Advances in distributed computing with modern drug discovery

Journal:	*Expert Opinion On Drug Discovery*
Manuscript ID	EODC-2018-0116.R1
Manuscript Type:	Review (Invited)
Keywords:	Distributed computing, Drug discovery, High performance computing, Computational chemistry

Advances in distributed computing with modern drug discovery

Antonio Jesús Banegas-Luna[1], Baldomero Imbernón[1], Antonio Llanes Castro[1], Alfonso Pérez-Garrido[1], José Pedro Cerón-Carrasco[1], Sandra Gesing[2], Ivan Merelli[3], Daniele D'Agostino[4], Horacio Pérez-Sánchez[1,*]

[1]*Bioinformatics and High Performance Computing Research Group (BIO-HPC), Universidad Católica de Murcia (UCAM), Spain.*

[2]*Center for Research Computing, University of Notre Dame, Notre Dame, IN, USA.*

[3]*Institute for Biomedical Technologies, National Research Council of Italy, Segrate (Milan), Italy.*

[4]*Institute for Applied Mathematics and Information Technologies "E. Magenes", National Research Council of Italy, Genoa, Italy.*

*Corresponding author: Horacio Pérez-Sánchez.: Tel: (0034) 968278819. E-mail: hperez@ucam.edu

ABSTRACT

Introduction: Computational chemistry dramatically accelerates the drug discovery process and high-performance computing (HPC) can be used to speed up the most expensive calculations. Supporting a local HPC infrastructure is both costly and time consuming and therefore many research groups are moving from in-house solutions to remote distributed computing platforms.

Areas covered: The authors focus on the use of distributed technologies, solutions and infrastructures to gain access to HPC capabilities, software tools and datasets to run the complex simulations required in computational drug discovery.

Expert opinion: The use of computational tools can decrease the time to market of new drugs. HPC has a crucial role in handling the complex algorithms and large volumes of data required to achieve specificity and avoid undesirable side-effects. Distributed computing environments have clear advantages over in-house solutions in terms of cost and sustainability. The use of infrastructures relying on virtualization reduces set-up costs. Distributed computing resources can be difficult to access, although web-based solutions are becoming increasingly available. There is a trade-off between cost effectiveness and accessibility in using on-demand computing resources rather than free/academic resources. Graphics processing unit computing, with its outstanding parallel computing power, is becoming increasingly important.

HIGHLIGHTS

- Virtual screening is a computational chemistry approach that is currently used to reduce the number of wet laboratory experiments required in drug discovery campaigns. However, it may require a considerable amount of computing power, the availability of which is an issue for many research groups and small- or medium-sized companies.

- Distributed computing is a cost-effective solution in drug discovery research and can be implemented in a number of ways. Grid computing is the most efficient way of

distributing the demand involved in calculations across a network of available computing units. Cloud computing is deployed through virtual hardware resources and is a more flexible approach than grid computing because it can be configured and scaled depending on the complexity of the task.

- Within distributed computing platforms, graphics processing unit computing increasingly has a key role as a result of the parallel nature of the hardware, which increases the throughput in most scientific computing tasks. This can be useful for the computationally demanding tasks involved in drug discovery. However, not all computational chemistry codes are amenable to efficient parallelization.

- Cloud applications are now available to support the deployment of simulations to non-expert users at an early stage. Web servers are available for virtual screening calculations, which implement many different computational drug discovery techniques, although most of these are for ligand-based methods because these are computationally cheaper than structure-based methods.

- Even when using distributed computing approaches, the size of the chemical space is still too large (millions of compounds) to be covered by one single virtual screening technique. Therefore a hierarchical virtual screening approach is often adopted: inexpensive methods (e.g., similarity searching or pharmacophore modeling) are applied first to create small, focused libraries, followed by more computationally expensive methods (e.g., molecular docking and molecular dynamics) to achieve the best results.

- Structure-based virtual screening approaches, such as molecular docking or molecular dynamics, generally require more computational resources than ligand-based methods. However, ligand-based techniques are less accurate than structure-based methods and therefore a trade-off is required when performing computations in distributed computing platforms to obtain the best results according to the resources available.

KEYWORDS

Cloud computing; computational chemistry; distributed computing; drug discovery; grid computing; high performance computing; virtual screening.

1. Introduction

It is now widely accepted that the discovery of new drugs can be aided by the use of computational drug discovery (CDD) techniques and many approved drugs have reached and passed clinical trials with their help. New terms have been added to the vocabulary of researchers, including computer-aided drug design (CADD) [1], computer-aided molecular design (CAMD) [2] and computer-aided molecular modeling (CAMM).

Most drug discovery studies focus on enhancing specific parts of the processes involved in the development of new drugs, which can be divided into three phases: (1) a discovery phase, in which millions of candidate compounds are screened; (2) a selection phase, in which the candidate drugs undergo preclinical research; and (3) an assessment phase, in which the drug is developed and extensive clinical trials conducted. *In silico* solutions are usually carried out in the discovery phase, whereas screening was previously carried out in a laboratory over several years with significant economic costs.

Many computational techniques are now available to study the molecular interactions relevant to drug discovery, such as virtual screening (VS) [3], which is used to simulate a large number of interactions between proteins (also known as receptors and/or enzymes) and small molecule drug candidates (ligands). Docking software is usually tested on protein families where there are the most crystal structures and therefore it is common practice to test many proteins in parallel [4–6]. Side-effects caused by off-target bindings should be avoided and therefore the most promising compounds are usually tested against many other proteins. The docking

conformations that describe the interactions between each compound and the corresponding target are optimized through molecular dynamics (MD) simulations to relax the system and improve the accuracy with which the binding energy is calculated. Molecular dynamics is a physics-based simulation method in which Newton's equations of motion are solved for each atom of the system considering all the forces involved in their interactions [7]. Depending on the number of atoms involved, it can be computationally demanding. Other cheaper techniques, such as chemical similarity and calculating the proximity matrix, are also used in the drug discovery process [8–10].

CDD methods can be used in either a predictive or a prospective way. In predictive CDD, calculations are carried out to process a database of compounds and to anticipate which compounds are most likely to be of interest before their characterization in the laboratory. In prospective CDD, the experimental results obtained after screening compounds in the laboratory are analyzed by CDD methods to try to understand why the compounds were selected. In both these approaches, the complexity of the computations depends on the size of the database and/or the accuracy of the methods used. The use of high performance computing (HPC) techniques is now mandatory in the development of efficient and scalable tools for CDD. Many different HPC approaches can be used, including graphics processing units (GPUs), in-house clusters of computers, remote supercomputers and distributed computing infrastructures.

The creation and maintenance of a local computing facility capable of managing the huge amount of data obtained from state-of-the-art acquisition instruments and simulation tools in drug discovery projects is too expensive for many small- or medium-sized biotechnology laboratories [11] and the use of remote computational services is a cost-effective solution. Remote computing infrastructures provide researchers with the ability to adjust the

computational resources according to their actual requirements, whereas doubling the size of a traditional cluster is expensive and can result in many idle resources during off-peak periods. In a distributed environment, requests to double the amount of access to resources simply involve paying for twice the capacity. Usability is the most crucial aspect in this scenario because end-users require solutions that simplify their activities rather than making them more complex.

We review here the current trends and advances in the application of distributed computing infrastructures to drug discovery. Section 2 reviews the most commonly used tools in ligand- or structure-based virtual screening. The tools discussed here provide a diversity of screening services, such as quantitative structure–activity relationship (QSAR) modeling, docking and molecular docking, and MD. Section 3 introduces the currently available distributed computing environments, including advances in the use of GPU-based HPC platforms. Section 4 gives several examples and case studies showing how large-scale distributed platforms based on the grid and cloud computing paradigms have been used successfully in CDD projects. Section 5 presents our conclusions and discusses the future perspectives for drug discovery in combination with advanced computational techniques.

2. Virtual screening methods

This section reviews some of the methods—such as QSAR, docking and molecular docking, and MD—that are run over distributed computing infrastructures and used routinely in VS calculations.

2.1. QSAR

QSAR models can be defined as regression or classification models that relate several variables, called descriptors, with bioactivity values to predict the activity of new compounds.

These descriptors codify several chemical features of compounds, including their physicochemical properties and experimental measurements. QSAR models are developed using different computational strategies [12]—such as statistical methods, which include multivariate linear regression analysis (MLR) [13], principal components analysis (PCA) [14], partial least-squares (PLS) analysis [15] and linear discriminant analysis (LDA) [16]—or artificial intelligence approaches, such as extreme learning machines (ELMs) [17], neural networks (NN) [18] and support vector machines (SVM) [19].

The applicability of QSAR models in drug discovery requires that the QSAR model has good predictability and provides a physicochemical interpretation of the possible mechanism of action. The development of QSAR models, their application to large datasets, the calculation of quantum chemical descriptors and the use of advanced statistical methods or artificial intelligence approaches demands large amounts of computing resources [20] and mandatory access to HPC resources.

Tetko et al. [21] reviewed various public modeling environments for the development of QSAR models, such as OpenMolGRID [22,23] and its successor Chemomentum [24] for grid computing, and their application to aquatic toxicity [25], acute toxicity [26] and the discovery of HIV-1 protease inhibitors [27]. Other QSAR tools running on distributed computing environments include: Simplex Representation of Molecular Structure–Structural and PhysiCochemical Interpretation [28], which can exploit ensemble predictions, classification techniques and non-linear methods, although the model-building parameters are kept fixed; ISIDA/QSPR [29,30], which also exploits ensemble predictions; and DTC Lab. Software Tools [31], which includes a validation method for the selection of models, is also valid for non-linear techniques.

2.2. Molecular docking

Ligand-based methods (e.g. QSAR, similarity searching, pharmacophore modeling and docking) represent worthwhile solutions in drug discovery. However, QSAR and similarity searching do not take into account knowledge about the binding site within the protein target and this can reduce the accuracy of the calculations. To overcome this issue, structure-based methods are the preferred choice when the 3D structure of the target is known, although they are usually computationally more expensive than ligand-based approaches. In such cases, it is studied how the activity of proteins may be altered when small ligands dock into the well-defined cavities of protein receptors. These ligands can act as molecular switches and control the activity of the protein. For proteins involved in a metabolic pathway related to a disease, artificial ligands can act as drugs [32]. As more metabolic pathways and their associated key proteins are identified, the search for artificial ligands has intensified as a method of improving the treatment of various diseases. The number of known protein structures continues to grow exponentially, a trend increasingly complemented by initiatives in structural genomics [33]. Molecular docking identifies the lead compounds that can bind to a target protein with high affinity [34]. This is achieved by calculating the optimum binding position for each molecule in a large database of potential targets using heuristics and then ranking the database with a scoring function according to the estimated affinity [35].

Docking methods have been investigated for many years and several compounds have been identified and developed as drugs [36]. Several docking methods are currently available—including AutoDock4 [37], AutoDock Vina [38], Glide [39] and Lead Finder [40]—each of which has different scoring functions and optimization methods. All of these methods use an atomistic representation of the protein and the ligand and allow the exploration of thousands of

possible binding positions and ligand conformations in the coupling process [41,42]. As a result, the binding modes for many complexes are reliably predicted. However, unbiased comparative evaluations of the estimations of affinity by molecular docking show little correlation between the measured and predicted affinities over a wide range of receptor–ligand complexes [43], suggesting that more advanced approaches are required to increase the accuracy of the total binding energies. The use of explicit solvent molecules and the addition of dynamic effects to the system may partially circumvent the limitations associated with classical docking simulations.

2.3. Molecular dynamics

The dynamic nature of real biological environments needs to be considered if meaningful predictions are to be made. As it is not possible to apply the most demanding simulations directly to large libraries of compounds, an efficient workflow should start by first using inexpensive techniques, such as QSAR, and then proceed with more expensive molecular docking techniques. The resulting best-ranked drug candidates can then be selected and implemented in more advanced simulations, such as MD. This approach can include the combined effects of the solvent and temperature in the evolution of the system over relatively long time trajectories—that is, on the scale of nanoseconds to microseconds. MD has been used to rank a series of biologically active ligands docked into the herceptin antibody, an efficient biological molecule able to localize malignant cells in patients with breast cancer [44]. The MD simulations allowed ligands that produced an early release from the binding site (during first 100 ns) to be discarded because they were incompatible with a stable interaction.

MD has both advantages and disadvantages in drug discovery research. It has the advantage that the stability of binding sites can be validated before cell-based assays are carried out, but the production of MD trajectories requires large amounts of computational resources,

usually defined as the number of nanoseconds computed per day. Huge efforts have been devoted to speed up simulations and most of the currently available molecular dynamics codes include GPU-accelerated versions. Unfortunately, such codes require previous expertise from the user because molecules with chemical features that are not implemented in the force field parameters need to be optimized/implemented by the user. As an alternative, some web servers running under distributed computing infrastructures are available that may help non-expert users to perform short molecular dynamics simulations in a friendly framework, which might be used as a first proof of concept. A series of representative examples of online solutions are MDWeb [45], the Gromacs server as implemented in Haddock [46], Vienna-PTM [47], CABS-flex [48], Protein structure REFinement via MD [49], MoMA-LigPath [50] and MoSGrid [51]. Most of these solutions offer predefined protocols to guide the preparation of the structure (i.e. the experimental PDB file), which should be "cured" and then transformed to a specific format to run standard molecular dynamics simulations. These listed tools also allow the trajectories produced to be analyzed by monitoring the stability of the whole system using the root-mean-square deviation (RMSD) as well as the geometrical parameters of the binding site (the non-covalent interactions between the target protein and ligand).

One of the major drawbacks of classical MD calculations is the assessment of entropic terms, thermodynamic parameters that are required to determine reliable absolute free energies. There is no universal solution for the refinement of the molecular dynamics protocol and several methods have been proposed to better capture the ligand–protein problem in drug discovery, including free energy perturbations, umbrella sampling, the potential of the mean force and metadynamics. All these additional descriptors may be used to produce a more accurate prediction. However, their computational cost restricts their application to small and rigid

systems, which, in turn, prevents the implementation of molecular dynamics techniques in the servers currently available. Further development is needed before molecular dynamics can be systematically applied to complex systems because it requires sampling at large dimensions to produce accurate values for the entropic and solvation contributions to the free energy [52].

3. Distributed computing environments

Biomedicine was one of the first areas of research [53] to move from the use of in-house computing facilities or single supercomputing centers to distributed infrastructures, in particular grid and cloud platforms. These infrastructures are based on the vision of providing services to users through the sharing of capabilities and resources. The core idea is that any simulation can be achieved using the concept of service: a workstation or a supercomputer represents a computing service, but a database, a domain-specific application or an authentication mechanism are also services. It is possible to find dozens of published definitions of these platforms, which have been modified during their development. Foster [54]—one of the first developers of grid computing technologies—defined a computational grid as an infrastructure of both hardware and software that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities. Foster [54] suggested that the essence of grid computing can be described by three concepts: (1) the coordination of resources that are not subject to centralized control (see Figure 1); (2) access to, and the management of, resources using standard, open, general purpose protocols and interfaces; and (3) the delivery of non-trivial qualities of service. Many years after the spread of this technology began, we can now describe grid computing as a wrapper with which to freely access remote multi-institutional resources, with either dedicated or shared computational time. Grid computing paved the way for cloud computing.

Cloud computing has been defined by the US National Institute of Standards and Technology as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or interaction with the service provider. It is based on three service models (see Figure 2): (1) the software as a service (e.g., Dropbox); (2) the platform as a service (e.g., the Google App Engine); and (3) the infrastructure as a service (i.e., the possibility of managing virtual machines). In contrast with the definition of grid computing, cloud computing can be defined as a way to use the internet to deliver on-demand, scalable, pay-per-use services that rely on computational facilities hosted by a single institution.

Cloud infrastructures can be used for business, commercial or research purposes, but the concept is always based on access to on-demand resources on a pay-per-use basis. By contrast, grid platforms are used for research by virtual organizations [55], which are dynamic sets of individuals and/or institutions sharing a goal to be pursued using the grid resources without any additional charge. The users of a cloud infrastructure are customers, but the users of a grid platform are members of one or more virtual organizations.

The following sections analyze the relevant tools and infrastructures based on the use of non-local computational resources, with a particular focus on architectures that have been used for CDD projects.

3.1. Grid computing

Although initially exploited in the field of high-energy physics, the biomedical community showed an immediate interest in grid computing—for drug discovery applications in particular [56]—and many computational challenges have been met in this context. In addition to

large campaigns for the analysis of biosequences [57], grid computing has been widely exploited in structural biology for large-scale virtual screenings of neglected diseases [58,59] and to determine the molecular dynamics of huge biomolecular systems [60].

3.1.1. Service-based grid computing

Most of the grid infrastructures rely on middleware toolkits—that is, a software layer that lies between the operating system and the applications. The first example of middleware was the Globus Toolkit [61], followed by others such as the Advanced Resource Connector and the Uniform Interface to Computing Resources. Many different platforms have been proposed on top of these middleware components for drug discovery projects and these usually follow one of two approaches.

The first approach relies on general purpose grid service environments coordinated by virtual organizations. This paradigm can be exploited through both service and software frameworks, such as the distributed infrastructure with remote agent control (DIRAC) [62], the distributed analysis environment (DIANE) [63] and the gLite workload management system (WMS) [64], which are able to provide high-level job submission services at the front end and the possibility of interacting with heterogeneous systems at the back end (see Figure 1). The second approach aims to standardize the creation, representation and sharing of the so-called computational workflows that tie a number of software tools together into a single analysis. This is built on application-oriented grid service infrastructures where the focus is on the software executed over the clusters provided within a grid (see Figure 1). This is the case for myGrid [65], a widely used infrastructure for composing pipelines in the field of bioinformatics and structural biology using the Taverna Workbench or the Virtual Laboratory for e-Science (VL-e) [66],

which rely on a grid service infrastructure and use Nimrod-G semantics to compose complex workflows.

Service-based grid computing is usually free, but the user needs to belong to an academic institution. Association with a virtual organization is required for grids relying on such organizations.

3.1.2. Desktop-based grid computing

The other grid approach that gained progressive success in the scientific community is known as desktop grid or volunteer computing because it often relies on the general public to donate resources. Desktop computing and, more generally, volunteer computing, is usually free and only registration is required (see Figure 1).

This paradigm relies on middleware oriented to peer-to-peer architectures. This means that there is no clear distinction between the grid entities owned by users (e.g., a workstation) to access the infrastructure (e.g., a cluster shared via grid middleware) and every resource can act as both a client and a server at any one time. Any user can bring resources into the grid because the installation and maintenance of the software is intuitive, requiring no special expertise, which enables a large number of donors to contribute to the pool of shared resources.

The most popular middleware for desktop grid applications is the Berkeley Open Infrastructure for Network Computing (BOINC) [58], which was originally developed in the context of the SETI@home project [67] to search for extraterrestrial intelligence. The main feature of BOINC is represented by its simple API, which allows easy interaction with other environments, and its great community support. For example, the EDGeS project [68], which aims to create an infrastructure combining the advantages of service (the EGEE infrastructure) and desktop grids, is using BOINC.

3.2. Cloud computing

Grid computing infrastructures are not completely satisfactory when running complex applications, in particular for industrial companies [69], because they do not provide a flexible and reliable environment. Most of the users submit batch jobs to remote clusters with little or no interactivity or the possibility of customizing the environment. The management of the storage of distributed data is complex, especially the administration of geographically dispersed databases. Misconfigured nodes on the grid are common, which results in jobs that fail continuously, emptying the grid queue, an effect known as shrink hole.

By contrast, cloud computing aims to provide the elements of a classical computational infrastructure (from a single workstation running the company website to a fully operational HPC cluster with high-speed network connections) as an on-demand service using virtual machines and virtual clusters, with development and execution frameworks and applications as services (see Figure 2). This paradigm addresses the key demands of creating an easily accessible and flexible environment (see Figure 2), able to support data processing and, in general, to provide everything required to perform complex analyses [70]. Cloud computing vendors such as Amazon [71] and Google [72] provide specialized environments to ease biomolecular data processing in the cloud.

Cloud computing allows the adoption of novel programming models—such as MapReduce and Spark, both of which rely on the Hadoop file system, an open source framework that enables the distributed processing of large datasets—which are exploited for sequence alignment [73–75], drug discovery [76–78] and many other applications [79]. In MapReduce, the input of a computation is split into independent chunks, which are then processed by the map tasks in a parallel manner. The results are then sorted and processed by the reduce tasks to provide the final output. By contrast, Spark works as a toolset for distributed programs and offers

a (deliberately) restricted form of distributed shared memory, facilitating the implementation of iterative algorithms that visit their dataset multiple times.

Some initiatives support the execution of virtual machines on top of grid resources, such as the Worker Nodes on Demand Services (WNoDes) framework [80]. This approach has been exploited for macromolecular characterization and the estimation of free energy in protein–protein docking [81].

3.3. GPUs and distributed computing

GPUs can be used to reduce the execution time of scientific applications [82,83] within distributed computing scenarios. In particular, GPUs have been ranked as one of the platforms with the highest projection for implementing algorithms that simulate complex scientific problems. Since their first appearance, the development of GPUs has been marked by the world of video games, which have reached high levels of popularity as more realism is achieved. In 2006, NVIDIA, the leader in the manufacturer of GPUs, made a breakthrough in the world of HPC when they released the Compute Unified Device Architect (CUDA) development kit. This architecture makes it possible to use GPUs for the development of scientific applications. Six of the ten most powerful supercomputers in the world [84] currently have coprocessor (GPU or similar vector-based devices) accelerators.

In the limited subset of problems for which enough effort can be invested to ensure that specific drug discovery software supports GPU computing, these devices can greatly accelerate calculations. Not all algorithms are good candidates for acceleration [85,86], but scientists are now aware of the benefits that HPC can bring to their research, either by including these hardware platforms in their studies or by directly designing and implementing specific software that can be used on these platforms.

4. Examples of applications

This section discusses examples of drug discovery using different distributed computing architectures and ligand-based virtual screening (LBVS) servers.

4.1 Grid-based examples

One of the most interesting grid projects is Wide in Silico Docking on Malaria (WISDOM) [87], which has the goal of using the computing resources offered by the Enabling Grids for E-Science in Europe (EGEE) project (the largest grid project in Europe developed for the Large Hadron Collider at CERN in Geneva) to select drug-like molecules active on a biological target to fight malaria [88]. Malaria was chosen because tropical diseases often suffer from a lack of research due to the cost of bringing new drugs to market. The project developed *in silico* approaches to VS, mainly using resources from the EGEE project, but also from EUChinaGrid and TWGrid in Asia, EUMEDGRID in Africa, and OSG and Digital Ribbon in the USA [59]. Several of the drug-like molecules selected *in silico* have been confirmed by *in vitro* tests to be active inhibitors and the most promising molecules have been patented [89].

An infrastructure similar to that used in WISDOM was also used to perform a large VS project (see Figure 3) aimed at identifying new drugs for potential variants of the influenza A virus [58]. About 300,000 compounds selected from the ZINC database and an *ad hoc* chemical combinatorial library were screened against eight variants of neuraminidases predicted by homology modeling. The distributed analysis environment was used as the job dispatcher and a portal was developed to visualize the achieved outcome to make the docking conformations and scoring results available to biologists. Using thousands of CPU cores on a grid, a six-week long, high-throughput screening activity was accomplished, performing >100 CPU core years of calculations, producing around 600 gigabytes of results, and identifying about 100 compounds for further biological analysis and testing (see Figure 3).

Among the grid initiatives oriented toward structural biology analysis and relevant to drug discovery, the European Model for Bioinformatics Research and Community Education (Embrace) network of excellence [90] has been particularly important in the promotion, development and implementation of standards for the interoperability of resources in the life sciences community. Another example is the BIOINFOGRID project [91], which aims to exploit the existing EGEE infrastructure for large-scale modeling and simulations of biological problems and in which docking and molecular dynamics had a primary role. Many attempts have been made to perform molecular dynamics on grid [92,93] and cloud [94,95] platforms, although the only approach that gained some popularity was the Hamiltonian Replica Exchange method [96].

In addition to dedicated virtual organizations, several *ad hoc* grid systems have been proposed for the execution of bioinformatics and drug discovery workflows [97]. One of the most effective service-based environments for running workflows is Taverna [98]. Many projects oriented toward the annotation, modeling and simulations of biological macromolecules, such as myGrid [65] and the VL-e [66], rely on this infrastructure. Taverna was designed to combine distributed grid services and/or local tools into complex analysis pipelines to tackle a wide range of scientific research [99]. Once constructed, the workflows become reusable resources. Many of these workflows are oriented toward structural biology and drug discovery, in particular the VS and identification of protein binding sites, by leveraging Taverna's technology for data storage, workflow enactment, change event notification, resource discovery and provenance management [100].

The VL-e environment, by leveraging existing grid technologies, enables molecular modeling for drug design using geographically distributed resources. The concept involves screening millions of compounds in a chemical database against a protein target to identify those

molecules with potential use in drug design. The VL-e uses the Nimrod-G parameter specification language to transform the existing molecular docking application into a parameter sweep application suitable for execution on distributed systems. New tools have been developed to enable access to ligand records/molecules in chemical databases from remote resources. The Nimrod-G resource broker, along with the chemical database data broker, is used for scheduling and the on-demand processing of docking jobs on the grid resources. The results show the ease of use and power of the Nimrod-G language and VL-e tools for drug discovery on grid computing platforms [101].

Moving toward the desktop grid paradigm, one important example is represented by the Drug Discovery Grid (DDGrid) [102] project. This focuses on providing drug screening services, such as building docking processes for the virtual high-throughput screening of the avian influenza virus [103]. The project relies on a grid environment and the grid computation resources of the Grid@Asia project. DDGrid leverages BOINC with grid technologies to harness the power of clusters and supercomputing systems owned by different organizations in China and South Korea.

Rosetta@home, a distributed computing project for the prediction of protein structures, also relies on the BOINC platform and aims to model protein–protein docking and protein structures. Rosetta@home has been used to validate proteins that neutralize influenza [104,105] and enzymes that break down gluten for the treatment of celiac disease, all of which are moving through animal trials and into clinical trials [106,107]. Rosetta@home has also been applied to research on malaria, Alzheimer's disease and other pathologies [108].

In addition to disease-related research, the Rosetta@home network serves as a testing framework for new methods in structural biology. Such methods are then used in other Rosetta-

based applications, such as RosettaDock [109], RosettaDesign and Robetta [110]. Rosetta@home consistently ranks as one of the foremost docking predictors and is one of the best predictors of tertiary structure currently available [111]. Using hundreds of thousands of cellular telephones and other mobile devices, Rosetta@home was used to validate peptide macrocycles designed with rigid structures, which could be useful as peptide therapeutics [112,113].

The conformational states from Rosetta's software can be used to initialize a Markov state model as starting points for simulations using Folding@home [114], which is a distributed computing platform for studying protein folding and other types of problems that can be solved with molecular dynamics. As Rosetta only tries to predict the final folded state, and not how folding proceeds, Rosetta@home and Folding@home are complementary and address very different molecular questions. Folding@home has been recognized as the most powerful distributed computing network in the world [115] and, in contrast to other similar projects, it does not rely on BOINC, but on a specific networking library called Cosm [116].

Large-scale executions of drug discovery applications on volunteer grids are no longer simply research projects, but an effective way to run *in silico* simulations. The main issue is the ratio of volunteer resources needed to achieve the computing power of a grid or cloud node, which is, on average, greater than three [117,118].

4.2 Cloud-based examples

Many examples of cloud computing rely on the platform as a service (PaaS) model, where applications are built using higher level platforms/frameworks. Many MapReduce-inspired frameworks developed by the Apache Software Foundation are currently available to manage and process large amounts of high-throughput omic data. For example, the Cancer

Genome Atlas project made use of Hadoop to split genome data into chunks distributed over the cluster for parallel processing [119,120].

The Collaborative Genomic Data Model (CGDM) [121] uses Hadoop to boost the querying speed for the main classes of query on genomic databases. MetaSparks [121] uses Spark to recruit large-scale metagenomics reads to reference genomes, achieving better scalability and sensitivity than programs based on a single machine [122], a principle that has also been applied to drug discovery [123].

4.3 GPU-based examples

GPUs have recently started to be used outside their traditional role as a graphical component of computers. This new application is known as general purpose computing on graphics processing units (GPGPU).

An example of GPU sharing within a distributed infrastructure is GPUGRID.net [124], a distributed computing infrastructure devoted to biomedical research. It consists of many GPUs joined together to deliver high performance all-atom biomolecular simulations. The simulations aim to develop better drugs by determining the mechanisms of drug resistance in cancers, modeling HIV maturation and investigating the features of neurologically important proteins.

The possibility of accessing GPUs on distributed facilities is particularly appealing because the most popular manufacturer, NVIDIA, markets itself as the best way to accelerate this kind of application. They have reported that molecular dynamics applications, such as ACEMD [125] and GROMACS [126], traditionally CPU-based, achieve increases in speed of up to eight times with the incorporation of their massive parallel platforms.

Researchers have carried out comparisons showing the reduced times achieved by performing tasks in these new platforms relative to traditional computation. Marin et al. [127]

compared the performance of a parallelization on multicore systems against a GPU and showed that the GPU implementation was up to 33 times faster. Analyses by Chiappori et al. [94] showed that these simulation techniques are time consuming and therefore parallel computing and GPU computations are required to reduce computation times.

Interesting papers have been published in this area of research—for example, Hung et al. [128] reviewed the two main CADD-based approaches (structure- and ligand-based drug design) and concluded that both multiple computers and GPGPU approaches can significantly improve the CADD performance. Vogt et al. [129] reviewed the current trends in method development, including the implementation of GPUs.

Ma et al. [130] used GPUs to accelerate the chemical similarity calculation that plays a major part in VS. Malhat et al. [131] saved between 76 and 99% of their computation time by using GPUs in their implementation of the Ward algorithm to group similar chemical compounds. Lo et al. [132] used CUDA to accelerate the prediction of protein–ligand binding regions using geometrical features.

Docking applications based on the scoring function, such as MetaDock [133], use a metaheuristic scheme to generate a large number of heuristic strategies for VS. MetaDock is designed to take full advantage of parallel and heterogeneous architectures, including multiprocessor chipsets and NVIDIA GPUs. Figure 4 shows how the MetaDock data distribution model works in GPUs. In this schema, the receptor molecule is stored in the shared memory of the multiprocessors to make better use of the memory accesses. The drug candidates are grouped in blocks of threads and are able to share the data from the receptor between the threads of the same multiprocessor. The application uses a computational molecular coupling methodology that seeks to predict the non-covalent binding of molecules or, more often, a macromolecule

(receptor) and a small molecule (ligand). The aim is to predict the bound conformations and affinity of the union—that is, the strength of association—which is usually measured by a scoring function [134,135].

Another approach that takes advantage of parallel programming is the migration of tools into this new parallel paradigm. McIntosh-Smith et al. [136] used this new GPU programming paradigm to increase the overall performance of the drug screening process, porting their tools to OpenCL. They describe the BUDE (Bristol University Docking Engine), which is a structure-based virtual screening (SBVS) engine, and present it as one of the first applications migrated to modern hardware [136]. Fang et al. [137] describe GeauxDock, a new molecular docking program migrated to run over parallel platforms. Some of these research groups are also involved in the review of new programs developed to take advantage of these parallel platforms [138]. Krige et al. [139] ported a commercial application (blazeV10) of VS to these platforms and compared an OpenCL version on a broad range of devices. Harvey et al. [140] implemented ACEMD, a production-class biomolecular dynamics simulation program specifically designed for GPUs. Sukhwani et al. [141] accelerated a production mapping software program using NVIDIA GPUs.

In addition to migrating tools to a distributed infrastructure to take advantage of the massive parallel platforms, another approach is to implement new programs to perform some tasks. For example, McArt et al. [142] implemented a new program to perform connectivity mapping, which is a computational technique dedicated to drug repurposing around differential gene expression analysis. They drastically reduced the computational times; previous implementations took up to seven days, whereas using a GPU reduced the time required to just ten minutes.

Studies in parallel algorithms that have led to new advances in this area include the work of Hu et al. [143], who developed a new GPU-based analytical method focusing on risk epistasis in a genome-wide association study of complex traits.

4.4. Applications of virtual screening web servers

HPC has been extensively applied in virtual screening. When full information about the target protein is available, SBVS methods are preferred and several servers are available to perform docking calculations. DOCK Blaster [144], Haddock [46], iScreen [145], SwissDock [146] and VSDocker [147] are examples of this type of server. Achilles [148] is a web tool implementing the blind docking approach. A distinguishing feature of Achilles is that it provides detailed reports, including the most likely ligand structure clusters, after processing the whole surface of the chosen protein.

SBVS calculations are usually very accurate, but the required information from the target protein is not always available. In such cases, LBVS is used as an alternative approach. A number of different servers implementing different approaches to LBVS are available on the web. As an example, SwissSimilarity [149] performs both 2D and 3D similarity searching, whereas HybridSim-VS [150] and USR-VS [151] focus on 3D similarity, including geometrical information about molecules. Other servers, such as iDrug [152], ChemMapper [153] and ZINCPharmer [154], use pharmacophore modeling to assess similarity. LBVS servers can efficiently scan millions or billions of compounds in a short time and are often a cost-effective solution to screening.

These examples have focused on the core of VS calculations. However, HPC can also help in the early stages of the screening process. For example, the prediction of molecular descriptors or fingerprints (e.g. molecular weight, number of rotatable bonds, number of

acceptor/donor hydrogens) is a sensitive task that benefits from high computing power. BioTriangle [155] is a complete toolkit for the characterization of complex biological molecules and their interactions. This tool is not only useful in the field of cheminformatics, but also in bioinformatics. ChemDes is another web tool for fingerprinting and molecular descriptor calculations. It hides the complexity of many other open source packages from the users. An extensive compilation of tools and databases for drug discovery is maintained by the Swiss Institute of Bioinformatics [156].

5. Conclusions

Distributed computing infrastructures, in particular those relying on GPUs, are very important in drug discovery programs. Although Big Pharma can rely on proprietary infrastructures to perform their analyses in-house, small- or medium-sized biotechnology laboratories need to exploit distributed infrastructures to achieve a flexible and cost-effective platform to perform their simulations. With distributed computing platforms, all players can combine the use of third-party software made available through a web interface with their own tools to exploit the full potential of this approach. There is a wide range of online VS servers providing services for many of the steps in the drug discovery process, including screening and fingerprint calculations.

The use of GPUs as computing platforms for drug discovery processes will further accelerate computations, as shown by the fact that commercial tools now include GPUs in their implementation to take advantage of their enhanced performance. Working with heterogeneous computation resources can help to improve the performance of applications. The main issues in these technological approaches are related to the cost-effective exploitation of the available computing capabilities because each GPU model has different features depending on its family

and generation and variations in the number of cores and performance. There is a need to develop applications with a load-balancing technique to correctly account for the characteristics of each device when distributing the workload through a distributed computing environment. Grid and cloud science gateways are an effective solution to providing user-friendly access to computational power and tools, while solving security issues and the problem of moving data between centers. These infrastructures are typically free for scientific purposes.

6. Expert opinion

Rapid developments in computational capabilities, combined with the explosion of research into personalized medicine, is currently leading the third wave of drug discovery. This new approach of personalized drug discovery should greatly improve the therapeutic effects of drugs while reducing their side-effects. The screening and validation of functional gene and protein targets in the early stages of drug discovery, and the development and optimization of selected molecules, requires huge amounts of computational power, which is difficult to buy and operate for small- and medium-sized biotechnology laboratories and academic research groups. In addition, perhaps surprisingly, large pharmaceutical companies are increasingly outsourcing research and computing activities to cut costs and to access state-of-the-art knowledge and technologies. We therefore see leading scientific institutions and computational chemistry companies shifting their business model from releasing commercial packages to providing distributed/cloud support for drug discovery research, while pharmaceutical companies exploit cloud services for drug development. As an example, in 2014, the Novartis Institutes for Biomedical Research used Amazon infrastructures to build a platform for the virtual screening of compounds against a common cancer target [157], leveraging the power and agility of cloud computing to conduct 39 years of computing in just 11 hours, creating an on-demand computing

system that would cost an estimated US$44 million to build for a cost of only US$5000. It is now almost impossible for researchers to enter the world of drug discovery without using these platforms.

These developments lead to the key message of this paper: recent advances in distributed computing technologies and infrastructures mean that they are now essential in the field of drug discovery. This was not true before the emergence of the grid computing paradigm about 30 years ago and, later, the cloud computing paradigm, but is currently very clear. The possibility of exploiting these paradigms presents great opportunities, as demonstrated by the large simulations conducted on grids and clouds by both pharmaceutical companies and public institutions, but also presents some weaknesses in terms of the effectiveness, accuracy and usability of software and hardware in distributed environments.

Effectiveness and accuracy mean the use of the correct methods in QSAR, VS, lead optimization and molecular dynamics. Therefore, after the development of new algorithms, usually by academic researchers, it is necessary to support the software through its full development for commercial use. The most common business model is to create spin-off companies to consolidate the development of these approaches, closing the gap between academia and the market, with sponsorship from pharmaceutical companies. These companies should also support users in applying these new approaches in the correct way, preventing failures in handling each specific tool. Some successful examples are available in the field of drug discovery, such as AMBER, one of the most popular software packages for molecular dynamics. Usability refers to the development of environments that lower the barriers to applying distributed and parallel computing infrastructures for drug discovery and offer support in handling failures.

Despite some open issues, drug discovery is moving towards distributed infrastructures because they provide high performance and cost-effective access to software tools, data and infrastructures. From the computational point of view, *in silico* analyses that were, until recently, impossible, are becoming increasingly feasible by applying the most suitable distributed computing infrastructure. This represents an incredible potential for this field of research, but more effort is required to develop and automate the functionalities that are crucial in enabling agile and flexible predictive modeling and simulation protocols. Workflow management systems can aid in many of these challenges, but the currently available systems are not suitable for users unfamiliar with ICT systems.

By considering the cost effectiveness of the different distributed infrastructures available to researchers for drug discovery, a recent review of computational models in computational biology evaluated the performance of a single application exploiting grid and cloud computing [70]. This study reported comparable results in terms of execution time, but also demonstrated that grid platforms have overheads due to failures caused by server misconfigurations and waiting times in cluster queues. Cloud infrastructures should therefore be preferred over grid approaches if the budget is sufficient, whereas if cost is a major issue, such as for neglected diseases, grid platforms may still be a valid solution.

Cloud platforms have cost advantages compared with buying local facilities because dynamic access to resources is less expensive than having a private infrastructure. In particular, if spot instances—cheap resources that can be turned off by the vendor when there is a high on-demand computational load—can be used, which may occur in virtual screening but not in molecular dynamics simulations, then cloud computing could be cheaper than buying computing

power from a high performance computational facility. This is particularly true if the cloud servers are equipped with high-end GPU devices and the applied cost model is appropriate.

Grid infrastructures and supercomputing centers require the pre-installation and tuning of software, whereas cloud solutions can exploit the pre-configured virtual machines released by vendors (e.g., Amazon and Google) or specialized companies (e.g., Eagle Genomics or Cloud Pharmaceuticals). However, it is important to highlight the fact that containers specialized for drug discovery analysis—for example, those relying on Docker [158]—can be used to rapidly create custom environments on all these platforms, reducing the time needed to set up and configure the software [159].

Future work in this area should be directed toward simplifying analyses and improving cooperation among researchers by providing input data, sharing the parameterization settings and releasing non-sensitive results to the scientific community. Open access to data and the reproducibility of methods have gained much attention in recent years and this is particularly true for approaches to developing new drugs. Tools are being developed to support researchers in this regard, but they require improvements and the trust of the scientific community. Workflow management systems and virtualization platforms are currently available to help improve the reproducibility and sharing of results and methods, although this is seldom considered before a publication or a patent is accepted as a result of the costly and long-running analyses required to achieve significant results and due to privacy/security issues. We postulate that open sharing is beneficial to the research community and research is more efficient when sharing is possible while protecting sensitive data.

A particular area of interest that, in our opinion, represents the next revolution in drug discovery is related to deep learning approaches [160], both for selecting compounds and for

their optimization. With the support of NVIDIA, some tools have already been released to improve the quantum mechanics energy function and to produce computationally fast and accurate molecular energy surfaces, geometries and forces. As a result of the complexity of these approaches, the use of HPC in a distributed infrastructure will be unavoidable.

Funding:

This work was funded by a grant from the Spanish Ministry of Economy and Competitiveness (CTQ2017-87974-R) and by the Spanish MEC and European Commission FEDER (TIN2016-78799-P, AEI/FEDER, UE). This research was partially supported by the supercomputing infrastructure of Poznan Supercomputing Center, the e-infrastructure program of the Research Council of Norway, the supercomputer center of UiT – the Arctic University of Norway and by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA−CIEMAT), funded by the European Regional Development Fund (ERDF). CETA−CIEMAT belongs to CIEMAT and the Government of Spain. The authors also acknowledge the computing resources and technical support provided by the Plataforma Andaluza de Bioinformática of the University of Málaga. Powered@NLHPC research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02).

Declaration of Interest:

The authors have no other relevant affiliations or financial involvement with any organization or entity with a financial interest in or financial conflict with the subject matter or materials discussed in the manuscript apart from those disclosed.

Reviewer Disclosures:

One referee is an employee of Merck & Co.

Abbreviations

CADD = Computer-Aided Drug Design

CAMD = Computer-Aided Molecular Design

CAMM = Computer-Aided Molecular Modeling

CDD = Computational Drug Discovery

ELM = Extreme Learning Machines

GPGPU = General Purpose Computing on Graphics Processing Units

GPU = Graphics Processing Unit

HPC = High Performance Computing

LBVS = Ligand-Based Virtual Screening

LDA = Linear Discriminant Analysis

MD = Molecular Dynamics

MLR = Multi-Linear Regression

NN = Neural Networks

PCA = Principal Components Analysis

PLS = Partial Least-Squares

QSAR = Quantitative Structure–Activity Relationships

SBVS = Structure-Based Virtual Screening

SVM = Support Vector Machines

VS = Virtual Screening

REFERENCES

1. Naylor CB, Richards WG. Computer-aided Drug Design [thesis]. Oxford (UK): University of Oxford; 1984.

2. McCammon JA. Computer-aided molecular design. Science. 1987; 238(4826):486-91.

3. Shoichet BK. Virtual screening of chemical libraries. Nature. 2004; 432(7019):862-865.

4. Morris GM, Huey R, Lindstrom W, et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. J Comput Chem. 2009; 30(16):2785-91.

5. Friesner RA, Murphy RB, Repasky MP, et al. Extra precision glide: docking and scoring incorporating a model of hydrophobic enclosure for protein-ligand complexes. J Med Chem. 2006; 49(21):6177-96.

6. Merelli I, Cozzi P, D'Agostino D, et al. Image-based surface matching algorithm oriented to structural biology. IEEE/ACM Trans Comput Biol Bioinform. 2011 Jul-Aug;8(4):1004-16.

7. Merelli I, Morra G, Milanesi L. Evaluation of a grid based molecular dynamics approach for polypeptide simulations. IEEE Trans Nanobioscience. 2007; 6(3):229-34.

8. Sheridan RP, Kearsley SK. Why do we need so many chemical similarity search methods? Drug Discov Today. 2002; 7(17):903-911.

9. Rudin M, Weissleder R. Molecular imaging in drug discovery and development. Nat Rev Drug Discov. 2003; 2(2):123-131.

10. Wang L, Wu YJ, Deng YQ, et al. Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field. J Am Chem Soc. 2015; 137(7):2695-2703.

11. D'Agostino D, Clematis A, Quarati A, et al. Cloud infrastructures for in silico drug discovery: economic and practical aspects. Biomed Res Int. 2013; 2013:138012.

12. Roy K, Kar S, Das RN. A Primer on QSAR/QSPR Modelling. 1st ed.: Springer, Cham; 2015. Chapter 2, Statistical Methods in QSAR/QSPR; p. 37.

13. Pérez-Garrido A, Girón-Rodríguez F, Helguera AM, et al. Topological structural alerts modulations of mammalian cell mutagenicity for halogenated derivatives. SAR and QSAR in Environmental Research. 2014; 25(1):17-33.

14. Yoo C, Shahlaei M. The applications of PCA in QSAR studies: A case study on CCR5 antagonists. Chem Biol Drug Des. 2018; 91(1):137-152.

15. Luco JM, Ferretti FH. QSAR Based on Multiple Linear Regression and PLS Methods for the Anti-HIV Activity of a Large Group of HEPT Derivatives. J Chem Inf Comput Sci. 1997; 37(2):392-401.

16. Pérez-Garrido A, Helguera AM, Rodríguez FG, et al. QSAR models to predict mutagenicity of acrylates, methacrylates and alpha,beta-unsaturated carbonyl compounds. Dent Mater. 2010; 26(5):397-415.

17. Pérez-Garrido A, Girón-Rodríguez F, Bueno-Crespo A, et al. Fuzzy clustering as rational partition method for QSAR. Chemometr Intell Lab Syst. 2017; 166:1-6.

18. Ghasemi F, Mehridehnavi A, Pérez-Garrido A, et al. Neural network and deep-learning algorithms used in QSAR studies: merits and drawbacks. Drug Discov Today. 2018; S1459-6446(17):30476-2.

19. Darnag R, Mazouz ELM, Schmitzer A, Villemin D, Jarid A, Cherqaoui D. Support vector machines: Development of QSAR models for predicting anti-HIV-1 activity of TIBO derivatives. Eur J Med Chem. 2010; 45(4):1590-1597.

20. Darvas F, Papp Á, Bágyi I, et al. OpenMolGRID, a GRID Based System for Solving Large-Scale Drug Design Problems. In: Dikaiakos M.D, editor. Grid Computing. Berlin, Heidelberg: Springer. 2004; 69-76.

21. Tetko IV, Maran U, Tropsha A. Public (Q)SAR services, integrated modeling environments, and repositories on the web: state of the art and perspectives for future development. Mol Inf. 2017; 35:1600082.

*** A review of public modeling environments for QSAR development. The authors explain the transition from the old-fashioned models to the new online approach.**

22. Sild S, Maran U, Lomaka A, et al. Open computing grid for molecular science and engineering. J Chem Inf Model. 2006; 46:953-959.

23. Sild S, Maran U, Romberg M, et al. OpenMolGRID: Using Automated Workflows in GRID Computing Environment. In: Sloot PMA, Hoekstra AG, Priol T, Reinefeld A, Bubak M, editors. Advances in Grid Computing - EGC 2005. Berlin, Heidelberg: Springer. 2005; 464-473.

24. Schuller B, Demuth B, Mix H, et al. Chemomentum - UNICORE 6 Based Infrastructure for Complex Applications in Science and Technology. In: Bougé L, Forsell M, Träff JL, Streit A, Ziegler W, Alexander M, Childs S, editors. Proceedings of the Euro-Par 2007 Workshops Parallel Processing; 2007; Rennes, France: Springer-Verlag. 2008;82-93.

25. Mazzatorta P, Smiesko M, Lo Piparo E, et al. QSAR Model for Predicting Pesticide Aquatic Toxicity. J Chem Inf Model. 2005; 45:1767-1774.

26. Maran U, Sild S, Mazzatorta P, et al. Grid Computing for the Estimation of Toxicity: Acute Toxicity on Fathead Minnow (Pimephales promelas). In: Dubitzky W, Schuster A, Sloot PMA, Schroeder M, Romberg M, editors. Distributed, High-Performance and Grid Computing in Computational Biology. Berlin, Heidelberg: Springer; 2007. p. 60-74.

27. Maran U, Sild S, Kahn I, et al. Mining of the chemical information in GRID environment. Future Gen Comput Syst. 2007; 23:76–83.

28. Polishchuk P, Tinkov O, Khristova T, et al. Structural and Physico-Chemical Interpretation (SPCI) of QSAR Models and Its Comparison with Matched Molecular Pair Analysis. J Chem Inf Model. 2016; 56(8):1455-1469.

29. Varnek A, Fourches D, Horvath D, et al. ISIDA - platform for virtual screening based on fragment and pharmacophoric descriptors. Curr Comput Aided Drug Des. 2008; 4:191-198.

30. Ruggiu F, Marcou G, Varnek A, et al. ISIDA Property-Labelled Fragment Descriptors. Mol Inform. 2010; 29(12):855-868.

31. Ambure P, Aher RB, Gajewicz A, et al. NanoBRIDGES software: Open access tools to perform QSAR and nano-QSAR modelling. Chemom Intell Lab Syst. 2015; 147:1-13.

32. Ekins S, Mestres J, Testa B. In silico pharmacology for drug discovery: Methods for virtual ligand screening and profiling. Br J Pharmacol. 2007; 152(1):9–20.

33. Kanehisa M, Furumichi M, Tanabe M, et al. KEGG: New perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. 2017; 45(D1):D353–361.

34. Ferreira L, dos Santos RN, Oliva G, et al. Molecular docking and structure-based drug design strategies. Molecules. 2015; 20:13384-13421.

35. Kitchen DB, Decornez H, Furr JR, et al. Docking and scoring in virtual screening for drug discovery: methods and applications. Nat Rev Drug Discov. 2004; 3(11): 935-949.

36. Talele TT, Khedkar SA, Rigby AC. Successful applications of computer aided drug discovery: moving drugs from concept to the clinic. Curr Top Med Chem. 2010; 10(1): 127-141.

37. Morris GM, Huey R, Lindstrom W, et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. J Comput Chem. 2009; 30(16):2785–2791.

38. Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. J Comput Chem. 2010; 31(2):455–461.

39. Friesner RA, Murphy RB, Repasky MP, et al. Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein−ligand complexes. J Med Chem. 2006; 49(21): 6177-6196.

40. Stroganov OV, Novikov FN, Stroylov VS, et al. Lead finder: an approach to improve accuracy of protein−ligand docking, binding energy estimation, and virtual screening. J Chem Inf Model. 2008; 48(12): 2371-2385.

41. Chiappori F, D'Ursi P, Merelli I, et al. In silico saturation mutagenesis and docking screening for the analysis of protein-ligand interaction: the Endothelial Protein C Receptor case study. BMC bioinformatics. 2009; 10(12):S3.

42. D'Ursi P, Chiappori F, Merelli I, et al. Virtual screening pipeline and ligand modelling for H5N1 neuraminidase. Biochem Bioph Res Co. 2009; 383(4):445-449.

43. Bock J, Gough D. A New Method to Estimate Ligand-Receptor Energetics. Mol Cell Proteomics. 2002; 11:904.

44. Cerón-Carrasco JP, Pérez-Sánchez H, Zúñiga J, et al. Antibodies as Carrier Molecules: Encapsulating Anti-Inflammatory Drugs inside Herceptine. J Phys Chem B. 2018; 122(7): 2064-2072.

45. Molecular Dynamics on Web. Available from: http://mmb.irbbarcelona.org

46. van Zundert GCP, Rodrigues JPGLM, Trellet M, et al. The HADDOCK2.2 webserver: User-friendly integrative modeling of biomolecular complexes. J Mol Biol. 2015; 428:720-725.

47. Margreitter C, Petrov D, Zagrovic B. Vienna-PTM webserver: a toolkit for MD simulations of protein post-translational modifications. Nucleic Acids Res. 2013; 41:W422-426.

48. CAB-S flex server. Laboratory of Theory of Biopolimers, Faculty of Chemistry. University of Warsaw. Available from: http://biocomp.chem.uw.edu.pl/CABSflex

49. Heo L, Feig M. PREFMD: a web server for protein structure refinement via molecular dynamics simulations. Bioinformatics. 2018; 34(6):1063-1065.

50. Devaurs D, Bouard L, Vaisset, M, et al. MoMA-LigPath: a web server to simulate protein-ligand unbinding. Nucleic Acids Res. 2013; 41(W1):W297-302.

51. Krüger J, Grunzke S, Gesing S, et al. The MoSGrid Science Gateway - A Complete Solution for Molecular Simulations. J Chem Theory Comput. 2014; 10(6):2232-2245.

52. Chen PC, Kuyucak S. Accurate Determination of the Binding Free Energy for KcsA-Charybdotoxin Complex from the Potential of Mean Force Calculations with Restraints. Biophys J. 2011; 100(10): 2466-2474.

53. Chiappori F, Merelli I, Milanesi L, et al. Static and dynamic interactions between GALK enzyme and known inhibitors: Guidelines to design new drugs for galactosemic patients. Eur J of Med Chem. 2013; 63: 423-34.

54. Foster I, Kesselman C. The history of the grid. Advances in Parallel Computing. 2011; 20:3-30.

**** This article describes the current status and future perspective of grid computing.**

55. Foster I, Kesselman C, Nick JM, et al. Grid computing: Making the global infrastructure a reality: Wiley Series; 2003. Chapter 8, The physiology of the grid; p. 217-249.

56. Chien A, Foster I, Goddette D. Grid technologies empowering drug discovery. Drug Discov Today. 2002; 7(20 Suppl):S176-80.

57. Trombetti GA, Merelli I, Orro A, et al. BGBlast: a BLAST grid implementation with database self-updating and adaptive replication. Stud Health Technol Inform. 2007; 126:23-30.

58. Anderson DP. Boinc: A system for public-resource computing and storage. In Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing 2004. 2004: 4-10. IEEE.
**** This article describes the BOINC distributed infrastructure.**

59. Jacq N, Breton V, Chen HY, et al. Virtual screening on large scale grids. Parallel Comput. 2007; 33(4): 289-301.

60. Chiappori F, Pucciarelli S, Merelli I, et al. Structural thermal adaptation of β-tubulins from the Antarctic psychrophilic protozoan Euplotes focardii. Proteins. 2011; 80(4):1154-1166

61. Foster I, Kesselman C. The globus toolkit. The grid: blueprint for a new computing infrastructure. 1999; 259-278.

62. Fifield T, Carmona A, Casajús A, et al. Integration of cloud, grid and local cluster resources with DIRAC. In Journal of Physics: Conference Series, IOP Publishing. 2011; 331(6): 062009.

63. Germain-Renaud C, Loomis C, Moscicki J, et al. Scheduling for Responsive Grids. J Grid Comput. 2008; 6(1):15-27.

64. Cecchi M, Capannini F, Dorigo A, et al. The glite workload management system. In International Conference on Grid and Pervasive Computing. Springer Berlin Heidelberg. 2009: 256-268.

65. Stevens RD, Robinson AJ, Goble CA. myGrid: personalised bioinformatics on the information grid. Bioinformatics. 2003; 19 Suppl 1:i302-4.

66. Rauwerda H, Roos M, Hertzberger BO, et al. The promise of a virtual lab in drug discovery. Drug Discov Today. 2006; 11(5-6):228-36.

67. Anderson DP, Cobb J, Korpela E, et al. SETI@home: an experiment in public-resource computing. Comm ACM. 2002; 45(11): 56-61.

68. Fedak G, He H, Lodygensky O, et al. EDGeS: a bridge between desktop grids and service grids. In ChinaGrid Annual Conference, 2008. ChinaGrid'08. The Third. 2008, Aug: 3-9. IEEE.

69. Merelli I, Cozzi P, Ronchieri E, et al. Porting bioinformatics applications from grid to cloud: a macromolecular surface analysis application case study. Int J High Perform Comput Appl. 2017; 31(3): 182-95.

70. Kasson PM. Computational Biology in the Cloud: Methods and New Insights from Computing at scale. Biocomputing 2013: World Scientific; 2012. p. 451-453.

71. Murty J. Programming amazon web services: S3, EC2, SQS, FPS, and SimpleDB. Sebastopol(CA): O'Reilly Media, Inc; 2008.

72. Gruber K. Google for genomes. Nature Research. 2014.

73. Matsunaga A, Tsugawa M, Fortes J. Cloudblast: Combining mapreduce and virtualization on distributed resources for bioinformatics applications. eScience. IEEE Fourth International Conference; 2008 Dec; 2008. p. 222-29.

74. Forer L, Lipic T, Schonherr S, et al. Delivering bioinformatics mapreduce applications in the cloud. Information and Communication Technology, Electronics and Microelectronics (MIPRO), 37th International Convention; 2014 May; IEEE; 2014. p. 373-77.

75. Schatz MC. CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics. 2009; 25(11):1363-9.

76. Ahmed L, Edlund A, Laure E, et al. Using iterative MapReduce for parallel virtual screening. IEEE 5th International Conference on Cloud Computing Technology and Science 2; 2013. p. 27-32.

77. Zhao J, Zhang R, Zhao Z, et al. Hadoop MapReduce framework to implement molecular docking of large-scale virtual screening. Services Computing Conference (APSCC); 2012 Dec; IEEE Asia-Pacific; 2012. p. 350-53.

78. Constantine RM, Batouche M. Drug discovery for breast cancer based on big data analytics techniques. Information & Communication Technology and Accessibility (ICTA), 5th International Conference; 2015 Dec; 2015. p. 1-6.

79. Zou Q, Li XB, Jiang WR, et al. Survey of MapReduce frame operation in bioinformatics. Brief Bioinform. 2014; 15(4):637-47.

80. Salomoni D, Italiano A, Ronchieri E. WNoDeS, a tool for integrated Grid and Cloud access and computing farm virtualization. Journal of Physics: Conference Series, IOP Publishing; 2011; 331(5):052017.

81. Ronchieri E, Cesini D, D'Agostino D, et al. The WNoDeS cloud virtualization framework: A macromolecular surface analysis application case study. Parallel, Distributed and Network-Based Processing (PDP), 22nd Euromicro International Conference; 2014 Feb; IEEE; 2014. p. 218-22.

82. Weber R, Gothandaraman A, Hinde RJ, et al. Comparing hardware accelerators in scientific applications: A case study. IEEE Trans Parallel Distr Syst. 2011; 22(1):58-68.

83. Couturier R. Designing Scientific Applications on GPUs. Belfort (France): CRC/Taylor & Francis; 2014.

84. TOP500 list. Available from: https://www.top500.org

85. Amdahl GM. Validity of the single processor approach to achieving large scale computing capabilities. In Proceedings of spring joint computer conference. Atlantic City, New Jersey. ACM; 1967. p. 483-485.

86. Gustafson JL. Reevaluating Amdahl's law. Commun ACM.1988; 31(5):532-533.

87. Kasam V, Salzemann J, Botha M, et al. WISDOM-II: screening against multiple targets implicated in malaria using computational grid infrastructures. Malar J. 2009 May 1;8:88.
**** This article describes the WISDOM drug discovery challenge**

88. Breton V, Jacq N, Kasam V, et al. Grid-added value to address malaria. IEEE Trans Inf Technol Biomed. 2008; 12(2):173-81.

89. Pharmaceutical composition for preventing and treating Malaria, containing compounds that inhibit Plasmepsin II activity, and method of treating Malaria using the same. Available from: https://patents.google.com/patent/WO2009131384A2/en

90. Pettifer S, Ison J, Kalas M, et al. The EMBRACE web service collection. Nucleic Acids Res. 2010; 38(Web Server issue):W683-8.

91. Milanesi L. BIOINFOGRID: Bioinformatics simulation and modelling based on grid. Model Simulat Sci. 2007: 178-186.

92. Chiappori F, Mattiazzi L, Milanesi L, et al. A novel molecular dynamics approach to evaluate the effect of phosphorylation on multimeric protein interface: the alphaB-Crystallin case study. BMC Bioinformatics. 2016;17 Suppl 4:57.

93. Dove MT, Sullivan LA, Walker AM, et al. Molecular dynamics in a grid computing environment: experiences using DL_POLY_3 within the e Minerals escience project. Molecular Simulation. 2006; 32(12-13):945-952.

94. Harvey MJ, De Fabritiis G. AceCloud: Molecular Dynamics Simulations in the Cloud. J Chem Inf Model. 2015; 55(5):909-14.

95. Addison E, Keinan S. Using Quantum Molecular Design & Cloud Computing to Improve the Accuracy & Success Probability of Drug Discovery. Drug Development & Delivery. 2016; 16(2):62-66.

96. Jiang W, Phillips JC, Huang L, et al. Generalized Scalable Multiple Copy Algorithms for Molecular Dynamics Simulations in NAMD. Comput Phys Commun. 2014; 185(3):908-916.

97. Merelli I, Morra G, D'Agostino D, et al. High performance workflow implementation for protein surface characterization using grid technology. BMC Bioinformatics. 2005; 6 Suppl 4:S19.

98. Hull D, Wolstencroft K, Stevens R, et al. Taverna: a tool for building and running workflows of services. Nucleic Acids Res. 2006; 34(Web Server issue):W729-32.

99. Oinn T, Li P, Kell DB, et al. Taverna/my Grid: aligning a workflow system with the life sciences community. In Workflows for e-Science. Springer, London. 2007: 300-19.

100. Stevens R, McEntire R, Goble C, et al. myGrid and the drug discovery process. Drug Discov Today. 2004; 2(4):140–148.

101. Buyya R, Branson K, Giddy J, et al. The Virtual Laboratory: A toolset to enable distributed molecular modelling for drug design on the world-wide grid. Concurr Comput Pract Exp. 2003; 15(1):1–25.

102. Zhang W, Du X, Ma F, et al. DDGrid: Harness the Full Power of Supercomputing Systems. In Grid and Cooperative Computing Workshops, 2006. GCCW'06. Fifth International Conference. 2006, Oct: 1-4. IEEE.

103. Wang Q, Ye Y, Yu K, et al. A Graphical Workflow Modeler for Docking Process in Drug Discovery. In Grid and Cloud Computing: Concepts, Methodologies, Tools and Applications. 2012: 1408-22. IGI Global.

104. Koday MT, Nelson J, Chevalier A, et al. A Computationally Designed Hemagglutinin Stem-Binding Protein Provides In Vivo Protection from Influenza Independent of a Host Immune Response. PLoS Pathog. 2016; 12(2):1–23.

105. Fleishman SJ, Whitehead TA, Ekiert DC, et al. Computational design of protein targeting the conserved stem region of influenza hemagglutinin. Science. 2011; 332(6031):816–821.
*** The paper describes a computational method for designing proteins. The method is applied to the design of different proteins.**

106. Gordon SR, Stanley EJ, Wolf S, et al. Computational design of an α-gliadin peptidase. J Am Chem Soc. 2012; 134(50):20513–20520.

107. Wolf C, Siegel JB, Tinberg C, et al. Engineering of Kuma030: A Gliadin Peptidase That Rapidly Degrades Immunogenic Gliadin Peptides in Gastric Conditions. J Am Chem Soc. 2015; 137(40):13106–13113.

108. Das R, Qian B, Raman S, et al. Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@ home. Proteins. 2007; 69(S8): 118-28.

109. Lyskov S, Gray JJ. The RosettaDock server for local protein-protein docking. Nucleic Acids Res. 2008; 36(Web Server issue):W233-8.

110. Kim DE, Chivian D, Baker D. Protein structure prediction and analysis using the Robetta server. Nucleic Acids Res. 2004; 32(Supp 2):W526-W531.

111. Lensink MF, Mendez R, Wodak SJ. Docking and scoring protein complexes: CAPRI 3rd Edition. Proteins. 2007; 69(4):704-18.

112. Bhardwaj G, Mulligan VK, Bahl CD, et al. Accurate de novo design of hyperstable constrained peptides. Nature. 2016; 538(7625):329–335.

113. Hosseinzadeh P, Bhardwaj G, Mulligan VK, et al. Comprehensive computational design of ordered peptide macrocycles. Science. 2017; 358(6369):1461–1466.

114. Jayachandran G, Vishal V, Pande VS. Using massively parallel simulation and Markovian models to study protein folding: examining the dynamics of the villin headpiece. J Chem Phys. 2006; 124(16):164902.

115. Most powerful distributed computing network [Internet]. 2007 Sep 16. Available from: http://guinnessworldrecords.com.

116. Beberg A, Ensign D, Jayachandran G, et al. Folding@home: Lessons From Eight Years of Volunteer Distributed Computing. In: Parallel & Distributed Processing, IEEE International Symposium: 1–8. 2009.
*** Authors describe the larger distributed platform worldwide for protein structural analysis**

117. Kondo D, Javadi B, Malecot P, et al. Cost-benefit analysis of cloud computing versus desktop grids. In Parallel & Distributed Processing; 2009 May 23-29. IPDPS 2009. IEEE International Symposium. 2009. p. 1-12.

118. Bertis V, Bolze R, Desprez F, et al. Large scale execution of a bioinformatic application on a volunteer grid. Parallel and Distributed Processing; 2008 Apr 14-18; Miami (FL); IPDPS 2008. IEEE International Symposium. 2008. p. 1-8.

119. Merelli I, Pérez-Sánchez H, Gesing S, et al. Managing, analysing, and integrating big data in medical bioinformatics: open problems and future perspectives. Biomed Res Int. 2014; 2014:134023.

120. McKenna A, Hanna M, Banks E, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. 2010; 20(9):1297-303.

121. Zhou W, Li R, Yuan S, et al. MetaSpark: a Spark-based distributed processing tool to recruit metagenomic reads to reference genomes. Bioinformatics. 2017; 33(7):1090-1092.

122. Niu B, Zhu Z, Fu L, et al. FR-HIT, a very fast program to recruit metagenomic reads to homologous reference genomes. Bioinformatics. 2011; 27(12):1704-5.

123. Harnie D, Saey M, Vapirev AE, et al. Scaling machine learning for target prediction in drug discovery using Apache Spark. Future Generat Comput Syst. 2017; 67:409-17.

124. Buch I, Harvey MJ, Giorgino T, et al. High-throughput all-atom molecular dynamics simulations using distributed computing. J Chem Inf Model. 2010; 50(3):397-403.

125. Harvey MJ, Giupponi G, Fabritiis GD. ACEMD: accelerating biomolecular dynamics in the microsecond time scale. J Chem Theory Comput. 2009; 5(6): 1632-1639.

126. Hess B, Kutzner C, Van der spoel D, et al. GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. J Chem Theory Comput. 2008; 4(3): 435-447.

127. Marin I, Goga N, Goga M. Benchmarking MD Systems Simulations on the Graphics Processing Unit and Multi-core Systems. IEEE International Symposium on Systems Engineering (ISSE); 2016 Oct 3-5; Edinburgh (UK); 2016.

128. Hung CL, Chen CC. Computational approaches for drug discovery. Drug Dev Res. 2014; 75(6):412-8.

129. Vogt M, Bajorath J. Chemoinformatics: a view of the field and current trends in method development. Bioorg Med Chem. 2012; 20(18):5317-23.

130. Ma C, Wang L, Xie XQ. GPU accelerated chemical similarity calculation for compound library comparison. J Chem Inf Model. 2011; 51(7):1521-7.

131. Malhat MG, El-Sisi AB. Parallel Ward Clustering for Chemical Compounds Using OpenCL. Tenth International Conference on Computer Engineering & Systems; 2015 Dec 23-24; Cairo (Egypt); ICCES; 2016.

132. Lo YT, Wang HW, Pai TW, et al. Protein–ligand binding region prediction (PLB-SAVE) based on geometric features and CUDA acceleration. BMC Bioinformatics. 2013; 14 Suppl 4:S4.

133. Imbernón B, Cecilia JM, Pérez-Sánchez H, et al. METADOCK: A parallel metaheuristic schema for virtual screening methods. Int J High Perform Comput Appl. 2017.

134. Yuriev E, Holien J, Ramsland PA. Improvements, trends, and new ideas in molecular docking: 2012-2013 in review. J Mol Recognit. 2015; 28(10):581-604.

135. Lionta E, Spyrou G, Vassilatis DK, et al. Structure-based virtual screening for drug discovery: principles, applications and recent advances. Curr Top Med Chem. 2014; 14(16):1923-38.

136. McIntosh-Smith S, Price J, Sessions RB, et al. High performance in silico virtual drug screening on many-core processors. Int J High Perform Comput Appl. 2015; 29(2):119-134.

137. Fang Y, Ding Y, Feinstein WP, et al. GeauxDock: Accelerating Structure-Based Virtual Screening with Heterogeneous Computing. PLoS One. 2016; 11(7):e0158898.

138. Feinstein W, Brylinski M. Structure-Based Drug Discovery Accelerated by Many-Core Devices. Curr Drug Targets. 2016; 17(14):1595-1609.

139. Krige S, Mackey M, McIntosh-Smith S, et al. Porting a Commercial Application to OpenCL. Proceedings of the International Workshop on OpenCL 2013 & 2014 - IWOCL '14; 2014.

140. Harvey MJ, Giupponi G, Fabritiis GD. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. J Chem Theory Comput. 2009; 5(6):1632-9.

141. Sukhwani B, Herbordt MC. Fast Binding Site Mapping Using GPUs and CUDA. 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW); 2010; 04.

142. McArt DG, Bankhead P, Dunne PD, et al. cudaMap: a GPU accelerated program for gene expression connectivity mapping. BMC Bioinformatics. 2013; 14:305.

143. Hu X, Liu Q, Zhang Z, et al. SHEsisEpi, a GPU-enhanced genome-wide SNP-SNP interaction scanning algorithm, efficiently reveals the risk genetic epistasis in bipolar disorder. Cell Res. 2010; 20(7):854-7.

144. Irwin JJ, Shoichet BK, Mysinger MM, et al. Automated docking screens: a feasibility study. J Med Chem. 2009; 52(18):5712-5720.

145. Tsai TY, Chang KW, Yu-Chian Chen C. iScreen: world's first cloud-computing web server for virtual screening and de novo drug design based on TCM database@Taiwan. J Comput Aided Mol Des. 2011; 25(6):525-531.

146. Grosdidier A, Zoete V, Michielin O. SwissDock, a protein-small molecule docking web service based on EADock DSS. Nucleic Acids Res. 2011; 39(Web Server issue):W270-7.

147. Prakhov ND, Chernorudskiy AL, Gainullin MR. VSDocker: a tool for parallel high-throughput virtual screening using AutoDock on Windows-based computer clusters. Bioinformatics. 2010; 26(10):1374-1375.

148. Sánchez-Linares I, Pérez-Sánchez H, Cecilia JM, et al. High-Throughput parallel blind Virtual Screening using BINDSURF. BMC Bioinformatics. 2012; 13 Suppl 14:S13.

149. Zoete V, Daina A, Bovigny C, et al. SwissSimilarity: A Web Tool for Low to Ultra High Throughput Ligand-Based Virtual Screening. J Chem Inf Model. 2016; 56(8):1399-404.

150. Shang J, Dai X, Li Y, et al. HybridSim-VS: a web server for large-scale ligand-based virtual screening using hybrid similarity recognition techniques. Bioinformatics. 2017; 33(21):3480-3481.

151. Li H, Leung KS, Wong MH, et al. USR-VS: a web server for large-scale prospective virtual screening using ultrafast shape recognition techniques. Nucleic Acids Res. 2016; 44(W1):W436-W441.

152. Wang X, Chen H, Yang F, et al. iDrug: a web-accessible and interactive drug discovery and design platform. J Cheminform. 2014; 6(28).

153. Gong J, Cai C, Liu X, et al. ChemMapper: a versatile web server for exploring pharmacology and chemical structure association based on molecular 3D similarity method. Bioinformatics. 2013; 29(14):1827-1829.

154. Koes DR, Camacho CJ. ZINCPharmer: pharmacophore search of the ZINC database. Nucleic Acids Res. 2012; 40(W1):W409-W414.

155. Dong J, Yao ZJ, Wen M, et al. BioTriangle: a web-accessible platform for generating various molecular representations for chemicals, proteins, DNAs/RNAs and their interactions. J Cheminform. 2016; 8:34.

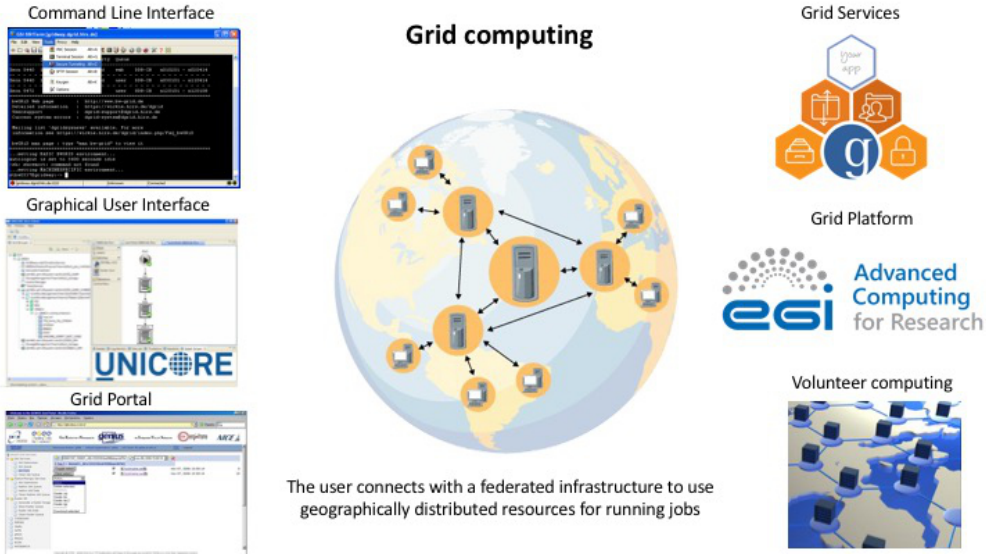
156. Click2Drug. Swiss Institute of Bioinformatics. Available from: http://www.click2drug.org

157. Novartis Case Study. Amazon Web Services [Internet]. Available from: https://aws.amazon.com/it/solutions/case-studies/novartis/.

158. List M. Using Docker Compose for the Simple Deployment of an Integrated Drug Target Screening Platform. J Integr Bioinform. 2017; 14(2).

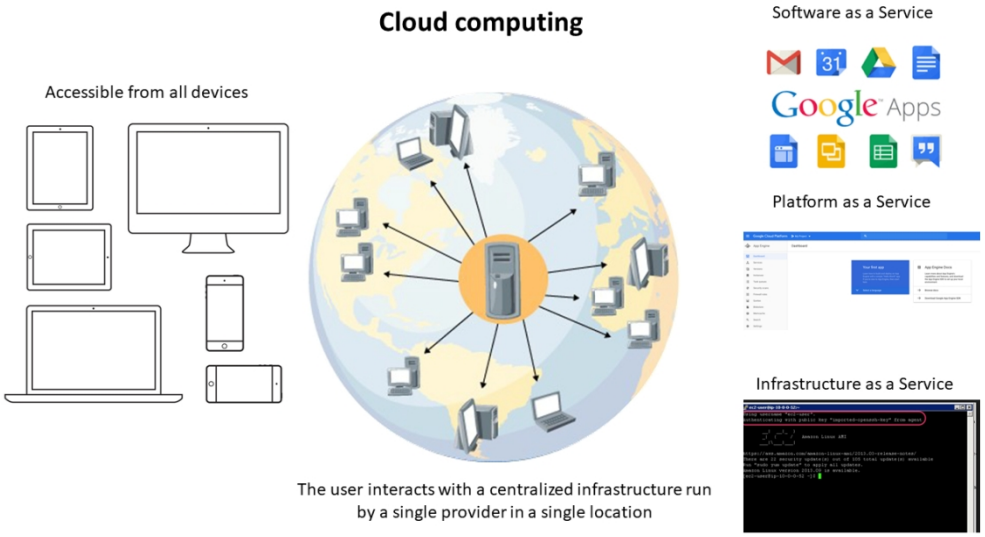
159. Merelli I, Fornari F, Tordini F, et al. Exploiting Docker Containers over Grid Computing for a comprehensive Study of Chromatin Conformation in different cell types. Journal of Parallel and Distributed Computing, in press.

160. AstraZeneca taps AI for drug discovery in deal with Berg. Reuters. 2018. Available from: https://www.reuters.com/article/us-astrazeneca-ai-berg/astrazeneca-taps-ai-for-drug-discovery-in-deal-with-berg-idUSKCN1B81G1



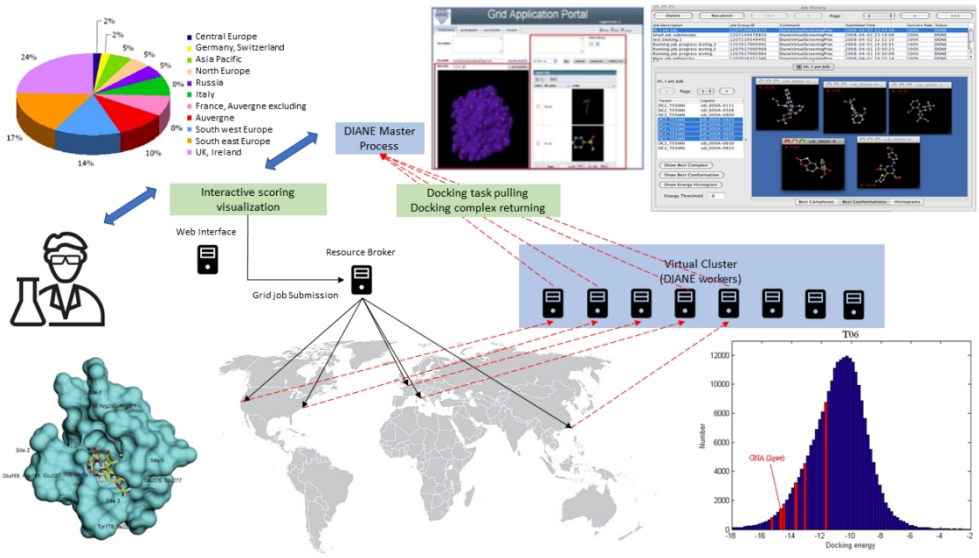
The Grid computing paradigm. On the left side, three examples of user interface are shown (Command Line Interface, Graphical User Interface, Grid Portal). The central panel shows the common federated and geographical dispersed infrastructure typical of grid computing. On the right side, the three general types of grid infrastructures are reported (Grid Services, Grid Platform, Volunteer Computing).

338x190mm (54 x 54 DPI)



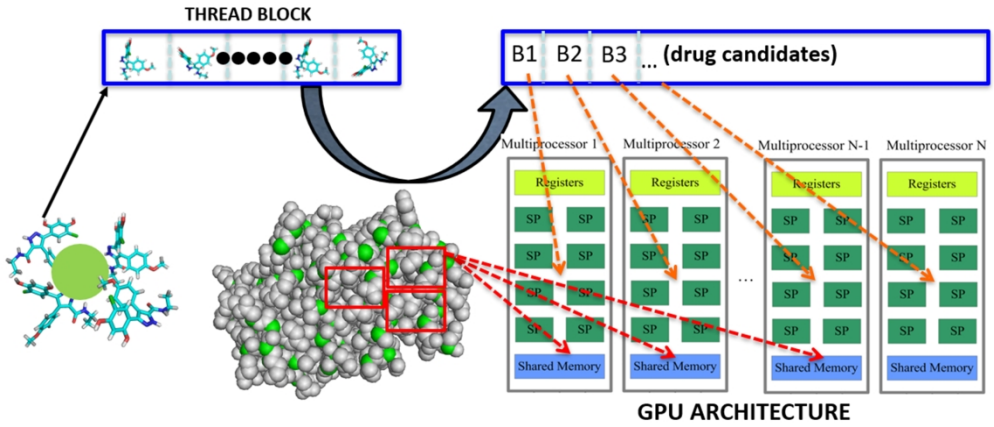
The Cloud computing paradigm. On the left side, the figure shows how cloud infrastructures are easily accessible from any kind of device. The central panel shows the common centralized and single-entry point infrastructure typical of Cloud Computing. On the right side, three popular examples of Cloud platforms are shown (Software as a Service, Platform as a Service, Infrastructure as a Service).

108x60mm (300 x 300 DPI)



The architecture of the grid-based infrastructure adopted for WISDOM and the Avian flu virtual screenings. DIANE was used as job dispatcher and a Grid Application Portal was developed to visually inspect achieved results. In the top left corner, the distribution of the jobs among the different geographical regions of the world. In the bottom left corner, the structure of the Neuraminidase that was target of the screening. In the top right corner, the job monitoring application used in the virtual screening. In the bottom right corner, the enrichment curve of one of the leading compounds identified with the virtual screening.

338x190mm (96 x 96 DPI)



Data distribution model on GPU in METADOCK. Drug candidates are grouped in blocks of threads and mapped to the multiprocessors for their computation. The shared memory, in the lower part, stores parts of the receptor molecule for reuse by the candidates that integrate the same block of threads.

440x186mm (300 x 300 DPI)