

Bioconductor and R 4.0

(recap/extension of previous November presentation)

<http://bioconductor.org/developers/how-to/troubleshoot-build-report/>

S3 method registration!!

Many packages are currently failing because of undeclared S3 methods in the NAMESPACE. There is some background found on the [R developers blog](#).

Solution: Register the S3 method in the package NAMESPACE

```
S3method(<function>, <dispatch>)
```

Examples:

```
S3method(plot, TCC) #TCC package
```

```
S3method(transform, ProbeSet) #affypdnn package
```

S3 method registration!!

The resulting ERROR seems to take many forms depending on package code. Some of the current ERRORS seen on the build report related to S3 method declaration:

- Cannot coerce class `<structure>` to a `data.frame`
- Cannot coerce type `'S4'` to vector of type `'double'`
- No applicable method for `<foo>` applied to an object of class `<bar>`
- `'X'` is a list, but does not have components `'x'` and `'y'`

Removed Setting in R CMD config

The source of the ERROR is utilizing settings in package configure script that have been removed or replaced. There is a section of R NEWS “R CMD config no longer knows about the unused settings F77 and FCPIFCPLAGS, nor CXX98 and similar.” Executing the configuration script when installing the package fails, and the output contains lot of messages along the lines of the following:

- configure: WARNING: The flags FFLAGS="" do not work
- checking whether the ERROR: no information for variable ‘F77’
- configure: WARNING: This value for FFLAGS does not work.

Solution: Replace instances of “`${R} CMD config F77`” with “`${R} CMD config FC`”

Conditional lengths > 1

R 4.0 current condition has length > 1 will produce a Warning message.
Bioconductor build systems treat this as an ERROR in R CMD check

Traditionally `if` / `while` statements could act on vectors using the first element as the conditional value. This now produces a warning as seen in this dummy example and documented on [R developer blog](#)

```
> if (c(TRUE, FALSE)) {}  
NULL  
Warning message:  
In if (c(TRUE, FALSE)) { :  
  the condition has length > 1 and only the first element will be used
```

Solution: Generally the solution is to use `any()` or `all()` surrounding the vector
See also: <https://stat.ethz.ch/pipermail/bioc-devel/2020-January/016081.html>

Scalar / Vector Logic

This is not a change in R yet but we have been notified that it is forthcoming and have escalated to an ERROR on our daily builders in preparation. This type of ERROR occurs with the misuse of `&&` and `||`. The double `&&` and `||` imply a scalar comparison rather than a vector comparison that the singular `&` and `|` expect. See the dummy example below:

```
> c(TRUE, TRUE) && TRUE)
Error in c(TRUE, TRUE) && TRUE :
  'length(x) = 2 > 1' in coercion to 'logical(1)'
```

Solution: Most cases are misjudgment and misunderstanding of the use of a scalar comparison from a vector comparison. Changing the double `&&` / `||` to a singular `&` / `|` will generally be sufficient if a vector comparison is intended or having the vector argument use an appropriate `any()` or `all()` surrounding the vector will result in the appropriate scalar comparison. **Note:** If this comparison is in a conditional please see the section above; `any()` or `all()` will most likely be a better alternative.

See also: <https://stat.ethz.ch/pipermail/bioc-devel/2020-January/016081.html>

`class(x) == "foo" vs is(x, "foo")/inherits(x, "foo")`

There is a much better discussion and explanation on [R developer blog](#)

The sum up:

`class(x) == "foo"` should be avoided. Misleading if classes extend other class

`is / inherits` safer, cleaner, avoids assumptions

This has been encouraged as part of [Bioconductor coding best practices](#)

Matrix class now extends array

In R 4.0, the matrix class now extends the array class. There are many ERRORS from this change in addition to the logical changes already addressed.

```
> m = matrix()
> class(m)
[1] "matrix" "array"
> class(m) == "matrix"
[1] TRUE FALSE
> if ( class(m) == "matrix"){
Error in if (class(m) == "matrix") { : the condition has length > 1
```

Solution: Generally the best solution as mentioned on the previous slide would be to use `is()` or `inherits()`: `is(m, "matrix")` , `inherits(m, "matrix")`

data.frame stringAsFactors

In R 4.0, the default for data.frame argument `stringsAsFactors` changed from TRUE to FALSE. This change is causing the most breakage in tests where there are checks for particular factor levels or constructing factor levels.

The ERROR's take many different forms. The simple solution is to change or add the `stringAsFactors=TRUE` to the data.frame call, however maintainers may want to re-evaluate code for potential restructuring or ease of use.

stats::smoothEnds()

A recent change to the `stats::smoothEnds()`, now returns an integer vector when the input is an integer vector. Previously it could return a numeric vector.

Example R 3.6.3:

```
> class(smoothEnds(c(401:403)))  
[1] "integer"  
> class(smoothEnds(c(401:403, 555L)))  
[1] "numeric"
```

Example R 4.0.0:

```
> class(smoothEnds(c(401:403)))  
[1] "integer"  
> class(smoothEnds(c(401:403, 555L)))  
[1] "integer"
```

grid() changes

I don't have much information on this. We were only made aware

I am about to commit some internal changes to 'grid' units (for, in some cases, 100x speed-up of unit operations).

A number of packages have already been fixed to work with these changes, but, according to my testing, the following CRAN packages will still fail R CMD check.

Some of those are cascades ('armada', 'countToFPKM', and 'wilson' from 'ComplexHeatmap' - see below - and 'fingertipscharts' from 'lemon' and 'xpose' is actually a 'ggforce' problem), but all of the other package authors have been notified and several are already working on fixes.

The most serious of those is 'ComplexHeatmap' because it causes multiple follow-on failures, the CRAN ones above and others on BioConductor:

Again, the main package authors have been notified and the "ComplexHeatmap" author is working on an update.

plot generic moved package location

plot() generic from graphics to base

This can cause a variety of ERROR's from packages loading the plot function in their NAMESPACE

```
Error in getGeneric(f, TRUE, envir, package) :
```

```
no generic function found for 'plot'
```

plot generic moved package location

But has also appeared (similar to S3 method declaration ERROR):

```
Error in as.double(y) :  
  cannot coerce type 'S4' to vector of type 'double'
```

Explanation:

"The namespace controls the search strategy for variables used by functions in the package. If not found locally, R searches the package namespace first, then the imports, then the base namespace and then the normal search path." as per <https://cran.r-project.org/doc/manuals/r-devel/R-exts.html#Package-namespaces>:

CRAN and Bioconductor had a few packages that "worked" because the right plot() was found in the normal search path, but now fail because it's calling the one in base instead.

Partial Argument Matching

There is now a more strict check on argument matching with regards to partial argument matching.

```
setGeneric("mycoolfunction", function(object, breaks)  
  standardGeneric("mycoolfunction")  
setMethod("mycoolfunction",  
  signature=c(object="GRanges", break="GRanges"),  
  <code>)
```

Partial argument matching when invoking functions should be avoided too!

Package inputenc ERROR: Invalid UTF8

This ERROR started to appear on tokay2 (windows) in Spring 2020. We are not sure the exact source of the ERROR (change in MiKTeX, Change in R, other?) but the solution is simple:

Please place `\usepackage[utf8]{inputenc}` in the beginning of your Sweave vignette right after the `\documentclass` line.

Package Dependency Issues

- CRAN binaries for 4.0 are not available yet. As binaries become available for mac and windows they will be added to the daily builder automatically. Out of our hands and we will not take extraordinary efforts to find work arounds. When they are available - they will be added.
- Packages have been removed from CRAN. If this occurs Bioconductor packages must alter their package to not utilize the code. We will only support dependencies actively maintained on CRAN or Bioconductor.
- Packages have been removed from Bioconductor. We try to be more aware of orphaned packages and packages that remain broken for extended periods of time. Package deprecation and removal occurs and packages will have to alter to not utilize code from these packages or could potential offer to take over maintenance of broken packages but that would require original maintainers permission. (SNPchip, GenomeGraphs, ...[List of Deprecated Package 3.10](#))

Deprecated Functions

Most ERROR have a message indicating what function should be used instead

- RangedData

Error : RangedData objects are defunct. Please migrate your code to use GRanges or GRangesList objects instead. See IMPORTANT NOTE in ?RangedData

- Normalize

Error: processing vignette 'using_schex.Rmd' failed with diagnostics:
'normalize' is defunct.
Use "normalize,SingleCellExperiment-method" is defunct.
Use 'logNormCounts' instead' instead.

- calculateQCMetrics

Error: processing vignette 'MAST-interopability.Rmd' failed with diagnostics:
'calculateQCMetrics' is defunct.
Use 'perCellQCMetrics' instead.

Useful Links

R developers blog:

<https://developer.r-project.org/Blog/public/>

Troubleshooting Bioconductor Documentation:

<https://bioconductor.org/developers/how-to/troubleshoot-build-report/>