

# Analyzing dose-volume histograms using DVHmetrics for R

Daniel Wollschlaeger                      Heiko Karle  
wollschlaeger@uni-mainz.de              karle@uni-mainz.de

University Medical Center Mainz, Germany  
January 29, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Interfaces</b>	<b>2</b>
2.1	R command line interface . . . . .	2
2.2	Web application . . . . .	2
<b>3</b>	<b>Read DVH text data</b>	<b>3</b>
<b>4</b>	<b>DVH metrics</b>	<b>5</b>
4.1	Calculate DVH metrics . . . . .	5
4.2	Save DVH metrics to file . . . . .	9
<b>5</b>	<b>DVH diagrams</b>	<b>9</b>
5.1	Plot DVH diagrams . . . . .	9
5.2	Save cumulative DVH diagrams to file . . . . .	13
<b>6</b>	<b>Quality assurance constraints on the dose-volume relationship</b>	<b>13</b>
6.1	Define constraints . . . . .	13
6.2	Check constraints . . . . .	15
6.3	Visualize constraints . . . . .	16
	<b>References</b>	<b>17</b>

## 1 Introduction

DVHmetrics is an add-on package for the free statistical environment R<sup>1</sup> (R Development Core Team, 2014) with applications in radiation oncology. It provides functionality to read dose-volume-histogram

---

<sup>1</sup>Free short introductions to R can be found at <http://www.splusebook.com/RIntro/RCourse.pdf> and <http://www.statmethods.net/>.

(DVH) text files, to calculate DVH metrics, and to plot DVHs. In addition, it checks and visualizes quality assurance constraints for the DVH.<sup>2</sup>

To install `DVHmetrics`, you need a current version of R and be online. Preferably, a free development environment like RStudio ([RStudio Inc, 2015](#)) or Architect ([OpenAnalytics BVBA, 2015](#)) 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 application

For users who are unfamiliar with R, `DVHmetrics` includes a `shiny`-based web application ([RStudio Inc., 2015](#)) 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:

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

---

<sup>2</sup>For a solution that also reads files in DICOM-RT format and has a broader scope in supporting radiation oncology, see the `RadOnc` package for R ([Thompson, 2014](#)).

### 3 Read DVH text data

To import DVH data into R, it should be exported as a plain text file from Varian Eclipse<sup>(TM)</sup> (Aria Version 10 or 11), from CadPlan<sup>(TM)</sup>, or from MasterPlan<sup>(TM)</sup>. Cumulative and differential DVHs are supported, as are sum plans. The measurement unit for absolute dose can be Gy or cGy, the measurement unit for volume has to be cm<sup>3</sup>. DVH text files can be read with `readDVH()`.

Example: Read one Eclipse file `dvhFile.txt` from folder `"c:/folder"` and save the result in object `res`.<sup>3</sup>

```
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.<sup>4</sup>

```
print(dataMZ)
```

```
Data courtesy of Department of Radiation Oncology (Prof. Dr. Schmidberger), University Medical
```

```
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`.

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

```
Data courtesy of Department of Radiation Oncology (Prof. Dr. Schmidberger), University Medical
```

```
DVH list of 3 lists - 1 for each patient:
```

```
DVH list:
```

```
Patient 'John Doe' (ID P123, prescription dose 60GY) with 7 Structures:
```

---

<sup>3</sup>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.

<sup>4</sup>Data courtesy of Department of Radiation Oncology (Prof. Dr. Schmidberger), University Medical Center Mainz, Germany.

```
DVH: Structure HEART (600CC), Dose: 0-62GY
DVH: Structure AOVALVE (12CC), Dose: 0-62GY
DVH: Structure AMYOCR (57CC), Dose: 0-62GY
DVH: Structure PULMVALVE (3.8CC), Dose: 0-62GY
DVH: Structure MYOCARD (310CC), Dose: 0-62GY
DVH: Structure AMYOCL (60CC), Dose: 0-62GY
DVH: Structure AVNODE (38CC), Dose: 0-62GY
```

DVH list:

Patient 'Jane Doe' (ID P234, prescription dose 60GY) with 7 Structures:

```
DVH: Structure AMYOCR (53CC), Dose: 0-63GY
DVH: Structure AVNODE (43CC), Dose: 0-63GY
DVH: Structure HEART (730CC), Dose: 0-63GY
DVH: Structure AOVALVE (15CC), Dose: 0-63GY
DVH: Structure PULMVALVE (6.7CC), Dose: 0-63GY
DVH: Structure MYOCARD (330CC), Dose: 0-63GY
DVH: Structure AMYOCL (81CC), Dose: 0-63GY
```

DVH list:

Patient 'Jane Smith' (ID P345, prescription dose 60GY) with 7 Structures:

```
DVH: Structure HEART (550CC), Dose: 0-62GY
DVH: Structure AOVALVE (11CC), Dose: 0-62GY
DVH: Structure PULMVALVE (5.7CC), Dose: 0-62GY
DVH: Structure MYOCARD (270CC), Dose: 0-62GY
DVH: Structure AMYOCL (43CC), Dose: 0-62GY
DVH: Structure AMYOCR (52CC), Dose: 0-62GY
DVH: Structure AVNODE (24CC), 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")
```

## 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 structure.
    - \* "DMAX": The maximum dose of the structure.
    - \* "DSD": The standard deviation of the dose in the structure.
    - \* "DRX": The prescription dose.
  - A *free* DVH metric is a character string which has three mandatory elements and one optional element in the following order:
    - \* 1<sup>st</sup> letter "D" or "V": "D" If the requested value is a dose, "V" if it is a volume.
    - \* 2<sup>nd</sup> 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.
    - \* 3<sup>rd</sup> element `<measurement unit>`: The measurement unit for the 2<sup>nd</sup> element of the metric. Absolute volumes are indicated by "CC" for cm<sup>3</sup>, relative volumes by "%". Absolute doses are indicated by "Gy" for Gray or "cGy" for Centigray, relative doses by "%".
    - \* Optional 4<sup>th</sup> element `_<measurement unit>`: The measurement unit of the output value. Possible units are the same as for the 3<sup>rd</sup> 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. Metrics are calculated using linear interpolation between actual DVH points – without extrapolating beyond the observed volume or dose.
- Option `patID`: Which patient IDs should be analyzed. IDs are interpreted as regular expressions matched against those found in the DVH files. Also supply `fixed=TRUE` for exact matching. If missing, the metrics are calculated for all patients.
- Option `structure`: Which structure should be analyzed. Structure names are interpreted as regular expressions matched against those found in the DVH files. Also supply `fixed=TRUE` for exact matching. 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

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 <sup>3</sup>
"V10%"	relative dose	%	relative volume	%
"V10%_CC"	relative dose	%	absolute volume	cm <sup>3</sup>
"D10CC"	absolute volume	cm <sup>3</sup>	absolute dose	as in DVH
"D10%_cGy"	relative volume	%	absolute dose	cGy
"DMEAN"	—	—	absolute dose	as in DVH
"DSD"	—	—	absolute dose	as in DVH
"DMIN"	—	—	absolute dose	as in DVH
"DMAX"	—	—	absolute dose	as in DVH

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")

  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	AMYOCL	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"))
```

	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%	AMYOCL	P234
7	23.208	D10%	AMYOCL	P123
5	24.391	D10%	PULMVALVE	P123
1	25.056	D10%	AOVALVE	P123
3	30.710	D10%	AMYOCL	P123
10	0.000	V5GY	AMYOCL	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`.

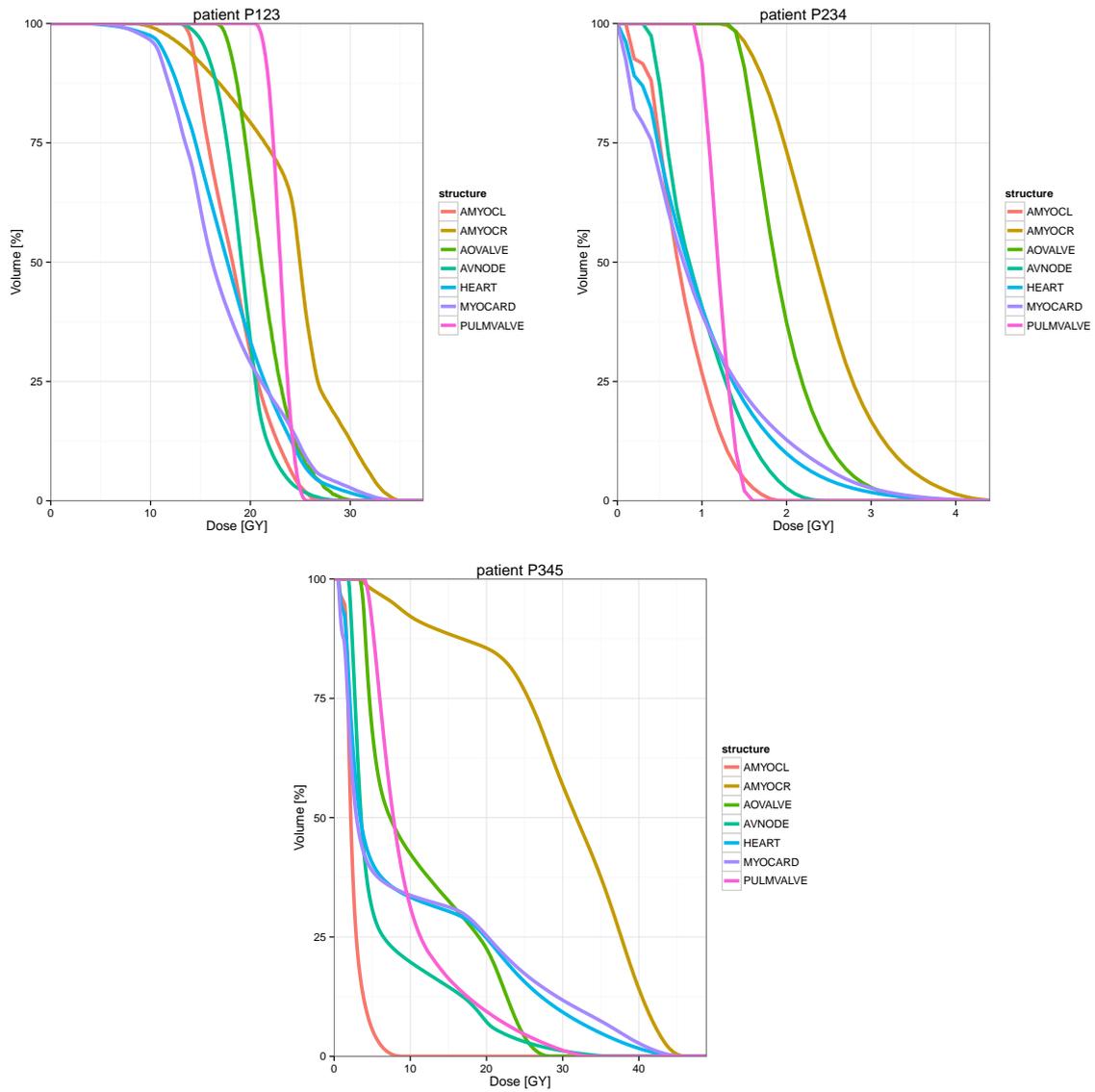
```
saveMetric(met, file="c:/folder/metrics.txt", quote=TRUE)
```

## 5 DVH diagrams

### 5.1 Plot DVH diagrams

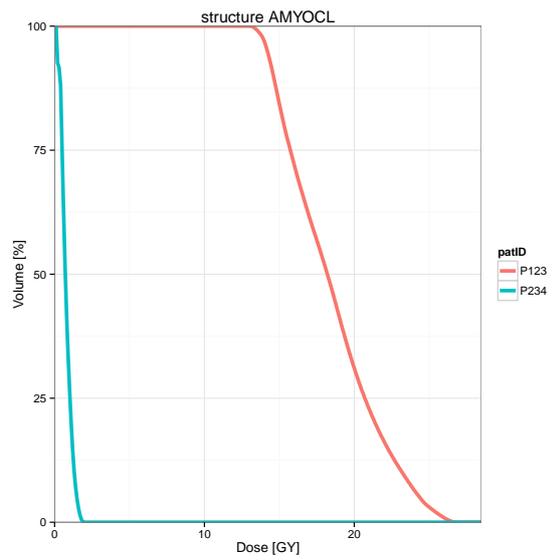
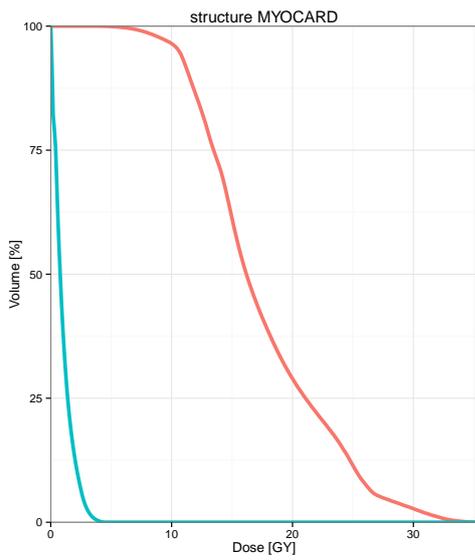
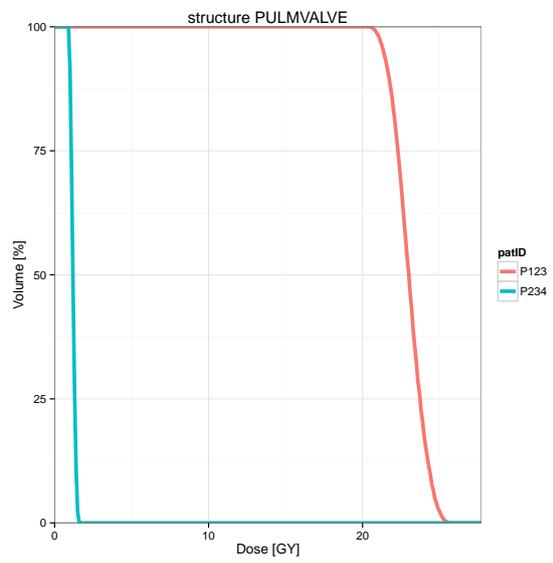
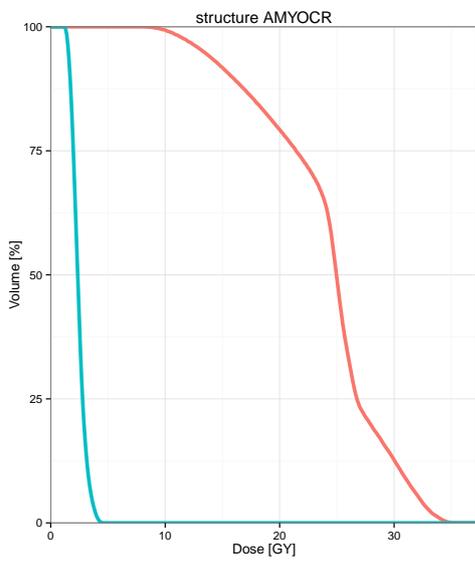
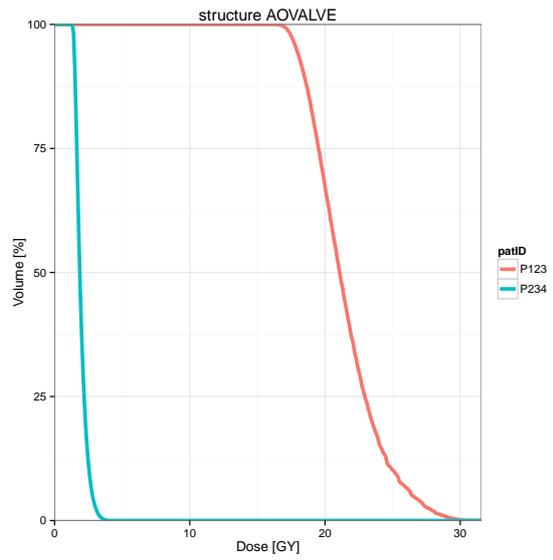
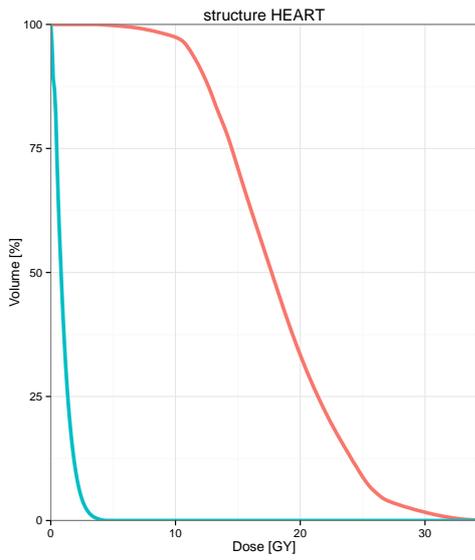
Cumulative as well as differential DVH diagrams can be generated with `showDVH()`. If you are using RStudio or Architect, all produced diagrams are accessible in the plots tab by clicking on the left and right arrows. Depending on the option `byPat`, each DVH diagram either shows one patient with multiple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

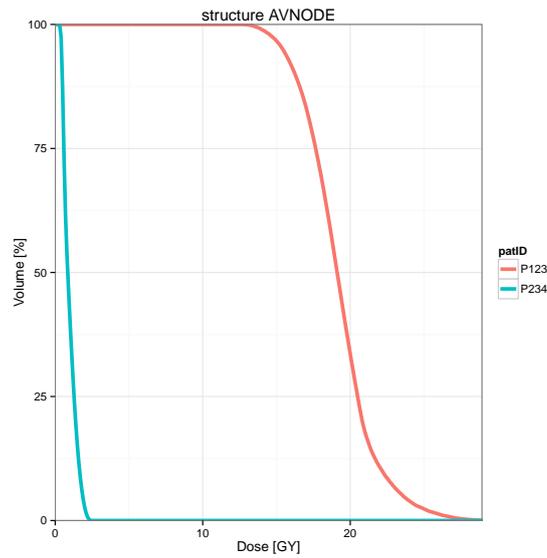
```
showDVH(dataMZ, byPat=TRUE)
```



Patient IDs and structures can be selected with the `patID="⟨ID⟩"` option and the `structure="⟨NAME⟩"` option. Both accept regular expressions. For exact matching, also supply `fixed=TRUE`. By default, all patients/structures are shown.

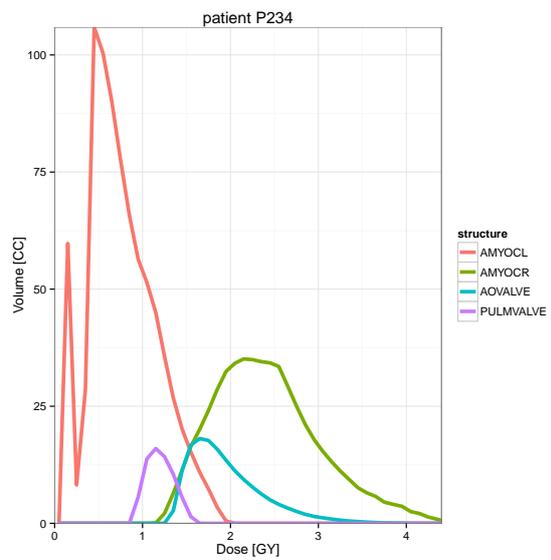
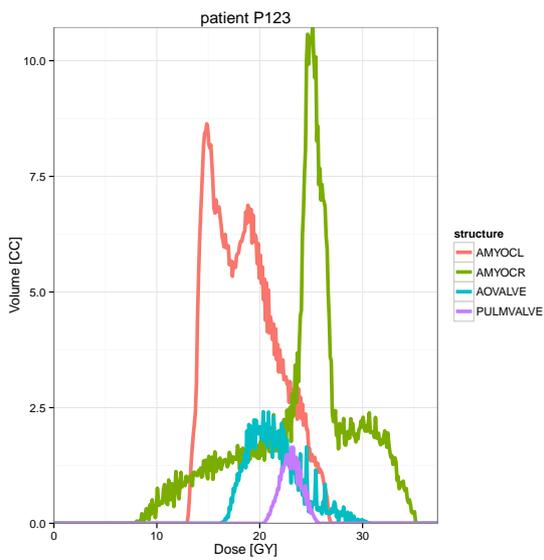
```
showDVH(dataMZ, byPat=FALSE, patID=c("P123", "P234"))
```

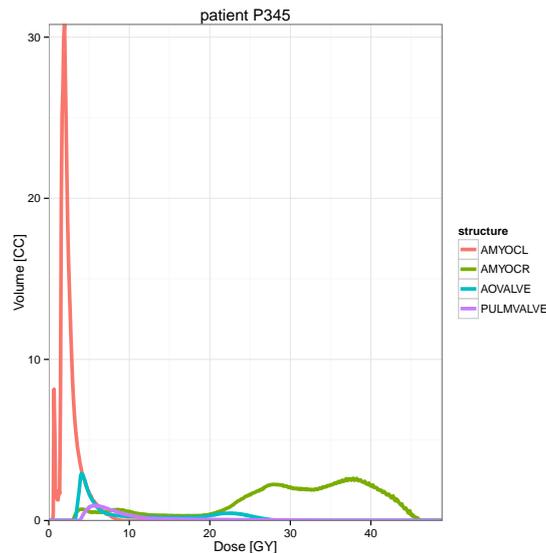




By default, the relative DVH is shown. Absolute volume can be plotted with the `rel=FALSE` option. For differential DVH, set `cumul=FALSE`.

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





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)
```

## 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<sup>3</sup> of the structure should have received 20% of the prescription

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"

dose), or "D10CC > 500cGy" (The “hottest” 10cm<sup>3</sup> of the structure should have received more than 500cGy).

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
```

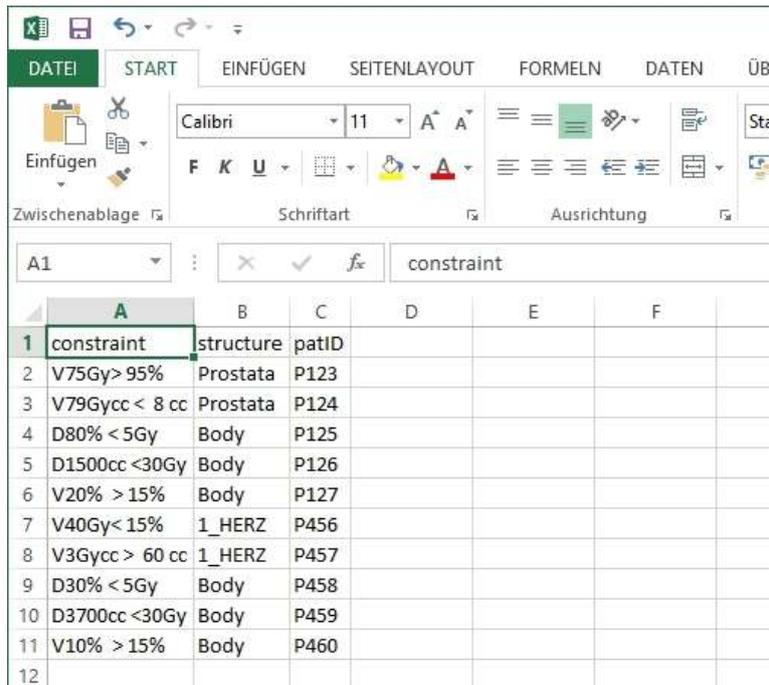


Figure 1: Defining constraints in a spreadsheet program like Excel

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.

## 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.

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	AOVALVE D80%_GY < 1.8GY	1.6	TRUE	-23	-29
3	P234	HEART D200CC_% > 2%	2.2	TRUE	525	262
4	P345	HEART V4%_% > 25%	100.0	TRUE	75	300
5	P345	HEART V7.5GY_% < 10%	35.6	FALSE	26	256

	deltaD	deltaDpc	dstMin	ptMinD	ptMinV
1	17.38	1448.4	17.36	18.5	61
2	-0.20	-11.2	0.22	1.6	80
3	0.16	7.8	1.89	2.2	198
4	29.11	727.8	13.61	12.2	36
5	21.79	290.5	17.93	23.2	19

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

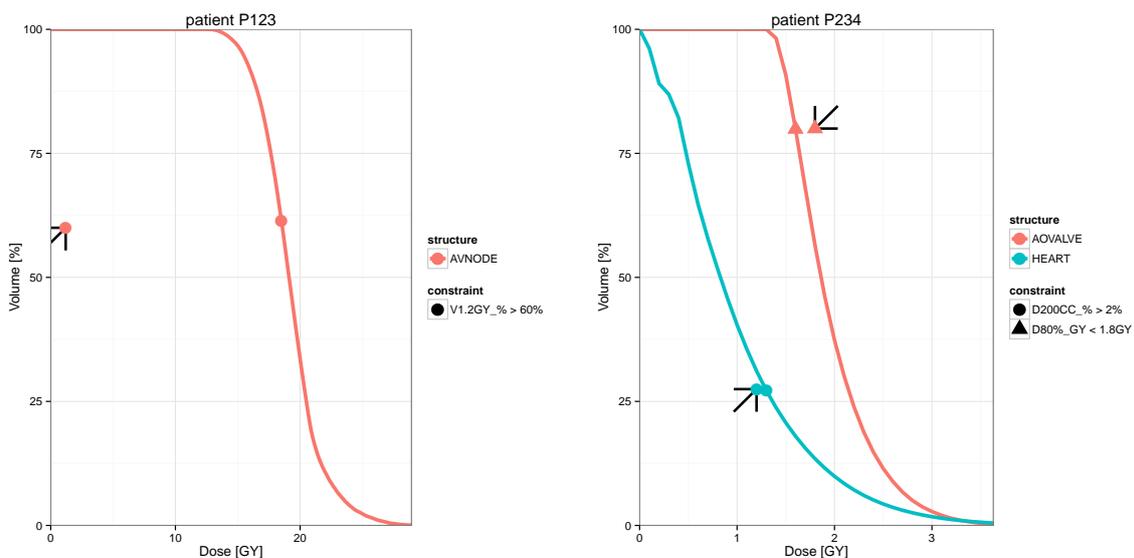
```
saveConstraint(cc, file="c:/folder/constrCheck.txt")
```

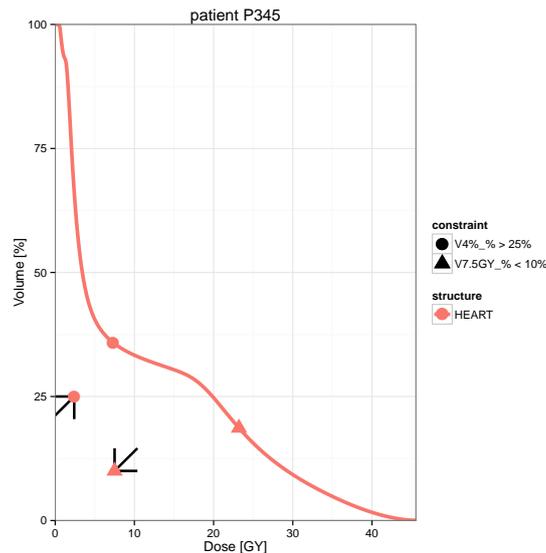
### 6.3 Visualize constraints

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")
```

## Acknowledgements

The authors thank Marcus Stockinger for ideas on checking quality assurance constraints, Sandra Bührdel for example DVH files exported from CadPlan, and Hannes Rennau for example DVH files exported from MasterPlan.

## References

- OpenAnalytics BVBA. (2015). Architect [Computer software]. URL <http://www.openanalytics.eu/architect/> (Version 0.9.5)
- R Development Core Team. (2014). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. URL <http://www.r-project.org/>
- RStudio Inc. (2015). RStudio [Computer software]. URL <http://www.rstudio.org/> (Version 0.99.175)
- RStudio Inc. (2015). shiny: Web application framework for R [Computer software]. URL <http://CRAN.R-project.org/package=shiny> (R package version 0.11)
- Thompson, R. F. (2014). RadOnc: Analytical tools for radiation oncology [Computer software]. URL <http://CRAN.R-project.org/package=RadOnc> (R package version 1.0.9)