GITIRBio: A Semantic and Distributed Service Oriented-Architecture for Bioinformatics Pipeline

Luis F. Castillo[1], Germán López-Gartner[2], Gustavo A. Isaza[1*], Mariana Sánchez[2], Jeferson Arango[1], Daniel Agudelo-Valencia[2], Sergio Castaño[1]

[1] Systems and Informatics Department, GITIR Research Group. Caldas University, Street 65 # 26-10 Manizales, Colombia
[2]Biology Sciences Department, GITIR Research Group. Caldas University, Street 65 # 26-10 Manizales, Colombia

Summary

The need to process large quantities of data generated from genomic sequencing has resulted in a difficult task for life scientists who are not familiar with the use of command-line operations or developments in high performance computing and parallelization. This knowledge gap, along with unfamiliarity with necessary processes, can hinder the execution of data processing tasks. Furthermore, many of the commonly used bioinformatics tools for the scientific community are presented as isolated, unrelated entities that do not provide an integrated, guided, and assisted interaction with the scheduling facilities of computational resources or distribution, processing and mapping with runtime analysis. This paper presents the first approximation of a Web Services platform-based architecture (*GITIRBio*) that acts as a distributed front-end system for autonomous and assisted processing of parallel bioinformatics pipelines that has been validated using multiple sequences. Additionally, this platform allows integration with semantic repositories of genes for search annotations. GITIRBio is available at: http://c-head.ucaldas.edu.co:8080/gitirbio

1 Introduction

A comprehensive database of biological data, particularly genomic data, is being developed in the scientific community for a wide variety of organisms. This database will not only characterize individuals within species but also species as a whole, driving the exponential growth of biological databases and creating the need for powerful new tools to organize, analyze, and visualize this information. This trend is affecting all fields of knowledge, including medicine, agriculture, animal and plant breeding, ecology and environmental sciences as well as general industry, allowing the emergence of new research specialties, such as computational biology, bioinformatics and biocomputing. New databases constantly appear that describe sequences of new genomes, transcriptomes, proteomes and everything occurs at a rate that exceeds the capacity to process such data by several orders of magnitude. Researchers need to analyze the immense amount of genomic data available today to assign the biological functions of complex genetic, biochemical and physiological processes. This analysis requires high computational capability to perform numerous tasks efficiently, including but not limited to sequence assembly, sequence alignment, functional and structural annotation, structural biology analysis, molecular modeling, gene interaction networks, molecular phylogenetics and comparative genomics.

The overall annotation process consists of identifying biological characteristics associated

[*] To whom correspondence should be addressed. Email: gustavo.isaza@ucaldas.edu.co

with DNA sequences, which is typically accomplished by comparing data from the unknown sequence with sequences previously studied in the laboratory and referenced in databases. Therefore, candidate genes associated with traits of interest, such as susceptibility, disease resistance, environmental adaptation, or animal or plant production, are identified. Given the overwhelming volume of information, it must be presented in a structured manner so that comparisons can be performed quickly by software agents to generate new knowledge by discovering innovative features and relationships between genes.

GITIRBio is the result of this interdisciplinary effort. It was developed as a high-performance software tool based on Semantic Web Services standards for the study of multiple genomes and sequences, and it was validated using the transcriptome of a macromycete fungus. The proposed methodology includes the annotation of genes associated with important metabolic pathways and the study of the functional relationships between genes through innovative software tools that implement Linked Open Data (LOD) search and reconstruct Gene Ontology (GO) type based on semantic language. In addition, comparative analysis was performed with the networks of genes and metabolic pathways described in the KEGG reference database.

Different tools have been presented in the scientific community that achieve some of these purposes, such as Bioconductor [1], Bioperl [2], EMBOSS [3], [4] and Galaxy [5], [6]. However, some behave as asynchronously developed wrapper tools and are federally restricted schemes with respect to some standards. The application of systems biology and bioinformatics methods in biochemistry and biomedicine have recently been reviewed, making the need to develop novel bioinformatics and computational biology techniques evident [7]. Other important efforts, such as WebLab [8] and BioManager [9], provide tools to load, manipulate, share, exchange and analyze biological data, offering an important number of integrated tools to be used for these purposes. By focusing on service-oriented standardized protocols, the advantages that *GITIRBio* offers over other tools include a queuing method within the processing pipeline that was structured according the requirements of researchers together with architecture components that have a degree of interaction according to clearly validated methods established by the scientific community for sequence analysis. Moreover, the platform provides an API that allows integration of new components within the structure, finding information from multiple repositories, both conventional and ontological, as well as different display modules. Furthermore, *GITIRBio* includes management processes with ontological repository modules that interact with the system queue clusters and results are displayed in multiple modes.

This database was developed to minimize the reasons that many researchers claim reduce their use of command line based interaction. This database will automatically develop scripts, parallelize tasks to analyze large data sets of sequences, arrange a significant amount of systems applications, and avoid messy and chaotic processing that with a low level of integration eventually makes interpretation difficult. The *GITIRBio* architecture offers an environment with optimal usability to load, configure, plan and execute distributive processing as well as visualize the results for different sequence formats. The architecture achieves this by utilizing an extensive pipeline to link parallel environments, integrating assemblies, alignments and annotations based on either GO functional relationships or LOD-based gene repositories, and then it displays the results.

2 Materials and Methods

The data used to perform the tests are predominantly from a simulated fungal sequence obtained using next generation sequencing reads as seeds for a simulation process. This decision arose from previous work this group accomplished with transcriptomes from macromycetes fungi; however, the platform supports many different types of biological sequences.

1.1. Modules and components

The underlying system, or the back-end, has two parts, the communication module and the processing module. The communication module acquires the user request through the user interface and then delivers it to the process module. The process module is a wrapper that holds all interactions with the bioinformatics tools. These interactions involve directory management, parameter setup, output management, error handling and notification sending.

Figure 1 shows the logical relationship of the main system modules used in bioinformatics processing and in the general use case, which is presented in Figure 2.

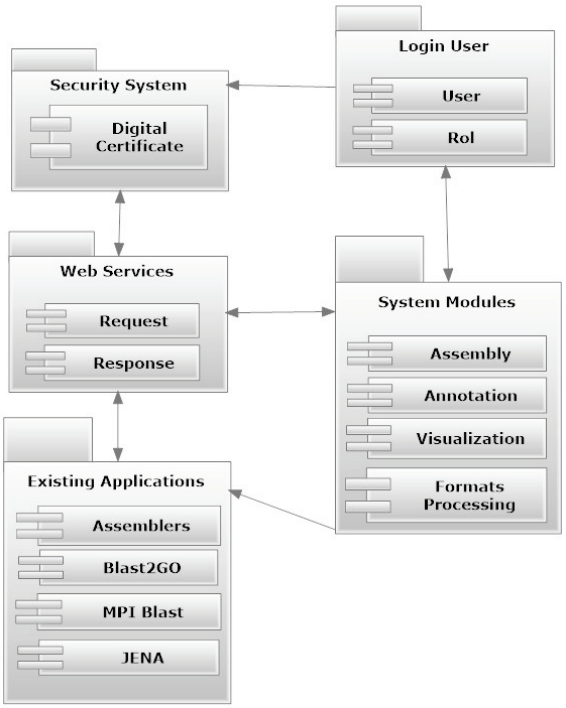


Figure 1: Main modules for *GITIRBio*

Below is a description of the relationships between the modules and their classes:

1.1.1. Security System

This module has the digital certificate and uses X.509 and user identification for their respective roles in the execution of processes in the system. The authentication uses TLS integrated with the Java Cryptography Architecture. The digital certificates are generated via Grid proxy authentication using DOE Grids CA.

1.1.2. System modules

This module contains the bioinformatics pipeline for assembly and annotation. It also displays the genetic correlations found in queries or comparisons with Gene Ontology.

1.1.3. Web Services

This module enables communication with other platforms for distributed and parallel processing, thus optimizing resources.

1.1.4. Existing applications

This module has applications developed to support information processing (i.e., for different stages of the pipeline and data visualization).

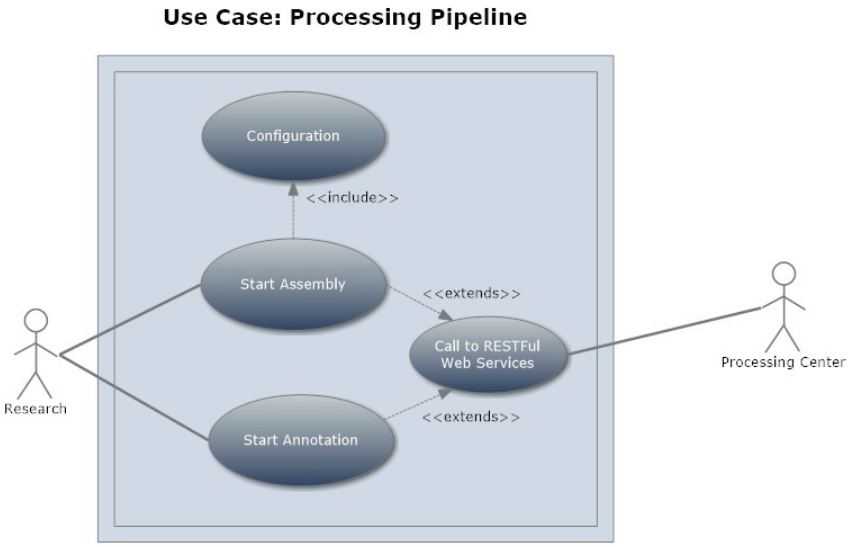


Figure 2: Case Use Scenario

Use case	Processing pipeline	
Objective in context	The researcher begins a pipeline process (assembly and annotation).	
Pre-conditions	The environment variables must be configured for use in each activity.	
Success condition	Starts the activity of assembly or annotation.	
Failure condition	The server responds with error and does not initiate the activity	
Primary actors	Research	
Secondary actors	Processing center	
Main flow	Step	Actions
	1	The researcher selects the configured options.
	2	A file is attached with the sequences to be processed
	3	The pipeline process starts
Extensions	Step	Actions
	1.1	The research did not find the configured options.
	2.1	The research cannot upload the file.
	2.2	The file is not in the correct format.
	3.1	Unable to start the pipeline process.

1.2. Architecture

The components have been developed using the REST services standards. The REST services allow developers to use the HTTP protocol. This permits communication between machines directly with CRUD operations (create, read, update and delete) and HTTP methods.

The platform has been designed to perform data transfer via a communication service to other web applications and web services, which employs a high level of cohesion and low coupling. Because REST works on the HTTP protocol, these services enable interoperability of different platforms without creating complex adaptations. Communication within the platform is achieved using Web Services running processes on servers, where the necessary scripts are executed to perform different stages of the pipeline. The data necessary to identify the process to be performed on the server are sent in JSON serialized format.

The key concept of the SOC (Service-oriented computing) paradigm is interoperability between different software applications running on a variety of hardware and software platforms.

The general architectural model for Web Services is composed of a service provider (SP), a service record (SR) and a service consumer (SC). The SP publishes services to a SR, and then the SC looks for a service in the SR. If the user finds the desired service, a connection between the SC and the SP is established that, using REST signatures, allows the creation and implementation of new modules without creating changes in pre-existing ones. The general architecture is presented in Figures 3 and 4.

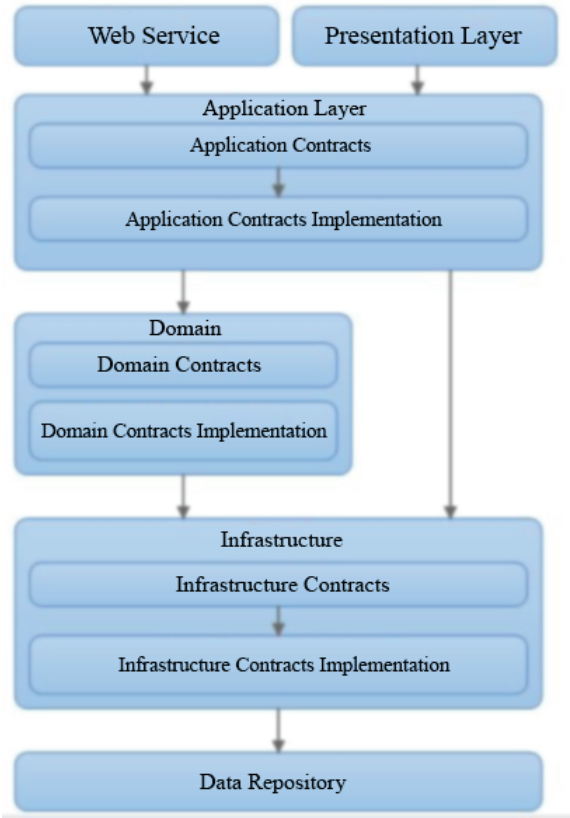


Figure 3: *GITIRBio* High Level Architecture.

The functionality and responsibility of each of the layers can be described as follows:

2.2.1 Presentation Layer. This layer contains the visual components and is where the user can launch a process and watch its status.

2.2.2 Web Services. The web services enable interaction with the integrated and external systems. This allows the platform functionality to communicate with other applications and to receive notifications of task completion.

2.2.3 Application. This layer knows the business logic and will therefore not enter any other logic that is made here to synchronize the overall system actions. This layer can send notifications to the system administrator, upload files to a FTP server if necessary, and run other actions related to the process launched.

2.2.4. Domain. The domain layer executes all business logic related to the assembler or the annotation parameters, and it can also validate the tree or graphs generated in the information visualization process.

2.2.5. Infrastructure. This layer interacts with the repositories, each process is registered here, and it can help in the auditing process. It can also interact with files or other web services.

2.2.6. Data Repository. This layer consists of repositories and the files used in the different stages of the pipeline.

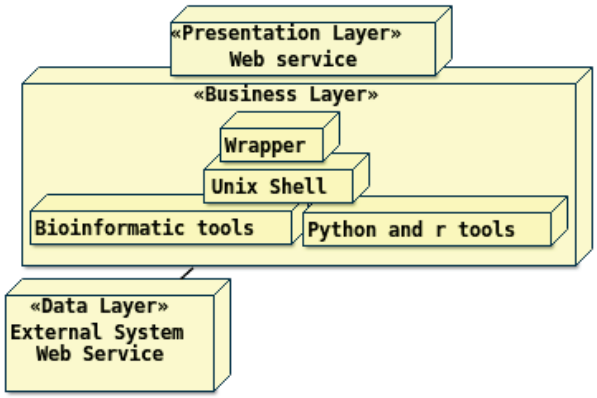


Figure 4: *GITIRBio* Low Level Architecture.

For the purposes of auditing, a security log record is generated from the data produced as the system processes each task and phase in the pipeline. These data relate to the sequence assembly, which may be performed by one of a number of different platform-specific methods, to the annotation and to the visualization of the assembled sequence according to the method selected from several options. This processing behavior is illustrated in Figure 5.

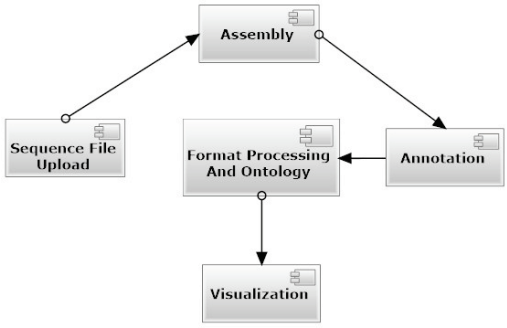


Figure 5: Linked Processes for the *GITIRBio* pipeline.

The platform is designed to be extensible with low coupling between modules, so that continuous development of functionality can be ongoing without impeding the existing platform. The system can be extended using method signatures added into the embedded interfaces for the REST services.

Figure 6 illustrates the assembly process, the stage of the pipeline where the configuration of the required parameters is performed as well as the assignment of the type of technology to use and the desired assembler (section 1). These are the parameters that the platform uses to communicate with the REST web services. Section 2 of Figure 6 has the option of uploading a file with the sequences to be processed and that file is referenced in the chosen Web service. Section 3 has the list of assemblies that have been prepared as well as their status and a description of the parameters used for execution; the status notification has 3 states (Asking, Processing, and Completed).

In Figure 7, the annotation process based on the file generated in the assembly stage is shown. There are two methods of annotation, traditional or utilizing Gene Ontology annotation. From an uploaded file, the data can undergo the assembly process and then proceed to the annotation process by invoking *WebService* to be in charge of the process. Similar to the assembly process, annotation has a history of processes and their status for each of the records.

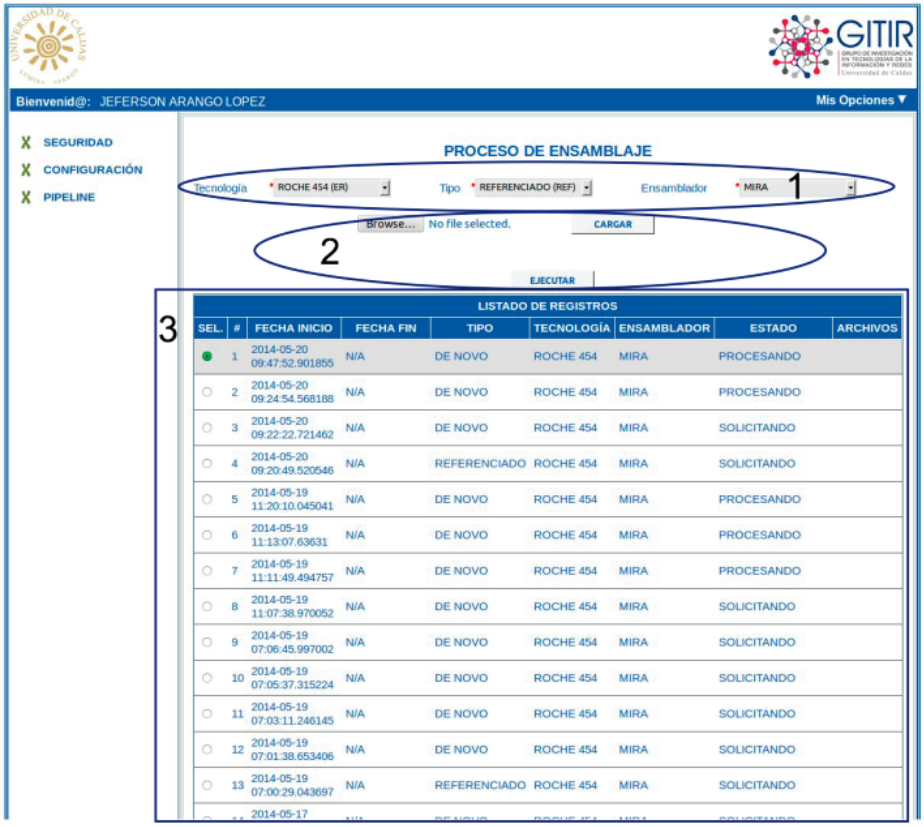


Figure 6: *GITIRBio* Assembly process.

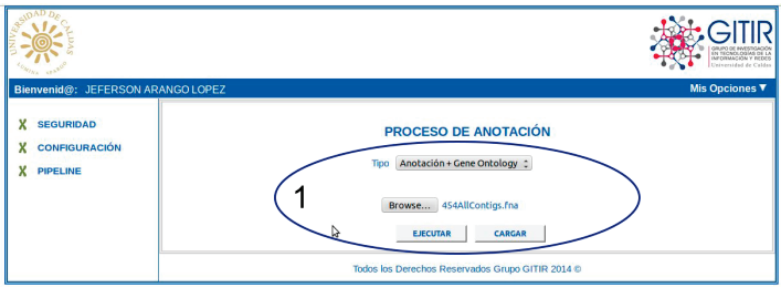


Figure 7: *GITIRBio* Annotation via Gene Ontology (GO).

1.3. Semantic Component

Multiple approaches based on the Semantic Web for integrating biomedical data have recently been proposed [10], [11], [12], [13].

In the biomedical field, exemplary progress has been achieved by Bio2RDF [14], a system that allows the integration of a large number of biomedical databases through Semantic Web RDF access technologies, for representing data using SPARQL protocol and RDF query language. For this purpose, many databases have been converted to RDF by special scripts called *RDFizers*, while some information systems offer a variable format that interfaces directly with the system. An extension of this approach is evident in references [15] and [16], where the Chem2Bio2RDF Linked Open Data (LOD) portal for chemical and biological systems is presented to facilitate drug discovery. It converts approximately 25 datasets of genes, compounds, drugs, side effects, diseases and RDF triples to links to other LOD bubbles, such as Bio2RDF, LODD and DBPedia. The portal is based on D2R server and provides a SPARQL endpoint by adding in a few facets of RDF features, an easy to use SPARQL query generator, MEDLINE/PubMed cross validation service, and visualization via Cytoscape.

In *GITIRBio*, the semantic component is treated with two main processes. The first process is the verification of annotations against GO (*Gene Ontology)*, invoking native Java methods to Blast2GO for genes annotated via GO. The second is the *RDF-Ization* of conventional BLAST annotations to triples, where individual annotations are obtained to corroborate functional relationships against different biomedical ontologies using SPARQL. These approaches have already been validated by the authors with other sequences, such as coffee and coffee rust fungus [17], [18].

1.4. Bioinformatics Validation

2.4.1 The bioinformatics pipeline

The processing sequence for this stage consists of assembly, annotation, alignment and visualization, which is exhaustively followed using the *GITIRBio* services through the pipeline implemented as shown in Figure 8. For *in silico* analysis of a biological organism, it is a common strategy to obtain a digital version of the information found in molecules such as DNA or RNA using high performance sequencing techniques. In the case of the fungus studied herein, a digital representation of the RNA-seq was developed using a simulation of next generation sequencing reads seeded by preliminary sequences obtained previously in the laboratory (unpublished data). The scripts used for this purpose were NeSSM [19] and Plantagora Tools [20]. We linked this fungal dataset because efforts are underway to proceed with the sequencing of a species of fungi in the *Macromycetes* family. However, the *GITIRBio* platform is open to many different types of sequencing data (DNA, RNA or protein).

To validate the pipeline, it initiated using simulated RNA-seq data (a simulated transcriptome) in *fasta* format. To extract the biological information associated with the simulated transcriptome, the computational processes called were implemented from the *GITIRBio* services and incorporated into a pipeline at the SFG from Stanford University [21]. In the sequencing process an extended collection of sequencing reads is generated; these represent scattered sectors of the RNA molecules that will eventually represent the transcriptome. Therefore, the pipeline begins with the assembly phase. Because the reference database lacked a comparable transcriptome originating from a close relative species, a *de novo* assembly strategy was implemented instead of a referenced one.

The second phase of the embedded pipeline consists of two simultaneous processes, annotation and results analysis of data from the assembly phase. In the assembly results analysis process, a set of tools written in *python* and *R* were created to obtain visual information about the quality of the assembly. The annotation process involves the identification or characterization of any gene outcome of the assembly phase. Therefore, applications that search the biological databases for information matching our simulated data were used or at least associated with the genes reported from experimental sources. BLAST is a set of bioinformatics applications and local alignment tool that implements the Smith-Waterman algorithm. The strategy of this algorithm is to find similar regions between an unidentified sequence and another identified sequence through the use of a scoring system that can be determined by the user. For the annotation process, three protein databases, *nr, uniprot sprot* and *uniprot trembl*, were searched with *BlastX* to compare transcriptomes in nucleotide and amino acid notation. *BlastX*translates the amino acid transcript and then begins the process of local alignment. In the next annotation process step, a set of tools were adapted to obtain information via Web Services using the identifiers of the characterized sequences, finally creating a consolidated result of the entire annotation process. The last stage of the pipeline was the results analysis, where a set of tools written in python were created using *R* and *bash* scripting to facilitate visualization of information concerning quality annotation parameters.

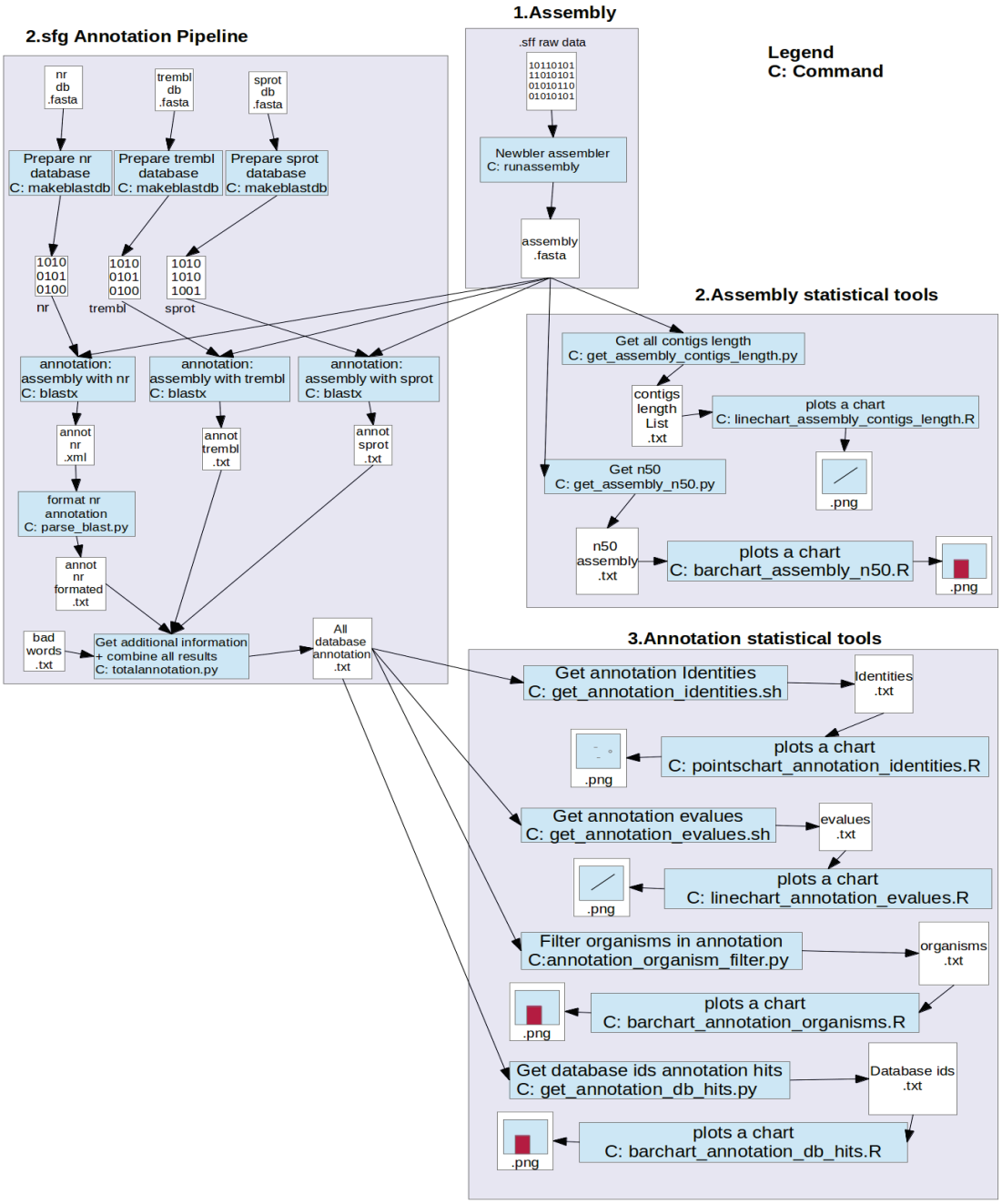


Figure 8: *GITIRBio* pipeline.

The processing module has 3 main components, the interface module, the wrapper and the bioinformatics processing of notifications and persistence (Figure 9). The interface module is responsible for receiving the user requests to initiate the bioinformatics process. It is implemented as a *RESTful* Web Service, thus technology deployment in the user interface does not affect its operation. The data format for communication between the user interface and processing module is JSON, which is lightweight and easy to process. Consequently, the parameters and phases of bioinformatics to start the processing are easily defined. For information security, encryption techniques are applied on the JSON data traveling between the two parties, and adding *CORS* into *RESTful* allowed refusals of requests for services to unauthorized domains. The bioinformatics processing wrapper is responsible for managing the execution phase, for generating results using a set of tools developed in *python*, *bash* and *R*, producing notification messages to the module responsible for persistence and sending notifications to the database and the user interface.

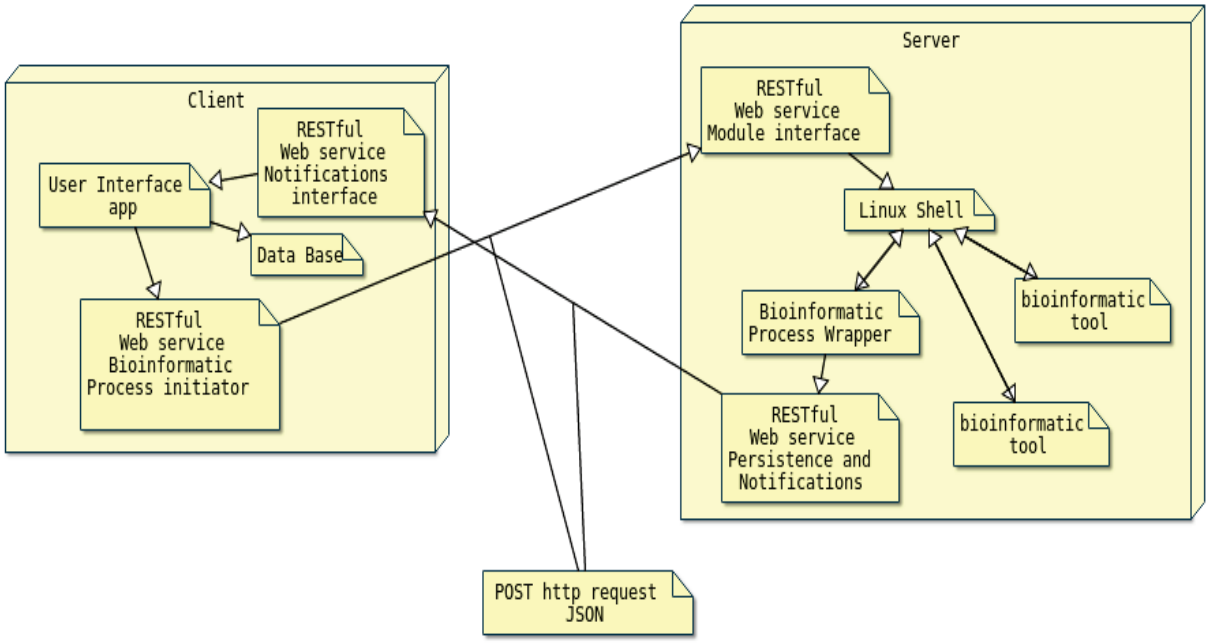


Figure 9: Deployment diagram of bioinformatics platform, where the JSON data format is the way the user interface and processing modules communicate.

The distribution and parallelization behavior is implemented by invoking MPI processes for the assemblers and annotation. This supports the parallelism via distributed memory scheme, as is the case with *MIRA*, *Velvet* and *MPI_BLAST*, and the decomposition of some repositories of annotation data to find functional relationships. Moreover, we are integrating the platform with the *HTCondor* Grid-based middleware for scheduling, queuing and process migration in an opportunistic computing environment model.

3 Results and discussion

Applying the pipeline generated by this service architecture, multiple phases can be invoked using multiple choices. For the assembly step, three assemblers are integrated in the *GITIRBio* platform, *Newbler, Mira and Velvet*; however, we are examining new service configurations to link other assemblers. For this research evaluating a simulated generic fungal organism, the assembly and annotation did not require an exhaustive biological analysis. Furthermore, Figures 10a and 10b are evidence that the display module is integrated into the developed tool referenced and described in this paper. Accordingly, the annotation process, displayed in Figures 11a and 11b, is invoked from a *GITIRBio* process for shared memory, conventional *BLAST* and clustering distributed *MPI_BLAST*, which are automatically decomposed and parallelized. In this case, depending on the settings for the queue manager and the scheduling system[2], the processes are parallelized in any of these modes. For processing the functional annotations, parallelization is derived from decomposition of the files listed according to the number of nodes. This runs an agent responsible for dividing data between processes and communicating via MPI to obtain the results of sequence comparison using the SPARQL LOD (Linked Open Data) repositories based on Open Biomedical Ontologies (OBO), which allows classification and has other methods for querying biological relational databases. The

[2] For the cluster tested in GITIRBio, *HTCondor* is the job and scheduler manager. GITIRBio can be accessed at http://c-head.ucaldas.edu.co:8080/gitirbio/

visualization component is parameterized with some conventional queries based on the specifications of experts in biology to analyze the results of each stage in the workflow. As feedback is provided by life sciences experts, additional and improved options associated with the graphics pipeline can be added for the automatic generation of graphs and a more user-friendly experience during the data analysis process.

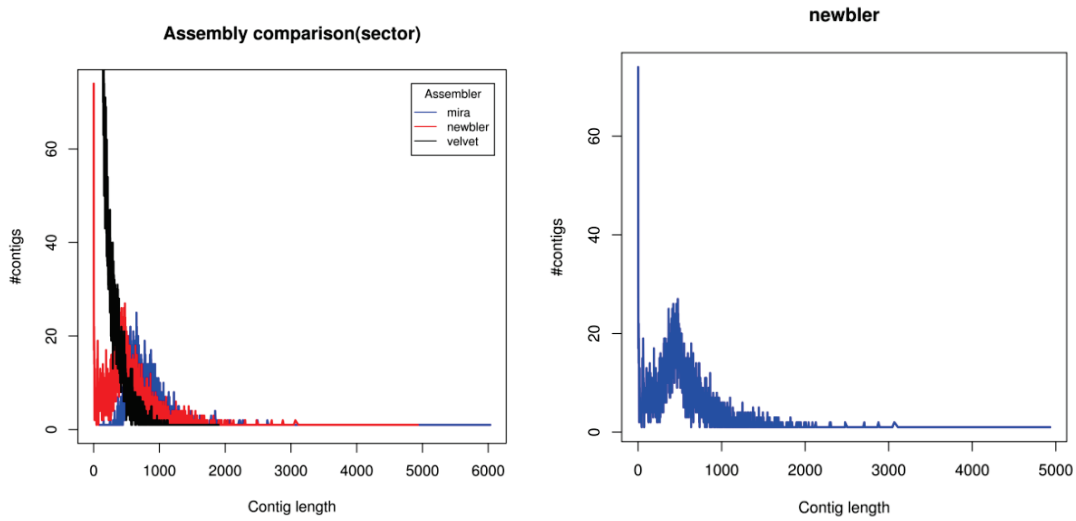


Figure 10: (a) Assembly comparison using the BioPython R component integrated in *GITIRBio*. (b) Assembly ContigLength using the BioPython R component integrated in GITIRBio.

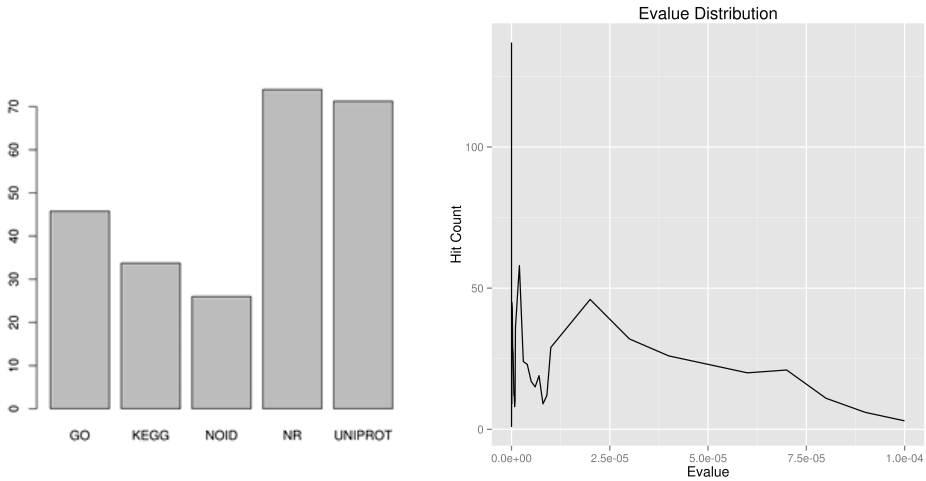


Figure 11: (a) Annotation using the BioPython R component integrated into *GITIRBio*, where x-axis the repository and y-axis is # annotation hits found. (b) Annotation E-Value using the BioPython R component integrated into *GITIRBio*.

Furthermore, network visualization of genes and functional relationships from *D3* library are included as resources, as demonstrated in Figure 12, where the semantic graphs of these links are shown in a triplet subject-predicate-object model according to the results obtained against OBO. In this figure, the relationship between different types of organisms in the kingdom Fungi are depicted in three levels of relationships; however, the platform supports n levels of relationships, allowing the visualization of organisms through the kingdom and their orders. The data for this experiment met the Linked Open Data standards. Therefore, queries can be processed through bioinformatics repositories and the results can be displayed not only in the shape of HTML tables, graphs or relational databases but also in network representations.

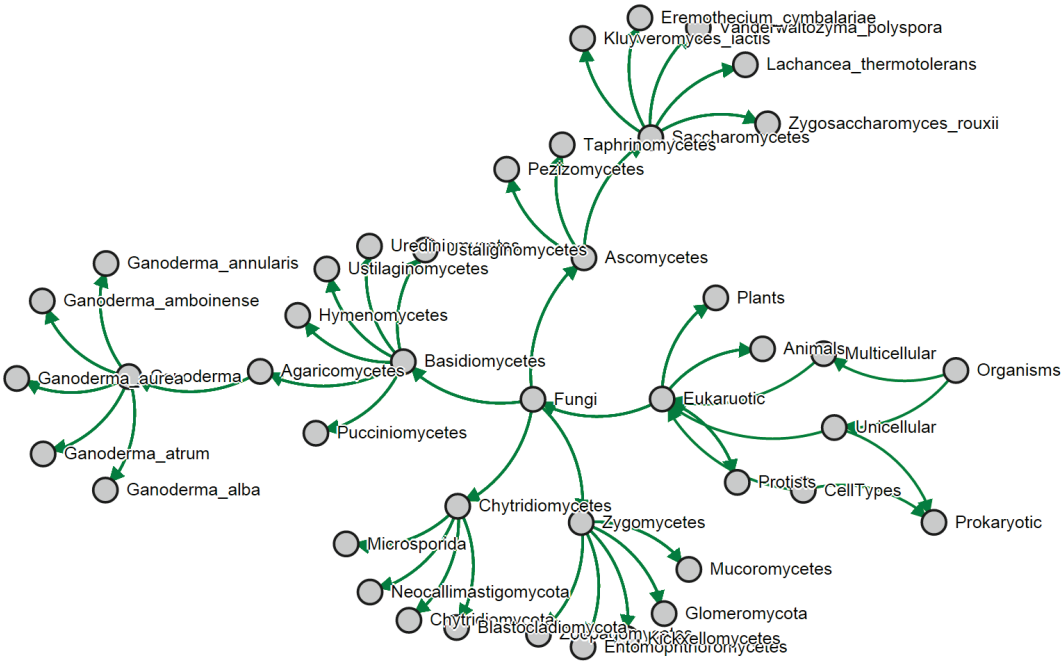
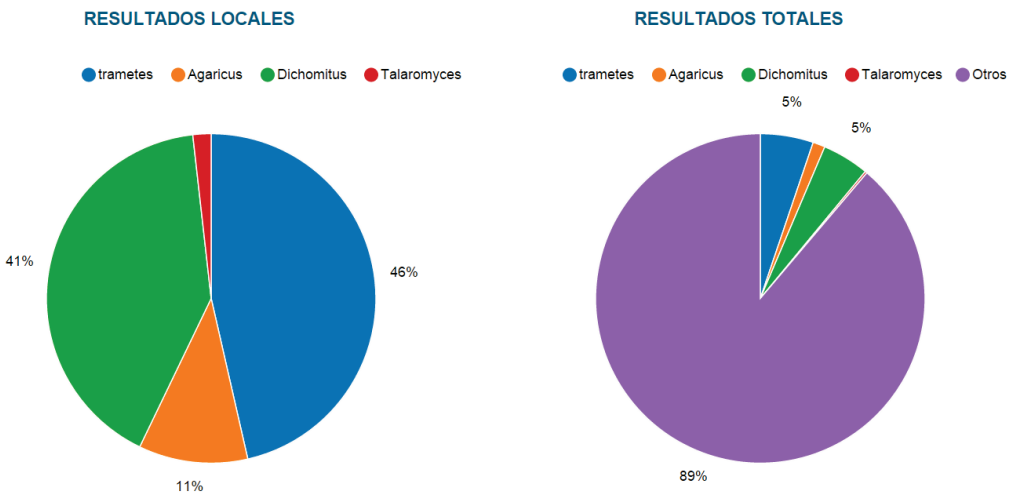


Figure 12: *GITIRBio* functional relationship using the D3 Viz component.

The ratios obtained and plotted not only represent organisms but the relationships between genes or proteins can also be graphed, provided a LOD format is used. The *GITIRBio* platform allows searching for terms related to genes, organisms, proteins or other biologically relevant items that are present in a data log file. This process is done dynamically; thus, the platform user can enter the search terms and indicate what sector of the file needs to be processed. The platform performs the search and displays graphical statistics for the data it found (Local Results) and also for the total terms found in the archive (Total Results). Therefore, the user can see multiple relationship views, as shown in Figure 13.



LISTADO DE TÉRMINOS			
Término	Número de Repeticiones	Porcentaje Local	Porcentaje Total
trametes	26	46 %	5.2 %
Agaricus	6	11 %	1.2 %
Dichomitus	23	41 %	4.6 %
Talaromyces	1	2 %	0.2 %
Otros	444		88.8 %

Figure 13: *GITIRBio* search term visualization component

For the behavior of the *GITIRBio* distribution, a test was executed in the HPC cluster at the University of Caldas, comparing it with other conventional pipeline tools (via command line) and using Galaxy to run jobs in the same cluster. The results of this analysis are shown in the following table:

Table 1: Performance using GITIRBio

Software/Tools	Process	Length Sequence Tested	Cores and Mem used for Test	Execution Time (mins)
Pipeline using directly command line tools	Assembly	12.500 reads	32 Cores	98
	MIRA	90 contigs	64 GB RAM	
	Newbler	143 contigs		
	Velvet	74 contigs		129
	Annotation			134
	BLAST			
	HMM			
	GO			390
Galaxy	Assembly	12.500 reads	32 Cores	98
	Velvet		64 GB RAM	
	Annotation			
				129
				34
GITIRBio	Assembly	12.500 reads	32 Cores	46
	MIRA	90 contigs	64 GB RAM	
	Newbler	143 contigs		
	Velvet	74 contigs		91
	Annotation Conventional BLAST			77
	BLAST with MPI / using HTCondor +			19
	Torque + Maui			127
	HMM Distributed			86
	Annotation using GO			244

As demonstrated, the use of a platform that interacts with an underlying distributed system accelerates several pipeline tasks in addition to facilitating interaction with its users by not requiring either multiple remote interventions or the manual execution of tasks. However, the advantages of *GITIRBio* are supported not only by these criteria but also on its ability to integrate new components via registered partnerships within the service.

4 Conclusions

The development of new computing methodologies to assist and expedite the work of scientists in the life sciences is well justified and requires interdisciplinary work. In this paper, the implementation of a service-based platform that automates the most important phases of a bioinformatics pipeline was presented that operates in a distributed environment, integrating semantic annotations against biological ontology repositories and visualization components. The platform can scale any plug-in based protocol to be implemented in the layers of the architecture services using signatures. Additionally, the system is being integrated with a queue manager and planning, such as *HTCondor* that is running on the University of Caldas cluster, where concurrent user requests can be controlled. The system has been tested with

simulated as well as real sequences of coffee rust fungus and other organisms of the genus fungi, one of the major interests of our research group. The integration of multiple assemblers, parallel decomposition models for automatic assembly and annotation, and the search and exploitation of semantic updates provides an innovative framework for this platform. Finally, an added value provided by the *GITIRBio* platform to the conventional front-end systems is the ability to integrate display layers into each of the workflow tasks.

GITIRBio is available at: http://c-head.ucaldas.edu.co:8080/gitirbio.

Acknowledgments

This work was supported by the Call for Special Project funding for Research and Innovation at the University of Caldas (2013).

References

[1] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome. Biol.*, 5(10):R80, 2004.

[2] J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, et al. The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, 12(10):1611–1618, 2002.

[3] P. Rice, I. Longden, and A. Bleasby. Emboss: the European molecular biology open software suite. *Trends Genet.*, 16:276–277, 2000.

[4] A. Rosenthal, P. Mork, M. H. Li, J. Stanford, D. Koester, and P. Reynolds. Cloud computing: a new business paradigm for biomedical information sharing. *J. Biomed. Inform.*, 43(2):342–353, 2010.

[5] B. Liu, B. Sotomayor, R. Madduri, K. Chard, and I. Foster. Deploying Bioinformatics Workflows on Clouds with Galaxy and Globus Provision. In: *Proceedings of SC Companion: High Performance Computing, Networking Storage and Analysis*, 1087–95, 2012.

[6] B. Liu, R. K. Madduri, B. Sotomayor, K. Chard, L. Lacinski, U. J. Dave, et al. Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses. *J. Biomed. Inform.,* 49:119-133, 2014. http://dx.doi.org/10.1016/j.jbi.2014.01.005

[7] Y. Cai, T. Huang, L. Chen, and B. Niu (Editors). Application of Systems Biology and Bioinformatics Methods in Biochemistry and Biomedicine. *BioMed Research International*., Volume 2013, Article ID 651968, 2013. http://dx.doi.org/10.1155/2013/651968

[8] X. Liu, J. Wu, J. Wang, X. Liu, S. Zhao, Z. Li, et al. WebLab: a data-centric, knowledge-sharing bioinformatic platform. *Nucleic Acids Res.*, 37(Web Server issue), W33–W39, 2009. http://dx.doi.org/10.1093/nar/gkp428

[9] S. Cattley and J. W. Arthur. BioManager: the use of a bioinformatics web application as a teaching tool in undergraduate bioinformatics training. *Briefings in Bioinformatics*, 8(6):457–465, 2007. http://dx.doi.org/10.1093/bib/bbm039

[10] S. Stephens, D. LaVigna, and M. DiLascio, J. Lucian. Aggregation of bioinformatics data using Semantic Web technology. *Journal of Web Semantics,* 4(3):216–221, 2006.

[11] L. Dhanapalan and J. Y. Chen. A case study of integrating protein interaction data using semantic web technology. *Int. J. Bioinform. Res. Appl.,* 3(3):286–302, 2007.

[12] H. F. Deus, R. Stanislaus, D. F. Veiga, C. Behrens, I. I. Wistuba, J. D. Minna, et al. A Semantic Web management model for integrative biomedical informatics, 3(8):e2946, 2008. http://dx.doi.org/10.1371/journal.pone.0002946

[13] A. Miles, J. Zhao, G. Klyne, H. White-Cooper, and D. Shotton. OpenFlyData: an exemplar data web integrating gene expression data on the fruit fly Drosophila melanogaster. *J. Biomed. Inform.* 43(5):752–761, 2010.

[14] F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *J. Biomed. Inform.* 41(5):706–716, 2008.

[15] B. Chen, X. Dong, D. Jiao, H. Wang, Q. Zhu, Y. Ding, et al. 2010. Chem2Bio2RDF, a semantic framework for linking and data mining chemogenomic and systems chemical biology data. *BMC Bioinform.* 11: 255, 2010. http://dx.doi.org/10.1186/1471-2105-11-255

[16] B. Chen, Y. Ding, H. Wang, D. J. Wild, X. Dong, Y. Sun, et al. Chem2Bio2RDF: A Linked Open Data Portal for Systems Chemical Biology. Web Intelligence and Intelligent Agent Technology 1:232–239, 2010. http://dx.doi.org/10.1109/WI-IAT.2010.183

[17] L. Castillo, N. Galeano, G. Isaza, and A. Gaitán. Construction of coffee transcriptome networks based on gene annotation semantics. *J. Integr. Bioinform.*, 9(3):1–14, 2012. http://dx.doi.org/10.2390/biecoll-jib-2012-205

[18] L. Bertel, G. Isaza, L. Castillo, A. Gaitán. Towards a Linked Open Data Model for Coffee Functional Relationships. *Advances In Intelligent Systems And Computing,* 232:121–126, 2013.

[19] Ch. Wei. NeSSM: a Next-generation Sequencing Simulator for Metagenomics. Taken from: http://cbb.sjtu.edu.cn/~ccwei/pub/software/NeSSM.php Consulted: April 2014.

[20] Plantagora: Simulation of Next Gen Sequencing Reads. Taken from: http://www.plantagora.org/tools_downloads/read_simulation.html Consulted: February 2014.

[21] P. De Wit, M. H. Pespeni, J. T. Ladner, D. J. Barshis, F. Seneca, H. Jaris, et al. The simple fool's guide to population genomics via RNA-Seq: an introduction to high-throughput sequencing data analysis. *Molecular Ecology Resources* 12(6):1058–1067, 2012.