ITMO UNIVERSITY

INFOCHEMISTRY SCIENTIFIC CENTER

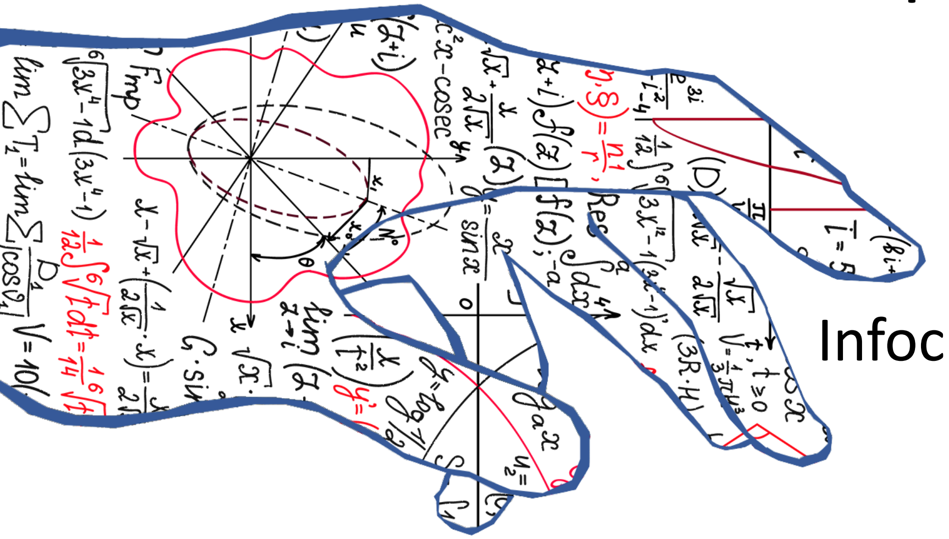
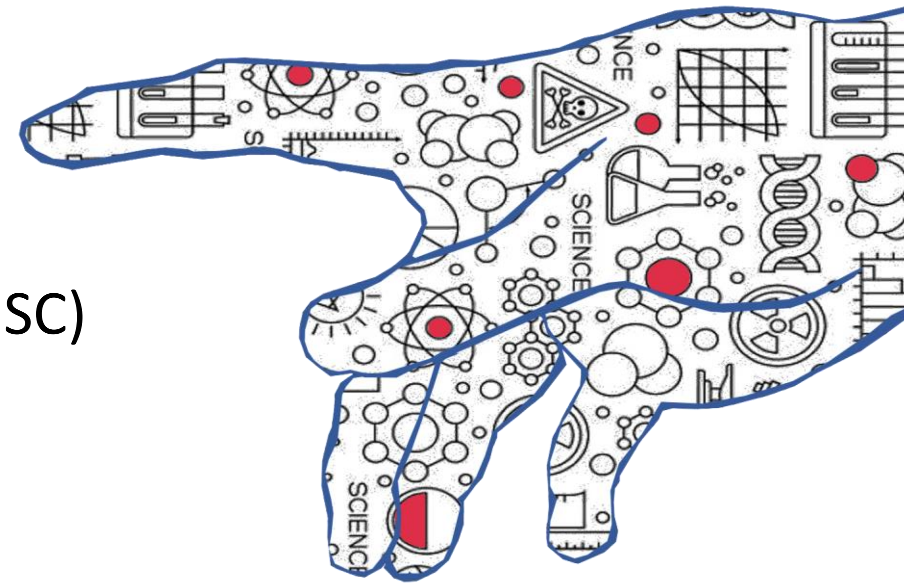
Cheminformatics and synthetic biology: computational methods and projects

Prof. Sergey Shityakov

Infochemistry Scientific Center (ISC)

ITMO University

Saint-Petersburg, 2024

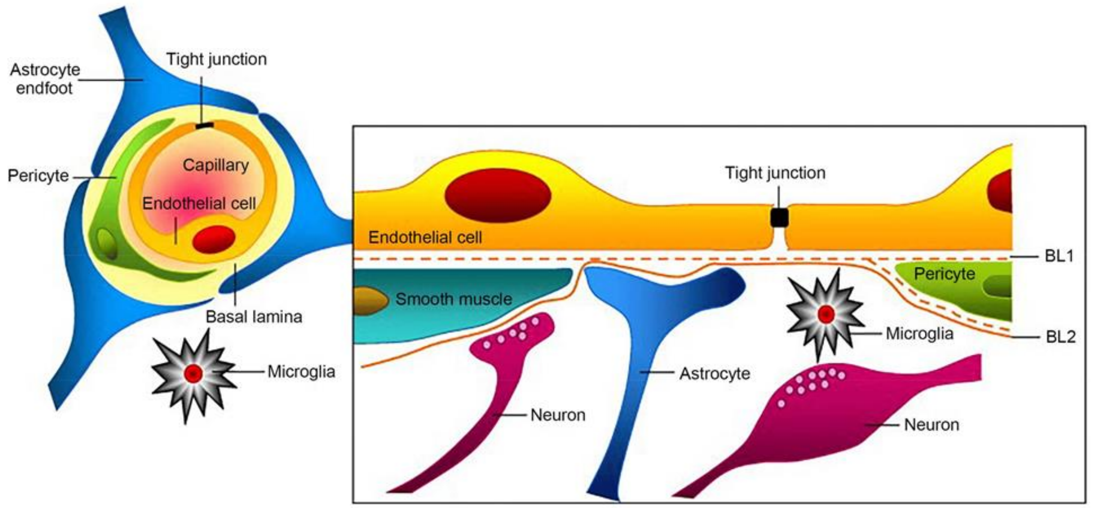
State of the art

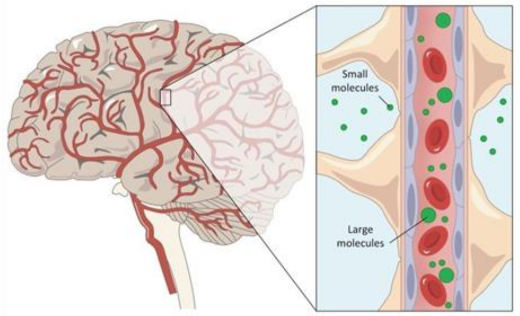
- Blood-brain barrier (BBB) is a semi–permeable barrier between blood and nervous tissue that protects the central nervous system (CNS) from the penetration of undesirable substances

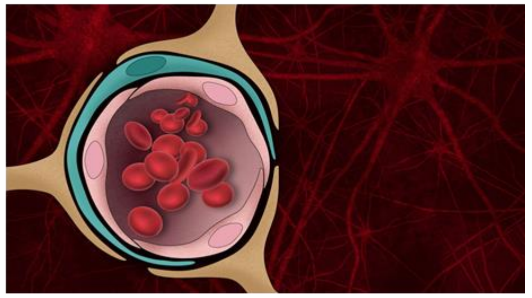
- **The ability to penetrate BBB is one of the most important parameters in pharmacology**: all drugs that affect the central nervous system should easily overcome BBB. Some other classes of drugs, on the contrary, should overcome the BBB as badly as possible in order to avoid side effects

- It is very difficult to measure the permeability experimentally.

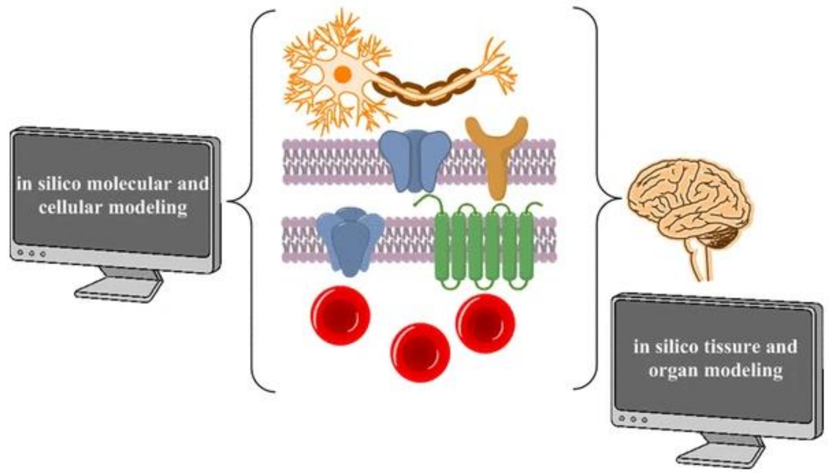
- **Therefore, scientists have high hopes for machine learning to predict BBB penetration by drugs before difficult and time-consuming experiments.**

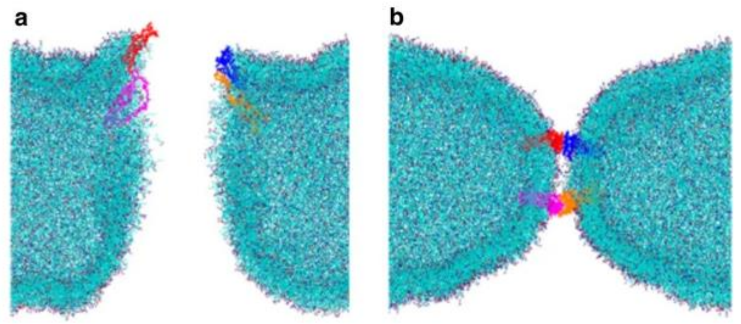


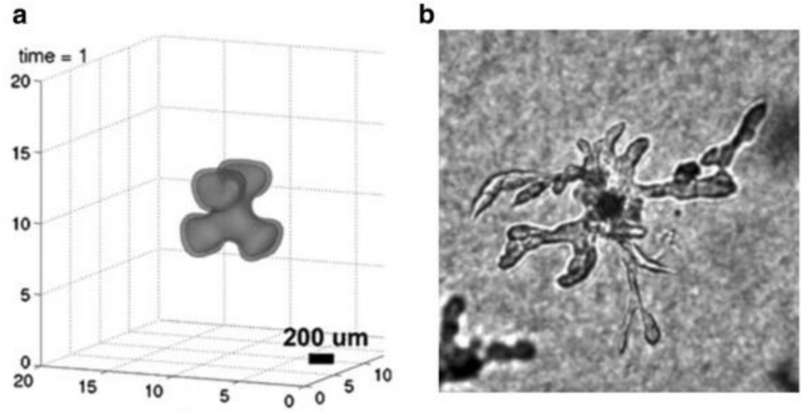


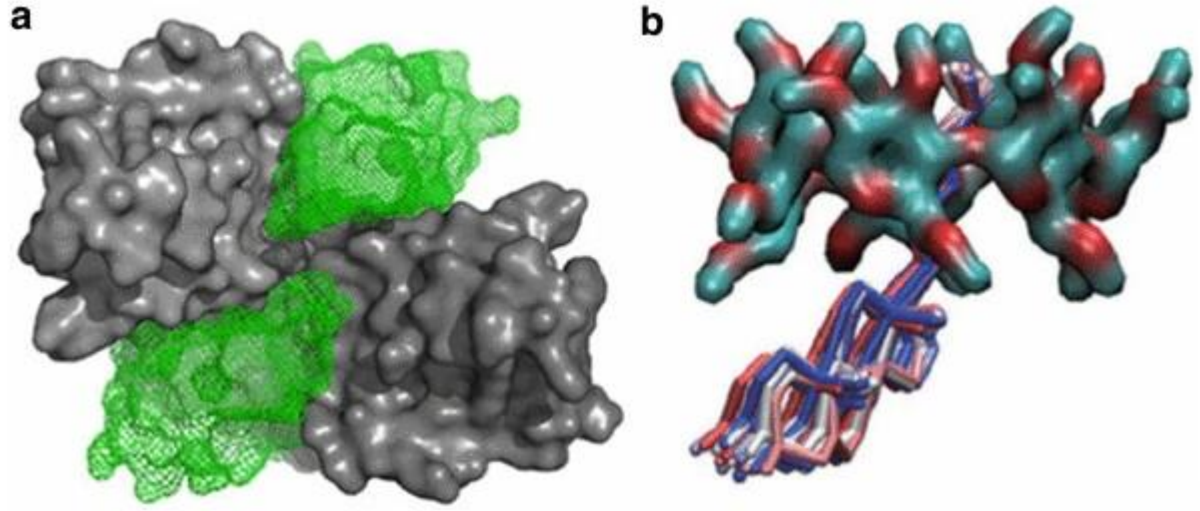


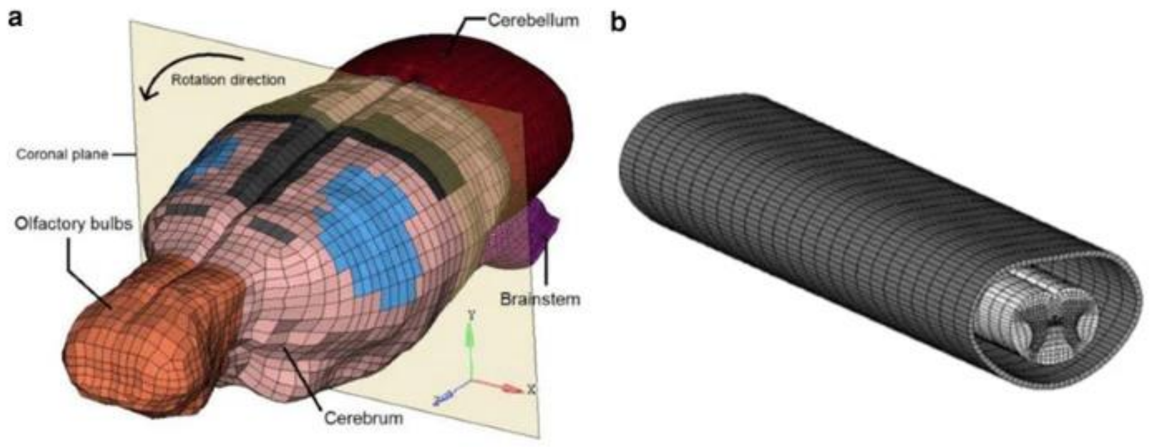
Computational methods


in silico molecular and cellular modeling
in silico tissure and organ modeling

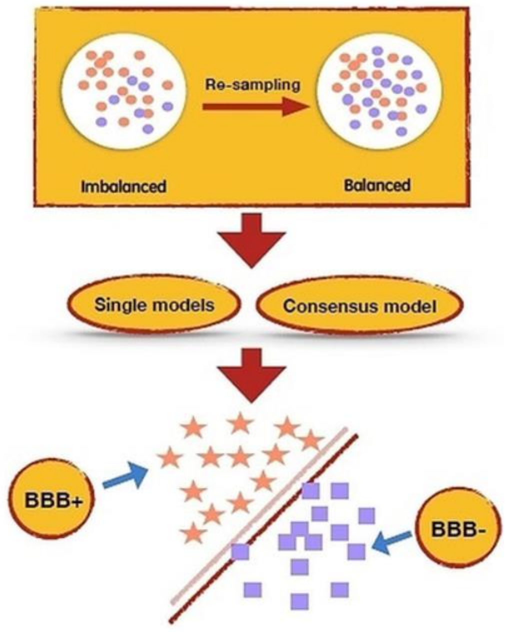






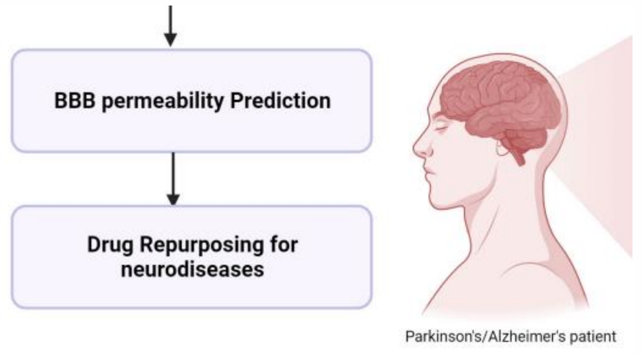


ML classification task: whether the molecule penetrates BBB



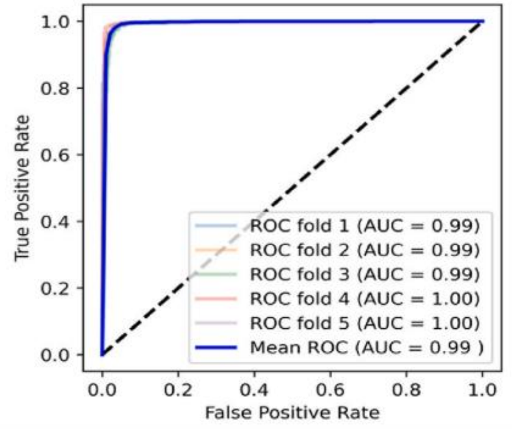
Wang et. al (2018)

- ✓ ROC_AUC_{cv}= 0.919*
- ✓ Publicly available
- ✗ Relatively small dataset (2358 compounds)
- ✗ Synthetic data was used to overcome class imbalance issue (synthetic data may not capture the intricacies of real-world data)


BBB permeability Prediction
Drug Repurposing for neurodiseases
Parkinson's/Alzheimer's patient

Ansari et. al (2022)

- ✓ ROC_AUC_{cv} = 0.96
- ✓ Relatively large dataset provided (7807 compounds)
- ✗ Publicly unavailable
- ✗ Synthetic data was used to overcome class imbalance issue



Mazumdar et. al (2023)

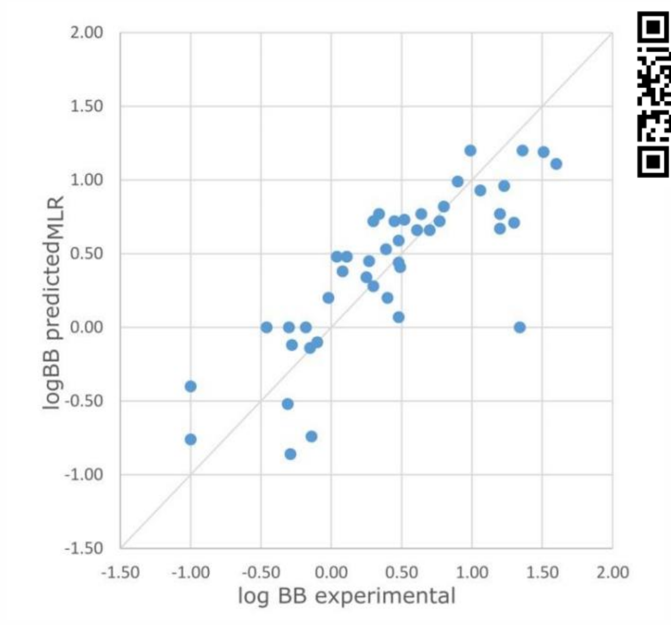
- ✓ ROC_AUC_{cv} = 0.99
- ✓ Relatively large dataset provided (8153 compounds)
- ✗ Publicly unavailable

Main problem: models with highest evaluation metrics are publicly unavailable, so it is impossible to reproduce their experiments and use these models for your own research

**ideal value is 1*

ITsMOre than a UNIVERSITY

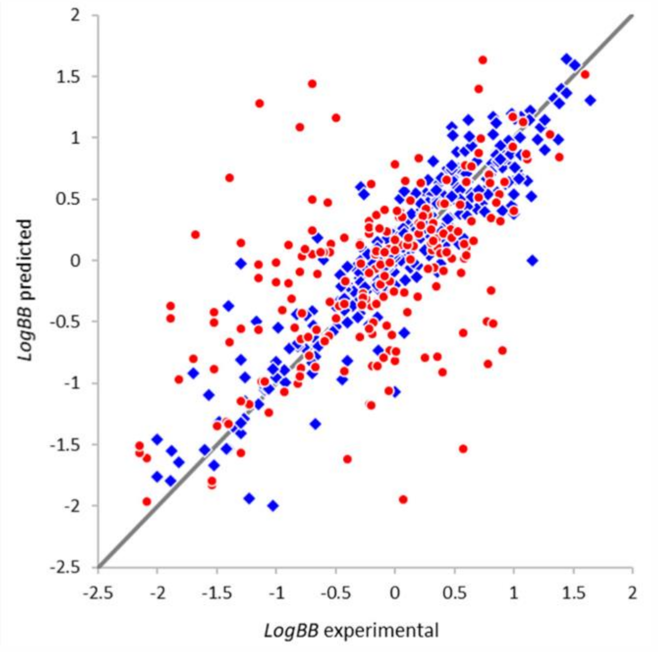
ML regression task: prediction of the concrete logBB value



Ciura et. al (2020)

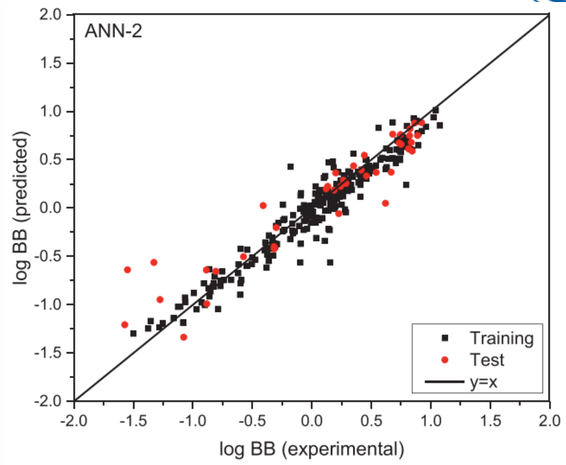
- ✓ $R^2 = 0.759$, $Q^2 = 0.731$, $RMSE_{cv} = 0.31$*
- ✗ Extremely small dataset (45 compounds)
- ✗ Publicly unavailable

**ideal values of R^2, Q^2 and $RMSE_{cv}$ are 1, 1 and 0 respectively*



Radchenko et. al (2020)

- ✓ $Q^2 = 0.816$, $RMSE_{cv} = 0.318$
- ✓ Publicly available
- ✗ Small dataset (529 compounds)



Wu et. al (2021)

- ✓ $RMSE_{test} = 0.236$
- ✗ Publicly unavailable
- ✗ Small dataset (300 compounds)
- ✗ RMSE value measured on one sample ($RMSE_{test}$) is much less reliable than the mean of several samples ($RMSE_{cv}$)

Main problem: all existing models were trained on small datasets, so the obtained results are not reliable enough. These models are also publicly unavailable, so the results cannot be reproduced

Aim and objectives

Aim: to develop more reliable publicly available machine learning models for classification and regression tasks, trained on large datasets without using synthetic data generation technologies.

Objectives:

Collect data for machine learning (molecules in SMILES format*)

Clean up the found data

Calculate various parameters of molecules that can be used as features for a machine learning model: combine physico-chemical characteristics with different types of fingerprints** like Avalon, MACCS, CATS2D, Pubchem, ECFP6, as well as fragment-based descriptors generated by our own (Fragments)

Use both artificial neural networks (ANNs) and ensemble machine learning algorithms (CatBoost, XGBoost, Random Forest, LGBM) to predict the target variable and then compare their performance

Try to use a classification label as an additional feature for the regression task

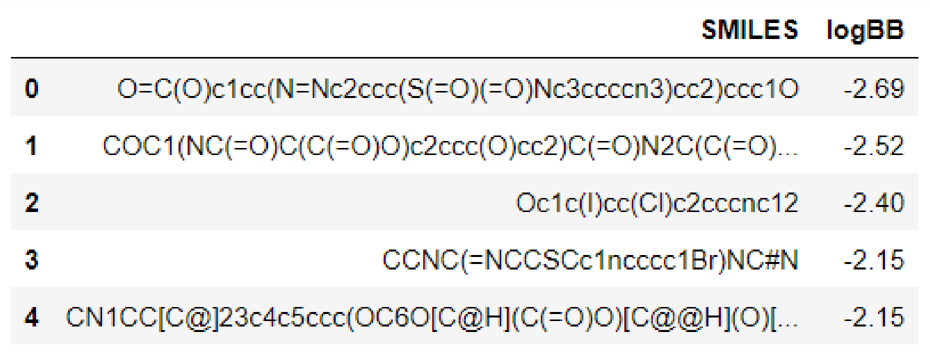
New!

*the most popular and convenient molecules notation

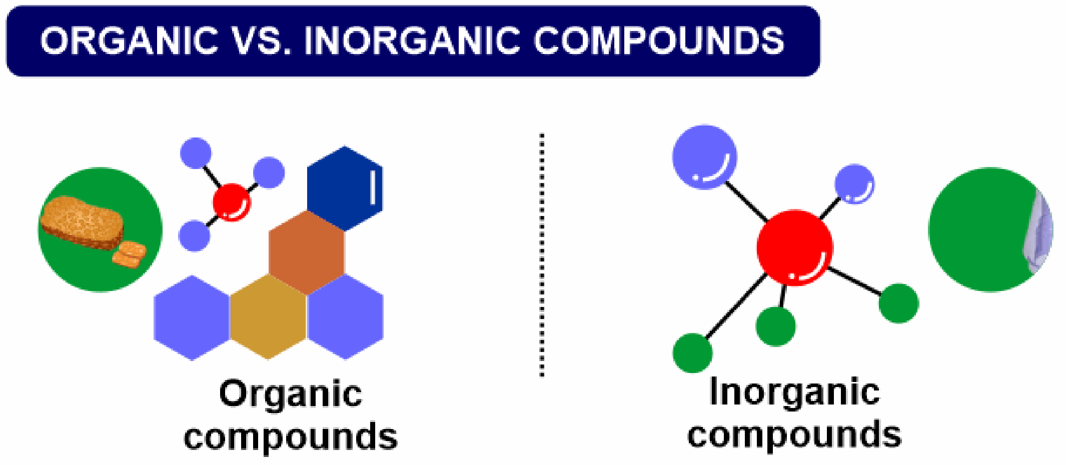
**a bit string that captures information about the molecular structure

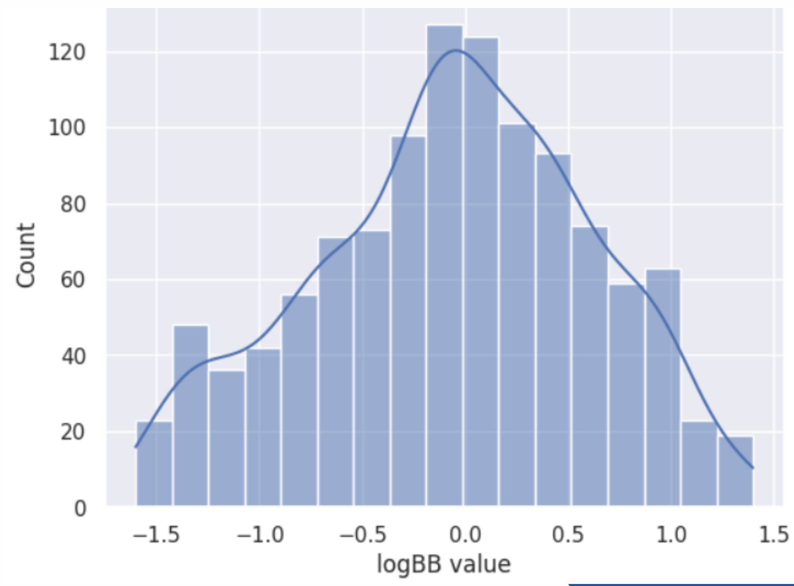
Data preprocessing

- Data sources: **B3DB database**, molecules from the dataset collected by ***Tevosyan et. al***
- Bring all SMILES to the canonical form
- Drop duplicates and NaNs
- Delete all inorganic molecules
- In the case of the regression task, remove those molecules which logBB exceeds two standard deviations from the mean value



	SMILES	logBB
0	O=C(O)c1cc(N=Nc2ccc(S(=O)(=O)Nc3ccccn3)cc2)ccc1O	-2.69
1	COC1(NC(=O)C(C(=O)O)c2ccc(O)cc2)C(=O)N2C(C(=O)...	-2.52
2	Oc1c(I)cc(Cl)c2cccnc12	-2.40
3	CCNC(=NCCSCc1ncccc1Br)NC#N	-2.15
4	CN1CC[C@]23c4c5ccc(OC6O[C@H](C(=O)O)[C@@H](O)[...	-2.15





General working pipeline

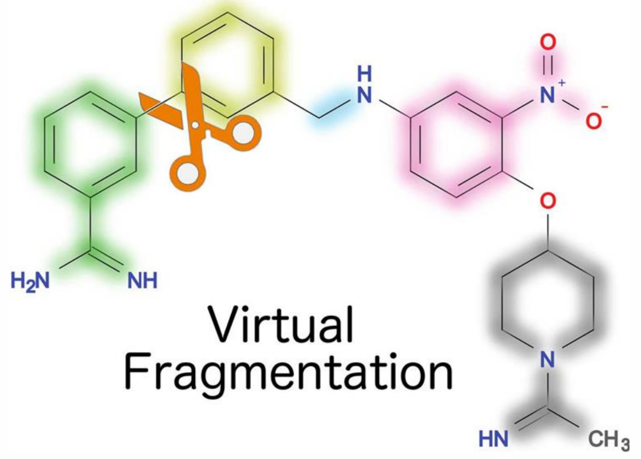
- To evaluate the quality of models, 5-fold cross-validation was performed: the **training sample** was **80%** of the data, the **test sample** was **20%**

- As a basic set of features, we took physico-chemical descriptors from the **RDKit python library**

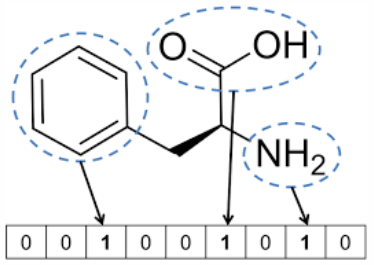
- From each set of features, **75%** of the best were selected using the *SelectKBest* method of the **sklearn python library**

- The feature selection was carried out according to the **mutual information criterion**

- We also provided our **own fragment-based descriptors** using the fragment generator *Chem.FragmentCatalog.FragCatGenerator()* of the python RDKit library. Fragments with a frequency occurrence of less than 1% throughout the dataset were screened out

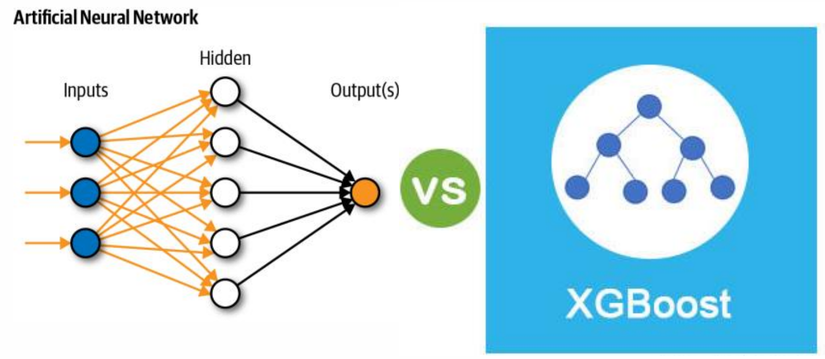


Molecular fingerprints

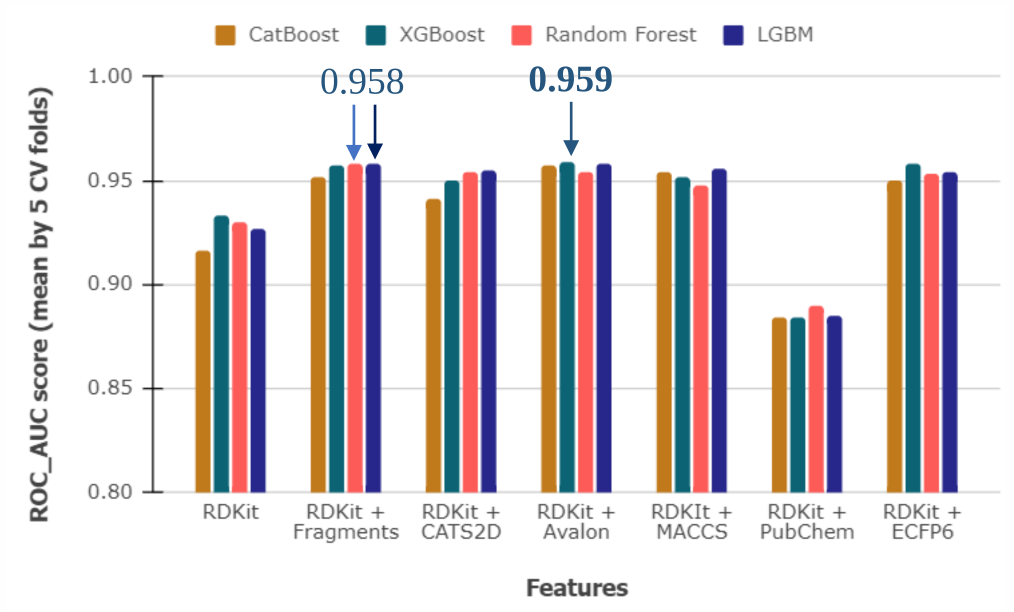


Results: classification task

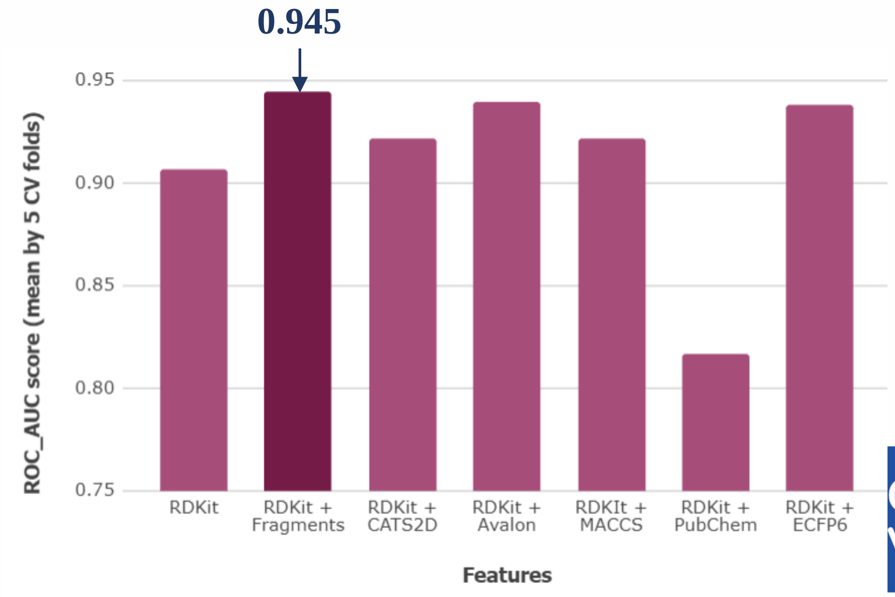
- After data processing, **7795** molecules were left: 4944 BBB permeable (BBB+) and 2851 BBB non-permeable (BBB-)
- The results obtained using various ensemble algorithms do not have a statistically significant difference due to the overlap of the confidence intervals (see *Supplementary materials).* However, the best performance was gained by the **XGBoost** model: $\boldsymbol{ROC\_AUC_{cv}}$**=0.959**
- ANNs turned out to be worse than ensemble models
- PubChem substructure fingerprints don't describe the target variable well enough
- For other types of fingerprints, there is no statistically significant difference. Nevertheless, the best result was achieved on **Avalon**
- **Fragments** performance is quite close to the leading one: ROC_AUC_{cv} = 0.958 vs 0.959



Classical ensemble models

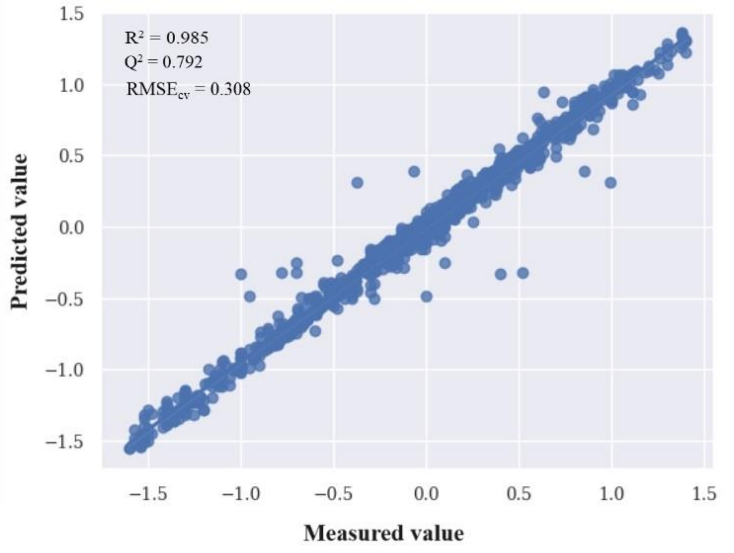


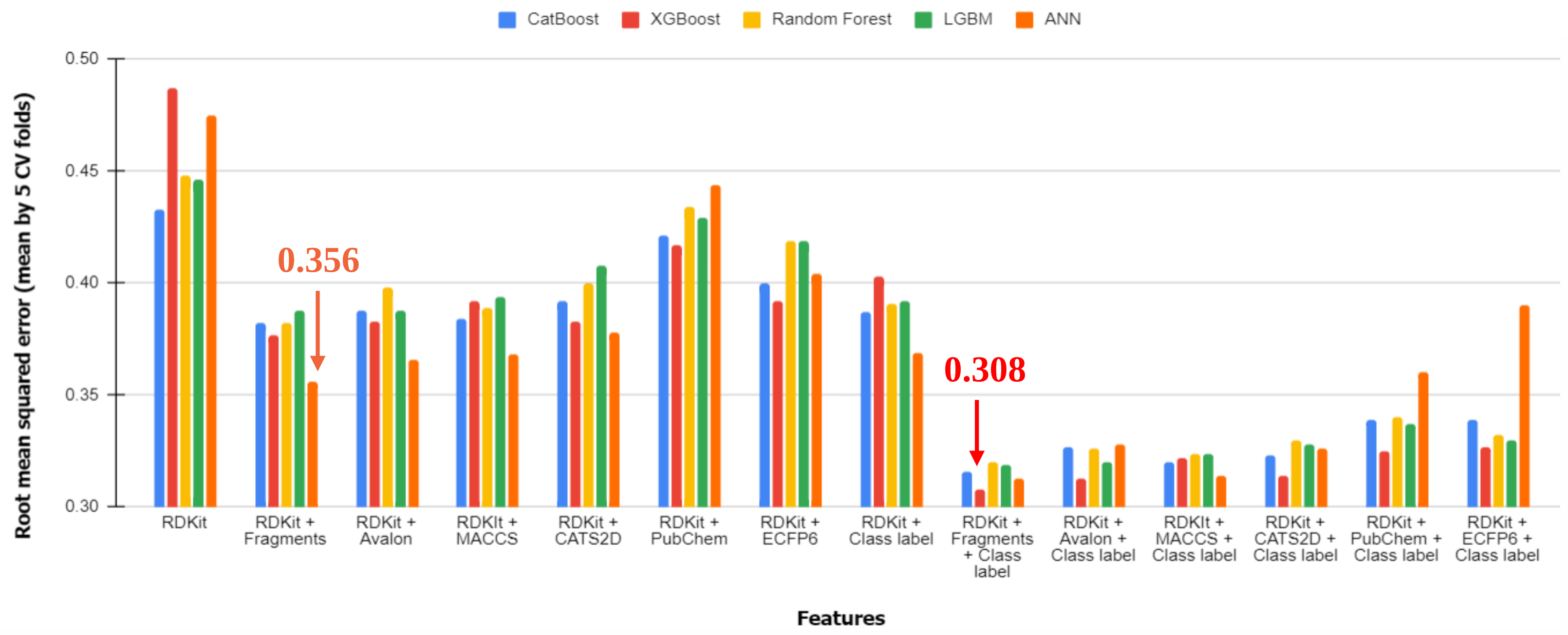
Artificial neural networks



Results: regression task

- After data processing, **1130** molecules were left
- The results obtained using various ensemble algorithms do not have a statistically significant difference . Nonetheless, the best performance was gained by the **ANN** model: **R^2 = 0.903, Q^2 = 0.668, $RMSE_{cv}$ = 0.356**
- However, generally, ANNs did not provide any significant improvements compared with ensemble models
- There is no statistically significant distinction in the use of different types of fingerprints. Nevertheless, the best result was achieved on **Fragments**
- The use of the classification label as a descriptor for the regression task significantly increased the values of the **,** Q^2 and $RMSE_{cv}$ metrics by an average of **10%** and **6%**, respectively. Here, the best score was achieved by **XGBoost** model: **R^2 = 0.985, Q^2 = 0.798, $RMSE_{cv}$ = 0.308**

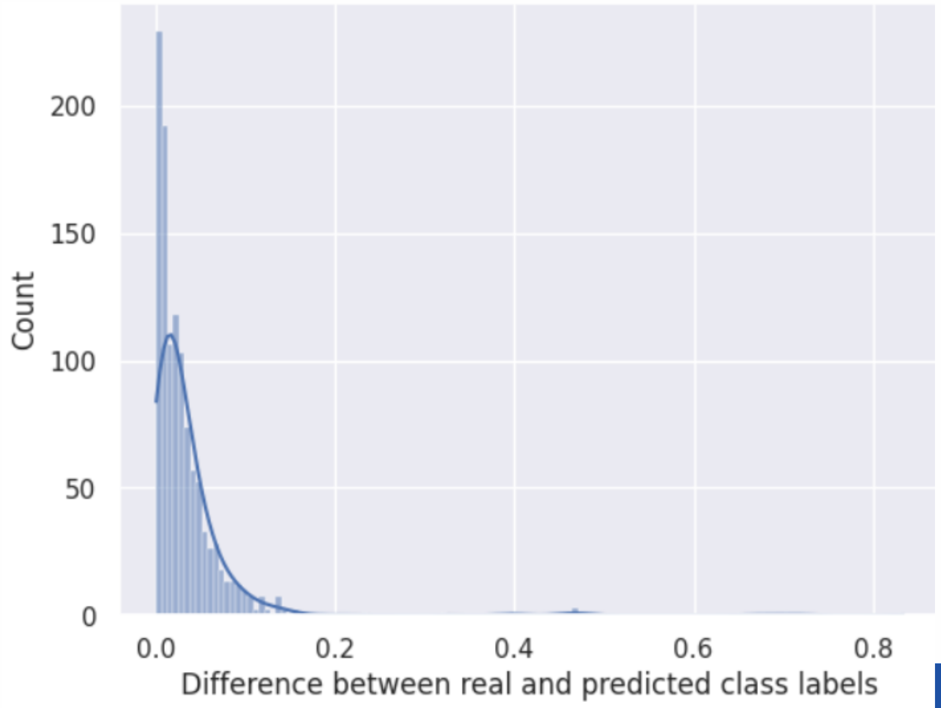




Predict that we do not know

- There is a clear relationship between the class label and the logBB value: **if logBB ≥ -1, the molecule overcomes the BBB, otherwise not**
- It was shown that the idea of using classification label as an additional descriptor for the regression task improved the quality of models significantly. However, in real life we usually do not such labels
- Classification **labels can be obtained from a predictive model** for the classification task. Since we have classification models with fairly high accuracy (ROC_AUC_{cv} about 0.96), this approach **will not lead to a conspicious deterioration** in the regression result

Algorithm	**Real label**			**Predicted label**		
	R^2	**Q^2**	**$RMSE_{cv}$**	**R^2**	**Q^2**	**$RMSE_{cv}$**
XGBoost	0.985	0.792 (± 0.021)	0.308 (± 0.022)	0.984	0.754 (± 0.035)	0.334 (± 0.025)
CatBoost	0.986	0.782 (± 0.021)	0.316 (± 0.02)	0.988	0.742 (± 0.028)	0.342 (± 0.03)
LightGBM	0.987	0.778 (± 0.02)	0.319 (± 0.021)	0.985	0.738 (± 0.036)	0.346 (± 0.029)
RF	0.965	0.777 (± 0.014)	0.32 (± 0.015)	0.959	0.74 (± 0.034)	0.344 (± 0.026)
ANN	0.914	0.748 (± 0.056)	0.313 (±0.035)	0.89	0.71 (± 0.052)	0.336 (± 0.028)



Conclusions

- Publicly available machine learning models for classification and regression tasks were created (see our *GitHub repository*)
- In case of the regression task, the results obtained in our study are more reliable than previously published
- The best scores were reached using XGBoost ML algorithm
- Overall, neural networks did not give substantial gain in comparison with ensemble models
- Fragments proved to be the optimal set of input data for all machine learning algorithms in the case of the regression task and for most algorithms in the case of the classification task
- The use of the classification label as a feature can substantially increase the quality of regression models
- The use of class labels obtained by a predictive model does not lead to a dramatical deterioration in the result compared to the use of real, pre-known labels
- ❗**Next plans are to test our algorithm for even larger dataset**

Supplementary materials

Classification metrics obtained by different algorithms and descriptor sets

Features	CatBoostClassifier				XGBoostClassifier				RandomForestClassifier				LGBMClassifier				ANN			
RDKit	0.888 (± 0.008)	0.916 (± 0.009)	0.94	0.976	0.896 (± 0.006)	0.933 (± 0.009)	0.985	0.999	0.894 (± 0.005)	0.93 (± 0.011)	0.986	0.998	0.892 (± 0.007)	0.927 (± 0.013)	0.984	0.998	0.865 (± 0.007)	0.907 (± 0.013)	0.925	0.954
RDKit + Fragments	0.912 (± 0.005)	0.952 (± 0.004)	0.972	0.994	0.917 (± 0.004)	0.957 (± 0.003)	0.986	0.995	0.917 (± 0.006)	0.958 (± 0.006)	0.985	0.999	0.912 (± 0.004)	0.958 (± 0.004)	0.983	0.998	0.943 (± 0.004)	0.945 (± 0.006)	0.952	0.985
RDKit + CATS2D	0.904 (± 0.005)	0.941 (± 0.008)	0.97	0.994	0.911 (± 0.007)	0.95 (± 0.007)	0.985	0.999	0.912 (± 0.005)	0.954 (± 0.007)	0.985	0.999	0.909 (± 0.004)	0.955 (± 0.001)	0.984	0.999	0.922 (± 0.017)	0.922 (± 0.004)	0.932	0.968
RDKit + Avalon	0.912 (± 0.003)	0.957 (± 0.005)	0.984	0.999	0.914 (± 0.005)	0.959 (± 0.004)	0.977	0.996	0.897 (± 0.004)	0.954 (± 0.005)	0.944	0.987	0.913 (± 0.007)	0.958 (± 0.004)	0.985	0.999	0.94 (± 0.013)	0.94 (± 0.005)	0.94	0.974
RDKIt + MACCS	0.917 (± 0.006)	0.954 (± 0.006)	0.984	0.998	0.913 (± 0.006)	0.952 (± 0.007)	0.985	0.999	0.898 (± 0.004)	0.948 (± 0.006)	0.936	0.982	0.914 (± 0.006)	0.956 (± 0.001)	0.985	0.999	0.922 (± 0.006)	0.922 (± 0.006)	0.927	0.964
RDKit + PubChem	0.861 (± 0.006)	0.884 (± 0.005)	0.997	1.0	0.915 (± 0.011)	0.884 (± 0.004)	0.966	0.993	0.869(± 0.008)	0.89 (± 0.008)	0.953	0.985	0.891 (± 0.034)	0.885 (± 0.011)	0.963	0.992	0.817 (± 0.016)	0.817 (± 0.013)	0.867	0.923
RDKit + ECFP6	0.91 (± 0.003)	0.95 (± 0.006)	0.981	0.997	0.911 (± 0.005)	0.958 (± 0.004)	0.985	0.999	0.911(± 0.004)	0.953 (± 0.006)	0.985	0.999	0.91 (± 0.007)	0.954 (± 0.003)	0.985	0.999	0.938 (± 0.007)	0.938 (± 0.007)	0.95	0.98

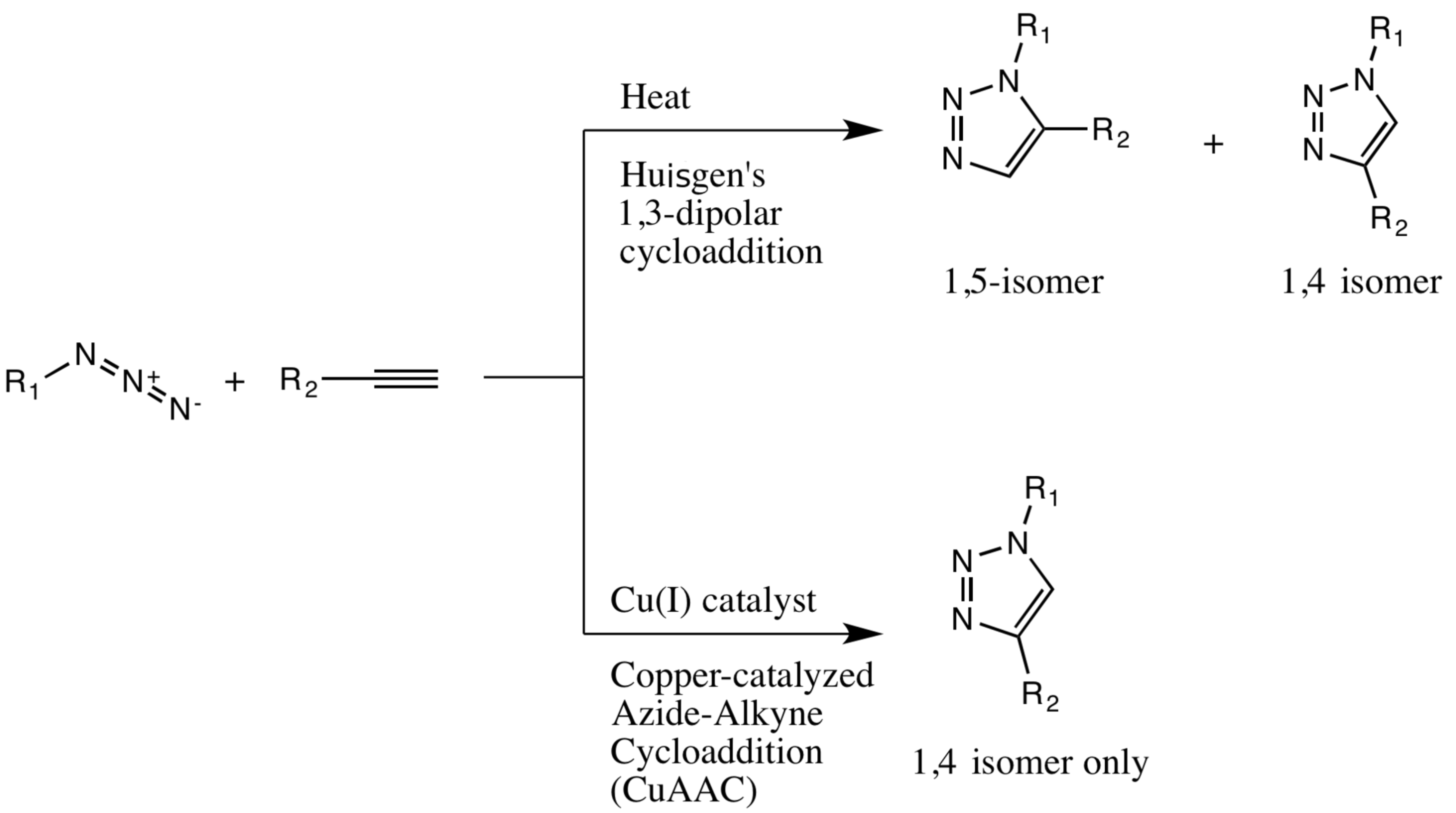
Supplementary materials

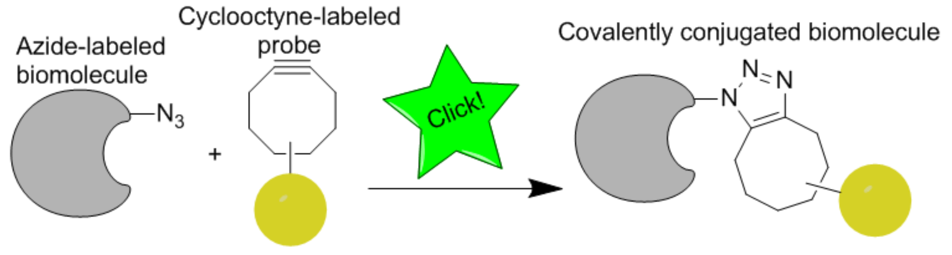
Regression metrics obtained by different algorithms and descriptor sets

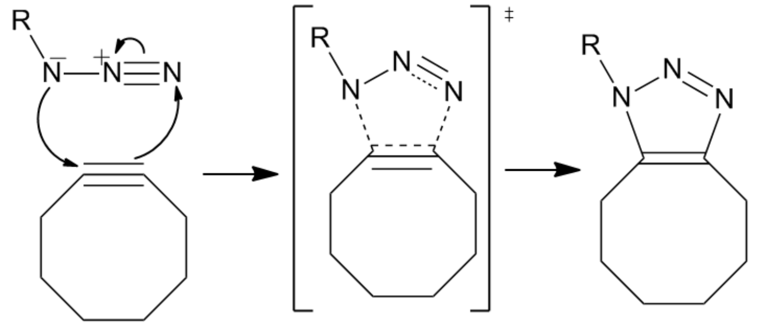
Features	CatBoostRegressor			XGBoostRegressor			RandomForestRegressor			LGBMRegressor			ANN		
	R^2	Q^2	$RMSE_{cv}$	R^2	Q^2	$RMSE_{cv}$	R^2	Q^2	$RMSE_{cv}$	R^2	Q^2	$RMSE_{cv}$	R^2	Q^2	$RMSE_{cv}$
RDKit	0.922	0.59 (± 0.052)	0.433 (± 0.029)	0.812	0.55 (± 0.073)	0.487 (± 0.044)	0.94	0.541 (± 0.031)	0.448 (± 0.039)	0.91	0.572 (± 0.058)	0.446 (± 0.045)	0.81	0.53 (± 0.079)	0.475 (± 0.052)
RDKit + Class label	0.94	0.671 (± 0.07)	0.387 (± 0.033)	0.897	0.651 (± 0.05)	0.403 (± 0.027)	0.95	0.642 (± 0.047)	0.391 (± 0.025)	0.98	0.668 (± 0.049)	0.392 (± 0.032)	0.92	0.639 (± 0.056)	0.369 (± 0.027)
RDKit + Fragments	0.985	0.678 (±0.063)	0.382 (±0.042)	0.987	0.687 (± 0.052)	0.377 (± 0.035)	0.991	0.681 (± 0.043)	0.382 (± 0.027)	0.985	0.67 (± 0.047)	0.388 (± 0.03)	0.903	0.668 (±0.045)	0.356 (±0.02)
RDKit + Fragments + Class label	0.986	0.782 (±0.021)	0.316 (±0.02)	0.985	0.792 (± 0.023)	0.308 (± 0.021)	0.965	0.777 (± 0.014)	0.32 (± 0.015)	0.987	0.778 (± 0.02)	0.319 (± 0.021)	0.914	0.748 (±0.056)	0.313 (±0.035)
RDKit + Avalon	0.976	0.669 (±0.05)	0.388 (±0.035)	0.989	0.679 (± 0.034)	0.383 (± 0.021)	0.991	0.652 (± 0.041)	0.398 (± 0.024)	0.988	0.671 (± 0.033)	0.388 (± 0.025)	0.902	0.656 (±0.048)	0.366 (±0.026)
RDKit + Avalon + Class label	0.981	0.764 (±0.03)	0.327 (±0.023)	0.989	0.785 (± 0.018)	0.313 (± 0.02)	0.963	0.768 (± 0.016)	0.326 (± 0.019)	0.989	0.776 (± 0.031)	0.32 (± 0.027)	0.912	0.734 (±0.043)	0.328 (±0.026)
RDKIt + MACCS	0.987	0.674 (±0.055)	0.384 (±0.036)	0.982	0.664 (± 0.04)	0.392 (± 0.026)	0.991	0.669 (± 0.04)	0.389 (± 0.026)	0.982	0.659 (± 0.042)	0.394 (± 0.03)	0.92	0.632 (±0.063)	0.368 (±0.03)
RDKIt + MACCS + Class label	0.977	0.777 (±0.014)	0.32 (±0.018)	0.99	0.773 (± 0.02)	0.322 (± 0.019)	0.964	0.771 (± 0.019)	0.324 (± 0.02)	0.986	0.771(± 0.017)	0.324 (± 0.018)	0.923	0.742 (±0.053)	0.314 (±0.038)
RDKit + CATS2D	0.978	0.661 (±0.047)	0.392 (±0.031)	0.987	0.679 (± 0.035)	0.383 (± 0.021)	0.991	0.65 (± 0.029)	0.4 (± 0.02)	0.983	0.635 (± 0.039)	0.408 (± 0.024)	0.9	0.628 (±0.033)	0.378 (±0.016)
RDKit + CATS2D + Class label	0.985	0.772 (±0.02)	0.323 (±0.019)	0.982	0.785 (± 0.016)	0.314 (± 0.019)	0.963	0.762 (± 0.019)	0.33 (± 0.019)	0.986	0.764 (± 0.014)	0.328 (± 0.015)	0.931	0.738 (±0.054)	0.326 (±0.036)
RDKit + PubChem	0.969	0.749 (±0.022)	0.421 (±0.02)	0.999	0.608 (± 0.02)	0.417 (± 0.017)	0.946	0.557 (± 0.022)	0.434 (± 0.024)	0.911	0.587 (± 0.026)	0.429 (± 0.022)	0.867	0.468 (±0.049)	0.444 (±0.036)
RDKit + PubChem + Class label	0.996	0.605 (±0.045)	0.339 (±0.023)	0.984	0.726 (± 0.032)	0.325 (± 0.017)	0.964	0.701 (± 0.029)	0.34 (± 0.011)	0.939	0.707 (± 0.029)	0.337 (± 0.013)	0.869	0.618 (±0.057)	0.36 (±0.029)
RDKit + ECFP6	0.964	0.648 (±0.044)	0.4 (±0.03)	0.986	0.662 (± 0.047)	0.392 (± 0.03)	0.945	0.617 (± 0.038)	0.419 (± 0.025)	0.9	0.617 (± 0.03)	0.419 (± 0.026)	0.875	0.566 (±0.04)	0.404 (±0.023)
RDKit + ECFP6 + Class label	0.969	0.749 (±0.022)	0.339 (±0.023)	0.975	0.766 (± 0.017)	0.327 (± 0.02)	0.962	0.759 (± 0.018)	0.332 (± 0.022)	0.979	0.762 (± 0.021)	0.33 (± 0.024)	0.884	0.604 (±0.048)	0.39 (±0.028)

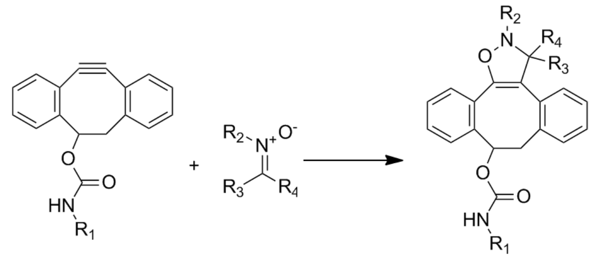
ITsMOre than a UNIVERSITY

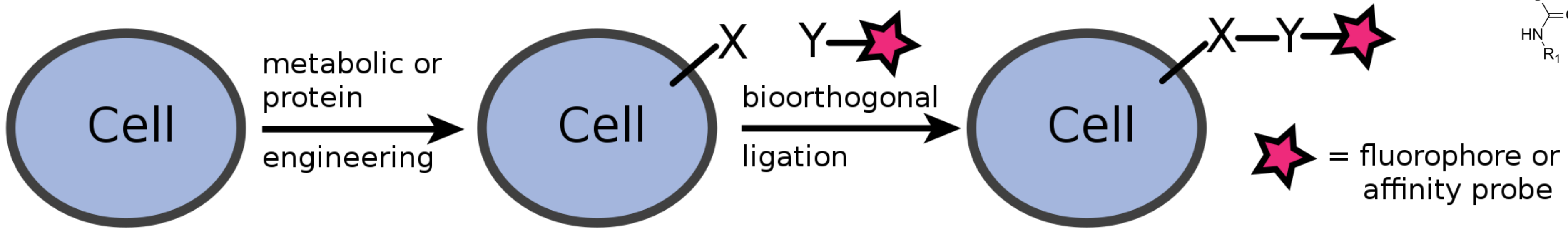
Click reaction and orthogonal chemistry


$R_1-N=N^+=N^-$ + $R_2-C\equiv CH$
Heat
Huisgen's 1,3-dipolar cycloaddition
1,5-isomer
+
1,4 isomer
Cu(I) catalyst
Copper-catalyzed Azide-Alkyne Cycloaddition (CuAAC)
1,4 isomer only

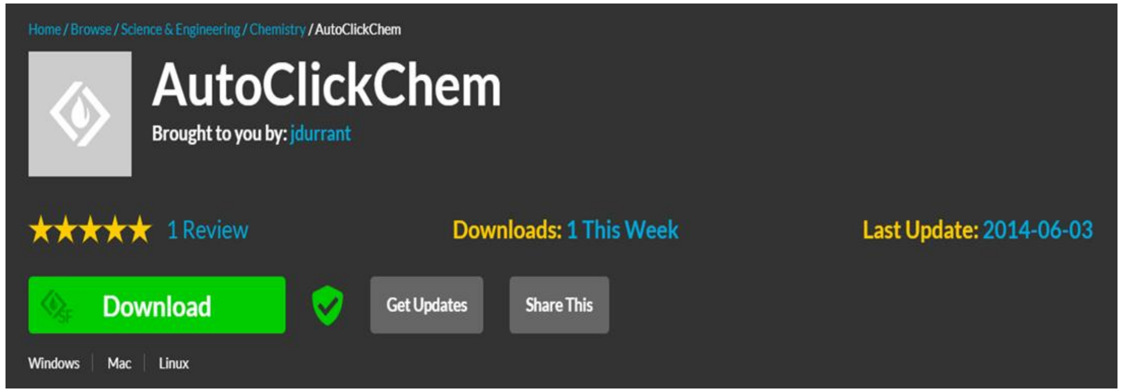

Azide-labeled biomolecule
N_3
+
Cyclooctyne-labeled probe
Click!
Covalently conjugated biomolecule






Cell
metabolic or protein engineering
Cell
X
Y
bioorthogonal ligation
Cell
X—Y
= fluorophore or affinity probe

NuroClick software



Home / Browse / Science & Engineering / Chemistry / AutoClickChem

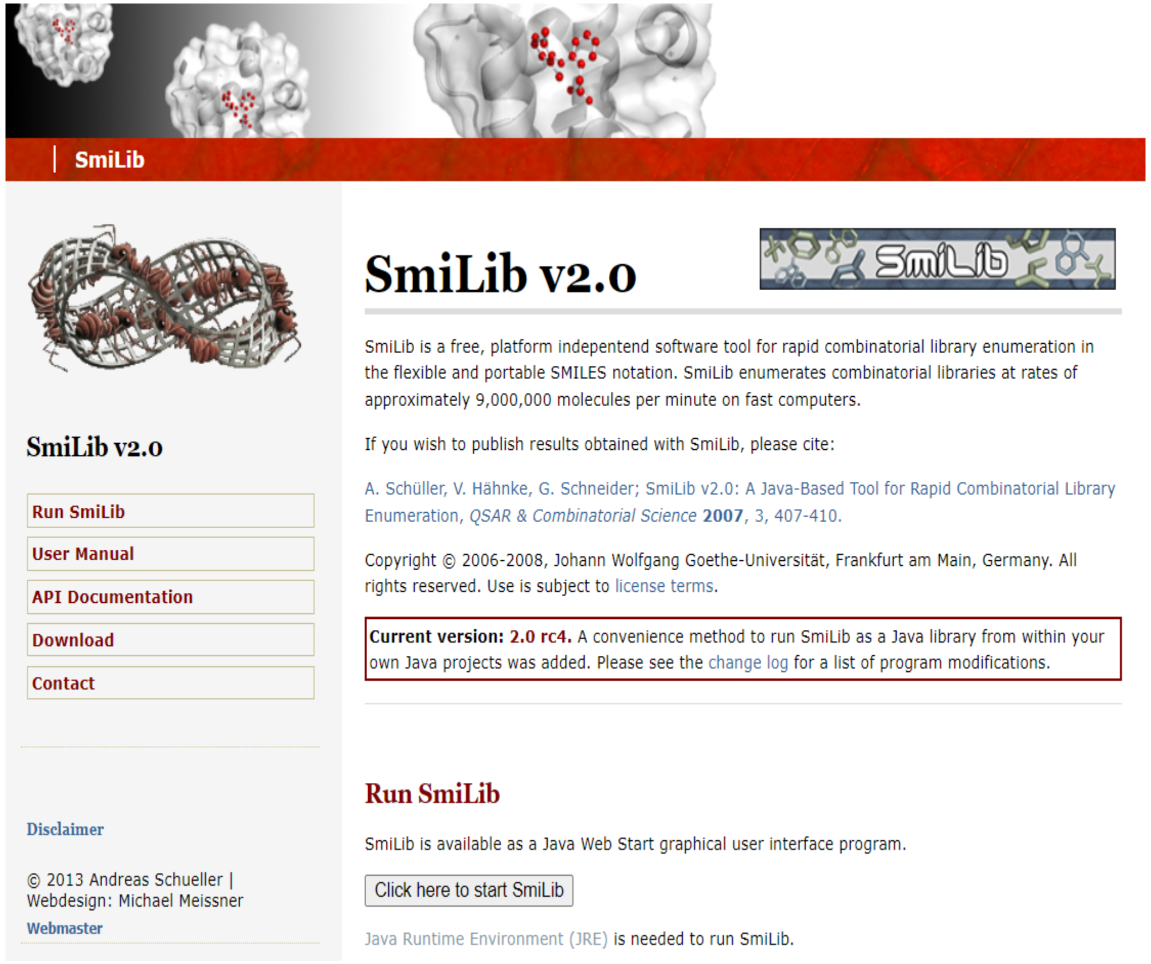
AutoClickChem

Brought to you by: jdurrant

★★★★★ 1 Review

Downloads: 1 This Week

Last Update: 2014-06-03



© 2013 Andreas Schueller | Webdesign: Michael Meissner

Webmaster

SmiLib v2.0

SmiLib is a free, platform independent software tool for rapid combinatorial library enumeration in the flexible and portable SMILES notation. SmiLib enumerates combinatorial libraries at rates of approximately 9,000,000 molecules per minute on fast computers.

If you wish to publish results obtained with SmiLib, please cite:

A. Schüller, V. Hähnke, G. Schneider; SmiLib v2.0: A Java-Based Tool for Rapid Combinatorial Library Enumeration, *QSAR & Combinatorial Science* **2007**, 3, 407-410.

Copyright © 2006-2008, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Germany. All rights reserved. Use is subject to license terms.

Current version: 2.0 rc4. A convenience method to run SmiLib as a Java library from within your own Java projects was added. Please see the change log for a list of program modifications.

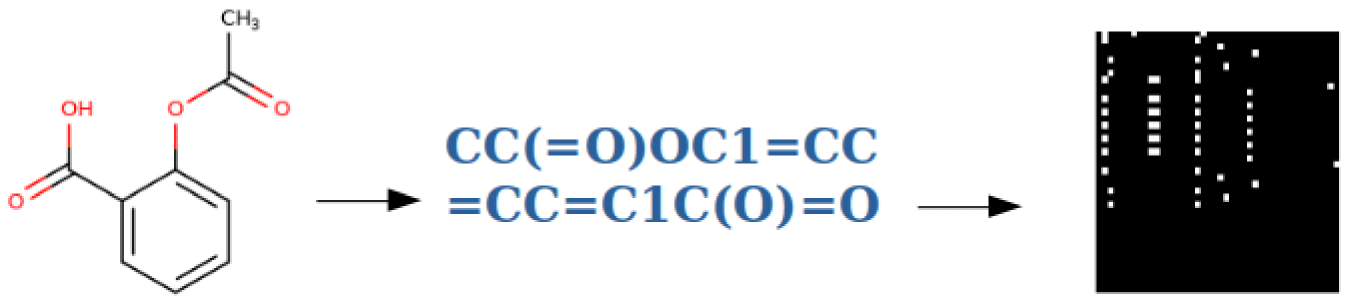
Run SmiLib

SmiLib is available as a Java Web Start graphical user interface program.

Click here to start SmiLib

Java Runtime Environment (JRE) is needed to run SmiLib.

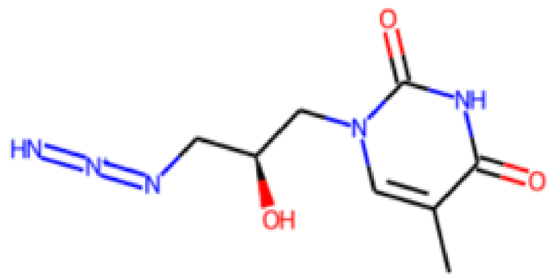
NuroClick software

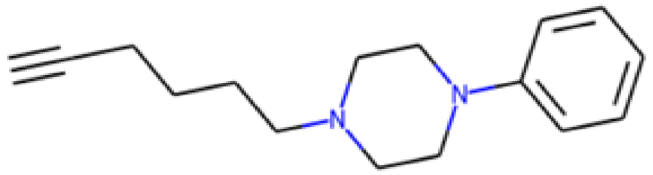


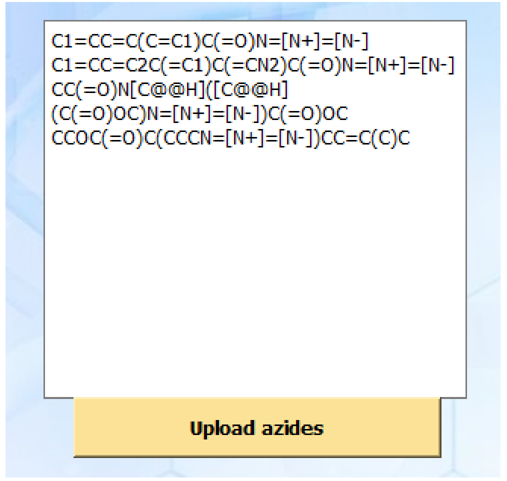
Compound SMILES notation SMILES feature matrix

How it works?

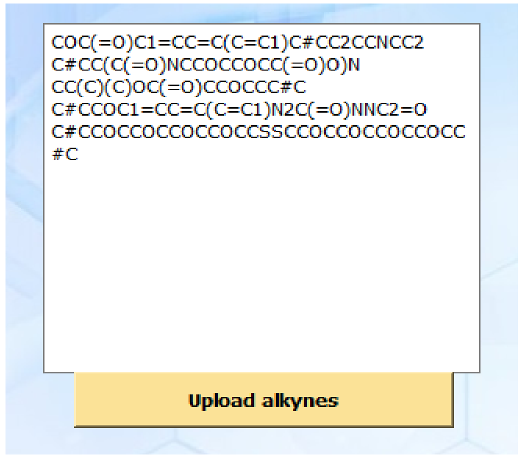
Step 1: reagents upload





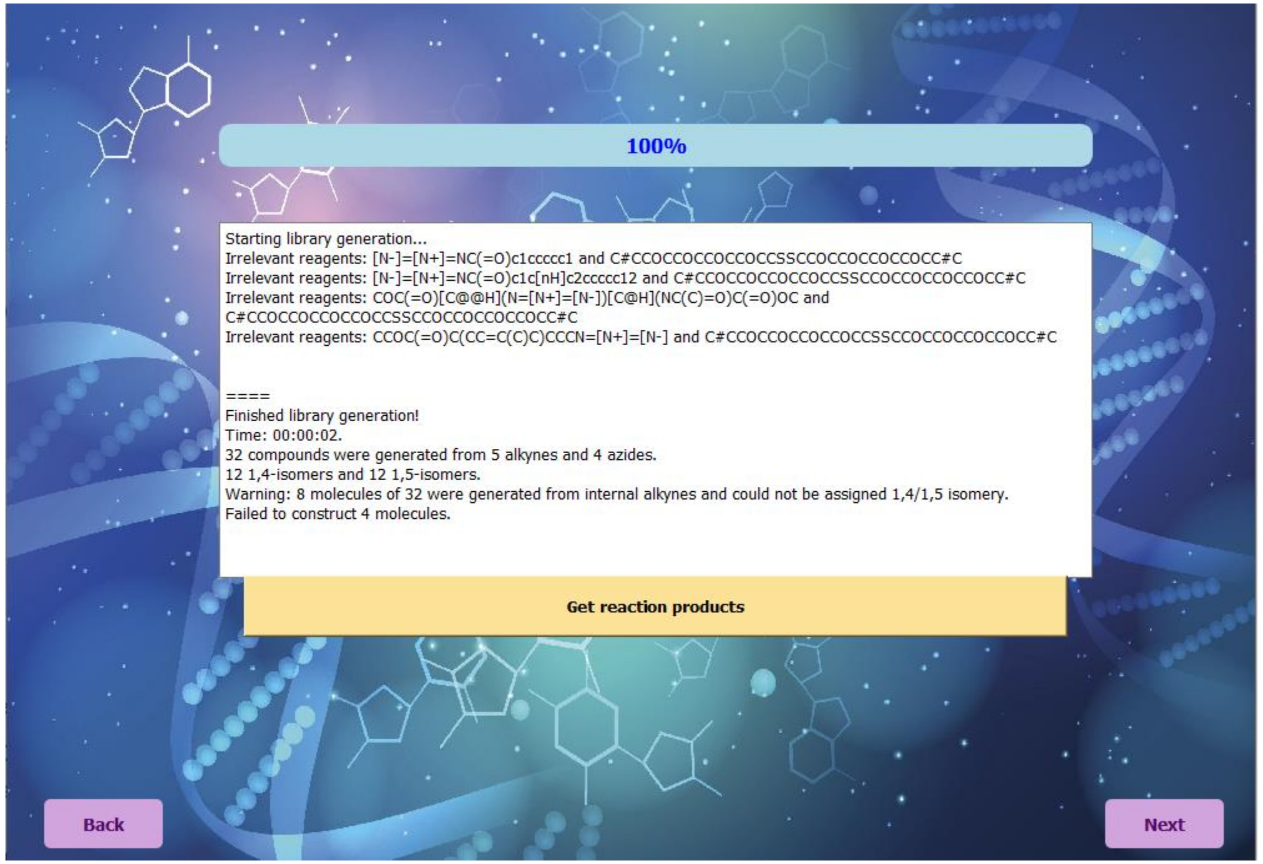


```
C1=CC=C(C=C1)C(=O)N=[N+]=[N-]
C1=CC=C2C(=C1)C(=CN2)C(=O)N=[N+]=[N-]
CC(=O)N[C@@H]([C@@H]
(C(=O)OC)N=[N+]=[N-])C(=O)OC
CCOC(=O)C(CCCN=[N+]=[N-])CC=C(C)C
```



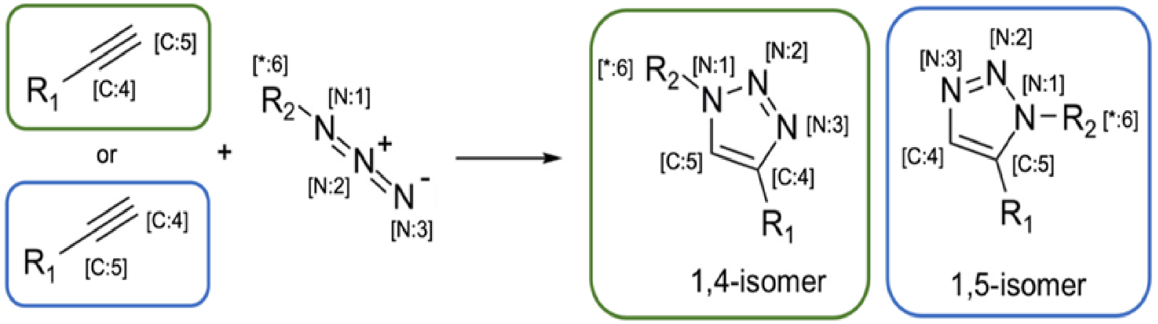
```
COC(=O)C1=CC=C(C=C1)C#CC2CCNCC2
C#CC(C(=O)NCCOCCOCC(=O)O)N
CC(C)(C)OC(=O)CCOCCC#C
C#CCOC1=CC=C(C=C1)N2C(=O)NNC2=O
C#CCOCCOCCOCCOCCSSCCOCCOCCOCCOCC
#C
```

Step 2: click reaction initiation

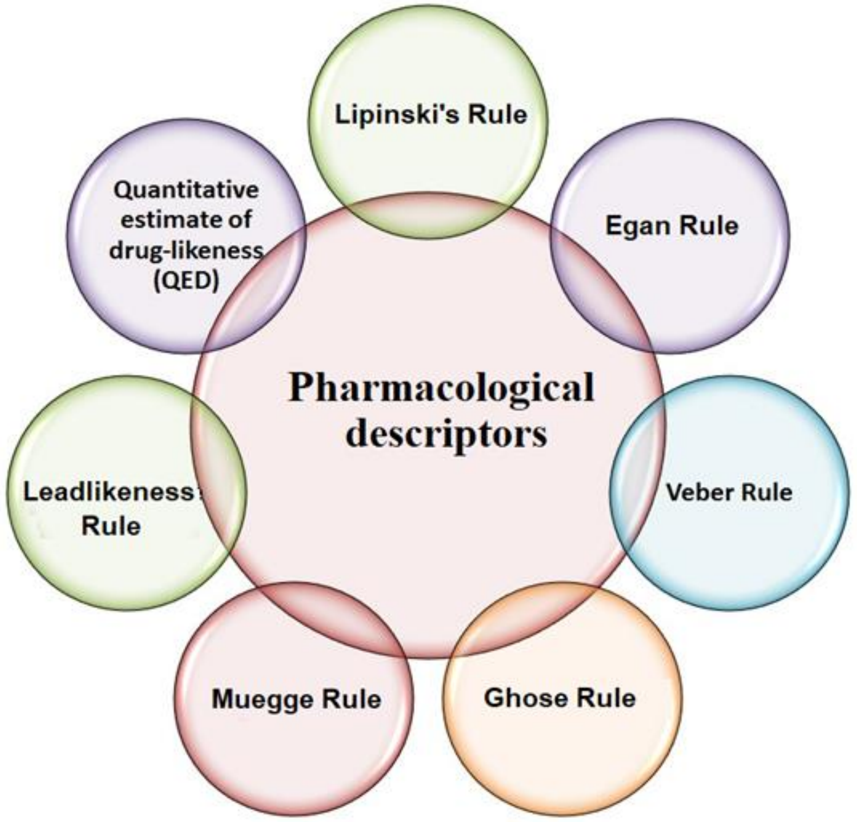


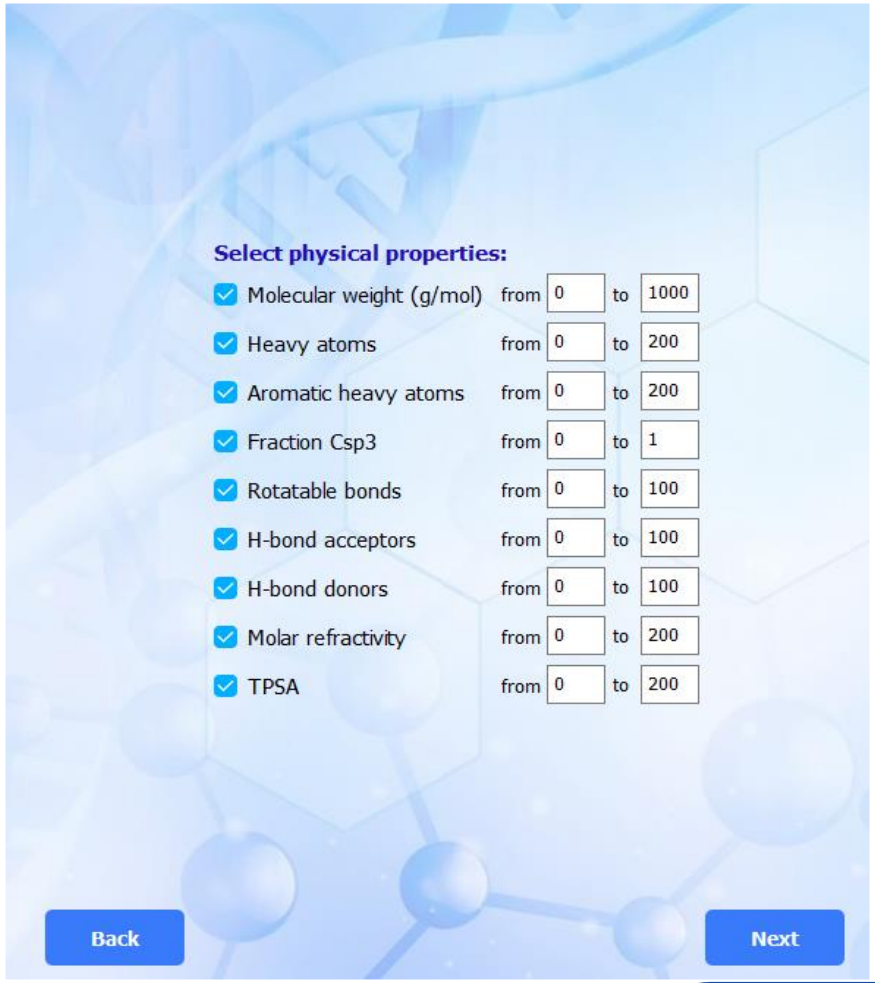
```
Starting library generation...
Irrelevant reagents: [N-]=[N+]=NC(=O)c1ccccc1 and C#CCOCCOCCOCCOCCSSCCOCCOCCOCCOCC#C
Irrelevant reagents: [N-]=[N+]=NC(=O)c1c[nH]c2ccccc12 and C#CCOCCOCCOCCOCCSSCCOCCOCCOCCOCC#C
Irrelevant reagents: COC(=O)[C@@H](N=[N+]=[N-])[C@H](NC(C)=O)C(=O)OC and
C#CCOCCOCCOCCOCCSSCCOCCOCCOCCOCC#C
Irrelevant reagents: CCOC(=O)C(CC=C(C)C)CCCN=[N+]=[N-] and C#CCOCCOCCOCCOCCSSCCOCCOCCOCCOCC#C


====
Finished library generation!
Time: 00:00:02.
32 compounds were generated from 5 alkynes and 4 azides.
12 1,4-isomers and 12 1,5-isomers.
Warning: 8 molecules of 32 were generated from internal alkynes and could not be assigned 1,4/1,5 isomery.
Failed to construct 4 molecules.
```


R1 [C:5] [C:4]
or
R1 [C:4] [C:5]
+
[*:6] R2 [N:1] [N:2] [N:3]
[*:6] R2 [N:1] [N:2] [N:3] [C:5] [C:4] R1
1,4-isomer
[N:3] [N:2] [N:1] R2 [*:6] [C:4] [C:5] R1
1,5-isomer

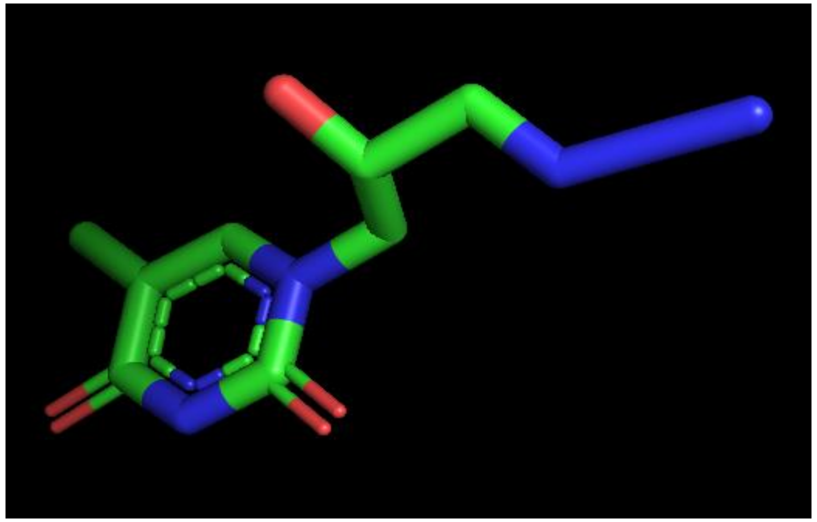
Step 3: PK/PD descriptor calculation

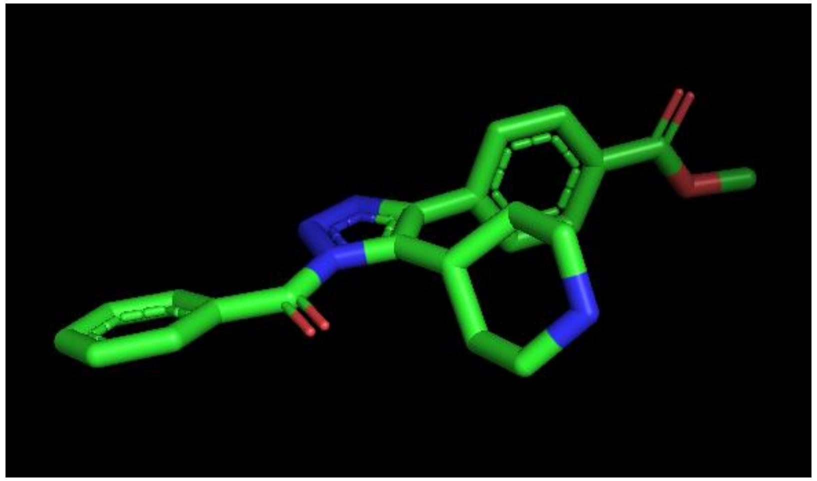


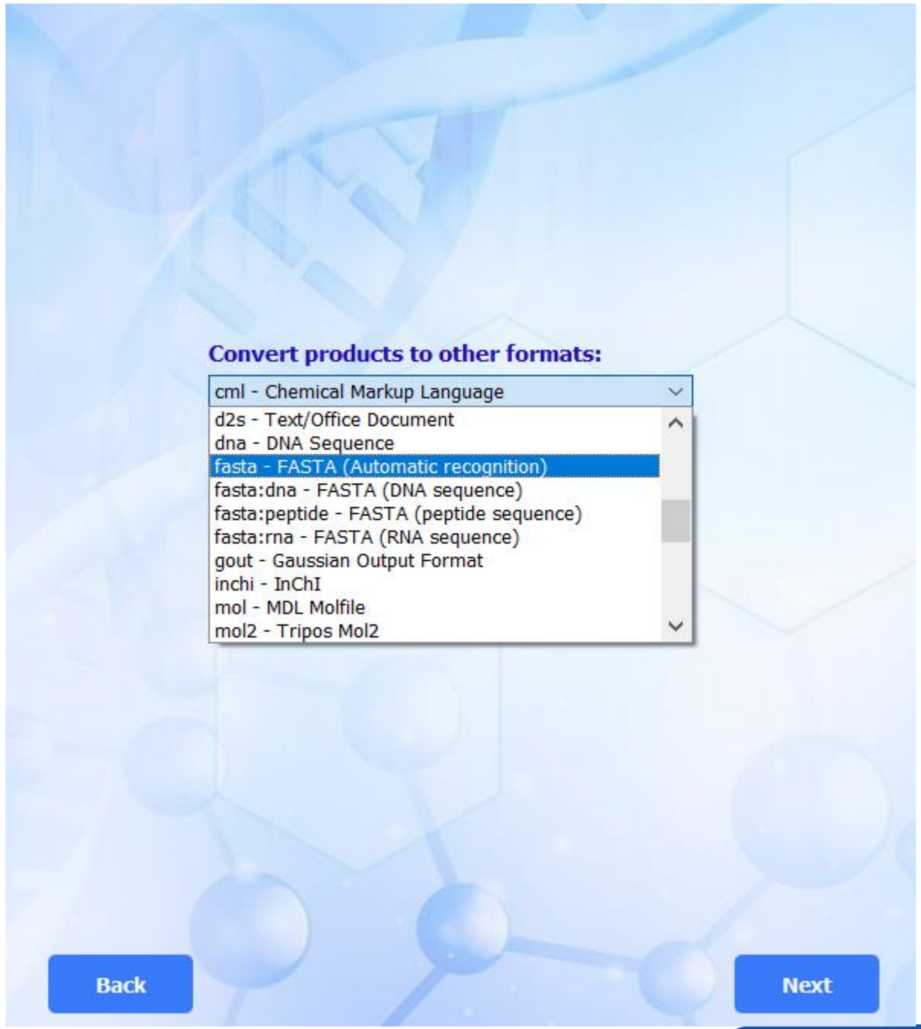

Select physical properties:

✓	Molecular weight (g/mol)	from	0	to	1000
✓	Heavy atoms	from	0	to	200
✓	Aromatic heavy atoms	from	0	to	200
✓	Fraction Csp3	from	0	to	1
✓	Rotatable bonds	from	0	to	100
✓	H-bond acceptors	from	0	to	100
✓	H-bond donors	from	0	to	100
✓	Molar refractivity	from	0	to	200
✓	TPSA	from	0	to	200

Step 4: save the results







Convert products to other formats:

cml - Chemical Markup Language

d2s - Text/Office Document
dna - DNA Sequence
fasta - FASTA (Automatic recognition)
fasta:dna - FASTA (DNA sequence)
fasta:peptide - FASTA (peptide sequence)
fasta:rna - FASTA (RNA sequence)
gout - Gaussian Output Format
inchi - InChI
mol - MDL Molfile
mol2 - Tripos Mol2

ITMO UNIVERSITY

INFOCHEMISTRY SCIENTIFIC CENTER

Thank you for your attention