Expanding the Knowledge Economy: Issues, Applications, Case Studies
Paul Cunningham and Miriam Cunningham (Eds)
IOS Press, 2007 Amsterdam
ISBN 978-1-58603-801-4

Multi-Science Decision Support for HIV Drug Resistance Treatment

Peter SLOOT, Alfredo TIRADO-RAMOS, Marian T. BUBAK
Computational Science Section, University of Amsterdam, Amsterdam, 1098SJ, The Netherlands
Tel: +31 20 5257460, Fax: + 31 20 5257490, Email: {sloot, alfredo,bubak}@science.uva.nl

Abstract: The complete cascade from genome, proteome, metabolome, and physiome, to health forms multiscale, multiscience systems and crosses many orders of magnitude in temporal and spatial scales. The interactions between these systems create exquisite multitiered networks, with each component in nonlinear contact with many interaction partners. Understanding, quantifying, and handling this complexity is one of the biggest scientific challenges of our time. In this paper we argue that computer science in general, and Grid computing in particular, provide the language needed to study and understand these systems, and discuss a case study in decision support for HIV drug resistance treatment within the European ViroLab project.

1. From Molecule To Man

'During the next decade, the practice of medicine will change dramatically, through personalized, targeted treatments that will enable a move beyond prevention to pre-emptive strategies.' [1]

Humans are complex systems: from a biological cell made of thousands of different molecules that work together, to billions of cells that build our tissue, organs and systems, to our society, six billion unique interacting individuals. Such complex systems are not made of identical and undistinguishable components: rather each gene in a cell, each cell in the immune system, and each individual have their own characteristic behavior and provide unique value and contributions to the systems in which they are constituents. The complete cascade from the genome, proteome, metabolome, physiome to health constitutes multi-scale, multi-science systems, and crosses many orders of magnitude in temporal and spatial scales [2], as seen in Figure 1. The interactions between these systems form exquisite multi-tiered networks, each component being in non-linear contact with many selected interaction partners. These networks are not just complicated; they are complex. Understanding, quantifying and handling this complexity is one of the biggest scientific challenges of our time [3].

It is the central assertion of this paper that computer science is the language to study and understand these systems, and that the same laws and organizing principles that dictate biomedical systems are reflected in the architecture of simulating computer systems.

We discuss some of these laws and organizing principles required to build systems for individualized biomedicine that can account for variations in physiology, treatment and drug response.

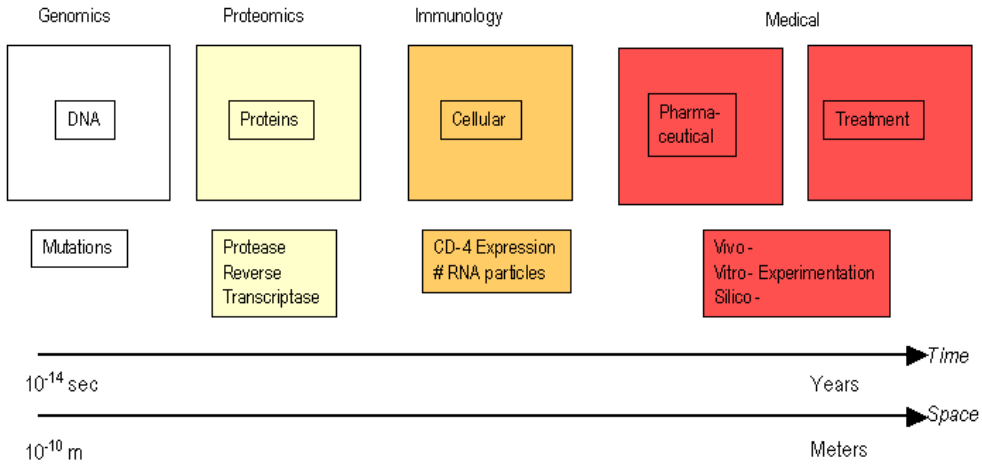

Figure 1. Multi-scale, multi-science models and techniques are needed to cover the huge spatial and temporal scales in studying complex problems such as drug response in infectious diseases.

1.1 Pushing and Pulling

We observe an application pull from biomedicine that is changing the scientific paradigm to emphasis for in silico studies, where more and more details of biomedical processes are simulated in addition to in vivo and in vitro studies. These simulated processes are being used to support medical doctors in making decisions through exploration of different scenarios. Typical examples are pre-operative simulation and visualization of vascular surgery [4] and expert systems for drug ranking [5]. At the same time we observe a technology push from computing and large amounts of data availability [6].

In the field of high-performance computing there have been changes from sequential to parallel to distributed computing, where the 'killer applications' moved from mathematics to physics to chemistry to biology to medicine, thus increasing the complexity of the systems under study with the complexity of the computational systems being required. In addition, with the advances in Internet technology and Grid computing [7], huge amounts of data from sensors, experiments and simulations have become available. There are, however, significant computational, integration, collaboration, and interaction gaps between these observed application pull and technology push that need to be addressed.

1.2 Bridging the Gaps

In order to close the computational gap in systems biology, we need to construct, integrate and manage a plethora of models. A bottom-up data-driven approach will not work for this. Web and Grid services are needed to integrate often incompatible applications and tools for data acquisition, registration, storage, provenance, organization, analysis and presentation, thus bridging the integration gap. Even if we manage to solve the computational and integration challenges, we still need a system-level approach to share processes, data, information and knowledge across geographic and organizational boundaries within the context of distributed, multidisciplinary and multi-organizational collaborative teams, or 'virtual organizations' as they are often called, thus closing the collaboration and interaction gap.

Finally, we need intuitive methods to streamline all these processes dynamically depending on their availability, reliability and the specific interests of the end-users (medical doctors, surgeons, clinical experts, and researchers). Such methods can be captured into 'scientific workflows' in which the flow of data and control from one step to

another is expressed in a workflow language [8,9]. A general scheme for conducting such type of e-Science research is depicted in Figure 2.

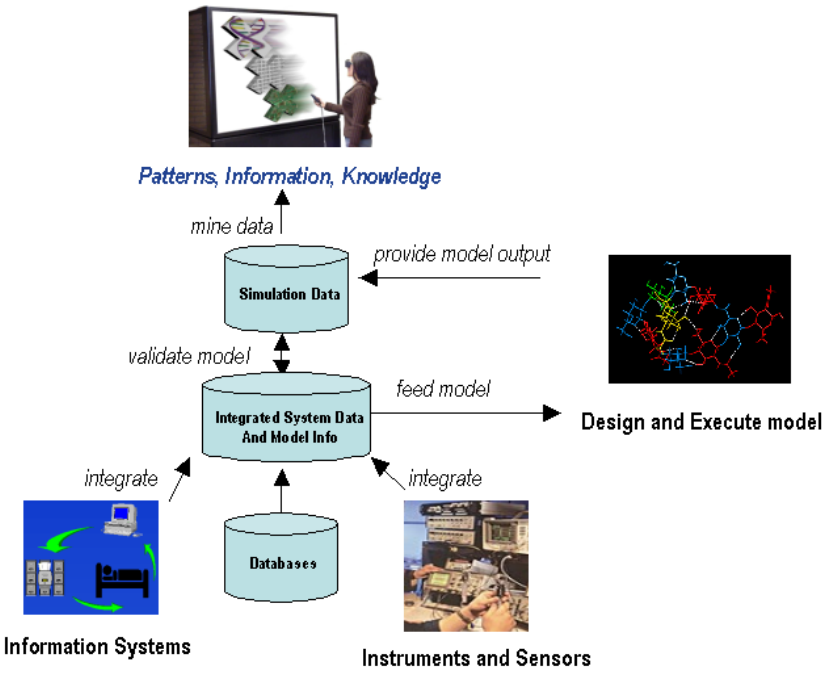

Figure 2. General architecture for conducting e-Science research: information systems integrate available data with data from specialized instruments and sensors into distributed repositories. Computational models are then executed using the integrated data, providing large quantities of model output data, which is mined and processed in order to extract useful knowledge.

We discuss the development of a Grid based decision support system consisting of modules such as the one depicted in Figure 2, for individualized drug ranking in Human Immunodeficiency Virus (HIV-1) diseases, called ViroLab [10]. The reason for using this complex problem of HIV drug resistance as a prototype for our system-level approach is twofold. First of all, HIV drug resistance is becoming an increasing problem worldwide, with a considerable number of HIV infected patients developing failure of complete suppression of the virus despite combination therapy with antiretroviral drugs. Second, HIV drug resistance is one of the few areas in medicine where genetic information is widely available and used for a considerable number of years. As a consequence, large numbers of complex genetic sequences are available, in addition to clinical data.

2. ViroLab: Collaborative Decision Support System in Viral Disease Treatment

'A process cannot be understood by stopping it. Understanding must move with the flow of the process, must join it and flow with it' (First Law of Mentat), Frank Herbert, Dune.

2.1 Background

During the past ten years significant progress has been made in the treatment of viral disease infected patients. For instance, around 20 antiretroviral drugs are now available for treatment of HIV, divided into four classes, with patients taking a combination of at least two different classes of antiretroviral drugs in order to achieve complete suppression of the virus [11]. In a considerable proportion of patients however, the drugs fail to completely suppress the viral disease, resulting in the rapid selection of drug-resistant viruses and loss of drug effectiveness. This complicates the clinician's decision process, as the basis for

clinical interpretation is based on datasets relating mutations to changes in drug sensitivity and relating mutations present in the virus to clinical responses to specific regiments.

A number of genotypic resistance interpretation tools that assist clinicians and virologists in choosing effective therapeutic alternatives have been developed in recent years. However, there is significant interpretation discordance among the available systems for interpreting, e.g., HIV-1 genotypic resistance. There is an urgent need for a joint effort to develop, validate, publish standardized rules as well as definition criteria for genotypic resistance interpretation, and to provide accessible tools for interpretation that help improve the clinical usefulness of genotypic assay results. The application of artificial intelligence and computational techniques applied to biomedicine has resulted in the development of computer-based decision support systems (DSSs). Recent developments in distributed computing further allow the virtualization of massive data, computational, and software resources that complex DSSs require.

Our vision is to provide researchers and medical doctors with a virtual laboratory for infectious diseases, called ViroLab [10], to enable easy access to distributed simulations as well as sharing, processing and analyzing virological, immunological, clinical and experimental data. Currently, virologists browse journals, pick results, compile them for discussion, and derive rules for ranking and making decisions. ViroLab advances the state of the art by offering clinicians a distributed virtual laboratory securely accessible from their hospitals and institutes distributed all over Europe. A typical usage scenario can be:

- A scientist from a laboratory for clinical and epidemiological virology in Utrecht, The Netherlands securely accesses virus sequence, amino acid or mutations data from a hospital's AIDS lab in Rome, Italy, using data Grid technology components running in Stuttgart, Germany.
- The scientist applies quality indicators needed for data provenance tracking using provenance server components running in Cracow, Poland.
- This data is used as input to (molecular dynamics) simulations and immune system simulations running on Grid-nodes that reside in London, UK and Amsterdam, The Netherlands.
- The virtualized DSS automatically derives meta-rules.
- Intelligent system components from Amsterdam use first order logic to clean rules, identify conflicts, redundancy and check logical consistency.
- The scientist validates new rules automatically uploaded into the virtualized DSS.
- A new ranking is presented and made available for all virtual organization members.

The ViroLab is based on a Grid-based virtual laboratory for infectious diseases that facilitates medical knowledge discovery, providing the medical doctors with a decision support system to rank drugs targeted at patients. Its infrastructure provides virologists with an advanced environment to study trends on an individual, population and epidemiological level. That is, by virtualizing the hardware, computing infrastructure, and databases, the virtual laboratory offers a user-friendly environment. It also offers tailored workflow templates to harness and automate such diverse tasks as data archiving, data integration, data mining and analysis, modeling and simulation, integrating the biomedical information from viruses (proteins and mutations), patients (e.g. viral load) and literature (drug resistance experiments), resulting in a rule-based distributed decision support system for drug ranking.

In ViroLab we put a virtualized DSS and an interpretation tool at the center of the distributed virtual laboratory. One of such interpretation tools is Retrogram [10]. Retrogram estimates the drug sensitivity for available drugs by interpreting the genotype of a patient using mutational algorithms, developed by experts on the basis of scientific literature, taking into account the published data relating genotype to phenotype. In addition, the ranking is based on data from clinical studies on the relationship between the presence of

particular mutations and clinical or virological outcome. For the system to support Grid-based distributed data access and computation, the virtualization of its components is of primary importance. ViroLab includes advanced tools for (bio)-statistical analysis, visualization, modeling and simulation that enable prediction of the temporal virological and immunological response of viruses with complex mutation patterns for drug therapy, as seen in Figure 3.

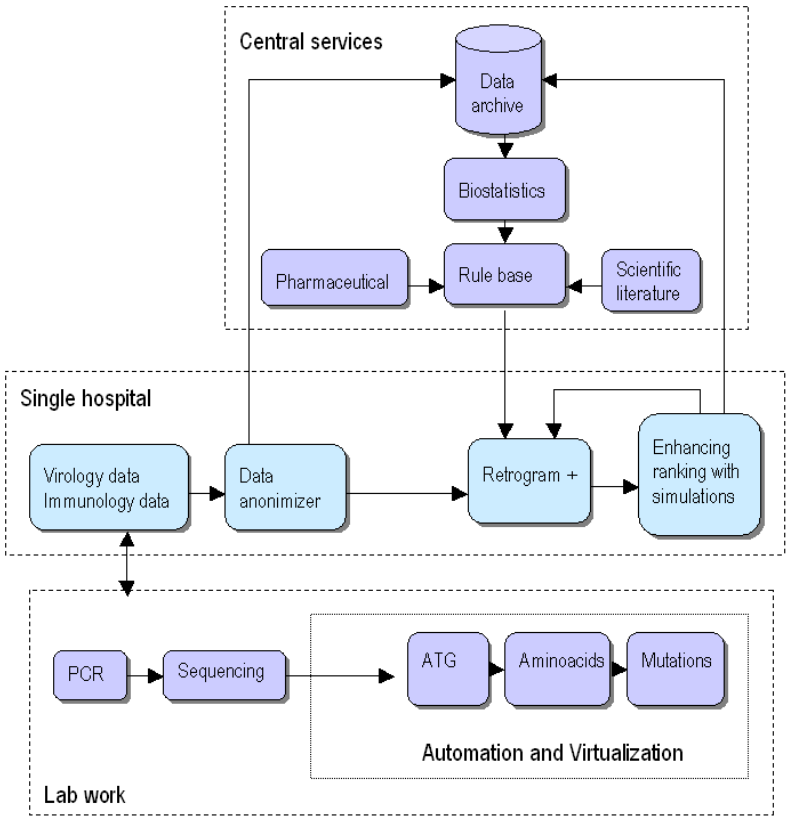

Figure 3. Schematic drawing of the ViroLab data and control flow: manual wet-lab is automated and virtualized, with the resulting data being fed to anonimizing components as well as directly to the decision support system to be ranked. Simulation components enhance output rankings, which are stored before they are applied to rule-based algorithms, and then fed back for prediction of the drug sensitivity of the virus.

2.2 ViroLab Architecture

In *ViroLab*, we consider each 'experiment' as a set of interconnected activities. The design of the *ViroLab* system guarantees the interaction between a user and running applications, similar to the way it is done in real experiments, so that the user would be able to change a selected set of input data or parameters at runtime.

In addition to the DSS, the patient databases, data analysis tools and the simulation software, the runtime system of *ViroLab* consists of:

- a distributed and fault-tolerant registry for storing, updating and publishing semantic information about available resources and executed applications,
- a tool for composing new experiments or modifying experiments already performed,
- an execution engine to enact workflows according to flow of data and actions,
- a scheduler for dynamic selection of resources to be used to run a given experiment in an efficient way.

ViroLab workflows enable dynamic workflow execution, lazy scheduling and runtime re-composition. They also support two levels of abstraction that are necessary to operate separately on abstract workflows (workflow templates) and on concrete workflow instances (executables). Major challenges to be addressed in the development of a virtual laboratory

are the highly distributed and heterogeneous nature of the data (virological, immunological, clinical and experimental), the high dimensionality and complexity of the (genetic and patient) data as well as the inaccessibility and (lack of) interoperability of advanced modeling, simulation and analyses tools. Recent advances in Grid computing tackle these problems by virtualizing the resources (data, instruments, compute nodes, tools, users, etc.) and making them transparently available. In Grid computing, the basic unit of organization is the Virtual Organization (VO), where a VO is a set of Grid entities, such as individuals, institutions, applications, services or resources, which are related to each other by some level of trust. These ideas are summarized in Figure 4.

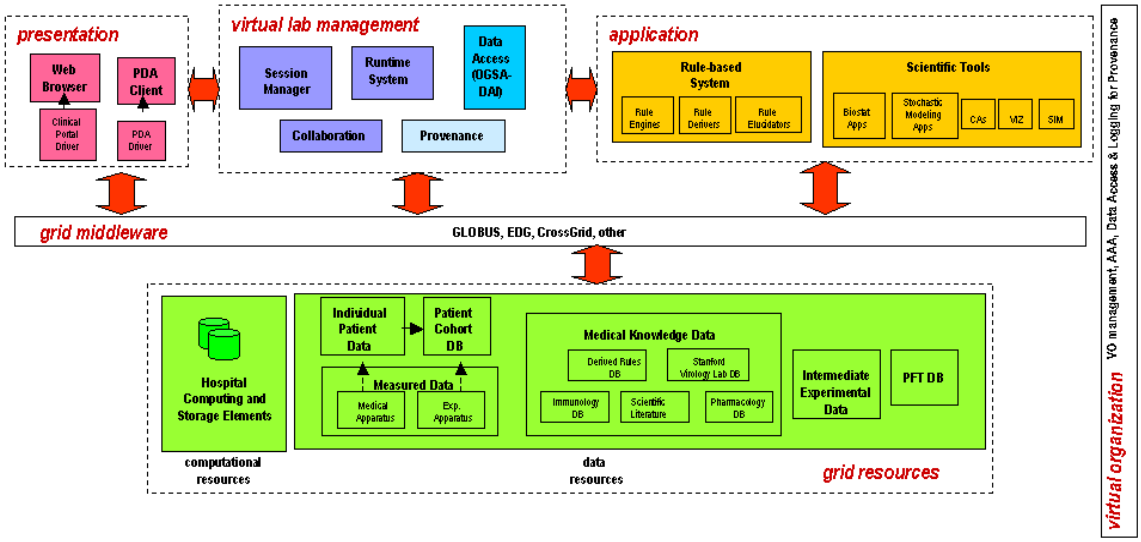

Figure 4 Architecture of the ViroLab system. Distributed resources (computing elements, data and storage) used by the biomedical applications are coordinated with the Grid middleware and a Grid runtime system.

We use data and user information to enable ViroLab users to verify and identify the origin (provenance) of data and to re-run experiments when required. We extend this feature by categorizing the level of information including the data and workflow process. Examples of saving provenance information are: keeping track of the level of information to be saved, the format of information and where to save it, dynamic data and parameter changes during the run and in time, saving workflow instances, and the information on how and by whom the run was made. The collected provenance information is archived in our portal and accessible through search and discovery methods.

Technical requirements for building such a system include efficient data management, integration and analysis, error detection, recovery from failures, logging information for each workflow, allowing status checks on running workflows, on the fly updates, detached execution of data- and compute-intensive tasks, visualization and image processing on the data flowing through the analysis steps, semantics and metadata-based data access, authentication and authorization. Additional technical challenges arise through the introduction of different, heterogeneous distributed network computing systems, data sources and instruments.

2.3 Interactivity in ViroLab

In the context of *ViroLab*, an important research problem is the availability of Grid services and tools for interactive [12] compute- and data-intensive applications. Here we build on the EU IST CrossGrid Project (www.crossgrid.org) that developed a unified approach for running interactive distributed GRID applications by providing solutions to these issues:

- automatic porting of applications to Grid environments,

- user interaction services for interactive startup of applications, online output control, parameter study, and runtime steering,
- advanced user interfaces that enable easy plug-in of applications and tools, like interactive performance analysis combined with on-line monitoring,
- scheduling of distributed interactive applications,
- benchmarking and performance prediction, and optimization of data access to different storage systems.

These functionalities were recently tested in a system that supports Grid-based vascular blood flow reconstruction through by-pass surgery, which automates the process flow of MRI scan-data, 3D visualization and virtual by-pass creation and evaluation [4]. The developed computational components were executed efficiently as a custom-built application using the developed CrossGrid infrastructure, thus supporting scientists to carry out their scientific processing flows and run their analyses on both local and distributed resources. Once a process flow has been developed, it can be re-used, shared and revised/modified by members of the virtual organization who has access to the resources that the tasks were distributed to.

2.4 Workflows as a System Science Language for ViroLab.

While an increasing number of computational tools for distributed computing in science have become available in the last couple of years, they are mostly at an infrastructural level making it difficult to use for the domain scientist. Scientific workflow environments [13,14] improve this situation by enabling usage of different tools and technologies in a user-friendly, visual programming environment. These environments provide domain-independent customizable graphical user interfaces to combine different e-Science technologies along with efficient methods for using them, and thus increase the efficiency of the scientist and promote scientific discovery. A custom-built application approach is not sufficient when applications are becoming more and more complex. Service-based distributed applications are ideal to automate and generalize using scientific workflows. They can be used to combine data integration, analysis, and visualization steps into larger, automated 'knowledge discovery pipelines' and 'Grid workflows'.

One of our goals in building an interactive scientific workflow environment for *ViroLab* is to add flexibility and extensibility and adding service-oriented interfaces through a workbench-style collaborative portal, such that any user with the right privileges can use the set of applications and datasets. One very important issue is to be able to register and publish derived data and processes, and to keep track of the provenance of information flowing through the generated pipelines as well as accessing existing (patient and scientific literature) data and acquiring new data from scientific instruments. These domain-independent features can be customized for a *specific* scientific domain by adding domain specific components and semantic annotation of the components and the data being used.

2.5 Semantic Assistance

Finally a very important feature needed to automate the construction of workflow applications is the ability to generate ontological descriptions of services, system components and their infrastructure [15]. Usually, all semantic data is stored in a form of registry that contains OWL-based descriptions of service class functionality, instance properties and performance records. The user provides a set of initial requirements about the workflow use, and then an abstract workflow is built using the knowledge about services' functionality that has been supplied to the registry by service providers. Subsequently, the semantic information on service properties, which results from analyzing the monitoring data of services and resources, is applied to steer running workflows that

still have multiple possibilities of concrete Web service operations. The selection of the preferable service class results from comparison of semantic descriptions of the available services classes and the matching of features of the classes to the actual requirements.

3. Results

'Progress in natural sciences comes from taking things apart; progress in computer science comes from bringing things together.' [16]

In *ViroLab*, we need statistical and immunological models to study the dynamics of the HIV populations and molecular dynamics models to study drug affinities, in addition to rule-based and parameter-based decision support. We enhanced *ViroLab* by adding cellular automata (CA) and molecular dynamics modeling of HIV infection and AIDS onset. These approaches render the highly dimensional and complex data more amenable.

3.1 HIV Simulation

A mesoscopic model to study the evolution of HIV infection and the onset of AIDS is used in ViroLab. The model takes into account the global features of the immune response to any pathogen, the fast mutation rate of the HIV, and a fair amount of spatial localization, which may occur in the lymph nodes. Ordinary (or partial) differential equation models are insufficient for describing the two extreme time scales involved in HIV infection (days and decades), as well as the implicit spatial heterogeneity. We developed a non-uniform Cellular Automata model to study the dynamics of drug therapy of HIV infection, which simulates four-phases (acute, chronic, drug treatment response and onset of AIDS). Three different drug therapies (mono-therapy, combined drug therapy and highly active antiretroviral therapy) can also be studied in this model. Our model for prediction of the temporal behavior of the immune system to drug therapy qualitatively corresponds to clinical data [17].

3.2 BioStatistics

The bio-statistical analysis of the HIV-1 genotype datasets aims to identify patterns of mutations (or naturally occurring polymorphisms) associated with resistance to antiviral drugs and to predict the degree of in-vitro or in-vivo sensitivity to available drugs from an HIV-1 genetic sequence. The statistical challenges in doing such analyses arise from the high dimensionality of these data [18]. Direct application of the well-known mathematical approaches to analysis of HIV-1 genotype results in a lot of problems. The problem stems from the fact that in HIV DNA analysis, the main scope of interest is the so-called relevant mutations, a set of mutations associated specifically with the drug resistance. These mutations might exist in different positions over the amino-acid chains. Moreover, the sheer complexity of the disease and data require the development of the reliable statistical technique for its analysis and modeling.

3.3 Decision Support System and Presentation

The output of our initial version of ViroLab, consists of a prediction of the drug sensitivity of the virus generated by comparing the viral genotype to a relational database containing a large number of phenotype-genotype pairs. The decision software interprets the genotype of a patient by using rules developed by experts on the basis of the literature, taking into account the relationship of the genotype and phenotype. In addition, it is based on available data from clinical studies and on the relationship between the presence of genotype and the clinical outcome.

A Proxy and J2ME method is implemented for accessing ViroLab from mobile devices, thus lowering the barrier to access the system. A mini-navigator script is responsible for

taking the patient data and communication with the remote server where the ranking takes place. Figure 5 shows a typical output of the ranking using the latter method.

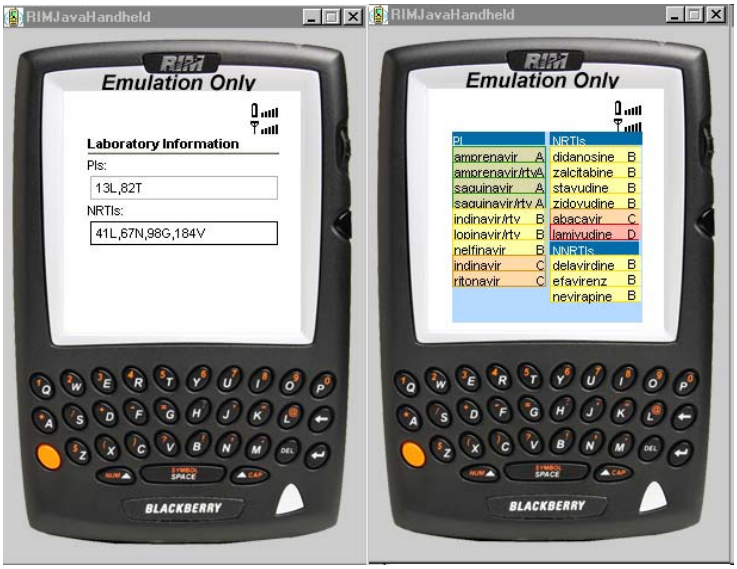

Figure 5. Presentation of drug rankings. Left: patient specific Reverse Transcriptase and Protease mutations. Right: resulting drug ranking

4. Discussion and Future Work

With the increasing availability of genetic information and extensive patient records the time has come to study diseases from the DNA level all the way up to medical responses. What is more, the long-standing challenges of individual-based, targeted treatments come within reach. What is necessary is to provide integrating technology to the medical doctors and researchers bridging the gaps in multi-scale models, data fusion and cross-disciplinary collaboration.

Indeed, one of the main issues for near future work is collaboration enhancement. Collaboration technology aims to enhance the productivity and effectiveness of multi-disciplinary biomedical research. To this effect, Grid technology offers the possibility of leveraging computing tools into distributed collaborative environments, or collaboratories. The basic collaboration functionality for ViroLab users is offered by basic portlet functionality, such as the one provided by the GridSphere installation. That is, a flexible XML-based portal presentation description can be easilymodified to create customized portal layouts, built-in support for Role Based Access Control (RBAC), separating users into guests, users, administrators and super users, and integrated JUnit/Cactus unit tests for complete server side testing of portlet services including the generation of test reports. The basic collaboration features are seamlessly integrated in the ViroLab Grid portal, offering functionality such as interactive chat, file sharing and whiteboard applets.

This system science approach is studied in ViroLab, where a prototype personalized drug-ranking system is used, as presented in this paper. Although the research is still in its infancy, interesting results have already been obtained, showing the viability and the extensibility of the approach taken. The system we described is still under development with new functionalities added frequently through extensive usability studies in a series of hospitals across Europe.

Acknowledgements

The authors would like to thank the *ViroLab* consortium and in particular Dr. M. Bubak from the University of Amsterdam, Ilkay Altintas from San Diego Super Computing Center, and Dr. D. van de Vijver from the University Medical Centre Utrecht, The Netherlands. This research was supported by the Dutch Bsik project VLe: Virtual Laboratory for e-Science and the European Commission *ViroLab* grant INFSO-IST-027446. We would like to thank Carl Kesselman and Ian Foster for proofreading and many valuable suggestions to improve the overall quality of the paper.

References

[1] W.H. Frist, "Health Care in the 21st Century", New England Journal of Medicine, Vol. 352, Jan. 2005, pp. 267-272

[2] A. Finkelstein, J. Hetherington, L. Li, O. Margoninski, P. Saffrey, R. Seymour and A. Warner, “Computational Challenges of System Biology”, Computer, Vol. 37, No. 5, 2004, pp. 26-33.

[3] A. Barabasi, “Taming Complexity”, Nature Physics, Vol 1, November 2005, pp 68-70.

[4] A. Tirado-Ramos, P.M.A. Sloot, A.G. Hoekstra and M. Bubak, “An Integrative Approach to High-Performance Biomedical Problem Solving Environments on the Grid”, Parallel Computing, Special Issue on High-Performance Parallel Bio-computing, Vol. 30, nr 9-10, 2004, pp. 1037-1055.

[5] P.M.A. Sloot, A.V. Boukhanovsky, W. Keulen, A. Tirado-Ramos and C.A. Boucher, “A Grid-Based HIV Expert System”, Journal of Clinical Monitoring and Computing, Vol. 19, Numbers 4-5, October 2005 pp. 263 – 278.

[6] A.J.G. Hey and A.E. Trefethen, “The Data Deluge: An e-Science Perspective”, Grid Computing - Making the Global Infrastructure a Reality, chapter 36, 2003, pp. 809-824.

[7] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations", International Journal of High Performance Computing Applications, Vol. 15, 2001, pp. 200-222.

[8] I. Altintas, A. Birnbaum, K.K. Baldridge, W. Sudholt, M. Miller, C. Amoreira, Y. Potier and B. Ludäscher, “A Framework for the Design and Reuse of Grid Workflows”, SAG 2004, Beijing, China, September 20-24, 2004, Vol. 3458 (3), pp. 119-132.

[9] F. Neubauer, A. Hoheisel and J. Geiler, “Workflow-based Grid applications”, Future Generation Computer Systems, Vol. 22, Issues 1-2, January 2006, pp. 6-15.

[10] ViroLab, EU project INFSO-IST-027446, http://www.virolab.org/.

[11] S.G. Deeks, “Treatment of antiretroviral-drug-resistant HIV-1 infection”, Lancet, 2003, 362(9400), pp. 2002-2011.

[12] M. Bubak, M. Malawski, K. Zajac, “Architecture of the Grid for Interactive Applications”, Lecture Notes in Computer Science, Vol. 2657, Part I, Springer, 2003, pp. 207-213, www.eu-crossGrid.org.

[13] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao and Y. Zhao, “Scientific Workflow Management and the Kepler System”, Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, 2005, http://kepler-project.org/.

[14] M. Bubak, T. Gubala, M. Kapalka, M. Malawski, K. Rycerz, “Workflow Composer and Service Registry for Grid Applications”, Future Generation Computer Systems, Vol.21, no. 1, 2005, pp. 77-86.

[15] OntoGrid Project, http://www.ontogrid.net, and K-WfGrid Project, http://www.kwfgrid.net.

[16] W.D. Hillis, “New Computer Architectures and Their Relationships to Physics", International Journal of Theoretical Physics, Vol. 21, 1982, pp. 255-62.

[17] P.M.A. Sloot, F. Chen and C.A. Boucher, “Cellular automata model of drug therapy for HIV infection”, Lecture Notes in Computer Science, Vol. 2493, October 2002, pp. 282–293.

[18] T.E. Scheetz, N. Trivedi, K.T. Pedretti, T.A. Braun and T.L. Casava, “Gene transcript clustering: a comparison of parallel approaches”, Future Generation Computer Systems, Vol. 21, Issue 5, May 2005, pp. 731-735.