Analyzing dose-volume histograms using `DVHmetrics` for R

Daniel Wollschlaeger
wollschlaeger@uni-mainz.de

Heiko Karle
karle@uni-mainz.de

University Medical Center Mainz, Germany
July 31, 2025

Contents

1 Introduction **2**

2 Interfaces **2**
2.1 R command line interface . . . 2
2.2 Web-based graphical user interface . . . 2

3 Read DVH text data **3**

4 DVH metrics **5**
4.1 Calculate DVH metrics . . . 5
4.2 Save DVH metrics to file . . . 9
4.3 Convenience functions for DMEAN, gEUD, NTCP, TCP . . . 10
4.4 Point-wise mean DVH with standard deviations . . . 11

5 DVH diagrams **12**
5.1 Plot DVH diagrams . . . 12
5.2 Save cumulative DVH diagrams to file . . . 16

6 Quality assurance constraints on the dose-volume relationship **16**
6.1 Define constraints . . . 16
6.2 Check constraints . . . 18
6.3 Visualize constraints . . . 19

7 BED, EQD2, Isoeffective Dose **20**

References **22**

1 Introduction

`DVHmetrics` is an add-on package for the free statistical environment R[1] (R Development Core Team, 2025) with applications in radiation oncology. It provides functionality to read dose-volume-histogram (DVH) text files, to calculate DVH metrics, and to plot DVHs. In addition, it checks and visualizes quality assurance constraints for the DVH.[2]

To install `DVHmetrics`, you need a current version of R and be online. Preferably, a free development environment like RStudio (Posit Inc, 2023) should be used.

```
# install DVHmetrics from the CRAN online package repository
install.packages("DVHmetrics")
```

2 Interfaces

`DVHmetrics` provides two interfaces geared towards users with different levels of familiarity with R: The regular command line functions and a built-in web application.

2.1 R command line interface

Users familiar with R can use the `DVHmetrics` package functions from the R command line. This facilitates statistical post-processing of results with the full capabilities of R. After installing `DVHmetrics`, you should be able to run (function `getMetric()` is explained in section 4.1):

```
## load DVHmetrics package - required for all following tasks
library(DVHmetrics, verbose=FALSE)

## calculate a DVH metric for built-in data
getMetric(dataMZ, metric="DMEAN", structure="HEART")
```

```
##   observed metric structure patID
## 1   18.110  DMEAN     HEART  P123
## 2    0.995  DMEAN     HEART  P234
## 3   10.389  DMEAN     HEART  P345
```

2.2 Web-based graphical user interface

For users who are unfamiliar with R, `DVHmetrics` includes a `shiny`-based web application (Chang, Cheng, Allaire, Xie, & McPherson, 2024) running locally that eliminates the need to use R syntax. For information on how to use this app, see the documentation by running this from the command line:

[1] A free short introduction to R can be found at `https://www.statmethods.net/`.
[2] For solutions that also read files in DICOM format, see R packages `RadOnc` (Thompson, 2025) and `espadon` and ().

```
vignette("DVHshiny")
```

3 Read DVH text data

To import DVH data into R, it should be exported as a plain text file from Varian Eclipse™ (Versions 10–15), CadPlan™, Pinnacle³ ™ (version 9[3]), Oncentra MasterPlan™ (version 4.3), Elekta Monaco™ (version 5), TomoTheray HiArt™, RaySearch Labs RayStation™, Medcom ProSoma™, PRIMO (version 0.3.1.1558), or Mirada. DVH files from different TPSs can be combined into one set of DVHs. Cumulative and differential DVHs are supported, as are sum plans. Eclipse uncertainty plans are supported. The measurement unit for absolute dose can be Gy, cGy, or eV/G for uncalibrated PRIMO files. The measurement unit for volume has to be cm^3. Multiple DVH text files can be read with `readDVH()` in one step.

Example: Read one Eclipse file `dvhFile.txt` from folder `"c:/folder"` and save the result in object `res`.[4]

```
res <- readDVH("c:/folder/dvhFile.txt", type="Eclipse")
```

Basic information about the files can be displayed with `print()`, or just by entering the name of a DVH object at the prompt – here used with built-in DVHs from three patients with radiotherapy, each with seven heart structures.[5]

```
print(dataMZ)


## DVH list of 3 lists - 1 for each patient:
##
## DVH list:
## Patient 'John Doe' (ID P123, prescription dose 60GY) with 7 Structures:
## HEART, AOVALVE, AMYOCR, PULMVALVE, MYOCARD, AMYOCL, AVNODE
##
## DVH list:
## Patient 'Jane Doe' (ID P234, prescription dose 60GY) with 7 Structures:
## AMYOCR, AVNODE, HEART, AOVALVE, PULMVALVE, MYOCARD, AMYOCL
##
## DVH list:
## Patient 'Jane Smith' (ID P345, prescription dose 60GY) with 7 Structures:
## HEART, AOVALVE, PULMVALVE, MYOCARD, AMYOCL, AMYOCR, AVNODE
```

Display more information on structures with `verbose=TRUE`.

[3] Pinnacle³ files have to be exported such that information from one patient is contained in one directory. The directory layout and required files are explained in `help(readDVH)`.

[4] Note that the way to indicate the path to these files is different from the usual Windows style path: Instead of writing the backslash `"\"` as folder separator, the forward slash `"/"` must be used.

[5] Sample data courtesy of Department of Radiation Oncology (Prof. Dr. Schmidberger), University Medical Center Mainz, Germany.

```
print(dataMZ, verbose=TRUE)


## DVH list of 3 lists - 1 for each patient:
##
## DVH list:
## Patient 'John Doe' (ID P123, prescription dose 60GY) with 7 Structures:
## DVH: Structure 'HEART' (600 CC), Dose: 0-62GY
## DVH: Structure 'AOVALVE' (12 CC), Dose: 0-62GY
## DVH: Structure 'AMYOCR' (57 CC), Dose: 0-62GY
## DVH: Structure 'PULMVALVE' (3.8 CC), Dose: 0-62GY
## DVH: Structure 'MYOCARD' (310 CC), Dose: 0-62GY
## DVH: Structure 'AMYOCL' (60 CC), Dose: 0-62GY
## DVH: Structure 'AVNODE' (38 CC), Dose: 0-62GY
##
## DVH list:
## Patient 'Jane Doe' (ID P234, prescription dose 60GY) with 7 Structures:
## DVH: Structure 'AMYOCR' (53 CC), Dose: 0-63GY
## DVH: Structure 'AVNODE' (43 CC), Dose: 0-63GY
## DVH: Structure 'HEART' (730 CC), Dose: 0-63GY
## DVH: Structure 'AOVALVE' (15 CC), Dose: 0-63GY
## DVH: Structure 'PULMVALVE' (6.7 CC), Dose: 0-63GY
## DVH: Structure 'MYOCARD' (330 CC), Dose: 0-63GY
## DVH: Structure 'AMYOCL' (81 CC), Dose: 0-63GY
##
## DVH list:
## Patient 'Jane Smith' (ID P345, prescription dose 60GY) with 7 Structures:
## DVH: Structure 'HEART' (550 CC), Dose: 0-62GY
## DVH: Structure 'AOVALVE' (11 CC), Dose: 0-62GY
## DVH: Structure 'PULMVALVE' (5.7 CC), Dose: 0-62GY
## DVH: Structure 'MYOCARD' (270 CC), Dose: 0-62GY
## DVH: Structure 'AMYOCL' (43 CC), Dose: 0-62GY
## DVH: Structure 'AMYOCR' (52 CC), Dose: 0-62GY
## DVH: Structure 'AVNODE' (24 CC), Dose: 0-62GY
```

Multiple files with the same name pattern can be specified using wildcards like `*`. Example: Read all CadPlan files with the file name pattern `dvhFile*.txt` from folder `"c:/folder"` and save the result in object `res`.

```
res <- readDVH("c:/folder/dvhFile*.txt", type="Cadplan")
```

When no file pattern is specified, multiple files can be selected using the standard Windows file picker dialogue. On MacOS and Linux, only a single file can be selected interactively.

```
res <- readDVH(type="Eclipse")      # opens interactive file picker
```

For DVH files from a sum plan, prescribed dose can be encoded in the plan name like `name_70Gy_etc`. It will then be assumed that "% for dose" is 100.

```
res <- readDVH("c:/folder/*", type="Eclipse", planInfo="doseRx")
```

If files contain special characters, it may be necessary to specify the file encoding using options `encoding="UTF-8"` or `encoding="UTF-8-BOM"` (when a byte-order-mark is used). If Eclipse uncertainty plans are present, option `uncertainty=TRUE` is necessary. Structures with uncertainty plans get separate structure names by appending the actual structure name with a suffix derived from the uncertainty plan name.

4 DVH metrics

4.1 Calculate DVH metrics

Function `getMetric()` calculates freely-defined DVH metrics based on data that has been read in as demonstrated in section 3. `getMetric()` has the following options:

- Option `x`: The DVH data.
- Option `metric` – one or many of the following:
  - A *pre-specified* DVH metric is one of the following character strings:
    - `"DMEAN"`: The volume-weighted mean dose of the structure.
    - `"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.
    - `"DHI"`: The Homogeneity Index according to ICRU 83: (D2%-D98%)/D50%.
    - `"DEUD"`: The generalized equivalent uniform dose (gEUD, Niemierko, 1999; Wu, Mohan, Niemierko, & Schmidt-Ullrich, 2002). This can be based on EQD_2 values if information on the fractionation is provided as well (IAEA & ICRU, 2008).
    - `"DNTCP"`: The normal tissue complication probability (NTCP) according to the Lyman (1985) probit model, the Niemierko (1999) logit model, the Poisson model (equations (1) to (3) in Källman, Ågren, & Brahme, 1992 with gEUD plugged in for D), or the relative seriality model (equation (18)). This can be based on EQD_2 values if information on the fractionation is provided as well.
    - `"DTCP"`: The tumor control probability (TCP) according to the same models as NTCP.
  - 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 cm^3, 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 the same as for the 3rd element. If missing, dose is reported as absolute dose in the measurement unit used in the DVH. If the measurement unit is missing, volume is reported as relative volume in %.
  - Example metrics are listed in table 1. By default, metrics are calculated using linear interpolation between adjacent supporting points of the cumulative DVH – without extrapolating beyond the observed volume or dose. The interpolation method can be changed to use monotone Hermite splines (Fritsch & Carlson, 1980), or to local polynomial regression with a Gaussian kernel (Wand & Jones, 1995). The kernel bandwidth is then determined by the direct plug-in method (Ruppert, Sheather, & Wand, 1995).
- Option `patID`: Which patient IDs should be analyzed. With `fixed=FALSE`, IDs are interpreted as regular expressions matched against those found in the DVH files. By default, IDs are matched exactly. If missing, the metrics are calculated for all patients.
- Option `structure`: Which structure should be analyzed. With `fixed=FALSE`, structure names are interpreted as regular expressions matched against those found in the DVH files. By default, structure names are matched exactly. If missing, the metrics are calculated for all structures.
- Option `sortBy`: Results can be sorted according to these variables:
  - `"observed"`: observed value of the metric
  - `"structure"`: structure for which the metric is calculated
  - `"metric"`: type of calculated metric
  - `"patID"`: patient ID
- Option `splitBy`: Results can be divided into different tables according to these variables:
  - `"structure"`: structure for which the metric is calculated
  - `"metric"`: type of calculated metric
  - `"patID"`: patient ID

If volume or dose values outside the range of possible values for a structure are requested, it may be that metrics cannot be calculated, and the result will be `NA` (missing value) with a warning.

In the following examples, we use object `dataMZ` that is built into the `DVHmetrics` package. `dataMZ` was the result from reading three Eclipse DVH files, each with seven structures – as demonstrated in section 3.

Calculate metric `DMEAN` for all structures for all patients in `dataMZ`.

```
getMetric(dataMZ, metric="DMEAN")
```

Table 1: Examples of possible DVH metrics

Metric	Reference	Unit reference	Result	Unit result
`"V10Gy"`	absolute dose	Gy	relative volume	%
`"V10cGy_CC"`	absolute dose	cGy	absolute volume	cm^3
`"V10%"`	relative dose	%	relative volume	%
`"V10%_CC"`	relative dose	%	absolute volume	cm^3
`"D10CC"`	absolute volume	cm^3	absolute dose	as in DVH
`"D10%_cGy"`	relative volume	%	absolute dose	cGy
`"DMEAN"`	—	—	absolute dose	as in DVH
`"DEUD"`	—	—	absolute dose	as in DVH
`"DSD"`	—	—	absolute dose	as in DVH
`"DMIN"`	—	—	absolute dose	as in DVH
`"DMAX"`	—	—	absolute dose	as in DVH
`"DHI"`	—	—	dose ratio	—
`"DNTCP"`	—	—	probability	—
`"DTCP"`	—	—	probability	—

```
##    observed metric structure patID
## 1    18.110  DMEAN     HEART  P123
## 2    21.454  DMEAN   AOVALVE  P123
## 3    24.046  DMEAN    AMYOCR  P123
## 4    23.015  DMEAN PULMVALVE  P123
## 5    17.492  DMEAN   MYOCARD  P123
## 6    18.496  DMEAN    AMYOCL  P123
## 7    19.200  DMEAN    AVNODE  P123
## 8     2.421  DMEAN    AMYOCR  P234
## 9     0.968  DMEAN    AVNODE  P234
## 10    0.995  DMEAN     HEART  P234
## 11    1.959  DMEAN   AOVALVE  P234
## 12    1.198  DMEAN PULMVALVE  P234
## 13    0.990  DMEAN   MYOCARD  P234
## 14    0.775  DMEAN    AMYOCL  P234
## 15   10.389  DMEAN     HEART  P345
## 16   11.283  DMEAN   AOVALVE  P345
## 17   10.019  DMEAN PULMVALVE  P345
## 18   10.639  DMEAN   MYOCARD  P345
## 19    2.549  DMEAN    AMYOCL  P345
## 20   30.009  DMEAN    AMYOCR  P345
## 21    6.676  DMEAN    AVNODE  P345
```

Calculate metric `D5cc` just for structure `HEART` for all patients in `dataMZ`.

```
getMetric(dataMZ, metric="D5cc", structure="HEART")
```

```
##   observed metric structure patID
## 1   31.482   D5CC     HEART  P123
## 2    3.474   D5CC     HEART  P234
## 3   41.618   D5CC     HEART  P345
```

Calculate metric `D5cc` just for structure `HEART` for all patients in `dataMZ`, and sort result by the observed value of the metric.

```
getMetric(dataMZ, metric="D5cc", structure="HEART", sortBy="observed")
```

```
##   observed metric structure patID
## 2    3.474   D5CC     HEART  P234
## 1   31.482   D5CC     HEART  P123
## 3   41.618   D5CC     HEART  P345
```

Calculate metrics `D10%` and `V5Gy` for all structures containing the text `AMYOC` or `VALVE`, for patient IDs in `dataMZ` containing the text `23`, and sort result by metric and observed value.

```
getMetric(dataMZ, metric=c("D10%", "V5Gy"),
          structure=c("AMYOC", "VALVE"),
          patID="23",
          sortBy=c("metric", "observed"),
          fixed=FALSE)
```

```
##    observed metric structure patID
## 15    1.312   D10%    AMYOCL  P234
## 13    1.404   D10% PULMVALVE  P234
## 11    2.556   D10%   AOVALVE  P234
## 9     3.263   D10%    AMYOCR  P234
## 7    23.208   D10%    AMYOCL  P123
## 5    24.391   D10% PULMVALVE  P123
## 1    25.056   D10%   AOVALVE  P123
## 3    30.710   D10%    AMYOCR  P123
## 10    0.000   V5GY    AMYOCR  P234
## 12    0.000   V5GY   AOVALVE  P234
## 14    0.000   V5GY PULMVALVE  P234
## 16    0.000   V5GY    AMYOCL  P234
## 2   100.000   V5GY   AOVALVE  P123
## 4   100.000   V5GY    AMYOCR  P123
## 6   100.000   V5GY PULMVALVE  P123
## 8   100.000   V5GY    AMYOCL  P123
```

Calculate metrics `DMEAN` and `D5cc` for structure `HEART` for all patients in `dataMZ`, sort by the observed value of the metric, and split the output such that one table is generated for each metric.

```
getMetric(dataMZ, metric=c("DMEAN", "D5cc"), structure="HEART",
          sortBy="observed", splitBy="metric")
```

```
## $D5CC
##   observed metric structure patID
## 4    3.474   D5CC     HEART  P234
## 2   31.482   D5CC     HEART  P123
## 6   41.618   D5CC     HEART  P345
```

```
##
## $DMEAN
##   observed metric structure patID
## 3    0.995  DMEAN     HEART  P234
## 5   10.389  DMEAN     HEART  P345
## 1   18.110  DMEAN     HEART  P123
```

Calculate metrics `DMEAN` and `D5cc` for structures `HEART` and `AOVALVE` for all patients in `dataMZ`, sort by observed value, and split the output such that one table is generated for each combination of structure and metric. Also store the result in object `met` that can be saved later.

```
met <- getMetric(dataMZ, metric=c("DMEAN", "D5cc"),
                 structure=c("HEART", "AOVALVE"),
                 sortBy="observed",
                 splitBy=c("structure", "metric"))
met                              # print the calculated results


## $AOVALVE.D5CC
##    observed metric structure patID
## 8     2.064   D5CC   AOVALVE  P234
## 12    8.337   D5CC   AOVALVE  P345
## 4    21.752   D5CC   AOVALVE  P123
##
## $HEART.D5CC
##    observed metric structure patID
## 6     3.474   D5CC     HEART  P234
## 2    31.482   D5CC     HEART  P123
## 10   41.618   D5CC     HEART  P345
##
## $AOVALVE.DMEAN
##    observed metric structure patID
## 7     1.959  DMEAN   AOVALVE  P234
## 11   11.283  DMEAN   AOVALVE  P345
## 3    21.454  DMEAN   AOVALVE  P123
##
## $HEART.DMEAN
##   observed metric structure patID
## 5    0.995  DMEAN     HEART  P234
## 9   10.389  DMEAN     HEART  P345
## 1   18.110  DMEAN     HEART  P123
```

4.2 Save DVH metrics to file

The calculated DVH metrics can be saved to tab-delimited text files with `saveMetric()`. These files are easy to import, e. g., into spreadsheets like Excel or into other statistics programs.

Assume object `met` has been calculated before as demonstrated in section 4.1. If `met` is not split into different tables, the following command saves `met` to the file `metrics.txt`. If `met` is divided into

multiple tables, this saves `met` into different files that all have the name pattern `metrics_NAME.txt`, where `NAME` stands, e. g., for the names of different structures.

```
saveMetric(met, file="c:/folder/metrics.txt")
```

Per default, numbers use the `.` as decimal separator. This can be changed with option `dec=","` .

```
saveMetric(met, file="c:/folder/metrics.txt", dec=",")
```

If text should be set in quotes in the output file, use `quote=TRUE` .

```
saveMetric(met, file="c:/folder/metrics.txt", quote=TRUE)
```

4.3 Convenience functions for DMEAN, gEUD, NTCP, TCP

DMEAN, gEUD, NTCP and TCP may be calculated together with other metrics using `getMetric()`, but there are specialized convenience functions for this task as well. In particular, `getDMEAN()` calculates the dose mean, median, mode, minimum, and maximum based on the (interpolated) differential DVH instead of relying on the values exported by the TPS.

```
dmean <- getDMEAN(dataMZ[[1]])
subset(dmean, select=c(doseAvg, doseMed, doseMin, doseMax))

##   doseAvg doseMed doseMin doseMax
## 1   18.11   17.70   3.499   35.20
## 2   21.45   21.08  16.205   30.69
## 3   24.05   24.96   7.398   35.20
## 4   23.01   23.00  20.400   25.70
## 5   17.49   16.19   3.905   35.20
## 6   18.50   18.21  12.995   26.89
## 7   19.20   19.11  12.503   29.80
```

gEUD is calculated by `getEUD()`.

```
# note that different tissues should have different parameter values,
# this is just for demonstration purposes
getEUD(dataMZ[[1]], EUDa=2)

##     EUD patID structure
## 1 18.77  P123     HEART
## 2 21.61  P123   AOVALVE
## 3 24.63  P123    AMYOCR
## 4 23.04  P123 PULMVALVE
## 5 18.33  P123   MYOCARD
## 6 18.77  P123    AMYOCL
## 7 19.35  P123    AVNODE
```

NTCP and TCP are calculated by `getNTCP()` and `getTCP()`, respectively.

```
# note that different tissues should have different parameter values,
# this is just for demonstration purposes
getNTCP(dataMZ[[1]], NTCPtd50=40, NTCPm=0.6, NTCPn=0.5, NTCPtype="probit")
```

```
##     NTCP patID structure
## 1 0.1882  P123     HEART
## 2 0.2217  P123   AOVALVE
## 3 0.2609  P123    AMYOCR
## 4 0.2398  P123 PULMVALVE
## 5 0.1832  P123   MYOCARD
## 6 0.1882  P123    AMYOCL
## 7 0.1948  P123    AVNODE
```

4.4 Point-wise mean DVH with standard deviations

Function `getMeanDVH()` 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. Before calculating the point-wise mean and SD, DVHs are first linearly interpolated such that they possess the same set of nodes. This feature can be useful for evaluating different plan options: The DVHs for each plan need to be exported using a different patient ID which thus serves as a plan identifier. See section 5.1 to show the mean DVH with SD regions.

```
# point-wise mean and SD for structure HEART over all patients
m1 <- getMeanDVH(dataMZ, fun=list(M=mean, SD=sd), byPat=FALSE, structure="HEART")
head(m1)
```

```
##   structure dose  doseRel volumeM volumeRelM volumeSD volumeRelSD
## 1     HEART  0.0 0.000000   623.0     100.00    94.25       0.000
## 2     HEART  0.1 0.001585   613.6      98.71    78.67       2.237
## 3     HEART  0.2 0.003170   596.4      96.36    51.42       6.312
## 4     HEART  0.3 0.004754   591.2      95.63    43.68       7.563
## 5     HEART  0.4 0.006339   579.6      94.04    29.61      10.309
## 6     HEART  0.5 0.007924   557.3      90.98    33.83      15.597
##            patID
## 1 P123_P234_P345
## 2 P123_P234_P345
## 3 P123_P234_P345
## 4 P123_P234_P345
## 5 P123_P234_P345
## 6 P123_P234_P345
```

When using option `returnDVHObj=TRUE`, the function returns a DVH object that behaves like a regular DVH, and can be used in functions such as `showDVH()` or `getMetric()`

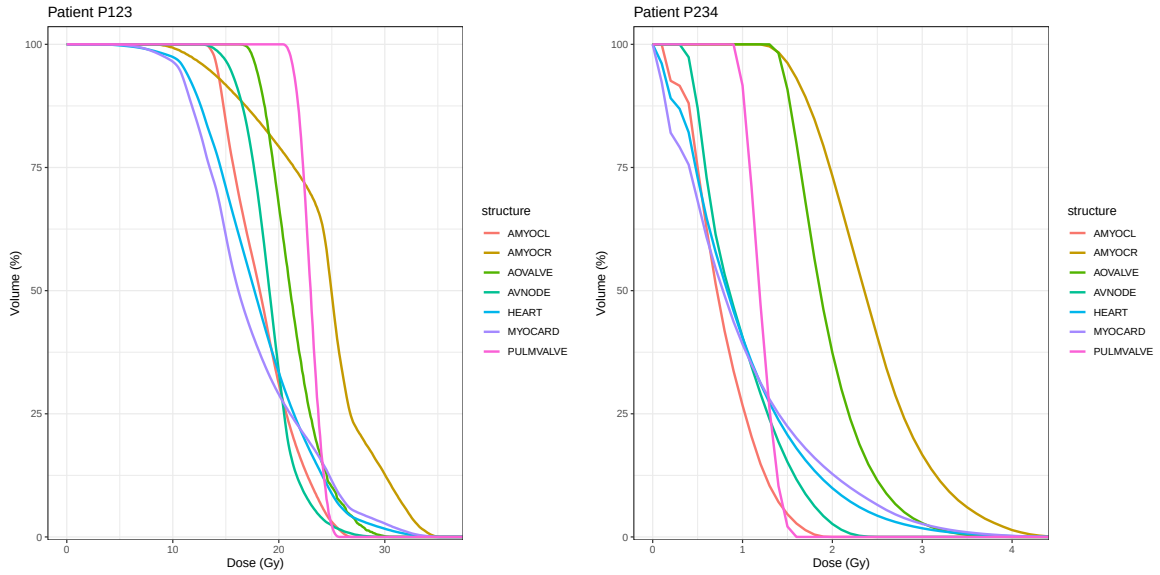
```
# point-wise mean for structure HEART over all patients
m2 <- getMeanDVH(dataMZ, fun=list(mean), byPat=FALSE, structure="HEART",
                 returnDVHObj=TRUE)

getMetric(m2, metric="V5GY")


##   observed metric structure
## 1    46.78   V5GY     HEART
```

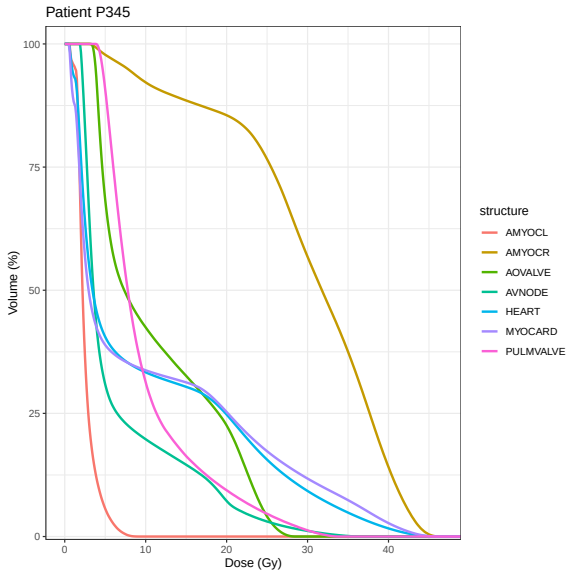
5 DVH diagrams

5.1 Plot DVH diagrams

Cumulative as well as differential DVH diagrams can be generated with `showDVH()`. If you are using RStudio or Architect, all produced diagrams are accessible in the plots tab by clicking on the left and right arrows. Depending on the option `byPat`, each DVH diagram either shows one patient with multiple structures (`byPat=TRUE`) or one structure with multiple patients (`byPat=FALSE`).

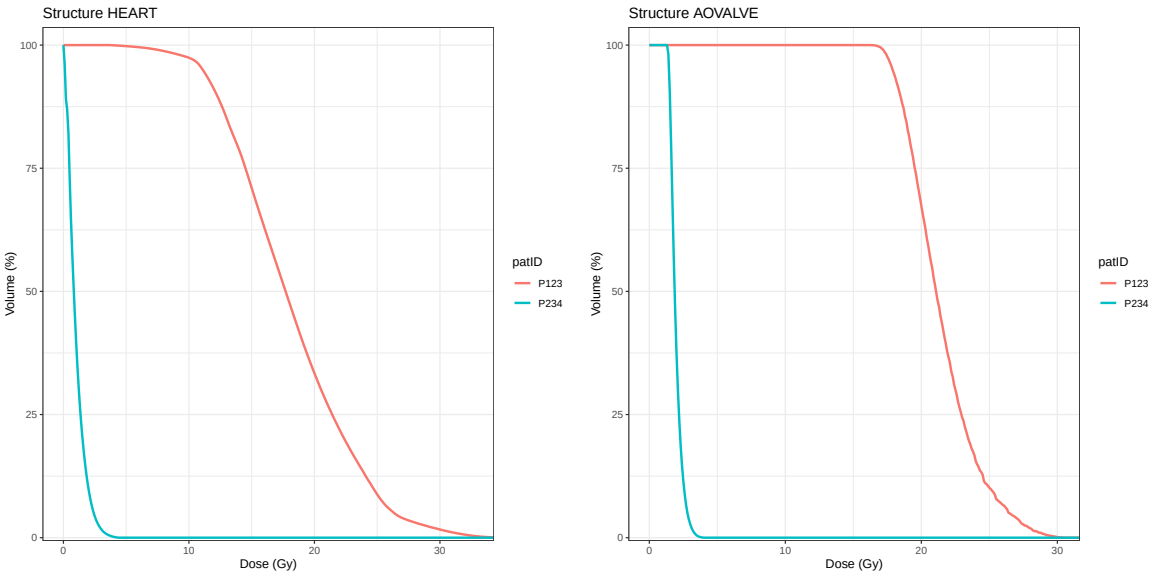
```
showDVH(dataMZ, byPat=TRUE)
```

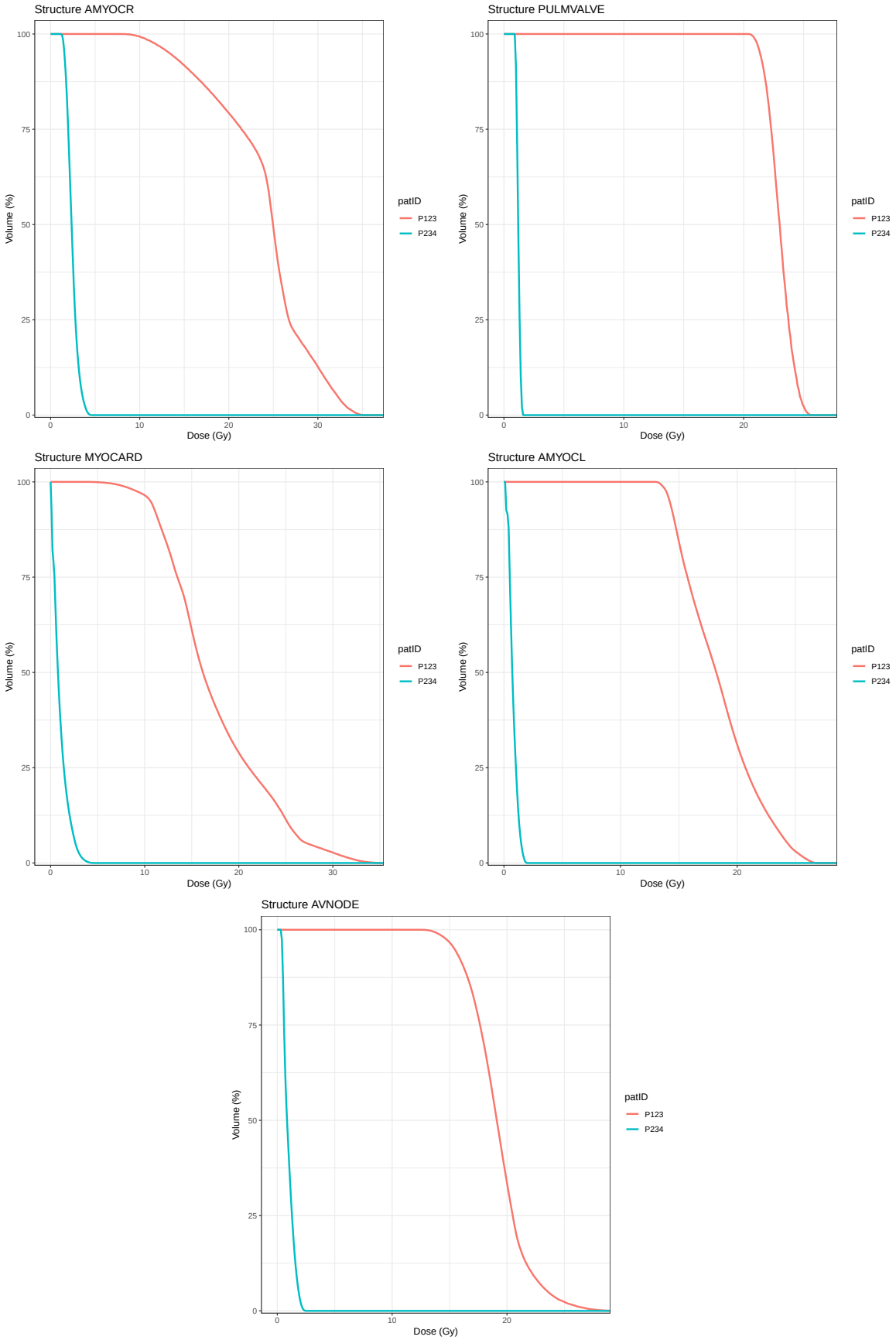




Patient IDs and structures can be selected with the `patID="⟨ID⟩"` option and the `structure="⟨NAME⟩"` option. With `fixed=FALSE`, both accept regular expressions. By default, IDs and structure names are matched exactly. By default, all patients/structures are shown.

```
showDVH(dataMZ, byPat=FALSE, patID=c("P123", "P234"))
```

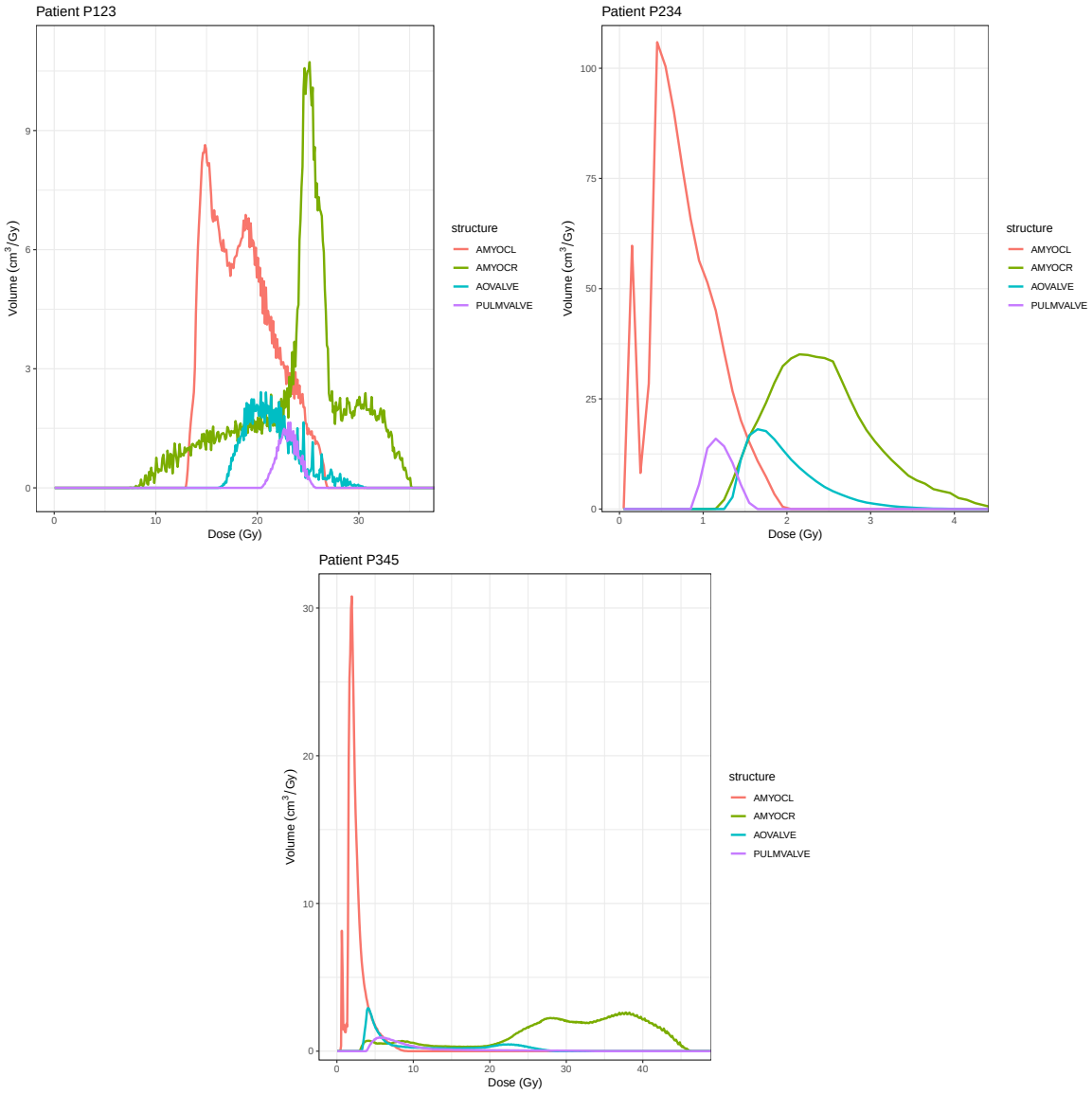




By default, the relative DVH is shown. Absolute volume (if available in the input files) can be plotted

with the `rel=FALSE` option. For differential DVH, set `cumul=FALSE`.

```
# match structures containing "VALVE" and "AMYOC"
showDVH(dataMZ, cumul=FALSE, rel=FALSE,
        structure=c("VALVE", "AMYOC"), fixed=FALSE)
```



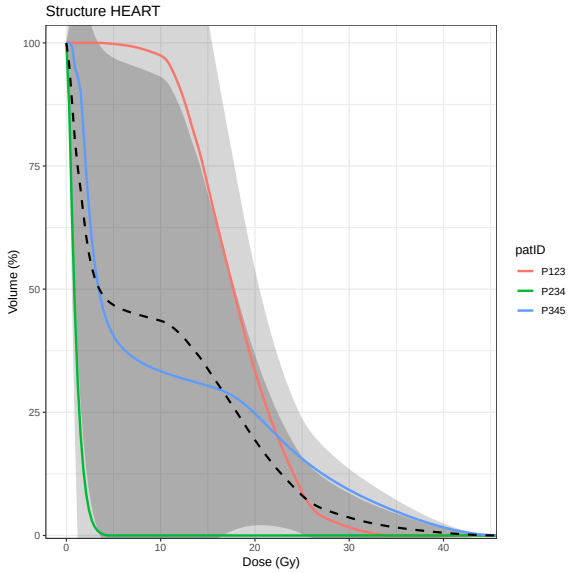
Option `thresh` allows to restrict the range of the x-axis such that only relative volumes larger than `thresh` appear. Use option `show=FALSE` to prevent the diagrams from being shown if you just need the returned object (here: `dvhPlot`) to later save the diagrams to file.

```
# just save the diagram but don't show it
dvhPlot <- showDVH(dataMZ, structure=c("HEART", "AOVALVE", "AVNODE"),
                   rel=FALSE, thresh=0.001, show=FALSE)
```

With option `addMSD=TRUE`, the diagram also shows the point-wise mean DVH as a black curve as well as grey shaded areas for the regions defined by the point-wise 1 standard deviation and 2 standard

deviations around this mean (see section 4.4). With `byPat=FALSE` and `addMSD=TRUE`, the average DVH for one structure over many patients can be visually assessed.

```
# add point-wise mean DVH and 1 SD/2 SD regions
showDVH(dataMZ, structure="HEART", byPat=FALSE, addMSD=TRUE)
```



5.2 Save cumulative DVH diagrams to file

DVH diagrams can be saved to file using `saveDVH()`. A file name pattern can then be supplied to option `file`. By using different file extensions like `.pdf`, `.jpg`, `.png`, different graphics formats can be automatically selected. In addition, the width and height of the diagram can be specified in inch.

```
saveDVH(dvhPlot, file="c:/folder/dvh.pdf", width=7, height=5)
```

6 Quality assurance constraints on the dose-volume relationship

For quality assurance, it is possible to define, check, and visualize constraints on the dose-volume relationship for DVHs.

6.1 Define constraints

A DVH constraint is a character string that consists of three parts: The DVH metric (see section 4.1), a comparison operator among `<, >, <=, >=`, and the reference value together with the measurement unit – one among among `Gy, cGy, cc, %`. 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 10cm^3 of the structure should have received 20% of the prescription dose), or `"D10CC > 500cGy"` (The "hottest" 10cm^3 of the structure should have received more than

Table 2: Example for pasted constraints.

Constraints that apply to all patients and to all structures

```
"D10cc < 20%"
"V5cGy > 100cc"
"DMEAN < 10Gy"
```

Constraints that apply to some patients and to all structures

"constraint"	"patID"
"D10cc < 20%"	"P123"
"V5cGy > 100cc"	"*"
"DMEAN < 10Gy"	"P234"

Constraints that apply to some patients and to some structures

"constraint"	"patID"	"structure"
"D10cc < 20%"	"P123"	"*"
"V5cGy > 100cc"	"*"	"HEART"
"DMEAN < 10Gy"	"P234"	"AOVALVE"

500cGy). Constraints can also apply to the dose mean, median, and standard deviation as well as to the gEUD and to the (N)TCP.

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, the constraint can be a `character` vector with elements like the examples above.
- If constraints apply only to some patients/structures, the constraint 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 the constraint applies to all patients. `structure` must be either a character string with a valid structure name, or `"*"` if the 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.

Alternatively, it is possible to specify a set of constraints as a table in a text file with one row per constraint and one column for the constraint expression, structure, and patient ID. A table like this can be created in a spreadsheet program like Excel (fig. 1), be exported to a tab-delimited text-file, and be read in by function `readConstraints()`. Table 2 shows some examples.

```
dataConstr <- readConstraints("constraints.txt", dec=".", sep="\t")
```

The constraint data frame `dataConstr` is built into `DVHmetrics` and applies to the `dataMZ` DVH data.

```
dataConstr     # show defined constraints and their scope

##          constraint structure patID
```

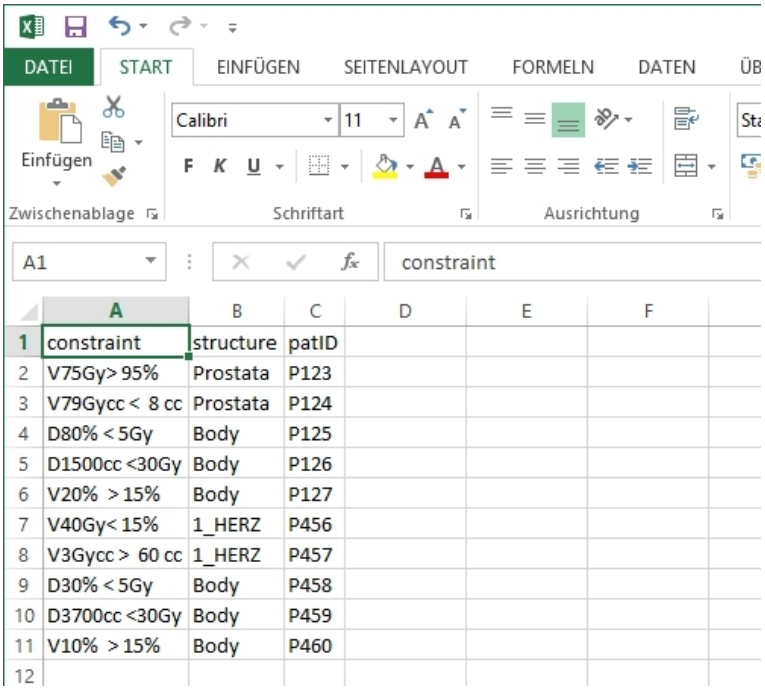


Figure 1: Defining constraints in a spreadsheet program like Excel

```
## 1       V1.2Gy> 60%    AVNODE  P123
## 2 V10Gy_cc <  18 cc    AMOYCL  P123
## 3      D80% < 1.8Gy   AOVALVE  P234
## 4        D200cc >2%     HEART  P234
## 5        V4%  > 25%     HEART  P345
## 6       V7.5Gy< 10%     HEART  P345
```

For checking constraints, and for calculating the difference between the observed DVH and the constraint, the DVH is linearly interpolated, by using monotone Hermite splines, or by local polynomial kernel regression.

6.2 Check constraints

Constraints are checked with `checkConstraint()`. The output returns information on the observed value of the tested metric, on the compliance with respect to this metric, and on the absolute/relative deviation in volume as well as in dose to the specified constraint value. The units for the absolute deviation are those used in the constraint expression. When the constraint defines a point in dose-volume space, `checkConstraint()` reports another quantitative measure for the degree of violation: The closest point on the DVH to the constraint as well as its Euclidean distance to the constraint point.

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 closest axis (dose and volume) along the same direction. In doing so, the

devation from the expected volume per dose and the devation from the expected dose per volume are condensed in a single metric.

As an example, we use the DVHs and corresponding constraints that are built into the `DVHmetrics` package.

```
## store result in object cc to save to file later
cc <- checkConstraint(dataMZ, constr=dataConstr)
print(cc, digits=2)            # show output with 2 decimal places

##   patID structure      constraint observed compliance deltaV deltaVpc
## 1  P123    AVNODE  V1.2GY_% > 60%    100.0       TRUE     40       67
## 2  P234     HEART   D200CC_% > 2%      2.2       TRUE     27       13
## 3  P234   AOVALVE D80%_GY < 1.8GY      1.6       TRUE    -23      -29
## 4  P345     HEART     V4%_% > 25%     66.2       TRUE     41      165
## 5  P345     HEART  V7.5GY_% < 10%     35.6      FALSE     26      256
##   deltaD deltaDpc dstMin dstMinRel ptMinD ptMinV
## 1  17.38   1448.4  17.35    1443.7   18.5     61
## 2   0.16      7.8   0.16       7.8    2.2    200
## 3  -0.20    -11.2   0.20      11.2    1.6     80
## 4  29.11    727.8  13.61     204.8   12.2     36
## 5  21.79    290.5  17.93     209.3   23.2     19
```

The result from a constraint check can be saved with function `saveConstraint()` that works like `saveMetric()` (see section 4.2).

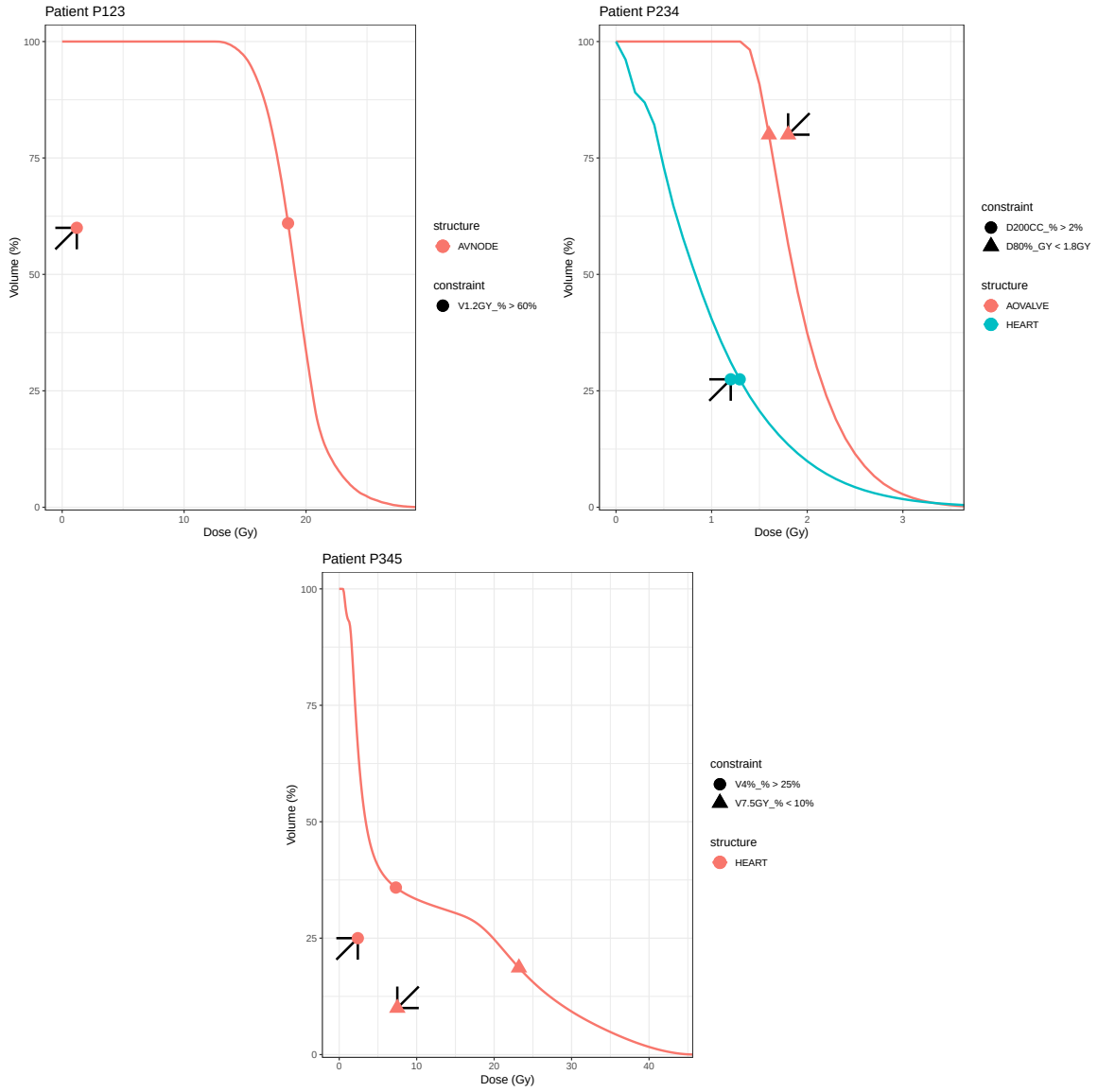
```
saveConstraint(cc, file="c:/folder/constrCheck.txt")
```

6.3 Visualize constraints

Constraints that define a point in dose-volume space can be visualized in a DVH with relative volume or absolute volume. The constraints will be converted to match the DVH plot. Only patients and structures within the scope of the defined constraints are shown. The diagram also shows the point on the DVH closest to the constraint. This can be verified visually only if the aspect ratio of the diagram is 1.

As in `showDVH()` (see section 5.1), either one diagram per patient with multiple structures is shown (`byPat=TRUE`), or one diagram per structure with multiple patients (`byPat=FALSE`).

```
## plot relative volume
showConstraint(dataMZ, constr=dataConstr, byPat=TRUE)
```




```
## plot absolute volume - store result in sc to save to file later
sc <- showConstraint(dataMZ, constr=dataConstr,
                     byPat=FALSE, rel=FALSE)
```

The result can be saved using `saveDVH()` as demonstrated in section 5.2.

```
saveDVH(sc, file="c:/folder/dvhConstraint.pdf")
```

7 BED, EQD2, Isoeffective Dose

The linear-quadratic model for the proportion of surviving cells S after dose d is (IAEA & ICRU, 2008):

$$S = e^{-(\alpha d + \beta d^2)}$$

According to the model, the biologically effective dose (BED) with total dose D, fraction dose d and a tissue-dependent α/β ratio is:

$$\text{BED} = D\left[1 + \frac{d}{\alpha/\beta}\right]$$

Given two different fractionation schemes, the total dose D_2 for the new fraction dose d_2 that corresponds to the total dose D_1 from the reference fraction dose d_1 can be calculated from solving the following equation for the desired measure:

$$\frac{D_2}{D_1} = \frac{d_1 + (\alpha/\beta)}{d_2 + (\alpha/\beta)}$$

As a special case, the dose in 2Gy fractions biologically equivalent dose (EQD_2) is given by:

$$\text{EQD}_2 = D_1 \cdot \frac{d_1 + (\alpha/\beta)}{2 + (\alpha/\beta)} = \frac{\text{BED}}{1 + \frac{2}{\alpha/\beta}}$$

The following convenience functions allow for easy calculation of these measures:

```
getBED(D=50, fd=2.5, ab=c(2, 3, 4))
```

```
##      BED fractDose ab
## 1 112.50       2.5  2
## 2  91.67       2.5  3
## 3  81.25       2.5  4
```

```
getEQD2(D=50, fd=2.5, ab=c(2, 3, 4))
```

```
##    EQD2 fractDose ab
## 1 56.25       2.5  2
## 2 55.00       2.5  3
## 3 54.17       2.5  4
```

```
getIsoEffD(D1=70, fd1=2, fd2=3, ab=c(3.5, 10))
```

```
## [1] 59.23 64.62
```

The same functions can be used to convert complete DVHs to BED, EQD2, or to the iso-effective dose corresponding to some other fraction dose.

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

```
## DVH: Patient 'John Doe' (ID P123), structure 'HEART' (600 CC), Dose: 0.11-68GY
```

Acknowledgements

The authors thank Marcus Stockinger for ideas on checking quality assurance constraints as well as Sandra Bührdel, Hannes Rennau, Ulrich Wolf, Bjorne Riis, Nico Banz, and Michael R. Young for example DVH files exported from different treatment planning systems.

References

Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2024). shiny: Web application framework for R [Computer software]. URL https://CRAN.R-project.org/package=shiny (R package version 1.11.1)

Fritsch, F. N., & Carlson, R. E. (1980). Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, *17*, 238–246.

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).

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

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 [abstract]. *Medical Physics*, *26*(6), 1100.

Posit Inc. (2023). RStudio: Integrated development environment for R [Computer software]. URL https://posit.co/products/open-source/rstudio/ (Version 2023.03.0)

R Development Core Team. (2025). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. URL https://www.r-project.org/

Ruppert, D., Sheather, S. J., & Wand, M. P. (1995). An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, *90*(1), 1257–1270.

Thompson, R. F. (2025). RadOnc: Analytical tools for radiation oncology [Computer software]. URL https://CRAN.R-project.org/package=RadOnc (R package version 1.1.9)

Wand, M. P., & Jones, M. C. (1995). *Kernel smoothing*. Boca Raton, FL: Chapman & Hall/CRC.

Wu, Q., Mohan, R., Niemierko, A., & Schmidt-Ullrich, R. (2002). Optimization of intensity-modulated radiotherapy plans based on the equivalent uniform dose. *International Journal of Radiation Oncology · Biology · Physics*, *52*(1), 224–235.