

Package ‘DVHmetrics’

November 27, 2015

Type Package

Title Analyze Dose-Volume Histograms and Check Constraints

Version 0.3.4

Date 2015-11-27

Depends R (>= 3.0.0)

Imports ggplot2, grid, markdown, reshape2, shiny, KernSmooth, stats,
utils

Suggests knitr

VignetteBuilder knitr

LazyData yes

Description Functionality for analyzing dose-volume histograms (DVH) in radiation oncology: Read DVH text files, calculate DVH metrics as well as generalized equivalent uniform dose (gEUD), biologically effective dose (BED), equivalent dose in 2 Gy fractions (EQD2), normal tissue complication probability (NTCP), and tumor control probability (TCP). Show DVH diagrams, check and visualize quality assurance constraints for the DVH. Includes web-based graphical user interface.

License GPL (>= 2)

URL <https://github.com/dwoll/DVHmetrics/>

NeedsCompilation no

Author Daniel Wollschlaeger [aut, cre],
Heiko Karle [aut],
Heinz Schmidberger [ctb]

Maintainer Daniel Wollschlaeger <wollschlaeger@uni-mainz.de>

Repository CRAN

Date/Publication 2015-11-27 14:59:38

R topics documented:

DVHmetrics-package	2
checkConstraint	3
convertDVH	6
dataConstr	7
dataMZ	8
getBED	9
getDMEAN	10
getEQD2	12
getEUD	13
getIsoEffD	14
getMeanDVH	16
getMetric	17
getNTCP	21
getTCP	22
mergeDVH	24
print.DVHs	25
readConstraint	26
readDVH	27
runGUI	29
saveConstraint	30
saveDVH	31
saveMetric	32
showConstraint	33
showDVH	34
Index	37

DVHmetrics-package *Analyze Dose-Volume Histograms and Check Constraints*

Description

Functionality for analyzing dose-volume histograms (DVH) in radiation oncology: Read DVH text files, calculate DVH metrics, gEUD, BED, EQD2, NTCP, TCP, show DVH diagrams, check and visualize quality assurance constraints for the DVH. Includes web-based graphical user interface.

Details

Package: DVHmetrics
 Type: Package
 Version: 0.3.4
 Date: 2015-11-27
 License: GPL (>= 2)

Author(s)

Daniel Wollschlaeger and Heiko Karle

Maintainer: Daniel Wollschlaeger <wollschlaeger@uni-mainz.de>

References

For a solution that also reads files in DICOM-RT format, see the RadOnc package: <http://CRAN.R-project.org/package=RadOnc>.

Examples

```
showDVH(dataMZ[[1]])
checkConstraint(dataMZ, "D1CC < 10Gy")
```

checkConstraint	<i>Check constraints on dose-volume histograms (DVH)</i>
-----------------	--

Description

Simultaneously checks one or more quality assurance constraints on one or more DVHs. Reports compliance with each constraint as well as observed difference between linearly interpolated DVHs and the given constraints in terms of (relative) dose, (relative) volume, and (relative) minimal Euclidean distance.

Usage

```
checkConstraint(x, constr, byPat=TRUE, semSign=FALSE,
               sortBy=c("none", "observed", "compliance", "structure",
                        "constraint", "patID", "deltaV", "deltaD",
                        "dstMin", "dstMinRel"),
               interp=c("linear", "spline", "smooth"), ...)
```

```
## S3 method for class 'DVHs'
checkConstraint(x, constr, byPat=TRUE, semSign=FALSE,
               sortBy=c("none", "observed", "compliance", "structure",
                        "constraint", "patID", "deltaV", "deltaD",
                        "dstMin", "dstMinRel"),
               interp=c("linear", "spline", "smooth"), ...)
```

```
## S3 method for class 'DVHlst'
checkConstraint(x, constr, byPat=TRUE, semSign=FALSE,
               sortBy=c("none", "observed", "compliance", "structure",
                        "constraint", "patID", "deltaV", "deltaD",
                        "dstMin", "dstMinRel"),
```

```

interp=c("linear", "spline", "smooth"), ...)

## S3 method for class 'DVHLstLst'
checkConstraint(x, constr, byPat=TRUE, semSign=FALSE,
               sortBy=c("none", "observed", "compliance", "structure",
                        "constraint", "patID", "deltaV", "deltaD",
                        "dstMin", "dstMinRel"),
               interp=c("linear", "spline", "smooth"), ...)

```

Arguments

x	A single DVH (object of class DVHs), multiple DVHs from one patient/structure (object of class DVHLst), or multiple DVHs from many patients/structures (object of class DVHLstLst). See readDVH .
constr	One or more constraints - given as a character vector or as a data.frame. See Details .
byPat	logical. Relevant if multiple DVHs are given. If x has class DVHLst: byPat=TRUE means that the DVHs are for one patient with multiple structures. byPat=FALSE means that the DVHs are for one structure from multiple patients. If x has class DVHLstLst: byPat=TRUE means that the DVHs are for multiple patients (list components of x) with multiple structures. byPat=FALSE means that the DVHs are for multiple structures (list components of x) from multiple patients.
semSign	logical. Meaning of the sign of the observed dose/volume differences between DVHs and constraints. semSign=TRUE means that negative differences indicate constraint compliance, positive differences indicate constraint violations. With semSign=FALSE, the algebraic differences are returned as is.
sortBy	character vector. Sorting criteria for the output data frame.
interp	character. Method of interpolation between DVH points: Linear interpolation using approx , monotone Hermite spline interpolation using spline , or local polynomial regression using locpoly with kernel bandwidth chosen by the direct plug-in method using dpill .
...	Additional parameters passed to getMetric . Use for constraints on EUD (see getEUD for parameter names), TCP (see getTCP), and NTCP (see getNTCP).

Details

A DVH constraint is a character string that consists of three parts: The DVH metric, the comparison operator (<, >, <=, >=), and the reference value together with the measurement unit. See [getMetric](#) for defining a DVH metric, as well as for possible measurement units for dose and volume. For constraints involving the relative dose, the DVH must contain the prescription dose.

Some example constraints are "V10Gy > 80%" (more than 80% of the structure should have received 10Gy), "V20% < 10CC" (less than 10cm³ of the structure should have received 20% of the prescription dose), or "D10CC > 500cGy" (The "hottest" 10cm³ of the structure should have received more than 500cGy).

For constraints on DEUD, DNTCP and DTCP (see [getMetric](#)), the reference measurement unit must be Gy, cGy, even though NTCP and TCP are probabilities. Example: "DNTCP < 0.5Gy".

A DVH constraint can apply to a specific patient or to all patients, and to a specific structure or to all structures.

- If constraints apply to all patients/structures in `x`, `constr` can be a character vector with elements like the examples above.
- If constraints apply only to some patients/structures, `constr` must be a data frame with variables `constraint`, `patID` and `structure`. Each row then defines one constraint and its scope: `constraint` must be a character string with one constraint definition as in the examples above. `patID` must be either a character string with a valid patient ID or "*" if the constraint applies to all patients. `structure` must be either a character string with a valid structure or "*" if the constraint applies to all structures. If variable `patID` is missing from the data frame, the constraints apply to all available patients. If variable `structure` is missing from the data frame, the constraints apply to all available structures. See [readConstraint](#) for reading appropriate constraint data frames from external text files.

For calculating the minimal Euclidean distance between the constraint point and the DVH, the constraint point is orthogonally projected onto each DVH segment between (interpolated) DVH nodes. The relative Euclidean distance is the minimum of these distances divided by the distance of the constraint point to the closer one of both axes (dose and volume).

If volume or dose values outside the range of possible values for a structure are requested, metrics cannot be calculated, and the result will be NA with a warning.

Value

A data frame with details on constraint compliance / violation.

<code>patID</code>	Patient ID
<code>structure</code>	Structure
<code>constraint</code>	The checked constraint
<code>observed</code>	The observed value for the metric given in the constraint
<code>compliance</code>	Does the DVH satisfy the constraint?
<code>deltaV</code>	Volume difference between constraint and observed DVH (for the constraint dose) in measurement unit specified by constraint
<code>deltaVpc</code>	Percent volume difference between constraint and observed DVH (for the constraint dose) relative to constraint volume
<code>deltaD</code>	Dose difference between constraint and observed DVH (for the constraint volume) in measurement unit specified by constraint
<code>deltaDpc</code>	Percent dose difference between constraint and observed DVH (for the constraint volume) relative to constraint dose
<code>dstMin</code>	Minimal Euclidean distance between constraint and the cumulative DVH, using linear interpolation
<code>ptMinD</code>	Dose coordinate of closest point on cumulative DVH to constraint
<code>ptMinV</code>	Volume coordinate of closest point on cumulative DVH to constraint

See Also

[getMetric](#), [getEUD](#), [getNTCP](#), [getTCP](#), [readConstraint](#), [saveConstraint](#), [showConstraint](#)

Examples

```

res <- checkConstraint(dataMZ, c("D10CC < 10Gy", "V20Gy < 20%"))
head(res)

# define constraints
constr <- data.frame(
  patID=c("P123", "P234"),
  structure=c("HEART", "*"),
  constraint=c("D1CC < 20Gy", "V10% > 8CC"),
  stringsAsFactors=FALSE) # this is important
checkConstraint(dataMZ, constr=constr)

```

convertDVH

Convert between differential and cumulative DVH

Description

Convert between differential and cumulative DVH as well as between dose units.

Usage

```

convertDVH(x, toType=c("asis", "cumulative", "differential"),
  toDoseUnit=c("asis", "GY", "CGY"),
  interp=c("asis", "linear", "spline", "ksmooth", "smoothSpl"),
  nodes=NULL, rangeD=NULL, perDose=TRUE)

## S3 method for class 'matrix'
convertDVH(x, toType=c("asis", "cumulative", "differential"),
  toDoseUnit=c("asis", "GY", "CGY"),
  interp=c("asis", "linear", "spline", "ksmooth", "smoothSpl"),
  nodes=NULL, rangeD=NULL, perDose=TRUE)

## S3 method for class 'DVHs'
convertDVH(x, toType=c("asis", "cumulative", "differential"),
  toDoseUnit=c("asis", "GY", "CGY"),
  interp=c("asis", "linear", "spline", "ksmooth", "smoothSpl"),
  nodes=NULL, rangeD=NULL, perDose=TRUE)

## S3 method for class 'DVHLst'
convertDVH(x, toType=c("asis", "cumulative", "differential"),
  toDoseUnit=c("asis", "GY", "CGY"),
  interp=c("asis", "linear", "spline", "ksmooth", "smoothSpl"),
  nodes=NULL, rangeD=NULL, perDose=TRUE)

## S3 method for class 'DVHLstLst'
convertDVH(x, toType=c("asis", "cumulative", "differential"),
  toDoseUnit=c("asis", "GY", "CGY"),

```

```
interp=c("asis", "linear", "spline", "ksmooth", "smoothSpl"),
nodes=NULL, ranged=NULL, perDose=TRUE)
```

Arguments

x	One DVH (object of class <code>matrix</code> or <code>DVHs</code> , multiple cumulative DVHs from one patient with multiple structures (object of class <code>DVHLst</code>), or multiple cumulative DVHs from many patients, each with multiple structures (object of class <code>DVHLstLst</code>). See readDVH .
toType	character. Convert the DVH to this type. "asis" keeps the current DVH type.
toDoseUnit	character. Convert the DVH to this dose unit. "asis" keeps the current dose unit.
interp	character. Interpolation method for the differential DVH. "asis" and "linear" for no interpolation. "spline" for spline interpolation using splinefun ("fmm" for differential, "monoH.FC" for cumulative DVHs), "ksmooth" for local polynomial regression using locpoly with kernel bandwidth chosen by the direct plug-in method using dpill , "smoothSpl" for a smoothing spline using smooth.spline , with the smoothing parameter chosen by generalized crossvalidation.
nodes	numeric. Minimum number of nodes to use in interpolation for method "ksmooth". Number of available nodes is kept as is for NULL or if larger than nodes.
ranged	numeric. Dose range for interpolation methods "linear", "spline", "smoothSpl". If NULL it is determined individually for each DVH.
perDose	logical. Are the differential DVH volume values per unit dose?

Value

Depending on the input, an object of class `matrix`, `DVHs`, `DVHLst`, or `DVHLstLst`.

See Also

[readDVH](#), [showDVH](#)

Examples

```
res <- convertDVH(dataMZ[[c(1, 1)]],
                  toType="cumulative",
                  toDoseUnit="CGY")
```

dataConstr

Constraint data frame

Description

Data frame with quality assurance constraints to use with built-in DVH object [dataMZ](#).

Usage

```
data(dataConstr)
```

Format

A data frame with 6 entries for the following 3 variables.

constraint The constraint character string.

patID The patient ID character string or * wildcard.

structure The structure character string or * wildcard.

Details

See [checkConstraint](#) for the definition of a constraint.

See Also

[readConstraint](#) [checkConstraint](#) [showConstraint](#)

Examples

```
checkConstraint(dataMZ, constr=dataConstr)
```

dataMZ

DVH data from 3 patients

Description

Data from 3 patients with radiotherapy. DVHs for 7 heart structures.

Usage

```
data(dataMZ)
```

Format

Object of class DVHLstLst with 3 components corresponding to 3 patients.

P123 Object of class DVHLst. 7 objects of class DVHs for structures AMYOCL (left anterior heart wall), AMYOCR (right anterior heart wall), AOVALVE (aortic valve), AVNODE (AV node), HEART (complete heart), PULMVALVE (pulmonary valve), MYOCARD (heart wall)

P234 Object of class DVHLst. 7 objects of class DVHs for the same structures as patient P123.

P345 Object of class DVHLst. 7 objects of class DVHs for the same structures as patient P123.

Details

Data courtesy of Department of Radiation Oncology (Prof. Dr. Schmidberger), University Medical Center Mainz, Germany.

See [readDVH](#) for classes DVHLstLst, DVHLst, and DVHs.

See Also

[readDVH](#), [print.DVHs](#)

Examples

```
print(dataMZ, verbose=TRUE)
```

getBED	<i>Calculate biologically effective dose (BED)</i>
--------	--

Description

Calculate biologically effective dose (BED) according to the linear-quadratic model.

Usage

```
getBED(D=NULL, fd=NULL, fn=NULL, ab=NULL)

## Default S3 method:
getBED(D=NULL, fd=NULL, fn=NULL, ab=NULL)

## S3 method for class 'DVHs'
getBED(D=NULL, fd=NULL, fn=NULL, ab=NULL)

## S3 method for class 'DVHLst'
getBED(D=NULL, fd=NULL, fn=NULL, ab=NULL)

## S3 method for class 'DVHLstLst'
getBED(D=NULL, fd=NULL, fn=NULL, ab=NULL)
```

Arguments

D	Default: Total dose. If NULL, fn must be given. Alternative: One cumulative DVH (object of class DVHs, multiple cumulative DVHs from one patient with multiple structures (object of class DVHLst), or multiple cumulative DVHs from many patients, each with multiple structures (object of class DVHLstLst). See readDVH .
fd	Fractional dose. If D is some kind of DVH object, only the first element will be used.

fn	Number of fractions. If NULL, D must be the total dose. Ignored if D is some kind of DVH object.
ab	alpha/beta ratio for the relevant tissue. If some kind of DVH object, only the first element will be used.

Value

Default method: A data frame with variables BED, fractDose, ab.

If D is some kind of DVH object, the same kind of object is returned with the individual dose values converted to BED.

References

Fowler, J. F. (2010). 21 years of Biologically Effective Dose. *British Journal of Radiology*, 83, 554-568.

See Also

[getEQD2](#), [getIsoEffD](#)

Examples

```
getBED(D=50, fd=2.5, ab=c(2, 3, 4))
getBED(D=dataMZ[[c(1, 1)]], fd=1.8, ab=3)
```

getDMEAN	<i>DMEAN and other dose metrics</i>
----------	-------------------------------------

Description

Calculate DMEAN and other dose metrics from the (interpolated) differential DVH without relying on the values exported by the TPS.

Usage

```
getDMEAN(x, interp=c("linear", "spline", "ksmooth", "smoothSpl"),
         nodes=5001L)

## S3 method for class 'DVHs'
getDMEAN(x, interp=c("linear", "spline", "ksmooth", "smoothSpl"),
         nodes=5001L)

## S3 method for class 'DVHLst'
getDMEAN(x, interp=c("linear", "spline", "ksmooth", "smoothSpl"),
         nodes=5001L)

## S3 method for class 'DVHLstLst'
getDMEAN(x, interp=c("linear", "spline", "ksmooth", "smoothSpl"),
         nodes=5001L)
```

Arguments

x	One DVH (object of class DVHs, multiple DVHs from one patient with multiple structures (object of class DVHLst), or multiple DVHs from many patients, each with multiple structures (object of class DVHLstLst). See readDVH .
interp	character. Method of interpolation between DVH points: Linear interpolation using approx , spline interpolation using splinefun ("fmm" for differential, "monoH.FC" for cumulative DVH), local polynomial regression using locpoly with kernel bandwidth chosen by the direct plug-in method using dpill , or a smoothing spline using smooth.spline with smoothing parameter selected by generalized crossvalidation.
nodes	numeric. Minimum number of nodes to use in interpolation. Number of available nodes is kept as is for NULL or if larger than nodes.

Value

A data frame with the following value(s).

patID Patient ID.

structure Structure name.

doseMin Minimum dose.

doseMax Maximum dose.

doseAvg Mean dose.

doseMed Median dose.

doseSD Dose standard deviation.

doseMode Dose mode.

doseAvgTPS Mean dose as exported from the TPS (if available).

doseMedTPS Median dose as exported from the TPS (if available).

doseMinTPS Minimum dose as exported from the TPS (if available).

doseMaxTPS Maximum dose as exported from the TPS (if available).

See Also

[getMetric](#), [approxfun](#), [splinefun](#), [smooth.spline](#), [dpill](#), [locpoly](#)

Examples

```
getDMEAN(dataMZ[[1]], interp="linear")
```

 getEQD2

 2Gy fractions biologically equivalent dose (EQD2)

Description

Calculate dose in 2Gy fractions biologically equivalent dose according to the linear-quadratic model.

Usage

```
getEQD2(D=NULL, fd=NULL, fn=NULL, ab=NULL)
```

```
## Default S3 method:
```

```
getEQD2(D=NULL, fd=NULL, fn=NULL, ab=NULL)
```

```
## S3 method for class 'DVHs'
```

```
getEQD2(D=NULL, fd=NULL, fn=NULL, ab=NULL)
```

```
## S3 method for class 'DVHLst'
```

```
getEQD2(D=NULL, fd=NULL, fn=NULL, ab=NULL)
```

```
## S3 method for class 'DVHLstLst'
```

```
getEQD2(D=NULL, fd=NULL, fn=NULL, ab=NULL)
```

Arguments

D	Default: Total dose. If NULL, fn must be given. Alternative: One cumulative DVH (object of class DVHs, multiple cumulative DVHs from one patient with multiple structures (object of class DVHLst), or multiple cumulative DVHs from many patients, each with multiple structures (object of class DVHLstLst). See readDVH .
fd	Fractional dose. If D is some kind of DVH object, only the first element will be used.
fn	Number of fractions. If NULL, D must be given. Ignored if D is some kind of DVH object.
ab	alpha/beta ratio for the relevant tissue. If D is some kind of DVH object, only the first element will be used.

Details

EQD2 is a special case of isoeffective dose calculation with fractional dose $d_2=2$, see [getIsoEffD](#).

Value

Default method: A data frame with variables EQD2, fractDose, ab.

If D is some kind of DVH object, the same kind of object is returned with the individual dose values converted to EQD2.

References

IAEA, & ICRU. (2008). Relative biological effectiveness in ion-beam therapy (Tech. Rep. No. IAEA-TR 461). Vienna, Austria: IAEA (International Atomic Energy Agency) and ICRU (International Commission on Radiation Units and Measurements).

See Also

[getBED](#), [getIsoEffD](#)

Examples

```
getEQD2(D=50, fd=2.5, ab=c(2, 3, 4))
getEQD2(dataMZ[[c(1, 1)]], fd=1.8, ab=3)
```

<code>getEUD</code>	<i>Generalized equivalent uniform dose (gEUD)</i>
---------------------	---

Description

Calculate generalized equivalent uniform dose (gEUD). May be based on EQD2.

Usage

```
getEUD(x, EUDa, EUDfd=NULL, EUDab=NULL, ...)

## S3 method for class 'DVHs'
getEUD(x, EUDa, EUDfd=NULL, EUDab=NULL, ...)

## S3 method for class 'DVHLst'
getEUD(x, EUDa, EUDfd=NULL, EUDab=NULL, ...)

## S3 method for class 'DVHLstLst'
getEUD(x, EUDa, EUDfd=NULL, EUDab=NULL, ...)
```

Arguments

<code>x</code>	One cumulative DVH (object of class <code>DVHs</code> , multiple cumulative DVHs from one patient with multiple structures (object of class <code>DVHLst</code>), or multiple cumulative DVHs from many patients, each with multiple structures (object of class <code>DVHLstLst</code>). See readDVH .
<code>EUDa</code>	Exponential parameter <code>a</code> .
<code>EUDfd</code>	If <code>gEUD</code> should be based on EQD2: Fraction dose.
<code>EUDab</code>	If <code>gEUD</code> should be based on EQD2: alpha/beta ratio for the relevant tissue.
<code>...</code>	Ignored. Used to catch additional arguments passed from getMetric .

Value

A data frame with variables EUD, patID, and structure.

References

Niemierko, A. (1999). A generalized concept of equivalent uniform dose. *Medical Physics*, 26(6), 1100.

Wu et al. (2002). Optimization of intensity modulated radiotherapy plans based on the equivalent uniform dose. *International Journal of Radiation Oncology Biology Physics*, 52, 224-235.

See Also

[getEQD2](#), [getMetric](#)

Examples

```
getEUD(dataMZ[[1]], EUDa=2)

# based on EQD2
getEUD(dataMZ[[1]], EUDa=2, EUDfd=1.8, EUDab=4)
```

getIsoEffD

Isoeffective dose calculation

Description

Convert given (fractional) dose into a corresponding (fractional) dose for a different total dose / fractionation schedule according to the linear-quadratic model.

Usage

```
getIsoEffD(D1=NULL, D2=NULL, fd1=NULL, fd2=NULL, ab=NULL)

## Default S3 method:
getIsoEffD(D1=NULL, D2=NULL, fd1=NULL, fd2=NULL, ab=NULL)

## S3 method for class 'DVHs'
getIsoEffD(D1=NULL, D2=NULL, fd1=NULL, fd2=NULL, ab=NULL)

## S3 method for class 'DVHlst'
getIsoEffD(D1=NULL, D2=NULL, fd1=NULL, fd2=NULL, ab=NULL)

## S3 method for class 'DVHlstLst'
getIsoEffD(D1=NULL, D2=NULL, fd1=NULL, fd2=NULL, ab=NULL)
```

Arguments

D1	Default: numeric vector. Total dose 1. Alternative: One cumulative DVH (object of class DVHs, multiple cumulative DVHs from one patient with multiple structures (object of class DVHLst), or multiple cumulative DVHs from many patients, each with multiple structures (object of class DVHLstLst). See readDVH .
D2	numeric vector. Total dose 2. Ignored if D is some kind of DVH object.
fd1	numeric vector. Fractional dose 1. If D is some kind of DVH object, only the first element will be used.
fd2	numeric vector. Fractional dose 2. If D is some kind of DVH object, only the first element will be used.
ab	numeric vector. alpha/beta ratio for the relevant tissue in the linear-quadratic model. If D is some kind of DVH object, only the first element will be used.

Details

DVH methods: Calculate D2 based on D1, fd1, fd2, and ab. The default method can also calculate fd2 based on D1, D2, fd1, and ab.

Value

The (vector of) corresponding (fractional) dose value(s). If D is some kind of DVH object, the same kind of object is returned with the individual dose values converted to D2.

References

IAEA, & ICRU. (2008). Relative biological effectiveness in ion-beam therapy (Tech. Rep. No. IAEA-TR 461). Vienna, Austria: IAEA (International Atomic Energy Agency) and ICRU (International Commission on Radiation Units and Measurements).

See Also

[getBED](#), [getEQD2](#)

Examples

```
# reference: 70Gy in 2Gy fractions
# new fractionation: 3Gy fractions
# calculate corresponding dose
(D2 <- getIsoEffD(D1=70, fd1=2, fd2=3, ab=c(3.5, 10)))

getIsoEffD(D1=dataMZ[[c(1, 1)]], fd1=1.8, fd2=2, ab=3.5)
```

getMeanDVH

Point-wise mean DVH with point-wise SDs

Description

Returns the point-wise mean and median DVH with the point-wise standard deviation for a given list of input DVHs. Other point-wise measures may be calculated as well.

Usage

```
getMeanDVH(x, fun=list(mean=mean, median=median, sd=sd),
           cumul=TRUE, thin=1, byPat=TRUE, patID=NULL, structure=NULL,
           interp=c("linear", "spline", "smoothSpl"), fixed=TRUE)

## S3 method for class 'DVHs'
getMeanDVH(x, fun=list(mean=mean, median=median, sd=sd),
           cumul=TRUE, thin=1, byPat=TRUE, patID=NULL, structure=NULL,
           interp=c("linear", "spline", "smoothSpl"), fixed=TRUE)

## S3 method for class 'DVHLst'
getMeanDVH(x, fun=list(mean=mean, median=median, sd=sd),
           cumul=TRUE, thin=1, byPat=TRUE, patID=NULL, structure=NULL,
           interp=c("linear", "spline", "smoothSpl"), fixed=TRUE)

## S3 method for class 'DVHLstLst'
getMeanDVH(x, fun=list(mean=mean, median=median, sd=sd),
           cumul=TRUE, thin=1, byPat=TRUE, patID=NULL, structure=NULL,
           interp=c("linear", "spline", "smoothSpl"), fixed=TRUE)
```

Arguments

x	A single DVH (object of class DVHs), multiple DVHs from one patient/structure (object of class DVHLst), or multiple DVHs from many patients/structures (object of class DVHLstLst). See readDVH .
fun	Named list of functions that should be applied to yield 1 point-wise DVH measure. Functions must have exactly 1 return value.
cumul	logical. Get point-wise mean and SD for cumulative or differential (per unit dose) DVH?
thin	numeric. The number of DVH nodes (dose values) is reduced by 1/thin of the maximum number of nodes in x before interpolating and averaging.
byPat	logical. Relevant if multiple DVHs are given. byPat=TRUE means that for each patient, DVHs for multiple structures are averaged point wise. byPat=FALSE means that for each structure, DVHs for multiple patients averaged point wise.
patID	character vector. Include DVHs for these patients only when calculating mean/SD. If missing, all patients are used. Can be a regular expression with fixed=FALSE, see regex .

structure	character vector. Include DVHs for these structures only when calculating mean/SD. If missing, all structures are used. Can be a regular expression with fixed=FALSE, see regex .
interp	character. Interpolation method for the differential DVH. "asis" and "linear" for no interpolation. "spline" for spline interpolation using splinefun ("fmm" for differential, "monoH.FC" for cumulative DVHs), "smoothSpl" for a smoothing spline using smooth.spline , with the smoothing parameter chosen by generalized crossvalidation.
fixed	logical. Use fixed=FALSE for regular expression matching of patID and structure.

Details

Before calculating the point-wise mean and SD, DVHs in x are first linearly interpolated with [convertDVH](#) using the same set of nodes.

Value

Returns a data frame with point-wise mean DVH averaged over structures (byPat=TRUE) or over patients (byPat=FALSE) including the point-wise standard deviation or other measures as controlled by fun.

See Also

[showDVH](#), [convertDVH](#)

Examples

```
res1 <- getMeanDVH(dataMZ, byPat=TRUE, structure=c("HEART", "AMYOCL"))
head(res1)

# average differential DVHs, use kernel smoothing
# matches patients P123 and P234
res2 <- getMeanDVH(dataMZ, fun=list(min=min, max=max),
                   cumul=FALSE, byPat=FALSE,
                   patID="23", interp="smoothSpl", fixed=FALSE)
head(res2)
```

getMetric

Calculate dose-volume-histogram metrics

Description

Simultaneously calculates multiple metrics for multiple cumulative DVHs.

Usage

```

getMetric(x, metric, patID, structure,
          sortBy=c("none", "observed", "patID", "structure", "metric"),
          splitBy=c("none", "patID", "structure", "metric"),
          interp=c("linear", "spline", "ksmooth"), fixed=TRUE, ...)

## S3 method for class 'DVHs'
getMetric(x, metric, patID, structure,
          sortBy=c("none", "observed", "patID", "structure", "metric"),
          splitBy=c("none", "patID", "structure", "metric"),
          interp=c("linear", "spline", "ksmooth"), fixed=TRUE, ...)

## S3 method for class 'DVHLst'
getMetric(x, metric, patID, structure,
          sortBy=c("none", "observed", "patID", "structure", "metric"),
          splitBy=c("none", "patID", "structure", "metric"),
          interp=c("linear", "spline", "ksmooth"), fixed=TRUE, ...)

## S3 method for class 'DVHLstLst'
getMetric(x, metric, patID, structure,
          sortBy=c("none", "observed", "patID", "structure", "metric"),
          splitBy=c("none", "patID", "structure", "metric"),
          interp=c("linear", "spline", "ksmooth"), fixed=TRUE, ...)

```

Arguments

x	One cumulative DVH (object of class DVHs, multiple cumulative DVHs from one patient with multiple structures (object of class DVHLst), or multiple cumulative DVHs from many patients, each with multiple structures (object of class DVHLstLst). See readDVH .
metric	character vector defining one or more DVH metrics. See Details for their definition. For metrics involving the relative dose, the DVH must contain the prescription dose.
patID	character vector. Calculate given DVH metrics for these patients only. If missing, DVH metrics are calculated for all patients. Can be a regular expression if additional argument <code>fixed=FALSE</code> is supplied as well, see regex .
structure	character vector. Calculate given DVH metrics for these structures only. If missing, DVH metrics are calculated for all structures. Can be a regular expression if additional argument <code>fixed=FALSE</code> is supplied as well, see regex .
sortBy	character vector giving the sorting criteria for the output data frame.
splitBy	character vector. Split results into a list of data frames where list components are defined by groups from combining these variables.
interp	character. Method of interpolation between DVH points: Linear interpolation using approx , monotone Hermite spline interpolation using splinefun , or local polynomial regression using locpoly with kernel bandwidth chosen by the direct plug-in method using dpill .

fixed	logical. Use fixed=FALSE for regular expression matching of patID and structure.
...	Further arguments passed to getEUD (for metric="DEUD"), getTCP (for metric="DTCP"), or getNTCP (for metric="DNTCP").

Details

A *pre-specified* DVH metric is one of the following character strings:

- "DMEAN": The volume-weighted mean dose of the structure.
- "DMEDIAN": Median dose, equal to D50%
- "DMIN": The minimum dose of the non-zero-dose voxels in the structure.
- "DMAX": The maximum dose of the non-zero-dose voxels in the structure.
- "DSD": The standard deviation of the dose in the structure.
- "DRX": The prescription dose.
- "DEUD": The generalized equivalent uniform dose (gEUD). See [getEUD](#) for mandatory and optional parameters.
- "DNTCP": The normal tissue complication probability (NTCP). See [getNTCP](#) for mandatory and optional parameters.
- "DTCP": The tumor control probability (TCP). See [getNTCP](#) for mandatory and optional parameters.

A *free* DVH metric is a character string which has three mandatory elements and one optional element in the following order:

- 1st letter "D" or "V": "D" if the requested value is a dose, "V" if it is a volume.
- 2nd element <number>: If the first letter is "D", this gives the volume for which the dose value of the cumulative DVH should be reported. If the first letter is "V", this gives the dose for which the volume value of the cumulative DVH should be reported.
- 3rd element <measurement unit>: The measurement unit for the 2nd element of the metric. Absolute volumes are indicated by "CC" for cubic centimeter, relative volumes by "%". Absolute doses are indicated by "Gy" for Gray or "cGy" for Centigray, relative doses by "%".
- Optional 4th element _<measurement unit>: The measurement unit of the output value. Possible units are as for the 3rd element. If missing, dose is reported as absolute dose in the measurement unit used in the DVH. Volume is reported as relative volume in %.

Examples:

- "D1%": Minimal absolute dose for the "hottest" 1% of the structure, i.e., the maximally irradiated 1% of the structure was exposed to at least this absolute dose.
- "D1CC_%": Minimal relative dose (% of prescription dose) for the maximally irradiated cm³ of the structure.
- "V500cGy": Relative structure volume in % that was exposed to at least 500cGy.
- "V10%_CC": Absolute structure volume in cm³ that was exposed to at least 10% of prescription dose.

If volume or dose values outside the range of possible values for a structure are requested, metrics cannot be calculated, and the result will be NA with a warning.

DMEAN, DMEDIAN, DMIN, DMAX, DSD are taken from the exported DVH if present. Otherwise, the differential DVH is generated and used for calculating these metrics.

Value

A data frame or a list with details on the calculated metrics.

patID	Patient ID
structure	Structure
metric	The calculated DVH metric
observed	The observed value for the DVH metric

References

Rancati et al. (2004). Fitting late rectal bleeding data using different NTCP models: results from an Italian multi-centric study (AIROPROS0101). *Radiotherapy Oncology*, 73, 21-32.

Wu et al. (2002). Optimization of intensity modulated radiotherapy plans based on the equivalent uniform dose. *International Journal of Radiation Oncology Biology Physics*, 52, 224-235.

See Also

[saveMetric](#), [getEUD](#), [getNTCP](#), [getTCP](#), [getEQD2](#), [approxfun](#), [splinefun](#), [dpill](#), [locpoly](#)

Examples

```
getMetric(dataMZ, c("D1CC", "V10%_CC"),
          sortBy=c("metric", "structure", "observed"))

# matching patients are P123 and P234
# matching structures are AMYOCL and AMYOCL
getMetric(dataMZ, c("D1CC", "V10%_CC"),
          patID="23",
          structure=c("AMYOC", "VALVE"),
          splitBy="patID",
          fixed=FALSE)

# gEUD with a=2
getMetric(dataMZ[[c(1, 1)]], "DEUD", EUDa=2)

# gEUD based on EQD2 with a=2, 20 fractions
getMetric(dataMZ[[c(1, 1)]], "DEUD", EUDa=2, EUDfd=1.8)

# NTCP Lyman probit model with TD50=20, m=4, n=0.5
getMetric(dataMZ[[c(1, 1)]], "DNTCP",
          NTCPtd50=20, NTCPm=4, NTCPn=0.5, NTCPtype="probit")
```

getNTCP *Normal tissue complication probability (NTCP)*

Description

Calculate normal tissue complication probability (NTCP) from Lyman's probit model, Niemierko's logit model, or Kaellman's Poisson model. May be based on EQD2.

Usage

```
getNTCP(x, NTCPTd50=NULL, NTCpm=NULL, NTCpn=NULL, NTCPgamma50=NULL,
        EUDa=NULL, EUDfn=NULL, EUDab=NULL,
        NTCptype=c("probit", "logit", "poisson"), ...)
```

```
## S3 method for class 'DVHs'
getNTCP(x, NTCPTd50=NULL, NTCpm=NULL, NTCpn=NULL, NTCPgamma50=NULL,
        EUDa=NULL, EUDfn=NULL, EUDab=NULL,
        NTCptype=c("probit", "logit", "poisson"), ...)
```

```
## S3 method for class 'DVHLst'
getNTCP(x, NTCPTd50=NULL, NTCpm=NULL, NTCpn=NULL, NTCPgamma50=NULL,
        EUDa=NULL, EUDfn=NULL, EUDab=NULL,
        NTCptype=c("probit", "logit", "poisson"), ...)
```

```
## S3 method for class 'DVHLstLst'
getNTCP(x, NTCPTd50=NULL, NTCpm=NULL, NTCpn=NULL, NTCPgamma50=NULL,
        EUDa=NULL, EUDfn=NULL, EUDab=NULL,
        NTCptype=c("probit", "logit", "poisson"), ...)
```

Arguments

x	One cumulative DVH (object of class DVHs, multiple cumulative DVHs from one patient with multiple structures (object of class DVHLst), or multiple cumulative DVHs from many patients, each with multiple structures (object of class DVHLstLst). See readDVH .
NTCPTd50	Tolerance dose with 50% complication probability.
NTCPm	Probit/logit Parameter m. Equal to $1 / (NTCPgamma50 * \sqrt{2 * \pi})$.
NTCPn	Parameter n. Equal to $1/a$ with exponential gEUD parameter a.
NTCPgamma50	Poisson parameter gamma50. Equal to $1 / (NTCPm * \sqrt{2 * \pi})$
EUDa	If gEUD should be based on EQD2: Exponential parameter a.
EUDfn	If gEUD should be based on EQD2: Number of fractions.
EUDab	If gEUD should be based on EQD2: alpha/beta ratio for the relevant tissue.
NTCptype	"probit" - Lyman probit model, "logit" - Niemierko logit model, "poisson" - Kaellman Poisson (relative seriality) model.
...	Ignored. Used to catch additional arguments passed from getMetric .

Details

For DVH reduction, gEUD is used. This is equivalent to the Kutcher-Burman DVH reduction scheme.

Value

A data frame with variables NTCP, patID, and structure.

References

Kaellman, P., Agren, A., & Brahme, A. (1992). Tumor and normal tissue responses to fractionated non-uniform dose delivery. *International Journal of Radiation Biology*, 62(2), 249-262.

Kutcher, G. J., Burman, C., Brewster L., Goitein, M., & Mohan, R. (1991). Histogram reduction method for calculating complication probabilities for threedimensional treatment planning evaluations. *International Journal of Radiation Oncology Biology Physics*, 21(1), 137-146.

Lyman, J. T. (1985). Complication probability as assessed from dose volume histograms. *Radiation Research*, 104(2), S13-19.

Niemierko, A. (1999). A generalized concept of equivalent uniform dose. *Medical Physics*, 26(6), 1100.

Rancati et al. (2004). Fitting late rectal bleeding data using different NTCP models: results from an Italian multi-centric study (AIROPROS0101). *Radiotherapy Oncology*, 73, 21-32.

See Also

[getTCP](#), [getEUD](#), [getMetric](#)

Examples

```
getNTCP(dataMZ[[1]],
        NTCPtd50=40, NTCPm=0.6, NTCPn=0.5, NTCPtype="probit")
```

getTCP

Tumor control probability (TCP)

Description

Calculate tumor control probability (TCP) from Lyman's probit model, Niemierko's logit model, or Kaellman's Poisson model. May be based on EQD2.

Usage

```
getTCP(x, TCPtd50=NULL, TCPm=NULL, TCPn=NULL, TCPgamma50=NULL,
       EUDa=NULL, EUDfn=NULL, EUDab=NULL,
       TCPtype=c("probit", "logit", "poisson"), ...)
```

Arguments

x	One cumulative DVH (object of class DVHs, multiple cumulative DVHs from one patient with multiple structures (object of class DVHlst), or multiple cumulative DVHs from many patients, each with multiple structures (object of class DVHlstLst). See readDVH .
TCPtcd50	Tolerance dose with 50% tumor control probability.
TCPm	Probit/logit Parameter m. Equal to $1 / (\text{NTCP}_{\text{gamma50}} \cdot \sqrt{2 \cdot \pi})$.
TCPn	Parameter n. Equal to $1/a$ with exponential gEUD parameter a.
TCPgamma50	Poisson parameter gamma50. Equal to $1 / (\text{NTCP}_m \cdot \sqrt{2 \cdot \pi})$
EUDa	If gEUD should be based on EQD2: Exponential parameter a.
EUDfn	If gEUD should be based on EQD2: Number of fractions.
EUDab	If gEUD should be based on EQD2: alpha/beta ratio for the relevant tissue.
TCPtype	"probit" - Lyman probit model, "logit" - Niemierko logit model, "poisson" - Kaellman Poisson (relative seriality) model.
...	Ignored. Used to catch additional arguments passed from getMetric .

Details

For DVH reduction, gEUD is used. This is equivalent to the Kutcher-Burman DVH reduction scheme.

Value

A data frame with variables TCP, patID, and structure.

References

- Kaellman, P., Agren, A., & Brahme, A. (1992). Tumor and normal tissue responses to fractionated non-uniform dose delivery. *International Journal of Radiation Biology*, 62(2), 249-262.
- Kutcher, G. J., Burman, C., Brewster L., Goitein, M., & Mohan, R. (1991). Histogram reduction method for calculating complication probabilities for threedimensional treatment planning evaluations. *International Journal of Radiation Oncology Biology Physics*, 21(1), 137-146.
- Lyman, J. T. (1985). Complication probability as assessed from dose volume histograms. *Radiation Research*, 104(2), S13-19.
- Niemierko, A. (1999). A generalized concept of equivalent uniform dose. *Medical Physics*, 26(6), 1100.
- Rancati et al. (2004). Fitting late rectal bleeding data using different NTCP models: results from an Italian multi-centric study (AIROPROS0101). *Radiotherapy Oncology*, 73, 21-32.

See Also

[getNTCP](#), [getEUD](#), [getMetric](#)

Examples

```
getTCP(dataMZ[[1]],
        TCPtcd50=40, TCPm=0.6, TCPn=0.5, TCPtype="probit")
```

mergeDVH	<i>Merge existing DVH objects</i>
----------	-----------------------------------

Description

Combine several existing DVH objects into one object.

Usage

```
mergeDVH(...)
```

Arguments

```
...          DVHlstLst objects.
```

Details

The first object determines whether the resulting object is organized by patient or by structure. Objects need not originally come from the same treatment planning system.

Value

Returns an object of class DVHlstLst.

Examples

```
## Not run:
# pick some DVH files interactively
a <- readDVH(type="Cadplan")

# pick other DVH files interactively
b <- readDVH(type="Eclipse")

# combine DVH data
res <- mergeDVH(a, b)
res

## End(Not run)
```

print.DVHs	<i>Print basic information about one or more DVHs</i>
------------	---

Description

Print basic information (patients, structures, dose range) about one or more DVHs.

Usage

```
## S3 method for class 'DVHs'  
print(x, ...)  
  
## S3 method for class 'DVHLst'  
print(x, ...)  
  
## S3 method for class 'DVHLstLst'  
print(x, ...)
```

Arguments

x	A single DVH (object of class DVHs), multiple DVHs from one patient/structure (object of class DVHLst), or multiple DVHs from many patients/structures (object of class DVHLstLst). See readDVH .
...	Further arguments: <code>print.DVHLst(x, verbose=TRUE)</code> prints more information about each DVH.

Value

Prints summary information about the DVHs.

See Also

[readDVH](#)

Examples

```
print(dataMZ)  
print(dataMZ, verbose=TRUE)
```

readConstraint	<i>Read constraint definitions from text file</i>
----------------	---

Description

Reads the definition of quality assurance constraints from a text file.

Usage

```
readConstraint(x, ...)
```

Arguments

x	character string giving the path to a single text file with the constraint definition. May contain globbing symbols understood by Sys.glob . If missing and in interactive mode, readDVH opens a file selector widget. See Details.
...	Further arguments passed to read.table , e.g., sep="\t" to define the column separator as tab.

Details

This is a wrapper for [read.table](#).

The text file should contain three columns with the column names patID, structure, constraint in the first line. Each further line then defines one constraint and the scope it applies to in terms of patients and structures. See [checkConstraint](#) for the definition of a constraint and for the definition of a scope. Example content:

```
"patID" "structure" "constraint"
"*"     "HEART"     "D1CC < 20Gy"
"234"   "*"        "V10% > 8CC"
```

Value

A data.frame with columns patID, structure, constraint that can be used in functions [checkConstraint](#) and [showConstraint](#).

See Also

[read.table](#), [checkConstraint](#), [saveConstraint](#), [showConstraint](#)

Examples

```
## Not run:
readConstraint("constraint.txt")
readConstraint()

## End(Not run)
```

readDVH	<i>Read DVH text files</i>
---------	----------------------------

Description

Reads single or multiple DVH text files as exported from Varian Eclipse(TM), CadPlan(TM), On-Centra MasterPlan(TM), Philipps Pinnacle3 (TM), Elekta Monaco (TM), Tomo HiArt (TM), or RaySearch Labs RayStation (TM). Supports cumulative and differential DVHs.

Usage

```
readDVH(x,
        type=c("Eclipse", "Cadplan", "Masterplan",
              "Pinnacle", "Monaco", "HiArt", "RayStation"),
        planInfo=FALSE, courseAsID=FALSE, add)
```

Arguments

x	character vector giving paths to DVH text files. May contain globbing symbols understood by Sys.glob . If missing and in interactive mode, readDVH opens a file selector widget. Under Windows, this widget allows selecting multiple files simultaneously. For type="Pinnacle", x should be one of the following: A directory with information for one patient, a directory with several sub-directories (one for each patient), or a zip file of such directories. Under Windows, if x is missing and type="Pinnacle", readDVH opens a folder selector widget.
type	character. Indicates which program the DVH text files were exported from. Supported: "Cadplan" (tested with version 6.4.7), "Eclipse" (tested with Varian Eclipse version 8), "Masterplan" (tested with OnCentra MasterPlan version 4.3), "Pinnacle" (tested with Pinnacle3 version 9, see Details), "Monaco" (Elekta Monaco), "HiArt" (TomoTherapy HiArt), or "RayStation" (RaySearch Labs RayStation).
planInfo	Experimental: Either FALSE or character string. In the latter case, readDVH tries to extract additional information from the Plan field in the DVH file, e.g., the prescription dose for a sum plan or the boost quadrant. Undocumented, see source.
courseAsID	logical. If TRUE, the Course entry in the header section of a DVH file is appended to the regular patient ID. Currently supported only for type="Eclipse".
add	DVHList object. Existing object that should be merged with the new data from the files.

Details

Absolute dose values need to be given in Gy or cGy, absolute volume values in cm³.

Differential DVHs are automatically converted to cumulative DVHs, but the differential DVH information is kept.

Sum plans are supported.

For RayStation, only cumulative DVHs with absolute volume are currently supported. Volume is assumed to be measured in cm^3 .

To export data from Tomo HiArt, copy to clipboard and then save to file from a text editor. Support for Tomo HiArt files is currently limited to those with absolute dose and relative volume. Please send an anonymized sample file if you need this functionality. Since HiArt files do not contain info on patient ID, the current workaround is to generate a random ID.

Pinnacle3 files have to be exported using its own scripting facility such that information from one patient is contained in one directory. A suitable export script is available on request from the package authors. The directory layout for one patient has to be as follows (experimental, likely to change in future versions):

- Files (CSV format with column headers):
 - DoseInfo.csv (variables "PrescriptionDose cGy", "NumberOfFractions", "Dosis cGy")
 - PatInfo.csv (variables "LastName", "FirstName", "MedicalRecordNumber")
 - PlanInfo.csv (variable "PlanName")
- Directory: Data:
 - Info.csv (variables "Filename", "RegionOfInterestName", "DoseMin cGy", "DoseMax cGy", "DoseMean cGy", "Volume ccm")
 - DVH1.csv, DVH2.csv, ... - the actual DVH data files with names defined in Info.csv variable "Filename". They should look like NumberOfDimensions = 2; NumberOfPoints = 431; Points[] = {


```
0,0
10,0
...
4000,100
};
```

Value

Returns an object of class DVHLstLst. This is a list (one component with class DVHLst for each original file from one patient) of lists (each component is an object of class DVHs). A DVHs object is a list with the following components:

dvh matrix - cumulative DVH values

dvhDiff matrix - differential DVH values, only created a) if original file contained a differential DVH or b) by [convertDVH](#)

patID character string - patient ID

date character string - date of DVH export

type character string - cumulative or differential DVH

plan character string - plan name

course character string - course - currently Eclipse only

structure character string - structure name

```
structVol numeric - structure volume
doseUnit character string - measurement unit dose
volumeUnit character string - measurement unit volume
doseRx numeric - prescription dose
isoDoseRx numeric - iso-dose percentage
doseMin numeric - minimum dose from DVH file
doseMax numeric - maximum dose from DVH file
doseAvg numeric - average dose from DVH file
doseMed numeric - median dose from DVH file
doseSD numeric - dose standard deviation from DVH file
```

See Also

[Sys.glob](#), [print.DVHs](#), [showDVH](#), [getMetric](#), [checkConstraint](#), [convertDVH](#)

Examples

```
## Not run:
# pick DVH files interactively
res <- readDVH()
res

# read all txt files in subdirectory DVH
res <- readDVH("DVH/*.txt", type="Eclipse")
res

## End(Not run)
```

runGUI

Open web-based GUI in browser

Description

Opens the web-based GUI in an external browser.

Usage

```
runGUI(...)
```

Arguments

... Arguments passed to [runApp](#). Supply `port=80` if a web browser refuses to connect to the randomly chosen port for security reasons.

Details

This function calls `runApp` to run the included DVHshiny application. See `vignette("DVHshiny")` for documentation.

See Also

[runApp](#)

Examples

```
## Not run:  
runGUI()  
  
## End(Not run)
```

saveConstraint	<i>Save constraint result to file</i>
----------------	---------------------------------------

Description

Saves results from `checkConstraint` to a text file.

Usage

```
saveConstraint(x, ...)
```

Arguments

x	data.frame - the result from <code>checkConstraint</code> .
...	Further arguments passed to <code>write.table</code> - e.g., <code>file="<filename>"</code> for the output filename, <code>dec="."</code> to define the decimal separator as point or <code>sep="\t"</code> to define the column separator as tab.

Details

This is a wrapper for `write.table`.

See Also

[write.table](#), [checkConstraint](#)

Examples

```
res <- checkConstraint(dataMZ, c("D10CC < 10Gy", "V20Gy < 20%"))  
## Not run:  
saveConstraint(res, file="constrResults.txt", sep="\t")  
  
## End(Not run)
```

saveDVH	<i>Save DVH diagram to file</i>
---------	---------------------------------

Description

Saves one or multiple DVH diagrams to file.

Usage

```
saveDVH(x, file="", ...)
```

Arguments

x	A single ggplot object or a list of multiple ggplot objects as returned by showDVH or showConstraint .
file	character. Path to file. The file-ending determines what kind of file is written, e.g., "filename.pdf" will write a pdf document, "filename.jpg" a JPEG image.
...	Further arguments passed to ggsave , e.g., width and height to determine the figure size.

Details

This is a wrapper for [ggsave](#).

Value

If x is a list of [ggplot](#) objects, one file is written for each list component. If x is a single [ggplot](#) object, one file is written.

See Also

[ggsave](#), [showDVH](#), [showConstraint](#)

Examples

```
res <- showDVH(dataMZ, byPat=TRUE, structure=c("HEART", "AMYOCL"))
## Not run:
saveDVH(res, "out.pdf")

## End(Not run)
```

saveMetric	<i>Save DVH metrics to file</i>
------------	---------------------------------

Description

Saves results from [getMetric](#) to a text file.

Usage

```
saveMetric(x, file = "", ...)  
  
## S3 method for class 'data.frame'  
saveMetric(x, file = "", ...)  
  
## S3 method for class 'list'  
saveMetric(x, file = "", ...)
```

Arguments

x	data.frame or list - the result from getMetric .
file	character. Path to file.
...	Further arguments passed to write.table - e.g., dec="." to define the decimal separator as point or sep="\t" to define the column separator as tab.

Details

This is a wrapper for [write.table](#).

Value

If x is a list, one text file is written for each list component. If x is a data.frame, one file is written.

See Also

[write.table](#), [getMetric](#)

Examples

```
res <- getMetric(dataMZ, c("D1CC", "V10%_CC"),  
                 sortBy=c("metric", "structure"),  
                 splitBy="patID")  
  
## Not run:  
# not run  
saveMetric(res, file="metricsResults.txt", sep="\t")  
  
## End(Not run)
```

showConstraint	<i>Display constraints for cumulative dose-volume histograms</i>
----------------	--

Description

Displays quality assurance constraints for cumulative dose-volume histograms: Either one diagram per patient - including multiple structures. Or one diagram per structure - including multiple patients.

Usage

```
showConstraint(x, constr, byPat=TRUE, rel=TRUE, guessX=TRUE, thresh=1)
```

```
## S3 method for class 'DVHs'
showConstraint(x, constr, byPat=TRUE, rel=TRUE, guessX=TRUE, thresh=1)
```

```
## S3 method for class 'DVHLst'
showConstraint(x, constr, byPat=TRUE, rel=TRUE, guessX=TRUE, thresh=1)
```

```
## S3 method for class 'DVHLstLst'
showConstraint(x, constr, byPat=TRUE, rel=TRUE, guessX=TRUE, thresh=1)
```

Arguments

x	A single DVH (object of class DVHs), multiple DVHs from one patient/structure (object of class DVHLst), or multiple DVHs from many patients/structures (object of class DVHLstLst). See readDVH . See Details.
constr	One or more constraints - given as a character vector or as a data.frame. See checkConstraint for their definition.
byPat	logical. Relevant if multiple DVHs are given. If x has class DVHLstLst: byPat=TRUE means that one diagram shows DVHs from one patient with multiple structures. byPat=FALSE means that one diagram shows DVHs for one structure from multiple patients.
rel	logical. Show relative volume?
guessX	logical. Try to clip the x-axis for better visibility of main DVH range?
thresh	numeric value. Relative volume threshold used with guessX=TRUE. Clip x-axis (+10%) such that the "highest" DVH is cut off at this relative volume.

Details

Constraints are shown as points in the cumulative DVH with an additional arrow indicating where the cumulative DVH curve should lie relative to the constraint. On each DVH curve, the point with the minimal Euclidean distance to the constraint is indicated. Note that, visually, this point only has the minimal apparent distance if the aspect ratio of the diagram is 1.

If multiple diagrams are produced, they are shown in the same graphics device. If interactive inspection is required, make sure you use an R development environment that saves previous diagrams and allows navigating between them - e.g., [RStudio](#) or [OpenAnalytics Architect](#).

Value

Silently returns a `ggplot` diagram object, or - when multiple diagrams are constructed - a list of `ggplot` diagram objects.

See Also

`checkConstraint`, `saveDVH`

Examples

```
data(dataMZ)

# define constraints
constr <- data.frame(
  patID=c("P123", "P234"),
  structure=c("HEART", "*"),
  constraint=c("D1CC < 20Gy", "V10% > 8CC"),
  stringsAsFactors=FALSE) # this is important
showConstraint(dataMZ, constr=constr, byPat=FALSE)
```

showDVH

Display dose volume histograms

Description

Displays dose volume histograms: Either one diagram per patient - including multiple structures. Or one diagram per structure - including multiple patients.

Usage

```
showDVH(x, cumul=TRUE, byPat=TRUE, patID=NULL, structure=NULL,
        rel=TRUE, guessX=TRUE, thresh=1, addMSD=FALSE, show=TRUE, fixed=TRUE)

## S3 method for class 'DVHs'
showDVH(x, cumul=TRUE, byPat=TRUE, patID=NULL, structure=NULL,
        rel=TRUE, guessX=TRUE, thresh=1, addMSD=FALSE, show=TRUE, fixed=TRUE)

## S3 method for class 'DVHlst'
showDVH(x, cumul=TRUE, byPat=TRUE, patID=NULL, structure=NULL,
        rel=TRUE, guessX=TRUE, thresh=1, addMSD=FALSE, show=TRUE, fixed=TRUE)

## S3 method for class 'DVHlstLst'
showDVH(x, cumul=TRUE, byPat=TRUE, patID=NULL, structure=NULL,
        rel=TRUE, guessX=TRUE, thresh=1, addMSD=FALSE, show=TRUE, fixed=TRUE)
```

Arguments

x	A single DVH (object of class DVHs), multiple DVHs from one patient/structure (object of class DVHLst), or multiple DVHs from many patients/structures (object of class DVHLstLst). See readDVH . See Details.
cumul	logical. Show cumulative or differential (per unit dose) DVH?
byPat	logical. Relevant if multiple DVHs are given. If x has class DVHLstLst: byPat=TRUE means that one diagram shows DVHs from one patient with multiple structures. byPat=FALSE means that one diagram shows DVHs for one structure from multiple patients.
patID	character vector. Show diagram for these patients only. If missing, all patients are shown. Can be a regular expression with fixed=FALSE, see regex .
structure	character vector. Show diagram for these structures only. If missing, all structures are shown. Can be a regular expression with fixed=FALSE, see regex .
rel	logical. Show relative volume?
guessX	logical. Try to clip the x-axis for better visibility of main DVH range?
thresh	numeric value. Relative volume threshold used with guessX=TRUE. Clip x-axis (+10%) such that the "highest" DVH is cut off at this relative volume.
addMSD	logical. If TRUE, diagram shows the point-wise mean DVH as well as shaded areas for point-wise 1-standard deviation and 2-standard deviations around this mean. See details.
show	logical. If TRUE, diagrams are shown, if FALSE diagrams are not shown - only ggplot diagram objects are silently returned.
fixed	logical. Use fixed=FALSE for regular expression matching of patID and structure.

Details

If multiple diagrams are produced, they are shown in the same graphics device. If interactive inspection is required, make sure you use an R development environment that saves previous diagrams and allows navigating between them - e.g., [RStudio](#) or [OpenAnalytics Architect](#).

For addMSD=TRUE, the number of DVH nodes (dose values) is reduced by 1/3 of the maximum number of nodes in x. Before calculating the point-wise mean andSD, DVHs in x are first linearly interpolated using the same set of nodes.

Value

Silently returns a [ggplot](#) diagram object, or - when multiple diagrams are constructed - a list of [ggplot](#) diagram objects.

See Also

[ggplot](#), [readDVH](#), [saveDVH](#), [getMeanDVH](#)

Examples

```
showDVH(dataMZ, byPat=TRUE, structure=c("HEART", "AMYOCL"))
```

```
# matches patients P123 and P234
```

```
showDVH(dataMZ, byPat=FALSE, patID="23", fixed=FALSE)
```

Index

- *Topic **datasets**
 - dataConstr, 7
 - dataMZ, 8
- *Topic **package**
 - DVHmetrics-package, 2

- approx, 4, 11, 18
- approxfun, 11, 20

- checkConstraint, 3, 8, 26, 29, 30, 33, 34
- convertDVH, 6, 17, 28, 29

- dataConstr, 7
- dataMZ, 7, 8
- dpill, 4, 7, 11, 18, 20
- DVHmetrics (DVHmetrics-package), 2
- DVHmetrics-package, 2

- getBED, 9, 13, 15
- getDMEAN, 10
- getEQD2, 10, 12, 14, 15, 20
- getEUD, 4, 5, 13, 19, 20, 22, 23
- getIsoEffD, 10, 12, 13, 14
- getMeanDVH, 16, 35
- getMetric, 4, 5, 11, 13, 14, 17, 21–23, 29, 32
- getNTCP, 4, 5, 19, 20, 21, 23
- getTCP, 4, 5, 19, 20, 22, 22
- ggplot, 31, 34, 35
- ggsave, 31

- locpoly, 4, 7, 11, 18, 20

- mergeDVH, 24

- print.DVHList (print.DVHs), 25
- print.DVHListList (print.DVHs), 25
- print.DVHs, 9, 25, 29

- read.table, 26
- readConstraint, 5, 8, 26

- readDVH, 4, 7, 9, 11–13, 15, 16, 18, 21, 23, 25, 27, 33, 35
- regex, 16–18, 35
- runApp, 29, 30
- runGUI, 29

- saveConstraint, 5, 26, 30
- saveDVH, 31, 34, 35
- saveMetric, 20, 32
- showConstraint, 5, 8, 26, 31, 33
- showDVH, 7, 17, 29, 31, 34
- smooth.spline, 7, 11, 17
- spline, 4
- splinefun, 7, 11, 17, 18, 20
- Sys.glob, 26, 27, 29

- write.table, 30, 32