# Protocolo seguido para generar el modelo de CYP1A2

## Endpoint

Sustratos o no sustratos para CYP1A2

## Origen de los datos

Los datos vienen de diversas bases de datos:

* **CYP1A2\_CYProduct**
* **CYP1A2\_CYPstrate**
* **CYP1A2\_Deep-PK**
* **CYP1A2\_Metabase**
* **CYP1A2\_Terri**

## Tratamiento de los datos

* **CYP1A2\_CYProduct:**

Los datos para crear los modelos de sustratos de las enzimas CYP se han obtenido a partir del siguiente artículo: https://doi.org/10.1021/acs.jcim.1c00144

Se ha descargado la base de datos de EBoMD.sdf y EBoMD2.sdf, compuesta por dos archivos en formato sdf y que contenía las columnas: Molecule Name, Name, InChiKey, PubChem, CYP (1A2, 2A6, 2B6, 2C8, 2C9, 2C19, 2D6, 2E1 y 3A4), References y Comments. Se utilizó el script "sdf\_to\_csv.py" para añadir la columna con los SMILES y transformar el archivo sdf a csv. A continuación se modificaron las columnas de las CYP, mediant el script "Zaretzki\_dataset\_modf", para que apareciese 1 o 0, en función de si aparecía alguna información o no en la celda de cada CYP. Finalmente, mediante un script (CYPstrate\_DB\_processing), he creado las bases de datos de sustratos/no-sustratos para cada enzima y he añadido los compuestos de la base de datos de Hunt a cada una (ya que los sustratos para cada enzima están separados en hojas de Excel).

* **CYP1A2\_CYPstrate**

Los datos para crear los modelos de sustratos de las enzimas CYP se han obtenido a partir del siguiente artículo: 10.3390/molecules26154678

Para el desarrollo del modelo, se compiló un "conjunto de datos principal" de sustratos y no sustratos de nueve isoenzimas humanas del CYP (1A2, 2A6, 2B6, 2C8, 2C9, 2C19, 2D6, 2E1 y 3A4) a partir de los trabajos de Hunt et al. y Tian et al. El conjunto de datos de Hunt es una colección bien curada de un total de 484 sustratos conocidos de siete de los nueve CYP investigados (excluyendo CYPs 2A6 y 2B6). El conjunto de datos de Tian contiene un total de 1800 sustratos y no sustratos conocidos para los mismos nueve CYPs. Los sustratos incluidos en este conjunto de datos provienen, en gran parte, del conjunto de datos de sitios de metabolismo de Zaretzki et al. Dado que el conjunto de datos de Zaretzki ha sido revisado desde entonces, la parte de los datos que proviene del conjunto original de Zaretzki fue reemplazada por los datos del conjunto de datos revisado de Zaretzki para los fines de este estudio.

En primer lugar se ha descargado la base de datos de Tian, compuesta por dos archivos (Train y Test) en formato sdf y que contenía las columnas: Molecule Name, Name, InChiKey, CYP (1A2, 2A6, 2B6, 2C8, 2C9, 2C19, 2D6, 2E1 y 3A4), References y Comments. Se utilizó el script "sdf\_to\_csv.py" para añadir la columna con los SMILES y transformar el archivo sdf a csv. A continuación se eliminaron todos aquellos compuestos marcados con una R, que hacían referencia a los sustratos.

En segundo lugar se descargó la base de datos de sustratos actualizada de Zaretzki. Se utilizó el script "sdf\_to\_csv.py" para añadir la columna con los SMILES y transformar el archivo sdf a csv. A continuación se modificaron las columnas de las CYP, mediant el script "Zaretzki\_dataset\_modf", para que apareciese 1 o 0, en función de si aparecía algún numero o no en la celda de cada CYP

En tercer lugar se creó una hoja de excel llamada CYP\_generalDB.csv. En este archivo se juntaron manualmente la información de las tres bases de datos.

Finalmente, mediante un script (CYPstrate\_DB\_processing), he creado las bases de datos de sustratos/no-sustratos para cada enzima y he añadido los compuestos de la base de datos de Hunt a cada una (ya que los sustratos para cada enzima están separados en hojas de Excel).

* **CYP1A2\_Deep-PK**

Se obtienen un total de tres archivos csv, separados en test, train y validation.

* **CYP1A2\_Metabase**

Se descargaron los datos disponibles en formato .txt para las CYP (1A2, 2C9, 2C19, 2D6, 3A4, 2C8, 2E1) del repositorio de Metrabase, en el siguiente link: http://www-metrabase.ch.cam.ac.uk.

El archivo de texto fue procesado para generar un archivo csv que guardase las columnas cmpd\_id cmpd\_name, y action\_type. Mediante un script en python se transformo la información de action\_type (inhibitor, substrate, non-substrate) en números (-1, 0, 1). Además se obtuvieron los smiles también por el script. Aquellos smiles que no fueron obtenidos por el script se comprobaron manualmente.

* **CYP1A2\_Terri**

Los datos para crear los modelos de sustratos de las enzimas CYP se han obtenido a partir del siguiente artículo: https://s3-us-west-2.amazonaws.com/drugbank/cite\_this/attachments/files/000/001/582/original/cyp450\_drug\_interactions.pdf?1537393944

El nombre de los compuestos y sus smiles aparecen en el propio artículo, se creó un excel con el nombre del compuesto, el smiles y se etiquetaron todos como 1 al ser sustratos.

A continuación, se eliminaron los Outliers de todas las bases de datos, mediante el script “Elbow\_Undersampling.py”. Las bases de datos finales se concatenaron mediante un script de Python y se procesaron mediante HYGIEIA.

## Transformación de la “y”

No hay transformación

## Train/test ratio

Se ha realizado una partición del 75% para el train, 10% para el test y 15% para el validation. Durante el proceso de partición se mantienen todos 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/TK\_CYP1A2inh/

Please input your MODEL NAME (enter to: Af\_MIC80\_no3D): TK\_CYP1A2inh\_DeepPK\_final

######################### 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/TK\_CYP1A2inh/" folder is needed

This file must be called: "TK\_CYP1A2inh\_DeepPK\_final-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/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-calculated\_preimputation.csv

[+] dataset imputation

Size of the database, preimputation: (19882, 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: (19882, 4519)

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

0 1.0 4.0 0.0 0.0 0.0 3.0 0.0 ... 0.0 1.68137 0.52011 1.451280 0.18233 2.67560 109.4560

1 0.0 4.0 0.0 1.0 0.0 4.0 0.0 ... 0.0 1.84171 0.79763 1.678620 0.39537 2.54790 100.3468

2 0.0 0.0 0.0 0.0 2.0 5.0 0.0 ... 0.0 1.90482 0.86372 1.699380 0.55517 2.50359 118.8892

3 0.0 0.0 0.0 0.0 0.0 1.0 0.0 ... 0.0 1.85721 0.65751 1.368910 0.65751 1.36230 89.0152

4 0.0 0.0 0.0 0.0 2.0 1.0 0.0 ... 0.0 2.13745 1.24081 1.557130 0.78926 3.34380 92.3787

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

19877 0.0 4.0 1.0 0.0 2.0 2.0 0.0 ... 0.0 1.91333 0.84422 1.896570 0.84422 2.18010 82.1138

19878 1.0 7.0 2.0 0.0 0.0 2.0 0.0 ... 0.0 2.14157 1.26223 2.042357 1.26223 2.43070 70.5090

19879 1.0 1.0 0.0 0.0 0.0 2.0 0.0 ... 0.0 1.87953 0.72765 1.592340 0.33411 2.19898 91.4227

19880 1.0 0.0 0.0 0.0 0.0 1.0 0.0 ... 0.0 2.07746 1.18286 1.943170 0.76265 4.19772 83.6387

19881 4.0 4.0 0.0 0.0 1.0 2.0 0.0 ... 0.0 1.89209 0.58831 1.620010 0.21044 2.37852 98.0665

[19882 rows x 4519 columns]

The following files have been created:

C:/Users/Enrique/Documents/GitHub/IRB/Models/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-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/TK\_CYP1A2inh/" folder is needed

This file must be called: "TK\_CYP1A2inh\_DeepPK\_final-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.

2162 features with a correlation magnitude greater than 0.90.

725 features with a single unique value.

Data has not been one-hot encoded

Removed 2887 features including one-hot features.

The following files have been created:

C:/Users/Enrique/Documents/GitHub/IRB/Models/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-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/TK\_CYP1A2inh/" folder is needed

This file must be called: "TK\_CYP1A2inh\_DeepPK\_final-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: {1, 19866, 3, 12}

ALERTS!!

compound number: 10412

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

compound number: 11865

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 1.0

compound number: 14220

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: 18698

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

[10412, 11865, 14220, 18698]

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: {1, 19866, 15}

ALERTS!!

compound number: 11865

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 1.0

[11865]

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: {19875, 7}

ALERTS!!

[]

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

0 0 CCOC(=O)Cc1csc(NC(=O)c2ccc(S(=O)(=O)N3CCCCC3)c... 1.0 1.0 4.0 ... 0.52011 1.451280 0.18233 2.67560 0

1 1 O=S(=O)(c1ccccc1)N1CCC2(CCN(c3ncccn3)CC2)CC1 0.0 0.0 4.0 ... 0.79763 1.678620 0.39537 2.54790 0

2 2 COc1ccccc1Cn1c(=S)[nH]c2cc(C(=O)N3CCN(C)CC3)cc... 0.0 0.0 0.0 ... 0.86372 1.699380 0.55517 2.50359 0

3 3 O=C(CP(=O)(c1ccccc1)c1ccccc1)Nn1cnnc1 0.0 0.0 0.0 ... 0.65751 1.368910 0.65751 1.36230 0

4 4 CN(C)Cc1ccccc1-c1cncnc1Nc1ccncc1 1.0 0.0 0.0 ... 1.24081 1.557130 0.78926 3.34380 0

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

19877 19877 CN(C)CCOC(=O)C(c1ccccc1)C1(O)CCCC1 0.0 0.0 4.0 ... 0.84422 1.896570 0.84422 2.18010 0

19878 19878 CC(=O)OC1C2CCCC1C([N+]1CCCC1)CC2 0.0 1.0 7.0 ... 1.26223 2.042357 1.26223 2.43070 0

19879 19879 CCNc1ncc2nc(-c3ccccc3)c(=O)n(CCC#N)c2n1 1.0 1.0 1.0 ... 0.72765 1.592340 0.33411 2.19898 0

19880 19880 Cc1n[nH]c(SCc2cccc(Oc3ccccc3)c2)n1 1.0 1.0 0.0 ... 1.18286 1.943170 0.76265 4.19772 0

19881 19881 CCCCN(CCCC)c1nc(OC)nc(-n2nnc(C(C)=O)c2C)n1 1.0 4.0 4.0 ... 0.58831 1.620010 0.21044 2.37852 0

[19882 rows x 1636 columns]

0

1

cluster0

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

0 0 CCOC(=O)Cc1csc(NC(=O)c2ccc(S(=O)(=O)N3CCCCC3)c... 1.0 1.0 4.0 ... 0.52011 1.451280 0.18233 2.67560 0

1 1 O=S(=O)(c1ccccc1)N1CCC2(CCN(c3ncccn3)CC2)CC1 0.0 0.0 4.0 ... 0.79763 1.678620 0.39537 2.54790 0

2 2 COc1ccccc1Cn1c(=S)[nH]c2cc(C(=O)N3CCN(C)CC3)cc... 0.0 0.0 0.0 ... 0.86372 1.699380 0.55517 2.50359 0

3 3 O=C(CP(=O)(c1ccccc1)c1ccccc1)Nn1cnnc1 0.0 0.0 0.0 ... 0.65751 1.368910 0.65751 1.36230 0

4 4 CN(C)Cc1ccccc1-c1cncnc1Nc1ccncc1 1.0 0.0 0.0 ... 1.24081 1.557130 0.78926 3.34380 0

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

19877 19877 CN(C)CCOC(=O)C(c1ccccc1)C1(O)CCCC1 0.0 0.0 4.0 ... 0.84422 1.896570 0.84422 2.18010 0

19878 19878 CC(=O)OC1C2CCCC1C([N+]1CCCC1)CC2 0.0 1.0 7.0 ... 1.26223 2.042357 1.26223 2.43070 0

19879 19879 CCNc1ncc2nc(-c3ccccc3)c(=O)n(CCC#N)c2n1 1.0 1.0 1.0 ... 0.72765 1.592340 0.33411 2.19898 0

19880 19880 Cc1n[nH]c(SCc2cccc(Oc3ccccc3)c2)n1 1.0 1.0 0.0 ... 1.18286 1.943170 0.76265 4.19772 0

19881 19881 CCCCN(CCCC)c1nc(OC)nc(-n2nnc(C(C)=O)c2C)n1 1.0 4.0 4.0 ... 0.58831 1.620010 0.21044 2.37852 0

[19875 rows x 1636 columns]

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

9017 9017 COc1ccc(NC(=O)N2CCC3(CCCN(C(=O)c4cccn4C)C3)C2)cc1 0.0 0.0 3.0 ... 0.75770 1.82866 0.71681 3.19390 0

2811 2811 Cc1cc(CNC(=O)C2C[C@H]2[C@@H](NP(=O)(c2ccccc2)c... 0.0 1.0 1.0 ... 1.05043 1.91148 0.75399 4.24082 0

8294 8294 Fc1ccc(Nc2ncnc3ccc(-c4ccoc4)cc23)cc1 1.0 0.0 0.0 ... 1.01686 1.40537 -0.27283 4.77250 0

11967 11967 COC(=O)C1C[C@H]1[C@@H](NC(=O)OCc1ccccc1)c1ccccc1 1.0 0.0 1.0 ... 0.95423 1.89735 0.92386 3.46320 0

14650 14650 C[C@]1(c2ccccc2)O[C@@H]2C[C@@H]3[C@@H]4CC=C5CC... 0.0 3.0 4.0 ... 0.74595 1.86444 0.74595 3.55560 0

[5 rows x 1636 columns]

cluster1

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

10273 10273 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 ... -0.16226 0.76920 -0.88220 0.1062 1

10412 10412 C=C(NC(=O)C(C)=NC(=O)c1csc(C2=NC3c4csc(n4)C4NC... 0.0 12.0 3.0 ... 0.22265 1.24983 -0.45358 1.4891 1

11865 11865 O=C1c2ccccc2C(=O)c2c(Nc3cc4c5c(ccc6c7ccc8c9c(c... 1.0 0.0 0.0 ... 0.80409 1.67448 0.24075 13.1980 1

14174 14174 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 ... -0.16226 0.76920 -0.88220 0.1062 1

14220 14220 C=C(NC(=O)C(=C)NC(=O)c1csc(C2=N[C@@H]3c4csc(n4... 0.0 11.0 3.0 ... 0.22176 1.24879 -0.45472 1.1315 1

15622 15622 CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=... 0.0 4.0 3.0 ... -0.16226 0.76920 -0.88220 0.1062 1

18698 18698 CC(C)(C)c1cc2c(O)c(c1)Cc1cc(C(C)(C)C)cc(c1O)Cc... 0.0 24.0 8.0 ... 0.97946 2.03815 0.07130 20.7512 1

[7 rows x 1636 columns]

Traceback (most recent call last):

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\NEO.py", line 1439, in <module>

main()

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\NEO.py", line 512, in main

back\_to\_main()

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\NEO.py", line 170, in back\_to\_main

main()

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\NEO.py", line 626, in main

back\_to\_main()

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\NEO.py", line 170, in back\_to\_main

main()

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\NEO.py", line 679, in main

data\_df,train\_set, test\_set = split\_by\_kmeans.create\_clusters(initial\_red, SEED, TEST\_SIZE,TARGET\_COL, model\_for\_split)

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\modules\split\_by\_kmeans.py", line 214, in create\_clusters

cluster\_train\_dict[train\_cluster\_name], cluster\_test\_dict[test\_cluster\_name] = split\_cluster(cluster\_value, SEED, TEST\_SIZE,TARGET\_COL, model\_for\_split)

File "C:\Users\Enrique\Documents\GitHub\generate\_models\NEO\modules\split\_by\_kmeans.py", line 41, in split\_cluster

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

File "C:\Users\Enrique\.conda\envs\Protocosas\lib\site-packages\sklearn\model\_selection\\_split.py", line 2441, in train\_test\_split

train, test = next(cv.split(X=arrays[0], y=stratify))

File "C:\Users\Enrique\.conda\envs\Protocosas\lib\site-packages\sklearn\model\_selection\\_split.py", line 1600, in split

for train, test in self.\_iter\_indices(X, y, groups):

File "C:\Users\Enrique\.conda\envs\Protocosas\lib\site-packages\sklearn\model\_selection\\_split.py", line 1940, in \_iter\_indices

raise ValueError(

ValueError: The least populated class in y has only 1 member, which is too few. The minimum number of groups for any class cannot be less than 2.

(Protocosas) C:\Users\Enrique\Documents\GitHub\generate\_models\NEO>python NEO.py

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

######################### 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/TK\_CYP1A2inh/

Please input your MODEL NAME (enter to: Af\_MIC80\_no3D): TK\_CYP1A2inh\_DeepPK\_final

######################### 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/TK\_CYP1A2inh/" folder is needed

This file must be called: "TK\_CYP1A2inh\_DeepPK\_final-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: {1, 19866, 3, 12}

ALERTS!!

compound number: 10412

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

compound number: 11865

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 1.0

compound number: 14220

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: 18698

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

[10412, 11865, 14220, 18698]

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)?1

Ok,eliminating

(19882, 1635)

(19878, 1635)

NUMBER OF CLUSTERS: 5

SETS: {3, 9, 175, 20, 19671}

ALERTS!!

compound number: 10273

SMILE: CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=O)NC3C(=O)NC(C(=O)NC(C(=O)O)c4cc(O)cc(O)c4-c4cc3ccc4O)[C@H](O)c3ccc(c(Cl)c3)Oc3cc2cc(c3O[C@@H]2O[C@@H](CO)[C@@H](O)[C@@H](O)[C@H]2O[C@H]2C[C@](C)(N)[C@H](O)[C@H](C)O2)Oc2ccc(cc2Cl)[C@@H]1O 0.0

compound number: 14172

SMILE: CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=O)NC3C(=O)NC(C(=O)NC(C(=O)O)c4cc(O)cc(O)c4-c4cc3ccc4O)C(O)c3ccc(c(Cl)c3)Oc3cc2cc(c3OC2OC(CO)C(O)C(O)C2OC2CC(C)(N)C(O)C(C)O2)Oc2ccc(cc2Cl)C1O 0.0

compound number: 15619

SMILE: CNC(CC(C)C)C(=O)NC1C(=O)NC(CC(N)=O)C(=O)NC2C(=O)NC3C(=O)NC(C(=O)NC(C(=O)O)c4cc(O)cc(O)c4-c4cc3ccc4O)[C@H](O)c3ccc(c(Cl)c3)Oc3cc2cc(c3O[C@H]2O[C@@H](CO)[C@@H](O)[C@@H](O)[C@@H]2O[C@@H]2C[C@](C)(N)[C@H](O)[C@H](C)O2)Oc2ccc(cc2Cl)[C@@H]1O 0.0

[10273, 14172, 15619]

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)?1

Ok,eliminating

(19878, 1635)

(19875, 1635)

NUMBER OF CLUSTERS: 5

SETS: {18693, 9, 79, 1074, 20}

ALERTS!!

[]

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

0 0 CCOC(=O)Cc1csc(NC(=O)c2ccc(S(=O)(=O)N3CCCCC3)c... 1.0 1.0 4.0 ... 0.52011 1.451280 0.18233 2.67560 0

1 1 O=S(=O)(c1ccccc1)N1CCC2(CCN(c3ncccn3)CC2)CC1 0.0 0.0 4.0 ... 0.79763 1.678620 0.39537 2.54790 0

2 2 COc1ccccc1Cn1c(=S)[nH]c2cc(C(=O)N3CCN(C)CC3)cc... 0.0 0.0 0.0 ... 0.86372 1.699380 0.55517 2.50359 0

3 3 O=C(CP(=O)(c1ccccc1)c1ccccc1)Nn1cnnc1 0.0 0.0 0.0 ... 0.65751 1.368910 0.65751 1.36230 0

4 4 CN(C)Cc1ccccc1-c1cncnc1Nc1ccncc1 1.0 0.0 0.0 ... 1.24081 1.557130 0.78926 3.34380 0

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

19877 19877 CN(C)CCOC(=O)C(c1ccccc1)C1(O)CCCC1 0.0 0.0 4.0 ... 0.84422 1.896570 0.84422 2.18010 0

19878 19878 CC(=O)OC1C2CCCC1C([N+]1CCCC1)CC2 0.0 1.0 7.0 ... 1.26223 2.042357 1.26223 2.43070 0

19879 19879 CCNc1ncc2nc(-c3ccccc3)c(=O)n(CCC#N)c2n1 1.0 1.0 1.0 ... 0.72765 1.592340 0.33411 2.19898 0

19880 19880 Cc1n[nH]c(SCc2cccc(Oc3ccccc3)c2)n1 1.0 1.0 0.0 ... 1.18286 1.943170 0.76265 4.19772 0

19881 19881 CCCCN(CCCC)c1nc(OC)nc(-n2nnc(C(C)=O)c2C)n1 1.0 4.0 4.0 ... 0.58831 1.620010 0.21044 2.37852 0

[19875 rows x 1636 columns]

0

1

2

3

4

cluster0

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

0 0 CCOC(=O)Cc1csc(NC(=O)c2ccc(S(=O)(=O)N3CCCCC3)c... 1.0 1.0 4.0 ... 0.52011 1.451280 0.18233 2.67560 0

1 1 O=S(=O)(c1ccccc1)N1CCC2(CCN(c3ncccn3)CC2)CC1 0.0 0.0 4.0 ... 0.79763 1.678620 0.39537 2.54790 0

2 2 COc1ccccc1Cn1c(=S)[nH]c2cc(C(=O)N3CCN(C)CC3)cc... 0.0 0.0 0.0 ... 0.86372 1.699380 0.55517 2.50359 0

3 3 O=C(CP(=O)(c1ccccc1)c1ccccc1)Nn1cnnc1 0.0 0.0 0.0 ... 0.65751 1.368910 0.65751 1.36230 0

4 4 CN(C)Cc1ccccc1-c1cncnc1Nc1ccncc1 1.0 0.0 0.0 ... 1.24081 1.557130 0.78926 3.34380 0

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

19877 19877 CN(C)CCOC(=O)C(c1ccccc1)C1(O)CCCC1 0.0 0.0 4.0 ... 0.84422 1.896570 0.84422 2.18010 0

19878 19878 CC(=O)OC1C2CCCC1C([N+]1CCCC1)CC2 0.0 1.0 7.0 ... 1.26223 2.042357 1.26223 2.43070 0

19879 19879 CCNc1ncc2nc(-c3ccccc3)c(=O)n(CCC#N)c2n1 1.0 1.0 1.0 ... 0.72765 1.592340 0.33411 2.19898 0

19880 19880 Cc1n[nH]c(SCc2cccc(Oc3ccccc3)c2)n1 1.0 1.0 0.0 ... 1.18286 1.943170 0.76265 4.19772 0

19881 19881 CCCCN(CCCC)c1nc(OC)nc(-n2nnc(C(C)=O)c2C)n1 1.0 4.0 4.0 ... 0.58831 1.620010 0.21044 2.37852 0

[18693 rows x 1636 columns]

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

13326 13326 N#Cc1ccc(C2C3=C(CCCC3=O)N=C3CCCC(=O)C32)cc1 1.0 0.0 6.0 ... 0.94852 1.76547 0.58574 3.47278 0

1387 1387 CC(N)Cc1c[nH]c2ccc(O)cc12 0.0 1.0 1.0 ... 1.15824 1.67509 0.29447 1.76320 0

18192 18192 O=C(CSc1nc(=O)c2c(C(F)(F)F)cc(-c3cccs3)nc2[nH]... 1.0 0.0 0.0 ... 0.56998 0.84721 -1.10890 4.06690 0

5219 5219 CC(C(=O)NCC1CC1)[C@@H]1C[C@@]1(C)[C@@H](NC(=O)... 1.0 2.0 3.0 ... 1.03681 1.92546 0.94942 4.84260 0

6432 6432 CC(=NOC(=O)c1ccccc1)c1cc(-c2ccc(Cl)cc2)no1 1.0 1.0 0.0 ... 0.86433 1.65305 0.41742 4.57600 0

[5 rows x 1636 columns]

cluster1

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

242 242 CC(=O)O[C@@H]1CC[C@]2(C)C(=C[C@@H]([C@@H]3C=C4... 0.0 10.0 16.0 ... 1.300887 1.92985 0.55366 12.1636 1

411 411 Oc1ccc2c3c1OC1c4[nH]c5c(c4CC4(O)C(C2)N(CC2CC2)... 0.0 0.0 10.0 ... 1.186660 1.77258 0.14163 3.8241 1

670 670 CC(=O)OC1CCC2(C)C(=CC(C3C=C4CC(OC(C)=O)CCC4(C)... 0.0 10.0 16.0 ... 1.300887 1.92985 0.55366 12.1636 1

10735 10735 COC1C(O)C(CO)OC(OC2C(O)COC(OC3C(CO)OC(OC4C(OC5... 0.0 8.0 9.0 ... 1.529340 2.24439 1.37772 -3.8524 1

12051 12051 C=CCN1/C(=C\CO)[C@H]2C[C@@H]3[C@@H]1CC[C@]31c3... 0.0 0.0 6.0 ... 1.529340 2.24439 1.37772 6.3736 1

12179 12179 Oc1ccc2c3c1O[C@@H]1c4[nH]c5c(c4C[C@@]4(O)[C@@H... 0.0 0.0 10.0 ... 1.186660 1.77258 0.14163 3.8241 1

13245 13245 C=CCN1/C(=C\CO)[C@H]2C[C@@H]3[C@@H]1CC[C@]31c3... 0.0 0.0 6.0 ... 1.529340 2.24439 1.37772 6.3736 1

13489 13489 CC(=O)O[C@@H]1CC[C@]2(C)C(=C[C@@H]([C@@H]3C=C4... 0.0 10.0 16.0 ... 1.300887 1.92985 0.55366 12.1636 1

15802 15802 C=CC[N+]1C(=CCO)C2CC3C1CCC31c3ccccc3N3C=C4C5CC... 0.0 0.0 6.0 ... 1.529340 2.24439 1.37772 6.0498 1

[9 rows x 1636 columns]

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

15802 15802 C=CC[N+]1C(=CCO)C2CC3C1CCC31c3ccccc3N3C=C4C5CC... 0.0 0.0 6.0 ... 1.529340 2.24439 1.37772 6.0498 1

411 411 Oc1ccc2c3c1OC1c4[nH]c5c(c4CC4(O)C(C2)N(CC2CC2)... 0.0 0.0 10.0 ... 1.186660 1.77258 0.14163 3.8241 1

13489 13489 CC(=O)O[C@@H]1CC[C@]2(C)C(=C[C@@H]([C@@H]3C=C4... 0.0 10.0 16.0 ... 1.300887 1.92985 0.55366 12.1636 1

242 242 CC(=O)O[C@@H]1CC[C@]2(C)C(=C[C@@H]([C@@H]3C=C4... 0.0 10.0 16.0 ... 1.300887 1.92985 0.55366 12.1636 1

670 670 CC(=O)OC1CCC2(C)C(=CC(C3C=C4CC(OC(C)=O)CCC4(C)... 0.0 10.0 16.0 ... 1.300887 1.92985 0.55366 12.1636 1

[5 rows x 1636 columns]

cluster2

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

956 956 O=C1c2cc([N+](=O)[O-])ccc2-c2ccc3c4ccc5c6c(ccc... 1.0 0.0 0.0 ... 0.84245 1.348380 -0.129140 7.97640 2

1017 1017 CCC1(O)CC2CN(CCc3c([nH]c4ccccc34)C(C(=O)OC)(c3... 0.0 3.0 6.0 ... 1.01918 1.944880 0.455190 3.99090 2

1962 1962 Cc1ccc(C(=O)NC2C(=O)NC(C(C)C)C(=O)N3CCCC3C(=O)... 0.0 12.0 4.0 ... 0.39286 1.349660 -0.622110 0.84834 2

2201 2201 O=C(Nc1ccc(C(=O)Nc2ccc(C(=O)Nc3ccc(S(=O)(=O)O)... 1.0 0.0 0.0 ... 0.24340 0.309420 -1.246660 6.12640 2

2416 2416 CCC1(O)CC2CN(CCc3c([nH]c4ccccc34)C(C(=O)OC)(c3... 0.0 3.0 6.0 ... 0.96585 1.744880 0.307050 3.51750 2

3522 3522 CCC1(O)CC2CN(CCc3c([nH]c4ccccc34)C(C(=O)OC)(c3... 0.0 2.0 6.0 ... 0.96279 1.737250 0.300550 3.37500 2

4188 4188 O=C1CNCCCN2CCN(CCCNCC(=O)Nc3ccc(cc3)S(=O)(=O)c... 0.0 0.0 4.0 ... 0.48804 1.465250 0.086960 1.97400 2

4216 4216 Cc1ccc(C(=O)NC2C(=O)NC(C(C)C)C(=O)N3CCCC3C(=O)... 0.0 12.0 4.0 ... 0.39286 1.349660 -0.622110 0.84834 2

5964 5964 C=CCN1CC[C@@]23c4c5ccc(O)c4O[C@H]2/C(=N\N=C2/C... 0.0 0.0 8.0 ... 1.10588 1.704530 0.088480 3.27520 2

9701 9701 CC(=O)O[C@H]1C[C@@H](O[C@@H]2[C@H](C)O[C@H](O[... 0.0 8.0 11.0 ... 1.03006 1.540263 0.661113 1.39330 2

9890 9890 CCC1(O)CC2CN(CCc3c([nH]c4ccccc34)C(C(=O)OC)(c3... 0.0 2.0 6.0 ... 1.01612 1.937250 0.448700 3.84840 2

12068 12068 CC[C@]1(O)C[C@@H]2CN(CCc3c([nH]c4ccccc34)[C@](... 0.0 2.0 6.0 ... 1.01612 1.937250 0.448700 3.84840 2

12549 12549 CC[C@]1(O)C[C@H]2CN(CCc3c([nH]c4ccccc34)[C@](C... 0.0 3.0 6.0 ... 0.96585 1.744880 0.307050 3.51750 2

12636 12636 CC1=C2N=C(C=C3N=C(C(C)=C4NC(C(CC(N)=O)C4(C)CCC... 0.0 11.0 11.0 ... 0.99215 1.458130 0.947430 3.16324 2

12857 12857 C=CCN1CCC23c4c5ccc(O)c4OC2C(=NN=C2CCC4(O)C6Cc7... 0.0 0.0 8.0 ... 1.10588 1.704530 0.088480 3.27520 2

14936 14936 CC[C@]1(O)C[C@@H]2CN(CCc3c([nH]c4ccccc34)[C@](... 0.0 2.0 6.0 ... 0.96279 1.737250 0.300550 3.37500 2

16983 16983 CC[C@]1(O)C[C@H]2CN(CCc3c([nH]c4ccccc34)[C@](C... 0.0 3.0 6.0 ... 1.01918 1.944880 0.455190 3.99090 2

17379 17379 CC1=C2N=C(C=C3N=C(C(C)=C4N[C@@H]([C@@H](CC(N)=... 0.0 11.0 11.0 ... 0.99215 1.458130 0.947430 3.16324 2

17464 17464 CC(=O)OC1CC(OC2C(O)CC(OC3C(O)CC(OC4CCC5(C)C(CC... 0.0 8.0 11.0 ... 1.03006 1.540263 0.661113 1.39330 2

19021 19021 O=C1CN2CCN(CC2)CC(=O)Nc2ccc(cc2)S(=O)(=O)c2ccc... 0.0 0.0 0.0 ... 0.44693 1.431440 0.052640 2.05520 2

[20 rows x 1636 columns]

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

12857 12857 C=CCN1CCC23c4c5ccc(O)c4OC2C(=NN=C2CCC4(O)C6Cc7... 0.0 0.0 8.0 ... 1.10588 1.704530 0.088480 3.27520 2

2201 2201 O=C(Nc1ccc(C(=O)Nc2ccc(C(=O)Nc3ccc(S(=O)(=O)O)... 1.0 0.0 0.0 ... 0.24340 0.309420 -1.246660 6.12640 2

17464 17464 CC(=O)OC1CC(OC2C(O)CC(OC3C(O)CC(OC4CCC5(C)C(CC... 0.0 8.0 11.0 ... 1.03006 1.540263 0.661113 1.39330 2

9890 9890 CCC1(O)CC2CN(CCc3c([nH]c4ccccc34)C(C(=O)OC)(c3... 0.0 2.0 6.0 ... 1.01612 1.937250 0.448700 3.84840 2

12636 12636 CC1=C2N=C(C=C3N=C(C(C)=C4NC(C(CC(N)=O)C4(C)CCC... 0.0 11.0 11.0 ... 0.99215 1.458130 0.947430 3.16324 2

[5 rows x 1636 columns]

cluster3

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

125 125 C/C=C/CC(C)C(O)C1C(=O)NC(CC)C(=O)N(C)CC(=O)N(C... 0.0 17.0 6.0 ... 0.576243 1.417357 0.353703 3.26900 3

197 197 COc1cc2c3cc1Oc1cc(ccc1O)CC1c4c(cc(OC)c(O)c4Oc4... 0.0 0.0 4.0 ... 1.257510 1.698780 0.000670 6.84560 3

899 899 CC1CCC2(C(=O)OC3OC(COC4OC(CO)C(OC5OC(C)C(O)C(O... 0.0 7.0 8.0 ... 0.999603 1.806020 0.461000 -1.03280 3

1050 1050 NCC(=O)NC1CSSCC(C(N)=O)NC(=O)C(Cc2c[nH]c3ccccc... 0.0 0.0 7.0 ... 0.661870 1.671230 0.613870 -2.08430 3

1187 1187 CO[C@H]1C=CO[C@]2(C)Oc3c(C)c(O)c4c(O)c(c5c(c4c... 0.0 10.0 2.0 ... 0.156250 1.392963 -0.489320 4.74942 3

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

18649 18649 Oc1ccc2c3c1O[C@H]1c4c(c5ccccc5n4Cc4ccccc4)C[C@... 1.0 0.0 5.0 ... 1.244640 1.799500 0.195380 5.09430 3

18763 18763 CC[C@H](C)[C@@H]1O[C@]2(CC[C@@H]1C)C[C@H]1C[C@... 0.0 8.0 8.0 ... 0.204743 1.392963 -0.576663 5.60140 3

19219 19219 CC1=C2N=C(CC3=NC(C(C)C4=NC(C(CC(N)=O)C4(C)CCC(... 1.0 9.0 12.0 ... 0.746333 1.424773 0.321017 1.82430 3

19803 19803 CC(=O)NC1C(OP(=O)(O)OP(=O)(O)OCC=C(C)CCC=C(C)C... 1.0 17.0 25.0 ... 0.553323 1.226000 -0.040363 13.84000 3

19831 19831 CC(C)CCCCC(=O)NC(CCN)C(=O)NC(C(=O)NC(CCN)C(=O)... 0.0 8.0 11.0 ... 1.006110 1.755657 0.428123 -5.97400 3

[79 rows x 1636 columns]

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

11415 11415 CO[C@@H]1C=CO[C@]2(C)Oc3c(C)c(O)c4c(O)c(c(CN5C... 0.0 8.0 0.0 ... 0.130690 1.395037 -0.549570 4.50152 3

18763 18763 CC[C@H](C)[C@@H]1O[C@]2(CC[C@@H]1C)C[C@H]1C[C@... 0.0 8.0 8.0 ... 0.204743 1.392963 -0.576663 5.60140 3

12955 12955 CCC(C)C(=O)O[C@@H]1[C@H](O)[C@@H]2[C@H](CN3C[C... 0.0 9.0 7.0 ... 1.247383 1.696723 0.000747 1.60720 3

13209 13209 C[C@@H]1[C@H](C)CC[C@@]2(C(=O)O[C@H]3O[C@@H](C... 0.0 7.0 8.0 ... 0.999603 1.806020 0.461000 -1.03280 3

9703 9703 C/C=C\C[C@@H](C)[C@@H](O)C1C(=O)NC(CC)C(=O)N(C... 1.0 17.0 6.0 ... 0.576243 1.417357 0.353703 3.26900 3

[5 rows x 1636 columns]

cluster4

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

10 10 CC1=C2C(=O)C3C(CC=C4CC(O)CCC43C)C2CCC12OC1CC(C... 0.0 4.0 7.0 ... 1.229137 2.049063 1.10393 4.18090 4

21 21 CO[C@@H]1COC(=O)C(COCc2ccccc2)NC(=O)CC=C[C@@H]... 0.0 2.0 2.0 ... 0.918110 1.885540 0.91811 3.68440 4

30 30 CC(=O)OC[C@@H]1O[C@@H](O/N=C2/C[C@@H](O)[C@@H]... 0.0 4.0 5.0 ... 0.458100 1.831750 -0.23546 0.30550 4

39 39 CN1CC(C(=O)N[C@]2(C)O[C@]3(O)[C@H]4CCCN4C(=O)C... 0.0 1.0 5.0 ... 1.309740 1.884430 0.87423 2.08110 4

41 41 COc1ccc(NS(=O)(=O)c2ccc(NC(=O)c3cnn4c(C(F)F)cc... 1.0 1.0 0.0 ... 0.998760 1.166740 -0.40413 5.70402 4

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

19792 19792 COC(=O)[C@@]1(Cc2ccc(OC)cc2)[C@H]2c3cc(C(=O)N4... 0.0 0.0 4.0 ... 0.851730 1.746880 0.42954 2.65230 4

19793 19793 COC(=O)C1(Cc2ccc(F)cc2)C2c3cc(C(=O)N4CCCC4)n(C... 0.0 0.0 5.0 ... 1.020990 1.372120 -0.37909 5.06730 4

19797 19797 CC(=O)OC[C@H]1O[C@@H](CCO/N=C2/[C@H]3O[C@@H]3[... 0.0 2.0 2.0 ... 0.453500 1.685710 0.45350 0.08470 4

19825 19825 N=C(N)Nc1ccc2[nH]c3c(c2c1)C[C@]1(O)[C@@H]2Cc4c... 0.0 0.0 5.0 ... 1.197200 1.749300 0.15217 2.87797 4

19852 19852 C[C@H]1CC[C@@]2(NC1)OC1C[C@H]3[C@@H]4CC=C5C[C@... 0.0 4.0 9.0 ... 1.330877 2.047377 0.95069 5.28690 4

[1074 rows x 1636 columns]

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

8716 8716 CC(=O)OCC1OC(ON=C2CC(O)C(O)C3c4c(c(O)n(C5CCCCC... 0.0 4.0 8.0 ... 0.520670 1.831750 -0.141570 1.6317 4

3005 3005 CC1(C)[C@@H](OC(=O)CCC(=O)O)CC[C@]2(C)C3C(=O)C... 0.0 7.0 11.0 ... 0.034263 0.776767 -0.884517 6.8283 4

1547 1547 O=S1(=O)[C@@H](c2ccccc2)[C@H](N2CCCCC2)[C@H]1[... 0.0 0.0 4.0 ... 1.410470 1.982270 0.968600 5.6765 4

13125 13125 COc1cc2c(cc1OC)C13CC[N+]4(C)CC5=CCOC6CC(=O)N2C... 0.0 0.0 3.0 ... 1.045020 1.948170 0.619570 1.5743 4

18180 18180 CCn1c(O)c2c(c1O)[C@H]1[C@H](O)[C@H](O)C/C(=N/O... 0.0 5.0 3.0 ... 0.470930 1.346763 -0.219720 0.1464 4

[5 rows x 1636 columns]

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

13326 13326 N#Cc1ccc(C2C3=C(CCCC3=O)N=C3CCCC(=O)C32)cc1 1.0 0.0 6.0 ... 0.94852 1.76547 0.58574 3.47278 0

1387 1387 CC(N)Cc1c[nH]c2ccc(O)cc12 0.0 1.0 1.0 ... 1.15824 1.67509 0.29447 1.76320 0

18192 18192 O=C(CSc1nc(=O)c2c(C(F)(F)F)cc(-c3cccs3)nc2[nH]... 1.0 0.0 0.0 ... 0.56998 0.84721 -1.10890 4.06690 0

5219 5219 CC(C(=O)NCC1CC1)[C@@H]1C[C@@]1(C)[C@@H](NC(=O)... 1.0 2.0 3.0 ... 1.03681 1.92546 0.94942 4.84260 0

6432 6432 CC(=NOC(=O)c1ccccc1)c1cc(-c2ccc(Cl)cc2)no1 1.0 1.0 0.0 ... 0.86433 1.65305 0.41742 4.57600 0

[5 rows x 1636 columns]

Train set contains:

8177 negative values

7722 positive values

ratio neg / pos: 1.0589225589225588

Test set contains:

2045 negative values

1931 positive values

ratio neg / pos: 1.0590367685137234

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/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-cleaned\_from\_kmeans.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-train\_set.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-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/TK\_CYP1A2inh/" folder are needed

These files must be called:

"TK\_CYP1A2inh\_DeepPK\_final-train\_set.csv"

"TK\_CYP1A2inh\_DeepPK\_final-test\_set.csv"

Continue (Y/n)?y

The following files have been created:

C:/Users/Enrique/Documents/GitHub/IRB/Models/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-stand\_train\_set.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-stand\_test\_set.csv

C:/Users/Enrique/Documents/GitHub/IRB/Models/TK\_CYP1A2inh/TK\_CYP1A2inh\_DeepPK\_final-alldataset.sca

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

Thanks for using NEO!