

Getting started with espadon

Table of content

1	Introduction. What we will learn?	3
2	Objects representation and handling	3
2.1	2D representations	3
2.2	Binary masks	4
2.3	3D representations	5
3	Basic dosimetry studies	6
3.1	Dose-Volume Histograms	7
3.2	Depth-dose	8
3.3	Conformity indexes.....	9
4	Miscellaneous.....	10
4.1	DICOM files automatic pseudonymisation	10
4.2	Raw DICOM content access.....	10
4.3	Basic rtstruct features and ROI naming.....	11
5	What we learned	13

1 Introduction. What we will learn?

In this document, we will shortly illustrate espadon main functionalities. In order to play with the different examples, you should have at hand 1 rt-Dose, 1 CT and 1 rt-Struct. Files should be in a dedicated folder, for instance, in our case "E:\\dicom data\\patient001", or "E:/dicom data/patient001". Then, execute the following lines:

```
rm (list=ls ())
require (espadon)

pat.dir <- "E:\\dicom data\\patient001"

pat <- load.patient.from.dicom (pat.dir, data = TRUE)
View (pat)

CT <- pat$ct[[1]]
D <- pat$rtdose[[1]]
S <- pat$rtstruct[[1]]
```

Note: espadon handles native DICOM files and tries to make what we called a Uniform General View (UGV) of the documents. That is what we get executing `View (pat)`. In fact espadon has got its own storage format which is faster and takes less disk space, which is useful for big studies. If you are interested in space/time saving, have a look at the "data loading" document.

The first section will be devoted to objects representation and handling in espadon. The second section will introduce basic instructions used to perform dosimetry studies. Then the last section will explore miscellaneous possibilities of espadon.

2 Objects representation and handling

From the simplest 2D representation up to powerful 3D representation, espadon package allows you to get quick and easy results.

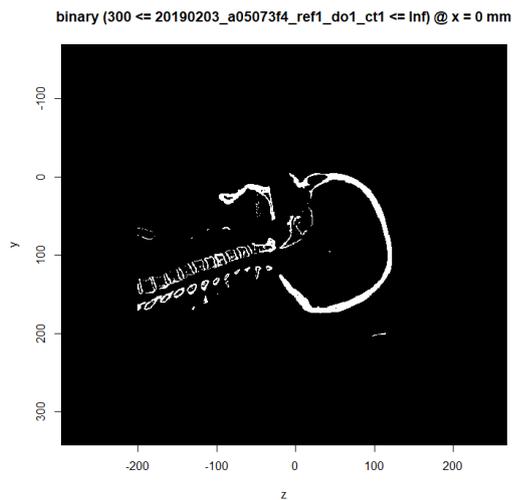
In order to know how to represent the objects, espadon has defined a class (enumerated by the `espadon.class ()` function) for each of them:

- There is for example the "volume" class, which contains all the objects that define a 3D volume with voxels, such as CT, MR, rt-Dose modalities.
- The "struct" class, contains among others, all the contour data, and is assigned to the rt-Struct modality objects
- There are also the classes "dvh", "histo2D", "mesh", "reg", "t.mat", and finally the class "undef", for objects not yet defined.

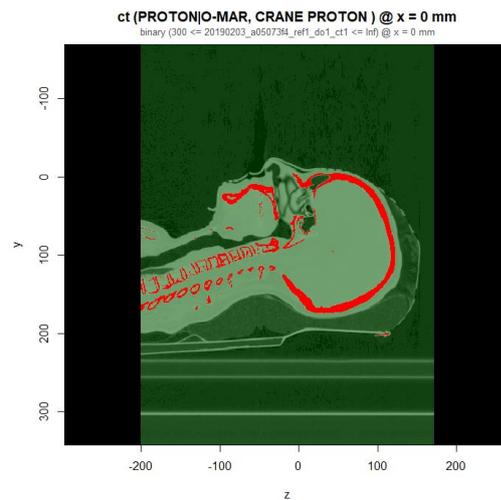
2.1 2D representations

For instance, to get the CT, with an overlay of dose and the structures, just do:

```
dev.new (width = 10, height = 10, noRStudioGD = T)
par (mar = c (4, 4, 4, 5))
display.plane (bottom = CT, top = D, struct = S)
```

Binary selection of CT > 300 HU



Binary selection added on top of the CT

The selection is operated using the `bin.from.vol` function. The first call to `display.plane` illustrates the binary selection (in sagittal view). The second one represents this selection on top of the CT. Value at FALSE are represented in transparent green, and TRUE are in opaque red, using the option `top.col`.

We may also want the selection be operated on one or more `rt-Struct` RoI, and maybe make computation with this selection. For instance, we could make a “negative expansion” for the production of the “skin” selection of the patient this way:

```
B.pat <- bin.from.roi (CT, S, roi.name = "patient")
B.out.pat <- bin.inversion (B.pat)
B.out.pat.3mm <- bin.dilation (B.out.pat, radius = 3)
B.skin1 <- bin.intersection (B.pat, B.out.pat.3mm)

S$roi.info$color[select.names(S$roi.info$roi.pseudo, roi.name = "patient")] <- "green"

display.plane (B.pat, view.type = "sagi", struct = S, roi.name = "patient")           # image 1
display.plane (B.out.pat, view.type = "sagi", struct = S, roi.name = "patient")       # image 2
display.plane (B.out.pat.3mm, view.type = "sagi", struct = S, roi.name = "patient")  # image 3
display.plane (B.skin1, view.type = "sagi")                                         # image 4 = "skin"
```

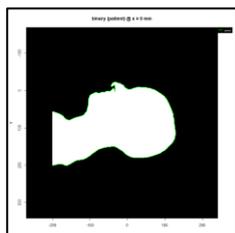


Image 1

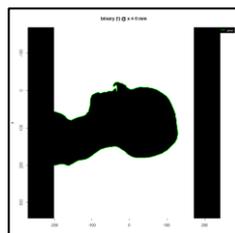


Image 2

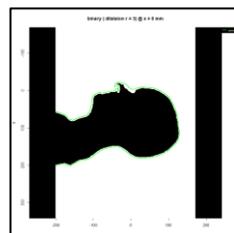


Image 3

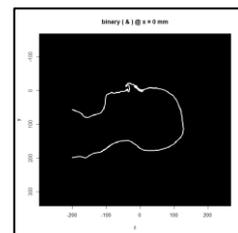


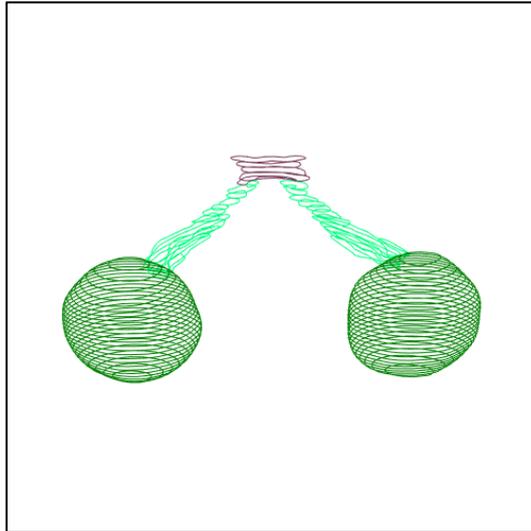
Image 4

2.3 3D representations

There are several ways to display objects in 3D in espadon. RoI of `rt-Structs` can be directly displayed in the form of planes contours. Binary objects offer more possibilities when converted into meshes. For this purpose, espadon uses `rgl` and `rvcg` packages where you can find powerful possibilities for your work. If you want to use them (as in this example), you should download them from CRAN.

Here a simple example of 3D representation:

```
require (rgl)
par3d (windowRect = c (100, 100, 600, 600)) ## rgl instruction
display.3D.contour(S, roi.sname = c ("oeil", "no", "chias"))
```

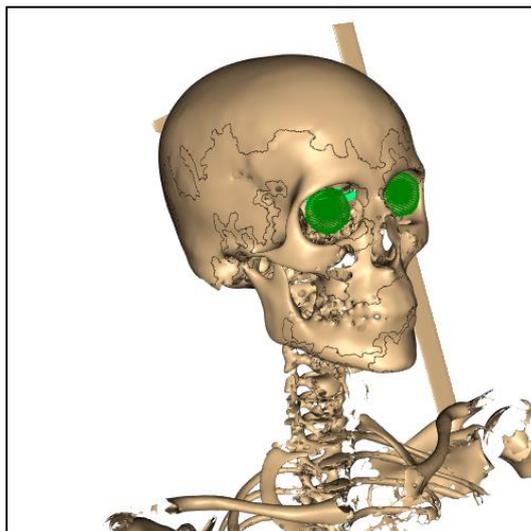


Display of Roi eyes, optical nerves, and chiasma

The `display.3D.contour` instruction uses the structures stored in `S`. Here we decided to display both (left and right) eyes (“oeil” in French), both optical nerves (“nerfs optiques” in French abbreviated as “no”) and chiasma.

```
bg3d ("white") ## another rgl instruction
B.skel <- bin.from.vol (CT, min = 300)
M.skel <- mesh.from.bin (B.skel, smooth.iteration = 10, verbose = TRUE)
display.3D.mesh (M.skel)

rgl.snapshot ("dummy.png") ## rgl way to store images in a file
```



Mesh of skeleton

3 Basic dosimetry studies

As espadon is primary a tool for studying dose effects on tumor and healthy tissue, it contains lot of functions for handling dose. As dose is basically an imaging object of the volume class, all these instructions also work for CT, MR, and PET.

3.1 Dose-Volume Histograms

Dose-Volume Histograms (DVH) are probably the main tool for radio-physicists. TPS manufacturers have plenty of ways to compute and store them, which is not practical in terms of homogenization and plan comparison. In espadon, you have the choice to use the DVH stored in the rt-Dose, or to compute them from scratch (and use espadon functionalities, such as Monte-Carlo).

For instance, let us check what contours we have at hand:

```
S$roi.info$roi.pseudo

 [1] "patient"      "encephale"      "troncint"      "oeild"         "oeilg"
 [6] "cristald"    "cristalg"      "nod"          "nog"          "chiasma"
[11] "gtvck16en8"  "ctv50.4gyrbe"  "hypophyse"    "bandeletteoptd" "bandeletteoptg"
[16] "cerveletpost" "cochleed"      "cochleeg"    "conduitintd"  "conduitintg"
[21] "hippocamped" "hippocampeg"   "lacrymald"    "lacrymalg"    "ncviid"
[26] "ncviig"      "ncviid"        "ncviig"      "cerveletant"  "peau3mm"
[31] "troncext"    "atmd"          "atmg"        "corneed"      "corneeg"
[36] "retined"     "retineg"       "parotided"   "parotideg"    "moelle"
[41] "external"    "picotsupd"     "picotsupg"   "picotd"       "picotg"
[46] "ptv"         "patient-ptv"   "voiesoptiques" "oreilled"     "oreilleg"
[51] "tronc"      "macula45"

length (select.names (S$roi.info$roi.pseudo, roi.sname = "ctv"))

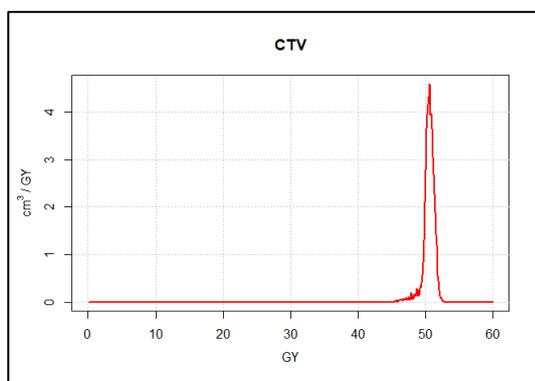
[1] 1
```

We have exactly one ROI whose name contains "ctv". Let us display its (differential) dose histogram, and (cumulative) DVH:

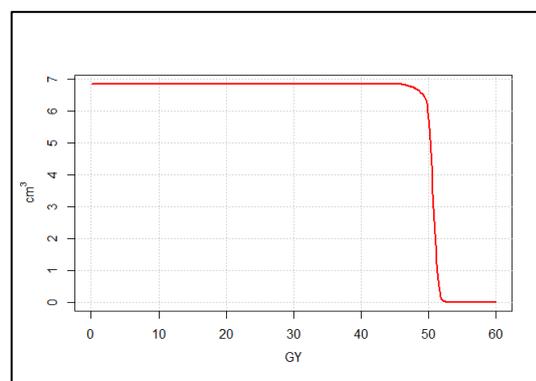
```
dev.new (width = 7, height = 5, noRStudioGD = T)

H <- histo.from.roi (D, S, roi.sname = "ctv", breaks=seq (0, 60, by=0.1))
display.dV_dx (H, lwd = 2, xlim = c (0, 60), col = "#ff0000", main="CTV")

dev.new (width = 7, height = 5, noRStudioGD = T)
DVH <- histo.DVH (H)
display.DVH (DVH, lwd = 2, xlim = c (0, 60), col = "#ff0000")
```



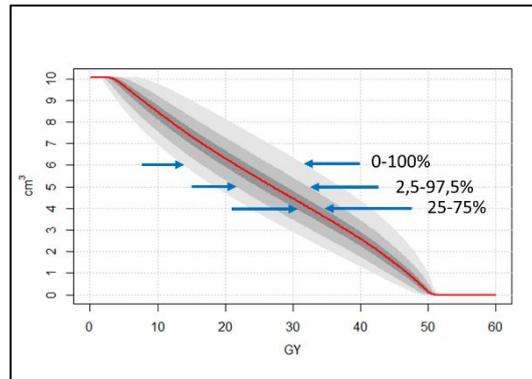
Differential dose histogram in CTV



DVH in CTV

In addition, espadon has a Monte-Carlo (MC) functionality on histograms. It can simulate “organ movements” (strictly speaking, contour movements) by applying a random normal shift in every (or user defined) direction. This MC can be time consuming since, for each shift, a binary volume is reconstructed in order to get the dose. It is strongly recommended to restrict the volume of interest to the organ volume, augmented by a margin. For instance:

```
roi.D <- nesting.roi (D, S, roi.name = "oeild", xyz.margin = c(10, 10, 10))
H <- histo.from.roi(roi.D, S, roi.sname = "oeild", breaks=seq (0, 60, by=0.1), MC=500)
DVH <- histo.DVH (H)
display.DVH (DVH, lwd = 2, xlim = c (0, 60), MC.plot=TRUE, col = "#ff0000")
```



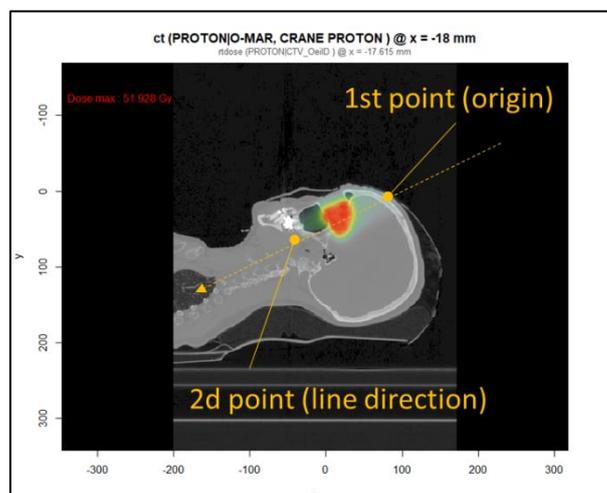
DVH in right eye, with simulation of 1 mm organ movements

The grey colors area correspond to the given fraction of the MC simulations. A black line (not visible here as it is under the reference DVH in red) is plotted for the median of the MC.

3.2 Depth-dose

Espadon has several technics for getting and displaying dose. One of the most simple to understand is depth dose (which can be used for tissue along the same axis, for instance). In the example below, we plot depth dose along a user defined axis in the sagittal plane including CTV gravity center:

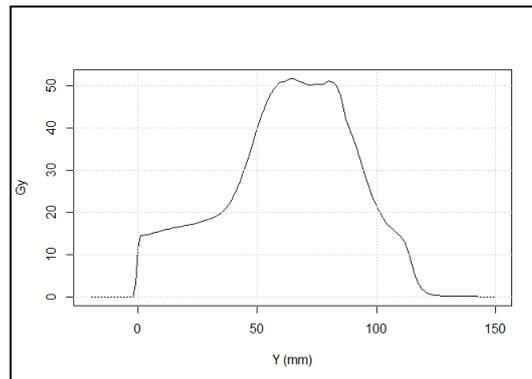
```
idx <- select.names(S$roi.info$name, roi.sname = "ctv")
CTV.CoG <- as.numeric (S$roi.info[idx, c("Gx", "Gy", "Gz")])
display.plane(bottom=CT, top=D, view.type = "sagi", view.coord = CTV.CoG[1])
M <- locator (n=2) # select two points on the image (1st = origin; 2d = direction)
```



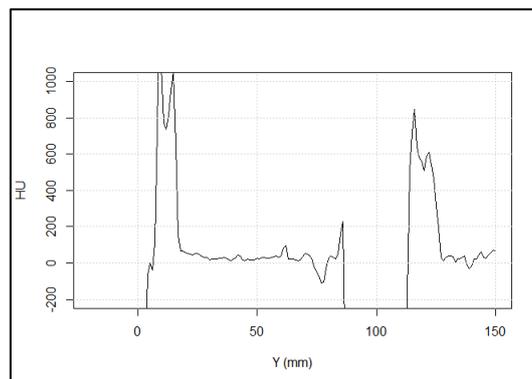
CT at CTV gravity center

Then, we can get the depth dose and the Hounsfield units of the CT (for instance):

```
Depth.dose <- get.line (D, origin = c (CTV.CoG[1], M$y[1], M$x[1]),
                        orientation = c(0, M$y[2] - M$y[1], M$x[2] - M$x[1]),
                        grid=seq (-20, 150, by=1), interpolate = TRUE)
plot (Depth.dose$s, Depth.dose$value, type="l", xlab="Y (mm)", ylab="Gy")
grid ()
HU.ct <- get.line (CT, origin = c (CTV.CoG[1], M$y[1], M$x[1]),
                  orientation = c(0, M$y[2] - M$y[1], M$x[2] - M$x[1]),
                  grid=seq (-20, 150, by=1), interpolate = TRUE)
plot (HU.ct$s, HU.ct$value, type="l", xlab="Y (mm)", ylab="HU", ylim=c(-200, 1000))
grid ()
```



Depth Dose



Hounsfield units of the CT

3.3 Conformity indexes

The espadon package directly computes the standard conformity indices. They can be retrieved using a single ROI, or a selection with a binary volume. **The meaning and interpretation of these indicators depend on the nature of the tissue and care should be taken as espadon compute them without knowing if they are relevant or not.**

The functions `rt.indices.from.bin` and `rt.indices.from.roi`, compute standard radiotherapy indices:

```
rt.indices.from.roi(vol=D, struct=S, target.roi.sname = "ptv", healthy.roi.sname = NULL, presc.dose = 50)
```

```
processing [=====] 100%
$dosimetry
```

```
  D.min  D.max  D.mean  STD
```

```

PTV          40.065 74.744 67.714692 3.869503
$volume
      V_tot      area V_50Gy
isodose      NA      NA 277.784
PTV          135.731 157.9252 135.653

$conformity
  presc.dose      target      PITV      PDS CI.lomax2003      CN      NCI      DSC
1           50      PTV 2.04657742 2.047754 0.99942533 0.488059229 2.048932 0.65609712
      CDI      CS3      ULF
1 0.7335748 0.99771607 0.0005746661

$homogeneity
  presc.dose      target HI.RTOG.max_ref HI.RTOG.5_95 HI.ICRU.max_min HI.ICRU.2.98_ref
1           50      PTV 1.49488 1.201004 1.865568 32.250
HI.ICRU.2.98_50 HI.ICRU.5.95_ref HI.mayo2010 HI.heufelder
1 23.36787 23.940 1.269082 0.9986857

$gradient
  presc.dose      target GI.ratio.50      mGI
1           50      PTV 2.295924 4.701488

```

There are many radiotherapy indices and this field is in constant evolution. Only few of them are computed in this example. For an up-to-date view of what is integrated in espadon (definitions, papers, ...) please have a look at the “dosimetry indicators” vignette.

4 Miscellaneous

4.1 DICOM files automatic pseudonymisation

Espadon has a pseudonymisation functionality that can be useful when original DICOM files have been automatically extracted (and are not anonymized). The job is performed using the `dicom.raw.data.anonymizer` instruction. It is a low level function explained in the vignette dealing with raw DICOM handling.

Note that espadon DICOM pseudonymisation deletes the content of every field that could deal with personal information, but also every field not belonging to the internal DICOM dictionary (as we do not know what they could content). For more information on automatic pseudonymisation, see the example in espadon documentation.

4.2 Raw DICOM content access

Despite espadon works a user friendly way on `rt-Struct`, `rt-Dose`, `CT`, `MR`, `PT`, it is sometime useful to have a look at raw DICOM data, especially for modalities not listed above (as for instance `rt-Plan` or `rt-Image`) or to get specific information not included natively in espadon.

As an example, we will parse a `rt-Plan` file. From `Rdcm` format, you just have to read your file this way:

```

RP <- load.Rdcm.raw.data (file.path(pat.dir, pat$rtplan[[1]]$file.basename))
head (RP$data[[1]])

$` (0002,0000) `
[1] 202

```

```

$`(0002,0001)`
[1] 0 1

$`(0002,0002)`
[1] "1.2.840.10008.5.1.4.1.1.481.8"

$`(0002,0003)`
[1] "1.2.752.243.1.1.20190219160017934.3000.38685"

$`(0002,0010)`
[1] "1.2.840.10008.1.2.1"

$`(0002,0012)`
[1] "1.2.826.0.1.3680043.1.2.100.6.40.0.76"

idx <- grepl ("([300A,0070]) item[:,digit:]+ ([300A,0078])", names (RP$data[[1]]))
RP$data[[1]][idx] # number of fractions

$`(300A,0070) item1 (300A,0078)`
[1] "28"

idx <- grepl ("([300A,0070]) item[:,digit:]+ ([300A,0080])", names (RP$data[[1]]))
RP$data[[1]][idx] # number of beams

$`(300A,0070) item1 (300A,0080)`
[1] "2"

```

Note the use of `load.Rdcm.raw.data` instruction, instead of `load.obj.from.Rdcm` previously explained. `rt-Plan` are not integrated natively in `espadon`, you will have to know exactly what are searching for... For instance:

```

idx <- grepl ("([300A,0070]) item[:,digit:]+ ([300A,0078])", names (RP$data[[1]]))
RP$data[[1]][idx] # number of fractions

$`(300A,0070) item1 (300A,0078)`
[1] "28"

idx <- grepl ("([300A,0070]) item[:,digit:]+ ([300A,0080])", names (RP$data[[1]]))
RP$data[[1]][idx] # number of beams

$`(300A,0070) item1 (300A,0080)`
[1] "2"

```

For more explanations, possibilities and illustrations, do not hesitate to have a look at the appropriate vignette on raw DICOM handling.

4.3 Basic rtstruct features and ROI naming

At loading, useful information is computed on each ROI of a `rt-Struct` file (especially if `data=TRUE`):

```

str (S$roi.info)

'data.frame':  52 obs. of  17 variables:
 $ number      : chr  "1 " "2 " "3 " "4 " ...
 $ name        : chr  "Patient " "Encephale " "TroncInt" "Oeild " ...
 $ description  : chr  "" "" "" "" ...
 $ generation.algorithm: chr  "SEMIAUTOMATIC " "SEMIAUTOMATIC " "SEMIAUTOMATIC " ...
 $ color       : chr  "#ffffff" "#ff8080" "#ffaa00" "#008000" ...
 $ roi.pseudo  : chr  "patient" "encephale" "troncint" "oeild" ...
 $ min.x       : num  -212.2 -66.5 -14.5 -39.5 19 ...
 $ max.x       : num  219.5 66.9 18.3 -11.5 46.5 ...

```

\$ min.y	: num	-22.61 2.13 83.78 3.66 6.91 ...
\$ max.y	: num	217.1 169.3 110.9 30.2 30.9 ...
\$ min.z	: num	-200 -21 -29 -5 -4 -2 -1 14 15 26 ...
\$ max.z	: num	130 116 36 21 22 4 5 26 26 30 ...
\$ vol	: num	7201.52 1389.14 14.74 10.06 9.41 ...
\$ Gx	: num	4.392 0.634 2.496 -25.798 32.611 ...
\$ Gy	: num	109.5 92 98.8 17.1 19.1 ...
\$ Gz	: num	-66.79 51.47 15.01 8.57 9.03 ...
\$ continue	: logi	TRUE TRUE TRUE TRUE TRUE TRUE ...

The name field is the “official” name of the ROI (as red in the DICOM file). As names can contain accented letters (especially in French), more or less spaces and separators, upper and lower cases, espadon uses `roi.pseudo` for selecting the ROI a way explained above. A good practice could consist in keeping the official names unchanged, and renaming their `roi.pseudo`, after loading, once the quality control has been performed. This way, you can be sure you will always find the right ROI pseudonym for the right ROI. For instance, you could decide to rename ROI like “PTV-45Gy”, “PTV primary”, ..., as “myptv” (or any name meaning something for you that cannot be found in other ROI).

The color field is the color extracted from DICOM. You can change it according your own wish (it is especially useful when comparing contours operated by several physicians for contouring challenges).

This data frame also contains min/max x/y/z data representing the box enclosing the full ROI contours. It is useful for restricting time-consuming computations on these ROI. Note that instructions using this possibility allow to increase the box size, if needed. One get also coordinates of the gravity center (Gx/y/z) and the volume expressed in cm^3 .

Note: volume and gravity center are computed according to ROI contours. They can also be computed using binary selections. Despite the two method should give the same result, slight variation can occur (mainly due to space sampling).

The way espadon uses ROI naming is important to understand as it can affect what we are working on. First, as mentioned, ROI selection is operated on `roi.pseudo`, not on name. Espadon internally uses the `select.names` instruction:

```
S$roi.info$roi.pseudo

 [1] "patient"      "encephale"      "troncint"       "oeild"          "oeilg"
 [6] "cristald"     "cristalg"       "nod"            "nog"            "chiasma"
[11] "gtvck16en8"   "ctv50.4gyrbe"   "hypophyse"      "bandeletteoptd" "bandeletteoptg"
[16] "cerveletpost" "cochleed"       "cochleeg"       "conduitintd"    "conduitintg"
[21] "hippocamped" "hippocampeg"    "lacrymald"      "lacrymalg"      "ncviid"
[26] "ncviig"       "ncviid"         "ncviig"         "cerveletant"    "peau3mm"
[31] "troncext"     "atmd"           "atmg"           "corneed"        "corneeg"
[36] "retined"      "retineg"        "parotided"      "parotideg"      "moelle"
[41] "external"     "picotsupd"      "picotsupg"      "picotd"         "picotg"
[46] "ptv"          "patient-ptv"    "voiesoptiques" "oreilled"       "oreilleg"
[51] "tronc"        "macula45"

select.names(S$roi.info$roi.pseudo, roi.name = "patient", roi.sname = "PT", roi.idx = 51:100)

 [1] 1 14 15 46 47 48 51 52
```

This instruction has three ways to select a name:

- 1) By exact name comparison (`roi.name` option) giving “patient” = 1 above.
- 2) By short name comparison (`roi.sname` option) giving 14, 15, 46, 48 as all these ROIs contain “pt” in their pseudo.

- 3) By direct index selection (`roi.idx` option) giving 51 and 52, as we requested 51 to 100 ROIs.

It returns all the values selected by the list of options.

Beware that some espadon instruction manage multiple RoI, like `display.plane`, some need only one selected RoI, like `histo.from.roi`, for instance.

5 What we learned

Espadon was developed for simplifying the usage of DICOM objects, without the need of a Treatment Planning System for instance. It is not a surrogate of the TPS as the TPS incorporates many more functionalities (dose computation, automated tissue segmentation ...). It is, in fact, a simple tool bridging the gap between imaging (CT, MRI, ...), dosimetry (rt-Dose), geometry (rt-Struct), on one side, and the powerful packages for data modeling everybody can find in R. For this reason, espadon has many functionalities (the simplest having been explored in this document) that can be combined for doing very powerful jobs. We describe any complex examples, in details, in dedicated vignettes.