`apt-get install cran bioc`: On automated builds of 1700 R packages for Debian

Dirk Eddelbuettel*
Debian

David Vernazobres
Universität Münster

Albrecht Gebhardt
Universität Klagenfurt

Steffen Möller
Universität Lübeck

May 2007

Abstract

Within the world of the R system, language and environment, the CRAN and BioConductor archives have achieved remarkable success in attracting a consistent inflow of new packages of high quality contributions and extensions to the R system.

At the same time, the Debian distribution (and its derivatives such as Ubuntu) has continued to make it easier for users to obtain a consistent and complete software installation. In Debian's case, this has resulted in an unprecedented ten installable architectures. For Ubuntu, a focus on easier installation and added polish means that the 'barriers to entry' for new users have been lowered, which has resulted in increased market- and mind share for Debian and Ubuntu.

This paper presents an effort to bring the R package repositories and the Debian Linux distribution together. This provides a unique statistical environment: essentially all CRAN, BioConductor and Omegahat packages can be installed automatically onto Debian (or Ubuntu) from pre-built binary packages with a single command. Our initial reference builds cover well over 1700 packages taken from the CRAN, BioConductor and Omegahat repositories.

1 Introduction

This paper[1] presents a novel approach to the automatic generation over 1700 binary R packages that can be reliably installed and used on Debian (or Ubuntu). In this section, we briefly discuss the main components that form the foundation of our work: the R environment and language for statistics, the CRAN and BioConductor R package repositories, and the Debian Linux distribution.

*Corresponding author – Email edd@debian.org

[1] Möller et al. (2007) present related work in the context of the semantic web for computational clusters.

R The R system (R Development Core Team, 2007) has become a de-facto standard for modern statistical applications and research. One can think of several key qualities and attributes of the R system that have driven this success. However, in this discussion, we want to emphasize one particular aspect: the Comprehensive R Archive Network (CRAN)[2], BioConductor (Gentleman et al., 2004) and Omegahat (Temple Lang, 2000) repositories.

CRAN CRAN offers an essentially open, but at the same time rigorously quality-controlled, repository to which 'community members' at large can submit packages provided they pass the R package check.[3] Given this open nature, the number of CRAN packages has grown dramatically: from a start of a few dozen packages to over 200 in early 2003, over 400 in the summer of 2004, over 550 in the summer of 2005, almost 750 in the spring of 2006 up to over one thousand in 2007.[4] CRAN has a large network of mirrors across the globe.

BioConductor During the same time period, the BioConductor Project (Gentleman et al., 2004) was started. It too has experienced rather remarkable growth to become one of the key technologies in bioinformatics and computational biology. It hosts its own repository and mirrors. Its upload policy differs slightly from CRAN and corresponds more closely to a peer-review approach employed by academic journals, or the Boost.Org C++ software project.

At this point, CRAN and BioConductor (as well as the experimental Omegahat repository) provide an unparalleled source of high quality packages for R. Moreover, these packages can be installed and used in an entirely standardized manner which really suggests automation.

Debian Debian is a volunteer-driven[5] Linux distribution. Debian and its derivatives like Ubuntu employ one of the most advanced package management systems. Debian packages can be installed using tools like `dpkg` or with the more advanced `apt-get` program. The latter will automatically resolve dependencies between packages by installing missing components along with the target packages in question, but refrain from doing so if incompatible versions are detected. This advanced technology, together with a reputation for quality and a large base of over 18,000 packages have made Debian (and Ubuntu) a popular choice of Linux installations.[6]

[2]The CRAN name is a tongue-in-cheek reference to the predecessors 'CTAN' (for the TEX community, hence the 'T') and 'CPAN' for the Perl programming language.

[3]Any given R package can be checked using the `R CMD check` command which utilises a wide variety of hard-coded tests that ensure (rather high) minimum standards for the package. It is worth noting that the tests work both ways: core R development is also tested against the corpus of CRAN packages to minimize unwarranted code regressions.

[4]This count was computed using copies of the page `http://cran.r-project.org/src/contrib/checkSummary.html` at the 'Wayback Machine' of the Internet Archive at `http://web.archive.org/` and is meant to be indicative-only.

[5]Several members do have full-time jobs that permit, or in some cases even focus on, Debian work.

[6]While 'market share' is difficult to measuer, the long-running Linux Counter (at `http://counter.li.org/reports/machines.php`) shows a percentage of 20% for Debian with an additional 11% for Ubuntu/Kubuntu giving over 30% to Debian whereas Red Hat and Fedora Core add up to around 15%.

Debian also has an automated build system that creates its binary packages for all ten release 'architectures' ranging from small-scale embedded architectures (like arm) to workstations (sparc, ia64) and even mainframes (s390).

Turning R packages into Debian components The preceding paragraphs outlined a few key aspects of the R software system and its repositories. The Debian distribution provides a natural mechanism for building and distributing binary components from CRAN, BioConductor and Omegahat.

The rest of the paper is organised as follows. The next section presents the motivation behind our initiative. Section 4 describes the technology and approach used. Section 5 raises a number of open issues before section 5 concludes.

2 Motivation

The motivation for turning R source packages into directly installable Debian (or Ubuntu) packages was first laid out in Bates and Eddelbuettel (2003) and Bates et al. (2004). A slightly updated list of reasons follows:

Dependencies: automated and reliable resolution of dependencies should prevent compile-time errors (such as 'header file not found') or run-time errors ('library not found') from frustrating our users, and permit them to concentrate on *applying* R and CRAN, BioConductor or Omegahat packages to their work, rather than re-building the components;

Convenience: installing pre-built packages via `apt-get` (or its front-ends as e.g. `aptitude` or `wajig`) is orders of magnitude easier than building from source, in particular for the important cases of a) more complicated packages requiring lesser known, or difficult to build, components or libraries to be installed, or b) in the case of less experienced administrators;

Quality control: build daemons creating packages from a minimal setup ensure that all required components are listed, adding an additional layer of assurance to the build process and generally preventing surprises; a related advantage is the use of full archive-build regressions to test major new releases of the key libraries or the compiler and linker toolchain;

Scalability: building a binary packages once and subsequently installing it multiple times over is very attractive for larger scale computer farms, clusters and grids; or the simpler case of installing to both home and office machines, or onto installations of co-workers or students;

Common platform: building on the Debian infrastructure carries over to Ubuntu and other derivatives; this provides a wider choice of installation methods and allows us to plug into the relative strengths of either distribution;

Different architectures: once the framework is in place, Debian packages can be made available for 32- or 64-bit Intel/AMD systems, PowerPCs, UltraSPARCs, ... taking advantage of the over ten architectures on which Debian is available, as well as the three Ubuntu variants;

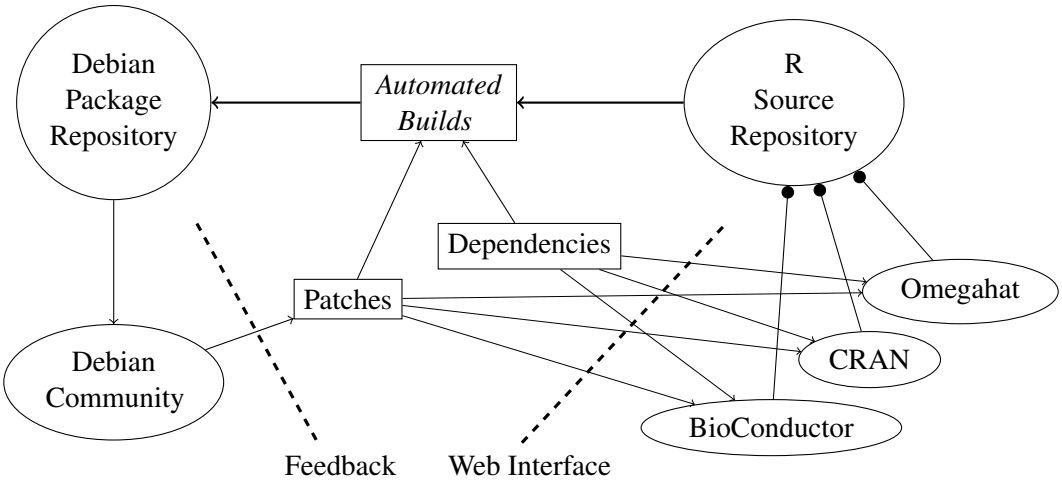

Figure 1: **Interaction of the Debian and R communities for the dissemination of packages.** *The R community provides the R source packages which are automatically transformed into binary packages for the Debian Linux distribution. Additional input is provided by a database of build dependencies, with the possibility of providing arbitrary patches. Mailing lists and the use of the CVS repository help in the collection of community feedback. Changes are presented via a web interface.*

Audience: Debian and Ubuntu installations, and specialised custom distributions such as Quantian (Eddelbuettel, 2003; Lawrence and Eddelbuettel, 2005), permit extending the reach of R and its packages even further.

Figure 1 sketches the interaction between the R and Debian communities. The next section provides some brief details about how the packages are being built.

3 Technology and Approach

This section sketches the current implementation, and its usage.[7]

The code is written in a combination of Bash and Perl scripts. A substantial amount of the underlying technology (package relations, automated builds) comes directly from Debian. A core component of the functionality we added consists in the mapping between the repository and package information to the corresponding package information for Debian so that R package dependencies can be resolved dynamically.

In essence, the workflow is as follows:

1. Check out the newest version of the tools from the version control system of the project:

```
cvs -z3 -d:pserver:anonymous@cvs.alioth.debian.org:/cvsroot/pkg-bioc
    co tools
```

[7]This process is described in more detail at http://wiki.debian.org/AliothPkgBioc.

2. Install the required Debian packages

   ```
   apt-get install `perl ./cran2deb.pl --listRequiredDebianPackages`
   ```

3. Prepare directories, links, package cache information as well as the lists of packages already available in Debian:

   ```
   sh ./r_pkg_prepare.sh --create-all --us
   ```

   which uses the North American mirrors via the '`--us`' switch.

4. Prepare the mirror and the automated build environment:

   ```
   ./r_pkg_update.pl --doupdate --dopbuilderupdate
   ```

5. Change into the corresponding repository directory and start the build:

   ```
   cd ../cran && ./cran2deb.pl
   ```

   which will

   (a) cycle through a graph structure to build 'leaf' packages (that do not depend on other packages) prior to 'node' packages which require other packages;

   (b) augment the build environment as needed, mapping R 'Depends' information from DESCRIPTION files into Debian package names;

   (c) where needed, resolve cross-repository dependencies between CRAN and BioConductor;

   (d) log stateful information in a relational database.

   This main step can fail as some inter-package dependencies may not be completely resolved. Restarting the script `cran2deb.pl` a few times should help in building most if not all of the packages.

6. Upload the packages to the (not-yet announced) repository.

The next section discusses some open issues.

4 Open Issues

At present, over 1700 packages from CRAN, BioConductor and Omegahat have been built using the tools described in the previous section. Several open issues remain to be addressed.

Public server, storage, bandwidth needed An experimental server is being set up with subdirectories for each of the three main repositories. However, as of mid-May 2007, the server has not yet been publically announced. The project would profit greatly from having a mirror and backup host. It is probably not feasible to force all 1700 packages directly into Debian. The distribution of packages directly via the main Debian distribution may be consideration for packages that a) are among the most popular and widely used packages, and/or b) that are rather more difficult to build and install.

Better / Automated Depends The packaging process for Debian is completely automated using the `R CMD build` directive. It should be noted that the process outlined in the previous section is strongly dependent on the contents of the (required) DESCRIPTION file of the R source package. Many of the build dependencies that are declared in these files were found to be insufficient for automated package compilation.[8] One obvious case is where packages require external libraries. A small database was developed to keep track of the mapping between the fields in the DESCRIPTION file, and the package names used by Debian.[9] For a considerable number of packages, some required R packages were found missing during the build process or required as extra functionality for the `R CMD check` step. In order to guarantee the completeness of build instructions, the build process is performed in clean chroot environments using Debian's `pbuilder` tool[10]. In that process, all required (C or R) libraries are added for each individual build, and the chroot is reset between packages. These automated builds have already been performed on the x86 and amd64 platforms.

Package descriptions For the Debian packages being created, a well-written description is an important source of information for users unfamiliar with the package. Our tools retrieve the matching content from the DESCRIPTION file. As some of the CRAN or BioConductor packages are aimed at domain specialists, the wording of the description may be rather sparse given that the main focus of documentation system is on the R documentation system (from which on-line, web and print versions can be generated). As the Debian packages are aimed at larger audiences of non-specialists, we would prefer more extensive descriptive text. We are looking for ways to communicate our (often manual) changes to the descriptions back to the upstream developers. For dependencies, which are stored within the `cran2deb.pl` script, a routine was added to generate a web page that can be disseminated. Other non-critical changes to the DESCRIPTION files or other parts of the source code are collected as patches in a separate directory of the publicly accessible CVS tree of the pkg-bioc alioth project.

[8] The non-interactive minimal chroot-based 'pbuilder' environment tends to reveal such missing Depends. Similar observations have been made in the course of packaging the existing R/CRAN Debian packages.

[9] A related effort was once started for Gentoo Linux, with a suitably cross-platform design and a 'grammar' to map correctly between distributions. But the code archive for the project at `http://code.google.com/p/cran2ebuild/` reveals that this project may have stalled.

[10] See `http://www.netfort.gr.jp/~dancer/software/pbuilder.html`

Information reuse The infrastructure that is being prepared should provide enough material to allow other projects to build on top of it. For example, a graphical user interface that allows one to peruse a repository, call up package information, possibly show package dependencies and reverse dependencies as well as changelogs and release histories could take advantage of the database which contains the package metadata.

Dual package management conflicts An unresolved open issue centers around the fact that R's internal package management system is unaware of the outer Debian package management system. This could potentially result in installations where versions are mismatched, or where packages are only partially installed. Once our infrastructure has proven to be viable, it may be worth considering the alteration of R's internal system to reflect the 'outer' packagement system. Changes could be made incrementally between the semi-annual R releases.

Integration with CRAN Several of the central computers of the CRAN network run on Debian. These machines already undertake numerous quality assurrance tests. If we can demonstrate a history of automated builds with only minimal human attention, it may be worth discussing if automated Debian (and / or Ubuntu) package builds can be added to the workload of these CRAN servers as this would permit direct distribution of Debian / Ubuntu binaries via CRAN.

5 Conclusion

This paper present some first results from an ongoing effort to provide Debian packages for all of the CRAN, BioConductor and Omegahat repositories for the R systems.

The system that has been developed and implemented by the authors of this paper permits a semi-automated build of almost all packages from the three repositories. Building on the success on Debian and Ubuntu, and leveraging their build infrastructure permits us to 'stand on the shoulder of giants' and push the envelope a little further. Providing 1700 automatically installable binary packages that extend the R system may be a first step in introducing even more users to R, and to the wealth of packages provided through the R repositories.

Acknowledgements

The authors would like to thank all contributors to the pkg-bioc project, the R Core team, and of course all authors of R packages.

References

Douglas Bates and Dirk Eddelbuettel. Debian R policy: Draft proposal, 2003. URL http://lists.debian.org/debian-devel-0312/msg02332.html.

Douglas Bates, Dirk Eddelbuettel, and Albrecht Gebhardt. Using R on Debian: Past, present, and future, 2004. URL http://www.ci.tuwien.ac.at/Conferences/useR-2004/abstracts/Eddelbuettel+Bates+Gebhardt.pdf.

Dirk Eddelbuettel. Quantian: A scientific computing environment. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, 2003. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Pro\discretionary{-}{}{}ceed\discretionary{-}{}{}ings/Eddelbuettel.pdf. ISSN 1609-395X.

Robert C Gentleman, Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony J. Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Y. H. Yang, and Jianhua Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004. URL http://genomebiology.com/2004/5/10/R80.

Christopher N. Lawrence and Dirk Eddelbuettel. Quantian: A comprehensive statistical computing environment. *The Political Methodologist*, 13(2):10–13, Fall 2005. URL http://polmeth.wustl.edu/tpm/tpm_v13_n2.pdf.

Steffen Möller, Daniel Bayer, David Vernazobres, Albrecht Gebhardt, and Dirk Eddelbuettel. Scientific grid computing via community-controlled autobuilding of software packages across architectures. In *Proceedings of NETTAB 2007, A Semantic Web for Bioinformatics: Goals, Tools, Systems, Applications*, Pisa, Italy, June 12-14th 2007. URL http://www.nettab.org.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. URL http://www.R-project.org. ISBN 3-900051-07-0.

Duncan Temple Lang. The omegahat environment: New possibilities for statistical computing. *Journal of Computational and Graphical Statistics*, 9(3), September 2000. URL http://www.amstat.org/publications/JCGS/index.cfm?fuseaction=Lang2000.