BMC Bioinformatics

Research article

AnaBench: a Web/CORBA-based workbench for biomolecular sequence analysis

Elarbi Badidi*, Cristina De Sousa, B Franz Lang and Gertraud Burger

Address: The Canadian Institute for Advanced Research, Program in Evolutionary Biology. Département de Biochimie, Université de Montréal, 2900, Boul. Édouard Montpetit, Montréal, QC H3T 1J4, Canada

Email: Elarbi Badidi* - me.badidi@umontreal.ca; Cristina De Sousa - cristina_de_sousa@hotmail.com; B Franz Lang - franz.lang@umontreal.ca; Gertraud Burger - gertraud.burger@umontreal.ca

* Corresponding author

Published: 16 December 2003

BMC Bioinformatics 2003, **4**:63

Received: 24 July 2003

Accepted: 16 December 2003

This article is available from: http://www.biomedcentral.com/1471-2105/4/63

Abstract

Background: Sequence data analyses such as gene identification, structure modeling or phylogenetic tree inference involve a variety of bioinformatics software tools. Due to the heterogeneity of bioinformatics tools in usage and data requirements, scientists spend much effort on technical issues including data format, storage and management of input and output, and memorization of numerous parameters and multi-step analysis procedures.

Results: In this paper, we present the design and implementation of AnaBench, an interactive, Web-based bioinformatics **Ana**lysis work**Bench** allowing streamlined data analysis. Our philosophy was to minimize the technical effort not only for the scientist who uses this environment to analyze data, but also for the administrator who manages and maintains the workbench. With new bioinformatics tools published daily, AnaBench permits easy incorporation of additional tools. This flexibility is achieved by employing a three-tier distributed architecture and recent technologies including CORBA middleware, Java, JDBC, and JSP. A CORBA server permits transparent access to a workbench management database, which stores information about the users, their data, as well as the description of all bioinformatics applications that can be launched from the workbench.

Conclusion: AnaBench is an efficient and intuitive interactive bioinformatics environment, which offers scientists application-driven, data-driven and protocol-driven analysis approaches. The prototype of AnaBench, managed by a team at the Université de Montréal, is accessible on-line at: http://malawimonas.bcm.umontreal.ca:8091/anabench. Please contact the authors for details about setting up a local-network AnaBench site elsewhere.

Background

Introduction

To conduct sequence analysis, biologists use a variety of bioinformatics software in a sequential fashion. For example, phylogenetic analysis of newly sequenced protein-coding genes involves translation of the nucleotide sequence to protein sequence in six frames (e.g., using FLIP [1]), identification of ORFs that correspond to conserved proteins by similarity search (e.g., using FASTA [2]), retrieval of protein sequences from GenBank using ENTREZ [3], multiple protein alignment (e.g., using CLUSTALW [4]), extraction of well aligned sequence stretches, as well as tree inferences and testing (e.g., using PHYLIP [5]). A major bottleneck is that most software applications are incompatible with one another because they use different file formats. As a consequence, the output of one tool cannot be used as an input for another, without data format conversion. A further complication is that the user has to define a multitude of parameters and options according to the particular data or aim of the analysis. Finally, as bioinformatics tools are written in various programming languages and for different operating systems, installation, configuration, and maintenance of these numerous software components is time consuming, costly and requires IT expertise.

Current methods for bioinformatics software integration

In the past few years, several interactive environments have been developed to facilitate bioinformatics analyses. Some of these environments are commercial products (e.g., BIONAVIGATOR [6], ISYS [7,8] and TURBOBENCH [9]); others are freely accessible (e.g., NCSA biology workbench [10] and GWFASTA [11]), or open-source (e.g., EMBOSS [12], ERATO SBW [13,14], and APPLAB [15]). These environments may be classified into three main categories.

Web-based environments

Bioinformatics tools are accessible through Web interfaces using HTML and various scripting languages (CGI, Perl, etc.). BIONAVIGATOR, NCSA biology workbench, and GWFASTA are examples for such environments. The advantage of these Web-based workbenches is that the user does not need to install bioinformatics tools on his/her computer, but only requires a Web browser to launch applications on the centralized server. However, the drawback of Web-based environments is that interactive bioinformatics tools cannot be easily integrated.

Client-side environments

These environments do not have the above limitation, because the environment and bioinformatics software tools are installed and executed on the user's computer. The integration of tools is typically achieved by means of wrappers, and the interoperation between tools by a specific application programming interface (API) for the exchange of messages. ERATO SBW, ISYS, TURBOBENCH, EMBOSS, and APPLAB are examples for such environments. The drawback here is that the user requires IT expertise for the installation and configuration of the environment and tools.

Client-side/Web-based environments

These environments combine several advantages of the former two categories and resolve their major drawbacks. In this category, the environment and some bioinformatics applications are installed on the user's computer, while providing a gateway to a Web-based environment. BIONODE-BIONAVIGATOR is an example for such a system.

Results

Objectives

None of the current open-source bioinformatics analysis environments provides both accessibility from any platform and location and easy integration of biological databases and tools developed by us and others. Therefore, we set out to develop 'AnaBench' a new Web-based environment that combines these two features.

Our goal was to build a Web-based workbench bioinformatics infrastructure, which fulfills the following requirements:

For the user (biologist):

1) Access to the workbench by employing only a Web browser without the need to download and configure software on his/her computer;

2) Access to individual user workspace on our central servers to save biological data and analysis results, and organization of this workspace in terms of projects;

3) Execution of bioinformatics tools offered in the workbench, with input data selected from the user's projects, and saving of results into selected projects.

For the workbench administrator:

1) Straightforward description of the bioinformatics tools in terms of parameters, data types, and data formats they can handle;

2) Easy integration of in-house and public biological databases;

3) Launching of applications hosted on our servers as well as remote applications available from third parties.

Workbench Architecture

To meet the above requirements, we chose a three-tier architecture for the workbench, an architecture that has gained popularity in the software industry. However, its use in the bioinformatics field is very recent. It usually consists of a thin client providing presentation logic, a middle-tier containing the business logic, and a back-end database. Compared to conventional client-server applications, three-tier applications provide several advantages including: (i) *Scalability*, by distributing application components across multiple servers; (ii) *Flexibility* to changes, by separating the business logic from its presentation logic; (iii) *Reusability* of components, by dividing the application into multiple layers; and (iv) *Reliability*, by implementing multiple levels of redundancy. Figure 1

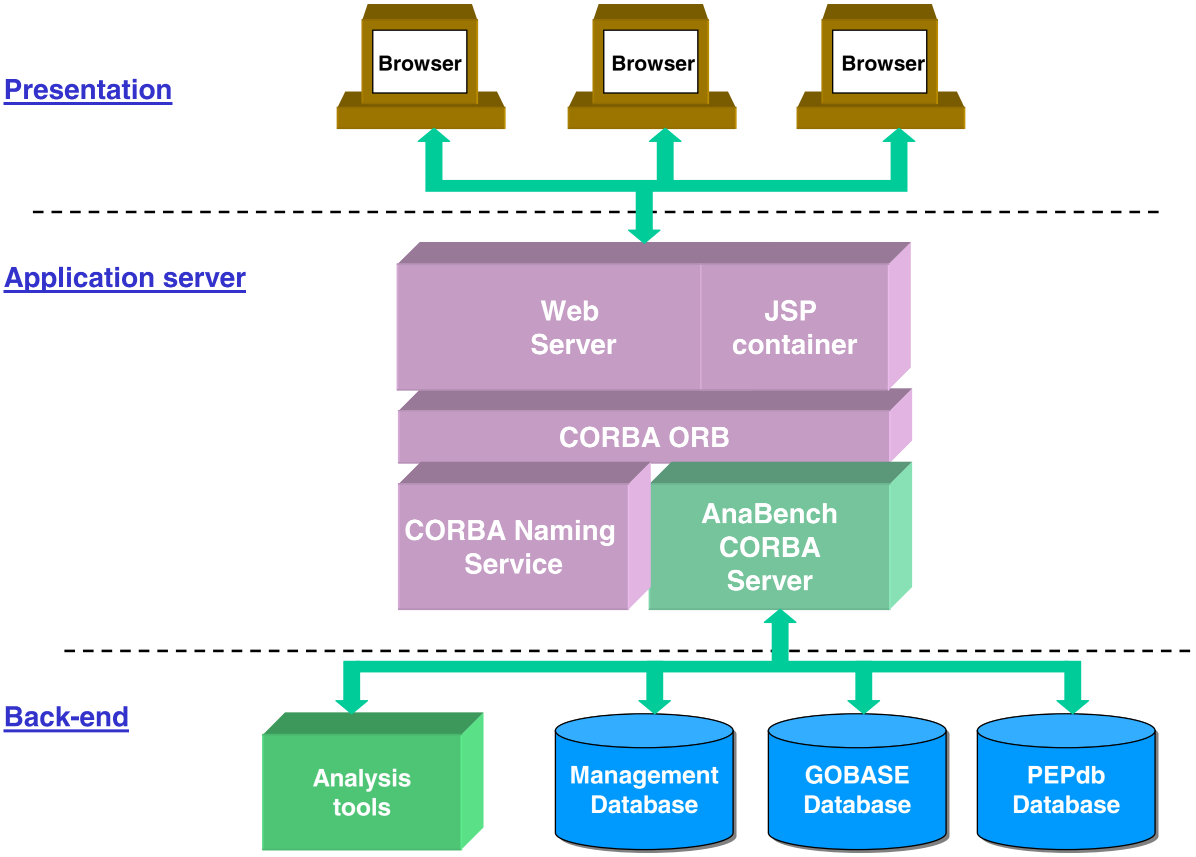


Figure 1
AnaBench architecture. JSP, Java Server pages; ORB, Object Request Broker, the CORBA middleware; GOBASE, Organelle Genome Database [18,19]; PEPdb, Protist EST Program database [20].

shows the main components of the workbench architecture.

Presentation level

Users interact with the workbench through a Web browser using HTML and JSP (Java Server Pages) screen forms. JSP [16] is a technology for controlling the content and appearance of Web pages through the use of servlets, i.e, small programs that run on the Web server to dynamically generate the requested Web page before sending it to the user.

Application server level

The requests submitted by users at the presentation level are handled by a Web server equipped with a servlet and JSP container, and CORBA (Common Object Request Broker Architecture) middleware (Figure 1).

CORBA is the standard distributed object architecture [17] that is characterized by an open software bus called Object Request Broker (ORB) through which heterogeneous object components can interoperate across networks. The interfaces to CORBA objects are specified by the Interface Definition Language (IDL). CORBA objects differ from typical programming language objects in three ways: (1) they may reside anywhere on a network; (2) they are able to interoperate with objects written on other platforms; and (3) they may be written in any programming language for which there is a mapping from IDL to that particular language.

The CORBA server is responsible for launching the analysis tools and interfacing with the back-end databases. The naming service is a standard CORBA object service, which provides the mapping between object names and object references.

Back-end level

In this tier, we find the management database, the biological databases and an analysis tools server, which contains bioinformatics tools. The management database stores information about users, their projects and data, and descriptions of bioinformatics tools as outlined above. Workbench users will be able to save the results of database queries and data analysis in their workspace for further processing.

The biological databases that are currently accessible through the workbench include: GOBASE, a taxonomically broad organelle genome database that organizes and integrates diverse data related to organelles [18,19], the Protist EST database (PEPdb), both developed in-house [20], and GenBank [21] via remote BLAST [22].

Workbench Design

In this section, we describe the design and functionalities of AnaBench, as well as sustainability issues.

Use-cases

Figure 2 presents the use-cases that have been identified in the course of the requirement analysis of the workbench.

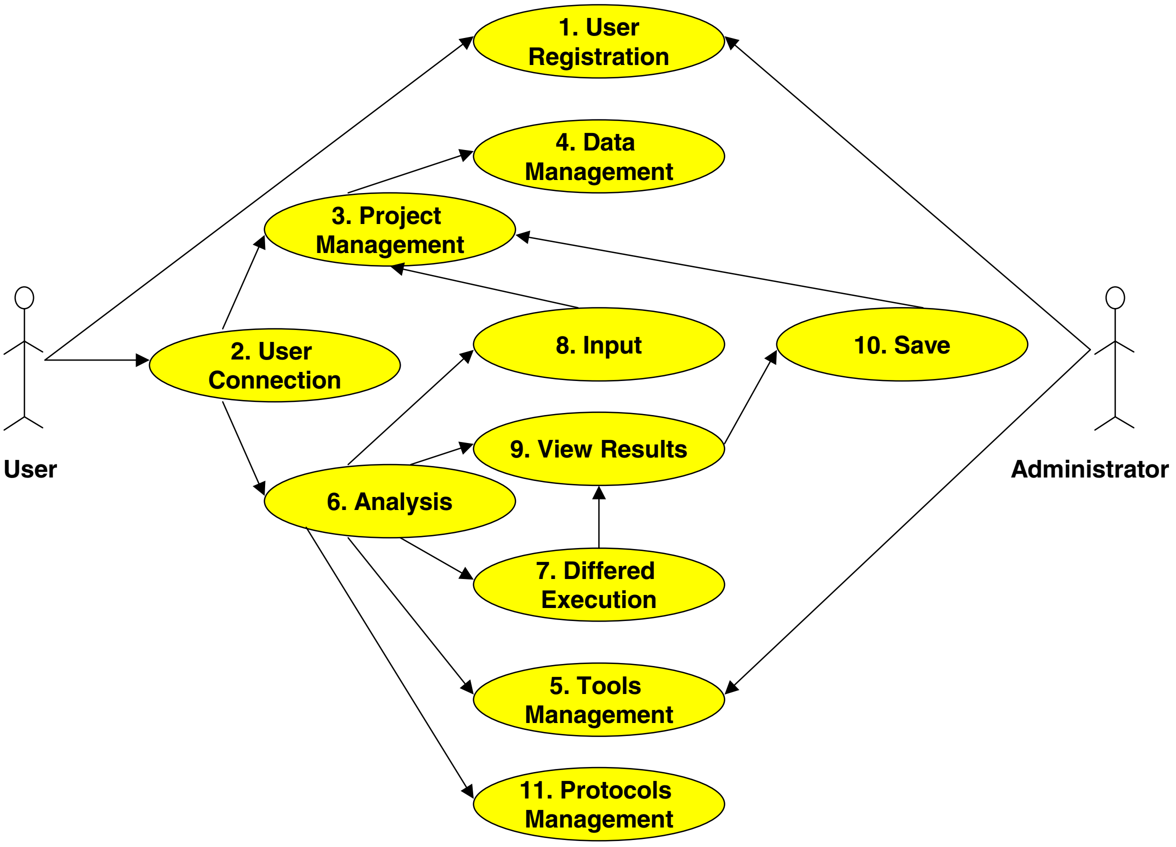


Figure 2
AnaBench use-cases. (1) User Registration: provide name, login name, email, and password. (2) User Connection: provide login name and password. (3) Project Management: create new projects, edit or delete an existing project, and list all user's projects. (4) Data Management: add, edit or delete data, and list all data of a project. (5) Tools Management: create, edit, delete, and list analysis categories, applications, and parameters. (6) Analysis: launch analysis applications, display results, and save them in a user's project. (7) Deferred Execution: launch analysis application in deferred mode. (8) Input: select input data for analysis applications from the user's projects. (9) View Results: display analysis results on the user's browser. (10) Save: save analysis results in an existing or in a new project. (11) Protocol Management: create new protocols, edit or delete an existing protocol, and list all user's protocols.

User Registration (UC1): The main page gives access to the registration of users, who have to provide their name, email, login name and password.

User Connection (UC2): In order to access individual workspace projects and the analysis tools, the user connects to the main Web page and identifies him/herself by entering login name and password. The system validates these data, rejects the connection if the identification is incorrect, and provides access to the 'User Registration' use-case (UC1), if the user is not yet registered.

Project Management (UC3): Users manage their own workspace projects, i.e., create a project, list all projects, edit or delete selected projects. The users have only access to this use-case after they have been identified by the system through 'User Connection' (UC2).

Data Management (UC4): This use-case is generated by 'Project Management' (UC3). It allows users to manage their sequence data within their projects, i.e., add data, list all data of a project, edit or delete selected data. Data can be added in a project using copy and paste, by uploading local files from the user's computer, and by saving results of analyses and external database queries. The users have only access to this use-case if they have selected one of their projects through UC3.

Tools Management (UC5): The workbench administrator describes with this use-case the parameters of bioinformatics applications to be integrated in the workbench.

Analysis (UC6): The user selects an analysis tool and provides the appropriate input data, an action that generates the 'Input' use-case (UC8). The results produced by a given tool may be saved in an existing or a new workspace project. Users have only access to this use-case if they have been identified by the system through 'User Connection' (UC2).

Deferred Execution (UC7): Users launch the execution of an application in the deferred mode, which means that they do not have to wait for the application to terminate before carrying out other tasks within the workbench. This mode is suited for tools that require intensive computation, as well as for remote applications managed by third parties. The system will inform the user by email once a deferred execution is completed.

Input (UC8): The user specifies the input data for a given analysis tool by selecting the data from the user's projects. This use-case is generated by 'Analysis' (UC6).

View Results (UC9): The results generated by an analysis tool are displayed on the user's Web browser. This use-case follows 'Analysis' (UC6) or 'Deferred Execution' (UC7).

Save (UC10): The user saves the results of an analysis in a selected workspace project, a use-case generated by 'View Results' (UC9).

Protocol Management (UC11): Protocols are fully interactive analysis pathways for addressing specific biological questions. Users manage their own protocols, i.e., create new protocols using the analysis tools provided by AnaBench, list all protocols, edit or delete selected protocols. The users have only access to this use-case after they have been identified by the system through 'User Connection' (UC2).

Most of these use-cases are divided into sub-use cases. For example, 'Project Management' (UC3) consists of 'Add a project', 'Modify a project', 'Delete a project', and 'List all projects of a given user'. In the following subsection, we describe in more details the 'Tools Management' use-case (UC5), which handles the installation of new bioinformatics software in AnaBench.

The Tools Management use-case

This use-case allows to save in the management database a description of the bioinformatics tools, including parameter descriptions, data types, and data formats. These descriptions are consulted for the design of screen forms that allow input of parameters values. Only the workbench administrator uses this use-case; it is invisible to the biologist. To facilitate the management of the tools, we classified them into several categories based on their functionality, for example:

- Nucleotide sequence analysis (length, composition, molecular weight)

- Nucleotide sequence comparison

- Nucleotide sequence alignment

- Search for promoters, transcription terminators and other motifs

- Gene prediction

- RNA secondary structure prediction

- Nucleotide to protein translation

- Protein sequence analysis (composition, molecular weight, isoelectric point, length)

- Protein sequence comparison

Table 1: Tools description entities

ENTITY	ATTRIBUTES
Analysis_Category	(category_id, category_name, category_description)
Application	(application_id, application_name, category_id, application_description, application_versionnumber)
Parameter	(parameter_id, parameter_name, application_id, parameter_label, parameter_description, parameter_prefix, parameter_abbreviation, parameter_valuetype, parameter_required, parameter_requiredvalues, parameter_syntax, parameter_min, parameter_max, parameter_default)
Application_datatype	(application_id, datatype_id)
Application_dataformat	(application_id, dataformat_id)

- Protein sequence alignment

- Search for protein domains, motifs, and signatures

- Protein secondary structure prediction

- Phylogenetic inference

There are significant differences between bioinformatics tools with respect to their parameters (number of input data and format) and parameter syntax. The implementation of the analysis part of the workbench requires for each application a precise description of all parameters. Keeping this information in the management database simplifies considerably the implementation of the 'Tools Management' use-case. Three major entities describe the parameters of the tools: *Analysis_Category*, *Application*, and *Parameter*, while two minor entities *Application_datatype* and *Application_dataformat* specify the data types and data formats handled by the tools. Table 1 describes the attributes of each entity.

CORBA Server

From the analysis of the workbench use-cases, we draw the core entities of the system: *User*, *Project*, *Data* (biological data and analysis results), *DataType*, *DataFormat*, *Analysis_Category*, *Application*, *Parameter*, *Application_datatype*, *Application_dataformat*, *Execution*, *Input*, *Output*, and *Protocol*. These entities constitute the management database tables.

The CORBA server provides the connection between the Web server and the workbench backend, including sequence analysis tools and databases. The IDL specification of the server includes the following interfaces: *UserManager*, *ProjectManager*, *DataManager*, *CategoryManager*, *ApplicationManager*, *ParameterManager*, *DeferredResultManager*, and *ProtocolManager*. Figure 3 shows the objects of the CORBA server in AnaBench and their interactions with the backend. JSP files act as CORBA clients that invoke methods provided by the CORBA server. The complete IDL specification of AnaBench is available upon request.

Implementation

The prototype of AnaBench uses the following platform:

- Hardware: Two double-processor Linux servers,

- Programming language: Java,

- CORBA environment: ORBACUS 4.0 for Java,

- JSP engine: Jakarta Tomcat 3.2,

- Web server: Apache advanced extranet server 1.3.23,

- Relational Database Management System: MYSQL with JDBC driver.

At the time of writing, the following programs had been integrated in AnaBench: CLUSTALW [4] for multiple alignments; FLIP [1] for translation of nucleotide sequence; Remote NCBI BLAST [22], BLASTPEP for local Blast searches in PEPdb [20], READSEQ [12] for format conversion of sequences, RNAmot [23] for finding secondary structures in nucleic acid sequences, TANDEMREPEATS [24] for identifying tandem repeats in DNA sequences, and TRNASCAN_SE [25] for finding tRNA genes. From the descriptions of the above tools, we generate, by using a JSP Generator program, the JSP screen forms, in which the user enters the parameter values of the tool, or, more precisely, modifies the default values we have set. As the output of most analysis tools is either in plain/text format or in HTML, it is easily displayed on the main frame of the workbench, without further processing.

An example for tools description

We have incorporated into AnaBench a 'Tools Description' module that streamlines addition or upgrades of analysis tools. The workbench administrator performs the description of tools by means of JSP pages that allow to specify application categories, applications, and their

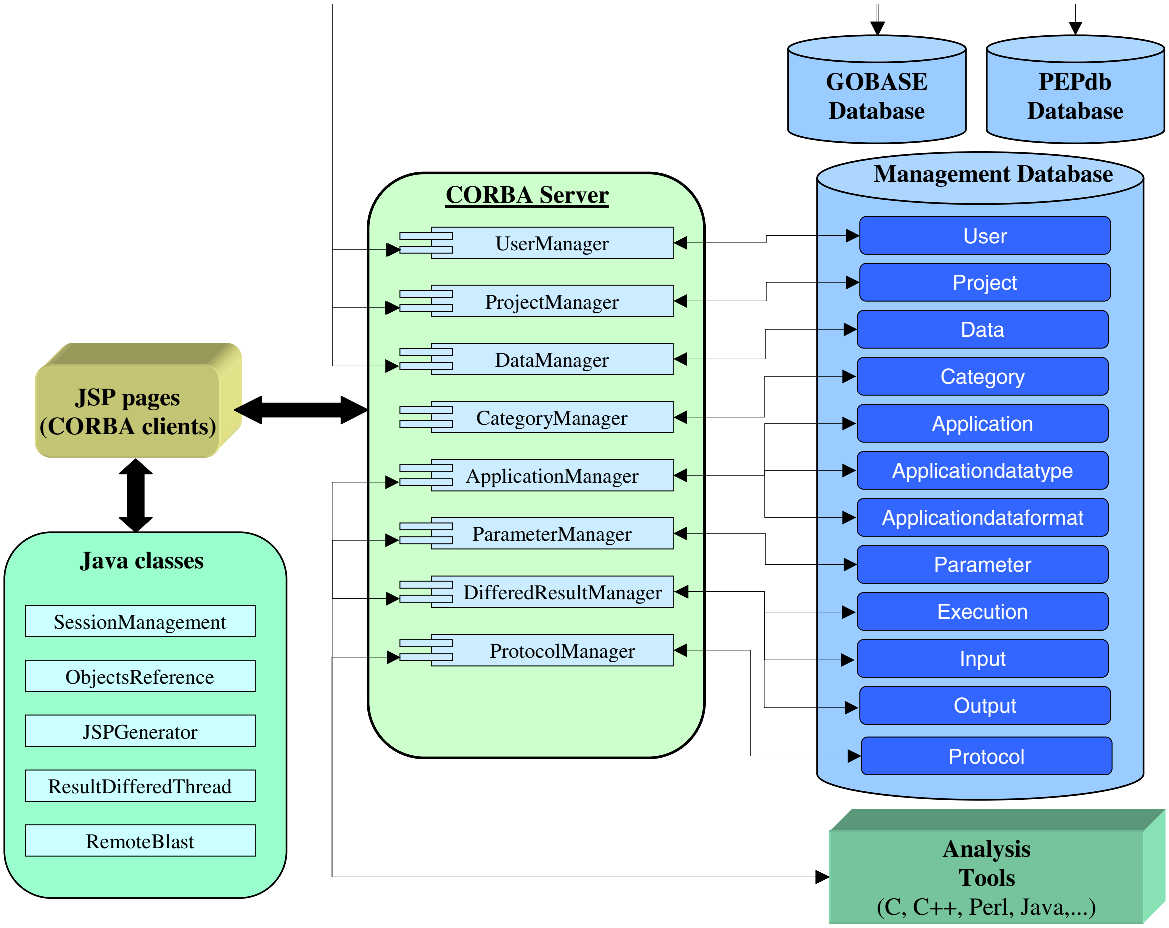


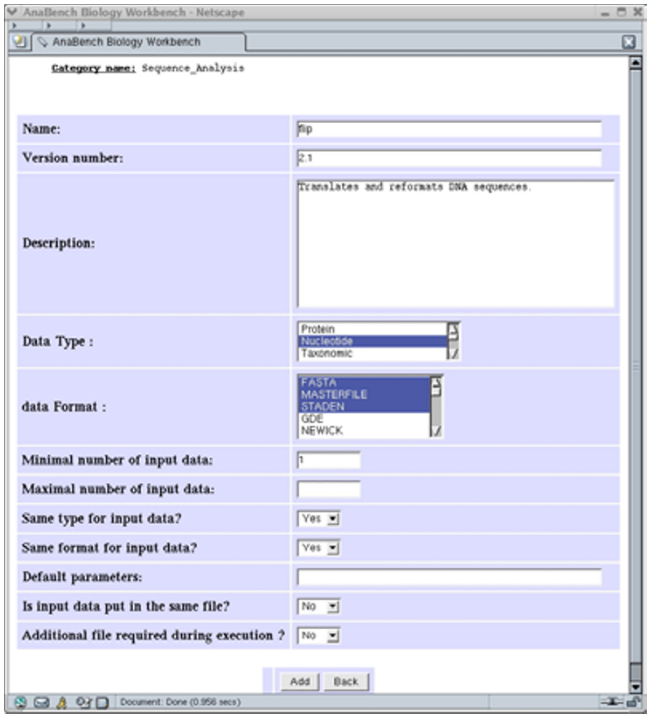
Figure 3
AnaBench CORBA Server objects. Communication between CORBA server objects and biological databases takes place through the JDBC (Java Database Connectivity) API. Communication between JSP pages and CORBA objects is performed through the IIOP (Internet Inter-ORB Protocol) protocol. Java classes are utilities used by JSP pages and include: *SessionManagement* manages session information; *ObjectsReference* finds object reference of CORBA objects; *JSPGenerator* generates JSP screen form from tools description; *ResulDeferredThread* launches a new thread for a deferred execution of a tool; *RemoteBlast* sends a BLAST request to the NCBI BLAST server.

parameters. These descriptions are saved in the management database. In the following, we present, as an example, the addition of *FLIP* into AnaBench. The steps are as follows:

First, we create a category of applications called "Sequence Analysis" provided it does not already exist in the management database. This is performed through the "addCategory.jsp" page, which interacts with the *CategoryManager* interface of the CORBA server in order to add a new record to the category table.

Second, we select from the list of categories displayed by "listCategories.jsp" the category "Sequence Analysis" created in the first step, and then we select the '*add application*' button to provide information about the FLIP application (see figure 4). Herewith a new record has been created in the application table of the management database through the *ApplicationManager* interface of the CORBA server.

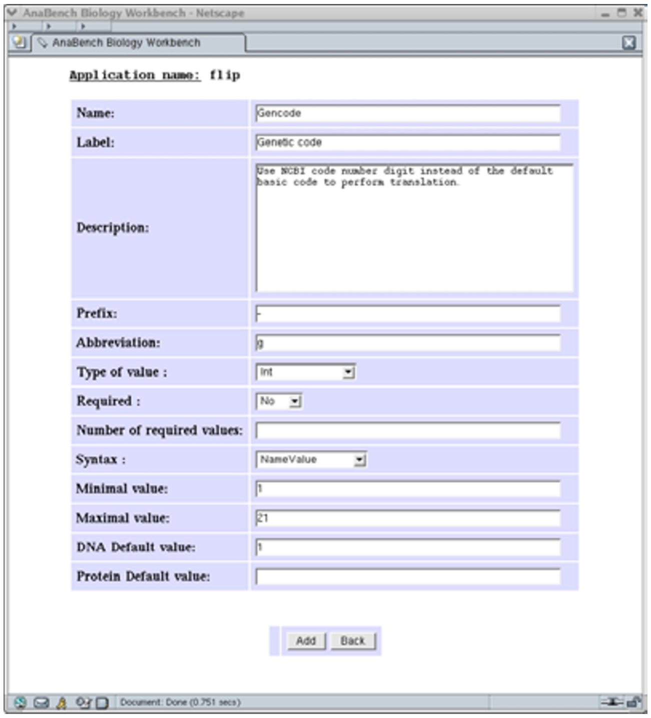
Third, from the list of applications provided by "editApplication.jsp", we select the '*add parameter*' button to fill in the description of the parameters of FLIP. For each parameter, a new record is added into the parameter table through the *ParameterManager* interface of the CORBA server (see figure 5).


AnaBench Biology Workbench - Netscape
AnaBench Biology Workbench
Add
Back
Document: Done (0.956 secs)

Category name: Sequence_Analysis

Name:	flip
Version number:	2.1
Description:	Translates and reformats DNA sequences.
Data Type :	Protein Nucleotide Taxonomic
data Format :	FASTA MASTERFILE STADEN GDE NEWICK
Minimal number of input data:	1
Maximal number of input data:	
Same type for input data?	Yes
Same format for input data?	Yes
Default parameters:	
Is input data put in the same file?	No
Additional file required during execution ?	No

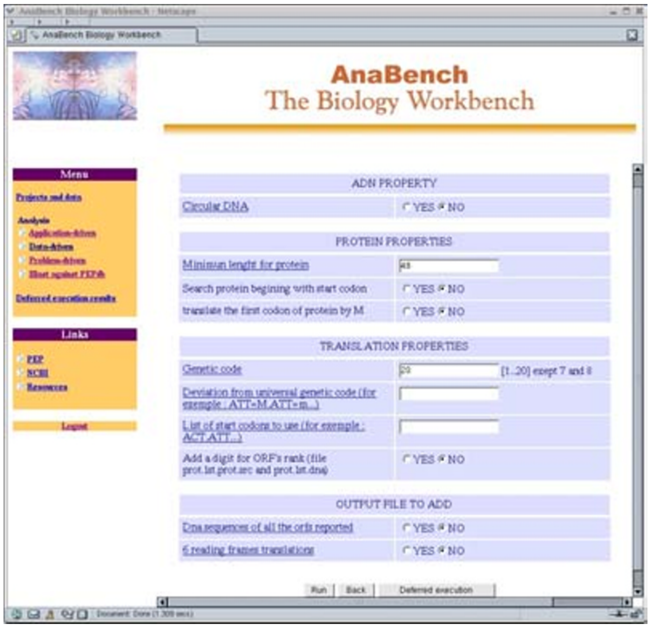
Figure 4
Adding a new application to a specific category.


AnaBench Biology Workbench - Netscape
AnaBench Biology Workbench
Add
Back
Document: Done (0.751 secs)

Application name: flip

Name:	Gencode
Label:	Genetic code
Description:	Use NCBI code number digit instead of the default basic code to perform translation.
Prefix:	-
Abbreviation:	g
Type of value :	Int
Required :	No
Number of required values:	
Syntax :	NameValue
Minimal value:	1
Maximal value:	21
DNA Default value:	1
Protein Default value:	

Figure 5
Adding a parameter description (case of the FLIP application)



ADN PROPERTY	
Circular DNA	○ YES ● NO
PROTEIN PROPERTIES	
Minimun lenght for protein	43
Search protein begining with start codon	○ YES ● NO
translate the first codon of protein by M	○ YES ● NO
TRANSLATION PROPERTIES	
Genetic code	20 [1..20] exept 7 and 8
Deviation from universal genetic code (for exemple : ATT=M,ATT=m...)	
List of start codons to use (for exemple : ACT,ATT...)	
Add a digit for ORF's rank (file prot.lst,prot.rec and prot.lst.dna)	○ YES ● NO
OUTPUT FILE TO ADD	
Dna sequences of all the orfs reported	○ YES ● NO
6 reading frames translations	○ YES ● NO

Figure 6
FLIP JSP screen form.

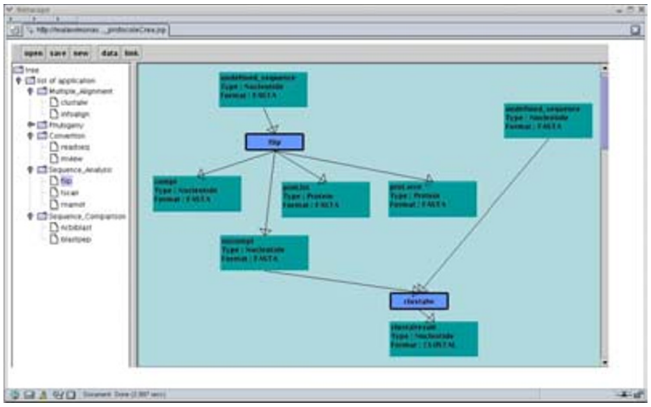


Figure 7
The graphical tool for the design and execution of pathways.

Fourth, from the description of the FLIP application together with its parameters, we generate, by using a Java program called "JSPGenerator", a screen form with which the end-user can specify the parameter values. The generated screen is shown in figure 6.

Sequence Analysis approaches in AnaBench

AnaBench implements three approaches for sequence analysis: application-driven, data-driven, and protocol-driven approaches. The application-driven approach starts by the selection of a tool. Once a tool is selected, the AnaBench system filters the entire data of the user and only presents those that are valid inputs for the selected tool. In the data-driven approach, a researcher can ask, "which tools can I use to mine for information in this data?", as in this case AnaBench shows all available tools that can extract information from a given data set. In the protocol-driven approach, the biologist can employ fully interactive analysis pathways for addressing specific biological questions. Protocols can be defined and executed using a graphical design system with which tool icons are dragged and dropped into a canvas and then linked together. Figure 7 presents a screenshot of this graphical tool. The above analysis approaches and the various actions the user can perform can be inspected online.

Discussion

We have built an interactive Web-based sequence analysis environment with a three-tier architecture including CORBA as a middleware to allow: (i) easy development and deployment of clients and servers, (ii) the ability to use heterogeneous languages and machines, and (iii) the implementation of light Internet clients in the presentation layer using HTML and JSP.

Automated integration of command-line bioinformatics tools in AnaBench has been achieved by two means: the tabular description of categories, applications, and parameters as shown in the previous section, and the utilization of the JSPGenerator. The programmatic effort to incorporate a new tool into AnaBench mainly consists in customizing the automatically generated JSP page for a given tool. As a future enhancement, we consider the integration of tools represented by Java classes that can be dynamically loaded into the system. We successfully tested this approach with the 'TVIEW' tool that displays the results of 'TRNASCAN_SE' and inserts feature annotations into nucleotide sequence files in MASTERFILE format [26]. It should be noted that packages like EMBOSS and BioNavigator took a different approach by using configuration files in the ACD format to describe command-line tools and their parameters. In contrast to ACD configuration files, the tabular tool description we use is straightforward since it does not require a parser for the processing of configuration files.

The integration of local and remote applications in AnaBench is performed in the same manner except that remote applications require dedicated APIs offered by service providers. In the case of remote NCBI BLAST, we have implemented the access to this service using the QBLAST API provided by NCBI.

To avoid crashes at the application server level, the CORBA server could be replicated across the machines of a local network. If one of the replicas fails, the surviving ones can continue the processing of the business logic and thus provide continuous service to the clients. In such a scenario, the CORBA Naming Service will locate an appropriate available CORBA server. Likewise, to avoid crashes of the Web server and to handle a great number of simultaneous requests as demand increases, a cluster of Web servers in combination with a load balancer may be deployed. Several Web cluster solutions are now available.

Conclusions

In this paper, we present the design and implementation of a new Web-based biological workbench environment. AnaBench is innovative in three aspects: (**i**) its three-tier architecture allows easy integration of heterogeneous bioinformatics tools; (**ii**) the recent technologies that we employed for distributed server-side Web computing, such as CORBA, RMI, JDBC, and JSP, provide excellent performance with modest hardware; (**iii**) the user data are stored in a relational database rather than in flat files providing flexibility as to data extraction and interrogation. Another distinctive feature of AnaBench is that it offers three types of sequence analysis: application-driven, data-driven, and protocol-driven approaches.

Authors' contributions

EB designed the architecture of AnaBench, defined the IDL specification of the CORBA server, performed its implementation, and performed the integration of several bioinformatics tools in AnaBench. CS participated in the design of AnaBench, integrated CLUSTALW into AnaBench, and implemented JSP modules for the display of analysis results. GB supervised the project and GB and BFL specified the functionalities of AnaBench. All authors read and approved the final manuscript.

Acknowledgements

This work is a part of the Protist EST program (PEP), funded by Genome-Canada and Genome-Quebec. Funding from CIHR (Canadian Institutes of Health Research, grant nr. GX-15331) and salary and interaction support from the CIAR (Canadian Institute for Advanced Research) to GB and BFL are acknowledged.

The authors thank Bruno Duclouet, Kun Wang, Sabrina Carpentier, Julien Rondeau, and Nicolas Moiroud for their contributions in the development and documentation of AnaBench, and Amy Hauth for her suggestions to the manuscript.

References

1. Lang BF, Burger G: **A collection of programs for nucleic acid and protein analysis, written in FORTRAN 77 for IBM compatible microcomputers.** *Nucleic Acids Res* 1986, **14:**455-465.

2. Pearson WR, Lipman DJ: **Improved Tools for Biological Sequence Comparison.** In *Proc Natl Acad Sci USA* 1988:2444-2448.
3. Schuler GD, Epstein JA, Ohkawa H, Kans JA: **Entrez: molecular biology database and retrieval system.** *Methods Enzymol* 1996, **266:**141-62.
4. Thompson JD, Higgins DG, Gibson TJ: **CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.** *Nucleic Acids Res* 1994, **22:**4673-4680.
5. Felsenstein J: **PHYLIP Phylogeny Inference Package (Version 3.2).** *Cladistics* 1989, **5:**164-166.
6. Entigen inc.: **BioNavigator, The Bioinformatics Workspace.** 1997 [http://www.entigen.com/library/white_papers/BioNavigator.pdf].
7. Siepel A, Farmer A, Tolopko A, Zhuang M, Mendes P, Beavis W, Sobral B: **ISYS: a decentralized, component-based approach to the integration of heterogeneous bioinformatics resources.** *Bioinformatics* 2001, **17:**83-94.
8. Siepel A, Tolopko A, Farmer A, Steadman P, Schilkey F, Perry B, Beavis W: **An integration platform for heterogeneous bioinformatics software components.** *IBM SYSTEMS JOURNAL* 2001, **40:**570-591.
9. TurboGenomics inc.: **TurboBench overview.** [http://www.turboworx.com/files/gen_news.pdf].
10. Unwin R, Fenton J, Whitsitt M, Jamison C, Stupar M, Jakobsson E, Subramaniam S: **Biology Workbench: A WWW-based Virtual Computing and Analysis Environment for the Biological Sciences.** 1999 [http://bsw.ncsa.uiuc.edu/Papers/].
11. Issac B, Raghava GP: **GWFASTA: Server for FASTA Search in Eukaryotic and Microbial Genomes.** *BioTechniques* 2002, **33:**548-556.
12. Rice P, Longden I, Bleasby A: **EMBOSS: The European Molecular Biology Open Software Suite.** *Trends in Genetics* 2000, **16:**276-277.
13. Hucka M, Sauro H, Finney A, Bolouri H: **Introduction to the Systems Biology Workbench.** 2001 [http://www.sbw.sourceforge.net/sbw/docs/intro/intro.pdf].
14. Hucka M, Sauro H, Finney A, Bolouri H, Doyle J, Kitano H: **The ERATO Systems Biology Workbench: Enabling Interaction and Exchange between Software Tools for Computational Biology.** In *Proceeding of the Pacific Symposium on Biocomputing* 2002:450-461.
15. Senger M: **AppLab – A CORBA-Java based Application Wrapper.** 1999 [http://www.hgmp.mrc.ac.uk/CCP11/CCP11newletters/CP11newsletterIsuue8.pdf].
16. Hanna P: **JSP: The Complete Reference.** *McGraw-Hill* 2001.
17. Object Management Group: **The Common Object Request Broker: Architecture and Specification.** 1999.
18. Korab-Laskowska M, Rioux P, Brossard N, Littlejohn TG, Gray MW, Lang BF, Burger G: **The organelle genome database project (GOBASE).** *Nucleic Acids Res* 1998, **26:**138-144.
19. Shimko N, Liu L, Lang BF, Burger G: **GOBASE: the Organelle Genome Database.** *Nucleic Acids Res* 2001, **29:**128-132.
20. Burger G, Lang BF, Golding GB: **PEPdb: The Protist EST Database.** 2002 [http://amoebidia.bcm.umontreal.ca/public/pepdb/welcome.php].
21. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL: **GenBank.** *Nucleic Acids Res* 2003, **31:**23-7.
22. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool.** *J Mol Biol* 1990, **215:**403-10.
23. Macke TJ, Ecker DJ, Gutell RR, Gautheret D, Case DA, Sampath R: **RNAMotif-an RNA secondary structure definition and search algorithm.** *Nucleic Acids Res* 2001, **29:**4724-35.
24. Hauth AM, Joseph DA: **Beyond Tandem Repeats: Complex Pattern Structures and Distant Regions of Similarity.** *Bioinformatics* 2002, **Suppl 1:**S31-7.
25. Lowe TM, Eddy SR: **tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence.** *Nucleic Acids Res* 1997, **25:**955-964.
26. The Organelle Genome Megasequencing Program: **The OGMP Masterfile Format.** [http://megasun.bch.umontreal.ca/ogmp/masterfile/intro.html].