RJafroc Documentation

Dev P. Chakraborty, PhD

2020-03-17

Contents

Preface

- This book, an extended documentation of the **RJafroc** package, is undergoing extensive edits.
- It should not be used by the casual user until I give the go ahead.
- It bypasses the file size limits of **CRAN**, currently 5 MB, which severely limits the extent of the documentation that can be included with the CRAN version of the package.
- I welcome corrections and comments by the not-so-casual-user.
- Please use the GitHub website to raise issues and comments:

    – https://github.com/dpc10ster/RJafrocBook

Chapter 1

Introduction

- This is the book desribing the **RJafroc** package.
- The name of the book is RJafrocBook
- Modality and treatment are used interchangeably.
- Reader is a generic radiologist, or a computer aided detection algorithm, or any algorithmic "reader"
- TBA

1.1 References

Chapter 2

ROC DATA FORMAT

$$\theta = \frac{1}{N_L N_N} \sum_k \sum_{k'} \sum_{r=1}^{n_k^L} \sum_{r'=1}^{n_{k'}^N} \psi(X_{kr}, Y_{k'r'})$$

$$\frac{d}{dx} \left(\int_a^x f(u)\,du \right) = f(x)$$

$$\theta = \frac{1}{N_L N_N}$$

2.1 Introduction

- The purpose of this vignette is to explain the data format of the input Excel file and to introduce the capabilities of the function `DfReadDataFile()`. Background on observer performance methods are in my book (Chakraborty, 2017).
- I will start with Receiver Operating Characteristic (ROC) data (Metz, 1978), as this is by far the simplest paradigm.
- In the ROC paradigm the observer assigns a rating to each image. A rating is an ordered numeric label, and, in our convention, higher values represent greater certainty or **confidence level** for presence of disease. With human observers, a 5 (or 6) point rating scale is typically used, with 1 representing highest confidence for *absence* of disease and 5 (or 6) representing highest confidence for *presence* of disease. Intermediate values represent intermediate confidence levels for presence or absence of disease.
- Note that location information associated with the disease, if applicable, is not collected.

- There is no restriction to 5 or 6 ratings. With algorithmic observers, e.g., computer aided detection (CAD) algorithms, the rating could be a floating point number and have infinite precision. All that is required is that higher values correspond to greater confidence in presence of disease.

2.2   Note to existing users

- The Excel file format has recently undergone changes resulting in 4 extra `list` members in the final created `dataset` object (i.e., 12 members instead of 8).
- Code should run on the old format Excel files as the 4 extra list members are simply ignored.
- Reasons for the change will become clearer in these vignettes
- Basically they are needed for generalization to other data collection paradigms instead of crossed, for example to the split-plot data acquisition paradigm, and for better data entry error control.

2.3   The Excel data format

- The Excel file has three worksheets.
- These are named

  - `Truth`,
  - `NL` (or `FP`),
  - `LL` (or `TP`).

2.4   Illustrative toy file

- *Toy files* are artificial small datasets intended to illustrate essential features of the data format.

- The examples shown in this vignette corresponds to Excel file `inst/extdata/toyFiles/ROC/rocCr.xlsx` in the project directory.
- To view these files one needs to `clone` the source files from `GitHub`.

2.5   The `Truth` worksheet

- The `Truth` worksheet contains 6 columns: `CaseID`, `LesionID`, `Weight`, `ReaderID`, `ModalityID` and `Paradigm`.

- For ROC data the first five columns contain as many rows as there are cases (images) in the dataset.
- `CaseID`: unique integers, one per case, representing the cases in the dataset.
- `LesionID`: integers 0 or 1, with each 0 representing a non-diseased case and each 1 representing a diseased case.
- In the current toy dataset, the non-diseased cases are labeled `1`, `2` and `3`, while the diseased cases are labeled `70`, `71`, `72`, `73` and `74`. The values do not have to be consecutive integers; they need not be ordered; the only requirement is that they be **unique**.
- `Weight`: Not used for ROC data, a floating point value, typically filled in with 0 or 1.
- `ReaderID`: a **comma-separated** listing of reader labels, each represented by a **unique string**, that have interpreted the case. In the example shown below each cell has the value `0, 1, 2, 3, 4` meaning that each of the readers, represented by the strings "0", "1", "2", "3" and "4", have interpreted all cases (hence the "crossed" design). **With reader names that could be confused with integers, each cell in this column has to be text formatted as otherwise Excel will not accept it.** [Try entering `0, 1, 2, 3, 4` in a numeric formatted Excel cell.]
- The reader names could just as well have been `Rdr0, Rdr1, Rdr2, Rdr3, Rdr4`. The only requirement is that they be unique strings.
- Look in in the `inst/extdata/toyFiles/ROC` directory for files `rocCrStrRdrsTrts.xlsx` and `rocCrStrRdrsNonUnique.xlsx` for examples of data files using longer strings for readers. The second file generates an error because the reader names are not unique.
- `ModalityID`: a comma-separated listing of modalities (one or more modalities), each represented by a **unique string**, that are applied to each case. In the example each cell has the value `"0", "1"`. **With treatment names that could be confused with integers, each cell has to be text formatted as otherwise Excel will not accept it.**
- The treatment names could just as well have been `Trt0, Trt1`. Again, the only requirement is that they be unique strings.
- `Paradigm`: this column contains two cells, `ROC` and `crossed`. It informs the software that this is an ROC dataset, and the design is crossed, meaning each reader has interpreted each case in each modality (in statistical terminology: modality and reader factors are "crossed").
- There are 5 diseased cases in the dataset (the number of 1's in the `LesionID` column of the `Truth` worksheet).
- There are 3 non-diseased cases in the dataset (the number of 0's in the `LesionID` column).
- There are 5 readers in the dataset (each cell in the `ReaderID` column contains the string `0, 1, 2, 3, 4`).
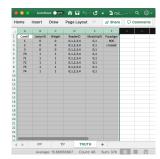- There are 2 modalities in the dataset (each cell in the `ModalityID` column contains the string `0, 1`).

Figure 2.1: Truth worksheet for file rocCr.xlsx

2.6   The structure of an ROC dataset

In the following code chunk the first statement retrieves the name of the data file, located in a hidden directory that one need not be concerned with. The second statement reads the file using the function `DfReadDataFile()` and saves it to object `x`. The third statement shows the structure of the dataset object `x`.

```
rocCr <- system.file("extdata", "toyFiles/ROC/rocCr.xlsx",
                        package = "RJafroc", mustWork = TRUE)
x <- DfReadDataFile(rocCr, newExcelFileFormat = TRUE)
str(x)
#> List of 12
#>  $ NL          : num [1:2, 1:5, 1:8, 1] 1 3 2 3 2 2 1 2 3 2 ...
#>  $ LL          : num [1:2, 1:5, 1:5, 1] 5 5 5 5 5 5 5 5 5 5 ...
#>  $ lesionVector : int [1:5] 1 1 1 1 1
#>  $ lesionID    : num [1:5, 1] 1 1 1 1 1
#>  $ lesionWeight : num [1:5, 1] 1 1 1 1 1
#>  $ dataType    : chr "ROC"
#>  $ modalityID  : Named chr [1:2] "0" "1"
#>   ..- attr(*, "names")= chr [1:2] "0" "1"
#>  $ readerID    : Named chr [1:5] "0" "1" "2" "3" ...
#>   ..- attr(*, "names")= chr [1:5] "0" "1" "2" "3" ...
#>  $ design      : chr "CROSSED"
#>  $ normalCases : int [1:3] 1 2 3
#>  $ abnormalCases: int [1:5] 70 71 72 73 74
#>  $ truthTableStr: num [1:2, 1:5, 1:8, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
```

- In the above code chunk flag `newExcelFileFormat` is set to `TRUE` as otherwise columns D - F in the `Truth` worksheet are ignored and the dataset is assumed to be crossed, with `dataType` automatically determined from the contents of the FP and TP worksheets.
- Flag `newExcelFileFormat = FALSE` is for compatibility with older

JAFROC format Excel files, which did not have these columns in the `Truth` worksheet. Its usage is deprecated.

- The dataset object `x` is a `list` variable with 12 members.
- The `x$NL` member, with dimension [2, 5, 8, 1], contains the ratings of normal cases. The extra values in the third dimension, filled with `NAs`, are needed for compatibility with FROC datasets, as unlike ROC, false positives are possible on diseased cases.
- The `x$LL`, with dimension [2, 5, 5, 1], contains the ratings of abnormal cases.
- The `x$lesionVector` member is a vector with 5 ones representing the 5 diseased cases in the dataset.
- The `x$lesionID` member is an array with 5 ones.
- The `x$lesionWeight` member is an array with 5 ones.
- The `lesionVector`, `lesionID` and `lesionWeight` members are not used for ROC datasets. They are there for compatibility with FROC datasets.
- The `dataType` member indicates that this is an `ROC` dataset.
- The `x$modalityID` member is a vector with two elements `"0"` and `"1"`, naming the two modalities.
- The `x$readerID` member is a vector with five elements `"0"`, `"1"`, `"2"`, `"3"` and `"4"`, naming the five readers.
- The `x$design` member is CROSSED; specifies the dataset design, which is "CROSSED".
- The `x$normalCases` member lists the integer names of the normal cases, 1, 2, 3.
- The `x$abnormalCases` member lists the integer names of the abnormal cases, 70, 71, 72, 73, 74.
- The `x$truthTableStr` member quantifies the structure of the dataset, as explained in **Chapter 00 Vignette #3-#5**.

2.7 The false positive (FP) ratings

These are found in the `FP` or `NL` worksheet, see below.

Figure 2.2: FP worksheet for file rocCr.xlsx

- It consists of 4 columns, each of length 30 (= # of modalities times number of readers times number of non-diseased cases).
- `ReaderID`: the reader labels: `0`, `1`, `2`, `3` and `4`. Each reader label occurs 6 times (= # of modalities times number of non-diseased cases).
- `ModalityID`: the modality or treatment labels: `0` and `1`. Each label occurs 15 times (= # of readers times number of non-diseased cases).
- `CaseID`: the case labels for non-diseased cases: `1`, `2` and `3`. Each label occurs 10 times (= # of modalities times # of readers).
- The label of a diseased case cannot occur in the FP worksheet. If it does the software generates an error.
- `FP_Rating`: the floating point ratings of non-diseased cases. Each row of this worksheet contains a rating corresponding to the values of `ReaderID`, `ModalityID` and `CaseID` for that row.

2.8   The true positive (TP) ratings

These are found in the `TP` or `LL` worksheet, see below.

Figure 2.3: TP worksheet for file rocCr.xlsx

- It consists of 5 columns, each of length 50 (= # of modalities times number of readers times number of diseased cases).
- `ReaderID`: the reader labels: `0`, `1`, `2`, `3` and `4`. Each reader label occurs 10 times (= # of modalities times number of diseased cases).
- `ModalityID`: the modality or treatment labels: `0` and `1`. Each label occurs 25 times (= # of readers times number of diseased cases).
- `LesionID`: For an ROC dataset this column contains fifty 1's (each diseased case has one lesion).
- `CaseID`: the case labels for non-diseased cases: `70`, `71`, `72`, `73` and `74`. Each label occurs 10 times (= # of modalities times # of readers). The label of a non-diseased case cannot occur in the TP worksheet.
- `TP_Rating`: the floating point ratings of diseased cases. Each row of this worksheet contains a rating corresponding to the values of `ReaderID`, `ModalityID`, `LesionID` and `CaseID` for that row.

2.9 Correspondence between `NL` member of dataset and the `FP` worksheet

- The list member **x$NL** is an array with `dim = c(2,5,8,1)`.

    - The first dimension (2) comes from the number of modalities.
    - The second dimension (5) comes from the number of readers.
    - The third dimension (8) comes from the **total** number of cases.
    - The fourth dimension is alway 1 for an ROC dataset.

- The value of **x$NL[1,5,2,1]**, i.e., 5, corresponds to row 15 of the FP table, i.e., to `ModalityID` $= 0$, `ReaderID` $= 4$ and `CaseID` $= 2$.
- The value of **x$NL[2,3,2,1]**, i.e., 4, corresponds to row 24 of the FP table, i.e., to `ModalityID` 1, `ReaderID` 2 and `CaseID` 2.
- All values for case index > 3 are `-Inf`. For example the value of **x$NL[2,3,4,1]** is `-Inf`. This is because there are only 3 non-diseased cases. The extra length is needed for compatibility with FROC datasets.

2.10 Correspondence between `LL` member of dataset and the `TP` worksheet

- The list member **x$LL** is an array with `dim = c(2,5,5,1)`.

    - The first dimension (2) comes from the number of modalities.
    - The second dimension (5) comes from the number of readers.
    - The third dimension (5) comes from the number of diseased cases.
    - The fourth dimension is alway 1 for an ROC dataset.

- The value of **x$LL[1,1,5,1]**, i.e., 4, corresponds to row 6 of the TP table, i.e., to `ModalityID` $= 0$, `ReaderID` $= 0$ and `CaseID` $= 74$.
- The value of **x$LL[1,2,2,1]**, i.e., 3, corresponds to row 8 of the TP table, i.e., to `ModalityID` $= 0$, `ReaderID` $= 1$ and `CaseID` $= 71$.
- There are no -Inf values in x$LL: `any(x$LL == -Inf)` = FALSE.

2.11 Correspondence using the `which` function

- Converting from **names** to **subscripts** (indicating position in an array) can be confusing.
- The following example uses the `which` function to help out.
- The first line says that the `abnormalCase` named 70 corresponds to subscript 1 in the LL array case dimension.
- The second line prints the NL rating for `modalityID` $= 0$, `readerID` $= 1$ and `normalCases` $= 1$.

- The third line prints the LL rating for `modalityID = 0`, `readerID = 1` and `abnormalCases = 70`.
- The last line shows what happens if one enters an invalid value for name; the result is a `numeric(0)`.
- Note that in each of these examples, the last dimension is 1 because we are dealing with an ROC dataset.
- The reader is encouraged to examine the correspondence between the NL and LL ratings and the Excel file using this method.

```
which(x$abnormalCases == 70)
#> [1] 1
x$NL[which(x$modalityID == "0"),which(x$readerID == "1"),which(x$normalCases == 1),1]
#> [1] 2
x$LL[which(x$modalityID == "0"),which(x$readerID == "1"),which(x$abnormalCases == 70),
#> [1] 5
x$LL[which(x$modalityID == "a"),which(x$readerID == "1"),which(x$abnormalCases == 70),
#> numeric(0)
```

2.12   References

Chapter 3

Metz Eqn36 numerical check

3.1 Helper functions

3.2 Main code and output

```r
npts <-  10000
for (i in 1:2) {
  for (j in 1:5) {
    C   <-  c1[i,j]
    da  <-  d_a1[i,j]
    ret <- GetLimits(da,C)
    LL <- ret$LL;UL <- ret$UL
    vc  <-  seq (LL, UL, length.out = npts)
    TPF  <-  TruePositiveFraction (vc, da, C)
    FPF <- FalsePositiveFraction (vc, da, C)
    FPF <- rev(FPF);TPF <- rev(TPF)
    df2 <- data.frame(FPF = FPF, TPF = TPF)
    # do integral numerically
    numAuc <- trapz(FPF, TPF)
    # Implement Eqn. 36 from Metz-Pan paper
    rho <- -(1-C^2)/(1+C^2);sigma <- rbind(c(1, rho), c(rho, 1))
    lower <- rep(-Inf,2);upper <- c(-da/sqrt(2),0)
    aucProproc <- pnorm(da/sqrt(2)) + 2 * pmvnorm(lower, upper, sigma = sigma)
    aucProproc <-  as.numeric(aucProproc)
    cat("i = ", i,"j = ", j,"C = ", C, ", da = ", da, "aucProproc =", aucProproc, "Norm. Diff. =
```

17

```
  }
}
#> i =   1 j =   1 C =   -0.1322804 , da =   1.197239 aucProproc = 0.8014164 Norm. Diff. =
#> i =   1 j =   2 C =   -0.08696513 , da =   1.771176 aucProproc = 0.8947898 Norm. Diff.
#> i =   1 j =   3 C =   -0.1444419 , da =   1.481935 aucProproc = 0.8526605 Norm. Diff. =
#> i =   1 j =   4 C =   0.08046016 , da =   1.513757 aucProproc = 0.8577776 Norm. Diff. =
#> i =   1 j =   5 C =   0.2225588 , da =   1.740157 aucProproc = 0.8909392 Norm. Diff. =
#> i =   2 j =   1 C =   -0.08174248 , da =   0.6281251 aucProproc = 0.6716574 Norm. Diff.
#> i =   2 j =   2 C =   0.04976448 , da =   0.9738786 aucProproc = 0.7544739 Norm. Diff.
#> i =   2 j =   3 C =   -0.1326126 , da =   1.155871 aucProproc = 0.7931787 Norm. Diff. =
#> i =   2 j =   4 C =   0.1182226 , da =   1.620176 aucProproc = 0.8740274 Norm. Diff. =
#> i =   2 j =   5 C =   0.0781033 , da =   0.8928816 aucProproc = 0.7360989 Norm. Diff. =
```

3.3   Discussion

Note the close correspondence between the formula, Eqn. 36 in the Metz-Pan
paper and the numerical estimate. As a historical note, Eqn. 31 and Eqn. 36
(they differ only in parameterizations) in the referenced publication are provided
without proof – it was probably obvious to Prof Metz or he wanted to leave it
to us "mere mortals" to figure it out, as a final parting gesture of his legacy.
The author once put a significant effort into proving it and even had a bright
graduate student from the biostatistics department work on it to no avail. The
author has observed that these equations always yield very close to the numerical
estimates, to within numerical precisions, so the theorem is correct empirically,
but he has been unable to prove it analytically. It is left as an exercise for a
gifted reader to prove/disprove these equations.

Chapter 4

ROI paradigm data

4.1 Introduction; this vignette is under construction!

- In the region-of-interest (ROI) paradigm (Obuchowski, 1997, Obuchowski et al. (2000)) each case is regarded as consisting of Q_k ($Q_k \geq 1$) "quadrants" or "regions-of-interest" or ROIs, where k is the case index ($k = 1, 2, ..., K$) and K is the total number of cases (i.e., case-level non-diseased plus case-level diseased cases). Each ROI needs to be classified, by the investigator, as either ROI-level-non-diseased (i.e., it has no lesions) or ROI-level-diseased (i.e., it has at least one lesion). **Note the distinction between case-level and ROI-level truth states.** One can have ROI-level non-diseased regions in a case-level diseased case. A case-level diseased case must contain at least one ROI-level diseased region and a case-level non-diseased case cannot have any ROI-level diseased regions.
- The observer gives a single rating (in fact an ordered label) to each ROI, denoted R_{kr} ($r = 1, 2, ..., Q_k$). Here r is the ROI index and k is the case index. The rating can be an integer or quasi- continuous (e.g., $0 - 100$), or a floating point value, as long as higher numbers represent greater confidence in presence of one or more lesions in the ROI.
- The ROI paradigm is not restricted to 4 or even a constant number of ROIs per case. That is the reason for the k subscript in Q_k.
- The ROI data structure is a special case of the FROC data structure, the essential difference being that the number of ratings per case is an a-priori known value, equal to Q_k.
- ROI-level non-diseased region ratings are stored in the `NL` field and ROI-level diseased region ratings are stored in the `LL` field.

- One can think of the ROI paradigm as similar to the FROC paradigm, but

with localization accuracy restricted to belonging to a region (one cannot distinguish multiple lesions within a region). Unlike the FROC paradigm, a rating *is required* for every ROI.

4.2   An example ROI dataset

An example simulated ROI dataset is included as `datasetROI`.

```
str(datasetROI)
#> List of 8
#>  $ NL         : num [1:2, 1:5, 1:90, 1:4] 0.95 0.927 0.556 0.805 1.421 ...
#>  $ LL         : num [1:2, 1:5, 1:40, 1:4] 1.57 2.31 2.3 2.34 2.34 ...
#>  $ lesionVector: int [1:40] 2 3 2 2 3 3 1 2 3 3 ...
#>  $ lesionID    : num [1:40, 1:4] 2 1 1 1 1 2 4 1 1 1 ...
#>  $ lesionWeight: num [1:40, 1:4] 0.5 0.333 0.5 0.5 0.333 ...
#>  $ dataType    : chr "ROI"
#>  $ modalityID  : Named chr [1:2] "1" "2"
#>   ..- attr(*, "names")= chr [1:2] "1" "2"
#>  $ readerID    : Named chr [1:5] "1" "2" "3" "4" ...
#>   ..- attr(*, "names")= chr [1:5] "1" "2" "3" "4" ...
datasetROI$NL[1,1,1,]
#> [1]  0.9498680 -0.0582497 -0.7763780  0.0120730
mean(datasetROI$NL[,,1:50,])
#> [1] 0.1014348
datasetROI$NL[1,1,51,]
#> [1] 1.01867 0.34710    -Inf    -Inf
datasetROI$lesionVector[1]
#> [1] 2
datasetROI$LL[1,1,1,]
#> [1] 1.56928 2.05945    -Inf    -Inf
x <- datasetROI$LL;mean(x[is.finite(x)])
#> [1] 1.815513
```

Examination of the output reveals that:

- This is a 2-treatment 5-reader dataset, with 50 non-diseased cases and 40 diseased cases, and $Q_k = 4$ for all k.

- For treatment 1, reader 1, case 1 (the 1st non-diseased case) the 4 ratings are 0.949868, -0.0582497, -0.776378, 0.012073. The mean of all ratings on non-diseased cases is 0.1014348.

- For treatment 1, reader 1, case 51 (the 1st diseased case) the NL ratings are 1.01867, 0.3471. There are only two finite values because this case

has two ROI-level-diseased regions, and 2 plus 2 makes for the assumed 4-regions per case. The corresponding `$lesionVector` field is 2.

- The ratings of the 2 ROI-level-diseased ROIs on this case are 1.56928, 2.05945. The mean rating over all ROI-level-diseased ROIs is 1.8155127.

4.3   The ROI Excel data file

- An Excel file in JAFROC format containing simulated ROI data corresponding to `datasetROI`, is included with the distribution. The first command (below) finds the location of the file and the second command reads it and saves it to a dataset object `ds`. !!!DPC!!!
- The `DfReadDataFile` function automatically recognizes that this is an *ROI* dataset. Its structure is similar to the JAFROC format Excel file, with some important differences, noted below. It contains three worksheets:

```
## fileName <- system.file(
##     "extdata", "RoiData.xlsx", package = "RJafroc", mustWork = TRUE)
## ds <- DfReadDataFile(fileName)
## ds$dataType
```
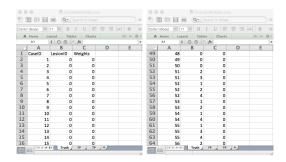

Figure 4.1: Fig. 1 two views of Truth worksheet

- The `Truth` worksheet, Fig. 1, indicates which cases are diseased and which are non-diseased and the number of ROI-level-diseased region on each case.

  – There are 50 non-diseased cases (labeled 1-50) under column `CaseID` and 40 diseased cases (labeled 51-90).

  – The `LesionID` field for each non-diseased case (e.g., `CaseID = 1`) is zero and there is one row per case. For diseased cases, this field has a variable number of entries, ranging from 1 to 4. As an example,

there are two rows for `CaseID` $= 51$ in the Excel file: one with
`LesionID` $= 2$ and one with `LesionID` $= 3$.

– The `Weights` field is always zero (this field is not used in ROI anal-
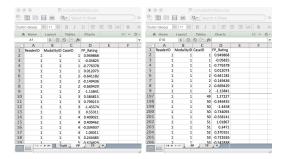ysis).

Figure 4.2: Fig. 2 two views of FP worksheet

- The `FP` (or `NL`) worksheet - this lists the ratings of ROI-level-non-diseased
regions.

  – For `ReaderID` $= 1$, `ModalityID` $= 1$ and `CaseID` $= 1$ there are 4
  rows, corresponding to the 4 ROI-level-non-diseased regions in this
  case. The corresponding ratings are 0.949868, -0.0582497, -0.776378,
  0.012073. The pattern repeats for other treatments and readers, but
  the rating are, of course, different.

  – Each `CaseID` is represented in the `FP` worksheet (a rare exception
  could occur if a case-level diseased case has 4 diseased regions).

Figure 4.3: Fig. 2 TP worksheet

- The `TP` (or `LL`) worksheet - this lists the ratings of ROI-level-diseased
regions.

– Because non-diseased cases generate TPs, one does not find any entry with `CaseID` = 1-50 in the `TP` worksheet.

– The lowest `CaseID` in the `TP` worksheet is 51, which corresponds to the first diseased case.

– There are two entries for this case, corresponding to the two ROI-level-diseased regions present in this case. Recall that corresponding to this `CaseID` in the `Truth` worksheet there were two entries with `LesionID` = 2 and 3. These must match the `LesionID`'s listed for this case in the `TP` worksheet. Complementing these two entries, in the `FP` worksheet for `CaseID` = 51, there are 2 entries corresponding to the two ROI-level-non-diseased regions in this case.

– One should confirm that for each diseased case the sum of the number of entries in the `TP` and `FP` worksheets is always 4.

4.4 Next, TBA

The next vignette illustrates significance testing for this paradigm.

4.5 References

Chapter 5

Analyzing data acquired according to the ROI paradigm

5.1 Introduction; this vignette is under construction!

5.2 Note to self (10/29/19) !!!DPC!!!

The FOM and DeLong method implementations need checking with a toy dataset.

5.3 Introduction

- For an ROI dataset `StSignificanceTesting()` automatically defaults to `method = "ORH"`, `covEstMethod = "DeLong"` and `FOM = "ROI"`.

- The covariance estimation method is based on the original DeLong method (DeLong et al., 1988), which is valid only for the trapezoidal AUC, i.e. ROC data, as extended by (Obuchowski, 1997) to ROI data, see formula below.

- The essential differences from conventional ROC analyses are in the definition of the ROI figure of merit, see below, and the procedure developed by (Obuchowski, 1997) for estimating the covariance matrix. Once the

25

covariances are known, `method = "ORH"` can be applied to perform significance testing, as described in (Obuchowski and Rockette, 1995) and (Chakraborty, 2017, Chapter 10).

5.4 The ROI figure of merit

Let X_{kr} denote the rating for the r$^{\text{th}}$ **lesion-containing** ROI in the k$^{\text{th}}$ case and let n_k^L be the total number of lesion-containing ROIs in the k$^{\text{th}}$ case. Similarly, let Y_{kr} denote the rating for the r$^{\text{th}}$ **lesion-free** ROI in the k$^{\text{th}}$ case and n_k^N denote the total number of lesion-free ROIs in the k$^{\text{th}}$ case. Let N_L denote the total number of lesion-containing ROIs in the image set and N_N denote the total number of lesion-free ROIs. These are given by:

$$N_L = \sum_k n_k^L$$

and

$$N_N = \sum_k n_k^N$$

. The ROI figure of merit θ is defined by:

$$\theta = \frac{1}{N_L N_N} \sum_k \sum_{k'} \sum_{r=1}^{n_k^L} \sum_{r'=1}^{n_{k'}^N} \psi(X_{kr}, Y_{k'r'})$$

$$\theta = \frac{1}{N_L N_N} \sum_k \sum_{k'} \sum_{r=1}^{n_k^L} \sum_{r'=1}^{n_{k'}^N} \psi(X_{kr}, Y_{k'r'})$$

The kernel function $\Psi(X, Y)$ is defined by:

$$\psi(X, Y) = \begin{bmatrix} 1 & \text{if} & X < Y \\ 0.5 & \text{if} & X = Y \\ 0 & \text{if} & X > Y \end{bmatrix}$$

The ROIs are *effectively regarded as mini-cases* and one calculates the FOM as the Wilcoxon statistic considering the mini-cases as actual cases. The correlations between the ratings of ROIs on the same case are accounted for in the analysis.

5.5 Calculation of the ROI figure of merit.

```
UtilFigureOfMerit(datasetROI, FOM = "ROI")
#>           Rdr1      Rdr2      Rdr3      Rdr4      Rdr5
#> Trt1 0.9057239 0.8842834 0.8579279 0.9350207 0.8352103
#> Trt2 0.9297186 0.9546035 0.8937652 0.9531716 0.8770076
fom <- UtilFigureOfMerit(datasetROI, FOM = "ROI")
```

- If the correct `FOM` is not supplied, it defaults to `FOM = ROI`.

- This is a 2-treatment 5-reader dataset.

- For treatment 1, reader 1 the figure of merit is 0.9057239.

- For treatment 2, reader 5 the figure of merit is 0.8770076.

- Etc.

5.6   Significance testing

When `dataset$dataType == "ROI"` the FOM defaults to "ROI" (meaning the above formula) and the covariance estimation method defaults to `covEstMethod = "DeLong"`.

```
ret <- StSignificanceTesting(datasetROI, FOM = "Wilcoxon")
#> ROI dataset: forcing method = `ORH`, covEstMethod = `DeLong` and FOM = `ROI`.
str(ret)
#> List of 14
#>  $ fomArray         : num [1:2, 1:5] 0.906 0.93 0.884 0.955 0.858 ...
#>   ..- attr(*, "dimnames")=List of 2
#>   .. ..$ : chr [1:2] "Trt1" "Trt2"
#>   .. ..$ : chr [1:5] "Rdr1" "Rdr2" "Rdr3" "Rdr4" ...
#>  $ meanSquares      :'data.frame':   1 obs. of  3 variables:
#>   ..$ msT : num 0.00361
#>   ..$ msR : num 0.00256
#>   ..$ msTR: num 0.000207
#>  $ varComp          :'data.frame':   1 obs. of  6 variables:
#>   ..$ varR : num 0.00108
#>   ..$ varTR: num 0.000153
#>   ..$ cov1 : num 0.000247
#>   ..$ cov2 : num 0.000187
#>   ..$ cov3 : num 0.000154
#>   ..$ var  : num 0.000333
#>  $ FTestStatsRRRC   :'data.frame':   1 obs. of  4 variables:
```

```
#>    ..$ fRRRC  : num 9.76
#>    ..$ ndfRRRC: num 1
#>    ..$ ddfRRRC: num 12.8
#>    ..$ pRRRC  : num 0.00817
#> $ ciDiffTrtRRRC      :'data.frame':    1 obs. of  8 variables:
#>    ..$ Treatment: chr "Trt1-Trt2"
#>    ..$ Estimate : num -0.038
#>    ..$ StdErr   : num 0.0122
#>    ..$ DF       : num 12.8
#>    ..$ t        : num -3.12
#>    ..$ PrGTt    : num 0.00817
#>    ..$ CILower  : num -0.0643
#>    ..$ CIUpper  : num -0.0117
#> $ ciAvgRdrEachTrtRRRC :'data.frame':    2 obs. of  6 variables:
#>    ..$ Treatment: Factor w/ 2 levels "Trt1","Trt2": 1 2
#>    ..$ Area     : num [1:2] 0.884 0.922
#>    ..$ StdErr   : num [1:2] 0.0232 0.0197
#>    ..$ DF       : num [1:2] 12.2 10.1
#>    ..$ CILower  : num [1:2] 0.833 0.878
#>    ..$ CIUpper  : num [1:2] 0.934 0.966
#> $ FTestStatsFRRC      :'data.frame':    1 obs. of  4 variables:
#>    ..$ fFRRC  : num 16.6
#>    ..$ ndfFRRC: num 1
#>    ..$ ddfFRRC: num Inf
#>    ..$ pFRRC  : num 4.58e-05
#> $ ciDiffTrtFRRC       :'data.frame':    1 obs. of  8 variables:
#>    ..$ Treatment: chr "Trt1-Trt2"
#>    ..$ Estimate : num -0.038
#>    ..$ StdErr   : num 0.00933
#>    ..$ DF       : num Inf
#>    ..$ t        : num -4.08
#>    ..$ PrGTt    : num 4.58e-05
#>    ..$ CILower  : num -0.0563
#>    ..$ CIUpper  : num -0.0197
#> $ ciAvgRdrEachTrtFRRC :'data.frame':    2 obs. of  6 variables:
#>    ..$ Treatment: Factor w/ 2 levels "Trt1","Trt2": 1 2
#>    ..$ Area     : num [1:2] 0.884 0.922
#>    ..$ StdErr   : num [1:2] 0.0163 0.0129
#>    ..$ DF       : num [1:2] Inf Inf
#>    ..$ CILower  : num [1:2] 0.852 0.896
#>    ..$ CIUpper  : num [1:2] 0.916 0.947
#> $ ciDiffTrtEachRdrFRRC:'data.frame':    5 obs. of  9 variables:
#>    ..$ Reader   : Factor w/ 5 levels "Rdr1","Rdr2",..: 1 2 3 4 5
#>    ..$ Treatment: Factor w/ 1 level "Trt1-Trt2": 1 1 1 1 1
#>    ..$ Estimate : num [1:5] -0.024 -0.0703 -0.0358 -0.0182 -0.0418
```

```
#>   ..$ StdErr   : num [1:5] 0.01025 0.01448 0.01648 0.00928 0.01398
#>   ..$ DF       : num [1:5] Inf Inf Inf Inf Inf
#>   ..$ t        : num [1:5] -2.34 -4.86 -2.17 -1.96 -2.99
#>   ..$ PrGTt    : num [1:5] 1.93e-02 1.20e-06 2.97e-02 5.05e-02 2.79e-03
#>   ..$ CILower  : num [1:5] -0.0441 -0.0987 -0.0681 -0.0363 -0.0692
#>   ..$ CIUpper  : num [1:5] -3.90e-03 -4.19e-02 -3.53e-03 3.88e-05 -1.44e-02
#>  $ varCovEachRdr      :'data.frame':   5 obs. of  3 variables:
#>   ..$ Reader: Factor w/ 5 levels "Rdr1","Rdr2",..: 1 2 3 4 5
#>   ..$ Var   : num [1:5] 0.000269 0.000227 0.000481 0.000168 0.000522
#>   ..$ Cov1  : num [1:5] 0.000216 0.000122 0.000345 0.000125 0.000424
#>  $ FTestStatsRRFC      :'data.frame':   1 obs. of  4 variables:
#>   ..$ fRRFC  : num 17.5
#>   ..$ ndfRRFC: num 1
#>   ..$ ddfRRFC: num 4
#>   ..$ pRRFC  : num 0.0139
#>  $ ciDiffTrtRRFC       :'data.frame':   1 obs. of  8 variables:
#>   ..$ Treatment: chr "Trt1-Trt2"
#>   ..$ Estimate : num -0.038
#>   ..$ StdErr   : num 0.00909
#>   ..$ DF       : num 4
#>   ..$ t        : num -4.18
#>   ..$ PrGTt    : num 0.0139
#>   ..$ CILower  : num -0.0633
#>   ..$ CIUpper  : num -0.0128
#>  $ ciAvgRdrEachTrtRRFC :'data.frame':   2 obs. of  6 variables:
#>   ..$ Treatment: Factor w/ 2 levels "Trt1","Trt2": 1 2
#>   ..$ Area     : num [1:2] 0.884 0.922
#>   ..$ StdErr   : num [1:2] 0.0175 0.0157
#>   ..$ DF       : num [1:2] 4 4
#>   ..$ CILower  : num [1:2] 0.835 0.878
#>   ..$ CIUpper  : num [1:2] 0.932 0.965
```

- While `ret` is a list with many (22) members, their meanings should be clear from the notation. As an example:

- The variance components are given by:

```
ret$varComp
#>        varR       varTR         cov1         cov2         cov3         var
#> 1 0.001082359 0.0001526084 0.0002465125 0.0001870571 0.0001543764 0.0003333119
```

5.6.1 RRRC analysis

```
ret$FTestStatsRRRC$fRRRC
#> [1] 9.763602
ret$FTestStatsRRRC$ndfRRRC
#> [1] 1
ret$FTestStatsRRRC$ddfRRRC
#> [1] 12.82259
ret$FTestStatsRRRC$pRRRC
#> [1] 0.008173042
```

- The F-statistic is , with `ndf = 1` and `ddf = `, which yields a p-value of .

- The confidence interval for the reader averaged difference between the two treatments is given by:

```
ret$ciDiffTrtRRRC
#>   Treatment   Estimate     StdErr       DF        t       PrGTt    CILower
#> 1 Trt1-Trt2 -0.03802005 0.01216768 12.82259 -3.124676 0.008173042 -0.06434373
#>      CIUpper
#> 1 -0.01169636
```

- The FOM difference (treatment 1 minus 2) is -0.03802, which is significant, p-value = 0.008173, F-statistic = 9.7636016, ddf = 12.8225898. The confidence interval is (-0.0643437, -0.0116964).

5.6.2 FRRC analysis

```
ret$FTestStatsFRRC$fFRRC
#> [1] 16.6135
ret$FTestStatsFRRC$ndfFRRC
#> [1] 1
ret$FTestStatsFRRC$ddfFRRC
#> [1] Inf
ret$FTestStatsFRRC$pFRRC
#> [1] 4.582365e-05
```

- The F-statistic is 16.6135014, with `ndf = 1` and `ddf = Inf`, which yields a p-value of 4.5823651×10^{-5}.

- The confidence interval for the reader averaged difference between the two treatments is given by:

```
ret$ciDiffTrtFRRC
#>   Treatment     Estimate     StdErr  DF         t        PrGTt     CILower
#> 1 Trt1-Trt2 -0.03802005 0.009327861 Inf -4.075966 4.582365e-05 -0.05630232
#>      CIUpper
#> 1 -0.01973778
```

5.6.3 RRFC analysis

```
ret$FTestStatsRRFC$fRRFC
#> [1] 17.48107
ret$FTestStatsRRFC$ndfRRFC
#> [1] 1
ret$FTestStatsRRFC$ddfRRFC
#> [1] 4
ret$FTestStatsRRFC$pRRFC
#> [1] 0.01390667
```

- The F-statistic is 17.4810666, with `ndf = 1` and `ddf = 4`, which yields a p-value of 0.0139067.

- The confidence interval for the reader averaged difference between the two treatments is given by:

```
ret$ciDiffTrtRRFC
#>   Treatment     Estimate    StdErr DF         t      PrGTt     CILower
#> 1 Trt1-Trt2 -0.03802005 0.00909345  4 -4.181037 0.01390667 -0.06326751
#>      CIUpper
#> 1 -0.01277258
```

5.7 Summary

TBA

5.8 References

Bibliography

Chakraborty, D. P. (2017). *Observer Performance Methods for Diagnostic Imaging - Foundations, Modeling, and Applications with R-Based Examples.* CRC Press, Boca Raton, FL.

DeLong, E. R., DeLong, D. M., and Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44:837–845.

Metz, C. (1978). Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 8(4):283–298.

Obuchowski, N. A. (1997). Nonparametric analysis of clustered roc curve data. *Biometrics*, 53:567–578.

Obuchowski, N. A., Lieber, M. L., and Powell, K. A. (2000). Data analysis for detection and localization of multiple abnormalities with application to mammography. *Acad. Radiol.*, 7(7):516–525.

Obuchowski, N. A. and Rockette, H. (1995). Hypothesis testing of the diagnostic accuracy for multiple diagnostic tests: An anova approach with dependent observations. *Communications in Statistics: Simulation and Computation*, 24:285–308.