A comparison of nine machine learning models based on an expanded mutagenicity dataset and their application for predicting pyrrolizidine alkaloid mutagenicity

Christoph Helma[*1], Verena Schöning[4], Philipp Boss[3], and Jürgen Drewe[2]

[1]in silico toxicology gmbh, Rastatterstrasse 41, 4057 Basel, Switzerland

[2]Zeller AG, Seeblickstrasse 4, 8590 Romanshorn, Switzerland

[3]Berlin Institute for Medical Systems Biology, Max Delbrück Center for Molecular Medicine in the Helmholtz Association, Robert-Rössle-Strasse 10, Berlin, 13125, Germany

[4]Clinical Pharmacology and Toxicology, Department of General Internal Medicine, Bern University Hospital, University of Bern, Inselspital, 3010 Bern, Switzerland

[*] Correspondence: Christoph Helma <helma@in-silico.ch>

Random forest, support vector machine, logistic regression, neural networks and k-nearest neighbor (`lazar`) algorithms, were applied to new *Salmonella* mutagenicity dataset with 8309 unique chemical structures. The best prediction accuracies in 10-fold-crossvalidation were obtained with `lazar` models and MolPrint2D descriptors, that gave accuracies (84%) similar to the interlaboratory variability of the Ames test.

TODO: PA results

Introduction

TODO: rationale for investigation

The main objectives of this study were

- to generate a new mutagenicity training dataset, by combining the most comprehensive public datasets
- to compare the performance of MolPrint2D (*MP2D*) fingerprints with PaDEL descriptors
- to compare the performance of global QSAR models (random forests (*RF*), support vector machines (*SVM*), logistic regression (*LR*), neural nets (*NN*)) with local models (`lazar`)
- to apply these models for the prediction of pyrrolizidine alkaloid mutagenicity

Materials and Methods

Data

Mutagenicity training data

An identical training dataset was used for all models. The training dataset was compiled from the following sources:

- Kazius/Bursi Dataset (4337 compounds, Kazius, McGuire, and Bursi (2005)): http://cheminformatics.org/datasets/bursi/cas_4337.zip
- Hansen Dataset (6513 compounds, Hansen et al. (2009)): http://doc.ml.tu-berlin.de/toxbenchmark/Mutagenicity_N6512.csv
- EFSA Dataset (695 compounds EFSA (2016)): https://data.europa.eu/euodp/data/storage/f/2017-0719T142131/GENOTOX%20data%20and%20dictionary.xls

Mutagenicity classifications from Kazius and Hansen datasets were used without further processing. To achieve consistency with these datasets, EFSA compounds were classified as mutagenic, if at least one positive result was found for TA98 or T100 Salmonella

strains.

Dataset merges were based on unique SMILES (*Simplified Molecular Input Line Entry Specification*) strings of the compound structures. Duplicated experimental data with the same outcome was merged into a single value, because it is likely that it originated from the same experiment. Contradictory results were kept as multiple measurements in the database. The combined training dataset contains 8309 unique structures.

Source code for all data download, extraction and merge operations is publicly available from the git repository https://git.in-silico.ch/mutagenicity-paper under a GPL3 License. The new combined dataset can be found at https://git.in-silico.ch/mutagenicity-paper/data/mutagenicity.csv.

Pyrrolizidine alkaloid (PA) dataset

The testing dataset consisted of 602 different PAs.

The PA dataset was created from five independent, necine base substructure searches in PubChem (https://pubchem.ncbi.nlm.nih.gov/) and compared to the PAs listed in the EFSA publication EFSA (2011) and the book by Mattocks Mattocks (1986), to ensure, that all major PAs were included. PAs mentioned in these publications which were not found in the downloaded substances were searched individually in PubChem and, if available, downloaded separately. Non-PA substances, duplicates, and isomers were removed from the files, but artificial PAs, even if unlikely to occur in nature, were kept. The resulting PA dataset comprised a total of 602 different PAs.

The PAs in the dataset were classified according to structural features. A total of 9 different structural features were assigned to the necine base, modifications of the necine base and to the necic acid:

For the necine base, the following structural features were chosen:

- Retronecine-type (1,2-unstaturated necine base)
- Otonecine-type (1,2-unstaturated necine base)
- Platynecine-type (1,2-saturated necine base)

For the modifications of the necine base, the following structural features were chosen:

- N-oxide-type
- Tertiary-type (PAs which were neither from the N-oxide- nor DHP-type)
- DHP-type (pyrrolic ester)

For the necic acid, the following structural features were chosen:

- Monoester-type
- Open-ring diester-type
- Macrocyclic diester-type

The compilation of the PA dataset is described in detail in Schöning et al. (2017).

Descriptors

MolPrint2D (*MP2D*) fingerprints

MolPrint2D fingerprints (O'Boyle et al. (2011)) use atom environments as molecular representation. They determine for each atom in a molecule, the atom types of its connected atoms to represent their chemical environment. This resembles basically the chemical concept of functional groups.

In contrast to predefined lists of fragments (e.g. FP3, FP4 or MACCs fingerprints) or descriptors (e.g PaDEL) they are generated dynamically from chemical structures. This has the advantage that they can capture substructures of toxicological relevance that are not included in other descriptors.

Chemical similarities (e.g. Tanimoto indices) can be calculated very efficiently with Mol-

Print2D fingerprints. Using them as descriptors for global models leads however to huge, sparsely populated matrices that cannot be handled with traditional machine learning algorithms. In our experiments none of the R and Tensorflow algorithms was capable to use them as descriptors.

MolPrint2D fingerprints were calculated with the OpenBabel cheminformatics library (O'Boyle et al. (2011)).

PaDEL descriptors

Molecular 1D and 2D descriptors were calculated with the PaDEL-Descriptors program (http://www.yapcwsoft.com version 2.21, Yap (2011)).

As the training dataset contained over 8309 instances, it was decided to delete instances with missing values during data pre-processing. Furthermore, substances with equivocal outcome were removed. The final training dataset contained 8080 instances with known mutagenic potential.

During feature selection, descriptors with near zero variance were removed using '*NearZeroVar*'-function (package 'caret'). If the percentage of the most common value was more than 90% or when the frequency ratio of the most common value to the second most common value was greater than 95:5 (e.g. 95 instances of the most common value and only 5 or less instances of the second most common value), a descriptor was classified as having a near zero variance. After that, highly correlated descriptors were removed using the '*findCorrelation*'-function (package 'caret') with a cut-off of 0.9. This resulted in a training dataset with 516 descriptors. These descriptors were scaled to be in the range between 0 and 1 using the '*preProcess*'-function (package 'caret'). The scaling routine was saved in order to apply the same scaling on the testing dataset. As these three steps did not consider the dependent variable (experimental mutagenicity), it was decided that they do not need to be included in the cross-validation of the model. To

further reduce the number of features, a LASSO (*least absolute shrinkage and selection operator*) regression was performed using the '*glmnet*'-function (package '*glmnet*'). The reduced dataset was used for the generation of the pre-trained models.

PaDEL descriptors were used in global (RF, SVM, LR, NN) and local (`lazar`) models.

Algorithms

`lazar`

`lazar` (*lazy structure activity relationships*) is a modular framework for read-across model development and validation. It follows the following basic workflow: For a given chemical structure `lazar`:

- searches in a database for similar structures (neighbours) with experimental data,
- builds a local QSAR model with these neighbours and
- uses this model to predict the unknown activity of the query compound.

This procedure resembles an automated version of read across predictions in toxicology, in machine learning terms it would be classified as a k-nearest-neighbour algorithm.

Apart from this basic workflow, `lazar` is completely modular and allows the researcher to use arbitrary algorithms for similarity searches and local QSAR (*Quantitative structure–activity relationship*) modelling. Algorithms used within this study are described in the following sections.

Neighbour identification

Utilizing this modularity, similarity calculations were based both on MolPrint2D fingerprints and on PaDEL descriptors.

For MolPrint2D fingerprints chemical similarity between two compounds a and b is

expressed as the proportion between atom environments common in both structures $A \cap B$ and the total number of atom environments $A \cup B$ (Jaccard/Tanimoto index).

$$sim = \frac{|A \ \cap B|}{|A \ \cup B|}$$

For PaDEL descriptors chemical similarity between two compounds a and b is expressed as the cosine similarity between the descriptor vectors A for a and B for b.

$$sim = \frac{A \cdot B}{|A||B|}$$

Threshold selection is a trade-off between prediction accuracy (high threshold) and the number of predictable compounds (low threshold). As it is in many practical cases desirable to make predictions even in the absence of closely related neighbours, we follow a tiered approach:

- First a similarity threshold of 0.5 is used to collect neighbours, to create a local QSAR model and to make a prediction for the query compound. This are predictions with *high confidence*.
- If any of these steps fails, the procedure is repeated with a similarity threshold of 0.2 and the prediction is flagged with a warning that it might be out of the applicability domain of the training data (*low confidence*).
- Similarity thresholds of 0.5 and 0.2 are the default values chosen by the software developers and remained unchanged during the course of these experiments.

Compounds with the same structure as the query structure are automatically eliminated from neighbours to obtain unbiased predictions in the presence of duplicates.

Local QSAR models and predictions

Only similar compounds (neighbours) above the threshold are used for local QSAR models. In this investigation, we are using a weighted majority vote from the neighbour's experimental data for mutagenicity classifications. Probabilities for both classes (mutagenic/non-mutagenic) are calculated according to the following formula and the class with the higher probability is used as prediction outcome.

$$p_c = \frac{\sum \mathrm{sim}_{n,c}}{\sum \mathrm{sim}_n}$$

p_c Probability of class c (e.g. mutagenic or non-mutagenic)

$\sum \mathrm{sim}_{n,c}$ Sum of similarities of neighbours with class c

$\sum \mathrm{sim}_n$ Sum of all neighbours

Applicability domain

The applicability domain (AD) of `lazar` models is determined by the structural diversity of the training data. If no similar compounds are found in the training data no predictions will be generated. Warnings are issued if the similarity threshold had to be lowered from 0.5 to 0.2 in order to enable predictions. Predictions without warnings can be considered as close to the applicability domain (*high confidence*) and predictions with warnings as more distant from the applicability domain (*low confidence*). Quantitative applicability domain information can be obtained from the similarities of individual neighbours.

Availability

- `lazar` experiments for this manuscript: https://git.in-silico.ch/mutagenicity-paper (source code, GPL3)
- `lazar` framework: https://git.in-silico.ch/lazar (source code, GPL3)

- `lazar` GUI: https://git.in-silico.ch/lazar-gui (source code, GPL3)
- Public web interface: https://lazar.in-silico.ch

R Random Forest, Support Vector Machines, and Deep Learning

The RF, SVM, and DL models were generated using the R software (R-project for Statistical Computing, https://www.r-project.org/*;* version 3.3.1), specific R packages used are identified for each step in the description below.

Random Forest (*RF*)

For the RF model, the '*randomForest*'-function (package '*randomForest*') was used. A forest with 1000 trees with maximal terminal nodes of 200 was grown for the prediction.

Support Vector Machines (*SVM*)

The '*svm*'-function (package 'e1071') with a *radial basis function kernel* was used for the SVM model.

TODO: **Verena, Phillip** Sollen wir die DL Modelle ebenso wie die Tensorflow als Neural Nets (NN) bezeichnen?

Deep Learning

The DL model was generated using the '*h2o.deeplearning*'-function (package '*h2o*'). The DL contained four hidden layer with 70, 50, 50, and 10 neurons, respectively. Other hyperparameter were set as follows: l1=1.0E-7, l2=1.0E-11, epsilon = 1.0E-10, rho = 0.8, and quantile_alpha = 0.5. For all other hyperparameter, the default values were used. Weights and biases were in a first step determined with an unsupervised DL model. These values were then used for the actual, supervised DL model.

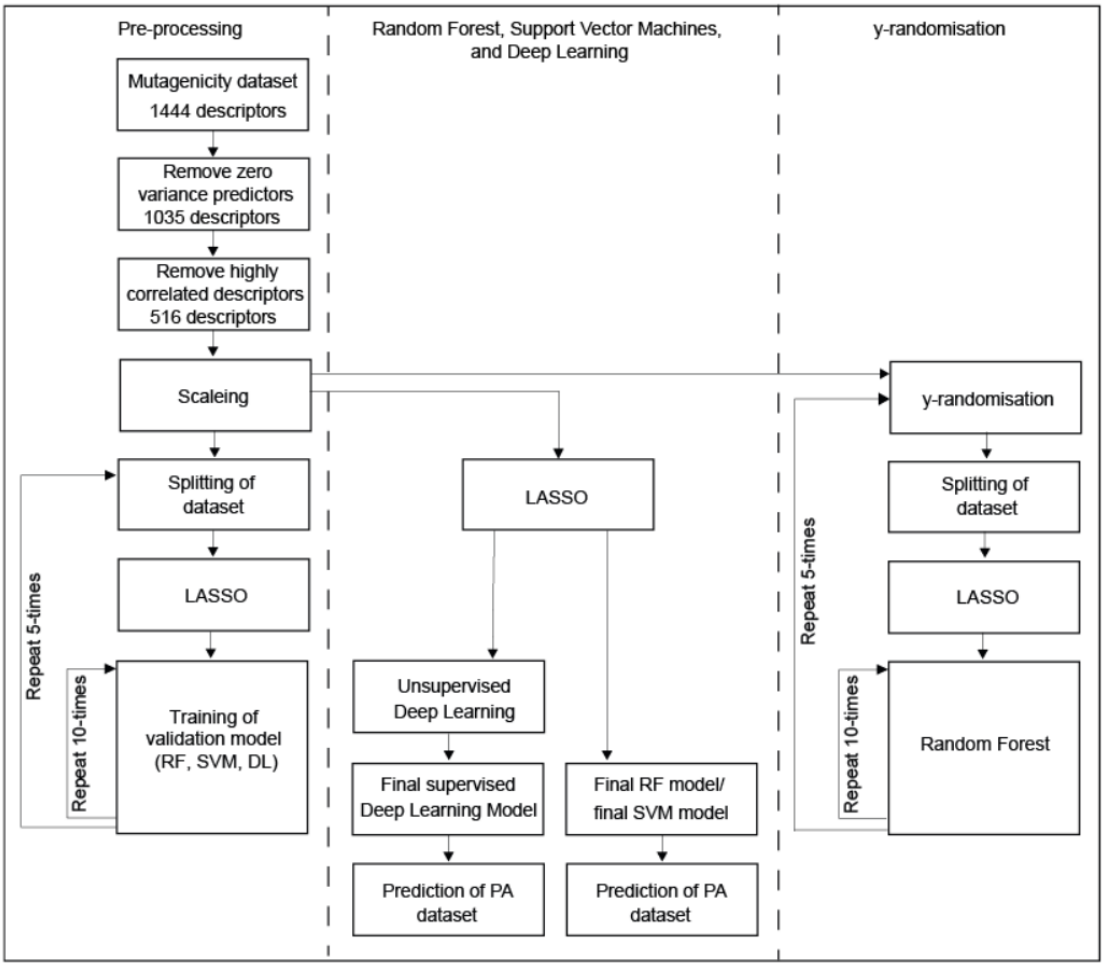


Figure 1: Flowchart of the generation and validation of the models generated in R-project

To validate these models, an internal cross-validation approach was chosen. The training dataset was randomly split in training data, which contained 95% of the data, and validation data, which contain 5% of the data. A feature selection with LASSO on the training data was performed, reducing the number of descriptors to approximately 100. This step was repeated five times. Based on each of the five different training data, the predictive models were trained and the performance tested with the validation data. This step was repeated 10 times.

Applicability domain

TODO: **Verena**: Mit welchen Deskriptoren hast Du den Jaccard index berechnet? Fuer den Jaccard index braucht man binaere Deskriptoren (zB MP2D), mit PaDEL Deskriptoren koennte man zB eine euklidische oder cosinus Distanz berechnen.

The AD of the training dataset and the PA dataset was evaluated using the Jaccard distance. A Jaccard distance of '0' indicates that the substances are similar, whereas a value of '1' shows that the substances are different. The Jaccard distance was below 0.2 for all PAs relative to the training dataset. Therefore, PA dataset is within the AD of the training dataset and the models can be used to predict the genotoxic potential of the PA dataset.

Availability

R scripts for these experiments can be found in https://git.in-silico.ch/mutagenicity-paper/scripts/R.

Tensorflow models

Data pre-processing was done by rank transformation using the '*QuantileTransformer*' procedure. A sequential model has been used. Four layers have been used: input layer, two hidden layers (with 12, 8 and 8 nodes, respectively) and one output layer. For the output layer, a sigmoidal activation function and for all other layers the ReLU ('*Rectified Linear Unit*') activation function was used. Additionally, a L^2-penalty of 0.001 was used for the input layer. For training of the model, the ADAM algorithm was used to minimise the cross-entropy loss using the default parameters of Keras. Training was performed for 100 epochs with a batch size of 64. The model was implemented with Python 3.6 and Keras.

TODO: **Philipp** Ich hab die alten Ergebnisse mit feature selection weggelassen, ist das ok? Dann muesste auch dieser Absatz gestrichen werden, oder?

TODO: **Philipp** Kannst Du bitte die folgenden Absaetze ergaenzen

Random forests (*RF*)

Logistic regression (SGD) (*LR-sgd*)

Logistic regression (scikit) (*LR-scikit*)

TODO: **Philipp, Verena** DL oder NN?

Neural Nets (*NN*)

Alternatively, a DL model was established with Python-based Tensorflow program (https://www.tensorflow.org/) using the high-level API Keras (https://www.tensorflow.org/guide/keras) to build the models.

Tensorflow models used the same PaDEL descriptors as the R models.

Validation

10-fold cross-validation was used for all Tensorflow models.

Availability

Jupyter notebooks for these experiments can be found in https://git.in-silico.ch/mutagenicity-paper/scripts/tensorflow.

Results

10-fold crossvalidations

Crossvalidation results are summarized in the following tables: Table 1 shows `lazar` results with MolPrint2D and PaDEL descriptors, Table 2 R results and Table 3 Tensorflow results.

Table 1: Summary of lazar crossvalidation results (all/high confidence predictions)

	MP2D	PaDEL
Accuracy	0.82/0.84	0.58/0.58
True positive rate/Sensitivity	0.85/0.89	0.32/0.32
True negative rate/Specificity	0.78/0.79	0.79/0.79
Positive predictive value/Precision	0.8/0.83	0.56/0.56
Negative predictive value	0.84/0.85	0.59/0.59
Nr. predictions	7781/5890	4089/4081

Table 2: Summary of R crossvalidation results

	RF	SVM	DL
Accuracy	0.64	0.61	0.56
True positive rate/Sensitivity	0.56	0.56	0.88
True negative rate/Specificity	0.71	0.67	0.24
Positive predictive value/Precision	0.66	0.62	0.53
Negative predictive value	0.62	0.61	0.67
Nr. predictions	8070	8070	8070

Table 3: Summary of tensorflow crossvalidation results

	RF	LR-sgd	LR-scikit	NN
Accuracy	0.64	0.62	0.63	0.63
True positive rate/Sensitivity	0.59	0.6	0.62	0.61
True negative rate/Specificity	0.7	0.65	0.63	0.64
Positive predictive value/Precision	0.66	0.63	0.62	0.63
Negative predictive value	0.63	0.62	0.63	0.63
Nr. predictions	8080	8080	8080	8080

Figure 2 depicts the position of all crossvalidation results in receiver operating characteristic (ROC) space.

Confusion matrices for all models are available from the git repository https://git.in-silico.ch/mutagenicity-paper/10-fold-crossvalidations/confusion-matrices/, individual predictions can be found in https://git.in-silico.ch/mutagenicity-paper/10-fold-crossvalidations/predictions/.

The most accurate crossvalidation predictions have been obtained with standard `lazar` models using MolPrint2D descriptors (0.84 for predictions with high confidence, 0.82 for all predictions). Models utilizing PaDEL descriptors have generally lower accuracies ranging from 0.56 (R deep learning) to 0.64 (R/Tensorflow random forests). Sensitivity and specificity is generally well balanced with the exception of `lazar`-PaDEL (low sensitivity) and R deep learning (low specificity) models.

Pyrrolizidine alkaloid mutagenicity predictions

Mutagenicity predictions from all investigated models for 602 pyrrolizidine alkaloids (PAs) are shown in Table 4. A CSV table with all predictions can be downloaded from

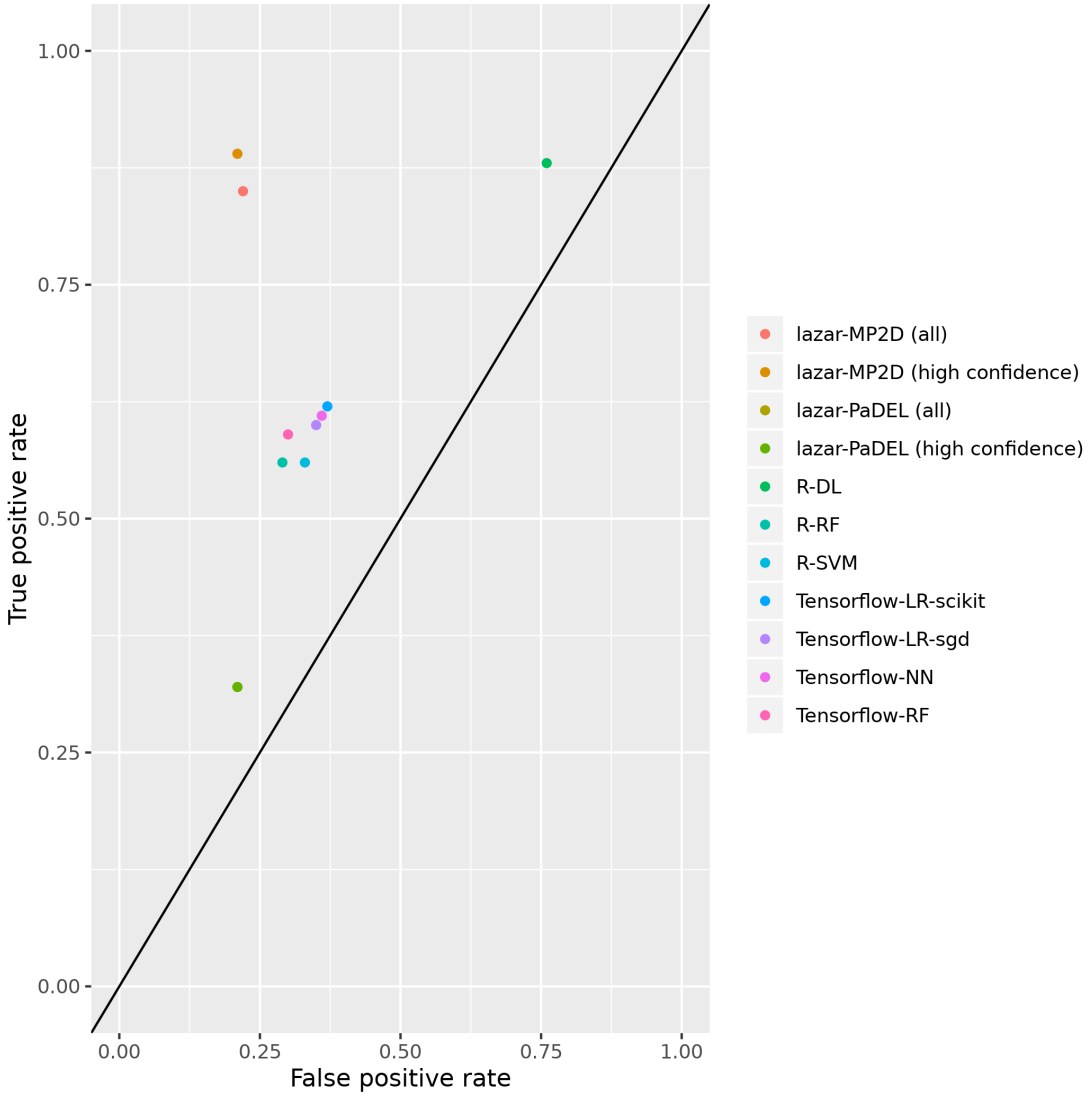


Figure 2: ROC plot of crossvalidation results.

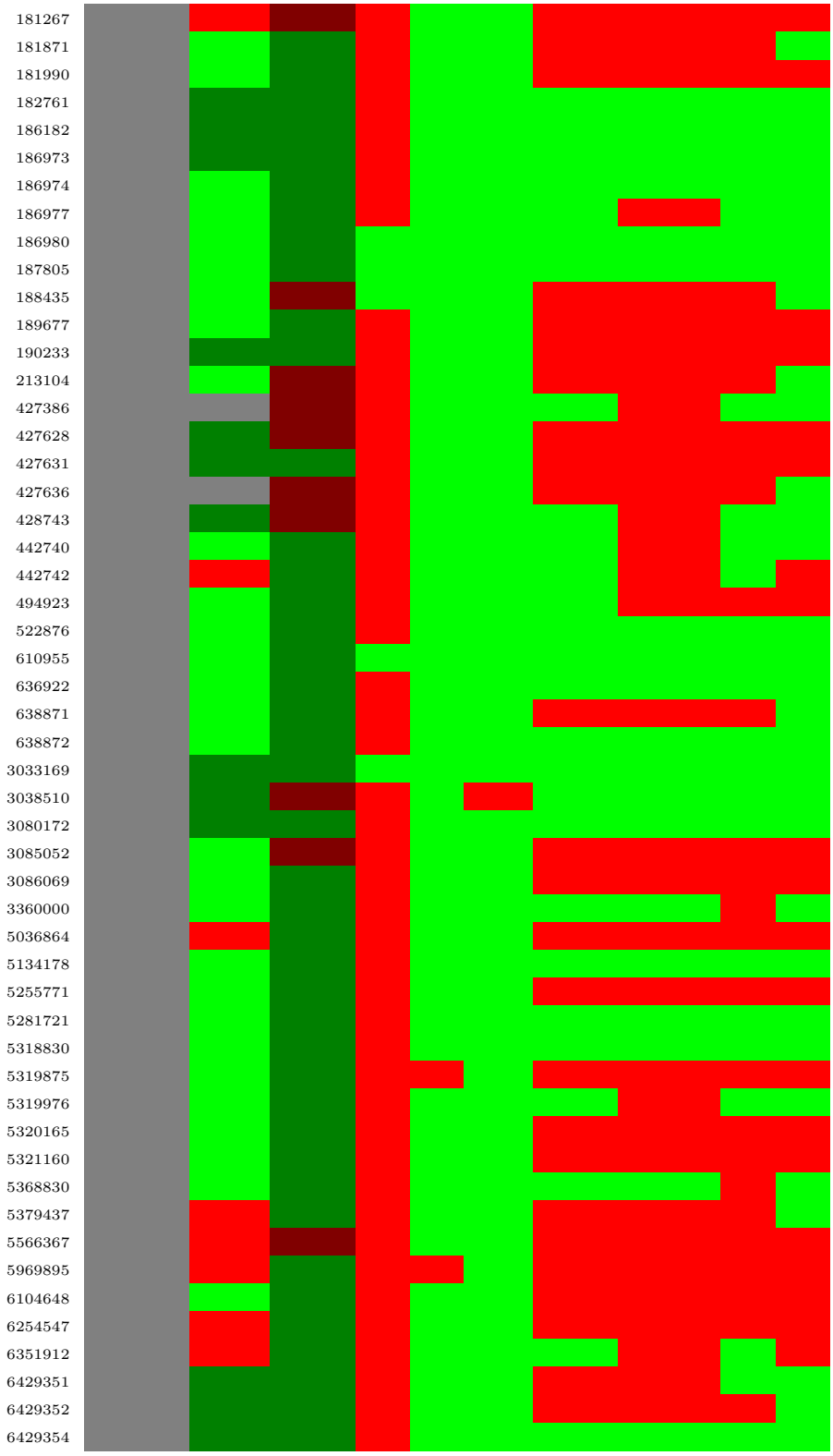
https://git.in-silico.ch/mutagenicity-paper/tables/pa-table.csv

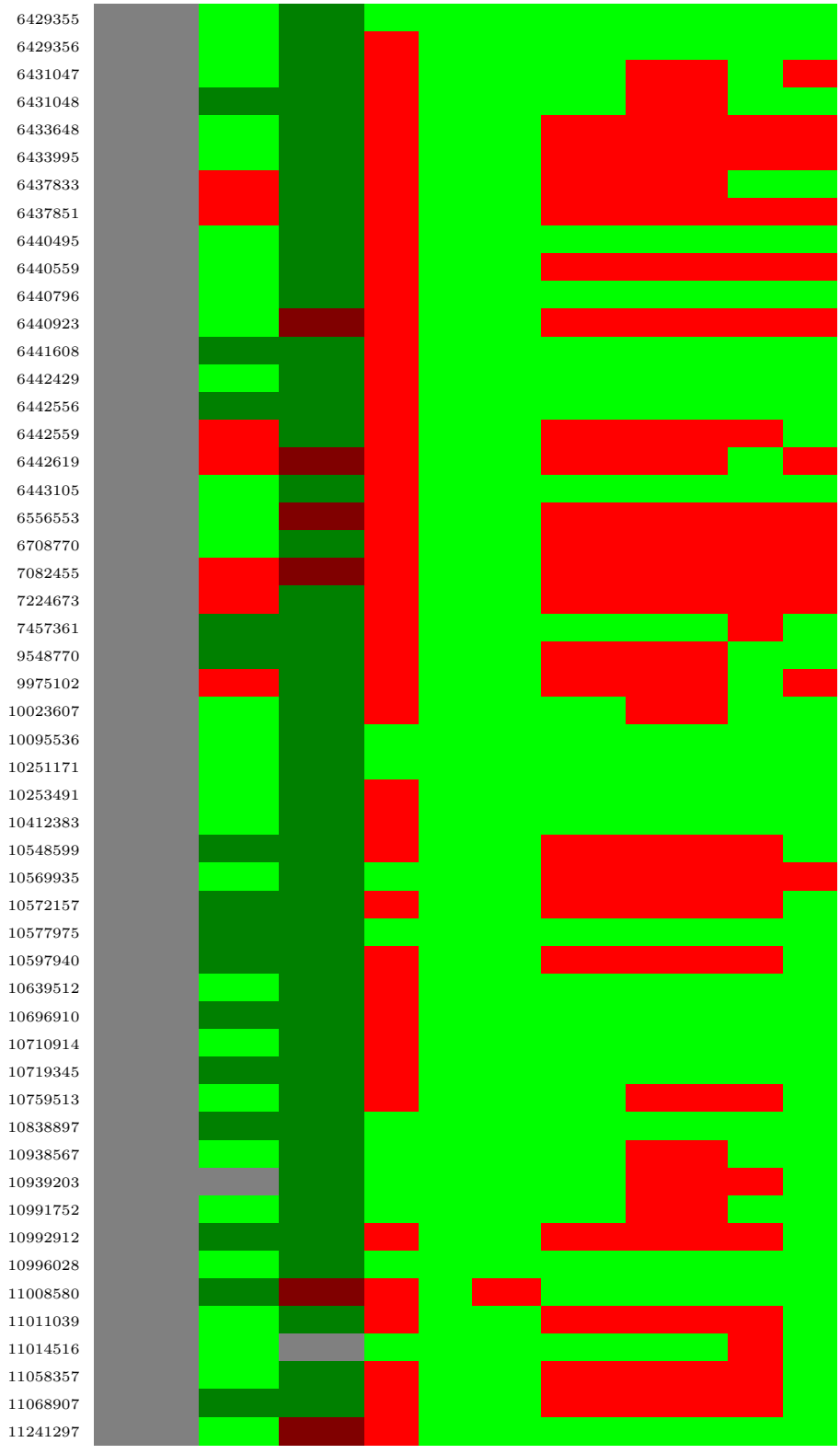
TODO Verena und Philipp Koennt Ihr bitte stichprobenweise die Tabelle ueberpruefen

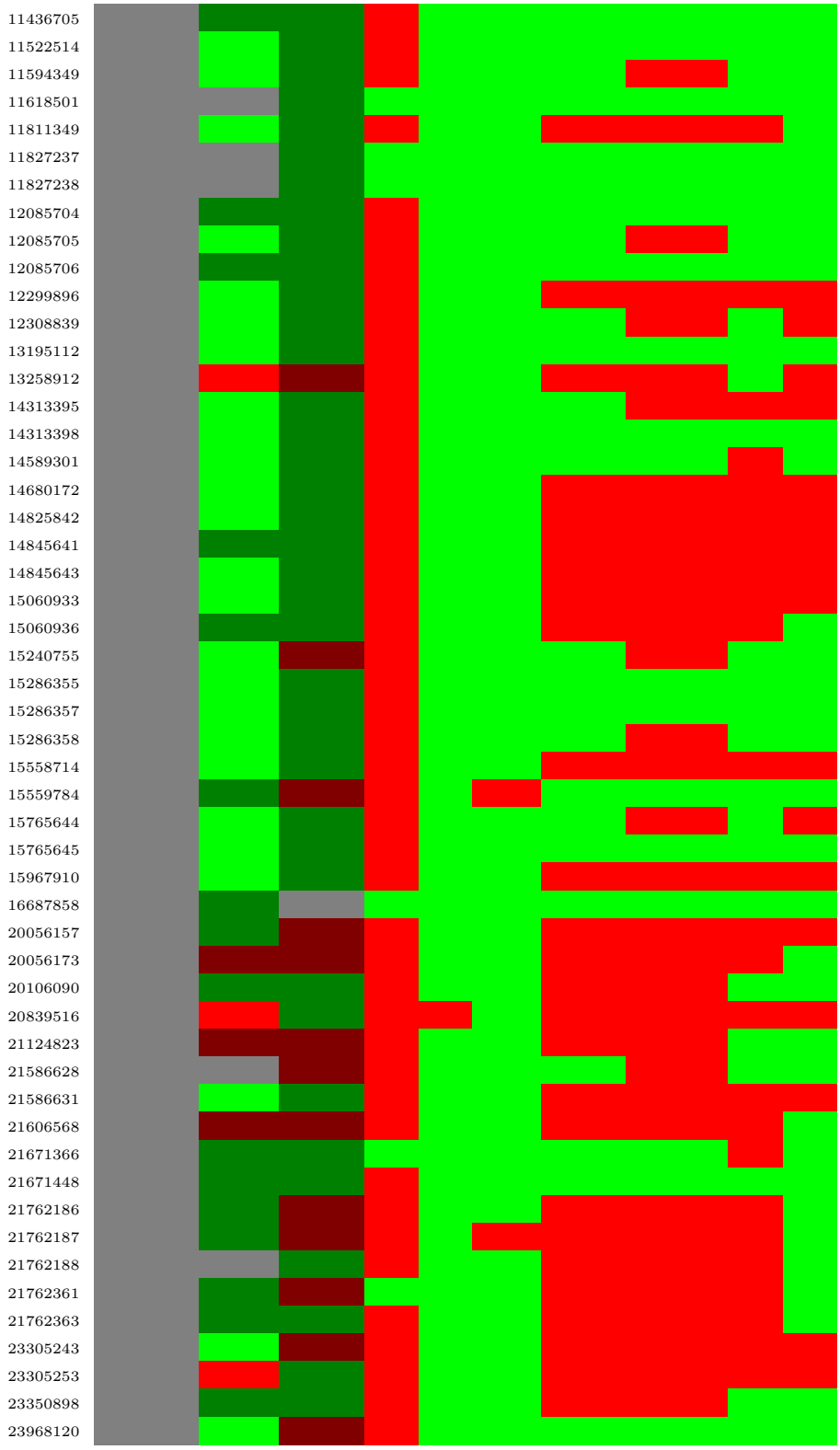
Table 4: Summary of pyrrolizidine alkaloid predictions: red: mutagen, green: non-mutagen, grey: no prediction, dark red/green: low confidence

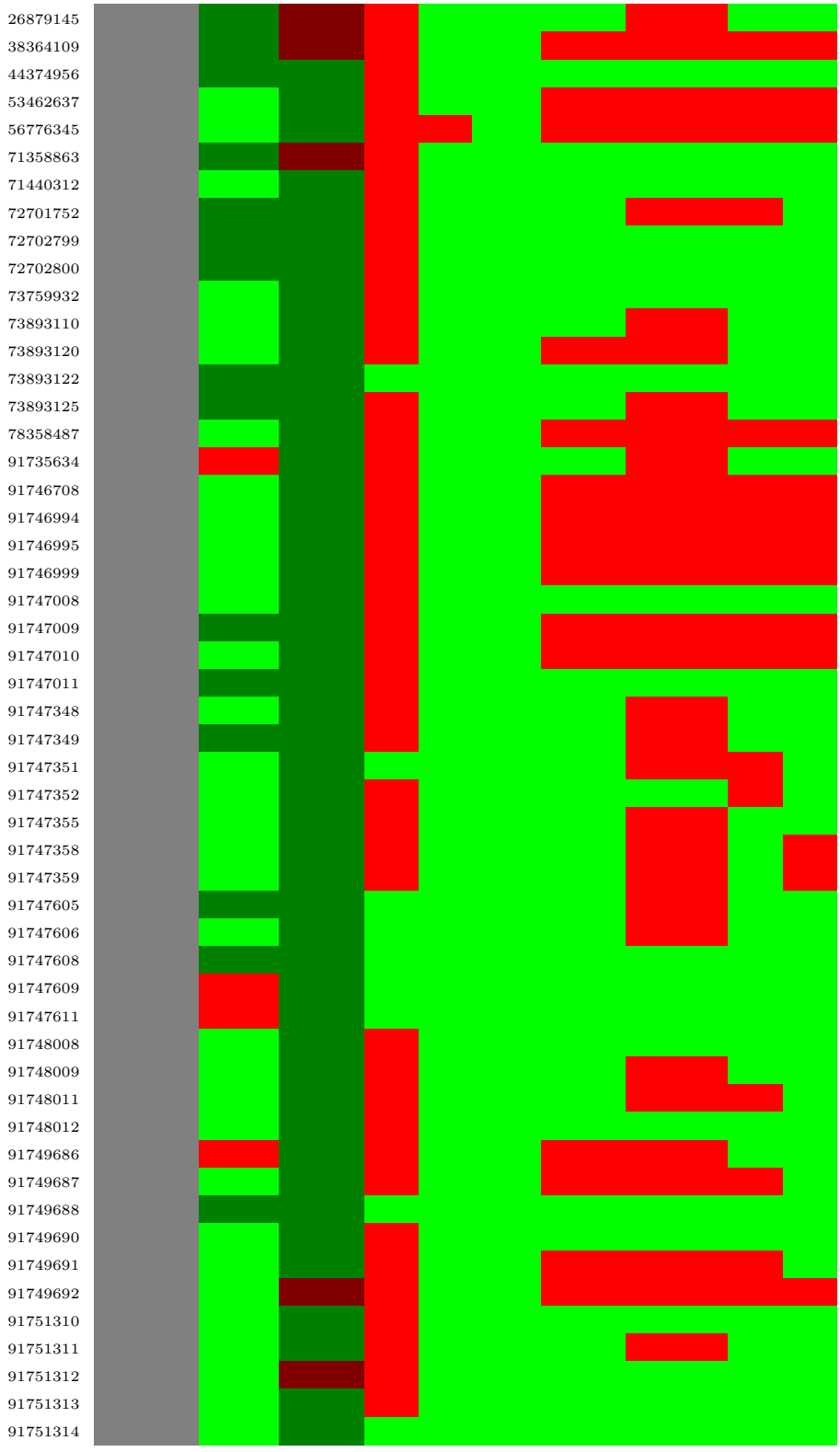
PubChem		lazar		R			Tensorflow			
CID	Measured	MP2D	PaDEL	DL	RF	SVM	LR-sgd	LR-scikit	NN	RF
9415										
5281743										
73614										
119040										
280564										
5280906										
5281733										
5281734										
5281744										
5281756										
5380876										
21606566										
613201										
5281750										
5355258										
9341										
10198										
577603										
99322										
185716										
442726										
5281753										
5281754										
6269253										
6440301										
6440889										
6441178										
12308873										
15120074										
15736564										
76957522										
98222										
197173										
259727										
268949										
279070										
323256										
333106										
333468										
333469										
340066										

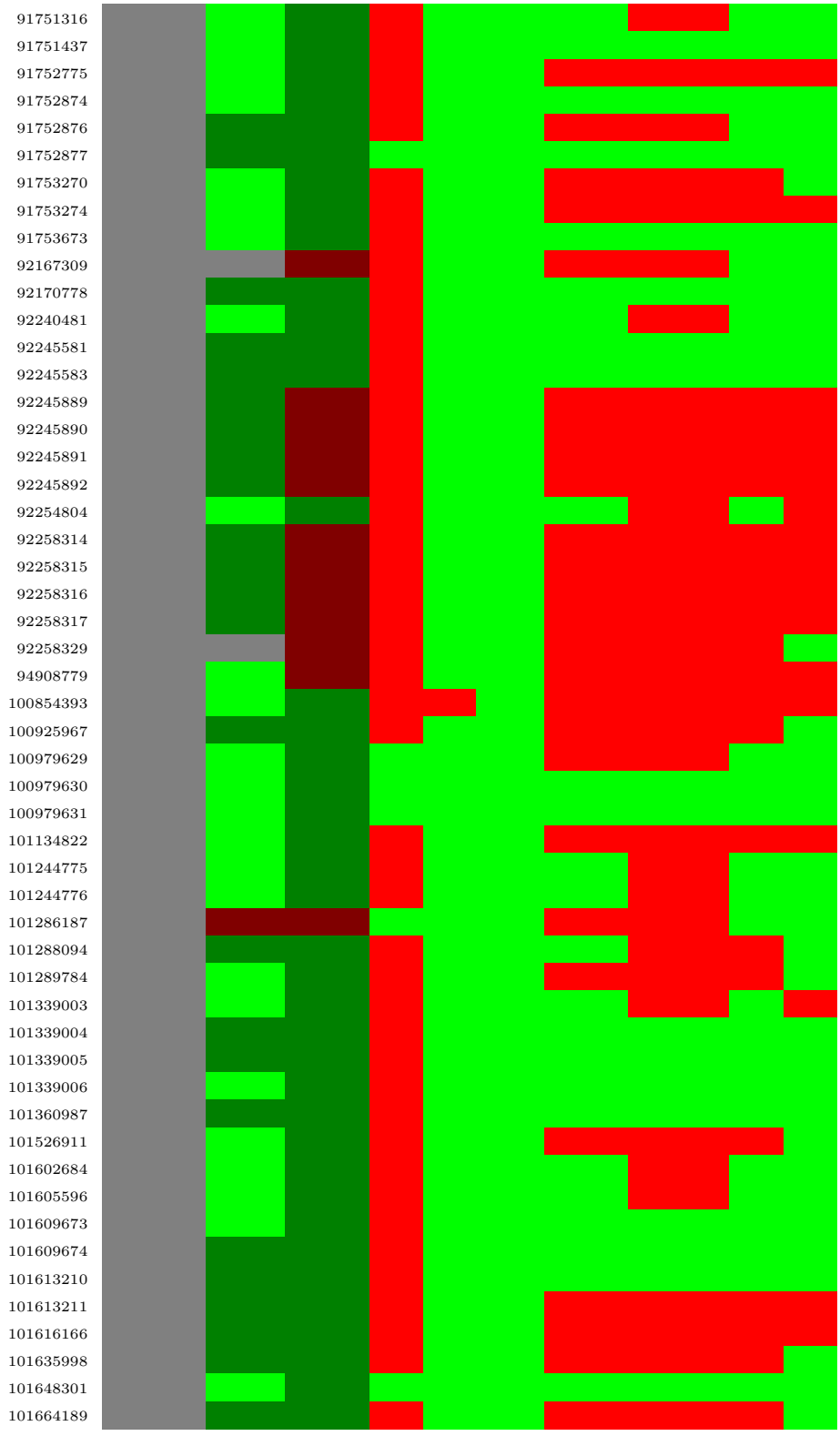

341142
357033
357034
5281729
5458800
6438237
6440654
6604611
10569934
21771304
44284782
44284870
44374459
44374601
44559289
71449185
71449186
71458133
76313095
76315880
76330453
76334042
59827875
91554673
31745
33555
43040
72614
107939
119052
119390
148316
148317
148318
148326
153499
156006
156038
156169
156170
156778
157017
158192
162662
165417
165551
167211
169196
169779
179107
180203
180509

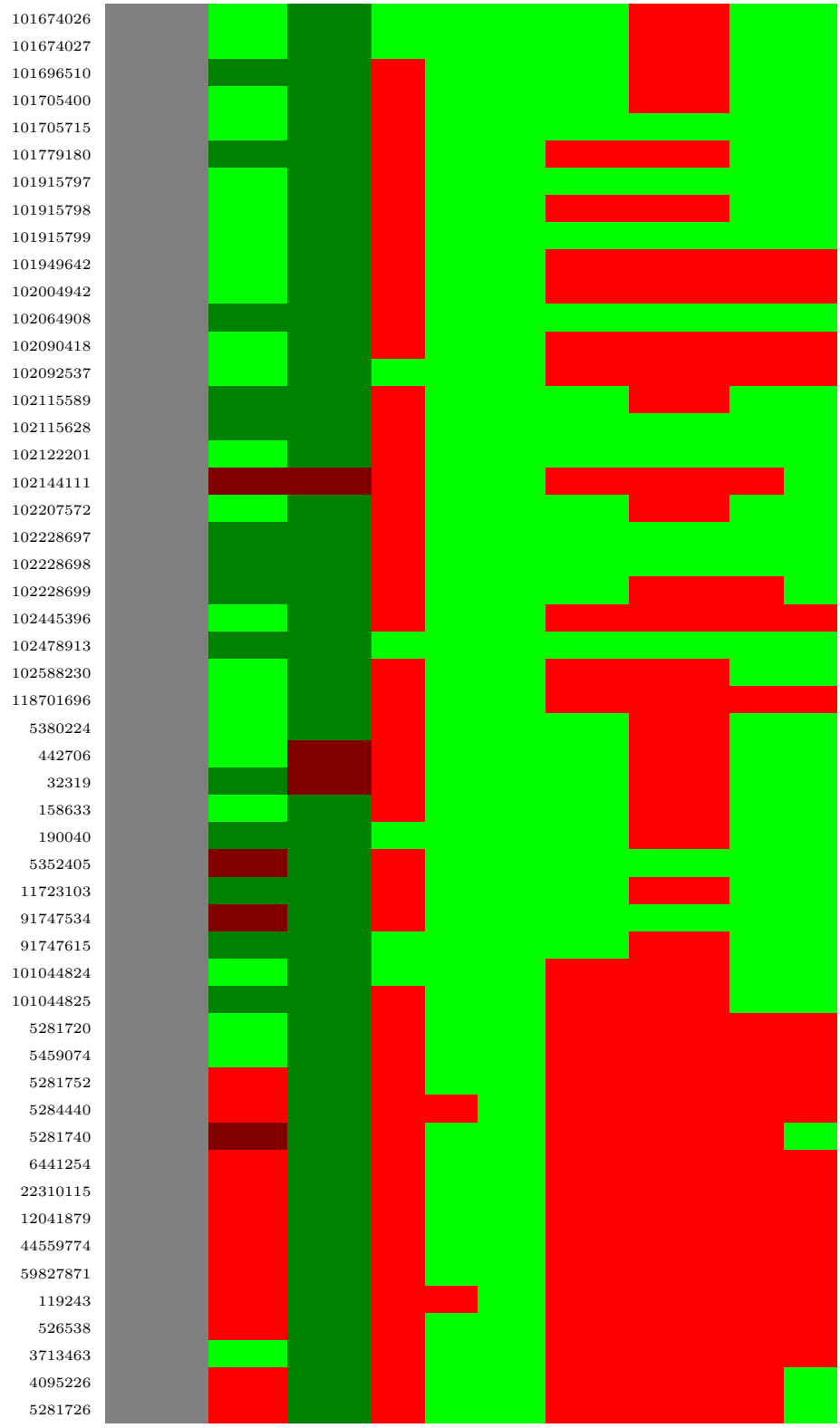

181267
181871
181990
182761
186182
186973
186974
186977
186980
187805
188435
189677
190233
213104
427386
427628
427631
427636
428743
442740
442742
494923
522876
610955
636922
638871
638872
3033169
3038510
3080172
3085052
3086069
3360000
5036864
5134178
5255771
5281721
5318830
5319875
5319976
5320165
5321160
5368830
5379437
5566367
5969895
6104648
6254547
6351912
6429351
6429352
6429354

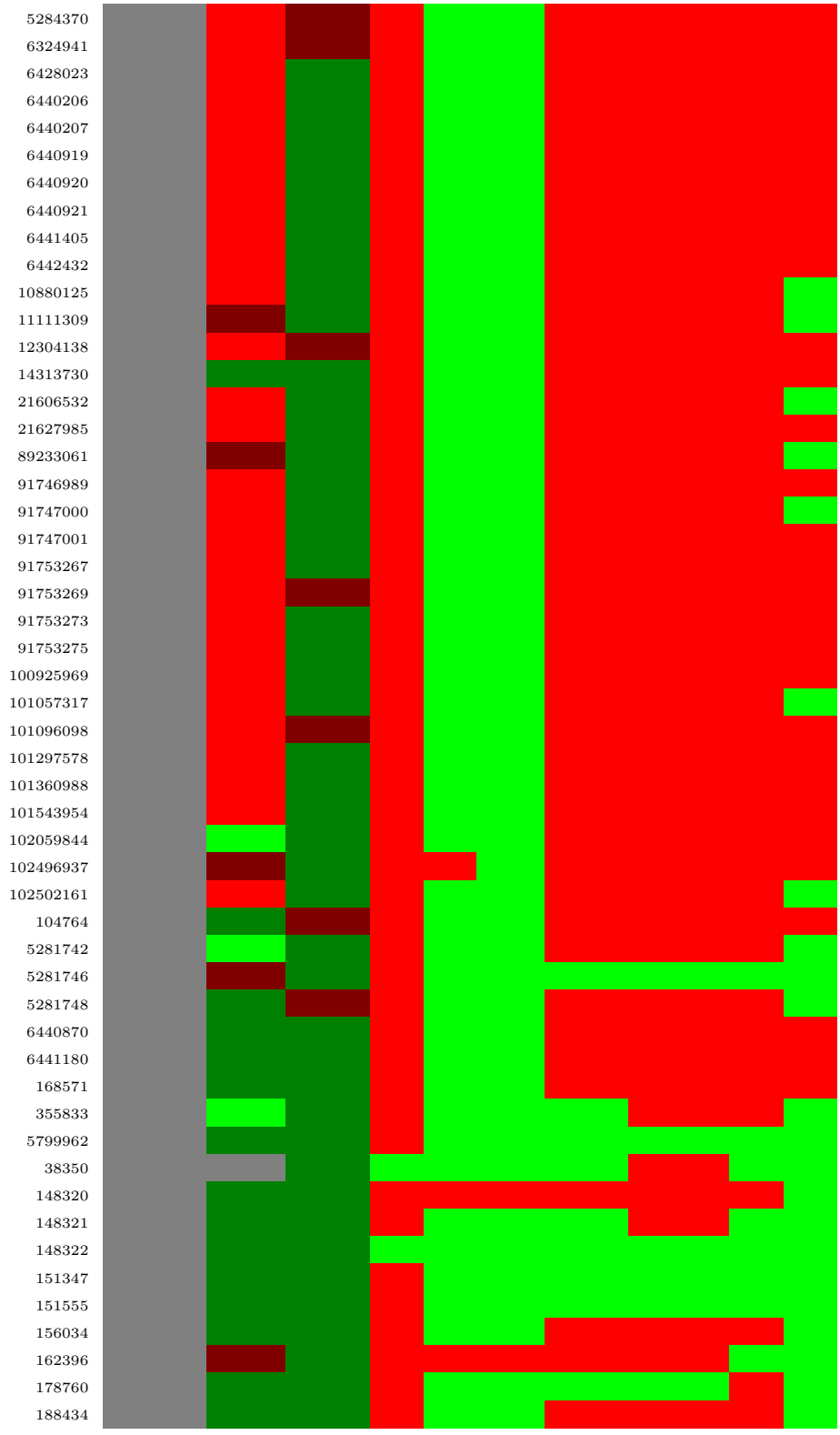

6429355
6429356
6431047
6431048
6433648
6433995
6437833
6437851
6440495
6440559
6440796
6440923
6441608
6442429
6442556
6442559
6442619
6443105
6556553
6708770
7082455
7224673
7457361
9548770
9975102
10023607
10095536
10251171
10253491
10412383
10548599
10569935
10572157
10577975
10597940
10639512
10696910
10710914
10719345
10759513
10838897
10938567
10939203
10991752
10992912
10996028
11008580
11011039
11014516
11058357
11068907
11241297

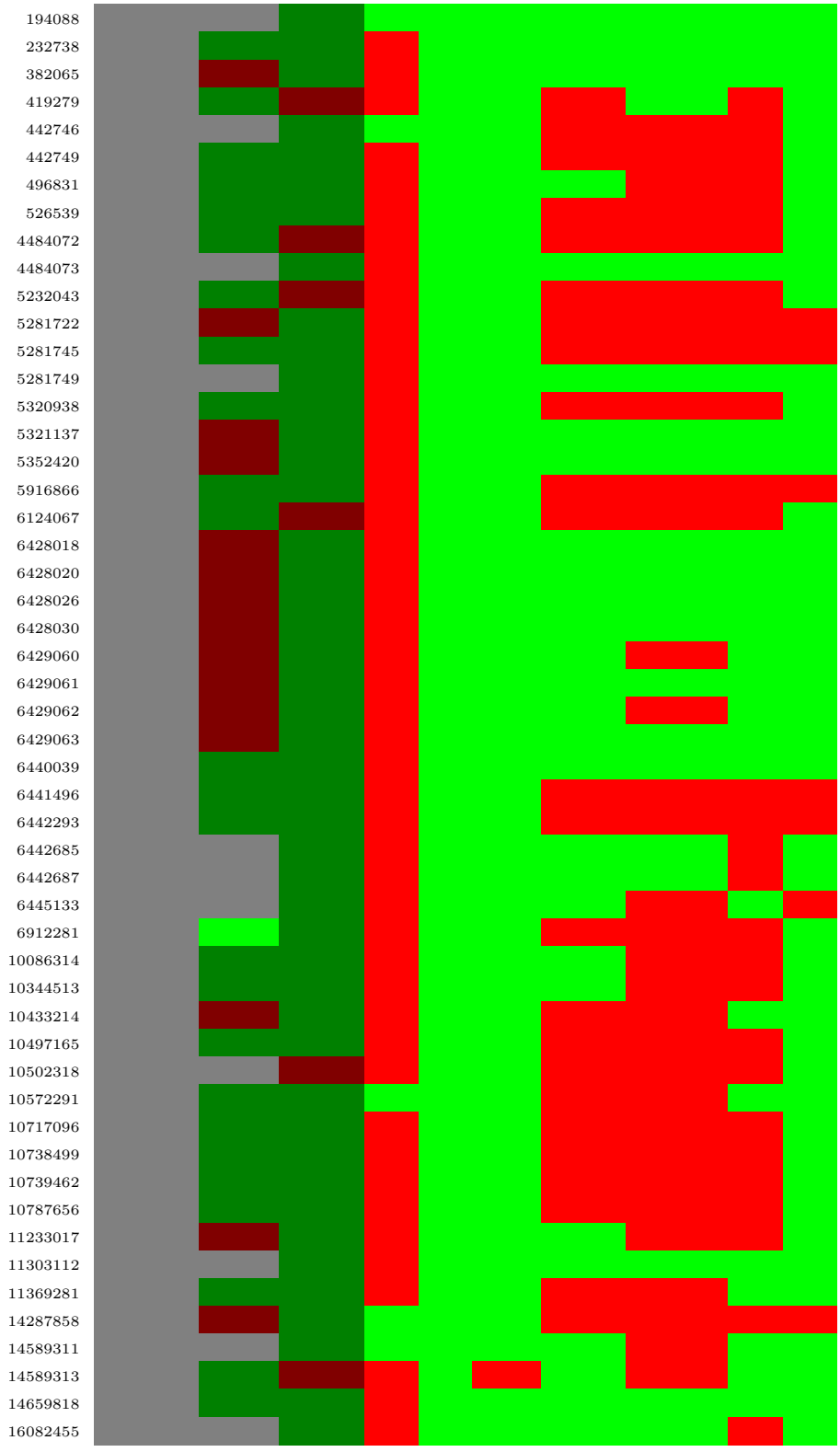

11436705
11522514
11594349
11618501
11811349
11827237
11827238
12085704
12085705
12085706
12299896
12308839
13195112
13258912
14313395
14313398
14589301
14680172
14825842
14845641
14845643
15060933
15060936
15240755
15286355
15286357
15286358
15558714
15559784
15765644
15765645
15967910
16687858
20056157
20056173
20106090
20839516
21124823
21586628
21586631
21606568
21671366
21671448
21762186
21762187
21762188
21762361
21762363
23305243
23305253
23350898
23968120

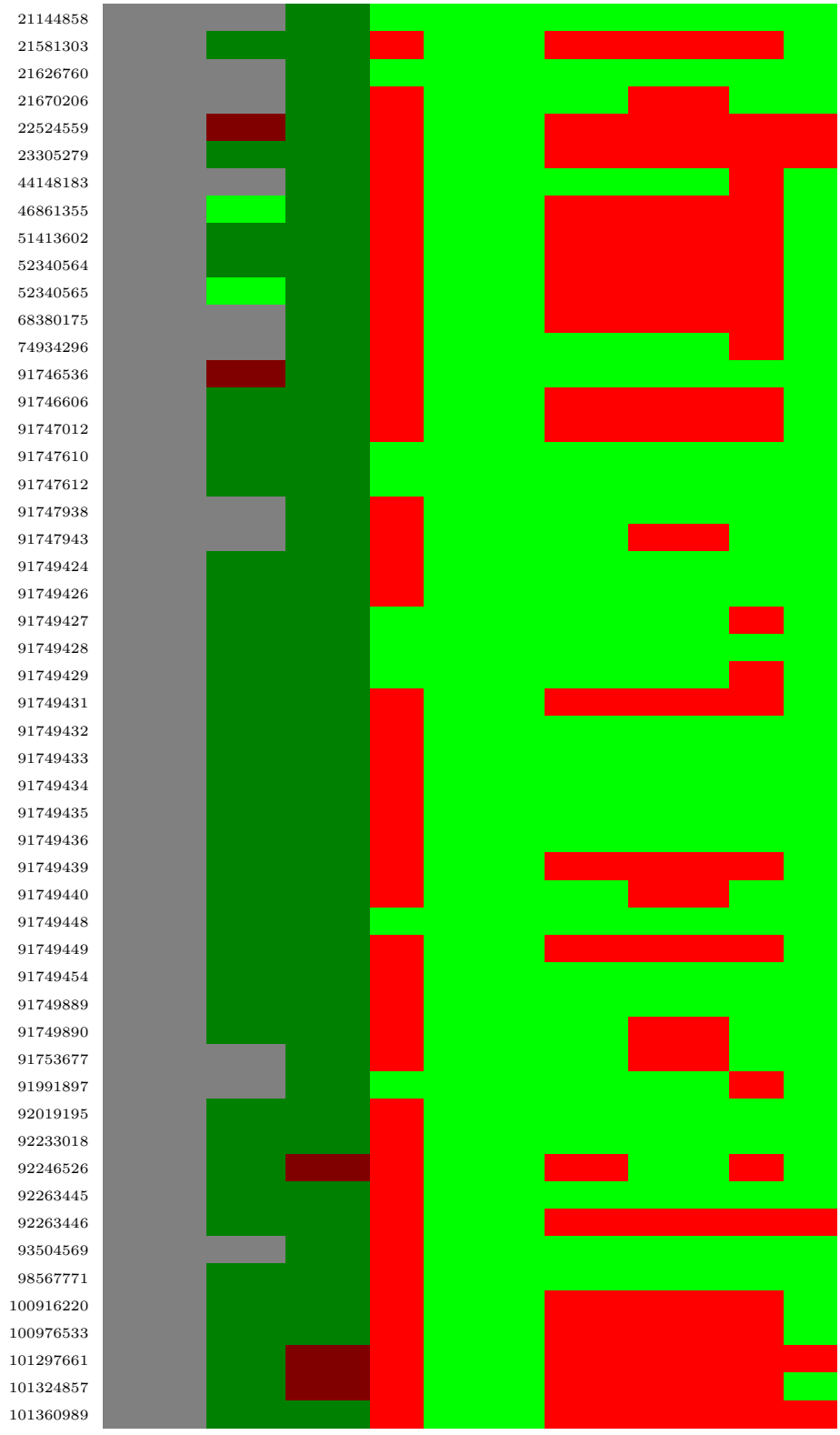

26879145
38364109
44374956
53462637
56776345
71358863
71440312
72701752
72702799
72702800
73759932
73893110
73893120
73893122
73893125
78358487
91735634
91746708
91746994
91746995
91746999
91747008
91747009
91747010
91747011
91747348
91747349
91747351
91747352
91747355
91747358
91747359
91747605
91747606
91747608
91747609
91747611
91748008
91748009
91748011
91748012
91749686
91749687
91749688
91749690
91749691
91749692
91751310
91751311
91751312
91751313
91751314


91751316
91751437
91752775
91752874
91752876
91752877
91753270
91753274
91753673
92167309
92170778
92240481
92245581
92245583
92245889
92245890
92245891
92245892
92254804
92258314
92258315
92258316
92258317
92258329
94908779
100854393
100925967
100979629
100979630
100979631
101134822
101244775
101244776
101286187
101288094
101289784
101339003
101339004
101339005
101339006
101360987
101526911
101602684
101605596
101609673
101609674
101613210
101613211
101616166
101635998
101648301
101664189


101674026
101674027
101696510
101705400
101705715
101779180
101915797
101915798
101915799
101949642
102004942
102064908
102090418
102092537
102115589
102115628
102122201
102144111
102207572
102228697
102228698
102228699
102445396
102478913
102588230
118701696
5380224
442706
32319
158633
190040
5352405
11723103
91747534
91747615
101044824
101044825
5281720
5459074
5281752
5284440
5281740
6441254
22310115
12041879
44559774
59827871
119243
526538
3713463
4095226
5281726


5284370
6324941
6428023
6440206
6440207
6440919
6440920
6440921
6441405
6442432
10880125
11111309
12304138
14313730
21606532
21627985
89233061
91746989
91747000
91747001
91753267
91753269
91753273
91753275
100925969
101057317
101096098
101297578
101360988
101543954
102059844
102496937
102502161
104764
5281742
5281746
5281748
6440870
6441180
168571
355833
5799962
38350
148320
148321
148322
151347
151555
156034
162396
178760
188434


194088
232738
382065
419279
442746
442749
496831
526539
4484072
4484073
5232043
5281722
5281745
5281749
5320938
5321137
5352420
5916866
6124067
6428018
6428020
6428026
6428030
6429060
6429061
6429062
6429063
6440039
6441496
6442293
6442685
6442687
6445133
6912281
10086314
10344513
10433214
10497165
10502318
10572291
10717096
10738499
10739462
10787656
11233017
11303112
11369281
14287858
14589311
14589313
14659818
16082455


21144858
21581303
21626760
21670206
22524559
23305279
44148183
46861355
51413602
52340564
52340565
68380175
74934296
91746536
91746606
91747012
91747610
91747612
91747938
91747943
91749424
91749426
91749427
91749428
91749429
91749431
91749432
91749433
91749434
91749435
91749436
91749439
91749440
91749448
91749449
91749454
91749889
91749890
91753677
91991897
92019195
92233018
92246526
92263445
92263446
93504569
98567771
100916220
100976533
101297661
101324857
101360989

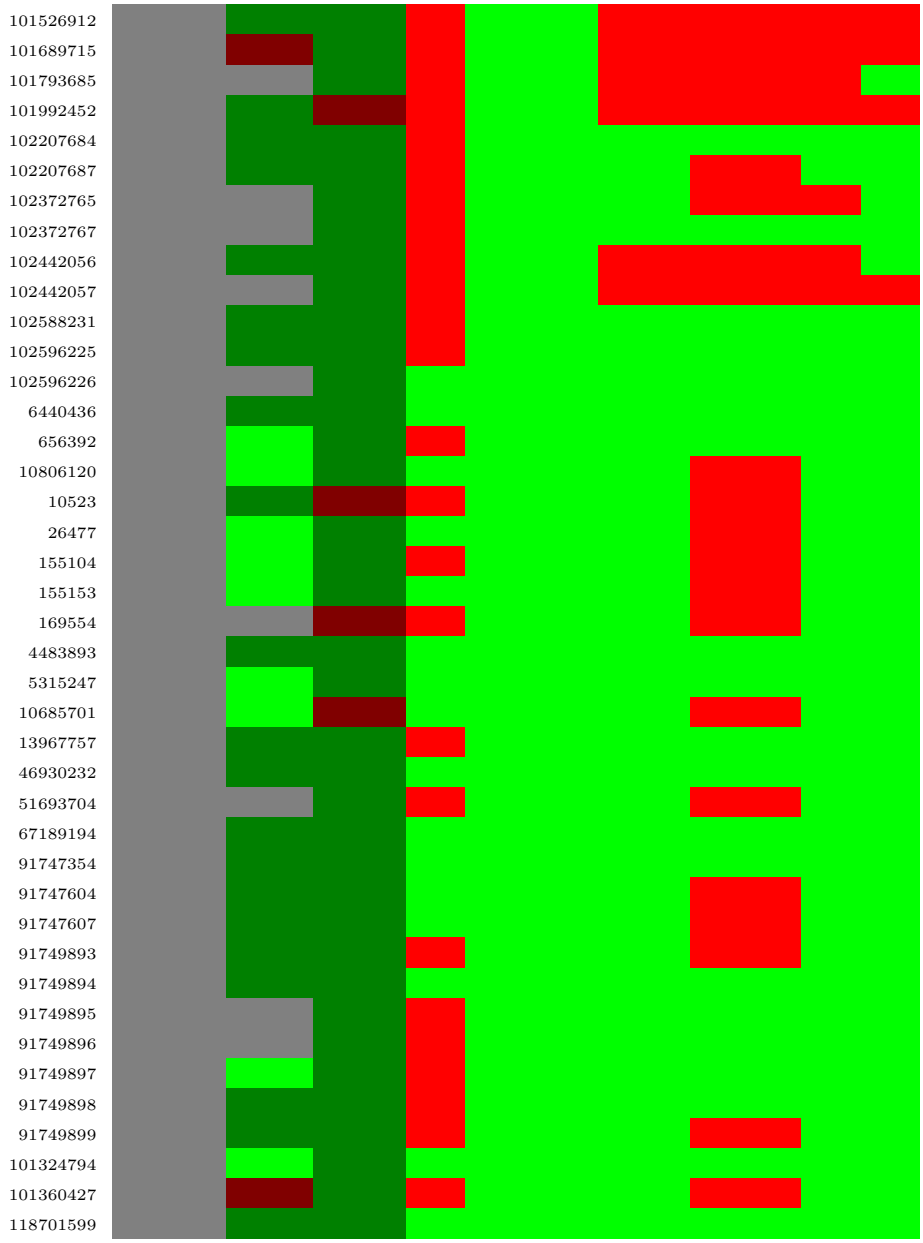


Table 5 summarises the number of positive and negative mutagenicity predictions for all investigated models.

Table 5: Summary of pyrrolizidine alkaloid mutagenicity predictions

Model	Nr.predictions	mutagenic	non-mutagenic
lazar-MP2D (all)	560 (93 %)	111 (20 %)	449 (80 %)
lazar-MP2D (high-confidence)	301 (50 %)	76 (25 %)	225 (75 %)
lazar-PaDEL (all)	600 (100 %)	83 (14 %)	517 (86 %)
lazar-PaDEL (high-confidence)	0 (0 %)	0 (0 %)	0 (0 %)
R-RF	602 (100 %)	18 (3 %)	584 (97 %)
R-SVM	602 (100 %)	11 (2 %)	591 (98 %)
R-DL	602 (100 %)	521 (87 %)	81 (13 %)
Tensorflow-RF	602 (100 %)	186 (31 %)	416 (69 %)
Tensorflow-LR-sgd	602 (100 %)	286 (48 %)	316 (52 %)
Tensorflow-LR-scikit	602 (100 %)	395 (66 %)	207 (34 %)
Tensorflow-NN	602 (100 %)	295 (49 %)	307 (51 %)

For the visualisation of the position of pyrrolizidine alkaloids in respect to the training data set we have applied t-distributed stochastic neighbor embedding (t-SNE, Maaten and Hinton (2008)) for MolPrint2D and PaDEL descriptors. t-SNE maps each high-dimensional object (chemical) to a two-dimensional point, maintaining the high-dimensional distances of the objects. Similar objects are represented by nearby points and dissimilar objects are represented by distant points.

Figure 3 shows the t-SNE of pyrrolizidine alkaloids (PA) and the mutagenicity training data in MP2D space (Tanimoto/Jaccard similarity).

Figure 4 shows the t-SNE of pyrrolizidine alkaloids (PA) and the mutagenicity training data in PaDEL space (Euclidean similarity).

Figure 3: t-SNE visualisation of mutagenicity training data and pyrrolizidine alkaloids (PA)

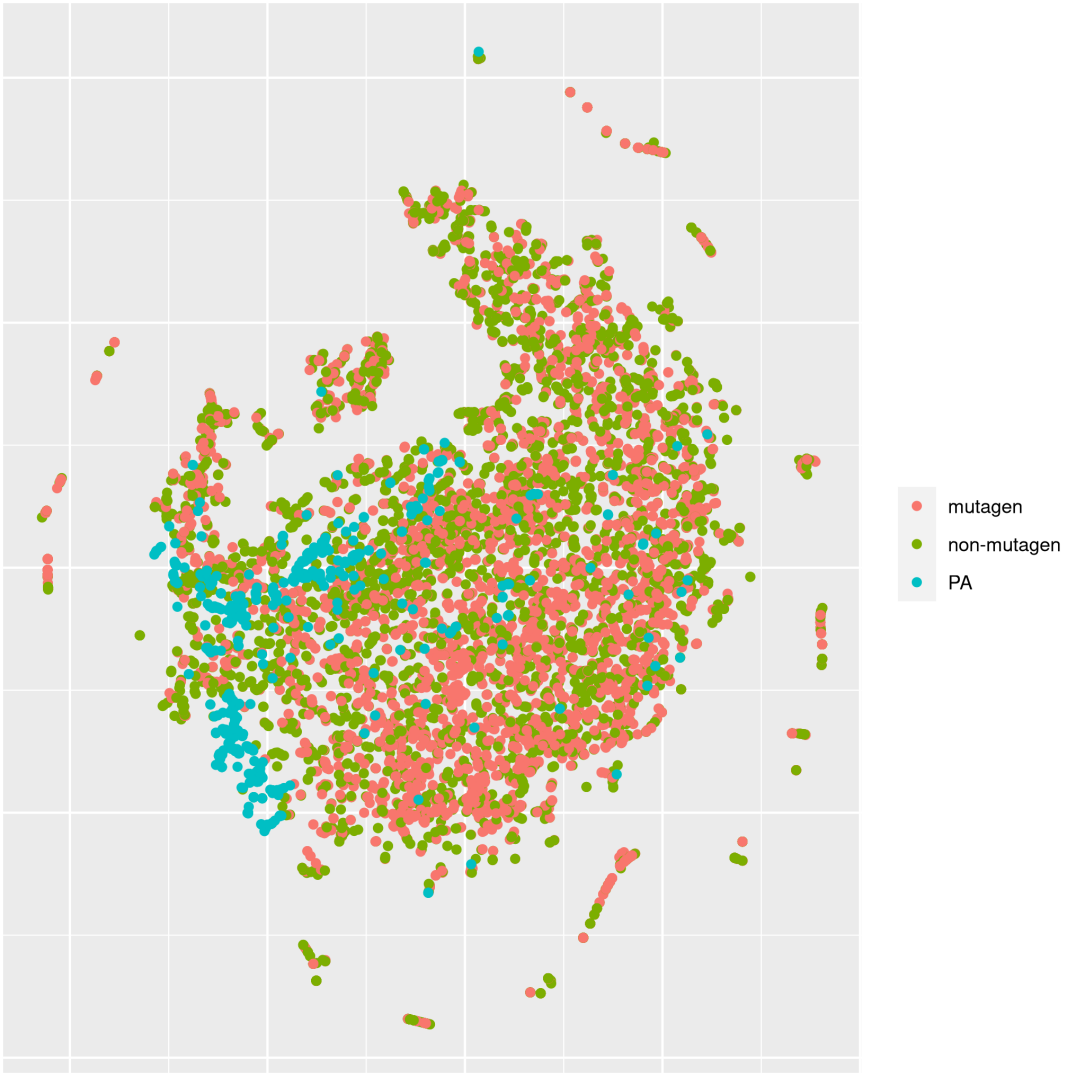


Figure 4: t-SNE visualisation of mutagenicity training data and pyrrolizidine alkaloids (PA)

Discussion

Data

A new training dataset for *Salmonella* mutagenicity was created from three different sources (Kazius, McGuire, and Bursi (2005), Hansen et al. (2009), EFSA (2016)). It contains 8309 unique chemical structures, which is according to our knowledge the largest public mutagenicity dataset presently available. The new training data can be downloaded from https://git.in-silico.ch/mutagenicity-paper/data/mutagenicity.csv.

Model performance

Table 1, Table 2, Table 3 and Figure 2 show that the standard `lazar` algorithm (with MP2D fingerprints) give the most accurate crossvalidation results. R Random Forests, Support Vector Machines and Tensorflow models have similar accuracies with balanced sensitivity (true position rate) and specificity (true negative rate). `lazar` models with PaDEL descriptors have low sensitivity and R Deep Learning models have low specificity.

The accuracy of `lazar` *in-silico* predictions are comparable to the interlaboratory variability of the Ames test (80-85% according to Benigni and Giuliani (1988)), especially for predictions with high confidence (84%). This is a clear indication that *in-silico* predictions can be as reliable as the bioassays, if the compounds are close to the applicability domain. This conclusion is also supported by our analysis of `lazar` lowest observed effect level predictions, which are also similar to the experimental variability (Helma et al. (2018)).

The lowest number of predictions (4081) has been obtained from `lazar`-PaDEL high confidence predictions, the largest number of predictions comes from Tensorflow models (). Standard `lazar` give a slightly lower number of predictions (7781) than R and Tensorflow models. This is not necessarily a disadvantage, because `lazar` abstains from

predictions, if the query compound is very dissimilar from the compounds in the training set and thus avoids to make predictions for compounds out of the applicability domain.

Descriptors

This study uses two types of descriptors for the characterisation of chemical structures:

MolPrint2D fingerprints (MP2D, Bender et al. (2004)) use atom environments (i.e. connected atom types for all atoms in a molecule) as molecular representation, which resembles basically the chemical concept of functional groups. MP2D descriptors are used to determine chemical similarities in the default `lazar` settings, and previous experiments have shown, that they give more accurate results than predefined fragments (e.g. MACCS, FP2-4).

In order to investigate, if MP2D fingerprints are also suitable for global models we have tried to build R and Tensorflow models, both with and without unsupervised feature selection. Unfortunately none of the algorithms was capable to deal with the large and sparsely populated descriptor matrix. Based on this result we can conclude, that MolPrint2D descriptors are at the moment unsuitable for standard global machine learning algorithms.

`lazar` does not suffer from the size and sparseness problem, because (a) it utilizes internally a much more efficient occurrence based representation and (b) it uses fingerprints only for similarity calculations and not as model parameters.

PaDEL calculates topological and physical-chemical descriptors.

TODO: **Verena** kannst Du bitte die Deskriptoren nochmals kurz beschreiben

PaDEL descriptors were used for `lazar`, R and Tensorflow models. All models based on PaDEL descriptors had similar crossvalidation accuracies that were significantly lower than `lazar` MolPrint2D results. Direct comparisons are available only for the `lazar`

algorithm, and also in this case PaDEL accuracies were lower than MolPrint2D accuracies.

Based on `lazar` results we can conclude, that PaDEL descriptors are less suited for chemical similarity calculations than MP2D descriptors. It is also likely that PaDEL descriptors lead to less accurate predictions for global models, but we cannot draw any definitive conclusion in the absence of MP2D models.

Algorithms

`lazar` is formally a *k-nearest-neighbor* algorithm that searches for similar structures for a given compound and calculates the prediction based on the experimental data for these structures. The QSAR literature calls such models frequently *local models*, because models are generated specifically for each query compound. R and Tensorflow models are in contrast *global models*, i.e. a single model is used to make predictions for all compounds. It has been postulated in the past, that local models are more accurate, because they can account better for mechanisms, that affect only a subset of the training data. Our results seem to support this assumption, because standard `lazar` models with MolPrint2D descriptors perform better than global models. The accuracy of `lazar` models with PaDEL descriptors is however substantially lower and comparable to global models with the same descriptors.

This observation may lead to the conclusion that the choice of suitable descriptors is more important for predictive accuracy than the modelling algorithm, but we were unable to obtain global MP2D models for direct comparisons. The selection of an appropriate modelling algorithm is still crucial, because it needs the capability to handle the descriptor space. Neighbour (and thus similarity) based algorithms like `lazar` have a clear advantage in this respect over global machine learning algorithms (e.g. RF, SVM, LR, NN), because Tanimoto/Jaccard similarities can be calculated efficiently with simple set

operations.

Pyrrolizidine alkaloid mutagenicity predictions

`lazar` models with MolPrint2D descriptors predicted 93% of the pyrrolizidine alkaloids (PAs) (50% with high confidence), the remaining compounds are not within its applicability domain. All other models predicted 100% of the 602 compounds, indicating that all compounds are within their applicability domain.

Mutagenicity predictions from different models show little agreement in general (table 4). 42 from 602 PAs have non-conflicting predictions (all of them non-mutagenic). Most models predict predominantly a non-mutagenic outcome for PAs, with exception of the R deep learning (DL) and the Tensorflow Scikit logistic regression models (and 66% positive predictions).

R RF and SVM models favor very strongly non-mutagenic predictions (only 3 and 2 % mutagenic PAs), while Tensorflow models classify approximately half of the PAs as mutagenic (RF 31%, LR-sgd {:n=>602, :mut=>286, :non_mut=>316, :n_perc=>100, :mut_perc=>48, :non_mut_perc=>52}%, LR-scikit:66, LR-NN:49%). `lazar` models predict predominately non-mutagenicity, but to a lesser extend than R models (MP2D:20, PaDEL:14).

It is interesting to note, that different implementations of the same algorithm show little accordance in their prediction (see e.g R-RF vs. Tensorflow-RF and LR-sgd vs. LR-scikit in Table 4 and Table 5).

TODO Verena, Philipp habt ihr eine Erklaerung dafuer?

Figure 3 and Figure 4 show the t-SNE of training data and pyrrolizidine alkaloids. In Figure 3 the PAs are located closely together at the outer border of the training set. In Figure 4 they are less clearly separated and spread over the space occupied by the

training examples.

This is probably the reason why PaDEL models predicted all instances and the MP2D model only 560 PAs. Predicting a large number of instances is however not the ultimate goal, we need accurate predictions and an unambiguous estimation of the applicability domain. With PaDEL descriptors *all* PAs are within the applicability domain of the training data, which is unlikely despite the size of the training set. MolPrint2D descriptors provide a clearer separation, which is also reflected in a better separation between high and low confidence predictions in `lazar` MP2D predictions as compared to `lazar` PaDEL predictions. Crossvalidation results with substantially higher accuracies for MP2D models than for PaDEL models also support this argument.

Differences between MP2D and PaDEL descriptors can be explained by their specific properties: PaDEL calculates a fixed set of descriptors for all structures, while MolPrint2D descriptors resemble substructures that are present in a compound. For this reason there is no fixed number of MP2D descriptors, the descriptor space are all unique substructures of the training set. If a query compound contains new substructures, this is immediately reflected in a lower similarity to training compounds, which makes applicability domain estimations very straightforward. With PaDEL (or any other predefined descriptors), the same set of descriptors is calculated for every compound, even if a compound comes from an completely new chemical class.

From a practical point we still have to face the question, how to choose model predictions, if no experimental data is available (we found two PAs in the training data, but this number is too low, to draw any general conclusions). Based on crossvalidation results and the arguments in favor of MolPrint2D descriptors we would put the highest trust in `lazar` MolPrint2D predictions, especially in high-confidence predictions. `lazar` predictions have a accuracy comparable to experimental variability (Helma et al. (2018)) for compounds within the applicability domain. But they should not be trusted blindly.

For practical purposes it is important to study the rationales (i.e. neighbors and their experimental activities) for each prediction of relevance. A freely accessible GUI for this purpose has been implemented at https://lazar.in-silico.ch.

TODO: **Verena** Wenn Du lazar Ergebnisse konkret diskutieren willst, kann ich Dir ausfuehrliche Vorhersagen (mit aehnlichen Verbindungen und deren Aktivitaet) fuer einzelne Beispiele zusammenstellen

Conclusions

A new public *Salmonella* mutagenicity training dataset with 8309 compounds was created and used it to train `lazar`, R and Tensorflow models with MolPrint2D and PaDEL descriptors. The best performance was obtained with `lazar` models using MolPrint2D descriptors, with prediction accuracies (84%) comparable to the interlaboratory variability of the Ames test (80-85%). Models based on PaDEL descriptors had lower accuracies than MolPrint2D models, but only the `lazar` algorithm could use MolPrint2D descriptors.

TODO: PA Vorhersagen

References

Bender, Andreas, Hamse Y. Mussa, Robert C. Glen, and Stephan Reiling. 2004. "Molecular Similarity Searching Using Atom Environments, Information-Based Feature Selection, and a Naïve Bayesian Classifier." *Journal of Chemical Information and Computer Sciences* 44 (1): 170–78. https://doi.org/10.1021/ci034207y.

Benigni, R., and A. Giuliani. 1988. "Computer-assisted Analysis of Interlaboratory Ames Test Variability." *Journal of Toxicology and Environmental Health* 25 (1): 135–48.

https://doi.org/10.1080/15287398809531194.

EFSA. 2011. "Scientific Opinion on Pyrrolizidine Alkaloids in Food and Feed." *EFSA Journal*, no. 9: 1–134.

———. 2016. "Guidance on the Establishment of the Residue Definition for Dietary Assessment: EFSA Panel on Plant Protect Products and Their Residues (PPR)." *EFSA Journal*, no. 14: 1–12.

Hansen, Katja, Sebastian Mika, Timon Schroeter, Andreas Sutter, Antonius ter Laak, Thomas Steger-Hartmann, Nikolaus Heinrich, and Klaus-Robert Müller. 2009. "Benchmark Data Set for in Silico Prediction of Ames Mutagenicity." *Journal of Chemical Information and Modeling* 49 (9): 2077–81. https://doi.org/10.1021/ci900161g.

Helma, Christoph, David Vorgrimmler, Denis Gebele, Martin Gütlein, Barbara Engeli, Jürg Zarn, Benoit Schilter, and Elena Lo Piparo. 2018. "Modeling Chronic Toxicity: A Comparison of Experimental Variability with (Q)SAR/Read-Across Predictions." *Frontiers in Pharmacology*, no. 9: 413.

Kazius, J., R. McGuire, and R. Bursi. 2005. "Derivation and Validation of Toxicophores for Mutagenicity Prediction." *J Med Chem*, no. 48: 312–20.

Maaten, L. J. P. van der, and G. E. Hinton. 2008. "Visualizing Data Using T-Sne." *Journal of Machine Learning Research*, no. 9: 2579–2605.

Mattocks, AR. 1986. *Chemistry and Toxicology of Pyrrolizidine Alkaloids*. Academic Press.

O'Boyle, Noel, Michael Banck, Craig James, Chris Morley, Tim Vandermeersch, and Geoffrey Hutchison. 2011. "Open Babel: An open chemical toolbox." *J. Cheminf.* 3 (1): 33. https://doi.org/doi:10.1186/1758-2946-3-33.

Schöning, Verena, Felix Hammann, Mark Peinl, and Jürgen Drewe. 2017. "Editor's

Highlight: Identification of Any Structure-Specific Hepatotoxic Potential of Different Pyrrolizidine Alkaloids Using Random Forests and Artificial Neural Networks." *Toxicol. Sci.*, no. 160: 361–70.

Yap, CW. 2011. "PaDEL-Descriptor: An Open Source Software to Calculate Molecular Descriptors and Fingerprints." *Journal of Computational Chemistry*, no. 32: 1466–74.