Package 'dChipIO'

October 13, 2022

Version 0.1.5

Title Methods for Reading dChip Files

Depends R (>= 2.6.0)

Date 2016-01-12

Author Henrik Bengtsson [aut, cre, cph]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Description Functions for reading DCP and CDF.bin files generated by the dChip software.

License LGPL (>= 2.1)

URL https://github.com/HenrikBengtsson/dChipIO

BugReports https://github.com/HenrikBengtsson/dChipIO/issues

LazyLoad TRUE

biocViews Infrastructure, DataImport

NeedsCompilation no

Repository CRAN

Date/Publication 2016-01-13 22:36:24

R topics documented:

---

dChipIO-package   *Package dChipIO*

Description

Functions for reading DCP and CDF.bin files generated by the dChip software.

The example data used in this package orginates from the Affymetrix Fusion SDK library [3].

To get started

To get started, see:

1. readCdfBin() - reads a dChip CDF.bin file.
2. readDcp() - reads a dChip DCP data file.

License

The releases of this package is licensed under LGPL version 2.1 or newer.

Author(s)

Henrik Bengtsson.

References

[1] The dChip software, http://www.dchip.org/

[2] Thread 'DCP File Format', 'dChip Software', Google Groups, December 2008. https://groups.google.com/forum/#!topic/dchip-software/Q7mTJPPpZ5U

[3] Affymetrix Inc, Fusion Software Developers Kit (SDK), 2008. http://www.affymetrix.com/estore/partners_programs/programs/developer/fusion/index.affx?terms=no

---

readCdfBin   *Reads a dChip CDF.bin file*

Description

Reads a dChip CDF.bin file.

Please note that this method is incomplete as it currently doesn't read all fields. It is only made available so that someelse can continue the development.

Usage

```
readCdfBin(con, units=NULL, ...)
```

Arguments

con	A connection or a character filename.
units	An integer vector specifying the units to be read. If NULL, all units are read.
...	Not used.

Value

Returns a list structure containing the file header and the unit data.

Author(s)

Henrik Bengtsson

See Also

To read only the CDF.bin file header, see readCdfBinHeader().

Examples

```
path <- system.file("exData", package="dChipIO")
chipType <- "Test3"
filename <- sprintf("%s.CDF.bin", chipType)
pathname <- file.path(path, filename)

hdr <- readCdfBinHeader(pathname)
print(hdr)

data <- readCdfBin(pathname)
str(data)

# Read a subset of the units
units <- c(10:11, 15:20, 150:105, 2,2,2)
dataT <- readCdfBin(pathname, units=units)
str(dataT)

# Assert correctness
for (ff in c("unitNames", "numProbes", "CellPos")) {
  stopifnot(length(dataT[[ff]]) == length(units))
  stopifnot(identical(dataT[[ff]], data[[ff]][units]))
}
```

---

readCdfBinHeader   *Reads the file header of a dChip CDF.bin file*

Description

Reads the file header of a dChip CDF.bin file.

Usage

```
readCdfBinHeader(con, ...)
```

Arguments

con	A connection or a character filename.
...	Not used.

Value

Returns a list structure containing the file header.

Author(s)

Henrik Bengtsson

See Also

To read the CDF.bin file data, see readCdfBin().

---

readDcp   *Reads a dChip DCP file*

Description

Reads a dChip DCP file.

Usage

```
readDcp(con, fields=c("rawIntensities", "normalizedIntensities", "calls", "thetas",
  "thetaStds", "excludes"), cells=NULL, units=NULL, .nbrOfUnits=NULL, ...)
```

Arguments

con	A connection or a character filename.
fields	A character vector specifying the fields to be read.
cells	An integer vector specifying the indices of the cell data to be read.
units	An integer vector specifying the indices of the unit data to be read.
.nbrOfUnits	A integer specifying the number of units available in the file. If NULL, this is inferred from the file size and the file header. The dChip software itself instead uses the corrsponding value in the CDF.bin file, but that file is specified by the user leaving room for errors.
...	Not used.

Value

Returns a [list](#) structure containing the file header and the requested data fields.

Author(s)

Henrik Bengtsson

See Also

To read only the DCP file header, see [readDcpHeader](#)().

Examples

```
path <- system.file("exData", package="dChipIO")

filename <- "Test3-1-121502.dcp"
pathname <- file.path(path, filename)

hdr <- readDcpHeader(pathname)
print(hdr)

data <- readDcp(pathname)
str(data)

# Read a subset of the units
units <- c(10:11, 15:20, 150:105, 2,2,2)
dataT <- readDcp(pathname, units=units)
str(dataT)

# Assert correctness
for (ff in c("calls", "thetas", "thetaStds", "excludes")) {
  stopifnot(length(dataT[[ff]]) == length(units))
  stopifnot(identical(dataT[[ff]], data[[ff]][units]))
}
```

---

readDcpHeader *Reads the file header of a dChip DCP file*

Description

Reads the file header of a dChip DCP file.

Usage

```
readDcpHeader(con, ...)
```

Arguments

con   A [connection](#) or a [character](#) filename.
...   Not used.

Value

Returns a list structure containing the file header.

Author(s)

Henrik Bengtsson

See Also

To read also the DCP file data, see readDcp().

---

readDcpRectangle   *Reads a spatial subset of probe-level data from a dChip DCP file*

Description

Reads a spatial subset of probe-level data from a dChip DCP file.

Usage

```
readDcpRectangle(filename, fields=c("rawIntensities", "normalizedIntensities"),
  xrange=c(0, Inf), yrange=c(0, Inf), ..., asMatrix=TRUE)
```

Arguments

filename	The pathname of the DCP file.
fields	The cell fields to be read.
xrange	A numeric vector of length two giving the left and right coordinates of the cells to be returned.
yrange	A numeric vector of length two giving the top and bottom coordinates of the cells to be returned.
asMatrix	If TRUE, the CEL data fields are returned as matrices with element (1,1) corresponding to cell (xrange[1],yrange[1]).
...	Additional arguments passed to readDcp().

Value

A named list CEL structure similar to what readDcp(). In addition, if asMatrix is TRUE, the CEL data fields are returned as matrices, otherwise not.

Author(s)

Henrik Bengtsson

See Also

The [readDcp]() method is used internally. This method was inspired by readCelRectangle() of the **affxparser** package.

Examples

```
path <- system.file("exData", package="dChipIO")

filename <- "Test3-1-121502.dcp"
pathname <- file.path(path, filename)

data <- readDcpRectangle(pathname)

layout(matrix(1:4, nrow=2, byrow=TRUE))
image(data$rawIntensities, main="Raw probe signals")
image(data$normalizedIntensities, main="Normalized probe signals")
```

Index