BIOINFORMATICS ORIGINAL PAPER
Vol. 21 no. 12 2005, pages 2812–2813
doi:10.1093/bioinformatics/bti424

Genome analysis

EGene: a configurable pipeline generation system for automated sequence analysis

Alan M. Durham[1,*], André Y. Kashiwabara[1], Fernando T. G. Matsunaga[2], Paulo H. Ahagon[2], Flávia Rainone[1], Leonardo Varuzza[2] and Arthur Gruber[2,*]

[1]Depto. de Ciências da Computação, Instituto de Matemática e Estatística and [2]Depto. de Patologia, Faculdade de Medicina Veterinária e Zootecnia, Universidade de São Paulo, São Paulo SP, 05508-900, Brazil

Received on December 19, 2004; revised on February 17, 2005; accepted on March 31, 2005
Advance Access publication April 6, 2005

ABSTRACT

Summary: EGene is a generic, flexible and modular pipeline generation system that makes pipeline construction a modular job. EGene allows for third-party programs to be used and integrated according to the needs of distinct projects and without any previous programming or formal language experience being required. EGene comes with CoEd, a visual tool to facilitate pipeline construction and documentation. A series of components to build pipelines for sequence processing is provided.

Availability: http://www.lbm.fmvz.usp.br/egene/

Contact: alan@ime.usp.br; argruber@usp.br

Supplementary information: http://www.lbm.fmvz.usp.br/egene/

INTRODUCTION

Large scale DNA sequencing became a worldwide-used methodology. In both genome and EST projects, sequence reads must be processed in a multistep protocol, an integrated chain of several interconnected programs. The usual solution is to devise a script, denominated 'pipeline', containing code to perform all the tasks in the desired order. However, due to the plethora of input/output formats and conventions, it is very hard to directly perform this association. Many individual pipelines have been reported, of which we can mention the works of Ayoubi *et al*. (2002), Chou and Holmes (2001) and Paquola *et al*. (2003). In this paper, we describe EGene, an integrated and customizable pipeline generation system that enables the quick implementation of pipelines. EGene is extensible, since it provides a simple standard to component creation that is easy to understand even to relatively novice programmers.

SYSTEM ARCHITECTURE

Pipelines are generally represented as a series of interconnected 'boxes' with the connections denoting information flow. Therefore, if we define a set of 'box types', each representing a specific processing task, and if we standardize the input/output format, one can develop a system where pipelines are described graphically, in a notation familiar to the biologist. If no restriction is imposed on the interconnections, we call these systems 'workflows'. EGene does not implement generic workflows, only pipelines. In a pipeline, there is a single information flow, represented as a path. The objects are processed one at a time by each of the pipeline steps. Once the first step of the pipeline has been performed in the first object, the resulting object is passed to the next step, and the first step can start being performed by the second object. In spite of being more restricted, pipelines have the advantage of having an embedded sense of fine grain parallelism.

Currently, EGene supports two formats: XML and database. By offering an XML alternative, EGene can be easily installed in simple workstations, with no need for database management skills. This makes the system suitable for big projects as well as for individual researchers. The key element for ensuring abstraction of representation is the component SequenceObject.pm, a Perl module that encapsulates all functionality related to sequence manipulation. Using this module, we can write components that are completely independent of the final representation of the sequence (XML or database). Also, writing new components for the framework is much easier, since all sequence-related operations are abstracted from the code.

In our system, a pipeline is described by creating a configuration file that specifies all the processing steps. There is a special Perl program that reads the configuration files and starts each processing step. EGene's pipelines can be executed in concurrent mode, where a Unix process is associated with each step and communication between steps is performed through Unix pipes. In a multiprocessor machine, Unix itself can perform load balancing and parallel execution. The pipeline execution program can be embedded in automated sequence submission scripts. EGene's configuration files are easily read by non-skilled end users and can be created and edited with standard text editors, or through a graphical tool, CoEd. CoEd is a Java visual configuration editor (Fig. 1) that facilitates building pipelines as it describes all mandatory and optional arguments for components, as well as default values. Pipelines can be run directly from CoEd or, alternatively, the user can save the configuration in EGene format and call manually the pipeline execution program. In addition, CoEd itself can be customized for different problem domains, each corresponding to a specific set of components. Finally, EGene helps the user to have a cleaner environment. Independently of the third-party software used by the components, it is up to the user to decide which data should be left in the host computer, where input data should be fetched and where output data should be stored.

*To whom correspondence should be addressed.

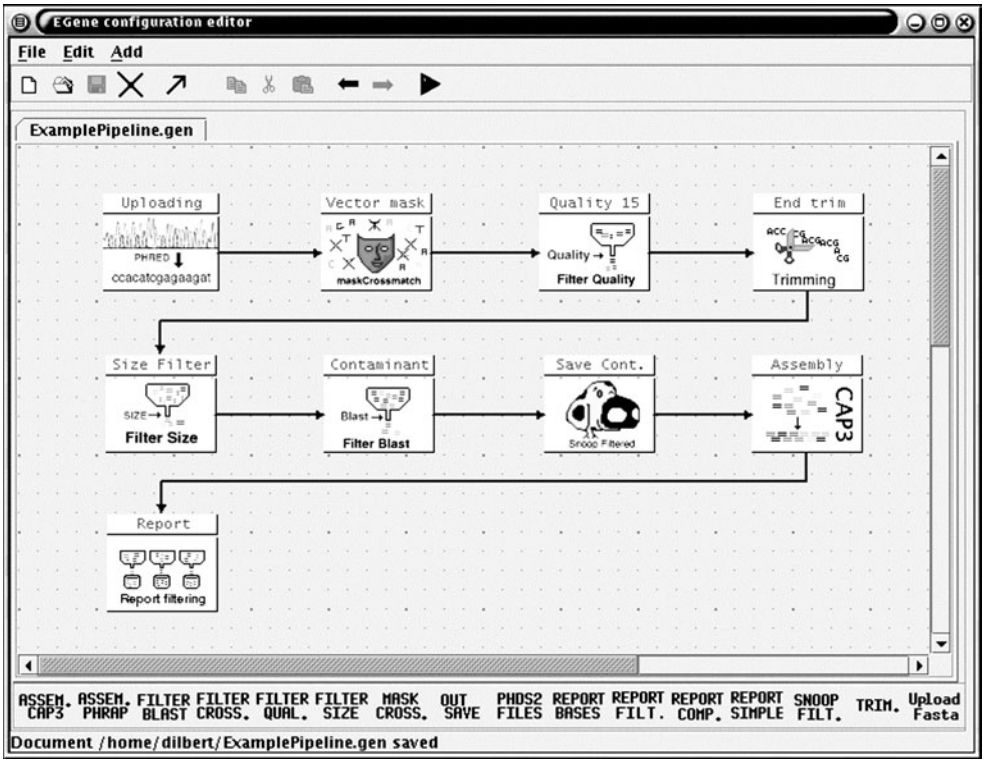

Fig. 1. Screenshot of CoEd, EGene's graphical configuration editor. The icons on the canvas represent the different nodes of the pipeline and the arrows represent the data flow.

IMPLEMENTATION

A complete list of the several components that have been developed for EGene so far, targeting pre-annotation sequence analysis, is provided in the Supplementary Material. EGene was written in Perl and is designed to run on Unix/Linux operating systems. CoEd was written in Java. Source codes, installation instructions, tutorials, documentation and some example datasets are available at EGene's website.

DISCUSSION

EGene has been used in our own laboratory for more than 48 000 reads of the *Eimeria* ORESTES Project (Shirley *et al*., 2004), as well as in several other laboratories, including an EST sequencing project of *Plasmodium vivax*.

There are two other systems with approaches similar to EGene: Pegasys (Shah *et al*., 2004) and Biopipe (Hoon *et al*., 2003). Both systems describe generic workflows, not pipelines. However, this greater flexibility comes with a price. First, Biopipe and Pegasys present a much coarser-grained model for parallelism, attained at the dataset level; that is, there is no parallelism if there are no branches in the workflow. Also, parallelism is achieved only with the use of third-party load management software. Second, with a richer data model, components are harder to write and to use, since it involves a full understanding of the data type conventions. Third, for both systems the development of new pipeline components involves much more computer science expertise. Finally, both Pegasys and BioPipe imply the use of a complex database to record and maintain both the configuration and the execution status of a workflow.

The next step in the development of EGene will be implementing support for sequence annotation. We are currently enriching EGene's data model to incorporate annotation information into the representation of the sequences. We also plan to extend EGene to accept sequence-based workflows that can be represented by a tree, maintaining pipeline-style parallelism. We expect and encourage other programmers to contribute with new specific components for this Open Source platform.

ACKNOWLEDGEMENTS

This work was supported by FAPESP, CNPq and the Ludwig Institute for Cancer Research, São Paulo, Brazil. We would like to thank Dr. Hernando del Portillo for helping to validate the pipeline in his laboratory.

REFERENCES

Ayoubi,P. *et al*. (2002) PipeOnline 2.0: automated EST processing and functional data sorting. *Nucleic Acids Res.*, **30**, 4761–4769.

Chou,H-H, and Holmes,M.H. (2001) DNA sequence quality trimming and vector removal. *Bioinformatics*, **17**, 1093–1104.

Hoon,S. *et al*. (2003) Biopipe: a flexible framework for protocol-based bioinformatics analysis. *Genome Res.*, **13**, 1904–1915.

Paquola,A.C. *et al*. (2003) ESTWeb: bioinformatics services for EST sequencing projects. *Bioinformatics*, **19**, 1587–1588.

Shah,S.P. *et al*. (2004) Pegasys: software for executing and integrating analyses of biological sequences. *BMC Bioinform.*, **5**, 40.

Shirley,M.W. *et al*. (2004) The *Eimeria* genome projects: a sequence of events. *Trends Parasitol.*, **20**, 199–201.