BIOINFORMATION
Discovery at the interface of physical and biological sciences

open access www.bioinformation.net Software

Volume 9(16)

AutoAssemblyD: a graphical user interface system for several genome assemblers

Adonney Allan de Oliveira Veras[1], Pablo Henrique Caracciolo Gomes de Sá[1], Vasco Azevedo[2], Artur Silva[1] & Rommel Thiago Jucá Ramos[1]*

[1]Institute of Biological Sciences, Federal University Pará, Belém, Pará, Brazil; [2]Institute of Biological Sciences, Federal University Minas Gerais, Belo Horizonte, Minas Gerais, Brazil; Rommel Thiago Jucá Ramos – Email: rommelthiago@gmail.com; *Corresponding author

Received August 27, 2013; Accepted August 28, 2013; Published September 23, 2013

Abstract:
Next-generation sequencing technologies have increased the amount of biological data generated. Thus, bioinformatics has become important because new methods and algorithms are necessary to manipulate and process such data. However, certain challenges have emerged, such as genome assembly using short reads and high-throughput platforms. In this context, several algorithms have been developed, such as Velvet, Abyss, Euler-SR, Mira, Edna, Maq, SHRiMP, Newbler, ALLPATHS, Bowtie and BWA. However, most such assemblers do not have a graphical interface, which makes their use difficult for users without computing experience given the complexity of the assembler syntax. Thus, to make the operation of such assemblers accessible to users without a computing background, we developed AutoAssemblyD, which is a graphical tool for genome assembly submission and remote management by multiple assemblers through XML templates.

Availability: AssemblyD is freely available at https://sourceforge.net/projects/autoassemblyd. It requires Sun jdk 6 or higher.

Keywords: Next-generation sequencing, Genome Assembly, Bioinformatics.

Background:
Next-generation sequencing technologies (NGS) revolutionized biology, as they reduced costs and increased the speed of genome sequencing, consequently generating large amounts of data compared with the Sanger method **[1].**

However, challenges have emerged, such as processing the large amounts of data generated by such platforms, especially without a robust computer infrastructure, and conducting genome assembly from short reads, which requires multiple software programs and parameters for optimization **[2-4].** The primary assemblers include Velvet, ABySS, AllPaths, SOAPdenovo and MAQ **[5].**

Most such assemblers operate through lengthy command lines composed of one or several parameters that influence the assembly results, which can be difficult for users with little computing experience **[4-6].**

Thus, a graphical interface facilitates algorithm use, such as in the VAGUE software **[7],** which uses a graphical interface for Velvet assembler operation that allows the user to set the parameter values necessary for operation. However, given the wide variety of tools available, such data analyses are limited where a graphical interface is only available for the Velvet assembler **[5].** Therefore, this study presents **AutoAssemblyD**, which is a graphical tool used for submitting and managing genome assembly from different assemblers through XML templates. This tool also facilitates assembly on remote devices through distributed programming.

Methodology:
Programming Language
The Java programming language was used for AutoAssemblyD development. Remote method invocation (RMI) and Socket are used in the AutoAssemblyD application and were implemented through the java.net and java.io package **[8].**

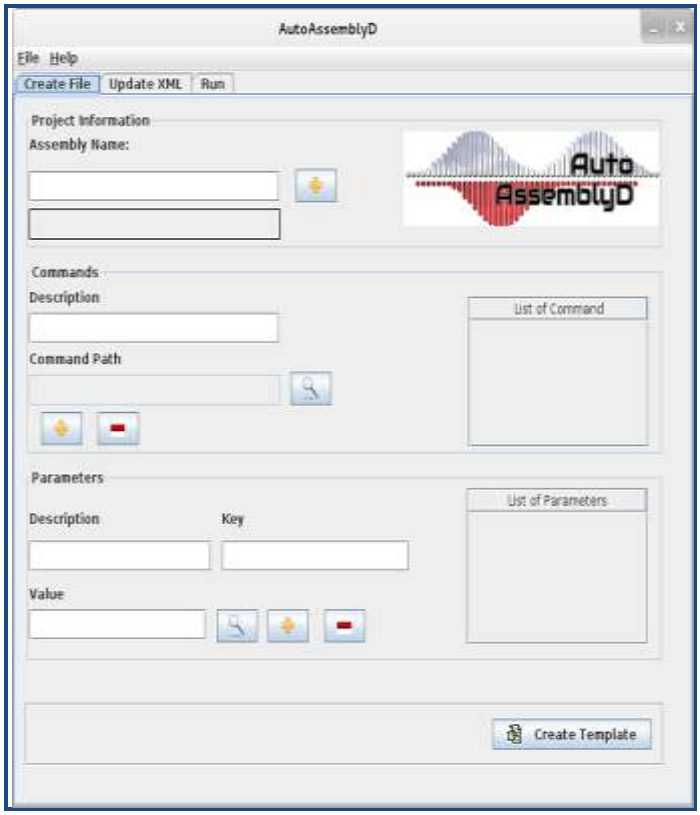

Figure 1: Standard window for template creation.

The Template

Standard Extensible Markup Language (XML) was used to generate the templates as well as identify the assembler, parameters and respective values in accordance with the user-defined configuration.

Remote Management

RMI was used to manage the remote assembly; the clients establish communication with the server upon initiation, which can be selected to perform the assembly process. Communication between the clients and server is tested every 3 seconds, and the server application shows the connection status to the user. The Socket application is used for file transfer between the clients and server; this application initiates upon client and server execution.

AutoAssemblyD Modules

AutoAssemblyD is composed of two modules: local and remote. The local module can be used when the computer has the necessary resources (memory, disk space and CPU) to process the data. The remote module facilitates file transfer and remote processing; thereafter, the results are transmitted to the server.

AutoAssemblyD

AutoassemblyD comprises three interfaces: template creation, template update and assembly operation. The assembler commands, parameters and their respective values are input at the template creation interface **(Figure 1).**

After the assembler informations are defined, the user selects the option *creates file* to generate the XML template used for assembly operation. The template is updated through the *UpdateXML* window that facilitates template reuse in future operations with other input data and parameters, which increases its flexibility. To operate the assembly, the template file must be loaded, and the client that performs the process must be selected. The user guide is available at: https:// sourceforge.net/projects/ autoassemblyd.

Acknowledgement:

This work was part of the Genomics and Proteomics Network of Pará (Rede Paraense de Genômica e Proteômica - RPGP) supported by the Research Foundation of the State of Pará (Fundação de Amparo a Pesquisa do Estado do Pará - FAPESPA), National Council for Scientific and Technological Development (Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq) and Pronex Amazon Center of Excellence in Genomics of Microorganisms (Núcleo Amazônico de Excelência em Genômica de Microorganismos). Funding: FAPESPA, CNPq and Coordination for the Improvement of Higher Education Personnel (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES).

References:

[1] Henson J *et al. Pharmacogenomics*. 2012 **13**: 901 [PMID: 22676195]
[2] Vezzi F *et al. Plos One*. 2012 **7**:e31002 [PMID: 22319599]
[3] Cerdeira LT *et al. J Microbiol Methods*. 2011 **86**: 218 [PMID: 21620904]
[4] Ramos RTJ *et al. Bioinformation*. 2012 **8**: 996 [PMID: 23275695]
[5] Earl D *et al. Genome Res*. 2011 **21**: 2224 [PMID: 21926179]
[6] Powell DR & Seemann T, *Bioinformatics*. 2012 **29**: 264 [PMID: 23162059]
[7] Ramos RTJ *et al. Microb Biotechnol*. 2012 **6**: 150 [PMID: 23199210].
[8] http://www.java.com

Edited by P Kangueane
Citation: **Veras *et al.*** Bioinformation 9(16): 840-841 (2013)