BIOINFORMATICS APPLICATIONS NOTE
Vol. 19 no. 17 2003, pages 2330–2331
DOI: 10.1093/bioinformatics/btg319

# NCL: a C++ class library for interpreting data files in NEXUS format

*Paul O. Lewis*

*Department of Ecology and Evolutionary Biology, University of Connecticut,
75 North Eagleville Road, Unit 3043, Storrs, CT 06269-3043, USA*

Received on March 7, 2003; revised on May 22, 2003; accepted on June 5, 2003

**ABSTRACT**

**Summary:** The NEXUS Class Library (NCL) is a collection of C++ classes designed to simplify interpreting data files written in the NEXUS format used by many computer programs for phylogenetic analyses. The NEXUS format allows different programs to share the same data files, even though none of the programs can interpret all of the data stored therein. Because users are not required to reformat the data file for each program, use of the NEXUS format prevents cut-and-paste errors as well as the proliferation of copies of the original data file. The purpose of making the NCL available is to encourage the use of the NEXUS format by making it relatively easy for programmers to add the ability to interpret NEXUS files in newly developed software.

**Availability:** The NCL is freely available under the GNU General Public License from http://hydrodictyon.eeb.uconn.edu/ncl/

**Contact:** paul.lewis@uconn.edu

**Supplementary information:** Documentation for the NCL (general information and source code documentation) is available in HTML format at http://hydrodictyon.eeb.uconn.edu/ncl/

The NEXUS file format (Maddison *et al.*, 1997) is currently used by a diversity of programs for storing sequences, discrete morphological data, and/or tree descriptions primarily for purposes of phylogenetic analysis. Programs employing the NEXUS format include: COMPONENT (Page, 1993; http://taxonomy.zoology.gla.ac.uk/rod/cpw.html); GeneTree (Page and Charleston, 1997a,b); MacClade (Maddison and Maddison, 2000, http://macclade.org/); MrBayes (Huelsenbeck and Ronquist, 2001, http://morphbank.ebc.uu.se/mrbayes/); NDE (Page, 1998, http://taxonomy.zoology.gla.ac.uk/rod/NDE/nde.html); PAUP* (Swofford, 2002, http://paup.csit.fsu.edu/); r8s (Sanderson, 2003, http://ginger.ucdavis.edu/r8s/) SplitsTree (Huson, 1998, http://www.mathematik.uni-bielefeld.de/~huson/phylogenetics/splitstree.html); and TreeView (Page, 1996, http://taxonomy.zoology.gla.ac.uk/rod/treeview.html). By adopting the NEXUS file format for their programs, programmers save their users from the hazards of making copies of their original data to accommodate the differing file formats of the programs they use. Just as with gene duplication events, divergence is the rule once there is more than one copy of a data file, and one of the primary benefits of the NEXUS format lies in promoting data integrity by reducing the number of times the original data file needs to be copied and modified.

Unfortunately, the NEXUS file format is complex due to its flexibility. This complexity means that few programmers have invested the time needed to write the code for reading and interpreting files in the NEXUS file format. The primary purpose of the NEXUS Class Library (NCL) is to reduce the cost of adopting the NEXUS file format (in terms of person-hours of programming effort). A secondary benefit of the NCL is consistency. Programs that have already adopted the NEXUS format differ in their implementation of the format. Each program produces a different message for the same underlying error in the NEXUS file, and most fail to adhere strictly to the published format. Having a NEXUS class library ensures consistent behavior and error reporting among the programs that incorporate it.

The NCL comprises 15 C++ classes that together confer the ability to interpret most NEXUS data files in existence. NEXUS files are composed of discrete sections known as blocks. The NCL provides classes to encapsulate these blocks, each derived from a common base class, NxsBlock. For example, the NEXUS TAXA, TREES and DISTANCES blocks are fully implemented by the NxsTaxaBlock, NxsTreesBlock and NxsDistancesBlock classes, respectively. The TAXSETS, CHARSETS and EXSETS commands from the ASSUMPTIONS block have been implemented in the NxsAssumptionsBlock class, and most CHARACTERS and DATA blocks can be read by the NxsCharactersBlock and NxsDataBlock classes. Some of the more complicated blocks require helper classes: NxsDistanceDatum, NxsDiscreteDatum, NxsDiscreteMatrix and NxsSetReader are examples of such classes. NxsReader encapsulates the NEXUS file interpreter, orchestrating the process of reading a NEXUS file using the NxsToken class to read individual tokens from the file. Finally, NxsString provides additional functionality over the standard string class, and NxsException objects are thrown when an error is detected while reading a file.

Like most programs that have adopted the NEXUS file format, the NCL does not fully implement the format as described by Maddison *et al*. (1997). For example, some blocks (e.g. the UNALIGNED block) do not appear in many actual data files and, thus, have received low priority. In CHARACTERS and DATA blocks, DATATYPE = CONTINUOUS has not yet been implemented, as well as the less common choices for ITEMS (e.g. MIN, MAX, MEDIAN etc.) and STATESFORMAT (e.g. INDIVIDUALS, COUNT FREQUENCY).

Included with the NCL are three example applications that use it: `nclsimplest` is the example described in detail in the documentation for the library; `ncltest` is the simplest application that nevertheless incorporates all existing capabilities of the library; and `BASICCMDLINE` provides an example of a command-driven application, and also illustrates how a private NEXUS block can be used to allow batch-mode processing.

The distribution (see http://hydrodictyon.eeb.uconn.edu/ncl/) includes: (1) all of the source code for the NCL and example applications; (2) three example NEXUS data files; and (3) complete source code documentation in the form of HTML pages. The standard GNU build system (employing a configure script created by GNU Autoconf and makefiles created by GNU Automake) is included for Linux/Unix builds. The library has been successfully compiled under the following combinations of operating sytem/compiler: Mac OS 10.1.5/gcc 2.95.2, Mac OS 10.2.6/gcc 3.1, Windows XP Professional/Visual C++ 6.0, Red Hat Linux 7.3 x86/gc 2.96, Red Hat 7.0 Alpha/gcc 2.96. The NCL is free software and falls under the GNU General Public License (http://www.gnu.org/licenses/licenses.html).

## ACKNOWLEDGEMENTS

I am deeply indebted to Mark Holder, Kent Holsinger, Rod Page, and David Swofford for their assistance and feedback during the development of the NEXUS Class Library. Thanks also to Rod Page, Fredrick Ronquist and one anonymous reviewer, for their valuable comments on this application note. I also gratefully acknowledge the financial assistance provided by grant 98-4-5 ME from the Alfred P. Sloan Foundation/National Science Foundation.

## REFERENCES

Huelsenbeck,J.P. and Ronquist,F.R. (2001) MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, **17**, 754–755.

Huson,D.H. (1998) SplitsTree: a program for analyzing and visualizing evolutionary data. *Bioinformatics*, **14**, 68–73.

Maddison,D.R., Swofford, D.L. and Maddison,W.P. (1997) NEXUS: an extensible file format for systematic information. *Sys. Biol.*, **46**, 590–621.

Maddison,W. and Maddison,D. (2000) *MacClade, Version 4.0.* Sinauer, Sunderland, Massachusetts.

Page,R.D.M. (1993) *COMPONENT, Version 2.0.* The Natural History Museum, London.

Page,R.D.M. (1996) TREEVIEW: an application to display phylogenetic trees on personal computers. *Comp. Appl. Biosci*., **12**, 357–358.

Page,R.D.M. (1998) NDE: NEXUS data editor for Windows.

Page,R.D.M. and Charleston, M.A. (1997a) From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. *Mol. Phylogenet. Evol.*, **7**, 231–240.

Page,R.D.M. and Charleston,M.A. (1997b) Reconciled trees and incongruent gene and species trees. In Mirkin,B. McMorris,F.R. Roberts,F.S. and Rzhetsky,A. (eds), *Mathematical Hierarchies in Biology*. American Mathematical Society, Providence, Rhode Island, pp. 57–70.

Sanderson,M.J. (2003) r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, **19**, 301–302.

Swofford,D.L. (2002) *PAUP*: Phylogenetic Analysis Using Parsimony (and Other Methods), Version 4.0b10.* Sinauer, Sunderland, Massachusetts.