

# RbcBook1

October 25, 2011

---

bcStangle *A function to extract the code chunks from the book for the online*

---

## Description

A function to extract the code chunks from the book for the online supplement.

## Usage

```
bcStangle(files = .RbcBook1Files(), outfile= "bioCSpringer.R")
tangleToSingleFiles()
```

## Arguments

`files` character vector with filenames of book chapters. Must have extension `.Rnw`  
`outfile` character of length 1 with the name of the output file.

## Details

It should suffice to change to the book's top level directory (e.g. `.../Springer1/trunk/Book1`) and run `bcStangle()`.

This function is specialized to its one purpose given in the title. It is not envisaged that it could be useful beyond, or even replace `Stangle`.

`tangleToSingleFiles` creates a directory named `Rfiles` and tangles each chapter separately into this directory. You must have the current working directory set to the top level directory of the book (`Springer/Book1`).

## Value

Function is called for its side effects.

## Author(s)

Wolfgang Huber <huber@ebi.ac.uk>

## See Also

`.RbcBook1Files`, `Stangle`

**Examples**

```
## bcStangle()
```

---

bcSweave	<i>Wrapper for Sweave</i>
----------	---------------------------

---

**Description**

Wrapper for Sweave - allows to set options, or perform additional preprocessing or QC steps.

**Usage**

```
bcSweave(f)
```

**Arguments**

`f` character, filename. Must have extension `.Rnw`

**Details**

No details.

**Value**

Function is called for its side effects.

**Author(s)**

Wolfgang Huber <huber@ebi.ac.uk>

**Examples**

```
## bcSweave("Intro.Rnw")
```

---

bcr.cor	<i>Illustrative datasets for distance measures</i>
---------	--

---

**Description**

Illustrative datasets for distance measures

**Usage**

```
data(ALL.dist) # all the components live in here
```

**Details**

These datasets are used in conjunction with the `bioDist` package for the distance chapter of the monograph

**Value**

these are `dist` objects

**Author(s)**

Vince Carey <stvjc@channing.harvard.edu>

**Examples**

```
data(ALL.dist)
objects()
dim(as.matrix(bcr.cor))
```

---

checkingBookSources

*Formatting and standardization checks on book chapters*

---

**Description**

Formatting and standardization checks on book chapters

**Usage**

```
checkVerbatim(files = .RbcBook1Files(ext=".tex"),
              maxc = 70,
              which = "both",
              verbose = TRUE)

checkRnw(files = .RbcBook1Files(),
         verbose = TRUE, stopOnError=FALSE)

checkPackage(files = .RbcBook1Files(ext=".Rnw"),
            verbose = TRUE)

.RbcBook1Files(ext=".Rnw")
```

**Arguments**

<code>files</code>	character vector with names (and path) of chapter source files
<code>ext</code>	character of length 1. File name extension. <code>.Rnw</code> (the default) and <code>.tex</code> should be the most important cases.
<code>maxc</code>	integer of length 1. Maximal number of characters in a verbatim line. All lines exceeding this limit will be reported in the return value of this function.
<code>which</code>	character of length 1. 'Sinput' will look at Sinput environments, 'Soutput' at Soutput environments, 'both' at both.
<code>verbose</code>	Logical.
<code>stopOnError</code>	Logical. If FALSE, first error found will lead to stop. If TRUE, try to continue checking.

**Details**

`checkVerbatimLines` sees whether all verbatim lines have length at most `maxc`.

`checkPackage` finds all the occurrences of `\Rpackage{...}` in the text, checks whether the package is known, and returns a named list with all occurrences of the packages.

**Value**

For `checkVerbatimLines` and `checkSetup`, a data frame with one row for each offending line and various columns describing it.

For `.RbcBook1Files`, a character vector.

**Author(s)**

Wolfgang Huber <huber@ebi.ac.uk>

**Examples**

```
f = tempfile()
zap = function(n) paste(formatC(1:n, width=2), collapse=" ")
writeLines(c("*begin{Sinput}", zap(20), zap(30), "*end{Sinput}"), con=f)
checkVerbatim(files=f)
```

---

imageSize

*File info for all image files in the book*

---

**Description**

File info for all image files in the book

**Usage**

```
imageSize(dirs= c("Preproc", "Analysis", "Metadata", "Graphs", "CaseStudies"), ext=c
```

**Arguments**

`dirs` character vector with directories.  
`ext` character vector with file name extensions.

**Details**

No details.

**Value**

Data frame.

**Author(s)**

Wolfgang Huber <huber@ebi.ac.uk>

**Examples**

```
# a = imageSize()
# print(a[1:10,c(1,4)])
```

---

`perf`*Results of tedious computations for computational inference*

---

**Description**

Results of tedious computations for computational inference

**Usage**

```
data(Survperformance)
```

**Details**

These datasets are used in conjunction with the computational inference chapter of the monograph

**Value**

these are `data.frame` objects

**Author(s)**

Vince Carey <stvjc@channing.harvard.edu>

**Examples**

```
data(Survperformance)
objects()
dim(performance)
```

---

`require.RbcBook1`*Load a lengthy list of packages that are used in the Book*

---

**Description**

Load a lengthy list of packages that are used in the Book

**Usage**

```
require.RbcBook1()
.RbcBook1.pkgs()
```

**Details**

`require.RbcBook1` can be used to see at once whether all packages required for the book can be loaded.

To do for `.RbcBook1.pkgs`: version numbers?

**Value**

`.RbcBook1.pkgs` returns a character vector with the names of all packages used in the book.

**Author(s)**

Wolfgang Huber <huber@ebi.ac.uk>

**Examples**

```
## Not run:
library(reposTools)
install.packages2(.RbcBook1.pkgs(), develOK=TRUE)
require.RbcBook1()

## End(Not run)
```

---

rpart2gNEL

*rpart to graphNEL Converter*


---

**Description**

Creates an object of class `graphNEL` from one of class `rpart`.

**Usage**

```
rpart2gNEL(tr, remap=function(x) x, nsep="\n")
```

**Arguments**

<code>tr</code>	instance of <code>rpart</code> class from <code>rpart</code> package.
<code>remap</code>	function that maps node names in tree to node names in graph.
<code>nsep</code>	token that separates node name and vote tally in the rendering.

**Details**

Not explicitly used in the book, but the basis for a certain graph illustrating application of `rpart`. The newer coin/party packages have more interesting rendering approaches based on `graphviz`.

`grabSplitV` is a utility function for acquiring the variable names, `remapAff` will remap affymetrix probe names to gene symbols for use in the `remap` parameter.

**Value**

a `graphNEL-class` instance with nodes constructed to convey information on the data tree fit

**Author(s)**

Vince Carey <stvjc@channing.harvard.edu>

**See Also**

`rpart`, `graphNEL-class`

**Examples**

```
library(rpart)
example(rpart)
ff = rpart2gNEL(fit2)
ff
nodes(ff)
```

# Index

## \*Topic **data**

bcr.cor, 2  
perf, 5

## \*Topic **misc**

bcStangle, 1  
bcSweave, 2  
checkingBookSources, 3  
imageSize, 4  
require.RbcBook1, 5

## \*Topic **models**

rpart2gNEL, 6  
.RbcBook1Files  
(checkingBookSources), 3

ALLperformance (perf), 5

bcr.cor, 2  
bcr.euc (bcr.cor), 2  
bcr.kldist (bcr.cor), 2  
bcr.man (bcr.cor), 2  
bcr.mi (bcr.cor), 2  
bcr.spear (bcr.cor), 2  
bcr.tau (bcr.cor), 2  
bcStangle, 1  
bcSweave, 2

checkingBookSources, 3  
checkPackage  
(checkingBookSources), 3  
checkRnw (checkingBookSources), 3  
checkVerbatim  
(checkingBookSources), 3

dist, 3

grabSplitV (rpart2gNEL), 6  
graphNEL-class, 6

imageSize, 4

kidpackperformance (perf), 5

perf, 5  
performance (perf), 5

remapAff (rpart2gNEL), 6

require.RbcBook1, 5  
rpart, 6  
rpart2gNEL, 6

Survperformance (perf), 5

tangleToSingleFiles (bcStangle), 1