An example session for analyzing Inversion Recovery MRI and MR Elastography data

Jörg Polzehl and Karsten Tabelow

March 6, 2025

This document illustrates the workflow of analyzing Inversion Recovery Magnetic Resonance Imaging (IRMRI) data. The example uses noisy IR data created from a small sub cube of an artificial IR image (Infinity Inversion Time), a corresponding segmentation image and MR Elastography data. For neuroimaging bacckground we refer to (**MNRLilai21b**) (IRMRI) and (**MNRLilai21a**) (MRE).

For an more extended introduction we refer to **MRBIbook2** Chapter 6.

1 Generating the IR MRI data

First, we specify the directory where the data are stored within the package

```
> dataDir0 <- system.file("extdataIR", package = "qMRI")
> dataDir <- tempdir()
> library(oro.nifti)
```

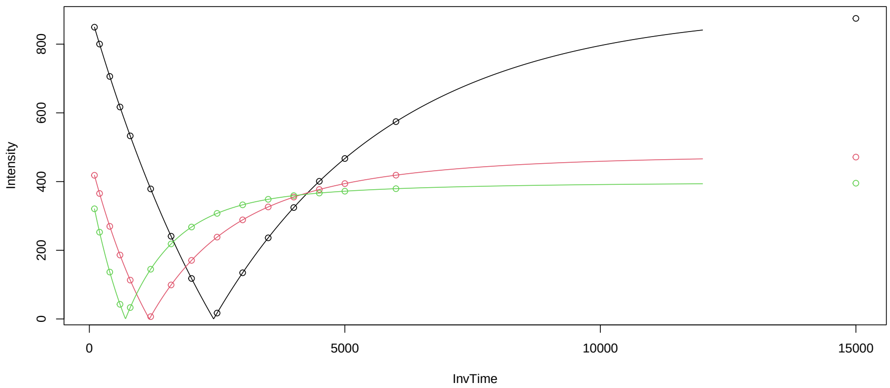
We now generate IRMRI data following a model that assumes voxel to contain a mixture of a solid tissue (either DM or WM) and fluid.

```
> library(qMRI)
> segm <- readNIfTI(file.path(dataDir0,"Brainweb_segm"))
> Sf <- 900
> Rf <- 0.000285
> Sgm <- 400
> Rgm <- 0.00075
> fgm <- .15
> Swm <- 370
> Rwm <- 0.0011
> fwm <- .05
> InvTimes <- c(100, 200, 400, 600, 800, 1200, 1600, 2000, 2500, 3000,
+               3500, 4000, 4500, 5000, 6000, Inf)
> InvTimes0 <- c(100, 200, 400, 600, 800, 1200, 1600, 2000, 2500, 3000,
+               3500, 4000, 4500, 5000, 6000, 15000)
```

Typical intensities as functions of inversion times an tissue type (black for CSF, red for GM and green for WM) are illustrated in Figure **??**

```
> x <- seq(100,12000,10)
> fintCSF <- qMRI:::IRhomogen(c(Sf,Rf),InvTimes0)
```

```
> fintGM <- qMRI:::IRmix2(c(fgm,Rgm,Sgm),InvTimes0,Sf,Rf)
> fintWM <- qMRI:::IRmix2(c(fwm,Rwm,Swm),InvTimes0,Sf,Rf)
> plot(InvTimes0,fintCSF,xlab="InvTime",ylab="Intensity")
> points(InvTimes0,fintGM,col=2)
> points(InvTimes0,fintWM,col=3)
> lines(x,qMRI:::IRhomogen(c(Sf,Rf),x))
> lines(x,qMRI:::IRmix2(c(fgm,Rgm,Sgm),x,Sf,Rf),col=2)
> lines(x,qMRI:::IRmix2(c(fwm,Rwm,Swm),x,Sf,Rf),col=3)
```



We generate artificial Rician distributed data with standard deviation $\sigma = 40$

```
> sigma <- 40
> nTimes <- length(InvTimes0)
> nCSF <- sum(segm==1)
> nGM <- sum(segm==2)
> nWM <- sum(segm==3)
> IRdata <- array(0,c(nTimes,prod(dim(segm))))
> IRdata[,segm==1] <- sqrt(rnorm(nTimes*nCSF,fintCSF,sigma)^2+
+                          rnorm(nTimes*nCSF,0,sigma)^2)
> IRdata[,segm==2] <- sqrt(rnorm(nTimes*nGM,fintGM,sigma)^2+
+                          rnorm(nTimes*nGM,0,sigma)^2)
> IRdata[,segm==3] <- sqrt(rnorm(nTimes*nWM,fintWM,sigma)^2+
+                          rnorm(nTimes*nWM,0,sigma)^2)
> dim(IRdata) <- c(nTimes,dim(segm))
> for(i in 1:9) writeNIfTI(as.nifti(IRdata[i,,,]),
+                          file.path(dataDir,paste0("IR0",i)))
> for(i in 10:nTimes) writeNIfTI(as.nifti(IRdata[i,,,]),
+                          file.path(dataDir,paste0("IR",i)))
```

2 Analysis of IR MRI data

We now illustrate the analysis pipeline for IRMRI data. First we generate an IRdata object

```
> library(qMRI)
> t1Files <- list.files(dataDir,"*.nii.gz",full.names=TRUE)
```

```
> segmFile <- file.path(dataDir0,"Brainweb_segm")
> IRdata <- readIRData(t1Files, InvTimes0, segmFile, sigma=sigma,
+                          L=1, segmCodes=c("CSF","GM","WM"))
```

In a first analysis step parameters S_f and R_f characterizing fluid are obtained from voxel that are classified as CSF using the model

$$\xi(TI; S^f, R_1^f) = |S^f \left(1 - 2\mathrm{e}^{-TI \cdot R_1^f}\right)| \tag{1}$$

with data for inversion time TI distributed as $Rician(\xi(TI; S_f, R_1^f), \sigma)$.

The parameters S_f and R_1^f are assumed not to vary within CSF.

```
> setCores(2) # parallel mode using 2 threads
> IRfluid <- estimateIRfluid(IRdata, method="NLR", verbose=FALSE)
> cat("Estimated parameters Sf:", IRfluid$Sf,
+                             " Rf:", IRfluid$Rf, "\n")

Estimated parameters Sf: 902.0022  Rf: 0.0002862264
```

We here use nonlinear regression instead of the more adequate quasi-likelihood method (`method="QL"`) In the next step we evaluate a mixture model

$$\xi(TI; f, S^f, R_1^f, S^s, R_1^s) = |(1-f)S^f \left(1 - 2\mathrm{e}^{-TI \cdot R_1^f}\right) + fS^s \left(1 - 2\mathrm{e}^{-TI \cdot R_1^s}\right)|, \tag{2}$$

for voxel classified as GM or WM with parameters S_f and R_1^f plugged in.

```
> IRmix <- estimateIRsolid(IRfluid, verbose=FALSE)
```

Parameters S^s and R_1^s characterizing solid material in GM and WM can be assumed to be spatially smooth within the respective tissue types. Parameter f characterizes the proportion of fluid within a voxel. This parameter is difficult to estimate in model 2. We therefor apply an adaptive smoothing procedure within segments characterizing GM and WM to reduce the variance of the estimates of S^s and R_1^s

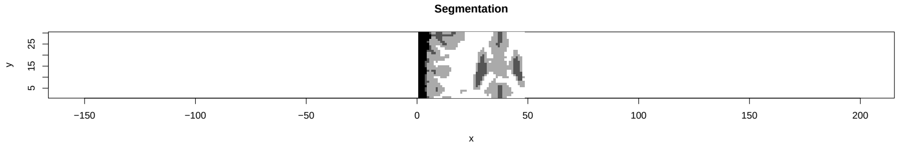
```
> sIRmix <- smoothIRSolid(IRmix, alpha=1e-4, verbose=FALSE,partial=FALSE)
```

and then re-estimate the fluid proportion f

```
> sIRmix <- estimateIRsolidfixed(sIRmix, verbose=FALSE)
```

We shortly illustrate the estimated maps (central slice) that we gain

```
> oldpar <- par(mfrow=c(1,4),mar=c(3,3,3,.5),mgp=c(2,1,0))
> on.exit(par(oldpar))
> library(adimpro)
> rimage(segm[,,2])
> title("Segmentation")
> rimage(sIRmix$Sx[,,2],zlim=c(250,500))
> title("solid intensity map")
> rimage(sIRmix$Rx[,,2],zlim=c(0,.0015))
> title("solid relaxation rate map")
> rimage(sIRmix$fx[,,2],zlim=c(0,.4))
> title("fluid proportion map")
```



All analysis steps can be combined, in this case using quasi-likelihood, simply calling

```
> sIRmix <- estimateIR(IRdata, method="QL")
```