

G-Language Genome Analysis Environment

Kazuharu Arakawa^{1,2}
t98901ka@sfc.keio.ac.jp

Koya Mori^{1,3}
s98982km@sfc.keio.ac.jp

Masaru Tomita^{1,2}
mt@sfc.keio.ac.jp

¹ Institute for Advanced Biosciences, Keio University, Tsuruoka 997-0035, Japan

² Department of Environmental Information, Keio University, Fujisawa 252-8520, Japan

³ Department of Policy Management, Keio University, Fujisawa 252-8520, Japan

Keywords: analysis software, development environment, computational analysis, bioinformatics

1 Introduction

The short but grand history of bioinformatics has clarified the fact that it must gain higher efficiency in order to process the huge masses of information that it faces. G-language Project in Institute for Advanced Biosciences aims to solve this task by:

1. Constructing an integrated environment for the development of analysis software.
2. Systematically accumulating existing analysis software, methodologies for analysis and their results.
3. Constructing generic analysis packages that allow users to avoid redundancy in the process of analysis.

We have been developing a generic analysis environment called the G-language Genome Analysis Environment to fulfill above requirements, and distributing the software system at our web site, <http://www.g-language.org/>.

2 Software Architecture

When designing a generic genome analysis environment, the flexibility of the system should be most taken into consideration. G-language Genome Analysis Environment is implemented with three-layer structure to realize this flexibility. The undermost core layer named “Prelude” is responsible for database and file IO, maintaining data in a unique structure with connection protocols open for the upper classes and layers. The middle layer named “Odyssey” is a group of classes incorporating variety of methods of analysis. This layer is covered by an interface layer, from which the data structure of the core layer and the methods of the middle layer can be natively accessed through a single protocol. Because the three layers are responsible for specific functions and are connected with unique protocol, developers and users need not to be aware of the difference of data formats of the databases. By embedding bioperl [3], the Prelude core layer supports most common genome database formats, such as Genbank, Fasta, EMBL, Swiss, SCF, PIR, GCG, Ace. Prelude core layer also supports native access to the R statistics language (<http://www.r-project.org>).

All system is developed using Perl programming language, and the analysis methods of the Odyssey layer are provided as functions of Perl programming language [1, 2]. The classes enclosing these functions are inherited by the interface class, thus the functions are natively provided to the user. This object oriented structure of G-language Genome Analysis Environment realizes an extended Perl programming language suited as a development/analysis environment for bioinformaticians.

There is also a graphical user interface for G-language Genome Analysis Environment developed with GTK+, enabling the user to easily manipulate the system and analyze complete genome databases. In the graphical user interface, methods from the middle layer are organized in order to

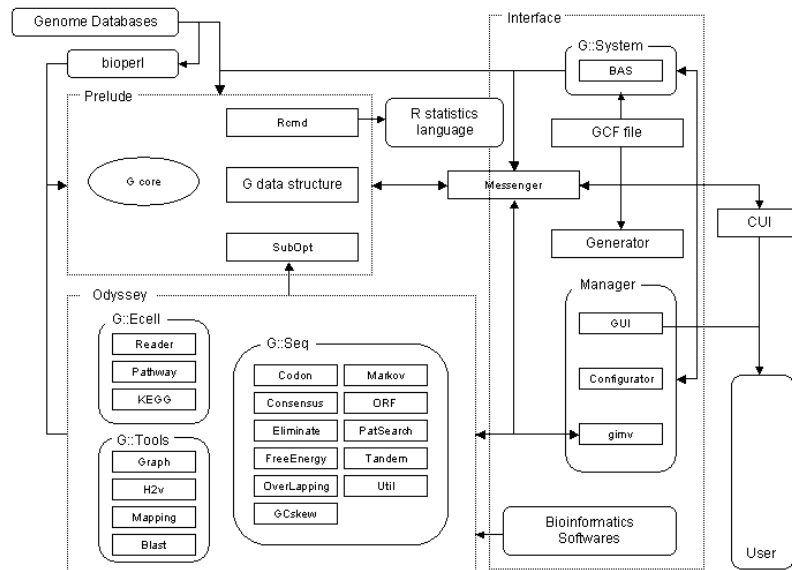


Figure 1: G-Language genome analysis environment architecture.

systemize certain analyses. For instance, Bacteria Analysis System implemented with current releases of G-language Genome Analysis Environment provides a package of analyses of bioinformatics for thorough computational analysis of bacterial genomes, through a graphical user interface. The Bacteria Analysis System is configured either by text file or by the graphical user interface. Users can also write their own short Perl scripts to extend the analysis system.

Located in the top layer in the G-language Genome Analysis Environment, the graphical user interface is tied with lower layers with a messaging protocol. The middle analysis layer also has a backward messaging interface to the user interface layer, altering the output format suited for the type of user interface. Therefore, based on this flexibility of the G-language Genome Analysis Environment, it is also possible for a user to create different user interfaces based on the two core layers, or to incorporate the core layer in other extended analysis systems. The analysis system itself can also be extended to implement file and database IO formats as a Plug-in function, making users possible to create access to other common bioinformatics softwares. G-language Genome Analysis Environment is thus a platform for bioinformatics analysis and software development, and this assists the efficiency of researches.

Early release of October 2001 is currently implemented with Bacteria Analysis System and basic functions and analyses for this purpose, but G-language Project is now developing other systems such as the cDNA Analysis System, Human Genome Analysis System, and other systems for more specific areas of bioinformatics.

Acknowledgements

This work is supported by the members of G-language Project. URL: <http://www.g-language.org/>. G-language Project official Web site.

References

- [1] Stein, L., How Perl saved the human genome project, *TPJ*, 1:5–9, 1996.
- [2] Wall, L., Christiansen, T., and Schwartz, R.L., *Programming Perl* (Second Ed.), O'Reilly & Associates, 1996.
- [3] <http://www.bioperl.org/>