Package ‘coreCT’

November 28, 2019

Type Package

Title Programmatic Analysis of Sediment Cores Using Computed Tomography Imaging

Version 1.3.2

Date 2019-11-27

Description Computed tomography (CT) imaging is a powerful tool for understanding the composition of sediment cores. This package streamlines and accelerates the analysis of CT data generated in the context of environmental science. Included are tools for processing raw DICOM images to characterize sediment composition (sand, peat, etc.). Root analyses are also enabled, including measures of external surface area and volumes for user-defined root size classes. For a detailed description of the application of computed tomography imaging for sediment characterization, see: Davey, E., C. Wigand, R. Johnson, K. Sundberg, J. Morris, and C. Roman. (2011) <DOI: 10.1890/10-2037.1>.

License GPL-3

URL https://github.com/troyhill/coreCT

BugReports https://github.com/troyhill/coreCT/issues

LazyData true

Depends R (>= 3.0), raster, igraph, oro.dicom, plyr

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

NeedsCompilation no

Author Troy D. Hill [aut, cre] (<https://orcid.org/0000-0003-2980-4099>), Earl Davey [ctb]

Maintainer Troy D. Hill <Hill.Troy@gmail.com>

Repository CRAN

Date/Publication 2019-11-28 11:20:02 UTC

R topics documented:

`conv` *Convert a matrix of semi-processed DICOM images to mass and volume of material classes. This is a deprecated version.*

Description

Converts raw CT units to material classes for each CT slice, directly replicating Earl Davey's manual classification approach. This method is deprecated as of coreCT version 1.3.0.

Usage

```
conv(mat.list, upperLim = 3045, lowerLim = -1025,
pixelA, thickness = 0.625, # all in mm
airHU = -850.3233, airSD = 77.6953,
SiHU = 271.7827, SiSD = 39.2814,
glassHU = 1345.0696, glassSD = 45.4129,
waterHU = 63.912, waterSD = 14.1728,
densities = c(0.0012, 1, 1.23, 2.2))
```

Arguments

`mat.list`	list of DICOM images for a sediment core (values in Hounsfield Units)
`upperLim`	upper bound cutoff for pixels (Hounsfield Units)
`lowerLim`	lower bound cutoff for pixels (Hounsfield Units)
`pixelA`	pixel area (mm2)
`thickness`	CT image thickness (mm)
`airHU`	mean value for air-filled calibration rod (Hounsfield Units)
`airSD`	standard deviation for air-filled calibration rod
`SiHU`	mean value for colloidal silica calibration rod

`SiSD`	standard deviation for colloidal Si calibration rod
`glassHU`	mean value for glass calibration rod
`glassSD`	standard deviation for glass calibration rod
`waterHU`	mean value for water filled calibration rod
`waterSD`	standard deviation for water filled calibration rod
`densities`	numeric vector of known cal rod densities. Format must be c(air, water, Si, glass)

Details

Calculates average Hounsfield units, cross-sectional areas (cm2), volumes (cm3), and masses (g) of material classes for each CT slice. This function assumes that core walls and all non-sediment material have been removed from the raw DICOM imagery. This function converts data from raw x-ray attenuation values to Hounsfield Units, and then uses user-defined calibration rod inputs to categorize sediment components: air, roots and rhizomes, peat, water, particulates, sand, and rock/shell.

Value

value `conv` returns a dataframe with one row per CT slice. Values returned are the average Hounsfield Unit value, the area (cm2), volume (cm3), and mass (grams) of 7 material classes: gas, peat, roots and rhizomes, particulates, sand, water, and rock/shell. If <code>rootData = TRUE</code>, data for specified root size classes are also returned. See <code>rootSize</code> for more detail on those values.

See Also

`rootSize` operates similarly.

Examples

```
ct.slope <- unique(extractHeader(core_426$hdr, "RescaleSlope"))
ct.int   <- unique(extractHeader(core_426$hdr, "RescaleIntercept"))
# convert raw units to Hounsfield units
HU_426 <- lapply(core_426$img, function(x) x*ct.slope + ct.int)

materials <- conv(HU_426, pixelA = 0.0596)

## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
mass.long <- reshape2::melt(materials, id.vars = c("depth"),
   measure.vars = grep(".g", names(materials)))
ggplot2::ggplot(data = mass.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("mass per section (g)")

## End(Not run)
```

convDir *Convert a directory of raw DICOM images to material classes*

Description

Calculates the area and volume of material classes for each CT slice in a directory. This approach directly replicates Earl Davey's manual classification approach. This method is deprecated as of coreCT version 1.3.0.

Usage

```
convDir(directory = file.choose(), upperLim = 3045, lowerLim = -1025,
airHU = -850.3233, airSD = 77.6953,
SiHU = 271.7827, SiSD = 39.2814,
glassHU = 1345.0696, glassSD = 45.4129,
waterHU = 63.912, waterSD = 14.1728,
densities = c(0.0012, 1, 1.23, 2.2),
rootData = TRUE,
diameter.classes = c(1, 2, 2.5, 10),
class.names = diameter.classes,
pixel.minimum = 4)
```

Arguments

`directory` a character string that can be a matrix of DICOM images or the address of an individual DICOM file in a folder of DICOM images. The default action is <code>file.choose()</code>; a browser menu appears so the user can select the the desired directory by identifying a single DICOM file in the folder of images.

`upperLim` upper bound cutoff for pixels (Hounsfield Units)

`lowerLim` lower bound cutoff for pixels (Hounsfield Units)

`airHU` mean value for air-filled calibration rod (Hounsfield Units)

`airSD` standard deviation for air-filled calibration rod

`SiHU` mean value for colloidal silica calibration rod

`SiSD` standard deviation for colloidal Si calibration rod

`glassHU` mean value for glass calibration rod

`glassSD` standard deviation for glass calibration rod

`waterHU` mean value for water filled calibration rod

`waterSD` standard deviation for water filled calibration rod

`densities` numeric vector of known cal rod densities. Format must be c(air, water, Si, glass)

`rootData` if TRUE, `rootSize` is also called on the matrix

`diameter.classes`
if rootData is TRUE, this argument provides an integer vector of diameter cut points used by `rootSize`. Units are mm (zero is added in automatically).

`class.names` placeholder, not used presently

`pixel.minimum` minimum number of pixels needed for a clump to be identified as a root

Details

Calculates the area and volume of material classes for each CT slice in a directory. Unlike `conv`, `convDir` accepts a folder of raw values and makes the conversion to Hounsfield Units using the metadata associated with the DICOM images.

Value

value `convDir` returns a dataframe with one row per CT slice. Values returned are the area and volume of seven material classes: gas, peat, roots and rhizomes, rock and shell, fine mineral particles, sand, and water. If `rootData = TRUE`, the output will also contain data on the abundance (number of particles), volume (cm3), and external surface area (cm2) of the root size classes specified in the `diameter.classes` argument.

See Also

`convDir` is a wrapper for `conv`. `rootSizeDir` operates similarly.

Examples

```
materials <- convDir("core_426", rootData = FALSE)


## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
mass.long <- reshape2::melt(materials, id.vars = c("depth"),
   measure.vars = grep(".g", names(materials)))
ggplot2::ggplot(data = mass.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("mass per section (g)")

## End(Not run)
```

`convert` *Convert a matrix of semi-processed DICOM images to mass and volume of material classes*

Description

Converts raw CT units to material classes for each CT slice. This version accommodates calibration curves with >4 calibrants, and uses density thresholds converted to Hounsfield Units using the calibration curve (rather than direct calibration rod values) to partition sediment components.

Usage

```
convert(mat.list, upperLim = 3045, lowerLim = -1025,
pixelA, thickness = 0.625, # all in mm
means     = c(-850.3233, 63.912, 271.7827, 1345.0696),
sds       = c(77.6953, 14.1728, 39.2814, 45.4129),
densities = c(0.0012, 1, 1.23, 2.2))
```

Arguments

`mat.list`	list of DICOM images for a sediment core (values in Hounsfield Units)
`upperLim`	upper bound cutoff for pixels (Hounsfield Units)
`lowerLim`	lower bound cutoff for pixels (Hounsfield Units)
`pixelA`	pixel area (mm2)
`thickness`	slice thickness for computed tomography image series (mm)
`means`	mean values (units = Hounsfield Units) for calibration rods used.
`sds`	standard deviations (units = Hounsfield Units) for calibration rods used. Must be in the same order as `means`.
`densities`	numeric vector of known cal rod densities. Must be in the same order as `means` and `sds`.

Details

Calculates average Hounsfield units, cross-sectional areas (cm2), volumes (cm3), and masses (g) of material classes for each CT slice. This function assumes that core walls and all non-sediment material have been removed from the raw DICOM imagery. This function converts data from raw x-ray attenuation values to Hounsfield Units, and then uses user-defined calibration rod inputs to categorize sediment components: air, roots and rhizomes, peat, water, particulates, sand, and rock/shell. The input style for calibration rods ensures sediment components are partitioned following the density divisions in Davey et al. 2011. Calibration rods and are used to develop the calibration curve. Separately, the densities used for partitioning in Davey et al. 2011 (0.0012, 1, 1.23, 2.2 g/cm3) are converted to Hounsfield Units and used for partitioning sediment components. The standard deviation for the calibration rod nearest to the target value is used for the standard deviation for the division between two sediment components.

Value

value `convert` returns a dataframe with one row per CT slice. Values returned are the average Hounsfield Unit value, the area (cm2), volume (cm3), and mass (grams) of 7 material classes: gas, peat, roots and rhizomes, particulates, sand, water, and rock/shell. If <code>rootData = TRUE</code>, data for specified root size classes are also returned. See <code>getRoots</code> for more detail on those values.

See Also

`getRoots` operates similarly.

Examples

```
ct.slope <- unique(extractHeader(core_426$hdr, "RescaleSlope"))
ct.int   <- unique(extractHeader(core_426$hdr, "RescaleIntercept"))
# convert raw units to Hounsfield units
HU_426 <- lapply(core_426$img, function(x) x*ct.slope + ct.int)

materials <- convert(HU_426, pixelA = 0.0596)

## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
mass.long <- reshape2::melt(materials, id.vars = c("depth"),
   measure.vars = grep(".g", names(materials)))
ggplot2::ggplot(data = mass.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("mass per section (g)")

## End(Not run)
```

convertDir *Convert a directory of raw DICOM images to material classes*

Description

Calculates the area and volume of material classes for each CT slice in a directory. This version accommodates calibration curves with >4 calibrants, and uses density thresholds converted to Hounsfield Units using the calibration curve (rather than direct calibration rod values) to partition sediment components.

Usage

```
convertDir(directory = file.choose(), upperLim = 3045, lowerLim = -1025,
means     = c(-850.3233, 63.912, 271.7827, 1345.0696),
sds       = c(77.6953, 14.1728, 39.2814, 45.4129),
densities = c(0.0012, 1, 1.23, 2.2),
rootData = TRUE,
diameter.classes = c(1, 2, 2.5, 10),
class.names = diameter.classes,
pixel.minimum = 4)
```

Arguments

`directory` a character string that can be a matrix of DICOM images or the address of an individual DICOM file in a folder of DICOM images. The default action is <code>file.choose()</code>; a browser menu appears so the user can select the the desired directory by identifying a single DICOM file in the folder of images.

`upperLim` upper bound cutoff for pixels (Hounsfield Units) `lowerLim` lower bound cutoff for pixels (Hounsfield Units)

`means` mean values (units = Hounsfield Units) for calibration rods used.

`sds` standard deviations (units = Hounsfield Units) for calibration rods used. Must be in the same order as `means`.

`densities` numeric vector of known cal rod densities. Format must be c(air, water, Si, glass)

`rootData` if TRUE, `rootSize` is also called on the matrix

`diameter.classes` if rootData is TRUE, this argument provides an integer vector of diameter cut points used by `rootSize`. Units are mm (zero is added in automatically).

`class.names` placeholder, not used presently

`pixel.minimum` minimum number of pixels needed for a clump to be identified as a root

Details

Calculates the area and volume of material classes for each CT slice in a directory. Unlike `conv`, `convDir` accepts a folder of raw values and makes the conversion to Hounsfield Units using the metadata associated with the DICOM images.

Value

value `convertDir` returns a dataframe with one row per CT slice. Values returned are the area and volume of seven material classes: gas, peat, roots and rhizomes, rock and shell, fine mineral particles, sand, and water. If `rootData` = `TRUE`, the output will also contain data on the abundance (number of particles), volume (cm3), and external surface area (cm2) of the root size classes specified in the `diameter.classes` argument.

See Also

`convertDir` is a wrapper for `convert`. `getRootsDir` operates similarly.

Examples

```
materials <- convertDir("core_426", rootData = FALSE)


## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
mass.long <- reshape2::melt(materials, id.vars = c("depth"),
   measure.vars = grep(".g", names(materials)))
ggplot2::ggplot(data = mass.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("mass per section (g)")

## End(Not run)
```

coreHist *Whole-core frequency distribution of Hounsfield units*

Description

Provides the raw data and plots a frequency distibution for Hounsfield Units in the entire core, also delineating material classes. As of coreCT version 1.3.0, this code accommodates calibration curves with >4 calibrants, and uses density thresholds converted to Hounsfield Units using the calibration curve (rather than direct calibration rod values) to partition sediment components.

Usage

```
coreHist(directory = file.choose(),
units = "percent",
upperLim = 3045, lowerLim = -1025,
means     = c(-850.3233, 63.912, 271.7827, 1345.0696),
sds       = c(77.6953, 14.1728, 39.2814, 45.4129),
densities = c(0.0012, 1, 1.23, 2.2),
returnData = TRUE, pngName = NULL)
```

Arguments

`directory` a character string that can be (1) a matrix of DICOM images that exists in the global environment, or (2) the address of an individual DICOM file in a folder of DICOM images. The default action is <code>file.choose()</code>; a browser menu appears so the user can select the the desired directory by identifying a single DICOM file in the folder of images.

`units` units to be used for plotting purposes: either "percent" (the default) or "absolute"

`upperLim` upper bound cutoff for pixels (Hounsfield Units); upper bound is inclusive

`lowerLim` lower bound cutoff for pixels (Hounsfield Units); lower bound is exclusive

`means` mean values (units = Hounsfield Units) for calibration rods used.

`sds` standard deviations (units = Hounsfield Units) for calibration rods used. Must be in the same order as `means`.

`densities` numeric vector of known cal rod densities. Must be in the same order as `means` and `sds`.

`returnData` if `TRUE`, voxel counts for each Hounsfield unit from `lowerLim` to `upperLim` are returned, as are material class definitions. These are the data needed to re-create and modify the frequency plot.

`pngName` if this is not `NULL`, the frequency plot is saved to disk. In that case, `pngName` should be a character string containing the name and address of the file.

Value

list if `returnData = TRUE`, a list is returned containing (1) the frequencies for each Hounsfield unit value from `lowerLim` to `upperLim`, (2) the boundaries for material classes, and (3) a summary of the calibration curve applied. Lower boundaries for a component class are exclusive, while upper bounds are inclusive. These materials allow the frequency distribution to be plotted by the user. If `returnData = FALSE` the data are plotted in the graphics window, but nothing is preserved.

Examples

```
# data(core_426)
coreHist("core_426", returnData = FALSE)
```

core_426 *Three computed tomography scans from a Spartina alterniflora core*

Description

Three computed tomography scans from a Spartina alterniflora core

Usage

```
data(core_426)
```

Format

A list of 3 matrices, each with two elements: header and image data

getRoots *Convert a matrix of semi-processed DICOM images to root particle counts, volumes, and surface areas*

Description

Calculates the number of root/rhizome particles, volumes, and surface areas, for different size classes. This version accommodates calibration curves with >4 calibrants, and uses density thresholds converted to Hounsfield Units using the calibration curve (rather than direct calibration rod values) to partition sediment components.

Usage

```
getRoots(mat.list, pixelA, diameter.classes = c(1, 2, 2.5, 10),
class.names = diameter.classes,
thickness = 0.625,
means     = c(-850.3233, 63.912, 271.7827, 1345.0696),
sds       = c(77.6953, 14.1728, 39.2814, 45.4129),
densities = c(0.0012, 1, 1.23, 2.2),
pixel.minimum = 4)
```

Arguments

`mat.list` list of DICOM images for a sediment core (values in Hounsfield Units)

`pixelA` pixel area (mm2)

`diameter.classes`
an integer vector of diameter cut points. Units are mm (zero is added in automatically).

`class.names` not used presently

`thickness` slice thickness for computed tomography image series (mm)

`means` mean values (units = Hounsfield Units) for calibration rods used.

`sds` standard deviations (units = Hounsfield Units) for calibration rods used. Must be in the same order as `means`.

`densities` numeric vector of known cal rod densities. Must be in the same order as `means` and `sds`.

`pixel.minimum` minimum number of pixels needed for a clump to be identified as a root

Details

Calculates the number of root/rhizome particles, volumes, and surface areas, for different size classes. This function requires that values be Hounsfield Units (i.e., data must be semi-processed from the raw DICOM imagery).

Value

value `getRoots` returns a dataframe with one row per CT slice. Values returned are the number, volume (cm3), and surface area (cm2) of particles in each size class with an upper bound defined in `diameter.classes`.

See Also

`convert`

Examples

```
ct.slope <- unique(extractHeader(core_426$hdr, "RescaleSlope"))
ct.int   <- unique(extractHeader(core_426$hdr, "RescaleIntercept"))
# convert raw units to Hounsfield units
HU_426 <- lapply(core_426$img, function(x) x*ct.slope + ct.int)

rootChars <- getRoots(HU_426, pixelA = 0.0596,
diameter.classes = c(2.5, 10))

## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
area.long <- reshape2::melt(rootChars, id.vars = c("depth"),
   measure.vars = grep("Area", names(rootChars)))
ggplot2::ggplot(data = area.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("root external surface area per slice (cm2)")
```

```
## End(Not run)
```

---

getRootsDir *Convert a directory of raw DICOM images to root particle counts and surface areas*

Description

Calculates the number of root/rhizome particles and surface areas, for different size classes

Usage

```
getRootsDir(directory = file.choose(),
diameter.classes = c(1, 2, 5, 10, 20),
class.names = diameter.classes,
means     = c(-850.3233, 63.912, 271.7827, 1345.0696),
sds       = c(77.6953, 14.1728, 39.2814, 45.4129),
densities = c(0.0012, 1, 1.23, 2.2),
pixel.minimum = 1)
```

Arguments

`directory`	a character string that can be a matrix of DICOM images or the address of an individual DICOM file in a folder of DICOM images. The default action is <code>file.choose()</code>; a browser menu appears so the user can select the the desired directory by identifying a single DICOM file in the folder of images.
`diameter.classes`	an integer vector of diameter cut points. Units are mm (zero is added in automatically).
`class.names`	not used presently
`means`	mean values (units = Hounsfield Units) for calibration rods used.
`sds`	standard deviations (units = Hounsfield Units) for calibration rods used. Must be in the same order as `means`.
`densities`	numeric vector of known cal rod densities. Must be in the same order as `means` and `sds`.
`pixel.minimum`	minimum number of pixels needed for a clump to be identified as a root

Details

Calculates the number of root/rhizome particles and surface areas, for different size classes. Unlike `getRoots`, `getRootsDir` accepts a folder of raw values and makes the conversion to Hounsfield Units using the metadata associated with the DICOM images. This version accommodates calibration curves with >4 calibrants, and uses density thresholds converted to Hounsfield Units using the calibration curve (rather than direct calibration rod values) to partition sediment components.

Value

value `getRootData` returns a dataframe with one row per CT slice. Values returned are the number, volume (cm3), and surface area (cm2) of particles in each size class with an upper bound defined in `diameter.classes`.

See Also

`getRootsDir` is a wrapper for `getRoots`. `getRootsDir` operates similarly.

Examples

```
rootChars <- getRootsDir("core_426", diameter.classes = c(2.5, 10))

## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
area.long <- reshape2::melt(rootChars, id.vars = c("depth"),
   measure.vars = grep("Area", names(rootChars)))
ggplot2::ggplot(data = area.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("root external surface area per slice (cm2)")

## End(Not run)
```

getSurface *Remove artificial surface layers from processed CT data*

Description

Identifies and removes artificial surface layers from processed CT data

Usage

```
getSurface(x, material = "particulates", threshold = 0.40,
start = "top", thickness = 0.625)
```

Arguments

`x`	dataframe created by `conv`
`material`	material used for determining where the surface begins
`threshold`	decimal fraction of total area, used to determine the surface layer. Surface slices where `material` exceeds threshold value are removed.
`start`	should core be processed from the top, bottom, or both?
`thickness`	CT image thickness (mm)

Details

Identifies and removes artificial surface layers from processed CT data. Areas can be removed from one or both ends of the core (set by `start`), based on exceeding a `threshold` proportion of material (e.g., 75

Value

value `getSurface` shortens the output of `conv` to remove artificial surface layers. The output is thus a subset of the input, and identical in structure to the /codeconv output.

See Also

`conv`

Examples

```
### Not run:
## Not run: data(core_426)
ct.slope <- unique(extractHeader(core_426$hdr, "RescaleSlope"))
ct.int   <- unique(extractHeader(core_426$hdr, "RescaleIntercept"))
# convert raw units to Hounsfield units
HU_426 <- lapply(core_426$img, function(x) x*ct.slope + ct.int)

materials <- conv(HU_426)
head(materials[, 1:6], 20)

materials2 <- getSurface(materials)
head(materials2[, 1:6])

## End(Not run)
```

---

rootSize — *Convert a matrix of semi-processed DICOM images to root particle counts, volumes, and surface areas*

Description

Calculates the number of root/rhizome particles, volumes, and surface areas, for different size classes. This approach directly replicates Earl Davey's manual classification approach. This method is deprecated as of coreCT version 1.3.0.

Usage

```
rootSize(mat.list, pixelA, diameter.classes = c(1, 2, 2.5, 10),
class.names = diameter.classes,
thickness = 0.625,
airHU = -850.3233,
airSD = 77.6953,
waterHU = 63.912,
waterSD = 14.1728,
pixel.minimum = 4)
```

Arguments

`mat.list` list of DICOM images for a sediment core (values in Hounsfield Units)

`pixelA` pixel area (mm2)

`diameter.classes` an integer vector of diameter cut points. Units are mm (zero is added in automatically).

`class.names` not used presently

`thickness` CT image thickness (mm)

`airHU` mean value for air-filled calibration rod (all rod arguments are in Hounsfield Units)

`airSD` standard deviation for air-filled calibration rod

`waterHU` mean value for water-filled calibration rod

`waterSD` standard deviation for water-filled calibration rod

`pixel.minimum` minimum number of pixels needed for a clump to be identified as a root

Details

Calculates the number of root/rhizome particles, volumes, and surface areas, for different size classes. This function requires that values be Hounsfield Units (i.e., data must be semi-processed from the raw DICOM imagery).

Value

value `rootSize` returns a dataframe with one row per CT slice. Values returned are the number, volume (cm3), and surface area (cm2) of particles in each size class with an upper bound defined in `diameter.classes`.

See Also

`conv`

Examples

```
ct.slope <- unique(extractHeader(core_426$hdr, "RescaleSlope"))
ct.int   <- unique(extractHeader(core_426$hdr, "RescaleIntercept"))
# convert raw units to Hounsfield units
HU_426 <- lapply(core_426$img, function(x) x*ct.slope + ct.int)

rootChars <- rootSize(HU_426, pixelA = 0.0596,
diameter.classes = c(2.5, 10))

## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
```

```
area.long <- reshape2::melt(rootChars, id.vars = c("depth"),
   measure.vars = grep("Area", names(rootChars)))
ggplot2::ggplot(data = area.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("root external surface area per slice (cm2)")

## End(Not run)
```

`rootSizeDir` *Convert a directory of raw DICOM images to root particle counts and surface areas*

Description

Calculates the number of root/rhizome particles and surface areas, for different size classes. This approach directly replicates Earl Davey's manual classification approach. This method is deprecated as of coreCT version 1.3.0.

Usage

```
rootSizeDir(directory = file.choose(), diameter.classes = c(1, 2, 5, 10, 20),
class.names = diameter.classes,
airHU = -850.3233,
airSD = 77.6953,
waterHU = 63.912,
waterSD = 14.1728,
pixel.minimum = 1)
```

Arguments

`directory` a character string that can be a matrix of DICOM images or the address of an individual DICOM file in a folder of DICOM images. The default action is <code>file.choose()</code>; a browser menu appears so the user can select the the desired directory by identifying a single DICOM file in the folder of images.

`diameter.classes` an integer vector of diameter cut points. Units are mm (zero is added in automatically).

`class.names` not used presently

`airHU` mean value for air-filled calibration rod (all rod arguments are in Hounsfield Units)

`airSD` standard deviation for air-filled calibration rod

`waterHU` mean value for water-filled calibration rod

`waterSD` standard deviation for water-filled calibration rod

`pixel.minimum` minimum number of pixels needed for a clump to be identified as a root

Details

Calculates the number of root/rhizome particles and surface areas, for different size classes. Unlike rootSize, rootSizeDir accepts a folder of raw values and makes the conversion to Hounsfield Units using the metadata associated with the DICOM images.

Value

value rootSize returns a dataframe with one row per CT slice. Values returned are the number, volume (cm3), and surface area (cm2) of particles in each size class with an upper bound defined in diameter.classes.

See Also

rootSizeDir is a wrapper for rootSize. rootSizeDir operates similarly.

Examples

```
rootChars <- rootSizeDir("core_426", diameter.classes = c(2.5, 10))

## Not run:
# plot using "ggplot" package after transforming with "reshape2" package
area.long <- reshape2::melt(rootChars, id.vars = c("depth"),
   measure.vars = grep("Area", names(rootChars)))
ggplot2::ggplot(data = area.long, ggplot2::aes(y = -depth, x = value,
   color = variable)) + ggplot2::geom_point() + ggplot2::theme_classic() +
   ggplot2::xlab("root external surface area per slice (cm2)")

## End(Not run)
```

---

voxDims *Extract voxel dimensions from DICOM image*

Description

Extract pixel area and slice thickness from DICOM header to characterize voxel (3D pixel) dimensions.

Usage

```
voxDims(directory = file.choose())
```

Arguments

directory	a character string that can be a matrix of DICOM images or the address of an individual DICOM file in a folder of DICOM images. The default action is <code>file.choose()</code>; a browser menu appears so the user can select the the desired directory by identifying a single DICOM file in the folder of images.

Value

value voxDims returns a two-column dataframe showing the pixel area and slice thickness. Values in the DICOM headers are assumed to be millimeters; pixel area and slice thickness columns are labeled based on this assumption.

Examples

```
# data(core_426)
voxDims("core_426")
```

Index