This is a really cool package that uses state of the art tech with ggplot and shiny, as well as interactive plots, to display ROC curves.
Unfortunately, the package is rather difficult to use and quite inconsistent. It also produces some bogus results (see below). As it is now, it looks more like a pretty cool tech demo than a scientific software I would use in my own research. Here are a few comments to try to improve it, as well as the manuscript itself.

Bug report:
The confidence intervals are apparently bogus. The following code executed in Rstudio or from the command line, with R 3.1.3 on Ubuntu 14.04 (from CRAN's Debian repository) and R 3.1.2 on Mac OS Yosemite (from Fink):

```
set.seed(42)
D.ex <- rbinom(100, size = 1, prob = .5)
M.ex <- rnorm(100, mean = D.ex)
roc.estimate <- calculate_roc(round(M.ex, 2), D.ex, ci = TRUE)
plot_interactive_roc(ggroc(roc.estimate, ci = TRUE))
```

After clicking on a point around 1.6, I got the screenshot shown in Figure 1. This seems related to the rounding (here to the 2nd decimal).

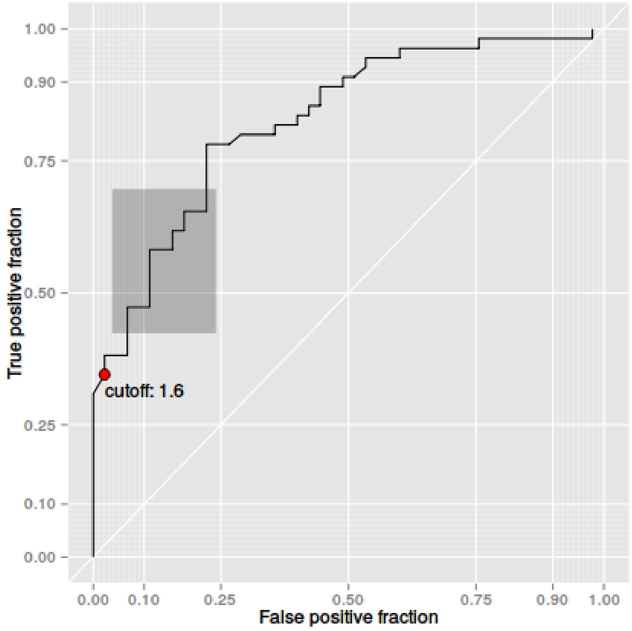

Figure 1: the CI region doesn't span the cutoff point.

General:

- The manuscript is written partially in the first person. It is surprising and not totally consistent throughout the manuscript, some of which is written in passive tense (for instance “Labels can be added easily [...]”).
- The abstract highlights the literature review as a significant finding. Surprisingly

is is only briefly mentioned in the introduction. This seems a bit inconsistent.

- Strictly speaking one could argue the plotROC package "obscures the cutoffs", as they are not displayed by default when calling the ggroc function. This is not so different from existing packages (as reviewed in subsection 1.3) that allow accessing the thresholds and adding them to the plot in different ways. The user has to request an interactive or journal plot to get them. The abstract and introduction should be rephrased in a more neutral way on this topic, as this is not the key point addressed by this package.

Introduction:

- The introduction states that "ROC curves have been used for decades in signal detection theory". In fact, they originate from this field (s/n in radio receivers, hence the name)
- The "good outcome", "bad outcome" is confusing to design correct / incorrect classifications. Especially in the medical field. I don't think it is necessary.
- "A perfect test [...] has TPF(c) = 1 and FPF(c) = 0 for all c". This is not correct. It is sufficient that this equality holds true for a single c.
- The formula hat(TPF(c)) is slightly confusing. It may be useful to first introduce the confusion table with true | false positive | negatives. It seems correct nonetheless.
- "The greater the AUC, the more informative the test". This statement is incomplete at best. What matters is | AUC - 0.5 |. An AUC = 0 is also the result of a perfect test (for which $M < c$ if I am not mistaken).
- Existing software: pROC allows to plot cutoffs quite easily (with print.thres=TRUE, just checked now). Of course it still uses the base R plots.
- I am surprised a Pubmed search returned only 54 papers. This part would probably deserve to be expanded a little bit, and maybe turned into a whole section.

Usage of the package:

- Plots such as fig 4 will be hard to read in small figures, as is often seen in journals.
- Fig 5 is unclear. Especially, what was the cutoff used to compute the region, as several ones are displayed in the region?
- In general, the use of shaded region instead of bars is quite surprising, although perhaps more accurate than bars.
- Why are the regions rectangle-shaped and not oval? That would seem even more correct, although probably quite difficult to set up.
- The gridlines in figures 3-5 & 7 are so thin and light-coloured that they did not appear when I printed this manuscript on paper. The printer was not (yet)

running out of ink.

- When reporting an AUC and confidence intervals in a publication, it seems to me at least as important to report values in addition to a plot. One can annotate a curve with the AUC (p. 13), with, for instance, label = “AUC=0.80”. But this value comes out of nowhere, and does not even look realistic (to me it looks more like AUC = 0.7). In addition, it isn't an “Advanced option”, as the subsection suggests, but rather as something quite basic. An easy way out would be to add a few additional lines computing this AUC to the manuscript. The case of CI seem a bit more difficult to deal with.

Package:

- The usage requires a lot of typing, and does not appear to be straightforward. For instance, the function calculate_roc must be used to create the FPF and TPF matrix. This will create a lot of headaches for infrequent users: is it calculate_, or create_, or generate_, or …?
- This wouldn't be an issue if users could browse the functions of the package as one can usually do by typing ?plotROC. Unfortunately, this leads to a stub that has never been authored. Please update man/plotROC-package.R with some useful pointers. Of course one can always find this paper, but it is much easier to get a proper help.
- It is of course a design choice, but the use of underscores over dots (or camelCase) in function names seems quite surprising and may disturb some users. See Bååth, R. (2012). The State of Naming Conventions in R. R Journal, 4, 74-75. R Foundation for Statistical Computing for a review of current usage of function name styles.
- Wouldn't it be more idiomatic to introduce the roc as a ggplot geometry? That would allow some much more straightforward syntax like:
  data.for.roc <- data.frame(M.ex, D.ex)
  qplot(M.ex, D.ex, data = data.for.roc) + geom_roc()
  or suchlike that would be much easier to remember.
- I don't understand why one has to pass a ggroc argument to plot_interactive_roc. That is a lot of functions to use in row and in the right order to get an interactive plot, and 3 times (_)?roc. I suspect a few well designed objects would make it much more user-friendly.
- What do the modCss and modJs functions do? The documentation is really limited and an example would help. Same with verify_d. As it stands now these functions seem useless.
- The help files lack examples altogether. Not only does it help a lot to have copy-pasteable code snippets as a start, but it can also help checking the package for potential regressions during updates.

- Performance: as most R packages doing ROC curves, this one scales pretty badly with larger sample sizes. I recommend using the cumsum function to compute the FPF/TPF. This will be much faster than the current implementation with sapply (which is really just a hidden slow loop). See the ROCR package that does it correctly. The calculate_roc function with the sample data of subsection 2.4 scaled to 1E5 data points literally collapsed and took > 10 minutes to compute. Such sample sizes may be uncommon in oncology, but are not rare in genomics or computer science. ROCR returns nearly instantly with such a low (sic!) sample size and scales much higher.
- How do I add the cut-offs to the plot? If I say ggplot(..., ci.at = 0.5) I get an unused argument error and nothing is added. Is this only available in the interactive plot?

Shiny app:

- The shiny application has an hard dependency on ggthemes and is unusable without that package. Unfortunately it is only “Suggest”ed, and and error message is printed only after a query to the shiny web application that doesn't work. PlotROC should either “Depend” on that package, as for shiny, or catch its absence and print the error message immediately when the shiny app is launched.
- On my Ubuntu system, the Download button opens a 404 error page with URL as http://127.0.0.1:3048/session/61edd0aac13532dc34788b220e9d27c8/download/printDownload?w= instead of downloading the image. I can right click the image and “View image” normally. The download works properly on my Mac where a tab shortly opens before the download starts.
- The browse button appears very low in the page and is not visible enough IMO.