

Using samtools C libraries with *Rsamtools*

Martin Morgan

Modified: 20 March, 2015. Compiled: October 30, 2017

Contents

| | | |
|-----|------------------------------------|---|
| 1 | Background | 1 |
| 2 | Use | 1 |
| 2.1 | Discover header files | 2 |
| 2.2 | Link to static libraries | 2 |

This document is meant for package developers wanting to use the [samtools](#) C libraries provided by [Rsamtools](#). The instructions here are based on the 'Writing R Extensions' manual associated with R-2.13; consult the current version of the manual for up-to-date instructions.

1 Background

[Rsamtools](#) arranges to install static versions of the `libbam` and `libbcf` libraries. The idea is that other packages can use these to implement C functionality that uses these libraries. This means that the samtools libraries are available in a consistent version to users across platforms, without requiring installation of additional software.

[Rsamtools](#) takes the following approach. On installation, [Rsamtools](#) contains a snapshot of the samtools library source code under `src/samtools`. [Rsamtools](#) makes static version(s) of the samtools libraries `libbam.a` and `libbcf.a`. These static libraries are specific to the operating system on which [Rsamtools](#) is being installed, are found under `usr/lib${R_ARCH}` in the user library location specified by the mechanism (e.g., `biocLite`, `install.packages`) used to install [Rsamtools](#). At the same time, [Rsamtools](#) copies headers required to use the library to the location `include/samtools`.

2 Use

To use these libraries, the third party package developer needs to (1) discover the appropriate header files when their package is built, and (2) link in the libraries.

Note that in order to link correctly across platforms your package must provide *both* the respective `src/Makevars` and `src/Makevars.win` files.

2.1 Discover header files

To discover appropriate header files at package installation time, add

LinkingTo: Rsamtools

to the DESCRIPTION file, and reference the relevant include files as, for instance,

```
#include "samtools/bam.h"
```

The content of the include files can be found in the *Rsamtools* source (under src/samtools) or at their installed location.

2.2 Link to static libraries

Linking to the static libraries is accomplished by including the following code in src/Makevars.win for *Windows*:

```
SAMVARS=$(shell echo 'cat(Rsamtools:::pkgMk())' |\  
  "${R_HOME}/bin/R" --vanilla --slave)  
include $(SAMVARS)
```

```
PKG_LIBS=$(SAMTOOLS_LIBS)  
PKG_CPPFLAGS=$(SAMTOOLS_CPPFLAGS)
```

and with the following code in src/Makevars for all other platforms:

```
SAMTOOLS_PATH=\`  
  echo 'cat(Rsamtools:::pkgLd())' |\  
  "${R_HOME}/bin/R" --vanilla --slave\  
SAMTOOLS_LIBS="$(SAMTOOLS_PATH)/libbam.a" "$(SAMTOOLS_PATH)/libbcf.a"\  
  "$(SAMTOOLS_PATH)/libtabix.a" -lz -pthread  
SAMTOOLS_CPPFLAGS=-D_USE_KNETFILE -DBGZF_CACHE -D_FILE_OFFSET_BITS=64 \  
  -D_LARGEFILE64_SOURCE  
  
PKG_LIBS=$(SAMTOOLS_LIBS)  
PKG_CPPFLAGS=$(SAMTOOLS_CPPFLAGS)
```

This updates the environment variables \$PKG_CPPFLAGS and \$PKG_LIBS; if your Makevars/Makevars.win modifies these also, do so by adding to the respective line, e.g.,

```
PKG_LIBS=$(SAMTOOLS_LIBS) -lfoo  
PKG_CPPFLAGS=$(SAMTOOLS_CPPFLAGS) -I/path/bar
```