An application of some machine learning methods for biological data modeling

An application of some machine learning methods for biological data modeling

Fatima Sapundzhi[1], **Meglena Lazarova**[2], **Tatyana Dzimbova**[1,3], **Slavi Georgiev**[4,5]

[1] Department of Communication and Computer Engineering, Faculty of Engineering, South-West University "Neofit Rilski", 66 Ivan Myhailov Str., 2700 Blagoevgrad, Bulgaria
[2] Faculty of Applied Mathematics and Informatics, Technical University of Sofia, 8 "St. Kliment Ohridski", Blvd., 1000 Sofia, Bulgaria
[3] Institute of Molecular Biology, Bulgarian Academy of Sciences, Acad. G.Bonchev Str, bl. 21, 1113 Sofia, Bulgaria,
[4] Department of Information Modeling, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., bl. 8, 1113 Sofia, Bulgaria
[5] Department of Applied Mathematics and Statistics, Faculty of Natural Sciences and Education, University of Ruse, 8 Studentska Str., 7004 Ruse, Bulgaria

sapundzhi@swu.bg

Abstract. The development of fast and reliable methods for predicting the biological activity of the substances in computational biology is of a great importance. This improves the development of some new compounds while keeping costs low. Among many scientists, an attractive target for docking experiments is the Delta-opioid receptor (DOR) and delta-opioid ligands (DOL). Their biological efficacy can be measured by various techniques, which could facilitate the establishment of the relationship between the structure of the compounds and their biological effect. The relationship between the results of the computer experiments and the biological activity of these compounds is modelled by using machine learning regressors. The primary goal of this study is to determine the most appropriate neural network for modelling the relationship between *in vitro* and *in silico* results for DOR and delta-opioid ligands.

1. Introduction

In our society there is a growing need to treat pain, but usually the current treatments are often ineffective and have side effects for the patients. Morphine is a powerful analgesic that can lead to physical dependence and addiction which drives the search for new pain medications with minimal side effects, see [1-3]. The delta-opioid receptor (DOR) is an attractive target for the development of new pain therapies. DOR is highly selective for encephalins and produces both spinal and supraspinal analgesia. The encephalin and its analogs have been extensively studied in order to discover potent and selective DOR ligands because they are endogenous opioid pentapeptides with DOR preference [4-9].

During the early stage of drug's discovery, the computer-aided drug's usage design (CADD) techniques was very attractive in comparison to the manual, time-consuming, and labor-intensive methods of the traditional drug design and testing. The use of CADD is possible to reduce the time and the costs associated with bringing a drug to market and accelerate research. Generally, CADD can be

classified into two types: ligand-based methods and structure-based methods. Among these, the ligand-based methods assume that a drug's chemical structure has an effect on its biological activity while the structure-based methods assume that a target's 3D structure has an effect on its biological activity. In these methods, small molecules are matched to select compounds for further study based on their properties. In the early stages of drug discovery, the molecules are evaluated based on some characteristics such as similarity, correlation with specific properties or binding energy [10-16].

The main purpose of this study is to construct a model structure-activity relationship between a series of delta-opioid ligands and molecular docking results with a model of the delta-opioid receptor (DOR) by using machine learning. The next section briefly mentions the software we use to obtain the regression data. Section 3 begins with the description of the data itself. Then the available polynomial regression results are recalled, and the new machine learning methods are explained. Further the calibration algorithm is visually described, and eventually the results are discussed. The paper finishes with a conclusion section.

2. Experimental

2.1. Docking procedure

The docking experiments were carried out with the software GOLD and GoldScore algorithm, see [13,17]. The scale of the score gives an information how good the pose is.

2.2. Receptor

A theoretical model of DOR (PDBe:1ozc) is obtained from Protein Data Bank in Europe (https://www.ebi.ac.uk/pdbe/) [13].

2.3. Ligands

The data for the DOR analogs from *in vitro* test used in this investigation is presented in Table 1, see [13, 18, 19]. The ligand's preparation was done by the program Molegro Molecular Viewer V 2.5 (MMV), (www.clcbio.com). The prepared structures were used for molecular docking by the software GOLD 5.2. The total energies of the formed ligand-receptor complex after docking were calculated by MMV 2.5, using the MolDock scoring function.

3. Results and discussions

The computer modeling and the docking experiments with DOR (PDBe:1ozc) and delta-opioid ligands, presented in Table 1, were carried out with software GOLD 5.2 and all optimization algorithms in the program: GoldScore, ChemScore, ASP and ChemPLP [17]. In order to establish the relationship between the docking results (scoring function) and the values of the biological activity of the delta-opioid ligands, a correlation between these two sets of data is searched for. The correlation analysis was performed by GraphPad Prism software (http://www.graphpad.com/). In some previous studies, a significant correlation between the values of the optimization function GoldScore and the values of the biological activity of the investigated ligands, see [13] where the Pearson correlation coefficient's value is R=-0.72730 was found.

Table 1. Data for the ligands from in vitro test and docking studies (GoldScore function from GOLD and MolDock function from MMV), see [13]

Ligand	*GoldScore*	*Total energy*	*Ligand efficacy*
[Cys(Bzl)^2-Leu^5]-enk	64.68	-107.022	9.3

[Cys(Bzl)^2-Met^5]-enk	81.49	-89.091	3.5
[Cys(O_2NH_2)^2-Leu^5]-enk	67.72	-97.619	29.2
[Cys(O_2NH_2)^2-Met^5]-enk	73.91	-91.246	7.3
[DCys(O_2NH_2)^2-Leu^5]-enk	74.73	-84.852	7.4
[DCys(O_2NH_2)^2-Met^5]-enk	75.13	-86.221	7.1
DPDPE [D-Pen^{2,5}]-enkephalin	57.67	-109.709	30.2
[HCys(O_2NH_2)^2-Leu^5]-enk	68.43	-62.774	3.4
[HCys(O_2NH_2)^2-Met^5]-enk	78.65	-93.301	4.5
[Leu^5]-enkephalin	73.42	-81.869	5.8
[Met^5]-enkephalin	73,26	-118.971	3.6

For determining the relationship between the biological activity and the docking results we use the MATLAB software and the Surface Curve Fitting Toolbox. The Surface Curve Fitting Toolbox provides applications and functions for fitting curves and surfaces to data. A fitting of the experimental data for DOR (PDBe:1ozc) with a polynomial function $z = f(x,y)$ was carried out, see [13]. The values $(z_1, z_2, \dots, z_n)$ of the dependent variable z represent the values of the biological activity of the delta-opioid ligands. The values $(x_1, x_2, \dots, x_n)$ of the independent variable x represent the results from the docking, where the values of the Gold Score function is calculated by GOLD. The values $(y_1, y_2, \dots, y_n)$ of the independent variable y represents the total energies for the ligand-receptor complex formed after the docking procedure, where the values of the MolDock function is calculated by MMV.

In our previous analysing investigation the behaviour of the dependent variable, which depends on more independent variables is applied by using the Surface Fitting Toolbox of MATLAB (http://www.mathworks.com/products/matlab), see [13]. The obtained model could be interpreted as a surface-fitting function and analysed as an experimental data by using the least squares method. To evaluate the goodness of fit, we use the following statistical parameters:

SSE (Sum of squares due to error),
$R - Square$ (R^2), $Adjusted\ R^2$,
$RMSE$ (Root Mean Squared Error).

The results are presented in the following Table 2:

Table 2. Goodness of fit for the polynomial models obtained by the least squares method.

Models Poly (x,y)	Degree of x	Degree of y	SSE	R^2	Adj R^2	RMSE
Poly11	1	1	443.5817	0.5446	0.4308	7.4463
Poly22	2	2	167.1000	0.8285	0.6569	5.7810
Poly33	3	3	0.0092	1.000	0.9999	0.0960

The third-degree polynomial model for fitting the experimental data showed good fitting properties and significant predictive ability. It is suitable for determining the relationship structure and biological activity ($R^2 = 1.0$, $SSE = 0.0092$, $adj\ R^2 = 0.9999$, $RMSE = 0.0960$). Similar studies had been conducted with other compounds, see [13-16].

In this research, our aim is to obtain the relationship $z = f(x,y)$, where z represents the biological activity values of delta-opioid ligands, and the independent variables x and y correspond to the values of the GoldScore function and respectively to the MolDock function. Our intent is to define this nonlinear relationship by utilizing machine learning techniques, see [19].

The regression models that we employed are the k-Nearest Neighbors, Gradient Boosting, Random Forest and the Extra Trees. The latter three are ensemble methods which are anticipated to offer superior

performance in comparison to the first traditional method incorporated here as a standard of comparison, see [20-23].

The **k-Nearest Neighbors** (k-NN) technique is an adaptable and an intuitive algorithm that is employed in numerous machine learning applications including classification and regression tasks. More precisely, the k-NN regression, a derivative of this method, is a potent instrument for predicting continuous variables.

The fundamental principle of the k-NN regression stems from the belief that the data points in the dataset, which are close to each other in the feature space, are expected to have similar target values. The method works by pinpointing the 'k' data points in the training set that are closest to a new unseen data point based on a certain distance metric – typically such metric is the Euclidean distance in the multi-dimensional feature space. Once these 'k' nearest points are found, the algorithm generates a prediction for the new data point. In regression scenarios this prediction is commonly the mean of the target variable for the 'k' closest points. It means that the predicted value for an unseen data point is the mean of the values of its nearest neighbors. One of the major strengths of the k-NN regression is its simplicity and intuitive nature. It does not enforce any explicit assumptions about the underlying structure of the data. Hence it is a non-parametric method. This flexibility enables the k-NN regression to adapt effectively to complex non-linear data. However, the k-NN regression has its share of complications. It is a complex task to decline the optimal 'k', because the lower 'k' might result in a model that is overly susceptible to noise whereas the larger 'k' might lead to over-smoothing the model and ignoring the key patterns in the data. Also, the k-NN regression may encounter difficulties with high-dimensional datasets, which is a problem known as the "curse of dimensionality".

The Gradient Boosting represents a potent machine learning method, which is applied across diverse regression and classification applications. Essentially, the gradient boosting regression is an ensemble technique that develops a predictive model by sequentially aligning a sequence of weak learners to the data. Each learner attempts to amend the mistakes of the one before it. Usually, these weak learners are decision trees. Of course, alternative base models can also be applied. The fundamental premise is to amalgamate the outputs of the multiple simplistic models in order to generate a singular, highly precise prediction. This "boosting" principle stems from the idea that the assembly of the weak learners when it is appropriately combined could be transformed into a robust learner. The progressive incorporation of weak learners in gradient boosting effectively acts as a method of the steepest descent. The algorithm approximates the gradient of the loss function (the metric that evaluates how accurately the model's predictions align with the actual values) with respect to the model parameters and how accurately it integrates the new models pointed in the direction that reduces the loss. For the regression tasks, the objective is to forecast a continuous outcome variable. In the context of the Gradient Boosting Regression the ensemble of the trees is trained to predict the residuals or errors of the prior trees. Consequently, each subsequent tree is effectively inching closer to the true, unknown function that we are trying to approximate. Of course, the Gradient Boosting has its merits and demerits like all models. It is an exceedingly effective technique which is capable to fit intricate and non-linear data. Frequently it performs well even on datasets with a blend of categorical and numerical features. However, the gradient boosting models can be susceptible to overfitting, especially if the data is noisy. They can also be computationally demanding and requiring a diligent tuning of several hyperparameters, such as the number of estimators, the depth of the trees, and the learning rate.

Random Forest is another renowned machine learning technique, commonly used in a variety of regression and classification tasks. The Random Forest model, as its name implies, is a conglomerate of individual decision trees each constructed to ensure a rich diversity that consequently guarantees resilience in their combined predictive capability. The Random Forest Regressor functions develop numerous individual decision trees for each regression tasks during the training phase. Each tree is learning from a random subset of the given data and is making its own prediction. When a new prediction is called for, each tree in the forest produces its own prediction. The final output is the average of these individual predictions. This methodology renders the Random Forest Regressor proficient at capturing intricate and non-linear relationships within data. The foundational concept of the Random Forest rests

on the proposition that a collective of "weak learners" can combine to form a "strong learner". Every decision tree within the forest is a weak learner trained on a random subset of the data using a random subset of the features at each bifurcation. Such technique is called a bootstrap aggregating or "bagging" along with feature randomness. The potency of the Random Forest is anchored in its simplicity and adaptability. It is relatively resistant to overfitting thanks to the randomness integrated into its creation. It handles both numerical and categorical data effectively, and also manages missing data proficiently. Additionally, it includes an inherent mechanism for assessing feature importance which can aid in deciphering the model. However, like all models, the Random Forest has its constraints. It can be computationally heavy and slower in both training and prediction, particularly as the number of trees escalates. Moreover, it may underperform with very high dimensional sparse data such as text data and can be less decipherable compared to the individual decision trees.

Extra Trees Regressor is an abbreviation for "Extremely Randomized Trees". It is another machine learning technique employed for regression tasks and it is a part of the ensemble learning class. Like the Random Forest, it creates multiple decision trees during the training phase. However, it adds an additional level of randomness and enhance its robustness and reduce its susceptibility to overfitting. In a conventional decision tree, the optimal split from a random feature subset in the node is chosen when expanding the tree. But in Extra Trees the random value is selected for the split for each considered feature rather than the best split point. Consequently, the Extra Trees Regressor injects randomness not only at the sample level but also at the level of each split in every decision tree. The prediction procedure in regression tasks mirrors that of the Random Forest Regressor i. e. each tree in the ensemble makes a prediction. So the final output is the average of these individual predictions. The randomness in the Extra Trees Regressor enables it to capture complex, non-linear relationships and rendering it a highly effective tool for regression. Nonetheless, it is crucial to distinguish the differences between the Extra Trees and the Random Forest methods. While they both utilize bagging and random feature subsets, the primary distinction lies on how they split the nodes. The Extra Trees method is quicker due to its randomness at each split. This results in faster training process. However, this might occasionally lead to a slight performance dip, indicating a trade-off to consider. The Extra Trees is less susceptible to overfitting. It is capable to manage both numerical and categorical data and can effectively handle the missing data. Similar to the Random Forest, it offers a built-in method for feature importance estimation. On the other hand, similar to other tree-based methods, it may not perform well with very high-dimensional sparse data like text data. Its ensemble nature makes it less interpretable compared to the individual decision trees.

In this work, we employ the four machine learning algorithms to create a multivariate regression. During the computing process we performed 10-fold cross-validation for each algorithm's indicators. Furthermore, we implemented a hyperparameter optimization routine to select the optimal values. Table 3 contains the optimal hyperparameters, and Fig. 1 illustrates the algorithm's flow.

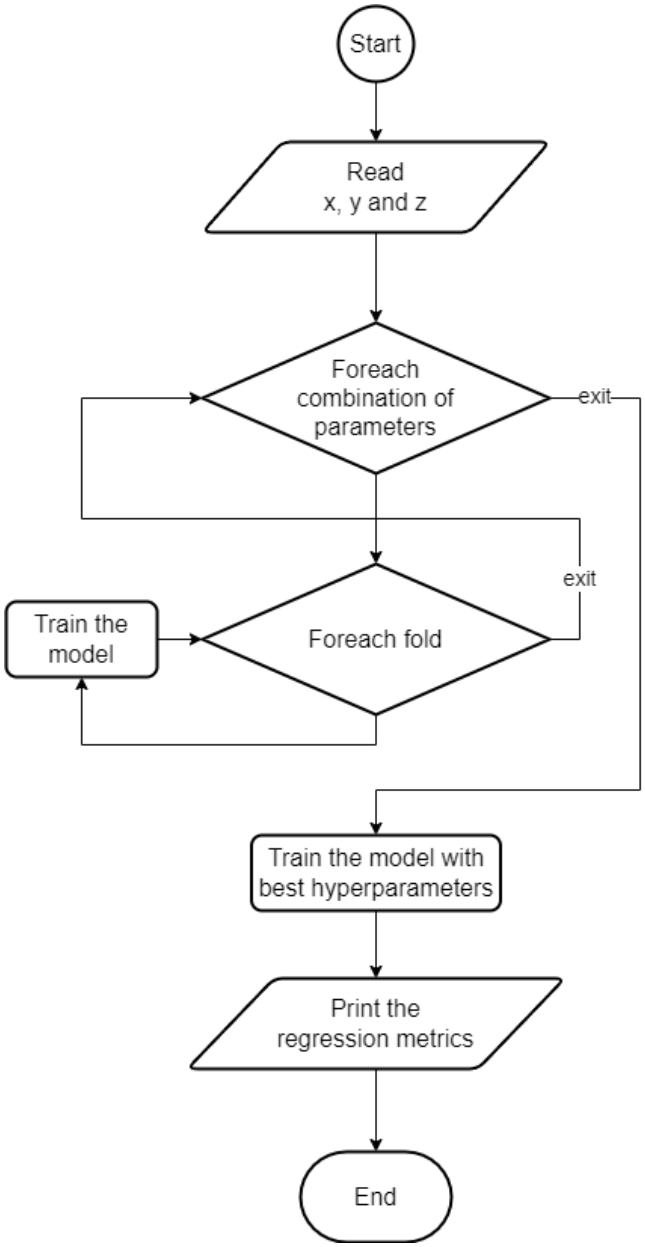

Fig. 1. Flowchart of the regression algorithm.

Cross-validation serves as a technique to gauge how the machine learning model is likely to perform on an unfamiliar dataset (i. e. the data is not previously encountered by the model). This method assists in preventing overfitting – a situation, where a model becomes overly specialized to the training data and performs poorly on some new unseen data. We utilize the most prevalent form of the cross-validation, known as k-fold cross-validation. For this method, the initial sample is randomly divided into k equally sized segments. Out of these k segments, a single one is kept as validation data for the model testing while the remaining k-1 segments serve as training data. This cross-validation procedure is repeated k times. Each time a different segment as validation data is used. The results from the k iterations can then be averaged (or otherwise merged) to yield a single measure of model performance.

On the other hand, hyperparameter optimization involves the locating of the best set of hyperparameters for a machine learning model. Hyperparameters are those parameters that the model

does not learn from the data but are preset before the learning procedure commences. For instance, in a Random Forest model the hyperparameters could consist of the count of decision trees in the forest or the count of the features each tree examines when partitioning a node. Unlike model parameters, the hyperparameters are preset before the learning process starts and remain static during the learning. Hyperparameters can have a significant impact on the model performance, hence the importance of the thoughtful selection. For our hyperparameter optimization, we employ a 'grid search' strategy where we define a set of possible values for each hyperparameter and then assess the performance of each possible combination.

Table 3: Modeling by using machine learning regressors.

Regressor	**Best hyperparameters**	**SSE**	**R-square**	**RMSE**
k-Nearest Neighbors	{'n_neighbors': 2, 'weights': 'uniform'}	351.0150	0.6397	5.6489
Gradient Boosting	{'max_depth': 2, 'n_estimators': 50}	7.9934	0.9918	0.8525
Random Forest	{'max_depth': 2, 'n_estimators': 50}	232.3306	0.7615	4.5958
Extra Trees	{'max_depth': 2, 'n_estimators': 100}	299.7497	0.6923	5.2202

In the second column of Table 3, the best combination of hyperparameters is shown for each method. The same metrics as in [14] are provided in order to compare them. Obviously, **Gradient Boosting** performs best. Its $\mathbf{R^2}$ is almost 1 and the errors are very small. Compared to Table 2, it is almost as good as the third-order polynomial Poly33. The other methods exhibit larger unacceptable errors. This approach could be successfully used to forecast the biological activity of new ligands via the optimization function GoldScore.

4. Conclusions

We conducted docking experiments to explore the interactions between delta-opioid ligands and DOR (PDBe: 1ozc). The established relationship between docking results and in vitro testing allowed us to predict the biological activity of newly synthesized analogs, exhibiting different activity than other compounds in the same series. This analysis provides deeper insights into the correlation between the biological effects of compounds and in silico experiments.

Furthermore, our investigation serves as a valuable tool to verify the alignment between the biological macromolecule (DOR) models and the actual three-dimensional molecular structure. The derived results equip us with the ability to anticipate the effectiveness of compounds with known docking scores and total energies. Such predictive capabilities will play a crucial role in shaping the direction of future studies.

By enhancing our understanding of these relationships and validating our models' accuracy, we are better positioned to make informed predictions about the biological activity of new compounds based on their docking scores and total energies. This valuable knowledge paves the way for future research and discovery in this field, potentially leading to the development of more effective pharmaceutical compounds.

Acknowledgments

This research is supported by the Bulgarian National Science Fund under Project KP-06-M62/1 "Numerical deterministic, stochastic, machine and deep learning methods with applications in computational, quantitative, algorithmic finance, biomathematics, ecology and algebra" from 2022.

References

[1] Rosenblum A., Marsch L., Joseph H., Portenoy R. 2008 *Experimental and clinical psychopharmacology* **16(5)**, 405–416 https://doi.org/10.1037/a0013628
[2] Portenoy R., Payne R., Passik S., Lowinson J., Ruz P., Millman R., Langrod J. 2004 *Acute and Chronic Pain,* 863–904 https://doi.org/10.1037/a0013628
[3] Valentino R, Volkow N 2018 *Neuropsychopharmacology* **43(13)** 2514-2520 https://doi.org/10.1038/s41386-018-0225-3
[4] Darcq E, Kieffer B. 2018 *Nat Rev Neurosci* **19(8)** 499–514 https://doi.org/10.1038/s41583-018-0028-x
[5] Kane B., Svensson B., Ferguson D. 2006 *The AAPS journal* **8(1)** E126-E137. https://doi.org/10.1208/aapsj080115
[6] Lord J, Waterfield A., Hughes J., Kosterlitz H 1977 *Nature* **267** 495-499 https://doi.org/10.1016/0024-3205(80)90312-4
[7] Maldonado R., Severini R., Hans W., Kiefer B., Melchiorri P., Negri L 2001 *British Journal of Pharmacology* **132** 1485–1492.
[8] Pradhan A., Befort K., Nozaki C., Gavériaux-Ruff C., Kieffer B. 2011 *Trends in pharmacological sciences* **32(10)** 581–590. https://doi.org/10.1016/j.tips.2011.06.008
[9] Le Merrer J, Becker J, Befort K, Kieffer L 2009 *Physiol Rev*. **89(4)** 1379–1412 https://doi.org/10.1152/physrev.00005.2009
[10] Morgan M, Christie M 2011 *British journal of pharmacology* **164(4)** 1322–1334 https://doi.org/10.1111/j.1476-5381.2011.01335.x
[11] Dorahy G., Chen J., Balle T. 2023 *Molecules* **28** 1324. https://doi.org/10.3390/molecules28031324
[12] Sapundzhi F, Dzimbova T. 2019 *International Journal of Online and Biomedical Engineering* **15(15)** 39–56 https://doi.org/10.3991/ijoe.v15i15.11566
[13] Sapundzhi F., Dzimbova T., Milanov P., Pencheva N., 2015 *J. Comput. Methods Molec. Design*, **5 (2)**, 98-108.
[14] Sapundzhi F 2019 *Bulgarian Chemical Communications* **51 (4),** 569-579.
[15] Sapundzhi F, Prodanova K, Lazarova M 2019 *AIP Conference Proceedings*, 2172, 100008 1-6
[16] Sapundzhi F 2019 *International Journal of Online and Biomedical Engineering* **15(11),** 39-145.
[17] Jones G, Willett P, Glen R, Leach A, Taylor R 1997 *J. Mol. Biol.* **267**, 727-748. https://doi.org/10.1006/jmbi.1996.0897
[18] Pencheva N., Bocheva A, Dimitrov E., Ivancheva C 1996 *Eur. J. Pharmacol*. **304** 99-108.
[19] Pencheva N., Milanov P., Vezenkov L., Pajpanova T., Naydenova E. 2004 *Eur. J. Pharmacol.* **498** 249-256.
[20] Traykov M, Trenchev I, 2016 *Genetika* **52(9)**, 1089–1096, https://doi.org/10.7868/s0016675816080130
[21] Russel S, Norvig P 2020 *Artificial Intelligence: A Modern Approach* 4th Ed Pearson USA
[22] Kelleher J, Namee B, D'Arcy A 2020 *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies* 2nd Ed The MIT Press USA
Nedyalkov I, Stefanov A, Georgiev G, 2018 *IX National Conference with International Participation (ELECTRONICA)*, Sofia, Bulgaria, 1-4, https://doi.org/10.1109/ELECTRONICA.2018.8439127