Co-funded by the Horizon 2020
Framework Programme of the
European Union

Grant agreement No. 671555

ExCAPE
Exascale Compound Activity Prediction Engine

Future and Emerging Technologies (FET)

Call: H2020-FETHPC-2014
Topic: FETHPC-1-2014
Type of action: RIA

Deliverable D1.2

Report: Metamodel Report

Due date of deliverable: 01.03.2016
Actual submission date: 18.04.2016

Start date of Project: 1.9.2015 Duration: 36 months

Responsible Consortium Partner: JP
Name of author(s) and contributors: Vladimir Chupakhin (JP)
Revision: V2
Internal reviewer(s): Paolo Toccaceli (RHUL)

Thomas J.

Project co-funded by the European Union within the Horizon 2020 Framework Programme (2014-2020)		
Dissemination Level		
PU	Public	**PU**
PP	Restricted to other programme participants (including the Commission Services	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document revision tracking

This page is used to follow the deliverable production from its first version until it has been reviewed by the assessment team. Please give details in the table below about successive releases.

Release number	Date	Reason of this release and/or validation	Dissemination of this release (task level, WP/ST level, Project Office Manager, Industrial Steering Committee, etc)
V1	18.04.2016	First version	Consortium
V2	24.07.2016	Second version after project review	Public

Glossary

HPC	High-performance computing
pK_i	The (negative log of the) dissociation constant of a complex
pIC_{50}	The (negative log of the) concentration of an inhibitor where the response (or binding) is reduced by half.
SVM	Support Vector Machine
JSON	JavaScript Object Notation
XML	Extensible Markup Language

Link to Tasks

Task number	Work from task carried out	Deviations from task technical content
T1.1.1	Coordination of the interactions between industry and academy partners	None
T1.1.2	Coordination of the interfacing between different developed algorithms	None
T1.2.1	Create meta-algorithm(s) for selecting best modeling scenario (description of meta-models used to evaluate algorithms)	None

Table of contents

1 Executive summary

The purpose of the ExCAPE project is to produce better chemogenomics models using large-scale multi-task machine learning techniques. Because different machine learning algorithms produce different solutions, there is a need for a well-defined methodological framework to compare those models in terms of quality, computational benchmarks and other related information, and to show how to combine models. In this framework, three broad categories can be identified: *meta-information* – i.e. any information related to the model, such as quality metric used (discussed in D3.3), computational time taken, hyperparameters tested, folding and sampling used, etc.; *calibration* – i.e. how to represent the results of the different approaches in a way that allows a fair comparison (for example, by using probability distributions); and finally, *ensembling* – i.e. combining several model into one model that probably can give better results compared to the individual models.

This document is a set of guidelines to help to develop a pipeline to compare and evaluate the quality of the models and build a final ensemble model at the end of the project. This guideline should be taken into account in all developmental stages and packages: WP1 – integration of calibration into the machine learning algorithms; WP2 – efficient implementation of the data handling needed for the cross-validation setup and parameter evaluation; WP3 – generation of the data according to the pipeline, quality criteria and possible ensembling techniques.

2 Introduction – Aim

One of the purposes of the ExCAPE project is to develop and evaluate large-scale machine learning methods for the prediction of activities and toxicities of chemical compounds. The methods considered are mainly group around deep learning [1] and matrix factorization [2]. There are several other important objectives that we would like to achieve in the project. One of them is to determine how good multi-task models are in comparison with single task models. For this purpose we deploy SVM models developed in collaboration between WP2 and WP3 [D2.1, D2.2 and D3.7] as a baseline for comparison. Another import objective is to assess how good different compound encodings are for building predictive models, and especially a comparison of biological descriptors versus chemical descriptors to understand the performance and cost-benefit ratio for experimental biological descriptors, imaging and L1000 gene expression. The diversity of possible solutions and possible modeling scenarios requires a guideline to compare, evaluate and reuse the models within the project – a metamodel. There are three main parts that the metamodel is composed of: meta-information, calibration and ensembling. The purpose of this deliverable is to give an overview of those parts.

Meta-information is any information associated with the model that does not include the data itself. This can include a computational benchmark, metrics used for quality estimation and hyperparameters used to build a model, links to the folder with the data used to train and validate the model and many other related pieces of information that can be useful later in the project and can be used to reproduce the results. An example of python code with meta-information used to build a single task SVM model is given below.

```
QUALITY_METRIC = ["ROC_AUC"]
INNER_FOLDS = [1, 2, 3, 4, 5]
DESCRIPTOR = [“ECFP6"]
OUTER_FOLDS = [1, 2, 3, 4, 5]
COST_GRID = [1.0, 3.1623, 10.0, 31.6228, 100.0, 316.2278]
GAMMA_GRID = [1.8, 1.16, 1.32, 1.64, 1.128, 1.256, 1.512, 1.1024, 1.2048,
1.4096]
TARGETS = ["gene_MAPK1"]
AFFINITIES = [5]
SAMPLES  = [1]
IA_RATIOS = [100]
KERNEL = 2
PROBABILITY_ESTIMATES = 0
KAPPA_VALUE = 1
```

In the literature there are several approaches that describe the storage of the meta-information and models themselves e.g. ModelDB, a model management system [3] or FGLab, a machine learning dashboard [4]. The main aim of the ExCAPE project is to research algorithms for creating the models rather than the annotation and storage of models themselves. Due to the diversity of the partners and the diversity of the algorithms used in the project we decided not to put limits on the partners on how to represent and store this information, but have a requirement that it should be annotated and stored in a machine readable format, (e.g. JSON, XML-like or Python as exemplified above). This is particularly important with respect to the industrial life sciences partners. Trying to standardise a meta-information format for the pharma industry would clearly be a large and lengthy undertaking. Thus, whilst it would potentially be a valuable exercise, it is completely out of scope for a project such as ExCAPE. Consequently partners will use their own preferred in-house strategies as they see fit.

An important part of meta-information is the annotation of bounds of reasonable use of the model, also known as the applicability domain. There are two main approaches to applicability domain specification: confidence estimation of predictions and a bounded applicability domain in molecular descriptor space. How to represent these in meta-information are both out of the scope of the project, and also at the current stage there is no clear consensus in the community on the best solution [8].

Calibration is the cornerstone for comparison of the different models, thus we put it into a specific section of this deliverable. There are two main sources of diversity in ExCAPE: the various machine learning approaches falling into classification or regression, and diversity of the features that can be used to build a model. SecondA third source of diversity is chemical compounds that can be described by various biological and *in silico* chemical descriptors. The latter can be encoded as graph, a 2D representation of the compound [5], or a 3D representation of the compound. Calibration guarantees the ability to compare the diverse results by mapping the algorithm output into probability distributions derived from identical conditions used to split the data and used in the modeling process. We are chiefly going to apply two calibration algorithms, namely: Platt scaling [6] and conformal prediction [7], developed in the deliverables of the project - D1.4 and D1.9.

In general, all models are learned independently and later transferred to the calibration stage (if required), and further to ensembling schemes. WP3 provides the data and helps with the analysis of the best performing method, best performing feature space and/or combination of the two coming from the ensembling algorithm. WP2 will participate in the optimization of the calibration stages, and if needed will help with HPC enablement of those stages taking into account the data storage system and memory related questions.

Finally, an important aspect is that of defining the applicability, that is the region of the space of compounds for which the prediction is meaningful. The characterization of the applicability domain can be approached in (at least) two ways: via the expected reliability of the prediction or in terms of the molecular descriptor space (novelty detection) [8]. At the present stage, there is no consensus on the best solution and, in any case, both methods are outside the scope of the project.

3 Model calibration

There are two main classifier-calibration methods widely used: Platt scaling and isotonic regression. Conformal prediction [7] can be also treated as calibration and it is covered in detail in the project deliverables D1.4 and D1.9

3.1 Platt scaling

Platt scaling [6] or Platt calibration transforms the output of a binary classification model into a probability distribution over classes. It was developed originally for support vector machines, but it can be also applied to other classification methods [9]. Platt scaling produces probability estimates $P(y = 1|x) = \frac{1}{1+\exp(Af(x+B)}$ via logistic transformation of the classifier scores *f*(x), where A and B are two scalar parameters that are learned by the algorithm. The parameters A and B are estimated using a maximum likelihood method that uses the same training set as the original classifier *f*. To avoid overfitting to this set, a held-out calibration set or cross-validation can be used, but Platt additionally suggests transforming the labels *y* to target probabilities: $T_+ = \frac{N_+ +1}{N_- +2}$ for positive samples and $T_- = \frac{1}{N_- +2}$ for negative ones. Here, N_+ and N_- are the number of positive and negative samples. This transformation is then followed by Bayes' rule applied to a model of out-of-sample data that has a uniform prior over the labels. Platt suggested to use the Levenberg–Marquardt algorithm to optimize the parameters, but a Newton algorithm was found to be more numerically stable [10]. The parameter derivation should be done using nested inner and outer cross-validation to avoid unwanted bias. The same cross-validation sets can be used for model validation (external validation) and parameter selection.

A Platt-scaling R package was developed by the University of Linz in the ChemBioBridge project (Flanders Grant IWT 135122) and has been successfully applied for calibrating the models built using several chemical descriptors and diverse machine learning algorithms. The calibrated predictions were successfully used in internal Janssen Pharmaceutica projects for target deconvolution and chemical library design. This package will be applied and further extended in the ExCAPE project if needed.

3.2 Isotonic regression

Platt scaling as described above is not well suited for some types of classifier output, especially when not enough training data is available [9]. Zadrozny and Elkan[11] applied isotonic regression to calibrate SVMs, Naive Bayes, boosted Naive Bayes, and decision trees. For a prediction f_i and given true classes/labels y_i the basic assumption in isotonic regression is $y_i = m(f_i) + \epsilon_i$ where m is an isotonic function. For a training set (f_i, y_i) isotonic regression is trying to find isotonic function $m(f_i)$ such that $\hat{m} = argmin_z \sum\left(y_i - z(f_i)\right)^2$.

The Pair-Adjacent Violators (PAV) algorithm [12] is one of the algorithms that find a stepwise constant solution for the isotonic function. Several variants based on isotonic regression were published, e.g. an ensemble of near isotonic regression methods eliminates the main limitation of isotonic regression, i.e. the monotonicity assumption of the predictions [13]. Another example is a combination of the parametric and non-parametric methods on the basis of isotonic regression as pr oposed in smooth isotonic regression [14].

3.3 Other classifier calibration options

The above-mentioned methods are the most popular classifier calibration methods that can be applied to the single class binary classification. For multiclass models there are several ways to either convert the problem into the set of binary tasks with further pairwise coupling [15] or a Dirichlet calibration where the outputs of the classifiers are transformed into a set of Beta-distributed random variables and then combined into a realization of Dirichlet-distributed random vectors [16].

4 Model ensembling

Ensembling of the output of the different models can achieved via methods listed below. One requirement for ensembling is that the output of the models should be either calibrated probabilities or Z-scores. After ensembling, the models are evaluated using the same metrics and the same folds used in the modeling process.

- **Mean or median**: Input can be probabilities or Z-score.
- **Geometric mean**: Input should be probabilities. Produces probabilities if probabilities are given, otherwise scores. Does not use any reference data.
- **Maximum probability**: Input should be probabilities. The method takes the probability with the highest confidence, meaning furthest away from 0.5 (<0.5 inactive, >0.5 active).
- **Inverse geometric mean**: $\tilde{y}_{i,j} = 1 - \sqrt[M]{\prod_{m=1}^{M} (1 - \hat{y}_{i,j,m})}$. Input must be probabilities, produces probabilities. Does not use any reference data.
- **Noisy OR**: $\tilde{y}_{i,j} = 1 - \prod_{m=1}^{M} (1 - \hat{y}_{i,j,m})$. Input must be probabilities. Does not use any reference data.
- **Bayes- like** [1]: Input must be probabilities. Produces probabilities and uses all training data as a reference set calculating:

$$\tilde{y}_{i,j} = 1 - \prod_{m=1}^{M} \hat{y}_{i,j,m} / (\prod_{m=1}^{M} \hat{y}_{i,j,m} + \prod_{m=1}^{M} (1 - \hat{y}_{i,j,m})(\frac{g}{1-g})^{M-1})$$

where $g = p(y_{ij} = 1|j)$ is the relative frequency of active compounds.

5 Metamodel proposal for the ExCAPE project

Meta-information for the models developed in ExCAPE project should contain all needed information to reproduce the experiment., but we do not attempt to fix the format for this ahead of time. Other requirements are:

a. Each model should produce calibrated output (via e.g. isotonic regression or Platt scaling);

b. Fusion of the models should be done on different types of compound fingerprints (features) to find the influence of them on the model quality;

c. An ensemble of the models should be composed using one of the methods proposed in Section 4 of this deliverable.

There are many possible combinations of the calibration types and ensemble types, but at the current stage of the project we propose to follow only two calibration schemes and all listed ensembling schemes. However, there are some specificities of the machine learning algorithms that should be taken into account for calibration, namely:

- **Group Factor Analysis** as described in D1.3 and publications [4] and [18] is a method that can combine different heterogeneous features as input and identify the significance of every group of them. The main limitation of the method is that it cannot handle missing features, and thus is limited to compounds with only complete descriptors in different feature spaces. This method does not require ensembling of the models from different descriptors, thus it can be used for testing the diversity of the chemical *in silico* descriptors. The output of Group Factor Analysis is a predicted distribution over the latent factors and loading matrix learned from the observed data. Posterior mean/expectation of the posterior samples for the distribution is used as the prediction by GFA.

- **Deep Learning** outputs real values and requires scaling for model ensembling and comparison.
- **Potential SVM** [18] output is probabilistic, but still requires an additional calibration step to be in the same scale of comparison as other models.
- **Conformal Prediction** outputs for every label or class a corresponding p-value, these p-values can be transformed into probability estimates [19], or used as variants of the conformal prediction, Venn prediction [20], or directly used for calibration.

6 Conclusion

The proposed guidelines in this document describe the three main steps for meta-model creation in the ExCAPE project: Meta-information – storage of the information related to the building of the model for reproducibility of the experiment; Calibration – an approach needed to correctly compare different machine learning algorithms and different compound encodings used and, finally, Ensembling - how to combine the output of the different models to obtain better predictions.

7 Bibliography

[1] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, “DeepTox: Toxicity Prediction using Deep Learning,” *Front. Environ. Sci.*, vol. 3, no. 80, 2015.

[2] A. Klami, S. Virtanen, E. Leppaaho, and S. Kaski, “Group Factor Analysis,” *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[3] M. Vartak, H. Subramanyam, W.-E. Lee, S. Viswanathan, S. Husnoo, S. Madden, and M. Zaharia, “ModelDB: A System for Machine Learning Model Management,” in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2016, p. 14:1--14:3.

[4] L. Gao, “Evaluating framework for hyperparameter optimization on large deep learning problem,” 2016.

[5] D. Rogers and M. Hahn, “Extended-connectivity fingerprints.,” *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–54, May 2010.

[6] J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Adv. large margin Classif.*, vol. 10, no. 3, pp. 61–74, 1999.

[7] U. Norinder, L. Carlsson, S. Boyer, and M. Eklund, “Introducing conformal prediction in predictive modeling for regulatory purposes. A transparent and flexible alternative to applicability domain determination,” *Regul. Toxicol. Pharmacol.*, vol. 71, no. 2, pp. 279–284, 2015.

[8] M. Mathea, W. Klingspohn, and K. Baumann, “Chemoinformatic Classification Methods and their Applicability Domain,” *Molecular Informatics*, vol. 35, no. 5. pp. 160–180, 2016.

[9] A. Niculescu-Mizil and R. Caruana, “Predicting good probabilities with supervised learning,” *Proc. 22nd Int. Conf. Mach. Learn. ICML 05*, no. 1999, pp. 625–632, 2005.

[10] H. T. Lin, C. J. Lin, and R. C. Weng, “A note on Platt’s probabilistic outputs for support vector machines,” *Mach. Learn.*, vol. 68, no. 3, pp. 267–276, 2007.

[11] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” *Proc. eighth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. KDD 02*, pp. 694–699, 2002.

[12] M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman, “An Empirical Distribution Function for Sampling with Incomplete Information,” *Ann. Math. Stat.*, vol. 26, no. 4, pp. 641–647, 1955.

[13] M. P. Naeini and G. F. Cooper, “Binary Classifier Calibration using an Ensemble of Near Isotonic Regression Models,” 2015.

[14] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado, “Smooth isotonic regression: a new method to calibrate predictive models.,” *AMIA Jt. Summits Transl. Sci. Proc. AMIA Summit Transl. Sci.*, vol. 2011, pp. 16–20, 2011.

[15] T. Hastie and R. Tibshirani, “Classification by pairwise coupling,” *Ann. Stat.*, vol. 26, no. 2, pp. 451–471, 1998.

[16] M. Gebel, “Multivariate calibration of classifier scores into the probability space,” no. April, 2009.

[17] S. Virtanen, A. Klami, S. A. Khan, and S. Kaski, “Bayesian Group Factor Analysis,” *arXiv Prepr. arXiv1110.3204*, vol. XX, p. 9, Oct. 2011.

[18] S. Hochreiter and K. Obermayer, “Support vector machines for dyadic data.,” *Neural Comput.*, vol. 18, no. 6, pp. 1472–510, Jun. 2006.

[19] V. Vovk, I. Petej, and V. Fedorova, “From conformal to probabilistic prediction,” *Artif. Intell. Appl. Innov.*, pp. 221–230, 2014.

[20] V. Vovk and I. Petej, “Venn-Abers predictors,” pp. 1–18, 2012.