

# How to forge a BSgenome data package

Hervé Pagès  
Gentleman Lab  
Fred Hutchinson Cancer Research Center  
Seattle, WA

April 11, 2014

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>How to forge a BSgenome data package with bare sequences</b>	<b>3</b>
2.1	Obtain and prepare the sequence data . . . . .	3
2.1.1	What do I need? . . . . .	3
2.1.2	An example . . . . .	4
2.1.3	The <code>&lt;seqs_srcdir&gt;</code> folder . . . . .	4
2.2	Prepare the BSgenome data package seed file . . . . .	4
2.2.1	Overview . . . . .	4
2.2.2	Standard DESCRIPTION fields . . . . .	5
2.2.3	Non-standard DESCRIPTION fields . . . . .	5
2.2.4	Other fields . . . . .	6
2.2.5	An example . . . . .	6
2.3	Forge the <i>target package</i> . . . . .	7
<b>3</b>	<b>How to forge a BSgenome data package with masked sequences</b>	<b>8</b>
3.1	Obtain and prepare the mask data . . . . .	8
3.1.1	What do I need? . . . . .	8
3.1.2	An example . . . . .	9
3.1.3	The <code>&lt;masks_srcdir&gt;</code> folder . . . . .	9

3.2	Prepare the seed file for the masked BSgenome data package . . . . .	10
3.2.1	Overview . . . . .	10
3.2.2	Standard DESCRIPTION fields . . . . .	10
3.2.3	Non-standard DESCRIPTION fields . . . . .	10
3.2.4	Other fields . . . . .	10
3.2.5	An example . . . . .	11
3.3	Forge the <i>2nd target package</i> . . . . .	12
<b>4</b>	<b>Session information</b>	<b>12</b>

# 1 Introduction

IMPORTANT NOTE: Starting with Bioconductor 2.14, the `forgeBSgenomeDataPkg` function doesn't handle masks anymore i.e. now it produces a *BSgenome data package* that contains only the bare sequences. A new function `forgeMaskedBSgenomeDataPkg` has been added to forge a *BSgenome data package* with masked sequences. This vignette has been updated to reflect these changes.

This vignette describes the process of forging a *BSgenome data package*. It is intended for Bioconductor users who want to make a new *BSgenome data package*, not for regular users of these packages.

Start R (make sure you are using the latest release version), load the BSgenome package, and use the `available.genomes` function to get the list of *BSgenome data packages* available in the current release version of Bioconductor. So you confirm that none of those genomes suits your needs? And you want to make your own package? If you answer yes to these 2 questions, then you've come to the right place.

Requirements:

- Some basic knowledge of the Unix/Linux command line is required. The commands that you will most likely need are: `cd`, `mkdir`, `mv`, `rmdir`, `tar`, `gunzip`, `unzip`, `ftp` and `wget`. Also you will need to create and edit some text files.
- You need access to a good Unix/Linux build machine with a decent amount of RAM ( $\geq 4$ GB), especially if your genome is big. For smaller genomes, 2GB or even 1GB of RAM might be enough.
- You need the latest release versions of R plus the `Biobstrings` and `BSgenome` packages installed on the build machine. To check your installation, start R and try to load the `BSgenome` package.
- Finally, you need to obtain the *source data files* of the genome that you want to build a package for. There are 2 groups of *source data files*: (1) the files containing the sequence data (those files are required), and (2) the files containing the mask data (those files are optional). For most organisms, these files have been made publicly available on the internet by genome providers like UCSC, NCBI, FlyBase, TAIR, etc. The next section of this vignette explains how to obtain and prepare these files.

Refer to the *R Installation and Administration* manual <sup>1</sup> if you need to install R or upgrade your R

---

<sup>1</sup><http://cran.r-project.org/doc/manuals/R-admin.html>

version, and to the *Bioconductor - Install* page <sup>2</sup> on the Bioconductor website if you need to install or update the *Biostrings* or *BSgenome* packages.

Questions, comments or bug reports about this vignette or about the *BSgenomeForge* functions are welcome. Please address them to the author ([hpages@fhcrc.org](mailto:hpages@fhcrc.org)) or post them on The Bioconductor Project Mailing List ([bioconductor@r-project.org](mailto:bioconductor@r-project.org)). Don't forget to visit the Bioconductor website <sup>3</sup> and subscribe to this list if you've not already done so.

In the next section (“How to forge a *BSgenome* data package with bare sequences”), we describe how to forge a *BSgenome data package* with sequences that have no masks on them. If this is what you're after, then you'll be done at the end of the section. Only if you need to forge a package with masked sequences, you'll have to keep reading but you'll have to forge a *BSgenome data package* with bare sequences first.

In this vignette, we call *target packages* the *BSgenome data packages* that we're going to forge (one target package with bare sequences and possibly a 2nd target package with masked sequences).

## 2 How to forge a *BSgenome* data package with bare sequences

### 2.1 Obtain and prepare the sequence data

#### 2.1.1 What do I need?

The sequence data must be in FASTA files. You need 1 file per sequence that you want to put in the *target package*. The name of each file must be of the form `<prefix><seqname><suffix>` where `<seqname>` is the name of the sequence in it and `<prefix>` and `<suffix>` are a prefix and a suffix (possibly empty) that are the same for all the files.

For example the FASTA files available at UCSC in the “Data set by chromosome” section for Stickleback (<http://hgdownload.cse.ucsc.edu/goldenPath/gasAcu1/chromosomes/>) could already be considered to have names of this form (`chrI.fa.gz`, `chrII.fa.gz`, `chrIII.fa.gz`, ..., `chrXXI.fa.gz`, `chrM.fa.gz` and `chrUn.fa.gz`). However, because the files will need to be uncompressed after download and before they can be used by the forging process, the suffix will need to be set to `.fa`, not `.fa.gz`. Also the prefix here is empty, not `chr`, because `chr` is considered to be part of the sequence names (a chromosome naming convention commonly used at UCSC). Note that, alternatively, you can download and extract the big `chromFa.tar.gz` tarball located in the “Full data set” section (aka the *bigZips* folder) for Stickleback (<http://hgdownload.cse.ucsc.edu/goldenPath/gasAcu1/bigZips/>): it should contain the same files as the “Data set by chromosome” folder.

You can use the `fasta.info` function from the *Biostrings* package to see what's in a FASTA file:

```
> library(Biostrings)
> file <- system.file("extdata", "ce2chrM.fa", package="BSgenome")
> fasta.info(file)
```

```
chrM
13794
```

---

<sup>2</sup><http://bioconductor.org/install/>

<sup>3</sup><http://bioconductor.org/>

### 2.1.2 An example

Here is how the *sequence data files* for the `BSgenome.Rnorvegicus.UCSC.rn4` package were obtained and prepared:

- Single sequences: file `chromFa.tar.gz` was downloaded from the UCSC *bigZips* folder <sup>4</sup> for `rn4` and extracted with:

```
tar xzf chromFa.tar.gz
```

- Multiple sequences: files `upstream1000.fa.gz`, `upstream2000.fa.gz` and `upstream5000.fa.gz` were downloaded from the same *bigZips* folder and uncompressed with:

```
for file in upstream*.fa.gz; do gunzip $file ; done
```

### 2.1.3 The `<seqs_srcdir>` folder

From now we assume that you've downloaded (checking the `md5sums` is always a good idea), extracted, and possibly renamed all the *sequence data files*, and that they are located in the `<seqs_srcdir>` folder.

Note that all the *sequence data files* should be located directly in the `<seqs_srcdir>` folder, not in subfolders of this folder. For example, depending on the genome, UCSC provides either a big `chromFa.tar.gz` or `chromFa.zip` file that contains the sequence data for all the chromosomes. But it could be that, after extraction of this big file, the individual FASTA files for each chromosome end up being located one level down the `<seqs_srcdir>` folder (granted that you were in this folder when you extracted the file). If this is the case, then you will need to move them one level up (use `mv -i */*.fa .` for this, then remove all the empty subfolders with `rmdir *`).

## 2.2 Prepare the `BSgenome` data package seed file

### 2.2.1 Overview

The *BSgenome data package seed file* will contain all the information needed by the `forgeBSgenomeDataPkg` function to forge the *target package*.

The format of this file is DCF (Debian Control File), which is also the format used for the `DESCRIPTION` file of any R package. The valid fields of a *seed file* are divided in 3 categories:

1. Standard `DESCRIPTION` fields. These fields are actually the mandatory fields found in any `DESCRIPTION` file. They will be copied to the `DESCRIPTION` file of the *target package*.
2. Non-standard `DESCRIPTION` fields. These fields are specific to *seed files* and they will also be copied to the `DESCRIPTION` file of the *target package*. In addition, the values of those fields will be stored in the *BSgenome* object that will be contained in the *target package*. This means that the users of the *target package* will be able to retrieve these values via the accessor methods defined for *BSgenome* objects. See the man page for the *BSgenome* class (`?'BSgenome-class'`) for a description of these methods.

---

<sup>4</sup><http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/>

### 3. Additional fields that don't fall in the 2 first categories.

The 3 following sub-subsections give an extensive description of all the valid fields of a *seed file*.

Alternatively, the reader in a hurry can go directly to the last sub-subsection of this subsection for an example of *seed file*.

#### 2.2.2 Standard DESCRIPTION fields

- **Package:** Name to give to the *target package*. The convention used for the packages built by the Bioconductor project is to use a name made of 4 parts separated by a dot. Part 1 is always **BSgenome**. Part 2 is the abbreviated name of the organism (when the name of the organism is made of 2 words, we put together the first letter of the first word in upper case followed by the entire second word in lower case e.g. **Rnorvegicus**). Part 3 is the name of the organisation who provided the genome (e.g. UCSC). Part 4 is the release string or number used by this organisation to identify this version of the genome (e.g. **rn4**).
- **Title:** The title of the package. E.g. **Full genome sequences for Rattus norvegicus (UCSC version rn4)**.
- **Description, Version, Author, Maintainer, License:** Like the 2 previous fields, these are mandatory fields found in any DESCRIPTION file. Please refer to the *The DESCRIPTION file* subsection of the *Writing R Extensions* manual <sup>5</sup> for more information about these fields. If you plan to distribute the package that you are going to forge, please pickup the license carefully and make sure that it is compatible with the license of the *sequence data files* if any.

#### 2.2.3 Non-standard DESCRIPTION fields

- **organism:** The full name of the organism (e.g. **Rattus norvegicus**).
- **species:** The name of the species. For the packages built by the Bioconductor project from a UCSC genome, this field corresponds to the SPECIES column of the *List of UCSC genome releases* table <sup>6</sup>.
- **provider:** The provider of the *sequence data files* e.g. UCSC, NCBI, BDGP, FlyBase, etc... Should preferably match part 3 of the package name (field **Package**).
- **provider\_version:** The provider-side version of the genome. Should preferably match part 4 of the package name (field **Package**). For the packages built by the Bioconductor project from a UCSC genome, this field corresponds to the UCSC VERSION field of the *List of UCSC genome releases* table.
- **release\_date:** When this assembly of the genome was released. For the packages built by the Bioconductor project from a UCSC genome, this field corresponds to the RELEASE DATE field of the *List of UCSC genome releases* table.
- **release\_name:** The release name or build number of this assembly of the genome. For the packages built by the Bioconductor project from a UCSC genome, this field corresponds to the RELEASE NAME field of the *List of UCSC genome releases* table.
- **source\_url:** The permanent URL where the *sequence data files* used to forge this package can be found.

---

<sup>5</sup><http://cran.r-project.org/doc/manuals/R-exts.html#The-DESCRIPTION-file>

<sup>6</sup><http://genome.ucsc.edu/FAQ/FAQreleases#release1>

- **organism\_biocview**: The official biocViews term for this organism. This is generally the same as the **organism** field except that spaces should be replaced by underscores. The value of this field matters only if the *target package* is going to be added to a Bioconductor repository since it will determine under which subview of the *Bioconductor Task View for Organism*<sup>7</sup> the package will appear. Note that this is the only field in this category that won't be stored in the *BSgenome* object that will be contained in the *target package*.

## 2.2.4 Other fields

- **BSgenomeObjname**: Should match part 2 of the package name (see **Package** field above).
- **seqnames**: An R expression returning the names of the single sequences to forge (in a character vector). E.g. `paste("chr", c(1:20, "X", "M", "Un", paste(c(1:20, "X", "Un"), "_random", sep="")), sep="")`.
- **circ\_seqs**: [OPTIONAL] An R expression returning the names of the circular sequences (in a character vector). This must be a subset of the **seqnames** specified above. E.g. "chrM" for rn4 or c("chrM", "2micron") for the sacCer2 genome (Yeast) from UCSC. The default value for **circ\_seqs** is NULL (no circular sequence).
- **mseqnames**: [OPTIONAL] An R expression returning the names of the multiple sequences to forge (in a character vector). E.g. `paste("upstream", c("1000", "2000", "5000"), sep="")`. The default value for **mseqnames** is NULL (no multiple sequence).
- **PkgDetails**: [OPTIONAL] Some arbitrary text that will be copied to the **Details** section of the man page of the *target package*.
- **SrcDataFiles1, SrcDataFiles2**: [OPTIONAL] Some arbitrary text that will be copied to the **Note** section of the man pages of the *target package*. **SrcDataFiles1** should describe briefly where the *sequence data files* are coming from. **SrcDataFiles2** can contain any extra information and/or URLs. Permanent URLs are a must.
- **PkgExamples**: [OPTIONAL] Some R code (possibly with comments) that will be added to the **Examples** section of the man page of the *target package*.
- **seqs\_srcdir**: The path to the `<seqs_srcdir>` folder.
- **seqfiles\_prefix, seqfiles\_suffix**: [OPTIONAL] The common prefix and suffix that need to be added to all the sequence names (fields **seqnames** and **mseqnames**) to get the name of the corresponding FASTA file. Default values are the empty prefix for **seqfiles\_prefix** and `.fa` for **seqfiles\_suffix**.

## 2.2.5 An example

The *seed files* used for the packages forged by the Bioconductor project are included in the *BSgenome* package:

```
> library(BSgenome)
> seed_files <- system.file("extdata", "GentlemanLab", package="BSgenome")
> tail(list.files(seed_files, pattern="-seed$"))
```

<sup>7</sup><http://bioconductor.org/packages/release/Organism.html>

```
[1] "BSgenome.Scerevisiae.UCSC.sacCer3-seed"
[2] "BSgenome.SsCrofa.UCSC.susScr3-seed"
[3] "BSgenome.SsCrofa.UCSC.susScr3.masked-seed"
[4] "BSgenome.Tguttata.UCSC.taeGut1-seed"
[5] "BSgenome.Tguttata.UCSC.taeGut1.masked-seed"
[6] "BSgenome.influenza.NCBI.20100628-seed"
```

```
> rn4_seed <- list.files(seed_files, pattern="\\.rn4-seed$", full.names=TRUE)
> cat(readLines(rn4_seed), sep="\n")
```

```
Package: BSgenome.Rnorvegicus.UCSC.rn4
Title: Full genome sequences for Rattus norvegicus (UCSC version rn4)
Description: Full genome sequences for Rattus norvegicus (Rat) as provided by UCSC (rn4, Nov. 2004) and
Version: 1.3.99
organism: Rattus norvegicus
species: Rat
provider: UCSC
provider_version: rn4
release_date: Nov. 2004
release_name: Baylor College of Medicine HGSC v3.4
source_url: http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/
organism_biocview: Rattus_norvegicus
BSgenomeObjname: Rnorvegicus
seqnames: paste("chr", c(1:20, "X", "M", "Un", paste(c(1:20, "X", "Un"), "_random", sep="")), sep="")
circ_seqs: "chrM"
mseqnames: paste("upstream", c("1000", "2000", "5000"), sep="")
SrcDataFiles1: chromFa.tar.gz, upstream1000.fa.gz, upstream2000.fa.gz, upstream5000.fa.gz
                from http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/
PkgExamples: genome$chr1 # same as genome[["chr1"]]
seqs_srcdir: /shared/silo_researcher/Morgan_M/BioC/BSgenomeForge/srcdata/BSgenome.Rnorvegicus.UCSC.rn4
```

From now we assume that you have managed to prepare the *seed file* for your package.

## 2.3 Forge the *target package*

To forge the package, start R, load the BSgenome package, and call the `forgeBSgenomeDataPkg` function on your *seed file*. For example, if the path to your *seed file* is `"path/to/my/seed"`, do:

```
> library(BSgenome)
> #forgeBSgenomeDataPkg("path/to/my/seed", mode="fa.rz") # not ready yet
> forgeBSgenomeDataPkg("path/to/my/seed", mode="rda")
```

Depending on the size of the genome and your hardware, this can take between 2 minutes and 1 or 2 hours. By default `forgeBSgenomeDataPkg` will create the source tree of the *target package* in the current directory.

The `mode` argument was added in Bioconductor 2.14 and can be either `"rda"`, `"fa"`, or `"fa.rz"`. See `?forgeBSgenomeDataPkg` for the details.

Once `forgeBSgenomeDataPkg` is done, ignore the warnings (if any), quit R, and build the source package (tarball) with

```
R CMD build <pkgdir>
```

where `<pkgdir>` is the path to the source tree of the package.

Then check the package with

```
R CMD check <tarball>
```

where `<tarball>` is the path to the tarball produced by `R CMD build`.

Finally install the package with

```
R CMD INSTALL <tarball>
```

and use it!

## 3 How to forge a BSgenome data package with masked sequences

### 3.1 Obtain and prepare the mask data

#### 3.1.1 What do I need?

The mask data are not available for all organisms. What you download exactly depends of course on what's available and also on what built-in masks you want to have in the *2nd target package*. 4 kinds of built-in masks are currently supported by BSgenomeForge:

- the masks of assembly gaps, aka “the AGAPS masks”;
- the masks of intra-contig ambiguities, aka “the AMB masks”;
- the masks of repeat regions that were determined by the RepeatMasker software, aka “the RM masks”;
- the masks of repeat regions that were determined by the Tandem Repeats Finder software (where only repeats with period less than or equal to 12 were kept), aka “the TRF masks”.

For the AGAPS masks, you need UCSC “gap” or NCBI “agp” files. It can be one file per chromosome or a single big file containing the assembly gap information for all the chromosomes together. In the former case, the name of each file must be of the form `<prefix><seqname><suffix>`. Like for the FASTA files in group 1, `<seqname>` must be the name of the sequence (sequence names for FASTA files and AGAPS masks must match) and `<prefix>` and `<suffix>` must be a prefix and a suffix (possibly empty) that are the same for all the files (this prefix/suffix doesn't need to be, and typically is not, the same as for the FASTA files in group 1).

You don't need any file for the AMB masks.

For the RM masks, you need RepeatMasker `.out` files. Like for the AGAPS masks, it can be one file per chromosome or a single big file containing the RepeatMasker information for all the chromosomes together. In the former case, the name of each file must also be of the form `<prefix><seqname><suffix>`. Same comments apply as for the AGAPS masks above.

For the TRF masks, you need Tandem Repeats Finder `.bed` files. Again, it can be one file per chromosome or a single big file. In the former case, the name of each file must also be of the form `<prefix><seqname><suffix>`). Same comments apply as for the AGAPS masks above.

Again, for some organisms none of the masks above are available or only some of them are.

### 3.1.2 An example

Here is how the *mask data files* for the `BSgenome.Rnorvegicus.UCSC.rn4` package were obtained and prepared:

- AGAPS masks: all the `chr*_gap.txt.gz` files (UCSC “gap” files) were downloaded from the UCSC *database* folder <sup>8</sup> for `rn4`. This was done with the standard Unix/Linux `ftp` command:

```
ftp hgdownload.cse.ucsc.edu # login as "anonymous"
cd goldenPath/rn4/database
prompt
mget chr*_gap.txt.gz
```

Then all the downloaded files were uncompressed with:

```
for file in chr*_gap.txt.gz; do gunzip $file ; done
```

- RM masks: file `chromOut.tar.gz` was downloaded from the UCSC *bigZips* folder and extracted with:

```
tar xzf chromOut.tar.gz
```

- TRF masks: file `chromTrf.tar.gz` was downloaded from the UCSC *bigZips* folder and extracted with:

```
tar xzf chromTrf.tar.gz
```

### 3.1.3 The `<mask_srcdir>` folder

From now we assume that you’ve downloaded (checking the `md5sums` is always a good idea) and possibly extracted all the *mask data files*, and that they are located in the `<mask_srcdir>` folder.

Note that all the *mask data files* should be located directly in the `<mask_srcdir>` folder, not in subfolders of this folder.

---

<sup>8</sup><http://hgdownload.cse.ucsc.edu/goldenPath/rn4/database/>

## 3.2 Prepare the seed file for the masked BSgenome data package

### 3.2.1 Overview

The *seed file* for the *BSgenome data package* with masked sequences (i.e. *2nd target package*) is similar to the *seed file* for the *BSgenome data package* containing the corresponding bare sequences (a.k.a. the *reference target package* i.e. the package you forged in the previous section). It will contain all the information needed by the `forgeMaskedBSgenomeDataPkg` function to forge the *2nd target package*.

The fields of this *seed file* are described in the 3 following sub-subsections.

Alternatively, the reader in a hurry can go directly to the last sub-subsection of this subsection for an example of *seed file*.

### 3.2.2 Standard DESCRIPTION fields

- **Package:** Name to give to the *2nd target package*. The convention used for the packages built by the Bioconductor project is to use the same name as the *reference target package* with the `.masked` suffix added to it.
- **Title:** The title of the package. E.g. `Full masked genome sequences for Rattus norvegicus (UCSC version rn4)`.
- **Description, Version, Author, Maintainer, License:** See previous section.

### 3.2.3 Non-standard DESCRIPTION fields

- `organism_biocview`: Same as for the *reference target package*.
- `source_url`: The permanent URL where the *mask data files* used to forge this package can be found.

### 3.2.4 Other fields

- `RefPkgname`: The name of the *reference target package*.
- `nmask_per_seq`: The number of masks per sequence (1 to 4).
- `PkgDetails`, `PkgExamples`: See previous section.
- `SrcDataFiles1`, `SrcDataFiles2`: [OPTIONAL] Some arbitrary text that will be copied to the `Note` section of the man pages of the *target package*. `SrcDataFiles1` should describe briefly where the *mask data files* are coming from. `SrcDataFiles2` can contain any extra information and/or URLs. Permanent URLs are a must.
- `masks_srcdir`: The path to the `<masks_srcdir>` folder.
- `AGAPSfiles_type`: [OPTIONAL] Must be `gap` (the default) if the *source data files* containing the AGAPS masks information are UCSC “gap” files, or `agp` if they are NCBI “agp” files.
- `AGAPSfiles_name`: [OPTIONAL] Omit this field if you have one *source data file* per single sequence for the AGAPS masks and use the `AGAPSfiles_prefix` and `AGAPSfiles_suffix` fields below instead. Otherwise, use this field to specify the name of the single big file.

- `AGAPSfiles_prefix`, `AGAPSfiles_suffix`: [OPTIONAL] Omit these fields if you have one single big *source data file* for all the AGAPS masks and use the `AGAPSfiles_name` field above instead. Otherwise, use these fields to specify the common prefix and suffix that need to be added to all the single sequence names (field `seqnames`) to get the name of the file that contains the corresponding AGAPS mask information. Default values are the empty prefix for `AGAPSfiles_prefix` and `_gap.txt` for `AGAPSfiles_suffix`.
- `RMfiles_name`, `RMfiles_prefix`, `RMfiles_suffix`: [OPTIONAL] Those fields work like the `AGAPSfiles*` fields above but for the RM masks. Default values are the empty prefix for `RMfiles_prefix` and `.fa.out` for `RMfiles_suffix`.
- `TRFfiles_name`, `TRFfiles_prefix`, `TRFfiles_suffix`: [OPTIONAL] Those fields work like the `AGAPSfiles*` fields above but for the TRF masks. Default values are the empty prefix for `TRFfiles_prefix` and `.bed` for `TRFfiles_suffix`.

### 3.2.5 An example

The *seed files* used for the packages forged by the Bioconductor project are included in the `BSgenome` package:

```
> library(BSgenome)
> seed_files <- system.file("extdata", "GentlemanLab", package="BSgenome")
> tail(list.files(seed_files, pattern="\\.masked-seed$"))

[1] "BSgenome.Ptroglyodytes.UCSC.panTro2.masked-seed"
[2] "BSgenome.Ptroglyodytes.UCSC.panTro3.masked-seed"
[3] "BSgenome.Rnorvegicus.UCSC.rn4.masked-seed"
[4] "BSgenome.Rnorvegicus.UCSC.rn5.masked-seed"
[5] "BSgenome.Sscrofa.UCSC.susScr3.masked-seed"
[6] "BSgenome.Tguttata.UCSC.taeGut1.masked-seed"

> rn4_masked_seed <- list.files(seed_files, pattern="\\.rn4\\.masked-seed$", full.names=TRUE)
> cat(readLines(rn4_masked_seed), sep="\n")

Package: BSgenome.Rnorvegicus.UCSC.rn4.masked
Title: Full masked genome sequences for Rattus norvegicus (UCSC version rn4)
Description: Full genome sequences for Rattus norvegicus (Rat) as provided by UCSC (rn4, Nov. 2004) and
Version: 1.3.99
RefPkgname: BSgenome.Rnorvegicus.UCSC.rn4
source_url: http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/
organism_biocview: Rattus_norvegicus
nmask_per_seq: 4
SrcDataFiles1: AGAPS masks: all the chr*_gap.txt.gz files from ftp://hgdownload.cse.ucsc.edu/goldenPath/
                RM masks: http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/chromOut.tar.gz
                TRF masks: http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/chromTrf.tar.gz
PkgExamples: genome$chr1 # a MaskedDNAString object!
              ## To get rid of the masks altogether:
              unmasked(genome$chr1) # same as BSgenome.Rnorvegicus.UCSC.rn4$chr1
masks_srcdir: /shared/silo_researcher/Morgan_M/BioC/BSgenomeForge/srcdata/BSgenome.Rnorvegicus.UCSC.rn4
```

From now we assume that you have managed to prepare the *seed file* for the *2nd target package*.

### 3.3 Forge the *2nd target package*

To forge the package, start R, make sure the *reference target package* is installed (try to load it), load the `BSgenome` package, and call the `forgeMaskedBSgenomeDataPkg` function on the *seed file* you made for the *2nd target package*. For example, if the path to your *seed file* is `"path/to/my/seed"`, do:

```
> library(BSgenome)
> forgeMaskedBSgenomeDataPkg("path/to/my/seed")
```

Depending on the size of the genome and your hardware, this can take between 2 minutes and 1 or 2 hours. By default `forgeMaskedBSgenomeDataPkg` will create the source tree of the *2nd target package* in the current directory.

Once `forgeMaskedBSgenomeDataPkg` is done, ignore the warnings (if any), quit R, and build the source package (tarball) with

```
R CMD build <pkgdir>
```

where `<pkgdir>` is the path to the source tree of the package.

Then check the package with

```
R CMD check <tarball>
```

where `<tarball>` is the path to the tarball produced by `R CMD build`.

Finally install the package with

```
R CMD INSTALL <tarball>
```

and use it!

If you want to distribute this package, you need to distribute it with the *reference target package* since it depends on it.

## 4 Session information

The output in this vignette was produced under the following conditions:

```
> sessionInfo()
```

```
R version 3.1.0 (2014-04-10)
Platform: x86_64-unknown-linux-gnu (64-bit)
```

```
locale:
```

```
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
```

```
[3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
[5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8    LC_NAME=C
[9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

attached base packages:

```
[1] parallel  stats      graphics  grDevices  utils      datasets  methods
[8] base
```

other attached packages:

```
[1] BSgenome_1.32.0      GenomicRanges_1.16.0 GenomeInfoDb_1.0.0
[4] Biostrings_2.32.0    XVector_0.4.0         IRanges_1.21.45
[7] BiocGenerics_0.10.0
```

loaded via a namespace (and not attached):

```
[1] Rsamtools_1.16.0 bitops_1.0-6      stats4_3.1.0     tools_3.1.0
[5] zlibbioc_1.10.0
```