RegionABEL package review

L.C. Karssen

<2015-01-14 w0>

Contents

1 Introduction

One of the goals of the GenABEL project is to be an environment in which people can work on the implementation of statistical methodology into user-friendly software packages.

In order for the project to be sustainable the packages that are accepted into the GenABEL suite must meet certain standards. Without certain standards/guidelines package maintenance will be difficult and time consuming. Moreover, if the user interface is awkward of if the package lacks documentation users will be less likely to use the package. In short these standard revolve around the following:

- *Maintainability of the package*: is the code understandable? Does it follow our coding standards? Is the code documented?
- *User-friendliness*: What is the quality of the user documentation? Are there any examples? Is the user interface compatible with what is to be expected?

One of the ways to ensure a healthy ecosystem is to have reviews of candidate packages. Such a review would be similar to the peer review done for scientific publications. In order to have good quality package reviews we have put together this document describing in a structured way the minimum questions a package reviewer should ask.

Please consider this document a working draft. Feedback is very much welcomed.

Note that our coding style closely follows the Google style guide and is documented here.

2 Legal issues

2.1 Is the copyright holder clearly mentioned?

Yes, the Author is listed in the DESCRIPTION file (and no Copyright field is used, so according to "Writing R Extensions" (http://cran.r-project.org/doc/manuals/R-exts.html#The-DESCRIPTION-file) the author is assumed to hold the copyright).

2.2 Is there a clear (standard) license?

Yes, the GPL (>=v2), listed in the DESCRIPTION file

2.3 Is the license GNU GPL-compatible?

Yes

3 Technical quality

3.1 Is the installation procedure clearly documented? Is the code easy to compile and run?

This is an R package, so in principle installation is as simple as running `install.packages(file.tar.gz)` and will be even easier once accepted into CRAN (because then the source code is downloaded automatically). However, the package depends on `biomaRt`, which is not available on CRAN but on BioConductor. This means the user needs to install `biomaRt` first. See e.g. http://stackoverflow.com/questions/14343817/cran-package-depends-on-bioconductor-package-installing-error#comment19941762_14345336 for ways of making this easier/instructions to include for the user.

3.2 [For R packages] Does the package pass CRAN checks (http://cran.r-project.org/doc/manuals/r-devel/R-exts.html#Checking-packages)? At minimum, run "R CMD check ..." and "R CMD check –as-cran ..."

- `R CMD check` on the `tar.gz` file works, leaving 2 NOTES
- `R CMD check --as-cran` on the `tar.gz` file works, leaving 3 NOTES
- Most NOTES are of the type: `gene.metanalysis: no visible global function definition for 'estlambda'` and are probably related to the earlier NOTE

```
Packages in Depends field not imported from:
'DatABEL' 'GenABEL' 'biomaRt' 'gplots'
These packages need to be imported from (in the NAMESPACE file)
for when this namespace is loaded but not attached.
```

3.3 Is the package documented? What is the quality of the documentation?

Yes.

- The functions (including internal functions) are documented using Roxygen.
- Comments in the code explain the function/use of various variables or sections of code.
- Note: the file/function `gene.summary()` is documented in Roxygen style, but without an apostrophe after the #. So the documentation doesn't show up in the manual.

3.4 [For R packages] Does `help(PackageName)` provide an adequate summary of the package and a review of the major functions?

Yes, but see comment about incorrect version number in the list or major issues at the end of the review.

3.5 [For R packages] Does the package use Roxygen2 for documentation?

Yes. The documentation can be improved by using the following Roxygen commands to create correctly formatted variable names and function references: `\code{}` and `\code{\link{somefunction}}`; see e.g. http://r-pkgs.had.co.nz/man.html#text-formatting

3.6 Are examples of usage provided?

Yes.

3.7 Does the package provide a tutorial/vignette? Can you comment on the tutorial?

Yes, a tutorial is included. Actually it is not a part of the package (yet?), but it was sent to me together with the package.

The tutorial provides a good introduction to the package. In section 2 it takes the reader through the necessary steps and also contains hints on what would be different in a real analysis. Showing how to correct for sample relatedness using the kinship matrix is a valuable addition for those with family-based studies, etc. The fact that section 2 ends by using GenABEL's `mmscore()` to double check the result is a nice touch.

- In the first sentence of the tutorial, could you change "RegionABEL is an R package based on the *ABEL packages" to "RegionABEL is an R package based on the GenABEL suite" (or "GenABEL suite of packages")? We prefer the use

of "GenABEL suite". The basic rationale is that the "GenABEL project" creates software packages of the "GenABEL suite" (among other things).

- Just above section 2 a reference is missing, is this a paper that is still to be published?

3.8 Is the source code of the tutorial/vignette provided?

No. It would be great to have the source, to allow others to contribute to it as well. In fact, I noticed some spelling errors that I could have fixed had the source been available :-).

3.9 Does the package make use of unit/integration/etc. tests?

No.

3.10 [For R packages] Does the package make use of RUnit testing?

No.

3.11 Does the code comply with the GenABEL coding standards?

In a large part is does and the code is in most cases very readable.

Things that need to be addressed:

- There is a mix of <- and = for assignments. Please use <- throughout, including a space before and after the <-.
- T and F are used instead of TRUE and FALSE
- Some lines are longer than 80 characters, e.g. because of an explanatory comment. In such a case the comment can usually be put above the line with the actual code.

Of lesser importance:

- There should be a space after a comma in function argument lists, for example: fun(a, b=TRUE, c="dummy") instead of fun(a,b=TRUE,c="dummy")
- The code becomes clearer if mathematical operators are surrounded by spaces, e.g.

  ```
  lik1 <- -2 * as.numeric(logLik(assoc))
  ```

  instead of

  ```
  lik1=-2*as.numeric(logLik(assoc))
  ```

3.12 Is the code readable/understandable?

Yes.

3.13 Does the code contain explanatory comments?

Yes (see earlier remarks on package documentation). Very helpful!

3.14 Were the design and methods implemented in package discussed during the development process (e.g. on the genabel-devel mailing list)?

No. At least not as far as I can remember.

4 Content

4.1 Does the package address a problem in the domain of statistical genomics?

Yes.

4.2 Is it streamlining analyses not covered elsewhere in the GenABEL suite? If not, does it improve the analysis already covered?

Yes.

4.3 Should it become a separate package or rather be incorporated into an existing package?

Yes.

4.4 Are any of the data types defined in other GenABEL packages used?

Yes, the `gwaa.data-class` object.

4.5 Are code/functions/data defined in other GenABEL packages used?

Yes, e.g. `estlambda()`, `databel()`, `polygenic()`, see also the NOTES when running the R checks. The `region.assoc()` function accepts a kinship matrix as calculated by `ibs()`.

5 Recommendations

5.1 What are the major issues which should be addressed?

- In `RegionABEL-package.R` incorrect version information & Date are being used. It is best to leave these out as they will most likely be forgotten in a future update, moreover, the correct information is printed on the title page of the PDF based on the information in the DESCRIPTION file.
- See comments on compliance with the GenABEL coding standards.
- Please fix the NOTES when running `R CMD check --as-cran`.
- Check the Roxygen documentation of the `gene.summary()` function, is it intentionally using # instead of #'?
- The object created by `region.assoc()` contains as 'strand' column, but this is coded as 1 and -1. It would be preferable to use the coding used by GenABEL (or do you see problems with incompatibilities?).

5.2 What other (optional) suggestions you could make to the author?

- Did you ever try to time a genome-wide or chromosome-wide sliding window scan using `region.assoc()` with e.g. 1kG imputed data? Any idea how long it would take? In other words, adding some information on the analysis time and how it scales with nr. of individuals and nr. of SNPs (in a region) would be great!
- Does it make sense to somehow cache the data downloaded from biomaRt (see `map.retriever()`, `gene.retreiver()`)? Can one expect the user to annotate several datasets like this (in a short timeframe)? I think it would be very useful (maybe for a second release if time doesn't permit to implement this now) to download the data once and check whether this data is up to date when the function is run a second time. Even more ideal would be to allow the user to specify the location of the cached data. In this way a system administrator can create a central directory so that multiple users don't start downloading the same data.
- Another point regarding `map.retriever()`: do you know the liftover tool (http://genome.sph.umich.edu/wiki/LiftOver)? It can be used in the same way (in principle). Does it make sense to let RegionABEL use liftover? On the one hand it means less code to maintain in RegionABEL, on the other hand it means that the user needs to install liftOver separately (and liftOver is Linux-only).
- Adding the source of the tutorial would be helpful as it is an important source of information to the user. Having the source available fits the spirit of the GenABEL project and will allow other to contribute to it as well.

- Adding some functional or unit tests (e.g. based on toy data) would help keep the package in shape and detect errors.
- Consider adding a third digit to the version number, e.g. 0.5-1, as suggested in "Writing R Extensions"
- In the package TITLE in the DESCRIPTION file some words start with a capital letters, some do not. I'd suggest using "Gene/Region-Wide Association Tools", including the hyphen in front of "Wide" (like in Genome-Wide Association Study). The same applies to the Description line.
- Consider adding spaces after the comma's in the Depends field of the DESCRIPTION file, while not mandatory, they increase legibility of the file.
- The function `svd()` is used several times with the `LINPACK=TRUE` argument. However, the `svd()` manual states that this argument is "Defunct and ignored".
- Some files/functions have Italian names (e.g. `mediatore()`), consider to convert these to English.
- Improve the Roxygen documentation be using `\code{}` and `\code{link{}}` where appropriate.