Some Improvements of the Byte-code Compiler Problems in Existing R/C Code

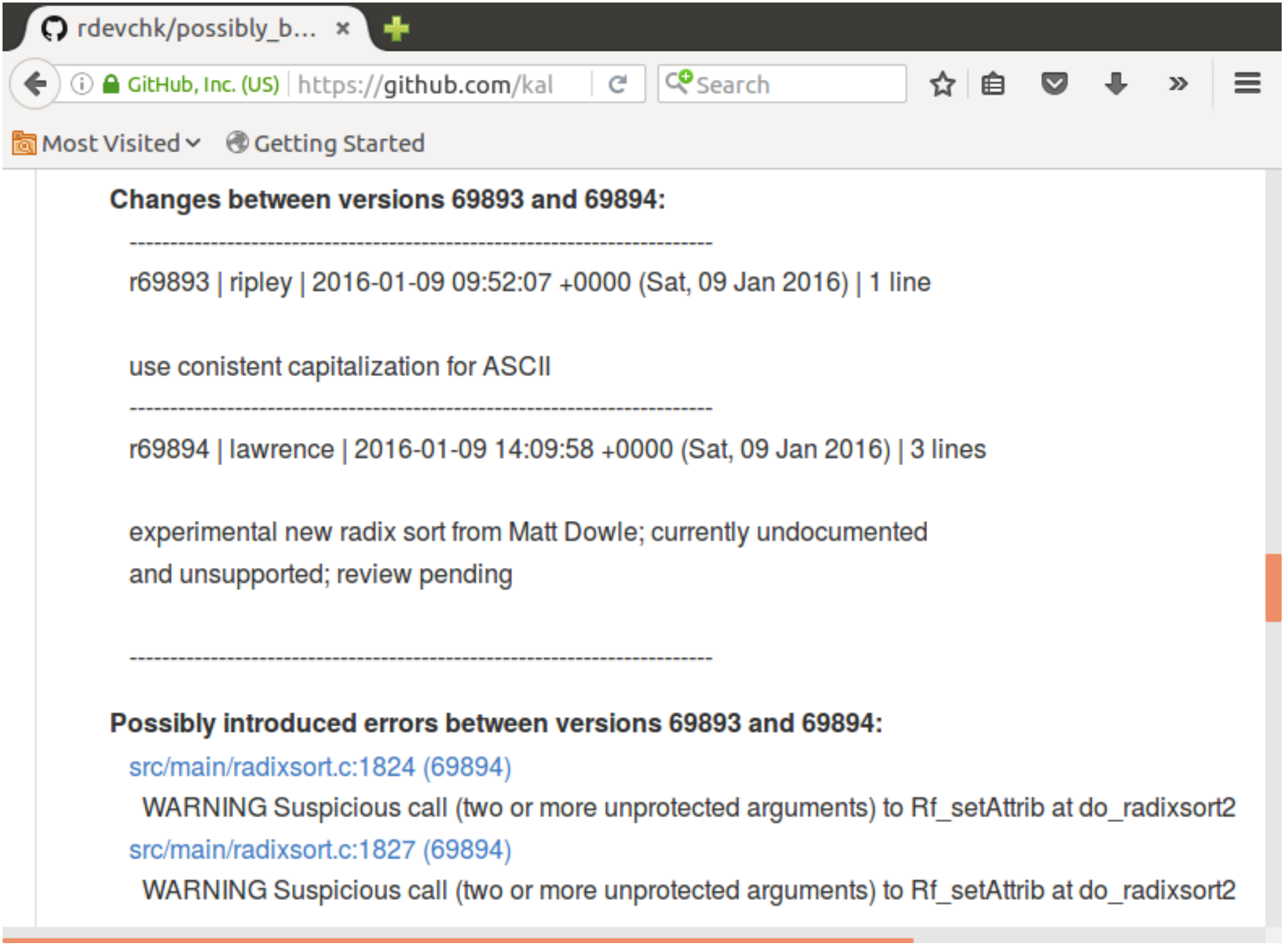
Tomas Kalibera

With Luke Tierney, Jan Vitek

Fighting PROTECT bugs

```
PROTECT(sb = coerceVector(CADR(args), CPLXSXP));
nb = XLENGTH(sb);
if (nb == 0) return allocVector(CPLXSXP, 0);
```

- Rchk, http://github.com/kalibera/rchk
  - Finds possible PROTECT errors in C code of R and packages, using static analysis
  - Improved precision - reduced false alarms
  - Automated install into virtualbox
- Rdevchk, http://github.com/kalibera/rdevchk
  - Newly introduced/fixed PROTECT errors in R-devel
  - Automated, http://github.com/kalibera/rchk-image



Changes between versions 69893 and 69894:

---

r69893 | ripley | 2016-01-09 09:52:07 +0000 (Sat, 09 Jan 2016) | 1 line

use conistent capitalization for ASCII

---

r69894 | lawrence | 2016-01-09 14:09:58 +0000 (Sat, 09 Jan 2016) | 3 lines

experimental new radix sort from Matt Dowle; currently undocumented
and unsupported; review pending

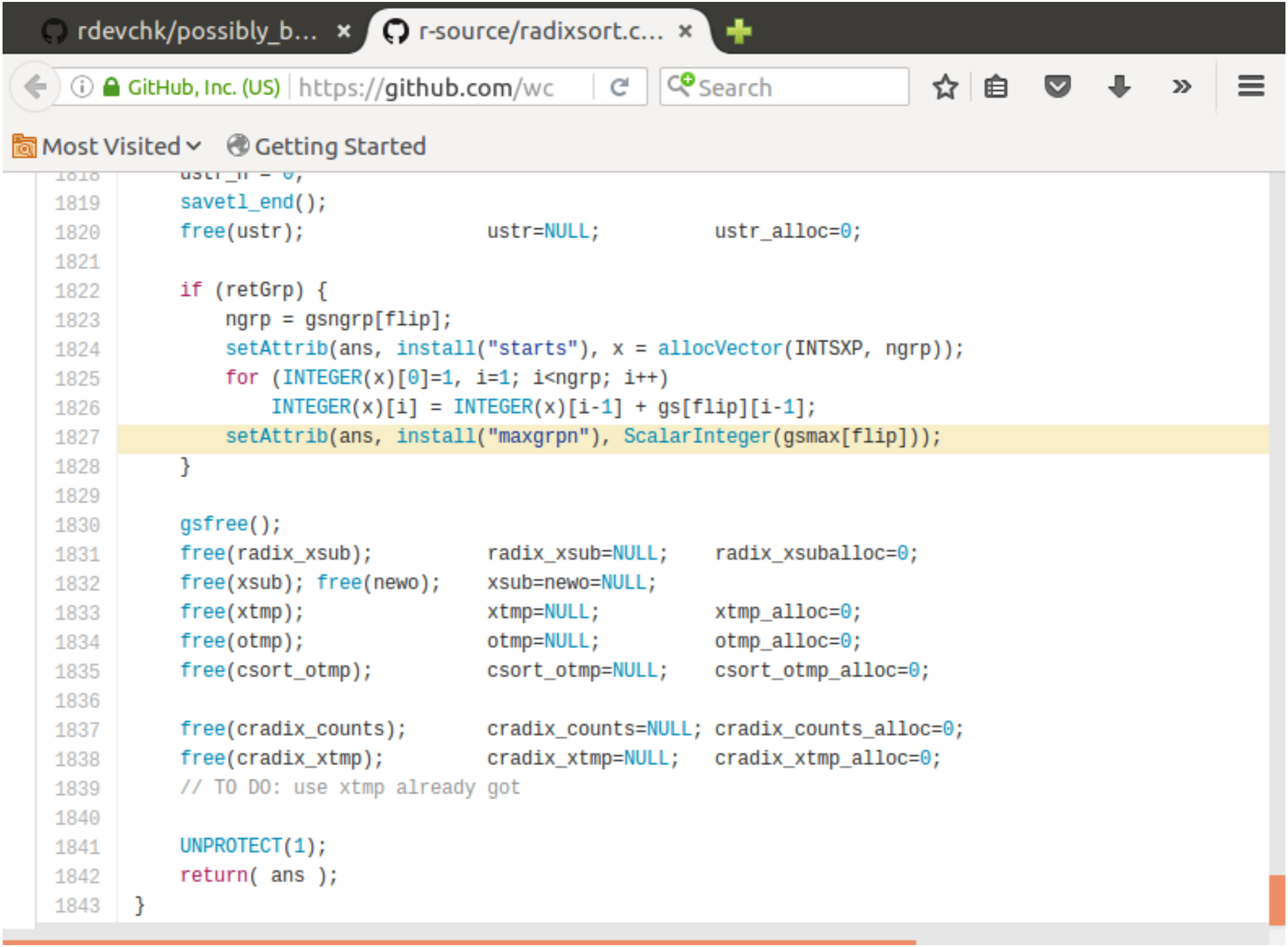
Possibly introduced errors between versions 69893 and 69894:

src/main/radixsort.c:1824 (69894)

WARNING Suspicious call (two or more unprotected arguments) to Rf_setAttrib at do_radixsort2

src/main/radixsort.c:1827 (69894)

WARNING Suspicious call (two or more unprotected arguments) to Rf_setAttrib at do_radixsort2


rdevchk/possibly_b...
r-source/radixsort.c...
GitHub, Inc. (US)
https://github.com/wc
Search
Most Visited
Getting Started
savetl_end();
free(ustr); ustr=NULL; ustr_alloc=0;
if (retGrp) {
ngrp = gsngrp[flip];
setAttrib(ans, install("starts"), x = allocVector(INTSXP, ngrp));
for (INTEGER(x)[0]=1, i=1; i<ngrp; i++)
INTEGER(x)[i] = INTEGER(x)[i-1] + gs[flip][i-1];
setAttrib(ans, install("maxgrpn"), ScalarInteger(gsmax[flip]));
}
gsfree();
free(radix_xsub); radix_xsub=NULL; radix_xsuballoc=0;
free(xsub); free(newo); xsub=newo=NULL;
free(xtmp); xtmp=NULL; xtmp_alloc=0;
free(otmp); otmp=NULL; otmp_alloc=0;
free(csort_otmp); csort_otmp=NULL; csort_otmp_alloc=0;
free(cradix_counts); cradix_counts=NULL; cradix_counts_alloc=0;
free(cradix_xtmp); cradix_xtmp=NULL; cradix_xtmp_alloc=0;
// TO DO: use xtmp already got
UNPROTECT(1);
return(ans);
}

Byte-code compiler/interpreter fixes

```
env R_ENABLE_JIT=3 R
compiler::enableJIT(3)
```

Regression tests now pass with the compiler/JIT enabled.
Package tests: 18 CRAN, 1 BIOC fail due to compiler.

- Source reference/expression tracking (not in yet)
- Robustness improvements
  - Loop compilation, structure of environments
- Corner-case fixes
  - Switch, super-assignment, constant folding, closures in AST
- Runtime fixes, package fixes

Problems in C (package) code: in-place modification of objects

```
iterpc_next_iterations ← function(I)
    if (I$status == -1L)
        ...
    C ← next_combinations(I$status)
    if (is.null(C))
        I$status ← -1L
    C
```

"I" is an environment

"status" changed in place to 0

"status" assigned a constant from pool

In-place modification of "status" to 0 turns all "-1L" constants in the function to 0.
Modifying "I$status" in R code works fine as "I" is an environment.

- Packages with constant pool corruption during tests:
  - CRAN:29, BIOC:0
- Many more packages with in-place changes
- Runtime checking of constants integrity

Packages failing tests due to compiler constants corruption

Packages affected – not necessarily each at fault, the problem is sometimes in a dependency

- CRAN (29)

  eiCompare ei flam gaston GetR GGMselect glinternet gRc HSAUR2 HSAUR MAclinical mboost mets mlr ModelGood ModelMap mombf NHMSAR nlmrt optimx ordinal party pec PSAboot R2BayesX sensR synthpop ucminf vcrpart

- BIOC (0)

```
env R_CHECK_CONSTANTS=5 R_ENABLE_JIT=3 R CMD check package.tar.gz
```

Tested June 28, 2016 with R-devel, JIT level 3, optimize level 2, checking level 5.

```
env R_CHECK_CONSTANTS=5 R_ENABLE_JIT=3 R CMD check gaston_1.4.5.tar.gz
```

```
ERROR: modification of compiler constant of type character,
    length 1
ERROR: the modified value of the constant is:
[1] "2\t1364 …
ERROR: the original value of the constant is:
[1] ""
ERROR: the modified constant is at index 20
ERROR: the modified constant is in this function body:
{
    filename <- path.expand(filename)
    xx <- vcf_open(filename)
…
Function read.vcf in namespace gaston has this body.
ERROR: detected compiler constant(s) modification after .Call
    invocation of function VCF_readLineRaw from library WhopGenome
    (/path/WhopGenome.so).
NOTE: .Call function VCF_readLineRaw modified its argument
    (number 2, type character, length 1)
Fatal error: compiler constants were modified (in .Call?)!
```

Problems in R package code

Re-evaluating a promise

```
“$.dyn” ← function(x, fun)
  e ← parent.frame()
  eval(substitute(unclass(x)$fun),e)
```

```
“$.dyn” ← function(x, fun)
  NextMethod(“$”)
```

Accessing caller frames: expecting library functions use certain number of calls

```
MakeBibLaTeX <- function(docstyle) local({
    docstyle <- get("docstyle", parent.frame(2))
    environment()
})
```

```
caller.name ← function (n = 2)
    as.character(sys.call(-n)[[1]])

subcrt ← function()
    if (identical(caller.name(3),"dPdTtr")) ...
```

Problems in R package code

Eval of unusual code hard to analyze at compile time

```
res<-paste("F<-function(",names(formals(f)),"){(",
             names(formals(f)),"-(",z0,"))*(",body2string(f),
           ")}",
           collapse="",sep="")
eval(parse(text=res))
```

```
F ← function(z) (z-z0)*f(z)
```

```
repeat
    eval(mpi.bcast.cmd(), envir=.GlobalEnv)
```

Summary

- Byte-code compiler is close to full compatibility with existing code
  - Regression tests (check-all) pass, at all optimization levels
  - Most CRAN/BIOC packages pass their tests
- Reaching to package maintainers
  - PROTECT errors
  - In-place modification of objects
  - Bugs, cleanups (restricting behavior)