Mixture Analysis of the Galaxy Data Using the Package mixAK

Arnošt Komárek

February 20, 2011

This document supplements a paper Komárek (2009) and shows an analysis of the Galaxy data introduced in the context of mixture modelling by Roeder (1990) using the R package mixAK.

1 Introduction

• Due to the fact that some code (especially MCMC) is time consuming, some code chunks found in this vignette are not run when compiling the package. You should set the variable RUN.TIMECONSUMING.CODE to TRUE to run full MCMC and related code.

• Having run full MCMC and related code, setting the variable RUN.ALLOUT to TRUE will cause that all output shown in this vignette is re-created and not taken from previously computed results.

• Results based on full MCMC runs are stored (chains excluded) in Galaxy-Result.RData in ./RESULT_OBJ directory and are used to create majority of this vignette at the time of compilation of the package.

R⇒ Setting variables RUN.ALLOUT and RUN.TIMECONSUMING.CODE.

```
> RUN.TIMECONSUMING.CODE <- FALSE
> RUN.ALLOUT <- FALSE
```

R⇒ Directory to store postscript files with figures. Figures which require chains are stored in FIGKEEPDIR directory all other figures are stored in FIGDIR directory.

```
> FIGDIR <- "./figures/"
> FIGKEEPDIR <- "./figuresKeep/"
```

R⇒ Directories with results computed in past. Objects with chains will be stored in directory specified by variable RESULTDIR. All other objects will be stored in directory RESULT2DIR.

```
> RESULTDIR <- "/home/komarek/RESULT_OBJ/mixAK-Galaxy-S081115/"
> RESULT2DIR <- "./RESULT_OBJ/"      ### available in package as /inst/doc/RESULT_OBJ
```

R⇒ Display options.

```
> options(width = 80)
```

R⇒ Load results computed in past.

```
> if (RUN.ALLOUT){
+   load(paste(RESULT2DIR, "Galaxy-Result.RData", sep=""))
+      ## contains RJModel2 (without chains), FixModel2 (without chains),
+      ##          PDensRJ2, PDensFix2
+   load(paste(RESULTDIR, "Galaxy-RJ2.RData", sep=""))
+      ## contains RJModel2 (contains chains as well)
+ }else{
+   load(paste(RESULT2DIR, "Galaxy-Result.RData", sep=""))
+      ## contains RJModel2 (without chains), FixModel2 (without chains),
+      ##          PDensRJ2, PDensFix2
+ }
```

R⇒ Load the package mixAK and package coda which will be used to perform some basic convergence diagnostics.

```
> library("mixAK")
> library("coda")
```

2 Exploration of the data

R⇒ The data are read and summarized as follows.

```
> data("Galaxy", package = "mixAK")
> summary(Galaxy)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  9.172  19.530  20.830  20.830  23.130  34.280
```

R⇒ Additionally, Figure 1 shows the histogram of the data.

```
> postscript(paste(FIGDIR, "figGalaxy01.ps", sep=""), width=6, height=6,
+            horizontal=FALSE)
> par(mfrow=c(1, 1), bty="n")
> hist(Galaxy, prob=TRUE, col="sandybrown", breaks=seq(7, 37, by=0.5),
+      xlab="Velocity (km/sec)", ylab="Density", main="")
> dev.off()
```

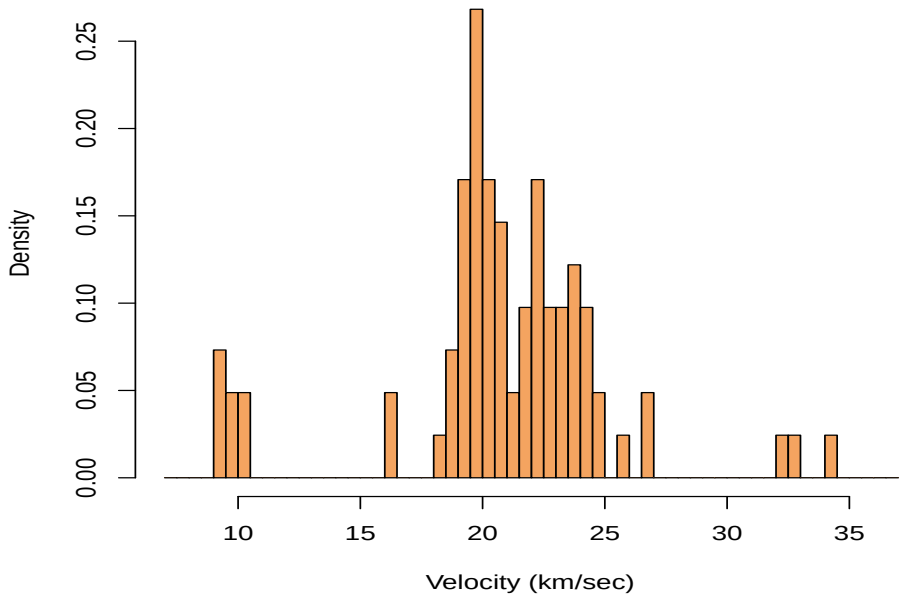

Figure 1: Histogram of Galaxy data.

3 Preparation of the MCMC

R⇒ Length of the MCMC simulation for all models in this document (burn-in of 100 000 iterations, additional 500 000 iterations are kept for the inference, thinning 1:10):

```
> nMCMC <- c(burn=100000, keep=500000, thin=10, info=10000)
```

R⇒ Grid of values where we evaluate and subsequently plot the predictive density for all models in this document:

```
> ygrid <- seq(5, 40, length = 500)
```

4 Model with a number of components estimated using the RJ-MCMC

4.1 Specification of the prior distributions and MCMC simulation

R⇒ The minimal specification of the prior distribution and running RJ-MCMC with default values for all prior parameters:

```
> RJPrior1 <- list(priorK="uniform", Kmax=30)
> if (RUN.TIMECONSUMING.CODE){
+   RJModel1 <- NMixMCMC(y0=Galaxy, prior=RJPrior1, nMCMC=nMCMC,
+                        scale=list(shift=0, scale=1), PED=TRUE)
+ }
```

R⇒ The prior distribution for the function NMixMCMC was the same as with

```
> RJPrior1 <- list(priorK="uniform", Kmax=30,
+                  delta=1,
+                  priormuQ="independentC", xi=21.7255, D=630.3614,
+                  zeta=2, g=0.2, h=0.01586391)
```

In the following, we will use the same prior hyperparameters and tuning parameters as in Richardson and Green (1997) when analyzing this dataset. Note that our prior for the mixture inverse variances is parametrized in terms of the Wishart distribution whereas Gamma distribution is used in Richardson and Green (1997). Hence our $\zeta = 2 \cdot 2$ corresponds to $\alpha = 2$ in [RG] and our $h = 0.016/2$ corresponds to $h = 0.016$ in [RG].

R⇒ Prior distribution of Richardson and Green (1997):

```
> RJPrior2 <- list(priorK="uniform", Kmax=30,
+                  delta=1,
+                  priormuQ="independentC", xi=21.73, D=630.5121,
+                  zeta=2*2, g=0.2, h=0.016/2)
```

R⇒ Parameters to tune RJ-MCMC:

```
> parRJMCMC2 <- list(par.u1=c(2, 2),
+                    par.u2=c(2, 2),
+                    par.u3=c(1, 1))
```

R⇒ Running the MCMC simulation.

R⇒ Two chains will be generated since the argument PED is set to TRUE (output is shown from MCMC simulation performed by author):

```
> if (RUN.TIMECONSUMING.CODE){
+   set.seed(770328)
+   RJModel2 <- NMixMCMC(y0=Galaxy, prior=RJPrior2, RJMCMC=parRJMCMC2,
+                        nMCMC=nMCMC, scale=list(shift=0, scale=1), PED=TRUE)
+ }


Chain number 1
==============
MCMC sampling started on Tue Nov 11 12:11:23 2008.
Burn-in iteration 100000
Iteration 600000
MCMC sampling finished on Tue Nov 11 12:22:08 2008.

Chain number 2
==============
MCMC sampling started on Tue Nov 11 12:22:09 2008.
Burn-in iteration 100000
Iteration 600000
MCMC sampling finished on Tue Nov 11 12:32:55 2008.

Computation of penalized expected deviance started on Tue Nov 11 12:32:57 2008.
Computation of penalized expected deviance finished on Tue Nov 11 12:36:54 2008.
```

R⇒ Information concerning the acceptance rates of different move types (separately for each chain):

```
> print(RJModel2[[1]]$moves)


                  Performed Accepted Proportion accepted (%)
Gibbs with fixed K   5000000  5000000                  100.00
Split                2502857   365907                   14.62
Combine              2497143   384026                   15.38
Birth                2501161   443183                   17.72
Death                2498839   425063                   17.01


> print(RJModel2[[2]]$moves)


                  Performed Accepted Proportion accepted (%)
Gibbs with fixed K   5000000  5000000                  100.00
Split                2500622   365327                   14.61
Combine              2499378   381750                   15.27
Birth                2502545   440529                   17.60
Death                2497455   424109                   16.98
```

4.2 Posterior inference

R⇒ Basic posterior summary of the fitted model:

```
> print(RJModel2)
```

```
     Normal mixture with at most 30 components estimated using RJ-MCMC
     ==================================================================
Posterior distribution of K:
----------------------------
               1        2        3        4       5        6        7        8
Chain 1 0.000136 0.004636 0.035326 0.113576 0.21326 0.245938 0.185912 0.108304
Chain 2 0.000136 0.004774 0.036136 0.114488 0.21342 0.246008 0.185468 0.108172
               9       10       11       12       13       14       15      16
Chain 1 0.054128 0.023652 0.009674 0.003634 0.001196 0.000422 0.000140 5.8e-05
Chain 2 0.053236 0.023544 0.009230 0.003478 0.001356 0.000394 0.000108 3.4e-05
            17    18    19
Chain 1 4.0e-06 2e-06 2e-06
Chain 2 1.4e-05 4e-06 0e+00

Posterior summary statistics for moments of mixture for original data:
-----------------------------------------------------------------------
Mean:
            Mean  Std.Dev.      Min.     2.5%  1st Qu.   Median  3rd Qu.
Chain 1 20.85665 0.7665924  1.508469 19.65970 20.49000 20.86112 21.22509
Chain 2 20.85673 0.7699914 -6.239395 19.65688 20.49049 20.85974 21.22313
          97.5%     Max.
Chain 1 22.03940 57.56010
Chain 2 22.03613 49.30073

Standard deviation:
            Mean Std.Dev.       Min.     2.5%  1st Qu.   Median  3rd Qu.
Chain 1 5.024763 1.020024 0.06109612 3.810018 4.469011 4.852147 5.302620
Chain 2 5.023523 1.024113 0.02530225 3.809339 4.466287 4.850280 5.298962
          97.5%     Max.
Chain 1 7.681152 45.97838
Chain 2 7.682431 35.00435
```

R⇒ Computation of the predictive density (separately for chain 1 and chain 2):

```
> if (RUN.TIMECONSUMING.CODE) {
+     PDensRJ2 <- list()
+     PDensRJ2[[1]] <- NMixPredDensMarg(RJModel2[[1]], grid = ygrid)
+     PDensRJ2[[2]] <- NMixPredDensMarg(RJModel2[[2]], grid = ygrid)
+ }
```

R⇒ Default `plot` method for the computed object (see Figure 2):

```
> postscript(paste(FIGDIR, "figGalaxy02.ps", sep = ""), width = 6,
+     height = 6, horizontal = FALSE)
> par(mfrow = c(1, 1), bty = "n")
> plot(PDensRJ2[[1]], xlab = "Velocity (km/sec)")
> dev.off()
```

R⇒ Plots of conditional predictive densities (given K) plus the overall predictive density (see Figure 3):

```
> postscript(paste(FIGDIR, "figGalaxy03.ps", sep=""), width=6, height=6,
+            horizontal=FALSE)
> par(mfrow=c(1, 1), bty="n")
> plot(PDensRJ2[[1]], K=c(0, 4:7), xlab="Velocity (km/sec)",
+      lty=c(1, rep(2, 4)), col=c("darkblue", rep("red", 4)),
+      lwd=c(2, rep(1, 4)))
> dev.off()
```

R⇒ Plot of the predictive density together with the histogram of the data for both chains (see Figure 4):

```
> postscript(paste(FIGDIR, "figGalaxy04.ps", sep=""), width=6, height=10,
+            horizontal=FALSE)
> par(mfrow=c(2, 1), bty="n")
> #
> ## Chain 1
> hist(Galaxy, prob=TRUE, col="sandybrown",
+      breaks=seq(7, 37, by=0.5),
+      xlab="Velocity (km/sec)", ylab="Density", main="Chain 1")
> lines(PDensRJ2[[1]]$x$x1, PDensRJ2[[1]]$dens[[1]],
+       col="darkblue", lwd=2)
> #
> ## Chain 2
> hist(Galaxy, prob=TRUE, col="sandybrown",
+      breaks=seq(7, 37, by=0.5),
+      xlab="Velocity (km/sec)", ylab="Density", main="Chain 2")
> lines(PDensRJ2[[2]]$x$x1, PDensRJ2[[2]]$dens[[1]],
+       col="darkblue", lwd=2)
> dev.off()
```

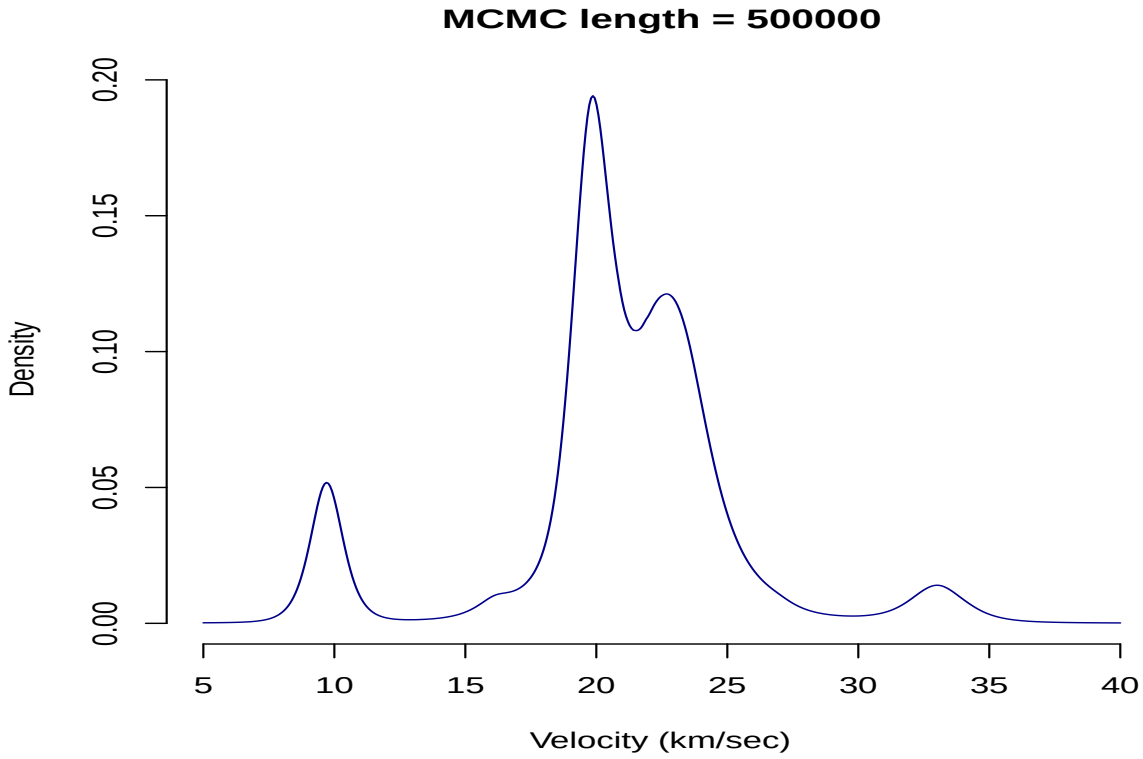

Figure 2: Predictive density based on the model with a random number of mixture components (results from chain 1).

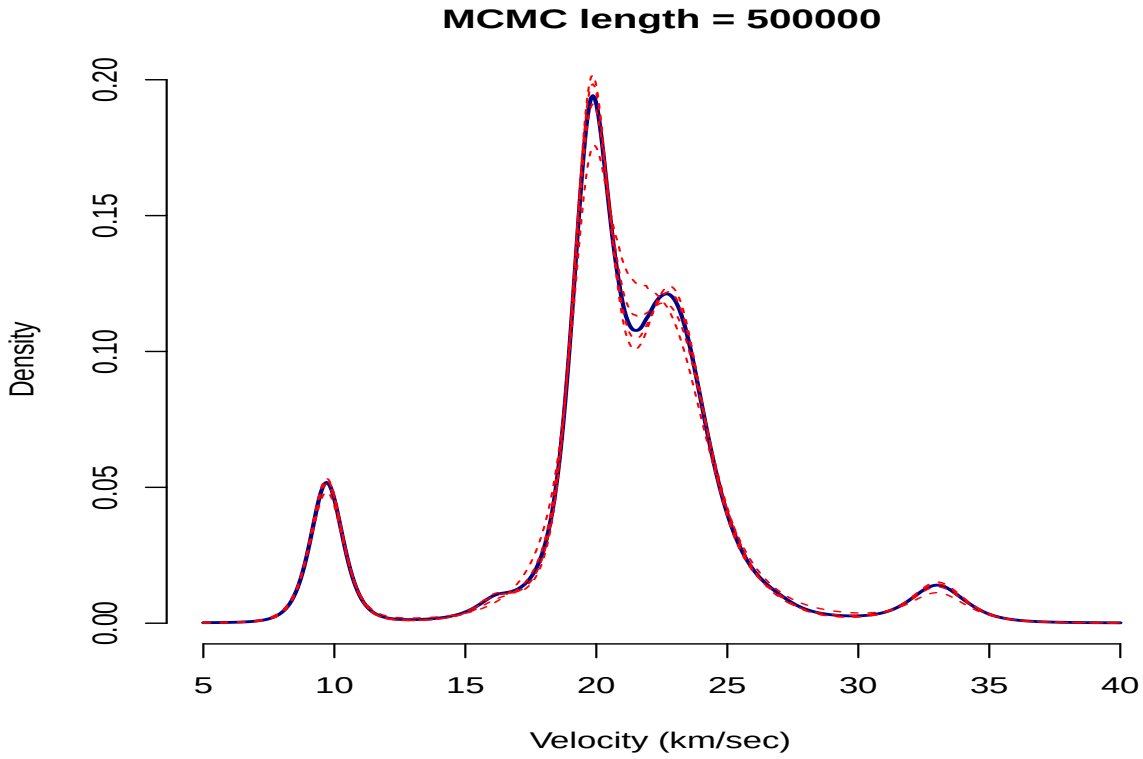

Figure 3: Overall predictive density based on the model with a random number of mixture components (in blue) and conditional predictive densities for $K = 4, 5, 6, 7$ (in red), results from chain 1.

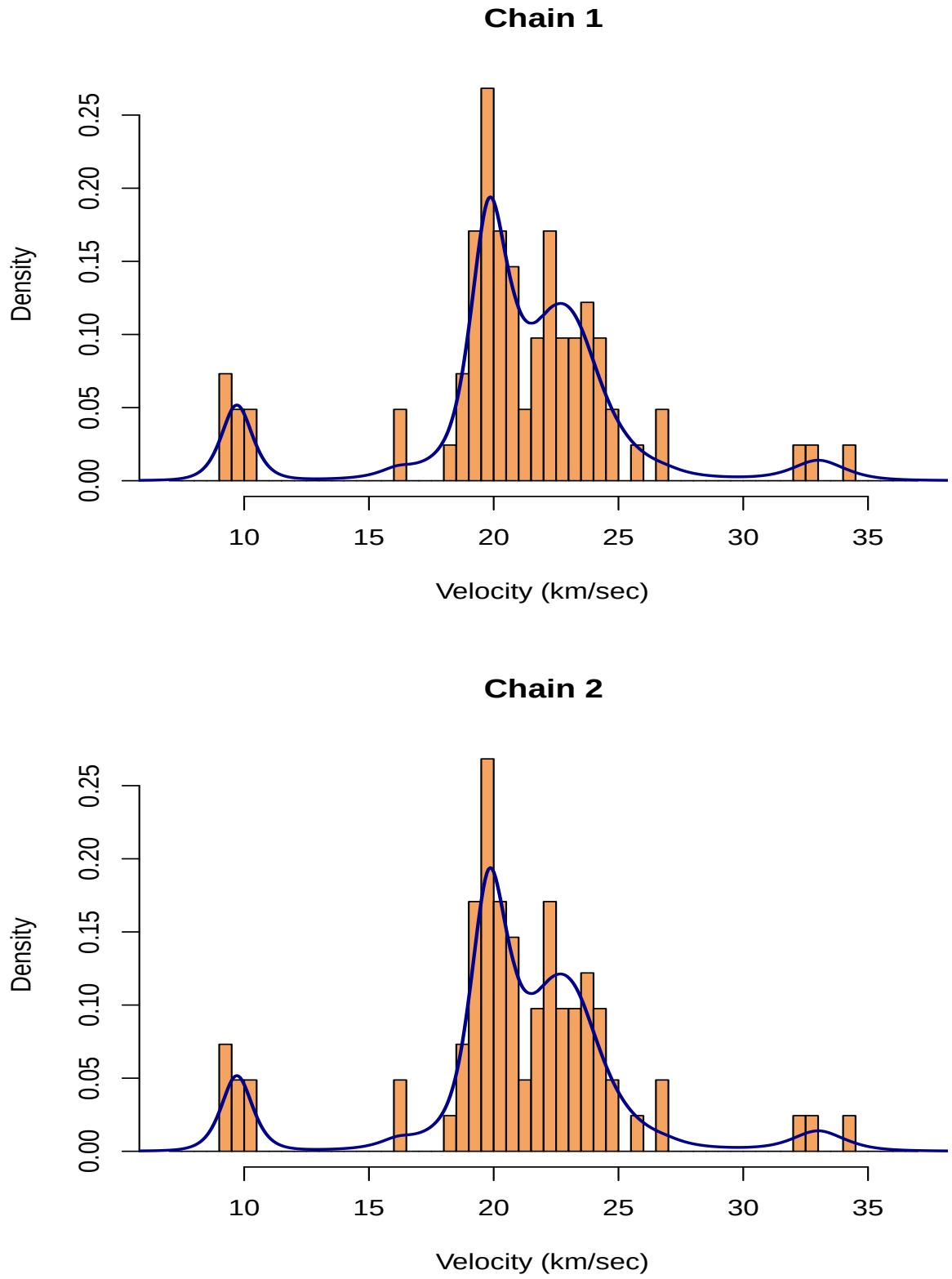

Figure 4: Predictive density (red line) based on the model with a random number of mixture components.

4.3 Convergence diagnostics

R⇒ Single chain convergence diagnostics using chain 1 will be shown here.

```
> CH <- 1
```

R⇒ Converting the chains into `mcmc` objects to be used in the package `coda`:

```
> if (RUN.ALLOUT){
+   start <- RJModel2[[CH]]$nMCMC["burn"] + 1
+   end <- RJModel2[[CH]]$nMCMC["burn"] + RJModel2[[CH]]$nMCMC["keep"]
+   chK <- mcmc(RJModel2[[CH]]$K, start=start, end=end)
+   chgammaInv <- mcmc(RJModel2[[CH]]$gammaInv, start=start, end=end)
+   chmixture <- mcmc(RJModel2[[CH]]$mixture, start=start, end=end)
+   chdeviance <- mcmc(RJModel2[[CH]]$deviance, start=start, end=end)
+ }
```

R⇒ Traceplots for selected parameters (not shown) can be drawn using the following commands:

```
> if (RUN.ALLOUT){
+   lwd <- 0.5
+   postscript(paste(FIGKEEPDIR, "figGalaxy07.ps", sep=""), width=6, height=9,
+              horizontal=FALSE)
+   par(mfrow=c(2, 2), bty="n")
+   traceplot(chK, smooth=FALSE, col="darkgreen", lwd=lwd, main="K")
+   traceplot(chgammaInv[, "gammaInv1"], smooth=FALSE,
+             col="brown", lwd=lwd, main="gamma^{-1}")
+   traceplot(chmixture[, "y.Mean.1"], smooth=FALSE,
+             col="darkblue", lwd=lwd, main="EY")
+   traceplot(chmixture[, "y.SD.1"], smooth=FALSE,
+             col="darkblue", lwd=lwd, main="sd(Y)")
+   dev.off()
+ #
+   postscript(paste(FIGKEEPDIR, "figGalaxy08.ps", sep=""), width=6, height=9,
+              horizontal=FALSE)
+   par(mfrow=c(2, 2), bty="n")
+   traceplot(chdeviance[, "LogL0"], smooth=FALSE,
+             col="red", lwd=lwd, main="Log(L0)")
+   traceplot(chdeviance[, "LogL1"], smooth=FALSE,
+             col="red", lwd=lwd, main="Log(L1)")
+   traceplot(chdeviance[, "dev.complete"], smooth=FALSE,
+             col="red", lwd=lwd, main="D(complete)")
+   traceplot(chdeviance[, "dev.observed"], smooth=FALSE,
+             col="red", lwd=lwd, main="D(observed)")
+   dev.off()
+ }
```

R⇒ On Figure 5, we show a traceplot of K of last 10 000 iterations drawn using the following commands:

```
> if (RUN.ALLOUT){
+   chKpart <- mcmc(RJModel2[[CH]]$K[490001:500000], start=start+490000, end=end)
+   postscript(paste(FIGKEEPDIR, "figGalaxy07a.ps", sep=""), width=9, height=6,
+              horizontal=FALSE)
+   par(mfrow=c(1, 1), bty="n")
+   traceplot(chKpart, smooth=FALSE, col="darkgreen", main="K")
+   dev.off()
+ }
```

R⇒ Posterior density estimates for selected parameters (see Figure 6):

```
> if (RUN.ALLOUT){
+   postscript(paste(FIGKEEPDIR, "figGalaxy09.ps", sep=""), width=6, height=9,
+              horizontal=FALSE)
+   par(mfrow=c(2, 2), bty="n")
+   densplot(chK, show.obs=FALSE, col="darkgreen", main="K")
+   densplot(chgammaInv[, "gammaInv1"], show.obs=FALSE,
+            col="brown", main="gamma^{-1}", xlim=c(0, 30))
+   densplot(chmixture[, "y.Mean.1"], show.obs=FALSE,
+            col="darkblue", main="EY", xlim=c(15, 25))
+   densplot(chmixture[, "y.SD.1"], show.obs=FALSE,
+            col="darkblue", main="sd(Y)", xlim=c(0, 12))
+   dev.off()
+ }
```

R⇒ Autocorrelation plots for selected parameters (see Figure 7):

```
> if (RUN.ALLOUT){
+   postscript(paste(FIGKEEPDIR, "figGalaxy10.ps", sep=""), width=6, height=9,
+              horizontal=FALSE)
+   par(mfrow=c(2, 2), bty="n")
+   autocorr.plot(chK, auto.layout=FALSE, ask=FALSE,
+                 col="darkgreen", lwd=2, main="K")
+   autocorr.plot(chgammaInv[, "gammaInv1"], auto.layout=FALSE, ask=FALSE,
+                 col="brown", lwd=2, main="gamma^{-1}")
+   autocorr.plot(chmixture[, "y.Mean.1"], auto.layout=FALSE, ask=FALSE,
+                 col="darkblue", lwd=2, main="EY")
+   autocorr.plot(chmixture[, "y.SD.1"], auto.layout=FALSE, ask=FALSE,
+                 col="darkblue", lwd=2, main="sd(Y)")
+   dev.off()
+ }
```

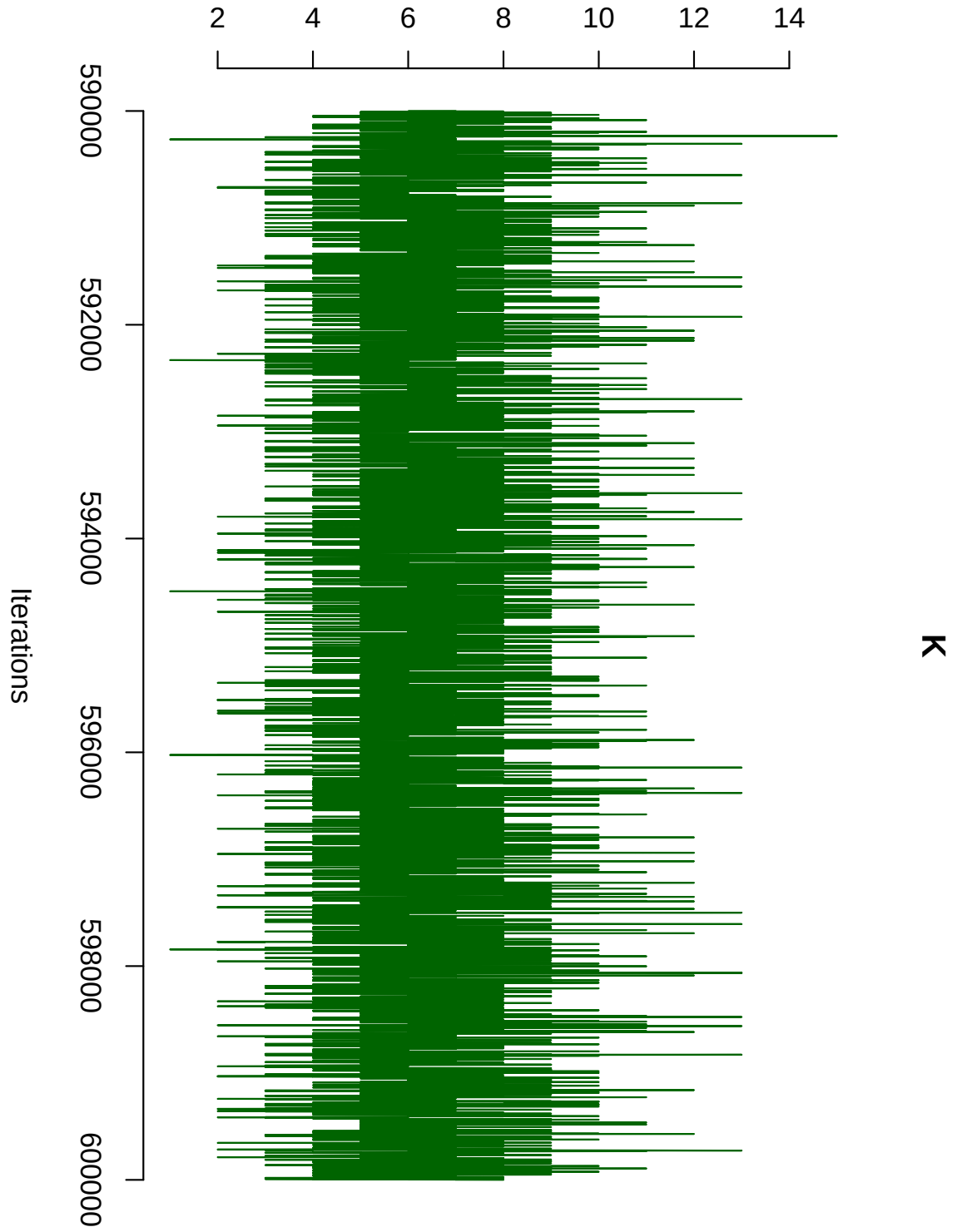

Figure 5: Model with a random number of mixture components. Traceplots for the number of mixture components K (last 10 000 iterations).

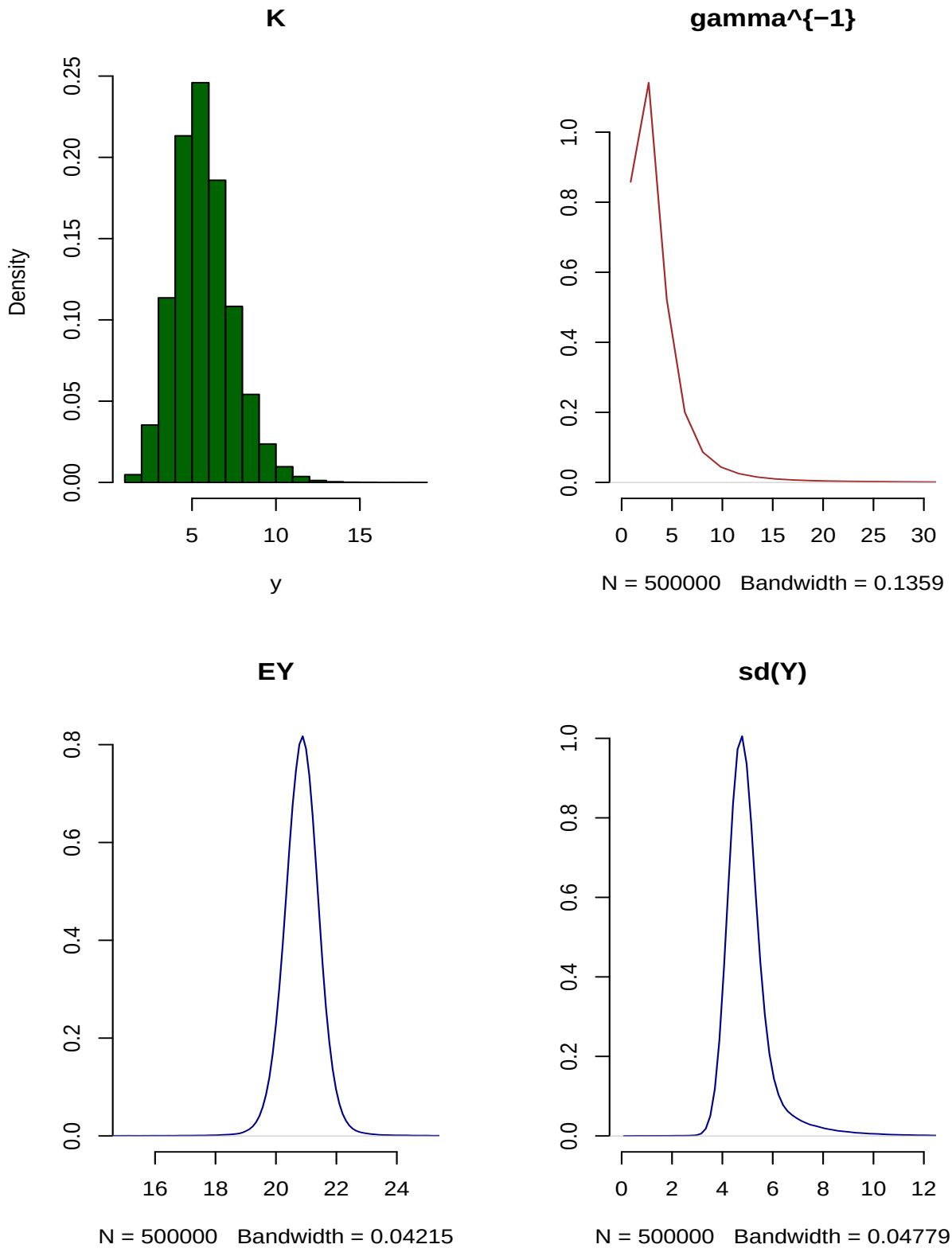

Figure 6: Model with a random number of mixture components. Posterior density estimates for selected parameters.

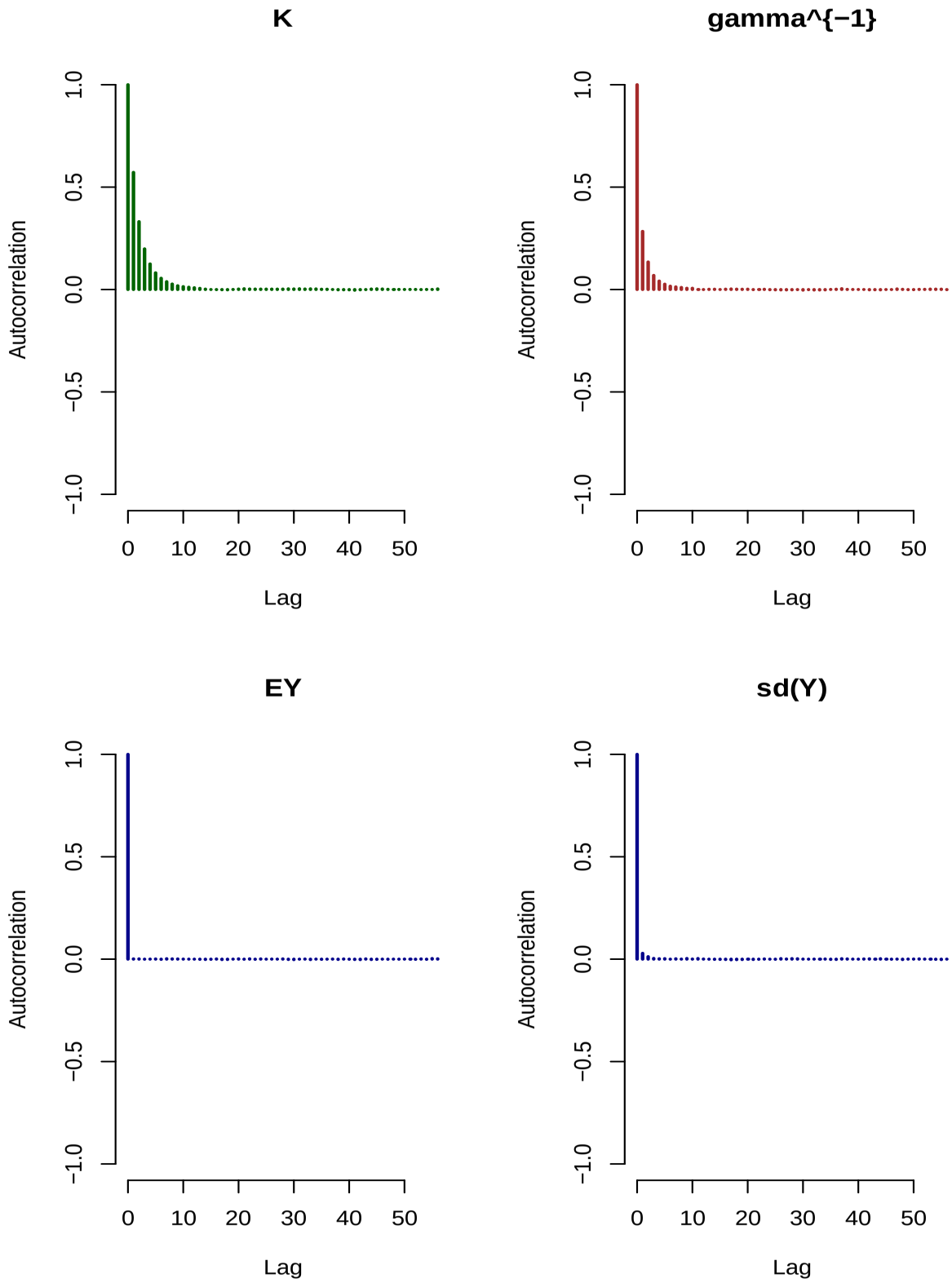

Figure 7: Model with a random number of mixture components. Autocorrelation plots for selected parameters.

5 Model with a fixed number of components

We will fit a mixture model for $K = 1, \ldots, 10$, compare the deviance based quantities and predictive densities. For the prior hyperparameters common to the model with a random K, we will use the same as in the model RJModel2.

R⇒ Specification of the prior hyperparameters:

```
> FixPrior2 <- list(priorK="fixed",
+                   delta=1,
+                   priormuQ="independentC", xi=21.73, D=630.5121,
+                   zeta=2*2, g=0.2, h=0.016/2)
```

R⇒ Running the MCMC simulation for $K = 1, \ldots, 10$ (output printed during the MCMC run on the screen not shown), computation of predictive densities.

R⇒ After predictive densities are computed, we remove all chains from resulting objects (to save some memory).

```
> if (RUN.TIMECONSUMING.CODE){
+   Keep <- c("iter", "nMCMC", "dim", "prior", "init", "RJMCMC",
+             "scale", "state", "freqK", "propK", "DIC", "moves",
+             "pm.y", "pm.z", "pm.indDev", "pred.dens", "summ.y.Mean",
+             "summ.y.SDCorr", "summ.z.Mean", "summ.z.SDCorr")
+   set.seed(770328)
+   FixModel2 <- list()
+   PDensFix2 <- list()
+   for (k in 1:10){
+     cat(paste("K = ", k, "\n------------------------------\n", sep=""))
+     PriorNow <- FixPrior2
+     PriorNow$Kmax <- k
+     FixModel2[[k]] <- NMixMCMC(y0=Galaxy, prior=PriorNow, nMCMC=nMCMC,
+                                scale=list(shift=0, scale=1), PED=TRUE)
+   #
+     cat(paste("\nComputation of pred. densities started on ", date(),
+               "\n", sep=""))
+     PDensFix2[[k]] <- list()
+     PDensFix2[[k]][[1]] <- NMixPredDensMarg(FixModel2[[k]][[1]], grid=ygrid)
+     PDensFix2[[k]][[2]] <- NMixPredDensMarg(FixModel2[[k]][[2]], grid=ygrid)
+     cat(paste("Computation of pred. densities finished on ", date(),
+               "\n\n\n", sep=""))
+   #
+     FixModel2[[k]][[1]] <- FixModel2[[k]][[1]][Keep]
+     FixModel2[[k]][[2]] <- FixModel2[[k]][[2]][Keep]
+     class(FixModel2[[k]][[1]]) <- class(FixModel2[[k]][[2]]) <- "NMixMCMC"
+   }
+ }
```

R⇒ Basic posterior summary of the fitted model with $K = 6$:

```
> print(FixModel2[[6]])


     6 component normal mixture estimated using MCMC
     ================================================

Penalized expected deviance:
----------------------------
D.expect   p(opt)      PED  wp(opt)     wPED
427.1929 118.7859 545.9789 136.7527 563.9456

Deviance information criteria:
------------------------------
             DIC       pD    D.bar D.in.bar
Chain 1 449.0657 26.08602 422.9797 396.8936
Chain 2 448.5626 25.84903 422.7136 396.8646

Posterior summary statistics for moments of mixture for original data:
----------------------------------------------------------------------
Mean:
            Mean  Std.Dev.     Min.     2.5%  1st Qu.   Median  3rd Qu.
Chain 1 20.85548 0.5750815 12.53002 19.71908 20.49663 20.85971 21.21807
Chain 2 20.85448 0.5738582 14.46291 19.71958 20.49497 20.85916 21.21748
           97.5%     Max.
Chain 1 21.97278 26.53756
Chain 2 21.96561 27.18524

Standard deviation:
            Mean  Std.Dev.      Min.     2.5%  1st Qu.   Median  3rd Qu.
Chain 1 4.974256 0.9143447 0.4844963 3.812832 4.467038 4.840967 5.264126
Chain 2 4.969868 0.9081553 0.5223116 3.816097 4.466204 4.839025 5.262335
           97.5%     Max.
Chain 1 7.254569 33.55014
Chain 2 7.216001 32.82954
```

R⇒ Summary of PED and DIC's for the fitted models:

```
> PED <- RJModel2$PED
> DIC <- list(Chain1 = RJModel2[[1]]$DIC, Chain2 = RJModel2[[2]]$DIC)
> for (k in 1:length(FixModel2)) {
+     PED <- rbind(PED, FixModel2[[k]]$PED)
+     DIC[[1]] <- rbind(DIC[[1]], FixModel2[[k]][[1]]$DIC)
+     DIC[[2]] <- rbind(DIC[[2]], FixModel2[[k]][[2]]$DIC)
+ }
> rownames(PED) <- rownames(DIC[[1]]) <- rownames(DIC[[2]]) <- c("RJ-MCMC",
+     paste("K = ", 1:length(FixModel2), sep = ""))
```

```
> print(PED)
```

```
        D.expect     p(opt)      PED    wp(opt)     wPED
RJ-MCMC 423.0743 114.340974 537.4152 126.374978 549.4492
K = 1   482.8301   4.070529 486.9006   4.087999 486.9181
K = 2   446.1402  11.436937 457.5771  11.845757 457.9859
K = 3   418.7662  20.787964 439.5542  20.947999 439.7142
K = 4   418.9034  42.658219 461.5616  50.636206 469.5396
K = 5   422.9906  73.942819 496.9334  88.074453 511.0651
K = 6   427.1929 118.785942 545.9789 136.752672 563.9456
K = 7   430.2111 169.979802 600.1909 190.077819 620.2889
K = 8   430.2424 223.377508 653.6199 242.123948 672.3664
K = 9   431.0306 283.241647 714.2723 300.760083 731.7907
K = 10  430.5812 345.667178 776.2484 360.986272 791.5675
```

```
> print(DIC)
```

```
$Chain1
             DIC        pD    D.bar D.in.bar
RJ-MCMC 442.6806 22.232147 420.4485 398.2163
K = 1   485.7709  2.944780 482.8261 479.8813
K = 2   451.1053  4.975623 446.1297 441.1540
K = 3   424.3881  5.633584 418.7545 413.1209
K = 4   430.7262 13.770323 416.9559 403.1856
K = 5   440.3760 20.656563 419.7195 399.0629
K = 6   449.0657 26.086023 422.9797 396.8936
K = 7   455.5718 30.243397 425.3284 395.0850
K = 8   457.5007 32.272065 425.2286 392.9565
K = 9   461.5795 35.464813 426.1146 390.6498
K = 10  463.4231 37.683232 425.7399 388.0567

$Chain2
             DIC        pD    D.bar D.in.bar
RJ-MCMC 442.6908 22.221437 420.4693 398.2479
K = 1   485.7925  2.958353 482.8341 479.8758
K = 2   451.7327  5.582011 446.1507 440.5686
K = 3   424.4238  5.645818 418.7779 413.1321
K = 4   431.3099 14.069949 417.2400 403.1700
K = 5   441.2720 21.105927 420.1661 399.0602
K = 6   448.5626 25.849028 422.7136 396.8646
K = 7   453.9765 29.467170 424.5093 395.0422
K = 8   457.7424 32.372631 425.3697 392.9971
K = 9   460.9831 35.177570 425.8055 390.6279
K = 10  463.7275 37.841151 425.8863 388.0452
```

R⇒ Plot of the predictive densities for different values of K based on chain 1 (see Figures 8 and 9):

```
> postscript(paste(FIGDIR, "figGalaxy05.ps", sep=""), width=6, height=6,
+            horizontal=FALSE)
> par(mfrow=c(1, 1), bty="n")
> hist(Galaxy, prob=TRUE, col="grey90", breaks=seq(7, 37, by=0.5),
+      xlab="Velocity (km/sec)", ylab="Density", main="")
> for (k in 1:10){
+   lines(PDensFix2[[k]][[1]]$x$x1, PDensFix2[[k]][[1]]$dens[[1]], col="red")
+ }
> dev.off()
```

```
> postscript(paste(FIGDIR, "figGalaxy06.ps", sep=""), width=6, height=9,
+            horizontal=FALSE)
> par(mar=c(3, 2, 2, 1)+0.1)
> par(mfrow=c(5, 2), bty="n")
> for (k in 1:10){
+   hist(Galaxy, prob=TRUE, col="lightblue", breaks=seq(7, 37, by=0.5),
+        xlab="", ylab="", main=paste("K = ", k, sep=""))
+   lines(PDensFix2[[k]][[1]]$x$x1, PDensFix2[[k]][[1]]$dens[[1]], col="red", lwd=2)
+ }
> dev.off()
```

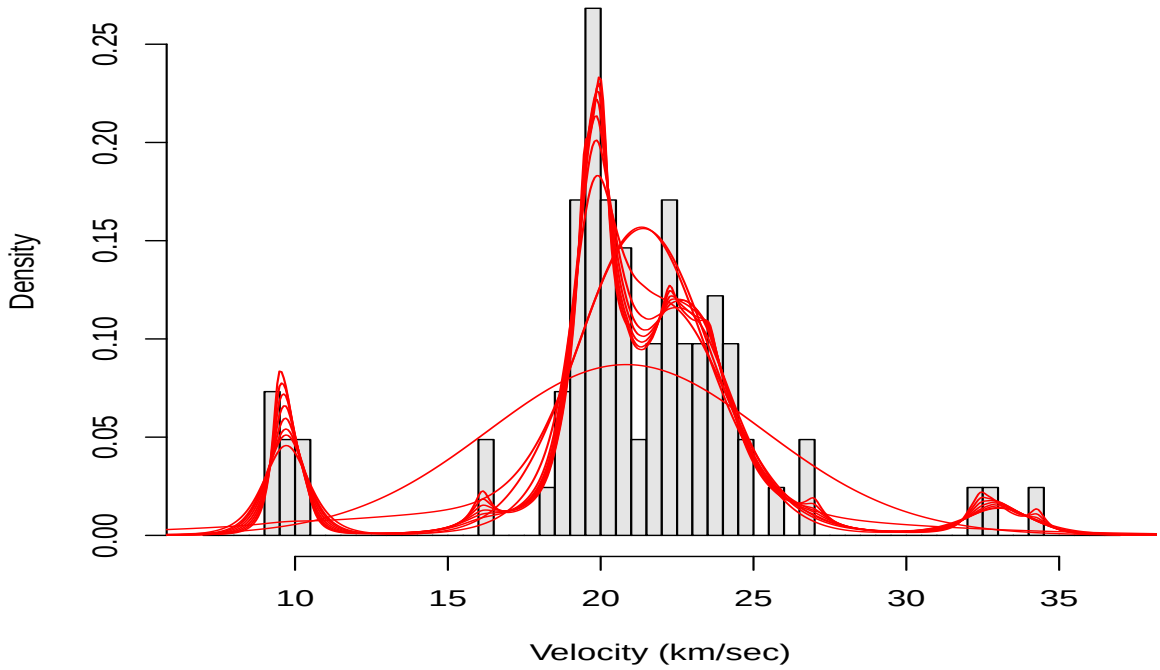

Figure 8: Predictive densities based on the models with a fixed number of mixture components (results from chain 1).

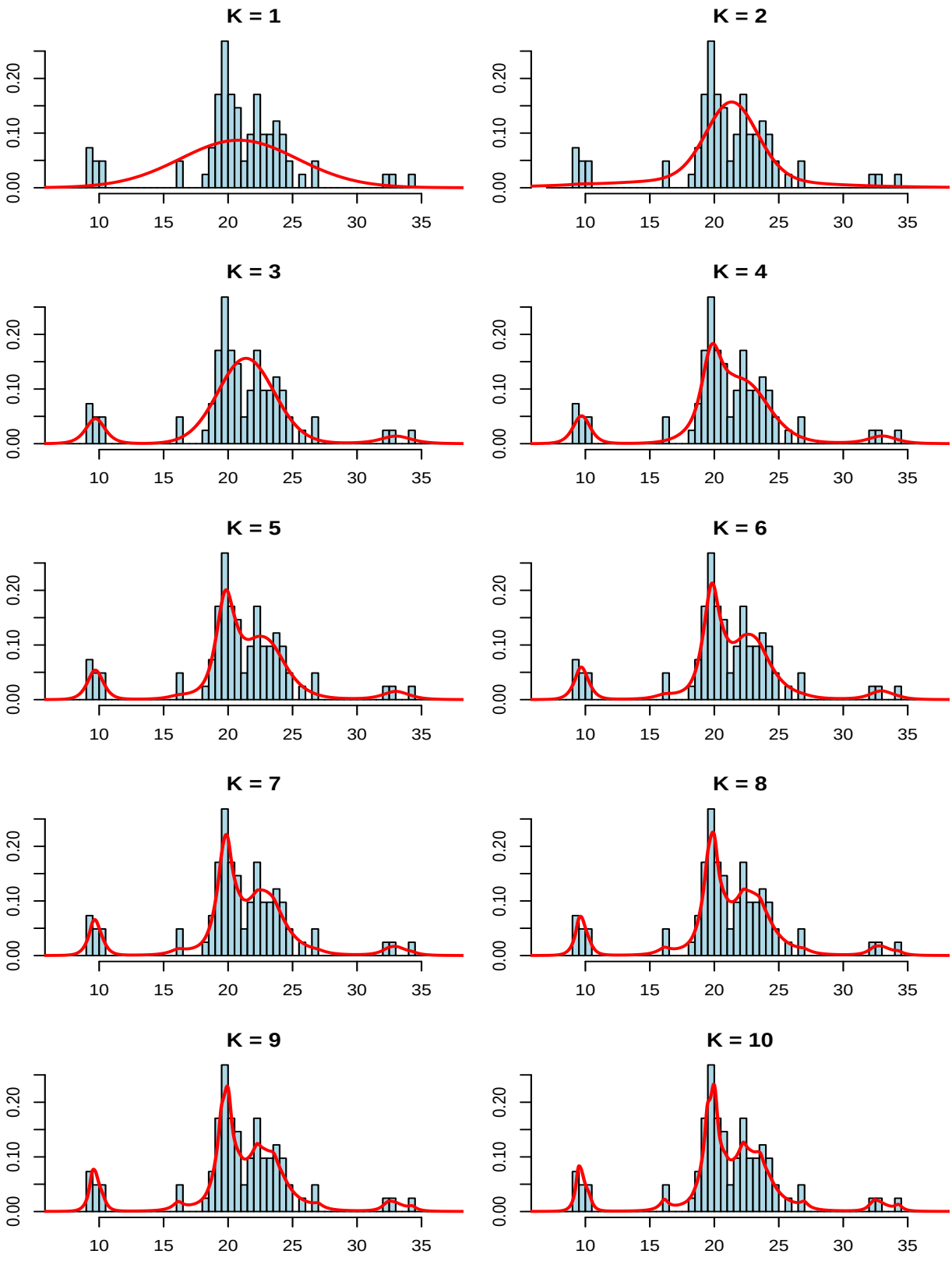

Figure 9: Predictive densities based on the models with a fixed number of mixture components (results from chain 1).

R⇒ Save results for future use. For file Galaxy-Result.RData, we exclude all chains from object RJModel2.

```
> if (RUN.TIMECONSUMING.CODE){
+   save(list="RJModel2",
+        file=paste(RESULTDIR, "Galaxy-RJ2.RData", sep=""))
+
+   Keep <- c("iter", "nMCMC", "dim", "prior", "init", "RJMCMC",
+             "scale", "state", "freqK", "propK", "DIC", "moves",
+             "pm.y", "pm.z", "pm.indDev", "pred.dens", "summ.y.Mean",
+             "summ.y.SDCorr", "summ.z.Mean", "summ.z.SDCorr")
+   #
+   RJModel2[[1]] <- RJModel2[[1]][Keep]
+   RJModel2[[2]] <- RJModel2[[2]][Keep]
+   class(RJModel2[[1]]) <- class(RJModel2[[2]]) <- "NMixMCMC"
+   #
+   save(list=c("RJModel2", "PDensRJ2", "FixModel2", "PDensFix2"),
+        file=paste(RESULT2DIR, "/Galaxy-Result.RData", sep=""))
+ }
```

References

Komárek, A. (2009). A new R package for Bayesian estimation of multivariate normal mixtures allowing for selection of the number of components and interval-censored data. *Computational Statistics and Data Analysis*, **53**, To appear.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with unknown number of components (with Discussion). *Journal of the Royal Statistical Society, Series B*, **59**, 731–792.

Roeder, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, **85**, 617–624.