How to forge a BSgenome data package

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

January 20, 2011

Contents

1 Introduction

This document 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 your answer is 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 (>= 4GB), 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 Biostrings 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 document explains how to obtain and prepare these files.

Refer to the *R Installation and Administration* manual [1] if you need to install R or upgrade your R version, and to the *Installation Instructions* page [2] on the Bioconductor website if you need to install or update the Biostrings or BSgenome packages.

Questions, comments or bug reports about this document or about the BSgenomeForge functions are welcome. Please address them to the author (`hpages@fhcrc.org`) or post them on The Bioconductor Project Mailing List (`bioconductor@stat.math.ethz.ch`). Don't forget to visit the Bioconductor website [3] and subscribe to this list if you've not already done so.

In this document, we call *target package* the *BSgenome data package* package that we want to forge.

2 Obtain and prepare the source data files

As mentioned earlier, there are 2 groups of *source data files*: (1) the files containing the sequence data (required), and (2) the files containing the mask data (optional).

[1] `http://cran.r-project.org/doc/manuals/R-admin.html`
[2] `http://bioconductor.org/install/`
[3] `http://bioconductor.org/`

2.1 Sequence data (group 1)

Group 1 must be 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 (eventually 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
```

2.2 Mask data (group 2)

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 *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 (eventually 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.

2.3 An example

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

- Group 1:
  - Single sequences: file `chromFa.tar.gz` was downloaded from the UCSC *bigZips* folder [4] for `rn4` and extracted with:

    ```
    tar zxf 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
    ```

- Group 2:
  - AGAPS masks: all the `chr*_gap.txt.gz` files (UCSC "gap" files) were downloaded from the UCSC *database* folder [5] 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 zxf chromOut.tar.gz
    ```

[4] `http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/`
[5] `http://hgdownload.cse.ucsc.edu/goldenPath/rn4/database/`

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

```
tar zxf chromTrf.tar.gz
```

2.4 The <*seqs_srcdir*> and <*masks_srcdir*> folders

From now we assume that you've downloaded (checking the md5sums is always a good idea), extracted, and eventually renamed all the *source data files*, and that they are located in the <*seqs_srcdir*> folder for group 1 and in the <*masks_srcdir*> folder for group 2.

Note that all the *source data files* should be located directly in the <*seqs_srcdir*> and <*masks_srcdir*> folders, not in subfolders of these folders. 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 *`).

3 Prepare the BSgenome data package seed file

3.1 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.
3. Additional fields that don't fall in the 2 first categories.

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

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

3.2 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 names 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. `Rattus norvegicus full genome (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* section of the *Writing R Extensions* manual [6] 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 *source data files* if any.

3.3 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 [7].
- `provider`: The provider of the *source 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 *source data files* used to forge this package can be found.
- `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* [8] 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*.

[6] `http://cran.r-project.org/doc/manuals/R-exts.html#The-DESCRIPTION-file`
[7] `http://genome.ucsc.edu/FAQ/FAQreleases#release1`
[8] `http://bioconductor.org/packages/release/Organism.html`

3.4 Other fields

- `BSgenomeObjname`: Should match part 2 of the package name (field `Package`).
- `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="")`.
- `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="")`.
- `nmask_per_seq`: [OPTIONAL] The number of masks per sequence (0 to 4).
- `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 *source data files* for the sequences are coming from. `SrcDataFiles2` should do the same for the masks. Permanent URLs are a must.
- `PkgExamples`: [OPTIONAL] Some R code (eventually with comments) that will be added to the `Examples` section of the man page of the *target package*.
- `seqs_srcdir`, `masks_srcdir`: The path to the <*seqs_srcdir*> and <*masks_srcdir*> folders, respectively.
- `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`.
- `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`.

3.5 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")
> list.files(seed_files, pattern="-seed$")
```

```
 [1] "BSgenome.Amellifera.BeeBase.assembly4-seed"
 [2] "BSgenome.Amellifera.UCSC.apiMel2-seed"
 [3] "BSgenome.Athaliana.TAIR.01222004-seed"
 [4] "BSgenome.Athaliana.TAIR.04232008-seed"
 [5] "BSgenome.Athaliana.TAIR.TAIR9-seed"
 [6] "BSgenome.Btaurus.UCSC.bosTau3-seed"
 [7] "BSgenome.Btaurus.UCSC.bosTau4-seed"
 [8] "BSgenome.Celegans.UCSC.ce2-seed"
 [9] "BSgenome.Celegans.UCSC.ce6-seed"
[10] "BSgenome.Cfamiliaris.UCSC.canFam2-seed"
[11] "BSgenome.Dmelanogaster.UCSC.dm2-seed"
[12] "BSgenome.Dmelanogaster.UCSC.dm3-seed"
[13] "BSgenome.Drerio.UCSC.danRer5-seed"
[14] "BSgenome.Drerio.UCSC.danRer6-seed"
[15] "BSgenome.Ecoli.NCBI.20080805-seed"
[16] "BSgenome.Ggallus.UCSC.galGal3-seed"
[17] "BSgenome.Hsapiens.UCSC.hg17-seed"
[18] "BSgenome.Hsapiens.UCSC.hg18-seed"
[19] "BSgenome.Hsapiens.UCSC.hg19-seed"
[20] "BSgenome.Mmusculus.UCSC.mm8-seed"
[21] "BSgenome.Mmusculus.UCSC.mm9-seed"
[22] "BSgenome.Ptroglodytes.UCSC.panTro2-seed"
[23] "BSgenome.Rnorvegicus.UCSC.rn4-seed"
[24] "BSgenome.Scerevisiae.UCSC.sacCer1-seed"
[25] "BSgenome.Scerevisiae.UCSC.sacCer2-seed"
[26] "BSgenome.influenza.NCBI.20100628-seed"
```

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

```
Package: BSgenome.Rnorvegicus.UCSC.rn4
Title: Rattus norvegicus (Rat) full genome (UCSC version rn4)
Description: Rattus norvegicus (Rat) full genome as provided by UCSC (rn4, Nov. 2004) and stored in Bi
Version: 1.3.16
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="")
mseqnames: paste("upstream", c("1000", "2000", "5000"), sep="")
nmask_per_seq: 4
SrcDataFiles1: sequences: chromFa.tar.gz, upstream1000.fa.gz, upstream2000.fa.gz, upstream5000.fa.gz
        from http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/
SrcDataFiles2: AGAPS masks: all the chr*_gap.txt.gz files from ftp://hgdownload.cse.ucsc.edu/goldenPat
        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: Rnorvegicus
        seqlengths(Rnorvegicus)
        Rnorvegicus$chr1  # same as Rnorvegicus[["chr1"]]
seqs_srcdir: /home/hpages/BSgenomeForge/srcdata/BSgenome.Rnorvegicus.UCSC.rn4/seqs
masks_srcdir: /home/hpages/BSgenomeForge/srcdata/BSgenome.Rnorvegicus.UCSC.rn4/masks
```

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

4 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")
```

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.

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!

5 Session information

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

```
> sessionInfo()

R version 2.12.1 (2010-12-16)
Platform: i386-pc-mingw32/i386 (32-bit)

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] BSgenome_1.18.3     GenomicRanges_1.2.3 Biostrings_2.18.2
[4] IRanges_1.8.8

loaded via a namespace (and not attached):
[1] Biobase_2.10.0 tools_2.12.1
```