Package ‘PBSadmb’

November 20, 2009

Version 0.51

Date 2009-11-20

Title PBS ADMB

Author Jon T. Schnute, Rowan Haigh, Anisa Egeli

Maintainer Jon T. Schnute ⟨Jon.Schnute@dfo-mpo.gc.ca⟩

Depends R (>= 2.7.0), PBSmodelling (>= 2.06)

Description R Support for ADMB (Automatic Differentiation Model Builder)

License GPL (>=2)

R topics documented:

`admb` *Start the PBS ADMB GUI*

Description

Start up the PBS GUI for running ADMB.

Usage

```
admb(prefix="", wdf="admbWin.txt", optfile="ADopts.txt")
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`wdf`	string name of the *window description file* that creates the GUI.
`optfile`	string name of options file (usually in user's working directory).

Author(s)

Rowan Haigh, Pacific Biological Station, Nanaimo BC, Canada

See Also

`makeADopts`

`ADMBcmd` *Database of ADMB Command Scripts*

Description

Command scripts for ADMB's convert, compile, and link routines

Usage

```
data(ADMBcmd)
```

Format

A data frame with the following 8 variables:

`OS` operating system

`Comp` C++ compiler type

`Index` index that indicates convert, compile or link with options: safe or optimize, random effects or normal.

`Step` description of processing step (convert, compile, or link)

`Safe` logical: if `TRUE` use safe mode; if not, use optimise mode.

`RanEff` logical: if `TRUE` use random effects model; if not, use normal model.

`Command` the command suitable for specified combination of `Step`, `Safe`, and `RanEff`

`Comment` comment about the command, if any

Details

This database represents a compilation of ADMB scripts for various operating systems and compilers. A user's project normally starts with a template file, named with a prefix to denote the project and a standard suffix `.tpl`. This file must go through three processing steps: conversion to C/C++ code, compilation by a specified compiler, and linking with ADMB libraries.

The reulting command depends on the operating system, compiler, processing step, and two binary options (safe/optimized; normal/random effects). In principle, the three processing steps and two binary options give 3x2x2=12 possibilities. However, conversion doesn't depend on the "safe/optimized" choice, and compilation doesn't depend on "normal/random effects". This reduction leaves only 8 possibilities, specified by an `index` in the range 1:8.

A variable in a `Command` string is designated by the prefix character `@`. We use this for convenient string substitution by `parseCmd`, the function that translates database strings into actual ADMB commands.

The subdirectoy `.../ADMB/scripts` in the installed package contains an Excel spreadsheet, used as the source file for this database. Currently, our database is incomplete, and we heartily encourage the ADMB community to make contributions for additional operating systems and compilers.

Source

Jon T. Schnute, Pacific Biological Station, Nanaimo BC

See Also

`parseCmd`

`appendLog` *Append Data to Log File*

Description

Append summary information or output to a previously created log file.

Usage

```
appendLog(prefix, lines)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`lines`	data to append to `'prefix'.log`).

Value

No explicit value reurned. Appends data into a log file `'prefix'.log`.

Note

A wrapper function that can be called from a GUI exists as `.win.appendLog`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`startLog`, `editADfile`

`checkADopts` *Check ADMB Options for Link Integrity*

Description

Check that `.ADopts` has all required components and that links point to actual files on the hard drive.

Usage

```
checkADopts(opts=getOptions(.PBSadmb), check=c("admpath","gccpath","editor"),
            warn=TRUE, popup=FALSE)
```

Arguments

`opts`	ADMB options values.
`check`	components of `.ADopts` to check.
`warn`	logical: if `TRUE`, print the results of the check to the R console.
`popup`	logical: if `TRUE`, display program location problems in a popup GUI.

Value

Boolean value where `TRUE` indicates all programs were located in the specified directories and `FALSE` if at least one program cannot be found. The returned Boolean scalar has two attributes:
`warn` - named list of test results, and
`message` - named vector of test results.

Note

A wrapper function that can be called from a GUI exists as `.win.checkADopts`.

Author(s)

Rowan Haigh, Pacific Biological Station, Nanaimo BC, Canada

See Also

`makeADopts`,`readADopts`

cleanAD — *Clean ADMB-Generated Files from the Working Directory*

Description

Detects files in the working directory with the specified `prefix` and removes them all save those with the suffix `.tpl`, `.dat`, and `.pin`.

Usage

```
cleanAD(prefix)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).

Details

Aside from potential garbage files with the specified `prefix`, other files associated with ADMB are detected. Also files `*.tmp` and `*.bak` are displayed. Calling `cleanAD` invokes the hidden function `.cleanUp`, which creates a GUI menu of the potential garbage files. The user can select whichever files s/he wishes for disposal.

Value

Returns nothing. Invokes a GUI menu of potential garbage files.

Note

A wrapper function that can be called from a GUI exists as `.win.cleanAD`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`makeAD`, `runAD`, `readRep`

compAD — *Compile C Code*

Description

Compile C++ code in `'prefix'.cpp` to create a binary object file `'prefix'.o`.

Usage

```
compAD(prefix, raneff=FALSE, safe=TRUE, logfile=TRUE, add=TRUE,
       verbose=TRUE, comp="GCC")
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`raneff`	logical: use the random effects model, otherwise use the normal model (currently does not influence the compile stage, but the argument is preserved here for future development).
`safe`	logical: if `TRUE`, use safe mode with bounds checking on all array objects, otherwise use optimized mode for fastest execution.
`logfile`	logical: if `TRUE`, create a log file of the messages from the shell call.
`add`	logical: if `TRUE`, append shell call messages to an exsiting log file.
`verbose`	logical: if `TRUE`, report the shell call an its messages to the R console.
`comp`	string: compiler to use - "GCC" is only currently supported

Details

This function uses the C++ comiler declared in `.ADopts`. If `logfile=TRUE`, any errors will appear in `'prefix'.log`. If `verbose=TRUE`, they will appear in the R console.

Value

Invisibly returns the shell call and its messages.

Note

A wrapper function that can be called from a GUI exists as `.win.compAD`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`convAD`, `linkAD`, `makeAD`

`convAD` *Convert TPL Code to CPP Code*

Description

Convert code in `'prefix'.tpl` to C++ code in `'prefix'.cpp`.

Usage

```
convAD(prefix, raneff=FALSE, logfile=TRUE, add=FALSE,
       verbose=TRUE, comp="GCC")
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`raneff`	logical: if `TRUE`, use the random effects model executable `tpl2rem.exe`, otherwise use the normal model executable `tpl2cpp.exe`.
`logfile`	logical: if `TRUE`, create a log file of the messages from the shell call.
`add`	logical: if `TRUE`, append shell call messages to an exsiting log file.
`verbose`	logical: if `TRUE`, report the shell call an its messages to the R console.
`comp`	string: compiler to use - "GCC" is only currently supported

Details

This function invokes the ADMB command `tpl2cpp.exe` or `tpl2rem.exe`, if `raneff` is `FALSE` or `TRUE` respectively. If `logfile=TRUE`, any errors will appear in `'prefix'.log`. If `verbose=TRUE`, they will appear in R console.

Value

Invisibly returns the shell call and its messages.

Note

A wrapper function that can be called from a GUI exists as `.win.convAD`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`compAD`, `linkAD`, `makeAD`

`copyFiles` *Copy System Files*

Description

Copy files with specified prefixes and suffixes from one location to another.

Usage

```
copyFiles(prefix, suffix=NULL, dir0=getwd(), dir1=getwd(), ask=TRUE)
```

Arguments

`prefix`	string scalar/vector of potential file prefixes.
`suffix`	string scalar/vector of potential file suffixes.
`dir0`	source directory from which to copy files.
`dir1`	destination directory to copy files to.
`ask`	logical: if `TRUE`, popup boxes will prompt the user for every instance that a file will be overwritten.

Details

This function uses R's `list.files` and `file.copy` functions. The pattern recognition tends not to work when given the wildcard character `*`; however, the user may use this character, and the code will interpret it.

Value

Invisibly returns a Boolean vector with names of files that have been copied or not.

Author(s)

Rowan Haigh, Pacific Biological Station, Nanaimo, BC

See Also

`editAD`

`editAD` *Edit ADMB Files*

Description

Edit files associated with specified prefix and suffixes.

Usage

```
editAD(prefix, suffix=c(".tpl",".cpp",".log"))
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`suffix`	string scalar/vector specifying one or more suffixes.

Value

Invisibly returns Boolean vector with elements `TRUE` if files exist, `FALSE` if they do not.

Note

A wrapper function that can be called from a GUI exists as `.win.editAD`.

This function explicitly uses the editor chosen for PBSadmb. PBSmodelling has another function `openFile` that uses Windows file associations or an application specified with `setPBSext`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`editADfile`, `makeADopts`

`editADfile` *Edit a File*

Description

Edit a file using the text editor specified in `.ADopts`.

Usage

```
editADfile(fname)
```

Arguments

`fname`	string name of file in current working directory (or elsewhere if path delimited by `/` or `\`).

Value

Returns Boolean: `TRUE` if file exists, `FALSE` if it does not.

Note

This function explicitly uses the editor chosen for PBSadmb. PBSmodelling has another function `openFile` that uses Windows file associations or an application specified with `setPBSext`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`editAD`, `makeADopts`

`installADMB` *Install windows admb binary (for gcc)*

Description

Only applicable for Windows: Downloads and installs the windows ADMB binary for gcc. ADMB is installed under PBSadmb's library directory under R.

Usage

```
installADMB()
```

Value

The path where ADMB was installed.

`linkAD` *Link Object Files to Make an Executable*

Description

Links the binary object file `'prefix'.o` to the ADMB libraries and produces the executable file `'prefix'.exe`.

Usage

```
linkAD(prefix, raneff=FALSE, safe=TRUE, logfile=TRUE, add=TRUE,
       verbose=TRUE, comp="GCC")
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`raneff`	logical: use the random effects model, otherwise use the normal model.
`safe`	logical: if `TRUE`, use safe mode with bounds checking on all array objects, otherwise use optimized mode for fastest execution.
`logfile`	logical: if `TRUE`, create a log file of the messages from the shell call.
`add`	logical: if `TRUE`, append shell call messages to an exsiting log file.
`verbose`	logical: if `TRUE`, report the shell call an its messages to the R console.
`comp`	string: compiler to use - "GCC" is only currently supported

Details

This function uses the C++ comiler declared in `.ADopts`. If `logfile=TRUE`, any errors will appear in `'prefix'.log`. If `verbose=TRUE`, they will appear in the R console.

Value

Invisibly returns the shell call and its messages.

Note

A wrapper function that can be called from a GUI exists as `.win.linkAD`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`convAD`, `compAD`, `makeAD`

`makeAD` *Make an Executable Binary File from a C File*

Description

Essentially a wrapper function that calls in sequence: `convAD`, `compAD`, and `linkAD`.

Usage

```
makeAD(prefix, raneff=FALSE, safe=TRUE, logfile=TRUE, verbose=TRUE)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`raneff`	logical: use the random effects model, otherwise use the normal model.
`safe`	logical: if `TRUE`, use safe mode with bounds checking on all array objects, otherwise use optimized mode for fastest execution.
`logfile`	logical: if `TRUE`, create a log file of the messages from the shell call.
`verbose`	logical: if `TRUE`, report the shell call an its messages to the R console.

Details

This function uses the C++ comiler declared in `.ADopts`. If `logfile=TRUE`, any errors will appear in `'prefix'.log`. If `verbose=TRUE`, they will appear in the R console.

Value

Returns nothing. The three functions called by `makeAD` each return the shell call and its messages.

Note

A wrapper function that can be called from a GUI exists as `.win.makeAD`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`convAD`, `compAD`, `linkAD`, `cleanAD`

`makeADopts` *Creates the ADMB Options List*

Description

Creates a global list object detailing the pathways to the ADMB directory, the GCC bin, and the user's preferred text editor.

Usage

```
makeADopts(admpath, gccpath, editor)
```

Arguments

`admpath`	explicit path to the user's ADMB directory.
`gccpath`	explicit path to the user's GCC bin (C-compiler) directory.
`editor`	explicit path and program to use for editing text.

Value

Creates a global, hidden list object called `.ADopts`.

Note

A wrapper function that can be called from a GUI exists as `.win.makeADopts`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`makeADopts`, `writeADopts`

`parseCmd` *Parse an Indexed ADMB Command*

Description

Parse an indexed ADMB command line for a specified `index`, operating system (`os`), and compiler (`comp`). The result depends on the project `prefix`, the path (`admpath`) to the ADMB home directory, and the path (`gccpath`) to the C++ compiler. Within the database, variables are denoted by leading `@` characters.

Usage

```
parseCmd(prefix, index, os=.Platform$OS, comp="GCC", admpath="", gccpath="")
```

Arguments

`prefix`	prefix for the ADMB project.
`index`	index that indicates one of eight possibilities related to three processing steps (convert, compile, link) and options: safe or optimize, random effects or normal.
`os`	operating system
`comp`	C++ compiler description
`admpath`	explicit path for the ADMB home directory.
`gccpath`	explicit path for the C++ bin directory.

Value

Character string, the ADMB command from `ADMBcmd` corresponding to the specified index, prefix, and system paths.

Author(s)

Rowan Haigh, Pacific Biological Station, Nanaimo BC

See Also

`ADMBcmd`

`plotMC` *Plot Results of MCMC Simulation*

Description

Plot results of an ADMB MCMC simulation using various plot methods.

Usage

```
plotMC(prefix, act="pairs", pthin=1, useCols=NULL)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`act`	string scalar: action describing plot type (current choices: `"pairs"`, `"eggs"`, `"acf"`, `"trace"`, and `"dens"`).
`pthin`	numeric scalar indicating interval at which to collect records from the `.mc.dat` file for plotting.
`useCols`	logical vector indicating which columns of `.mc.dat` to plot.

Note

A wrapper function that can be called from a GUI exists as `.win.plotMC`. Use the PBSadmb GUI to explore these plots easily.

Author(s)

Rowan Haigh, Pacific Biological Station, Nanaimo BC, Canada

See Also

`runMC`, `showADargs`

`readADopts` *Reads an ADMB Options List into Memory From a File*

Description

Reads ADMB options into a global, hidden list object called `.ADopts` from an ASCII text file using `PBSmodelling::readList`).

Usage

```
readADopts(optfile="ADopts.txt")
```

Arguments

`optfile`	string name of an ASCII text file containing ADMB options information.

Value

No values returned. Reads the ADMB options into the list object `.ADopts`.

Note

A wrapper function that can be called from a GUI exists as `.win.readADopts`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`makeADopts`,`writeADopts`

`readRep` *Read an ADMB Report into R Memory*

Description

Import ADMB-generated report files into R's memory using the names of the report files to name the R-objects.

Usage

```
readRep(prefix, suffix=c(".cor",".rep",".std",".mc.dat"), global=FALSE)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`suffix`	string scalar/vector specifying one or more suffixes.
`global`	logical: if `TRUE`, save the imported reports as objects to global environment using the same names as the report files.

Details

If the report object is one of `c(".cor", ".std", ".mc.dat")`, the report object is a data frame, otherwise it is a string vector. Multiple report objects are returned as a list of objects. A single report object is returned as the object itself.

This function attempts to detect the file format from a number of possibilities. For example, if the file has the special format recognized by PBSmodelling, then the function returns a list with named components. The example `vonb` included with this package shows how to write the template to get consistent variable names between ADMB and R. See the User's Guide for complete details.

Value

Invisibly returns the list of report objects. If only one report is imported, a single report object is returned.

Note

A wrapper function that can be called from a GUI exists as `.win.readRep`.

Author(s)

Rowan Haigh, Pacific Biological Station, Nanaimo BC, Canada

See Also

`editADfile`, `.win.viewRep`

runAD *Run an Executable Binary File*

Description

Run the executable binary file `'prefix'.exe` that was created by `makeAD`.

Usage

```
runAD(prefix, argvec="", logfile=TRUE, add=TRUE, verbose=TRUE)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`argvec`	string scalar/vector of arguments appropriate for the executable `'prefix'.exe`.
`logfile`	logical: if `TRUE`, create a log file of the messages from the shell call.
`add`	logical: if `TRUE`, append shell call messages to an exsiting log file.
`verbose`	logical: if `TRUE`, report the shell call an its messages to the R console.

Details

This function typically reads the two files `'prefix'.dat` and `'prefix'.pin`, although in same cases one or both of these files may not be necessary.

If `logfile=TRUE`, output (including error messages, if any) will appear in `'prefix'.log`. If `verbose=TRUE`, it will appear in the R console.

Value

Invisibly returns the results of the shell call.

Note

A wrapper function that can be called from a GUI exists as `.win.runAD`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`runMC`, `makeAD`, `cleanAD`

runMC *Run an Executable Binary File in MCMC Mode*

Description

Run the executable binary file `'prefix'.exe`, created by `makeAD`, to generate MCMC simulations.

Usage

```
runMC(prefix, nsims=2000, nthin=20, outsuff=".mc.dat",
      logfile=FALSE, add=TRUE, verbose=TRUE)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`nsims`	numeric scalar indicating number of MCMC simulations to perform.
`nthin`	numeric scalar indicating the sampling rate or thinning of the `nsims` MCMC simulations to report.
`outsuff`	string name suffix of the MCMC output data file.
`logfile`	logical: if `TRUE`, create a log file of the messages from the shell call.
`add`	logical: if `TRUE`, append shell call messages to an exsiting log file.
`verbose`	logical: if `TRUE`, report the shell call an its messages to the R console.

Details

This function runs `'prefix'.exe` twice, first with the arguments `-mcmc 'nsims' -mcsave 'nthin'` and second with the argument `-mceval`. By default, output goes to the file `'prefix'.mc.dat`, although a user can specify a different output suffix.

To see this function in action, use the PBSadmb GUI with the example `vonb` or `simpleMC`.

Value

Invisibly returns the results of the shell call.

Note

A wrapper function that can be called from a GUI exists as `.win.runMC`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`runAD`, `makeAD`, `cleanAD`

`showADargs` *Show All Arguments for an ADMB Executable*

Description

Show all arguments available for an ADMB executable in the default text editor.

Usage

```
showADargs(prefix, ed=TRUE)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).
`ed`	logical: if `TRUE`, write the ADMB arguments to a file and view them with the text editor, else display the arguments on the R console.

Value

Invisibly returns the argument list.

Note

A wrapper function that can be called from a GUI exists as `.win.showADargs`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`editADfile`, `runAD`

`startLog` *Start a Log File*

Description

Start a log file by removing any previous version and appending header information.

Usage

```
startLog(prefix)
```

Arguments

`prefix`	string name prefix of the ADMB project (e.g., `"vonb"`).

Value

No explicit value reurned. Writes header lines into a log file `'prefix'.log`.

Note

A wrapper function that can be called from a GUI exists as `.win.startLog`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`appendLog`, `editADfile`

`writeADopts` *Writes the ADMB Options List from Memory to a File*

Description

Writes the global ADMB options list to a file in 'PBS' format (see `PBSmodelling::writeList`).

Usage

```
writeADopts(optfile="ADopts.txt")
```

Arguments

`optfile`	string name of the intended output file.

Value

Returns `opts` invisibly. Writes the options list object to an ASCII file.

Note

A wrapper function that can be called from a GUI exists as `.win.writeADopts`.

Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

See Also

`makeADopts`, `readADopts`

Index