Sigeru Omatu Miguel P. Rocha José Bravo
Florentino Fernández Emilio Corchado
Andrés Bustillo Juan M. Corchado (Eds.)

Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living

10th International Work-Conference
on Artificial Neural Networks, IWANN 2009 Workshops
Salamanca, Spain, June 10-12, 2009
Proceedings, Part II

Springer

Volume Editors

Sigeru Omatu
Graduate School of Engineering, Osaka Prefecture University, Osaka, Japan
E-mail: omatu@cs.osakafu-u.ac.jp

Miguel P. Rocha
Department of Informatics / CCTC, University of Minho, Braga, Portugal
E-mail: mrocha@di.uminho.pt

José Bravo
MAmI Research Lab, University of Castilla-La Mancha, Ciudad Real, Spain
E-mail: jose.bravo@uclm.es

Florentino Fernández
Department of Informatics, University of Vigo, Ourense, Spain
E-mail: riverola@uvigo.es

Emilio Corchado
Higher Polytechnic School, University of Burgos, Burgos, Spain
E-mail: escorchado@ubu.es

Andrés Bustillo
Higher Polytechnic School, University of Burgos, Burgos, Spain
E-mail: abustillo@ubu.es

Juan M. Corchado
Department of Informatics, University of Salamanca, Salamanca, Spain
E-mail: corchado@usal.es

Library of Congress Control Number: Applied for

CR Subject Classification (1998): J.3, I.2, I.5, C.2.4, H.3.4, D.1, D.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-02480-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-02480-1 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12695317 06/3180 5 4 3 2 1 0

Intuitive Bioinformatics for Genomics Applications: Omega-Brigid Workflow Framework

David Díaz[1], Sergio Gálvez[1], Juan Falgueras[1], Juan Antonio Caballero[2], Pilar Hernández[3], Gonzalo Claros[4], and Gabriel Dorado[5]

[1] Dep. Lenguajes y Ciencias de la Computación, Universidad de Málaga
[2] Dep. Estadística, Campus de Rabanales C2-20N, Universidad de Córdoba, 14071 Córdoba
[3] Instituto de Agricultura Sostenible (IAS-CSIC), Alameda del Obispo, s/n, 14080 Córdoba
[4] Dep. Biología Molecular y Bioquímica, Universidad de Málaga
[5] Dep. Bioquímica y Biología Molecular, Universidad de Córdoba 14071 Córdoba, Spain
{david.diaz, galvez}@lcc.uma.es,
{bb1dopeg, vtic, ge1hemop}@uco.es,
{jfalgueras, claros}@uma.es

Abstract. The recent developments in life sciences and technology have produced large amounts of data in an extremely fast and cost-efficient way which require the development of new algorithms, coupled with massively parallel computing. Besides, biologists are usually non-programmers, thus demanding intuitive computer applications that are easy to use by means of a friendly GUI. In addition, different algorithms, databases and other tools usually lie on incompatible file formats, applications, operating systems and hardware platforms. It is therefore of paramount importance to overcome such limitations, so that bioinformatics becomes much more widely used amongst biologists. The main goal of our research project is to unify many of these existing bioinformatics applications and resources (local and remote) in one easy-to-use environment, independent of the computing platform, being a concentrator resource tool with a friendly interface. To achieve this, we propose a tool based on a new, open, free and well-documented architecture called Biomniverso. Two main elements make up such a tool: its kernel (Omega), which supplies services specifically adapted to allow the addition of new bioinformatics functionalities by means of plugins (like Minerva, which makes easy to detect SNP amongst a set of genomic data to discover fraudulent olive oil), and the interface (Brigid), which allows even non-programmer laboratory scientists to chain different processes into workflows and customize them without code writing.

Keywords: resource integration, SNP, plugins, GUI, online services, workflow.

1 Introduction

The recent advances in biology in general and molecular biology and genomics in particular have produced large amounts of data in an extremely fast and cost-efficient way. One of such accomplishments is the "next-generation sequencing", which is expected to allow the sequencing of a human genome in a single day for $1,000 [1]. Yet, such enormous amounts of data require the development of new algorithms,

S. Omatu et al. (Eds.): IWANN 2009, Part II, LNCS 5518, pp. 1084–1091, 2009.

coupled with massively parallel computing. Such goals can be accomplished now due to the availability of multicore processors and frameworks like the Open Computing Language (OpenCL) for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, and others kinds of processors. Besides, biologists are usually non-programmers, thus demanding intuitive computer applications that are easy to use by means of a Graphical User Interface (GUI) [2].

Bioinformatics is the integration of biology and computer sciences, in order to find solutions to current biology problems [3]. Yet, there is overwhelming bioinformatics casuistry, including many file formats, interfaces, approaches and platforms that in practice limit or even block the use of such tools by non-programmers in general and genome researchers in particular. For example, the GenBank data type for sequences is one of the most extended sequence representation formats, yet some tools like ClustalW2 do not recognize such a format for their inputs [4]. Thus, different algorithms, databases and tools have been developed using incompatible software and hardware resources, used for similar purposes but in a somehow chaotic approach, creating a lack of standards which hinders the progress of science and technology.

The main goal of our research project is to unify many of these existing bioinformatics applications and resources (local and remote) in one easy-to-use environment, which is independent of the operating system and hardware platform used, being a concentrator resource platform with a friendly interface. To achieve this, our platform is based on a new, open, free and well-documented architecture called Biomniverso. Two main elements make up such tool: its kernel, Omega, which supplies services specifically adapted to allow the addition of new bioinformatics functionalities by means of plugins (Core and Minerva plugins are supplied with the current implementation). On the other hand there is the interface, Brigid, which allows laboratory scientists to chain different processes together (workflows) with minimal effort and execute them with a intuitive user interface allowing a graphical control of the execution and quick experiencing with different data.

We propose the use of plugins for different resource unification instead of other techniques such as application linking [5] or wholly formalized connectors [6] and connector composition [7] for compatibility reasons. Plugins are easily extendable, have been widely used in biology [8] and they are an efficient way to make use of algorithms written for other operative systems. As many bioinformatics algorithms are available via web as remote services, Internet can be exploited to play a central role for the execution of different workflows though Omega-Brigid, which also accepts local resources and processes. The basis of this platform was sketched in [9] and this work focuses on its final architecture and implementation.

2 Workflow Development and Deployment

The Omega-Brigid framework was designed to concentrate and reuse bioinformatics resources by means of workflows. A workflow is a processes chain -displayed as a directed graph- in which a collection of tasks is automated and flows through processes (nodes), following different connections in order to execute them in each node of the drawn graph.

Thus, Omega-Brigid registers as many resources, processes and algorithms as possible using Omega plugins support and management. Additionally, we have

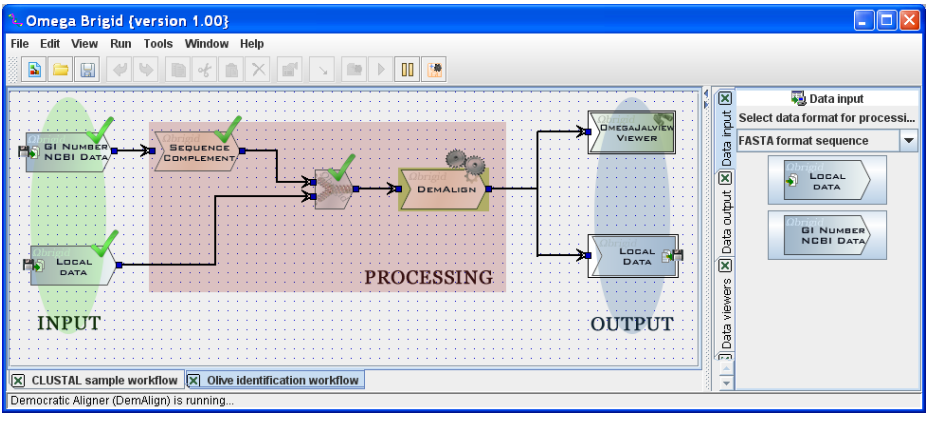

Fig. 1. Workflow example using the Omega-Brigid tool. A workflow has three main stages: *input*, *processing* and *output*. Input phase loads data from resources (here, from *local files* and a *remote NCBI database*). Processing stage carries out the data transformation using algorithms, filters, flow controlling, etc. (here, a sequence is translated into its *complement* and then joined to another sequence to undergo a multiple alignment with the *DemAlign* in the Minerva plugin). In the output phase, results may be displayed, printed saved and/or edited. Here, the data is saved in a *local file* and opened into a viewer customized for SNP finding (*OmegaJalview*).

developed the tool to allow the user to customize the workflow by means of the Brigid interface. The workflow enhancements and optimizations can be accomplished without code writing; thus even non-programmers can take advantage of such workflow customization. Figure 1 shows an example of a workflow created using Omega-Brigid, in which several nodes or cells are connected with arrows: every cell represents a bioinformatics process that can have inputs and outputs.

These arrows show the way in which data flows from one cell into another. The Omega-Brigid engine executes each workflow cell as soon as its inputs are ready, allowing parallel execution of graph's branches. To design and run workflows in Omega-Brigid is a very easy task due to its user-friendly GUI (Brigid), see Figure 1.

New processes and functionalities may be added by installing new plugins into Omega-Brigid. This way, the engine can manage plugins and their cells, albeit with no bioinformatics elements to work with, so the main algorithms are collected in the Core and Minerva plugins.

3 Biomniverso Architecture

The main task of the Biomniverso architecture is to enable users to register, arrange and use processes in workflows. However, this is a very complex task, because a process has three different states in Omega-Brigid (Fig. 1), corresponding to the three phases of the workflow: available workflow components, workflow design and workflow execution (Figure 2):

- **Level 1:** *Model*. To create new workflow elements (nodes) and data connections between node ports (points of input/output data from each node), the concept of

flavor (taken from Java) is implemented. In this way, a general behavior is defined for every cell of the same flavor: legal connections, icons, constants, tooltips, etc. Flavors are related hierarchically, as shown on top of Figure 2.

- **Level 2:** *View*. While designing the workflow, several cells of the same flavor can be inserted, allowing each one to have different configuration parameters. This gives rise to the interpreter concept. The interpreter has a 1:1 relationship with visual elements in the diagram, which make up the workflow in the Biomniverso architecture, as shown in the Figure 2 middle layer.
- **Level 3:** *Controller*. These interpreters can be employed to run processes. If a workflow is reused many times will be required several executions for the same cell. This leads to a new concept: the thread. A thread starts when a workflow cell is reached, but if the flow never reaches such a cell, its thread is not launched. Sample live threads are illustrated at the bottom layer of Figure 2.

The pattern Model-View-Controller may be seen from another perspective, where the flavors correspond to the Model, the nodes become the View, and the interpreters are interacting Controllers.

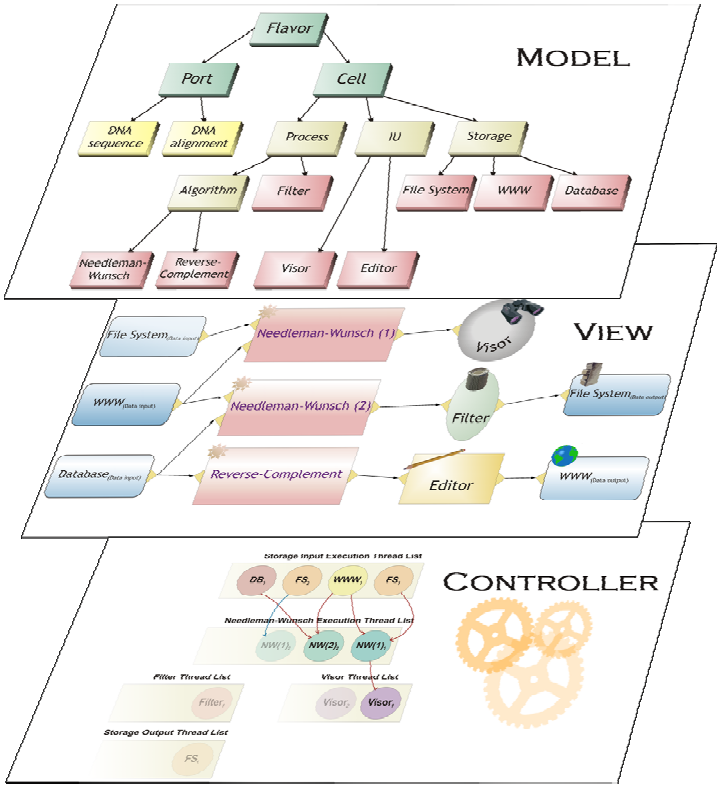

Fig. 2. The three levels of abstraction in Biomniverso architecture. The *Model-View-Controller* design pattern is shown, so each layer corresponds to a different workflow phase.

4 Omega-Brigid: Implementation of Biomniverso

The Omega and Brigid components are designed with an object-oriented programming language, because that suits best the plugin implementation (from a developer point of view) and the workflow modeling (from a user point of view). The best option among them is Java, for being one of the most used languages, open source, easily extensible, and with many libraries available, allowing the deployment of online applications. Another strategic advantage of Java is its universality, being operating system- and hardware-independent. Thus, can be overcome one of the main drawbacks of bio-informatics development and deployment, i.e. the incompatibility amongst platforms.

The Omega tool is a kernel with two main functionalities: 1) run/pause/stop workflow executions, and 2) integrate plugins to enhance operational capabilities.

The user interface that allows an intuitive user interaction with Omega is Brigid, allowing users to manage plugins and workflow processes, as well as visually designing workflows and running them. This effectively overcomes other of the current bioinformatics handicaps, allowing even non-programmers like biologists to use and customize the Omega-Brigid tool without code writing.

4.1 Omega Kernel and Plugin Management

The Omega kernel provides core functionalities in order to manage workflow processing: run, pause, continue, validation, control of threads, etc. In order to construct workflow diagrams, several plugins may be connected to Omega [10]. Omega controls such plugin management as well, providing standard mechanisms to add new workflow elements and to define default behaviors for the simplest tasks.

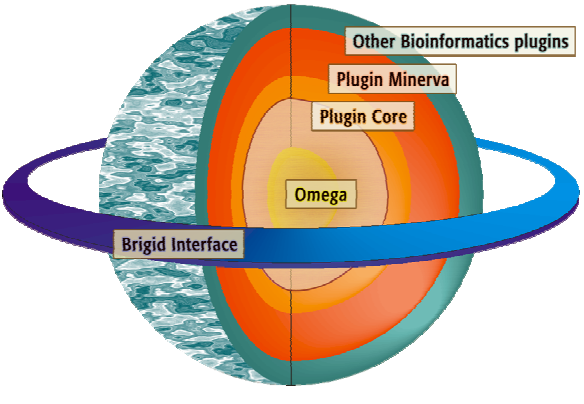

Fig. 3. The Biomniverso implementation allows adding new features and functionalities to application by means of plugins. In bioinformatics, our main plugin is named *Core*, defining the basis of bioinformatics workflow elements processing. High-level plugins are built above it, for more specific tasks, like the *Minerva* plugin. Furthermore, the Omega-Brigid is a general-purpose application, so its workflow concept can be extended to any other fields like network packet processing or queue simulation, implementing a Core non-dependant plugin. Every plugin is accessed by a common GUI, i.e. Brigid.

Every new process (cell) or data format (cell port) to be used in a workflow must follow the Omega directives, inheriting properties and behaviors defined in ancestor components of a hierarchy. For example, following this may be inferred relationships among different implementations of the Needleman-Wunsch algorithm [11].

For bioinformatics purposes, the Core plugin provides basic concepts to build workflows oriented to this field. It is based on BioJava [12] and supplies services and data types (flavors) represented by cells and arrows in the workflow diagram. In this sense, Core illustrates all the basic abilities of the system: read, process, write and view bioinformatics data, local and remote, public or user/password restricted, etc.

Many other plugins may be loaded into Omega. Any of them may use the resources supplied by a previous one, thus effectively providing high order services. The Minerva plugin developed by our research group is an example of this kind of plugin. Its main goal is to find and detect Single Nucleotide Polymorphisms (SNP) amongst a set a genomic data corresponding to different varieties, cultivars, breeds or strains. For instance, we have applied such a tool for quality control and to detect fraudulent olive oil [13]. Figure 3 deploys this abstraction.

4.2 Brigid Interface

We have taken special care to design the Brigid interface (named after the Celtic goddess of Unity), being one of the goals of this research and development work, for the reasons outlined before. We want to stress that this is a key element when comparing Omega-Brigid to other available tools, like the Taverna Workbench [14], the Kepler System [15], or Cyrille2 [16] and Scitegic Pipeline Pilot. Whereas such other tools focus mainly on functionality, the Brigid interface has been designed also with a high usability in mind, so that even biologists with no programming skills can use and customize the Omega-Brigid tool. As an example, the number of components displayed to the user at a particular time has been minimized. Likewise, all the information provided by Omega is displayed with graphics that have an intrinsic mean: red crosses, OK marks, gears of works, etc. A briefly comparison between Omega-Brigid and other tools may be found in the web site of Omega-Brigid.

This Brigid interface has a simple set of options that are enough for novel users, but at the same time is as powerful and flexible enough as to allow custom-tailored enrichment, including new plugins. Thus, the user can customize the set of cells to be used in a workflow.

A plugin may have many cells, being functionally grouped into tab panels: input, output, viewers, processing, searching, etc. A workflow is built by inserting cells into a blank page and connecting them by means of arrows, which represent the flow of data. The path of a data flow is made of input, processing and output, as shown in Figure 1.

Plugins may be obtained from the Internet by means of the Brigid configuration. Even more importantly, such plugins may contain processing cells that can use Internet services to achieve their goals. This situation is illustrated in Figure 4, where the right window shows a web form to invocate the SeqTrim remote algorithm [17]; a cell that allows using this form transparently by means of Brigid has been included in Core plugin. All the options of the SeqTrim algorithm have been reproduced in the configuration panel of the SeqTrim cell.

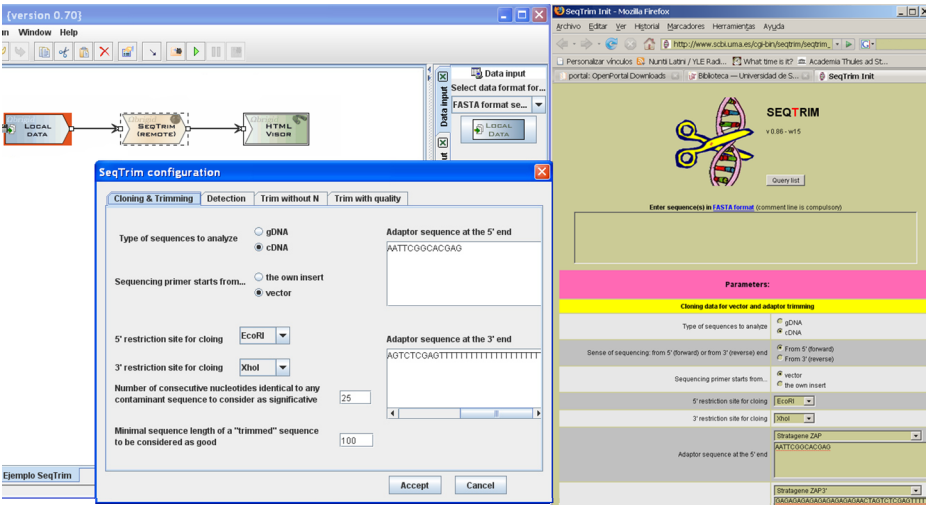

Fig. 4. The Brigid interface. This example shows the *SeqTrim web application* (*right*), and *Omega-Brigid SeqTrim component* to remotely access the service (*left*).

5 Conclusions and Further Work

Omega-Brigid is Java based and thus platform independent. Since it is executed by means of Java Web Start, no installation is needed and it runs on Mac OS X, Linux, Windows, Solaris, etc. Another advantage of JWS is that the latest version of the application is used when an Internet connection is available. The web site <www.sicuma.uma.es/omega> has documentation for developers and users, and allows launching Omega-Brigid through JWS. Thus, Internet plays a central role to take full advantage of Omega-Brigid potential. Additionally, its specially designed GUI allows biologists and other non-programmers to use and even customize and enhance this tool without code writing.

To further enhance the described Omega-Brigid tools, we are currently developing the following features:

- Encapsulate a piece of workflow into a cell, so that it can be reused transparently.
- Allow pipelined data processing.
- Allow users to discard some uninteresting cells when including a new plugin.
- Allow alternative paths in the workflow, with a special emphasis when a remote resource is not available during a limited time.
- Parallelize and implement algorithms to exploit new multicore processors.

Acknowledgments

Supported by grants AGL2006-12550-C02-01/02 of Ministerio de Educación y Ciencia, Project 041/C/2007 and PAI Group AGR-248 of Junta de Andalucía (Spain).

References

1. Dorado, G., Falgueras, J., Claros, M.G., Gálvez, S., Hernández, P.: Bioinformatics: from command-line to GUI and multithreading. In: EU Science Forum, Heidelberg, Germany (2006)
2. Dorado, G., Vásquez, V., Rey, I., Luque, F., Jiménez, I., Morales, A., Gálvez, M., Sáiz, J., Sánchez, A., Hernández, P.: Sequencing ancient and modern genomes (Review). Archaeobios 2, 75–80 (2008)
3. Emmerich, W., Wolf, A.L. (eds.): CD 2004. LNCS, vol. 3083. Springer, Heidelberg (2004)
4. Larkin, M.A., Blackshields, G., Brown, N.P., Chenna, R., et al.: Clustal W and Clustal X version 2.0. Bioinformatics 23, 2947–2948 (2006)
5. Kumar, S., Dudley, J.: Bioinformatics software for biologists in the genomics era. Bioinformatics 23, 1713–1717 (2007)
6. Bridget, S., David, G.: A compositional formalization of connector wrappers. In: Proceedings of the 25th Int. Conf. on Software Engineering. EEE Computer Society Technical Council on Software Engineering, Portland, Oregon, pp. 374–384 (2003)
7. Lopes, A., Wermelinger, M., Fiadeiro, J.: A Compositional Approach to Connector Construction. In: Cerioli, M., Reggio, G. (eds.) WADT 2001 and CoFI WG Meeting 2001. LNCS, vol. 2267, pp. 201–220. Springer, Heidelberg (2002)
8. Bridget Carragher, C.S.P.F.J.S.: Software Tools for Macromolecular Microscopy. Journal of Structural Biology 157, 1–2 (2007)
9. Díaz, D., Dorado, G., Hernández, P., Castillo, A., Claros, G., Falgueras, J., Gálvez, S.: Bioinformatics Approaches for Olive Oil Quality Control. In: Plant Genomics European Meetings (Plant GEM 6). P 06.9. Tenerife, Spain (2007)
10. Cervantes, H., Charleston-Villalobos, S.: Using a workflow engine in a plugin-based product line architecture. In: Gorton, I., Heineman, G.T., Crnković, I., Schmidt, H.W., Stafford, J.A., Szyperski, C., Wallnau, K. (eds.) CBSE 2006. LNCS, vol. 4063, pp. 198–205. Springer, Heidelberg (2006)
11. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48(3), 443–453 (1970)
12. Pocock, M., et al.: BioJava: Open Source Components for Bioinformatics. ACM SIGBIO Newsletter 20, 10–12 (2000)
13. Hernández, P., de la Rosa, R., Rallo, L., Martín, A., Dorado, G.: First evidence of a retrotransposon-like element in olive (*Olea europaea*): implications in plant variety identification by SCAR-marker development. In: TAG Theoretical and Applied Genetics, vol. 102, pp. 1082–1087. Springer, Heidelberg (2001)
14. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20, 3045–3054 (2004)
15. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B., Mock, S.: Kepler: An Extensible System for Design and Execution of Scientific Workflows. In: 16th International Conf. on Scientific and Statistical Database Management, Proceedings, pp. 423–424 (2004)
16. Fiers, M., van der Burgt, A., Datema, E., de Groot, J., van Ham, R.: High-throughput bioinformatics with the Cyrille2 pipeline system. BMC Bioinformatics 9, 96 (2008)
17. Falgueras, J., Lara, A., Cantón, F., Pérez-Trabado, G., Claros, G.: SeqTrim - A Validation and Trimming Tool for All Purpose Sequence Reads. In: Advances in Soft Computing, vol. 44, pp. 353–360. Springer, Heidelberg (2008)