# Protocolo seguido para generar el modelo de VDss

## Endpoint

Volume of Distribution at steady-state (VDss)

## Origen de los datos

Los datos vienen del Supplementary Information de los artículos:

* “***In Silico* Models of Human PK Parameters. Prediction of Volume of Distribution Using an Extensive Data Set and a Reduced Number of Parameters**” de Berellini (2020). DOI: <https://doi.org/10.1016/j.xphs.2020.08.023>
* **“An Accurate *In Vitro* Prediction of Human VDss Based on the Øie–Tozer Equation and Primary Physicochemical Descriptors. 3. Analysis and Assessment of Predictivity on a Large Dataset”** <https://doi.org/10.1124/dmd.119.088914>
* **“A Benchmarking Dataset with 2440 Organic Molecules for Volume Distribution at Steady State”** (<https://github.com/da-wen-er/VDss/tree/main/VDss_dataset>)

## Tratamiento de los datos

Del Excel original nos quedamos con las columnas “SMILES” y “human VDss (L/kg)”.

Después, y antes de pasar por Hygieia, se usó el script de Ágata (Elbow\_Undersampling.py) para quitar los outliers según peso molecular.

## Transformación de la “y”

Se ha llevado a cabo la transformación log10 (manualmente con numpy, no usando el 1/log que tiene Neo) del valor respuesta para realizar el modelo.

## Train/test ratio

Se ha realizado una partición del 70% para el train y del 30% para el test. Durante el proceso de partición se han mantenido los compuestos señalados como *quite dissimilar*.

## Scaler

Se ha usado el *Standard Scaler*

#########################################################################

######################### WELCOME TO NEO script #########################

#########################################################################

This script will allow you to:

- eliminate 3D descriptors

- "y" transformation

- perform the initial unsupervised feature reduction

- perform the train/test split based on kmeans

- descriptor standarization

- select the relevant features based on:

· Recursive feature elimination (RFE)

· Feature importance (FI) based on Ligth gradient boosting machine (LGBM)

· Permutation importance (PI)

- select your own features features

Please input your PATH (enter to: "../data/Af\_MIC80\_definitva/no3D/OWNdesc/"): C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/

Please input your MODEL NAME (enter to: Af\_MIC80\_no3D): CYP2C19\_Inhibitor

######################### MAIN MENU #########################

Please select what do you want to do:

[01] Elimination of 3D descriptors [your dataset will be saved as [Name]\_no3D]

[1] "y" transformation + dataset random order + Knn imputation

[2] Initial feature reduction: infinite, correlated, constant and empty values

[3] Generation of train and test sets based in kmeans

[4] Descriptor standarization

[5] Feature selection by RFE

[6] Feature selection by FI based on LGBM

[7] Feature selection by Permutation importance

[8] Select own features (inside the script)

[0] Exit NEO

Your choice: 1

This part of the code will do the y" transformation, randomization of the dataset order and Knn imputation.

From this version of NEO is its compulsory to perform the inputation here, as this will create the un-imputed file needed for reimputation.

[+] "y" transformation

A file located in "C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/" folder is needed

This file must be called: "CYP2C19\_Inhibitor-paralel\_calculated\_with\_y.csv"

Continue (Y/n)?Y

Please select your type of model:

[1] Regression

[2] Classification

Your choice: 2

I am so sorry, there is nothing yet for your request. Please try tomorrow with more coffee and cookies.

[+] dataset random sort

The following file has been created (save it as you will need it for feature reduction):

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-calculated\_preimputation.csv

[+] dataset imputation

Size of the database, preimputation: (19641, 4676)

Warning! Some of descriptors have too many NaN values and have been removed: ['LPRS', 'VDA', 'MDDD', 'MAXsLi', 'MAXssBe', 'MAXssssBe', 'MAXssBH', 'MAXsssB', 'MAXssssB', 'MAXsCH3', 'MAXdCH2', 'MAXssCH2', 'MAXtCH', 'MAXdsCH', 'MAXsssCH', 'MAXddC', 'MAXtsC', 'MAXdssC', 'MAXaaaC', 'MAXssssC', 'MAXsNH3', 'MAXsNH2', 'MAXssNH2', 'MAXdNH', 'MAXssNH', 'MAXaaNH', 'MAXtN', 'MAXsssNH', 'MAXdsN', 'MAXaaN', 'MAXsssN', 'MAXddsN', 'MAXaasN', 'MAXssssN', 'MAXsOH', 'MAXdO', 'MAXssO', 'MAXaaO', 'MAXsF', 'MAXsSiH3', 'MAXssSiH2', 'MAXsssSiH', 'MAXssssSi', 'MAXsPH2', 'MAXssPH', 'MAXsssP', 'MAXdsssP', 'MAXsssssP', 'MAXsSH', 'MAXdS', 'MAXssS', 'MAXaaS', 'MAXdssS', 'MAXddssS', 'MAXsCl', 'MAXsGeH3', 'MAXssGeH2', 'MAXsssGeH', 'MAXssssGe', 'MAXsAsH2', 'MAXssAsH', 'MAXsssAs', 'MAXsssdAs', 'MAXsssssAs', 'MAXsSeH', 'MAXdSe', 'MAXssSe', 'MAXaaSe', 'MAXdssSe', 'MAXddssSe', 'MAXsBr', 'MAXsSnH3', 'MAXssSnH2', 'MAXsssSnH', 'MAXssssSn', 'MAXsI', 'MAXsPbH3', 'MAXssPbH2', 'MAXsssPbH', 'MAXssssPb', 'MINsLi', 'MINssBe', 'MINssssBe', 'MINssBH', 'MINsssB', 'MINssssB', 'MINsCH3', 'MINdCH2', 'MINssCH2', 'MINtCH', 'MINdsCH', 'MINsssCH', 'MINddC', 'MINtsC', 'MINdssC', 'MINaaaC', 'MINssssC', 'MINsNH3', 'MINsNH2', 'MINssNH2', 'MINdNH', 'MINssNH', 'MINaaNH', 'MINtN', 'MINsssNH', 'MINdsN', 'MINaaN', 'MINsssN', 'MINddsN', 'MINaasN', 'MINssssN', 'MINsOH', 'MINdO', 'MINssO', 'MINaaO', 'MINsF', 'MINsSiH3', 'MINssSiH2', 'MINsssSiH', 'MINssssSi', 'MINsPH2', 'MINssPH', 'MINsssP', 'MINdsssP', 'MINsssssP', 'MINsSH', 'MINdS', 'MINssS', 'MINaaS', 'MINdssS', 'MINddssS', 'MINsCl', 'MINsGeH3', 'MINssGeH2', 'MINsssGeH', 'MINssssGe', 'MINsAsH2', 'MINssAsH', 'MINsssAs', 'MINsssdAs', 'MINsssssAs', 'MINsSeH', 'MINdSe', 'MINssSe', 'MINaaSe', 'MINdssSe', 'MINddssSe', 'MINsBr', 'MINsSnH3', 'MINssSnH2', 'MINsssSnH', 'MINssssSn', 'MINsI', 'MINsPbH3', 'MINssPbH2', 'MINsssPbH', 'MINssssPb'] The limit is marked to the 15.0 %

[+] fitting

[+] transforming

Size of the database, postimputation: (19641, 4519)

C-001 C-002 C-003 C-004 C-005 C-006 C-007 C-008 ... SsssPbH SssssPb MAXaaCH MAXaasC MINaaCH MINaasC SLogP SMR

0 5.0 0.0 2.0 0.0 0.0 2.0 0.0 5.0 ... 0.0 0.0 2.033120 1.11842 1.979430 1.118420 2.46450 114.1766

1 1.0 0.0 0.0 0.0 3.0 2.0 0.0 1.0 ... 0.0 0.0 2.119313 1.27597 2.032707 1.252347 1.17060 49.6924

2 0.0 2.0 2.0 0.0 3.0 2.0 0.0 0.0 ... 0.0 0.0 1.776000 0.67861 0.974170 -0.984360 6.61590 170.2700

3 3.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 ... 0.0 0.0 2.034120 1.11197 1.792560 0.512860 4.58564 103.2972

4 0.0 0.0 0.0 0.0 2.0 2.0 0.0 0.0 ... 0.0 0.0 1.832860 0.77957 1.745240 0.370680 1.53390 99.5040

... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...

19636 2.0 5.0 1.0 0.0 0.0 6.0 0.0 0.0 ... 0.0 0.0 1.898130 0.83874 1.755980 0.338280 3.13610 124.7410

19637 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 2.357640 1.41060 1.889530 1.410600 5.16510 84.0100

19638 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 1.988550 1.09636 1.703080 0.153710 3.81842 85.9490

19639 0.0 2.0 0.0 0.0 2.0 2.0 0.0 1.0 ... 0.0 0.0 2.104950 1.16907 2.030320 0.768930 3.52670 108.8154

19640 4.0 13.0 4.0 0.0 0.0 2.0 0.0 6.0 ... 0.0 0.0 1.951210 0.97689 1.906750 0.976890 1.99200 217.9762

[19641 rows x 4519 columns]

The following files have been created:

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-calculated\_imputed\_ytransformed.csv

Do you want to perform any other step?(y/n): y

######################### MAIN MENU #########################

Please select what do you want to do:

[01] Elimination of 3D descriptors [your dataset will be saved as [Name]\_no3D]

[1] "y" transformation + dataset random order + Knn imputation

[2] Initial feature reduction: infinite, correlated, constant and empty values

[3] Generation of train and test sets based in kmeans

[4] Descriptor standarization

[5] Feature selection by RFE

[6] Feature selection by FI based on LGBM

[7] Feature selection by Permutation importance

[8] Select own features (inside the script)

[0] Exit NEO

Your choice: 2

A file located in "C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/" folder is needed

This file must be called: "CYP2C19\_Inhibitor-calculated\_imputed\_ytransformed.csv"

Continue (Y/n)?y

[1] Initial feature reduction: infinite, correlated, constant and empty values

0 infinite values

0 features with greater than 0.00 missing values.

2164 features with a correlation magnitude greater than 0.90.

740 features with a single unique value.

Data has not been one-hot encoded

Removed 2904 features including one-hot features.

The following files have been created:

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-initial\_reduction.csv

Do you want to perform any other step?(y/n): y

######################### MAIN MENU #########################

Please select what do you want to do:

[01] Elimination of 3D descriptors [your dataset will be saved as [Name]\_no3D]

[1] "y" transformation + dataset random order + Knn imputation

[2] Initial feature reduction: infinite, correlated, constant and empty values

[3] Generation of train and test sets based in kmeans

[4] Descriptor standarization

[5] Feature selection by RFE

[6] Feature selection by FI based on LGBM

[7] Feature selection by Permutation importance

[8] Select own features (inside the script)

[0] Exit NEO

Your choice: 3

Please select your type of model:

[1] Regression

[2] Classification

Your choice (1/2)?: 2

Please input your desired TEST SIZE (enter to: "0.25"): 0.2

A file located in "C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/" folder is needed

This file must be called: "CYP2C19\_Inhibitor-initial\_reduction.csv"

Continue (Y/n)?y

[+] Generation of train and test sets based in kmeans

OPTIMAL NUMBER OF CLUSTERS: 4

NUMBER OF CLUSTERS: 4

SETS: {19624, 1, 3, 13}

ALERTS!!

compound number: 3482

SMILE: CC(C)(C)c1cc2c(O)c(c1)Cc1cc(C(C)(C)C)cc(c1O)Cc1cc(C(C)(C)C)cc(c1O)Cc1cc(C(C)(C)C)cc(c1O)Cc1cc(C(C)(C)C)cc(c1O)Cc1cc(C(C)(C)C)cc(c1O)Cc1cc(C(C)(C)C)cc(c1O)Cc1cc(C(C)(C)C)cc(c1O)C2 0.0

compound number: 4932

SMILE: O=C1c2ccccc2C(=O)c2c(Nc3cc4c5c(ccc6c7ccc8c9c(cc(Nc%10cccc%11c%10C(=O)c%10ccccc%10C%11=O)c(c3c56)c97)-c3ccccc3C8=O)C(=O)c3ccccc3-4)cccc21 0.0

compound number: 6600

SMILE: C=C(NC(=O)C(=C)NC(=O)c1csc(C2=N[C@@H]3c4csc(n4)[C@@H]4NC(=O)c5csc(n5)[C@@H]([C@](C)(O)[C@H](C)O)NC(=O)C5CSC(=N5)C(=CC)NC(=O)C([C@H](C)O)NC(=O)c5csc(n5)[C@]3(CC2)NC(=O)C(C)NC(=O)C(C)=NC(=O)C(C)NC(=O)C(C(C)CC)N[C@H]2C=Cc3c([C@H](C)O)cc(nc3[C@@H]2O)C(=O)O[C@H]4C)n1)C(N)=O 0.0

compound number: 16536

SMILE: C=C(NC(=O)C(C)=NC(=O)c1csc(C2=NC3c4csc(n4)C4NC(=O)c5csc(n5)C(C(C)(O)C(C)O)NC(=O)C5CSC(=N5)C(=CC)NC(=O)C(C(C)O)NC(=O)c5csc(n5)C3(CC2)NC(=O)C(C)NC(=O)C(C)=NC(=O)C(C)NC(=O)C(C(C)CC)NC2C=Cc3c(C(C)O)cc(nc3C2O)C(=O)OC4C)n1)C(N)=O 0.0

[3482, 4932, 6600, 16536]

you have some molecular alerts. It means that these molecules are quite dissimilar

You can (1) eliminate them or (2) maintain them

What is your choice (1/2)?2

Ok, continue with entire dataframe.

NUMBER OF CLUSTERS: 3

SETS: {19624, 16, 1}

ALERTS!!

compound number: 4932

SMILE: O=C1c2ccccc2C(=O)c2c(Nc3cc4c5c(ccc6c7ccc8c9c(cc(Nc%10cccc%11c%10C(=O)c%10ccccc%10C%11=O)c(c3c56)c97)-c3ccccc3C8=O)C(=O)c3ccccc3-4)cccc21 0.0

[4932]

you have some molecular alerts. It means that these molecules are quite dissimilar

You can (1) eliminate them or (2) maintain them

What is your choice (1/2)?2

Ok, continue with entire dataframe.

NUMBER OF CLUSTERS: 2

SETS: {10, 19631}

ALERTS!!

[]

index SMILES y C-001 C-002 C-003 ... MAXaaCH MAXaasC MINaaCH MINaasC SLogP cluster

0 0 CC(C)C(OCc1ccccc1)C(C)C=NOCC(O)C1OC2OC(C)(C)OC... 0.0 5.0 0.0 2.0 ... 2.033120 1.11842 1.979430 1.118420 2.46450 0

1 1 C[C@@H]1O[C@H](C[N+](C)(C)C)CS1 0.0 1.0 0.0 0.0 ... 2.119313 1.27597 2.032707 1.252347 1.17060 0

2 2 COC(=O)[C@@]1(Cc2ccc(F)cc2)[C@@H]2c3cc(C(=O)N(... 1.0 0.0 2.0 2.0 ... 1.776000 0.67861 0.974170 -0.984360 6.61590 0

3 3 Cc1ccc(C(=O)Nc2c(C#N)c(C)c(C)n2Cc2ccccc2)cc1 1.0 3.0 0.0 0.0 ... 2.034120 1.11197 1.792560 0.512860 4.58564 0

4 4 COC(=O)Cn1c(=O)c2c(nc(Br)n2Cc2ccccc2Cl)n(C)c1=O 1.0 0.0 0.0 0.0 ... 1.832860 0.77957 1.745240 0.370680 1.53390 0

... ... ... ... ... ... ... ... ... ... ... ... ... ...

19636 19636 CCCOc1ccc2oc(=O)c3c(c2c1)CCCN3C(=O)CN1CCCC(C(=... 1.0 2.0 5.0 1.0 ... 1.898130 0.83874 1.755980 0.338280 3.13610 0

19637 19637 CCc1cc2c3ccccc3sc2c2cnccc12 1.0 1.0 1.0 0.0 ... 2.357640 1.41060 1.889530 1.410600 5.16510 0

19638 19638 Cc1cccc(-c2noc(-c3cc4ccccc4oc3=O)n2)c1 1.0 1.0 0.0 0.0 ... 1.988550 1.09636 1.703080 0.153710 3.81842 0

19639 19639 CN(C)c1ccc(-c2nc(NC3CCNCC3)c3ccccc3n2)cc1 0.0 0.0 2.0 0.0 ... 2.104950 1.16907 2.030320 0.768930 3.52670 0

19640 19640 CC(C)C[C@@H](CC(=O)N[C@@H](CCC(=O)O)CC(=O)O)NC... 0.0 4.0 13.0 4.0 ... 1.951210 0.97689 1.906750 0.976890 1.99200 0

[19641 rows x 1619 columns]

0

1

cluster0

index SMILES y C-001 C-002 C-003 ... MAXaaCH MAXaasC MINaaCH MINaasC SLogP cluster

0 0 CC(C)C(OCc1ccccc1)C(C)C=NOCC(O)C1OC2OC(C)(C)OC... 0.0 5.0 0.0 2.0 ... 2.033120 1.11842 1.979430 1.118420 2.46450 0

1 1 C[C@@H]1O[C@H](C[N+](C)(C)C)CS1 0.0 1.0 0.0 0.0 ... 2.119313 1.27597 2.032707 1.252347 1.17060 0

2 2 COC(=O)[C@@]1(Cc2ccc(F)cc2)[C@@H]2c3cc(C(=O)N(... 1.0 0.0 2.0 2.0 ... 1.776000 0.67861 0.974170 -0.984360 6.61590 0

3 3 Cc1ccc(C(=O)Nc2c(C#N)c(C)c(C)n2Cc2ccccc2)cc1 1.0 3.0 0.0 0.0 ... 2.034120 1.11197 1.792560 0.512860 4.58564 0

4 4 COC(=O)Cn1c(=O)c2c(nc(Br)n2Cc2ccccc2Cl)n(C)c1=O 1.0 0.0 0.0 0.0 ... 1.832860 0.77957 1.745240 0.370680 1.53390 0

... ... ... ... ... ... ... ... ... ... ... ... ... ...

19636 19636 CCCOc1ccc2oc(=O)c3c(c2c1)CCCN3C(=O)CN1CCCC(C(=... 1.0 2.0 5.0 1.0 ... 1.898130 0.83874 1.755980 0.338280 3.13610 0

19637 19637 CCc1cc2c3ccccc3sc2c2cnccc12 1.0 1.0 1.0 0.0 ... 2.357640 1.41060 1.889530 1.410600 5.16510 0

19638 19638 Cc1cccc(-c2noc(-c3cc4ccccc4oc3=O)n2)c1 1.0 1.0 0.0 0.0 ... 1.988550 1.09636 1.703080 0.153710 3.81842 0

19639 19639 CN(C)c1ccc(-c2nc(NC3CCNCC3)c3ccccc3n2)cc1 0.0 0.0 2.0 0.0 ... 2.104950 1.16907 2.030320 0.768930 3.52670 0

19640 19640 CC(C)C[C@@H](CC(=O)N[C@@H](CCC(=O)O)CC(=O)O)NC... 0.0 4.0 13.0 4.0 ... 1.951210 0.97689 1.906750 0.976890 1.99200 0

[19631 rows x 1619 columns]

index SMILES y C-001 C-002 C-003 ... MAXaaCH MAXaasC MINaaCH MINaasC SLogP cluster

8263 8263 O=C(NCCCN1CCc2ccccc2C1)C1CCN(S(=O)(=O)N2CCCC2)CC1 0.0 0.0 6.0 1.0 ... 2.216840 1.45846 2.101420 1.419840 1.60360 0

8214 8214 O=C(O)C1C2CCC(O2)C1C(=O)O 0.0 0.0 2.0 2.0 ... 1.851030 0.69907 1.687383 0.659733 -0.05080 0

1087 1087 CS(=O)(=O)N1CC(C(=O)NC2CCCC2)Oc2ccc(Cl)cc21 1.0 0.0 4.0 0.0 ... 1.609230 0.41190 1.530790 0.345040 1.92580 0

7409 7409 COc1ccc(N2CCC(CNC(=O)Nc3ccc(OC)cc3OC)C2)cc1 1.0 0.0 1.0 1.0 ... 2.066770 1.17659 1.735430 0.556530 3.36040 0

13375 13375 COC(=O)C1=C(C)[C@@H](O)[C@@H](C)[C@H](c2c(OC)c... 1.0 3.0 0.0 1.0 ... 1.885963 0.68646 1.640320 0.380060 2.54472 0

[5 rows x 1619 columns]

cluster1

index SMILES y C-001 C-002 C-003 ... MAXaaCH MAXaasC MINaaCH MINaasC SLogP cluster

2729 2729 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 1.0 ... 1.21774 -0.16226 0.76920 -0.88220 0.1062 1

3482 3482 CC(C)(C)c1cc2c(O)c(c1)Cc1cc(C(C)(C)C)cc(c1O)Cc... 0.0 24.0 8.0 0.0 ... 2.03815 0.97946 2.03815 0.07130 20.7512 1

4932 4932 O=C1c2ccccc2C(=O)c2c(Nc3cc4c5c(ccc6c7ccc8c9c(c... 0.0 0.0 0.0 0.0 ... 2.02071 0.80409 1.67448 0.24075 13.1980 1

5209 5209 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 1.0 ... 1.21774 -0.16226 0.76920 -0.88220 0.1062 1

6600 6600 C=C(NC(=O)C(=C)NC(=O)c1csc(C2=N[C@@H]3c4csc(n4... 0.0 11.0 3.0 1.0 ... 1.46977 0.22176 1.24879 -0.45472 1.1315 1

7700 7700 Oc1ccc2c3c1O[C@H]1c4[nH]c5c(c4C[C@@]4(O)[C@@H]... 0.0 0.0 10.0 2.0 ... 2.06027 1.18666 1.77258 0.14163 3.8241 1

7823 7823 Oc1ccc2c3c1OC1c4[nH]c5c(c4CC4(O)C(C2)N(CC2CC2)... 0.0 0.0 10.0 2.0 ... 2.06027 1.18666 1.77258 0.14163 3.8241 1

8205 8205 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 1.0 ... 1.21774 -0.16226 0.76920 -0.88220 0.1062 1

14411 14411 Oc1ccc2c3c1O[C@@H]1c4[nH]c5c(c4C[C@@]4(O)[C@@H... 0.0 0.0 10.0 2.0 ... 2.06027 1.18666 1.77258 0.14163 3.8241 1

16536 16536 C=C(NC(=O)C(C)=NC(=O)c1csc(C2=NC3c4csc(n4)C4NC... 0.0 12.0 3.0 1.0 ... 1.47172 0.22265 1.24983 -0.45358 1.4891 1

[10 rows x 1619 columns]

index SMILES y C-001 C-002 C-003 ... MAXaaCH MAXaasC MINaaCH MINaasC SLogP cluster

8205 8205 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 1.0 ... 1.21774 -0.16226 0.76920 -0.88220 0.1062 1

3482 3482 CC(C)(C)c1cc2c(O)c(c1)Cc1cc(C(C)(C)C)cc(c1O)Cc... 0.0 24.0 8.0 0.0 ... 2.03815 0.97946 2.03815 0.07130 20.7512 1

14411 14411 Oc1ccc2c3c1O[C@@H]1c4[nH]c5c(c4C[C@@]4(O)[C@@H... 0.0 0.0 10.0 2.0 ... 2.06027 1.18666 1.77258 0.14163 3.8241 1

2729 2729 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 1.0 ... 1.21774 -0.16226 0.76920 -0.88220 0.1062 1

4932 4932 O=C1c2ccccc2C(=O)c2c(Nc3cc4c5c(ccc6c7ccc8c9c(c... 0.0 0.0 0.0 0.0 ... 2.02071 0.80409 1.67448 0.24075 13.1980 1

[5 rows x 1619 columns]

index SMILES y C-001 C-002 C-003 ... MAXaaCH MAXaasC MINaaCH MINaasC SLogP cluster

8263 8263 O=C(NCCCN1CCc2ccccc2C1)C1CCN(S(=O)(=O)N2CCCC2)CC1 0.0 0.0 6.0 1.0 ... 2.216840 1.45846 2.101420 1.419840 1.60360 0

8214 8214 O=C(O)C1C2CCC(O2)C1C(=O)O 0.0 0.0 2.0 2.0 ... 1.851030 0.69907 1.687383 0.659733 -0.05080 0

1087 1087 CS(=O)(=O)N1CC(C(=O)NC2CCCC2)Oc2ccc(Cl)cc21 1.0 0.0 4.0 0.0 ... 1.609230 0.41190 1.530790 0.345040 1.92580 0

7409 7409 COc1ccc(N2CCC(CNC(=O)Nc3ccc(OC)cc3OC)C2)cc1 1.0 0.0 1.0 1.0 ... 2.066770 1.17659 1.735430 0.556530 3.36040 0

13375 13375 COC(=O)C1=C(C)[C@@H](O)[C@@H](C)[C@H](c2c(OC)c... 1.0 3.0 0.0 1.0 ... 1.885963 0.68646 1.640320 0.380060 2.54472 0

[5 rows x 1619 columns]

Train set contains:

8697 negative values

7015 positive values

ratio neg / pos: 1.2397719173200286

Test set contains:

2175 negative values

1754 positive values

ratio neg / pos: 1.2400228050171038

If you find this imbalanced, try to decomment line 44 of split\_by\_kmeans.py module. It can give an error!

The following files have been created:

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-cleaned\_from\_kmeans.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-train\_set.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-test\_set.csv

Do you want to perform any other step?(y/n): y

######################### MAIN MENU #########################

Please select what do you want to do:

[01] Elimination of 3D descriptors [your dataset will be saved as [Name]\_no3D]

[1] "y" transformation + dataset random order + Knn imputation

[2] Initial feature reduction: infinite, correlated, constant and empty values

[3] Generation of train and test sets based in kmeans

[4] Descriptor standarization

[5] Feature selection by RFE

[6] Feature selection by FI based on LGBM

[7] Feature selection by Permutation importance

[8] Select own features (inside the script)

[0] Exit NEO

Your choice: 4

[+] Descriptor standarization

Please select the method to standarize the descriptors:

[1] StandardScaler

[2] MinMaxScaler

Your choice (1/2)?: 1

Two files located in "C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/" folder are needed

These files must be called:

"CYP2C19\_Inhibitor-train\_set.csv"

"CYP2C19\_Inhibitor-test\_set.csv"

Continue (Y/n)?y

The following files have been created:

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-stand\_train\_set.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-stand\_test\_set.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/CYP2C19 Inhibitor/CYP2C19\_Inhibitor-alldataset.sca

Do you want to perform any other step?(y/n): y

######################### MAIN MENU #########################

Please select what do you want to do:

[01] Elimination of 3D descriptors [your dataset will be saved as [Name]\_no3D]

[1] "y" transformation + dataset random order + Knn imputation

[2] Initial feature reduction: infinite, correlated, constant and empty values

[3] Generation of train and test sets based in kmeans

[4] Descriptor standarization

[5] Feature selection by RFE

[6] Feature selection by FI based on LGBM

[7] Feature selection by Permutation importance

[8] Select own features (inside the script)

[0] Exit NEO

Your choice: 0

Thanks for using NEO!