BMC Bioinformatics

Software

libcov: A C++ bioinformatic library to manipulate protein structures, sequence alignments and phylogeny

Davin Butt[1], Andrew J Roger[2,3] and Christian Blouin*[1,2,3]

Address: [1]Faculty of Computer Science, Dalhousie University, 6050 University Ave. Halifax, NS, B3H 1W5, Canada, [2]Dept. of Biochemistry and Molecular Biology, Dalhousie University, Tupper Medical Building, Halifax, NS, B3H 1X5, Canada and [3]Canadian Institute for Advanced Research (CIAR)

Email: Davin Butt - davin@cs.dal.ca; Andrew J Roger - andrew.roger@dal.ca; Christian Blouin* - cblouin@cs.dal.ca

* Corresponding author

Published: 06 June 2005

BMC Bioinformatics 2005, **6**:138 doi:10.1186/1471-2105-6-138

Received: 05 November 2004
Accepted: 06 June 2005

This article is available from: http://www.biomedcentral.com/1471-2105/6/138

Background: An increasing number of bioinformatics methods are considering the phylogenetic relationships between biological sequences. Implementing new methodologies using the maximum likelihood phylogenetic framework can be a time consuming task.

Results: The bioinformatics library libcov is a collection of C++ classes that provides a high and low-level interface to maximum likelihood phylogenetics, sequence analysis and a data structure for structural biological methods. libcov can be used to compute likelihoods, search tree topologies, estimate site rates, cluster sequences, manipulate tree structures and compare phylogenies for a broad selection of applications.

Conclusion: Using this library, it is possible to rapidly prototype applications that use the sophistication of phylogenetic likelihoods without getting involved in a major software engineering project. libcov is thus a potentially valuable building block to develop in-house methodologies in the field of protein phylogenetics.

Background

With the development of genomics, research in biology and systems biology is becoming increasingly data-driven. The feedback between available data and hypotheses has accelerated the pace at which innovative ideas are generated. Life scientists are in a position to design novel methodologies but do not necessarily have the in-house skills to produce software implementations. Simple methods, made of complex building blocks such as maximum likelihood calculations, require major software development projects before they can be prototyped. The use of libraries can help to rapidly prototype software implementations.

We present libcov, an object-oriented library to perform phylogenetic inference and the manipulation of protein sequences and structures. The library is written in C++, is compliant with the GNU standards and packaged as a dynamic library that can be installed on most Unix distributions (including MacOS X).

There are other bioinformatic libraries available, many of which overlap with libcov in their functionalities. The PAL library, for example, [1] is a Java implementation which offers a versatile object set for nucleotide and protein phylogeny. More generally, interested readers can visit the Open Bioinformatics Foundation [2] that links to a series of libraries written in various popular scripting languages

```
#include <cov/cov.h>
...

try {
        // Read in all trees in file "treeFile"
        vector<covTree*> trees = covIO::ReadTreeFile(treeFile);
        cout << "Analyzing " << trees.size() << " trees..." << endl;


        // Calculate Consensus Tree
        covTree & consensusTree = *Consense::ConsenseTrees(trees, true);

       // Display it to stdout
        cout << "Consensus Tree: ";
        consensusTree.DrawOnTerminal();

        // Read alignment (PHYLIP or FASTA format) in file "alignmentFile"
        covAlignment alignment = covIO::ReadSequenceFile(alignmentFile);

        // Create evolutionary model
        PMatrix model(JTT_MODEL);

        // Calculate ML (no Rate Across Sites (RAS))
        ML_BranchLength mlObj;

        // Calculate the Optimal likelihood
        double likelihood = mlObj.OptimizeBL(consensusTree, model, alignment);
        cout << "Likelihood: " << likelihood << endl;

        // Calcualte with RAS (4 rate categories, alpha = 0.6)
        likelihood = mlObj.OptimizeBL(consensusTree, model, alignment, 0.6, 4);
        cout << "Likelihood with RAS: " << likelihood << endl;


         // Write the Consensus Tree to a file in Newick format
        fstream fout;
        fout.open(outTree.c_str(), fstream::out);
        consensusTree.WriteOutTree(fout);



} catch (covException & e) {
                cerr << e.GetReport() << endl;
                exit(1);
}
```

Figure 1
The likelihood of a consensus tree. In this example, a file containing trees in NEWICK format is parsed and a consensus tree is resolved using the greedy majority-rule consensus algorithm [9, 21]. Finally, the likelihood of the resulting tree is calculated. Bolded lines are libcov API calls.

Table 1: High Level functionalities

Category	Method	Reference
I/O	Tree (NEWICK) Sequences (FASTA, PHYLIP) Protein Structure (PDB)	
Tree manipulation	Random/exhaustive: Subtree Pruning Regrafting (SPR) Tree Bissection Reconnection (TBR) Nearest Neighbor interchange (NNI) Branch Swapping Stepwise addition	[7]
Phylogeny	Neighbor Joining	[8]
	Greedy Majority-rule consensus	[9]
	Maximum Likelihood Rates across site modeling Estimation of shape parameter α	[10]
ML confidence intervals	KH	[11]
	SH	[12]
	RELL	[13]
	Expected Likelihood Weights	[14]
ML performance	P-matrix caching Chebyshev Polynomial approximation	[15]
Substitution matrices	JTT	[16]
	PAM	[17]
	WAG	[18]
Simulation	Protein Sequence Simulation (Rates across sites, rate shifts, site specific frequencies, multiple datasets, likelihood computation)	[5, 19]
	Random Number generation	[20] Acknowl. Z. Yang for implementation in PAML
Structural Biology	Manipulation / mapping Neighboring site anisotropy (NSA) Geometric transformations Distance/Contact Matrices	[6]

such as Perl and Python. Further, there are other libraries available in C++ such as the Bioinformatics Template library BTL [3], and the compBioTool++[4], both of which focus on sequence manipulation.

The scope of libcov is to offer a series of high-level functions that can be invoked in one line of code, and which does not force an implementation to adopt specialized custom types. As for any open source project, it is possible to use or extend the low-level Application Programming Interface (API) to add functionalities or entirely new modules.

Implementation

Libcov offers a high-level programming interface, using an Object-Oriented (OO) approach with classes to represent distinct identities. For example, the class covTree represents a phylogenetic tree, covAlignment is used to store alignments, and class PDBentity handles 3D protein structures from the PDB format. The PDBentity class is a hierarchical structure of peptide chains, residues and atoms. Other classes handle elements such as geometric transformations and substitution matrices.

Libcov is designed as a protein phylogeny library. The data structures and methods that its public interface offers can be integrated within application prototypes with a minimal impact on software design. Most of the return types are Standard Template Library (STL) containers, which can be seamlessly integrated into ongoing software projects. Specialized classes can be derived by consulting the online API documentation. Examples of integration of libcov within C++ source codes are presented in Figure 1.

A summary of the functions offered by libcov is presented in TABLE 1. A more complete list of methods is available at the project's website.

Currently, we have implemented three major applications using libcov. covTREE is our protein sequence simulator that has the ability to simulate complex patterns of protein evolution and phylogenetic artifacts[5]. It uses the Monte Carlo-based simulation functions that libcov provides. covSEARCH is a tree searching program using the maximum likelihood and tree re-arrangement algorithms in libcov. covARES maps sequence and phylogenetic information on to protein models [6]. These applications are also available to the research community under a GNU GPL license.

Conclusion

The libcov library is actively under development, and we will be frequently releasing updated versions. As libcov is the engine powering the phylogenetic application covSEARCH, future work will involve new algorithms of tree searching, confidence interval determination and the integration of structure-based models of substitution.

External contributions are welcomed as the functionality of the library will evolve to match the research interests of the developers of phylogenetics applications.

Availability and Requirements

Project's name: libcov

Project's website: http://www.cs.dal.ca/~cblouin/libcov/

Operating System: GNU C++ library. Tested on Linux, MacOSX and other Unix-based operating systems.

License: GPL

Non-academic licensing: None.

Authors' contributions

C. Blouin – Scientific Functionalities, High-level design, Redaction of manuscript.

D. Butt – Software design, implementation and testing.

A.J. Roger – Scientific functionalities, redaction of manuscript.

Acknowledgements

This work was supported by Genome Atlantic grant on *Prokaryotic genome diversity and evolution*, and by the NSERC Discovery grant 298397-04 (CB). The author would like to thank J. Murdoch for her contribution to the treeSPACE module.

References

1. Drummond A, Strimmer K: **PAL: an object-oriented programming library for molecular evolution and phylogenetics.** *Bioinformatics* 2001, **17:**662-663.
2. O|B|F: **Open Bioinformatics Foundation.** [http://www.open-bio.org].
3. Williams M: **The Bioinformatics Template Library (BTL).** [http://people.cryst.bbk.ac.uk/~classlib/bioinf/BTL99.html].
4. Durbin KJ: **CompBioTools++.** [http://people.cryst.bbk.ac.uk/~classlib/bioinf/BTL99.html].
5. Blouin C, Butt DJ, Roger AJ: **The impact of taxon sampling on the estimation of rates of evolution at sites.** *Mol Biol Evol* 2005, **22:**784-791.
6. Blouin C, Boucher Y, Roger AJ: **Inferring functional constraints and divergence in protein families using 3D mapping of phylogenetic information.** *Nucleic Acids Res* 2003, **31:**790-797.
7. Felsenstein J: **Inferring Phylogenies.** 1st edition. Sunderland, MA, Sinauer Associates, Inc.; 2004:664.
8. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol Biol Evol* 1987, **4:**406-425.
9. Felsenstein J: **PHYLIP (Phylogeny Inference Package) version 3.6.** Seattle, Wa., Distributed by the author, Dept. of Genetics, U. of Washington; 2002.
10. Yang Z: **Among-site rate variation and its impact on phylogenetic analyses.** *Trends Ecol Evol* 1996, **11:**367-372.
11. Kishino H, Hasegawa M: **Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea.** *J Mol Evol* 1989, **29:**170-179.
12. Shimodaira H, Hasegawa M: **Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference.** *Mol Biol Evol* 1999, **16:**1114-1116.
13. Kishino H, Miyata T, Hasegawa M: **Maximum Likelihood inference of protein phylogeny and the origin of chloroplasts.** *J Mol Evol* 1990, **30:**151-160.
14. Strimmer K, Rambaut A: **Inferring confidence sets of possibly misspecified gene trees.** *Proc R Soc Lond B Biol Sci* 2002, **269:**137-142.
15. Pupko T, Graur D: **Fast computation of maximum likelihood trees by numerical approximation of amino acid replacement probabilities.** *Computational Statistics & Data Analysis* 2002, **40:**285-291.
16. Jones DT, Taylor WR, Thornton JM: **The rapid generation of mutation data matrices from protein sequences.** *Comput Appl Biosci* 1992, **8:**275-282.
17. Dayhoff MO, Schwartz RM, Orcutt BC: **A model of evolutionary change in proteins.** In *Atlas of protein sequence and structure Volume 5*. Edited by: Dayhoff MO. Silver Spring, MA, National Biomedical Research Foundation; 1978:345-352.
18. Whelan S, Goldman N: **A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach.** *Mol Biol Evol* 2001, **18:**691-699.
19. Grassly NC, Adachi J, Rambaut A: **PSeq-Gen: an application for the Monte Carlo simulation of protein sequence evolution along phylogenetic trees.** *Comput Appl Biosci* 1997, **13:**559-560.
20. Wichmann BA, Hill ID: **An efficient and portable pseudo-random number generator.** *Appl Stat* 1982, **31:**188-190.
21. Bryant D: **A Classifcation of Consensus Methods for Phylogenetics.** In *BioConsensus* Edited by: Janowitz M, Lapointe FJ, McMorris FR, Mirkin B and Roberts FS. , DIMACS. AMS.; 2003:164-184.