



R news and tutorials contributed by (573) R bloggers

- [Home](#)
- [About](#)
- [RSS](#)
- [add your blog!](#)
- [R jobs](#) ♦♦♦
- [Contact us](#)

Welcome!

Follow

Here you will find daily
news and tutorials
about R, contributed by
over 573 bloggers.

There are many ways to
follow us -

[By e-mail:](#)

Your e-mail here

18204 readers

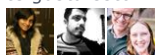
BY FEEDBURNER

[On Facebook:](#)



Me gusta esta página

Sé el primero de tus
amigos en indicar que
te gusta esto.



**If you are an R blogger
yourself** you are invited
to [add your own R
content feed to this site](#)
(**Non-English** R bloggers
should add themselves-
[here](#))

[Jobs for R- users](#)

- [Product Analyst
\(@ Pittsburgh \)](#)
- [Quantitative
Developer/Data
Scientist \(@
Mountain View\)](#)
- [Data Scientist
AWF](#)
- [Network Link
analysis Consultant
\(@ London\)](#)
- [Seeking R](#)

[Programmers for
long term contract
role](#)

Search & Hit Enter

Popular Searches

- [heatmap](#)
- [web scraping](#)
- [maps](#)
- [undefined](#)
- [hadoop](#)
- [shiny](#)
- [twitter](#)
- [boxplot](#)
- [animation](#)
- [ggplot2](#)
- [trading](#)
- [finance](#)
- [time series](#)
- [latex](#)
- [excel](#)
- [ggplot](#)
- [PCA](#)
- [quantmod](#)
- [googlevis](#)
- [eclipse](#)
- [market research](#)
- [rstudio](#)
- [how to import
image file to R](#)
- [tutorial](#)
- [knitr](#)
- [rattle](#)
- [coplot](#)
- [rcmdr](#)
- [gis](#)
- [sweave](#)

Recent Posts

- [STAN trailer
\[PG+53\]](#)
- [Survival Analysis –
2](#)
- [Using Google
Analytics with R](#)
- [R News From JSM
2015](#)
- [Mapping Historic
US Presidential
Election Results](#)
- [Showing a
distribution over
time: how many
summary stats?](#)
- [functional
enrichment for
GTEx paper](#)
- [Milestone: 7000
packages on
CRAN](#)

- [Differences in the network structure of CRAN and BioConductor](#)
- [SWMPPr 2.1.0 on CRAN](#)
- [Visualizing bdsns data using bdvis](#)
- [How Do Auction Values Differ by the Number of Teams in Your League? A Multilevel Model](#)
- [How do you know if your data has signal?](#)
- [Multilevel Models and Political Advertising](#)
- [JSM 2015 \[day #2\]](#)

Other sites

- [SAS blogs](#)
- [Jobs for R-users](#)
- [Statistics of Israel](#)

Using OpenMP-ized C code with R

August 11, 2011

By [mbq](#)

Like Share  0

(This article was first published on [Stack Exchange Stats Blog » R tips&tricks](#), and kindly contributed to [R-bloggers](#))

What is [OpenMP](#)? Basically a standard compiler extension allowing one to easily distribute calculations over multiple processors in a shared-memory manner (this is especially important when dealing with large data — simple separate-process approach usually requires as many copies of the working data as there are threads, and this may easily be an overkill even in overall size, not to mention the time wasted for copying).

The magic of OpenMP is that once you have a C or Fortran code, in most cases you need nothing more than a few additional compiler flags — thus the code remains as portable and as readable as before the modification. And is usually just nice and simple, not counting few common parallelism traps and some quirks related to the fact we want it to work with R.

In this post I don't want to make an OMP tutorial (the web is full of them), rather show how to use it with R. Thus, I'll use a toy example: a function that calculates the cumulative sum in an **unnecessary demanding** way:

```
#include <R.h>
#include <Rinternals.h>
SEXP dumbCumsum(SEXP a){
  SEXP ans;
  PROTECT(a=coerceVector(a, REALSXP));
  PROTECT(ans=allocVector(REALSXP, length(a)));
  double* Ans=REAL(ans);
  double* A=REAL(a);
  for(int e=0;e<length(a);e++){
    Ans[e]=0.;
    for(int ee=0;ee<e+1;ee++)
```

```

    Ans[e]+=A[ee];
}
UNPROTECT(2);
return(ans);
}

```

There is only one for loop responsible for most computational time and no race conditions, thus the OMP-ized version will look like this:

```

#include <R.h>
#include <Rinternals.h>
#include <omp.h>
SEXP dumbCumsum(SEXP a){
  SEXP ans;
  PROTECT(a=coerceVector(a,REALSXP));
  PROTECT(ans=allocVector(REALSXP,length(a)));
  double* Ans=REAL(ans);
  double* A=REAL(a);
  #pragma omp parallel for
  for(int e=0;e<length(a);e++){
    Ans[e]=0.;
    for(int ee=0;ee<e+1;ee++)
      Ans[e]+=A[ee];
  }
  UNPROTECT(2);
  return(ans);
}

```

Time for R-specific improvements; first of all, it is good to give the user an option to select number of cores to use (for instance he has 16 cores and want to use first 8 for one job and next 8 for something else — without such option he would have to stick to sequential execution); yet it is also nice to have some simple option to use the full capabilities of the system. To this end we will give our function an appropriate argument and use OMP functions to comply with it:

```

#include <R.h>
#include <Rinternals.h>
#include <omp.h>
SEXP dumbCumsum(SEXP a,SEXP reqCores){
  //Set the number of threads
  PROTECT(reqCores=coerceVector(reqCores,INTSXP));
  int useCores=INTEGER(reqCores)[0];
  int haveCores=omp_get_num_procs();
  if(useCores<=0 || useCores>haveCores) useCores=haveCores;
  omp_set_num_threads(useCores);
  //Do the job
  SEXP ans;
  PROTECT(a=coerceVector(a,REALSXP));
  PROTECT(ans=allocVector(REALSXP,length(a)));
  double* Ans=REAL(ans);
  double* A=REAL(a);
  #pragma omp parallel for
  for(int e=0;e<length(a);e++){
    Ans[e]=0.;
    for(int ee=0;ee<e+1;ee++)
      Ans[e]+=A[ee];
  }
  UNPROTECT(3);
  return(ans);
}

```

This code will also ensure that the number of threads won't be larger than the number of physical cores; doing this gives no speedup and comes with a performance loss caused by OS scheduler.

Finally, time to resolve small quirk — R has some code to guard the C call stack from overflows, which is obviously not thread-aware and thus have a tendency to panic and screw the whole R session up when running parallel code. To this end we need to disable it using the trick featured in R-ext. First, we include Rinterface to have an access to the variable with stack limit

```

#define CSTACK_DEFNS 7
#include "Rinterface.h"

```

and then set it to almost infinity in the code

```
R_CStackLimit=(uintptr_t)-1;
```

Voilà, the stack is now unprotected — the work with R just become a bit more dangerous, but we can run parallel stuff without strange problems. The full code looks like this:

```
#include <R.h>
#include <Rinternals.h>
#include <omp.h>
#define CSTACK_DEFNS 7
#include "Rinterface.h"
SEXP dumbCumsum(SEXP a,SEXP reqCores){
  R_CStackLimit=(uintptr_t)-1;
  //Set the number of threads
  PROTECT(reqCores=coerceVector(reqCores,INTSXP));
  int useCores=INTEGER(reqCores)[0];
  int haveCores=omp_get_num_procs();
  if(useCores<=0 || useCores>haveCores) useCores=haveCores;
  omp_set_num_threads(useCores);
  //Do the job
  SEXP ans;
  PROTECT(a=coerceVector(a,REALSXP));
  PROTECT(ans=allocVector(REALSXP,length(a)));
  double* Ans=REAL(ans);
  double* A=REAL(a);
  #pragma omp parallel for
  for(int e=0;e<length(a);e++){
    Ans[e]=0.;
    for(int ee=0;ee<e+1;ee++){
      Ans[e]+=A[ee];
    }
  }
  UNPROTECT(3);
  return(ans);
}
```

Now, time to make sure that R will compile our function with OMP support (and thus make it parallel). To this end, we create a Makevars file (in the src in case of package and in code directory when using dangling object files) with a following contents (for GCC):

```
PKG_CFLAGS=-fopenmp
PKG_LIBS=-lgomp
```

The first line will trigger parsing OMP pragmas, the latter will link the OMP library with `omp_*` functions.

We are ready to test our example:

```
$ R CMD SHLIB omp_sample.c
$ R
> dyn.load('omp_sample.so')
> .Call('dumbCumsum',runif(100000),0L)
```

Try to run sum system monitor (like `htop` or GUI one that comes with your desktop environment) and watch your powerful CPU being finally fully utilized (-;

To close with an optimistic aspect, few words about limitations. Don't even try to run any R code or use features like random number generation or `Ralloc` inside parallelized blocks — R engine is not thread-safe and thus this will end in a more or less spectacular failure.

Plus of course all issues of parallel programming and OMP itself also apply — but that is a different story.

Related



[Parallel Random Number Generation](#)

[Comments: 12](#)
[OpenMP Tutorial with R Interface](#)
In "R blog posts"

[Introducing RcppParallel: Getting R and C++ to work \(some more\) in parallel](#)
A common theme over the last few decades:

wing IRNG
In 'R. bloggers''

was that we could
afford to simply sit
back and let computer
In R. bloggers''

0

Like

Share

To leave a comment for the author, please follow the link and comment on his blog:
[Stack Exchange Stats Blog » R tips&tricks.](#)

R-bloggers.com offers [daily e-mail updates](#) about [R](#) news and [tutorials](#) on topics such as: visualization ([ggplot2](#), [Boxplots](#), [maps](#), [animation](#)), programming ([RStudio](#), [Sweave](#), [LaTeX](#), [SQL](#), [Eclipse](#), [git](#), [hadoop](#), [Web Scraping](#)) statistics ([regression](#), [PCA](#), [time series](#), [trading](#)) and more...

If you got this far, why not [subscribe for updates](#) from the site?
Choose your flavor: [e-mail](#), [twitter](#), [RSS](#), or [facebook](#)...

Like Share

0

Comments are closed.

Top 3 Posts from the past 2 days

- [Scatterplots](#)
- [In-depth introduction to machine learning in 15 hours of expert videos](#)
- [Installing R packages](#)

Search & Hit Enter

Top 9 articles of the week

1. [Scatterplots](#)
2. [In-depth introduction to machine learning in 15 hours of expert videos](#)
3. [Installing R packages](#)
4. [Using apply, sapply, lapply in R](#)
5. [Turning your R \(or Python\) models into APIs](#)
6. [Basics of Histograms](#)
7. [Differences in the network structure of CRAN and BioConductor](#)
8. [Read Excel files from R](#)
9. [Adding a legend to a plot](#)

Sponsors



Highland Statistics Ltd

Zero Inflated Models & GLMM

Beginner's Guide to GAM

Beginner's Guide to GLM & GLMM

Beginner's Guide to GAMM



R training

R consulting

QUANTIDE
knowledge from data

R Studio
open source & enterprise ready
professional software for R

REVOLUTION
ANALYTICS

R for the Enterprise

www.revolutionanalytics.com

Werden Sie zum Expe[R]ten mit der
R-Akademie von



Beratung | Software
Training | Lösungen



plotly online R graphing

[Plotly: collaborative, publication-quality graphing](https://plot.ly)

STATISTICS
VIEWS

Bringing Statistics Together



NYC DATA SCIENCE
ACADEMY

Sep 21 - Dec 11 | FULL TIME PROGRAM

12 - WEEK DATA SCIENCE BOOTCAMP

Hands-on R, Python & Hadoop Training

Mentorship from Top Data Scientists

Job Placement at Our 300+ Hiring Firms

Scholarship & Financial Aid Available

DEADLINE FOR APPLICATION: August 21st



DataCamp
Learn R Interactively



#ODSC

SAN FRANCISCO

NOV. 14TH - 15TH

www.opendatascicon.com

LEARN MORE

Save **25%** on
R Books



Use
Promo
Code
CZP40

Plus Free
Shipping

**Data Analysis
Modeling**
Cloud Solution

dotplot

Free Sign Up!



STATWORX

Consulting
Schulung
Data Mining

Mehr erfahren



Try the **FASTEST ML**
for **R**

Click for a Free Trial



Become a
Certified Data Scientist

ENROLL NOW

simpl|learn



EARN YOUR MASTER'S IN
DATA SCIENCE **online.**

Berkeley
UNIVERSITY OF CALIFORNIA

LEARN MORE



Search & Hit Enter

[Full list of contributing R-bloggers](#)

[R-bloggers](#) was founded by [Tal Galili](#), with gratitude to the [R](#) community.

Is powered by [WordPress](#) using a [bavotasan.com](#) design.

Copyright © 2015 [R-bloggers](#). All Rights Reserved. [Terms and Conditions](#) for this website