[RFC] A case for freezing CRAN

Tim Triche, Jr. Mar 20, 2014; 5:11pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More

23 posts

That doesn't make sense.

If an API changes (e.g. in Matrix) and a program written against the old API can no longer run, that is a very different issue than if the same numbers (data) give different results. The latter is what I am guessing you address. The former is what I believe most people are concerned about here. Or at least I hope that's so.

It's more an issue of usability than reproducibility in such a case, far as I can tell (see e.g. http://liorpachter.wordpress.com/2014/03/18/reproducibility-vs-usability/). If the same data produces substantially different results (not attributable to e.g. better handling of machine precision and so forth, although that could certainly be a bugaboo in many cases... anyone who has programmed numerical routines in FORTRAN already knows this) then yes, that's a different type of bug. But in order to uncover the latter type of bug, the code has to run in the first place. After a while it becomes rather impenetrable if no thought is given to these changes.

So the Bioconductor solution, as Herve noted, is to have freezes and releases. There can be old bugs enshrined in people's code due to using old versions, and those can be traced even after many releases have come and gone, because there is a point-in-time snapshot of about when these things occurred. As with (say) ANSI C++, deprecation notices stay in place for a year before anything is actually done to remove a function or break an API. It's not impossible, it just requires more discipline than declaring that the same program should be written multiple times on multiple platforms every time. The latter isn't an efficient use of anyone's time.

Most of these analyses are not about putting a man on the moon or making sure a dam does not break. They're relatively low-consequence exploratory sorties. If something comes of them, it would be nice to have a point-in-time reference to check and see whether the original results were hooey. That's a lot quicker and more efficient than rewriting everything from scratch (which, in some fields, simply ensures things won't get checked).

My $0.02, since we do still have those to bedevil cashiers.

Statistics is the grammar of science.
Karl Pearson <http://en.wikipedia.org/wiki/The_Grammar_of_Science>

On Thu, Mar 20, 2014 at 1:28 PM, Ted Byers <[hidden email]> wrote:

> On Thu, Mar 20, 2014 at 3:14 PM, Hervé Pagès <[hidden email]> wrote:
> >
> > > On 03/20/2014 03:52 AM, Duncan Murdoch wrote:
> > >
> > >> On 14-03-20 2:15 AM, Dan Tenenbaum wrote:
> > >>
> > >>>
> > >>>
> > >>> ----- Original Message -----
... [show rest of quote]
[[alternative HTML version deleted]]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Ted Byers Mar 20, 2014; 5:13pm Re: [RFC] A case for freezing CRAN Reply | Threaded | More

107 posts

In reply to this post by Jeroen Ooms.

On Thu, Mar 20, 2014 at 4:53 PM, Jeroen Ooms <[hidden email]>wrote:

> On Thu, Mar 20, 2014 at 1:28 PM, Ted Byers <[hidden email]> wrote:
>>
>> Herve Pages mentions the risk of irreproducibility across three minor
>> revisions of version 1.0 of Matrix. My gut reaction would be that if the
>> results are not reproducible across such minor revisions of one library,
>> they are probably just so much BS.
>>
>
> Perhaps this is just terminology, but what you refer to I would generally
... [show rest of quote]

> If someone is publishing results that I think are questionable and I
> cannot replicate them, I want to know exactly how those outcomes were
> obtained in the first place, so that I can 'debug' the problem. It's quite
> important to be able to trace back if incorrect results were a result of a
> bug, incompetence or fraud.
>
> OK. That is where archives come in. When I had to deal with that sort of thing, I provided copies of both data and code to whoever asked. It ought not be hard for authors to make an archive, to e.g. an optical disk, that includes the software used along with the data, and store it like any other backup, so it can be provided to anyone upon request.

> Let's take the example of the Reinhart and Rogoff case. The results
> obviously were not replicable, but without more information it was just the
> word of a grad students vs two Harvard professors. Only after reproducing
> the original analysis it was possible to point out the errors and proof
> that the original were incorrect.
>
>
>
>
... [show rest of quote]
to which everything relevant was stored would solve that problem (and it would be extremely easy for the researcher or his/her supervisor to do). I once had a reviewer complain he couldn't reproduce my results, so I sent him my code, which, translated into any of the Algol family of languages, would allow him, or anyone else, to replicate my results regardless of their programming language of choice. Once he had my code, he found his

error and reported back that he had finally replicated my results. Several of my colleagues used the same practice, with the same consequences (whenever questioned, they just provide their code, and related software, and then their results were reproduced). There is nothing like backups with due attention to detail.

Cheers

Ted

--
R.E.(Ted) Byers, Ph.D.,Ed.D.

[[alternative HTML version deleted]]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Ted Byers Mar 20, 2014; 5:24pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

107 posts

In reply to this post by Tim Triche, Jr.

On Thu, Mar 20, 2014 at 5:11 PM, Tim Triche, Jr. <[hidden email]>wrote:

> That doesn't make sense.
>
> If an API changes (e.g. in Matrix) and a program written against the old
> API can no longer run, that is a very different issue than if the same
> numbers (data) give different results. The latter is what I am guessing
> you address. The former is what I believe most people are concerned about
> here. Or at least I hope that's so.
>
> The problem you describe is the classic case of a failure of backward compatibility. That is completely different from the question of reproducibility or replicability. And, since I, among others, noticed the question of reproducibility had arisen, I felt a need to primarily address that.

I do not have a quibble with anything else you wrote (or with anything in this thread related to the issue of backward compatibility), and I have enough experience to know both that it is a hard problem and that there are a number of different solutions people have used. Appropriate management of deprecation of features is one, and the use of code freezes is another. Version control is a third. Each option carries its own advantages and disadvantages.

> It's more an issue of usability than reproducibility in such a case, far
> as I can tell (see e.g.
> http://liorpachter.wordpress.com/2014/03/18/reproducibility-vs-usability/). If the same data produces substantially different results (not
> attributable to e.g. better handling of machine precision and so forth,
> although that could certainly be a bugaboo in many cases... anyone who has
> programmed numerical routines in FORTRAN already knows this) then yes,
> that's a different type of bug. But in order to uncover the latter type of
> bug, the code has to run in the first place. After a while it becomes
... [show rest of quote]

Ted

[[alternative HTML version deleted]]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Tim Triche, Jr. Mar 20, 2014; 5:27pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

In reply to this post by Ted Byers

> There is nothing like backups with due attention to detail.


23 posts

Agreed, although given the complexity of dependencies among packages, this might entail several GB of snapshots per paper (if not several TB for some papers) in various cases. Anyone who is reasonably prolific then gets the exciting prospect of managing these backups.

At least if I grind out a vignette with a bunch of Bioconductor packages and call sessionInfo() at the end, I can find out later on (if, say, things stop working) what was the state of the tree when it last worked, and what might have changed since then. If a self-contained C++ or FORTRAN program is sufficient to perform an entire analysis, that's awesome, and it ought to be stuffed into revision control (doesn't everyone already do this?). But once you start using tools that depend on other tools, it becomes substantially more difficult to ensure that

1) a comprehensive snapshot is taken
2) reviewers, possibly on different platforms and/or major versions, can run using that snapshot
3) some means of a quick sanity check ("does this analysis even return sensible results?") can be run

Hopefully this is better articulated than my previous missive.

I believe we fundamentally agree; some of the particulars may be an issue of notation or typical workflow.

Statistics is the grammar of science.
Karl Pearson <http://en.wikipedia.org/wiki/The_Grammar_of_Science>

On Thu, Mar 20, 2014 at 2:13 PM, Ted Byers <[hidden email]> wrote:

> On Thu, Mar 20, 2014 at 4:53 PM, Jeroen Ooms <[hidden email]
> >wrote:
>
> > On Thu, Mar 20, 2014 at 1:28 PM, Ted Byers <[hidden email]>
> wrote:
> >>
> >> Herve Pages mentions the risk of irreproducibility across three minor
> >> revisions of version 1.0 of Matrix. My gut reaction would be that if
> the
... [show rest of quote]

[[alternative HTML version deleted]]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Ted Byers Mar 20, 2014; 5:55pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More


107 posts

On Thu, Mar 20, 2014 at 5:27 PM, Tim Triche, Jr. <[hidden email]>wrote:

> > There is nothing like backups with due attention to detail.
>
> Agreed, although given the complexity of dependencies among packages, this
> might entail several GB of snapshots per paper (if not several TB for some
> papers) in various cases. Anyone who is reasonably prolific then gets the
> exciting prospect of managing these backups.
>
> Isn't that what support staff is for? ;-) But, storage space is cheap,
and as tedious as managing backups can be (definitely not fun), it is
managable.

> At least if I grind out a vignette with a bunch of Bioconductor packages
> and call sessionInfo() at the end, I can find out later on (if, say, things
> stop working) what was the state of the tree when it last worked, and what
> might have changed since then. If a self-contained C++ or FORTRAN program
> is sufficient to perform an entire analysis, that's awesome, and it ought

> to be stuffed into revision control (doesn't everyone already do this?).
> But once you start using tools that depend on other tools, it becomes
> substantially more difficult to ensure that
>
... [show rest of quote]

I understand this, as I routinely work with complex distributed systems involving multiple programming languages and other diverse tools. But such is part of the overhead of doing quality work.

> I believe we fundamentally agree; some of the particulars may be an issue
> of notation or typical workflow.
>
>
> I agree that we fundamentally agree ;-)

>From my experience, the issues addressed in this thread are probably best handled by in the package developers and those authors that use their packages, rather than imposing additional work on those responsible for CRAN, especially when the means for doing things a little differently than how CRAN does it are readily available.

Cheers

Ted
R.E.(Ted) Byers, Ph.D.,Ed.D.

>
> Statistics is the grammar of science.
> Karl Pearson <http://en.wikipedia.org/wiki/The_Grammar_of_Science>
>
>
> On Thu, Mar 20, 2014 at 2:13 PM, Ted Byers <[hidden email]> wrote:
>
>> On Thu, Mar 20, 2014 at 4:53 PM, Jeroen Ooms <[hidden email]
>> >wrote:
... [show rest of quote]

--

[[alternative HTML version deleted]]

_____
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Hervé Pagès Mar 20, 2014; 6:23pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More


452 posts

In reply to this post by Ted Byers

On 03/20/2014 01:28 PM, Ted Byers wrote:

> On Thu, Mar 20, 2014 at 3:14 PM, Hervé Pagès <[hidden email]
> <mailto:[hidden email]>> wrote:
>
> On 03/20/2014 03:52 AM, Duncan Murdoch wrote:
>
> On 14-03-20 2:15 AM, Dan Tenenbaum wrote:
>
>
>
... [show rest of quote]

If you use R-2.13.2, you get Matrix 1.1-2-2 on Linux. AFAIK this is the most recent version of Matrix, aimed to be compatible with the most current version of R (i.e. R 3.0.3). However, it has never been tested with R-2.13.2. I'm not saying that it should, that would be a big waste of resources of course. All I'm saying it that it doesn't make sense to

serve by default a version that is known to be incompatible with the version of R being used. It's very likely to not even install properly.

For the apparently small differences between the versions you get on Windows and Mac, the Matrix package was just an example. With other packages you get (again if you use R-2.13.2):

```
           src   win    mac
abc        1.8   1.5    1.4
ape      3.1-1 3.0-1    2.8
BaSTA    1.9.3   1.1    1.0
bcrm     0.4.3   0.2    0.1
BMA   3.16.2.3  3.15 3.14.1
Boruta   3.0.0   1.6    1.5
...
```

Are the differences big enough?

Also note that back in October 2011, people using R-2.13.2 would get e.g. ape 2.7-3 on Linux, Windows and Mac. Wouldn't it make sense that people using R-2.13.2 today get the same? Why would anybody use R-2.13.2 today if it's not to run again some code that was written and used two years ago to obtain some important results?

Cheers,
H.

> My gut reaction would be that if
> the results are not reproducible across such minor revisions of one
> library, they are probably just so much BS. I am trained in
> mathematical ecology, with more than a couple decades of post-doc
> experience working with risk assessment in the private sector. When I
> need to do an analysis, I will repeat it myself in multiple products, as
> well as C++ or FORTRAN code I have hand-crafted myself (and when I wrote
> number crunching code myself, I would do so in multiple programming
> languages - C++, Java, FORTRAN, applying rigorous QA procedures to each
... [show rest of quote]

--
Hervé Pagès

Program in Computational Biology
Division of Public Health Sciences
Fred Hutchinson Cancer Research Center
1100 Fairview Ave. N, M1-B514
P.O. Box 19024
Seattle, WA 98109-1024

E-mail: [hidden email]
Phone: (206) 667-5791
Fax: (206) 667-1319

Uwe Ligges-3 Mar 20, 2014; 6:29pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

3583 posts

On 20.03.2014 23:23, Hervé Pagès wrote:

>
>
> On 03/20/2014 01:28 PM, Ted Byers wrote:
>> On Thu, Mar 20, 2014 at 3:14 PM, Hervé Pagès <[hidden email]
>> <mailto:[hidden email]>> wrote:
>>
>> On 03/20/2014 03:52 AM, Duncan Murdoch wrote:
>>
>> On 14-03-20 2:15 AM, Dan Tenenbaum wrote:

... [show rest of quote]

Not true, since Matrix 1.1-2-2 has

Depends: R (≥ 2.15.2)

Best,
Uwe Ligges

And none of them of course is what was

>> used
>> by the authors of the paper (they used Matrix 1.0-1, which is what
>> was
>> current when they ran their analysis).
>>
>> Initially this discussion brought back nightmares of DLL hell on
>> Windows. Those as ancient as I will remember that well. But now, the
>> focus seems to be on reproducibility, but with what strikes me as a
>> seriously flawed notion of what reproducibility means.
... [show rest of quote]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Hervé Pagès Mar 20, 2014; 7:28pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

452 posts

On 03/20/2014 03:29 PM, Uwe Ligges wrote:

>
>
> On 20.03.2014 23:23, Hervé Pagès wrote:
>>
>>
>> On 03/20/2014 01:28 PM, Ted Byers wrote:
>>> On Thu, Mar 20, 2014 at 3:14 PM, Hervé Pagès <[hidden email]
>>> <mailto:[hidden email]>> wrote:
>>>
... [show rest of quote]

OK. So that means Matrix is not available today for R-2.13.2 users on Linux:

```
> "Matrix" %in% rownames(available.packages()[ , ])
[1] FALSE
```

However since Matrix is a recommended package, it's included in the official R-2.13.2 source tarball so it gets installed when I install R:

```
> installed.packages()["Matrix", "Version", drop=FALSE]
       Version
Matrix "0.9996875-3"
```

As I mentioned earlier, the Matrix package was just an example. In the case of a non-recommended package, it will either be:

- unavailable by default (if the source package was removed or if the package maintainer consciously used the R >= x.y.z feature, e.g. the ape package),
- or available but incompatible (e.g. bcrm is broken with R-2.13.2 on Linux),
- or available and compatible, but with a very different version than the version that was available 2 years ago (e.g. BaSTA),
- or available and at the exact same version as 2 years ago (bingo!)

This is a very painful experience for anybody trying to install and use R-2.13.2 today to reproduce 2-year old results. Things could be improved a lot with very little changes.

Cheers,
H.

```
> sessionInfo()
R version 2.13.2 (2011-09-30)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=C              LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
[1] tools_2.13.2
```

>
>
> Best,
> Uwe Ligges
>
>
> And none of them of course is what was
>>> used
>>> by the authors of the paper (they used Matrix 1.0-1, which is what
... [show rest of quote]

--
Hervé Pagès

Program in Computational Biology
Division of Public Health Sciences
Fred Hutchinson Cancer Research Center
1100 Fairview Ave. N, M1-B514
P.O. Box 19024
Seattle, WA 98109-1024

E-mail: [hidden email]
Phone: (206) 667-5791
Fax: (206) 667-1319

Gábor Csárdi Mar 20, 2014; 9:23pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

97 posts

Much of the discussion was about reproducibility so far. Let me emphasize another point from Jeroen's proposal.

This is hard to measure of course, but I think I can say that the existence and the quality of CRAN and its packages contributed immensely to the success of R and the success of people using R. Having one central, well controlled and tested package repository is a huge advantage for the users. (I know that there are other repositories, but they are either similarly well controlled and specialized (BioC), or less used.) It would be great to keep it like this.

I also think that the current CRAN policy is not ideal for further growth. In particular, updating a package with many reverse dependencies is a frustrating process, for everybody. As a maintainer with ~150 reverse dependencies, I think not twice, but ten times if I really want to publish a new version on CRAN. I cannot speak for other maintainers of course, but I have a feeling that I am not alone.

Tying CRAN packages to R releases would help, because then I would not have to worry about breaking packages in the stable version of CRAN, only in

CRAN-devel.

Somebody mentioned that it is good not to do this because then users get bug fixes and new features earlier. Well, in my case, the opposite it true. As I am not updating, they actually get it (much) later. If it wasn't such a hassle, I would definitely update more often, about once a month. Now my goal is more like once a year.

Again, I cannot speak for others, but I believe the current policy does not help progress, and is not sustainable in the long run. It penalizes the maintainers of "more important" (= many rev. dependencies, that is, which probably also means many users) packages, and I fear they will slowly move away from CRAN. I don't think this is what anybody in the R community would want.

Best,
Gabor

[[alternative HTML version deleted]]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

William Dunlap Mar 20, 2014; 9:45pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

1661 posts

> In particular, updating a package with many reverse dependencies is a
> frustrating process, for everybody. As a maintainer with ~150 reverse
> dependencies, I think not twice, but ten times if I really want to publish
> a new version on CRAN.

It might be easier if more of those packages came with good test suites.

Bill Dunlap
TIBCO Software
wdunlap tibco.com

> -----Original Message-----
> From: [hidden email] [mailto:[hidden email]] On Behalf
> Of Gábor Csárdi
> Sent: Thursday, March 20, 2014 6:24 PM
> To: r-devel
> Subject: Re: [Rd] [RFC] A case for freezing CRAN
>
> Much of the discussion was about reproducibility so far. Let me emphasize
> another point from Jeroen's proposal.
... [show rest of quote]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Gábor Csárdi Mar 20, 2014; 10:15pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

97 posts

On Thu, Mar 20, 2014 at 9:45 PM, William Dunlap <[hidden email]> wrote:

> > In particular, updating a package with many reverse dependencies is a
> > frustrating process, for everybody. As a maintainer with ~150 reverse
> > dependencies, I think not twice, but ten times if I really want to
> publish
> > a new version on CRAN.
>
> It might be easier if more of those packages came with good test suites.
>

Test suites are great, but I don't think this would make my job easier.
More tests means more potential breakage. The extreme of not having any
examples and tests in these 150 packages would be the easiest for _me_,
actually. Not for the users, though.....

What would really help is either fully versioned package dependencies (daydreaming here), or having a CRAN-devel repository, that changes and might break often, and a CRAN-stable that does not change (much).

Gabor

[...]

[[alternative HTML version deleted]]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Tim Triche, Jr. Mar 20, 2014; 10:19pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More


23 posts

Heh, you just described BioC

--t

> On Mar 20, 2014, at 7:15 PM, Gábor Csárdi <[hidden email]> wrote:
>
> On Thu, Mar 20, 2014 at 9:45 PM, William Dunlap <[hidden email]> wrote:
>
>>> In particular, updating a package with many reverse dependencies is a
>>> frustrating process, for everybody. As a maintainer with ~150 reverse
>>> dependencies, I think not twice, but ten times if I really want to
>> publish
>>> a new version on CRAN.
... [show rest of quote]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Tim Triche, Jr. Mar 20, 2014; 10:20pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More


23 posts

In reply to this post by Gábor Csárdi

Except that tests (as vignettes) are mandatory for BioC. So if something blows up you hear about it right quick :-)

--t

> On Mar 20, 2014, at 7:15 PM, Gábor Csárdi <[hidden email]> wrote:
>
> On Thu, Mar 20, 2014 at 9:45 PM, William Dunlap <[hidden email]> wrote:
>
>>> In particular, updating a package with many reverse dependencies is a
>>> frustrating process, for everybody. As a maintainer with ~150 reverse
>>> dependencies, I think not twice, but ten times if I really want to
>> publish
>>> a new version on CRAN.
... [show rest of quote]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Dan Tenenbaum Mar 20, 2014; 10:48pm **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More


96 posts

In reply to this post by Gábor Csárdi

----- Original Message -----

> From: "Gábor Csárdi" <[hidden email]>
> To: "r-devel" <[hidden email]>
> Sent: Thursday, March 20, 2014 6:23:33 PM

> Subject: Re: [Rd] [RFC] A case for freezing CRAN
>
> Much of the discussion was about reproducibility so far. Let me
> emphasize
> another point from Jeroen's proposal.
... [show rest of quote]

These are good points. Not only do maintainers think twice (or more) before updating packages but it also seems that there are CRAN policies that discourage frequent updates. Whereas Bioconductor welcomes frequent updates because they usually fix problems and help us understand interoperability/dependency issues. Probably the main reason for this difference is the existence of a devel branch where breakage can happen and it's not the end of the world.

> Again, I cannot speak for others, but I believe the current policy
> does not
> help progress, and is not sustainable in the long run. It penalizes
> the
> maintainers of "more important" (= many rev. dependencies, that is,
> which
> probably also means many users) packages, and I fear they will slowly
> move
> away from CRAN. I don't think this is what anybody in the R community
... [show rest of quote]

Jari Oksanen Mar 21, 2014; 4:33am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

129 posts

In reply to this post by Hervé Pagès

Freezing CRAN solves no problem of reproducibility. If you know the sessionInfo() or the version of R, the packages used and their versions, you can reproduce that set up. If you do not know, then you cannot. You can try guess: source code of old release versions of R and old packages are in CRAN archive, and these files have dates. So you can collect a snapshot of R and packages for a given date. This is not an ideal solution, but it is the same level of reproducibility that you get with strictly frozen CRAN. CRAN is no the sole source of packages, and even with strictly frozen CRAN the users may have used packages from other source. I am sure that if CRAN would be frozen (but I assume it happens the same day hell freezes), people would increasingly often use other package sources than CRAN. The choice is easy if the alternatives are to wait for the next year for the bug fix release, or do the analysis now and use package versions in R-Forge or github. Then you could not assume that frozen CRAN packages were used.

CRAN policy is not made in this mailing list, and CRAN maintainers are so silent that it hurts ears. However, I hope they won't freeze CRAN.

Strict reproduction seems to be harder than I first imagined: ./configure && make really failed for R 2.14.1 and older in my office desktop. To reproduce older analysis, I would also need to install older tool sets (I suspect gfortran and cairo libraries).

CRAN is one source of R packages, and certainly its policy does not suit all developers. There is no policy that suits all. Frozen CRAN would suit some, but certainly would deter some others.

There seems to a common sentiment here that the only reason anybody would use R older than 3.0.3 is to reproduce old results. My experience form the Real Life(™) is that many of us use computers that we do not own, but they are the property of our employer. This may mean that we are not allowed to install there any software or we have to pay, or the Department of project has to pay, to the computer administration for installing new versions of software (our case). This is often called security. Personally I avoid this by using Mac laptop and Linux desktop: these are not supported by the University computer administration and I can do what I please with these, but poor Windows users are stuck. Computer classes are also maintained by centralized computer administration. This January they had new R, but last year it was still two years old. However, users can install packages in their personal "folders" so that they can use current packages even with older R. Therefore I want to take care that the packages I maintain also run in older R. Therefore I also applaud the current CRAN policy where new versions of packages are "backported" to previous R release: Even if you are stuck with stale R, you need not be stuck with stale packages. Currently I cannot test with older R than 2.14.2, though, but I do that regularly and certainly before CRAN releases. If somebody wants to prevent this, they can set their package to unnecessarily depend on the current version of R. I would regard this as antisocial, but

nobody would ask what I think about this so it does not matter.

The development branch of my package is in R-Forge, and only bug fixes and (hopefully) non-breaking enhancements (isolated so that they do not influence other functions, safe so that API does not change or format of the output does not change) are merged to the CRAN release branch. This policy was adopted because it fits the current CRAN policy, and probably would need to change if CRAN policy changes.

Cheers, Jari Oksanen

Rainer M Krug-3 Mar 21, 2014; 4:40am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

68 posts

In reply to this post by Jeroen Ooms.

This is a long and (mainly) interesting discussion, which is fanning out in many different directions, and I think many are not that relevant to the OP's suggestion.

I see the advantages of having such a dynamic CRAN, but also of having a more stable CRAN. I prefer CRAN as it is now, but ion many cases a more stable CRAN might b an advantage. So having releases of CRAN might make sense. But then there is the archiving issue of CRAN.

The suggestion was made to move the responsibility away from CRAN and the R infrastructure to the user / researcher to guarantee that the results can be re-run years later. It would be nice to have this build in CRAN, but let's stick at the scenario that the user should care for reproducability.

Leaving the issue of compilation out, a package which is creating a custom installation of the R version which includes the source of the R version used and the sources of the packages in a on Linux compilable format, given that the relevant dependencies are installed, would be a huge step forward.

I know - compilation on Windows (and sometimes Mac) is a serious problem), but to archive *all* binaries and to re-compile all older versions of R and all packages would be an impossible task.

Apart from that - doing your analysis in a Virtual Machine and then simply archiving this Virtual Machine, would also be an option, but only for the more tech savy users.

In a nutshell: I think a package would be able to provide the solution for a local archiving to make it possible to re-run the simulation with the same tools at a later stage - although guarantees would not be possible.

Cheers,

Rainer
--
Rainer M. Krug
email: Rainer<at>krugs<dot>de
PGP: 0x0F52F982

Rainer M Krug-3 Mar 21, 2014; 4:49am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

In reply to this post by Jari Oksanen

Jari Oksanen <[hidden email]> writes:

> Freezing CRAN solves no problem of reproducibility. If you know the


68 posts

> sessionInfo() or the version of R, the packages used and their
> versions, you can reproduce that set up. If you do not know, then you
> cannot. You can try guess: source code of old release versions of R
> and old packages are in CRAN archive, and these files have dates. So
> you can collect a snapshot of R and packages for a given date. This is
> not an ideal solution, but it is the same level of reproducibility
> that you get with strictly frozen CRAN. CRAN is no the sole source of
> packages, and even with strictly frozen CRAN the users may have used
... [show rest of quote]

Agree completely here - the solution would be a package, which is packaging the source (or even binaries?) of your local R setup including R and packages used. The solution is local - not on a server.

>
> CRAN policy is not made in this mailing list, and CRAN maintainers are
> so silent that it hurts ears.

+1

> However, I hope they won't freeze CRAN.

Yes and no - if they do, we need a devel branch which acts like the current CRAN.

>
> Strict reproduction seems to be harder than I first imagined:
> ./configure && make really failed for R 2.14.1 and older in my office
> desktop. To reproduce older analysis, I would also need to install
> older tool sets (I suspect gfortran and cairo libraries).

Absolutely - let's not go there. And then there is also the hardware issue.

>
> CRAN is one source of R packages, and certainly its policy does not
> suit all developers. There is no policy that suits all. Frozen CRAN
> would suit some, but certainly would deter some others.
>
> There seems to a common sentiment here that the only reason anybody
> would use R older than 3.0.3 is to reproduce old results. My
> experience form the Real Life(™) is that many of us use computers that
> we do not own, but they are the property of our employer. This may
... [show rest of quote]

Nicely put.

> Computer classes are also
> maintained by centralized computer administration. This January they
> had new R, but last year it was still two years old. However, users
> can install packages in their personal "folders" so that they can use
> current packages even with older R. Therefore I want to take care that
> the packages I maintain also run in older R. Therefore I also applaud
> the current CRAN policy where new versions of packages are
> "backported" to previous R release: Even if you are stuck with stale
> R, you need not be stuck with stale packages. Currently I cannot test
... [show rest of quote]

--
Rainer M. Krug
email: Rainer<at>krugs<dot>de
PGP: 0x0F52F982

attachment0 (504 bytes) Download Attachment

Philippe Grosjean Mar 21, 2014; 5:06am Re: [RFC] A case for freezing CRAN Reply | Threaded | More

In reply to this post by Jeroen Ooms.

This is becoming an extremely long thread, and it is going in too many directions. However, I would like to mention here our ongoing five years projects ECOS project for the study of Open Source Ecosystems,


279 posts

among which, CRAN. You can find info here: http://informatique.umons.ac.be/genlog/projects/ecos/. We are in the second year now.

We are currently working on CRAN maintainability questions. See:

- Claes Maelick, Mens Tom, Grosjean Philippe, "On the maintainability of CRAN packages" in IEEE CSMR-WCRE 2014 Software Evolution Week, Antwerpen, Belgique, 2014 (2014)

- Mens Tom, Claes Maelick, Grosjean Philippe, Serebrenik Alexander, "Studying Evolving Software Ecosystems based on Ecological Models" in Mens Tom, Serebrenik Alexander, Cleve Anthony, "Evolving Software Systems" , Springer, Mens Tom, Serebrenik Alexander, Cleve Anthony, 978-3-642-45397-7 (2014)

Currently, we are building an Open Source system based on Virtualbox and Vagrant to recreate a virtual machine under Linux (Debian and Ubuntu considered for the moment) that would be as close as possible as a "simulated CRAN environment as it was at any given date". Our plans are to replay CRAN back in time and to instrumentize that platform to measure what we need for our ecological studies of CRAN.

The connection with this thread is the possibility to reuse this system for proposing something useful for reproducible research, that is, a reproducible platform, in the definition of reproducibility vs replicability Jeroen Ooms mentions. It would then be enough to record the date some R code was run on that platform (and perhaps whether it is 32 or 64 bit system) to be able to rebuild a similar software environment with all corresponding CRAN packages of the right version easily installable. In case something specific is required in addition to software proposed by default, Vagrant allows provisioning the Virtual machine in an easy way too… but then, the provisioning script must be provided too (not much a problem). Info required to rebuild the platform is shrunk down to a few kb Ascii text file. This is something easy to put together with your R code in, say, additional material of a publication.

Please, keep in mind that many platform-specific features in R (graphic devices, string encoding, and many more) may be a problem too for reproducing published results. Hence, the idea to use a virtual box using only one OS, Linux, no matter if you work on Windows, or Mac OS X, or… Solaris (anyone there?).

PhG

On 20 Mar 2014, at 21:53, Jeroen Ooms <[hidden email]> wrote:

> On Thu, Mar 20, 2014 at 1:28 PM, Ted Byers <[hidden email]> wrote:
>>
>> Herve Pages mentions the risk of irreproducibility across three minor
>> revisions of version 1.0 of Matrix. My gut reaction would be that if the
>> results are not reproducible across such minor revisions of one library,
>> they are probably just so much BS.
>>
>
> Perhaps this is just terminology, but what you refer to I would generally
... [show rest of quote]

[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Jari Oksanen Mar 21, 2014; 5:17am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More


129 posts

In reply to this post by Rainer M Krug-3

On 21/03/2014, at 10:40 AM, Rainer M Krug wrote:

>
>
> This is a long and (mainly) interesting discussion, which is fanning out
> in many different directions, and I think many are not that relevant to
> the OP's suggestion.
>
> I see the advantages of having such a dynamic CRAN, but also of having a
> more stable CRAN. I prefer CRAN as it is now, but ion many cases a more
> stable CRAN might b an advantage. So having releases of CRAN might make
... [show rest of quote]

There are two different problems that alternate in the discussion: reproducibility and breakage of CRAN dependencies. Frozen CRAN could make *approximate* reproducibility easier to achieve, but real

reproducibility needs stricter solutions. Actual sessionInfo() is minimal information, but re-building a spitting image of old environment may still be demanding (but in many cases this does not matter).

Another problem is that CRAN is so volatile that new versions of packages break other packages or old scripts. Here the main problem is how package developers work. Freezing CRAN would not change that: if package maintainers release breaking code, that would be frozen. I think that most packages do not make distinction between development and release branches, and CRAN policy won't change that.

I can sympathize with package maintainers having 150 reverse dependencies. My main package only has ~50, and it is sure that I won't test them all with new release. I sometimes tried, but I could not even get all those built because they had other dependencies on packages that failed. Even those that I could test failed to detect problems (in one case all examples were \dontrun and passed nicely tests). I only wish that if people *really* depend on my package, they test it against R-Forge version and alert me before CRAN releases, but that is not very likely (I guess many dependencies are not *really* necessary, but only concern marginal features of the package, but CRAN forces to declare those).

Still a few words about reproducibility of scripts: this can be hardly achieved with good coverage, because many scripts are so very ad hoc. When I edit and review manuscripts for journals, I very often get Sweave or knitr scripts that "just work", where "just" means "just so and so". Often they do not work at all, because they had some undeclared private functionalities or stray files in the author workspace that did not travel with the Sweave document. I think these -- published scientific papers -- are the main field where the code really should be reproducible, but they often are the hardest to reproduce. Nothing CRAN people do can help with sloppy code scientists write for publications. You know, they are scientists -- not engineers.

Cheers, Jari Oksanen

>
> Leaving the issue of compilation out, a package which is creating a
> custom installation of the R version which includes the source of the R
> version used and the sources of the packages in a on Linux compilable
> format, given that the relevant dependencies are installed, would be a
> huge step forward.
>
> I know - compilation on Windows (and sometimes Mac) is a serious
> problem), but to archive *all* binaries and to re-compile all older
... [show rest of quote]

_____
[hidden email] mailing list
https://stat.ethz.ch/mailman/listinfo/r-devel

Rainer M Krug-3 Mar 21, 2014; 6:08am **Re: [RFC] A case for freezing CRAN** Reply | Threaded | More

68 posts

Jari Oksanen <[hidden email]> writes:

> On 21/03/2014, at 10:40 AM, Rainer M Krug wrote:
>
>>
>>
>> This is a long and (mainly) interesting discussion, which is fanning out
>> in many different directions, and I think many are not that relevant to
>> the OP's suggestion.
>>
>> I see the advantages of having such a dynamic CRAN, but also of having a
... [show rest of quote]
Breakage of CRAN packages is a problem, to which I can not comment much. I have no idea how this could be saved unless one introduces more checks, which nobody wants. CRAN is a (more or less) open repository for packages written by engineers / programmers but also scientists of other fields - and that is the strength of CRAN - a central repository to find packages which conform to a minimal standard and format.

>
> Still a few words about reproducibility of scripts: this can be hardly
> achieved with good coverage, because many scripts are so very ad
> hoc. When I edit and review manuscripts for journals, I very often get
> Sweave or knitr scripts that "just work", where "just" means "just so
> and so". Often they do not work at all, because they had some
> undeclared private functionalities or stray files in the author
> workspace that did not travel with the Sweave document.

One reason why I *always* start my R sessions --vanilla and ave a local initialization script which I call manually.

> I think these
> -- published scientific papers -- are the main field where the code
> really should be reproducible, but they often are the hardest to
> reproduce.

And this is completely ouyt of the hands of R / CRAN / ... and in the hand of Journals and Authors. But R could provide a framework to make this more easy in form of a package which provides functions to make this a one-command approach.

> Nothing CRAN people do can help with sloppy code scientists
> write for publications. You know, they are scientists -- not
> engineers.

Absolutely - and I am also a sloppy scientists - I put my code online, but hope that not many people ask me later about it.

Cheers,

Rainer

>
> Cheers, Jari Oksanen
>>
>> Leaving the issue of compilation out, a package which is creating a
>> custom installation of the R version which includes the source of the R
>> version used and the sources of the packages in a on Linux compilable
>> format, given that the relevant dependencies are installed, would be a
>> huge step forward.
>>
... [show rest of quote]
--
Rainer M. Krug
email: Rainer<at>krugs<dot>de
PGP: 0x0F52F982

attachment0 (504 bytes) Download Attachment