

Simulate MRMC Data

Here we demonstrate how to simulate Multi-reader Multi-case (MRMC) data using functions in the iMRMC package.

`sim.NormalIG.Hierarchical` is a function that simulates MRMC **agreement** data (no binary truth state returned). The model for the `sim.NormalIG.Hierarchical` function is described here:

- S. Wen and B. D. Gallas, “Three-Way Mixed Effect ANOVA to Estimate MRMC Limits of Agreement,” *Statistics in Biopharmaceutical Research**, 14, pp. 532–541, 2022, <https://www.doi.org/10.1080/19466315.2022.2063169>.

`sim.gRoeMetz` is a function that simulates MRMC **ROC** data. We note here that the model for `sim.gRoeMetz` does not yield readers with different average/expected levels of agreement with one another. This limitation was the reason for developing the model for the `sim.NormalIG.Hierarchical` simulation. The model for the `sim.gRoeMetz` function is described here:

- B. D. Gallas and S. L. Hillis, “Generalized Roe and Metz ROC model: analytic link between simulated decision scores and empirical AUC variances and covariances,” *J Med Img*, 1, no. 3, p. 031006, 2014, <https://www.doi.org/10.1117/1.JMI.1.3.031006>.

1. Using the new hierachical model to simulate the MRMC agreement data

The following is how to use the `sim.NormalIG.Hierachcial` function to simulate MRMC agreement data (with no truth state). First, we initialize a list of simulation parameters using the `sim.NormalIG.Hierarchical.config` function. There are many simulation parameters. All have default values that can be changed in the function call. There are some parameters returned by the function that are equal for convenience and might need to be changed before simulating the data. Below, we demonstrate specifying non-default numbers of readers and cases.

```
# configuration
nR = 5      #number of readers
nC = 100   #number of cases

config <- sim.NormalIG.Hierarchical.config(nR=nR, nC=nC, modalityID = c("testA","testB"))
```

Here are all the simulation parameters:

```
print(names(config))

## [1] "modalityID" "nR"          "nC"          "C_dist"      "mu"
## [6] "tau_A"       "tau_B"       "sigma_C"     "alpha_R"     "beta_R"
## [11] "sigma_C.A"   "alpha_R.A"   "beta_R.A"    "sigma_C.B"   "alpha_R.B"
## [16] "beta_R.B"    "C_scale"     "RC_scale"    "tauC_scale"  "tauRCE_scale"
```

Next, we set the random seed for reproducible simulations (if needed) and simulate the MRMC agreement data based on the `config` list of parameters. The data frame for 5 readers and 100 cases for 2 modalities has 1000 rows (5x100x2). The 4 columns specify the reader ID, the case ID, the modality ID, and the score, or observation.

```
# simulate MRMC study
set.seed(1, kind = "L'Ecuyer-CMRG")
dFrame.newH <- sim.NormalIG.Hierarchical(config)

# check the first and last few lines of the simulated dataframe
print(dFrame.newH[1:6, ])
```

```
##   caseID readerID modalityID   score
## 1 Case1  reader1     testA  0.6946134
## 2 Case2  reader1     testA  2.9965743
## 3 Case3  reader1     testA  1.7412831
## 4 Case4  reader1     testA -1.2549331
## 5 Case5  reader1     testA -0.2952628
## 6 Case6  reader1     testA  3.0468024
```

```
print(dFrame.newH[(nrow(dFrame.newH) - 5):nrow(dFrame.newH), ])
```

```
##      caseID readerID modalityID   score
## 995 Case95  reader5     testB -0.1549758
## 996 Case96  reader5     testB -1.2040039
## 997 Case97  reader5     testB  1.6544295
## 998 Case98  reader5     testB -1.2125349
## 999 Case99  reader5     testB -0.2017070
## 1000 Case100 reader5     testB  2.0605555
```

2. Using `sim.gRoeMetz` in `iMRMC` package to simulate MRMC ROC data

The following demonstrates how to use the `sim.gRoeMetz` function to simulate MRMC ROC data (with truth state). Again, the process starts by creating a `config` list of default simulation parameters that can be changed in the function call.

```
# configuration
nR = 5      #number of readers
nC.neg = 50 #number of positive cases
nC.pos = 50 #number of negative cases

config <- sim.gRoeMetz.config(nR = nR, nC.neg = nC.neg, nC.pos = nC.pos)
```

Here are the simulation parameters for the `sim.gRoeMetz` simulation function. Note that there are some parameters that are set equal for convenience and might need to be changed before simulating the data.

```
print(names(config))

## [1] "modalityID.A" "modalityID.B" "nR"           "nC.neg"       "nC.pos"
## [6] "mu.neg"       "var_r.neg"    "var_c.neg"    "var_rc.neg"   "mu.pos"
## [11] "var_r.pos"    "var_c.pos"    "var_rc.pos"   "mu.Aneg"      "var_r.Aneg"
```

```
## [16] "var_c.Aneg" "var_rc.Aneg" "mu.Apos" "var_r.Apos" "var_c.Apos"
## [21] "var_rc.Apos" "mu.Bneg" "var_r.Bneg" "var_c.Bneg" "var_rc.Bneg"
## [26] "mu.Bpos" "var_r.Bpos" "var_c.Bpos" "var_rc.Bpos"
```

We again set the seed for reproducible simulations (if needed) before we simulate the data. As we see below, the simulated data starts with truth state of each case. This is followed by the reading scores from each of the readers. Since we simulate 50 positive cases and 50 negative cases, there are 100 rows for the truth and $100 \times 5 \times 2 = 1000$ rows for the scores from 5 readers for 2 modalities. Therefore, the data frame has 1100 rows total.

```
# simulate MRMCC study
set.seed(1, kind = "L'Ecuyer-CMRG")
dFrame.gRM <- sim.gRoeMetz(config)

# check the first and last few lines of the simulated dataframe
print(dFrame.gRM[1:6, ])
```

```
## readerID caseID modalityID score
## 1 truth negCase1 truth 0
## 2 truth negCase2 truth 0
## 3 truth negCase3 truth 0
## 4 truth negCase4 truth 0
## 5 truth negCase5 truth 0
## 6 truth negCase6 truth 0
```

```
print(dFrame.gRM[(nrow(dFrame.gRM) - 5):nrow(dFrame.gRM), ])
```

```
## readerID caseID modalityID score
## 1095 reader5 posCase45 testB 2.9342657
## 1096 reader5 posCase46 testB 1.0639768
## 1097 reader5 posCase47 testB 0.9885389
## 1098 reader5 posCase48 testB -0.5130517
## 1099 reader5 posCase49 testB 0.9254049
## 1100 reader5 posCase50 testB 2.4164279
```

To combine the truth data and the reader scores and change the data to a dataframe with 5 columns (readerID, caseID, modalityID, score, truth) we can use the function `undoIMRMCdf` in `iMRMC` package.

```
dFrame.gRM.2 <- undoIMRMCdf(dFrame.gRM)

# check the first and last few lines of the simulated dataframe
print(dFrame.gRM.2[1:6, ])
```

```
## caseID readerID modalityID score truth
## 1 negCase1 reader1 testA 0.36795249 0
## 2 negCase1 reader2 testB 0.05607467 0
## 3 negCase1 reader3 testB -0.46914121 0
## 4 negCase1 reader4 testA -1.38686722 0
## 5 negCase1 reader1 testB -0.49895246 0
## 6 negCase1 reader5 testA -1.64836669 0
```

```
print(dFrame.gRM.2[(nrow(dFrame.gRM.2) - 5):nrow(dFrame.gRM.2), ])
```

##	caseID	readerID	modalityID	score	truth
## 995	posCase9	reader5	testA	0.9233879	1
## 996	posCase9	reader4	testB	4.1614474	1
## 997	posCase9	reader4	testA	3.0874475	1
## 998	posCase9	reader2	testB	3.4945769	1
## 999	posCase9	reader1	testA	2.2259825	1
## 1000	posCase9	reader5	testB	1.3017587	1