July 13, 2016

The results below are generated from an R script.

```
# diagnostics.R
#
# functions to help diagnose the output of MCMC routines.
```

```
mcmc.diag.plot <- function(chain) {

  # mcmc.diag.plot - Generate output plots to help assess the convergence
  #                  of MCMC output.
  # Inputs:
  #   chain  - (list) containing output data from an MCMC, including
  #   theta  - (array) n * ndim array of posterior samples
  #             n samples of ndim vectors of parameters
  #   method - (string) name of MCMC method used
  #   nwalkers - number of walkers used (if method = 'gw.mcmc')
  #
  # Value
  #   none
  #
  # Generate plots of chain traces (variable values vs. iteration)
  # auto-correlation plots and 1D densities (histograms).
  # History:
  #  14/04/16 - First working version
  #
  # Simon Vaughan, University of Leicester
  # Copyright (C) 2016 Simon Vaughan

  # check the input arguments
  if (missing(chain)) stop('Must specify chain input.')
  if (!"theta" %in% names(chain)) stop('** chain input list is missing theta.')

  m <- rbind(c(1, 1), c(2, 3))
  layout(m)
  par(mar = c(5, 5, 2, 1))

  # total length of chains
  n <- length(chain$theta[,1])

  # dimensions of density / no. variables
  ndim <- NCOL(chain$theta)

  nchains <- 1
  # no. walkers if using ensemble method
```

```
if (chain$method == 'gw.mcmc') {
  nchains <- chain$nwalkers
}
if (chain$method == 'mh.mcmc') {
  nchains <- chain$nchains
}

# total no. cycles
ncycles <- floor(n / nchains)

# do the columns of theta come with names?
if (is.null(colnames(chain$theta))) {
  names <- paste("parameter", 1:ndim)
} else {
  names <- colnames(chain$theta)
}

# prepare an array for output
zeroACF <- array(NA, dim = ndim)

# ----------------------------------------
# trace plots of walkers

if (nchains > 50) {
  walker.sample <- sample(1:nchains, 50)
} else {
  walker.sample <- 1:nchains
}

t <- 1:ncycles

for (i in 1:ndim) {

  # ----------------------------------------
  # Trace plot - for each walker plot theta[j] vs. iteration

  x <- chain$theta[,i]
  dim(x) <- c(ncycles, nchains)
  x.mean <- rowMeans(x)

  plot(x[,1], bty = "n", main = paste(names[i], "- trace"), type = "n",
       xlim = c(0, ncycles), ylim=range(x),
       xlab = "iteration", ylab = names[i])

  n.col <- length(walker.sample)
  k <- 0
  for (j in walker.sample) {
    k <- k + 1
    lines(t, x[,j], col=rainbow(n.col)[k], type = "l")
  }
  lines(t, x.mean, lwd = 3)

  # ----------------------------------------
```

```
    # Auto-correlation functions (ACFs)
    # Compute the mean of the ACFs of each walkers, and
    # the ACF of the mean of all the walkers.

    lag.max <- ncycles

    for (j in 1:nchains) {
      acf.i <- acf(x[,j], lag.max = lag.max, plot = FALSE)
      if (j == 1) acf.j <- rep(0, length(acf.i$lag))
      acf.j <- acf.j + acf.i$acf
    }
    acf.j <- acf.j / nchains
    lag.j <- acf.i$lag
    acf.mean <- acf(x.mean, lag.max = lag.max, plot = FALSE)

    plot(lag.j, acf.j, type = "l", bty = "n", ylim=c(-1,1), main = "ACF",
         xlab = "lag", ylab = "ACF")
    abline(h = 0, lty=2)
    lines(lag.j, acf.mean$acf, lwd=3)

    sign.change <- diff(sign(acf.mean$acf))
    if (!all(sign.change == 0)) zeroACF[i] <- min(which(sign.change != 0))

    # ----------------------------------------
    # 1D histogram

    if (exists('plot.dist')) {
      plot.dist(x, xlim = range(x), fill.col = "steelblue3", xlab = names[i],
                main = "density", breaks = 60)
      axis(1)
    } else {
      hist(x, breaks = 60, main = "density", border = NA,
           col = "steelblue3", xlab = names[i])
    }

  }
}
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 3.2.2 (2015-08-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 8 x64 (build 9200)
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252  LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252 LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
```

```
## other attached packages:
## [1] knitr_1.12.3
##
## loaded via a namespace (and not attached):
## [1] magrittr_1.5  formatR_1.2.1 tools_3.2.2   stringi_1.0-1 highr_0.5.1   stringr_1.0.0
## [7] evaluate_0.8
```

```
Sys.time()
```

```
## [1] "2016-07-13 08:47:39 BST"
```