JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

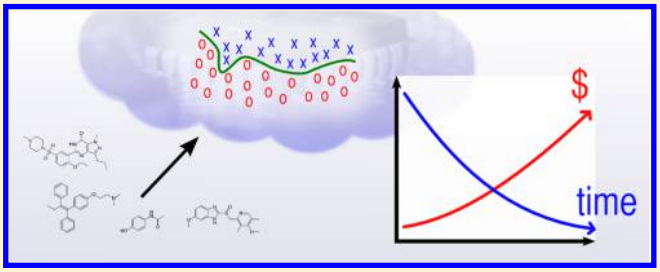
Scaling Predictive Modeling in Drug Development with Cloud Computing

Behrooz Torabi Moghadam,[†] Jonathan Alvarsson,[†] Marcus Holm,[‡] Martin Eklund,[†] Lars Carlsson,[∥] and Ola Spjuth*[,§]

[†]Department of Pharmaceutical Biosciences, [‡]Department of Information Technology, and [§]Department of Pharmaceutical Biosciences and Science for Life Laboratory, Uppsala University, SE-751 24 Uppsala, Sweden

[∥]Computational ADME and Safety, DSM, AstraZeneca Innovative Medicines and Early Development, SE-431 83 Mölndal, Sweden

ABSTRACT: Growing data sets with increased time for analysis is hampering predictive modeling in drug discovery. Model building can be carried out on high-performance computer clusters, but these can be expensive to purchase and maintain. We have evaluated ligand-based modeling on cloud computing resources where computations are parallelized and run on the Amazon Elastic Cloud. We trained models on open data sets of varying sizes for the end points logP and Ames mutagenicity and compare with model building parallelized on a traditional high-performance computing cluster. We show that while high-performance computing results in faster model building, the use of cloud computing resources is feasible for large data sets and scales well within cloud instances. An additional advantage of cloud computing is that the costs of predictive models can be easily quantified, and a choice can be made between speed and economy. The easy access to computational resources with no up-front investments makes cloud computing an attractive alternative for scientists, especially for those without access to a supercomputer, and our study shows that it enables cost-efficient modeling of large data sets on demand within reasonable time.



INTRODUCTION

New high-throughput technologies have transformed the life sciences into a data-intensive domain, with experimental technologies such as massively parallel sequencing and high-throughput screening generating huge amounts of data that needs considerable computational resources for management and analysis including data storage, backup, algorithms, and software, as well as high-performance computing (HPC) for data analysis.[1,2] Organizations lacking an adequate e-infrastructure for data-intensive computing consequently have problems dealing with data from high-throughput platforms.

A recent technology for providing easy access to computational and storage resources is cloud computing,[3] where computing resources are delivered as on-demand services (such as servers, storage, and software) over a network (such as the Internet).[3] An important criteria is that services should be rapidly provisioned and released with minimal management effort or service provider interaction, with users not needing to care about how the things are working behind the scenes. This enables users to pay only for the time that resources are used, without high setup and maintenance costs. Another appealing property of cloud computing is *elasticity*, which refers to the possibility to easily scale the number of resources used up or down as needed.

Cloud computing is based on *virtualization* technology, which enables the cloud to partition the underlying hardware resources into multiple *instances*, in which each instance can, e.g., run its own operating system isolated from other instances. The running instance can be serialized to disk as a *cloud image*, also known as simply an *image*. The user can then choose to start multiple copies of the same image and run them in parallel. A cloud can be private, which means it is managed within the organization it serves, or public, which means it is managed outside of the organization and commonly available to the general public over the Internet. Examples of public clouds include the Amazon Elastic Compute Cloud (EC2)[4] and Windows Azure Services Platform.[5] There are many organizations offering preconfigured cloud images that other people can download and start on their own cloud (public or private) such as the community project Cloud BioLinux.[6]

In the life sciences, cloud computing has been considered an important part of the necessary e-infrastructure in the future[7−10] and for example been applied to proteomics,[11] high-throughput DNA sequencing,[12] RNA-Seq differential expression analysis,[13] and SNP variant calling.[14] Big data deployed in the cloud includes the collections by Amazon Web services, with data from, e.g., the 1000 Genomes Project.[15] Cloud computing not only delivers on-demand, elastic computational and storage resources but a wide range of domain-centered services such as exchange of complete analysis systems for reproducible science[16] and access via programmatic interfaces.[17] There are also several online services with cloud-

Received: September 24, 2014
Published: December 10, 2014

DOI: 10.1021/ci500580y
J. Chem. Inf. Model. 2015, 55, 19−25

based web-portals and workflow systems applied in the life sciences including Galaxy,[18] Taverna,[19] and Yabi.[20]

Drug discovery faces the same data growth as many other scientific domains, and many analyses are computationally demanding and require high-performance computing (HPC).[21−23] One example is when predictive structure−activity relationship models are built from in vitro data. For organizations that have a continuous process of incoming experimental data from in vitro assays, such as AstraZeneca R&D, it is important to provide updated predictive models trained on the latest data. In these cases the modeling time may become a bottleneck, and it is important that the modeling not only delivers high accuracy but also completes within reasonable time. Parallelization of model building on a computer cluster is one way to address the problem, but this process might require higher computational expertise and significant up-front costs. Furthermore, not everyone has access to a computer cluster. A key difference between HPC and cloud computing is that in HPC the architecture is optimized for parallel execution and there is fast interconnect between nodes, while cloud computing delivers better scalability and is more suitable for embarrassingly parallel computations.

In this paper we evaluate the use of cloud computing for scaling ligand-based predictive modeling in drug discovery, demonstrated on Amazon EC2; one of the services available from Amazon Web Services (AWS).[24] This work differs from general virtual screening projects, such as the presented Novartis use of Amazon EC2 orchestrated by commercial products from Cycle Computing,[25] as we focus on the model building that can be used in virtual screening projects and for computational drug safety assessment. D'Agostino et al. have compiled a high-level overview of cloud computing with cost estimates for drug discovery;[26] however, it only briefly mentions QSAR and has focus on commercially available software in cloud environments. We show that for a set of open data sets of varying sizes we can get a substantial speedup in QSAR model building with easy setup and simple administration, and we compare the performance and costs with parallelization on a traditional computer cluster.

METHODS

Data. We downloaded a data set of chemical structures from OpenPhacts,[27] containing 362 035 structures with calculated partition coefficient (logP) values. We randomly extracted 50 000 structures to a test set, and from the remainder we constructed random training sets of sizes 300 000, 150 000, 75 000, and 20 000 chemical structures each.

We also used a data set for the Ames *Salmonella*/microsome mutagenicity assay (Ames test)[28] from a study by Kazius et al.[29] which was downloaded from (http://cheminformatics.org/datasets/bursi/). The data set contained 4337 chemical structures of which 2401 were classified as *mutagen* and 1936 as *nonmutagen*.

QSAR Modeling. Quantitative structure−activity relationship (QSAR) is a method which aims at relating molecular structure of an organic compound to some biological/chemical activity or physical property of interest.[30] It has for example been used to model carcinogenicity,[31,32] toxicity,[33,34] and solubility.[35,36] In QSAR, chemical structures are numerically described by descriptors, which for example can be calculated or measured (such as physicochemical) properties or enumerated fragments. The descriptors and the response values make up a data set, which is then subjected to statistical model building. QSAR models are commonly used in the early stages of drug discovery, for example to provide decision aid in toxicity and ADME (absorption, distribution, metabolism, excretion)[37,38] profiling.

In our study we used the signature molecular descriptor,[39,40] which has been shown to perform well in other QSAR problems,[41−43] and used this to train a support vector machine (SVM).[44] We used the implementation of the signature molecular descriptor in The Chemistry Development Kit (CDK)[45,46] and the single process SVM implementation libsvm[47] as well as the parallel version piSVM.[48]

We trained two types of SVM types, c-SVC and ε-SVR with a radial basis function (RBF) as the kernel. This requires two parameters to be tuned; γ and C, which we did using a grid search over a set of parameter combinations with 10-fold cross validation. We optimized the prediction accuracy in the case of c-SVC and root-mean-square deviation (RMSD) in the case of ε-SVR.

Computational Resources. Amazon Web Services (AWS) was chosen as the cloud provider since it is one of the largest, has advanced tooling for programmatic and manual access to services, and has previously been used in the life sciences.[2,12,49] There are different types of instances available on Amazon EC2 (see Table 1), each consisting of virtual CPUs (vCPU) with performance specified in EC2 Compute Units (ECU) where one ECU provides the equivalent CPU capacity of a 1.0−1.2 GHz 2007 Opteron or 2007 Xeon processor.

Table 1. Various Amazon EC2 Instance Types Tested[a]

type	memory (GiB)	vCPU	ECU	network performance	cost per hour ($)
m1.large	7.5	2	4.0	moderate	0.24
m1.xlarge	15.0	4	8.0	high	0.48
m2.4xlarge	68.4	8	26.0	high	1.64
cc1.4xlarge	7.0	8	33.5	high	1.30
cg1.4xlarge	22.5	8	33.5	10 Gb	2.10
cc2.8xlarge	60.5	16	88.0	10 Gb	2.40

[a]vCPU stands for number of virtual CPUs, ECU denotes EC2 computing unit. Information on prices was collected on September 8th, 2013; it should be noted that prices have decreased after the date of this study.

We also used resources on a computer cluster at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX),[50] the high-performance computing center at Uppsala University, Sweden, running a batch queuing system (SLURM) for allocating resources and which is used heavily in bioinformatics.[51] Whereas the cloud instances provided by Amazon are virtual compute cores, the reserved nodes at the UPPMAX cluster were made up of physical cores with exclusive access. Each node contained two Quad-core Intel Xeon 5520 (Nehalem 2.26 GHz, 8 MB cache) processors.

Parallelizing Model Building. We used a cloud computing approach to parallelize SVM model building in two separate ways: (1) to parallelize the grid search and (2) to parallelize the actual model building.

Parallelizing Parameter Tuning in SVM. In order to obtain the optimal SVM parameters γ and C, we distributed a grid search over N cloud instances where C was allowed to take the values 1, 10, ..., 10^4 and γ the values 10, 1, 10^{-1}, 10^{-2}, 10^{-3}. The parallelization process was implemented with input parameters N for the number of cloud instances, the AWS instance type,

DOI: 10.1021/ci500580y
J. Chem. Inf. Model. 2015, 55, 19−25

and the data set to use for training. The system used the AWS Toolkit for Eclipse[52] for interacting with AWS and for the modeling we used the libsvm framework.[47] The source code for the implementation is publicly available from Github.[53] The implemented system performed the following procedural steps:

1. Start *N* cloud instances with a preconfigured cloud image containing libsvm.
2. Copy the data set to each instance using secure copy (scp).
3. A threading system then distributes the jobs between the *N* cloud instances, sets up an SSH connection, starts the modeling via a command line execution, waits for the modeling to finish, and then returns the results.
4. Terminate all instances.

Parallelizing SVM Model Building. In order to parallelize the model building, we fixed the SVM parameter C to 50 and γ to 0.001 (values found to perform generally well in previous studies[41] and on internal data sets at AstraZeneca) and trained models for the four logP data sets of varying sizes using piSVM[54] on several cloud instances offered by Amazon (see Table 1). We also trained models on the same data sets using the high-performance computing resources at UPPMAX.

The main computational workload of SVM model training is in solving a minimization problem. We used PSMO, a parallelized sequential minimal optimization algorithm, implemented in piSVM using Message Passing Interface (MPI) to work in a distributed computing environment. We set up an Amazon Machine Image (AMI) with piSVM and all dependencies, which is publicly available from AWS with id *ami-89c4b4e0*. Experiments on the UPPMAX cluster in order to evaluate performance of piSVM parallelizing showed that piSVM code achieved a 27× speedup over serial libSVM using 32 cores on a 50 000 data set, but less speedup on smaller data sets (see Figure 1). We chose piSVM as it is widely used, open source, and publicly available. There are alternative SVM implementations supporting sparse data representation that can operate on distributed resources,[55] e.g., MIC-SVM[56] is an interesting implementation but the code is not yet publicly available.

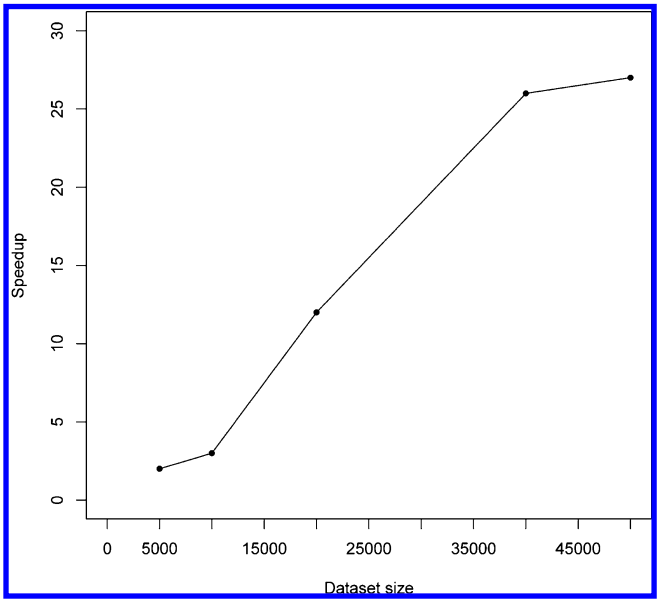


Figure 1. Speedup for different data set sizes (number of molecules) using the parallel SVM implementation piSVM with the PSMO optimization algorithm as a solver on 32 cores when comparing to the sequential implementation of libSVM.

RESULTS

Parallelizing Parameter Tuning in SVM. For the parallelizing of parameter tuning in SVM, we used the Ames mutagenicity data set and the libSVM package on different number of AWS instances of type m1.large and compared with results on the UPPMAX computer cluster with the same number of compute cores. The process was repeated three times, and Figure 2 shows the runtimes in minutes. We see a very good speedup which is expected since this is a trivially parallelized problem. The difference between AWS and UPPMAX is due to the fact that the UPPMAX computer cluster is not virtualized. We also note that the hardware at UPPMAX is a few years old, and with new HPC hardware, the HPC performance might be higher.

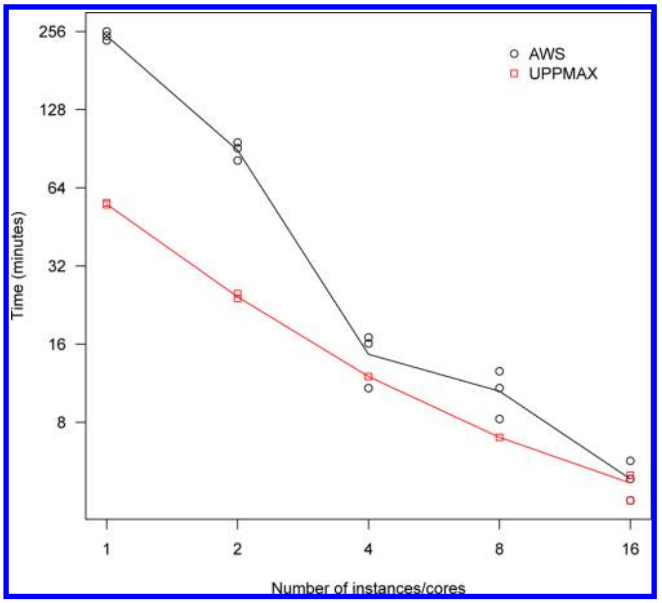


Figure 2. Run times for SVM parameter tuning, implemented as a grid search parallelized on different number of Amazon EC2 instances of type m1.large and different number of UPPMAX processor cores. Each experiment was repeated three times, and the line connects the mean values.

Parallelizing SVM Model Building. For the parallelizing of SVM model building we used the logP data and built models using piSVM[54] on four data sets of varying sizes (20 000, 75 000, 150 000, and 300 000) on six different AWS instance types and measured the runtimes; see Figure 3. We then repeated the experiment on UPPMAX where resources are exclusive to the user and not virtualized, see Figure 4 for a chart of the runtime per number of cores. UPPMAX has a policy that jobs should finish within 7 days, but job extensions are possible. However, when attempting to run the 300 000 data set on 1 core the job was terminated all three times because of much too long runtimes. We note that both increasing resources in terms of a bigger AWS instance as well as more cores on UPPMAX lead to a reduction in the runtime for model building. Table 2 presents the runtime and performance of the resulting models calculated as RMSD between the predicted and observed values for each model. A direct comparison of runtime between AWS

DOI: 10.1021/ci500580y
J. Chem. Inf. Model. 2015, 55, 19−25

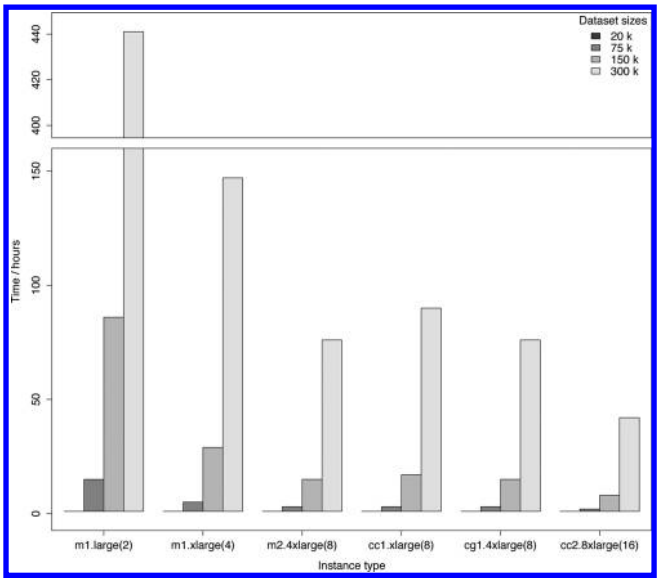


Figure 3. Run times for MPI-based SVM model building on different Amazon EC2 instances.

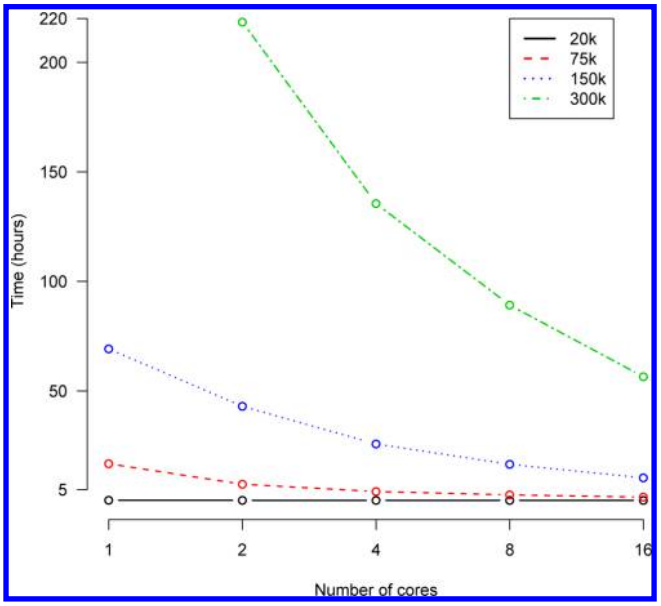


Figure 4. Run times for MPI-based SVM model building on different number of cores on a high-performance cluster without virtualized resources.

instances and number of cores at UPPMAX is not possible due to the differences in AWS instance types; they do not increase compute power linearly and also vary in terms of internal memory. However, we can calculate and compare the cost (in dollar) for each model building, which is included in Table 2. For AWS this cost is readily available in the Amazon Dashboard and billed to the registered credit card each month. For UPPMAX we calculated the actual cost for computer hardware, system administration, electricity, cooling, premises, and management and divided by the number of cores to get the estimate $0.02 per compute core hour (cpuh). This calculation however implies that we have a shared cluster that is fully utilized by a large amount of people, which might be the case in, e.g., a university or a large company. In the case of no existing compute resources, an alternative to using a public cloud provider like AWS would be to purchase a compute server with, e.g., 16 cores to be used exclusively for model building. The cost for such a machine (of same characteristics as an UPPMAX node) is estimated to $10,000. On top of that, system administration and electricity is needed which we estimate to $20,000 per year. During the first year, such a system produces 140 160 cpuh where each cpuh then costs $0.21 if the system was used 100%, or cost per hour of $3.40. If we calculate the hypothetical case of server use for model building for 84 h per week, this gives a resource utilization of 50% and cost per used cpuh will then be $0.42 (disregarding the lower costs for electricity). The cost for these hypothetical scenarios using a dedicated server for modeling are also included in Table 2 (labeled "dedicated"). We note that the performance of the different computer setups are not directly comparable, even if the number of cores are the same, but the objective is to compare the absolute cost of modeling jobs. It is also clear that the hardware and administration costs of a dedicated server will differ between different areas and organizations, and our TCO estimate of $20,000 is based on the assumption that no expertise and resources are available in-house but needs to be procured. This estimate can of course be adjusted to local conditions.

DISCUSSION

Cloud computing is an interesting technology for providing computational resources to drug development for many reasons. First, it requires little hardware investment if using a public cloud provider. Second, users only pay for the resources used during the analysis and can scale the number of resources up and down as needed. For predictive modeling this is appealing as most of the computational power is needed during a limited time, and it is also easy to quantify the cost of building a model. For model builders without access to computer clusters, cloud computing can be a very interesting technology for speeding up model building. In times when more and more organizational and administrative IT-infrastructure and services are being outsourced to cloud providers (a process sometimes referred to as cloudsourcing), it also makes it easier for research and development to take the same path as the formal and legal processes are already established.

In order to obtain an increased throughput with model building there are some requirements on the methodology and tools. The simplest way to parallelize is to, if possible, divide a problem into many pieces and run them in parallel (embarrassingly parallel), such as in the parameter tuning grid search where the runs are independent of each other. The most flexible approach, however, is a parallel implementation of the statistical method that can take advantage of many processors while building one model. In this manuscript we have shown that cloud computing is a viable choice for both these scenarios when doing ligand-based predictive modeling.

As Table 2 shows, the prediction error decreases as we use a bigger data set to train our model. Although the time and cost difference on specific instance types or number of cores is negligible when the data set is small, the difference increases with the size of the data set. As an illustrative example, we were not even able to finish model building on the 300 000 data set on a single core on UPPMAX due to the extremely long runtime. Comparing the costs and times for different AWS instance types shows that the cheapest option for building the model for our largest data set is m1.xlarge, but using cc2.8xlarge makes it 71% faster while the cost is 41% higher. Amazon has a

DOI: 10.1021/ci500580y
J. Chem. Inf. Model. 2015, 55, 19−25

Table 2. Time (h) and Cost (dollars) for Building logP Regression Models for the Specified Dataset Sizes on Different AWS EC2 Instances, UPPMAX Nodes, and Dedicated Servers with 100% and 50% Utilization (Hypothetical Scenario)[a]

data set size	20 000		75 000		150 000		300 000	
model performance (RMSD)	0.60		0.44		0.38		0.33	
	time/h	cost/$	time/h	cost/$	time/h	cost/$	time/h	cost/$
AWS.m1.large	0.11	0.03	14.20	3.41	85.66	20.56	440.30	105.68
AWS.m1.xlarge	0.05	0.02	4.46	2.15	28.01	13.45	146.58	70.36
AWS.m2.4xlarge	0.03	0.05	2.13	3.50	14.33	23.50	75.35	123.60
AWS.cc1.4xlarge	0.05	0.07	2.50	3.25	16.50	20.87	89.58	116.50
AWS.cg1.4xlarge	0.03	0.07	2.11	4.45	14.43	30.31	75.23	158.00
AWS.cc2.8xlarge	0.03	0.08	1.23	2.96	7.83	18.80	41.45	99.48
UPPMAX.2cores	0.07	0.01	7.41	0.30	43.00	1.72	218.37	8.73
UPPMAX.4cores	0.05	0.01	4.08	0.33	25.80	2.06	135.52	10.84
UPPMAX.8cores	0.05	0.02	2.57	0.41	16.52	2.64	89.18	14.27
UPPMAX.16cores	0.05	0.01	1.51	0.48	10.33	3.31	56.50	18.08
dedicated (16 cores, 100% util)	0.05	0.17	1.51	5.07	10.33	34.71	56.50	189.84
dedicated (16 cores, 50% util)	0.05	0.34	1.51	10.15	10.33	69.42	56.50	379.68

[a]Model performance is calculated as RMSD between predicted and observed values.

purchasing option called Spot Instances,[57] where users can bid on spare computational resources and their software is run whenever the bid exceeds the current Spot Price (which varies over time). Sacrificing immediate access with Spot instances can significantly lower the cost of EC2 instances; however, the implementations we use (libSVM and piSVM) do not support checkpointing, rendering spot instances not applicable.

The cost when having access to a shared compute cluster is almost a factor 10 lower than for using AWS; however, this requires hundreds of users and a high degree of resource utilization that makes it infeasible unless a large general need for computations exist in the organization. It might be more interesting to compare with the scenario of buying and administrating a dedicated server for modeling, and here we see advantages of using cloud computing in terms of both costs and flexibility; there is a possibility to easily scale up and down the resources with a cloud provider and still get the benefits of a cheaper model building. Much is of course due to the overhead in system administration of a local server, which is greatly simplified using cloud computing technology. A cloud environment is also easier to work in than a shared computer cluster as there is no need for queueing systems. We use Amazon EC2 as an infrastructure-as-a-service (IaaS) and do not consider the value-add options AWS has for platform-as-a-service (PaaS) and software-as-a-service (SaaS).

In drug development the data security aspects cannot be ignored, and using an external cloud provider requires a well-designed security policy as well as the use of encrypted data transfer with adequate authentication/authorization. This is however well understood by cloud providers, and most of them have high security standards. For most practical problems the security offered by cloud providers is high enough, and they are trusted by large companies including banks.[58] Setting up a secure environment using a cloud provider is however not trivial and requires substantial domain expertise. Nevertheless, we argue that cloud computing should not be dismissed simply because of security concerns. In our study all communication with the AWS instances were carried out over encrypted SSH connections using 1024-bit SSH-2 RSA keys to encrypt and decrypt login information.

It has been shown that nested (double) cross validation produces more conservative performance estimates for QSAR models and avoids overfitting.[59−61] Such methods require more computational power than simpler validation methods, but due to the embarrassingly parallel nature they are highly suitable for cloud computing environments. Another issue affecting the computational load is that QSAR models in many cases require optimization of parameters, e.g., in our work we have a height parameter for the signature descriptor. We have in this work used the parameter settings in Alvarsson et al.[62] as they perform good in the general case, but other types of descriptors have their own parameters that might need optimization. Within a nested CV this will be even more computationally heavy, and the suitability for cloud computing in these settings will be part of our future research program.

■ CONCLUSION

We have shown that cloud computing can increase the speed of model building in computational pharmacology. Cloud computing does not require up-front investments, users pay only for the time that they use the services, and the cost of modeling is easily quantified. If users have access to a supercomputer or computer cluster, this will be the cheapest and fastest option. If not, then cloud computing is an attractive option for establishing a flexible e-infrastructure for predictive modeling that provides a low entrance barrier.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: ola.spjuth@farmbio.uu.se.

Notes

The authors declare the following competing financial interest(s): O.S. and M.E. hold shares in Genetta Soft AB, a Swedish incorporated company.

■ ACKNOWLEDGMENTS

The authors thank Zeeshan Ali Shah, Åke Edlund, and Christofer Bäcklin for valuable technical assistance during the project. Computational resources were provided by Uppsala Multidisciplinary Center for Advanced Computational Science (SNIC-UPPMAX) and SNIC-CLOUD. Funding was provided by the Swedish strategic research program eSSENCE and the Swedish Research Council (VR-2011-6129).

DOI: 10.1021/ci500580y
J. Chem. Inf. Model. 2015, 55, 19−25

REFERENCES

(1) Maclean, D.; Kamoun, S. Big data in small places. *Nat. Biotechnol.* **2012**, *30*, 33−34.
(2) Schadt, E. E.; Linderman, M. D.; Sorenson, J.; Lee, L.; Nolan, G. P. Computational solutions to large-scale data management and analysis. *Nat. Rev. Genet.* **2010**, *11*, 647−657.
(3) Mell, P.; Grance, T. *The NIST Definition of Cloud Computing*; National Institute of Standards and Technology, Information Technology Laboratory, 2009.
(4) Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/ (Accessed Nov. 14, 2014).
(5) Microsoft Windows Azure Services Platform. http://www.windowsazure.com/ (Accessed Nov. 14, 2014).
(6) Krampis, K.; Booth, T.; Chapman, B.; Tiwari, B.; Bicak, M.; Field, D.; Nelson, K. E. Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community. *BMC Bioinformatics* **2012**, *13*, 42.
(7) Stein, L. D. The case for cloud computing in genome informatics. *Genome Biol.* **2010**, *11*, 207.
(8) Gathering clouds and a sequencing storm: why cloud computing could broaden community access to next-generation sequencing. *Nat. Biotechnol.* **2010**, *28*, 1.
(9) Sansom, C. Up in a cloud? *Nat. Biotechnol.* **2010**, *28*, 13−15.
(10) Schatz, M. C.; Langmead, B.; Salzberg, S. L. Cloud computing and the DNA data race. *Nat. Biotechnol.* **2010**, *28*, 691−693.
(11) Mohammed, Y.; Mostovenko, E.; Henneman, A. A.; Marissen, R. J.; Deelder, A. M.; Palmblad, M. Cloud parallel processing of tandem mass spectrometry based proteomics data. *J. Proteome Res.* **2012**, *11*, 5101−5108.
(12) Jourdren, L.; Bernard, M.; Dillies, M.-A.; Le Crom, S. Eoulsan: a cloud computing-based framework facilitating high throughput sequencing analyses. *Bioinformatics* **2012**, *28*, 1542−1543.
(13) Langmead, B.; Hansen, K. D.; Leek, J. T. Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome Biol.* **2010**, *11*, R83.
(14) Langmead, B.; Schatz, M. C.; Lin, J.; Pop, M.; Salzberg, S. L. Searching for SNPs with cloud computing. *Genome Biol.* **2009**, *10*, R134.
(15) Via, M.; Gignoux, C.; Burchard, E. G. The 1000 Genomes Project: new opportunities for research and social challenges. *Genome Med.* **2010**, *2*, 3.
(16) Dudley, J. T.; Butte, A. J. In silico research in the era of cloud computing. *Nat. Biotechnol.* **2010**, *28*, 1181−1185.
(17) Wagener, J.; Spjuth, O.; Willighagen, E. L.; Wikberg, J. E. S. XMPP for cloud computing in bioinformatics supporting discovery and invocation of asynchronous web services. *BMC Bioinformatics* **2009**, *10*, 279.
(18) Afgan, E.; Baker, D.; Coraor, N.; Goto, H.; Paul, I. M.; Makova, K. D.; Nekrutenko, A.; Taylor, J. Harnessing cloud computing with Galaxy Cloud. *Nat. Biotechnol.* **2011**, *29*, 972−974.
(19) Abouelhoda, M.; Issa, S. A.; Ghanem, M. Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support. *BMC Bioinformatics* **2012**, *13*, 77.
(20) Hunter, A. A.; Macgregor, A. B.; Szabo, T. O.; Wellington, C. A.; Bellgard, M. I. Yabi: An online research environment for grid, high performance and cloud computing. *Source Code Biol. Med.* **2012**, *7*, 1.
(21) Pitera, J. W. Current developments in and importance of high-performance computing in drug discovery. *Curr. Opin. Drug Discovery Devel.* **2009**, *12*, 388−396.
(22) Valerio, L. G., Jr.; Choudhuri, S. Chemoinformatics and chemical genomics: potential utility of in silico methods. *J. Appl. Toxicol.* **2012**, *32*, 880−889.
(23) Sun, H.; Xia, M.; Austin, C. P.; Huang, R. Paradigm shift in toxicity testing and modeling. *AAPS journal* **2012**, *14*, 473−480.
(24) Amazon Web Services. http://aws.amazon.com (Accessed Nov. 14, 2014).
(25) Advancing Drug Discovery with HPC Cloud. http://www.hpcwire.com/2014/07/10/advancing-drug-discovery-hpc-cloud/ (Accessed Nov. 14, 2014).
(26) D'Agostino, D.; Clematis, A.; Quarati, A.; Cesini, D.; Chiappori, F.; Milanesi, L.; Merelli, I. Cloud infrastructures for in silico drug discovery: economic and practical aspects. *Biomed Res. Int.* **2013**, 138012.
(27) Williams, A. J.; Harland, L.; Groth, P.; Pettifer, S.; Chichester, C.; Willighagen, E. L.; Evelo, C. T.; Blomberg, N.; Ecker, G.; Goble, C.; Mons, B. Open PHACTS: semantic interoperability for drug discovery. *Drug Discov Today* **2012**, *17*, 1188−1198.
(28) Mortelmans, K.; Zeiger, E. The Ames Salmonella/microsome mutagenicity assay. *Mutat. Res.* **2000**, *455*, 29−60.
(29) Kazius, J.; McGuire, R.; Bursi, R. Derivation and validation of toxicophores for mutagenicity prediction. *J. Med. Chem.* **2005**, *48*, 312−320.
(30) C, H. A Quantitative Approach to Biochemical Structure-Activity Relationships. *Acc. Chem. Res.* **1969**, *2*, 232−239.
(31) Helma, C. Lazy structure-activity relationships (lazar) for the prediction of rodent carcinogenicity and Salmonella mutagenicity. *Mol. Divers.* **2006**, *10*, 147−158.
(32) Helguera, A. M.; Gonzalez, M. P.; Dias Soeiro Cordeiro, M. N.; Cabrera Perez, M. A. Quantitative Structure - Carcinogenicity Relationship for Detecting Structural Alerts in Nitroso Compounds: Species, Rat; Sex, Female; Route of Administration, Gavage. *Chem. Res. Toxicol.* **2008**, *21*, 633−642.
(33) Spycher, S.; Smejtek, P.; Netzeva, T. I.; Escher, B. I. Toward a Class-Independent Quantitative Structure-Activity Relationship Model for Uncouplers of Oxidative Phosphorylation. *Chem. Res. Toxicol.* **2008**, *21*, 911−927.
(34) Guha, R.; Schürer, S. Utilizing High Throughput Screening Data for Predictive Toxicology Models: Protocols and Application to MLSCN Assays. *J. Comp. Aid. Mol. Des.* **2008**, *22*, 367−384.
(35) Johnson, S.; Chen, X.; Murphy, D.; Gudmundsson, O. A Computational Model for the Prediction of Aqueous Solubility That Includes Crystal Packing, Intrinsic Solubility, and Ionization Effects. *Mol. Pharmaceutics* **2007**, *4*, 513−523.
(36) Yan, A.; Gasteiger, J. Prediction of Aqueous Solubility of Organic Compounds Based on 3D Structure Representation. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 429−434.
(37) Munteanu, C. R.; Fernández-Blanco, E.; Seoane, J. A.; Izquierdo-Novo, P.; Rodrguez-Fernández, J. A.; Prieto-González, J. M.; Rabuñal, J. R.; Pazos, A. Drug discovery and design for complex diseases through QSAR computational methods. *Curr. Pharm. Des.* **2010**, *16*, 2640−2655.
(38) Gedeck, P.; Lewis, R. A. Exploiting QSAR models in lead optimization. *Curr. Opin. Drug Discovery Devel.* **2008**, *11*, 569−575.
(39) Faulon, J.-L.; Churchwell, C. J.; Visco, D. P. J. The signature molecular descriptor. 2. Enumerating molecules from their extended valence sequences. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 721−734.
(40) Faulon, J.-L.; Visco, D. P. J.; Pophale, R. S. The signature molecular descriptor. 1. Using extended valence sequences in QSAR and QSPR studies. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 707−720.
(41) Spjuth, O.; Eklund, M.; Ahlberg Helgee, E.; Boyer, S.; Carlsson, L. Integrated decision support for assessing chemical liabilities. *J. Chem. Inf. Model.* **2011**, *51*, 1840−1847.
(42) Norinder, U.; Ek, M. E. QSAR investigation of NaV1.7 active compounds using the SVM/Signature approach and the Bioclipse Modeling platform. *Bioorg. Med. Chem. Lett.* **2013**, *23*, 261−263.
(43) Alvarsson, J.; Eklund, M.; Engkvist, O.; Spjuth, O.; Carlsson, L.; Wikberg, J. E. S.; Noeske, T. Ligand-based target prediction with signature fingerprints. *J. Chem. Inf. Model.* **2014**, *54*, 2647−2653.
(44) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed.; Cambridge University Press, 2000.
(45) Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493−500.
(46) Steinbeck, C.; Hoppe, C.; Kuhn, S.; Floris, M.; Guha, R.; Willighagen, E. L. Recent developments of the chemistry development

DOI: 10.1021/ci500580y
J. Chem. Inf. Model. 2015, 55, 19−25

kit (CDK) - an open-source java library for chemo- and bioinformatics. *Curr. Pharm. Des.* **2006**, *12*, 2111−2120.
(47) Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27.
(48) piSVM. http://pisvm.sourceforge.net/ (Accessed Nov. 14, 2014).
(49) Fusaro, V. A.; Patil, P.; Gafni, E.; Wall, D. P.; Tonellato, P. J. Biomedical cloud computing with Amazon Web Services. *PLoS Comput. Biol.* **2011**, *7*, No. e1002147.
(50) Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX). http://www.uppmax.uu.se (Accessed Nov. 14, 2014).
(51) Lampa, S.; Dahlö, M.; Olason, P. I.; Hagberg, J.; Spjuth, O. Lessons learned from implementing a national infrastructure in Sweden for storage and analysis of next-generation sequencing data. *Gigascience* **2013**, *2*, 9.
(52) AWS Toolkit for Eclipse. https://aws.amazon.com/eclipse/ (Accessed Nov. 14, 2014).
(53) Github repository for AWS interaction. https://github.com/behroozt/Java-AWS-interaction (Accessed Nov. 14, 2014).
(54) Brugger, D. *Parallel Support Vector Machines*; WSI; Graphisch-Interaktive Systeme, Wilhelm-Schickard-Inst. für Informatik, 2006.
(55) Menon, A. K. Large-Scale Support Vector Machines: Algorithms and Theory. M.Sc. thesis, University of California, San Diego, 2009.
(56) You, Y.; Song, S.; Fu, H.; Marquez, A.; Dehnavi, M.; Barker, K.; Cameron, K.; Randles, A.; Yang, G. MIC-SVM: Designing a Highly Efficient Support Vector Machine for Advanced Modern Multi-core and Many-Core Architectures. *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, Phoenix, AZ, May 19−23, 2014; pp 809−818.
(57) Amazon EC2 Spot Instances. http://aws.amazon.com/ec2/purchasing-options/spot-instances/ (Accessed Nov. 14, 2014).
(58) Chopra, M.; Mungi, J.; Chopra, K. A Survey on Use of Cloud Computing in various Fields. *Int. J. of Sci. Eng. Technol. Res.* **2013**, *2*, 2.
(59) Hawkins, D. M. The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1−12.
(60) Reunanen, J. Overfitting in Making Comparisons Between Variable Selection Methods. *J. Mach. Learn. Res.* **2003**, *3*, 1371−1382.
(61) Eklund, M.; Spjuth, O.; Wikberg, J. E. The C1C2: a framework for simultaneous model selection and assessment. *BMC Bioinformatics* **2008**, *9*, 360.
(62) Alvarsson, J.; Eklund, M.; Andersson, C.; Carlsson, L.; Spjuth, O.; Wikberg, J. E. S. Benchmarking study of parameter variation when using signature fingerprints together with support vector machines. *J. Chem. Inf. Model.* **2014**, *54*, 3211−3217.

DOI: 10.1021/ci500580y
J. Chem. Inf. Model. 2015, 55, 19−25