Package 'BeadDataPackR'

November 11, 2026

Type Package

Title Compression of Illumina BeadArray data

Version 1.64.0

Date 2018-02-09

Author Mike Smith, Andy Lynch

Maintainer Mike Smith <grimbough@gmail.com>

Description Provides functionality for the compression and decompression of raw bead-level data from the Illumina BeadArray platform.

License GPL-2

LazyLoad yes

biocViews Microarray

VignetteBuilder knitr

Suggests BiocStyle, knitr

Imports stats, utils

git_url https://git.bioconductor.org/packages/BeadDataPackR

git_branch RELEASE_3_23

git_last_commit df1e5d8

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

compressBeadData 2
decompressBeadData 3
example 4
extractLocsFile 5
readCompressedData 5

Index **7**

`compressBeadData` *Write raw bead level data to a compressed format.*

Description

Given raw bead level data, in the form of .txt and .locs file, this function combines the two producing a new file with the data stored in a compressed format.

Usage

```
compressBeadData(txtFile, locsGrn, locsRed = NULL, outputFile = NULL,
                 path = NULL, nBytes = 8, base2 = TRUE, fullLocsIndex = FALSE,
                 nrow = NULL, ncol = NULL, progressBar = TRUE)
```

Arguments

`txtFile`	The name of the .txt file to be read in.
`locsGrn`	The locs file for the green channel.
`locsRed`	The locs file for the red channel. Only needed for two channel data.
`outputFile`	Name of the file to be created.
`path`	Path to where the input files can be found. If NULL the current working directory is used. This is also the directory where the output files will be written.
`nBytes`	Gives the number of bytes that are used to store the fractional parts of the bead coordinates. For a single channel array the maximum value is 4, whilst it is 8 for a two channel array. Any number larger than this is automatically set the the maximum value. If the maximum value is used the coordinates are stored in the .bab file as single precision floating point numbers, as they are in the .locs files. If a value smaller than the maximum is choosen then the integer parts of each coordinate are stored seperately. The requested number of bytes are then used to store the fractional parts, with a corresponding loss of precision as the number of bytes decreases.
`base2`	If not using the full precision coordinates, the approximations can be stored as either a binary or decimal fraction. Using a binary fraction (base2 = TRUE) provides a greater accuracy, but can lead to a meandering number of decimal places in the reconstructed .txt files. If one wants a consistent number of decimal places, set base2 = FALSE.
`fullLocsIndex`	Default value of 0 uses a linear model fitted to each segment of the array to allow reconstruct the locs file when the file is decompressed. Using a value of 1 a simple index is used to record the locs file order, but requires more space.
`ncol`	This specifies the number of columns in each grid segment on the array and, if left blank, can normally be infered from the grid coordinates. However, this can fail for particularly small grids. If one wants or needs to specify them explicitly, these values can be found in the .sdf which accompanies the bead level output from the scanner. The number of columns per segment can be found within the tag <SizeGridX>

nrow	See ncol. If needed can be found within the <SizeGridY> tag in the .sdf file.
progressBar	By default the function uses a txtProgressBar to indicate progress through the compression. Setting this argument to FALSE supresses the drawing of this progress bar.

Details

In the future the file names will be determined automatically, rather than requiring manual entry of each. The path argument may also be amended so there are seperate options for the locations of the input and output files.

Value

Primarily invoked for its side effect, which is to produce a compressed version of the input files. The function returns, invisibly, a logical TRUE if compression was successful.

Author(s)

Mike L. Smith

Examples

```
dataPath <- system.file("extdata", package = "BeadDataPackR")
## copy the files to a temp directory, and don't overwrite system files
file.copy( list.files(path = dataPath, pattern = "example", full.names = TRUE),
           tempdir() )
compressBeadData(txtFile = "example.txt", locsGrn = "example_Grn.locs",
outputFile = "example.bab", path = tempdir(), nBytes = 4, nrow = 326,
ncol = 4, fullLocsIndex = TRUE)
```

---

decompressBeadData *Decompress a file in the beadarray binary format*

Description

Decompressed a file create by BeadDataPackR. The original files that were compressed will be restored as accurately as possible, depending upon the degree of precision specified during the compression.

Usage

```
decompressBeadData(input, inputPath = ".", outputMask = NULL, outputPath = ".",
                outputNonDecoded = FALSE, roundValues = TRUE, progressBar = TRUE)
```

Arguments

input	The name of the .bab file(s) to be read. Can be a vector of file names, such as generated by `list.files()`.
inputPath	Path where the compress file is located. The default is to use the current working directory.
outputMask	Text specify the names of the output files. The output files will have ".txt", "_Grn.locs" and (if approriate "_Red.locs") appended to this mask. If left NULL the original names of the section will be used.
outputPath	Path to where the uncompressed version of the files should be written to. The default is to use the current working directory.
outputNonDecoded	If TRUE the undecoded beads will be included in the output .txt file. They will have ProbeID 0 and intensity 0, but the bead centre coordinates will be included.
roundValues	The original Illumina text files give the bead centre coordinates to 7 significant figures. When this argument is TRUE decompressed files are also truncated in this manner, whilst FALSE writes them to the full precision they are stored in the compressed file.
progressBar	By default the function uses a `txtProgressBar` to indicate progress through the compression. Setting this argument to FALSE supresses the drawing of this progress bar.

Value

Called primarily for its side effect, in which two (or three) files are written to the disk. These files should be representative of the original files that were compressed. The function returns, invisibly, the number of lines written in the .txt file.

Author(s)

Mike L. Smith

Examples

```
 dataPath <- system.file("extdata", package = "BeadDataPackR")
decompressBeadData(input = "example.bab", inputPath = dataPath, outputPath = tempdir())
```

`example` *Example bead-level data*

Description

Example bead-level data consisting of a `.txt` file, a `.locs` file and the `.bab` file that is produced from their compression.

extractLocsFile *Retrieve only the .locs file information*

Description

Provides a mechanism to extract the information from the original .locs file from a compressed .bab file, without the need to extract the intensity or probe ID values.

Usage

```
extractLocsFile(inputFile, path = ".")
```

Arguments

inputFile	The name of the .bab file to be read in.
path	Path to where the input file can be found. Default is the current working directory.

Value

A matrix with two columns (four if two-channel data) containing the X and Y values of the bead centre coordinates supplied in the original .locs file. For two-channel data the first two columns contain the coordinates from the green channel, with the red channel held in columns three and four.

Author(s)

Mike L. Smith

Examples

```
dataPath <- system.file("extdata", package = "BeadDataPackR")
locs <- extractLocsFile(inputFile = "example.bab", path = dataPath)
locs[1:10,]
```

readCompressedData *Extract data for specific bead-types from a compressed file*

Description

Given a list of probeIDs this function can scan a compressed .bab file for matching entries and return the data as a data.frame within R, rather than decompressing the data and generating new files.

Usage

```
readCompressedData(inputFile, path = ".", probeIDs = NULL)
```

Arguments

inputFile	The name of the .bab file to be read in.
path	Path to where the input file can be found. Default is the current working directory.
probeIDs	List the probe IDs for which data should be obtained. If left NULL then every probe on the array is returned.

Value

If the requested probe IDs are present the function returns a data.frame with one row per bead. If the probes are not found in the file then the function returns NULL and informs the user.

Author(s)

Mike L. Smith

Examples

```
  dataPath <- system.file("extdata", package = "BeadDataPackR")
readCompressedData(inputFile = "example.bab", path = dataPath, probeIDs = c(10008, 10010))
```

Index

∗ **IO**
    compressBeadData, 2
    decompressBeadData, 3
    extractLocsFile, 5
    readCompressedData, 5
∗ **datasets**
    example, 4

compressBeadData, 2

decompressBeadData, 3

example, 4
extractLocsFile, 5

readCompressedData, 5

txtProgressBar, *3*, *4*