Casal2
Third-Party Libraries

v2019.12

Author:
Scott Rasmussen
scott@zaita.com

Table of Contents:
Table of Contents

Document History

Version	Description	Author	Date
V2019.12	*Initial draft for version 1.0 of Casal2*	S.Rasmussen	16/12/2019

Casal2 Overview

Casal2 is NIWA's integrated assessment tool for modelling the population dynamics of marine species, including fishery stock assessments. Casal2 expands functionality and increases maintainability over its predecessor, CASAL. Casal2 can be used for quantitative assessments of populations, including fish, invertebrates, marine mammals, and seabirds.

Because Casal2 includes multiple automatic differentiation libraries, the way it incorporates third-party libraries is very specific. This document covers the considerations for including new third-party libraries, and the characteristics of the third-party libraries packaged with Casal2.

New Third-Party Library Considerations

Automatic Differentiation Compatibility

Casal2 has been designed to allow the inclusion of multiple automatic differentiation libraries (http://www.autodiff.org/). However, this feature introduces very strict restrictions on the types and scope of third-party libraries usage.

A third-party library can either:

1. Not contribute in any way to the likelihood function; OR
2. Not be used at all during estimation; OR
3. Allow override of the double type with a custom object; OR
4. The third-party library code can be modified to work with a custom object.

If a library meets any of the above requirements, then it is likely that the library can be used in Casal2. If it does not meet any of the above requirements, then the library cannot be used in Casal2.

As an example, the Boost Matrix library (https://www.boost.org/): this library is used for matrix manipulation and calculations during a model iteration (i.e., classes and methods that contribute to the likelihood score) and it does not allow the overriding of the double type.

So, this library cannot be used without modifying the [Q: Boost? Casal2? Both?] source code to work with a custom object. This restriction adds a significant amount of overhead to the Casal2 project, so it is recommended to not use the Boost library and instead to include a different library that is autodiff compatible, such as [please provide an example library].

Including new automatic differentiation libraries

Request: please describe what requirements need to be met and what classes, methods, and interfaces need to be developed to include and test a new automatic differentiation library, e.g., ensmallen (http://ensmallen.org/), ATL (https://github.com/msupernaw/ATL), or the Stan math library (https://mc-stan.org/users/interfaces/math.html).

Including new libraries for other functionality

Request: please describe what requirements need to be met and what classes, methods, and interfaces need to be developed to include and test libraries for other functionality, e.g., MCMCLib (https://github.com/kthohr/mcmc) or Stan (https://github.com/stan-dev/stan).

Building on Windows

Some third-party libraries have been designed primarily for Linux use, so compiling them on Windows can be challenging. Casal2 primarily uses the GCC compiler chain (https://gcc.gnu.org/), so that the Linux and Windows build differences are minimised.

Where the third-party library comes with build scripts for a Linux environment only, Casal2 will use MSYS (http://www.mingw.org/wiki/msys) as a POSIX-compliant command line for building on Windows. MSYS is preferred over other POSIX-compliant command line environments like Cygwin as MSYS does not impose any dependencies on the binary being built; Cygwin requires linking to a Cygwin DLL.

Each third-party library that requires a Linux-style command line for building is packaged with its own version of MSYS. The /etc/profile file was modified to run the compilation on start-up, then automatically exit. These tasks are monitored in the Casal2 build system for completion.

Performance Consideration and Testing

There are no performance considerations or tests for third-party libraries in Casal2.

The Casal2 package has been designed such that third-party libraries do not typically interact or interfere with each other. The third-party libraries that are using during the execution of a model (e.g., an autodiff minimiser) are all isolated from each other.

When adding new third-party libraries, performance of that library in the wider context of Casal2 is not considered. The restrictions on third-party library usage typically disallow general libraries to be added freely, which prevents this [Q: prevents what?].

Inclusion in Git

Casal2 packages all third-party libraries as part of the source code. This choice is deliberate so that developers can work on the entire Casal2 code base without an Internet connection (e.g., on a boat or plane).

With that in mind, git/GitHub/GitLab does not allow for files in the repository that are 100MB or larger. If the size of the installation or archive file for a library is 100MB or larger, then either the file needs to be split into two or more files, or some components of the library package need to be removed so that the file size is less than 100MB.

Existing Third-Party Libraries

ADOL-C

Link: https://projects.coin-or.org/ADOL-C
License: Eclipse Public Licence (EPL) or GNU General Public Licence v2 (GPL v2)
Version in Casal2: 2.5.2

ADOL-C is an automatic differentiation library that does not include a solver. Casal2 uses a copy of the FMM/Numerical Differences solver from the BetaDiff code to build an ADOL-C-based automatic differentiation minimiser.

ADOL-C does not build cleanly for Windows using MinGW, so the third-party build system includes MSYS for building ADOL-C.

Casal2 has not customised the ADOL-C code.

BetaDiff

Link: None

Licence: Unknown

Version in Casal2: 2012-11-08

BetaDiff is an automatic differentiation minimiser developed by NIWA. It was built on ADOL-C version 1.8.4 (Aug 16th 1999). The ADOL-C code has been heavily modified to provide an interface similar to that for AUTODIFF/ADMB.

Casal2 has customised the BetaDiff source code with overrides for converting from an adouble to a std::stringstream for outputting values in reports, etc.

The code for BetaDiff is no longer being actively maintained by NIWA, so the version in Casal2 is considered the latest. There is no repository for BetaDiff.

Boost

Link: https://www.boost.org
Licence: Boost Software Licence, Version 1.0
Version in Casal2: 1.58

Boost is a collection of C++ libraries together under a single project to provide a wide range of extra functionality to the base C++ language. Many Boost libraries eventually become part of the C++ standard.

Casal2 uses Boost for:

- Reading and parsing the command line parameters,
- String manipulation by using Boost::split() and Boost::join() methods, and
- Random number generation that is consistent across OS/CPU as the standard C++ random number generator is not.

Casal2 previously used Boost functionality for threading, but this was modified when the Boost threading functionality was incorporated into the C++ standard.

Casal2 does not modify the source code for Boost. Casal2 does strip all the documentation/man pages from the Boost archive file to reduce the file size to less than 100MB for inclusion in the repository. In the future, it may be useful to either use the GitHub LFS functionality (https://git-lfs.github.com/) or 7zip to address the repository file size restriction issue.

CppAD

Link: https://coin-or.github.io/CppAD/doc/cppad.htm
License: Eclipse Public License or GNU General Public License v2 (GPL v2)
Version in Casal2: 20141128

CppAD (C++ Algorithmic Differentiation package) is an automatic differentiation library and minimiser (using Ipopt solver, https://coin-or.github.io/Ipopt/, version 3.12.4).

CppAD does not build cleanly for Windows using MinGW, so the third-party build system includes MSYS for compiling CppAD.

Casal2 has not customised the CppAD code. The third-party libraries that CppAD requires have been downloaded and included with the CppAD third-party library. The CppAD build instructions have instructions for downloading these libraries with the included CppAD scripts.

DLib

Link: https://dlib.net/
License: Boost Software License V 1.0
Version in Casal2: 18.16

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. DLib provides minimisation capabilities but is not an automatic differentiation system. DLib is one of the non-autodiff minimisers included with Casal2.

Casal2 has not customised the DLib code.

Google Test and Google Mock

Link: https://github.com/google/googletest
License: BSD 3-Clause "New" or "Revised" License
Version in Casal2: Google Test 1.8.1, Google Mock 1.6.0

The Google Test library is the unit testing library used by Casal2. All unit tests in Casal2 are developed using Google Test.

The Google Mock library is used to create mock objects in the code to make unit testing easier. This allows for simple objects with scripted behaviour to be created which reduces the complexity of unit testing interconnected objects.

The Google libraries are run during unit testing only, so they do not need to be autodiff compatible. There are currently no autodiff unit tests.

Casal2 has not customised the Google Test and Google Mock code.

Parser

Link: http://www.gammon.com.au, https://github.com/nickgammon/parser
License: Unknown Open Source License

Version in Casal2: 1.1

The Gammon Parser is an expression parser than allows the use of user-defined equations in Casal2. It is a header-only library, so it does not need to be compiled as part of the build system.

The Gammon Parser has been heavily modified to allow it to support the use of Casal2 addressables. As the Gammon Parser appears to be abandoned by its owner, we do not expect any new versions to be released so do not have to consider the upgrade path.