Denoising EEG and MEG Recordings using `icaOcularCorrection`.

Antoine Tremblay
Dalhousie University, Halifax, Nova Scotia, Canada

Abstract

This vignette documents the steps in performing Independent Components Analysis (ICA) based eye-movement correction (HEOG and VEOG) and correction of other known (i.e., recorded) sources of noise (e.g., EMG, ECG, and GSR) in EEG and MEG recordings using package `icaOcularCorrection`. Please note that the vignette is still under construction. Therefore some sections will be empty and the structure of the vignette might change in later versions of the package.

Contents

Introduction

The correction method proposed in this package is largely based on the method described in Flexer, Bauer, Pripfl, & Dorffner (2005). The process of correcting electro- and magneto-encephalographic data (EEG/MEG) begins by running function `icac`, which first performs independent components analysis (ICA) to decompose the data frame into independent components (ICs) using function `fastICA` from the package of the same name (Marchini, Heaton, & Ripley, 2012). It then calculates for each trial the correlation between each IC and each one of the noise signals – there can be one or more, e.g., vertical and horizontal electro-oculograms (VEOG and HEOG), electro-myograms (EMG), electrocardiograms (ECG), galvanic skin responses (GSR), and other noise signals. Subsequently, portions of an IC corresponding to trials at which the correlation between it and a noise signal was at or above threshold (set to 0.4 by default; Flexer et al., 2005, p. 1001) are either zeroed-out in the source matrix, S, or subtracted from the data that was passed to function `icac`.

The user can then identify which ICs correlate with the noise signals the most by looking at the summary of the `icac` object (using function `summary.icac`), the scalp topography of the ICs (using function `topo_ic`), the time courses of the ICs (using functions `plot_tric` and `plot_nic`, and other diagnostic plots. Once these ICs have been identified, they can be completely zeroed-out or subtracted using function `update.icac` and the resulting correction checked using functions `plot_avgba` and `plot_trba`. Some worked-out examples with `R` code are provided in the following sections. Please contact Antoine Tremblay `trea26@gmail.com` to obtain the data to run the examples.

Correcting for Artifacts in EEG Recordings

Ocular (EOG) Artifacts

The data we use here was gathered in the context of a four-character sequence immediate recall experiment in Mandarin Chinese. A total of 432 four-character sequences were divided in blocks of six and presented one at a time on a computer screen for 1.5 seconds. After having seen six sequences, participants were asked to verbally recall as many of them as they could remember. The data used here also includes 40 practice trials that were presented before the 432 experimental ones.

While participants were performing the task, a dense array EEG system consisting of 129-electrode sensors connected to a `Net Amps` amplifier and `Net Station` software (version 4.3; Electrical Geodesics, Inc., Eugene, Oregon) was used for data acquisition at a sampling rate of 250 Hz. An on-line bandpass filter of 0.1-100 Hz was used, and all electrode impedances were maintained below 40 kΩ. Off-line data preprocessing involved the application of a 30 Hz low-pass filter using `NetStation`'s waveform tools. The filtered data were then PARE-corrected average re-referenced and exported to a `.mat` file. Finally, the data were imported into `R` using function `reshape.egi` from package `eRp`.

Running Function `icac`.. After loading in the data and performing some formatting, we run function `icac` with noise signals E14 and E21 (top of left and right eyes), and E126 and E127 (bottom of left and right eyes), as shown below.

```
> library(icaOcularCorrection)
> load("data/mc12.eeg.fil.mat.reshp.rda")
> chan <- colnames(egi129)[1:129]
> #
> res <- icac(x = egi129, channel = chan, noise.sig = c("E14",
+          "E21", "E126", "E127"))
> save(res, file = "models/res.rda", compress = "xz")
```

You'll need at least 8GB RAM to run this and the process will take approximately 70 minutes to complete.

Looking at the Results. Let's now have a look at the first 10 rows of the overall summary.

```
> smry  <- summary(res, print = FALSE)
> save(smry, file = "smryEGI129.rda")
> #
> smry  <- summary(res, ic = 52, print = FALSE)
> save(smry, file = "smryEGI129IC52.rda")
> #
> smry  <- summary(res, ic = 6, print = FALSE)
> save(smry, file = "smryEGI129IC6.rda")
> #
> smry  <- summary(res, ic = 43, print = FALSE)
> save(smry, file = "smryEGI129IC43.rda")
```

```
> load("smryEGI129.rda")
> smry[1:10,]
```

```
    IC NumTrials  MeanCorr
1   52      1339 0.6726820
2    6      1211 0.7114151
3   43      1147 0.8030763
4  112       614 0.5567361
5  115       611 0.5423470
6   48       471 0.5175990
7   78       459 0.5184090
8   84       448 0.5171411
9  111       447 0.5316383
10  41       443 0.5126144
```

Independent components 52, 6, and 43 correlate with channels E11, E21, E126, and E127 above the threshold of 0.4 in more trials than any other independent component (1339, 1211, and 1147 trials, respectively).

```
> load("smryEGI129IC52.rda")
> smry
```

```
  IC NoiseSignal NumTrials  MeanCorr
1 52         E14       365 0.7059010
2 52         E21       356 0.6884662
3 52        E126       300 0.6511167
4 52        E127       318 0.6372274
```

As can be seen from the summary, IC 52 correlated mostly with channel E14 (365 trials). Figure 1 on page 4 plots IC 52 against channel E14 at trials 85 to 104. The blue (at or above threshold) or grey (below threshold) lines are channel E14.

```
> # IC 52 mostly correlates with channel E14
> pdf(file="IC52E14.pdf")
> plot_tric(res, ic = 52, noise.sig = "E14",
+          trials = 85:104, n.win = 21, new.page = FALSE)
> dev.off()
```

We can also use function `plot_nic` to look at an IC and a noise signal at specific trials. In Figure 2 on page 5 we look at at trials 85 to 93.

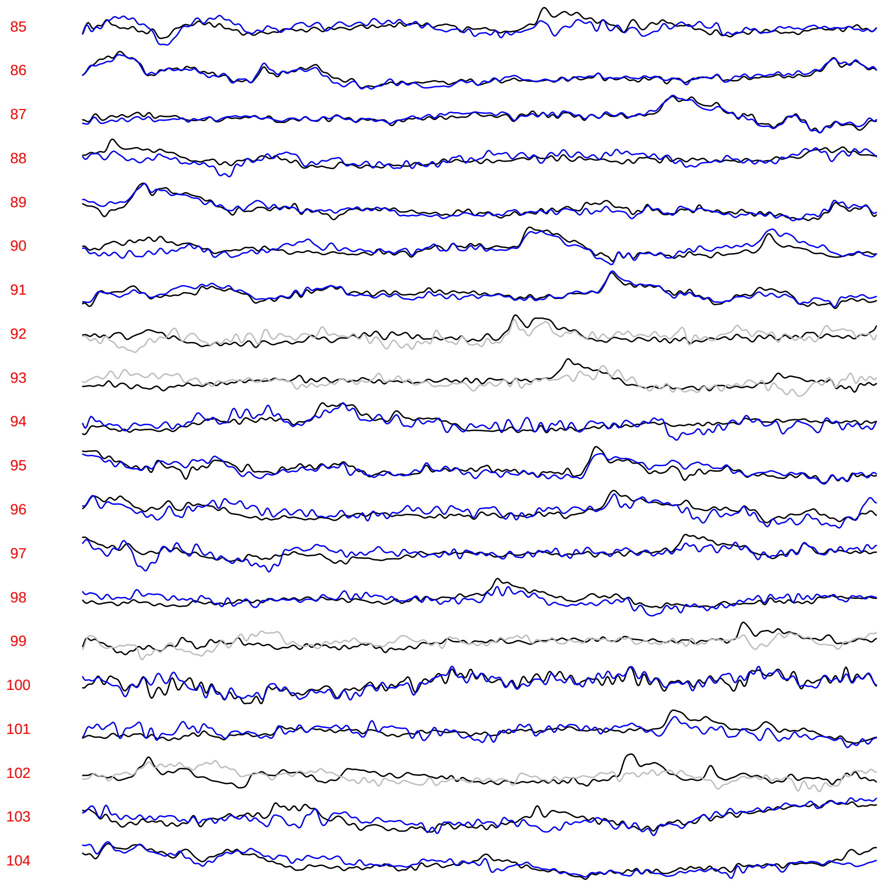

Figure 1. Independent component 52 (black lines) and channel E14 (right bottom eye; blue lines when correlation was at or above threshold; grey lines when it was below threshold) at trials 85 to 104. Positive is plotted up.

```
> pdf("plotNICIC52E14.pdf")
> par(mfrow = c(3, 3))
> for(i in 85:93){
+         plot_nic(x = res, data = egi129, ic = 52,
+                 trial = i, noise.sig = "E14")
+ }
> par(mfrow = c(1, 1))
> dev.off()
```

The activity captured in IC 52 should thus be apparent at the very front of the scalp, about where the eyes are. Figure 3 on page 6 shows that this is the case.

```
> pdf("topomapIC52.pdf")
> topo_ic(x = res, ic = 52, coords = "egi.129")
> dev.off()
```

Let's now have a look at the summary for IC 6.

```
> load("smryEGI129IC6.rda")
> smry
```

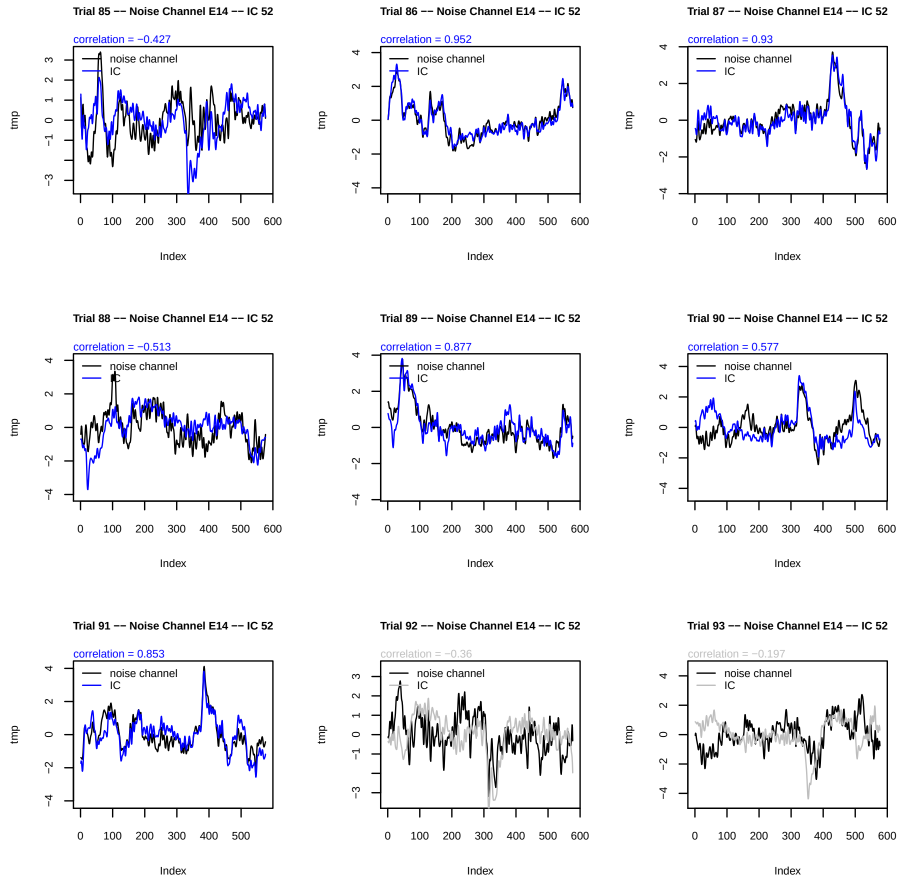

Figure 2. Independent component 52 and channel E14 at trials 85 to 93. The correlation between IC 52 and channel E14 at a specific trial is indicated at the top left of each plot. It is blue if the correlation is at or above threshold, but grey if it is below it. The black line is channel E14 and the blue or grey line is IC 52. Positive is plotted up.

```
  IC NoiseSignal NumTrials  MeanCorr
1  6         E14       305 0.7260039
2  6         E21       305 0.7237632
3  6        E126       299 0.6978538
4  6        E127       302 0.6976372
```

IC 6 correlates mostly with channel E14 (305 trials). We will nevertheless look at IC 6 against noise signal E126 (299 trials).

```
> pdf(file="IC6E126.pdf")
> plot_tric(res, ic = 6, noise.sig = "E126",
+         trials = 253:273, n.win = 21, new.page = FALSE)
> dev.off()
```

Figure 4 on page 7 shows another view of IC 6 and channel E126 at trials 253 to 273.

```
> pdf("plotNICIC6E126.pdf")
> par(mfrow = c(3, 3))
```

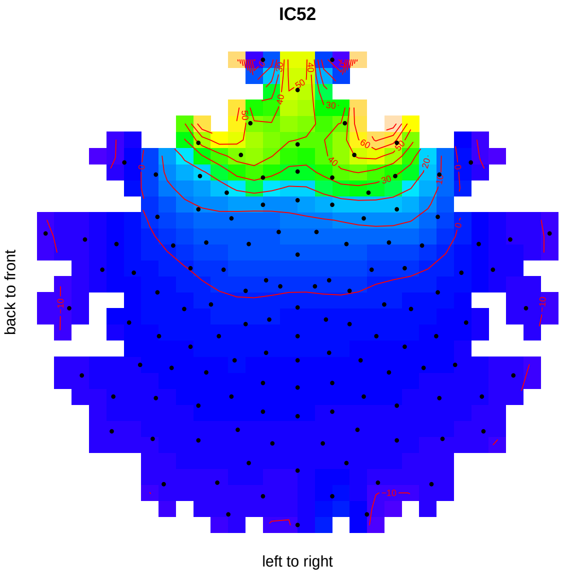

Figure 3. Topographic map of independent component 52. The bottom of the plot corresponds to the back of the head and the top of the plot to the front of the head. Yellow represents positive amplitudes and blue represents negative amplitudes.

```
> for(i in 253:261){
+         plot_nic(x = res, data = egi129, ic = 6,
+                 trial = i, noise.sig = "E126")
+ }
> par(mfrow = c(1, 1))
> dev.off()
```

```
> pdf("topomapIC6")
> topo_ic(x = res, ic = 6, coords = "egi.129")
> dev.off()
```

The centro-frontal location of IC 6 apparent in Figure 6 on page 9 corroborates that it is mostly composed of blinks. Finally, let's have a look at the summary for IC 43.

```
> load("smryEGI129IC43.rda")
> smry

  IC NoiseSignal NumTrials  MeanCorr
1 43         E14       284 0.8114560
2 43         E21       288 0.8150607
3 43        E126       288 0.7863471
4 43        E127       287 0.7995457
```

IC 43 correlates mostly with channel E21 (288 trials).

```
> pdf(file="IC43E21.pdf")
> plot_tric(res, ic = 43, noise.sig = "E21",
+         trials = 1:21, n.win = 21, new.page = FALSE)
> dev.off()
```

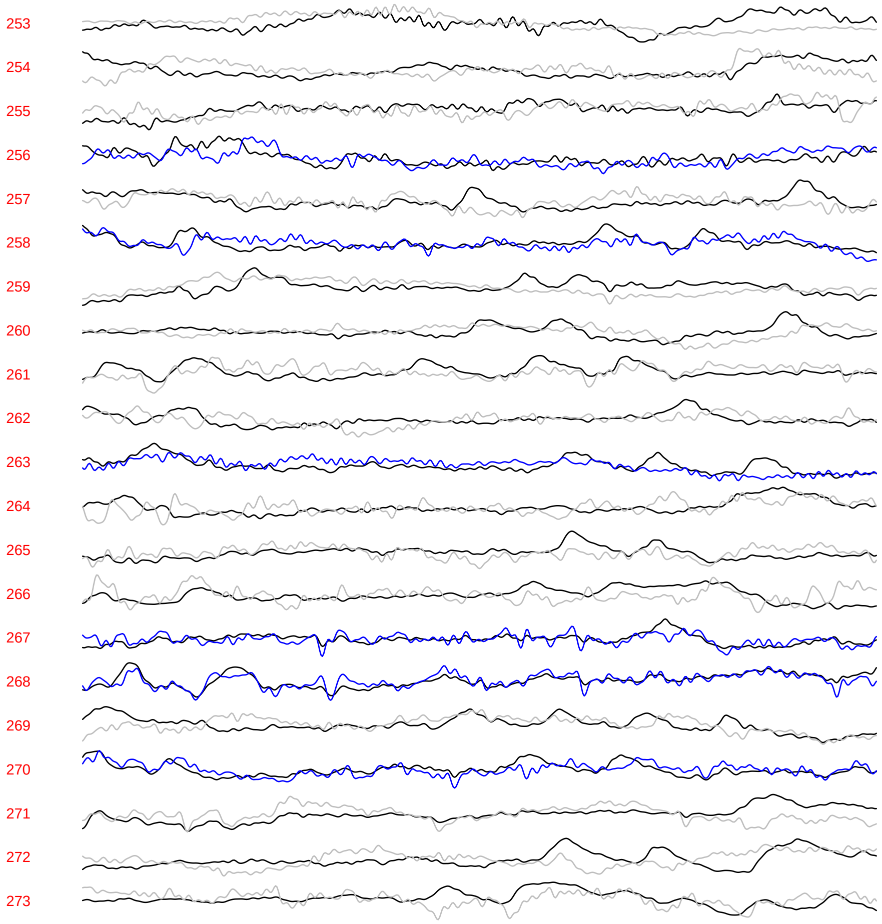

Figure 4. Independent component 6 (black lines) and channel E126 (blue lines when correlation was at or above threshold; grey lines when it was below threshold) at trials 253 to 273. Positive is plotted up.

Figure 8 on page 11 shows another view of IC 43 and channel E21 at trials 1 to 21.

```
> pdf("plotNICIC43E21.pdf")
> par(mfrow = c(3, 3))
> for(i in 1:9){
+         plot_nic(x = res, data = egi129, ic = 43,
+                  trial = i, noise.sig = "E21")
+ }
> par(mfrow = c(1, 1))
> dev.off()
```

```
> pdf("topomapIC43")
> topo_ic(x = res, ic = 43, coords = "egi.129")
> dev.off()
```

It is apparent in Figure 9 on page 12 that the scalp topography of IC 43 is also frontal, thus corroborating that it is mostly composed of blinks.

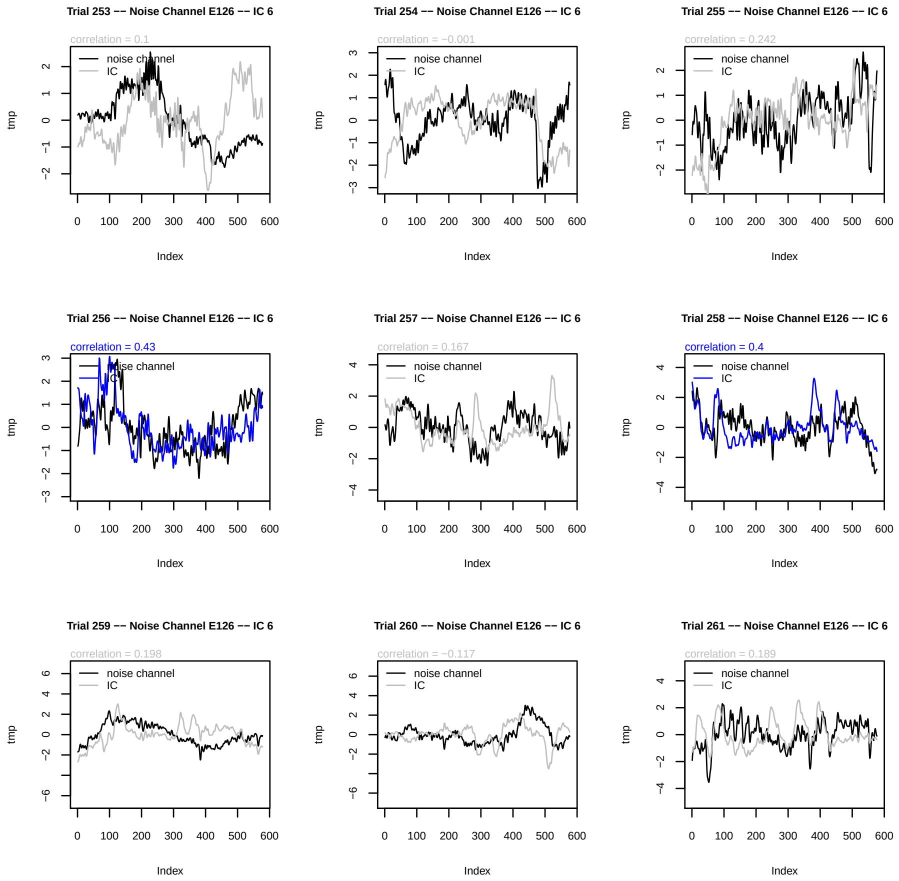

Figure 5. Independent component 6 and channel E126 at trials 253 to 261 The correlation between IC 6 and channel E126 at a specific trial is indicated at the top left of each plot. It is blue if the correlation is at or above threshold, but grey if it is below it. The black line is channel E126 and the blue or grey line is IC 6

Determining which ICs to Completely Zero-out & Updating the `icac` object. Let's see how well the by-trial correction of blinks and eye-movements performed. To do this we will (1) use function `get.peaks` to get peaks of every blink in channel E21 from the uncorrected data, (2) insert event code 777 at these points, (3) grab a 200 ms window of EEG data around each peak, (4) recompute the time column where time t = 0 will be at those peaks, and (5) finally average across each blink through time. We will subsequently re-use these peaks to compute an average of the blinks in the corrected data. Although this process can be a bit time consuming, it will enable us to get an idea of how well the correction performed.

```
> # you'll need library eRp for this.
> library(eRp)
> load("data/mc12.eeg.fil.mat.reshp.rda")
> # get peaks for egi129
> peaks.egi129 <- get.peaks(egi129,"E21",NULL)
> save(peaks.egi129, file = "data/peaks.egi129.rda", compress= "xz")
> #
```

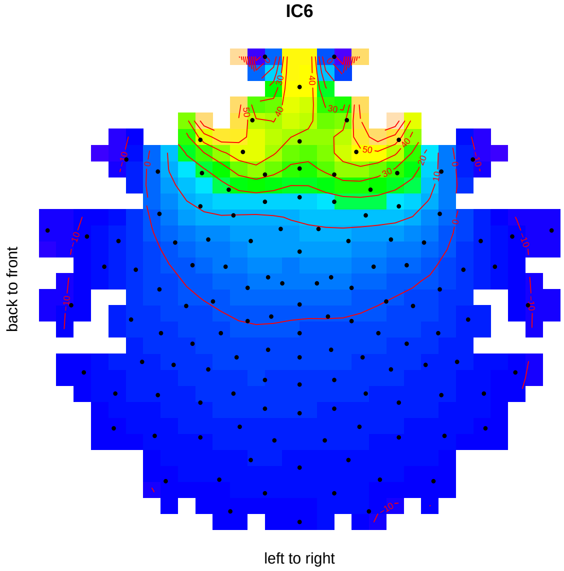

Figure 6. Topographic map of independent component 6. The bottom of the plot corresponds to the back of the head and the top of the plot to the front of the head. Yellow represents positive amplitudes and blue represents negative amplitudes.

```
> # insert event code 777 at each peak
> egi129$EventCode <- as.character(egi129$EventCode)
> pb<-txtProgressBar(min=1,max=length(peaks.egi129),char="=",
+         style=3)
> for(i in 1:length(peaks.egi129)){
+         setTxtProgressBar(pb,i)
+         tmp <- peaks.egi129[[i]]
+         if(!is.na(tmp[1])){
+                 for(j in 1:length(tmp)){
+                         egi129[egi129$Trial==i & egi129$Time==tmp[j],
+                                 "EventCode"] <- "777"
+                 }
+         }
+ }
> close(pb)
> #
> # save event codes to later merge with corrected data frame
> evts <- egi129$EventCode
> save(evts, file = "data/evts.peaks.egi129.rda", compress = "xz")
> #
> # grab a 200 ms window around each peak and
> # put into data frame
> x <- as.numeric(rownames(egi129[egi129$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- egi129[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
+         style = 3)
> for(i in 2:nrow(x)){
```

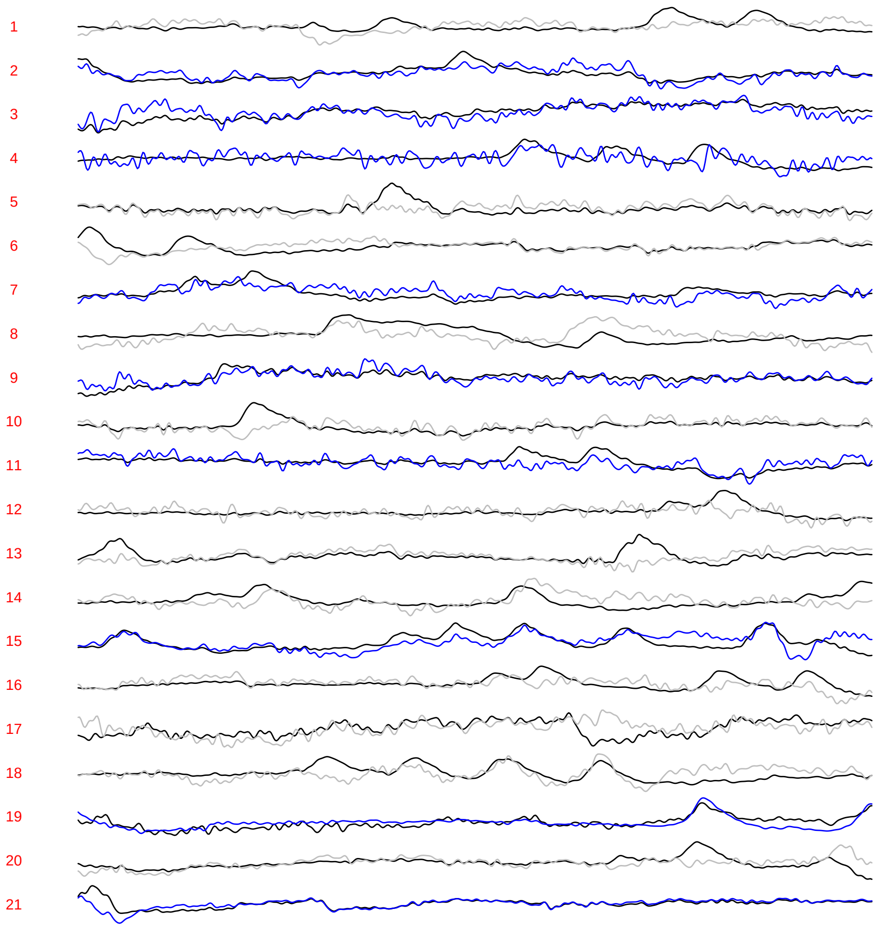

Figure 7. Independent component 43 (black lines) and channel E21 (blue lines when correlation was at or above threshold; grey lines when it was below threshold) at trials 1 to 21. Positive is plotted up.

```
+         setTxtProgressBar(pb, i)
+         tmp1 <- egi129[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
> rownames(tmp) <- 1:nrow(tmp)
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eog.uncor <- tmp
> save(eog.uncor, file = "data/eog.uncor.icaOC.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # compute blink average for corrected data
> datc <- res$data
> datc$EventCode <- evts
> #
> # grab a 200 ms window around each peak and
```

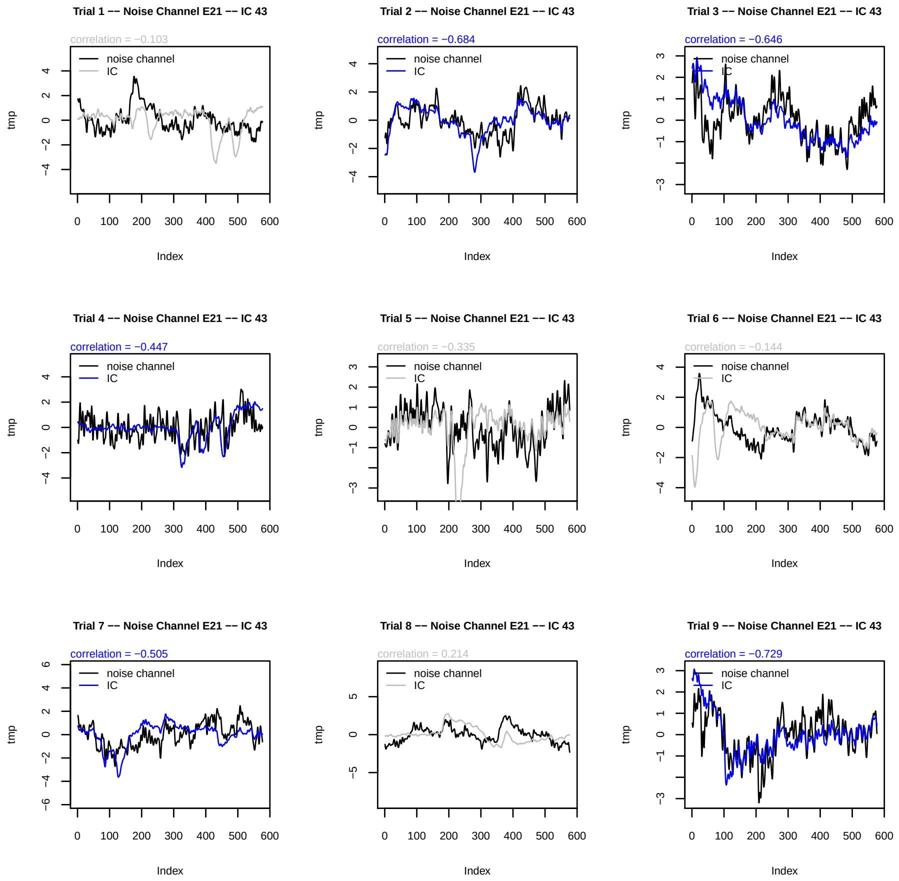

Figure 8. Independent component 43 and channel E21 at trials 1 to 21. The correlation between IC 43 and channel E21 at a specific trial is indicated at the top left of each plot. It is blue if the correlation is at or above threshold, but grey if it is below it. The black line is channel E21 and the blue or grey line is IC 43

```
> # put into data frame
> x <- as.numeric(rownames(datc[datc$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- datc[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
+         style = 3)
> for(i in 2:nrow(x)){
+         setTxtProgressBar(pb, i)
+         tmp1 <- datc[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
```

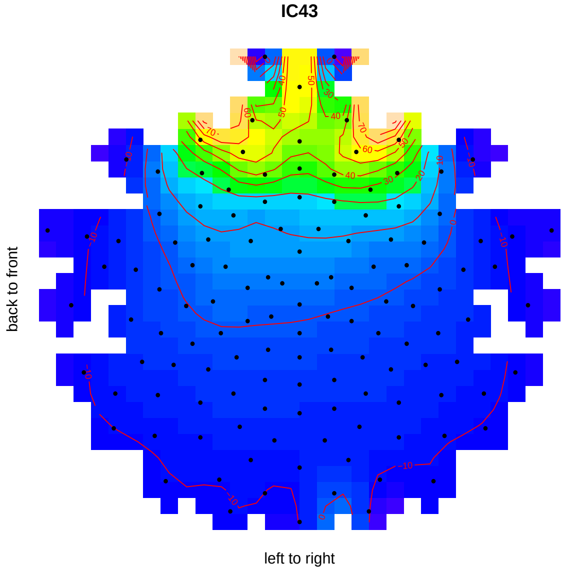

Figure 9. Topographic map of independent component 43. The bottom of the plot corresponds to the back of the head and the top of the plot to the front of the head. Yellow represents positive amplitudes and blue represents negative amplitudes.

```
> rownames(tmp) <- 1:nrow(tmp)
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eog.cor <- tmp
> save(eog.cor, file = "data/eog.cor.icaOC.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # topomap for uncorrected
> avg <- tapply(eog.uncor[,1], eog.uncor$Time, mean)
> chan<-colnames(egi129)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eog.uncor[,chan[i]], eog.uncor$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.dat.uncor<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.uncor<-gam(Amplitude~te(x,y,bs="ts",k=11),dat=avg.dat.uncor)
> #
> # topomap for corrected
> avg <- tapply(eog.cor[,1], eog.cor$Time, mean)
> chan<-colnames(egi129)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
```

```
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eog.cor[,chan[i]], eog.cor$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.dat.cor<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.cor<-gam(Amplitude~te(x,y,bs="ts",k=11),dat=avg.dat.cor)
> #
> # get plotting info
> pi.uncor<-plotGAM(m.uncor,too.far=des("egi.129")$too.far,plot=FALSE)
> pi.cor<-plotGAM(m.cor,too.far=des("egi.129")$too.far,plot=FALSE)
> #
> # waveforms
> avg.uncor <- tapply(eog.uncor$E21, eog.uncor$Time, mean)
> avg.cor <- tapply(eog.cor$E21, eog.cor$Time, mean)
> time <- as.numeric(names(avg.uncor))
> #
> # create plot
> pdf(file = "uncorrectedCorrected0.pdf")
> par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
> # plot waveforms
> plot(time, avg.uncor - min(avg.uncor), type = "l",
+         xlab = "Time (ms)", ylab = "Amplitude",ylim=c(0,400),
+         main = "E21")
> lines(time, avg.cor - min(avg.cor), col = 2)
> legend("topleft", legend = c("blinks in uncorrected data",
+         "blinks in corrected data"), lty = 1, col = 1:2, bty = "n")
> par(mar = c(1, 1, 1, 1))
> zlimit <- range(rbind(pi.uncor$mat, pi.cor$mat), na.rm = TRUE)
> # skip a plotting region
> plot.new()
> # plot uncorrected topomap
> image(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, add = TRUE)
> title(main = "Uncorrected", line = -12.5)
> # plot corrected topomap
> image(pi.cor$xm, pi.cor$xm, pi.cor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.cor$xm, pi.cor$xm, pi.cor$mat, add = TRUE)
> title(main = "Corrected", line = -12.5)
> par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
> dev.off()
```

A graph of the blinks before and after correction is shown in Figure 10. Although the trial-by-trial approach goes a long way in removing blinks from the data, it nevertheless misses a number of trials that contain them. If we look at a plot that shows the number of trials that correlated at or above threshold for each IC, a first break apparent at IC 43 as shown by the red dotted line in Figure 15. We have seen above that some noise channels did not correlate at or above threshold at some trials.

```
> pdf(file="ICNumTrials3.pdf")
> plot(smry$NumTrial, type = "h", xlab="IC", ylab = "Number of Trials",
+         xaxt = "n")
> myat<-pretty(1:nrow(smry), 10)
> myat[1]<-1
> myat<-myat[1:(length(myat)-1)]
> mylab<-smry$IC[myat]
> axis(side = 1, at = myat, labels = mylab, cex.axis = 0.85)
```

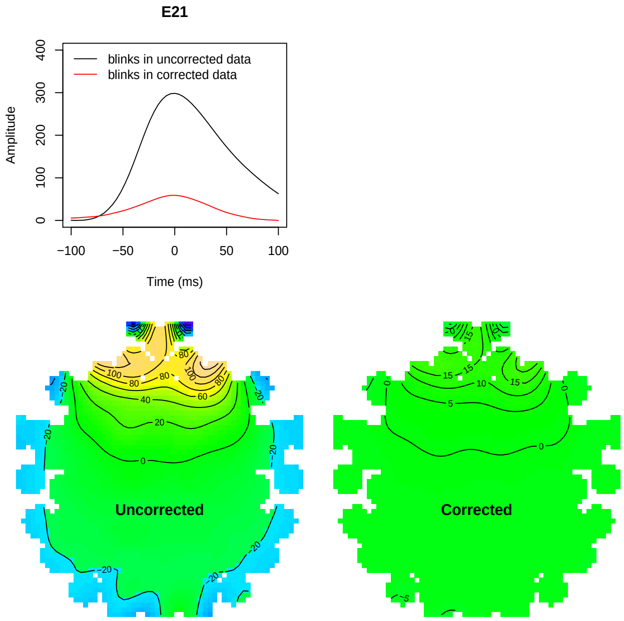

Figure 10. Top left panel: Uncorrected (black) and corrected (red) waveforms at channel E21. Bottom left panel: Topographic map of uncorrected blinks. Bottom righ panel: Topographic map of corrected blinks.

```
> abline(v = 3, col = 2, lty = 3)
> text(x = 4, y = 1000, labels = paste("IC",smry$IC[3]), cex = 0.85, adj = 0)
> dev.off()
```

Let's completely zero-out all the ICs up to that point (i.e., three ICs) and see if the correction improves.

```
> my.what <- list()
> for(i in 1:3){
+         my.what[[i]] <- c(smry$IC[i], "-")
+ }
> res.up <- update(object = res, what = my.what)
> save(res.up, file = "models/res.up.rda", compress = "xz")

> # compute blink average for corrected data
> datc <- res.up$data
> datc$EventCode <- evts
> #
> # grab a 200 ms window around each peak and
> # put into data frame
> x <- as.numeric(rownames(datc[datc$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- datc[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
+         style = 3)
> for(i in 2:nrow(x)){
+         setTxtProgressBar(pb, i)
```

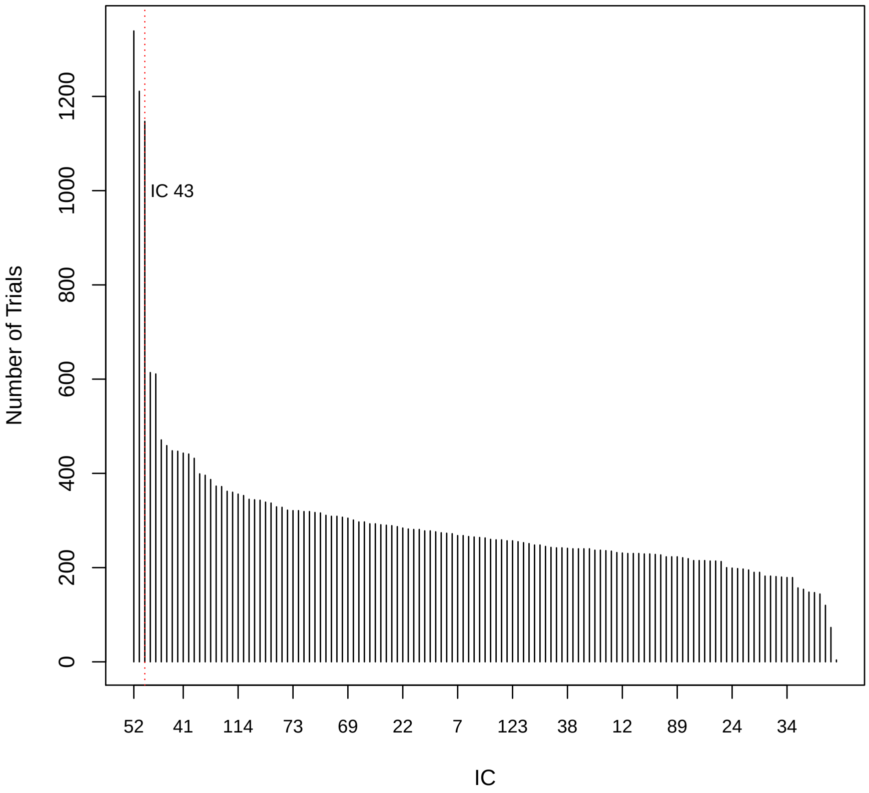

Figure 11. Number of Trials that correlate at or above threshold for each IC. Red dotted line shows where the first break occurs.

```
+         tmp1 <- datc[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
> rownames(tmp) <- 1:nrow(tmp)
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eog.cor2 <- tmp
> save(eog.cor2, file = "data/eog.cor2.icaOC.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # topomap for corrected
> avg <- tapply(eog.cor2[,1], eog.cor2$Time, mean)
> chan<-colnames(egi129)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eog.cor2[,chan[i]], eog.cor2$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.dat.cor2<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.cor2<-gam(Amplitude~te(x,y,bs="ts",k=11),dat=avg.dat.cor2)
> #
> # get plotting info
> pi.cor2<-plotGAM(m.cor2,too.far=des("egi.129")$too.far,plot=FALSE)
> #
> # waveforms
> avg.cor2 <- tapply(eog.cor2$E21, eog.cor2$Time, mean)
> time <- as.numeric(names(avg.uncor))
> #
> # create plot
> pdf(file = "uncorrectedCorrected3.pdf")
> par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
> # plot waveforms
> plot(time, avg.uncor - min(avg.uncor), type = "l",
+         xlab = "Time (ms)", ylab = "Amplitude",ylim=c(0,400),
+         main = "E21")
> lines(time, avg.cor2 - min(avg.cor2), col = 2)
> legend("topleft", legend = c("blinks in uncorrected data",
+         "blinks in corrected data"), lty = 1, col = 1:2, bty = "n")
> par(mar = c(1, 1, 1, 1))
> zlimit <- range(rbind(pi.uncor$mat, pi.cor$mat), na.rm = TRUE)
> # skip a plotting region
> plot.new()
> # plot uncorrected topomap
> image(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, add = TRUE)
> title(main = "Uncorrected", line = -12.5)
> # plot corrected topomap
> image(pi.cor2$xm, pi.cor2$xm, pi.cor2$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.cor2$xm, pi.cor2$xm, pi.cor2$mat, add = TRUE)
> title(main = "Corrected", line = -12.5)
> par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
> dev.off()
```

It is apparent from Figure 12 that the correction is improved. There nevertheless remains a small portion of eye-blinks as evidence by the small bump in the top left panel (red line). A plot of the number of trials that correlated at or above threshold for each IC (Figure 13) shows that there is a second break at IC 115.

```
> pdf(file="ICNumTrials5.pdf")
> plot(smry$NumTrial, type = "h", xlab="IC", ylab = "Number of Trials",
+         xaxt = "n")
> myat<-pretty(1:nrow(smry), 10)
> myat[1]<-1
> myat<-myat[1:(length(myat)-1)]
> mylab<-smry$IC[myat]
> axis(side = 1, at = myat, labels = mylab, cex.axis = 0.85)
> abline(v = 5, col = 2, lty = 3)
> text(x = 6, y = 1000, labels = paste("IC",smry$IC[5]), cex = 0.85, adj = 0)
> dev.off()
```

Let's see if we gain anything by completely zero-out the five top ICs.

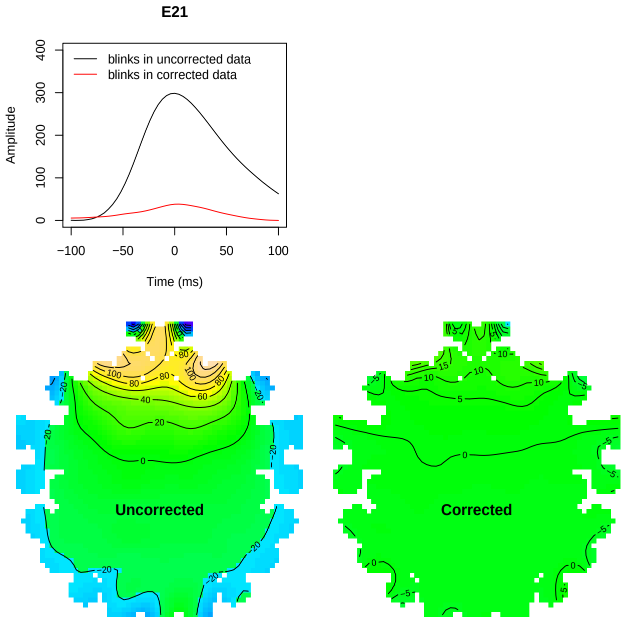

Figure 12. Top left panel: Uncorrected (black) and corrected (red) waveforms at channel E21. Bottom left panel: Topographic map of uncorrected blinks. Bottom righ panel: Topographic map of corrected blinks (top 3 ICs completely zerod-out).

```
> my.what <- list()
> for(i in 1:5){
+         my.what[[i]] <- c(smry$IC[i], "-")
+ }
> res.up2 <- update(object = res, what = my.what)
> save(res.up2, file = "models/res.up2.rda", compress = "xz")

> # compute blink average for corrected data
> datc <- res.up$data
> datc$EventCode <- evts
> #
> # grab a 200 ms window around each peak and
> # put into data frame
> x <- as.numeric(rownames(datc[datc$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- datc[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
+         style = 3)
> for(i in 2:nrow(x)){
+         setTxtProgressBar(pb, i)
+         tmp1 <- datc[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
```

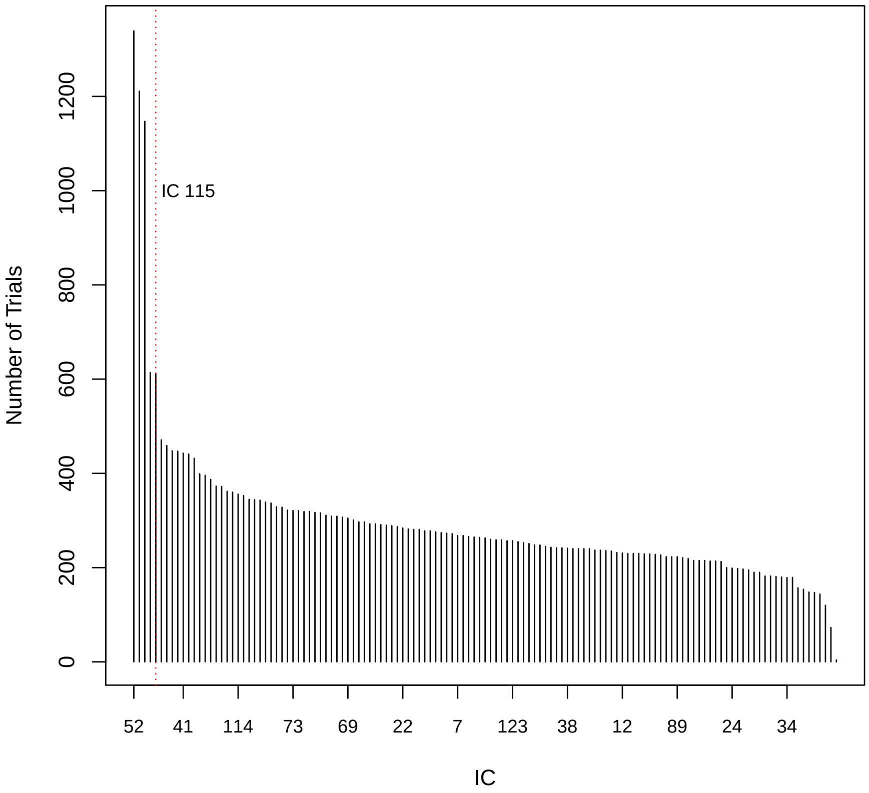

Figure 13. Number of Trials that correlate at or above threshold for each IC. Red dotted line shows where the second break occurs.

```
> rownames(tmp) <- 1:nrow(tmp)
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eog.cor2 <- tmp
> save(eog.cor2, file = "data/eog.cor2.icaOC.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # topomap for corrected
> avg <- tapply(eog.cor2[,1], eog.cor2$Time, mean)
> chan<-colnames(egi129)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eog.cor2[,chan[i]], eog.cor2$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.dat<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.cor2<-gam(Amplitude~te(x,y,k=11),dat=avg.dat)
> #
> # get plotting info
> pi.cor2<-plotGAM(m.cor2,too.far=des("egi.129")$too.far,plot=FALSE)
> #
> # waveforms
> avg.cor2 <- tapply(eog.cor2$E21, eog.cor2$Time, mean)
> time <- as.numeric(names(avg.uncor))
> #
> # create plot
> pdf(file = "uncorrectedCorrected2.pdf")
> par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
> # plot waveforms
> plot(time, avg.uncor - min(avg.uncor), type = "l",
+         xlab = "Time (ms)", ylab = "Amplitude",ylim=c(0,400),
+         main = "E21")
> lines(time, avg.cor - min(avg.cor), col = 2)
> legend("topleft", legend = c("blinks in uncorrected data",
+         "blinks in corrected data"), lty = 1, col = 1:2, bty = "n")
> par(mar = c(1, 1, 1, 1))
> zlimit <- range(rbind(pi.uncor$mat, pi.cor$mat), na.rm = TRUE)
> # skip a plotting region
> plot.new()
> # plot uncorrected topomap
> image(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, add = TRUE)
> title(main = "Uncorrected", line = -12.5)
> # plot corrected topomap
> image(pi.cor2$xm, pi.cor2$xm, pi.cor2$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.cor2$xm, pi.cor2$xm, pi.cor2$mat, add = TRUE)
> title(main = "Corrected", line = -12.5)
> par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
> dev.off()
```

```
> # compute blink average for corrected data
> datc <- res.up2$data
> datc$EventCode <- evts
> #
> # grab a 200 ms window around each peak and
> # put into data frame
> x <- as.numeric(rownames(datc[datc$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- datc[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
+         style = 3)
> for(i in 2:nrow(x)){
+         setTxtProgressBar(pb, i)
+         tmp1 <- datc[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
> rownames(tmp) <- 1:nrow(tmp)
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eog.cor3 <- tmp
> save(eog.cor3, file = "data/eog.cor3.icaOC.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # topomap for corrected
> avg <- tapply(eog.cor3[,1], eog.cor3$Time, mean)
> chan<-colnames(egi129)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eog.cor3[,chan[i]], eog.cor3$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.dat.cor3<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.cor3<-gam(Amplitude~te(x,y,bs="ts",k=11),dat=avg.dat.cor3)
> #
> # get plotting info
> pi.cor3<-plotGAM(m.cor3,too.far=des("egi.129")$too.far,plot=FALSE)
> #
> # waveforms
> avg.cor3 <- tapply(eog.cor3$E21, eog.cor3$Time, mean)
> time <- as.numeric(names(avg.uncor))
> #
> # create plot
> pdf(file = "uncorrectedCorrected5.pdf")
> par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
> # plot waveforms
> plot(time, avg.uncor - min(avg.uncor), type = "l",
+         xlab = "Time (ms)", ylab = "Amplitude",ylim=c(0,400),
+         main = "E21")
> lines(time, avg.cor3 - min(avg.cor3), col = 2)
> legend("topleft", legend = c("blinks in uncorrected data",
+         "blinks in corrected data"), lty = 1, col = 1:2, bty = "n")
> par(mar = c(1, 1, 1, 1))
> zlimit <- range(rbind(pi.uncor$mat, pi.cor$mat), na.rm = TRUE)
> # skip a plotting region
> plot.new()
> # plot uncorrected topomap
> image(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, add = TRUE)
> title(main = "Uncorrected", line = -12.5)
> # plot corrected topomap
> image(pi.cor3$xm, pi.cor3$xm, pi.cor3$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.cor3$xm, pi.cor3$xm, pi.cor3$mat, add = TRUE)
> title(main = "Corrected", line = -12.5)
> par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
> dev.off()
```

Figure 14 shows that the blinks are completely removed.

There is a third break apparent at IC 16 as shown by the red dotted line in Figure 15. Let's completely zero-out all the ICs up to that point (i.e., 12 ICs) and see if anything is gained in terms of the quality of the correction.

```
> pdf(file="ICNumTrials12.pdf")
> plot(smry$NumTrial, type = "h", xlab="IC", ylab = "Number of Trials",
```

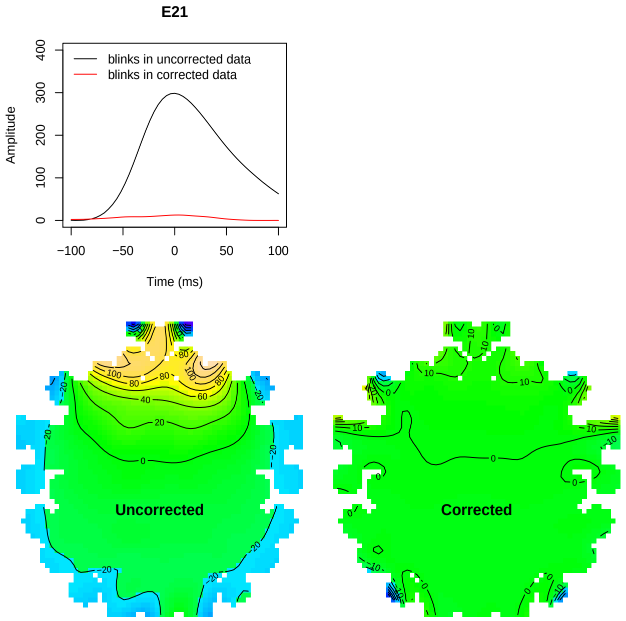

Figure 14. Top left panel: Uncorrected (black) and corrected (red) waveforms at channel E21. Bottom left panel: Topographic map of uncorrected blinks. Bottom righ panel: Topographic map of corrected blinks (top 5 ICs completely zerod-out).

```
+         xaxt = "n")
> myat<-pretty(1:nrow(smry), 10)
> myat[1]<-1
> myat<-myat[1:(length(myat)-1)]
> mylab<-smry$IC[myat]
> axis(side = 1, at = myat, labels = mylab, cex.axis = 0.85)
> abline(v = 12, col = 2, lty = 3)
> text(x = 13, y = 1000, labels = paste("IC",smry$IC[12]), cex = 0.85, adj = 0)
> dev.off()


> my.what <- list()
> for(i in 1:12){
+         my.what[[i]] <- c(smry$IC[i], "-")
+ }
> res.up3 <- update(object = res, what = my.what)
> save(res.up3, file = "models/res.up3.rda", compress = "xz")

> # compute blink average for corrected data
> datc <- res.up3$data
> datc$EventCode <- evts
> #
> # grab a 200 ms window around each peak and
> # put into data frame
> x <- as.numeric(rownames(datc[datc$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- datc[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
```

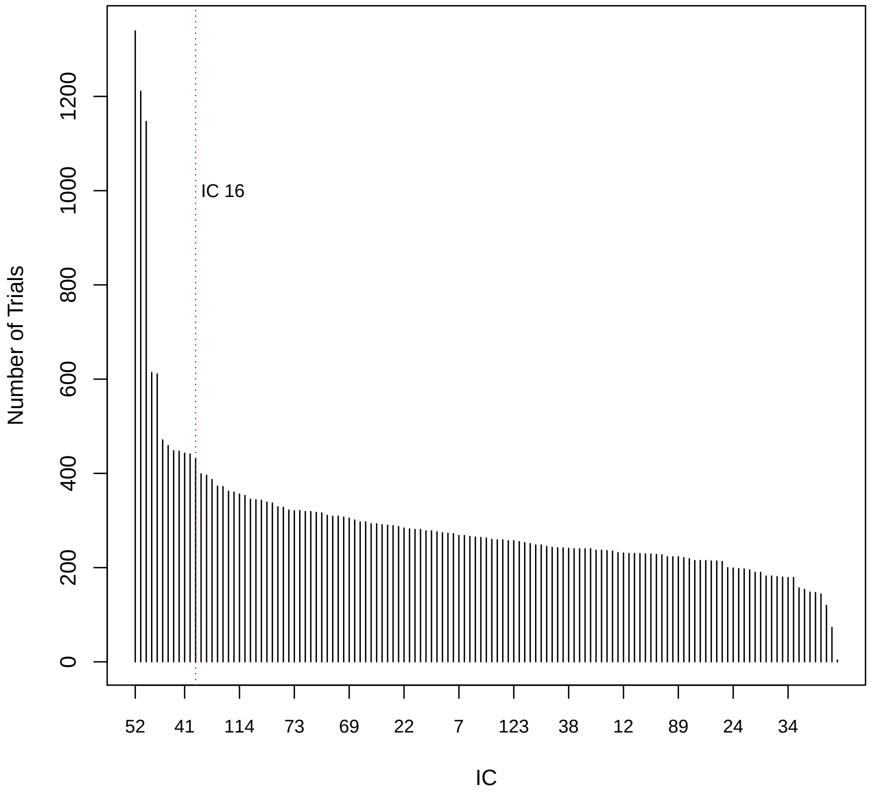

Figure 15. Number of Trials that correlate at or above threshold for each IC. Red dotted line shows where the third break occurs.

```
+         style = 3)
> for(i in 2:nrow(x)){
+         setTxtProgressBar(pb, i)
+         tmp1 <- datc[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
> rownames(tmp) <- 1:nrow(tmp)
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eog.cor4 <- tmp
> save(eog.cor4, file = "data/eog.cor4.icaOC.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # topomap for corrected
> avg <- tapply(eog.cor4[,1], eog.cor4$Time, mean)
> chan<-colnames(egi129)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eog.cor4[,chan[i]], eog.cor4$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.dat.cor4<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.cor4<-gam(Amplitude~te(x,y,bs="ts",k=11),dat=avg.dat.cor4)
> #
> # get plotting info
> pi.cor4<-plotGAM(m.cor4,too.far=des("egi.129")$too.far,plot=FALSE)
> #
> # waveforms
> avg.cor4 <- tapply(eog.cor4$E21, eog.cor4$Time, mean)
> time <- as.numeric(names(avg.uncor))
> #
> # create plot
> pdf(file = "uncorrectedCorrected12.pdf")
> par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
> # plot waveforms
> plot(time, avg.uncor - min(avg.uncor), type = "l",
+         xlab = "Time (ms)", ylab = "Amplitude",ylim=c(0,400),
+         main = "E21")
> lines(time, avg.cor4 - min(avg.cor4), col = 2)
> legend("topleft", legend = c("blinks in uncorrected data",
+          "blinks in corrected data"), lty = 1, col = 1:2, bty = "n")
> par(mar = c(1, 1, 1, 1))
> zlimit <- range(rbind(pi.uncor$mat, pi.cor$mat), na.rm = TRUE)
> # skip a plotting region
> plot.new()
> # plot uncorrected topomap
> image(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, add = TRUE)
> title(main = "Uncorrected", line = -12.5)
> # plot corrected topomap
> image(pi.cor4$xm, pi.cor4$xm, pi.cor4$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.cor4$xm, pi.cor4$xm, pi.cor4$mat, add = TRUE)
> title(main = "Corrected", line = -12.5)
> par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
> dev.off()
> #
> # save info
> save(avg.cor,avg.cor2,avg.cor3,avg.cor4,avg.dat.cor,avg.dat.cor2,
+         avg.dat.cor3,avg.dat.cor4,avg.dat.uncor,avg.uncor,m.cor,m.cor2,
+         m.cor3,m.cor4,m.uncor,pi.cor,pi.cor2,pi.cor3,pi.cor4,pi.uncor,
+         file="models/updating_and_topomaps.rda",compress="xz")
```

It is evident in Figure 16 that we are not gaining anything by completely zeroing-out seven additional ICs, at least regarding blinks. In fact, by removing these extra ICs we might be removing signal of interest.

We will thus settle with zeroing-out portions of ICs that correspond to the trials at which an IC correlated with a noise signal at or above threshold, in addition to the five ICs that correlated with the greatest number of trials. The uncorrected and corrected averages at ten midline electrodes are illustrated in Figure 17.

```
> # create egi129 net mask
```

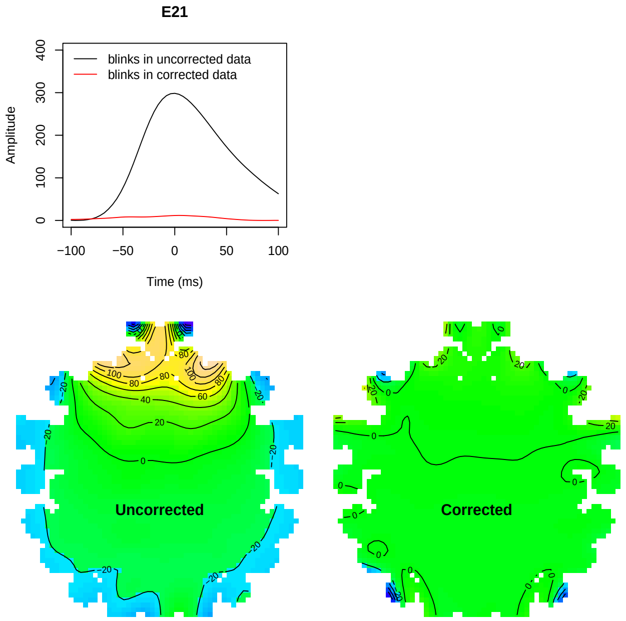

Figure 16. Top left panel: Uncorrected (black) and corrected (red) waveforms at channel E21. Bottom left panel: Topographic map of uncorrected blinks. Bottom righ panel: Topographic map of corrected blinks (top 12 ICs completely zerod-out).

```
> mat <- pi.cor$mat
> mat[!is.na(mat)]<-1
> rownames(mat) <- pi.cor$xm
> colnames(mat) <- pi.cor$ym
> save(mat, file = "data/egi129NetMask.rda", compress = "xz")
> #
> # plot before and after averages
> pdf(file = "beforeAfter5IC.pdf")
> plot_avgba(x = res.up2, data = egi129, channel = c("E17", "E11",
+         "E16", "E6", "E129", "E55", "E62", "E72", "E75", "E81"),
+         ylim = c(-5, 5), new.page = FALSE)
> #
> # add net and channel names
> par(new = TRUE)
> split.screen(c(3,3))
> screen(16)
> par(mar=c(0.5,0.5,0.5,0.5))
> image(as.numeric(rownames(mat)), as.numeric(colnames(mat)),
+         mat,col=rgb(190,190,190,100,maxColorValue=255), ann = FALSE,
+         axes = FALSE)
> coords<-des("egi.129")$cart
> for(i in 1:nrow(coords)){
+         if(coords[i,"Channel"]%in%c("E17", "E11",
+         "E16", "E6", "E129", "E55", "E62", "E72",
+         "E75", "E81")){
+                 mycol <- "red"
+         }else{
+                 mycol <- "black"
+         }
+         text(coords[i,"x"], coords[i,"y"], coords[i, "Channel"], cex = 0.35,
+                 col = mycol)
+ }
```

```
> dev.off()
```

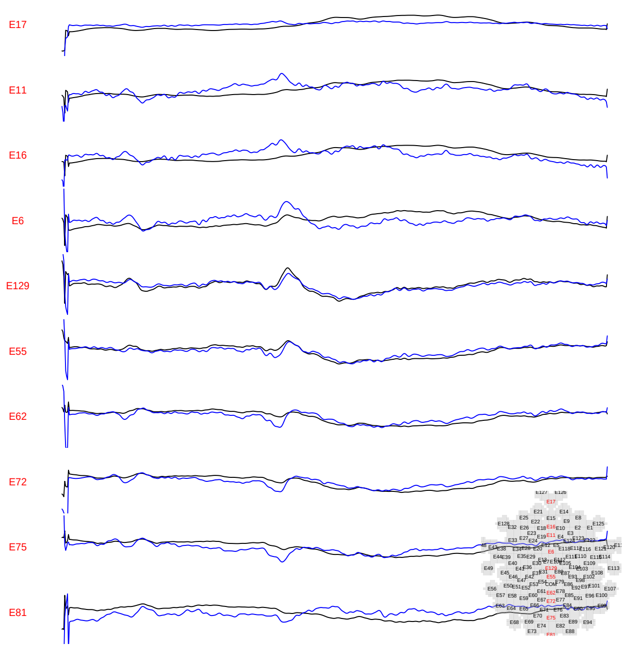

Figure 17. uncorrected averages (black lines) and corrected averages (blue lines) at midline channels (in red font in the scalp inset at the bottom right).

Comparing with EEGLAB's ICA Ocular Artifact Correction Function. Here we compare the difference in correction between EEGLAB and icaOcularCorrection. The comparison is illustrated in Figure 18 on page 30. The two corrections are comparable, but EEGLAB outperforms icaOcularCorrection. It is important to note, however, that the bad channels and trials were removed from the data prior to correcting it with EEGLAB. In another version we will use icaOcularCorrection to correct the data used in the EEGLAB correction.

```
> load("data/mc12postICAallchansEEGLAB.rda")
> dat <- dat[dat$Time >= -200 & dat$Time <= 1500, ]
> rownames(dat) <- 1:nrow(dat)
> eeglab <- list()
> eeglab$data <- dat
> rm(dat)
> gc(TRUE,TRUE)
> class(eeglab) <- "icac"
> #
> load("data/mc12artrejEEGLAB.rda")
> dat <- dat[dat$Time >= -200 & dat$Time <= 1500, ]
> rownames(dat) <- 1:nrow(dat)
> #
> # get blinks
> peaks.eeglab <- get.peaks(dat,"E21",NULL)
> save(peaks.eeglab, file = "data/peaks.eeglab.rda", compress= "xz")
> #
> # insert event code 777 at each peak
> dat$EventCode <- as.character(dat$EventCode)
> pb<-txtProgressBar(min=1,max=length(peaks.eeglab),char="=",
+         style=3)
> for(i in 1:length(peaks.eeglab)){
+         setTxtProgressBar(pb,i)
+         tmp <- peaks.eeglab[[i]]
+         if(!is.na(tmp[1])){
+                 for(j in 1:length(tmp)){
+                         dat[dat$Trial==i & dat$Time==tmp[j],
+                                 "EventCode"] <- "777"
+                 }
+         }
+ }
> close(pb)
> #
> # save event codes to later merge with corrected data frame
> evts <- dat$EventCode
> save(evts, file = "data/evts.peaks.eeglab.rda", compress = "xz")
> #
> # grab a 200 ms window around each peak and
> # put into data frame
> x <- as.numeric(rownames(dat[dat$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- dat[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
+         style = 3)
> for(i in 2:nrow(x)){
+         setTxtProgressBar(pb, i)
+         tmp1 <- dat[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> close(pb)
> rownames(tmp) <- 1:nrow(tmp)
> #
> table(tmp$EventCode)
> # For some reason, two 777 codes appear in some trials.
> # Remove the ones after the first 777.
> tmp<-add.trial2(tmp,list(begin=111111,finish=222222))
> pb <- txtProgressBar(min = 1, max = length(unique(tmp$Trial)),
+         char = "=", style = 3)
> for(i in 1:length(unique(tmp$Trial))){
+         setTxtProgressBar(pb, i)
+         while(length(which(tmp[tmp$Trial==i,"EventCode"]==777))>1){
+                 tmp[as.numeric(rownames(tmp[tmp$Trial==i&tmp$EventCode==777,
+                         ]))[2],"EventCode"]<-0
+         }
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eeglab.uncor <- tmp
> save(eeglab.uncor, file = "data/eeglab.uncor.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # compute blink average for corrected data
> datc <- eeglab$data
> datc$EventCode <- evts
> #
> # grab a 200 ms window around each peak and
>
> # put into data frame
> x <- as.numeric(rownames(datc[datc$EventCode=="777",]))
> x1 <- x - 25
> x2 <- x + 25
> x <- cbind(x1, x2)
> tmp <- datc[x[1, 1]:x[1, 2], ]
> tmp[1,]$EventCode <- "111111"
> tmp[nrow(tmp),]$EventCode <- "222222"
> pb <- txtProgressBar(min = 1, max = nrow(x), char = "=",
+         style = 3)
> for(i in 2:nrow(x)){
+         setTxtProgressBar(pb, i)
+         tmp1 <- datc[x[i, 1]:x[i, 2], ]
+         tmp1[1,]$EventCode <- "111111"
+         tmp1[nrow(tmp1),]$EventCode <- "222222"
+         tmp <- rbind(tmp, tmp1)
+ }
> rownames(tmp) <- 1:nrow(tmp)
> #
> table(tmp$EventCode)
> # For some reason, two 777 codes appear in some trials.
> # Remove the ones after the first 777.
> tmp<-add.trial2(tmp,list(begin=111111,finish=222222))
> pb <- txtProgressBar(min = 1, max = length(unique(tmp$Trial)),
+         char = "=", style = 3)
> for(i in 1:length(unique(tmp$Trial))){
+         setTxtProgressBar(pb, i)
+         while(length(which(tmp[tmp$Trial==i,"EventCode"]==777))>1){
+                 tmp[as.numeric(rownames(tmp[tmp$Trial==i&tmp$EventCode==777,
+                         ]))[2],"EventCode"]<-0
+         }
+ }
> close(pb)
> #
> # reset time with t = 0 at event code 777
> tmp <- add.time2(x = tmp, markers = list(begin = "111111",
+      ref = "777", finish = "222222"), sampling.rate = 250)
> eeglab.cor <- tmp
> save(eeglab.cor, file = "data/eeglab.cor.rda", compress = "xz")
> rm(tmp); gc(TRUE, TRUE)
> #
> # topomap for uncorrected
> avg <- tapply(eeglab.uncor[,1], eeglab.uncor$Time, mean)
> chan<-colnames(dat)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eeglab.uncor[,chan[i]], eeglab.uncor$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.eeglab.uncor<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.uncor<-gam(Amplitude~te(x,y,bs="ts",k=11),dat=avg.eeglab.uncor)
> #
> # topomap for corrected
> avg <- tapply(eeglab.cor[,1], eeglab.cor$Time, mean)
> chan<-colnames(dat)[1:129]
> avg.dat<-data.frame(Time=as.numeric(names(avg)),Amplitude=avg,Channel=chan[1])
> pb <- txtProgressBar(min = 1, max = length(chan)-1, char = "=",
+         style = 3)
> for(i in 2:length(chan)){
+         setTxtProgressBar(pb, i)
+         avg <- tapply(eeglab.cor[,chan[i]], eeglab.cor$Time, mean)
+         avg.dat<-rbind(avg.dat,data.frame(Time=as.numeric(names(avg)),
+                 Amplitude=avg,Channel=chan[i]))
+ }
> close(pb)
> avg.dat$Channel<-as.factor(avg.dat$Channel)
> coords<-des("egi.129")$cart
> avg.eeglab.cor<-merge(avg.dat,coords[,c("x","y","Channel")],by="Channel")
> m.cor<-gam(Amplitude~te(x,y,bs="ts",k=11),dat=avg.eeglab.cor)
> #
> # get plotting info
> pi.eeglab.uncor<-plotGAM(m.uncor,too.far=des("egi.129")$too.far,plot=FALSE)
> pi.eeglab.cor<-plotGAM(m.cor,too.far=des("egi.129")$too.far,plot=FALSE)
> #
> # waveforms
> avg.uncor <- tapply(eeglab.uncor$E21, eeglab.uncor$Time, mean)
> avg.cor <- tapply(eeglab.cor$E21, eeglab.cor$Time, mean)
> time <- as.numeric(names(avg.uncor))
> #
> # create plot
> pdf(file = "EEGLABuncorrectedCorrected.pdf")
> par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
> # plot waveforms
> plot(time, avg.uncor - min(avg.uncor), type = "l",
+         xlab = "Time (ms)", ylab = "Amplitude",ylim=c(0,500),
+         main = "E21")
> lines(time, avg.cor - min(avg.cor), col = 2)
> legend("topleft", legend = c("blinks in uncorrected data",
+         "blinks in corrected data"), lty = 1, col = 1:2, bty = "n")
> par(mar = c(1, 1, 1, 1))
> zlimit <- range(rbind(pi.eeglab.uncor$mat, pi.eeglab.cor$mat), na.rm = TRUE)
> # skip a plotting region
> plot.new()
> # plot uncorrected topomap
> image(pi.eeglab.uncor$xm, pi.eeglab.uncor$xm, pi.eeglab.uncor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.eeglab.uncor$xm, pi.eeglab.uncor$xm, pi.eeglab.uncor$mat, add = TRUE)
> title(main = "Uncorrected", line = -12.5)
> # plot corrected topomap
> image(pi.eeglab.cor$xm, pi.eeglab.cor$xm, pi.eeglab.cor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.eeglab.cor$xm, pi.eeglab.cor$xm, pi.eeglab.cor$mat, add = TRUE)
> title(main = "Corrected", line = -12.5)
> par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
> dev.off()
> #
> save(avg.eeglab.cor,avg.eeglab.uncor,eeglab.cor,eeglab.uncor,
+         pi.eeglab.cor,pi.eeglab.uncor,m.uncor,m.cor,
+         file="models/eeglab_correction.rda",compress="xz")
```

```
> pdf(file = "EEGLABvsICAocularCorrection.pdf")
> par(mfrow = c(2, 2), mar = c(5.1, 4.1, 4.1, 2.1))
> # plot waveforms
> avg.uncor <- tapply(eeglab.uncor$E21, eeglab.uncor$Time, mean)
> avg.cor <- tapply(eeglab.cor$E21, eeglab.cor$Time, mean)
> time <- as.numeric(names(avg.uncor))
> plot(time[1:51], avg.uncor[1:51] - min(avg.uncor[1:51]), type = "l",
+         xlab = "Time (ms)", ylab = "Amplitude",ylim=c(0,600),
+         main = "E21")
> lines(time[1:51], avg.cor[1:51] - min(avg.cor[1:51]), col = 2)
> avg.uncor <- tapply(eog.uncor$E21, eog.uncor$Time, mean)
> avg.cor <- tapply(eog.cor3$E21, eog.cor3$Time, mean)
> time <- as.numeric(names(avg.uncor))
> lines(time, avg.uncor - min(avg.uncor), col = 3)
> lines(time, avg.cor - min(avg.cor), col = 4)
> legend("topleft", legend = c("EEGLAB -- blinks in uncorrected data",
+         "EEGLAB -- blinks in corrected data",
+         "icOC -- blinks in uncorrected data",
+         "icOc -- blinks in corrected data"),
+          lty = 1, col = 1:4, bty = "n",cex=0.85)
> par(mar = c(1, 1, 1, 1))
> zlimit <- range(rbind(pi.uncor$mat, pi.cor3$mat,
+         pi.eeglab.cor$mat), na.rm = TRUE)
> # uncorrected topomap
> image(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.uncor$xm, pi.uncor$xm, pi.uncor$mat, add = TRUE)
> title(main = "icaOcularCorrection \n corrected", line = -12.5)
> #
> # icaOcularCorrection -- corrected topomap
> image(pi.cor3$xm, pi.cor3$xm, pi.cor3$mat, col = topo.colors(100),
+         xlab = "", ylab = "", zlim = zlimit, axes = FALSE)
> contour(pi.cor3$xm, pi.cor3$xm, pi.cor3$mat, add = TRUE)
> title(main = "icaOcularCorrection \n corrected", line = -12.5)
> #
> # EEGLAB --  corrected topomap
> image(pi.eeglab.cor$xm, pi.eeglab.cor$xm, pi.eeglab.cor$mat,
+         col = topo.colors(100), xlab = "", ylab = "", zlim = zlimit,
+         axes = FALSE)
> contour(pi.eeglab.cor$xm, pi.eeglab.cor$xm, pi.eeglab.cor$mat,
+         add = TRUE)
> title(main = "EEGLAB \n Corrected", line = -12.5)
> par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
> dev.off()
```

Galvanic Skin Response Noise

Coming soon.

Correcting for Artifacts in MEG Recordings

Coming soon.

Ocular (EOG) Artifacts

Coming soon.

Muscle (EMG) Artifacts

Coming soon.

Heart Beat (ECG/EKG) Artifacts

Coming soon.

Environmental Noise

Coming soon.

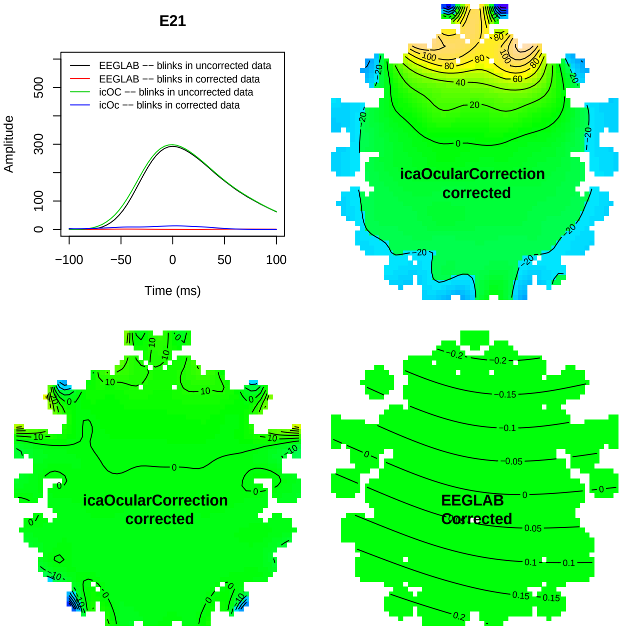

Figure 18. Top left panel: `EEGLAB` uncorrected (121 channels and 379 trials; black line), `EEGLAB` corrected (red), `icaOcularCorrection` uncorrected (129 channels and 472 trials; green line), and `icaOcularCorrection` corrected (blue line) waveforms at channel E21. Top right panel: Topographic map of uncorrected blinks. Bottom left panel: Topographic map of `icaOcularCorrection` corrected blinks (some trials + top 5 ICs completely zerod-out). Bottom righ panel: Topographic map of `EEGLAB` corrected blinks.

References

Flexer, A., Bauer, H., Pripfl, J., & Dorffner, G. (2005). Using ICA for removal of ocular artifacts in EEG recorded from blind subjects. *Neural Networks*, *18*, 998–1005.

Marchini, J., Heaton, C., & Ripley, B. (2012). *fastICA: FastICA Algorithms to perform ICA and Projection Pursuit* [Computer software manual]. Available from http://CRAN.R-project.org/package=fastICA (R package version 1.1-16).