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 D1.1

Report: Challenge Report

Due date of deliverable: 30.11.2015
Actual submission date: 21.01. 2016

Start date of Project: 01.09.2015 Duration: 36 months

Responsible Consortium Partner: JP
Contributing Consortium Partners: IT4I, IMEC
Name of author(s) and contributors: Vladimir Chupakhin (JP), Jan Martinovic (IT4I), Tom Ashby (IMEC)
Revision: V3
Internal reviewer(s): Nina Jeliazkova (IDEA)

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)
1	18.12.2015	Deliverable due	All project partners
2	19.07.2016	Correction of the documents (review)	All project partners
3	27.03.2017	Editing of the document prior to Review meeting	All project partners

Glossary

R&D	Research And Development
EEG	Electroencephalogram
ECG	Electrocardiogram
ADMET	Absorption, Distribution, Excretion, Metabolism
HPC	High-Performance Computing
HTS	High Throughput Screening
QSAR	Quantitative Structure Activity Relationship
ML	Machine Learning

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: alignment of the academic research with industry challenges	None

Table of contents

1 Executive summary

The pharmaceutical industry aims to deliver drugs to patients to cure disease or ease pathological symptoms. In the diversity of the experimental techniques used computational methods show significant benefits and help to reduce the costs, time and animal usage in the R&D pipeline. Computational chemogenomics predicts how compounds interact with the targets in a biological system. Compared to molecular informatics that is more focused on a single target, the purpose of chemogenomics is to find dependencies in the diversity of the data, revealing the hidden dependencies and hidden cross-target information to better predict the targets where compounds will be active.

There are strong challenges in computational chemogenomics in the pharmaceutical industry. The current academic research is more focused on small datasets compared to the highly imbalanced sparse data available in the industry. This report describes the prior art on the topic and the differences between the challenges already described in the literature, primarily academic research and the current challenges of big pharma, represented in the ExCAPE project by AstraZeneca and Janssen Pharmaceutica. This description of the challenges is a vital line that formulates the challenges in the application that is being used to drive the research into machine learning algorithms in the ExCAPE project. As well as describing the application area being used to anchor the algorithmic research, this report also gives a brief summary of some of the technology trends that may have an impact on the application.

This deliverable was originally slightly late due to unexpected discussions across work-packages about which parts to involve as a challenge. It was also substantially updated after the (less formal) month 9 review at the EC.

2 Introduction – Aim

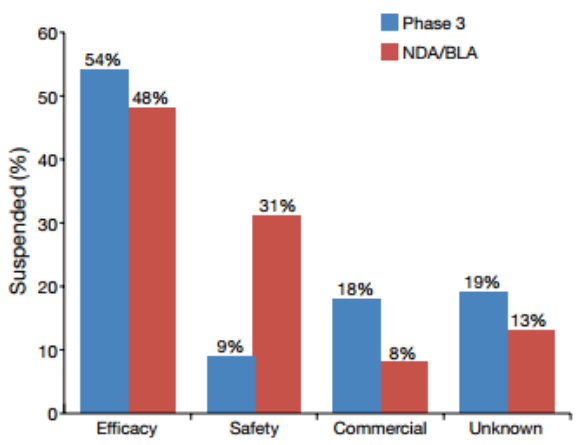

Figure 1. Root-cause analysis for 359 phase 3 and 95 New Drug Application/ Biologic license application suspended programs. A program was designated as 'suspended' when conclusive evidence had been gathered regarding a company's plans to discontinue development or communications with regulators were not reinitiated for several years [1].

Patients are waiting for new and better drugs. The pharmaceutical industry is currently facing issues that slow down the approval of new drugs: increasing safety requirements by governmental agencies, more complex diseases linked with lifestyle, a high attrition rate of

the compounds in clinical trials due to efficacy (48%) or toxicity (31%) [1] (Fig. 1). It is believed that cost-efficacy treatment reimbursement will result in a major change in the forthcoming future of the pharma industry, thus only drugs that are highly effective for a disease will be reimbursed by governmental and insurance agencies [2].
To improve drug efficacy and safety, biological biomarkers for a disease or condition are currently used to design a drug from early beginnings to clinic. Biological markers include gene expression, metabolite biomarkers, interaction with proteins and any other traceable biological endpoints (heart rate, EEG, ECG, etc). Data on different biological levels gives a full picture of drug influence on an organism with major ones being absorption, distribution, metabolism, excretion, toxicity (ADMET) and efficacy of treatment. A common way of encoding of this type of information is a numerical vector – sometimes it is called a biological fingerprint (biofingerprint).

The average cost of design and development of a drug is €930 million ($1.2 billion) and it takes 9 to 13 years for a compound to reach the patient [3]. At every step of drug design and development a control is needed to measure the drug against biological effects obtained in a patient. This can cut costs, save time, energy resources and usage of laboratory animals.

Gene expression profiles of compounds attempt to mimic the real organism answer yet have significant limitations: it is a secondary outcome, not a direct compound exposure; understanding of gene expression profiles requires significant work and is not fully automatized [4]. Interaction of the compound with proteins is considered more reliable information, but only a part of a human proteome is available in the form of biological assays. ADMET parameters also include cell- or animal-based toxicities, drug-drug interactions, adverse reactions and many other biological end-points. It is important to mention that those procedures are very costly and require also significant usage of laboratory animals.

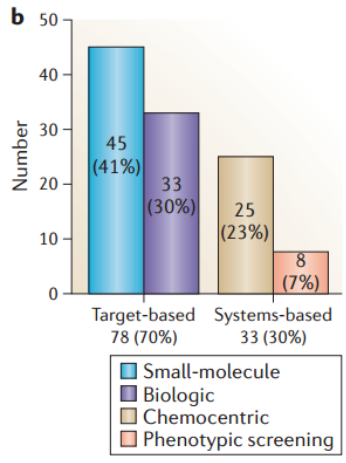

Figure 2 The majority of first-in-class drugs were discovered through target-based approaches with slightly more small-molecule drugs than biologics. Most drugs that were discovered through systems-based approaches originated from a known compound or compound class (that is, a chemocentric approach), and only a few were based on a phenotypic screen as defined in [5].

It is believed that we can reduce the drug attrition rate by using for initial screening a cell-based phenotypic screen instead of biochemical assays where the protein is extracted. A significant amount of compounds in clinical trials in 1990-2013 were designed using the target-based paradigm [5] (Fig. 2). Thus we move to a system that is closer to the real world [6] – drug to disease target is not a direct path and there are a lot of barriers on the way. In our opinion the truth, as usual, lies somewhere in the middle.

The purpose of computational chemogenomics is to deliver models for prediction of chemical compound – target interaction. In our case, the target is not only the protein

interaction but also an ADMET property. This will speed up the drug design and development at all stages: from HTS design to clinical trial candidate selection. Computational chemogenomics is not a new field, but in ExCAPE we would like to focus on the challenges specific for the pharmaceutical industry previously not addressed: chemogenomics models for highly imbalanced big sparse data, HPC enablement of the algorithms for their development and questions concerning the transparency and interpretability of the models. In WP1 we will be focusing on the prototype development, which will be further taken in WP2 for HPC enablement and optimization from computer science perspective, and WP3 is data preparation and running the resultant pipeline to obtain the benchmark. In current whitepaper we will focus on above mentioned computational challenges and prior knowledge available.

3 Large scale computational chemogenomics challenges: an industry perspective

In industry and the academy computational prediction and analysis are intensively used on all levels of biological data: linking compounds to biological end-points, analysis of compounds, targets and assays for similarities, ranking and classification. Predictions can be used from the beginning of the development workflow – to increase the hit rate during chemical library design for high throughput screening (HTS) [7] – up to analysis of the ADMET parameters and efficacy [8]. The database of all compounds versus all proteins or biological end-points, is called the chemogenomics matrix. There is little distinction if the columns (end-points) in this matrix are interaction with proteins, expression of a particular gene, hepatotoxicity or rate of blood-brain-barrier permeability. Currently, data for entries in this matrix are sparse with very few know positive and negative entries for compounds, thus most entries are unknown, and a lot of resources would be needed to fill the gaps with experimental data. Models that are trying to fill the empty space in the chemogenomics matrix often called chemogenomics models. Chemogenomics models also can expand a chemogenomics matrix to "uncharted space" – where no information about compound-target interaction is known, but can be inferred using e.g. target-target similarities.

Common pharma industry databases contain around 5-10 million compounds with biological data for several thousand biological end-points. For proteins there is a large overlap with public domain databases (~5000 targets [9]), but not in chemistry and biological endpoints. The human proteome has 17,294 confirmed proteins [10], while the theoretical approximation is around 20,000-25,000 proteins. Thus, it is only 35 % of the human proteome that has associated chemical compounds – compounds that can interact with those proteins and modulate their function. All known chemical compounds represent around 30 million compounds [11], either synthesized or extracted from natural resources, but only a small portion of those compounds have linked biological information. Some of the ADMET assays are widely used in academy and industry, but still the overlap is significantly less than for the proteins.

The "zero-th" stage of any drug design and development project begins with defining the target and/or biological fingerprint required for curing a disease. High throughput screening of a compound library is often done using a phenotypic assay, where a cell line is used as a disease model, or a biochemical assay, where direct interaction of the compound with the target is measured. Screening is completed using single concentrations of compounds. This step generates data for hundred thousands of compounds. Found hits are confirmed by varying the compound concentration to find out the stability of the induced effect and its level to remove false positives. In some situations it may be possible to apply the lean drug design approach which generates less data since there is no need to search for initial hits because there are many chemical tools known for this target and it is easy to generate new intellectual property on their basis.

Computational analysis and prediction of biological endpoints on the basis of compound structure is a branch of life sciences that combines chemistry, biology and computer science (machine learning and data mining). Building a model that connects compound chemical structure and biological end-point can be described as *f (structure) = activity* in the form of

regression or classification. Unsupervised methods are also widely used separately and in combination with supervised ones.

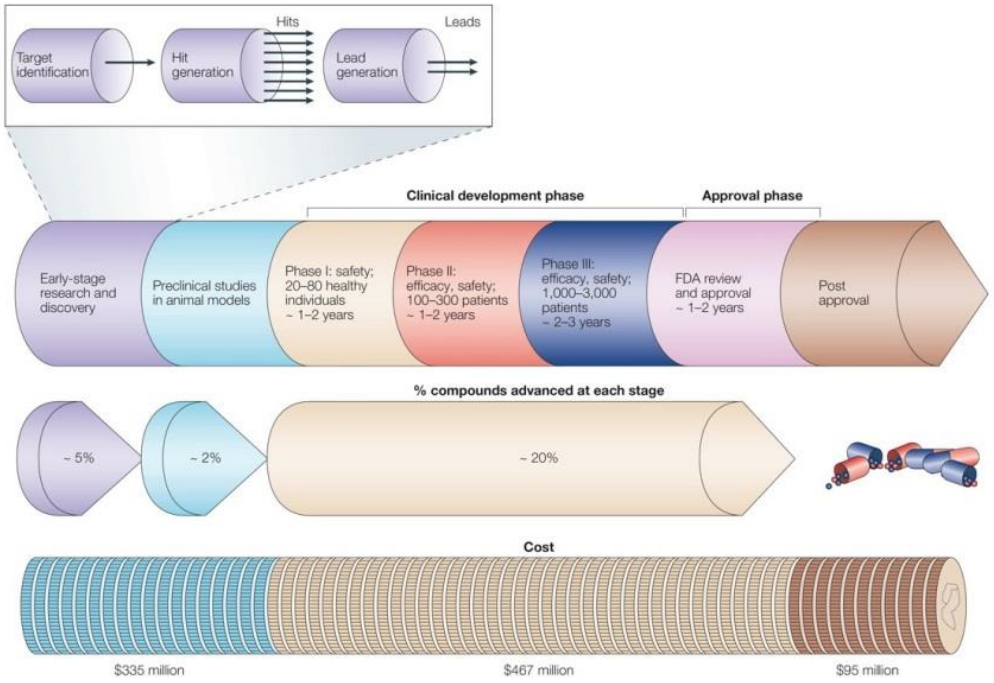

Figure 3 Drug design and development workflow from year 2004.

3.1 Limitations of current chemogenomics data and models

3.1.1 Limitations of the chemogenomics data

Industry data versus public data: Most of the data in the public domain are small datasets mainly composed of active compounds, while in industry datasets are big with a prevalence of inactive compounds. There is a very limited number of public ADMET databases, because it is a tremendous task to standardize and put various toxicity and ADME data together, as it was done e.g. in the eTOX project [12]. In 2010 there was a significant change in the scope of publically available data with the creation of the ChEMBL [9] and PubChem [11] databases. For a long time academic researchers had had access to a limited amount of data, but currently access to imbalanced and highly imbalanced big datasets similar to those used in industry is available to some extent (millions of compounds with ratio of active to inactive 1 to 100-1000), albeit with somewhat less data quality that in industry.

Data used in the pharmaceutical industry is intrinsically noisy: Every response of a living organism to its environment is not linear and depends on many factors, thus all biological and pharmacological data is noisy with a significant amount of false positive/negative data [13];

Data standards: There is a lack of common data standards that dates back, probably, to the beginning of drug design and discovery, but still common nowadays. There are discrepancies between protein and gene identifiers, compound encodings and biological end-point

completeness. There is a strong call from the community for data standardization and research reproducibility [14] in all fields of the life sciences, supported for example by the OpenPHACTS foundation [15] or the European Bioinformatics Institute, but standards are not yet accepted by everyone.

Bias toward "easy" targets: A significant amount of the chemogenomics data obtained in academia and industry is biased towards a portion of the target space because development of the new target is very hard due to the lack of the chemical tool compounds or antibodies, also development of the specific biomarkers for the unknown target require significant accumulated knowledge. The Protein Structure Initiative is aiming for large-scale structural annotation of proteins, thus trying to fill the knowledge gap [16] and enable work on more targets.

3.1.2 Limitations of the chemogenomics models

Chemogenomics model quality: Chemogenomics models heavily depend on the input data. Many developed methods capture only a single class for the targets. There is only very limited research on building models for high dimensional multi-label and multi-class data. Also, most of the published research is aimed at constructing models that are considered as a black box without providing practical insight to end-users: biologists and chemists.

Diversity of feature space: There is no consensus on how a compound should be encoded for a perfect model; a fragment- or pharmacophore-based descriptor, graph, 3D properties of a compound, a biological fingerprint or some combination there-of. This is a trial and error exercise and almost always project dependent.

Chemists, biologists and molecular informatics specialists are the main end users for chemogenomics predictions. They require slightly different features from their chemogenomics models, but do have some common requirements: (a) model-building algorithms that are able to handle big and imbalanced data with multiple classes and labels: millions of compounds with thousands of classes with a sparse feature space; (b) tools that can work with different compound encodings – various feature space inputs, ideally based on mixed heterogeneous descriptors [17]; (c) chemogenomics models that are highly reliable with clear confidence estimation; (d) chemogenomics models that are be able to identify new chemical scaffolds – compounds that share little similarity with known feature space; (e) ideally, chemogenomics models that cover the full proteome using target-target relationships, or, in case of ADMET end-points – assay-assay relationships; (f) other desirable but not required or hard-to-get features: incremental update of the models with a linear computational cost; and model transparency.

3.2 Large-scale chemogenomics models: the state of the art

The current *big data* revolution calls for new machine learning algorithms capable of dealing with imbalanced multi-label and multi-class big data with a large and often sparse and heterogeneous feature space. Individual workstations are not currently capable of working with such an amount of data. Data available in big pharma companies shares significant similarities with web-scale data from Google, Facebook or Twitter. Significant research in building large scale machine learning systems was done by those companies for recommendation, classification and labeling. There are several approaches for building such systems: developing new algorithms for effective model optimization in parallel; developing highly optimized machine learning algorithms for specific processor architectures: GPU [18] or FPGA are good examples; developing machine learning algorithms that can reduce computational time – e.g. budgeted or cost-efficient machine learning – for example, various approximated SVM solvers: LaSVM [19], PEGASOS [20], AMM [21], LLSVM [22]; ensembling of models is a common solution for dealing with big data in chemoinformatics. For example, the search for an optimal solution was done for the prediction of the solubility of compounds in dimethyl sulfoxide using various chemical descriptors and different machine learning algorithms [23].

Most of the machine learning algorithms for supervised classification are sensitive to the class balance. This can be solved using undersampling [23], oversampling [24] or ensembles of models [25]. Some algorithms can intrinsically handle imbalanced data, for example LIBLINEAR SVM [26], decision trees [27], Cost sensitive classifiers [28] or Influence Relevance Voter (IRV) [29]. The issue of dataset imbalance is common for many real life datasets – text classification, speech and image recognition, and disease diagnosis [30], and of course the results of an HTS screen, where the hit rate (i.e. the number of active compounds) is usually below 1%. Imbalanced data sets also entail special handling of model quality assessment, as classical ROC AUC is not well suited to models built from such sets; other metrics for model optimization and quality control are needed. For example the area under precision-recall curve, BEDROC, MCC, G-mean or the F1-measure are considered as good quality metrics for models built from imbalanced datasets. One can use a combination of those metrics for building better models, but in the end this is linked to the goal of modeling for the end user – e.g. do we need to have a high recall rate or good overall predictions?

Active learning algorithms attempt to intelligently choose the examples to label next in order to learn well with as few labeled examples as possible – a common use-case for chemogenomics data. Active learning perfectly mimics the logic of compound optimization – originally a small amount of compounds are tested, then a larger set of labels is predicted, a subset of which are tested and submitted back to the model building algorithm for optimization in a stepwise manner. One of the main benefits of active learning is better handling of imbalanced datasets. Examples of the active learning approach for imbalanced chemogenomics datasets include SVM [31] and query-by-bagging active learning algorithm [32].

Another interesting solution for large and imbalanced datasets is online incremental learning, a machine learning approach that learns one instance/batch in a time with further

optimization of the model after the true label of the entry is discovered. It is very efficient to update a model when more information becomes available rather than learning from scratch. An example for computational chemogenomics models is Online SVM (SVMTorch) applied to imbalanced datasets for prediction of biological activities [29].

Many pharmaceutical companies prefer to outsource computational costs required to build the models, and the question of data security and privacy is a very important one, as compound-target interaction is their most valuable asset, after patient private clinical data. Dyadic modeling can use similarity matrices as input for the modeling thus hiding instances of the compounds, as it is impossible to map back the structures of the compounds from similarities. Dyadic modeling uses relationships between the entries as an input space for building a model. This allows the building of multi-target holistic models for multiple proteins. The algorithms are usually amenable to imbalanced datasets. Dyadic modeling requires a large storage space and efficient memory usage for big datasets; the selection of the similarity metric requires some optimization; a combination of similarity matrices is possible via matrix factorization [33] or multiple kernel learning [34]. A portable out-of-the-box solution for fast similarity/relationship search is needed for the end-user for security reasons. There are several examples for prediction of mutagenicity, toxicity and anti-cancer activity [35]–[37], all on small to medium size datasets.

Matrix Factorization and related approaches are very effective and reliable machine learning methods able to help to build a model to make predictions in “unknown” parts of chemical space; they are also an effective search tool for similarities between datasets. There have been several notable successes with matrix factorization: disease-disease associations [38], and integrative and personalized drug design and discovery [39]. Matrix factorization can be used to combine different features and distance matrices to improve model performance [39].

Available computational power is growing every year driven by Moore’s law. A successful example of the use of ever-growing computational power is deep learning, based on artificial neural networks with many hidden layers. It was proposed quite a time ago [40], but was hardly feasible to be deployed and widely until recently. This algorithm is effective for big and imbalanced data. Deep Learning algorithm won the Merck Molecular Activity Challenge on Kaggle [41]. Another advantage of deep learning (and which also applies for most neural network-based algorithms) is an ability to predict several classes simultaneously – authors have shown that in most of the cases multiclass prediction gives a benefit compared to single class models. Recursive deep learning, as a variant of the deep neural network, was also successfully used for aqueous compound solubility, but in a single task setup [42].

3.3 Prediction and model interpretation: the state of the art

Confidence estimation is a very important part of any prediction. Wet-lab and predicted life science data is often grey (a continuous distribution), but practitioners have to make concrete black and white decisions, thus confidence estimation with clear cut-off defined is

very important for a usable model. There is still no consensus on what confidence metric is the best for a QSAR model especially for imbalanced, many-class life science data.

There are several approaches for the estimation of confidence intervals: for example, cross-validation based on leave-one-out or leave-group-out techniques and derived values like Q^2_{LOO} (only external predictions are used), or Q^2_{F1} (where the distance between sets is taken into account), or the concordance correlation coefficient (based on an analysis of the scatter plot of predicted versus real values [43]. Unfortunately, all of them are computationally intensive, which can be problematic for building models on big data.

Several methods have confidence estimation as a part of the machine learning algorithm implicitly – like Laplacian-modified Naïve Bayes Classifiers [44] or logistic regression [45]. Some algorithms have an explicit formulation of confidence estimation as a part of the model, e.g. conformal prediction [46] where the output is a prediction region, not an entity class *per se*. Conformal prediction uses a nonconformity measure for estimating the difference between old and new examples (the calibration set) and the prediction rule is update for every new example in an online manner. Confidence-weighted classification [47] is another example, trained in an online manner with model update based on confidence estimation on the basis of a Gaussian distribution over the weight vectors.

How well-suited a model is for a new dataset is an applicability domain problem that is sometimes solved together with confidence estimation, but which has gained more attention from the molecular informatics community, probably because it is less computationally intensive than full confidence estimation. Among various estimates widely used, criticized and yet proven to be very effective in practice, are a number of nearest neighbor and similarity to the closest point(s) techniques [48], [49]. Non-supervised and supervised machine learning methods can be used to map new datasets to existing training sets thus calculating the “closeness” to the existing model. Generative Topography Mapping can used for classification purposes but can also serve as an applicability domain visualization by mapping new datasets to the model map [50].

Interpretation of a computed model is very important. For example, for a chemist it is important to see which parts of the compound have more influence on activity or on some chemical property, whereas for biologists it is important to see which molecules have the most influence on activity prediction by the model. Most of the machine learning algorithms implicitly has feature influence estimation, but sometimes it is difficult to translate it directly to compound structure. Multiclass and imbalanced classification problems are even less easy to interpret.

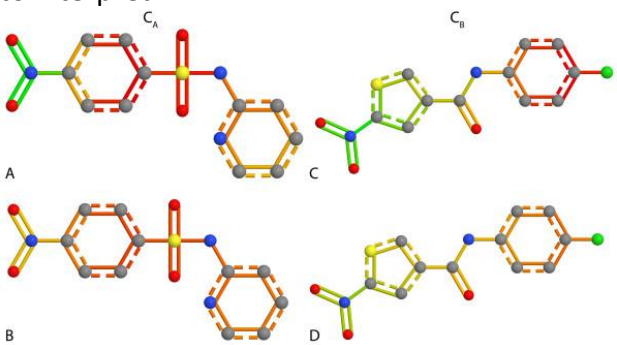

Figure 4. Heatmap of influence of a fragment on compound toxicity based on a linear SVM model (green –toxic, red - non-toxic. Taken from [51].

Compound fragments can be colored according to their influence on the model; for example, the toxicity of a compound predicted by a linear SVM model (Figure 4). However this approach tends to be inefficient in large and very diverse datasets [51]. Due to the nature of the Random Forest algorithm it is possible not only to interpret the model, but also visualize important fragments (features) of the model [52] directly.

Whereas fragment influence on the compound classification can be directly acquired from a model, the influence of the entity is less easy to calculate. Leave-one-out cross-validation can be used not only to estimate model applicability, but also influence points [53], at the cost of being a computationally intensive procedure. SVM models can be interpreted by visualizing the hyper-plane that separate two classes [54]. Other interesting approaches are support *feature* machines [55] that are based on features derived from kernels and support *cluster* machines [56] that are based either on a specific set of features derived from a support vector or clusters of entities – both are easy to interpret on the basis of the hyper-plane separating the entities or cluster of entities. Again, most examples are based on small and single class models, and may not be easily applicable to chemogenomics data.

4 Impact of Technology Developments

The development of Exascale computing, expected sometime in 2020 or thereafter, will signal a major milestone in the development of High Performance Computing and make powerful computing resources available to tackle problems such as chemogenomics modeling. As well as the development of raw computing power, there are various trends in technology around HPC that will have an impact on machine learning applications. Here we review a number that are relevant to Chemogenomics in the context of the ExCAPE project.

4.1 Software Tools and Frameworks

4.1.1 Work Flow software

Scientific workflows are software frameworks to connect together and coordinate other computer applications. Machine learning applications tend to consist of many jobs that are very similar but which differ in terms of their input parameters, such as the subset of the training data to use and the values of certain model parameters such as the number of hidden variables or the amount of regularisation. In addition, there are often chains of data preparation, learning, testing and selection tasks. Consequently, machine learning is likely to benefit from the application of workflow-like concepts.

In the context of the ExCAPE project, an ideal workflow language would be well suited to HPC machines, and well suited to expressing machine learning applications and the particular forms of instantiation of similar short pipelines of tasks that occur in that domain. From an implementation point of view, the workflow system should also make it possible to exploit any redundancy seen in ML learning schemas to enable the best possible performance in an HPC context.

There are many workflow systems available. We discuss here a few pertinent examples. Pegasus is a distributed workflow system from USC [57]. In common with many workflow systems it offers simple programming facilities to connect tasks, automatically handles the communication of data between tasks, and does some logging (provenance information). It also enables the distribution of the workflow execution across multiple machines (resources) at different sites, and has some reliability mechanisms such as detecting failure and restarting tasks. The Pegasus system has the capability to transform a workflow to improve performance and reliability, and to automatically allocate the task executions and data to the different resources available. The dispel4py package[58] is a python implementation of the dispel workflow language. It has the ability to automatically map a workflow and task description in the Python language to particular execution back-ends, namely Apache Storm or MPI clusters. The saves the user from having to implement the connection between the workflow system and the task invocation and makes the workflow specification platform independent (provided a mapping exists for the desired execution platform). Apache Airavata [59] is a workflow system that relies on many components developed by the Apache foundation and is notable for its open governance model. Whilst individual workflow systems are designed to offer portability of workflows expressed in their notation across different computer systems, portability of workflows across workflow systems themselves is not yet standard even though workflow languages are often rather simple. The Common

Workflow Language is a relatively recent development in this area aiming to provide such portability[60].

Workflow systems originally arose in the context of the creation of pipelines of existing tools where the tools themselves were not re-engineered and where the programming constructs offered by the workflow environment required limited sophistication and remained application agnostic. The boundary between the coordination language and subtasks in a project that is also considering programming models and creating the subtasks themselves is obviously much more fluid. In addition the programming sophistication required to achieve high performance and to support the machine learning application domain may be higher than that usually offered by workflow systems. At the same time, the engineering burden suffered by workflow systems to enable portability of pipelines across different computational resources (by ensuring consistent behaviour of component tasks in different environments) is not an issue that needs to be addressed given that the subtasks are not whole applications and are not coupled to their operating environment in the same way, and the ExCAPE programming model does not have such portability constraints. None the less, some concepts from this domain are expected to be relevant.

4.1.2 Java Big Data Ecosystem

The classical Big Data ecosystem revolves around a simplified programming model (Map-Reduce [61]) and open source implementations on the Java platform. The standard tools are Hadoop [62] and Spark [63]. While there have been various machine learning packages that have been built using them, there are a number of disadvantages to this ecosystem in the context of ML and HPC.

The Java platform provides a virtual machine (VM) that can simplify issues around code portability, in that implementations of the VM can be made for various different platforms enabling the application code to be directly reused. However, the VM introduces performance inefficiencies, and displaces the portability problem to the availability of the correct version of a VM on a platform. Performance problems can be resolved to some extent using sophisticated JIT compilers within the VM[64]. However, performance engineering for HPC is often based around a tool chain with static compilation, libraries, profilers and hardware accelerators etc, for which there is no generally accepted alternative in the Java ecosystem.

While it is possible to implement some ML algorithms on top of MapReduce (see the project described in [65] for some examples), that programming model has limitations for the implementation of some patterns that can occur in machine learning applications. For example, sophisticated techniques such as graph-based algorithms or matrix factorisation based on statistical sampling don't fit efficiently into the model. Taking the latter as an example, the distribution of objects and targets across the machine and the patterns of access to them determined by the application-specific pattern of scarce ratings data means that point-to-point communication between nodes is more natural and gives higher performance (see [66] for a discussion). Techniques such as asychronous communication between different parts of the sampling algorithm [67] are also ill-suited to MapReduce.

There are some important high-level concepts that have been promoted by the Java Big Data community. These include resilience at the software level rather than relying on hardware, the ability to scale to extremely large data sets, and ease of programming for algorithm developers. We expect these to be very relevant to the intersection of ML and HPC. However, the use of VMs, the lack of direct support for accelerators and the restricted programming model mean that the software tools are difficult to re-use directly. This is reflected for example in the fact that the Deep Learning community have mostly avoided the Java Big Data tools and opted instead for tools much more closely aligned with those used in traditional HPC.

4.1.3 High Performance ML Packages

There are a number of publically released machine learning software packages that have been engineered for high performance. The most prominent are those developed for neural networks, usually with an emphasis on the use of GPUs as accelerators. These include TensorFlow [68], Theano [69], etc. These implementations usually concentrate on single accelerator or single node performance, with little in the way of multinode implementations being publicly available.

Given that Neural Networks are one of the families of algorithms to be investigated in the project, it may be possible to reuse one of the existing packages, and this will be investigated. However, the lack of integration with multinode execution engines and the emphasis on one particular type of accelerator may prevent use of any of the well supported packages. For example, GPUs work best with very regularly structured computations, and the sparse feature sets from the chemogenomics domain are unlikely to fit onto this approach well, and this may also make it hard to use software packages that have been built with such a structured execution engine in mind.

4.2 Trends in HPC Hardware

The development towards Exascale is a huge effort by the HPC community. One of the main aims of the ExCAPE project is to develop algorithms that will be able to scale machine learning to make use of very large compute resources. These algorithms will have to take some characteristics of HPC hardware into account, although probably at a fairly high level of abstraction due to their complexity. The main trends that are likely to impact the performance of the algorithms are the development of the memory hierarchy, the number of threads of execution, and the increasing use of specialised compute elements and hardware accelerators.

4.2.1 Memory

Alternative memory technologies to standard DRAM and flash have been under development[70], with Intel XPoint Memory being an interesting example that has reached commercialisation. The main result will likely be the introduction of a layer in the memory hierarchy between DRAM and flash. What the exact dimensions of this layer are and how exactly it will be managed are still being worked out. In the case the layer becomes another

operationally transparent level in the hierarchy, the impact on the programming of ML algorithms will be minimal beyond the possible need to take an extra hierarchy layer into account when reasoning about memory behaviour. If the handling of the layer is explicit (e.g. in the form of memory mapped files) then implementations will have to exploit this to be able to take advantage of it. Still, it is not clear how much impact this will have on the design of the algorithms themselves.

On-package memory made of stacked DRAM (such as High Bandwidth Memory[71]) will also have a similar potential impact on the implementation of software, but through less drastic changes. Its introduction will change the dimensions (capacity and bandwidth) of one of the layers rather than inserting a new layer in the memory hierarchy. It will also likely be possible to treat on package memory as either a transparent caching layer, or as an explicitly managed pool of memory. The latter is currently the standard approach for on package memory for accelerators.

4.2.2 Accelerators

Accelerators and custom compute engines have been a common feature of the HPC landscape for at least a decade. GPUs in particular have become popular and are currently the dominant form of acceleration, although many-core products (e.g. Xeon Phi) from Intel are strong contenders. The use of accelerators in HPC is paralleled by the use of accelerators in the Neural Network sub-field of machine learning, so there is overlap between the two fields already.

In addition to GPUs and many-core accelerators, FPGAs are a potentially useful accelerator for machine learning workloads. However, they have yet to achieve widespread popularity in the HPC community despite some pilot systems[72].

4.2.3 Number of threads

Due to the limits in scaling processor speeds, the general trend in hardware is towards more parallelism to provide more performance[73]. This will translate to needing a huge number of threads to achieve exascale performance. The consequence for algorithm designers is that they should design algorithms that can make use of massive parallelism and ideally that are tolerant of high latency communications outside of groups of threads that co-exist on the same socket or node.

4.3 Parallel Computing and Performance Engineering Challenges

HPC machines are increasingly complex and consequently increasingly difficult to program. As outlined above, the hardware trends are towards increasing complexity of memory hierarchies, specialised compute units in accelerators and ever increasing parallelism to provide performance. Added to this is the increased likelihood of component failure. This translates into complexity in the tools and programming approaches that programmers and performance engineers are required to use to achieve reasonable efficiency from the hardware platforms that are available.

For example, an HPC machine consisting of multiple nodes that are multi socket with a GPU accelerator could easily require the combination of CPU vector instructions, MPI, OpenMP, and CUDA to achieve reasonable performance. Within a node this implies at least software management of the memory on the GPU card and probably some awareness of the NUMA effects across the sockets, as well as awareness of the impact of caching and prefetching within the CPUs. Across nodes it requires explicit handling of the communications between ranks, likely including issues such as message vectorisation and buffer handling.

As HPC machines develop towards exascale, it is highly likely that the architectures will get more complex. Consequently there will be an increased need for cooperation between algorithm designers and performance engineers, and a selection of tractable methodologies to enable the cooperation. This is likely to include algorithm prototyping in high level languages by machine learning experts, followed by a hand over of the prototype for re-engineering by performance experts.

4.3.1 Simulation

Computer system simulation is mostly used for the dimensioning and design space exploration of computer hardware, but is also used by performance engineers to locate bottlenecks and give an indication of how much return any given code improvement is likely to provide, especially in cases where the target hardware is not available for direct testing.

Due to the inherent trade-offs between accuracy, simulation time and size of simulated system, there are many different simulators occupying different niches in the trade-off space. Cycle-accurate simulators for CPUs are often used by computer hardware companies internally (and occasionally provided externally) to perform detailed architectural exploration, but they are usually limited to simulations of very small sequences of instructions running on a handful of cores. High level architectural simulators such as GEM5[74]and Sniper[75] allow much longer simulation runs and higher core counts with a relatively low cost in accuracy, allowing the exploration of CPU design on a larger scale, multi-socket simulations to model node behaviour, and useful levels of feedback for performance engineers based on analyses that are difficult to do using existing hardware profiling alone.

Simulation on a larger scale than an individual compute node requires some fairly drastic simplification of the architectural model and some loss of accuracy. Examples of large scale simulators aimed at more traditional HPC include (SST [76], Dimemas [77], SimCan[78], etc.). Accurate, large multi-node simulations would clearly be desirable in the context of performance engineering for traditional HPC, where fairly close coupling of a large number of nodes is often the norm for physics-like mathematical models. In such cases, scaling behaviour as the node count increases is of interest, and it is informative to simulate systems that are larger than those available. However, the need is much less for machine learning where individual jobs are much more likely to run on a single node or small number of nodes. In addition, the heterogeneity and interaction between the individual jobs in an ML workload is expected to be low or zero, and thus data centre level simulations for cloud workloads (e.g. CloudSim [79]) are also unlikely to be in the sweet spot for simulation trade-offs. As there is less or even no need for scaling analysis for single jobs, profiling on existing systems is likely to provide the vast majority of inter-node behavioural information required. Thus, the type of simulators that we expect to be of most use within the ExCAPE project are

individual node and socket simulations that enable code performance analysis, and simulations or analyses of accelerator performance.

5 Conclusion

In this document we have given an overview of issues in drug development and how modeling can help deal with them, outlined the state of the art in QSAR and Chemogenomics modeling, and given a detailed description of the modeling challenges faced by industry practitioners. We have also given an overview of various technology in this context.

Empirical activity data and other endpoints are very expensive to obtain, thus there is a major drive to develop methods capable of using all available data to make the most accurate possible models across all relevant targets. This subsequently entails dealing with the imbalance in the data sets and the wide diversity of features and end-points. Furthermore, the scope of information available from the models should improve, moving from what are often black-box predictions to actionable insights about the effect of different parts of compounds on the final prediction and the level of confidence that the model can ascribe to predictions.

Within the ExCAPE project we will work on creating machine learning algorithms that can leverage the huge amounts of computational power available from future HPC platforms. In this way we will contribute to solving problems such as chemogenomics modeling. In addition to the amount of computing power available, we have described in this report various other technology trends that we expect to have a major impact on this application area, and more widely in machine learning applications as a whole.

6 Bibliography

[1] M. Hay, D. W. Thomas, J. L. Craighead, C. Economides, and J. Rosenthal, “Clinical development success rates for investigational drugs,” vol. 32, no. 1, 2014.

[2] H. G. Eichler, B. Bloechl-Daum, E. Abadie, D. Barnett, F. Konig, and S. Pearson, “Relative efficacy of drugs: an emerging issue between regulatory agencies and third-party payers,” *Nat Rev Drug Discov*, vol. 9, no. 4, pp. 277–291, 2010.

[3] PhRMA, “2013 Biopharmaceutical Research Industry Profile,” *Biopharm. Res. Ind. Phrma*, pp. 1–78, 2013.

[4] M. R. Fielden and T. R. Zacharewski, “Challenges and Limitations of Gene Expression Profiling in Mechanistic and Predictive Toxicology,” *Toxicol. Sci.*, vol. 60, no. 1, pp. 6–10, Mar. 2001.

[5] J. Eder, R. Sedrani, and C. Wiesmann, “The discovery of first-in-class drugs: origins and evolution.,” *Nat. Rev. Drug Discov.*, vol. 13, no. 8, pp. 577–87, Aug. 2014.

[6] J. a Lee, M. T. Uhlik, C. M. Moxham, D. Tomandl, and D. J. Sall, “Modern phenotypic drug discovery is a viable, neoclassic pharma strategy.,” *J. Med. Chem.*, vol. 55, no. 10, pp. 4527–38, May 2012.

[7] P. M. Petrone, A. M. Wassermann, E. Lounkine, P. Kutchukian, B. Simms, J. Jenkins, P. Selzer, and M. Glick, “Biodiversity of small molecules - A new perspective in screening set selection,” *Drug Discovery Today*, vol. 18, no. 13–14. pp. 674–680, 2013.

[8] L. G. Valerio, “In silico toxicology for the pharmaceutical sciences,” *Toxicology and Applied Pharmacology*, vol. 241, no. 3. pp. 356–370, 2009.

[9] a P. Bento, A. Gaulton, A. Hersey, L. J. Bellis, J. Chambers, M. Davies, F. a Krüger, Y. Light, L. Mak, S. McGlinchey, M. Nowotka, G. Papadatos, R. Santos, and J. P. Overington, “The ChEMBL bioactivity database: an update.,” *Nucleic Acids Res.*, vol. 42, no. 1, pp. D1083-90, Jan. 2014.

[10] M.-S. Kim, S. M. Pinto, D. Getnet, R. S. Nirujogi, S. S. Manda, R. Chaerkady, A. K. Madugundu, D. S. Kelkar, R. Isserlin, S. Jain, J. K. Thomas, B. Muthusamy, P. Leal-Rojas, P. Kumar, N. a Sahasrabuddhe, L. Balakrishnan, J. Advani, B. George, S. Renuse, L. D. N. Selvan, A. H. Patil, V. Nanjappa, A. Radhakrishnan, S. Prasad, T. Subbannayya, R. Raju, M. Kumar, S. K. Sreenivasamurthy, A. Marimuthu, G. J. Sathe, S. Chavan, K. K. Datta, Y. Subbannayya, A. Sahu, S. D. Yelamanchi, S. Jayaram, P. Rajagopalan, J. Sharma, K. R. Murthy, N. Syed, R. Goel, A. a Khan, S. Ahmad, G. Dey, K. Mudgal, A. Chatterjee, T.-C. Huang, J. Zhong, X. Wu, P. G. Shaw, D. Freed, M. S. Zahari, K. K. Mukherjee, S. Shankar, A. Mahadevan, H. Lam, C. J. Mitchell, S. K. Shankar, P. Satishchandra, J. T. Schroeder, R. Sirdeshmukh, A. Maitra, S. D. Leach, C. G. Drake, M. K. Halushka, T. S. K. Prasad, R. H. Hruban, C. L. Kerr, G. D. Bader, C. a Iacobuzio-Donahue, H. Gowda, and A. Pandey, “A draft map of the human proteome.,” *Nature*, vol. 509, no. 7502, pp. 575–81, May 2014.

[11] Y. Wang, T. Suzek, J. Zhang, J. Wang, S. He, T. Cheng, B. a Shoemaker, A. Gindulyte, and S. H. Bryant, “PubChem BioAssay: 2014 update.,” *Nucleic Acids Res.*, vol. 42, no. 1, pp. D1075-82, Jan. 2014.

[12] M. Cases, M. Pastor, and F. Sanz, “The eTOX library of public resources for in silico toxicity prediction,” *Molecular Informatics*, vol. 32, no. 1. pp. 24–35, 2013.

[13] T. Kalliokoski, C. Kramer, and A. Vulpetti, “Quality Issues with Public Domain Chemogenomics Data,” *Mol. Inform.*, vol. 32, no. 11–12, pp. 898–905, Dec. 2013.

[14] M. Baker, “Independent labs to verify high-profile papers,” *Nature*, pp. 1–5, 2012.
[15] A. J. Williams, L. Harland, P. Groth, S. Pettifer, C. Chichester, E. L. Willighagen, C. T. Evelo, N. Blomberg, G. Ecker, C. Goble, and B. Mons, “Open PHACTS: Semantic interoperability for drug discovery,” *Drug Discovery Today*, vol. 17, no. 21–22. pp. 1188–1198, 2012.
[16] K. Khafizov, C. Madrid-Aliste, S. C. Almo, and A. Fiser, “Trends in structural coverage of the protein universe and the impact of the Protein Structure Initiative.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 10, pp. 3733–8, 2014.
[17] Y. J. Tseng, A. J. Hopfinger, and E. X. Esposito, “The great descriptor melting pot: mixing descriptors for the common good of QSAR models.,” *J. Comput. Aided. Mol. Des.*, vol. 26, no. 1, pp. 39–43, Jan. 2012.
[18] Q. Liao and J. Wang, “GPU accelerated support vector machines for mining high-throughput screening data,” *J. Chem. …*, pp. 2718–2725, 2009.
[19] A. Bordes, Ş. Ertekin, J. Weston, and L. Bottou, “Fast Kernel Classifiers with Online and Active Learning,” *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, 2005.
[20] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, “Pegasos: Primal estimated sub-gradient solver for SVM,” *Math. Program.*, vol. 127, no. 1, pp. 3–30, 2011.
[21] Z. Wang, N. Djuric, K. Crammer, and S. Vucetic, “Trading representability for scalability: adaptive multi-hyperplane machine for nonlinear classification,” *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 24–32, 2011.
[22] K. Zhang, L. Lan, Z. Wang, and F. Moerchen, “Scaling up Kernel SVM on Limited Resources: A Low-rank Linearization Approach,” *Aistats*, vol. XX, pp. 1425–1434, 2012.
[23] I. V Tetko, S. Novotarskyi, I. Sushko, V. Ivanov, A. E. Petrenko, R. Dieden, F. Lebon, and B. Mathieu, “Development of dimethyl sulfoxide solubility models using 163,000 molecules: using a domain applicability metric to select more reliable predictions.,” *J. Chem. Inf. Model.*, vol. 53, no. 8, pp. 1990–2000, Aug. 2013.
[24] C. Y. Chang, M. T. Hsu, E. X. Esposito, and Y. J. Tseng, “Oversampling to overcome overfitting: Exploring the relationship between data set composition, molecular descriptors, and predictive modeling methods,” *J. Chem. Inf. Model.*, vol. 53, no. 4, pp. 958–971, 2013.
[25] L. Rokach, “Ensemble-based classifiers,” *Artif. Intell. Rev.*, vol. 33, no. 1–2, pp. 1–39, Nov. 2009.
[26] G. Hinselmann, L. Rosenbaum, A. Jahn, N. Fechner, C. Ostermann, and A. Zell, “Large-scale learning of structure-activity relationships using a linear support vector machine and problem-specific metrics.,” *J. Chem. Inf. Model.*, vol. 51, no. 2, pp. 203–13, Feb. 2011.
[27] L. Han, Y. Wang, and S. H. Bryant, “Developing and validating predictive decision tree models from mining chemical structural fingerprints and high–throughput screening data in PubChem,” *BMC Bioinformatics*, vol. 9, no. 1, p. 401, 2008.
[28] A. C. Schierz, “Virtual screening of bioassay data.,” *J. Cheminform.*, vol. 1, p. 21, Jan. 2009.
[29] S. J. Swamidass, C. A. Azencott, T. W. Lin, H. Gramajo, S. C. Tsai, and P. Baldi, “Influence relevance voting: An accurate and interpretable virtual high throughput screening method,” *J. Chem. Inf. Model.*, vol. 49, no. 4, pp. 756–766, 2009.
[30] W. J. Lin and J. J. Chen, “Class-imbalanced classifiers for high-dimensional data,” *Briefings in Bioinformatics*, vol. 14, no. 1. pp. 13–26, 2013.

[31] M. K. Warmuth, J. Liao, G. Ratsch, M. Mathieson, S. Putta, and C. Lemmen, “Active learning with support vector machines in the drug discovery process,” *J Chem Inf Comput Sci*, vol. 43, pp. 667–673, 2003.

[32] Y. Fujiwara, Y. Yamashita, T. Osoda, M. Asogawa, C. Fukushima, M. Asao, H. Shimadzu, K. Nakao, and R. Shimizu, “Virtual screening system for finding structurally diverse hits by active learning.,” *J. Chem. Inf. Model.*, vol. 48, no. 4, pp. 930–40, Apr. 2008.

[33] X. Zheng, “Collaborative Matrix Factorization with Multiple Similarities for Predicting Drug-Target Interactions Categories and Subject Descriptors,” pp. 1025–1033.

[34] N. C. V. Pilkington, M. W. B. Trotter, and S. B. Holden, “Multiple Kernel Learning for Drug Discovery,” *Mol. Inform.*, vol. 31, no. 3–4, pp. 313–322, Apr. 2012.

[35] S. J. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi, “Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity,” *Bioinformatics*, vol. 21, no. SUPPL. 1, 2005.

[36] P. Willett, D. Wilton, B. Hartzoulakis, R. Tang, J. Ford, and D. Madge, “Prediction of ion channel activity using binary kernel discrimination,” *J. Chem. Inf. Model.*, vol. 47, no. 5, pp. 1961–1966, 2007.

[37] B. Chen, R. F. Harrison, K. Pasupa, P. Willett, D. J. Wilton, D. J. Wood, and X. Q. Lewell, “Virtual screening using binary kernel discrimination: Effect of noisy training data and the optimization of performance,” *J. Chem. Inf. Model.*, vol. 46, no. 2, pp. 478–486, 2006.

[38] M. Žitnik, V. Janjić, C. Larminie, B. Zupan, and N. Pržulj, “Discovering disease-disease associations by fusing systems-level molecular data.,” *Sci. Rep.*, vol. 3, p. 3202, 2013.

[39] M. Ammad-Ud-Din, E. Georgii, M. G??nen, T. Laitinen, O. Kallioniemi, K. Wennerberg, A. Poso, and S. Kaski, “Integrative and personalized QSAR analysis in cancer by kernelized Bayesian matrix factorization,” *J. Chem. Inf. Model.*, vol. 54, no. 8, pp. 2347–2359, 2014.

[40] J. Hochreiter and J. Schmidhuber, “Untersuchungen zu dynamischen neuronalen Netzen,” 1991.

[41] G. Dahl, N. Jaitly, and R. Salakhutdinov, “Multi-task Neural Networks for QSAR Predictions,” *arXiv Prepr. arXiv1406.1231*, pp. 1–21, 2014.

[42] A. Lusci, G. Pollastri, and P. Baldi, “Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules.,” *J. Chem. Inf. Model.*, vol. 53, no. 7, pp. 1563–75, Jul. 2013.

[43] N. Chirico and P. Gramatica, “Real external predictivity of QSAR models: How to evaluate It? Comparison of different validation criteria and proposal of using the concordance correlation coefficient,” *J. Chem. Inf. Model.*, vol. 51, no. 9, pp. 2320–2335, 2011.

[44] D. Rogers, R. D. Brown, and M. Hahn, “Using extended-connectivity fingerprints with Laplacian-modified Bayesian analysis in high-throughput screening follow-up.,” *J. Biomol. Screen.*, vol. 10, no. 7, pp. 682–6, Oct. 2005.

[45] A. T. Garc??a-Sosa, M. Oja, C. Het??nyi, and U. Maran, “DrugLogit: Logistic discrimination between drugs and nondrugs including disease-specificity by assigning probabilities based on molecular properties,” *J. Chem. Inf. Model.*, vol. 52, no. 8, pp. 2165–2180, 2012.

[46] 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.

[47] K. Crammer, M. Dredze, and F. Pereira, "Confidence-weighted Linear Classification for Text Categorization," *J. Mach. Learn. Res.*, vol. 13, pp. 1891–1926, 2012.

[48] R. P. Sheridan, "Using random forest to model the domain applicability of another random forest model.," *J. Chem. Inf. Model.*, vol. 53, no. 11, pp. 2837–50, Nov. 2013.

[49] F. Sahigara, D. Ballabio, R. Todeschini, and V. Consonni, "Defining a novel k-nearest neighbours approach to assess the applicability domain of a QSAR model for reliable predictions.," *J. Cheminform.*, vol. 5, no. 1, p. 27, Jan. 2013.

[50] H. A. Gaspar, G. Marcou, D. Horvath, A. Arault, S. Lozano, P. Vayer, and A. Varnek, "Generative topographic mapping-based classification models and their applicability domain: Application to the biopharmaceutics drug disposition classification system (BDDCS)," *J. Chem. Inf. Model.*, vol. 53, no. 12, pp. 3318–3325, 2013.

[51] L. Rosenbaum, G. Hinselmann, A. Jahn, and A. Zell, "Interpreting linear support vector machine models with heat map molecule coloring," *J. Cheminform.*, vol. 3, no. 3, 2011.

[52] V. E. Kuz'min, P. G. Polishchuk, A. G. Artemenko, and S. a. Andronati, "Interpretation of QSAR Models Based on Random Forest Methods," *Mol. Inform.*, vol. 30, no. 6–7, pp. 593–603, Jun. 2011.

[53] G. Tóth, Z. Bodai, and K. Héberger, "Estimation of influential points in any data set from coefficient of determination and its leave-one-out cross-validated counterpart.," *J. Comput. Aided. Mol. Des.*, vol. 27, no. 10, pp. 837–44, Oct. 2013.

[54] D. Baehrens and T. Schroeter, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, 2010.

[55] T. Maszczyk and W. Duch, "Support feature machines: Support vectors are not enough," in *Proceedings of the International Joint Conference on Neural Networks*, 2010.

[56] B. Li, M. Chi, J. Fan, and X. Xue, "Support cluster machine," *Proc. 24th Int. Conf. Mach. Learn. - ICML '07*, pp. 505–512, 2007.

[57] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira Da Silva, M. Livny, and K. Wenger, "Pegasus, a workflow management system for science automation," *Futur. Gener. Comput. Syst.*, vol. 46, pp. 17–35, 2015.

[58] R. Filgueira, A. Krause, M. Atkinson, I. Klampanos, A. Spinuso, and S. Sanchez-Exposito, "Dispel4py: An agile framework for data-intensive eScience," in *Proceedings - 11th IEEE International Conference on eScience, eScience 2015*, 2015, pp. 454–464.

[59] S. Marru, L. Gunathilake, C. Herath, P. Tangchaisin, M. Pierce, C. Mattmann, R. Singh, T. Gunarathne, E. Chinthaka, R. Gardler, A. Slominski, A. Douma, S. Perera, and S. Weerawarana, "Apache Airavata: A framework for distributed applications and computational workflows," *GCE'11 - Proc. 2011 ACM Work. Gatew. Comput. Environ. Co-located with SC'11*, pp. 21–28, 2011.

[60] A. Peter, C. Michael R., T. Nebojša, C. Brad, C. John, H. Michael, K. Andrey, L. Dan, M. Hervé, N. Maya, S. Matt, S.-R. Stian, and S. Luka, "Common Workflow Language, v1.0," Jul. 2016.

[61] J. Dean and S. Ghemawat, "MapReduce : Simplified Data Processing on Large Clusters," *Commun. ACM*, vol. 51, no. 1, pp. 1–13, 2008.

[62] T. White, *Hadoop: The definitive guide*, vol. 54. 2012.
[63] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark : Cluster Computing with Working Sets,” *HotCloud’10 Proc. 2nd USENIX Conf. Hot Top. cloud Comput.*, p. 10, 2010.
[64] M. G. Burke, J. Whaley, J.-D. Choi, S. Fink, D. Grove, M. Hind, V. Sarkar, M. J. Serrano, V. C. Sreedhar, and H. Srinivasan, “The Jalapeño dynamic optimizing compiler for Java,” in *Proceedings of the ACM 1999 conference on Java Grande - JAVA ’99*, 1999, pp. 129–141.
[65] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*. 2011.
[66] T. Vander Aa, I. Chakroun, and T. Haber, “Distributed Bayesian Probabilistic Matrix Factorization. BT - 2016 IEEE International Conference on Cluster Computing, CLUSTER 2016, Taipei, Taiwan, September 12-16, 2016.” pp. 346–349, 2016.
[67] A. Terenin, D. Simpson, and D. Draper, “Asynchronous Gibbs Sampling,” pp. 1–33, 2015.
[68] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: A system for large-scale machine learning,” *Google Brain*, p. 18, 2016.
[69] Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions,” *arXiv e-prints*, p. 19, 2016.
[70] J. Meena, S. Sze, U. Chand, and T.-Y. Tseng, “Overview of emerging nonvolatile memory technologies,” *Nanoscale Res. Lett.*, vol. 9, no. 1, p. 526, 2014.
[71] J. Kim and Y. Kim, “HBM: Memory solution for bandwidth-hungry processors,” in *2014 IEEE Hot Chips 26 Symposium (HCS)*, 2014, pp. 1–24.
[72] R. Baxter, S. Booth, M. Bull, G. Cawood, J. Perry, M. Parsons, A. Simpson, A. Trew, A. McCormick, G. Smart, R. Smart, A. Cantle, R. Chamberlain, and G. Genest, “Maxwell - A 64 FPGA supercomputer,” in *Proceedings - 2007 NASA/ESA Conference on Adaptive Hardware and Systems, AHS-2007*, 2007, pp. 287–294.
[73] P. Kogge and J. Shalf, “Exascale computing trends: Adjusting to the ‘new normal’ for computer architecture,” *Comput. Sci. Eng.*, vol. 15, no. 6, pp. 16–26, 2013.
[74] N. Binkert, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. M. D. D. Hill, D. A. D. A. A. Wood, B. Beckmann, G. Black, S. K. S. K. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, A. Basil, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. M. D. D. Hill, and D. A. D. A. A. Wood, “The gem5 Simulator,” *Comput. Archit. News*, vol. 39, no. 2, p. 1, 2011.
[75] T. E. Carlson, W. Heirmant, and L. Eeckhout, “Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation,” *2011 Int. Conf. High Perform. Comput. Networking, Storage Anal.*, no. September, pp. 1–12, 2011.
[76] H. Adalsteinsson, S. Cranford, D. A. Evensky, J. P. Kenny, J. Mayo, A. Pinar, and C. L. Janssen, “A Simulator for Large-Scale Parallel Computer Architectures,” *Int. J. Distrib. Syst. Technol.*, vol. 1, no. 2, pp. 57–73, Apr. 2010.
[77] S. Girona, J. Labarta, and R. M. Badia, “Validation of Dimemas Communication Model for MPI Collective Operations,” in *Proceedings of the 7th European PVM/MPI Users’ Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 2000, pp. 39–46.
[78] A. Núñez, J. Fernández, J. D. Garcia, L. Prada, and J. Carretero, “SIMCAN: A SIMulator

Framework for Computer Architectures and Storage Networks," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, 2008, p. 73:1--73:8.

[79] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. - Pract. Exp.*, vol. 41, no. 1, pp. 23–50, 2011.