ChemRuby:An open source library for integrating bioinformatics and cheminformatics

Nobuya Tanaka tanaka@kuicr.kyoto-u.ac.jp
Susumu Goto goto@kuicr.kyoto-u.ac.jp
Minoru Kanehisa kanehisa@kuicr.kyoto-u.ac.jp

Bioinformatics Center, Institute of Chemical Research, Kyoto University, Gokanosho, Uji, Kyoto 611, Japan

Keywords: ChemRuby, BioRuby, cheminformatics

1 Introduction

Integration of bioinformatics and cheminformatics is the most attractive issue in the modern pharmaceutical industry. The integration is rapidly accomplished by developing chemical databases focusing on the biochemical and biological aspects such as KEGG LIGAND[1, 2]. These databases are expected to decipher mechanisms of biology at a molecular level. Although bioinformatics and cheminformatics are rapidly integrated within databases, their integration techniques have not been established.

We have developed ChemRuby as an open source library that aims to integrate cheminformatics and bioinformatics. Although current ChemRuby is implemented only for cheminformatics, cooperation with BioRuby[9], that is implemented for bioinformatics, allows to process chemical and biological information simultaneously.

2 Design of the system

ChemRuby is a collection of reusable modules implemented with the object oriented language Ruby. To tackle with the issue of highly dynamic nature of cheminformatics, flexible features of Ruby are exploited.

Most of chemical systems that process chemical information have internal models for a molecule, and they try to interpret and normalize every chemical structural information into an average model. So the information that is hardly normalized into their models is simply ignored. This design is sufficient for converting a chemical file format into other formats.

Because one of the aims of ChemRuby is to extract biologically important descriptors from chemical structure, such information loss can lead to a serious problem. To prevent the loss, ChemRuby adopts two steps to interpret chemical information: reading chemical information without normalizing and subsequent specific interpretation. Since processes for interpretation of chemical information differ according to user's purpose, ChemRuby provide several appropriate modules. This mechanisms are implemented with the Ruby's Mix-in feature.

Currently, ChemRuby is divided into six parts.

1. Parsers for major chemical file formats (kegg chemical function file[7], MDL[12] molfile, sdfile and rxnfile, ChemDraw[11] cdx and cdxml Gaussian98[8], Mopac[5], Protein Data Bank file format[13])

2. Visualizers for chemical structure (Encapsulated PostScript, POV-Ray, Macromedia Flash and Scalable Vector Formats)

3. Wrappers for computational chemistry systems (Tinker[10]:minimize, protein, and nucleic)
4. Accessors for chemical structures (searching, generating and transforming chemical structures)
5. Evaluators for chemical structures (calculating molecular properties and descriptors[3, 4, 6])
6. Unit tests for above modules.

3 Conclusion

ChemRuby includes useful modules to extract chemical descriptors and to manipulate chemical structures. By collaborating with BioRuby, chemical and biological information are closely processable. Although ChemRuby is available for purposes to automate routine work, it would be also useful for rapid prototyping because of Ruby's agile nature.

References

[1] Goto, S., Nishioka, T., and Kanehisa, M., LIGAND: Chemical Database for Enzyme Reactions *Bioinformatics*, 14:591–599, 1998.

[2] Goto, S., Okuno, Y., Hattori, M., Nishioka, T., and Kanehisa, M., LIGAND: database of chemical compounds and reactions in biological pathways, *Nucleic Acids Research*, 30:402–404, 2002.

[3] Graham L. Patrick, *An Introduction to Medicinal Chemistry*, Oxford University Press, 2001.

[4] James Devillers and Alexandru T. Balaban, *Topological Indices and Related Descriptors in QSAR and QSPR*, Gordon and Breach Science Publishers, Amsterdam, The Netherlands, 1999.

[5] J. J. P. Stewart, *MOPAC 2002*, Fujitsu Limited, Tokyo, Japan, 1999.

[6] Mati Karelson, Victor S. Lobanov and Alan R. Katritzky, Quantum-Chemical Descriptors in QSAR/QSPR Studies, *Chem. Rev.*, 96(3):1027–1043, 1996.

[7] Minoru Kanehisa, Susumu Goto, Shuichi Kawashima, Yasushi Okuno, and Masahiro Hattori The KEGG resource for deciphering the genome, *Nucleic Acids Research*, 32:277–280, 2004.

[8] M. J. Frisch, et al. *Gaussian 98 (Revision A.7)*, Gaussian, Inc., Pittsburgh PA, 1998.

[9] Toshiaki Katayama, Shuichi Kawashima, Naohisa Goto, Mitsuteru C. Nakao, Yoshinori K. Okuji and Minoru Kanehisa, BioRuby: Object Oriented Open Source Library for Bioinformatics *Genome Informatics*, 13:248–249, 2002.

[10] `http://dasher.wustl.edu/tinker/`

[11] `http://www.cambridgesoft.com/`

[12] `http://www.mdli.com/`

[13] `http://www.rcsb.org/pdb/`