



# Neural networks in the design of molecules with affinity to selected protein domains.

Sieci neuronowe w projektowaniu cząsteczek z powinowactwem  
do wybranych domen białkowych.

2022

Damian Aleksander Nowak

433724

Field of study: General chemistry

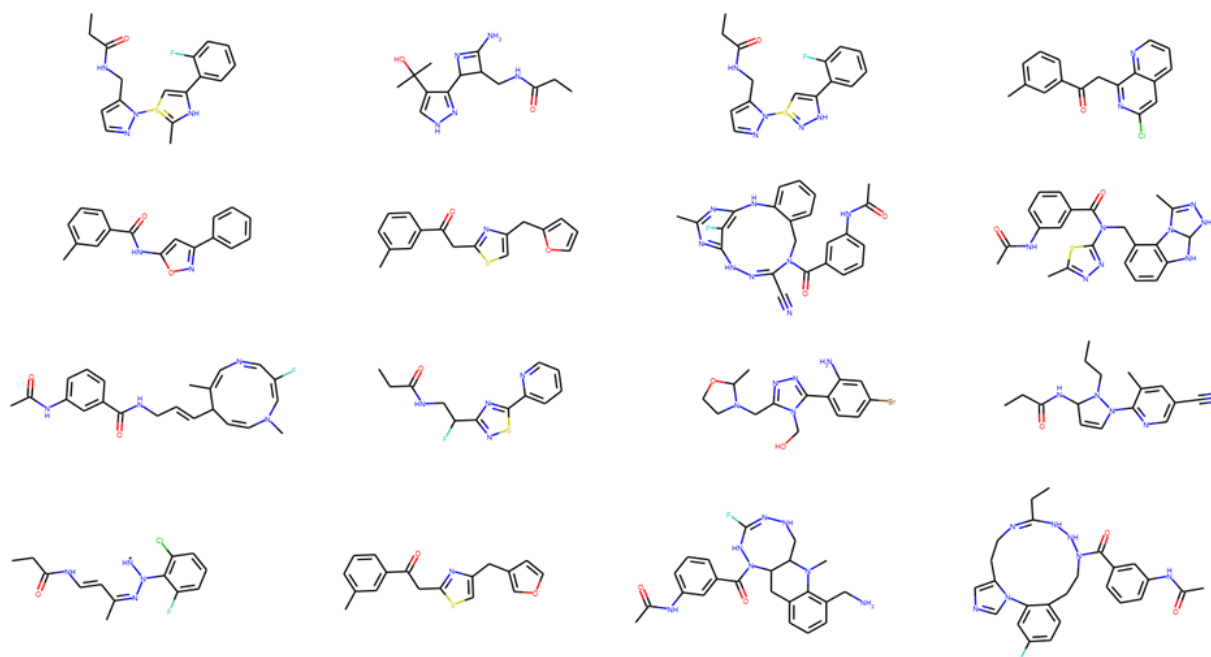
Specialty and research group: computational chemistry, Prof. Marcin Hoffmann's group

**Abstract:** Drug design via machine learning can speed up new drug discoveries. While current databases of known compounds are magnitudes of orders smaller (approximately  $10^8$ ), the number of small drug-like molecules is estimated to be between  $10^{23} - 10^{60}$ . The use of molecular docking algorithms can help in new drug development by sieving out the worst drug-receptor complexes. New chemical spaces can be efficiently searched with the application of artificial intelligence. From that, new structures can be proposed. The paper given below aims to create new chemical structures via a neural network that will possess affinity to selected protein domains. Transferring chemical structures into SELFIES code allowed us to pass chemical information to a neural network. On the basis of vectorized SELFIES new chemical structures can be created. With the use of the model, novel compounds, that are chemically sensible, can be generated. Newly created chemical structures are sieved by the Quantitative Estimation of Drug-Likeness descriptor and the SYnthetic Bayesian Accessibility classifier score. The affinity to selected protein domains has been checked with the use of the AutoDock tool. The results of the paper are the structures that possess affinity to selected protein domains (see **Visualization 1**).

**Keywords:** machine learning; neural networks; molecular docking; ROR- $\gamma$ ; drugs design

**Streszczenie:** Projektowanie leków za pomocą uczenia maszynowego może przyspieszyć odkrywanie nowych specyfików. Podczas gdy obecne bazy danych znanych związków chemicznych są o rzędy wielkości mniejsze (około  $10^8$ ), liczba małych cząsteczek podobnych do leków szacowana jest na  $10^{23} - 10^{60}$ . Zastosowanie algorytmów dokowania molekularnego może pomóc w opracowywaniu nowych leków poprzez odsiewanie najgorszych kompleksów lek-receptor. Dzięki zastosowaniu sztucznej inteligencji można efektywnie przeszukiwać nowe przestrzenie chemiczne. Na tej podstawie można zaproponować nowe struktury. Poniższa praca ma na celu stworzenie nowych struktur chemicznych za pomocą sieci neuronowej, które będą miały powinowactwo do wybranych domen białkowych. Przeniesienie struktur chemicznych do kodu SELFIES pozwoliło nam na przekazanie informacji chemicznej do sieci neuronowej. Na podstawie zwektoryzowanego kodu SELFIES można tworzyć nowe struktury chemiczne. Za pomocą tego modelu można generować nowe związki, które są sensowne chemicznie. Nowo utworzone struktury chemiczne są przesiewane przez deskryptor Quantitative Estimation of Drug-Likeness oraz wynik klasyfikatora SYnthetic Bayesian Accessibility. Przynależność do wybranych domen białkowych sprawdzono za pomocą narzędzia AutoDock. Wynikiem pracy są struktury, które wykazują powinowactwo do wybranych domen białkowych (patrz **Wizualizacja 1**).

**Słowa kluczowe:** uczenie maszynowe; sieci neuronowe; dokowanie molekularne; ROR- $\gamma$ ; projektowanie leków



**Visualization 1.** Generated structures that have affinity to selected protein domains.

## Table of Contents

<b>Introduction.....</b>	<b>2</b>
<b>Macromolecules .....</b>	<b>3</b>
<b>Selected protein domains.....</b>	<b>5</b>
<b>ROR-<math>\gamma</math> receptors characterization .....</b>	<b>6</b>
<b>Molecular docking preparation.....</b>	<b>7</b>
<b>SMILES and SELFIES molecular representations .....</b>	<b>8</b>
<b>Neural networks .....</b>	<b>9</b>
<b>Neural networks – selected parameters .....</b>	<b>11</b>
<b>Docking methods.....</b>	<b>12</b>
<b>Aim of study .....</b>	<b>13</b>
<b>Molecular docking algorithm description .....</b>	<b>13</b>
<b>Accuracy of AutoDock.....</b>	<b>16</b>
<b>Methodology .....</b>	<b>17</b>
<b>SELFIES coder .....</b>	<b>19</b>
<b>Environment preparation .....</b>	<b>19</b>
<b>Training data selection .....</b>	<b>19</b>
<b>Model.....</b>	<b>21</b>
<b>Data to predictions.....</b>	<b>25</b>
<b>Predictions .....</b>	<b>26</b>
<b>Similarity .....</b>	<b>28</b>
<b>Molecular docking .....</b>	<b>29</b>
<b>Results and discussion .....</b>	<b>30</b>
<b>Model.....</b>	<b>30</b>
<b>Data to predictions.....</b>	<b>31</b>
<b>Predictions and results of filtration.....</b>	<b>33</b>
<b>Similarity to initial structures and training data .....</b>	<b>38</b>
<b>Molecular docking of selected structures .....</b>	<b>41</b>
<b>Conclusions.....</b>	<b>45</b>
<b>Equations and methods .....</b>	<b>47</b>
<b>Attachments.....</b>	<b>54</b>
<b>Files .....</b>	<b>54</b>
<b>Acknowledgments .....</b>	<b>55</b>
<b>References.....</b>	<b>55</b>

## Introduction

Designing a molecule that can effectively bind to a target protein domain is essential in a drug discovery process [1, 2]. Computational methods are able to speed up screening, in a virtual manner, which allows researchers to reduce excessive costs and the long time that are necessary during the conduction of experimentally based techniques so-called in vitro studies [3].

The use of a neural network helps speed-up obtaining molecules that are similar to already known molecules of desired properties. Artificial intelligence enables obtaining new bioactive compounds derived from already known molecules via modifications necessary to better fit the pharmacological purpose based on molecular descriptors [3].

Crystallography and multidimensional nuclear magnetic resonance (NMR) [4] provide much structural information deposited in the protein data bank (PDB) [5] that can be used during a search for interactions between a newly designed potential drug and selected macromolecules.

Currently, the methods employed by most popular programs are assuming a ligand (small molecule) flexibility, and rigidity of a receptor. This leads to cost and time reduction. These programs (AutoDock [7], Flex [7], DOCK [7], GOLD [7], ICM [7], Glide [7], Ligand Fit [7], and others [7]) are fitting small molecules to a protein [7]. Molecular dynamics simulation can be used to dock ligands in a more flexible model, this approach enables seeing how flexible a potential drug is in relation to a binding flexible macromolecule. This approach is going to be used more widely as the computers will be getting more computational power and their costs will be reduced [7].

During the last decades variety of docking programs, have been developed for either academic or commercial use (AutoDock, Flex, Surflex, DOCK, GOLD, ICM, Glide, LigandFit, and many others). Different solutions, and strategies are exploited in the ligand placement in the protein. Four categories can be listed: stochastic Monte Carlo (Glide), fragment-based (Surflex, Flex), evolutionary-based (GOLD, AutoDock) and shape complementary methods (LigandFit). Systematical search is not used due to the impossibility of exploration of all degrees of freedom during molecular docking procedures. It is due to enormous computational costs. For example, if one is going to examine the cubic active site of  $10^3 \text{ \AA}^3$  with a simple ligand and when energy evaluation is done every  $10^\circ$  (change of the angle between the small molecule and receptor) as well as a rigid movement

every 0.5 Å [7] for a drug with four rotatable bonds only – there are  $6 * 10^{14}$  [7] conformations to be checked. If our computer is fast enough to compute 1000 conformations per second whole procedure would take 19025 years to finish this systematical approach [7].

The studies presented in this dissertation aimed at proposing new molecules which may be a possible ligand against selected protein domains (ROR- $\gamma$ ). This was achieved by the harnessing of artificial intelligence that handled new potential drug generation connected with a chosen molecular docking program (AutoDock [8]), which is responsible for checking the effectiveness of the binding of the new ligand to the chosen receptor.

## Macromolecules

Macromolecules are gigantic objects, such as proteins. They are constructed of thousands of covalently bonded atoms. A large group of them is composed of smaller parts called monomers. For purposes of this thesis biopolymers in general and proteins in detail are used. According to the IUPAC, a macromolecule is a molecule of high relative molecular mass or a structure arranged from multiple repetitions of units that have relatively low molecular mass [9]. From a chemical point of view, some macromolecules such as globular proteins or carbohydrates are soluble in water, on the other hand, fibrous proteins are not. Proteins often form the hydrophobic core and hydrophilic shell. Proteins show the possibility of denaturation, the process in which a highly ordered structure is destroyed (quaternary, tertiary, secondary structure). It can be a result of external stress coming from the presence of strong acid, base, inorganic salt, or organic solvents in the protein solution, radiation, or heat. Temperature, to some optimal point, leads to an increase in protein activity, and below this point, macromolecule starts to be denatured, and a decrease in activity in living organisms can be seen [10].

Proteins are built from twenty different amino acids (the smallest polymer units in this case) and their creation is done by ribosomes that interpret mRNA ciphered by codons in the gene and assemble the deciphered bricks of macromolecules in a process called translation [11]. The primary structure is created, at this stage – the linear polypeptide chain. A less complicated, nevertheless important, level of organization combines one amino acid alpha-carboxyl group with the alpha-amino group of the second in a peptide bond, a water molecule is eliminated during the formation of the peptide bond. Connected amino acid rests form polypeptides and their chains are built from regularly repeating parts which are named

the main chain (or backbone) and side chains are attached to the major branch. This structure owns a high affinity to create hydrogen bonds [11].

Secondary structure (alpha helix or beta-sheet) requires hydrogen bond formation between peptide group chains. In the case of alpha-helix, a tightly coiled backbone forms the inner part of the rod, and the rod is stabilized by hydrogen bonds that are formed between NH and CO groups of the major branch. Beta sheets are becoming stable due to hydrogen bonding between polypeptide strands. The antiparallel and parallel arrangements can be listed. The former is simpler, and NH and CO groups are created hydrogen bonds with adjacent partners present in the second chain. The latter is more complicated; the NH group is hydrogen-bonded with one CO part of the molecule and the CO part of the residue is connected with hydrogen that belongs to the NH part of the residue which is two amino acids farther along the chain [11].

Then folding of an alpha helix or a beta sheet takes place – a tertiary structure is formed. The formation of the hydrophobic core and hydrophilic cover can be seen at this stage. The most complicated three-dimensional structure consisting of multiple polypeptides is the quaternary structure. It is assembled from subunits, with more than one polypeptide chain. Each polypeptide chain is called a subunit. Spatial arrangements of subunits are covered by the quaternary structure [11].

The class of macromolecules accustomed in this paper are well-prepared for being a catalyst. These are responsible for catalyzing the biochemical reactions that nurture life. One of the roles that are assigned to proteins is the enzymatic role which can be found, for example, in the digestion enzyme in our stomachs that helps us to break down proteins in food, this protein is called pepsin [11].

Another role assigned to proteins is related to the immune system that produces antibodies (proteins) which are helpful in removing foreign substances and fighting infections.

The next function is played by DNA-associated proteins, which regulate chromosome structure [11] during cell division and moderate gene expression, for example, histones and cohesion proteins [11].

Other tasks related to the described group of macromolecules are muscle contraction and movement (actin and myosin), structural role (collagen, elastin), hormone proteins that co-ordinate functionalities of our bodies, for example, the insulin that controls our blood sugar concentration by regulating the uptake of glucose into cells. Last, but not least function is transportation, in this case, hemoglobin can be an example, it transports oxygen [11].

## Selected protein domains

Protein domains that are targets of this study belong to ROR-gamma receptors (NR1F3). They are called orphan receptors due to fact that exact natural ligands are not yet exactly known [12] [13].

These are a group of nuclear receptor transcription factors that possess an affinity to bind with DNA. Their activity is regulated by ligands. Ligands, in general, are small molecules that exhibit biological activity [13].

Given macromolecules are responsible for metabolism regulation, whole-body development, cell apoptosis, the persistence of homeostasis, or even circadian rhythms control, not all their functions are currently defined [13].

In our organisms, we can find them inside our lungs, liver, kidneys, and muscles [14].

Substances that show activity against ROR-gamma receptors are oxysterols, in which desmosterol has high affinity and it is a precursor of cholesterol, mainly activators of receptors, cardiac glycosides (generally deactivators), steroids [15], and inhibitors of kinases [16].

A ligand that is bound to the receptor leads to a change in protein conformation and in that manner influences protein activity.

These receptors are currently objects of interest due to fact that antagonism of drugs that can be efficiently docked to them may lead to application in inflammatory treatments, multiple sclerosis, rheumatoid arthritis, or psoriasis [15].

For a better understanding of different compounds that own affinity against ROR-gamma receptors, one should notice several types of drugs. Agonists [17] are a group of compounds that lead to creating interactions with macromolecule and as a result of that receptor functionality is activated. An inverse agonist [17] is called a compound that deactivates the receptor's functionality. Antagonists [17] are one group that can be further divided into competitive and non-competitive. In general, antagonists are drugs that ease agonists' influence. Competitive antagonists [17] bind in the same place as agonists do and by doing this receptor's active site is blocked so that no agonist can be inserted – the macromolecule's activity is limited. Non-competitive antagonists [17] bind to a place that is not reserved for agonists. It results in protein conformation change and that is how drugs can moderate the receptor's activity. Reversible antagonists [17] can be easily removed from the ligand-receptor complex, due to weak bonds only. Irreversible antagonists [17] cannot

be flushed out without difficulties due to stronger bonds created – bonds of covalent nature can be formed, and it results in the challenge of ligand detachment.

Already known compounds that are known to be active against ROR-gamma receptors with ligand classification: digoxin [18] – cardiac glycoside, inverse agonist – inhibits macromolecule functionality; ursolic acid [18] – steroid precursor, inverse agonist; ouabain [18] - cardiac glycoside, inverse agonist, HC9 – (22R-Hydroxycholesterol) – oxysterol, agonist – promotes macromolecule functionality, HC3 (25-Hydroxycholesterol) – oxysterol, agonist; stearic acid – fatty acid, antagonist – occupy the same active site as an agonist. An anti-cancer drug, LYC-55716 (called cintirotogon). It is used as an oral drug which is a selective agonist of ROR-gamma receptors [19]. Other studies found that AZ 5104 can be active against ROR-gamma receptors [52].

All of the substances listed above can play their roles by the interactions between ligand and receptor – these are crucial to moderate the operability of the macromolecule. They are necessary to form a stable ligand-receptor complex which should be composed of hydrogen bonds and other weak intermolecular interactions. It is so due to the possible reuse of macromolecule later, in the case of strong (covalent) bonding of ligand inside a receptor further usability of protein is excluded [17].

## **ROR- $\gamma$ receptors characterization**

The protein domains used as receptor targets in this paper are 7NPC, 7NP5, and 7KXD respectively, their names are related to the PDB database infrastructure.

The choice was done according to the given requirements:

- all of them are belonging to the ROR- $\gamma$  family
- they have a similar and good resolution of collected data
- each structure has one main chain
- these domains data were collected with the use of X-ray crystallography
- their publications time is similar (the year 2021)

7NPC – ROR- $\gamma$  nuclear receptor, a protein in which crystallographic data were collected with a resolution of 1.47 Å. Its publication date was 2021-02-26 [6, 20].

7NP5 – ROR- $\gamma$  nuclear receptor, a protein in which crystallographic data were collected with a resolution of 1.55 Å. Its publication date was 2021-06-02 [6, 20].

7KXD – ROR- $\gamma$  nuclear receptor, a protein in which crystallographic data were collected with a resolution of 1.62 Å. Its publication date was 2021-01-20 [6, 21].



## Molecular docking preparation

Active sites [22] are crucial in the case of moderation of macromolecule activity. This is the place where the ligand is placed and where the ligand-receptor complex is formed. This part can be defined from previously done experiments; it allows us to reduce the time of calculations due to the limited volume of macromolecule that will be searched. If this space is not known we should include a greater volume of protein to search in, but the calculations will take longer time. To maintain this step, AutoDock tool [8] is used. These centers are figured out by searching for the minimal value of energy in a complex that is formed between ligand and receptor by next attaching the ligand to the receptor. The lower the energy, the better the binding [8].

Macromolecules must be appropriately prepared to be used in molecular docking procedures. All water molecules must be removed as the ligand from the raw file downloaded from Protein DataBase. Only protein should be left. Then acidic hydrogens are added, which means those which are connected to oxygen or nitrogen or those that are potentially H-bonding hydrogen atoms. The next step is to distribute charges on the macromolecule; in this case Kollman charges [23, 24] for the macromolecule. Charges are calculated to obtain possible hydrogen bonds, during the molecular docking procedure. The following step is to determine a grid box inside which a ligand will be attached to the macromolecule and binding energy will be estimated [8].

Ligand has to be prepared similarly, it means that acidic hydrogens need to be inserted and Gasteiger charges should be calculated [23].

Kollman charges are template values for each amino acid that were derived from the corresponding electrostatic potential using quantum mechanics [23, 24].

Gasteiger charges are calculated on basis of electronegativity equilibration computed by AutoDockTools for a specific molecular system [23, 25].

Molecular docking can be done after the above steps of preparation. At this stage, the ligand will be quasi-randomly attached to the macromolecule and binding energy will be assessed. Previously prepared receptor and ligand are loaded into AutoDockTools [8] then the force field is generated by this tool. After this step, search parameters are chosen such as the algorithm of search, in the case of this study, it is a genetic algorithm, and the output file is generated with the use of the Lamarckian genetic algorithm option [23].

A genetic algorithm is a process derived from natural selection that belongs to a wider class of evolutionary algorithms (EA), this approach leads to effective results in better time

than when molecular docking is done without any restrictions and any learning during the entire process. Hopefully, a global minimum may be located. This procedure involves optimization and search problems by using mutations, crossovers, and selections during the propagation of the procedure. Better results can be obtained, since during molecular docking GA follows the movement of ligand in defined search space and finds the area of the greatest surface complementarity between the two. This procedure tries to perform a global search [26].

Lamarckian genetic algorithm is a hybrid algorithm that connects a genetic part with a local search that has enhanced performance compared to a genetic algorithm. In this approach, each generation is followed by a local search on a defined population [27].

## **SMILES and SELFIES molecular representations**

SMILES (Simplified Molecular Input Line Entry Specification) [28] codes allow us to encode chemical structure in a string of characters such as CCO which stands for ethanol molecule. This approach makes our research easier, because working on characters is simpler for computers than working with graphical representation. It has some limitations, such as carrying information only about the two-dimensional structure of molecules encoded via SMILES, but three-dimensional information can be calculated later based on generated two-dimensional structure. The 3D structure will be necessary for the molecular docking procedure.

SELFIES - (SELF-referencIng Embedded Strings) [29] codes can stand for chemical structures similarly to SMILES codes, but SELFIES are more robust against character mutations while still preserving similar chemical properties. Even randomly generated SELFIES produces much more semantically valid structures than SMILES. As feature representation in variational autoencoders, SELFIES provide a substantial improvement in the task of reconstruction, validity, and diversity. SELFIES allow for direct applications in machine learning, without the need for domain-specific adaptation of model architectures. Example of SELFIES code: ethanol - [C][C][O], but in this example complexity of SELFIES cannot be presented. Look at SMILES for desmosterol molecule: C[C@H](CCC=C(C)C)[C@H]1CC[C@@H]2[C@@]1(CC[C@H]3[C@H]2CC=C4[C@@]3(CC[C@@H](C4)O)C)C, where @ indicates S chirality marker and @@ indicates R chirality marker. Translated SELFIES code based on desmosterol is described by the given string

[C][C@H1][Branch1][=Branch2][C][C][C][=C][Branch1][C][C][C][C@H1][C][C][C@H1][C@H1][C@H1][C@H1][Ring1][Branch1][Branch2][Ring1][=Branch2][C][C][C@H1][C@H1][Ring1][=Branch1][C][C][=C][C@H1][Ring1][=Branch1][Branch1][#Branch2][C][C][C@H1][Branch1][Ring2][C][Ring1][=Branch1][O][C][C].

This maintenance of chemical information carries a more detailed description of the compound. SELFIES encoder should be fed with valid SMILES code, and it can be done by the RDKit library [30]. When the new SELFIES code is generated by a neural network it can be decoded into SMILES again, the compound will always be syntactically and semantically correct. The newly obtained potential drug is further converted into a three-dimensional structure that can be used during the molecular docking procedure. Previously Tanimoto similarity [31] to initial compounds is calculated due to the necessity of sieving out of the same structures as starting ones. Estimation of binding energy between new ligand and ROR- $\gamma$  receptor is done. On the basis of this calculation choice of potential drugs can be carried out.

## Neural networks

Neural networks are subsets of machine learning algorithms whose name and structure are inspired by the human brain; in that way, they mimic biological neurons that are signaling to one another. This definition gives a general overview of them [32].

The idea of artificial neural networks has been created by neurophysiologists Warren McCulloch and mathematician Walter Pitts in “*A Logical Calculus of Ideas Immanent in Nervous Activity*.” It was done in 1943 when the first architecture of this type of network, has been described [32].

The main limitations and reasons why this method was difficult to develop at those times were: negligible quantity of data available and therefore of potential applications, computational power, which is magnitudes of power greater currently - according to Moore’s equation, not well-defined algorithms, only limited number of people were interested in it [32].

This kind of getting knowledge from the data is powerful, scalable and, can be widely applied. Those enormously large and complicated datasets can be analyzed with the usage of them [32].

Fields of applications are related to classification – image, numerical, text, speech recognition, recommendation systems, learning of algorithms how to play, and, in the case of chemistry, de novo structure generation [32, 1, 2, 33].

Every neural network works and predicts with a base of mathematics (numbers). Data that are interesting to us should be converted into mathematical tensors and that way objects such as pictures, voices, or molecules can be described and effectively transferred to the network [32].

Artificial neurons, according to McCulloch and Pitt's idea, has at least one binary (0 or 1) input and only one binary output. The outcome can be activated when a certain number of inputs are activated [32].

Perceptron can also be distinguished as a modified artificial neuron, in which inputs and outputs are numbers, not binary states, and every connection between them has an assigned weight. In this case, a weighted sum of input signals is calculated and then the Heaviside step function (or signum function) is used on the previously computed sum. In that manner, the final output is formulated [32].

Let us take a closer look at the architecture of neural networks in general and deep neural networks in specific. This is applied in this paper. It refers to a network that owns not only input and output layers, but also multiple hidden layers. Their order is the following: firstly, the input layer is given then hidden layers occur, and at the end output layer. Nodes are connected to one another in the order given above, and these connections are associated with weight and threshold [32].

If the next node is activated, it means that output from previous data was above the threshold and deactivation takes place in case of a lower result. In that manner knowledge, in form of data, is transferred to the next layer of the network [32].

Each node can be thought to be a linear regression model which is constructed on basis of input data, weights, a bias (or threshold), and an outcome, see **Equation 11** [32].

Weights are useful for assignment and determination of the importance of any given variable. The larger the weight the more significant the contribution to the output [32].

The result from each node is passed through an activation function and this can lead to the activation or deactivation of the next node. This process is called feedforwarding [32].

The recurrent neural network (RNN) is one type of artificial neural network that is prepared and used to work with sequences and to predict as a sequence. They are derived from feedforward networks. They are widely applied in natural language processing

and with the usage of long-short term memory cells are able to learn rules that are included in given data [32].

LSTM (see **Method 2**) network is a type of RNN, so this type of unit has recurrent connections and in that manner state from the previous activations of the neuron from the earlier time step is used as a context for output formulation. They are composed of weights and gates. Three types of gates are given input, output, and forget gate, and their role is writing – which decides which values from the input are used to update the memory state, reading – which decides what to output based on input and the memory of the cell and resetting – decides what to discard from the cell, respectively. The network can interact with cells only via gates. The following weights are present: input, output, and internal state, first is used to weight input for the current time step, the second to weight output from the last time step, and the third is used for the calculation of the output for this time step [34]. The gates control cell internal state  $C$  and the state  $H$  is passed as a copy between the recurrent iterations, and it is crucial to choose what to do with the internal state by changing, closing off, inputting, and forgetting gates until a new condition or input opens them. In that manner, in drop-in replacements, RNN can recognize long-term dependencies [34].

### Neural networks – selected parameters

Loss measurement is the computation of how accurate the model on training data is. This should decrease in time, during a training procedure. For this purpose, for this experiment categorical cross-entropy, see **Equation 7**, is used to get loss and validation loss (the same as loss, but for the unseen data during training) [35].

Activation functions – a variety of different activation functions can be distinguished. Those used in this work are tanh [36], ReLU [37], and SoftMax [36]. They are the last just before the output from the node. Inputs and weights are multiplied by themselves and also bias is added. Then the activation function is applied to the previous result, and after that, the output is passed to the next layer. In that manner, we are going to the last layer [37].

Tanh activation function [36], see **Equation 3** is helping in non-zero centered problem-solving. Its results in range  $[-1,1]$ , it is a non-linear function.

ReLU – rectified linear unit [37], see **Equation 4**, is mostly used in the hidden layers of neural networks. It avoids the vanishing gradient problem. If activations are in the  $<0$  regions there is a danger of the gradient being 0 due to weights that will not get adjusted during optimization.

SoftMax activation function [36], see **Equation 5**, is mostly used at the last layer which calculates the probabilities distribution of the event over “n” different events, it can manage multiple classes.

The optimizer used here was the so-called Adam [38], see **Equation 6**. It is used as a replacement optimization algorithm for stochastic gradient descent for training artificial learning models. Here is a single learning rate (called alpha) for all weight updates and the learning rate does not change during training – the learning rate is maintained for each network weight and separately adapted as learning unfolds.

## Docking methods

As mentioned earlier there are many molecular docking approaches, but all of them are trying to solve the same issue, found optimal ligand pose within receptor conformation that will lead to the minimization of energy of the whole complex [39].

The conformations can give information about possible intramolecular interactions such as hydrogen bonds, electrostatic free energy, torsional free energy, dispersion, repulsion, desolvation total internal energy, and unbound system energy. All of them are summed up in the case of raw structure. After molecular docking, energy of binding is given. It is calculated as a difference between the raw macromolecular system energy and the energy of a complex consisting of macromolecule and ligand [8, 23, 39].

The simulation approach assumes the possibility of ligand binding into a groove target after a given number of runs in its conformational space. During the procedure, torsional angle rotations and translations can be done [39].

The shape complementarity approach assumes ligands and targets surface structural features that provide their molecular interactions. The surface of the target is shown concerning its solvent-accessible surface area and complementarity between two surfaces based on shape matching illustration helps in searching for the complementary groove for a ligand on the target’s surface [39].

A variety of different algorithms can be used for searching, they are genetic algorithms, fragment-based algorithms, Monte Carlo algorithms, and also molecular dynamics algorithms [39].

Molecular docking can be done in many modes, ligand/target can be flexible in some, and in others, the rigidity of them is imposed.

## **Aim of study**

The main aim of the examination is to check the possibility of new chemical structures' generation with the application of artificial intelligence: neural networks architecture, in this case. The machine learning model should be trained on how chemical structures are constructed. When the model has some chemical knowledge, it can generate chemically correct structures. The output is the SMILES code, which is extremely useful because it easily goes ahead further [1, 2].

The second aim is to conduct molecular docking via an automatic path done by python code. This solution leads to more effective energy binding calculation when the whole of the process can be done via computer script and many potential ligands can be checked inside specific macromolecule and their interactions can be compared to one another [46].

The use of IT tools makes molecular screening enhancement due to avoidance of the necessity of manual preparation of a ligand and a given receptor. This approach makes it more efficient, and many systems can be studied [46].

After the whole procedure, we obtained the best structures, which have the most favorable binding energies (lowest in values). They may be later synthesized, and experimental results recorded [18].

In the manner described above searching for new drugs can be boosted and the selection of structures that will most likely exhibit desired properties are selected.

## **Molecular docking algorithm description**

AutoDock is an automated procedure that allows users to predict interactions between ligand and a macromolecular target.

Some steps can be distinguished, the proper order is the following: preparation of coordinate files, precalculation of atomic affinities, docking of ligands, and analysis of results [8, 23].

The first step allows the user to prepare a model of a ligand and protein that includes polar hydrogen atoms, without hydrogen atoms bonded to carbon atoms. This tool takes advantage of the extended \*.pdb file format, so-called \*.pdbqt inside which several types of information are stored: coordinates, atomic partial charges, atom types, and torsional degrees of freedom. The used force field can distinguish between aliphatic and aromatic carbon atoms and also

separate types of polar atoms that can form hydrogen bonds and those which are not able to do that [8, 23].

When coordination files are prepared next step can be performed. The atomic affinity potentials are precalculated for each atom type present in a ligand structure. AutoGrid tool, which comes along with AutoDock Toolkit is used and it embeds the protein in a three-dimensional grid and a probe atom is placed at each grid point. The energy of interaction between a single atom and the protein is assigned to the grid point. These kinds of affinity grids are calculated for each type of atom in the ligand. Also, electrostatic and desolvation potentials are calculated. Then the energy of a certain ligand configuration is evaluated with the use of the values from the grids [8, 23].

The further step is to dock the ligand to a macromolecule. This action is conducted employing the Lamarckian genetic algorithm (LGA). It is done several times to give more docked conformations, and due to that further analysis can be done to identify the best pose [8, 23].

The last step is the analysis that visualizes the conformation with interactions between the ligand and macromolecule and displays the affinity potentials created by the AutoGrid tool [8,23].

The free energy scoring function given by AutoDock is computed with a semi-empirical free energy force field that evaluates conformations during the molecular docking procedure. The force field has been parametrized with the usage of a large number of protein-ligand systems for which the structure and inhibition constants are known. Binding is evaluated in two steps where ligand, and protein begin in an unbound state. Then intramolecular energies are estimated for the transition from unbound form to the bounded state. The further step is to evaluate the intramolecular energetics of the combination of the ligand and protein in their bound complex [8, 23].

The force field employs six pair-wise evaluations of potential energy (V) and estimation of the conformational entropy lost upon binding ( $\Delta S_{\text{conf}}$ ) – see **Equation 1**.

For each of the energetic V, terms (see **Equations 1 and 2**) are including evaluations for dispersion, repulsion, hydrogen bonding, electrostatic, and desolvation.

Weighting constants W, see **Equation 2**, have been optimized based on empirical free energy based on a set of experimentally determined binding constants.

First-term,  $W_{\text{vdw}} \sum_{i,j} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right)$ , is a typical 6/12 potential for dispersion/repulsion interactions [4, 23], the Lennard-Jones Potential [40].



Second,  $W_{\text{hbond}} \sum_{i,j} E(t) \left( \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right)$ , describes input from H-bond based on 10/12 potential. Parameters C and D are assigned to give the maximal energy outcome for hydrogen-oxygen and nitrogen H-bonds, which is about 5 kcal/mol at 1.9 Å length and with an energy of about 1 kcal/mol when an H-bond with sulfur is formed at 2.5 Å in length. Function E(t) provides energy change based on the angle t from ideal H-bonding geometry [4, 23].

Third term,  $W_{\text{elec}} \sum_{i,j} \frac{q_i q_j}{e(r_{ij}) r_{ij}}$ , describes screening Coulomb potential for electrostatics [8, 23, 40].

Fourth,  $W_{\text{sol}} \sum_{i,j} (S_i V_j + S_j V_i) e^{\left( \frac{-r_{ij}^2}{2\sigma^2} \right)}$ , so-called desolvation potential which is calculated on the volume of atoms (V) that are surrounding certain atom and shelter it from solvent, S parameter is used there as a weight. Also, an exponential term can be found, it is related to distance-weighting input and is given by  $\sigma$ , and equals 3.5 Å [8, 23].

**Equation 1.** The binding free energy calculation [23]

$\Delta G = (V_{\text{bound}}^{\text{L-L}} - V_{\text{unbound}}^{\text{L-L}}) + (V_{\text{bound}}^{\text{P-P}} - V_{\text{unbound}}^{\text{P-P}}) + (V_{\text{bound}}^{\text{P-L}} - V_{\text{bound}}^{\text{P-L}} + \Delta S_{\text{conf}})$ , where L refers to ligand and P indicates protein in the docking energy calculation. Each V parameters has unit kcal/mol.

**Equation 2.** Energetic terms calculation [23]

$$V = W_{\text{vdw}} \sum_{i,j} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right) + W_{\text{hbond}} \sum_{i,j} E(t) \left( \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right) + W_{\text{elec}} \sum_{i,j} \frac{q_i q_j}{e(r_{ij}) r_{ij}} + W_{\text{sol}} \sum_{i,j} (S_i V_j + S_j V_i) e^{\left( \frac{-r_{ij}^2}{2\sigma^2} \right)}$$

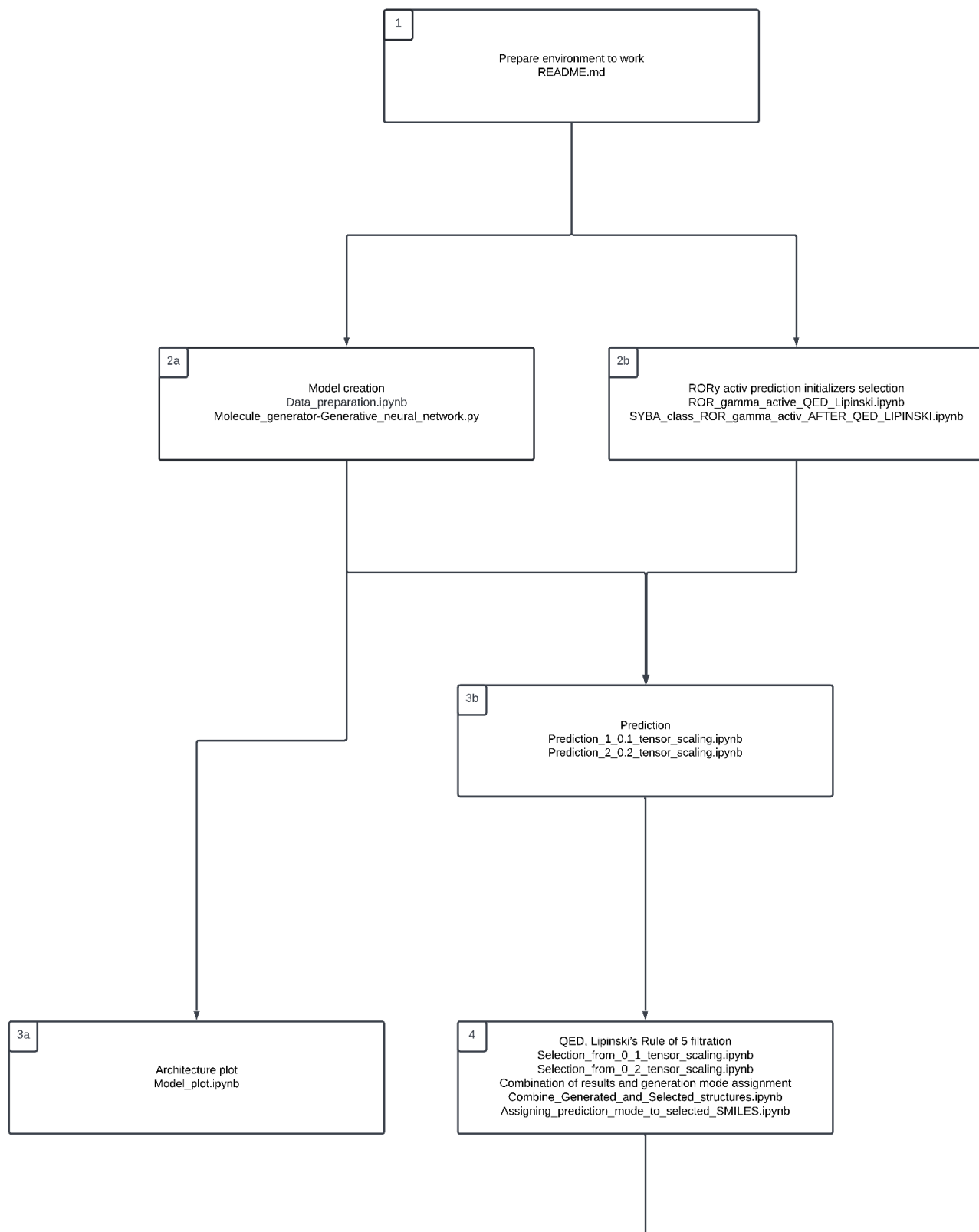
Where  $W_{\text{vdw}}$  means weighting constant for Van der Waals interactions,  $A_{ij} = 4\epsilon\sigma^{12}$  [52] ( $\epsilon$  means strength of attraction by particles,  $\sigma$  means van der Waals radius (equals ½ of the internuclear distance between nonbonding particles),  $B_{ij} = 4\epsilon\sigma^6$  [40], r is the distance of separation between both particles (from one center of the particle to the center of another particle).  $W_{\text{hbond}}$  means weighting constant for hydrogen-bonding.  $W_{\text{elec}}$  means weighting constant for electrostatics. q means the charge [40].  $W_{\text{sol}}$  means weighting constant for desolvation.

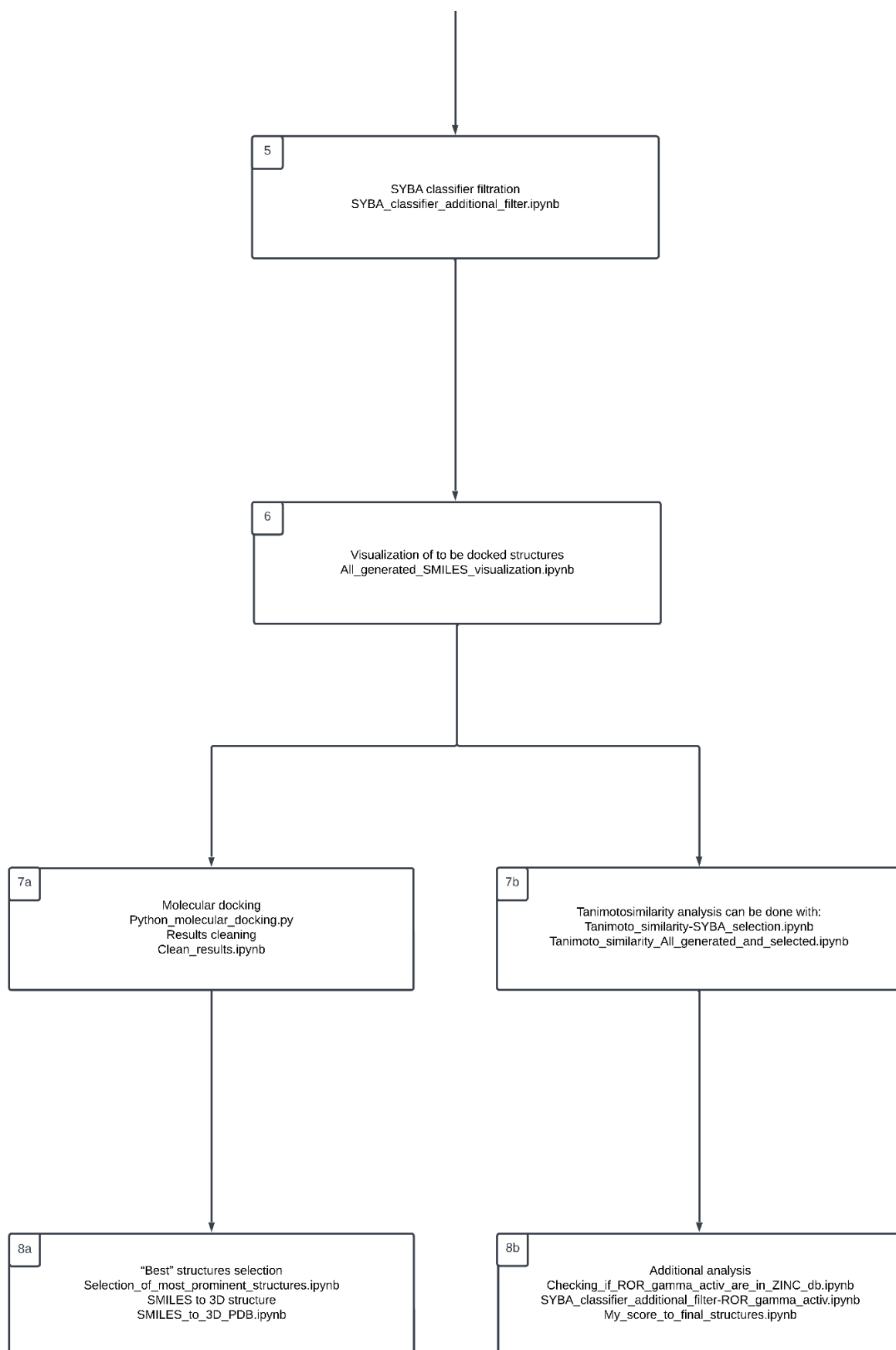
## Accuracy of AutoDock

Based on the study done by D. Plewczyski et al [7] the tool that was used during the actual investigation failed in nearly 90 cases while a total number of pairs of ligand-receptor was 1300. AutoDock has docked successfully about 93% of initial pairs. Computed ligands pose in the case of AutoDock lead to poor results when compared with the initial pose of the molecule, threshold, given in Å, was set up to 2 and everything below it was marked as successfully docked. In this case, the tool utilized during this study was in last place with Flex. One can notice that the native pose is not known before docking in most cases. It is because only in the example of a redocking native pose is previously defined and in other situations, it is not. The algorithm used by the tool requires a well-defined active site of a receptor, because if another local minimum of binding energy is being found genetic algorithm takes the best pose starting from that space during the next evaluation. According to D. Plewczyski et al AutoDock is performing well in the case of small and hydrophilic molecules with either strong or weak binding energies (about 50% accuracy – top score conformation-based analysis and about 76% accuracy – best pose conformation-based analysis) [7]. A small molecule is defined as an object that has up to 5 rotatable bonds. “In the usage of all tested docking programs, we can conclude that the larger and the more flexible the ligand is the hardest molecular docking becomes.” All the algorithms used achieved a weak correlation between in vivo derived binding energy and this computed by docking software (overall Pearson correlation for AutoDock, top score poses equals 25% and 19% in best pose comparison), it can be due to the relative simplicity of the scoring functions and other assumptions that are done during molecular docking virtually. That is why provided type of software is only a supporting tool in the drug design process. AutoDock is gaining better performance as time sampling is expanding. Along with all approaches researchers [7] agreed that attempts that employed genetic algorithms seemed to be the best option for the pose prediction.

The larger the ligand the more difficult is reaching of the protein, because the cell membrane crossing stage and software do not take this issue under consideration.

# Methodology





**Figure 1.** Overall workflow

## SELFIES coder

Functionality (see attachment 41), which name is the title of this paragraph, has been prepared by me due to the need for accurate conversion from SELFIES into molecular sequences.

Molecular structures encoded via SELFIES are more challenging to vectorize due to the mode of string to vector conversion. That is done character by a character which, in the case of SMILES is easier as SMILES can be vectorized, but SELFIES cannot. For example, a carbon atom in SMILES is coded as C and in SELFIES as [C], so when the second is taken to be vectorized it will be encoded with the use of three values and the possibility of correct recreation of three signs in the correct order is almost impossible.

To avoid potential problems, a molecular sequence term is created, which can be defined as encoded SELFIES into a string containing mono-signs that represent the respective SELFIES alphabet element.

As SELFIES encoding of molecules is more robust than SMILES, (invalid SMILES can be formed), this approach allows researchers to generate semantically correct structures in greater numbers when compared with a model using SMILES [1, 2, 41].

Arbitral mono-signs are taken based on these listed in attachment 40, and then further processing is possible.

This functionality has been developed by the author of this paper.

## Environment preparation

The first step as shown in **Figure 1**, is to set up the environment that will be able to maintain this project. All the necessary dependencies and libraries, along with steps for its installation, are given in the ReadMe.md attachment.

All the data collection and selection steps are done with the use of the Pandas library [53].

## Training data selection

In the second step, shown as 2a in **Figure 1**, SMILES of compounds are extracted from the ZINC database. Attachment 1 is the file in which the ZINC database download is done. This step saves about 885,780,663 substances [42] in approximately 1,800 tranches [42]. Then in attachment 2, a selection of data that will be used to create a model can be found. This attachment contains information about used tranches. When 935,475 unique isomeric

SMILES are collected, they are translated into SELFIES and the length of each SELFIES code is determined.

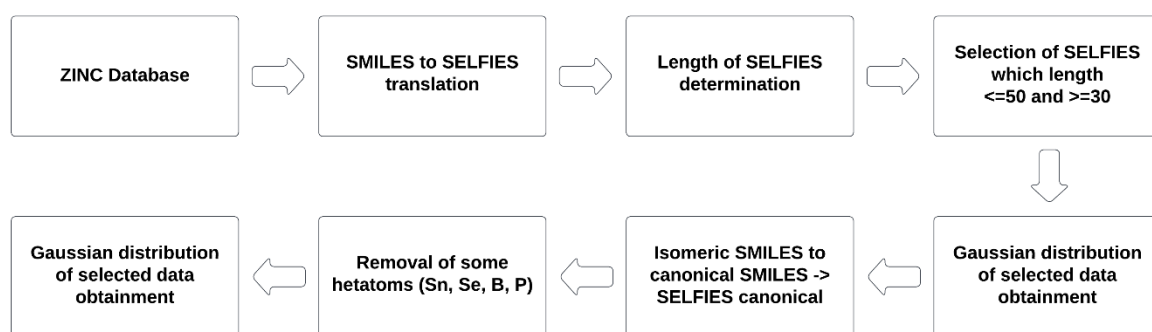
By the choice of the author of this paper for further processing, only compounds that have SELFIES length in the range of 30 to 50 are selected – 569,205 structures.

The following thing is the filtration of structures in a manner that will result in a Gaussian distribution of their length. The script, created by the author of this paper, assumes 21 bins as 21 unique SELFIES lengths are selected. To get the demanded distribution of SELFIES length, the number of structures in each bin increases by 2,000 structures. When the median bin, with a SELFIES length of 40, is reached, the further bins descend in a number of compounds by 2,000. In that manner, 241,956 structures are selected.

The next step is to take advantage of the canonical form of SMILES – a smaller charset in comparison to isomeric SMILES. From the previously obtained data, isomeric SMILES are converted into canonical SMILES. These SMILES are converted into SELFIES and their length is calculated.

Then compounds that have unwanted heteroatoms (see **Figure 2**) are removed from the dataset.

The last thing is to prepare 121,000 structures for a model. This is done in the same way as obtaining the Gaussian distribution, see attachment 3.



**Figure 2.** The procedure of data for model training preparation

## Model

This paragraph is related to step 2a in **Figure 1** and attachment 4.

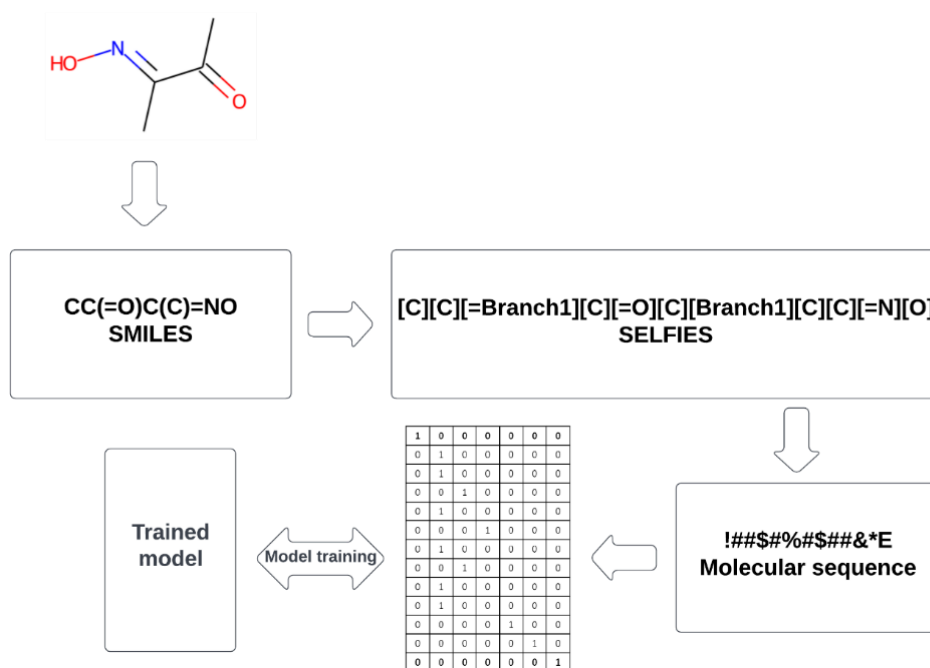
The basic idea was to prepare a model, a recurrent neural network, which would be able to generate molecules from those of known activity.

Training and testing data should be established before a model can be created. The data were obtained from the ZINC database [43] using the code shown in attachments 1 and 2.

Data collected in the previous section, see attachment 3, is used to create the model. 108,900 structures are used as training data, and 12,100 structures are used as validation data.

SMILES codes are translated into SELFIES and from SELFIES charsets are created (see attachment 5 and attachment 6). Then we have our molecules in the form of a so-called molecular sequence, and two additional charsets are created, one containing translations from arbitral mono-signs to numbers (see attachment 7) and from numbers to arbitral mono-signs (see attachment 8).

Now molecular sequences can be coded into latent space (see **Figure 3**). The vectorization step converts molecular sequences into one-hot encoded arrays. Additionally, start and stop characters are present (! – starting character, “E” – ending character). “E” works as a stop for molecule creation. It also plays the role of padding objects to obtain the same length of all vectors, thus batch mode training. The maximum length of the molecular sequence is called the embed value. It determines how long molecular sequences can be used during predictions.



**Figure 3.** Overall training workflow

Our vectors are NumPy [44] arrays, and the model (see **Figure 4**) can be trained and tested with them. The “Vectorize” function results in two X and Y types of vectors; the first is used for input, and the second for output. X starts with the starting character !. Y starts with the first character of the molecular sequence. For now, we can go forward and backward between molecular sequence and mathematical tensor.

The TensorFlow-Keras [45] library was used to build the model architecture – autoencoder. 128 long-short-term-memory (LSTM), see **Method 2**, cells were used as a decoder. These were used to read input molecular sequences. This includes the encoder. Encoder-decoder architecture can be seen.

Output was then moved through two dense layers to decode the states that were set on the LSTM decoding layer. The LSTM layer received input once again and the task of the next character prediction was requested. So, from the latent representation of the sequence, when the starting character “!” is present, the next character prediction runs until “E” is encountered, which stops the generation.

During training, callbacks were recorded such, as “loss” (see **Equation 7**), and “val\_loss” (see **Equation 7**), and they show the accuracy of prediction of the structures on unseen data in comparison with the training set.

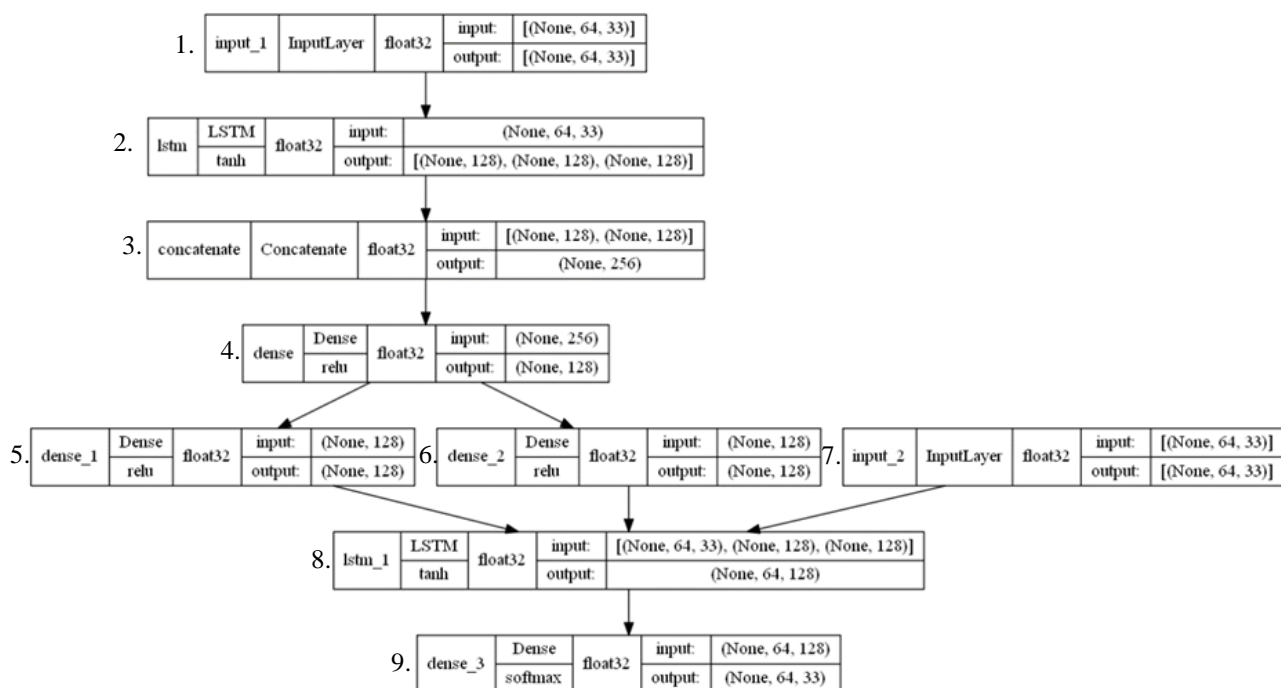
A reduction of the learning rate was also used until a training plateau was reached. The used optimizer was Adam, see **Equation 6**, and the loss function was measured on categorical cross-entropy, see **Equation 7**.

After training, the model can be divided into parts. The first part was constructed based on encoder inputs and 4<sup>th</sup> layer outputs (see **Figure 5**) and was saved in attachment 9. This part can take a vectorized molecular sequence and encode it into the latent space.

The second part of the model derived (see **Figure 6**) is responsible for latent space decoding into states that are necessary at the LSTM cell decoder. A new input shape, which matches the latent space, is set up, but the layers from before were reused to preserve the h and c states obtained during training – attachment 10.

The last extracted model (see **Figure 7**) is responsible for decoding into a tensor – vector. The model was trained in batch mode, but predictions were done in a stateful model, which means that during training it operates on a vast number of vectors representing molecular sequences, and during predictions, it will encounter one vector per time and predict one character. The weights from the trained model were transferred and saved in attachment 11.





**Figure 4.** The seq\_to\_seq model architecture.

**Figure 4** stands for the complete model workflow. The first layer is the input layer. At this point, vectors representing molecular sequences are inserted. 64 is connected to the maximal length of the molecular sequence, and 33 is the charset length. This is constructed based on zeros and ones.

The second layer, the LSTM layer (see **Method 2**) takes vectors from the first step and learns long-term dependencies. The output here is given by three new vectors. The C and H states are first and second, and the omitted output is the third. The results are in the range of the tanh function (see **Equation 3**).

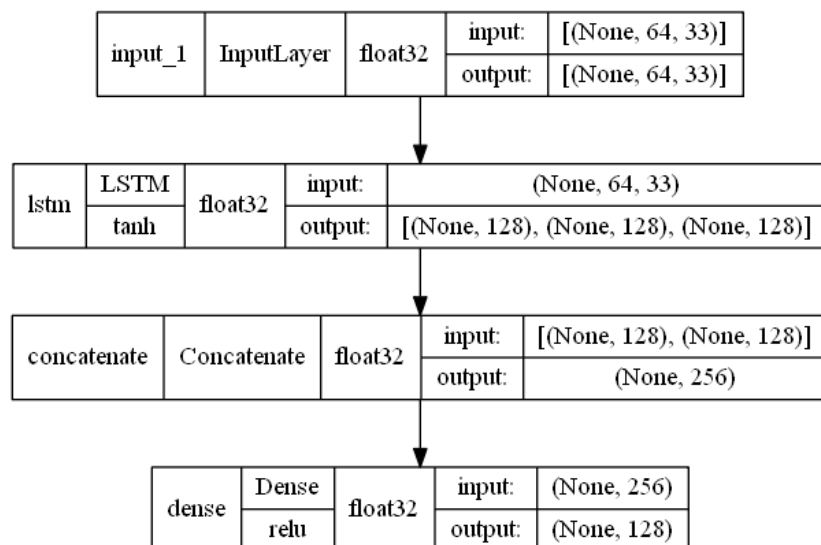
The third layer is a concatenation one where the C and H states from the second layer are combined into one vector.

The fourth layer is the so-called dense layer, in which concatenated C and H states are passed further with the use of the ReLU activation function (see **Equation 4**).

The fifth and sixth layers are the C and H states from the previous step. The used activation function to create output from them is also ReLU (see **Equation 4**).

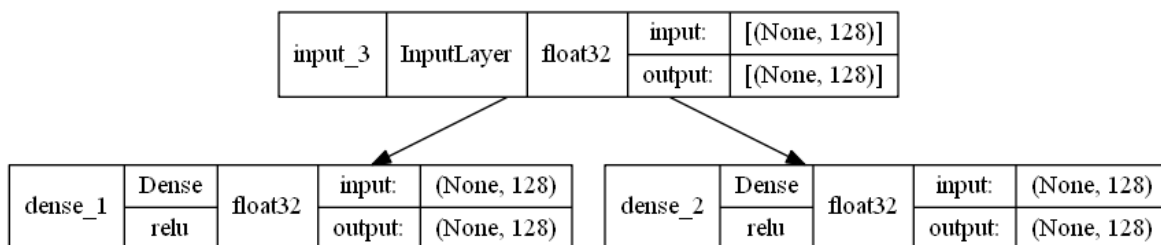
The seventh layer is a new LSTM layer, where the C and H state from the earlier step are used as input along with the vector representing a molecule (eighth layer (input)). As a result of the tanh function (see **Equation 3**) a new vector with a size of 64 (the maximal length of the molecular sequence) and combining C and H states is obtained along with the output from LSTM.

The ninth, and last layer is another dense layer that takes the previous tensor and, with the use of the SoftMax activation function (see **Equation 5**) gives an output of size 64 – the maximal length and 33 – the charset length (64x33).



**Figure 5.** The molecular sequence to a latent model is saved as another model and is reused to encode the molecular sequence into latent space.

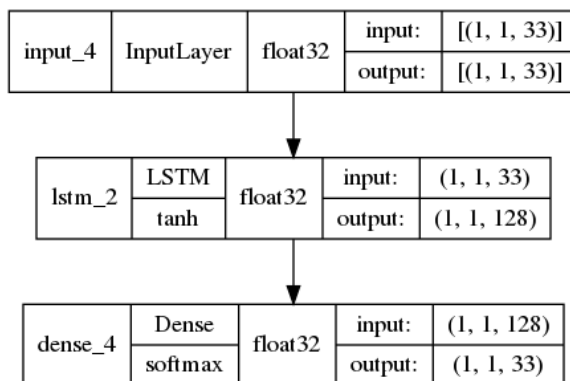
**Figure 5** shows the encoder part of the model. The first layer takes a vectorized molecular sequence and, with the use of the second (LSTM) and third (concatenated C and H states) layers encodes the vector into latent space.



**Figure 6.** The latent to states model takes the tensor of a given dimension and as a result, states are decoded.

**Figure 6** stands for the deciphering of latent space into states C and H.

This part takes encoded molecular sequence vectors and decodes them.



**Figure 7.** Sample model used to make predictions – character by character.

**Figure 7** represents the part that predicts new structures character by character. Slices of vectors are given as input (row by row), and their length is the length of the charset - 33. Then weights from the LSTM cell are transferred into the vector's slices. These new vectors are then activated with the use of the SoftMax function (see **Equation 5**), and the probability is given as the output – for each character and the length of the outcome is connected to the length of the vector that has been used as the prediction initializer.

### Data to predictions

The current paragraph is visualized in **Figure 1** as 2b part of it.

Based on the publication by Y. Zhang, et al. [15] ROR- $\gamma$  active compounds were collected (see attachment 12). Then five structures were selected (see **Figure 8**) to forecast new structures. These structures were selected after the following steps.

As the training data is constructed based on canonical SMILES, collected data should also be converted into that form.

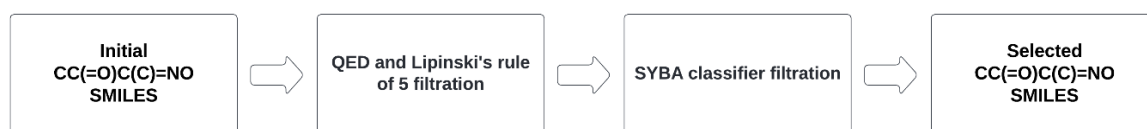
The first step of selection (see attachment 13) means that the QED descriptor (see **Equation 8**) is larger than 0.5.

The second filtration is done with the use of Lipinski's rule of five (see **Equation 9**), and only structures that are fulfilled the conditions are selected.

Then the results are saved in attachment 14. and then a third classifier is used, the SYBA algorithm, (see **Method 1**). That one's threshold is set to above 0. It decides if the structure is easier (greater value) or harder (lower value) to synthesize.

Then normalization of the QED descriptor and SYBA score is done.

The last calculated parameter is called “My score” and it takes the sum of normalized, (see **Equation 10**), QED, and normalized SYBA scores, multiplies it by 100, and divides it by 2. Then the results are saved in attachment 15. The results are sorted in descending order of My score and five structures are selected – two from the top, two from the middle, and the last one, and these are used to make predictions – saved in attachment 16.



**Figure 8.** Workflow of initializers selection

## Predictions

This paragraph is visualized as 3b in **Figure 1**.

Structure generation was done by tensor scaling and further deciphering (see **Figure 9**). It was initialized by previously selected structures – these are present in attachment 16. The saved models are used now, also the charsets.

The initial structures are then translated into molecular sequence and vectorized, so that our model can make predictions on them.

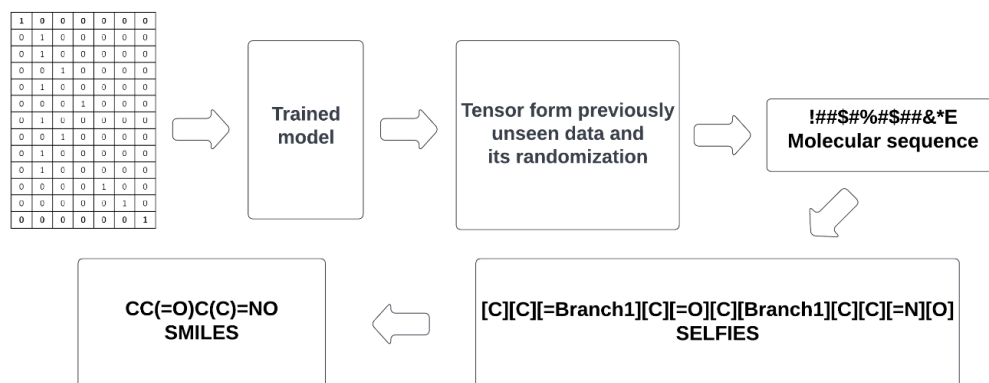
The function “latent to smiles” simply converts latent space to the molecular sequence. The scheme is that the states are taken from predictions made on newly obtained latent space, and the states of the sample model are reset with new ones. Then a for loop makes character-by-character predictions until the “E” character is outputted – then it stops creating a molecular sequence.

The decisive step is sampling around the latent vector of each initial molecule with a 0.1 and 0.2 scaling of the tensor. For each molecule, 20 tries are done. The final result should consist of 100 unique structures, but just in case there will be some repetitions, duplicates will be removed.

As a result, we get the molecular sequence, and this is translated with the use of charset that has as keys arbitral mono-signs and as values SELFIES alphabet elements. Translation into SELFIES is done. After that, SELFIES is translated into SMILES.

Then structures are printed, so they can be easily viewed and saved into attachment 18. and nineteen for 0.1 and 0.2 tensor scaling, respectively. Then the obtained SMILES codes are canonicalized and a search in the PubChem database is done – if any of the newly generated structures are present in the database, they will be printed in the document.

These steps are done in attachments 17 and 18, with 0.1 scaling and 0.2 scaling correspondingly. The results are saved in attachments 19 and 20.



**Figure 9.** Data prediction workflow

Below, procedures are marked as 4 in **Figure 1**.

The first data selection is done in attachments 21 and 22 in which the QED descriptor, see **Equation 8**, is discriminant if its value is lower than 0.5 (see **Figure 10**). As a second discriminant, Lipinski's rule of 5, see **Equation 9**, is used. This is done by fully – 1 or other - 0, fulfillment of this filter (see **Figure 10**).

The results of both selections were saved into attachments 23 and 24 appropriately.

Then these structures are combined into one file, which is done in attachment 25. After that, a prediction mode is assigned to each structure, (see attachment 26), and the results are saved in attachment 27.

The step described below is assigned to number 5 in **Figure 1**.

As forty-two structures passed the first two filtration steps, one additional was applied. It was done with the use of the SYBA classifier (see **Method 1**, see **Figure 10**). The threshold was set to 0, so that each structure that has an SYBA score above 0 passed the test and the other was rejected. The last reduction of results is done in attachment 28 and attachment 29 is created.



**Figure 10.** Workflow of generated structures filtration

The 6<sup>th</sup> step shown in **Figure 1** displays the possibility of visualization of structures that can be docked (see attachment 30). Just after this, molecular docking of selected structures can be applied. To the output file's name suffix, “\_blind\_try” should be added just before the .xlsx extension. If some structures prove problematic during the docking procedure, they can be reduced by rerunning the code present in attachment 30. Then molecular docking can be run one more time, but without problematic structures.

## Similarity

That part is marked as 7b in **Figure 1**.

Tanimoto similarity [31] (see **Method 3**) is measured to determine how comparable two structures are. Where one means the same structure and zero has no common parts.

Firstly, a comparison between selected structures, and five initial structures was made (see attachment 32).

Then Tanimoto similarity is calculated among structures that are going to be docked. This means structures that meet the filtration steps from the previous paragraph (see attachment 32).

The third check is comparing training data with newly achieved chemicals. The frequency axis has a much larger scale due to the number of compounds times the number of training structures (121,000) result to be displayed as calculations are done for a combination of each docked structure with each training structure (see attachment 32).

After that, there is a simple check if initial structures (used to make predictions) can be found in PubChem and the same is done for molecules that meet requirements (see attachment 32).

The same procedure as described above is applied to all structures that meet QED and Lipinski's rule of five thresholds (see attachment 33).

The next step is checking if the ROR- $\gamma$  active compounds used in this paper are present in the training dataset (see attachment 34). Further checking checks if any of the generated structures are present in the training dataset or ROR- $\gamma$  active compounds (see attachment 34).

## Molecular docking

Molecular docking is marked as 7a in **Figure 1**.

The search for possible ligand–macromolecule interactions was investigated via a molecular docking procedure (see **Figure 11**) and with the utilization of the pyscreener tool [44]. Where binding energy is the output, the more negative the result, the better the affinity of the ligand to the receptor.

The procedure has been conducted inside attachment 35 – it is an automated way of molecular docking via python code and its libraries. As a result, a file is created, see attachment 36. Binding energies, structures, and coordinates are stored inside attachment 36, see **Method 4**. Then results are extracted via the code present in attachment 37.

The steps below are assigned as 8a in **Figure 1**.

Average binding energies are calculated based on three experiments of molecular docking (see attachment 38). They are ordered in descending order by average binding energy. This was done to get data to make manual molecular docking in AutoDockTools 1.5.6 and get visualizations of what the results look like.

Then, using the code included in attachment 39, SMILES are converted into 3D structures that can be used during manual molecular docking procedures. It is done based on general chemistry rules of angles, hybridizations, and distances.



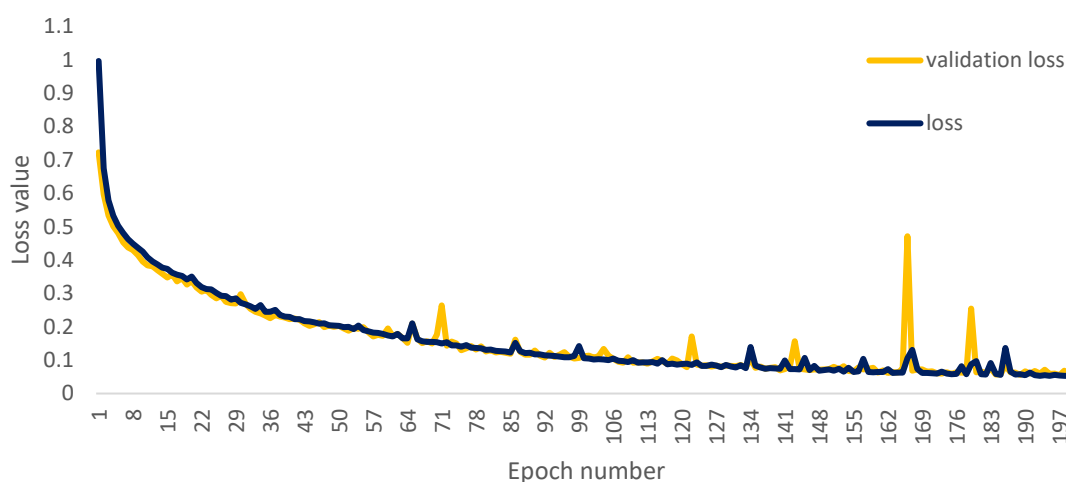
**Figure 11.** Workflow of molecular docking

## Results and discussion

### Model

The prepared sequence to sequence model can construct semantically correct structures. The model has been trained the with use of 121,000 structures. The loss function used (SoftMax (see **Equation 5**)) showed us that the model is performing well, because the loss value converges to 0 (see **Figure 12**).

Even so, a longer learning time may lead to even lower losses.



**Figure 12.** Loss and validation loss during seq\_to\_seq model creation. It follows the general rules of model training during time evaluation both the values are decreasing, which means that the model learns how to reconstruct the training molecules – molecular sequence.

The given model shows the possibility of learning chemistry via neural networks. Application of SELFIES codes leads to no errors during the prediction step. In the case of the application of SMILES chemical information, there is a possibility of the formation of wrong structures [41, 2]. This fact is due to the lower robustness of the SMILES notation in comparison to SELFIES [41, 2].

This model is able to handle molecules whose representation in SELFIES form is shorter than 64. The elements that can be effectively encoded with the use of it are listed below:

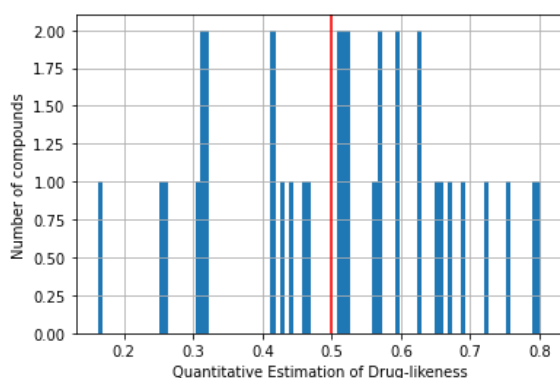
{[I]: #, [#Branch1]: \$, [#N]: %, [NH1]: &, [=S]: ', [#C]: (, [Ring2]: ), [N-1]: \*, [Ring1]: +, [=N]: ,, [N]: -, [Cl]: ., [=Branch1]: /, [=N-1]: 0, [=C]: 1, [CH0]: 2, [O-1]: 3, [=N+1]: 4, [Br]: 5, [O]: 6, [S]: 7, [C]: 8, [=Ring1]: 9, [N+1]: :, [Branch1]: ;, [#Branch2]: <, [=O]: =, [=Branch2]: >, [=Ring2]: @, [F]: A, [Branch2]: B}, where key : value architecture is maintained. It shows how SELFIES elements are translated into molecular sequences.



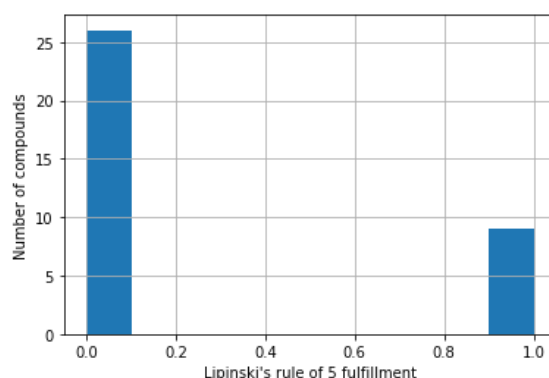
## Data to predictions

In the beginning, 36 structures from the Y. Zhang, et al. publication [15] (see attachment 12.) were present. After conversion, the R and S stereoisomers translated into the same canonical SMILES.

The distributions of the first and second discriminants are given below (see **Figure 13** and **Figure 14**).

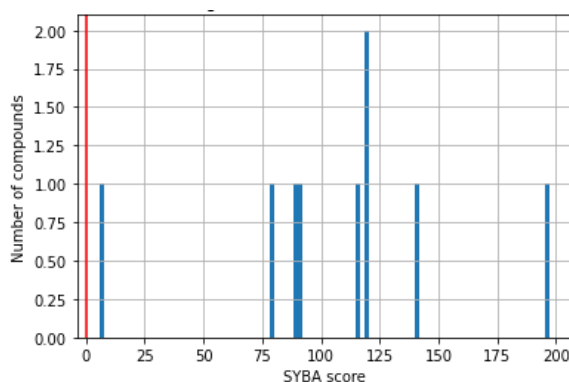


**Figure 13.** Histogram of QED distribution for ROR-y active compounds with a marked threshold (red vertical line)

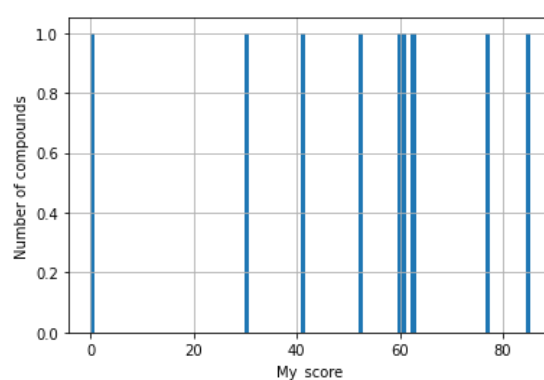


**Figure 14.** Histogram of Lipinski's rule of five fulfillment distribution for ROR-y active compounds

21 structures passed through the QED discriminator, and 9 through Lipinski's rule of 5. The SYBA score was calculated for each of the previously selected structures (see **Figure 15**).



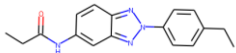
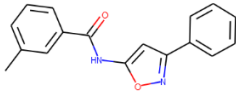
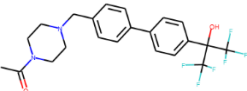
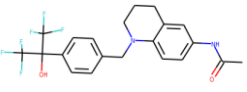
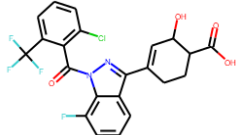
**Figure 15.** Histogram of SYBA score distribution for ROR-y active with a marked threshold (red vertical line)



**Figure 16.** My score distribution to 9 structures that passed through three filters

My score for structures that passed the SYBA threshold was calculated. Its distribution is shown above (see **Figure 16**).

**Table 1.** Prediction initializers: normalization, see **Equation 10**, is done based on ROR- $\gamma$  active compounds that passed QED and Lipinski's thresholds (9 SMILES strings) – attachments 15, 16.

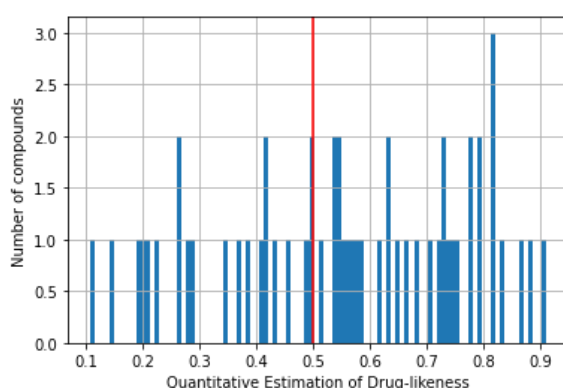
Structure	QED	QED normalized	SYBA score	SYBA score normalized	My score	ROR $\gamma$ activity
	0.80	1.00	140.57	0.70	85.19	agonist
	0.79	0.96	119.44	0.59	77.43	agonist
	0.69	0.60	119.78	0.59	59.77	inverse agonist
	0.66	0.48	116.61	0.58	52.72	inverse agonist
	0.52	0.00	5.72	0.00	0.00	inverse agonist

Structures that are used as prediction initializers are presented in **Table 1**. The first structure is known as GSK-1c, the second as GSK-1a, the third as SR15555, fourth N-(1-(4-(1,1,1,3,3,3-hexafluoro-2-hydroxypropan-2-yl)benzyl)-1,2,3,4-tetrahydroquinolin-6-yl)acetamide and fifth 4-(1-(2-chloro-6-(trifluoromethyl)benzoyl)-7-fluoro-1H-indazol-3-yl)-2-hydroxycyclohex-3-enecarboxylic acid.

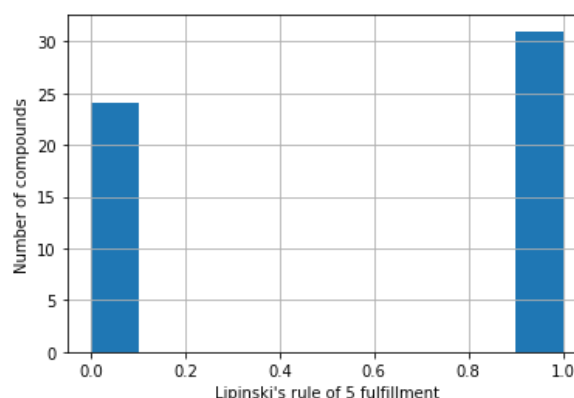
## Predictions and results of filtration

There were 55 unique structures generated with the use of 0.1 tensor scaling. Three of them were found in PubChem. Their CIDs are: 16445174, 18006105, and 129773833.

The QED descriptor calculation results (see **Figure 17**), along with Lipinski's rule of five fulfillment (see **Figure 18**), are shown below. Of these unique structures, thirty-four met the QED requirement and thirty-one met the second discriminator. The number of structures that met both is equal to twenty-six, 47.27% of generated compounds.



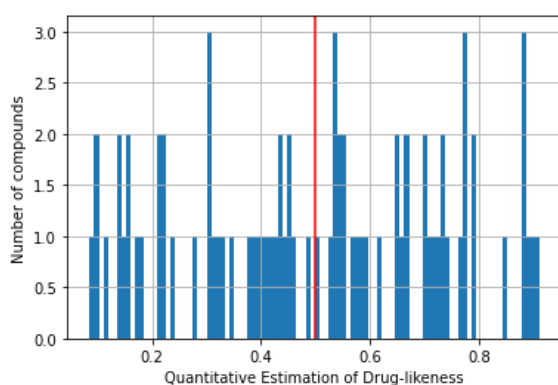
**Figure 17.** Histogram of QED distribution for the first prediction with a marked threshold (red line)



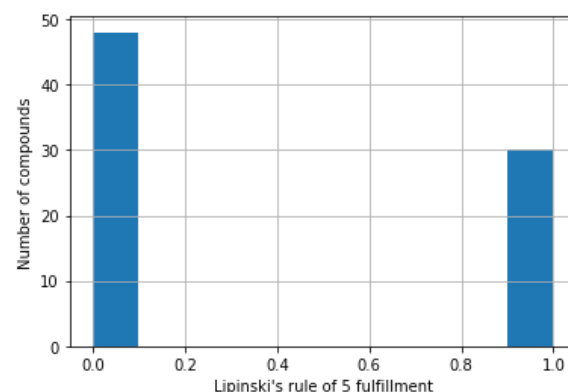
**Figure 18.** Histogram of Lipinski's rule of five fulfillment distribution for the first prediction

There were 78 unique structures generated with the use of 0.2 tensor scaling. Three of them were found in PubChem. Their CIDs are 16445174, 129773833, and 18006105

The QED descriptor calculation results (see **Figure 19**), along with Lipinski's rule of five fulfillment (see **Figure 20**), are shown below. From these unique structures, thirty-nine met the QED requirement, and thirty met the second discriminator. The number of structures that met both is equal to twenty-five - 32.05% of the generated compounds.

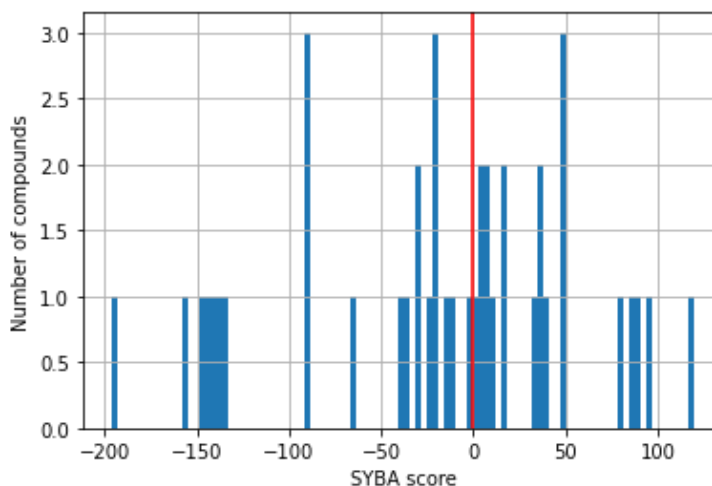


**Figure 19.** Histogram of QED distribution for the second prediction with a marked threshold (red vertical line)



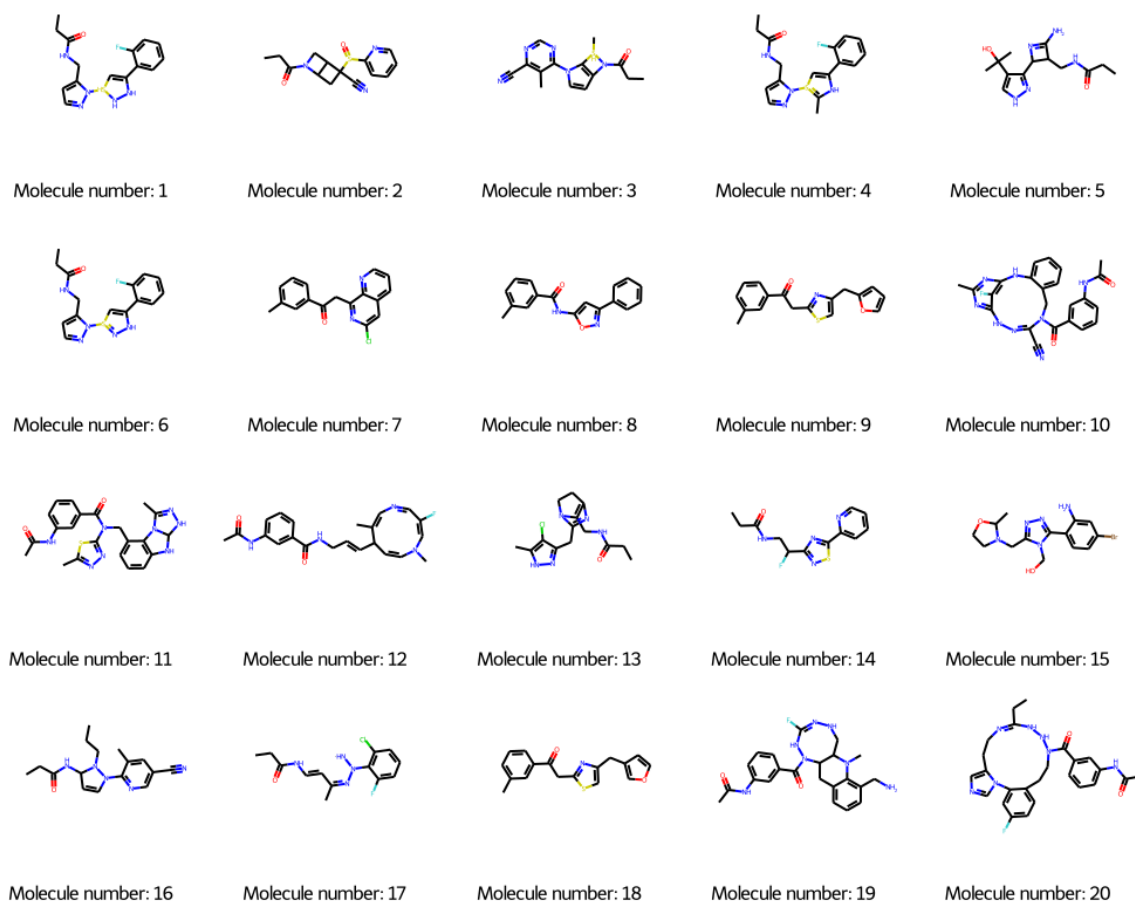
**Figure 20.** Histogram of Lipinski's rule of five fulfillment distribution for the second prediction

Then twenty-six from the first prediction and twenty-five from the second were combined and repetitions were removed. It results in forty-two unique structures. The third discriminator was applied – the SYBA classifier. The results of its application are shown below (see **Figure 21** and **Figure 22**).

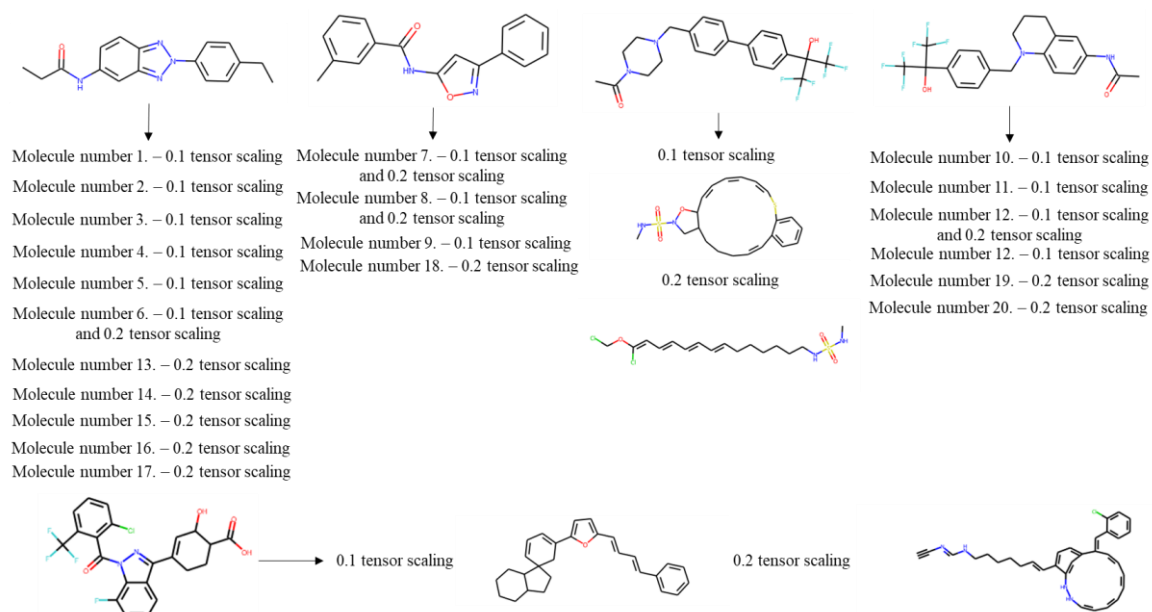


**Figure 21.** SYBA score distribution over seventy-six selected structures with a marked threshold (red vertical line)

Twenty unique structures persisted after SYBA score filtration, presented in **Figure 22**.



**Figure 22.** Selected molecules: QED, Lipinski's rule of five, and SYBA score. Visualization is done in attachment 30 and 29. information about them is saved.



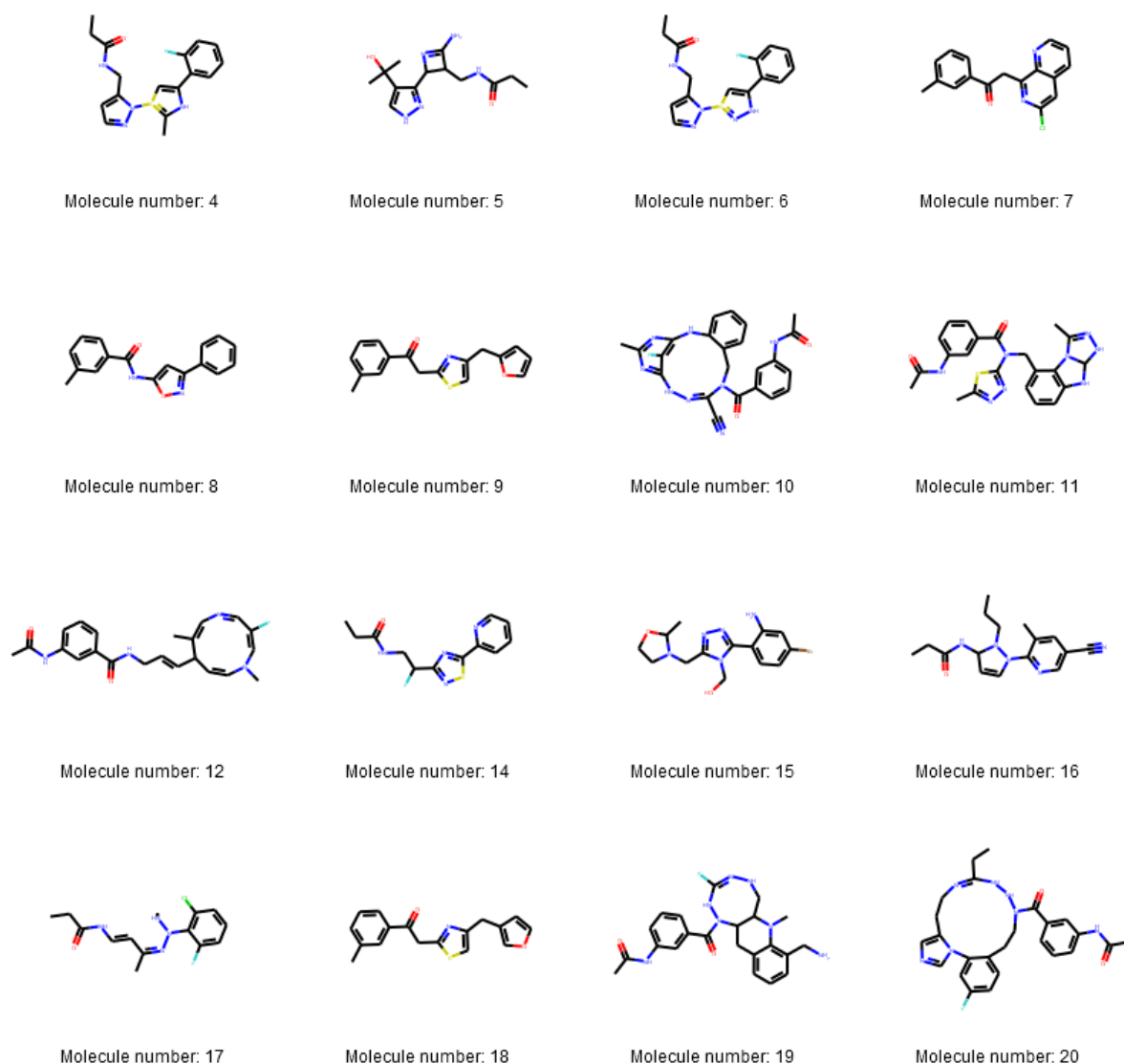
**Figure 23.** Initial structures with predicted assignment along with a show of structures that comes from 3<sup>rd</sup> and 5<sup>th</sup> initial structures which do not pass-through filters.

Eleven out of twenty structures come from the first structure tensor scaling (see **Figure 23**). This fact shows that the prediction given by this model is connected with the type of initial structure. By the meaning of the initial structure QED descriptor value and the fulfillment of Lipinski's Rule of five.

After the removal of structures that produce some problems during the molecular docking procedure, the analyzed molecules are reduced to sixteen. These are visualized below.

Molecules number 1, 2, and 3 have problems while converting into 3D structures with the use of the RDkit library. The issue was connected to sulfur atoms. This is the reason those were omitted.

Due to RDkit kekulization problems molecule, number 13 was also removed from the dataset.



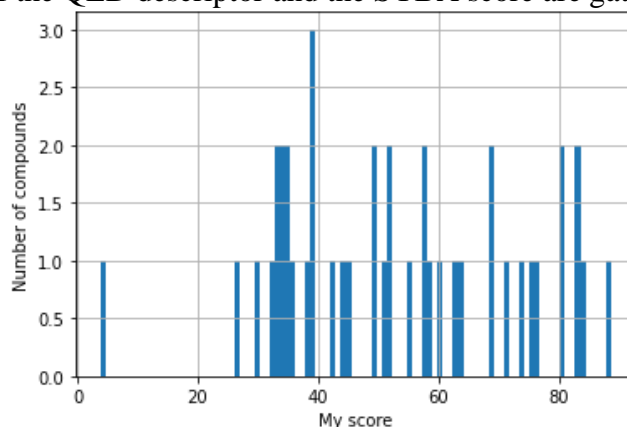
**Figure 24.** Structures that pass the first molecular docking procedure without errors

Seven out of sixteen predicted structures come from the first initial SMILES that has high values of QED and SYBA score (see **Figure 23** and **Figure 24**). From the second initial structure, four predicted structures pass through filtration steps. In the case of the third, no predicted structure met the filtration requirements. From the fourth structure, six structures met the filtration requirements. The last initial structure gives no predicted one that meets the given conditions. In that way, sixteen structures are analyzed with the use of molecular docking.

**Table 2.** My score, QED normalized and SYBA scores normalized, see **Equation 10**, for selected structures with taking attachment 27 - so all structures that met the QED threshold and Lipinski's rule of 5. They are used to determine normalized QED and SYBA score and my score (see attachment 31)

Molecule number	QED	QED normalized	SYBA score	SYBA score normalized	My score
4	0.82	0.76	39.47	0.75	75.10
5	0.62	0.22	5.24	0.64	42.70
6	0.88	0.92	35.55	0.73	82.67
7	0.54	0.02	79.24	0.87	44.65
8	0.79	0.68	119.44	1.00	84.00
9	0.67	0.35	90.13	0.91	63.02
10	0.53	0.00	47.43	0.77	38.68
11	0.53	0.00	85.03	0.89	44.54
12	0.74	0.56	7.05	0.64	60.04
14	0.91	1.00	47.22	0.77	88.54
15	0.79	0.69	17.76	0.68	68.36
16	0.90	0.97	4.28	0.63	80.42
17	0.67	0.36	15.68	0.67	51.67
18	0.67	0.35	95.91	0.93	63.93
19	0.54	0.01	6.40	0.64	32.45
20	0.54	0.01	47.67	0.77	39.20

Detailed values of the QED descriptor and the SYBA score are gathered in **Table 2**.

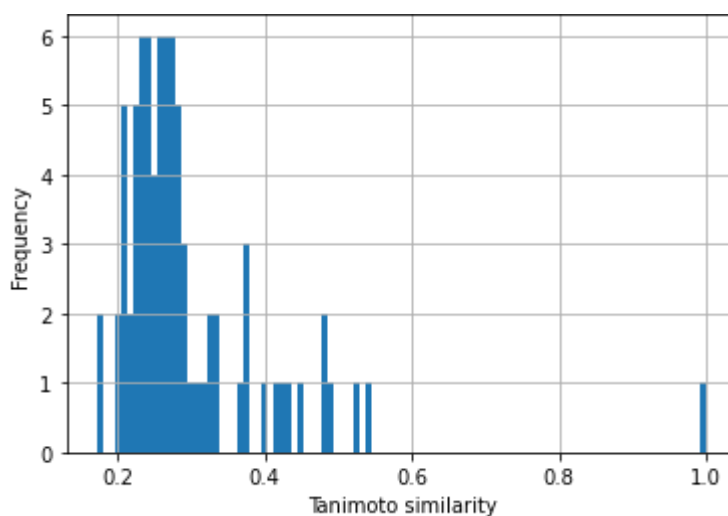


**Figure 25.** My score distribution for attachment 27. – all generated SMILES (forty-two structures)

Compounds that were preserved possess a generally high value of my score (see **Figure 25**) which means that both the descriptors – QED and SYBA score – are higher than lower. But in two cases, molecule numbers ten and eleven, the lowest QED descriptor is maintained. In comparison with all the data after the first two filtrations. These structures go further due to a high SYBA score.

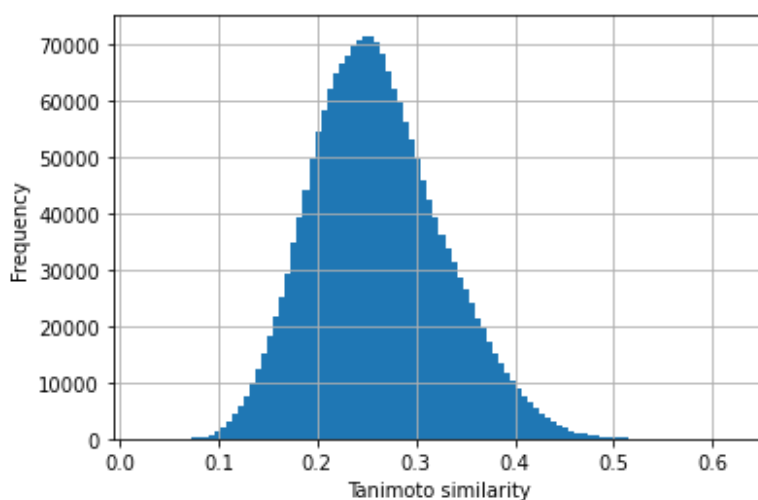
Some structures possess the highest QED (molecule number fourteen) and SYBA score (molecule number eight) in comparison to all generated structures after QED and SYBA filtration.

## Similarity to initial structures and training data



**Figure 26.** Tanimoto similarity distribution along with sixteen generated structures after selection (see **Figure 24**) and five initials

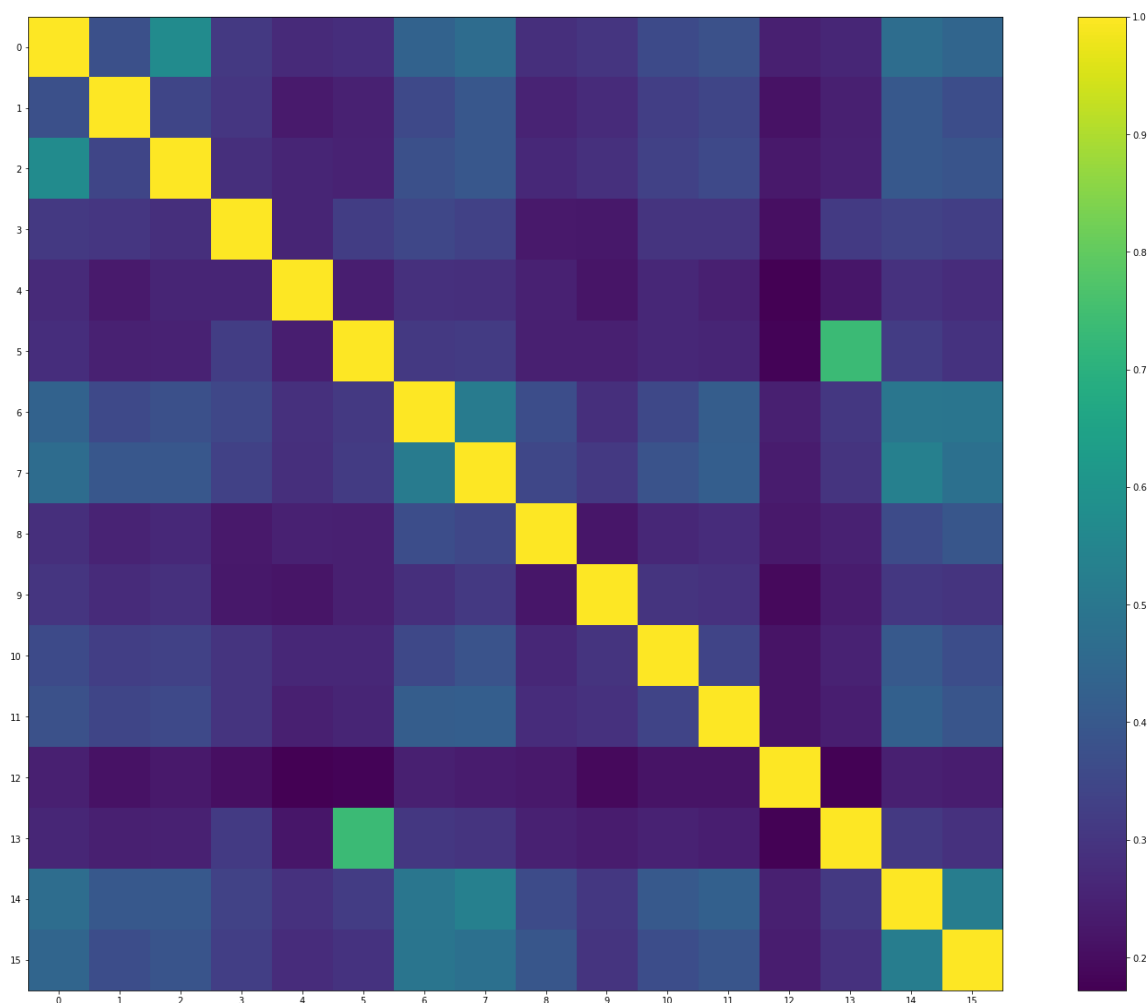
Structures from tensor scalation are not remarkably similar in comparison to initial structures (see **Figure 26**). The highest similarity value is found at 1.00, which means that one structure was recreated in 100% of the same shape. The lowest was at 0.17. Results are in a different molecular space than initials.



**Figure 27.** Histogram of Tanimoto similarity distribution between to be docked molecules and training data

Structures from tensor scalation are not highly similar to training molecules (see **Figure 27**). The highest similarity has been found at 0.62. The lowest similarity has been found at 0.02 – so the structures that give this result are different. The mean similarity is 0.26.



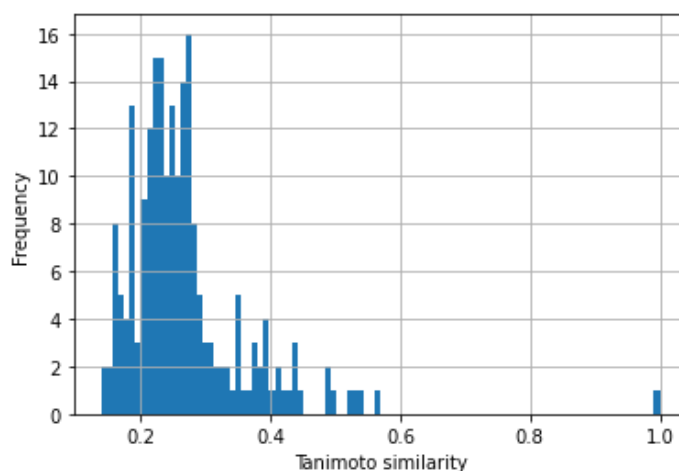


**Figure 28.** Tanimoto similarity distribution along to be docked molecules

The comparability of structures that are selected for molecular docking gives information that these structures are also not identical (see **Figure 28**), and the thirteenth is the outlier as it has the lowest Tanimoto similarity output in each comparison.

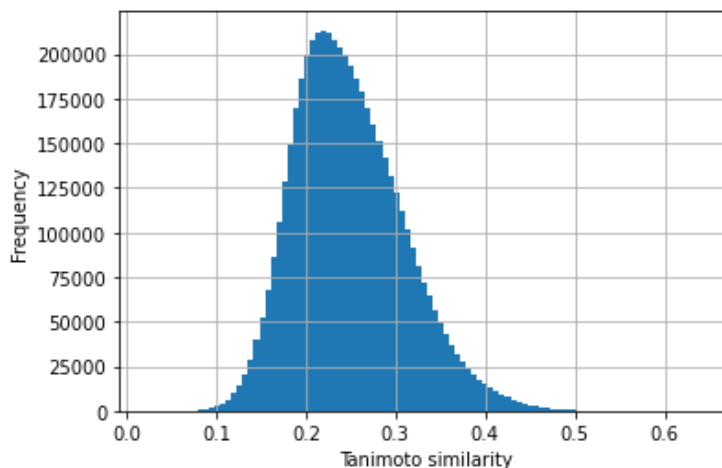
When using the pubchempy [47] library to see if initial SMILES are present in PubChem, the following PubChem CIDs are returned: 807146, 16445174, 71470549, 71811962, 135337558, and these structures are related to initial structures, respectively. The same search for ten structures after 3-step filtration gives 16445174 as an outcome. That means one generated structure by the neural network has been found in PubChem. All the above steps are done in attachment 32.

Questions can be raised about the similarity between structures after the first selection (42 objects) and the initial ROR- $\gamma$  active set of compounds containing five structures (see **Figure 29**). The highest hit is 1.00, so the one recreated structure. The lowest is 0.14.



**Figure 29.** Tanimoto similarity distribution between data after QED, Lipinski's rule of five selection and initials

The last chemical resemblance is computed for forty-two structures after QED and Lipinski's selection, and the whole training dataset consists of 121,000 entities (see **Figure 30**). The maximal and minimum values are 0.65 and 0.02, respectively. In this case, nine hundred and eighty-three hits are above the 0.5 threshold.



**Figure 30.** Histogram of Tanimoto similarity distribution between forty-six structures and training data

Pubchempy [47] was used to check whether some of these forty-two structures were present in the PubChem database. The results show that there is one structure found, CID: 16445174. All the above steps are done in attachment 33.

Inside attachment 34, there is a check to see if some ROR- $\gamma$  active compounds are included in the used database and if some of the reconstructed entries are here too. As a result, no structure from the whole ROR- $\gamma$  active dataset and no structure that was generated is present in the training dataset. Only one has been found in the initial five structures. That implies that one structure has been recreated.

## Molecular docking of selected structures

The results of molecular docking are collected in **Table 3**.

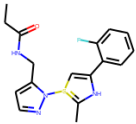
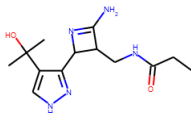
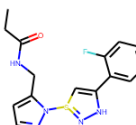
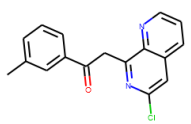
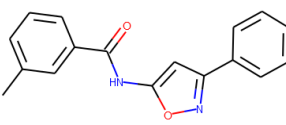
The lowest average binding energy was found for molecule number 12, which equals -9.80 kcal/mol. The second was molecule number 20, with -9.70 kcal/mol. The third was the 19<sup>th</sup> with -9.60 kcal/mol.

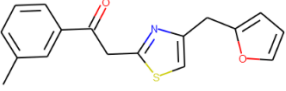
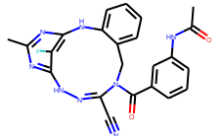
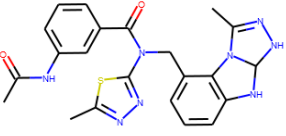
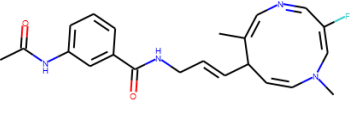
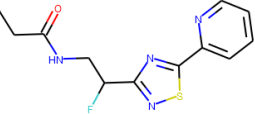
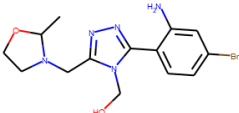
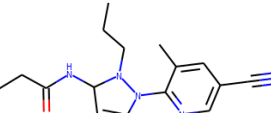
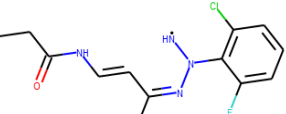
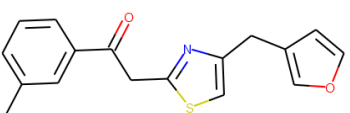
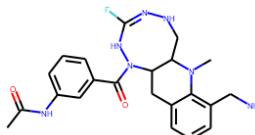
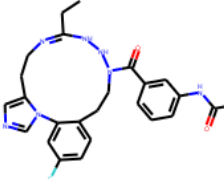
The minimal binding energy for the 7NPC - macromolecule was -10.0 kcal/mol, and the structure related to this result is molecule number 10. In the case of the 7NP5 -10.0 kcal/mol binding energy, is found to be the lowest value, and the corresponding structures are the 7<sup>th</sup>, 19<sup>th</sup>, and 20<sup>th</sup>. The 7KXD complex, with a minimum energy of -10.0 kcal/mol, has the lowest energy, and molecules 11<sup>th</sup>, 12<sup>th</sup>, and 19<sup>th</sup> are the causes of it.

The most favorable average binding energy is for molecule number 5, at -6.7 kcal/mol. It is the worst candidate for a new drug based on binding energy.

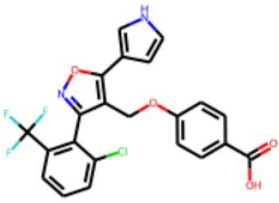
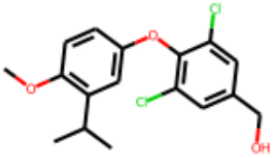
For each of the chosen macromolecules, the less favorable binding energies are -6.5 kcal/mol (7NPC), -6.9 kcal/mol (7NP5), and -6.7 kcal/mol (7KXD), and molecules are related to those results in the order of 5 (7NPC), 5 (7NP5), and 14 (7NP5), 5 (7KXD), respectively.

**Table 3.** Docked structures with docking scores, in kcal/mol unit, and IUPAC names – attachment 42.

Structure	Name IUPAC	7npc	7np5	7kxd
	N-({1-[4-(2-fluorophenyl)-2-methyl-3H-1lambda4,3-thiazol-1-yl]-1H-pyrazol-5-yl}methyl)propanamide	-7.9	-8.2	-8.6
	N-({4-amino-2-[4-(2-hydroxypropan-2-yl)-1H-pyrazol-3-yl]-2,3-dihydroazet-3-yl}methyl)propanamide	-6.5	-6.9	-6.7
	N-({1-[4-(2-fluorophenyl)-3H-1lambda4,2,3-thiadiazol-1-yl]-1H-pyrazol-5-yl}methyl)propanamide	-8.6	-8.1	-8.7
	2-(6-chloro-1,7-naphthyridin-8-yl)-1-(3-methylphenyl)ethan-1-one	-9.2	-10.0	-8.7
	3-methyl-N-(3-phenyl-1,2-oxazol-5-yl)benzamide	-8.8	-9.0	-9.2

	2-{4-[(furan-2-yl)methyl]-1,3-thiazol-2-yl}-1-(3-methylphenyl)ethan-1-one	-8.0	-8.2	-8.5
	N-(3-{11-cyano-18-fluoro-16-methyl-2,10,12,13,15,17-hexaazatricyclo[12.3.1.0 <sup>3,8</sup> ]octadeca-1(17),3,5,7,11,14(18),15-heptaene-10-carbonyl}phenyl)acetamide	-10.0	-9.1	-9.2
	3-acetamido-N-(5-methyl-1,3,4-thiadiazol-2-yl)-N-({3-methyl-2,4,5,7-tetraazatricyclo[6.4.0.0 <sup>2,6</sup> ]dodeca-1(8),3,9,11-tetraen-12-yl}methyl)benzamide	-8.7	-8.8	-10.0
	3-acetamido-N-[3-(3-fluoro-1,7-dimethyl-1,8-dihydro-1,5-diazecin-8-yl)prop-2-en-1-yl]benzamide	-9.9	-9.5	-10.0
	N-{2-fluoro-2-[5-(pyridin-2-yl)-1,2,4-thiadiazol-3-yl]ethyl}propanamide	-6.6	-6.9	-7.2
	[3-(2-amino-4-bromophenyl)-5-[(2-methyl-1,3-oxazolidin-3-yl)methyl]-4H-1,2,4-triazol-4-yl]methanol	-7.0	-7.1	-7.3
	N-[1-(5-cyano-3-methylpyridin-2-yl)-2-propyl-2,3-dihydro-1H-pyrazol-3-yl]propanamide	-7.5	-7.4	-7.7
	2-(2-chloro-6-fluorophenyl)-3-(4-propanamidobut-3-en-2-ylidene)triazan-1-yl	-7.0	-7.1	-7.3
	2-{4-[(furan-3-yl)methyl]-1,3-thiazol-2-yl}-1-(3-methylphenyl)ethan-1-one	-9.7	-8.1	-8.3
	N-{3-[8-(aminomethyl)-3-fluoro-7-methyl-1H,2H,5H,6H,6aH,7H,12H,12aH-[1,2,4,5]tetrazocino[7,6-b]quinoline-1-carbonyl}phenyl}acetamide	-8.8	-10.0	-10.0
	N-(3-{10-ethyl-19-fluoro-2,4,9,11,12,13-hexaazatricyclo[14.4.0.0 <sup>2,6</sup> ]icosa-1(20),3,5,9,16,18-hexaene-13-carbonyl}phenyl)acetamide	-9.5	-10.0	-9.6

**Table 4.** Results of molecular docking for structures present in raw file from Protein DataBase along with QED and SYBA score – see attachment 42, see attachment 28.

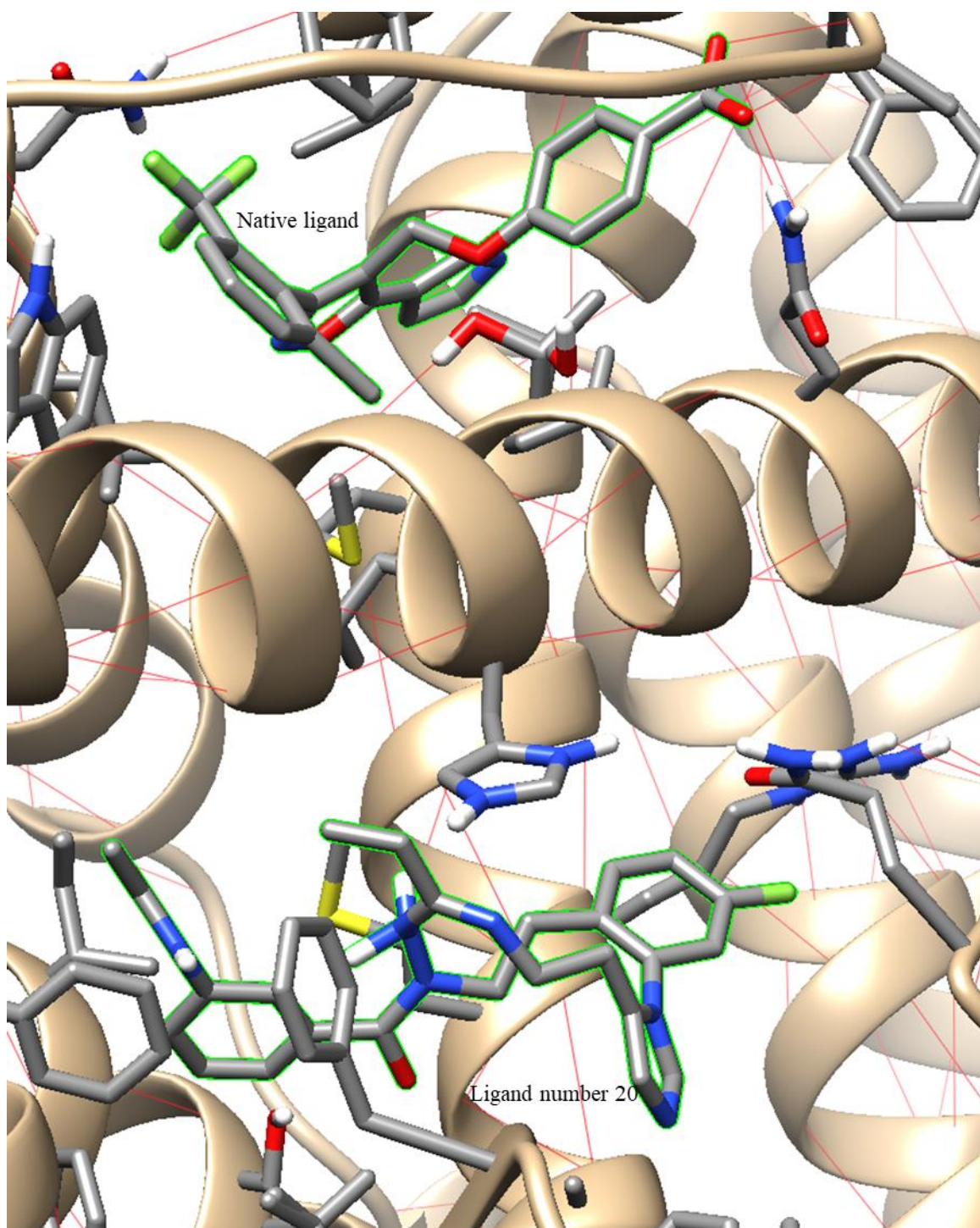
Structure	Name IUPAC	QED	SYBA score	Docking result/ name of domain
	4-[[3-[2-chloranyl-6-(trifluoromethyl)phenyl]-5-(1~{H}-pyrrol-3-yl)-1,2-oxazol-4-yl]methoxy]benzoic acid [39]	0.35	45.11	-13.84 kcal/mol/ 7npc
				-14.19 kcal/mol / 7np5
	{3,5-dichloro-4-[4-methoxy-3-(propan-2-yl)phenoxy]phenyl}methanol [38]	0.79	71.02	-8.66 kcal/mol / 7kxd

In **Table 4**, docking results are obtained for the redocking procedure as we know the exact place of binding. A recreation of the initial pose has been achieved in attachment 43 with the use of AutoDockTool 1.5.6. In the case of pythonic molecular docking, the results are significantly different (see attachment 36). They are as follows: -8.0 kcal/mol for the 7NPC ligand, -9.2 kcal/mol for the 7NP5 ligand, and -7.7 kcal/mol for the 7KXD ligand (see attachment 41).

These differences in results are observed due to the varied sizes of the search space; it shows how significant the search space size is. The time during which the genetic algorithm is allowed to search for the best poses is crucial.

So-called sieving of molecules has been used to get a general overview (see attachment 35) and mark those that are the most prominent. As for macromolecules, their activity can be modified in diverse ways by inhibitors, agonists, or antagonists.

The visualization of the results can be found in **Figure 31**.



**Figure 31.** Possible interactions between 7NP5 ROR- $\gamma$  receptor with molecule number 20 and also with a ligand that is present in raw the PDB file, hydrogen bonds are marked in red. Ligand present in raw file pose has been recreated – attachment 43.



## Conclusions

The used model can create molecules that, only in a limited manner, are similar to the training data and initial data. This can be due to the method of output formation – tensor scaling. That method takes advantage of more than one structure generation per single initial molecule as input.

Given the approach of potentially new drug discovery in that way, it is not characterized by infinite robustness, as many created structures are hard-to-synthesize according to the SYBA score. What is more, the use of SELFIES codes helps to reduce the formation of incorrect SMILES, which is a plus. In that utilization, all structures that were generated are semantically valid. But another problem was also observed - the problem related to the possibility of exotic structure formation, along with some problems while creating 3D structures. These challenging structures can be out.

Other solutions to the de novo drug designing problem can be employed, such as conditional recurrent neural networks [1], in which output is more targeted, fingerprints of known molecules with the usage of sequence-to-sequence reconstruction [33], and multi-layered Gated Recurrent Unit (GRU), also other RNNs architectures [41, 2]. In these approaches, the steering of what the neural network produces is different, in a positive sense. In the case of this paper, the new structures are more random. These approaches can be studied in future projects.

The sequence-to-sequence model employed in this paper has some drawbacks, for example, the limited length of molecular sequence that can be encoded. This fact is connected to the generally shorter SELFIES codes in length than the starting SMILES codes. One needs to consider that SELFIES length is related to the used molecular sequence length. This leads to a possible limitation of the initial structure's molecular sequence length. The loss and validation loss values can possibly be lower in value as the training epochs number will be greater, but this newly created one allows one to search for close molecular space, which can also be promising for new drug discoveries.

Most of the structures that are generated from the one that initially has high values of QED and SYBA score are above the threshold. The majority of the selected compounds are created from the tensor scaling of the first structure. This may lead to the conclusion that the initial structure has a significant role in the outcomes of the model. More outputs may pass

filtration steps if only initial structure number one is used, which is derived from ROR- $\gamma$  active compounds.

Differences in energies after molecular docking procedures can be observed. It can be so due to the different requirements of each search. First, the so-called screening approach can be seen, in which many ligands are evaluated with many macromolecular systems. The second, to prepare visualizations, was done in a larger search space. It was done that way to check if the new structure would be attached to the same active site of the protein. As is given in **Figure 31**, molecule number twenty is docked in another space than the ligand present in the raw file from the Protein DataBank. It should be remembered that proteins possess different active sites. Some are for agonists, while others are for antagonists, so being active against these ROR- $\gamma$  domains is not ruled out.

Used filters (QED, Lipinski's rule of five, and SYBA score) allow us to get structures that have good synthetical possibilities along with good binding energies.

There is still a necessity for confirmation by synthesis and conducted laboratory molecular docking if these structures have real affinities for selected protein domains.

This solution can be applied to other macromolecules of interest as well, where at least one active compound is known.

The generated structures have transferred weights, in the sense of chemical information, from the training dataset. Due to that, we are getting what we trained it for. Further preprocessing and selection of training data could be applied. More similar structures will possibly be generated with the use of training structures constructed with a charset that equals the charset of target structures.

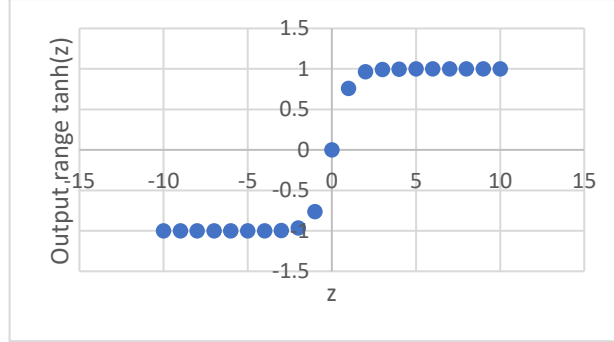
The possibility of new chemical structures' generation with the application of artificial intelligence has been proved. The machine learning model has some chemical knowledge. The second aim has been achieved with the use of python code (see attachment 35).



## Equations and methods

**Equation 3.** Tanh activation function - hyperbolic tangent [36] (see attachment 44)

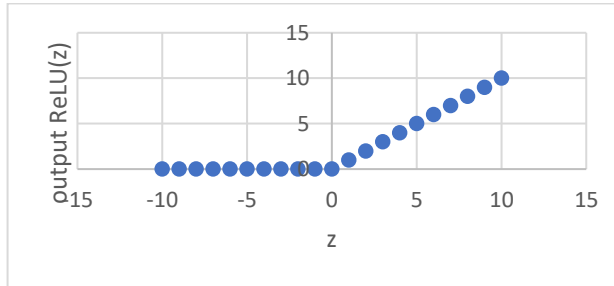
$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \text{ results in the range } (-1,1)$$



**Figure 32.** Tanh activation function visualization

**Equation 4.** ReLU activation function –rectified linear unit [37] (see attachment 44)

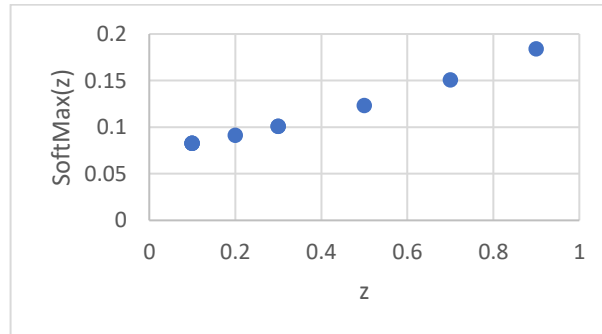
$$R(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}, \text{ results in the range } [0, \infty)$$



**Figure 33.** ReLU activation function visualization

**Equation 5.** SoftMax activation function [36] (see attachment 44)

$$\text{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K, \text{ results in the range } (0,1) - \text{probability distribution}$$



**Figure 34.** SoftMax activation function visualization

**Equation 6.** Adam optimizer general equations [38]

$$m_t = \beta_1 m_t + (1 - \beta_1) \left[ \frac{\partial L}{\partial w_t} \right] v_t = \beta_2 v_t + (1 - \beta_2) \left[ \frac{\partial L}{\partial w_t} \right]^2$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \widehat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$w_{t+1} = w_t - \widehat{m}_t \left( \frac{\alpha}{\sqrt{\widehat{v}_t + \epsilon}} \right)$ , where  $m_t$  – aggregate of gradients at the time (initially 0),  $w_t$  – weight at time  $t$ ,  $\alpha$  – learning rate (0.001),  $\partial L$  – a derivative of the loss function,  $\partial w_t$  – a derivative of the weights at time  $t$ ,  $\beta$  – moving average parameter (const, 0.9) for first and 0.999 for a second,  $\epsilon$  = A small positive constant ( $10^{-8}$ ),  $v_t$  – the sum of the square of past gradients ( $\partial L / \partial w_{t-1}$ )

This is used to update weights during the training procedures.

**Equation 7.** Categorical cross-entropy equation [35] (see attachment 44)

Loss =  $-\sum_i^{\text{output size}} y_i * \log(\widehat{y}_i)$ , where  $\widehat{y}_i$  is  $i$ -th scalar value in the model output,  $y_i$  is the corresponding target value, and output size (classes) is the number of scalar values in the model output.

**Table 5.** Categorical cross-entropy exemplary calculations

Target values	model output	Loss	Partial loss
Structure 1	Structure 1	Structure 1	
0	0.110	0.418	0.000
1	0.720		-0.143
1	0.530		-0.276
0	0.011		0.000
Structure 2		Structure 2	
1	0.560	0.293	-0.252
0	0.240		0.000
1	0.910		-0.041
0	0.110		0.000
Structure 3		Structure 3	
0	0.180	0.077	0.000
1	0.930		-0.032
1	0.920		-0.036
1	0.980		-0.009
Structure 4		Structure 4	
0	0.050	0.036	0.000
0	0.020		0.000
1	0.950		-0.022
1	0.970		-0.013

**Equation 8.** Quantitative estimation of drug-likeness [48] (see attachment 44)

$$QED_w = \exp \left[ \frac{W_{MW} \ln(d_{MW}) + W_{ALOGP} \ln(d_{ALOGP}) + W_{HBA} \ln(d_{HBA}) + W_{HBD} \ln(d_{HBD}) + W_{PSA} \ln(d_{PSA}) + W_{ROTB} \ln(d_{ROTB}) + W_{AROM} \ln(d_{AROM}) + W_{ALERTS} \ln(d_{ALERTS})}{W_{MW} + W_{ALOGP} + W_{HBA} + W_{HBD} + W_{PSA} + W_{ROTB} + W_{AROM} + W_{ALERTS}} \right], \text{ where } W_x \text{ means weight of natural}$$

logarithm of each molecular descriptor, for MW weight is 0.66 then  $- * \ln(d_{MW})$ , ALOGP weight is 0.46 then  $- * \ln(d_{ALOGP})$ , HBA weight is 0.05  $* \ln(d_{HBA})$ , HBD weight is 0.61 then  $- * \ln(d_{HBD})$ , PSA weight is 0.06 then  $- * \ln(d_{PSA})$ , ROTB weight is 0.65 then  $- * \ln(d_{ROTB})$ , AROM weight is 0.48 then  $- * \ln(d_{AROM})$ , ALERTS weight is 0.95 then  $- * \ln(d_{ALERTS})$ . Functions of desirability for each molecular descriptor:

$$d_x = a + \frac{b}{\left[ 1 + \exp \left( - \frac{x - c + \frac{d}{2}}{e} \right) \right]} \left[ 1 - \frac{1}{1 + \exp \left( - \frac{x - c - \frac{d}{2}}{f} \right)} \right]$$

$$d_{MW} = 2.817 + \frac{392.575}{\left[ 1 + \exp \left( - \frac{MW - 290.749 + \frac{2.420}{2}}{49.223} \right) \right]} \left[ 1 - \frac{1}{1 + \exp \left( - \frac{MW - 290.749 - \frac{2.420}{2}}{65.371} \right)} \right]$$

$$d_{ALOGP} = 3.173 + \frac{137.862}{\left[ 1 + \exp \left( - \frac{ALOGP - 2.535 + \frac{4.581}{2}}{0.823} \right) \right]} \left[ 1 - \frac{1}{1 + \exp \left( - \frac{ALOGP - 2.535 - \frac{4.581}{2}}{0.576} \right)} \right]$$

$$d_{HBA} = 2.949 + \frac{160.461}{\left[ 1 + \exp \left( - \frac{HBA - 3.615 + \frac{4.436}{2}}{0.290} \right) \right]} \left[ 1 - \frac{1}{1 + \exp \left( - \frac{HBA - 3.615 - \frac{4.436}{2}}{1.301} \right)} \right]$$

$$d_{HBD} = 1.619 + \frac{1010.051}{\left[ 1 + \exp \left( - \frac{HBD - 0.985 + \frac{10^{-9}}{2}}{0.714} \right) \right]} \left[ 1 - \frac{1}{1 + \exp \left( - \frac{HBD - 0.985 - \frac{10^{-9}}{2}}{0.921} \right)} \right]$$

$$d_{\text{PSA}} = 1.877 + \frac{125.223}{\left[1 + \exp\left(-\frac{\text{PSA} - 62.908 + \frac{87.834}{2}}{12.020}\right)\right]} \left[1 - \frac{1}{1 + \exp\left(-\frac{\text{PSA} - 62.908 - \frac{87.834}{2}}{28.513}\right)}\right]$$

$$d_{\text{ROTB}} = 0.010 + \frac{272.412}{\left[1 + \exp\left(-\frac{\text{ROTB} - 2.558 + \frac{1.566}{2}}{1.272}\right)\right]} \left[1 - \frac{1}{1 + \exp\left(-\frac{\text{ROTB} - 2.558 - \frac{1.566}{2}}{2.758}\right)}\right]$$

$$d_{\text{AROM}} = 3.218 + \frac{957.737}{\left[1 + \exp\left(-\frac{\text{AROM} - 2.275 + \frac{10^{-9}}{2}}{1.318}\right)\right]} \left[1 - \frac{1}{1 + \exp\left(-\frac{\text{AROM} - 2.275 - \frac{10^{-9}}{2}}{0.376}\right)}\right]$$

$$d_{\text{ALERTS}} = 0.010 + \frac{1199.094}{\left[1 + \exp\left(-\frac{\text{ALERT} + 0.090 + \frac{10^{-9}}{2}}{0.186}\right)\right]} \left[1 - \frac{1}{1 + \exp\left(-\frac{\text{ALERT} + 0.090 - \frac{10^{-9}}{2}}{0.875}\right)}\right]$$

Each of  $d_x$  value is then compared to the maximal possible  $d_{\text{max}}$ . The results are used during final QED calculations.

Example: QED for aspirin: MW = 180.16 g/mol; ALOGP = 1.31; HBA = 4; HBD = 1; PSA = 63.6 Å<sup>2</sup>; ROTB = 3, AROM = 1; ALERTS = 2.

$$\text{QED}_{\text{aspirin}} = \exp \left[ \frac{\begin{matrix} 0.66 \ln(0.337) + 0.46 \ln(0.846) + \\ 0.05 \ln(0.886) + 0.61 \ln(0.986) + \\ 0.06 \ln(0.976) + 0.65 \ln(0.992) + \\ 0.48 \ln(0.827) + 0.95 \ln(0.241) \end{matrix}}{\begin{matrix} 0.66 + 0.46 + 0.05 + 0.61 + \\ 0.06 + 0.65 + 0.48 + 0.95 \end{matrix}} \right] = 0.56$$

### Equation 9. Lipinski's rule of five [49]

Molecular descriptors set – Molecular weight – MW, octanol-water partition coefficient – LogP, number of hydrogen donors – HBD, number of hydrogen acceptors – HBA, number of rotatable bonds – ROTB.

To pass this test molecule should fulfill the below conditions:

$$\text{MW} \leq 500 \text{ g/mol}, \text{LogP} \leq 5, \text{HBD} \leq 5, \text{HBA} \leq 10, \text{ROTB} \leq 5$$

Example: Aspirin: MW = 180.16 g/mol; LOGP = 1.31; HBA = 4; HBD = 1; ROTB = 3

MW(aspirin) < 500, LogP < 5, HBA <10, HBD <5, ROTB < 5  
Aspirin passes this filtration.

**Equation 10.** Normalization function

$$\text{Normalized value} = \frac{\text{value} - \text{minimal value in dataset}}{\text{maximal value in dataset} - \text{minimal value in dataset}}$$

Example: six elements are given 3, 8, 15, 32, 12, 45.

Normalization results are:

$$\begin{aligned} \text{Normalized value}(3) &= \frac{3 - 3}{45 - 3} = 0.000 & \text{Normalized value}(32) &= \frac{32 - 3}{45 - 3} = 0.690 \\ \text{Normalized value}(8) &= \frac{8 - 3}{45 - 3} = 0.119 & \text{Normalized value}(12) &= \frac{12 - 3}{45 - 3} = 0.214 \\ \text{Normalized value}(15) &= \frac{15 - 3}{45 - 3} = 0.286 & \text{Normalized value}(45) &= \frac{45 - 3}{45 - 3} = 1.000 \end{aligned}$$

**Equation 11.** Node output general function [32]

Output = activation function(z);  $z = \sum_i w_i x_i + b$ , where  $w_i$  is the weight for each input,  $x_i$  is i-th input and b – a bias.

Example: ReLU as activation function  $w_1 = 0.2$ ,  $x_1 = 2$ ,  $w_2 = 0.1$ ,  $x_2 = 5$ ,  $b = 0.5$

$$z = 0.2*2 + 0.1*5 + 0.5 = 0.45$$

$$\text{ReLU}(z) = 0.45$$

$$\text{Tanh}(z) = 0.42$$

**Method 1.** SYBA classifier – a SYnthetic Bayesian Accessibility classifier is a tool that classifies organic compounds as easy-to-synthesize (ES) or hard-to-synthesize (HS). This algorithm is a fragment-based method. The analyzed molecule is decomposed into ECFP4-like fragments, and a score is assigned to each fragment then all scores are summed. If the resultant score is positive – the structure is considered to be easy-to-synthesized [50]. Each compound is represented by a binary fingerprint  $F = [f_1, f_2, f_3, \dots, f_M]$  of length M.  $f_i$  indicates the presence (1) or absence (0) of the specific fragment i in the compound. This fingerprint is used to assign the molecule to a class  $C \in \langle \text{ES}, \text{HS} \rangle$ . The Bayesian theorem is used  $p(C|F) = \frac{p(F|C)p(C)}{p(F)}$ , where  $p(C|F)$  is the posterior probability that a compound with

a certain set of molecular fragments  $F$  belongs to class  $C$ . The likelihood  $p(F|C)$  is the conditional probability that a compound from the class  $C$  contains a set of molecular fragments  $F$ . The marginal probabilities  $p(F)$  and  $p(C)$  express our belief to see a set of molecular fragments  $F$  and the molecule that belongs to the class  $C$ .

SYBA score is calculated by use of the equation shown below [51].

$$\text{SYBA}(F) = \sum_{i=1}^M \ln \left( \frac{p(f_i|ES)}{p(f_i|HS)} \right)$$

**Method 2.** LSTM cell from a mathematical point of view. [34]

Three types of gates are distinguished: input, forget and output gate. All they are sigmoid ( $\text{sigmoid}(t) = \frac{1}{1+e^{-t}}$ ), activation functions, so the output is in the range from 0 to 1. This is used to fulfill the necessity of positive output; this is due to fact that the answer is whether the particular feature should be kept or discarded. Zero as a result blocks the gate and one allows passing information through it.

The equations for each gate are given below:

$i_t = \delta(w_i[h_{t-1}, x_t] + b_i)$  – for the input gate, the latest information to be stored,

$f_t = \delta(w_f[h_{t-1}, x_t] + b_f)$  – for the forget gate, throwing away information from the cell,

$o_t = \delta(w_o[h_{t-1}, x_t] + b_o)$  – for the output gate, activation supplying and final output creation at given timestamp “t”

$\delta$  – stands for sigmoid function,  $w_x$  – weight for the respective gate(x),  $h_{t-1}$  – output of the previous LSTM block,  $x_t$  – input at the current timestamp,  $b_x$  – biases for respective gate

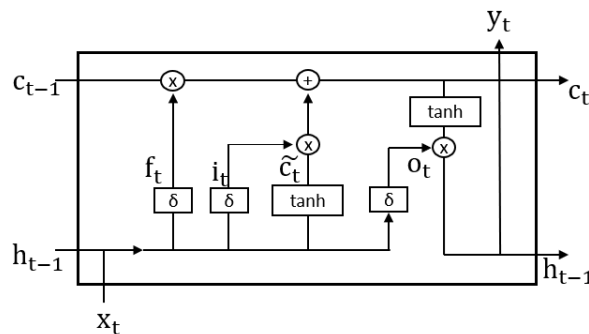
Equations for the cell state, candidate cell state, and final output:

$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c)$ ;  $\tilde{c}_t$  – candidate for cell state at timestamp

$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$ ;  $c_t$  – cell state at timestamp

$h_t = o_t * \tanh(c_t)$ ;  $h_t$  – hidden state at timestamp

To get the output SoftMax activation is applied:  $\text{Output} = \text{softmax}(h_t)$



**Figure 35.** LSTM scheme

**Method 3.** Tanimoto similarity [31] (see attachment 44)

Computation is done as an inverse of the distance of descriptor space measurement. For purpose of this work, molecular fingerprints are compared.

$Tc(A, B) = \frac{c}{a+b-c}$ , where a and b are representing several features present in compounds A and B and c is the number of features that are common for both.

This means that in the case of fingerprints usage feature means on-bits numbers similarly to arrays used when molecular sequences are transformed into numerical arrays.

Output is given in the range from 0 to 1.

Structures are considered to be similar if  $T > 0.85$ , but this does not give information about possible similar bioactivity, this parameter depends on many more variables.

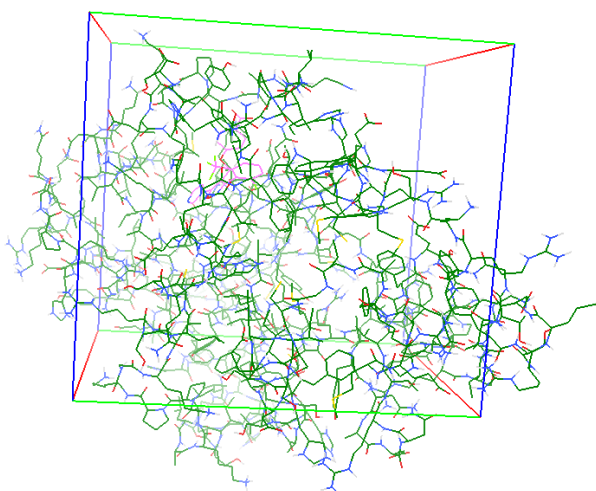
Should be mentioned that one as the outcome does not necessarily mean that our structures are identical, it means that they have the same fingerprints.



**Figure 36.** Similarity between two structures; tridecan-1-amine and [(pentylamino)methyl](tridecyl)amine Structures are transformed into molecular objects (RDkit library [30]) and corresponding fingerprints are created then they are compared, and the result is  $Tc(A, B) = 0$ . (44).

**Method 4.** Molecular docking visualization [23]

In figure shown below search space of molecular docking is visualized. This is where the ligand will be attached to the macromolecule.



**Figure 37.** Search space in molecular docking for 7NPC with native ligand visualization inside which grids are calculated and used genetic algorithm searches for best ligand pose

## Attachments

### Files

The architecture of the seq\_to\_seq\_and\_dock\_AMU folder is given inside the arch.txt file.

Attachment 1.	ZINC-downloader-2D-smi.wget
Attachment 2.	Data_preparation.ipynb
Attachment 3.	zinc20_selected_to_create_model_processed.parquet
Attachment 4.	Molecule_generator-Generative_neural_network.py
Attachment 5.	SELFIES_to_mol_seq.json
Attachment 6.	mol_seq_to_SELFIES.json
Attachment 7.	mol_seq_to_int.json
Attachment 8.	int_to_mol_seq.json
Attachment 9.	mol_seq2lat_ZINC.h5
Attachment 10.	lat2state_ZINC.h5
Attachment 11.	samplemodel_ZINC.h5
Attachment 12.	RORgamma_active_compounds.xlsx
Attachment 13.	ROR_gamma_active_QED_Lipinski.ipynb
Attachment 14.	ROR_gamma_active_QED_Lipinski.xlsx
Attachment 15.	SYBA_class_ROR_gamma_activ_AFTER_QED_LIPINSKI.ipynb
Attachment 16.	Prediction_initializers_Kin_Inh.xlsx
Attachment 17.	Prediction_1_0.1_tensor_scaling.ipynb
Attachment 18.	Prediction_2_0.2_tensor_scaling.ipynb
Attachment 19.	Molecules_generated_tensor_scaling_0_1.xlsx
Attachment 20.	Molecules_generated_tensor_scaling_0_2.xlsx
Attachment 21.	Selection_from_0_1_tensor_scaling.ipynb
Attachment 22.	Selection_from_0_2_tensor_scaling.ipynb
Attachment 23.	Selected_molecules_from_0_1_tensor_scaling.xlsx
Attachment 24.	Selected_molecules_from_0_2_tensor_scaling.xlsx
Attachment 25.	Combine_Generated_and_Selected_structures.ipynb
Attachment 26.	Assigning_prediction_mode_to_selected_SMILES.ipynb
Attachment 27.	All_generated_SMILES_QED_Lipinski.xlsx
Attachment 28.	SYBA_classifier_additional_filter.ipynb
Attachment 29.	All_generated_SMILES_SYBA_filtration.xlsx
Attachment 30.	All_generated_SMILES_visualization.ipynb
Attachment 31.	My_score_to_final_structures.ipynb
Attachment 32.	Tanimoto_similarity-SYBA_selection.ipynb
Attachment 33.	Tanimoto_similarity_All_generated_and_selected.ipynb
Attachment 34.	Checking_if_ROR_gamma_activ_are_in_ZINC_db.ipynb
Attachment 35.	Python_molecular_docking.py
Attachment 36.	dockingResults_ROR_gamma_SYBA_selected.xlsx
Attachment 37.	Clean_results.ipynb
Attachment 38.	Selection_of_most_prominent_structures.ipynb
Attachment 39.	SMILES_to_3D_PDB.ipynb
Attachment 40.	UniCode_char.csv
Attachment 41.	SELFIES_coder.py
Attachment 42.	dockingResults_ROR_gamma_SYBA_CLEAN.xlsx
Attachment 43.	Reference_dockings folder
Attachment 44.	master_thesis_calculations.xlsx

All files with description of usage can be found:

[https://github.com/XDamianX-coder/seq\\_to\\_seq\\_and\\_dock\\_AMU](https://github.com/XDamianX-coder/seq_to_seq_and_dock_AMU)



## Acknowledgments

The author thanks the Head of Molecular Modeling Laboratory of the Institute of Medical Biology Ph.D. Rafal Bachorz for giving a possibility of calculations inducement and for invaluable help during code preparation. Special thanks to Ph.D. Esben Jannik Bjerrum, AstraZeneca's Principal Scientist, for the concept of the predictive model. The author thanks also Professor Marcin Hoffmann for all the comments and discussions that lead to the publication of this paper.

## References

1. Kotsias, Panagiotis-Christos, Josep Arús-Pous, Hongming Chen, Ola Engkvist, Christian Tyrchan, and Esben Jannik Bjerrum. "Direct Steering of de Novo Molecular Generation with Descriptor Conditional Recurrent Neural Networks." *Nature Machine Intelligence* 2, no. 5 (May 2020): 254–65. <https://doi.org/10.1038/s42256-020-0174-5>.
2. Segler, Marwin H. S., Thierry Kogej, Christian Tyrchan, and Mark P. Waller. "Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks." *ACS Central Science* 4, no. 1 (January 24, 2018): 120–31. <https://doi.org/10.1021/acscentsci.7b00512>.
3. Xu, Yinqiu, Xuanyi Li, Hequan Yao, and Kejiang Lin. "Neural Networks in Drug Discovery: Current Insights from Medicinal Chemists." *Future Medicinal Chemistry* 11, no. 14 (July 2019): 1669–72. <https://doi.org/10.4155/fmc-2019-0118>.
4. Homans, Steve W. "NMR Spectroscopy Tools for Structure-Aided Drug Design." *Angewandte Chemie (International Ed. in English)* 43, no. 3 (January 3, 2004): 290–300. <https://doi.org/10.1002/anie.200300581>.
5. Berman, Helen M., Tammy Battistuz, T. N. Bhat, Wolfgang F. Bluhm, Philip E. Bourne, Kyle Burkhardt, Zukang Feng, et al. "The Protein Data Bank." *Acta Crystallographica. Section D, Biological Crystallography* 58, no. Pt 6 No 1 (June 2002): 899–907. <https://doi.org/10.1107/s0907444902003451>.
6. Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. "The Protein Data Bank." *Nucleic Acids Research* 28, no. 1 (January 1, 2000): 235–42. <https://doi.org/10.1093/nar/28.1.235>.
7. Plewczynski, Dariusz, Michał Łażniewski, Rafał Augustyniak, and Krzysztof Ginalski. "Can We Trust Docking Results? Evaluation of Seven Commonly Used Programs on PDBbind Database." *Journal of Computational Chemistry* 32, no. 4 (March 2011): 742–55. <https://doi.org/10.1002/jcc.21643>.
8. Morris, Garrett M., Ruth Huey, William Lindstrom, Michel F. Sanner, Richard K. Belew, David S. Goodsell, and Arthur J. Olson. "AutoDock4 and AutoDockTools4: Automated Docking with

- Selective Receptor Flexibility.” *Journal of Computational Chemistry* 30, no. 16 (December 2009): 2785–91. <https://doi.org/10.1002/jcc.21256>.
9. Jenkins, A. D., P. Kratochvíl, R. F. T. Stepto, and U. W. Suter. “Glossary of Basic Terms in Polymer Science (IUPAC Recommendations 1996).” *Pure and Applied Chemistry* 68, no. 12 (January 1, 1996): 2287–2311. <https://doi.org/10.1351/pac199668122287>.
  10. Denaturation characterization: <https://www.biologyonline.com/dictionary/denaturation> [29.12.2021].
  11. Berg, Jeremy M., John L. Tymoczko, Lubert Stryer, and Lubert Stryer. *Biochemistry*. 5th ed. New York: W.H. Freeman, 2002.
  12. Hirose, T., R. J. Smith, and A. M. Jetten. “ROR Gamma: The Third Member of ROR/RZR Orphan Receptor Subfamily That Is Highly Expressed in Skeletal Muscle.” *Biochemical and Biophysical Research Communications* 205, no. 3 (December 30, 1994): 1976–83. <https://doi.org/10.1006/bbrc.1994.2902>.
  13. Benoit, Gérard, Austin Cooney, Vincent Giguere, Holly Ingraham, Mitch Lazar, George Muscat, Thomas Perlmann, et al. “International Union of Pharmacology. LXVI. Orphan Nuclear Receptors.” *Pharmacological Reviews* 58, no. 4 (December 2006): 798–836. <https://doi.org/10.1124/pr.58.4.10>.
  14. Medvedev, A., Z. H. Yan, T. Hirose, V. Giguère, and A. M. Jetten. “Cloning of a cDNA Encoding the Murine Orphan Receptor RZR/ROR Gamma and Characterization of Its Response Element.” *Gene* 181, no. 1–2 (November 28, 1996): 199–206. [https://doi.org/10.1016/s0378-1119\(96\)00504-5](https://doi.org/10.1016/s0378-1119(96)00504-5).
  15. Zhang, Yan, Xiao-yu Luo, Dong-hai Wu, and Yong Xu. “ROR Nuclear Receptors: Structures, Related Diseases, and Drug Discovery.” *Acta Pharmacologica Sinica* 36, no. 1 (January 2015): 71–87. <https://doi.org/10.1038/aps.2014.120>.
  16. Zhang, Xiong, Zenghong Huang, Junjian Wang, Zhao Ma, Joy Yang, Eva Corey, Christopher P. Evans, Ai-Ming Yu, and Hong-Wu Chen. “Targeting Feedforward Loops Formed by Nuclear Receptor ROR $\gamma$  and Kinase PBK in MCRPC with Hyperactive AR Signaling.” *Cancers* 13, no. 7 (April 1, 2021): 1672. <https://doi.org/10.3390/cancers13071672>.
  17. Pharmacological glossary <https://www.tocris.com/resources/pharmacological-glossary> [29.12.2021]
  18. Huh, Jun R., and Dan R. Littman. “Small Molecule Inhibitors of ROR $\gamma$ t: Targeting Th17 Cells and Other Applications.” *European Journal of Immunology* 42, no. 9 (September 2012): 2232–37. <https://doi.org/10.1002/eji.201242740>.
  19. Study of LYC-55716 in Adult Subjects With Locally Advanced or Metastatic Cancer - <https://clinicaltrials.gov/ct2/show/NCT02929862> [29.12.2021]
  20. Meijer, Femke A., Annet O. W. M. Saris, Richard G. Doveston, Guido J. M. Oerlemans, Rens M. J. M. de Vries, Bente A. Somsen, Anke Unger, et al. “Structure-Activity Relationship Studies of Trisubstituted Isoxazoles as Selective Allosteric Ligands for the Retinoic-Acid-Receptor-Related Orphan Receptor  $\Gamma$ t.” *Journal of Medicinal Chemistry* 64, no. 13 (July 8, 2021): 9238–58. <https://doi.org/10.1021/acs.jmedchem.1c00475>.
  21. Ruan, Zheming, Peter K. Park, Donna Wei, Ashok Purandare, Honghe Wan, Daniel O’Malley, Sylwia Stachura, et al. “Substituted Diaryl Ether Compounds as Retinoic Acid-Related Orphan Receptor- $\Gamma$ t (ROR $\gamma$ t) Agonists.” *Bioorganic & Medicinal Chemistry Letters* 35 (March 1, 2021): 127778. <https://doi.org/10.1016/j.bmcl.2021.127778>.

22. Pan, Albert C., David W. Borhani, Ron O. Dror, and David E. Shaw. "Molecular Determinants of Drug-Receptor Binding Kinetics." *Drug Discovery Today* 18, no. 13–14 (July 2013): 667–73. <https://doi.org/10.1016/j.drudis.2013.02.007>.
23. AutoDock UserGuide [15.02.2022]  
[https://autodock.scripps.edu/wp-content/uploads/sites/56/2021/10/AutoDock4.2.6\\_UserGuide.pdf](https://autodock.scripps.edu/wp-content/uploads/sites/56/2021/10/AutoDock4.2.6_UserGuide.pdf)
24. Bayly, Christopher I., Piotr Cieplak, Wendy Cornell, and Peter A. Kollman. "A Well-Behaved Electrostatic Potential Based Method Using Charge Restraints for Deriving Atomic Charges: The RESP Model." *The Journal of Physical Chemistry* 97, no. 40 (October 1993): 10269–80. <https://doi.org/10.1021/j100142a004>.
25. Mortier, Wilfried J., Karin Van Genechten, and Johann Gasteiger. "Electronegativity Equalization: Application and Parametrization." *Journal of the American Chemical Society* 107, no. 4 (February 1985): 829–35. <https://doi.org/10.1021/ja00290a017>.
26. Mitchell, Melanie. *An Introduction to Genetic Algorithms*. Complex Adaptive Systems. Cambridge, Mass: MIT Press, 1996.
27. Ross, Brian. "A Lamarckian Evolution Strategy for Genetic Algorithms." In *Practical Handbook of Genetic Algorithms*, edited by Lance Chambers. CRC Press, 1998. <https://doi.org/10.1201/9781420050080.ch1>.
28. Weininger, David. "SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules." *Journal of Chemical Information and Modeling* 28, no. 1 (February 1, 1988): 31–36. <https://doi.org/10.1021/ci00057a005>.
29. Krenn, Mario, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. "Self-Referencing Embedded Strings (SELFIES): A 100% Robust Molecular String Representation," 2019. <https://doi.org/10.48550/ARXIV.1905.13741>.
30. RDKit: Open-source cheminformatics; <http://www.rdkit.org> [01.02.2022]
31. Maggiora, Gerald, Martin Vogt, Dagmar Stumpfe, and Jürgen Bajorath. "Molecular Similarity in Medicinal Chemistry: Miniperspective." *Journal of Medicinal Chemistry* 57, no. 8 (April 24, 2014): 3186–3204. <https://doi.org/10.1021/jm401411z>.
32. Géron, Aurélien, and an O'Reilly Media Company Safari. *Uczenie Maszynowe z Użyciem Scikit-Learn i TensorFlow*, 2020.
33. Xu, Zheng, Sheng Wang, Feiyun Zhu, and Junzhou Huang. "Seq2seq Fingerprint: An Unsupervised Deep Molecular Embedding for Drug Discovery." In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 285–94. Boston Massachusetts USA: ACM, 2017. <https://doi.org/10.1145/3107411.3107424>.
34. Brownlee, J. *Long Short-Term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning*. Jason Brownlee, 2017.
35. Categorical cross-entropy: <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy> [21.03.2022]
36. Activations functions [https://en.wikipedia.org/wiki/Activation\\_function](https://en.wikipedia.org/wiki/Activation_function) [21.03.2022]
37. Chollet, François. *Deep Learning with Python*. Shelter Island, New York: Manning Publications Co, 2018.

38. Kingma, Diederik P., and Jimmy Ba. "Adam: A Method for Stochastic Optimization," 2014. <https://doi.org/10.48550/ARXIV.1412.6980>.
39. Dar, Ayaz Mahmood, and Shafia Mir. "Molecular Docking: Approaches, Types, Applications and Basic Challenges." *Journal of Analytical & Bioanalytical Techniques* 08, no. 02 (2017). <https://doi.org/10.4172/2155-9872.1000356>.
40. Atkins, P. W., and Julio De Paula. *Physical Chemistry for the Life Sciences*. Oxford, UK : New York: Oxford University Press ; W.H. Freeman, 2006.
41. Bjerrum, Esben Jannik, and Richard Threlfall. "Molecular Generation with Recurrent Neural Networks (RNNs)," 2017. <https://doi.org/10.48550/ARXIV.1705.04612>.
42. ZINC database tranches <https://zinc20.docking.org/tranches/home/> [04.04.2022]
43. Irwin, John J., Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. "ZINC: A Free Tool to Discover Chemistry for Biology." *Journal of Chemical Information and Modeling* 52, no. 7 (July 23, 2012): 1757–68. <https://doi.org/10.1021/ci3001277>.
44. Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, et al. "Array Programming with NumPy." *Nature* 585, no. 7825 (September 2020): 357–62. <https://doi.org/10.1038/s41586-020-2649-2>.
45. Chollet, Francois and others. "Keras." GitHub, 2015. <https://github.com/fchollet/keras>.
46. Graff, David E., and Connor W. Coley. "Pyscreener: A Python Wrapper for Computational Docking Software," 2021. <https://doi.org/10.48550/ARXIV.2112.10575>.
47. PubChemPy open software cheminformatics; <https://github.com/mcs07/PubChemPy>
48. Bickerton, G. Richard, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. "Quantifying the Chemical Beauty of Drugs." *Nature Chemistry* 4, no. 2 (January 24, 2012): 90–98. <https://doi.org/10.1038/nchem.1243>.
49. Lipinski, C. A., F. Lombardo, B. W. Dominy, and P. J. Feeney. "Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings." *Advanced Drug Delivery Reviews* 46, no. 1–3 (March 1, 2001): 3–26. [https://doi.org/10.1016/s0169-409x\(00\)00129-0](https://doi.org/10.1016/s0169-409x(00)00129-0).
50. Voršilák, Milan, and Daniel Svozil. "Nonpher: Computational Method for Design of Hard-to-Synthesize Structures." *Journal of Cheminformatics* 9, no. 1 (March 20, 2017): 20. <https://doi.org/10.1186/s13321-017-0206-2>.
51. Voršilák, Milan, Michal Kolář, Ivan Čmelo, and Daniel Svozil. "SYBA: Bayesian Estimation of Synthetic Accessibility of Organic Compounds." *Journal of Cheminformatics* 12, no. 1 (December 2020): 35. <https://doi.org/10.1186/s13321-020-00439-2>.
52. Karaś, Kaja, Anna Sałkowska, Iwona Karwaciak, Aurelia Walczak-Drzewiecka, Jarosław Dastyh, Rafał A. Bachorz, and Marcin Ratajewski. "The Dichotomous Nature of AZ5104 (an EGFR Inhibitor) Towards ROR $\gamma$  and ROR $\gamma$ T." *International Journal of Molecular Sciences* 20, no. 22 (November 17, 2019): E5780. <https://doi.org/10.3390/ijms20225780>.
53. Reback, Jeff, Wes McKinney, Jbrockmendel, Joris Van Den Bossche, Tom Augspurger, Phillip Cloud, Gfyoung, et al. *Pandas-Dev/Pandas: Pandas 1.2.1* (version v1.2.1). Zenodo, 2021. <https://doi.org/10.5281/ZENODO.4452601>.