Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Biomedical Informatics 39 (2006) 440–450
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

KDE Bioscience: Platform for bioinformatics analysis workflows

Qiang Lu [a,c,*], Pei Hao [b], Vasa Curcin [c], Weizhong He [b], Yuan-Yuan Li [b], Qing-Ming Luo [a], Yi-Ke Guo [c], Yi-Xue Li [b]

[a] School of Life Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Shanghai Center for Bioinformation Technology, 12 Floor, 100 Qingzhou Road, Shanghai 210235, China
[c] Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2BZ, UK

Received 18 July 2005
Available online 11 October 2005

Abstract

Bioinformatics is a dynamic research area in which a large number of algorithms and programs have been developed rapidly and independently without much consideration so far of the need for standardization. The lack of such common standards combined with unfriendly interfaces make it difficult for biologists to learn how to use these tools and to translate the data formats from one to another. Consequently, the construction of an integrative bioinformatics platform to facilitate biologists' research is an urgent and challenging task. KDE Bioscience is a java-based software platform that collects a variety of bioinformatics tools and provides a workflow mechanism to integrate them. Nucleotide and protein sequences from local flat files, web sites, and relational databases can be entered, annotated, and aligned. Several home-made or 3rd-party viewers are built-in to provide visualization of annotations or alignments. KDE Bioscience can also be deployed in client-server mode where simultaneous execution of the same workflow is supported for multiple users. Moreover, workflows can be published as web pages that can be executed from a web browser. The power of KDE Bioscience comes from the integrated algorithms and data sources. With its generic workflow mechanism other novel calculations and simulations can be integrated to augment the current sequence analysis functions. Because of this flexible and extensible architecture, KDE Bioscience makes an ideal integrated informatics environment for future bioinformatics or systems biology research.

Keywords: Integration; Bioinformatics; Platform

1. Introduction

The rapid development of genome technologies, especially automatic sequencing techniques, has produced a huge amount of data consisting essentially of nucleotide and protein sequences. For instance, the number of sequences in GenBank increases exponentially and as of August 2003 (Release 137) it contained over 33.9 billion nucleotide bases from 27.2 million individual sequences [1]. To store, characterize, and mine such a large amount of data requires many databases and programs hosted in high-performance computers. Until now, there have been several databases, for example GenBank [1], Uniprot [2], PDB [3], KEGG [4], PubMed Medline, etc., covering not only nucleotide and protein sequences but also their annotations and related research publications. The programs include those for sequence alignment, prediction of genes, protein structures, and regulatory elements, etc., some of which are organized into packages such as EMBOSS [5], PHYLIP [6], and GCG Wisconsin (http://www.accelrys.com/products/gcg_wisconsin_package/program_ list.html).

In general, these databases are built independently by various academic or commercial organizations and their input and output data formats follow their own standards (e.g., Fasta, Genbank, EMBL, SRS, etc.), most of which are incompatible. The programs themselves are even more complex in that they are implemented using a variety of programming languages and on different operating systems, are operated in different ways using input and

* Corresponding author.
E-mail address: qianglu@doc.ic.ac.uk (Q. Lu).

1532-0464/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2005.09.001

output data in a wide range of formats. Biologists try to discover biological functions from sequences using informatics techniques but are frequently frustrated by the processes of searching for suitable tools, learning how to use these tools, and translating data formats between them.

To facilitate biologists' research, an integrative informatics platform is needed in which many kinds of databases and programs are integrated with a common input–output data format and uniform graphical user interface (GUI). To build such an integrative informatics platform, workflow is recognized as a potential solution. Some existing efforts include Biopipe [8], BioWBI [9], Taverna [10], Wildfire [11], etc. All of them provide mechanism to integrate bioinformatics programs into workflows. Biopipe is based on programming language perl. It looks lack of user-friendly interface for building workflow so far. BioWBI and Tarverna use Web-Services for components to construct workflows. However, to convert a 3rd-party program into Web-Services, they lack of integrative GUI environment. Wildfire aims at using workflow to provide huge computing capability to bioinformatics application. However, there is no integrative environment provided for multiple users to collaborate in the same large-scale bioinformatics project. In this paper, we present a significant integrative informatics platform, Knowledge Discovery Environment of Bioscience (KDE Bioscience), which is supposed to provide a solution of integration of biological data, algorithms, computing hardware, and biologist intelligence for bioinformatics.

2. Requirements

From the viewpoint of informatics, the requirements of an integrative informatics platform consist of four parts: integration of data, algorithms, computing hardware, and human intelligence.

The large scale of sequence and annotation data is one of the prominent characteristic of bioinformatics applications. Generally, to handle bulk data, a Database Management Systems (DBMS) is the best choice. With support for the Structure Query Language (SQL) standard, accessing the DBMS is machine-friendly in cases where the data are well-structured. However, since current biological researches generate such a large amount of data, that are usually far from complete and well-structured, it can be better to store them in flat files with a semi-structured format that allows for errors and redundancies. Additionally, biologists have become used to publishing their data on web pages that are usually in unstructured formats. Providing biologists with an easy-to-use bioinformatics platform requires the integration of sequence and annotation data in different formats from DBMS, flat files, and web pages.

Genomic data are so abundant that it defies simple intuitive analysis. Thus, many computer programs are employed to assist in such tasks including alignments of sequences, predictions of gene, protein structure, regulatory elements, and visualization. These programs reflect the up-to-the-minute progress in research and as a consequence lack standardization. Although there are a few de facto standards, too many varieties will continue to exist for the foreseeable future.

Practical bioinformatics task typically consist of some algorithms running in parallel or in series. Therefore, the key requirements of algorithms integration are (1) to collect a lot of specified programs and (2) to provide them an easy way of communications.

In bioinformatics, large-scale data needs bulk storage and time-consuming tasks such as alignment of genomes need powerful computing resources—however, such powerful hardware may be unaffordable to a single organization or ordinary researcher. A potential solution is to integrate the distributed storage and computing resources. In this sense, it is necessary for an integrative informatics platform to support distributed storage and computing.

Furthermore, it is not feasible for a genome project to be handled by only one person. Typically a team will be formed involving several experts who focus on different parts of the same project such as sequencing, micro-arrays, bioinformatics analysis, and experimental verification. Thus, it is necessary for an integrative informatics platform to provide a mechanism for biologists to work together and share data and designs—or even construct the same workflow. In addition, they can publish their designs as web pages, for access and re-use by other researchers.

3. Implementation

KDE Bioscience based on the Knowledge Discovery Environment (KDE) [7]. The basic idea is to represent a bioinformatics analysis process (task) as a workflow (pipeline) constructed from a series of linked nodes. There are two key concepts in this thought: (1) a data model, which represents data with specialized syntax, (2) nodes, which represent separate algorithms. The platform itself also provides functions for development, management, and execution of workflows.

3.1. Data model

Data in KDE Bioscience, such as sequences and related annotations are abstracted into data model. These data model, composed of data container and metadata, provide a general structure to transfer data between various programs, (or nodes) in KDE Bioscience. Metadata describes the properties of data including types, names, and structures, etc.

KDE Bioscience provides a mechanism for metadata processing that executes before the workflow operates on the actual data. Since metadata provides extra information, the data controls such as logical constraints can be implemented for workflow, for example, verification

of the compatibility of a particular algorithm to a given dataset.

There are two important data models in KDE Bioscience: sequence collection and table.

Sequence collection accommodates the main bioinformatics data of nucleotide/protein sequences and their related annotations. Considering the intrinsic linearity of the biological sequence, the sequence is represented as a string of characters, and thereafter its annotations or features are organized along the sequence with one-dimensional coordinates. To facilitate development, the interface of SequenceDB from the open source project biojava [12] is adopted as the Java interface for this container. To cope with large amounts of sequence data, our Java Class KSequenceDB implementing the SequenceDB interface is based on files in hard disk instead of memory. KSequenceDB metadata is organized as a tree, of which sequence types (DNA, RNA, or protein), names, and types of related annotations, etc. are described as leafs.

Output of most algorithms, not only those for bioinformatics but also those for general data processing can be mathematically generalized as two-dimensional matrices. Naturally, table is used to store the matrix. Table metadata specifies the column names and types. In KDE Bioscience, a Java class KResultSet is developed to create the table, which implements the interface of java.sql.ResultSet based on file system. By using ResultSet persistence can be implemented easily in a relational database. In this sense, tables provide a bridge between KDE Bioscience and other applications such as data warehouses for data mining and knowledge discovery.

There are some other specified data models involved in KDE Bioscience, such as ClustalWResult for result of alignment program clustalw, which are not to be illustrated in detail in this paper.

To explore the data several viewers can be attached for each data model using a simple configuration file. These viewers can be launched at any point of a workflow to visualize the corresponding data.

3.2. Node

Node is another basic component of a workflow. Usually, each node represents a distinct algorithm. KDE Bioscience has so far collected more than 60 commonly used bioinformatics programs covering the analysis and alignment of nucleotide and protein sequences. With the powerful software development kit (SDK) provided by KDE, algorithms with various implementations (with or without source code, hosted in local or remote machines) can be integrated. For example, two nodes for blast applications are provided, local blast and net blast and there are nodes for retrieving NCBI, PDB, and UniProt databases remotely.

All nodes are classified into several groups: Import/export, nucleotide analysis, protein analysis, remote query, alignment, visualization, and accessory tools.

3.2.1. Import and export nodes

Import and export nodes transfer sequence data between the KDE Bioscience workspace and outside—for instance, users can load and store sequence data from and to flat files with various formats such as FASTA, Genbank, EMBL, Swissprot, Genpept, and so on. Furthermore, the sequences can be imported and exported as XML or tables in any JDBC (Java Database Connectivity) supported relational DBMS. In addition, sequences can also be imported in the form of editable strings, and exported to the clipboard on Windows system. Generally, import nodes are the starting points of a workflow (task) and export nodes are the end points.

3.2.2. Nucleotide and protein analysis nodes

The nodes in these two groups provide algorithms to annotate the nucleotide and protein sequences, covering various functions: (1) nucleotide composition analysis, such as Compseq, DAN, FreekN [13], and GC3 calculation; (2) GpG island prediction, such as Cpgplot and Gpgreport [13]; (3) 2D nucleic structure prediction, such as RNAfold [14] and Einverted [13]; (4) nucleic motif analysis, such as Fuzznuc, Fuzztran, Restrict, and Tfscan [13]; (5) primer prediction, such as primer3 [15]; (6) promoter prediction, such as Neural network promoter prediction [16]; (7) repeat identification, such as Recon [17] and Repeatmasker [18]; (8) tRNA prediction, such as tRNAscan [19]; (9) gene finding, such as GeneScan [20], GetORF [13], and Glimmer [21]; (10) statistics, such as Geecee and Pepstat [13]; (11) protein composition analysis, such as Charge, Checktrans, Compseq, Freak, Iep, Octanol, and Pepinfo [13]; (12) Protein 2D structure prediction, such as Garnier, HelixturnHelix, Pepcoil, Pepwheel, and Tmap [13]; (13) protein motifs prediction, such as Antigenic, Digest, Fuzzpro, and Sigcleave [13]; (14) phylogeny analysis, such as Phylip [22]. Many of these come from the open source package EMBOSS [13].

More significant than the list of basic programs integrated is the fact that KDE Bioscience provides a mechanism—XML Integration Framework (XIF)—that enables programs to be integrated by users themselves without any programming. A standalone GUI application, XIF Studio, is provided to guide the end user in integrating command-line executable applications into the platform. The programs integrated can be executable programs or scripts written in Perl or any other shell language. With XIF, a trivial java class file and a XML file describing the user interface and command line options will be created automatically. After KDE Bioscience is rebooted the new nodes will appear in the user interface for use in workflows.

3.2.3. Remote query nodes

There are many popular bioinformatics applications and databases hosted remotely such as the queries of nucleotide sequence, protein sequence, and Medline at http://www.ncbi.nlm.nih.gov. As these databases or algorithms are difficult or impossible to install locally, KDE

Bioscience provides groups of nodes to access them instead of using a web browser. With these nodes, NCBI (http://www.ncbi.nlm.nih.gov), PDB (http://www.rcsb.org), SwissProt (http://us.expasy.org), SMART (http://smart.embl-heidelberg.de), KEGG (http://www.genome.ad.jp/kegg/), SRS integrated databases (http://www.scbit.org/srs7), and many other websites such as HUGO (http://www.gene. ucl.ac.uk/nomenclature/) can be accessed.

Instead of raw web pages, the outputs of queries will be translated into structured data automatically by KDE Bioscience. Thereafter, such data can be used in the workspace as sequence collections or tables for further processing by various nodes.

3.2.4. Alignment nodes

Alignment is the basic process in sequence analysis. In KDE Bioscience alignment programs such as Blast [23], ClustalW [24], Sim4 [25], MUMmer [26], Fastacmd [23], and Dotter [27] are integrated for this purpose.

3.2.5. Visualization nodes

Visualization nodes include several viewers for data models. These graphical representations allow biologists to get a better understanding of the data. In addition to the default text viewer for plain text, which represents the data with return value of the corresponding toString( ), there are some other graphical viewers such as FeatureVista for sequence collections, AlignmentTreeView for clustalW results, BlastViewer for Blast results, TableEditor for KResultSets, and so on. Some of them are home-made, while others are from open-source projects such as Gsviewer (http://www.lasergo.com/gsviewer.htm) for postscript files and Rasmol (http://www.umass.edu/microbio/rasmol/) for molecular structures.

3.2.6. Accessory tools

In addition to the core functions mentioned above, KDE Bioscience provides several nodes to assist the analysis—nodes for merging sequences and their annotations, nodes for extracting specific features and so on.

With data models and nodes, a variety of algorithms and bioinformatics data can be integrated to provide a powerful integrative environment for complex bioinformatics analysis.

3.3. Architecture

To support workflow construction, KDE Bioscience is built using the Java 2 Platform Enterprise Edition (J2EE) architecture. It consists of three layers: User Interface (UI) layer, Execution layer (KDE engine), and Component layer (refer to Fig. 1). The UI layer provides an interface for construction of workflows that handles the visual presentation of nodes, data, and widgets for parameter setting, plus drag-and-drop operations, and the actual execution of the workflow. The execution layer provides a mechanism for workflow execution including metadata processing, node invocation, and data transfer. The component layer includes many modules that implement the actual algorithms. The interface between the execution layer and component layer is defined by the SDK.

The UI layer provides interfaces to operate the KDE engine, including a human- and machine-friendly interface. KDE Bioscience has two kinds of user-friendly interfaces, one of which is implemented as Java GUI application, while the other is implemented as web pages. In general, the Java application GUI provides a visual interface for workflow construction and execution. After a workflow is constructed, it can be deployed as web pages for execution from a standard web browser. To support the construction of distributed applications, a machine friendly interface is also provided, where workflows can be built, modified, and executed via the Simple Object Access Protocol (SOAP) as Web-Services.

The execution layer transforms the graph of workflow in the KDE Bioscience GUI into a concrete execution plan. The layer itself describes the logical model of a workflow, and acts as a virtual machine for node processing. A framework for node invocation, data, and metadata processing has been implemented in which there are two separate aspects of workflow execution: one is for metadata, while the other is for the actual data. The execution of nodes can be divided into two phases, (1) "preparing," in which

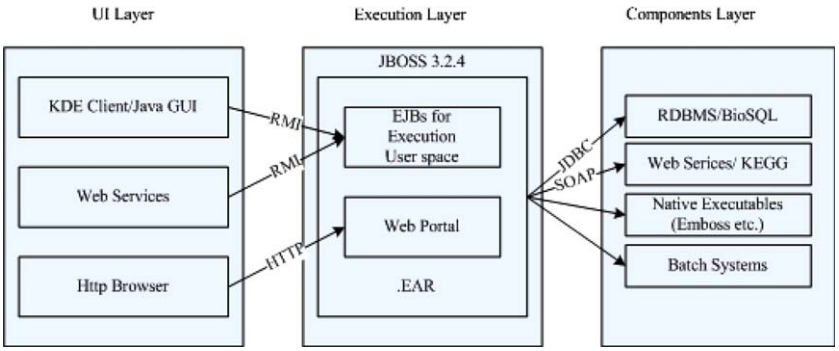

Fig. 1. KDE Bioscience Architecture. EAR denotes Enterprise Archive for server side code. RDBMS denotes Rational Database Management System. BioSQL is a part of biojava [12], and it provides an interface to DBMS. KEGG denotes an instance of Web-Services [4]. RMI denotes the method of Java Remote Method Invocation.

metadata is processed and any errors will prevent the workflow from starting execution, and (2) "processing," in which the actual data itself is processed. Here, any errors will stop the workflow executing beyond the point where the error occurred. The preparation phase provides a highly flexible mechanism for checking that workflows have been constructed correctly—for example, sequence types and table column metadata can be checked before the workflow is allowed to execute. This metadata verification significantly increases the likelihood of a successful workflow execution.

As we mentioned before, the interface to the execution framework is designed as SDK, with which algorithms can be plugged into KDE Bioscience as components. The code implementing the SDK builds the low-level component layer that behaves as the micro-code of a virtual machine, carrying out the actual computing task.

Beside these three basic layers, there are some other management modules aside, such as user management, user space (file system) management, node management and a special module for database access, that construct the platform for collaboration between users. Each user has a private user space where workflows and data are stored. Moreover, different users belonging to the same group can share a public user space. With this sharing mechanism different users can collaborate quite easily. Not only data and results can be shared, but also the same workflow can be edited and executed by different users in the same group. Therefore many users can work together on the same bioinformatics analysis.

Technically the framework of the KDE Bioscience system is implemented using J2EE. The application server adopted is JBOSS (http://www.jboss.org/), an open source project. Several Enterprise JavaBeans (EJBs, Server-side components in J2EE platform) carry out the above functions, for example, ExecutionBean for execution of workflows, ComponentsBean for management of nodes, UserBean for account management, UserspaceBean for operation of the user space, and so on. Tomcat embedded in JBOSS provides support for web-based access.

4. Results and discussion

4.1. Usage

In this section, a concise description is given to illustrate typical use of KDE Bioscience based on the Java

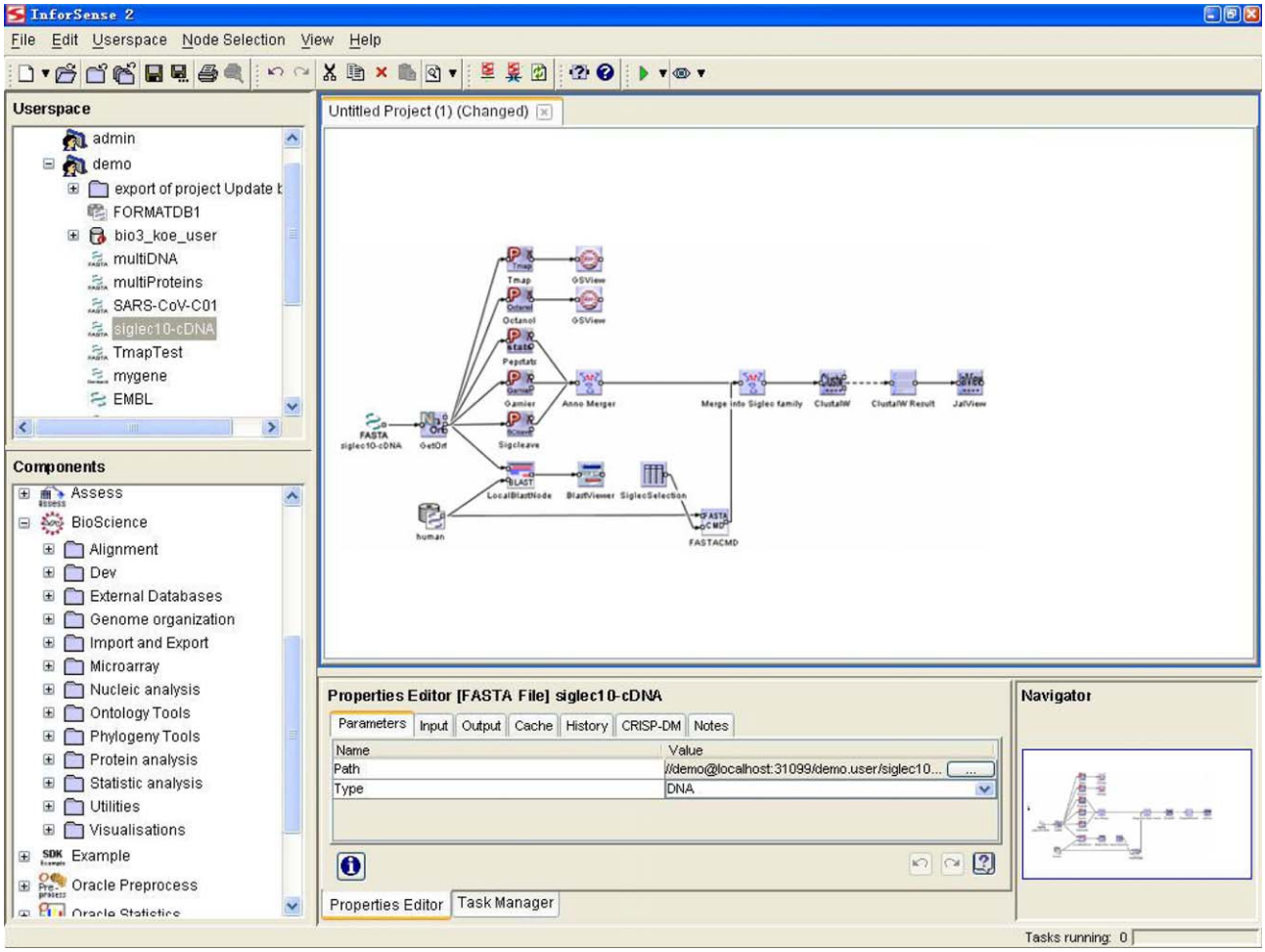

Fig. 2. KDE Bioscience Java GUI.

GUI (Fig. 2). Essentially there are 5 panels in the Java GUI. At the top-left, it is (1) User space panel, which provides the space to display data to be processed, results produced, and even workflows constructed. Different users belonging to the same group can exchange the data via copy-and-paste operations here. At the bottom-left there is (2) Component panel, where all the nodes are listed as a tree according to their groups of functions. Users can drag-and-drop the icons from user space panel (data) and components panel (algorithms) into the top-right (3) Workspace panel to construct the workflow. When a node is selected by a simple click in the workspace, its corresponding parameters can be set in the bottom-middle (4) Properties Editor panel. When a workflow is constructed and its parameters are set, the user can select one branch of the workflow and trigger execution via a toolbar icon or pop-up menu. When a workflow is very large, and (5) Navigator panel provides a global view for the entire workflow in contrast to the workspace panel that gives only the view of the part that the user is currently interested in.

4.2. Use cases

In this section, two use cases are presented to illustrate KDE's usability and function concisely.

4.2.1. SARS analysis

KDE Bioscience has been involved in the Severe Acute Respiratory Syndrome (SARS) research conducted at SCBIT from the beginning [28], and serves as the framework for further investigation [29]. Since SARS-coronavirus (SARS-CoV) was found as the causative virus, one important task in SARS research has been to examine the genomic variation between virus samples taken from different patients, and to find other homologous species. KDE Bioscience facilitated the necessary nucleotide and protein sequence analysis.

Following is a typical case for SARS research. First, we download all SARS genomes from the NCBI public database. Then, we compare the genomes downloaded (NCBI Sequence) and genomes (HP03/HP04/PC) sequenced by our collaborators using ClustalW to look for interesting variations (Fig. 3). With the variations

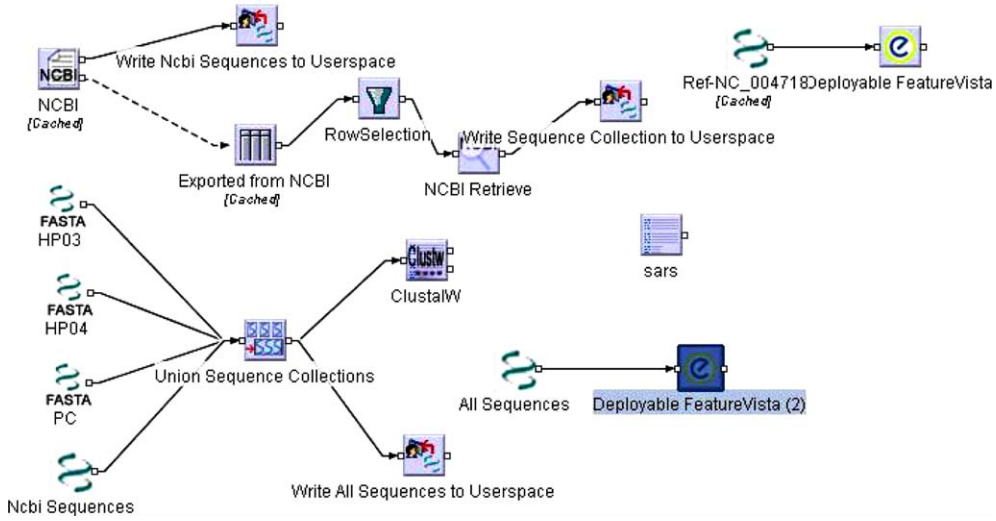

Fig. 3. Workflows for SARS genomes variation finding.

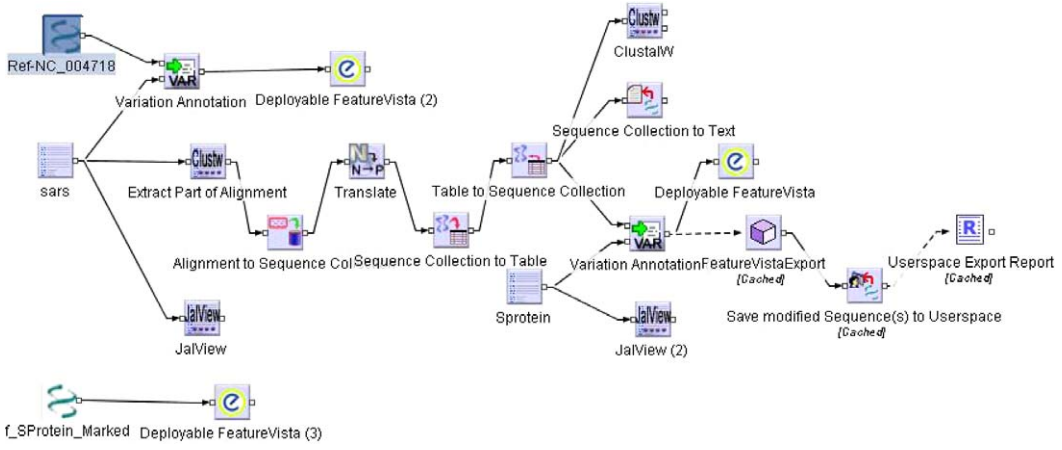

Fig. 4. Workflows for SARS S-Protein mark.

obtained, we are able to annotate the reference genome. Furthermore, we extract the varying sections and translate them into protein sequences to mark the reference protein sequence (for instance, S-Protein, Fig. 4).

Subsequently, the S-Protein marked with variation points is annotated with various protein analysis tools—for example, Tmap for transmembrane region prediction. With these annotations, we can find some interesting

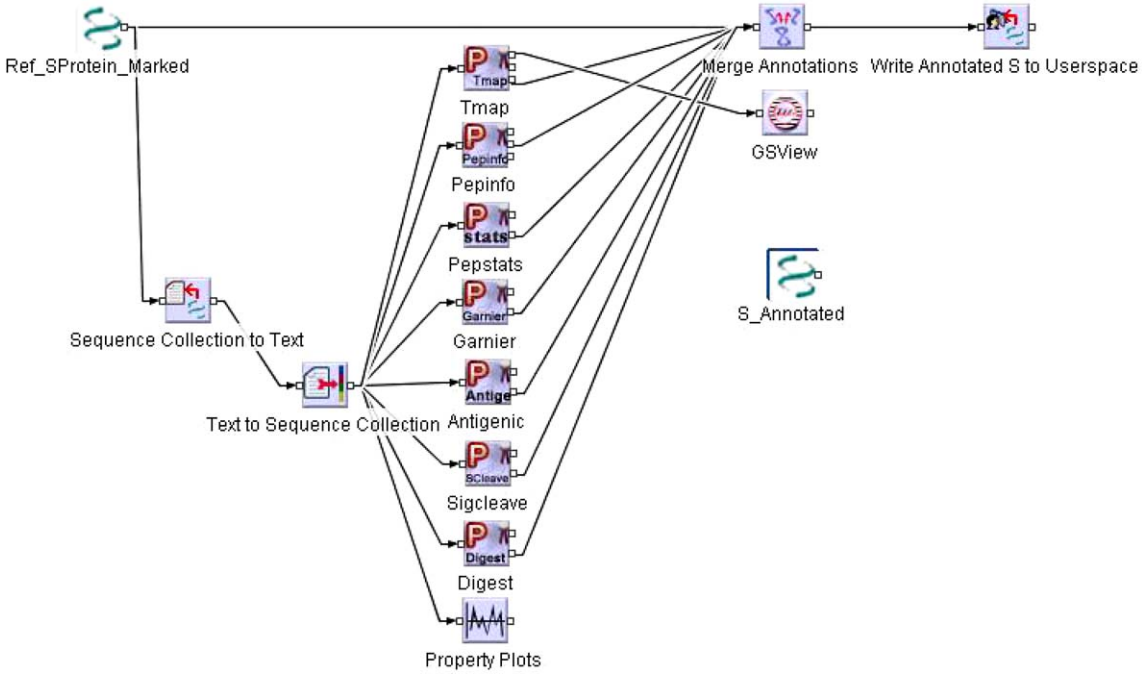

Fig. 5. Workflow for S-Protein annotation.

Fig. 6. Visualization of genome and its annotation with FeatureVista.

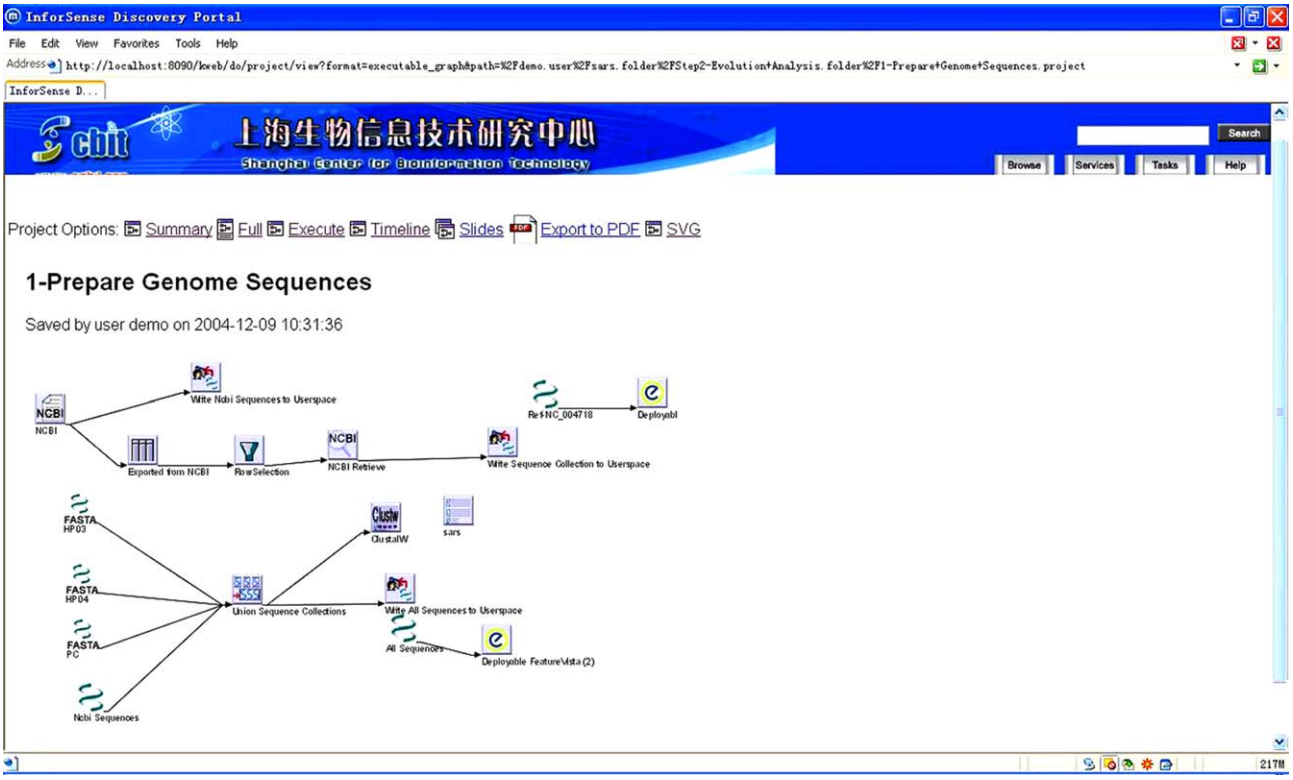

Fig. 7. Workflows deployed as web page.

Fig. 8. Visualization of genome in web pages.

properties of the variation regions (Fig. 5). At any time, the sequence can be viewed by a custom sequence viewer FeatureVista, where the annotations and features are listed and visualized (Fig. 6).

In addition to the standard analysis process, the workflows constructed can be deployed as web pages, and thus execute using a browser (Fig. 7). The results are visualized in an applet (Fig. 8).

4.2.2. SMIGA

The System for Microbial Genome Annotation (SMIGA) is a web server (http://www.scbit.org/smiga/index.html) provided by SCBIT for prokaryotic genome annotation, which is built using KDE Bioscience.

SMIGA users can log into the system to submit DNA or protein sequences for analysis. Thereafter, the system will take the user to the corresponding selection page, which lists

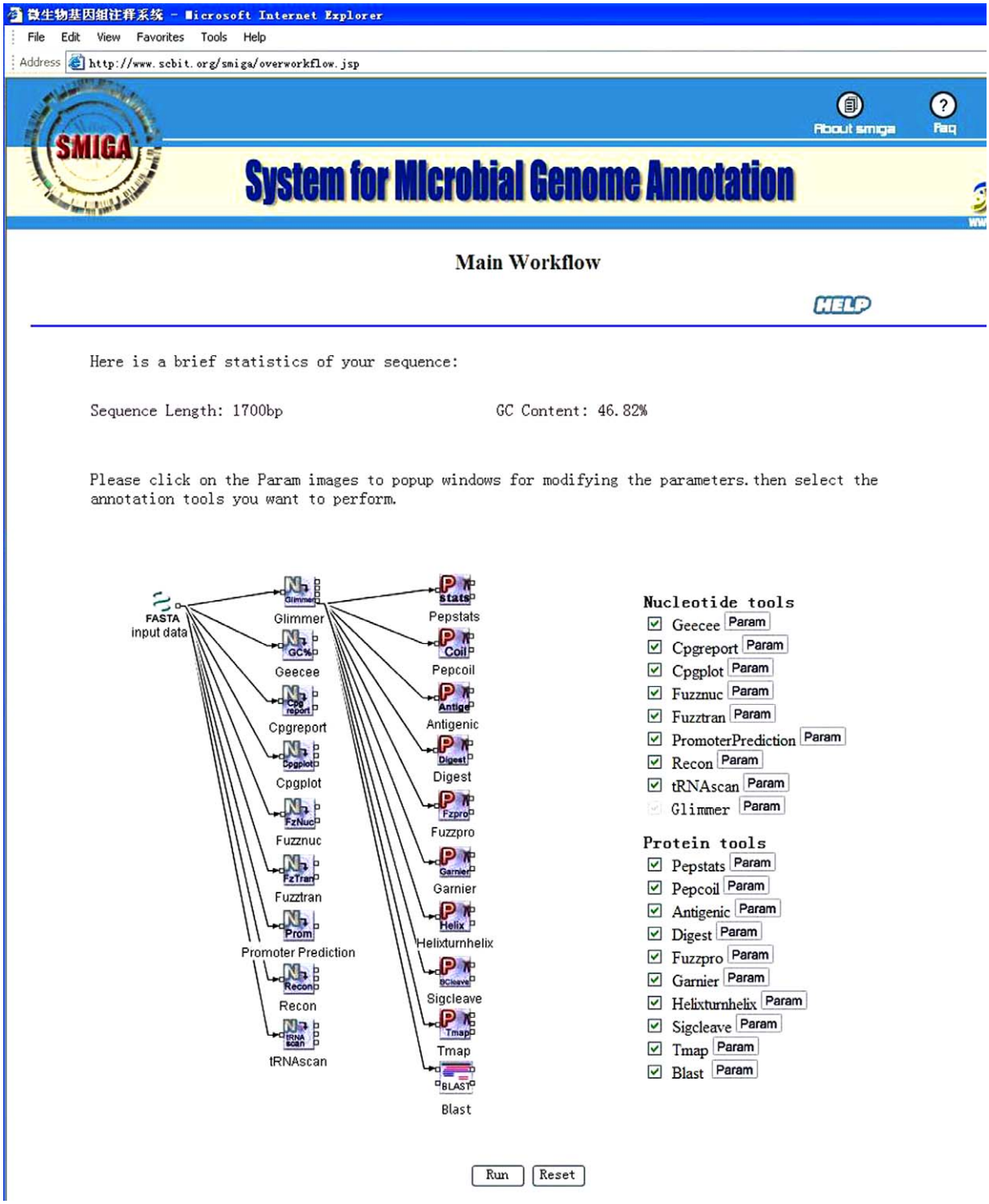

Fig. 9. Web page of SMIGA, an application instance of KDE Bioscience.

all possible functions related to a given type of sequence. For example, in Fig. 9, after a DNA sequence is submitted, a web page including functions such as tRNAscan, Glimmer, Garnier, Tmap, Antigenic, etc. is presented. Users can choose some or all functions according to their needs. They can also set or adjust the parameter settings for the selected nodes by clicking on their corresponding "param" buttons. Clicking "Submit" will trigger the system and automatically send a notification email when the job is done. Henceforth, users can come back to the system and view the results for all of the jobs submitted and finished.

SMIGA is a typical application instance of KDE Bioscience with a customized web UI. All annotation tasks and algorithms are managed and executed in the background. The algorithms are hosted in a distributed computing environment using KDE Bioscience infrastructure, which brings powerful computing capabilities to SMIGA.

5. Discussion

As we have illustrated, in workflows, algorithms are implemented as nodes. Roughly, these algorithms can cover any aspects of bioinformatics analysis regardless of their usage, purpose, programming languages, operating systems, and input–output data formats. With appropriate data models, multiple nodes can be linked together to form powerful workflows. Furthermore, the uniform interface of KDE Bioscience makes a variety of algorithms transparent to the biologist. In this way, biologists can avoid tiresome tasks caused by complicated software and concentrate on biological problems. This simple drag-and-drop operation offers a real opportunity to improve the efficiency of bioinformatics research.

With the well-structured KDE Bioscience SDK, the algorithms hosted in a distributed computing environment can be incorporated as nodes regardless of the invocation protocols. Also, the workflows constructed can be exported as Web-Services. In this sense, KDE Bioscience acts not only the portal to access a bioinformatics grid but also a grid computing service provider. It provides a simple solution to the integration of distributed computing resources. To support collaborative work, user and user space management provide facilities for multiple users to share data and workflows. As a result, several biologists can work concurrently on the same bioinformatics project without data collision. Moreover, the use of J2EE allows flexibility of architecture and portability of applications. Several UIs—Java GUI, web page, and SOAP application program interface—are presented to fit for various users' requirements. In addition, the robustness of KDE Bioscience benefits a lot from the robustness of Java language itself and J2EE.

It is plain that workflow provides a workable mechanism to integrate data and algorithms. KDE Bioscience, which adopts workflow and J2EE, provides an integrative platform for biologists to collaborate and use distributed computing resources in a simple manner.

Finally, it should be pointed out that, although our software brings a lot of advantages, the overhead caused cannot be ignored. The data transferring between different nodes cost much computing resource. While a large-scale date set is processed, the performance decline is noticeable, sometimes even intolerable. In the current version of KDE Bioscience, a practical solution is to transfer data address such as file location instead of data itself. However, this solution may limit its portability in some distributed computing environments.

6. Conclusions

In summary, we demonstrated in this paper an integrative platform, KDE Bioscience that provides a bioinformatics framework to integrate data, algorithms, computing resources, and human intelligence. Significantly, it allows biologists to simplify the usage of complicated bioinformatics software to concentrate more on biological questions. In fact, the power of KDE Bioscience comes from not only the flexible workflow mechanism but also more than 60 included programs. With workflows, not only the analysis of nucleotide and protein sequences but also other novel calculations and simulations can be integrated. In this sense, KDE Bioscience makes an ideal integrated informatics environment for bioinformatics or future systems biology research.

Acknowledgments

The research is funded by Hi-Tech research and development program of China, Grant No. 2003AA231011. The authors thank Alex Michie for his help and expertise.

References

[1] Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL. GenBank: update. Nucleic Acids Res 2004;32:23–6.
[2] Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, et al. UniProt: the universal protein knowledgebase. Nucleic Acids Res 2004;32:115–9.
[3] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The protein data bank. Nucleic Acids Res 2000;28:235–42.
[4] Kanehisa M, Goto S. KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 2000;28:27–30.
[5] Rice P, Longden I, Bleasby A. EMBOSS: the european molecular biology open software suite. Trends Genet 2000;16:276–7.
[6] Felsenstein J. PHYLIP: phylogeny inference package (Version 3.2). Cladistics 1989;5:164–6.
[7] Giannadakis N, Rowe A, Ghanem M, Guo Y. InfoGrid: providing integration for knowledge discovery. Inform Sci—Inform Comput Sci: An Int J 2003;155:199–226.
[8] Hoon S, Ratnapu K, Chia J, Kumarasamy B, Juguang X, Clamp M, et al. Biopipe: a flexible framework for protocol-based bioinformatics analysis. Genome Res 2003;13:1904–15.
[9] Leo P, Marinelli C, Pappadà G, Scioscia G, Zanchetta L. BioWBI: an Integrated Tool for building and executing Bioinformatic Analysis Workflows, Bioinformatics Italian Society Meeting (BITS 2004), Padova; 2004.
[10] Oinn T, Addis M, Ferris J, Marvin D, Greenwood M, Carver T, et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 2004;20:3045–54.

[11] Tang F, Chua C, Ho L, Lim Y, Issac P, Krishnan A. Wildfire: distributed, Grid-enabled workflow construction and execution. BMC Bioinformat 2005;6:69.
[12] Pecock M, Down T, Hubbard T. Biojava: open source components for bioinformatics. ACM SIGBIO Newlett 2000;20:10–2.
[13] Rice P, Longden I, Bleasby A. EMBOSS: the European molecular biology open software suite. Trends Genet 2000;16:276–7.
[14] Hofacker IL, Fontana W, Stadler PF, Bonhoeffer S, Tacker M, Schuster P. Fast folding and comparison of RNA secondary structures. Monatshefte f Chem 1994;125:167–88.
[15] Rozen S, Skaletsky H. Primer3 on the WWW for general users and for biologist programmers. In: Krawetz S, Misener S, editors. Bioinformatics methods and protocols in the series methods in molecular biology. Totowa, NJ: Humana Press; 2000, p. 365–86.
[16] Reese MG. Application of a time-delay neural network to promoter annotation in the *Drosophila melanogaster* genome. Comput Chem 2001;26:51–6.
[17] Bao Z, Eddy S. Automated de novo identification of repeat sequence families in sequenced genomes. Genome Res 2002;12(8):1269–76.
[18] Bedell JA, Korf I, Gish W. MaskerAid: a performance enhancement to RepeatMasker. Bioinformatics 2000;16:1040–1.
[19] Lowe TM, Eddy SR. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res 1997;25(5):955–64.
[20] Tiwari S, Ramachandran S, Bhattacharya A, Bhattacharya S, Ramaswamy R. Prediction of probable genes by Fourier analysis of genomic sequences. Comput Appl Biosci 1997;13(3):263–70.
[21] Salzberg SL, Pertea M, Delcher AL, Gardner MJ, Tettelin H. Interpolated Markov models for eukaryotic gene finding. Genomics 1999;59:24–31.
[22] Felsenstein J. PHYLIP—phylogeny inference package (Version 3.2). Cladistics 1989;5:164–6.
[23] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215:403–10.
[24] Wilbur WJ, Lipman DJ. Rapid similarity searches of nucleic acid and protein data banks. Proc Natl Acad Sci USA 1983;80:726–30.
[25] Florea L, Hartzell G, Zhang Z, Rubin GM, Miller WA. Computer program for aligning a cDNA sequence with a genomic DNA sequence. Genome Res 1998;8:967–74.
[26] Delcher AL, Kasif S, Fleischmann RD, Peterson J, White O, Salzberg SL. Align of whole genomes. Nucleic Acids Res 1999;27:2369–76.
[27] Sonnhammer ELL, Durbin R. A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis. Gene 1995;167:GC1–GC10.
[28] The Chinese SARS Molecular Epidemiology Consortium. Molecular evolution of the SARS coronavirus during the course of the SARS epidemic in China. Science 2004;303:1666–9.
[29] Curcin V, Ghanem M, Guo Y, He W, Li Y, Hao P. SARS analysis on the grid. UK e-science all hands meeting, Nottingham, UK; 2004.