SARS analysis on the Grid

Vasa Curcin, Moustafa Ghanem and Yike Guo
Department of Computing
Imperial College London
180 Queens Gate
London, SW7 2AZ
United Kingdom

Wayne He, Yuanyuan Li, Hao Pei, Lu Qiang

Shanghai Center for Bioinformation Technology
100 Qinzhou Rd. Fl.12
Shanghai 200235
People's Republic of China

Abstract

In this paper we show how DiscoveryNet acts as Grid middleware when applied to real-world scientific problems such as the research into the evolution of the SARS epidemic. The analysis presented was realized as part of collaboration between DiscoveryNet project and the Shanghai Center for Bioinformation Technology (SCBIT), where DiscoveryNet provided the software for the SCBIT users to create their analysis and map it to local and Grid resources. Data analysis workflows constructed using Discovery Net to integrate and co-ordinate a large number of bioinformatics tools and services for the analysis are described in detail and at the main analysis results are discussed.

1. Introduction

Severe Acute Respiratory Syndrome, or SARS, has spread rapidly from its site of origin in Guangdong Province, in Southern China, to a large number of countries throughout the world. Initial symptoms of the disease include high fever, malaise, rigor, headache, and a non-productive cough, and later can progress to generalized interstitial infiltrates in the lung, requiring incubation and mechanical ventilation. The mortality rate among patients meeting the current case definition of SARS is approximately 15%, while it can reach 50% for patients who are 60 years of age or older. Epidemiological evidence suggests that the transmission of this newly emerging pathogen occurs mainly through close contact, although other routes of transmission cannot be excluded.

An unprecedented worldwide collaboration led by the World Health Organization (WHO) has been established to study this threat to human health, involving hundreds of scientists around the world. Through their efforts, we now know that a novel coronavirus is the causative agent of SARS (SARS-CoV) [1]. One important part of this research was establishing the relationship between observed genomic variations in strains taken from different patients, and the biology of SARS. As described in [2] epidemiological and genetic evidence for viral adaptation to human beings was collected through molecular investigations of the characteristic viral lineages.

DiscoveryNet project [3], in collaboration with researchers from SCBIT [4] (Shanghai Centre for Bioinformation Technology), was actively involved in this research and the platform was subsequently used to capture the analysis based upon a service-based computing infrastructure and to serve as framework for further investigations over the collected data. In this paper we describe the data analysis workflows constructed using Discovery Net to integrate and co-ordinate a large number of bioinformatics tools and services for the analysis. We conclude by discussing the main analysis results.

2. Data

The main purpose of the workflows presented was to combine the sequence variation information on both genomic and proteomic level, and to use the available public annotation information to establish the impact of those variations on the SARS virus development. The data used consists of 31 human patient samples, 2 strains sequenced from palm civet samples, which were assumed to be the source of infection and 30

sequences that were committed to Genbank at the time of the analysis, including the SARS reference sequence (NC004718). The reference nucleotide sequence is annotated with the variation information from the samples, and overlaps between coding segments and variations are observed. Furthermore, individual coding segments are translated into five proteins that form the virus (Orf1A, Orf1B, S, M, E, N) and analysis is performed comparing the variation in these proteins in different strains.

3. Analysis

The first workflow deals with the basic data processing and alignment. As a first step we collect all known samples from NCBI (30 of them) and the 33 sample sequenced in the sequencing lab. Some of the NCBI samples have been partially annotated, but we are only using the annotations from the reference SARS genome sequence (NC004718) to avoid confusion.

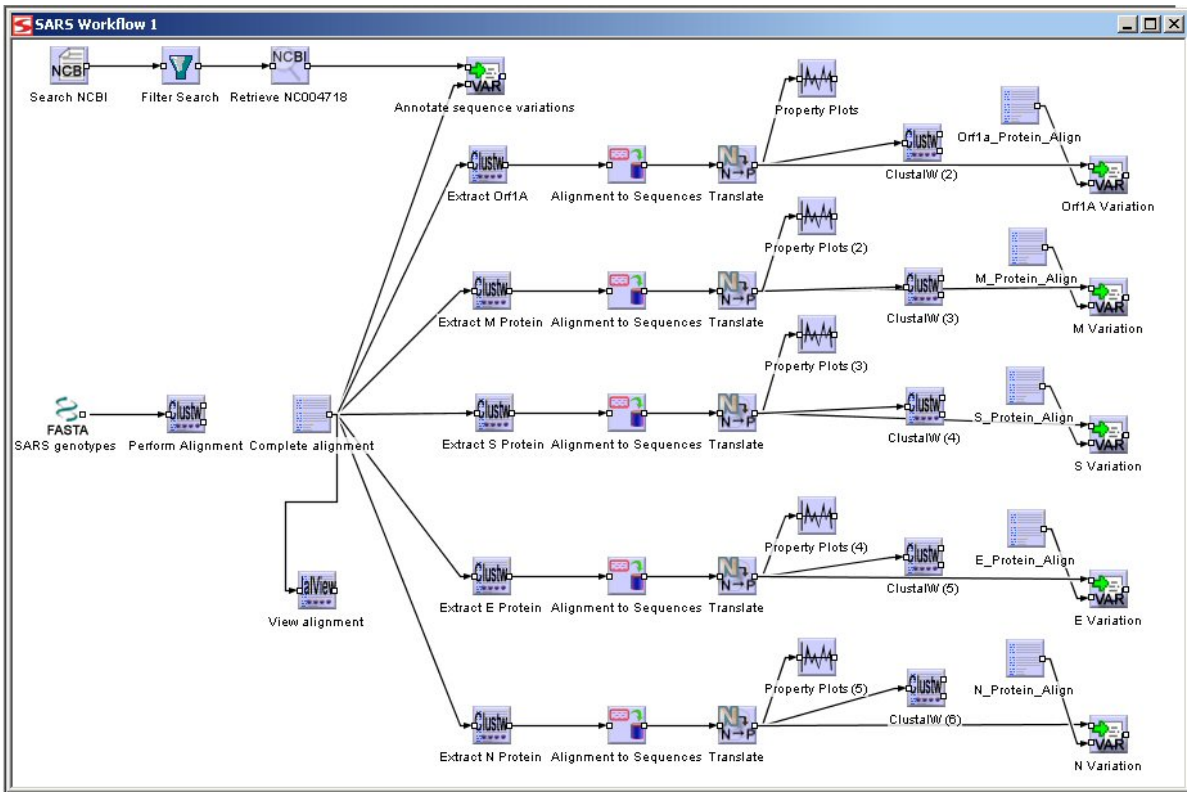

Figure 1: Nucleotide-level analysis

Then, we align all the samples in order to find the variation points, insertions and deletions. This is a time-consuming process, and the ClustalW component performing it has several implementations including local version and the distributed version. In this execution we used the Grid to invoke the distributed version on a remote cluster, reducing the calculation time from 3 days on a standard desktop machine to several hours.

It is interesting to note that a related project, Grid Application Development Software (GrADS [8]), also chose a sequence alignment program (FastA) to demonstrate the usefulness of treating sequence problems in the Grid environment. The motivation for moving to the Grid is the fundamental ambitiousness of these tasks – very often there is a need to search massive sequence databases such as latest editions of Genbank or SwissProt, to learn the nature of novel sequences. In the SARS case, we are dealing with highly similar sequences, which require a more sensitive comparison, such as ClustalW alignment, which is even more computationally intensive.

The variety of ClustalW chosen in this experiment was ClustalMPI, that parallelised all three steps involved in a typical Clustal operation: pair wise alignment, guide-tree generation and progressive alignment. Message Passing Interface (MPI) was used to achieve the parallelisation of the algorithm, thereby enabling it to run on both parallel CPUs and distributed platforms.

Since we are aligning the reference sequence as well, we can transfer this information from the alignment to the reference sequence. Then, we can use a visualization tool to see where these variation points and indels are located and how they overlap with significant features such as CDS-s. By analyzing the result we discover which area varies easily and which one is conservative. The results are shown in a sequence viewer.

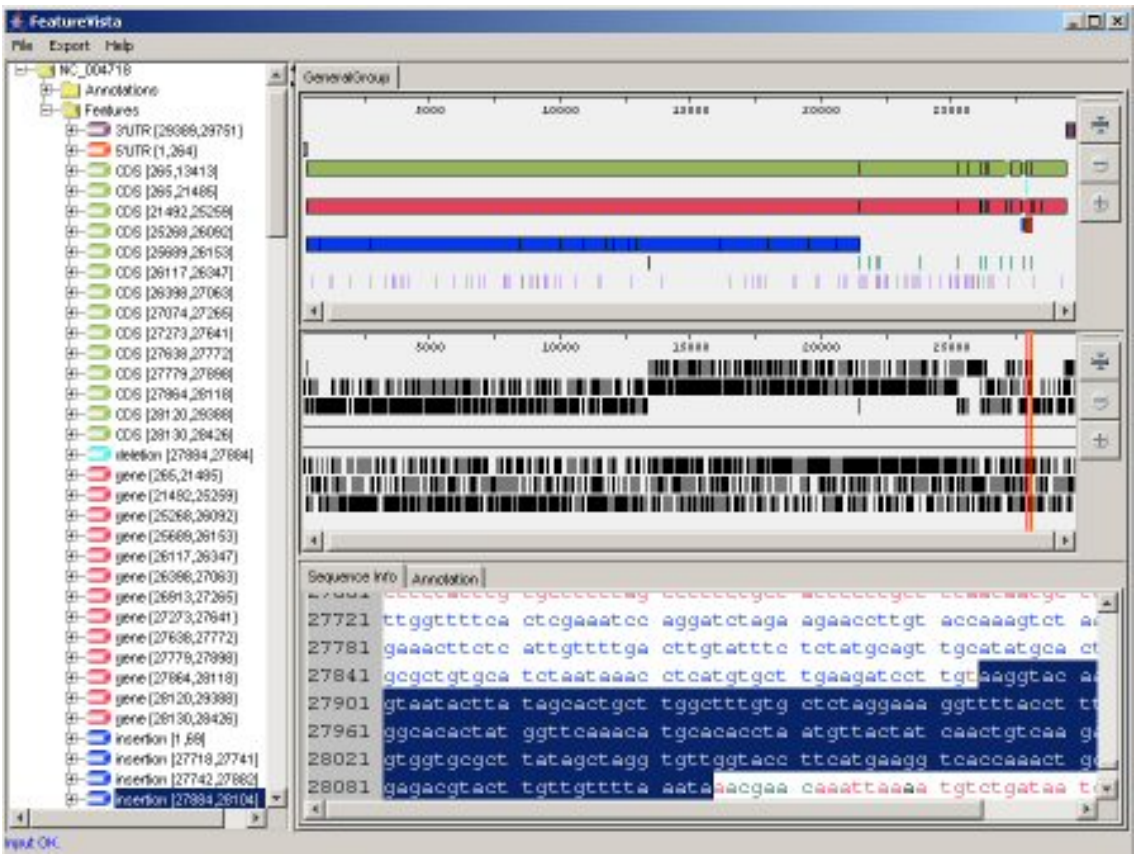

Figure 2: Reference information with added variation data

The sequence viewer used is FeatureVista, a visualisation component of the DiscoveryNet bioinformatics toolkit. It ties to the object model present in the system to achieve intuitive representation of the characteristics of nucleic and protein sequences.

Following the analysis of the variation on nucleic acid level, we focus on analyzing the interesting genes. We can extract the partial alignment out to isolate the section of the alignment which belongs to a particular gene. As these represent the same gene in all samples (with broadly the same structure), there is a very high similarity between them, which is why multiple alignment can be performed directly, rather than using BLAST or some other less sensitive similarity tool. Then, translation of these nucleic sequences is performed and proteins produced by each one of them are obtained. Note that this method is independent of the actual gene we choose to analyze.

The analysis then moves to the protein level. Again, we are dealing with very similar proteins, so we can align them to find out the variation information on this level. Once we have done this, we want to annotate the selected protein (S-Protein in this example) with a number of proteomic tools, to provide a comprehensive overview of various annotations and how they overlap with the conservative and variable segments of the sequence. The tools that we choose in this example are various applications from the EMBOSS package which have been integrated using XIF command-line tool framework. The main idea of the framework is to remove the need for programming from component construction in DiscoveryNet – by providing XML descriptors for tools that need to be integrated, mapping input and output parameters to input data and user parameters and, more interestingly, specifying the data format transformations and metadata produced by the tool.

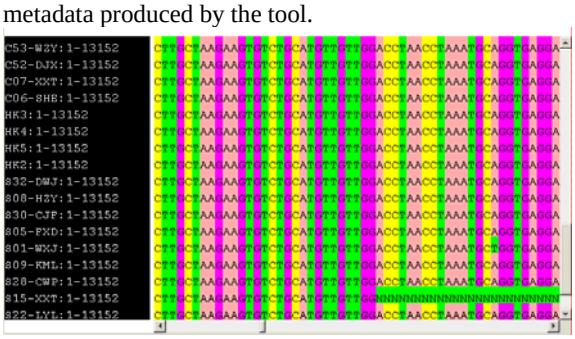

Figure 3: Alignment of patient samples of Orf1A gene

The integration itself was being done on-the-fly, taking on average 5 minutes per tool and adding the components to the server at runtime. The annotations are performed in parallel, allowing for the execution on different physical resources, and then merged and joined with the annotation information.

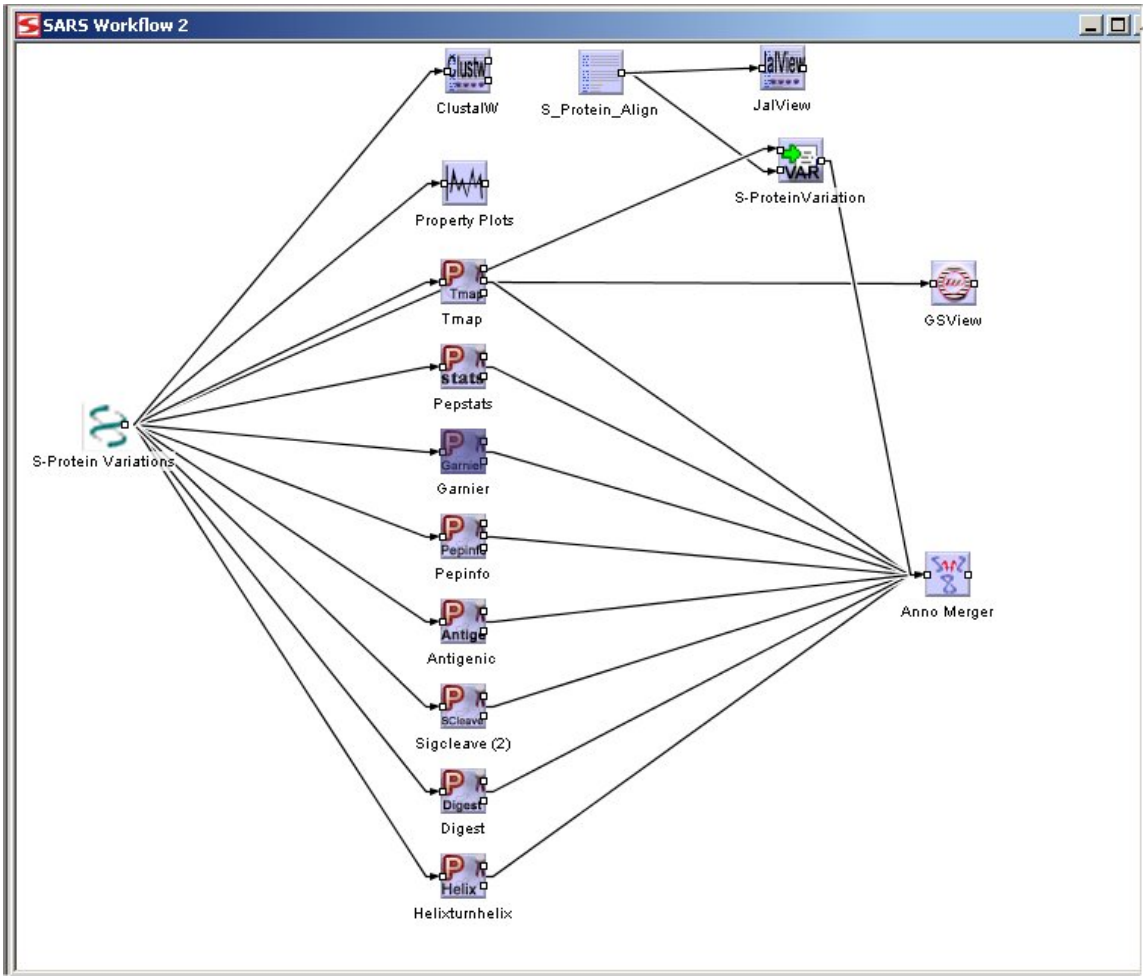

Figure 4: Protein-level analysis

Finally the annotated sequence collection is stored back into the repository, either a relational database, dedicated sequence database or a generic DiscoveryNet userspace.

4. Technology

DiscoveryNet represents a service composition pipeline framework running on top of an application server. The components within this framework represent elements of the workflow, which can be resource bound (implementation of a particular algorithm running on a particular machine) or resource free (executable Java code). Furthermore, the component can be atomic, or assembled from existing components and packaged as a new component (this packaging is performed on the instantiated components which are then given new degrees of freedom).

These capabilities make DiscoveryNet a superset of service composition tools (such as Imperial College e-Science Networked Infrastructure, ICENI [5] that are often mentioned in the context of the Grid technology.

Even though DiscoveryNet internally represents all the applications developed with it as workflows/dataflows, it is still just a middleware feature of the system, which is not directly connected to its component aspects. We will keep it out of the discussion for a moment, and concentrate on the component platform aspects.

Component within the DiscoveryNet is represented as a *node*. Its composition, integrity and execution logic are all captured within the node definition that is hidden from the user. Construction of a node can either be performed through Java code, or with the help of component-building tools provided.

Component frameworks are represented as groups of related nodes, with a well-defined set of input and output types, that cover a particular function, such as sequence alignments.

Component architectures are families of frameworks that share certain data models and modes of communications between themselves to support cross domain analysis. It is their role to specify allowed links with the underlying server functionality.

Deployment of a component within the platform is performed either by one of the builder tools, or by physically placing a compiled Java file in a specified location on the server.

Execution happens on the workflows constructed from instantiated components. The graph representing the workflow is interpreted and the required node result is produced.

Finally, executable workflows can be packaged as components in their own right. Since, at this stage, we have components that have already been instantiated, we are now, effectively, removing a part of their instantiation, by introducing degrees of freedom, that will be used to instantiate the new, composite, node.

DiscoveryNet started as a component-based data-mining system, effectively a component framework. With the introduction of cross domain node collections, it moved away from a single component framework to a component architecture, containing a number of frameworks. Finally, opening-up of its SDK toolkit moved it onwards into a component platform for designing component system architectures and hosting component systems built with it.

4.1 Component abstraction in DiscoveryNet

The key element of the DiscoveryNet view of components is that it allows the decoupling of abstract component definitions from its actual implementations. This separation of component interface from the component implementation is at the heart of any component architecture, but in DiscoveryNet, we introduce one additional level in-between in order to facilitate the workflow definition process. So, a component is defined in three decreasing levels of abstraction:

Connectivity defines the input and output ports of the component, specifying the data types that can be passed on those ports. These can be relational tables, structured non-relational data representing domain-specific information (sequences with annotations), analysis results or even generic binary objects. In this way, component is enabled to be composed with other components.

Metadata declaration works on the metadata being passed in and out of the components, as described in the section below. The metadata of the input is verified to ensure that it satisfies the logical constraints defined within the component and the output metadata is specified so that the other components connected to it can do the same. Various user-definable parameters of a component also belong to this level, since they strongly influence the definition logic, but they can also sometime work on the connectivity level (by changing the port number or type).

Execution specification defines where and how a node is executed. Separation of this level enables the user to define workflow graphs without having to decide if the node will be executed locally, as a Web service, Grid service, or at a dedicated resource.

With this separation in place, the analysis pipeline is defined in the terms of the functional aspects of the components. Thus, a component such as 'Multiple sequence alignment' used in the SARS example above may be an element in the workflow, taking as its input a set of sequences, producing an alignment object, with an associated set of parameters; based on some criteria (user-defined or automatic), it will be executed on the appropriate resource and in the appropriate form ClustalW or ClustalMPI distributed implementation [8].

4.2 Relationship to other views on DiscoveryNet

Through the development of DiscoveryNet platform, it has been used for design and deployment of various projects in areas as diverse as geodesics, finances, life sciences, oil and gas mining and business intelligence. Also, the technological aspects of DiscoveryNet presented would concentrate on the features relevant to the work at hand. Here we will give a brief overview of how these different aspects relate to each other and how they fit with the design that is presented in this work.

4.3 DiscoveryNet as a service composition environment

This particular usage of DiscoveryNet is very close to its component platform aspect. Issues of component definition and composition are common to both, but while generic component architectures are only concerned with specific data interchange and invocation issues within the individual framework designs, this aspect focuses on one particular subset of component architectures, where each component is a web or grid service. The data format used for data interchange is based on XML/SOAP and a registry service needs to support UDDI.

Also, while generic component approach that we are analysing here is concerned with the separation of component definition and implementation only on the design level, service composition within DiscoveryNet provides advanced methods for creating abstract components that are looking up its execution mechanism at runtime. While this is certainly interesting, it is not a generic component issue.

Finally, one incarnation of this aspect concentrates exclusively on the Grid environment, where each component is a service registered with a UDDI registry, and where workflows can be deployed as Grid-enabled components in such a registry and used by other Grid platforms

4.4 DiscoveryNet as a query/schema creation tool

DiscoveryNet has repeatedly been used to integrate data from various domains, using a number of tools provided: relational database support, web database access and local and remote data source support. Combination of these tools with generic data transformation components produces a dynamic

distributed query construction tool, where a component workflow is observed as a complex query.

Another usage of DiscoveryNet in this area is concerned with capturing the schema created by these queries. A component connects to the database and creates tables and views represented by this query. Thereby, DiscoveryNet can be used as a schema creation tool.

4.5 DiscoveryNet as a visual analysis capture mechanism

As described above, DiscoveryNet represents all of its analysis as pipelines constructed from nodes. Different modes of analysis, such as building classifiers while immediately observing how performance changes with different parameters, creating a graph to show the analysis results or observing how a particular clustering technique partitions a data set, are enabled through the concept of *studios* that provide a different user interface, similar to a desktop tool, where user's actions are recorded and captured. When the studio tool is closed, the actions and results are stored as components in the workflow. This connects directly to the area of information visualisation and analysis tracking and capture.

4.6 DiscoveryNet and Grid

Component architectures have been operating on ever-increasing levels of abstraction, starting from modular pieces of code, via well-defined libraries and executable units finally moving to high-level entities that capture elements of design logic. Concrete definitions change together with the level of abstraction that the observer chooses to adopt.

With DiscoveryNet, service composition, service publishing, tool integration, knowledge warehousing, visual analysis and other technological paradigms have been unified on the common abstraction level. This in itself is an achievement, but it is with the presence of the Grid as the resource provision environment that this approach realises its full potential.

Given the generic nature of DiscoveryNet components, they are easily placed over Grid middleware. Grid components are integrated into the DiscoveryNet environment: data sources are composed using the metadata provided; Web and Grid services can be used interchangeably to define analysis tasks and complex queries. Consequently, DiscoveryNet is then accessed through the Grid: the analysis tasks become Grid services, execution environment can accept Grid jobs and run them.

This approach makes the system usable in combination not only with Grid components but with other Grid middleware as well, opening up the field and not forcing the users to restrict themselves to a single middleware solution.

5. Mining the Grid

It was mentioned that Discovery Net offers warehousing capabilities - all the analysis that are designed and constructed in the system, are captured and indexed with rich provenance information, including a full record of all the changes applied to the task with timestamps and user information for each change. The warehoused workflows can then be queried and analysed to gain insight into the ways that the system is being used and how frequently particular pipelines are being executed. Moreover, each discovery result is stored with its associated workflow, parameter setting and all the relevant provenance information.

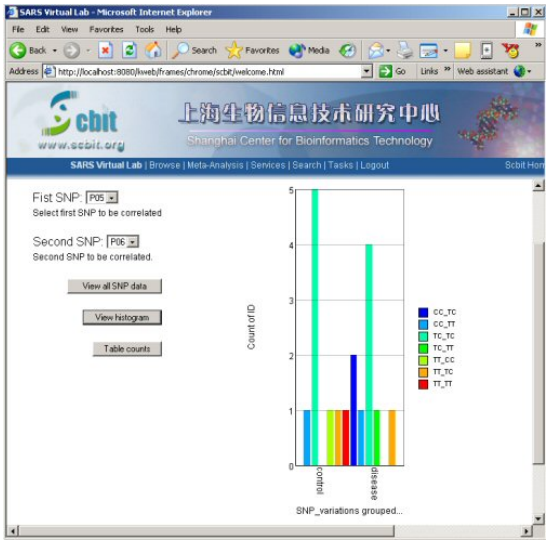

Figure 5: SNP Analysis

In that way, researchers can build their own pipeline libraries and make them accessible at various levels of granularity within the organization, Supporting this strategy is the hierarchical structure of actual components, which can be constructed at runtime from existing components without any coding involved. Such complex components are distributed via the registry service in the same way as atomic ones and can be reused.

When this concept is applied to the Grid, the libraries become virtual bookmark collections – entry points to the Grid analysis environment, that offer the user the selection of favourite tasks, together with the status of the analysis he is currently running, effectively providing a Grid browser.

The SARS examples detailed above are currently being placed within a SARS portal environment that will be updated as new data sets and analytical techniques become available. The longer term ambition of this collaboration is to establish a virtual Grid based research environment for investigating viral epidemics, storing analytical methods which have proven successful in the past and making them easily applicable to the problem at hand.

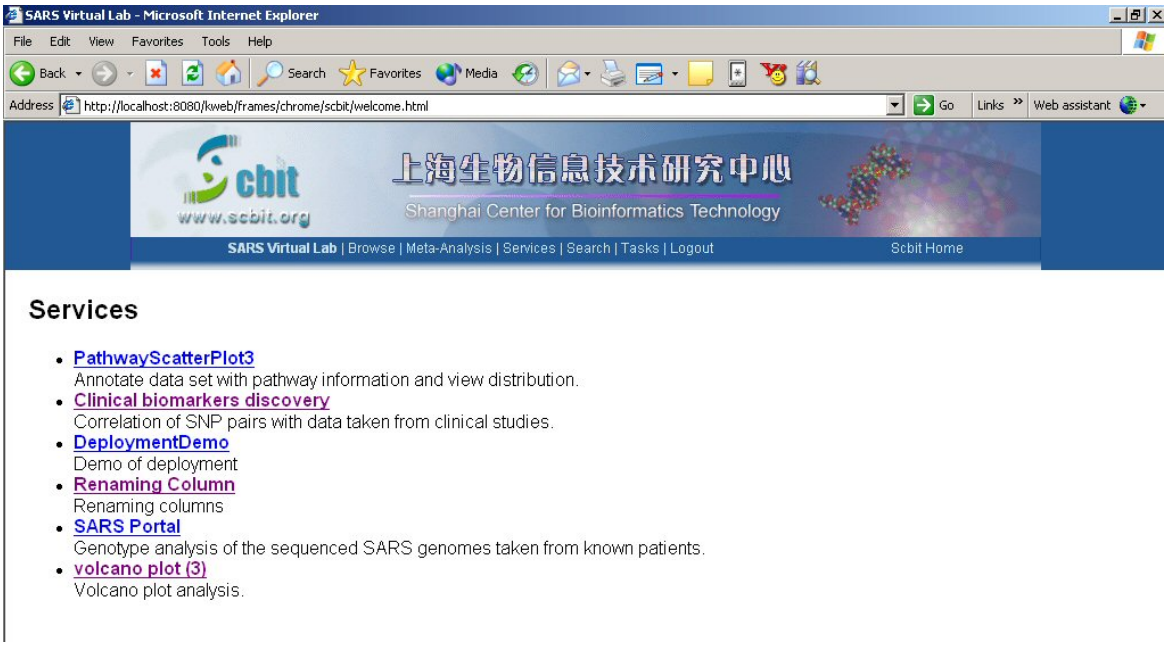

Figure 6: SARS Virtual Lab

6. Results and Discussion

The benefits of the DiscoveryNet technology were shown on the example of two SARS workflows. The first provides the overview of the analysis conducted, allowing that each step be modified and tweaked in a controlled manner, storing the change information (user, parameter changed, time) and visualising the data in real-time to observe the significance of the change. Also, it is easy to replace the reference sequence in the example with a novel SARS strain sequence and automatically obtain the annotation data and its significance. The latter workflow is reusable in a similar way, allowing automated distributed protein annotation to be combined with the variability information.

This research initiative is notable both for its scope and for the opportunity it has provided to evaluate the efficacy of post-genomic technologies applied to a new disease with epidemic and even pandemic potential.

Currently, the investigative methods used are being added to the SARS Virtual Lab within the SCBIT institute, providing a testbed for future development of DiscoveryNet technology.

References

[1] Ksiazek, Thomas G. and Erdman, Dean and Goldsmith, Cynthia S. and Zaki, Sherif R. and Peret, Teresa and Emery, . Shannon and Tong, Suxiang and Urbani, Carlo and Comer, James A. and Lim, Wilina and Rollin, Pierre E. and Dowell, Scott F. and Ling, Ai-Ee and Humphrey, Charles D. and Shieh, Wun-Ju and Guarner, Jeannette and Paddock, Christopher D. and Rota, Paul and Fields, Barry and DeRisi, Joseph and Yang, Jyh-Yuan and Cox, Nancy and Hughes, James M. and LeDuc, James W. and Bellini, William J. and Anderson, Larry J. and the SARS Working Group. A Novel Coronavirus Associated with Severe Acute Respiratory Syndrome New England Journal of Medicine.Vol. 348 No. 20, pp 1953-1966, 2003.

[2] Tsui, Stephen K.W. and Chim, Stephen S.C. and Lo, Y.M. Dennis and the Chinese University of Hong Kong (CUHK) Molecular SARS Research Group. Coronavirus Genomic-Sequence Variations and the Epidemiology of the Severe Acute Respiratory Syndrome. New England Journal of Medicine.Vol. 349 No. 2, pp 187-188, 2003.

[3] V. Curcin, M. Ghanem, Y. Guo, M. Kohler, A. Rowe, J Syed, P. Wendel. Discovery Net: Towards a Grid of Knowledge Discovery. Proceedings of KDD-2002. The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. July 23-26, 2002 Edmonton, Canada.

[4] SCBIT, http://www.scbit.org.

[5] Nathalie Furmento, Anthony Mayer, Stephen McGough, Steven Newhouse, Tony Field and John Darlington. ICENI: optimisation of component applications within a Grid environment. Journal of Parallel Computing. Vol. 28 No. 12, pp 1753-1772, 2002.

[6] Li, Kuo-Bin. ClustalW-MPI: ClustalW analysis using distributed and parallel computing. Journal of Bioinformatics. Vol. 19 No. 12. pp 1585-1586. 2003.

[7] The Chinese SARS Molecular Epidemiology Consortium. Molecular Evolution of the SARS Coronavirus During the Course of the SARS Epidemic in China. Science, Vol. 303, Issue 5664, pp 1666-1669, 12 March 2004.