In silico protein target prediction with reliability-density neighbourhood applicability domain analysis

Lewis Mervin and Natalia Aniceto

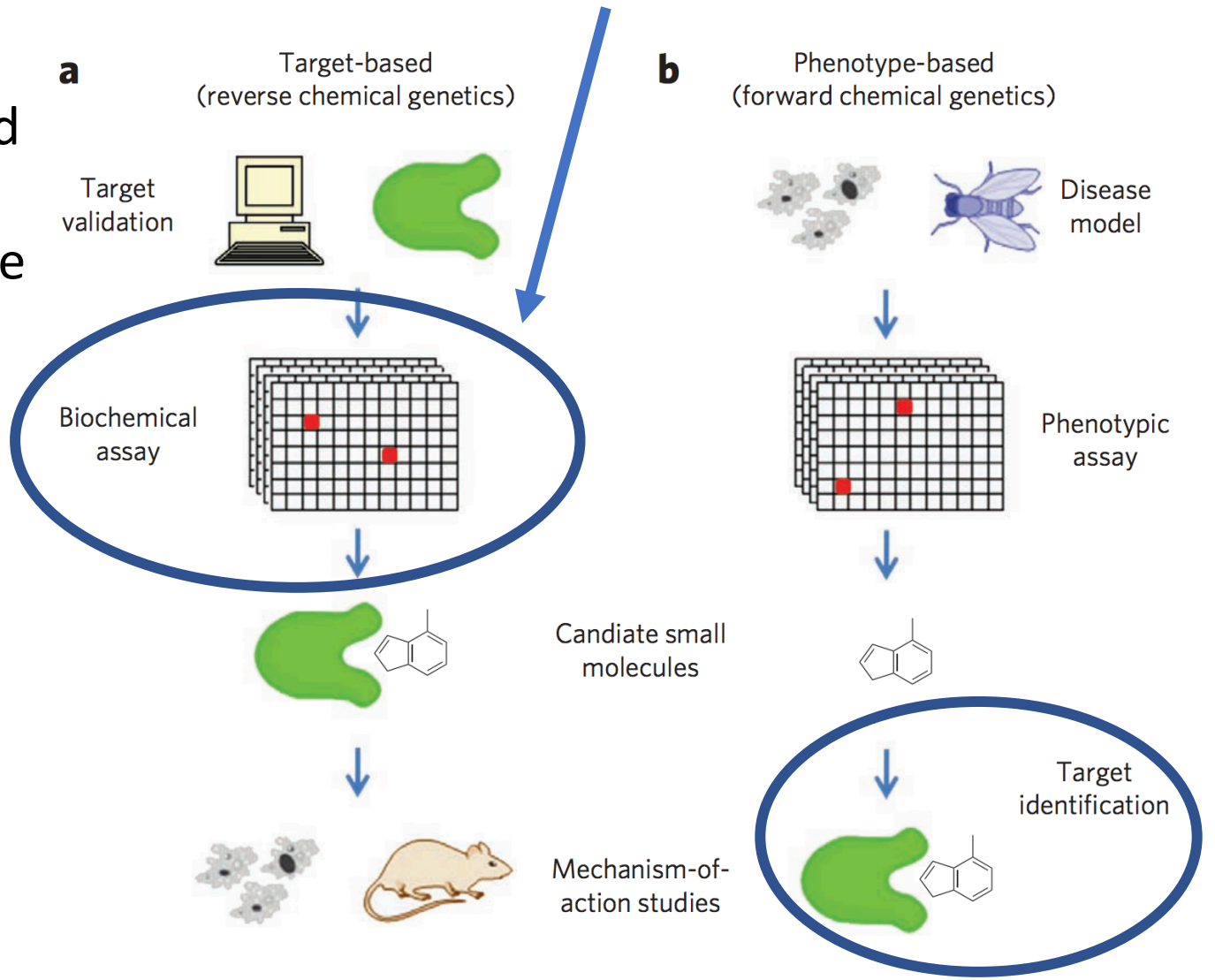
Postdoctoral Research Associates

Centre for Molecular Informatics
Department of Chemistry
University of Cambridge

UNIVERSITY OF CAMBRIDGE

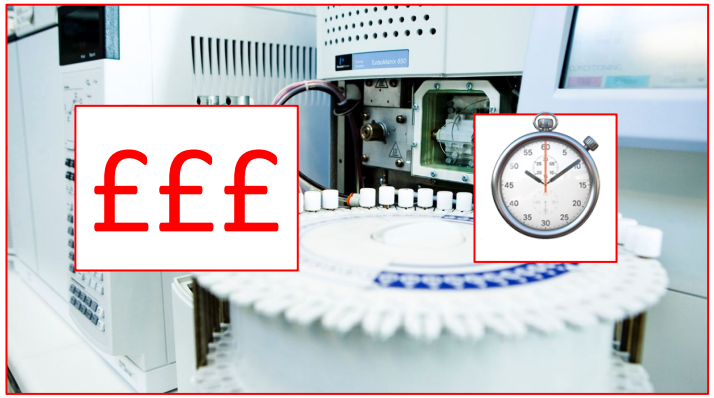
Why do target prediction?

- **Target-based screening (a)**
  - Target validation of protein established in pathway/disease
  - Biochemical assay to identify candidate small molecules
  - Mechanism-of-Action (MoA) studies validate cellular activity
- **Phenotypic-based screening (b)**
  - Phenotypic assay of small molecules that perturb the phenotype
  - Candidate molecules undergo target-deconvolution
  - MoA studies determine the protein responsible for phenotypic change



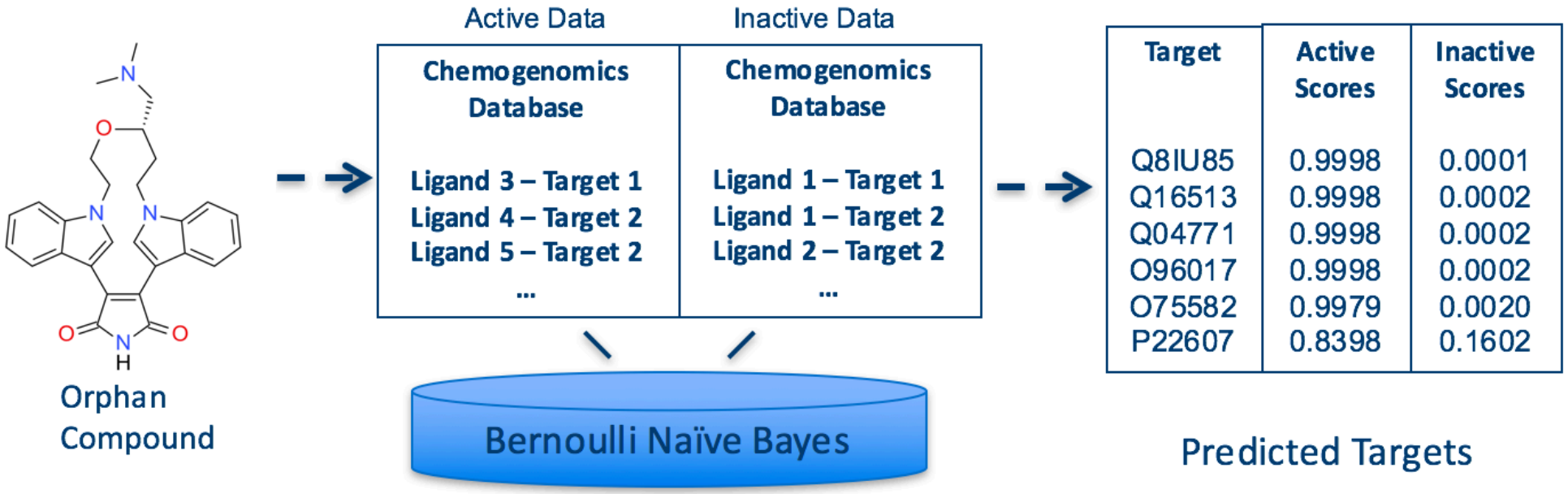
Schenone, Monica, et al. "Target identification and mechanism of action in chemical biology and drug discovery." *Nature chemical biology* 9.4 (2013): 232.

Why *in silico* prediction?



- **Biochemical methods** are available to explicitly discover target-ligand interactions of interest – but are **expensive** and **time-consuming**
- *In silico* Target prediction offers a **quick**, **inexpensive** alternative to **infer** target-ligand interactions using machine learning techniques
- Help **direct resources** for biochemical confirmation afterward

How does protein target prediction work?



PIDGIN v1: https://github.com/lhm30/PIDGIN

- **Machine learning** algorithms trained on **bioactivity data** (SAR relationships) from chemogenomic databases such as ChEMBL or PubChem
- Models provide a **probability of activity** for an orphan compound to bind a protein target based on information from **known ligand structures**
- Can be trained using the **fingerprints** of compound structures

Binary circular (ECFP_4) fingerprints

- ECFP_4 Morgan-like **Fingerprints** generated using Rdkit (2048 bits)

Activity	Bits ->
1	010001010001
1	010011010001
1	010001010101
y	0100 … X … 10101
0	0100 … 10001
0	01 … 01
0	01 … 01
0	011000010100

NumPy

```
fit (X, y, sample_weight=None)
```

Build a forest of trees from the training set (X, y).

Parameters: **X** : array-like or sparse matrix of s

```
>>> from sklearn.ensemble import RandomForestClassifier
>>> clf = RandomForestClassifier(n_estimators=10)
>>> clf = clf.fit(X, Y)
```

```
>>> class_weights = class_weight.compute_class_weight('balanced',np.unique(y),y)
>>> sw = np.array([class_weights[1] if i == 1 else class_weights[0] for i in y])
>>> clf = RandomForestClassifier(n_jobs=1, n_estimators=ntree, \
    class_weight='balanced',random_state=123)
>>> clf.fit(x,y,sample_weight = sw)
>>> probs = clf.predict_proba(test_x)[:,1]
```

- Bits represent the **presence** or **absence** of **chemical features** in a molecule

- Scikit-learn: to train a **classifier** using fingerprints

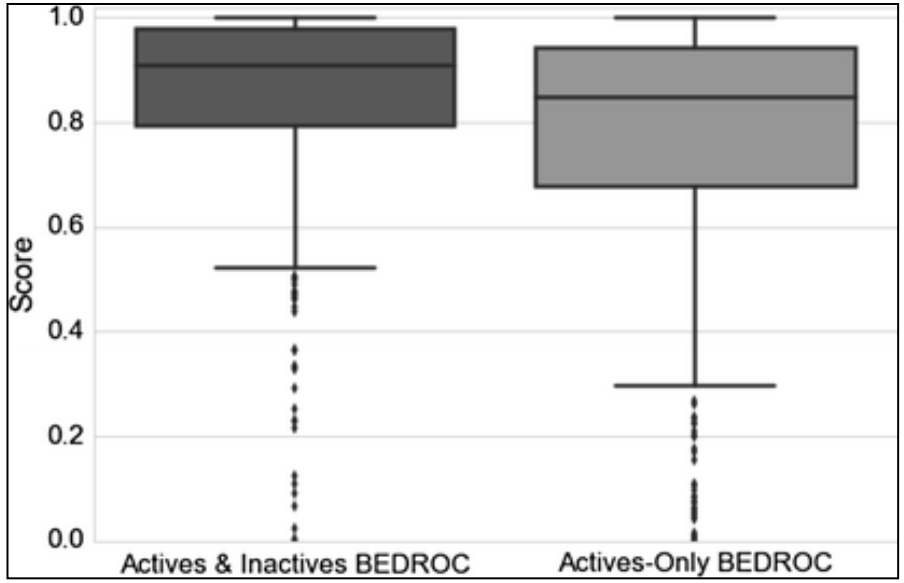
RDKit
Open-Source Cheminformatics and Machine Learning

Rogers D & Hahn M. Extended-connectivity fingerprints. J. Chem. Inf. Model. 50: 742-54 (2010).

Prediction IncluDinG Inactivity

Read the Docs

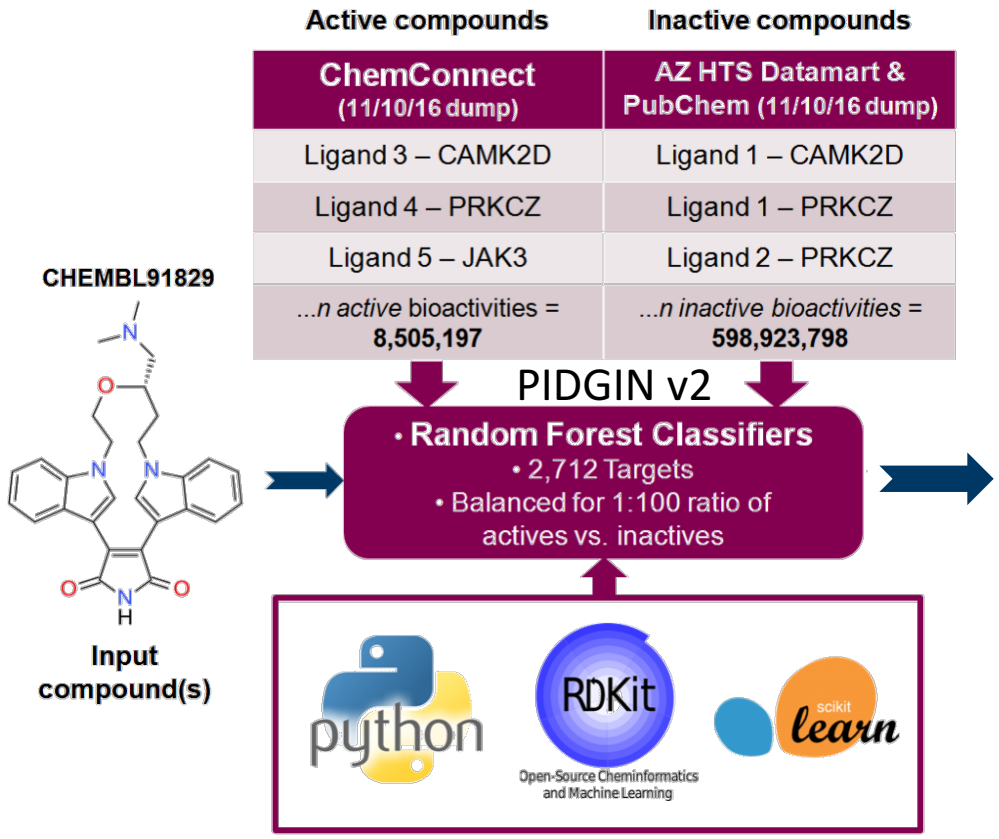
- PIDGIN: target prediction tools trained on actives and inactive bioactivity data from ChEMBL and PubChem
- Consider chemical features in both the active and negative bioactive sets
- **Data driven**: reliability of models depends on chemical space of bioactivity data
- Large chemical space (13+ million compounds)
- Benchmarked using WOMBAT and AstraZeneca data sources
- Models produce a probability of activity or inactivity for an orphan compound

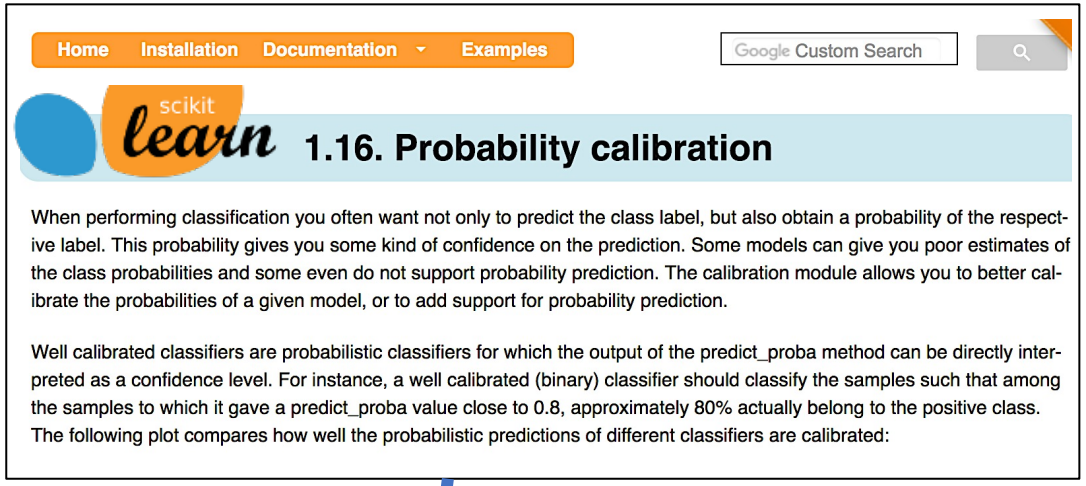


Mervin, L. H., et al., (2015). Target prediction utilising negative bioactivity data covering large chemical space. Journal of Cheminformatics, 7, 1-16.

Architecture of PIDGIN

- Published using ChEMBL and PubChem data[1] and AstraZeneca (proprietary) data[2]





Raw output

Target Identifier	Random Forest Score
PRKCZ	1.00
CAMK2D	1.00
PTK6	0.99
JAK3	0.99
...n targets =2,712	...

(Optional)

Scaled probability

Target Identifier	Scaled Probability
PRKCB	0.99999
PRKCZ	0.99966
CAMK4	0.99922
PTK6	0.99969
...n targets =2,712	...

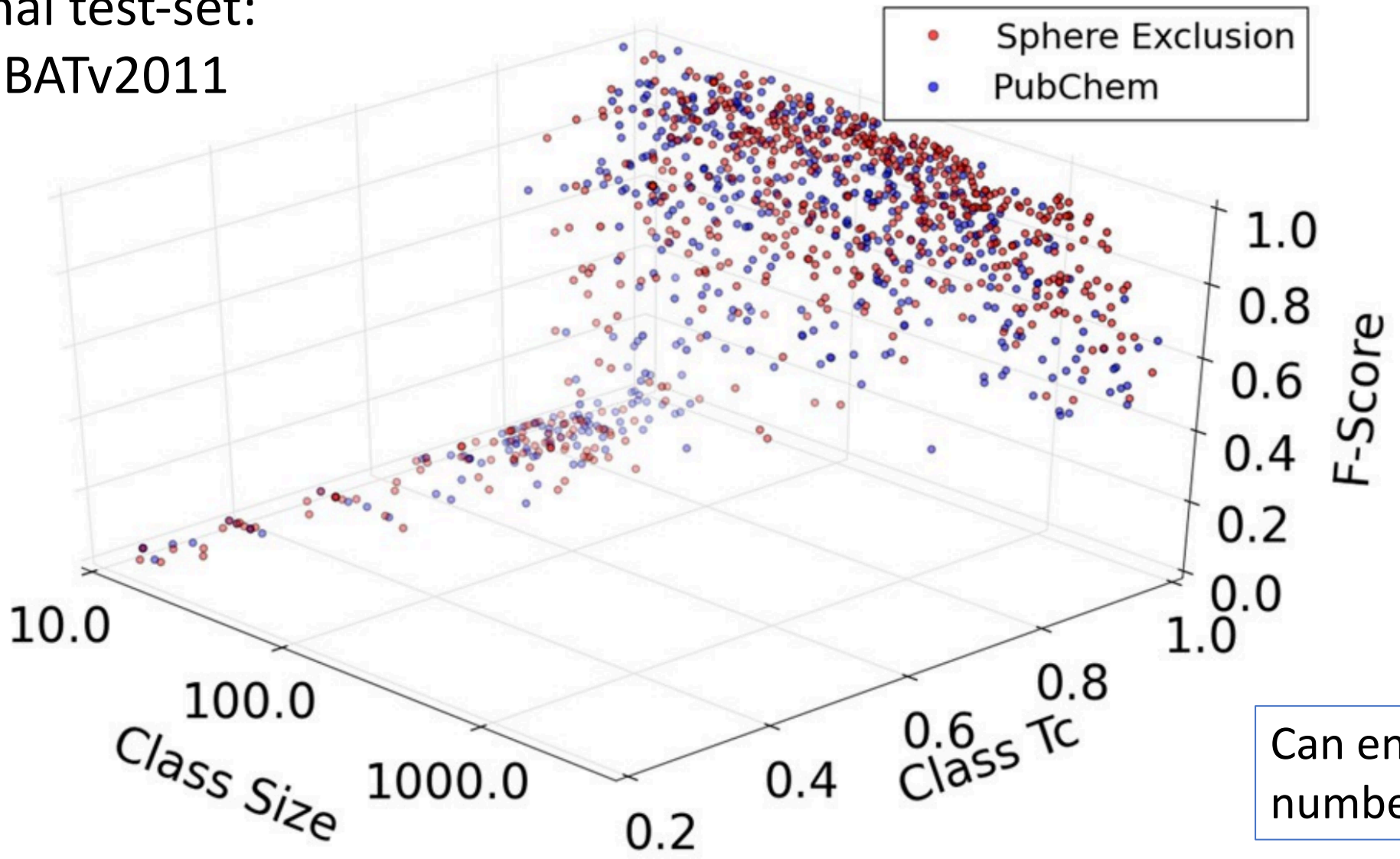
PIDGIN v2:
https://github.com/lhm30/PIDGINv2

[1] Mervin, L. H., *et al*. ***Target prediction utilising negative bioactivity data covering large chemical space***. *Journal of Cheminformatics*, *7*, 1-16. (2015)
[2] Mervin, L. H., *et al.* ***Understanding Cytotoxicity and Cytostaticity in a High-Throughput Screening Collection***. ACS Chem. Biol. (2016)

How do the models perform?

Choice of background/inactive set ***hugely*** impacts performance

External test-set:
WOMBATv2011

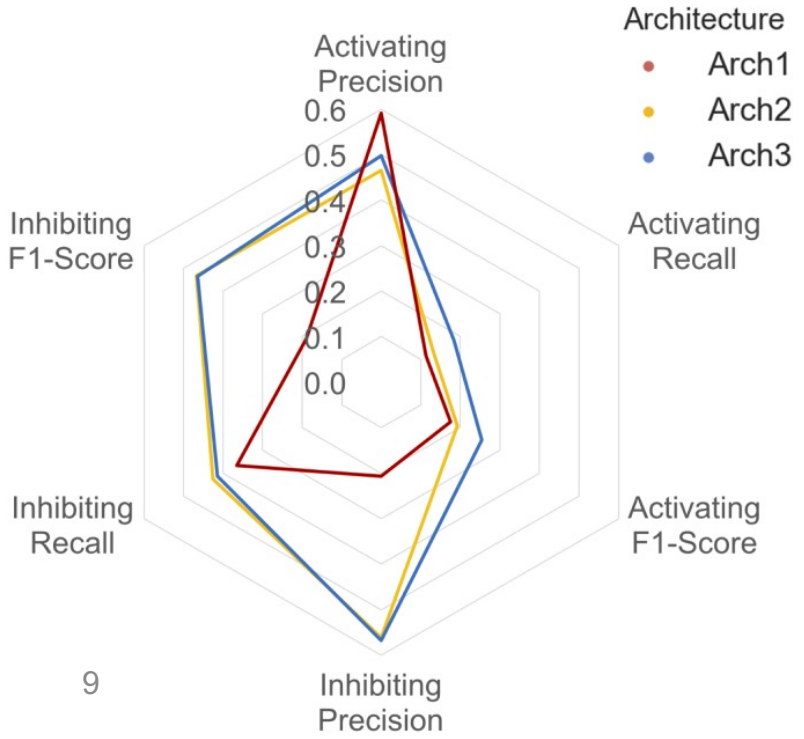


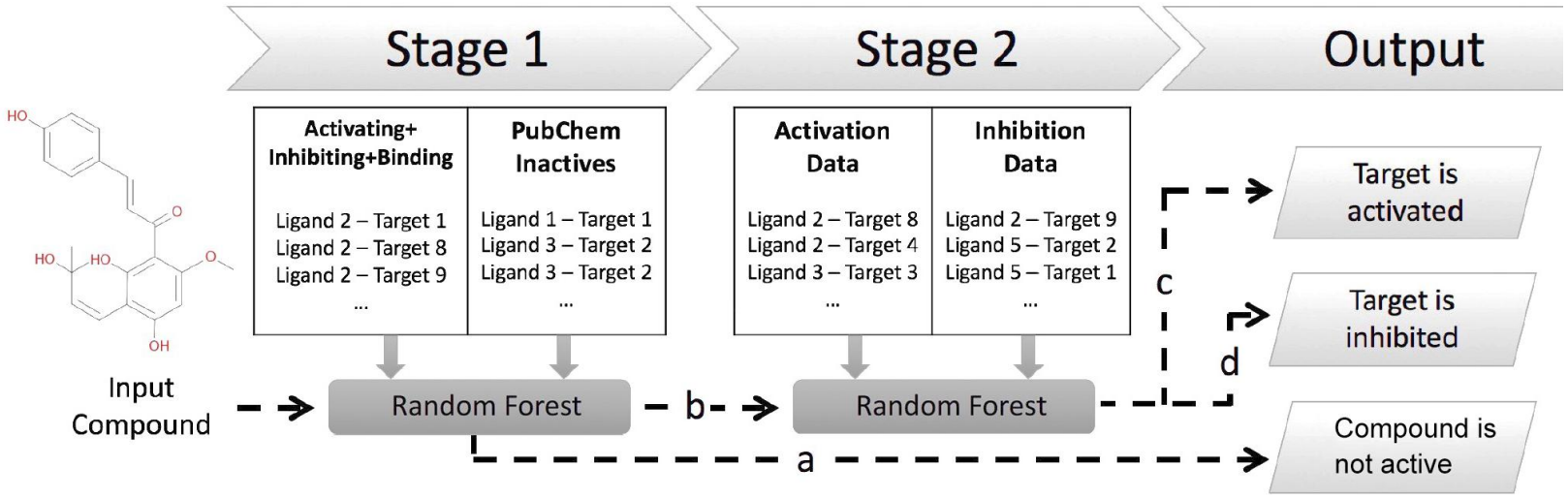
Can entirely choose numbers

Also working on function prediction

Prospective validation performed in-house

- Published cascaded models with AstraZeneca which extend target prediction with predicted functional effects
- Combine Stage 1 target prediction with Stage 2 function effect prediction
- Working on public models





Mervin, L.H, et al. ***Extending in silico protein target prediction models to include functional effects***. Frontiers in Pharmacology (2018).

PIDGIN version 3

https://pidginv3.readthedocs.io/en/latest/
https://github.com/lhm30/PIDGINv3

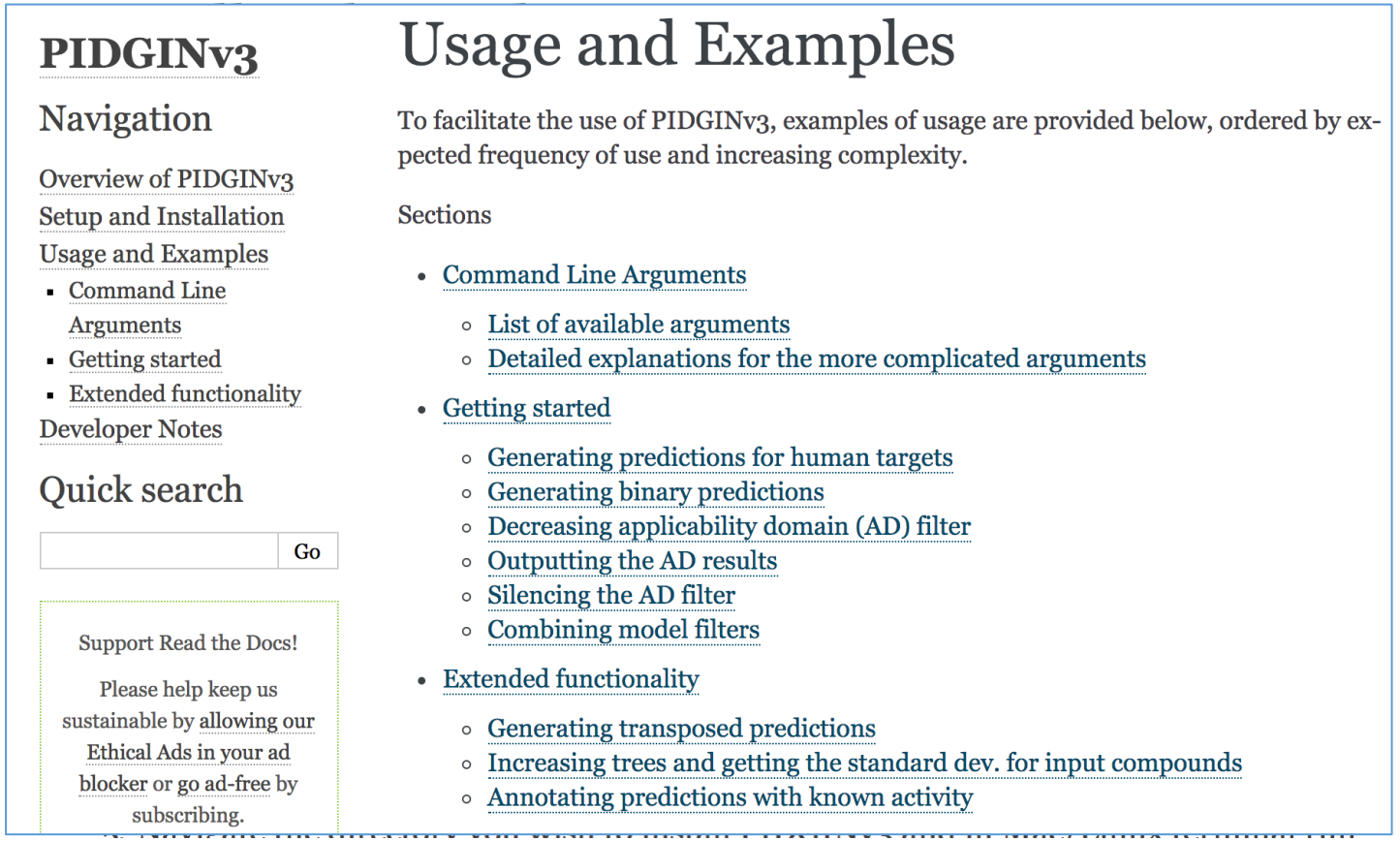
- Random Forests (RFs) trained on bioactivity data from PubChem (extracted 07/06/18) and ChEMBL (version 24)
- Employ a modification of the reliability-density neighbourhood Applicability Domain (AD) analysis
- Semi-quantitative: Models generated both with and without mapping to orthologues and at cut-offs of 100µM, 10 µM, 1 µM, and 0.1 µM

	Without orthologues	**With orthologues**
Distinct Models (various cut-offs)	10,446	14,678
Distinct Targets [exhaustive total]	7,075	16,623 [60,437]
Total Bioactivities Over all models	39,424,168	398,340,769
Actives	3,204,038	35,009,629
Inactives [of which are Sphere exclusion]	36,220,130 [27,435,133]	363,331,140 [248,782,698]

Downloading the tool

https://pidginv3.readthedocs.io/en/latest/

- Use Readthedocs
- Option 1: Navigate to https://github.com/lhm30/PIDGINv3 and and click on “Clone or Download” and then “Download Zip” to download PIDGIN
- Option 2: If using git from command line run “git clone https://www.github.com/lhm30/PIDGINv3”



PIDGINv3

Navigation

Overview of PIDGINv3
Setup and Installation
Usage and Examples
- Command Line Arguments
- Getting started
- Extended functionality

Developer Notes

Quick search

Go

Support Read the Docs!

Please help keep us sustainable by allowing our Ethical Ads in your ad blocker or go ad-free by subscribing.

Usage and Examples

To facilitate the use of PIDGINv3, examples of usage are provided below, ordered by expected frequency of use and increasing complexity.

Sections

- Command Line Arguments
  - List of available arguments
  - Detailed explanations for the more complicated arguments
- Getting started
  - Generating predictions for human targets
  - Generating binary predictions
  - Decreasing applicability domain (AD) filter
  - Outputting the AD results
  - Silencing the AD filter
  - Combining model filters
- Extended functionality
  - Generating transposed predictions
  - Increasing trees and getting the standard dev. for input compounds
  - Annotating predictions with known activity

Validating PIDGINv3

- Time series split cross validation (TSSCV)

```
def do_tsscv(actives,inactives,ntree,output_name):
    #split actives and inactives seperately
    tscv = TimeSeriesSplit(n_splits=5)
    a_split = list(tscv.split(actives))
    i_split = list(tscv.split(inactives))
    bedroc, rocs, prauc = [], [], []
    #build rf on test,train and calculate pr-auc & bedroc
    for spl in zip(a_split,i_split):
        a_train, a_test = spl[0][0], spl[0][1]
```

- Leave 50% of scaffolds out cross validation (L50SO)

Function Details

MakeScaffoldGeneric(mol)

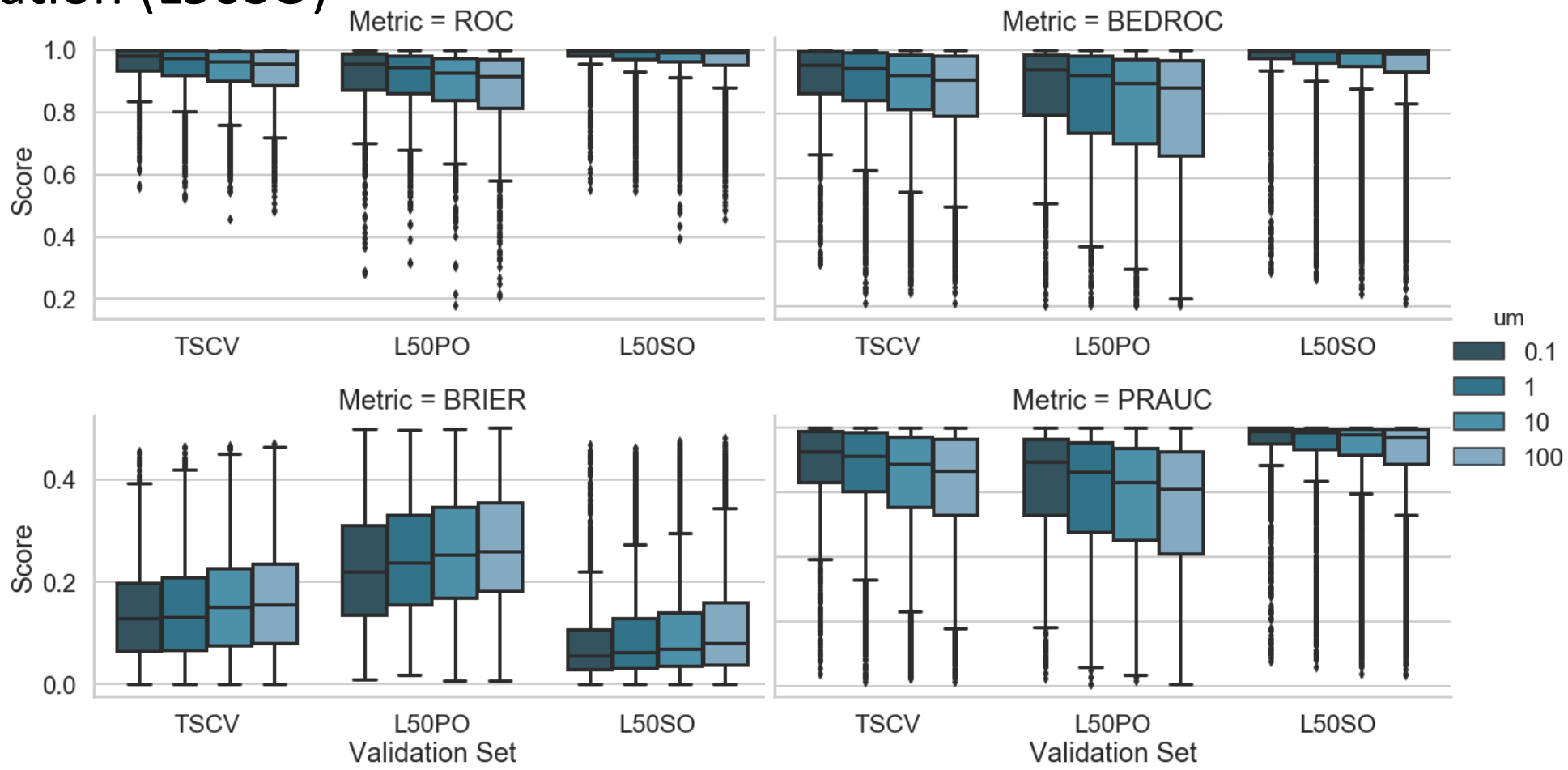
Makes a Murcko scaffold generic (i.e. all

```
def do_group_splitting(actives,groups,inactives,ntree,train_size=0.5):
    #check 4 groups (enough across splits)
    if not len(set(groups)) >= 4: return None
    gss = GroupShuffleSplit(n_splits=4, train_size=train_size, random_state=123)
    #check can split actives
    try: a_split = list(gss.split(actives, range(len(actives)), groups=groups))
    except ValueError: return None
    #check at least 5 comps acrross splits
    if all([len(splt[0])>=5 for splt in a_split]) == False: return err
    inact_ss = ShuffleSplit(n_splits=4, train_size=train_size,random_state=123)
    i_split = list(inact_ss.split(inactives))
    bedroc, rocs, prauc = [], [], []
    for spl in zip(a_split,i_split):
        a_train, a_test = spl[0][0], spl[0][1]
        i_train, i_test = spl[1][0], spl[1][1]
        x = np.vstack((actives[a_train],inactives[i_train]))
        y = [1] * len(a_train) + [0] * len(i_train)
        test_x = np.vstack((actives[a_test],inactives[i_test]))
        test_y = [1] * len(a_test) + [0] * len(i_test)
        class_weights = class_weight.compute_class_weight('balanced',np.unique(y),y)
        sw = np.array([class_weights[1] if i == 1 else class_weights[0] for i in y])
```

- Leave 50% of the ChEMBL publications out cross validation (L50PO)

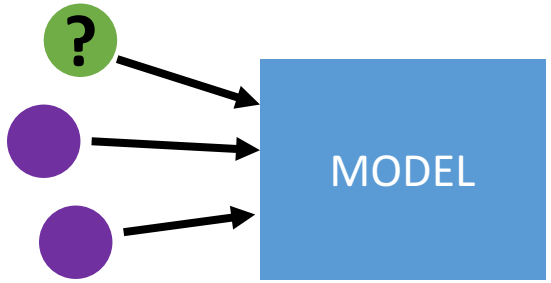
PIDGINv3 validation

- Time series split cross validation (TSCV)
- Leave 50% of scaffolds out cross validation (L50SO)
- Leave 50% of the ChEMBL publications out cross validation (L50PO)



Applicability Domain (AD) analysis

- The region of chemical space in which a model can be **reliably used to make predictions**.
- Defining this is essential to ensure the **modelled SAR remains valid**



- How different is *too* different? How to even define difference/similarity?
- **Usefulness is Subjective**: depends on the desired purpose
  - higher **precision** – i.e. lower numbers of inactive compounds classified as actives (false positives)
  - higher **recall** – i.e. retrieving the maximum number of active compounds (true positives)

Definition of the AD

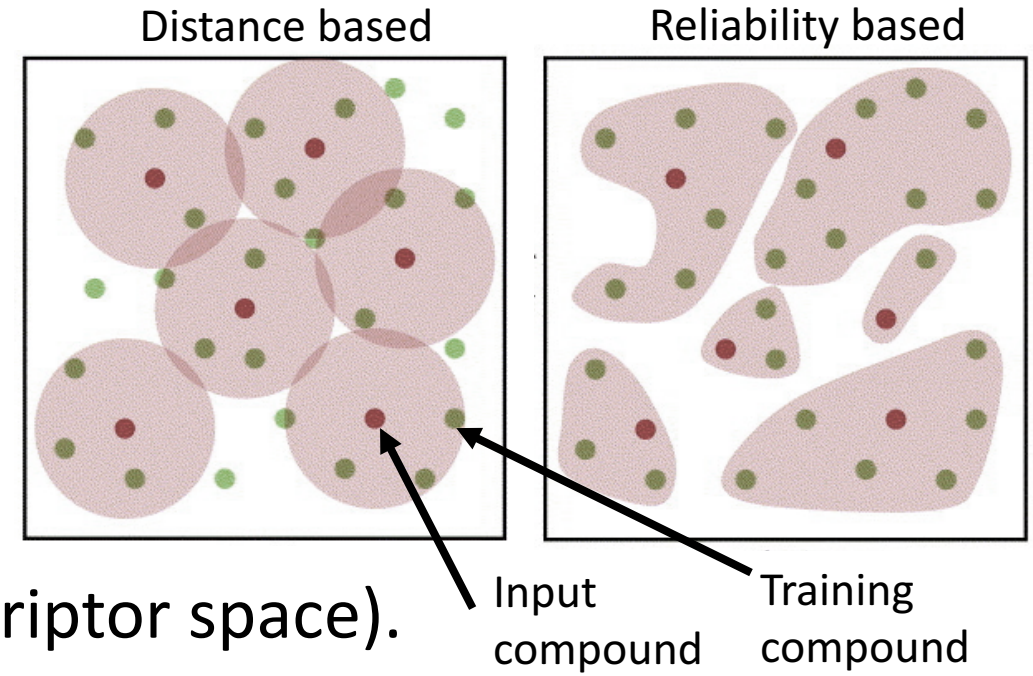
- A typical approach: distance (or “degree of extrapolation”) defined by similarity to the training set structures or training set features

Usually not a good correlation with error rate (!)

- AD is context-specific:
  - Near neighbour (NN) similarity = 50% has different meanings when applied to a model from a chemical series versus a diverse set of compounds

- QSAR model predictions typically depend on the **local characteristics**

- Misprediction can be largely boiled down to **insufficient data** (Density) or **poor input** (Experimental error, ambiguous descriptor space).

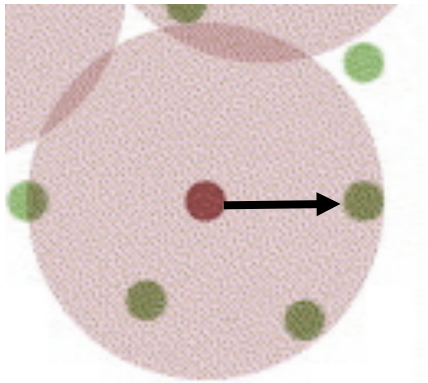


Reliability-density neighbourhood analysis

Training space: Local data **density**, local **bias** and local **precision**

SCENARIO #1

NN Sim = 0.6



BIAS = 10%
SD = 10%

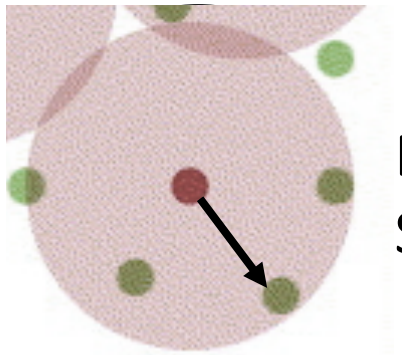
Sim* = 0.6/(0.1*0.1)
= **60**

Sim*=NN Sim/(BIAS * SD)

relative coverage adjusted to consider Reliability

SCENARIO #2

NN Sim = 0.6



BIAS = 80%
SD = 20%

Sim* = 0.6/(0.8*0.2)
= **3.75**

Both Covered?

BIAS Calculation:

```
1 - clf.predict_proba(mol)[:,true_label]
```

SD Calculation:

```
>>> sd = []
>>> for tree in range(len(clf.estimators_)):
>>>     tree_p = \
        clf.estimators_[tree].predict_proba(mol)[:,1])
>>>     sd.append(tree_p)
>>> np.std(sd,axis=0)
```

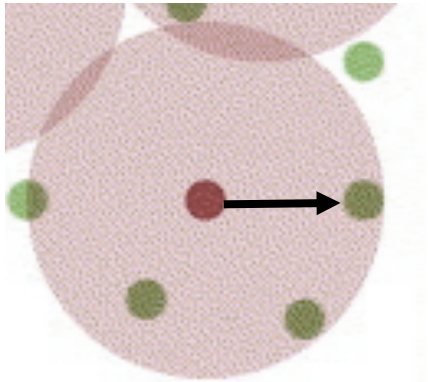
Aniceto, N *et al.* ***A novel applicability domain technique for mapping predictive reliability across the chemical space of a QSAR: reliability-density neighbourhood***. *Journal of cheminformatics* 8.1 (2016): 69

Reliability-density neighbourhood analysis

Training space: Local data **density**, local **bias** and local **precision**

SCENARIO #1

NN Sim = 0.6



BIAS = 10%
SD = 10%

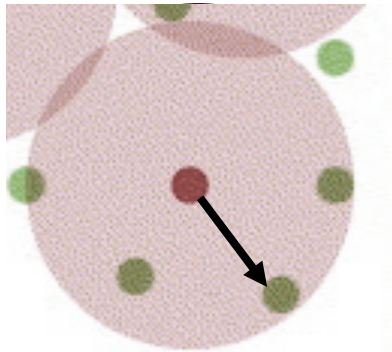
Sim* = 0.6/(0.1*0.1)
= **60**

Sim*= NN Sim / (BIAS * SD)

relative coverage adjusted to consider Reliability

SCENARIO #2

NN Sim = 0.6



BIAS = 80%
SD = 20%

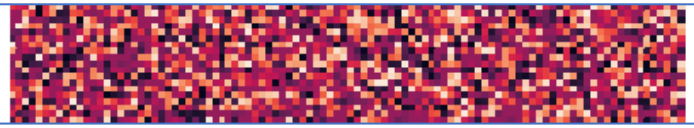
Sim* = 0.6/(0.8*0.2)
= **3.75**

Both Covered?

We now need to set **a new threshold for this scale,** which is dependent on the **training distribution**

1. Calculate all **Sim*** within Training

```
for idx, c in enumerate(c_array):
    BulkTanimotoSimilarity(c,c_array[:idx]+
     c_array[idx+1:])
```

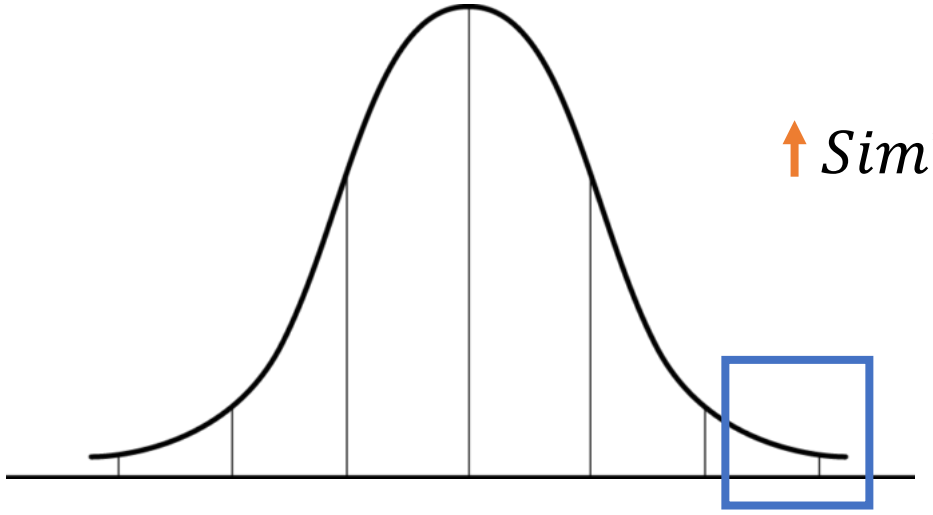


rdMetricMatrixCalc.GetTanimotoDistMat(c_array)

RETURNS: A numeric 1 dimensional array containing the lower triangle elements of the symmetric distance matrix

Neighbourhood analysis calculation

2. Get percentiles of training-training Sim* values



$$\uparrow Sim^* = \frac{Sim \uparrow}{BIAS \times SD \downarrow}$$

Highest similarity + Highest reliability

3. For a NEW compound & Sim*$_{lim}$:

- for each training compound:
  - a) calculate test-train Sim*
  - b) If Sim* > Sim*$_{lim}$ → **Compound is covered**

(!) Caveat: No distinction between having multiple neighbours and just one (control computational cost)

Testing PIDGIN's AD functionality

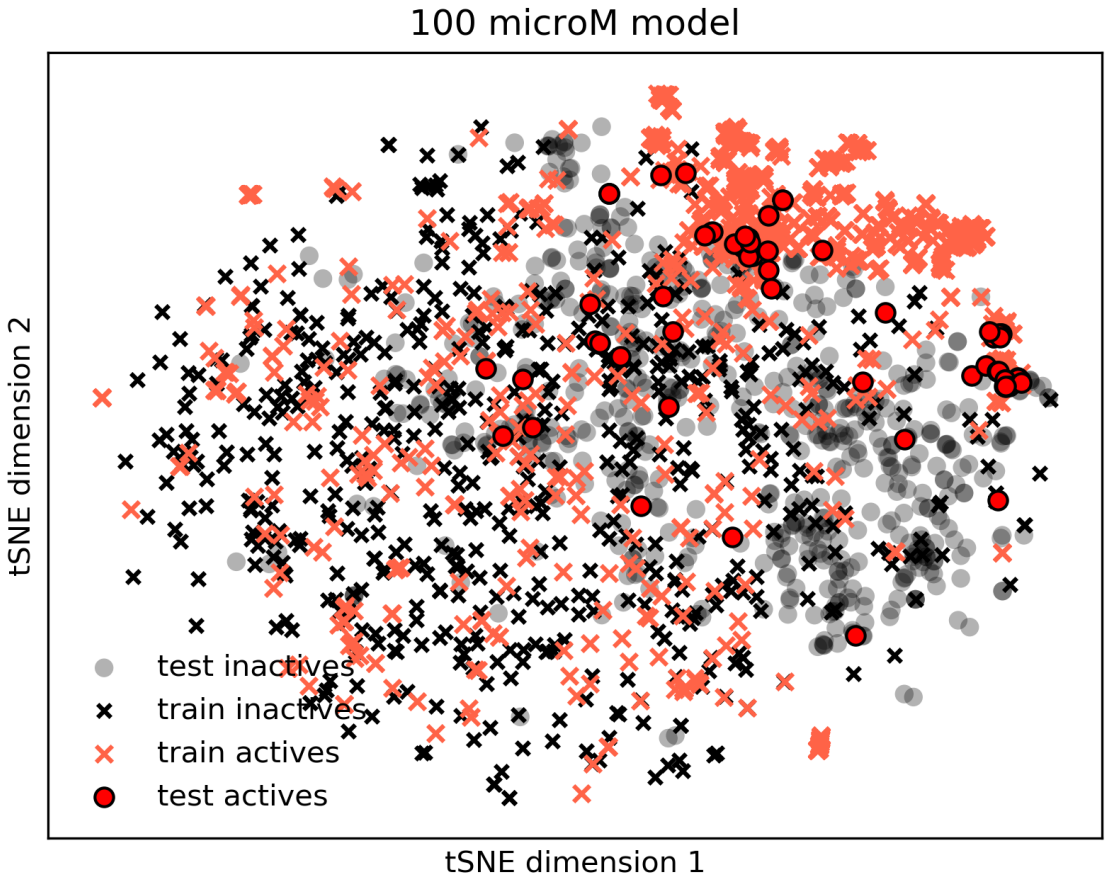
- 8 PubChem cell-based assays deposited in the last year
- reporter target bioactivity
- Removal of overlap with training

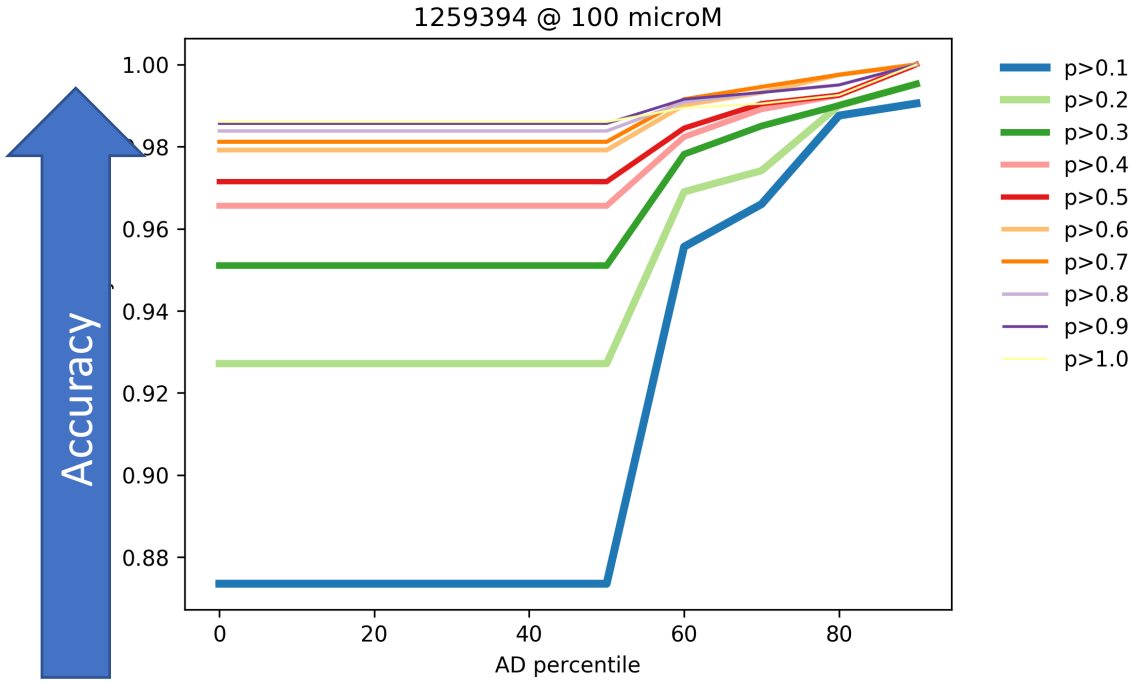
- AD percentile = [10 - 90]
- Classifier probability threshold = [0 - 1]

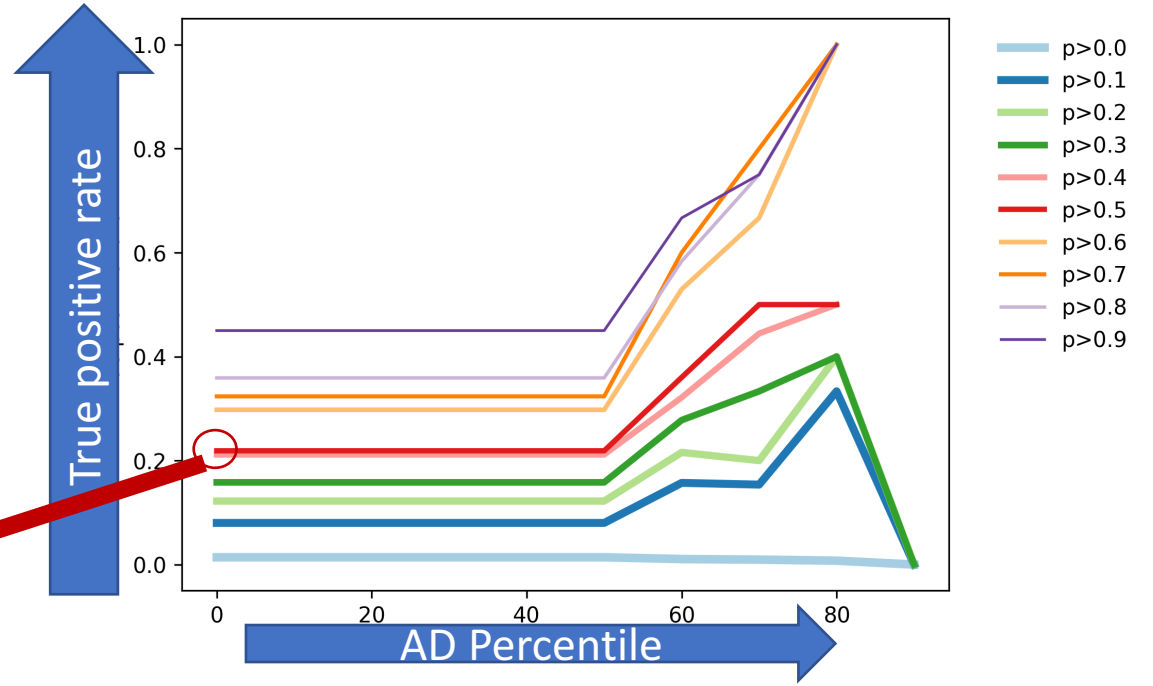
	0.1	1	10	100
#actives	2-9	4-32	10-115	13-490
#inactives		3k – 6k		

Hypothesis to be tested: ↑ AD should correlate with OVERALL **↓ misprediction rate** (no class-specific guarantee)

Estrogen Receptor 2

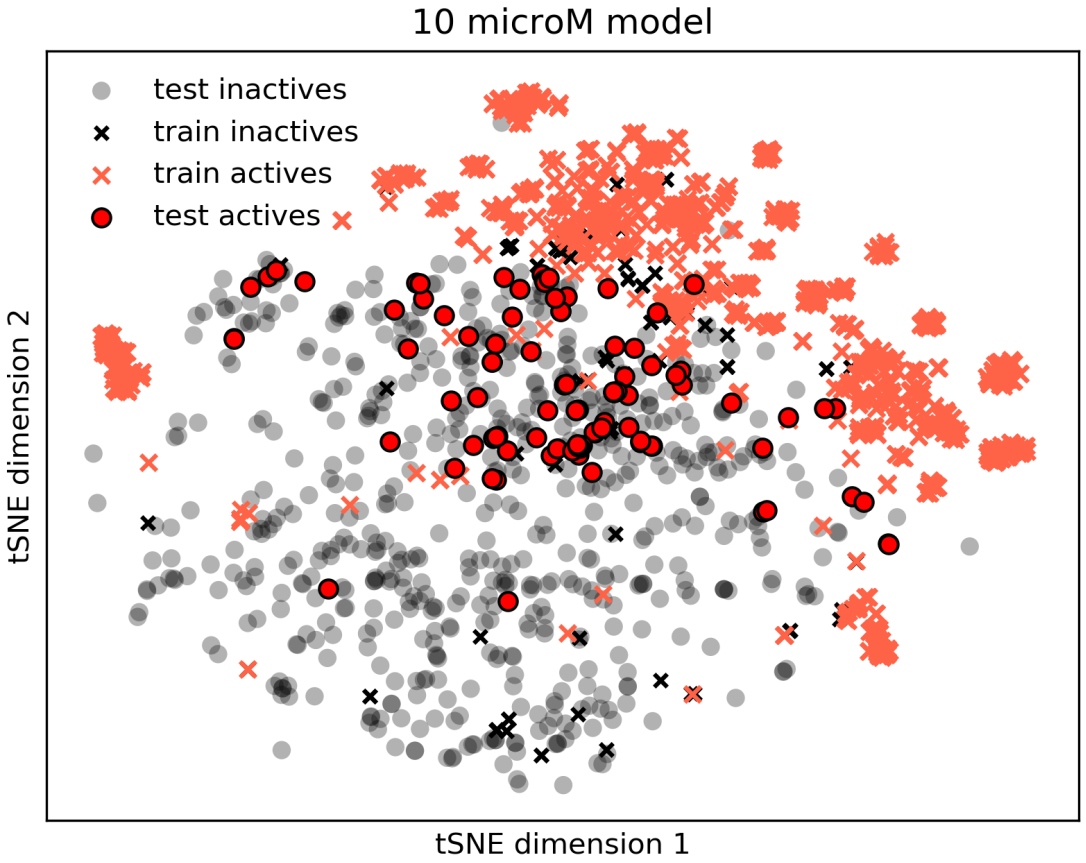


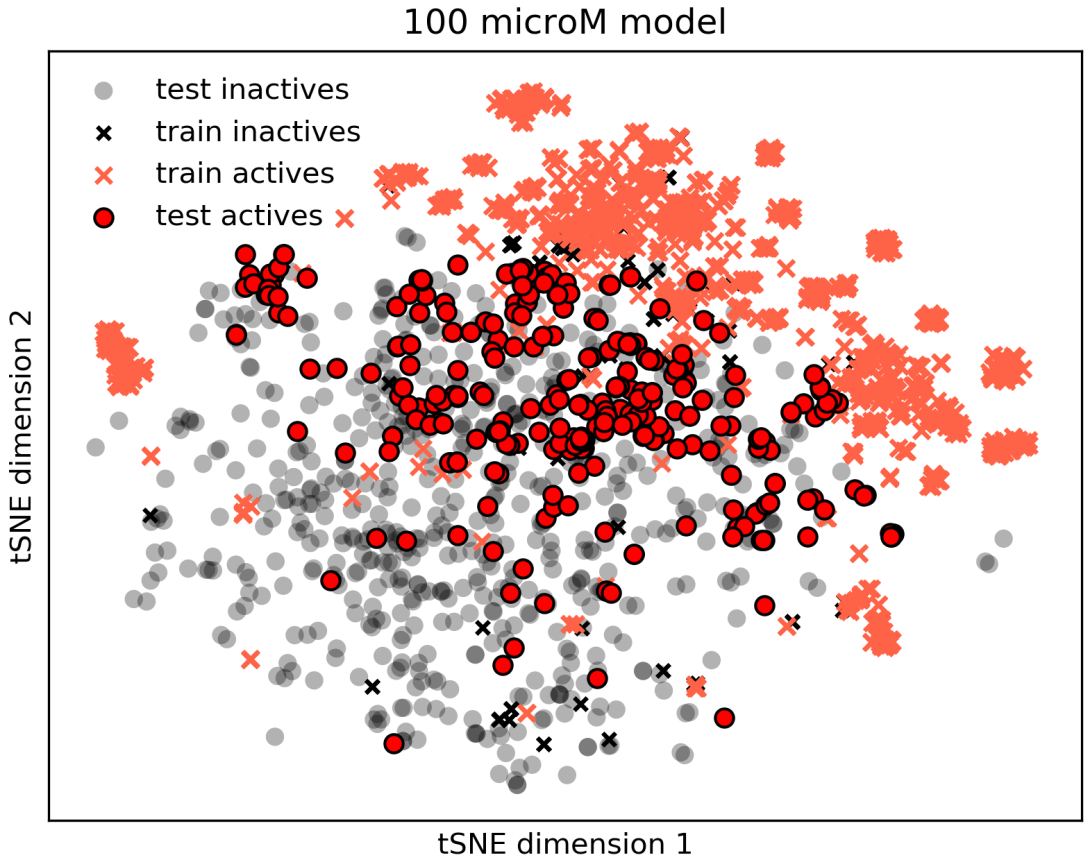


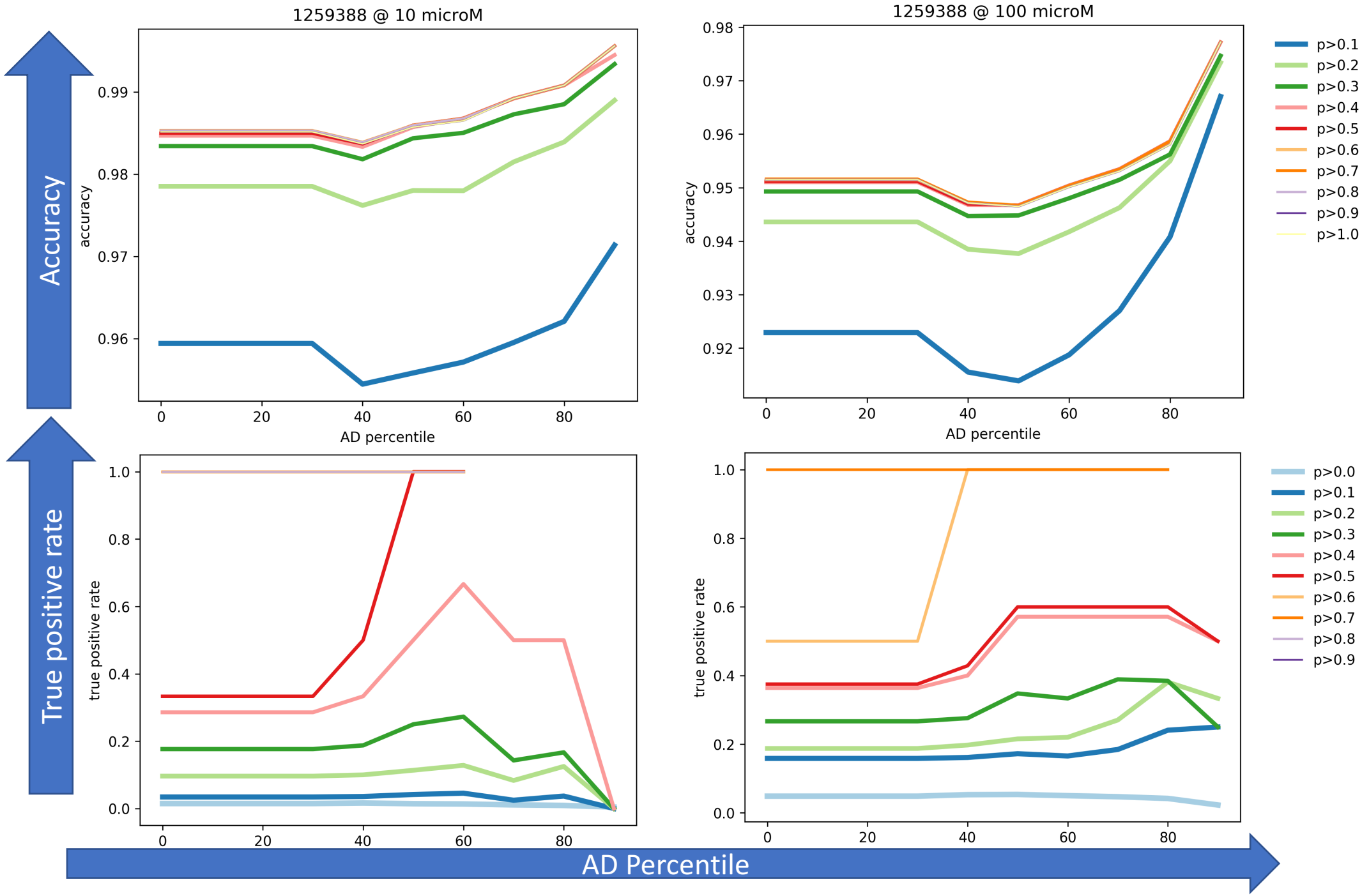


Without AD filter actives are accepted at **~80%** error rate → with AD: **up to 50%**

Histone Deacetylase 9 isoform 3





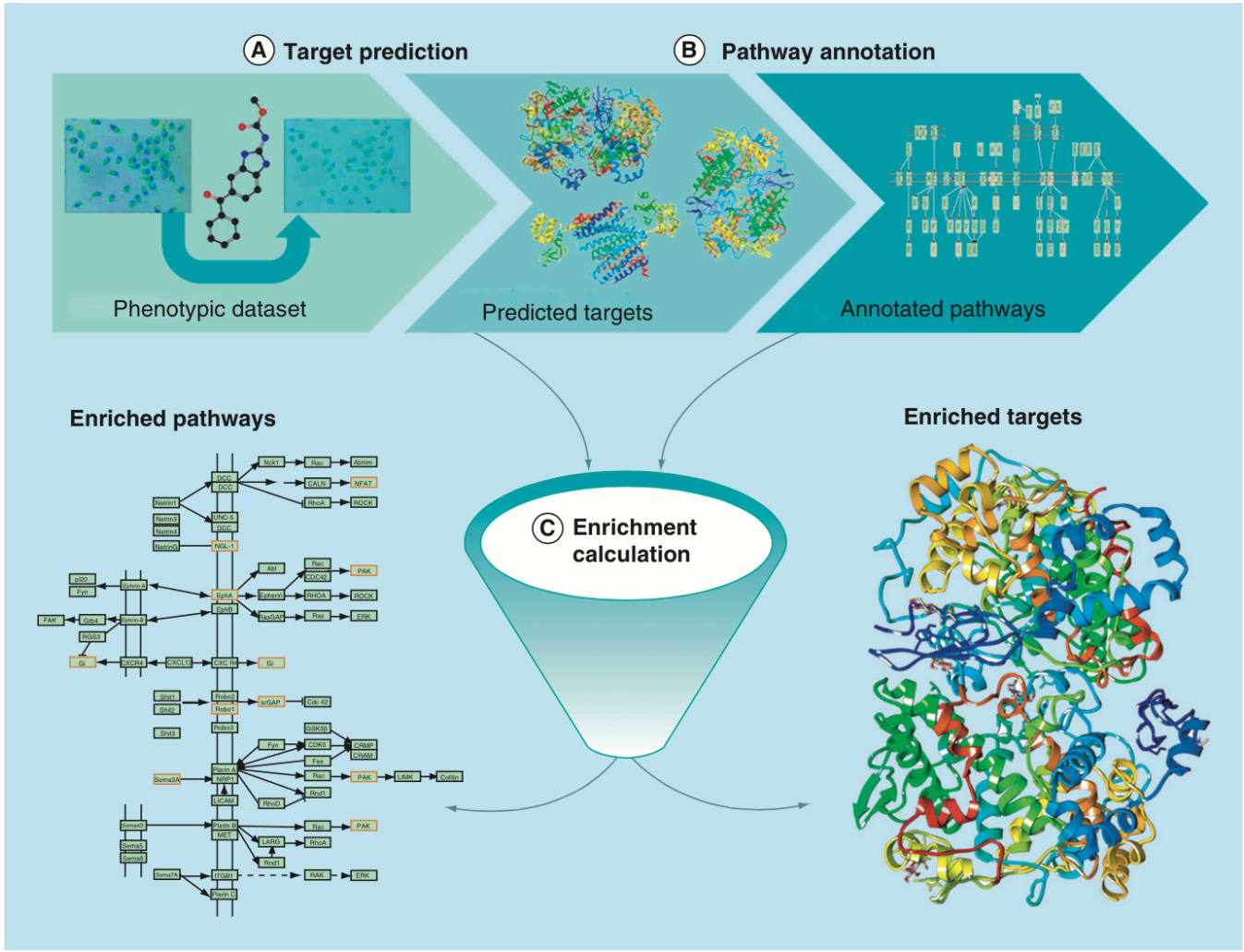

1259388 @ 10 microM
1259388 @ 100 microM
Accuracy
True positive rate
AD Percentile
accuracy
true positive rate
AD percentile
p>0.0
p>0.1
p>0.2
p>0.3
p>0.4
p>0.5
p>0.6
p>0.7
p>0.8
p>0.9
p>1.0

Citations/Prospective validation of PIDGIN

- A Nano-MgO and Ionic Liquid-Catalyzed 'Green' Synthesis Protocol for the Development of Adamantyl-Imidazolo-Thiadiazoles as Anti-Tuberculosis Agents Targeting Sterol 14α-Demethylase (CYP51)S Anusha, B Cp, *et al*. *PLoS One* (2015)
- Trisubstituted-Imidazoles Induce Apoptosis in Human Breast Cancer Cells by Targeting the Oncogenic PI3K/Akt/mTOR Signaling Pathway. CD Mohan, *et al*. *PLoS One* (2016)
- Novel Adamantanyl-Based Thiadiazolyl Pyrazoles Targeting EGFR in Triple-Negative Breast Cancer. A Sebastian, *et al*. *ACS Omega* (2016)
- Towards the mode of action of *Strobilanthes crispus* through integrated computational and experimental analyses. KK Wong, *et al*. *Journal of Plant Biochemistry and Biotechnology* (2017)
- Synthesis and evaluation of anticancer and trypanocidal activities of boronic tyrphostins. Martins, Daniela*, et al. ChemMedChem* (2018).
- A Machine Learning Approach for Predicting HIV Reverse Transcriptase Mutation Susceptibility of Biologically Active Compounds. Kaiser, Thomas M., *et al. Journal of chemical information and modeling* (2018).
- Modelling Cellular Permeability via Carrier Mediated Transport (Masters Thesis). TI Sarupinda. (2016) Available at http://uhra.herts.ac.uk/handle/2299/17256
- Clavis Aurea? Structure-enabled approaches of identifying and optimizing GPCR ligands (Masters Thesis). EB Lenselink (2017) Available at http://hdl.handle.net/1887/49402
- The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology (PhD Thesis). A Kadurin (2017) Available at http://dspace.kpfu.ru/xmlui/handle/net/113796
- A Biofocussed Chemoprospecting Approach to Drug Discovery: Design, Synthesis and Bioactivity Screening of Diverse Biofocussed Chemical Libraries (PhD Thesis). BS Thakkar (2017) Available at http://hdl.handle.net/10037/11155

Extending in silico mechanism-of-action analysis by annotating targets with **pathways**

- Annotate predicted targets with pathways from NCBI BioSystems database.
- Enrichment is calculated after comparison of phenotypic dataset or with a background of compounds



Liggi, Sonia, *et al.* **"Extending in silico mechanism-of-action analysis by annotating targets with pathways: application to cellular cytotoxicity readouts."** *Future medicinal chemistry* 6.18 (2014): 2029-2056.

BioSystems

KEGG Kyoto Encyclopedia of Genes and Genomes

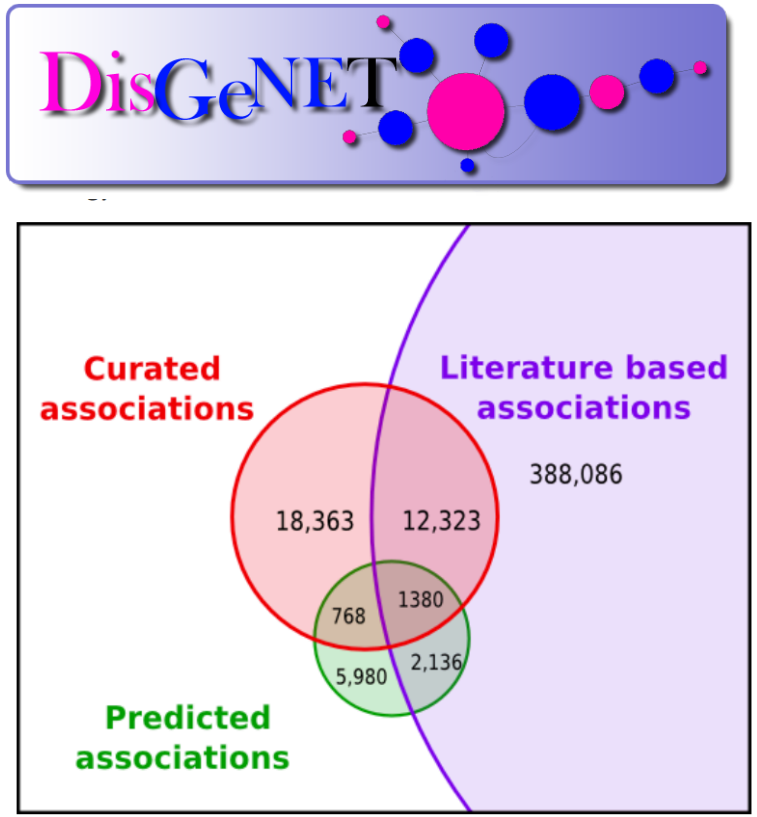
BETA WikiPathways *Pathways for the People*

BioCyc Database Collection

GO the Gene Ontology

Extending in silico mechanism-of-action analysis by annotating targets with **diseases**

- Annotate predicted targets with diseases from DisGeNET database
- Enrichment is calculated after comparison of phenotypic dataset or with a background of compounds
- A DisGeNET score >= 0.06 includes associations of curated sources/animal models supporting them or reported in 20-200 papers



DisGeNET Score: Confidence in a gene-disease association (GDA)

Piñero, Janet, et al. "DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants." Nucleic acids research (2016)

How to use PIDGIN

- SDF/Smiles support
- Provides pre-processing
- Change # cores
- Filter for organism/target class /cross validation performance
- Dynamically alter # trees
- Flag for known bioactivity within input
- Easy to alter AD filter

```
Options:
  -h, --help            show this help message and exit
  -f FILE               Input smiles or sdf file (required)
  -d DELIM, --smiles_delim=DELIM
                                                Input file (smiles) delimiter char (default: white
                                                space ' ')
  --smiles_column=SMICOL
                                                Input file (smiles) delimiter column (default: 0)
  --smiles_id_column=IDCOL
                                                Input file (smiles) ID column (default: 1)
  -o FILE               Optional output prediction file name
  -t, --transpose       Transpose output (rows are compounds, columns are
                                                targets)
  -n NCORES, --ncores=NCORES
                                                No. cores (default: 1)
  -b BIOACTIVITY, --bioactivity=BIOACTIVITY
                                                Bioactivity threshold (can use multiple split by ','.
                                                E.g. '100,10'
  -p PROBA, --proba=PROBA
                                                RF probability threshold (default: None)
  --ad=AD               Applicability Domain (AD) filter using percentile of
                                                weights [float]. Default: 90 (integer for percentile)
  --known_flag          Set known activities (annotate duplicates betweem
                                                input to train with correct label)
  --orthologues         Set to use orthologue bioactivity data in model
                                                generation
  --organism=ORGANISM   Organism filter (multiple can be specified using
                                                commas ',')
  --target_class=TARGETCLASS
                                                Target classification filter
  --min_size=MINSIZE    Minimum number of actives used in model generation
                                                (default: 10)
  --performance_filter=P_FILT
                                                Comma-seperated performance filtering using following
                                                nomenclature: validation_set[tsscv,l50so,l50po],metric
                                                [bedroc,roc,prauc,brier],performance_threshold[float].
                                                E.g 'tsscv,bedroc,0.5'
  --se_filter           Optional setting to restrict to models which do not
                                                require Sphere Exclusion (SE)
  --training_log        Optional setting to add training_details to the
                                                prediction file (large increase in output file size)
  --ntrees=NTREES       Specify the minimum number of trees for warm-start
                                                random forest models (N.B Potential large
                                                latency/memory cost)
  --preprocess_off      Turn off preprocessing using the flatkinson (eTox)
                                                standardizer (github.com/flatkinson/standardiser),
                                                size filter (100 >= Mw >= 1000 and organic mol check
                                                (C count >= 1)
  --std_dev             Turn on matrix calculation for the standard deviation
                                                of prediction across the trees in the forest
  --percentile          Turn on matrix calculation for the percentile of AD
```

Acknowledgements

- Marianna Trapotsi
- Anaïs Mokdad
- Chad Allen
- Avid Azfal
- Andreas Bender
- Bender group
- Ola Engkvist
- Mike Firth
- Ian Barrett
- Krishna Bulusu
- AstraZeneca group

Estrogen Nuclear Receptor Alpha