RPackUtils: R Package Dependencies Manager and Bioconductor & CRAN Mirroring Tool

Stephane Cano, Sylvain Gubian

PMI R&D, Philip Morris Products S.A., Quai Jeanrenaud 5, CH-2000 Neuchâtel, Switzerland

Abstract

The installation and maintenance of R packages from public repositories, such as the Comprehensive R Archive Network (CRAN) or Bioconductor, is a process made cumbersome by the need to manage package dependencies. Typically, this forces users to download the versions of the package that are currently available and potentially experience a trial-and-error cycle of installing and updating all dependencies. The reproducibility of an R installation, and therefore of the R code itself, is not guaranteed by the use of standard, out-of-the-box tools.

RPackUtils is an R package dependencies manager developed in Python with reproducibility in mind. **RPackUtils** can manage several public and private repositories (e.g., Artifactory), any existing R environment, or any local file system folder internally as well as install packages from all of them and mirror the main public repositories, CRAN and Bioconductor, to a specific snapshot in time or release.

Introduction

The command line tools of **RPackUtils** enable the following use cases:

- Mirror past and current CRAN snapshots published on MRAN
- Mirror past and current Bioconductor versions
- Clone existing R environments
- Install R packages with dependencies from different repository types (*providers*)
- Create R package dependency graphs

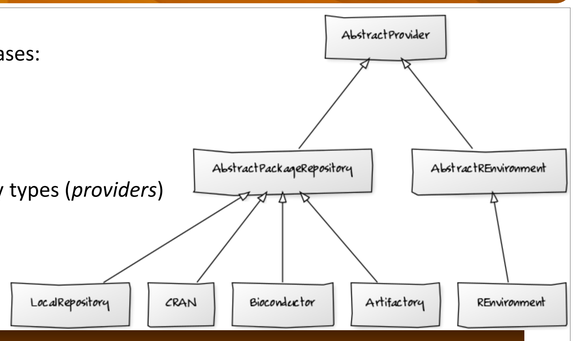


Table 1 describes the main commands available in the shell.

Command	Purpose
rpackbioc	Query the Bioconductor repository for available releases
rpackmran	Query the MRAN repository for available snapshots
rpackq	Search across repositories for a package or a list of packages
rpacki	Install R packages with resolved dependencies
rpackc	Install R packages based on an existing environments (clone)
rpackd	Download R packages and resolve dependencies
rpackm	Download R packages from a specified repository (CRAN, Bioconductor) and upload them to Artifactory (mirror)
rpackg	Generate a dependencies graph

Table 1: *List of available commands for RPackUtils (CLI).*

Comparison with Packrat

Both tools are meant for different use cases, the typical workflow of **Packrat** occurs inside user workspaces, whereas in **RPackUtils**, it occurs in entire R environments for all users.

Packrat is initialized inside an R project in a user's workspace. It maintains a library of packages for each **Packrat** -enabled project. It can install, remove, and snapshot dependencies. For collaboration and sharing, *bundle()* and *unbundle()* functions are provided. The creation of a bundle will actually fetch all necessary sources and make them available along with the project source in an archive.

RPackUtils is more administrator-oriented in the sense that it acts on entire R environments. It can mirror the major R repositories (CRAN and Bioconductor) even for old snapshots, clone R environments, and install packages.

Table 2 summarizes the main features of both tools.

	Packrat	RPackUtils
User workspace management	✓	✗
R environment management	✗	✓
Artifactory repository support	✗	✓
Local repository support	✓	✓
Clone an existing R environment	✗	✓
"Isolation" per package (a **Packrat** project has its own package library)	✓	✗
Reproductibility guaranteed	✓	✓
Resolve dependencies while installing an R package	✗	✓
Repository mirroring	✗	✓
License class filtering	✗	✓

Table 2: *Packrat and RPackUtils main features.*

Links

RPackUtils on github:
https://github.com/pmpsa-hpc/RPackUtils

R package repositories: CRAN MRAN Bioconductor

Packrat:
https://rstudio.github.io/packrat/

R releases and main package repositories

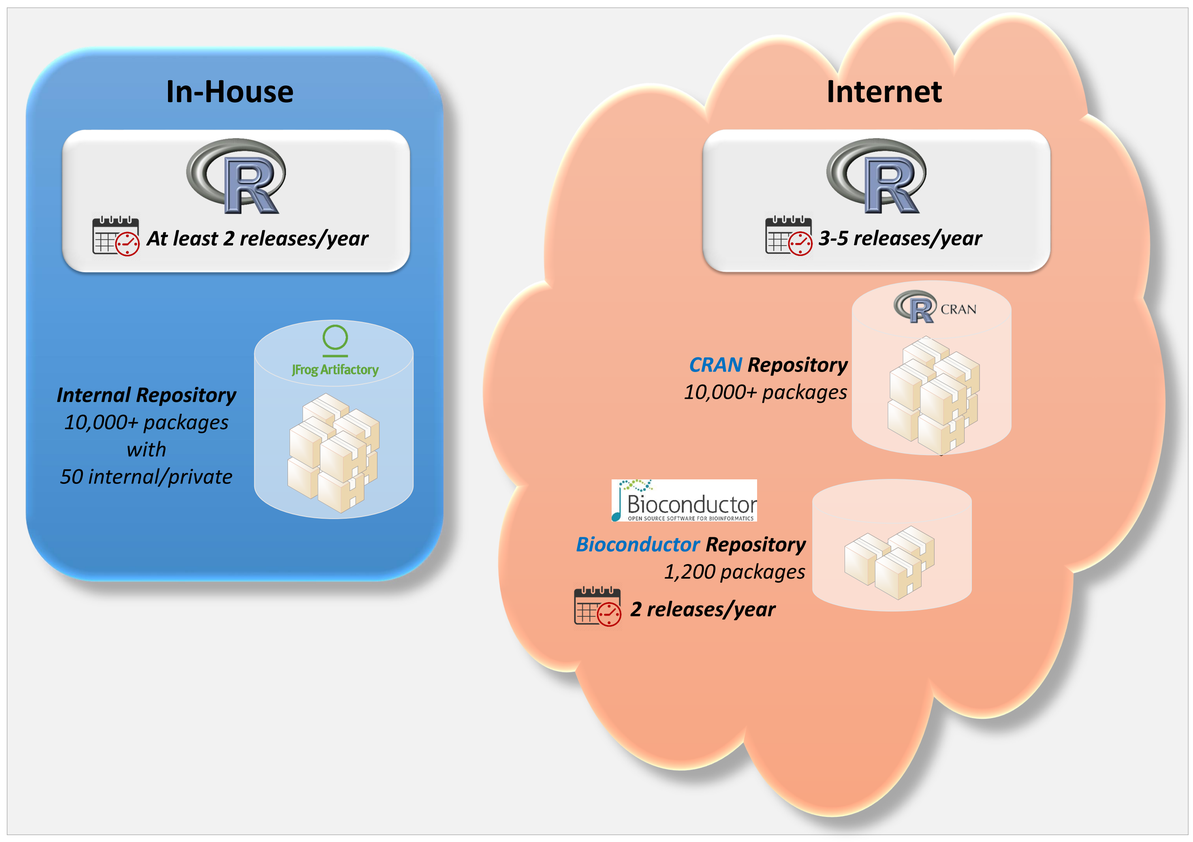


Figure 1: *Our in-house R environment and packages (for a given version of R) versus the outside situation.*

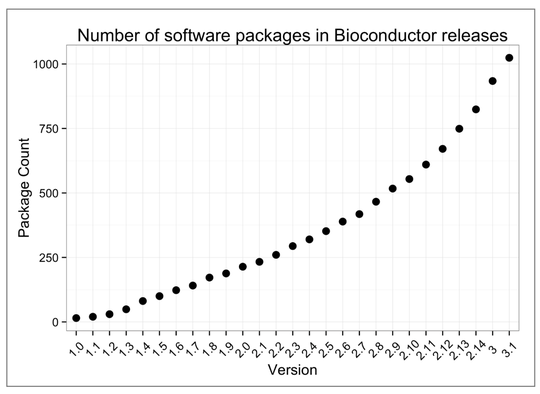


Figure 2: *Number of packages in Bioconductor.*

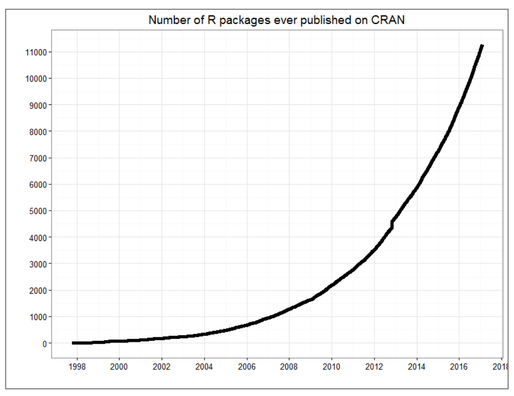
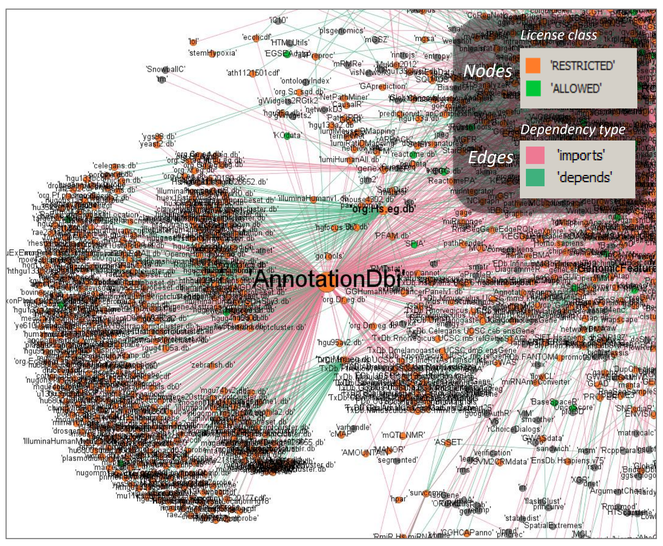


Figure 3: *Number of packages in CRAN.*

Dependencies graph



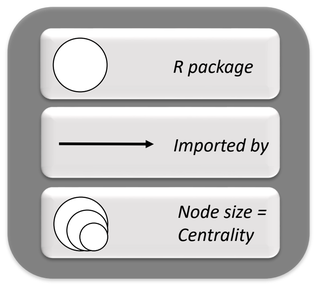
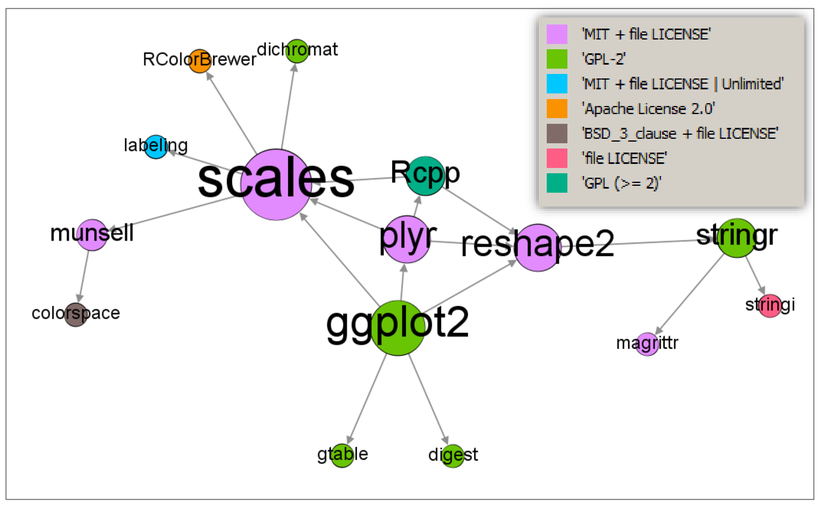


Figure 4: *Dependency graph for Bioconductor with license classes (detail).*

Figure 5: ggplot2 *dependencies with licenses.*



Bastian M., Heymann S., Jacomy M. (2009).
Gephi: an open source software for exploring and manipulating networks.
International AAAI Conference on Weblogs and Social Media.

PMI SCIENCE
PHILIP MORRIS INTERNATIONAL

Competing Financial Interest

The research described in this poster was sponsored by Philip Morris International