly using small local perturbations until no such perturbations yield an improvement in the objective function. This process is randomized such that there are occasional and intentional deviations from the solution to lessen the probability of becoming stuck in local optima. The use of GA to preselect descriptor subsets for SAR modeling of artificial and real data was shown to be successful in [13] where 2D descriptors were employed to discriminate between active and inactive compounds. Particle swarm optimization (PSO) [47] and ant colony optimization (ACO) [50] algorithms may also be employed for heuristic subset search. For instance, it has been shown that the ACO algorithm is a useful method for selecting descriptors for predicting cyclooxygenase inhibitors [50].

7.2.3 Embedded

Embedded feature selection methods incorporate feature selection into the model training process. Embedded feature learning, much like wrapper methods, takes the potential dependencies among features into consideration while being more computationally efficient and less prone to overfitting as compared to wrappers [18, 27, 28, 41]. A common embedded feature selection algorithm is random forest. A random forest is an ensemble of learners with a built-in mechanism for feature selection, such as ID3 and C4.5 [28, 51]. Base learners, i.e., decision trees, look at each feature in the feature space individually and assign importance to them based on how well they contribute to the model attaining an optimal fit. Features with the lowest importance are discarded, and the forest with the least number of features and highest predictive performance is selected [28] (Fig. 7.1c). Using the top 20 molecular descriptors from the random forest predictor importance method, Newby et al. [44] obtained more accurate decision tree classification models in most cases, compared to the use of filter methods such as information gain, chi-square, and greedy search.

Pruning is another embedded feature selection approach that has been applied to neural networks as well as classical learning algorithms, specifically support vector machines (SVMs) [25]. For instance, SVM-recursive feature elimination (SVM-RFE) begins with all the features and recursively removes features that do not contribute positively to the model's predictive accuracy. To determine the optimal number of features for an RFE-based model, cross-validation is used to evaluate and select the subset with the best performance. Hence, RFE can select the best features for a specific learning algorithm. RFE is considered to be computationally expensive as it traverses through all the features one after the other [41]. Weighted Kernels [49] and regularization methods [52], like Lasso, Ridge and Elastic net, have also gained prominence.

7.2.4 Hybrid and Ensemble Feature Selection

Hybrid methods for feature selection involve combining at least two different methods and applying them, usually in succession. Hybrid methods attempt to take advantage of the benefits of the constituent methods while leveraging their strengths. In

Table 7.1 A summary of feature selection techniques

Methods	Description	Strengths	Weaknesses	Examples
Filter	• Rank features using a criterion calculated based on the data properties	• Fast, computationally inexpensive, and as such, can be applied to higher dimensions of data • Multivariate methods take the relationship between features into consideration	• Univariate methods ignore feature dependencies • Insensitive to the learner's heuristics • Deciding on the best threshold when selecting from ranked features is not deterministic	• Information gain • Chi-square test • Fisher score • Correlation coefficient • Variance threshold
Wrapper	• Use search strategies to generate feature subsets which are then evaluated by a learner	• Dependencies between features in a subset are considered • Interaction with the learner results in better performance than filter	• Features are learner specific • Interaction with the learner increases the likelihood of overfitting • Computationally expensive	• Sequential feature selection or elimination (e.g. RFE) • Genetic algorithm • Simulated annealing
Embedded	• Are learning algorithms that can weigh the contribution of each feature to its performance	• Interacts with the learner but is less prone to overfitting • Computationally less expensive than wrapper and has better performance than filter • Dependencies between features are inherently considered	• Features selected are learning algorithm specific	• LASSO • Ridge Regression • Elastic Net • Decision Trees
Hybrid	• Combines other methods to achieve the accuracy of wrappers and the efficiency of filters	• Better performance than filters and less computationally demanding than wrappers	• The setbacks of the filter and wrapper methods are not eliminated, they are reduced. The features remain specific to the learning algorithm	• Filter followed by embedded methods • Hybrid genetic algorithms
Ensemble	• Aggregates the output of different feature selection methods or subsets	• Ensures stable and robust feature selection	• Depending on the constituent methods, it could be computationally expensive and difficult to understand	• Could be made up of multiple feature selection methods

the literature, the most reported is the combination of filter and wrapper methods. Their use has been widely reported for biomedical data [35]. Hsu et al. [49] separately filtered two sets of features using F-score or information gain as the filtering criterion. The resulting features were combined and further treated with wrappers (Fig. 7.1d). They reported improved predictions in comparison with using filters alone and a decreased computational time compared to using wrappers only. Reddy et al. [53] applied a hybrid GA-based descriptor optimization technique for consistently selecting descriptor subsets that represented the whole initial descriptor space. The weights of the selected subsets were analyzed to understand the contribution of each feature to the prediction of HIV protease inhibitors, revealing the role of hydrophobic interactions. This implies the interpretability of the method.

Ensemble methods represent the application of a feature selection method on different subsets of features obtained by using subsampling strategies like bootstrapping. The resulting features from each of the subsets are aggregated using mean, weights, or simple linear aggregation [38, 39] (Fig. 7.1e). This method is often used to deal with the challenges of perturbation and instability experienced by most feature selection methods. Seijo-Pardo et al. [39] provided an in-depth discussion of ensemble methods of feature selection. Dutta et al. [54] proposed an ensemble descriptor selection that searches for descriptor subsets using a genetic algorithm whose objective function is a linear combination of the root-mean-square deviation (RMSE) of all the models in the ensemble. They reported an improvement and found that the resulting model had good performance on the PDGFR and COX-2 datasets. A 96% reduction in noise and an improvement in performance was reported by Zhu et al. [55], using a recursive random forest to rule out a quarter of the least important descriptors at each iteration. This performed better than the least absolute shrinkage and selection operator (LASSO). The authors highlighted that the difference between the prediction performance of random forest and LASSO mainly resulted from the use of variables selected by different strategies, rather than from differences between the learning algorithms.

We have summarized the characteristics, strengths, and weaknesses of the five classes of feature selection methods described above in Table 7.1 in order to assist a user in choosing the appropriate tool based on user-specific requirements and/or goals.

7.3 Feature Extraction

The algorithms employed for mathematical representation of molecular descriptors and fingerprints are independent of the size of molecules, allowing the generation of a fixed length set of descriptors for every molecule regardless of size [7]. The generation of fixed length vectors can introduce redundant descriptors for certain molecules within a library. An optimized feature set achieved by feature extraction can minimize redundancy, noise, correlation between descriptors, and consequently generate classifiers with improved prediction accuracy [20].

A mathematical description of feature extraction is as follows: Considering a descriptor space, $x \in R^n$, find a mapping $y = f(x)$ to obtain transformed feature vector y, where $y \in R^k$ and $k < n$. The vector y should preserve the majority of molecular information in R^n. The goal is to achieve a reduction in dimension without negatively impacting the prediction performance. An optimal mapping, $y = f(x)$, is one that minimizes the prediction error.

Feature extraction transforms the initial feature space to a new, lower dimension feature space by combining the features in the original space. As a result, it is difficult to associate the new features with the old. Further analysis, such as feature importance explanation, becomes very difficult as there is no physical meaning for the newly mapped features that are obtained from feature extraction. Here we discuss some commonly used feature extraction techniques.

7.3.1 Principal Component Analysis

Principal component analysis (PCA) is a multivariate, nonparametric method employed for dimensionality reduction [56, 57]. It works by performing a linear combination of the features, also referred to as the principal components, to achieve the maximum variance. At its core, PCA is centered on determining the eigenvectors of the input data's covariance matrix. This linear transformation can minimize redundancy and reduce the number of features, which increases the information in the resulting features. Each of the resulting features, called principal components, is a combination of several original features. These principal components are also highly uncorrelated because the first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible [26]. A detailed discussion on the different applications of PCA in SAR modeling was provided in [57]. Klepsch et al. [58] applied PCA to a curated P-glycoprotein inhibitors data set of 1608 compounds, where the first two principal components were reported to explain 71.7% of the variance in the dataset. This approach was applied to classification and an analysis into the effect of the initial descriptors on these two components showed that hydrophobic information, such as the number of aromatic bonds and the partition coefficient, was the major contributor to the principal components. According to [59], 2-aryl-1,3,4-Thiadiazole derivatives were classified into distinct clusters of active or inactive molecules when PCA was performed instead of using all of the descriptors calculated.

Considering that principal components are combinations of the original features, all the original features are still available within the components. This is useful for interpretation of models because knowing the original features that contribute to a component can reveal the types of features that are closely related. A key challenge with PCA is that it is unable to handle data with complicated structures that may not be represented in a linear subspace [60]. Kernel PCA (KPCA) [61, 62] was designed to serve as the nonlinear form of PCA. KPCA is based on kernel functions that intrinsically perform a nonlinear mapping of the input space to a feature space followed by performing linear PCA in this feature space. KPCA generated vectors have been used to train SVM models [59], and it was shown that KPCA is efficient over a wide range of virtual screening dataset inputs using MACCS and ECFP fingerprints. It was also observed that the KPCA embedding largely depended on the properties of the underlying representation as its performance on the ECFP fingerprint varied with the hashing employed.

7.3.2 Autoencoder

Autoencoders [63, 64] are unsupervised neural networks with an odd number of hidden layers that can be applied for nonlinear feature extraction. They employ the backpropagation algorithm to try to create a set of output values which are equal to the input by minimizing the error between the output and the input layer. The network architecture can be designed such that the middle layer is smaller, i.e., has fewer nodes than the input and output layers (Fig. 7.2). In that case, the network is forced to learn a compact representation (embedding) of the input data [65]. In an early work, Hinton et al. [17] demonstrated that autoencoders generated embeddings of images that were used to reconstruct images. A major drawback of autoencoders is that physical meaning for theoretical insight will be lost. They are also complex to train because they typically require a large amount of training data and a search through many possible hyperparameter values. Blaschke et al. [66] employed generative autoencoders to design new molecules in silico based on

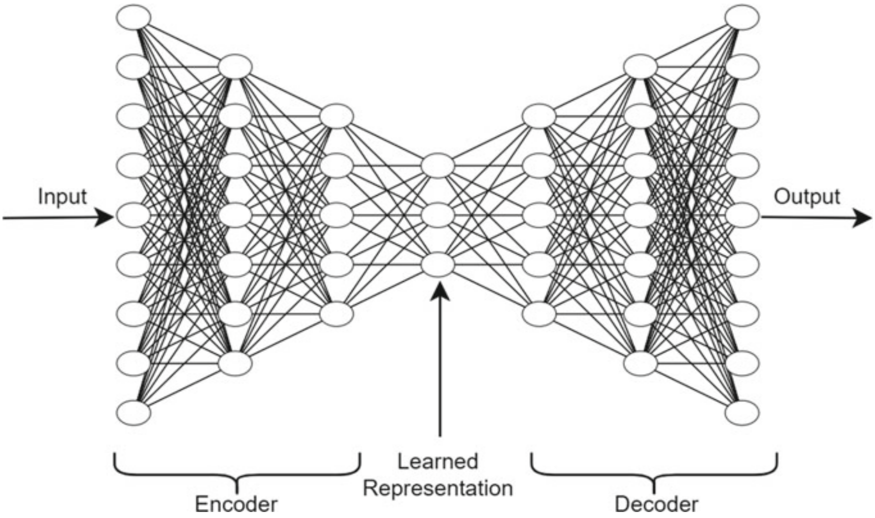

Fig. 7.2 An autoencoder indicating the reduced dimension in the middle layer

the recreated output layer. Burgoon [67] used autoencoders to screen chemicals for potential estrogenic activity by projecting the two neurons in the middle layer into a Cartesian plane. The application of autoencoders for toxicity prediction has not been widely reported, especially for feature extraction. This provides an opportunity for a future area of research.

7.3.3 Linear Discriminant Analysis

Like PCA, linear discriminant analysis (LDA) [65, 68] is a linear transformation technique commonly used for dimensionality reduction. However, LDA is supervised since the discrimination power of the features is taken into consideration. LDA computes an optimal transformation (projection) of the input data on to a line such that classes are separated as clusters. The goal of the projection is to ensure maximum class discrimination by minimizing the within-class distance while maximizing the between-class distance [26]. A weakness of LDA is that if the distribution of a dataset is significantly non-Gaussian, the LDA projections will not be able to preserve any complex structure of the data [69]. Thus, the resulting features may not have good discriminative power. Features extracted with LDA were used by Ren et al. [70] in a stepwise forward manner from a combined pool of experimental data, and chemical structure-based descriptors were employed for predicting aquatic toxicity mode of action. In this work, logistic regression was shown to have a better predictive performance than LDA using the extracted features, with a 7.3% improvement over previously reported classification rates.

In addition to the above-mentioned nonlinear dimensionality reduction techniques, there are also spectral and manifold learning methods, such as t-distributed Stochastic Neighbor Embedding (t-SNE) [71], multi-dimensional scaling (MDS) [72], spectral embedding [73], and isomap [74]. Manifold learning, a class of unsupervised nonlinear algorithms, assumes that the dimensionality of a datasets is only artificially high and thus attempts to uncover the intrinsic low dimensionality. Typically, these algorithms work by computing the similarities between points to find a nearest-neighbor, and then an eigen problem for embedding high-dimensional points into a lower dimensional space [75].

7.4 Miscellaneous

7.4.1 Feature Stability

It is common to use the performance of a model as the metric to evaluate the suitability of a feature reduction algorithm. Therefore, it is an obvious choice to optimize the selection process to obtain the best prediction power possible. However, the stability or degree of variance of feature selection methods becomes a crucial challenge when the task at hand goes beyond optimizing prediction accuracy to include improving interpretability. A simple scenario may be the case for using substructure-based descriptors for SAR modeling. It is common to consider a substructure that is very relevant for prediction as a major contributor to the activity of that molecule, implying a potential research target. However, many feature selection algorithms tend to be unstable and would yield a different subset if a little perturbation is applied (i.e., when new training samples are added or when some training samples are removed). If every perturbation results in wide variation in the selected subset, then it is difficult to conclude that a feature may be important to the molecule's activity.

Kalousis et al. [76] defined the stability of a feature selection algorithm as "the robustness of the feature subset the algorithm produces in the presence of perturbations in training sets drawn from the same generating distribution." Essentially, stability quantifies how different training sets affect the variation in the selected feature subset. Hence, a similarity measure is often employed to measure the stability of feature selection algorithms. A reliable algorithm should produce the same or similar subset for any perturbations in the training data. Alelyani et al. [77] performed experiments to investigate the causes of instability and reported that dimension, sample size, and the distribution of the training data influenced stability. Larger sample size translated to improved stability, while larger dimensions caused negative effects. Thus, researchers should pay attention to the characteristics of a training dataset. Certain algorithms are also more prone to instability than others. *ReliefF*-based feature selection is affected by the order of samples in a training set, while stochastic search algorithms like GA that use random initialization parameters tend to yield subsets that are unstable [78, 79]. Various metrics for measuring stability have been proposed [78]. To overcome the stability challenge, it has been suggested to employ ensemble selection algorithms based on the technicalities of the selection algorithm in use [78, 80, 81]. Some of these algorithms include Bootstrap sampling, random data partitioning, parameter randomization, or the combination of several of these. Developing algorithms for feature selection that are stable and possess high predictive power is still an open and challenging area. SAR-based toxicity prediction stands to gain a lot from such techniques that can improve speed and accuracy of predictions for regulatory as well as lead optimization purposes.

7.4.2 Validation of Feature Selection

In selecting the optimal feature subset, it is common to evaluate the performance of a learner based on its prediction error. A very common and overlooked mistake is to select features using the entire dataset as a preprocessing step. While this appears to be obviously wrong, it has been reported that many researchers, especially in the biomedical fields, continue to make this mistake and successfully publish in top-ranking journals [82, 83]. If a test set is to be used to evaluate the performance of a feature set, it must not be involved in the feature selection step as that will result in a selection bias that will yield overly optimistic performance estimates. This is because the features used will have an unfair advantage since they were chosen based on all of the samples. As a result, the model would have gained insight into the features which are more important in the test set. This challenge is more common with wrapper methods [83].

In many practical cases of SAR-based toxicity modeling, there are rarely a large number of compounds across the different end points to be predicted. This makes it difficult to set aside a reasonable batch of data for evaluation purposes. Methods such as cross-validation and bootstrap sampling can be used to avoid sampling bias [34, 82, 83]. Cross-validation techniques like leave-one-out cross-validation (LOOCV) and the *k*-fold method were suggested. Feature selection is to be done in the inner loop of the cross-validation procedure; hence, the algorithm takes the following form for a *k*-fold technique [82]:

(i) Randomly shuffle the data set.
(ii) Randomly split the dataset into *K* folds.
(iii) For each fold $k = 1, 2, \ldots, K$.

    a. Perform feature selection to obtain an optimal subset with good univariate correlation with the desired end point using all the data except the *k*th fold.
    b. Use the selected features and build a multivariate model with all data except the *k*th fold.
    c. Perform an evaluation using the *k*th fold.

(iv) Aggregate the performance across all *K* folds to get an unbiased evaluation.

7.5 Summary

QSAR-based predictive toxicity modeling methods are faced with input spaces of thousands of features. To improve the ability of a learner to find a generalizable relationship between molecular descriptors and the toxicity end point of interest, it is expedient to provide the learning algorithm with the minimum number of descriptors while ensuring that the resulting model is interpretable and computationally inexpensive to build. The relevance of a descriptor is assessed by its ability to discriminate between classes in qualitative classification or its correlation to a scalar in quantitative prediction.

In this review, we have discussed different feature selection and extraction methods applicable to SAR-based toxicity modeling. The strengths and weaknesses of each method are highlighted. The choice of which to use should largely depend on the available dataset, and we suggest beginning a new task with a few baseline performance values from a number of methods since no single approach is universally superior. Where the importance of descriptors is sought, feature selection methods such as *filter*, *wrapper*, *embedded* or their combinations (*hybrid* and *ensemble*) may apply. Feature extraction methods transform the features into a lower dimension while altering the physical meaning of the features. More analysis may be required to interpret the selected features. The stability of selected features and proper feature subset validation methods are often overlooked. Feature selection bias can be avoided by embedding the feature selection process within the inner loop of a cross-validation process to avoid an overly optimistic performance value. Although dimensionality reduction has been shown to improve model performance, there is still room for improvement when it comes to evaluating and validating feature selection and extraction methods and their stability. For the sake of reproducibility, researchers are encouraged to publish important parameters for feature selection or extraction methods they employed, such as the threshold for a variance score. Regardless of the choice of features (molecular descriptors, fingerprints or a combination) used for modeling, SAR models can benefit from dimensionality reduction techniques.

References

1. Lavecchia A (2015) Machine-learning approaches in drug discovery: methods and applications. Drug Discov Today 20(3):318–331
2. Raies AB, Bajic VB (2016) In silico toxicology: computational methods for the prediction of chemical toxicity. Wiley Interdiscip Rev Comput Mol Sci 6(2):147–172
3. Greene N, Pennie W (2015) Computational toxicology, friend or foe? Toxicol Res 4(5):1159–1172
4. Kruhlak NL, Benz RD, Zhou H, Colatsky TJ (2012) (Q)SAR modeling and safety assessment in regulatory review. Clin Pharmacol Ther 91(3):529–534
5. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29(6–7):476–488
6. Yang H, Sun L, Li W, Liu G, Tang Y (2018) In silico prediction of chemical toxicity for drug design using machine learning methods and structural alerts. Front Chem 6:30. https://doi.org/10.3389/fchem.2018.00030
7. Danishuddin Khan AU (2016) Descriptors and their selection methods in QSAR analysis: paradigm for drug design. Drug Discov Today 21(8):1291–1302
8. Leach AR, Gillet VJ (2007) Molecular descriptors. An introduction to chemoinformatics. Springer, Dordrecht, pp 53–74
9. Todeschini R, Consonni V (2000) Handbook of molecular descriptors. Wiley-VCH, Weinheim
10. Duan J, Dixon SL, Lowrie JF, Sherman W (2010) Analysis and comparison of 2D fingerprints: insights into database screening performance using eight fingerprint methods. J Mol Graph Model 29(2):157–170
11. National Institutes of Health (2009) PubChem substructure fingerprint. ftp://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem_fingerprints.txt. Accessed 10 Oct 2018
12. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50(5):742–754
13. Huang R, Xia M, Nguyen D-T et al (2016) Tox21Challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. Front Environ Sci 3:85. https://doi.org/10.3389/fenvs.2015.00085
14. Mayr A, Klambauer G, Unterthiner T, Hochreiter S (2016) DeepTox: toxicity prediction using deep learning. Front Environ Sci 3:80. https://doi.org/10.3389/fenvs.2015.00080
15. Subramanian J, Simon R (2013) Overfitting in prediction models—Is it a problem only in high dimensions? Contemp Clin Trials 36(2):636–641

16. Clarke R, Ressom HW, Wang A et al (2008) The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. Nat Rev Cancer 8(1):37–49
17. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
18. Ang JC, Mirzal A, Haron H, Hamed HNA (2016) Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. IEEE/ACM Trans Comput Biol Bioinform 13(5):971–989
19. Merkwirth C, Mauser H, Schulz-Gasch T, Roche O, Martin Stahl A, Lengauer T (2004) Ensemble methods for classification in cheminformatics. J Chem Inf Comput Sci 44(6):1971–1978
20. Venkatraman V, Dalby AR, Yang ZR (2004) Evaluation of mutual information and genetic programming for feature selection in QSAR. J Chem Inf Comput Sci 44(5):1686–1692
21. Bajorath J (2001) Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. J Chem Inf Comput Sci 41(2):233–245
22. Goodarzi M, Dejaegher B, Heyden YV (2012) Feature selection methods in QSAR studies. J AOAC Int 95(3):636–651
23. Shahlaei M (2013) Descriptor selection methods in quantitative structure—activity relationship studies: a review study. Chem Rev 113(10):8093–8103
24. Bellman R (2016) Adaptive control processes: a guided tour. Princeton University Press, New Jersey
25. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
26. Van Der Maaten L, Postma E, Van Den Herik J (2009) Dimensionality reduction: a comparative review. J Mach Learn Res 10:66–71
27. Cai J, Luo J, Wang S, Yang S (2018) Feature selection in machine learning: a new perspective. Neurocomputing 300:70–79
28. Tang J, Alelyani S, Liu H (2014) Feature selection for classification: a review. In: Aggarwal CC (ed) Data classification: algorithms and applications, 1st edn. CRC Press, Boca Raton, pp 37–64
29. Johnstone IM, Titterington DM (2009) Statistical challenges of high-dimensional data. Philos Trans A Math Phys Eng Sci 367(1906):4237–4253
30. Zhu X, Wu X (2004) Class noise versus attribute noise: a quantitative study. Artif Intell Rev 22(3): 177 –210
31. Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43(1):59–69
32. Sheikhpour R, Sarram MA, Gharaghani S, Chahooki MAZ (2017) A survey on semi-supervised feature selection methods. Pattern Recognit 64:141–158
33. Dy JG, Brodley CE (2004) Feature selection for unsupervised learning. J Mach Learn Res 5:845–889
34. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
35. Solorio-Fernandez S, Martinez-Trinidad JF, Carrasco-Ochoa JA, and Zhang Y-Q (2012) Hybrid feature selection method for biomedical datasets. In: 2012 IEEE symposium on computational intelligence in bioinformatics and computational biology (CIBCB), San Diego, 9–12 May 2012
36. Hsu H-H, Hsieh C-W, Lu M-D (2011) Hybrid feature selection by combining filters and wrappers. Expert Syst Appl 38(7):8144–8150
37. Guan D, Yuan W, Lee YK, Najeebullah K, Rasel MK (2014) A review of ensemble learning based feature selection. IETE Tech Rev 31(3):190–198
38. Brahim AB, Limam M (2017) Ensemble feature selection for high dimensional data: a new method and a comparative study. Adv Data Anal Classif 12(4):937–952
39. Seijo-Pardo B, Porto-Díaz I, Bolón-Canedo V, Alonso-Betanzos A (2017) Ensemble feature selection: homogeneous and heterogeneous approaches. Knowl Based Syst 118:124–139
40. Janecek A, Gansterer W, Demel M, Ecker G (2008) On the relationship between feature selection and classification accuracy. Proc Mach Learn Res 4:90–105

41. Hira ZM, Gillies DF (2015) A review of feature selection and feature extraction methods applied on microarray data. Adv Bioinformatics. https://doi.org/10.1155/2015/198363
42. Rajarshi G, Jurs PC (2004) Development of linear, ensemble, and nonlinear models for the prediction and interpretation of the biological activity of a set of PDGFR inhibitors. J Chem Inf Comput Sci 44(6):2179–2189
43. Guo G, Neagu D, Cronin MTD (2005) A study on feature selection for toxicity prediction. In: Wang L, Jin Y (eds) Fuzzy systems and knowledge discovery. Springer, Heidelberg, pp 31–34
44. Newby D, Freitas AA, Ghafourian T (2012) Pre-processing feature selection for improved C&RT models for oral absorption. J Chem Inf Model 53(10):2730–2742
45. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recognit Lett 15(11):1119–1125
46. Brendel M, Zaccarelli R, Devillers L (2010) A quick sequential forward floating feature selection algorithm for emotion detection from speech. In: INTERSPEECH-2010, Chiba, 26–30 September 2010
47. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN’95—international conference on neural networks, Perth, 27 November–1 December 1995
48. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Longman Publishing Co., Inc, Boston
49. Revathy N, Balasubramanian R (2012) GA-SVM Wrapper approach for gene banking and classificaiton using expressions of very few genes. J Theor Appl Inf Technol 40(2):113–119
50. Shen Q, Jiang J-H, Tao J et al (2005) Modified ant colony optimization algorithm for variable selection in QSAR modeling: QSAR studies of cyclooxygenase inhibitors. J Chem Inf Model 45(4):1024–1029
51. Jain D, Singh V (2018) Feature selection and classification systems for chronic disease prediction: a review. Egypt Informatics J 19(3):179–189
52. Osman H, Ghafari M, Nierstrasz O (2017) Automatic feature selection by regularization to improve bug prediction accuracy. In: 2017 IEEE workshop on machine learning techniques for software quality evaluation (MaLTeSQuE), Klagenfurt, 21 February 2017
53. Reddy AS, Kumar S, Garg R (2010) Hybrid-genetic algorithm based descriptor optimization and QSAR models for predicting the biological activity of tipranavir analogs for HIV protease inhibition. J Mol Graph Model 28(8):852–862
54. Dutta D, Guha R, Wild D, Chen T (2007) Ensemble feature selection: consistent descriptor subsets for multiple QSAR models. J Chem Inf Model 47(3):989–997
55. Zhu X-W, Xin Y-J, Ge H-L (2015) Recursive random forests enable better predictive performance and model interpretation than variable selection by LASSO. J Chem Inf Model 55(4):736–746
56. Lauria A, Ippolito M, Almerico AM. (2009) Combined use of PCA and QSAR/QSPR to predict the drugs mechanism of action. An application to the NCI ACAM database. QSAR Comb Sci 28(4):387–395
57. Yoo C, Shahlaei M (2018) The applications of PCA in QSAR studies: a case study on CCR5 antagonists. Chem Biol Drug Des 91(1):137–152
58. Klepsch F, Vasanthanathan P, Ecker GF (2014) Ligand and structure-based classification models for prediction of P-glycoprotein inhibitors. J Chem Inf Model 54(1):218–229
59. Hemmateenejad B, Miri R, Jafarpour M, Tabarzad M, Foroumadi A (2006) Multiple linear regression and principal component analysis-based prediction of the anti-tuberculosis activity of some 2-aryl-1,3,4-Thiadiazole derivatives. QSAR Comb Sci 25(1):56–66
60. Manikandan G, Abirami S (2018) A survey on feature selection and extraction techniques for high-dimensional microarray datasets. In: Anouncia SM, Wiil UK (eds) Knowledge computing and its applications. Springer, Singapore, pp 311–333
61. Reverter F, Vegas E, Oller JM (2014) Kernel-PCA data integration with enhanced interpretability. BMC Syst Biol 8(2):S6
62. Wang Q (2012) Kernel principal component analysis and its applications in face recognition and active shape models. https://arxiv.org/abs/1207.3538. Accessed 10 October 2018

63. Baldi P (2012) Autoencoders, unsupervised learning, and deep architectures. In: Proceedings of ICML workshop on unsupervised and transfer learning, Bellevue, 2 July 2012
64. Goh GB, Hodas NO, Vishnu A (2017) Deep learning for computational chemistry. J Comput Chem 38(16):1291–1307
65. Chandra B, Sharma RK (2015) Exploring autoencoders for unsupervised feature selection. In: 2015 international joint conference on neural networks (IJCNN), Killarney, 12–17 July 2015
66. Blaschke T, Olivecrona M, Engkvist O, Bajorath J, Chen H (2018) Application of generative autoencoder in de novo molecular design. Mol Inform 37(1–2):1700123
67. Burgoon LD (2017) Autoencoder predicting estrogenic chemical substances (APECS): an improved approach for screening potentially estrogenic chemicals using in vitro assays and deep learning. Comput Toxicol 2:45–49
68. Ye J, Ji S (2009) Discriminant analysis for dimensionality reduction: an overview of recent developments. In: Boulgouris NV, Plataniotis KN, Micheli-Tzanakou E (eds) Biometrics: theory, methods, and applications. IEEE Press, Piscataway, pp 1–20
69. Yan H, Dai Y (2011) The comparison of five discriminant methods. In: 2011 International conference on management and service science, Wuhan, 12–14 August
70. Ren YY, Zhou LC, Yang L, Liu PY, Zhao BW, Liu HX (2016) Predicting the aquatic toxicity mode of action using logistic regression and linear discriminant analysis. SAR QSAR Environ Res 27(9):721–746
71. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
72. Borg I, Groenen PJF (2005) Modern Multidimensional Scaling, 2nd edn. Springer Science + Business Media Inc, New York
73. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–1396
74. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
75. Izenman AJ (2012) Introduction to manifold learning. Wiley Interdiscip Rev Comput Stat 4(5):439–446
76. Kalousis A, Prados J, Hilario M (2007) Stability of feature selection algorithms: a study on high-dimensional spaces. Knowl Inf Syst 12(1):95–116
77. Alelyani S, Liu H, Wang L (2011) The effect of the characteristics of the dataset on the selection stability. In: 2011 IEEE 23rd international conference on tools with artificial intelligence, Boca Raton, 7–9 November 2011
78. Yang P, Zhou BB, Yang JY-H, Zomaya AY (2013) Stability of feature selection algorithms and ensemble feature selection methods in bioinformatics. In: Elloumi M, Zomaya AY (eds) Biological knowledge discovery handbook: preprocessing, mining, and postprocessing of biological data. John Wiley & Sons Inc, Hoboken, pp 333–352
79. Yang P, Ho JW, Yang Y, Zhou BB (2011) Gene-gene interaction filtering with ensemble of filters. BMC Bioinformatics 12:S10. https://doi.org/10.1186/1471-2105-12-S1-S10
80. Yang F, Mao KZ (2011) Robust feature selection for microarray data based on multicriterion fusion. IEEE/ACM Trans Comput Biol Bioinforma 8(4):1080–1092
81. Abeel T, Helleputte T, Van de Peer Y, Dupont P, Saeys Y (2010) Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics 26(3):392–398
82. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, 2nd edn. Springer-Verlag, New York
83. Ambroise C, McLachlan GJ (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Natl Acad Sci U S A 99(10):6562–6566

Gabriel Idakwo is Ph.D. student and research assistant in the School of Computing Sciences and Computer Engineering at the University of Southern Mississippi. His research has focused on machine learning applied to problems in the field of computational chemistry, including Chemical Structure-Activity-Relationship Modeling, toxicity prediction, and feature selection. Gabriel received his BS from Ahmadu Bello University in 2008 and his MS from the University of Southern Mississippi in 2014.

Joseph Luttrell IV is Ph.D. student and research assistant in the School of Computing Sciences and Computer Engineering at the University of Southern Mississippi. His research has focused on machine learning and its application to problems in the areas of bioinformatics and image processing, including protein residue–residue contact prediction, face recognition, and object detection. Joseph received his BS in computer science from the University of Southern Mississippi in 2016.

Minjun Chen is Principal Investigator working at the Division of Bioinformatics and Biostatics of the US FDA's National Center for Toxicological Research and serves as the adjunct faculty and mentor for the bioinformatics program offered jointly by the University of Arkansas at Little Rock (UALR) and the University of Arkansas for Medical Sciences (UAMS). He received the FDA award for outstanding junior investigator (2012), the NCTR scientific achievement award (2014) and the FDA scientific achievement award (2017). He has been co-chairing the FDA Liver Toxicity Working Group since 2014 and edited the book titled "Drug-induced Liver Toxicity" that was published by the Springer in 2018. His primary research interests encompass drug-induced liver injury, drug safety, bioinformatics, and personalized medicine. He has authored or co-authored more than 90 original publications and book chapters.

Huixiao Hong is Chief of Bioinformatics Branch, Division of Bioinformatics and Biostatistics, National Center for Toxicological Research (NCTR), US Food and Drug Administration (FDA), working on the scientific bases for regulatory applications of bioinformatics. Before joining the FDA, he was Manager of Bioinformatics Division of Z-Tech, an ICFI company. He was a research scientist at Sumitomo Chemical Company in Japan and a visiting scientist at National Cancer Institute at National Institutes of Health. He was also Associate Professor and the Director of Laboratory of Computational Chemistry at Nanjing University in China. Dr. Hong is a member of the steering committee of OpenTox, a member of board directors of US MidSouth Computational Biology and Bioinformatics Society, and is in the leadership circle of US FDA modeling and simulation working group. He published more than 180 scientific papers and served as Editor-in-Chief and Editorial Board member for multiple peer-reviewed journals. He received his Ph.D. from Nanjing University in China in 1990 and conducted research in Leeds University in England in 1990–1992.

Ping Gong is Principal Investigator with multidisciplinary expertise in environmental genomics, bioinformatics, bioengineering, mechanistic and predictive toxicology, and molecular modeling. He earned a Bachelor's degree in Environmental Biology and Ecology from Peking University and a Ph.D. in Environmental Toxicology from the Institute of Applied Ecology, Chinese Academy of Sciences. He completed his postdoctoral studies in the Technology University of Berlin (TU Berlin), Swedish University of Agricultural Sciences (SLU), and Biotechnology Research Institute of National Research Council of Canada. As a research biologist, he has led a number of multi-year projects that have greatly improved the fundamental and mechanistic understanding underlying the observed toxicities of military unique and other environmental contaminants. His current research interests focus on mode of action-guided predictive toxicology, genetic variations-conferred herbicide resistance, epigenetics-driven transgenerational inheritance of phenotypic traits, and synthetic biology-based bioengineering and biocontrol of invasive species. He has advised more than 18 graduate students and published over 75 peer-reviewed research articles and book chapters. He also served on the Board of Directors of MidSouth Computational Biology and Bioinformatics Society (MCBIOS) as well as the editorial boards of Frontiers in Genetics and Environmental Toxicology and Chemistry.

Chaoyang Zhang is Professor in the School of Computing Sciences and Computer Engineering at the University of Southern Mississippi (USM). He served as the Director of School of Computing at USM from 2008 to 2014. His research interests include data mining, machines learning, big data analytics, and their applications to the interdisciplinary areas of bioinformatics, health care and toxicity analysis. His research has been supported by the National Science Foundation (NSF), Department of Defense (DOD), National Academy of Sciences (NAS), and National Institute of Health (NIH). Dr. Zhang received his MS in computer science and Ph.D. in computational analysis and modeling from Louisiana Tech University. Prior to joining USM in 2003, he was a research assistant professor in the Department of Computer Science at the University of Vermont.