Radiation Oncology (RadOnc) Tools

Reid F. Thompson

February 20, 2014

Contents

1 Introduction

The *RadOnc* package provides a number of tools for the import and analysis of dose-volume histogram (DVH) data used routinely in Radiation Oncology clinical practice and research. Supported formats for data import currently include:

- Varian's Aria/Eclipse platform (v.10 and v.11)
- CadPlan (for historical/archival data)
- DICOM-RT files

The functionality contained herein also enables visualization of dosimetric and volumetric data, and statistical comparison among multiple DVHs and three-dimensional structures. In order to use these tools, you must first load the *RadOnc* package:

```
> library(RadOnc)
```

It is assumed that the reader is already familiar with DVH analysis. If this is not the case, consult the relevant literature for a thorough treatment of the subject (1).

Throughout this vignette, we will be exploring actual data for 2 patients, each possessing a set of 10 structures (including organs at risk and treatment planning volumes). We will also demonstrate rudimentary three-dimensional structural processing.

2 Changes for *RadOnc* in current release

- Functionality expanded for `[]` accessor method to `DVH.list` and `structure.list` objects to enable pattern matching (e.g. `janedoe["KIDNEY$"]` and handling of redundant structure names in `DVH.list`.
- New methods for `as()` function to convert a list of `DVH` and/or `DVH.list` objects to a single combined `DVH.list` object and a list of `structure3D` and/or `structure.list` objects to a single combined `structure.list` object.
- Expanded `read.DICOM.RT()` functionality to include calculation of DVHs from dose grid information. DVH calculations can be enabled by specifying logical parameter `DVH=TRUE` (default). Also added `calculate.DVH()` function to perform DVH calculation given one or more input structure set(s) and dose grid data. This function now also implicitly extends `read.DVH(..., method="dicom")` to enable DVH calculation from DICOM-RT data.
- `LQE()` now supports fractionation specification in either dose per fraction or total number of fractions (parameters `N` and `dose.units` were introduced for this purpose).
- Added `janedoe.RTdata` to package data contents in order to provide an example of a `RTdata` object.
- `DVH` class expanded to include patient identifying information (new parameters `patient` and `ID`)

3 DVH Analysis

3.1 DVH file import

The `read.DVH()` function is designed to take an input text file and output a list of DVH data objects containing all relevant data. Supported file types currently include Varian's Aria/Eclipse platform (v.10 and v.11) and CadPlan (for archival/historical data). Other treatment planning systems are not currently supported however will be included in future releases.

For Varian-specific file types, data must be exported directly from the treatment planning system and should include all DVH structures of interest. In Eclipse, this is accomplished via the "Export DVH in Tabular Format..." option, accessed by right-clicking over DVHs in Plan Evaluation mode. Exported files will adhere to the following form (an example file, 50 lines of which are shown here, is contained within this release of the *RadOnc* package):

```
Patient Name          : Doe, Jane (1111111111)
Patient ID            : 1111111111
Comment               : DVHs for one plan
Date                  : 05.24.2013  00:00:00
Type                  : Cumulative Dose Volume Histogram
Description           : The cumulative DVH displays the percentage (relative)
                        or volume (absolute) of structures that receive a dose
                        equal to or greater than a given dose.

Plan: PLAN_NAME
Prescribed dose [cGy]: 5500.0
% for dose (%): 100.0

Structure: LIVER
Approval Status: Unapproved
Plan: PLAN_NAME
Course: COURSE_1
Volume [cc]: 1635.9
Dose Cover.[%]: 100.0
Sampling Cover.[%]: 100.0
Min Dose [cGy]: 42.7
Max Dose [cGy]: 5634.2
Mean Dose [cGy]: 707.0
Modal Dose [cGy]: 99.5
Median Dose [cGy]: 276.4
STD [cGy]: 917.2
Equiv. Sphere Diam. [cm]: N/A
Conformity Index: N/A
Gradient Measure [cm]: N/A
```

Dose [cGy]	Relative dose [%]	Ratio of Total Structure Volume [%]
0	0	100
5	0.0909091	100
10	0.181818	100
15	0.272727	100
20	0.363636	100
25	0.454545	100
30	0.545455	100
35	0.636364	100
40	0.727273	100
45	0.818182	99.9247
50	0.909091	99.2638
55	1	98.1752
60	1.09091	96.8538
65	1.18182	95.3989
70	1.27273	93.8907
75	1.36364	92.3371
80	1.45455	90.7697
85	1.54545	89.2061
90	1.63636	87.6496

...
...

This DVH data may be imported using the `read.DVH()` function, with an example shown here:

```
> read.DVH(file="Jane_Doe.dvh", type="aria10", verbose=TRUE)

Reading DVH file ('/Library/Frameworks/R.framework/Versions/3.0/Resources/library/RadOnc/
  Patient: Doe, Jane (1111111111)
  Plan: PLAN_NAME
  Dose: 5500cGy (at 100% isodose line)
  ..Importing structure: LIVER  [volume: 1635.9cc, dose: 42.7 - 5634.2cGy]
  ..Importing structure: LEFT_KIDNEY  [volume: 195.7cc, dose: 75.8 - 3846.8cGy]
  ..Importing structure: STOMACH  [volume: 695.2cc, dose: 59 - 5353.2cGy]
  ..Importing structure: DUODENUM  [volume: 34.2cc, dose: 2707.8 - 5620.1cGy]
  ..Importing structure: RIGHT_KIDNEY  [volume: 223.9cc, dose: 102.4 - 4201.9cGy]
  ..Importing structure: CTV  [volume: 146.7cc, dose: 5168.6 - 5646.9cGy]
  ..Importing structure: PTV  [volume: 239.4cc, dose: 4749.8 - 5664.7cGy]
  ..Importing structure: SMALL_BOWEL  [volume: 232.2cc, dose: 59.6 - 4934.1cGy]
  ..Importing structure: CORD  [volume: 64.9cc, dose: 0 - 3442.8cGy]
  ..Importing structure: BODY  [volume: 25507.5cc, dose: 0 - 5664.7cGy]
```

Multiple files can be imported simultaneously, generating a list of `DVH.list` objects:

```
> DVHs <- read.DVH(file=c("Jane_Doe.dvh", "John_Doe.dvh"), type="aria10")
```

```
Reading DVH file ('/Library/Frameworks/R.framework/Versions/3.0/Resources/library/RadOnc/
  Patient: Doe, Jane (1111111111)
  Plan: PLAN_NAME
  Dose: 5500cGy (at 100% isodose line)
  ..Importing structure: LIVER  [volume: 1635.9cc, dose: 42.7 - 5634.2cGy]
  ..Importing structure: LEFT_KIDNEY  [volume: 195.7cc, dose: 75.8 - 3846.8cGy]
  ..Importing structure: STOMACH  [volume: 695.2cc, dose: 59 - 5353.2cGy]
  ..Importing structure: DUODENUM  [volume: 34.2cc, dose: 2707.8 - 5620.1cGy]
  ..Importing structure: RIGHT_KIDNEY  [volume: 223.9cc, dose: 102.4 - 4201.9cGy]
  ..Importing structure: CTV  [volume: 146.7cc, dose: 5168.6 - 5646.9cGy]
  ..Importing structure: PTV  [volume: 239.4cc, dose: 4749.8 - 5664.7cGy]
  ..Importing structure: SMALL_BOWEL  [volume: 232.2cc, dose: 59.6 - 4934.1cGy]
  ..Importing structure: CORD  [volume: 64.9cc, dose: 0 - 3442.8cGy]
  ..Importing structure: BODY  [volume: 25507.5cc, dose: 0 - 5664.7cGy]
Reading DVH file ('/Library/Frameworks/R.framework/Versions/3.0/Resources/library/RadOnc/
  Patient: Doe, John (5555555555)
  Plan: PLAN_NAME
  Dose: 5500cGy (at 100% isodose line)
  ..Importing structure: LIVER  [volume: 1366.8cc, dose: 0 - 5109.5cGy]
  ..Importing structure: SMALL_BOWEL  [volume: 206.2cc, dose: 0 - 5489cGy]
  ..Importing structure: DUODENUM  [volume: 93.1cc, dose: 0 - 5632cGy]
  ..Importing structure: STOMACH  [volume: 303.6cc, dose: 0 - 5571.5cGy]
  ..Importing structure: CTV  [volume: 88.4cc, dose: 5324 - 5643cGy]
  ..Importing structure: PTV  [volume: 155.7cc, dose: 4625.5 - 5643cGy]
  ..Importing structure: BODY  [volume: 17666.2cc, dose: 0 - 5643cGy]
  ..Importing structure: LEFT_KIDNEY  [volume: 154.2cc, dose: 0 - 2442cGy]
  ..Importing structure: RIGHT_KIDNEY  [volume: 155.1cc, dose: 0 - 5417.5cGy]
  ..Importing structure: CORD  [volume: 40.7cc, dose: 0 - 3052.5cGy]

$`1111111111_Doe, Jane`
[1] "List containing 10 DVH objects (LIVER, LEFT_KIDNEY, STOMACH, DUODENUM, RIGHT_KIDNEY

$`5555555555_Doe, John`
[1] "List containing 10 DVH objects (LIVER, SMALL_BOWEL, DUODENUM, STOMACH, CTV, PTV, BO
```

3.2 DVH calculation from dose grid

Alternatively, `read.DVH(..., method="dicom")` can be employed to extract DVH data from a DICOM-RT dataset including a dose grid (calculated from the treatment planning system) and one or more structure set(s). The DVH import implicitly calls the `calculate.DVH()` function without the user having to manually specify dose calculation, however the `calculate.DVH()` function can also be called explicitly to calculate or recalculate DVHs using specific input parameters (generally the user will not need to call this function explicitly). In any of the above cases, DVH calculation from dose grid data utilizes the method previously described by Straube et al. with trilinear interpolation of dosimetric information (2). Please note that this is a computationally intensive process and has not yet been optimized in the current release (future releases will include speed-ups of this dose calculation).

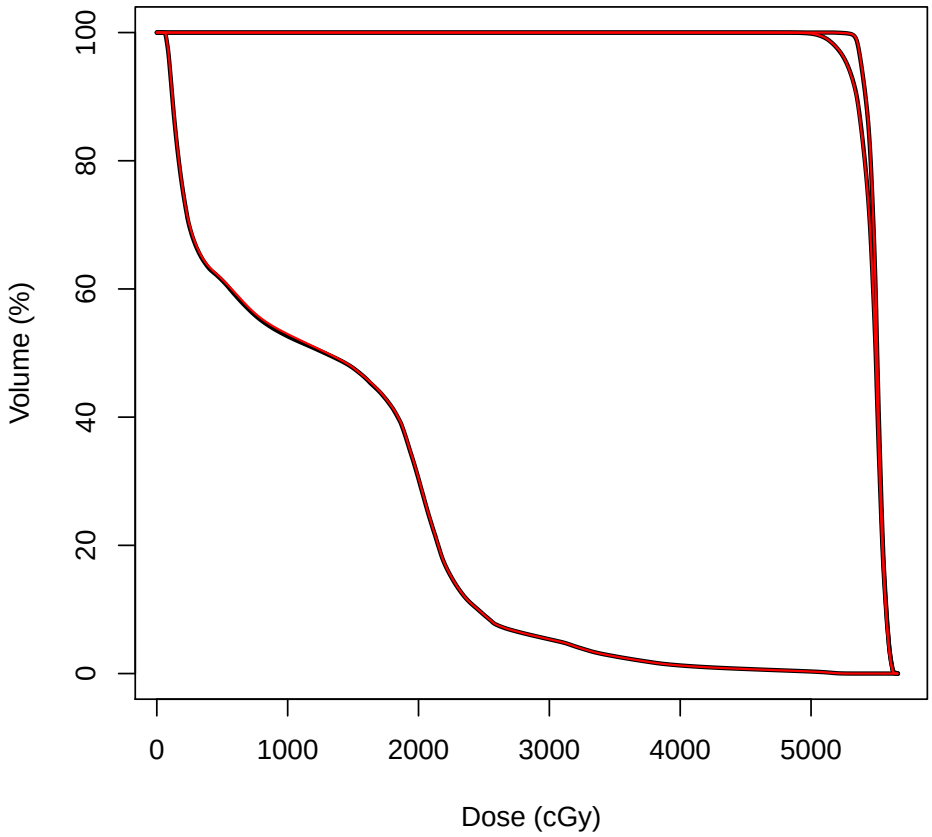

Figure 1: Comparison of DVH data for patient Jane Doe, calculated from dose grid data using the *RadOnc* package (black) or otherwise imported as pre-calculated DVH from the treatment planning system (red).

3.3 DVH list manipulation

The `read.DVH()` function returns a DVH list that can be manipulated in multiple ways. Subsets of DVH lists can be obtained using the `[]` modifier, and any number of DVH lists can be combined using the `c()` function. Additionally, single DVH objects can be directly accessed using the `[[]]` modifier, and individual elements of a DVH list may be directly replaced with other DVH objects using the `[[<-` function.

```
> janedoe[1:4]

[1] "List containing 4 DVH objects (LIVER, LEFT_KIDNEY, STOMACH, DUODENUM)"

> c(janedoe["PTV"], johndoe[c("CTV", "DUODENUM")])

[1] "List containing 3 DVH objects (PTV, CTV, DUODENUM)"

> johndoe[["CTV"]]

[1] "Structure: CTV (88.4cc), Dose: 96.80-102.60% (5500cGy prescribed), DVH: cumulative,

> janedoe[[1]] <- johndoe[["CTV"]]
> janedoe[1:4]

[1] "List containing 4 DVH objects (CTV, LEFT_KIDNEY, STOMACH, DUODENUM)"
```

Note that pattern matching (using regular expressions) may also be employed to select a subset of DVH elements from a DVH list:

```
> janedoe["KIDNEY$"]

[1] "List containing 2 DVH objects (LEFT_KIDNEY, RIGHT_KIDNEY)"

> janedoe[c(2,"IGHT.*")]

[1] "List containing 2 DVH objects (LEFT_KIDNEY, RIGHT_KIDNEY)"
```

Other list processing functions may be applied to DVH lists, enabling further data manipulation. The `rev()` function may be used to reverse the order of a DVH list, while the `names()` function may be used to extract (or set) the structure names for each DVH contained within the list. The `length()` function may be used to find the number of DVH objects contained within a DVH list, and the `lapply()` function can be used to perform a customizable set of operations on a DVH list and return a customizable set of values. Here are some examples employing each of these functions:

```
> names(janedoe)[1:4] <- c("A1", "B2", "C3", "D4")
> names(rev(janedoe[1:4]))

[1] "D4" "C3" "B2" "A1"
```

```
> length(johndoe)

[1] 10

> lapply(johndoe, function(DVH) { DVH[c("DMIN", "D50%", "DMAX", "V20%")] })

$LIVER
         cGy            %          cGy           cc
0.000000e+00 7.174856e-02 5.109500e+03 1.852850e+02

$SMALL_BOWEL
         cGy            %          cGy           cc
0.000000e+00 6.320805e-02 5.489000e+03 1.394640e+01

$DUODENUM
       cGy          %        cGy         cc
   0.00000   81.34012 5632.00000   83.10130

$STOMACH
         cGy            %          cGy           cc
0.000000e+00 6.726741e-02 5.571500e+03 3.195860e+01

$CTV
      cGy         %       cGy        cc
5324.0000 5324.0000 5643.0000   88.4095

$PTV
     cGy        %      cGy       cc
4625.500 4625.500 5643.000  155.735

$BODY
         cGy            %          cGy           cc
0.000000e+00 6.202236e-02 5.643000e+03 1.893130e+03

$LEFT_KIDNEY
         cGy            %          cGy           cc
0.000000e+00 7.279133e-02 2.442000e+03 1.860640e+01

$RIGHT_KIDNEY
       cGy          %        cGy         cc
   0.00000   24.25567 5417.50000   85.93270

$CORD
        cGy           %         cGy          cc
   0.000000    4.078601 3052.500000   17.496100
```

Patient name(s) and identifier(s) may be extracted from each DVH within the DVH list using the `$` modifier:

```
> janedoe[1:2]$patients

$LIVER

"Doe, Jane"

$LEFT_KIDNEY

"Doe, Jane"

> janedoe[3:4]$ID

$STOMACH

"1111111111"

$DUODENUM

"1111111111"
```

3.4 DVH data

Each DVH structure contains a variety of data related to the structure itself as well as the distribution of radiation dose within the structure volume. Detailed slot list and parameters are described in the `DVH-class` documentation accompanying the *RadOnc* package. Specific parameters can be extracted using the `[]` modifier, which can take as its argument a character string representation of the desired dose/volume parameter. For instance, the volume of duodenum receiving 20Gy or the dose to the top 2.5% (2.3286cc) of the volume can be extracted from DVH data as follows:

```
> johndoe[["DUODENUM"]]["V20Gy"]

     cc
76.5153

> johndoe[["DUODENUM"]]["D2.5%"]

       %
100.5605

> johndoe[["DUODENUM"]]["volume"] * 0.025

      cc
2.328598

> johndoe[["DUODENUM"]]["D2.3286cc"]

       %
100.5605
```

These parameters are entirely flexible and multiple parameters can be requested for a given DVH object at the same time. This functionality can also be applied to a DVH list using the `$` modifier.

```
> johndoe[["DUODENUM"]][c("V5%", "V20Gy", "D2.5%", "D2.3286cc", "Dmax")]

       cc        cc         %         %       cGy
  90.1912   76.5153  100.5605  100.5605 5632.0000

> johndoe[1:4]$"V20Gy,Dmax"

$LIVER
        cc        cGy
  21.35734 5109.50000

$SMALL_BOWEL
         cc         cGy
   9.370469 5489.000000
```

```
$DUODENUM
       cc       cGy
  76.5153 5632.0000

$STOMACH
        cc        cGy
  26.83795 5571.50000
```

Note that specific parameter keywords include: `"Dmax"` (maximum dose), `"Dmin"` (minimum dose), `"Dmean"` (mean dose), `"Dintegral"` (estimated integral dose), `"DRx"` (prescription dose), and `"volume"` (total structure volume). If an improper parameter is specified however, `NA` results will be returned for the affected parameter(s):

```
> johndoe[["DUODENUM"]][c("V5", "VGy", "volume", 2.5, "", "Dmax%")]

    <NA>     <NA>       cc     <NA>     <NA>        %
      NA       NA  93.1439       NA       NA 102.4000
```

Parameters may be further modified by `"()"` tags, which can alter the output format of data in the case of standard dosimetric parameters, for instance:

```
> johndoe[["LIVER"]][c("V10Gy(%)","D25%","D25%(Gy)")]

           %            %           Gy
14.191575545  0.833386291  0.008333863
```

Integral dose presents a special case, where additional tags can specify the range over which the integral dose is calculated (default is over the entire dose range).

```
> johndoe[["LIVER"]][c("Dintegral","Dintegral(>0cGy)")]

  cGy*cc   cGy*cc
368656.6 368656.6

> johndoe[["LIVER"]][c("Dintegral(<20Gy)","Dintegral(10-20Gy)")]

   Gy*cc    Gy*cc
3071.012 2493.167
```

3.5 Generalized equivalent uniform dose (gEUD) calculation

Generalized equivalent uniform doses (gEUD) can be calculated from one or more `DVH` or `DVH.list` objects using the `gEUD()` function. This calculation summarizes the entire DVH as a single dose which, when distributed evenly thorughout the structure volume, is biologically iso-effective with the heterogeneously distributed dose represented by the DVH (3). The calculation depends upon a user-specified tissue-specific parameter (`a`), which can account for behavior of both tumors and normal tissues. This parameter should be negative for target structures (e.g. tumor) and positive for organs at risk. For `a = 1`, the gEUD is equivalent to mean dose, while for `a = Inf` and `a = -Inf`, the gEUD is equivalent to maximum and minimum doses, respectively.

```
> gEUD(janedoe[1:3], 6:8)

      LIVER LEFT_KIDNEY     STOMACH
   2538.873    1997.308    2842.618

> gEUD(janedoe[1:3], 1) == unlist(janedoe[1:3]$"Dmean")

      LIVER LEFT_KIDNEY     STOMACH
       TRUE        TRUE        TRUE

> gEUD(janedoe[1:3], Inf) == unlist(janedoe[1:3]$"Dmax")

      LIVER LEFT_KIDNEY     STOMACH
       TRUE        TRUE        TRUE

> gEUD(janedoe[1:3], -Inf) == unlist(janedoe[1:3]$"Dmin")

      LIVER LEFT_KIDNEY     STOMACH
       TRUE        TRUE        TRUE
```

3.6 Linear quadratic extrapolated (LQE) dose conversion

Linear quadratic extrapolation (LQE) has been used to model iso-effective doses (4) and is implemented herein for conversion of and comparison among one or more `DVH` or `DVH.list` objects to account for varying dose fractionation. Two identical DVH curves may in fact represent two widely divergent treatments if, for instance, one patient is treated to 6000cGy in four 1500cGy fractions whereas another is treated to 6000cGy in thirty 200cGy fractions. In the case below, the `LQE()` function is used to determine the isoeffective doses for 4500, 5500, and 6000cGy when delivered in 30 instead of 20 daily fractions (20x 300cGy is approximately iso-effective with 30x 245cGy, assuming an alpha/Beta ratio of 3 [`aB = 3`]):

```
> LQE(c(4500, 5500, 6000), aB=3, fractions=c(20, 30))

[1] 4363.636 5333.333 5818.182
```

Note that the `LQE()` function may also be applied to `DVH` or `DVH.list` objects, wherein doses are converted according to specified fractionation and accounting for user-specified alpha/Beta ratio.

3.7 DVH plotting

Individual DVH plots can be generated by the `plot()` function, and may be altered to show dose and/or volume as relative or absolute values with DVH shown as cumulative or differential data.

```
> plot(janedoe[[3]], volume="relative", dose="absolute", type="cumulative")
```

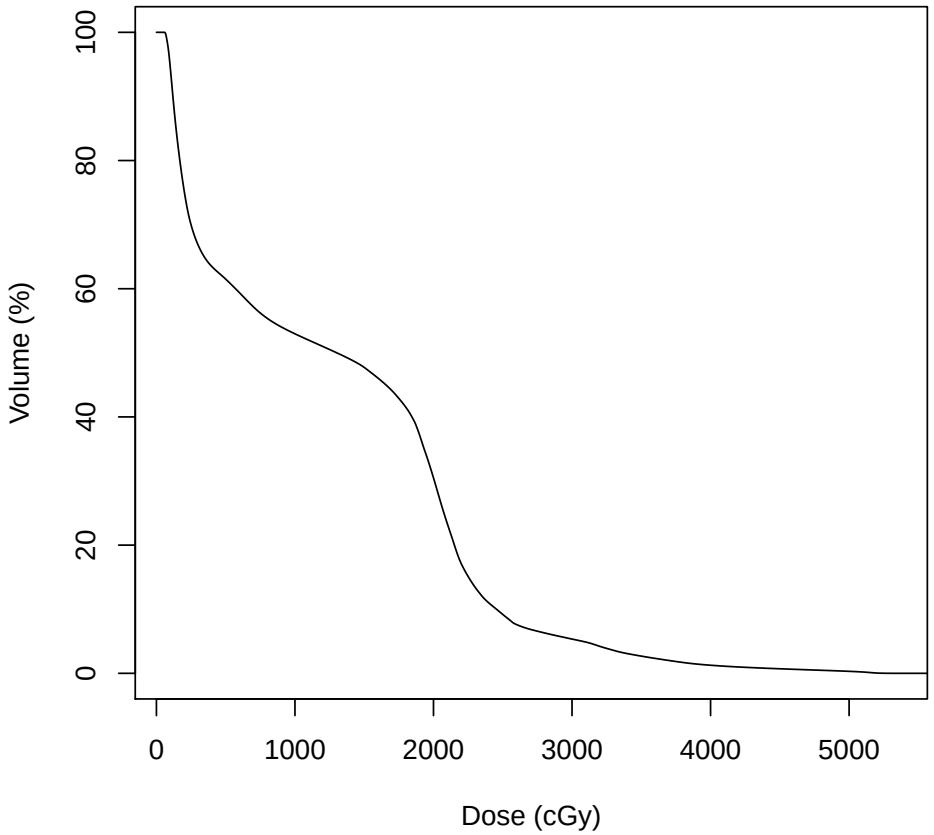

Figure 2: Standard dose-volume histogram for a single structure ("STOMACH") from patient Jane Doe. Data is shown as cumulative dose versus volume.

```
> plot(janedoe[1:3], plot.type="i", col=c("red", "green", "blue"),
+ legend="topright", legend.labels=names(janedoe[1:3]))
```

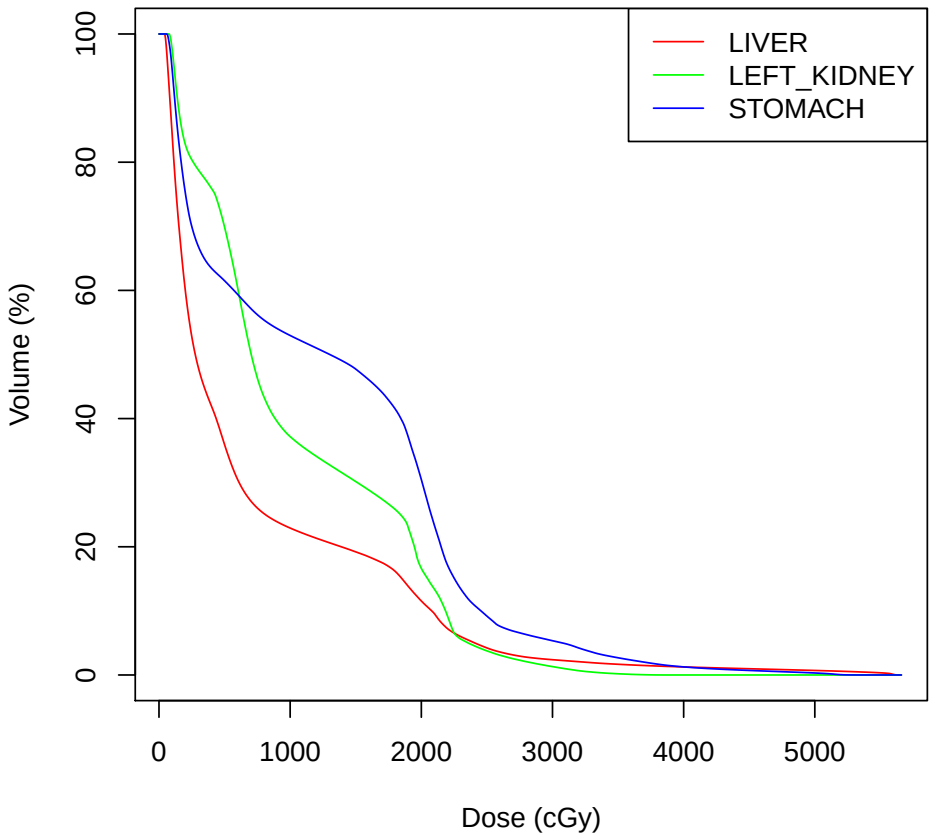

Figure 3: Standard dose-volume histogram for three structures from a single patient, Jane Doe. Data is shown as cumulative dose versus volume. Legend is displayed in the top right corner of the plot.

```
> plot(c(johndoe["STOMACH"],janedoe["STOMACH"]), #group 1
+ c(janedoe["LIVER"],johndoe["LIVER"]), #group 2
+ c(johndoe["DUODENUM"],janedoe["DUODENUM"]), #group 3
+ plot.type="g", dose="relative", col=c("blue", "red", "green"),
+ lwd=2, lty="dashed", fill.lty="solid", fill.transparency=0.3)
```

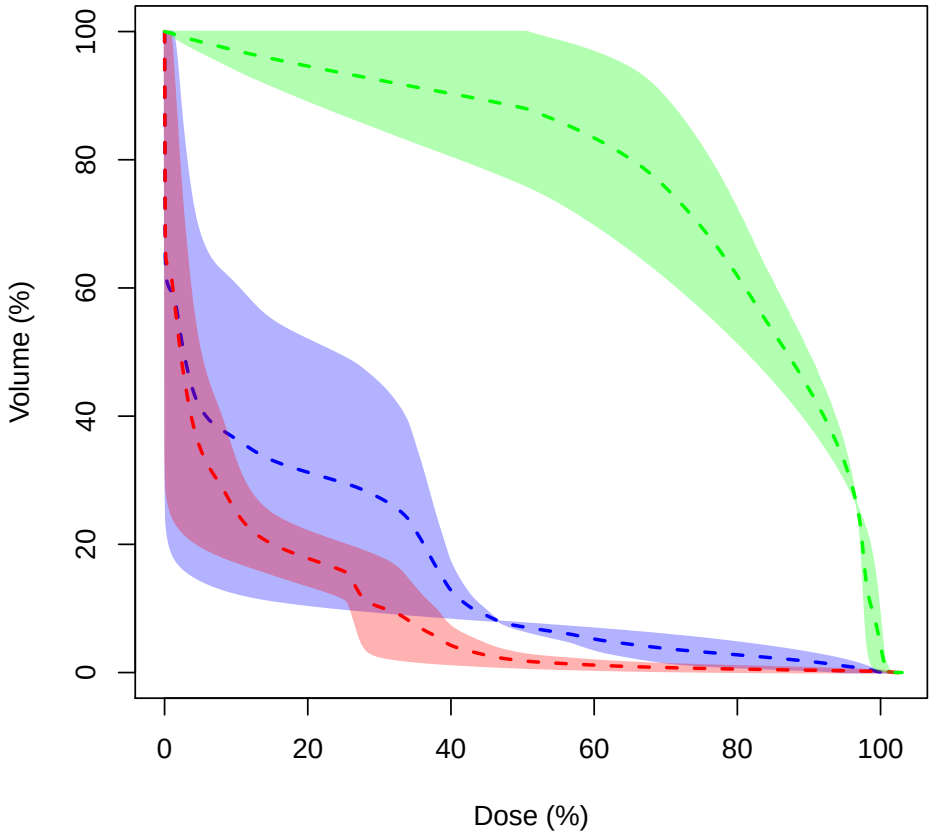

Figure 4: Mean dose-volume histograms are shown for three groups of DVHs, in this case corresponding to stomach, liver, and duodenum from two different patients (John Doe and Jane Doe). Data is shown as cumulative dose (relative) versus volume (relative). Shading represents the range of the data for each group (note that the width of the shading can be specified to represent other parameters instead of range – e.g. variance, standard deviation, interquartile range, median absolute deviation).

```
> group1 <- c("CTV", "PTV")
> group2 <- c("LIVER", "STOMACH", "SMALL_BOWEL")
> plot(c(johndoe[group1],janedoe[group1]),
+ c(janedoe[group2],johndoe[group2]),
+ plot.type="t", main="Target v. OAR t-Test", alpha=0.001,
+ col=c("red", "blue"), lty="dashed", fill.lty="solid")
```

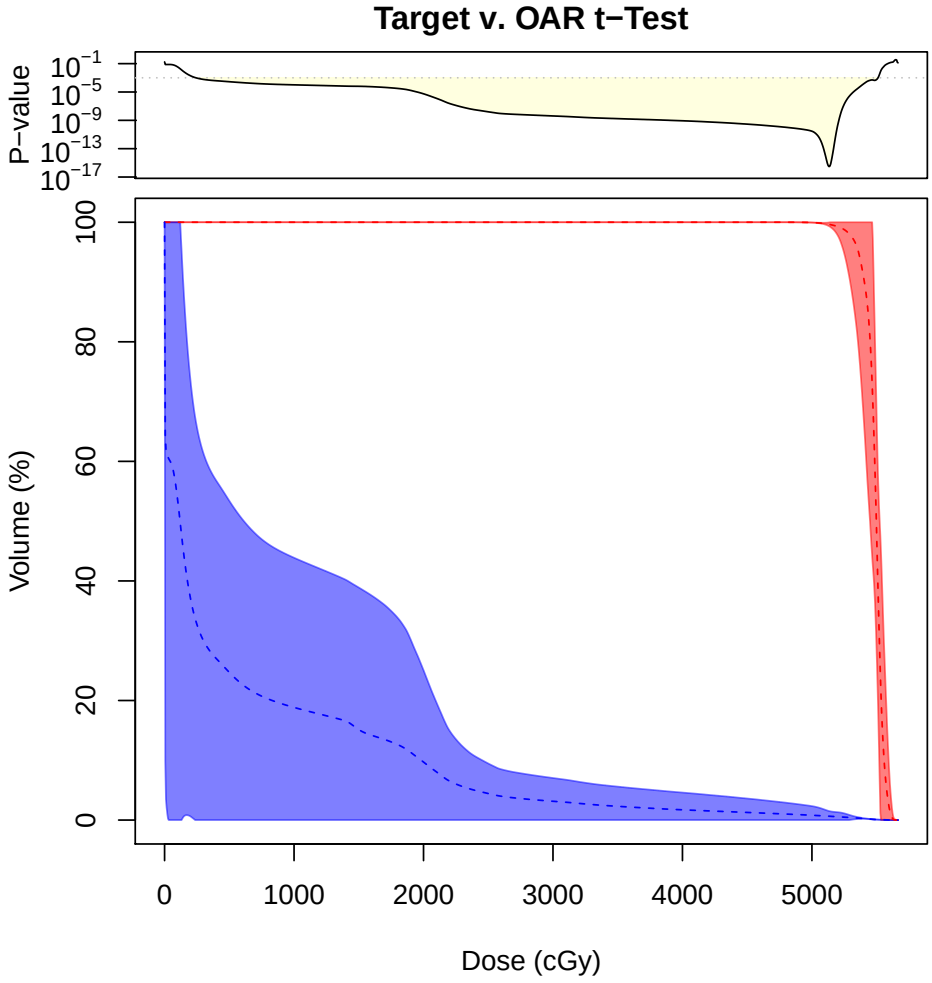

Figure 5: Mean dose-volume histograms are shown for two groups of DVHs, in this case corresponding to CTV/PTV and liver/stomach/small bowel from two different patients (John Doe and Jane Doe). Data is shown as cumulative dose (absolute) versus volume (relative). Shading represents the 99.9% confidence interval for each group (specified here by `alpha=0.001`). The corresponding p-values are shown in the upper panel, with corresponding significance threshold $p<0.001$.

```
> plot(janedoe[2:9], plot.type="b", volume="abs", dose="rel")
```

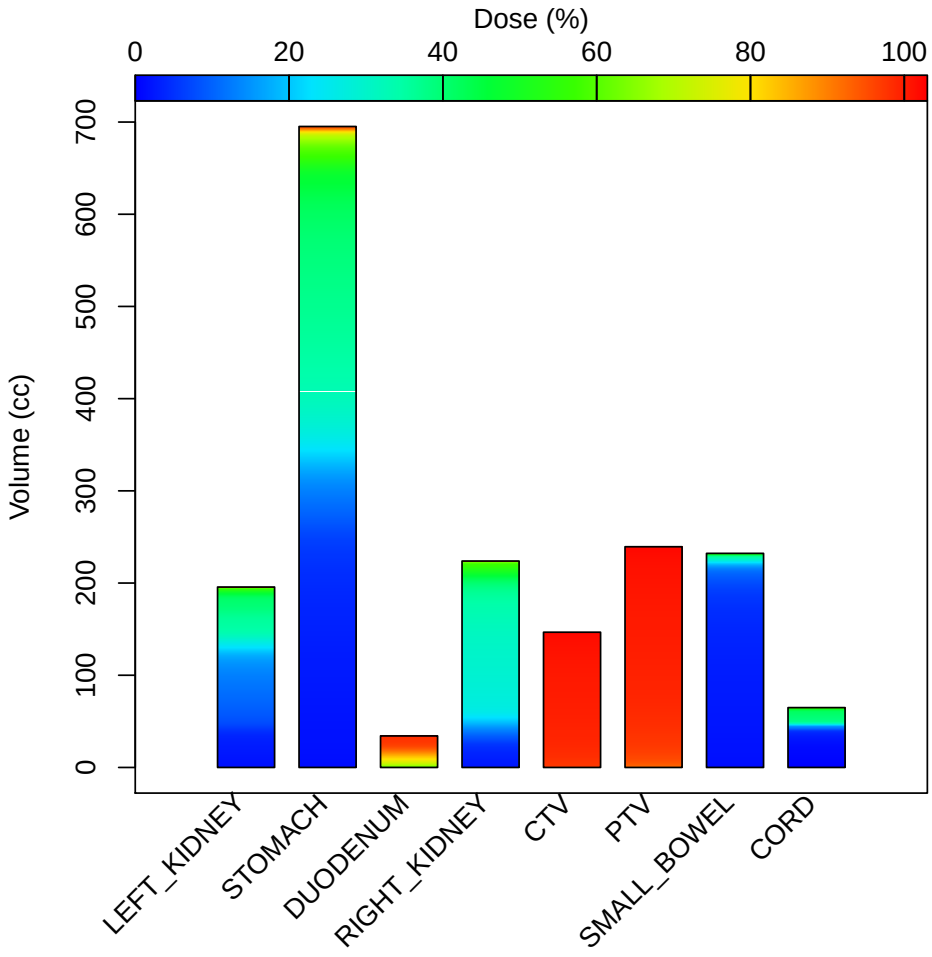

Figure 6: Bar representation of dose distributions for eight structures from a single patient (Jane Doe).

3.8 DVH statistics

Mean or median DVHs can be calculated using the `mean()` and `median()` functions, respectively. These functions take a DVH list as input and return a single object of class `DVH` representing the mean or median dose-volume histogram data calculated from the entire group.

```
> plot(janedoe)
> plot(median(janedoe), new=FALSE, col="red", lwd=2)
> plot(mean(janedoe), new=FALSE, col="blue", lwd=2, lty="dashed")
```

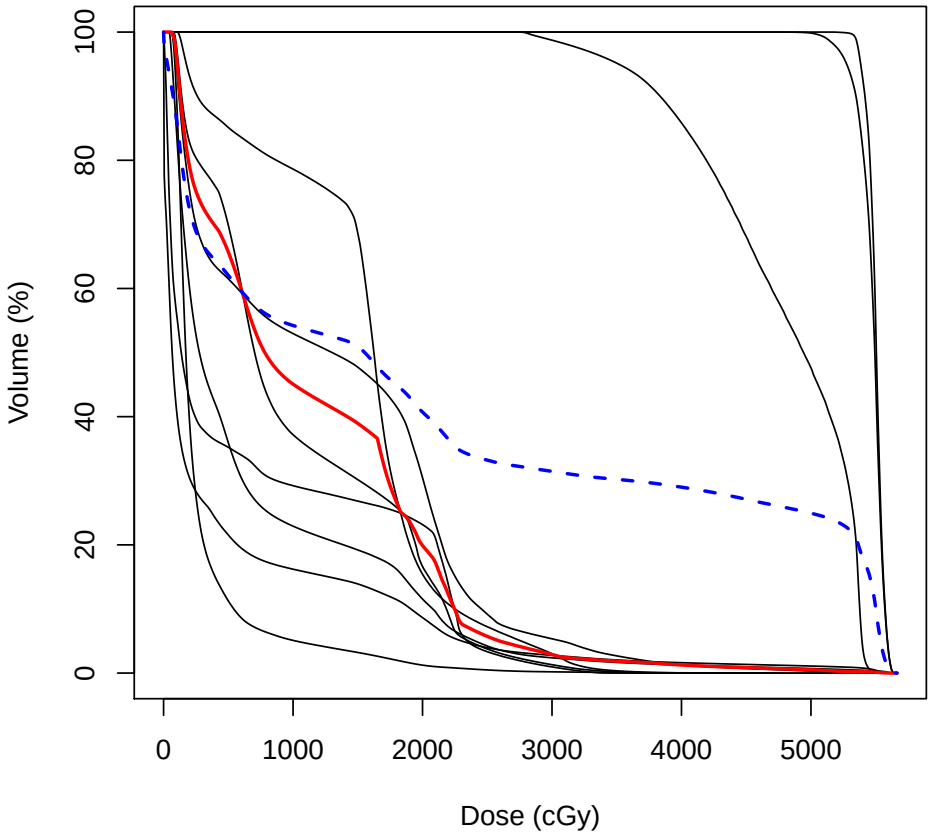

Figure 7: Mean and median DVHs are shown in blue dash and red, respectively.

Similarly, DVH list summation can be performed for two or more *non-overlapping* structures (e.g. bilateral lungs or kidneys) using the `sum()` function.

```
> L.kidney <- janedoe[["LEFT_KIDNEY"]]
> R.kidney <- janedoe[["RIGHT_KIDNEY"]]
> total.kidney <- sum(janedoe[c("LEFT_KIDNEY", "RIGHT_KIDNEY")])
> plot(total.kidney, type="diff", volume="abs")
> plot(L.kidney, new=FALSE, type="diff", volume="abs", col="red")
> plot(R.kidney, new=FALSE, type="diff", volume="abs", col="blue")
```

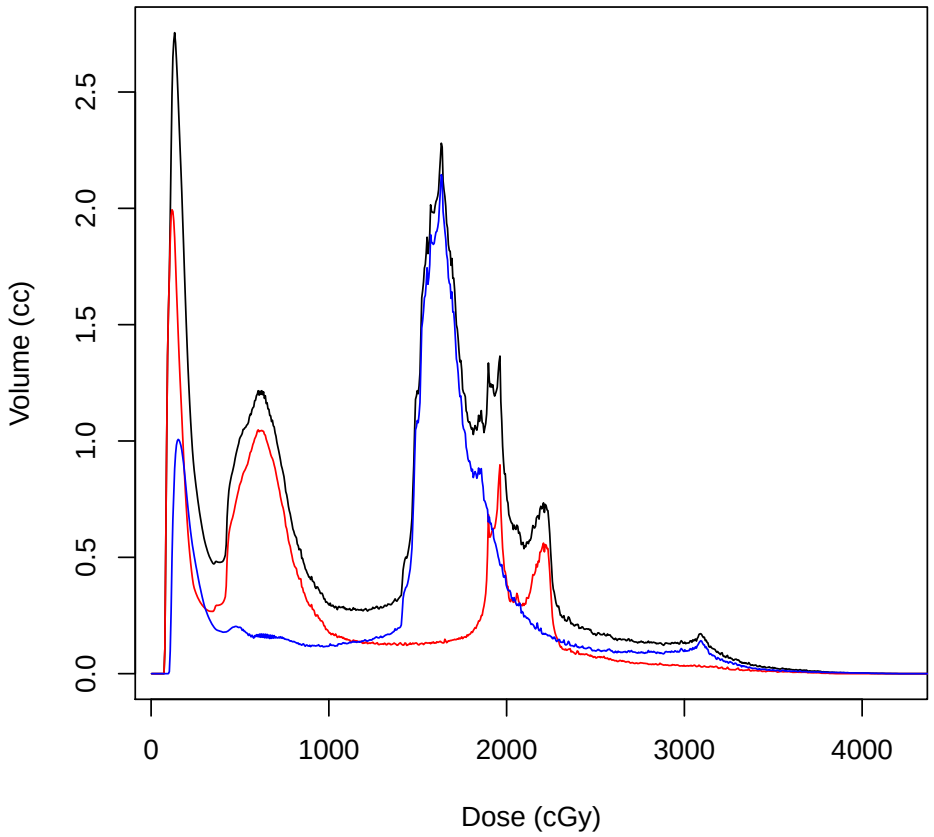

Figure 8: Plot of total kidney dose, calculated using `sum()` function. Left and right kidney doses are shown in red and blue, respectively.

In routine clinical practice and research, DVH comparisons are often performed at an individual parameter level (e.g. V20Gy from Group A compared to Group B). The *RadOnc* package enables automated comparison throughout the entire DVH. Functions such as `t.test()` and `wilcox.test()` are both enabled for DVH lists.

```
> groupA <- janedoe[c("LIVER","LEFT_KIDNEY","RIGHT_KIDNEY","CORD")]
> groupB <- janedoe[c("CTV", "PTV")]
> t.test(unlist(groupA$"V20Gy"), unlist(groupB$"V20Gy"))

        Welch Two Sample t-test

data:  unlist(groupA$V20Gy) and unlist(groupB$V20Gy)
t = -34.5156, df = 3, p-value = 5.347e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -90.99543 -75.63187
sample estimates:
mean of x mean of y
 16.68635 100.00000
```

```
> AvB <- t.test(groupA, groupB)
> plot(AvB$dose, AvB$p, type="l", log="y", xlab="Dose (cGy)", ylab="p-value")
```

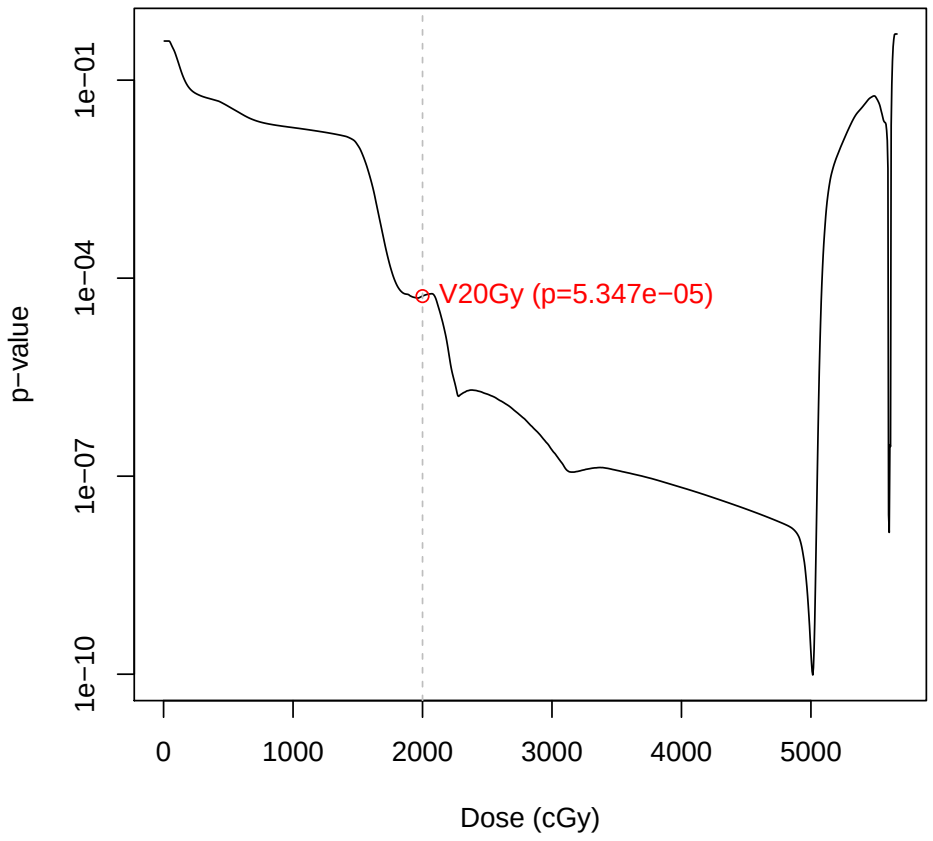

Figure 9: p-values from `t.test()` comparison as a function of dose. V20Gy is highlighted and its p-value corresponds closely to values generated from t-test of V20Gy directly.

4 Three-Dimensional Structure Analysis

4.1 DICOM-RT import

The `read.DICOM.RT()` function is designed to take input DICOM-RT file(s) and output a list of `structure3D` data objects containing all relevant data, in particular the axially-defined contours delineating each structure. Note that DICOM-RT file import was evaluated using Varian's Aria/Eclipse platform (v.10 and v.11). Other treatment planning systems may encode 3D structural information in a different format and this has not been evaluated in the current release of software.

For DICOM-RT data, the associated CT scan must be exported directly from the treatment planning system and should include all contoured structures of interest. In Eclipse, this can be accomplished via the "Export Wizard..." option in the "File" menu, accessed in either Countouring or Plan Evaluation modes. Note that the "Include structure set" option should be selected, and that Institution-specific filters will be required for proper data export. DICOM-RT data will consist of multiple files representing both the CT image as well as the relevant structure set(s).

DICOM-RT data may be imported using the `read.DICOM.RT()` function, with a mockup example shown here (note that "DICOM directory" should be replaced by the path to a specific directory containing the desired DICOM data):

```
> data <- read.DICOM.RT(path="<<DICOM directory>>", verbose=TRUE)
```

The DICOM-RT import process may take some time. We have included pre-loaded data for a single patient (included structures: spinal cord, mandible, teeth) which will be explored in this vignette.

4.2 3D structure manipulation

The `read.DICOM.RT()` function returns a `structure.list` object that can be manipulated in multiple ways. Subsets of structure lists can be obtained using the `[]` modifier, and any number of structure lists can be combined using the `c()` function. Additionally, single `structure3D` objects can be directly accessed using the `[[]]` modifier, and individual elements of a structure list may be directly replaced with other `structure3D` objects using the `[[<-` function.

```
> teeth[1:2]

[1] "List containing 2 structure3D objects (Tooth #1, Tooth #2)"

> c(cord, mandible)

[1] "List containing 2 structure3D objects (Spinal Cord, Mandible)"

> teeth[[1]]

[1] "Structure (Tooth #1) defined by 324 points in 23 axial slices"

> teeth[[1]] <- teeth[["Tooth #3"]]
> teeth

[1] "List containing 3 structure3D objects (Tooth #3, Tooth #2, Tooth #3)"
```

As with `DVH.list` objects, `structure.list` objects may be subsetted by pattern matching:

```
> teeth["Tooth.*"]

[1] "List containing 3 structure3D objects (Tooth #1, Tooth #2, Tooth #3)"
```

Other list processing functions may be applied to structure lists, enabling further data manipulation. The `rev()` function may be used to reverse the order of a structure list, while the `names()` function may be used to extract (or set) the structure names for each structure contained within the list. The `length()` function may be used to find the number of structures contained within a structure list, and the `lapply()` function can be used to perform a customizable set of operations on a structure list and return a customizable set of values. Here are some examples employing each of these functions:

```
> names(teeth) <- c("Larry", "Curly", "Moe")
> names(rev(teeth[1:3]))

[1] "Moe"   "Curly" "Larry"

> length(teeth)

[1] 3
```

```
> lapply(teeth, function(tooth) { range(tooth) })

$Larry
          x       y    z
min -31.71 -136.93 -111
max -16.82 -122.58  -90

$Curly
          x       y    z
min -31.43 -136.77 -111
max -16.27 -122.43  -90

$Moe
          x       y    z
min -31.28 -136.58 -111
max -16.49 -122.51  -90
```

4.3 Plotting 3D structures

Three dimensional surfaces renderings can be generated by the `plot()` function. The *RadOnc* package does not currently contain the functionality to generate surface triangulations for a given structure, however future releases of the package will implement surface triangulation. Thus, data imported using `read.DICOM.RT()` will not currently be processed for surface triangulation and will generate an empty plot if plotting is attempted. External applications such as MeshLab (5) can be used to generate triangulations which, for advanced users, can be imported into a given `structure3D` object.

```
> plot(mandible)
```

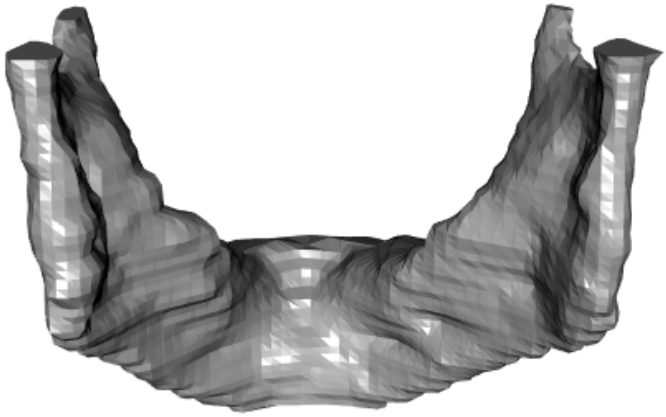

Figure 10: Three-dimensional surface reconstruction from triangulation of physician-contoured points for the mandible of a patient.

```
> plot(cord)
```

Figure 11: Three-dimensional surface reconstruction from triangulation of physician-contoured points for the spinal cord of a patient.

4.4 Structure comparison

Comparison of three-dimensional structures has numerous applications. In the case presented here, three physicians separately delineated a tooth on axial slices of a CT for a single patient. Variability among physician contours is demonstrated using the `compareStructures()` function:

```
> compareStructures(teeth, method="axial", plot=TRUE)
```

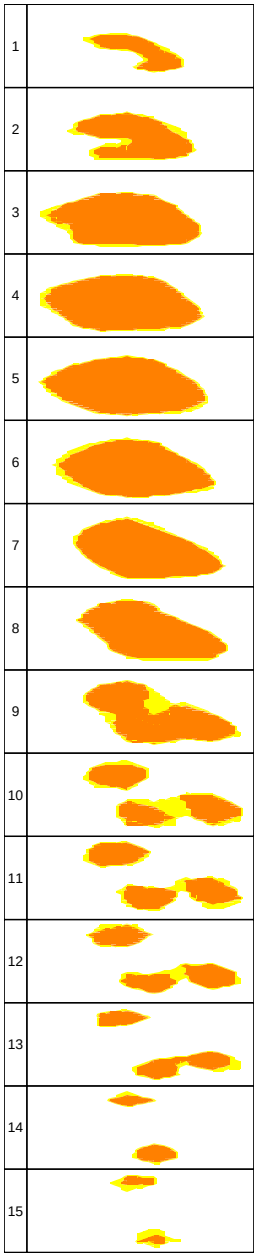

Figure 12: Axial comparison of overlap among three separate physician-defined contours for a single tooth. Red (and white) regions delineate consensus while decreasing degree of overlap is shown in decreasing shades of orange and yellow.

Structure comparison may also be performed by Hausdorff distance (6), which computes the distance between two points clouds, in this case structure surfaces. The absolute Hausdorff distance (`hausdorff.method="absolute"`) yields the maximum distance required to connect any point from one point cloud to its closest neighbor in the other. This metric is highly subject to outliers, thus an aggregate metric is implemented by selecting the average distance (`hausdorff.method="mean"` or `hausdorff.method="median"`) required to connect all points in one point cloud to their closest neighboring points in the other. Note that the Hausdorff distance between two completely superimposable point clouds is zero.

```
> compareStructures(teeth, method="hausdorff", hausdorff.method="mean")

Analyzing structure 1/2 (Tooth #1) ... FINISHED
Analyzing structure 2/2 (Tooth #3) ... FINISHED
           Tooth #1  Tooth #3
Tooth #1 0.0000000 0.4847732
Tooth #3 0.4847732 0.0000000
```

References

[1] R.E. Drzymala, R. Mohan, L. Brewster, J. Chu, M. Goitein, W. Harms, and M. Urie. Dose-volume histograms. *Int J Radiat Oncol Biol Phys*, 21(1):71–78, 1991.

[2] W. Straube, J. Matthews, W. Bosch, and J.A. Purdy. Dvh analysis: consequences for quality assurance of multi-institutional clinical trials. *Med Phys*, 32(6):2021, 2005.

[3] C. Thieke, T. Bortfeld, A. Niemierko, and S. Nill. From physical dose constraints to equivalent uniform dose constraints in inverse radiotherapy planning. *Med Phys*, 30(9): 2332–2339, 2003.

[4] G.W. Barendsen. Dose fractionation, dose-rate and iso-effect relationships for normal-tissue response. *Int J Radiat Oncol Biol Phys*, 8(11):1981–1997, 1982.

[5] Open Source actively supported by the 3D-CoForm project. Meshlab. 2005. URL `http://meshlab.sourceforge.net`.

[6] Felix Hausdorff. *Grundzuge der Mengenlehre*. Veit and Company, Leipzig, 1914.

A Acknowledgements

Many thanks to Brandon Whitcher for his phenomenal and generous support regarding the *oro.dicom* pacakge, and to Daniel Wollschlaeger for his insights and contributions regarding DVH file formatting, dose unit interconversion, and inclusion of CadPlan data.

B Previous Release Notes

- Initial release (v1.0.0) on 2013-07-07
- Subsequent release (v1.0.1) on 2013-10-23
- Subsequent release (v1.0.2) on 2013-11-25
- Subsequent release (v1.0.3) on 2014-01-17
- Subsequent release (v1.0.4) on 2014-01-31
- Last release (v1.0.5) on 2014-02-19

Please refer to NEWS.Rd file for more details.

```
> news(package="RadOnc")
```