

report15

Haoliang Zheng

2/22/2022

Contents

data	1
bdregjump_adapt_tempered_new	2
Trace plot	2
Acceptance rate	27
Inverse temperature	29

data

```
vote <- read.csv("./codes/vote_data.csv")
vote <- subset(vote, abs(from_requirement_threshold) < 0.5)
set.seed(1234)
train.ss <- sample(nrow(vote), 15e3)

preds <- model.matrix(~ #major.requirement
                      + ISS.against.recommendation
                      #+ shares.outstanding.base
                      #+ special.meeting
                      + analyst.coverage
                      #+ institutional.ownership
                      + past.stock.return
                      + q
                      + firm.size,
                      data = vote)[,-1]
preds.train <- scale(preds[train.ss,])
x <- cbind(1, preds.train)
x.test <- cbind(1, scale(preds[-train.ss,],
                           center=attr(preds.train,"scaled:center"),
                           scale=attr(preds.train,"scaled:scale")))
dimnames(x)[[2]][1] <- "Intercept"
dimnames(x.test)[[2]][1] <- "Intercept"
x.names <- dimnames(x)[[2]]

y <- 2*vote[train.ss,"from_requirement_threshold"]
```

```

y.test <- 2*vote[-train.ss,"from_requirement_threshold"]

bunch <- FALSE
pos.est <- TRUE

set.seed(123)
N_para <- 6
p <- 6
order <- 15
L <- 5

kbw <- 2/(order-1)
knots <- seq(-1,1,kbw)
bsd.sq <- (0.67*kbw)^2
gausskern <- lapply(knots, function(mu) return(function(x) return(exp(-0.5*(x-mu)^2/bsd.sq))))
get.poly.mat <- function(y) return(sapply(gausskern, function(f) f(y)))

yFn <- get.poly.mat

b_init <- replicate(N_para, matrix(rnorm(n = order*p, sd=5), order, p))
a_init <- replicate(N_para, rnorm(n = p, sd = 1.5))

```

Denote y_l^* as the missing history for chain l , then swap probability for chain 1 and 2 is:

$$\alpha(1,2) = 1 \wedge \frac{p_1(y, y_2^* | \theta_2) * p_2(y, y_1^* | \theta_1)}{p_1(y, y_1^* | \theta_1) * p_2(y, y_2^* | \theta_2)}$$

bdregjump_adapt_tempered_new

```

adapt_tempered_joint_new <- foreach(i = 1:N_para) %dorng%
  bdregjump_adapt_tempered(y=y, x=x, b=b_init[,i], burn=0, nsamp=5000, thin=1, order = 15,
                            jump=list(a=a_init[,i], prec = 1, positive=pos.est,
                                       persistence=0.68, update.jbw=FALSE),
                            adap=list(parallel = 5, alpha_star_PT = 0.234))

```

```
load(file = "./report/report15/adapt_tempered_joint_new.RData")
```

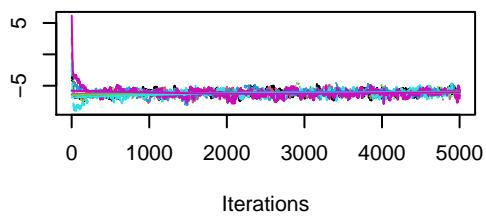
Trace plot

```

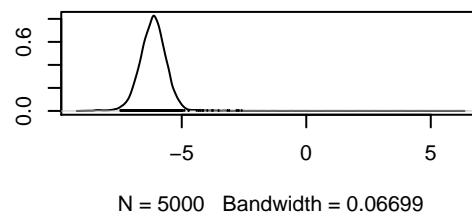
adapt_tempered_joint_new_coda <- multi_chain_parallel_coda(adapt_tempered_joint_new, burn=0, N_para, p,
plot(adapt_tempered_joint_new_coda)

```

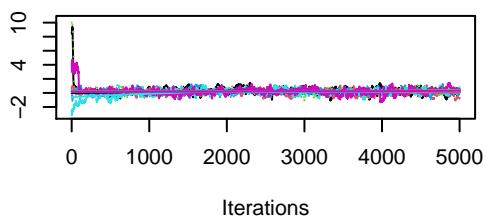
Trace of b11



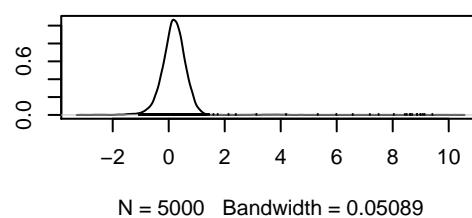
Density of b11



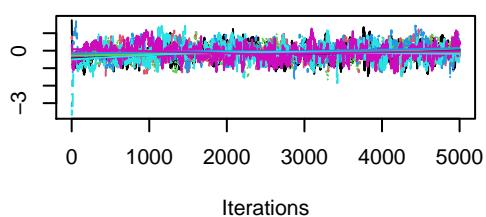
Trace of b12



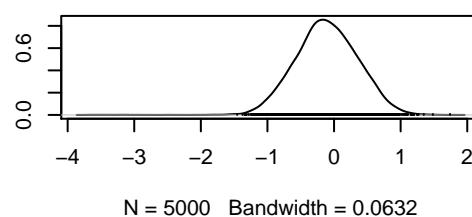
Density of b12



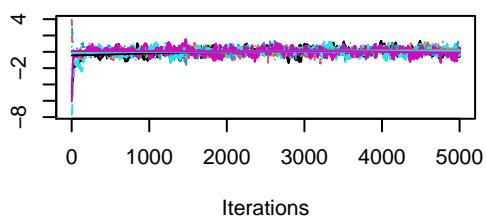
Trace of b13



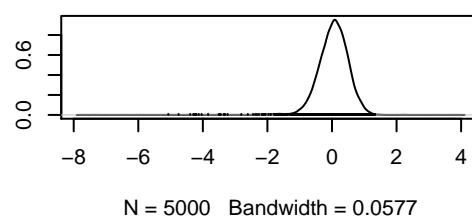
Density of b13



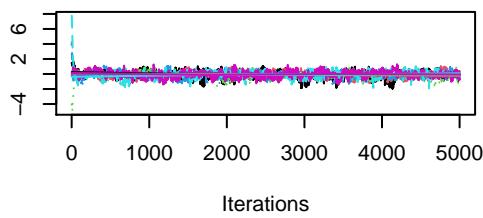
Trace of b14



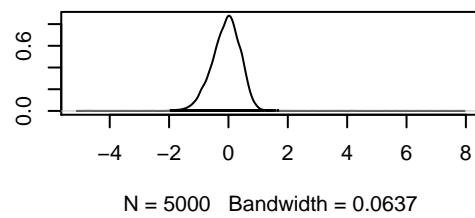
Density of b14



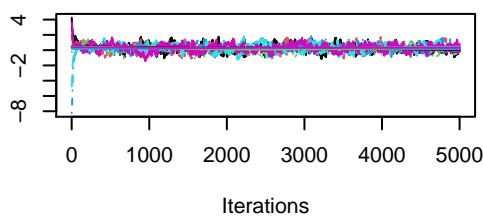
Trace of b15



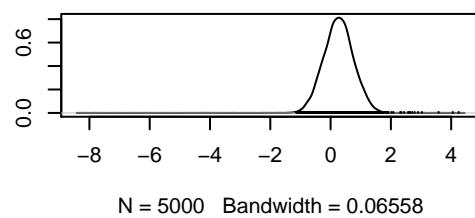
Density of b15



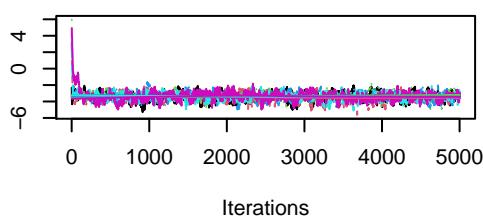
Trace of b16



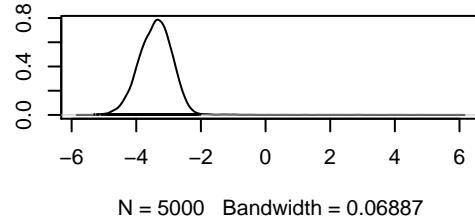
Density of b16



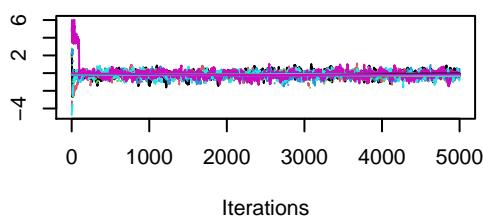
Trace of b21



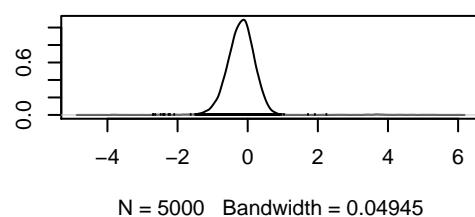
Density of b21



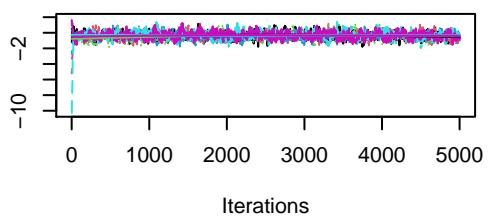
Trace of b22



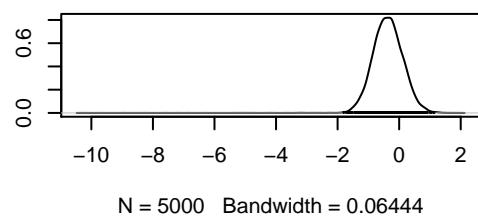
Density of b22



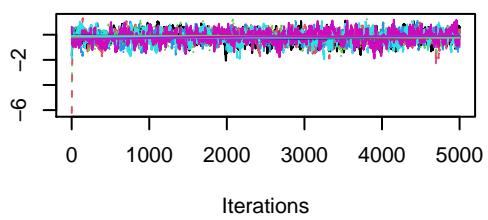
Trace of b23



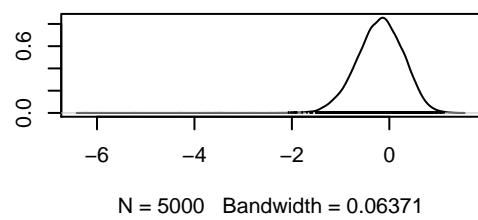
Density of b23



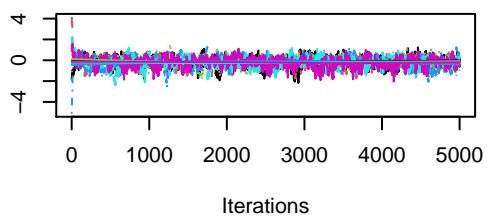
Trace of b24



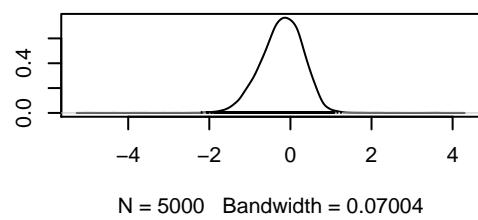
Density of b24



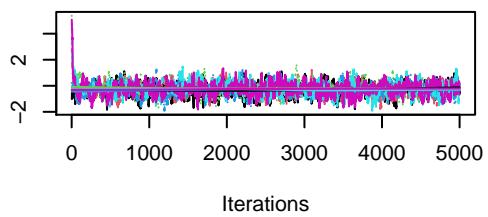
Trace of b25



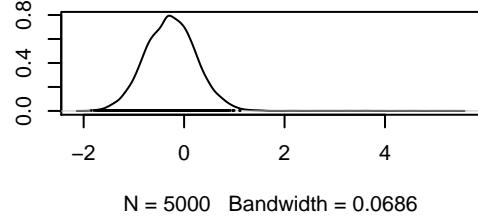
Density of b25



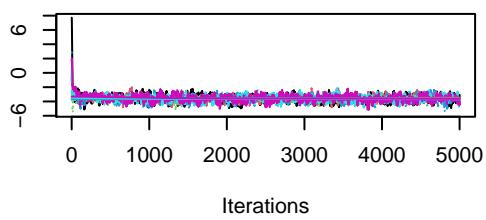
Trace of b26



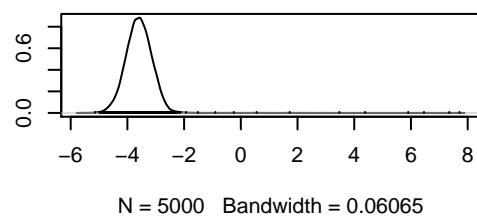
Density of b26



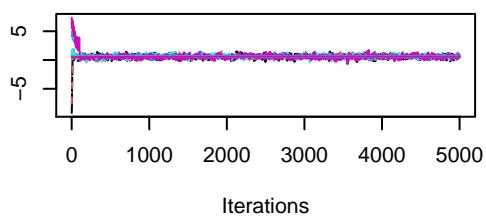
Trace of b31



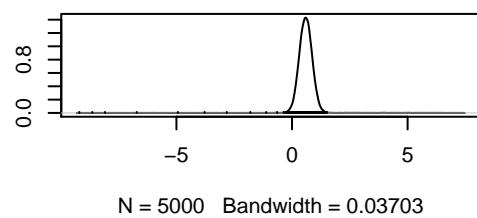
Density of b31



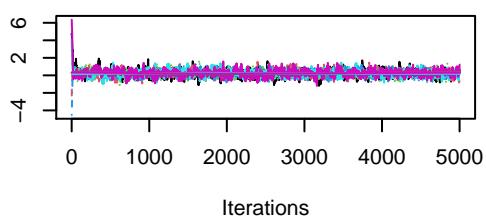
Trace of b32



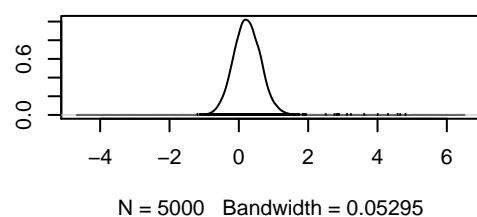
Density of b32



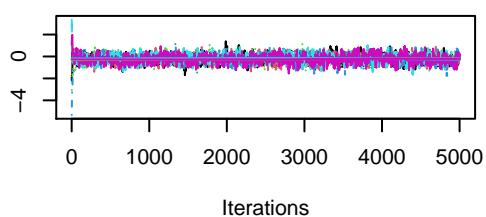
Trace of b33



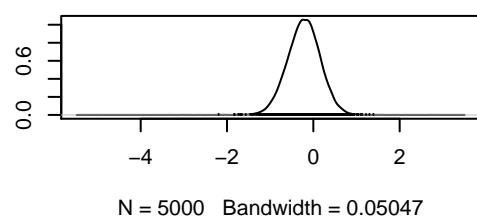
Density of b33



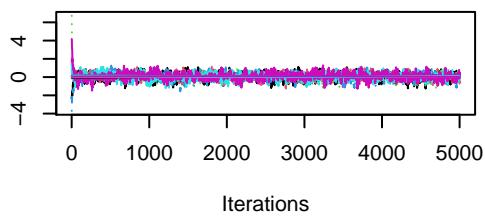
Trace of b34



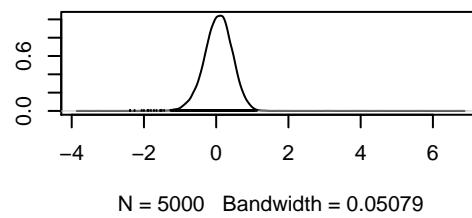
Density of b34



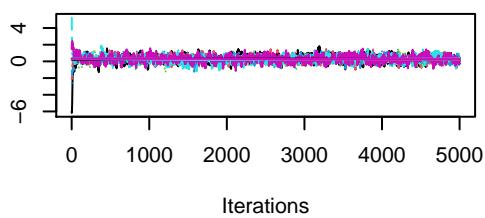
Trace of b35



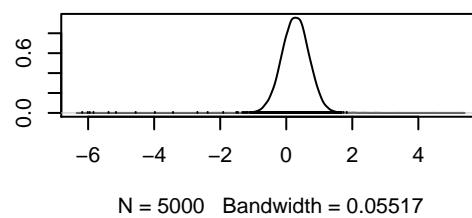
Density of b35



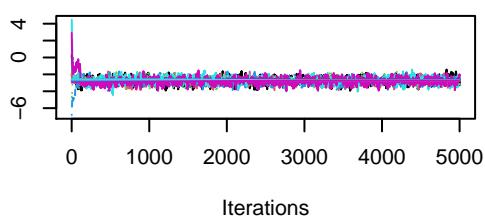
Trace of b36



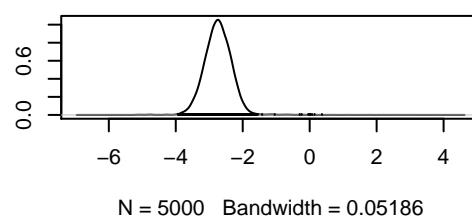
Density of b36



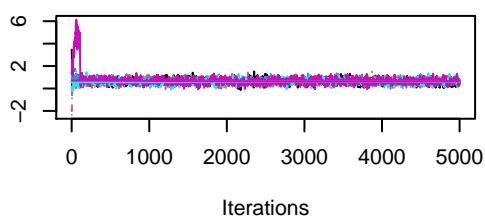
Trace of b41



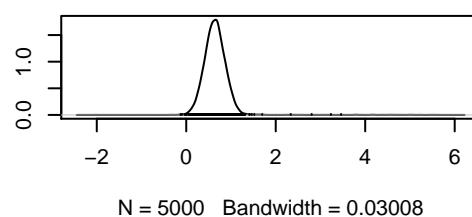
Density of b41



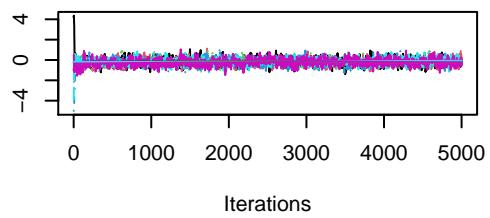
Trace of b42



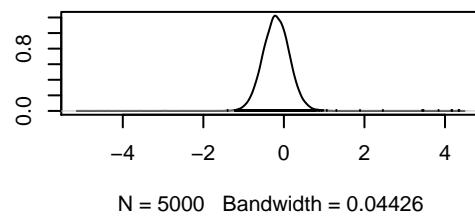
Density of b42



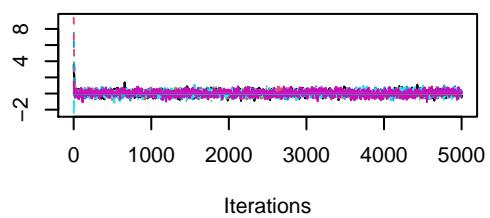
Trace of b43



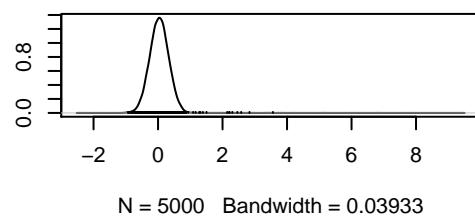
Density of b43



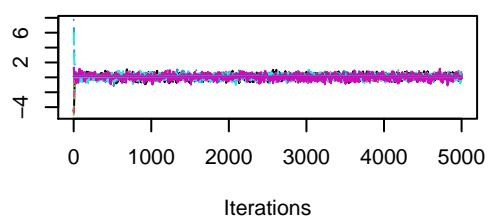
Trace of b44



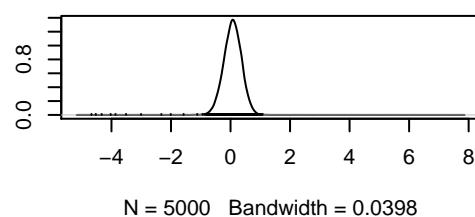
Density of b44



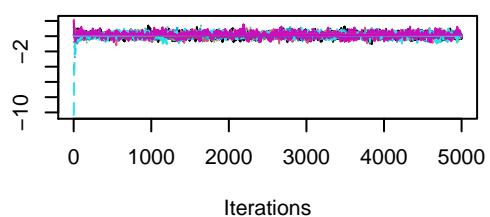
Trace of b45



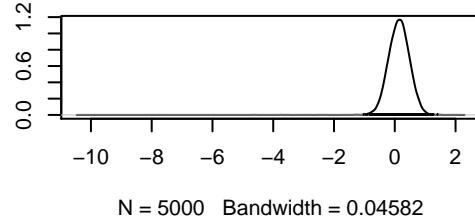
Density of b45



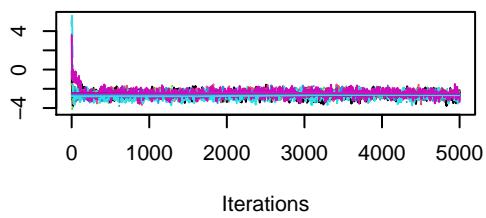
Trace of b46



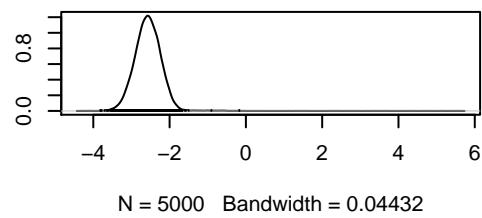
Density of b46



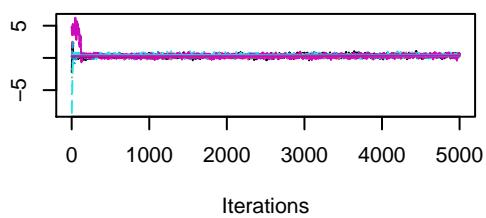
Trace of b51



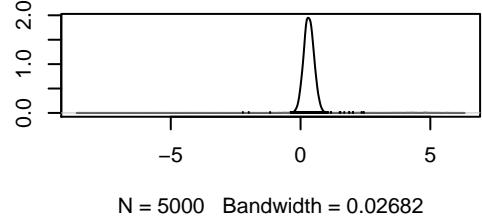
Density of b51



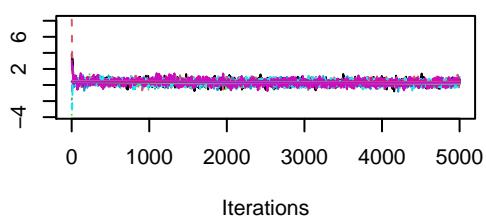
Trace of b52



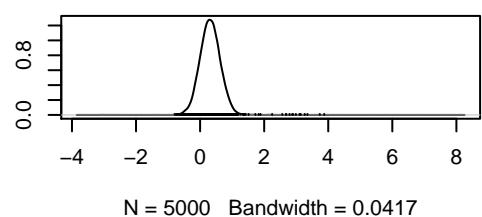
Density of b52



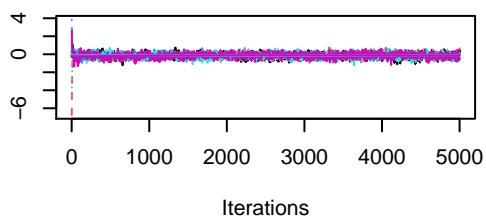
Trace of b53



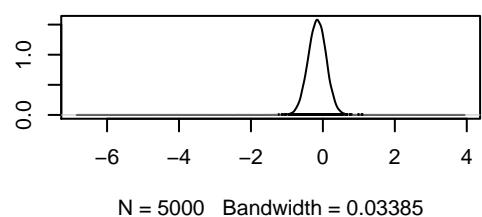
Density of b53



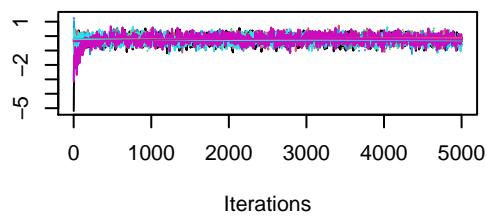
Trace of b54



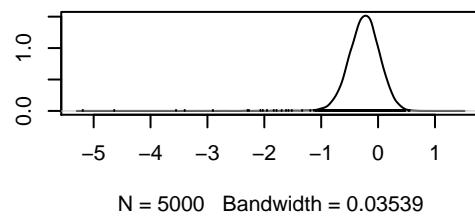
Density of b54



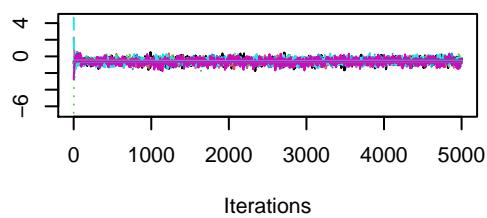
Trace of b55



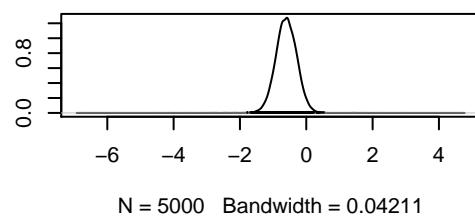
Density of b55



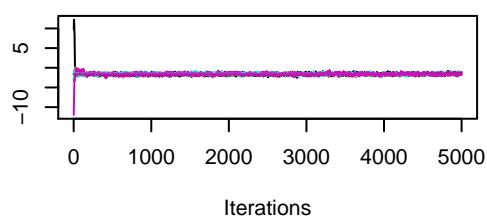
Trace of b56



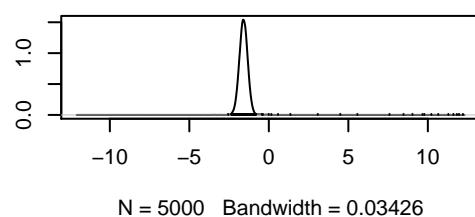
Density of b56



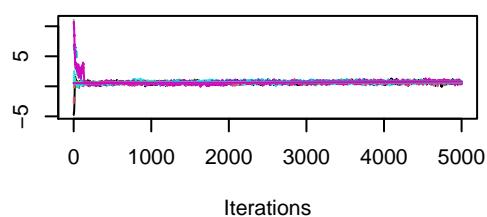
Trace of b61



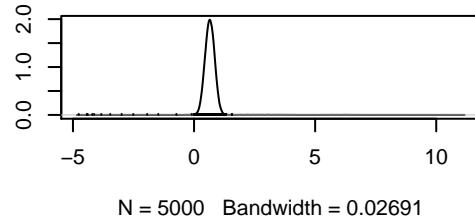
Density of b61



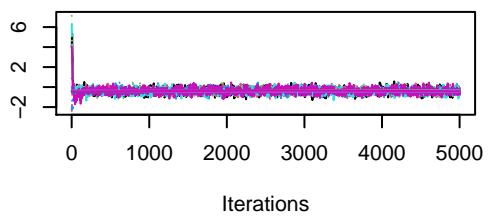
Trace of b62



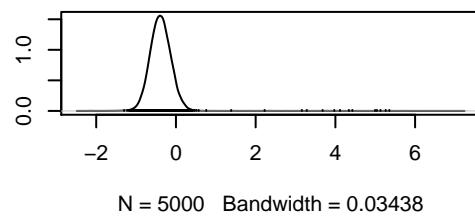
Density of b62



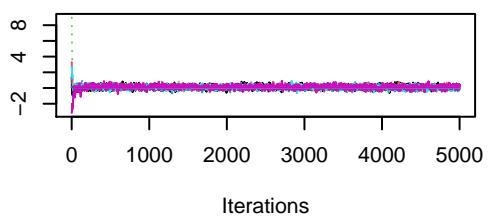
Trace of b63



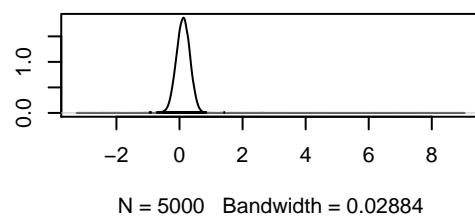
Density of b63



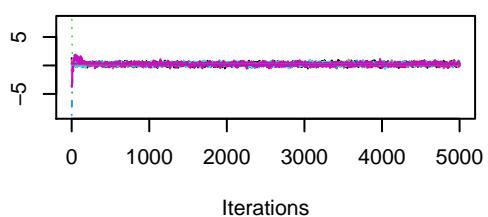
Trace of b64



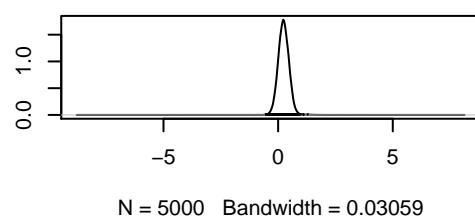
Density of b64



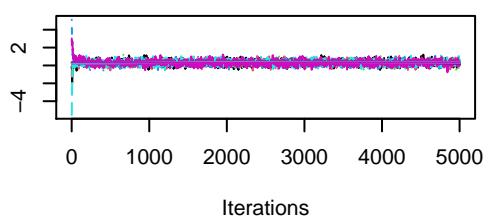
Trace of b65



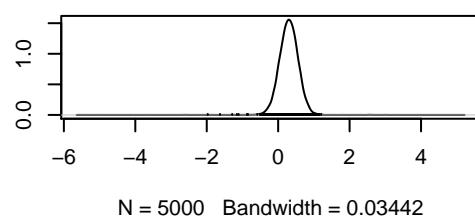
Density of b65



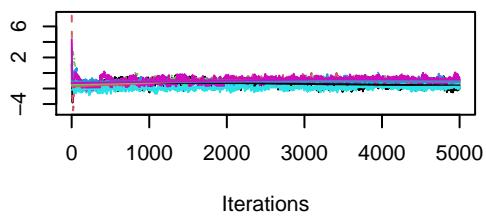
Trace of b66



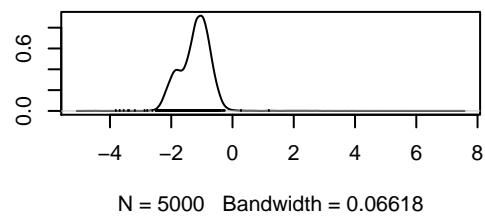
Density of b66



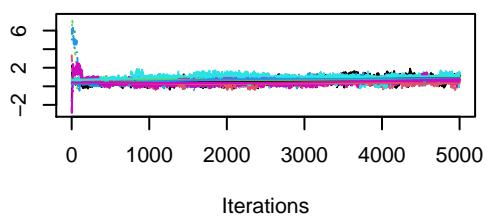
Trace of b71



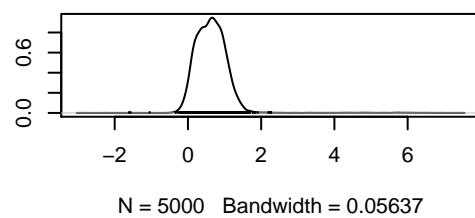
Density of b71



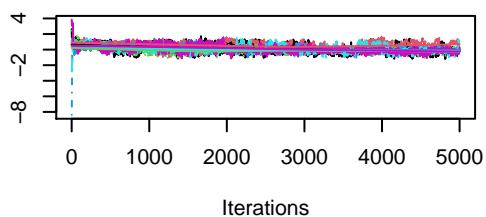
Trace of b72



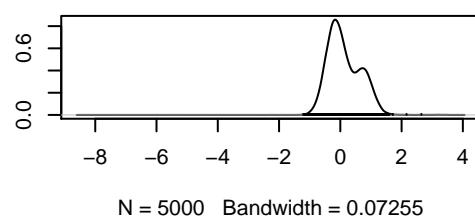
Density of b72



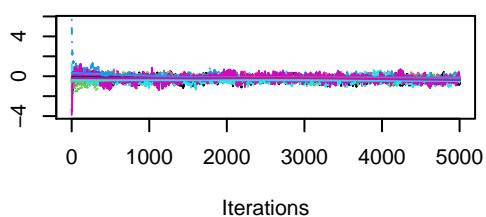
Trace of b73



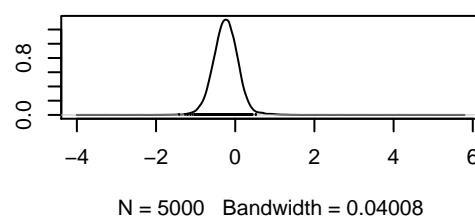
Density of b73



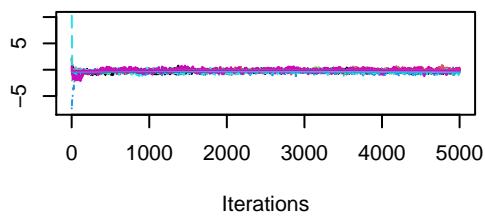
Trace of b74



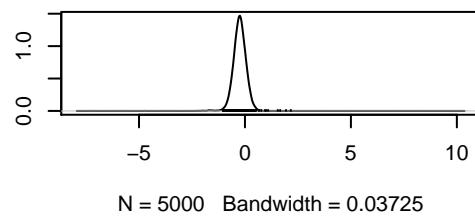
Density of b74



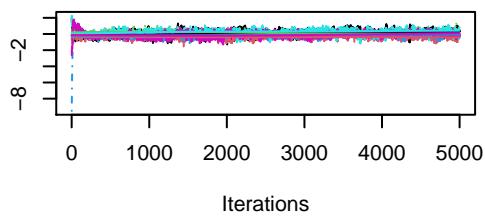
Trace of b75



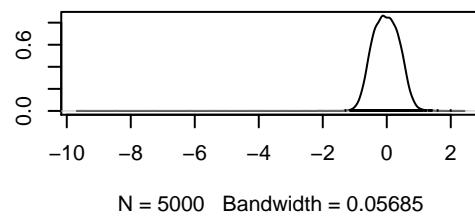
Density of b75



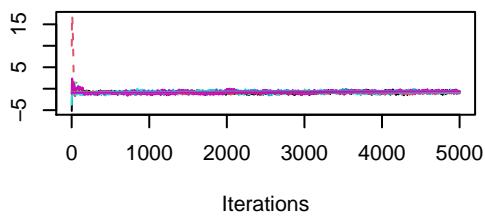
Trace of b76



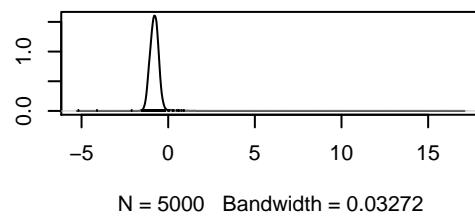
Density of b76



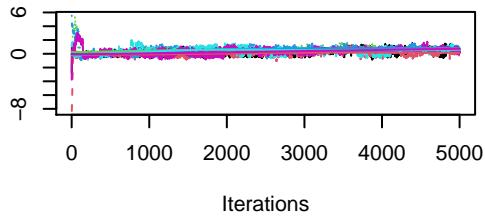
Trace of b81



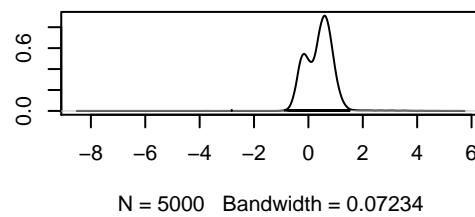
Density of b81



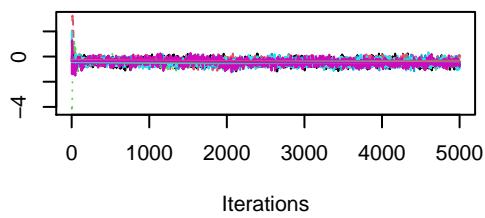
Trace of b82



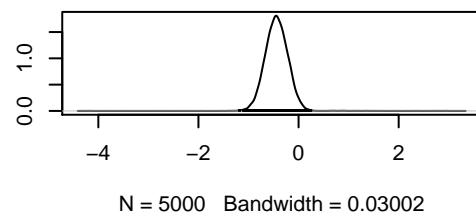
Density of b82



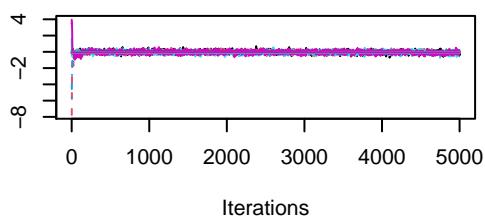
Trace of b83



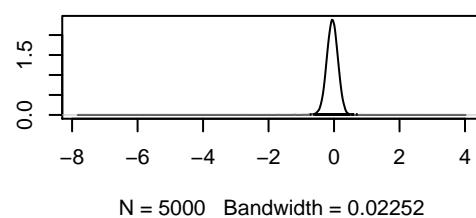
Density of b83



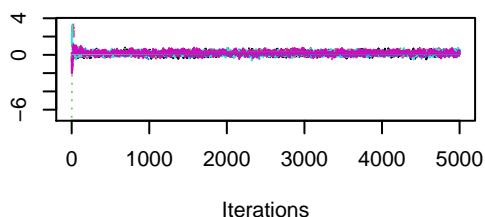
Trace of b84



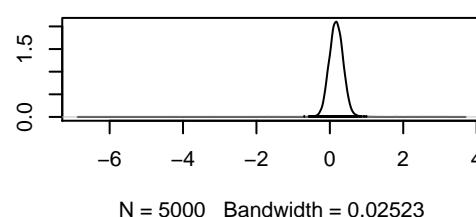
Density of b84



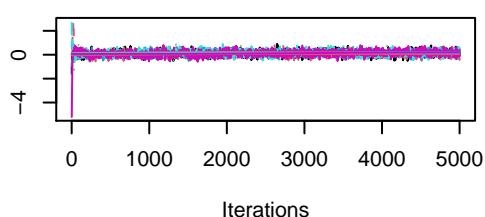
Trace of b85



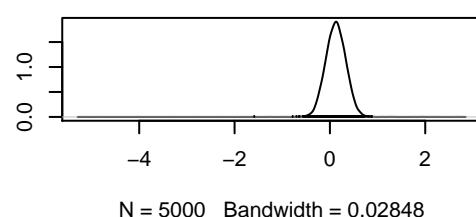
Density of b85



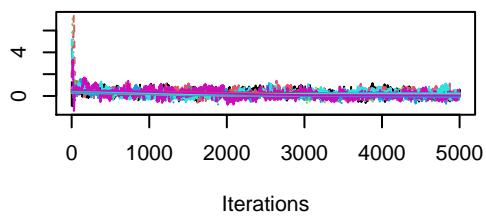
Trace of b86



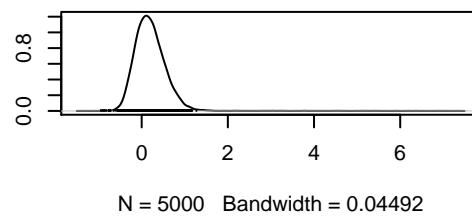
Density of b86



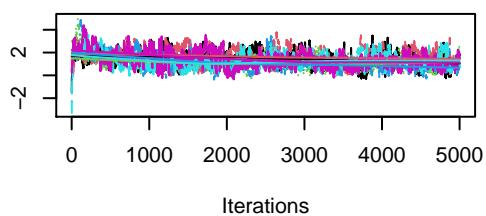
Trace of b91



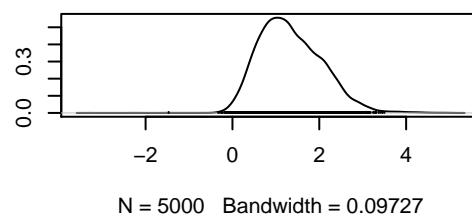
Density of b91



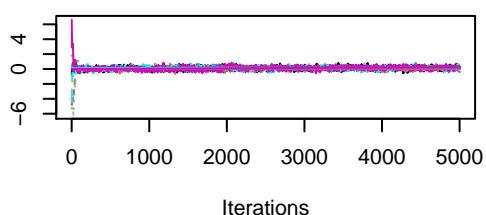
Trace of b92



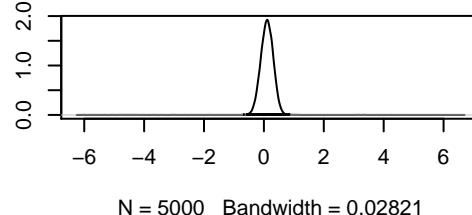
Density of b92



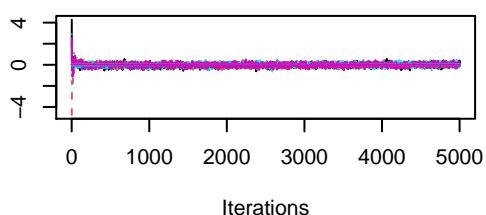
Trace of b93



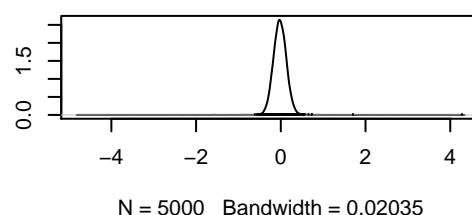
Density of b93



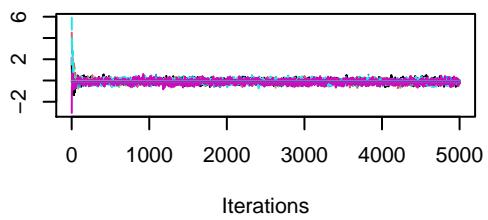
Trace of b94



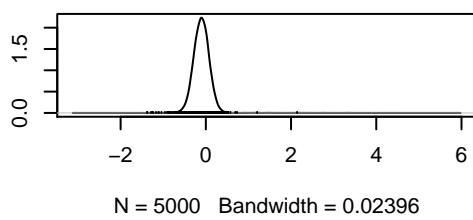
Density of b94



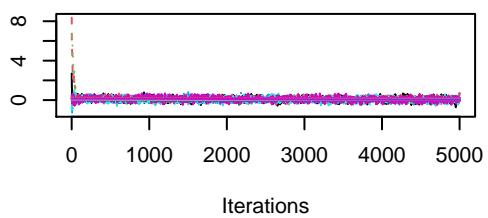
Trace of b95



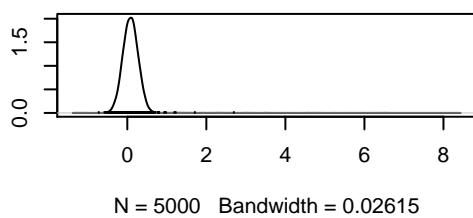
Density of b95



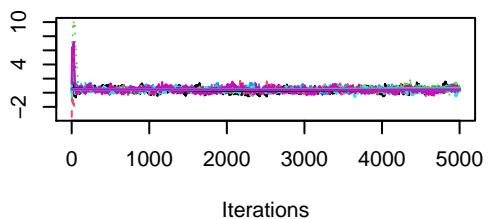
Trace of b96



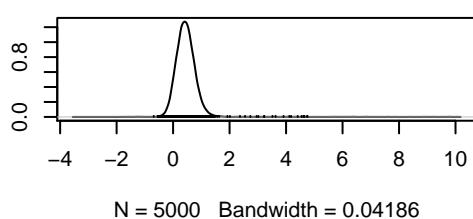
Density of b96



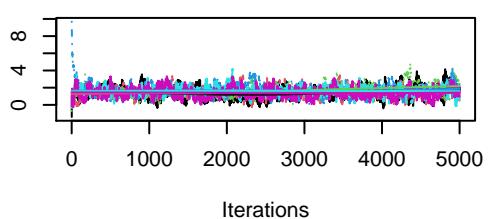
Trace of b101



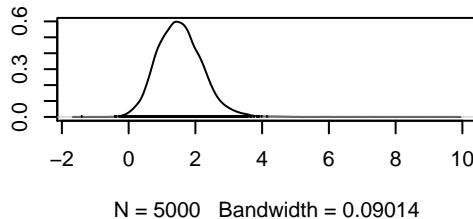
Density of b101



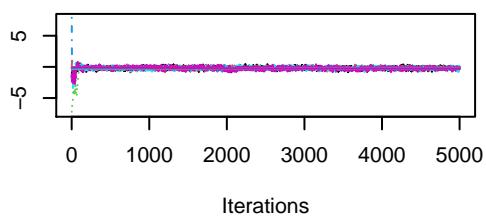
Trace of b102



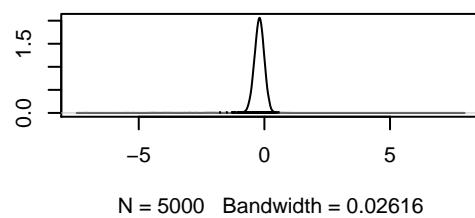
Density of b102



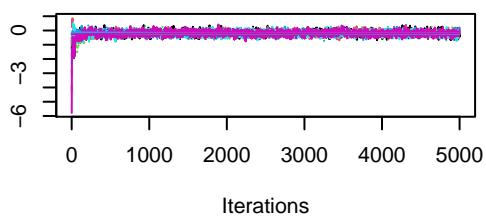
Trace of b103



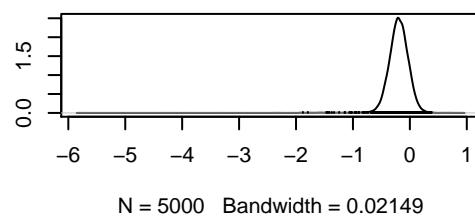
Density of b103



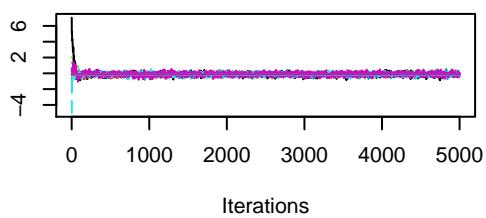
Trace of b104



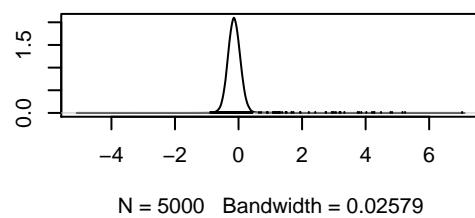
Density of b104



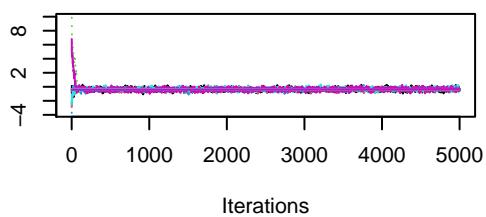
Trace of b105



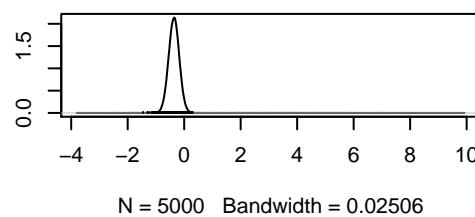
Density of b105



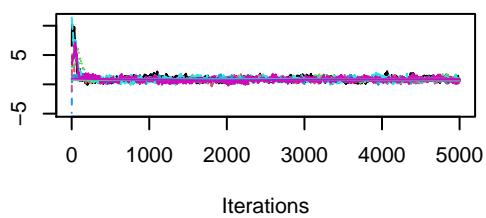
Trace of b106



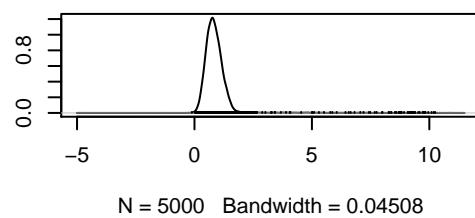
Density of b106



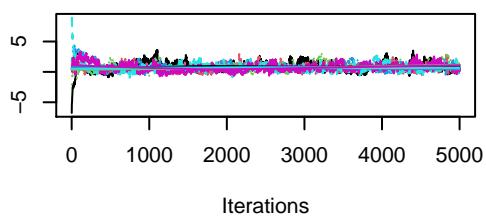
Trace of b111



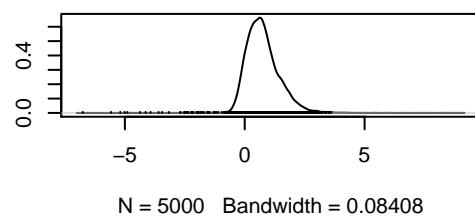
Density of b111



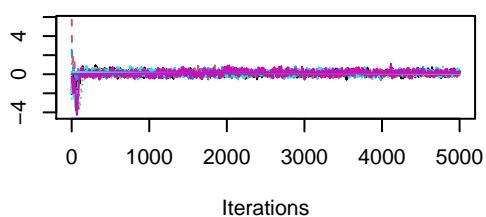
Trace of b112



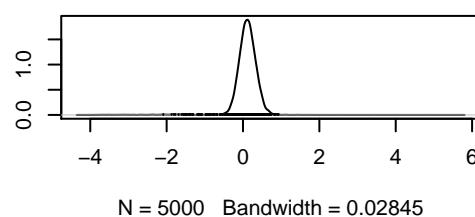
Density of b112



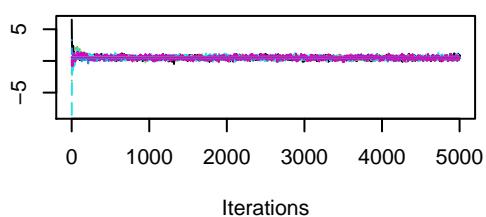
Trace of b113



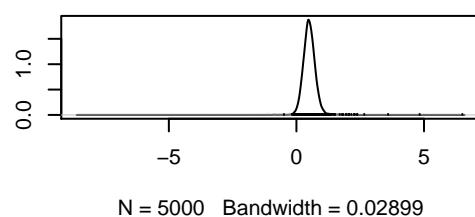
Density of b113



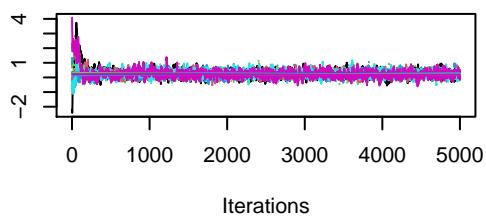
Trace of b114



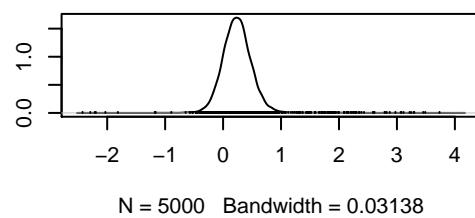
Density of b114



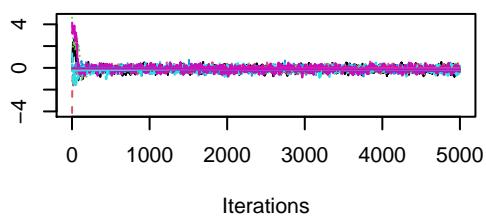
Trace of b115



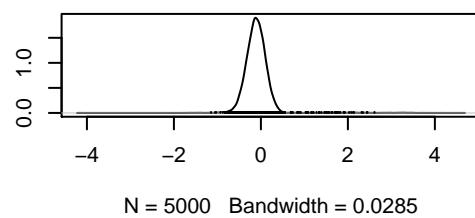
Density of b115



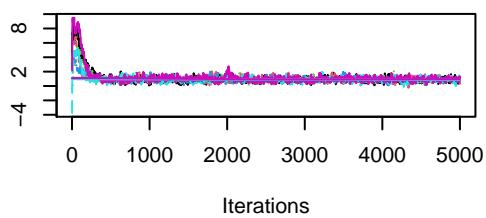
Trace of b116



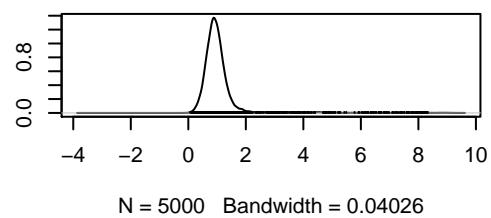
Density of b116



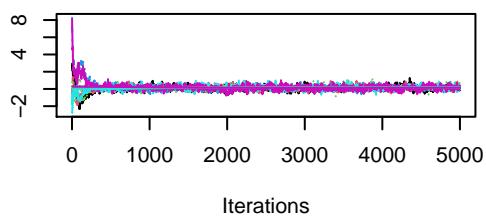
Trace of b121



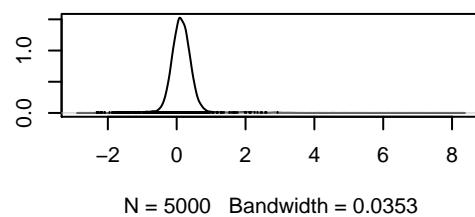
Density of b121



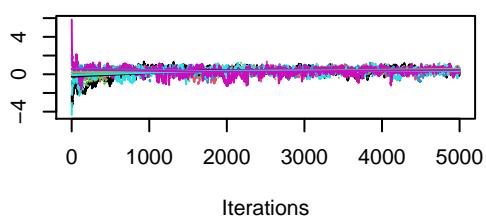
Trace of b122



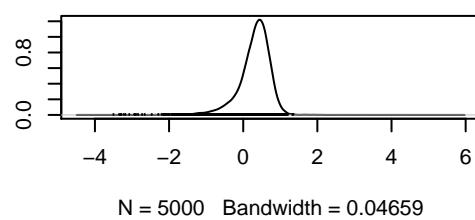
Density of b122



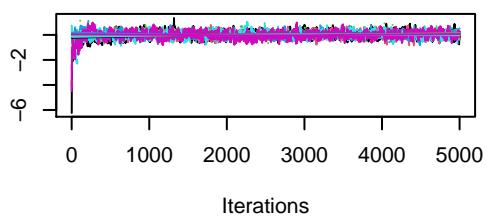
Trace of b123



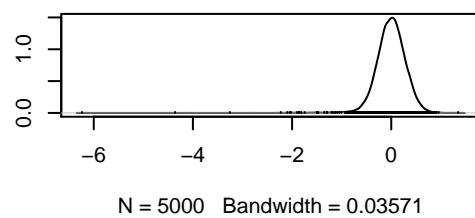
Density of b123



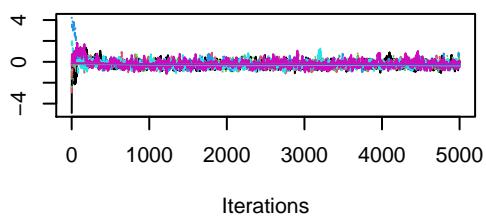
Trace of b124



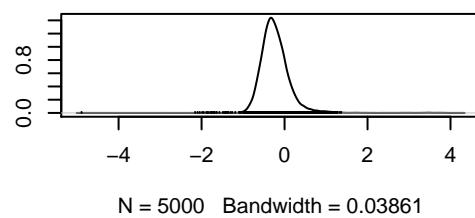
Density of b124



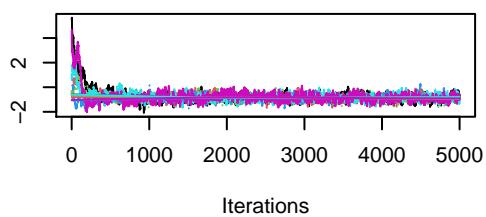
Trace of b125



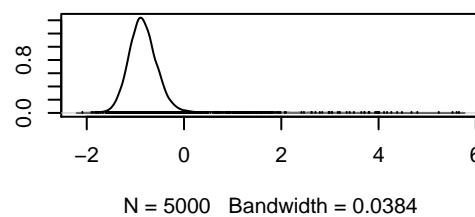
Density of b125



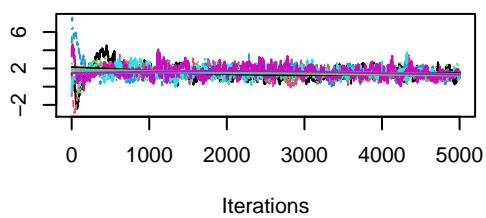
Trace of b126



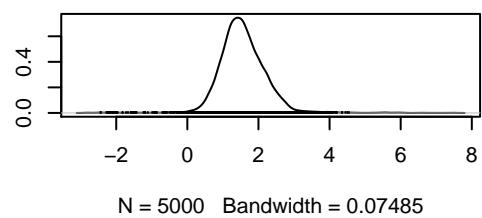
Density of b126



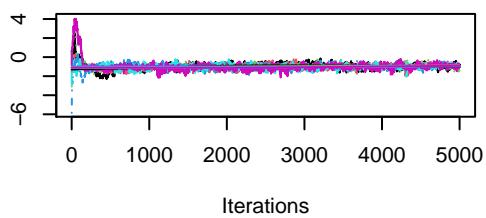
Trace of b131



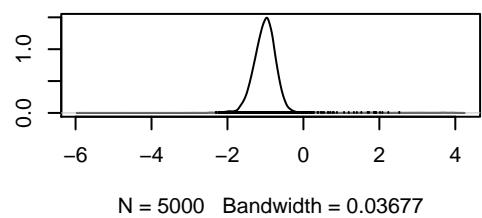
Density of b131



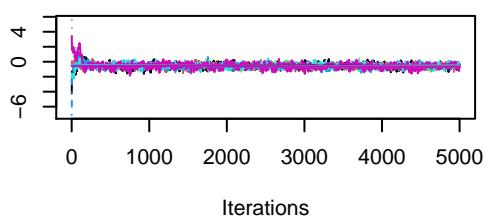
Trace of b132



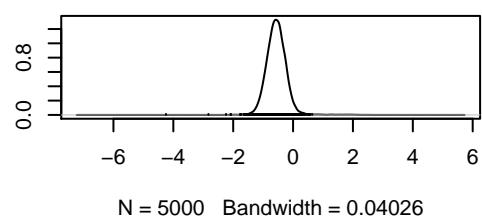
Density of b132



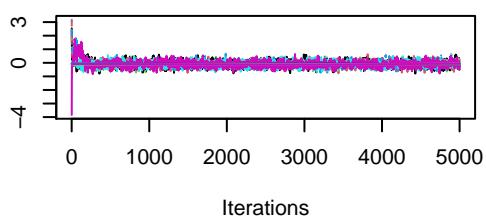
Trace of b133



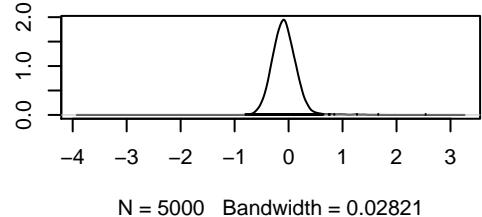
Density of b133



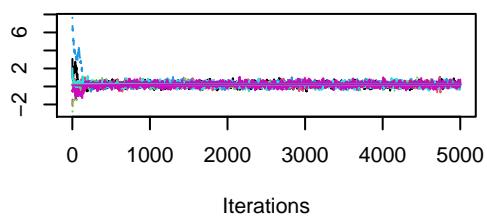
Trace of b134



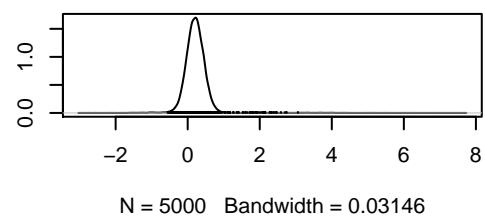
Density of b134



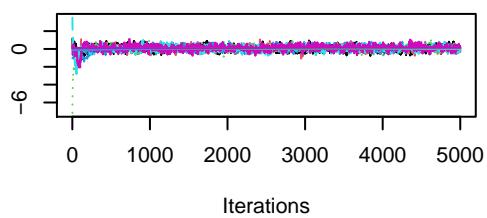
Trace of b135



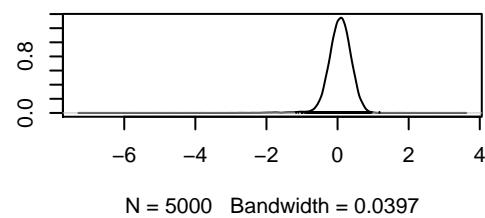
Density of b135



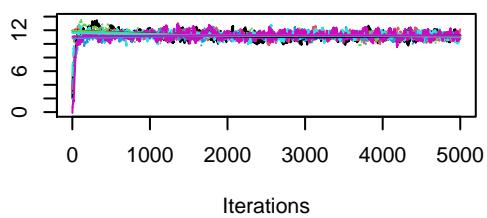
Trace of b136



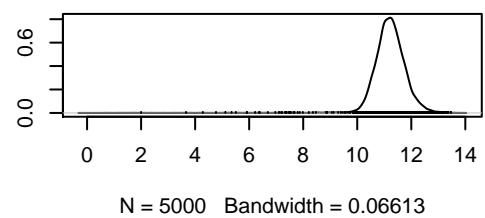
Density of b136



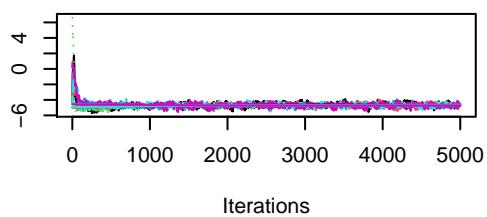
Trace of b141



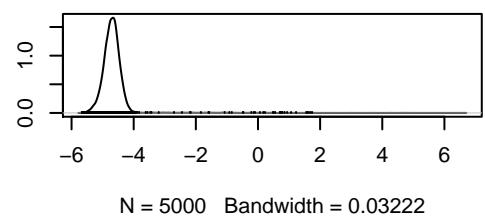
Density of b141



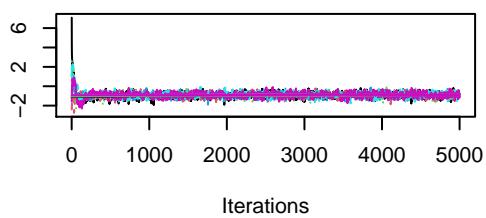
Trace of b142



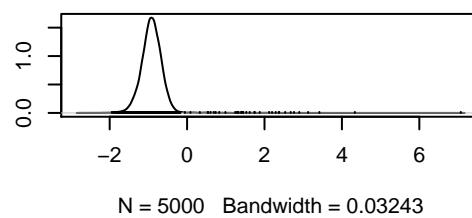
Density of b142



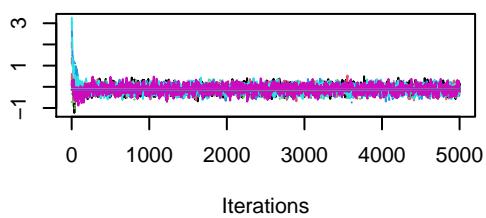
Trace of b143



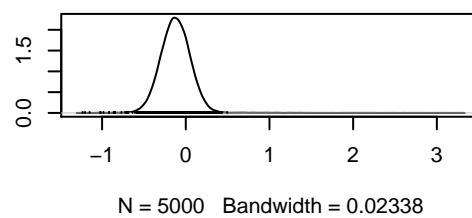
Density of b143



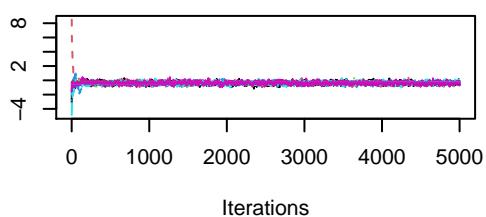
Trace of b144



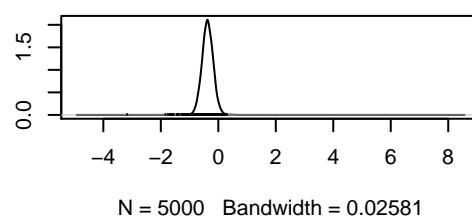
Density of b144



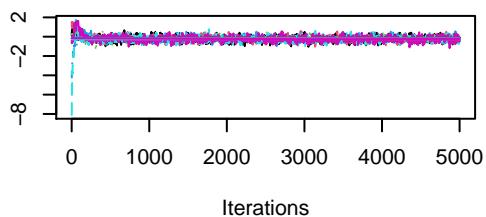
Trace of b145



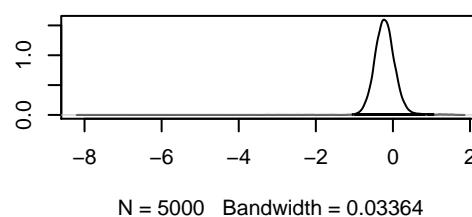
Density of b145



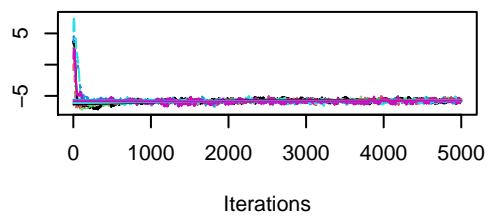
Trace of b146



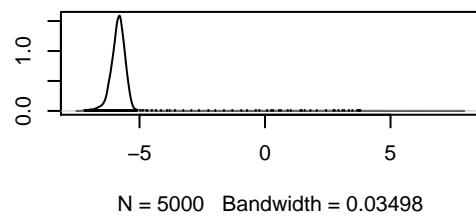
Density of b146



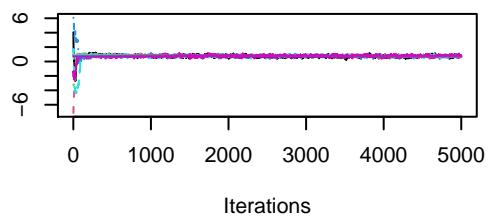
Trace of b151



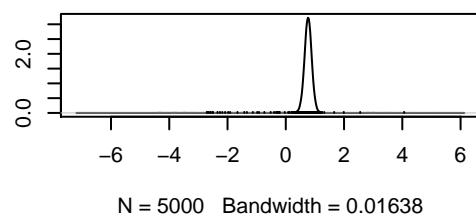
Density of b151



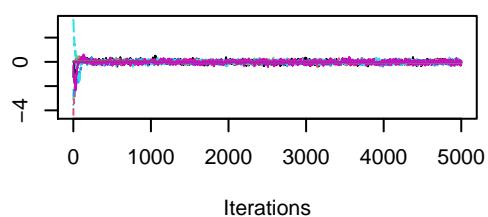
Trace of b152



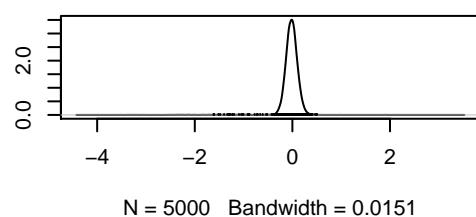
Density of b152



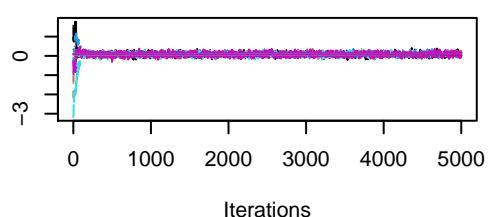
Trace of b153



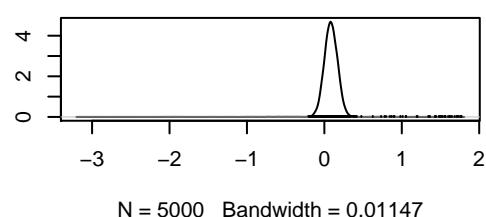
Density of b153



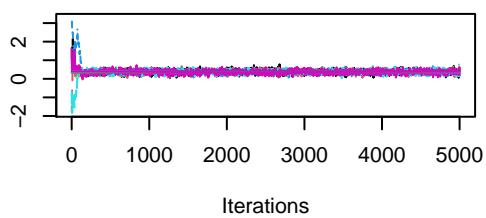
Trace of b154



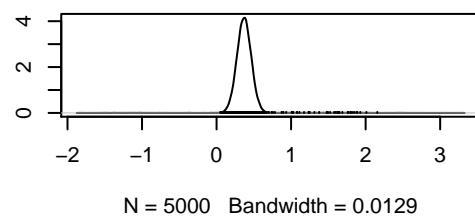
Density of b154



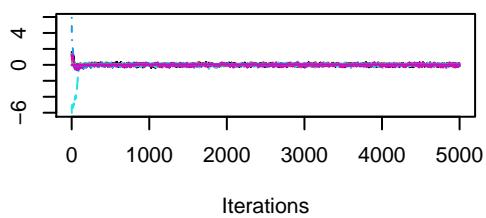
Trace of b155



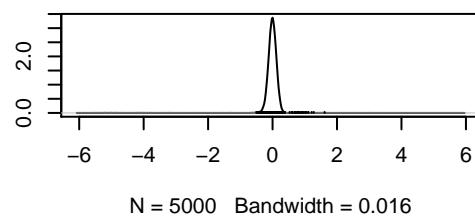
Density of b155



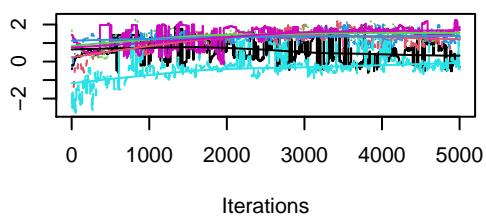
Trace of b156



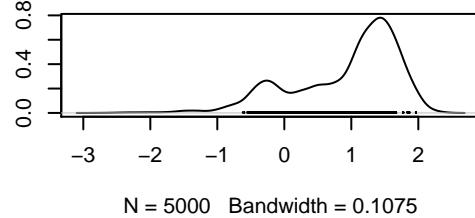
Density of b156



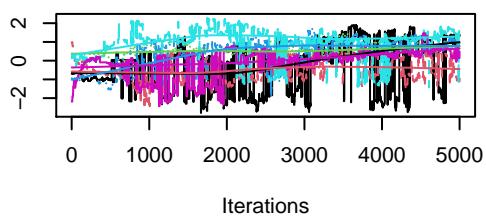
Trace of a1



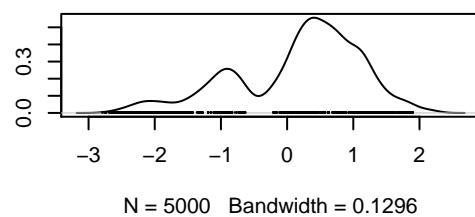
Density of a1

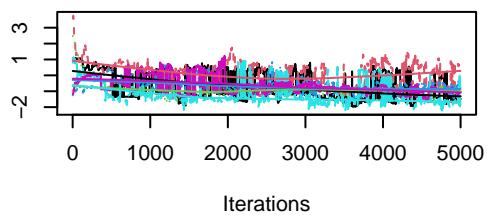
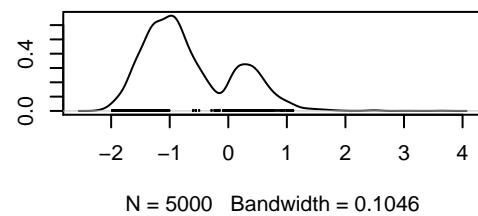
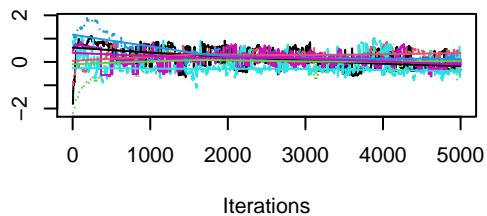
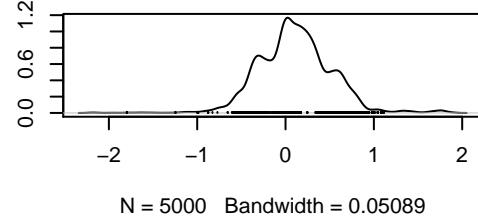
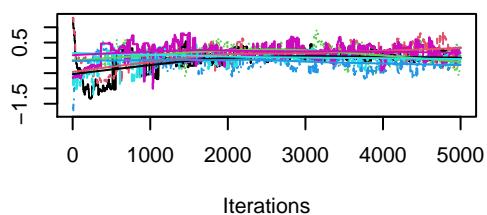
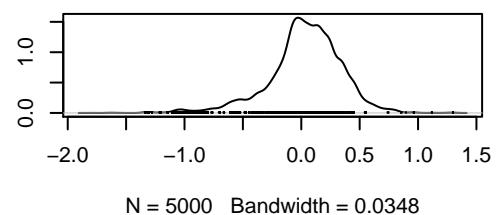
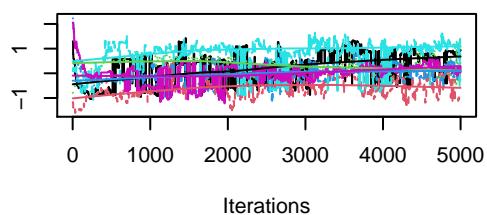
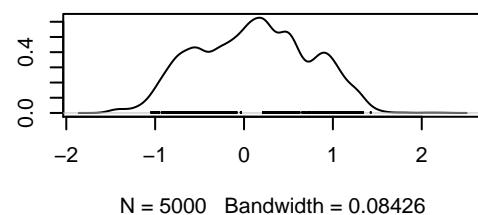
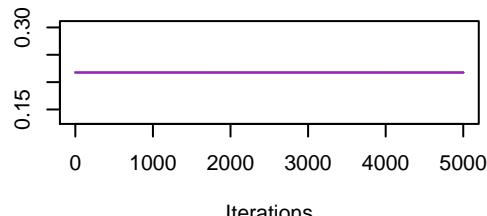
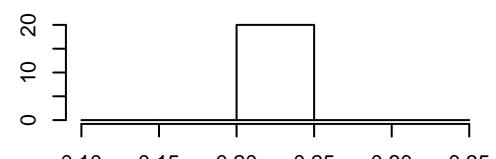


Trace of a2



Density of a2



Trace of a3**Density of a3****Trace of a4****Density of a4****Trace of a5****Density of a5****Trace of a6****Density of a6****Trace of jbw_result****Density of jbw_result**

From the trace plot, we can see clear sign of swapping.

Acceptance rate

Overall acceptance rate of Metropolis and swap for 6 different initializing points:

```
sapply(1:6, function(x) adapt_tempered_joint_new[[x]]$acpt)[c(1,3),]
```

```
##      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## a    0.1398  0.1654  0.1748  0.1860  0.1684  0.1902
## swap 0.2040  0.2086  0.2112  0.2118  0.2072  0.1972
```

Average acceptance rate of swap for each pair of chains from 6 different initializing points:

```
get.swap.ratio(adapt_tempered_joint_new)
```

```
##      [,1]     [,2]     [,3]     [,4]
## [1,] 0.2157506 0.2109501 0.2043974 0.1861175
## [2,] 0.2088509 0.2038835 0.2145816 0.2072498
## [3,] 0.2129555 0.2021773 0.2093596 0.2204600
## [4,] 0.2115987 0.2145188 0.2172879 0.2037187
## [5,] 0.2107280 0.2180209 0.1971714 0.2024000
## [6,] 0.1870110 0.1934501 0.2044335 0.2048699
```

According to the acceptance rate, we can see the result seems to be good. The acceptance rate of swap is around the setting of 0.234.

I also check the last 10 computed acceptance probabilities from 6 different initializing points to make sure that they are not taking values only 0 or 1.

```
adapt_tempered_joint_new[[1]]$swap_accept_prob[, 4991:5000]
```

```
##      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 0.002767508 0.49339389 0.01269335 0.077701292 0.053862628 0.06332043
## [2,] 1.000000000 0.18536537 0.93661273 1.000000000 1.000000000 0.02112318
## [3,] 0.003790572 0.01214730 0.01632248 0.003167342 0.001874987 0.03867605
## [4,] 0.132259742 0.04028831 0.35068504 0.222725936 0.202193585 1.000000000
##      [,7]     [,8]     [,9]     [,10]
## [1,] 0.178581946 0.005019876 0.241362383 0.032362228
## [2,] 1.000000000 1.000000000 0.005661739 0.240740406
## [3,] 0.001583271 0.002710892 0.015092545 0.003061901
## [4,] 1.000000000 0.105459754 1.000000000 0.644189423
```

```
adapt_tempered_joint_new[[2]]$swap_accept_prob[, 4991:5000]
```

```
##      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 0.058426331 0.097496729 0.750337620 0.008708556 0.672763932 0.260548159
## [2,] 0.162807722 0.042713527 0.109936116 1.000000000 0.280064752 0.122480274
## [3,] 0.006153514 0.003698777 0.116760850 0.003842692 0.001540253 0.005571195
## [4,] 1.000000000 0.943671611 0.009348717 1.000000000 1.000000000 1.000000000
##      [,7]     [,8]     [,9]     [,10]
## [1,] 0.36719264 0.1217327225 1.000000000 0.196000689
## [2,] 0.06148840 0.0967307756 0.206606612 0.047683994
## [3,] 0.03694692 0.5578110287 0.077229867 0.024663016
## [4,] 0.17715108 0.0007077746 0.001335545 0.002436993
```

```
adapt_tempered_joint_new[[3]]$swap_accept_prob[, 4991:5000]
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.040470112 0.01128985 1.000000000 0.051528910 0.014383969 0.182893893
## [2,] 0.407501667 0.25297557 0.40240246 1.000000000 0.154085027 1.000000000
## [3,] 0.009538039 0.03695020 0.01254710 0.002679811 0.682263708 0.001630832
## [4,] 0.840643252 0.03959832 0.02162642 0.266153563 0.007944492 1.000000000
##          [,7]      [,8]      [,9]      [,10]
## [1,] 0.30733482 0.416505145 0.05208484 0.05695430
## [2,] 0.01039755 0.006388895 0.07656975 0.01782183
## [3,] 1.000000000 0.417608985 0.25039570 1.000000000
## [4,] 0.01442744 0.075272197 0.01288400 0.00141199
```

```
adapt_tempered_joint_new[[4]]$swap_accept_prob[, 4991:5000]
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.002649073 0.021591235 0.0166639776 0.047234308 0.0150517571 0.0061376211
## [2,] 0.115060247 0.031315625 1.0000000000 0.345223916 1.0000000000 0.4386426467
## [3,] 0.013848483 0.004249631 0.0001693107 0.001055223 0.0001678011 0.0005239958
## [4,] 1.000000000 1.000000000 0.0481606028 0.011724313 0.0229368390 0.5790107899
##          [,7]      [,8]      [,9]      [,10]
## [1,] 0.0020644225 0.03836287 0.002409029 0.5284286437
## [2,] 1.0000000000 0.03344332 0.182570448 0.0001869949
## [3,] 0.0005083283 0.04414164 0.036489369 1.0000000000
## [4,] 0.3742522650 0.24059668 0.922060228 0.0061643756
```

```
adapt_tempered_joint_new[[5]]$swap_accept_prob[, 4991:5000]
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.040905255 0.007292781 0.004061047 0.02122921 0.01093218 0.01627840
## [2,] 0.030651939 0.631535229 0.383398167 0.81296852 0.30449856 0.14340446
## [3,] 0.008025124 0.005100709 0.052202535 0.12985461 0.01473431 0.04119661
## [4,] 1.000000000 0.330834871 0.021894608 0.01730039 0.67060859 1.000000000
##          [,7]      [,8]      [,9]      [,10]
## [1,] 0.005592362 0.01842380 0.509716221 0.12522870
## [2,] 1.000000000 0.02431492 0.006563735 0.08070543
## [3,] 0.005435400 0.27744527 0.324376900 0.03372364
## [4,] 1.000000000 0.06343382 0.261782164 0.38370377
```

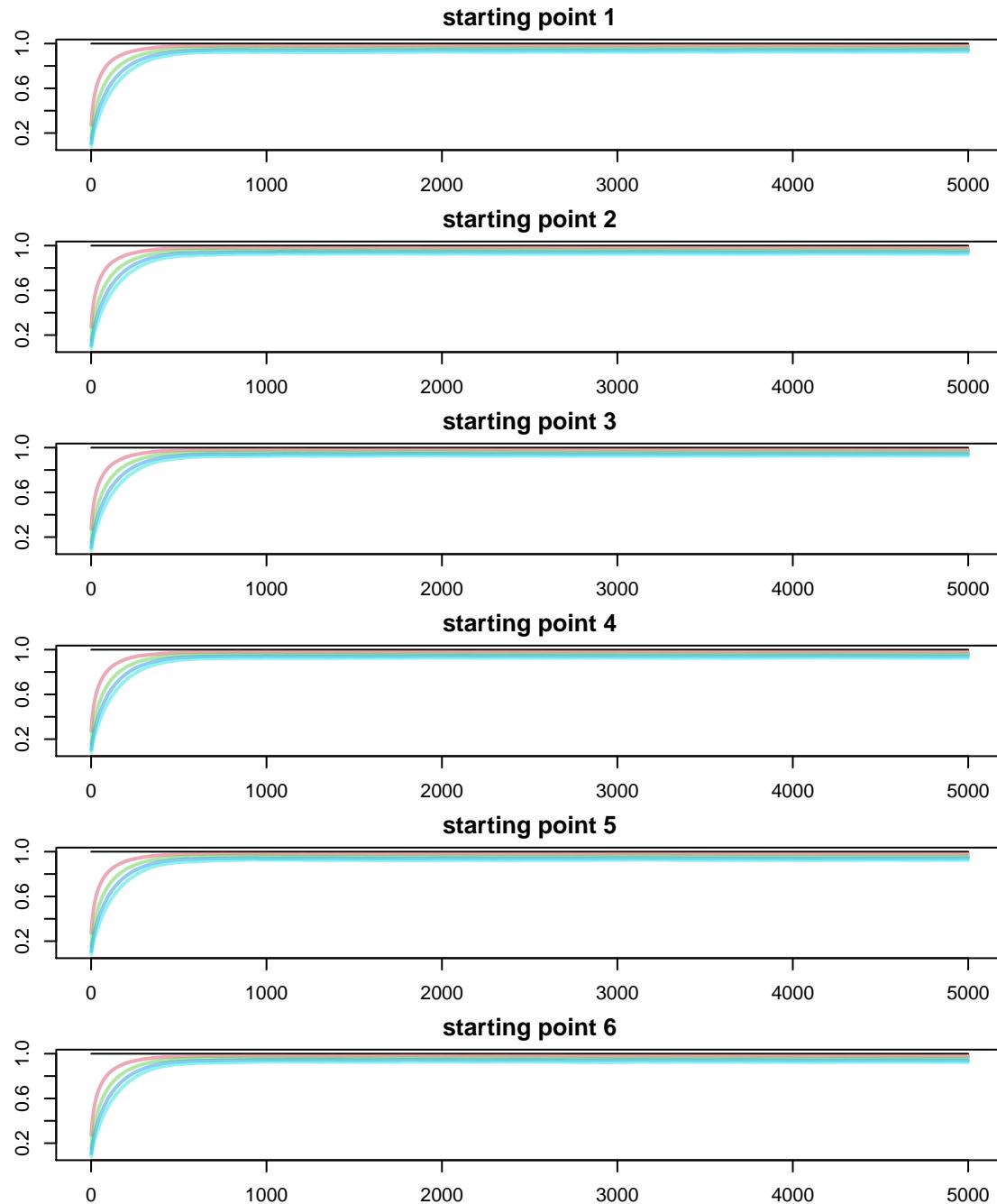
```
adapt_tempered_joint_new[[6]]$swap_accept_prob[, 4991:5000]
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.012767132 0.116644342 0.229215806 0.10202284 0.002227672 0.235548702
## [2,] 0.053471872 0.094686571 0.009169553 0.16296813 1.000000000 0.158076698
## [3,] 0.006911529 0.009721748 0.127693126 0.03592046 0.008658946 0.006496024
## [4,] 1.000000000 0.170524782 0.766776720 0.31983717 0.501476847 0.305101744
##          [,7]      [,8]      [,9]      [,10]
## [1,] 0.025702442 0.2229051724 1.0000000000 1.0000000000
## [2,] 0.003096486 0.0488335471 0.0138869455 0.2571813779
## [3,] 0.019946204 0.0003459921 0.0006292222 0.0005621225
## [4,] 1.000000000 0.9872950372 0.0694909001 0.0912382378
```

Inverse temperature

One thing not ideal is that other four inverse temperatures are all close to 1. However, the good thing is that they are not extremely close to 1. It seems that they converge to 0.98, 0.96, 0.94 and 0.92.

```
para_temp_plot(adapt_tempered_joint_new, 5, N_para)
```



```
sapply(1:6, function(x) adapt_tempered_joint_new[[x]]$temp[,5000])
```

```
##      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
```

```
## [1,] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## [2,] 0.9802783 0.9799704 0.9809234 0.9799904 0.9805748 0.9802447  
## [3,] 0.9625618 0.9610828 0.9624019 0.9612936 0.9606578 0.9615528  
## [4,] 0.9440591 0.9430492 0.9438993 0.9431239 0.9430140 0.9418570  
## [5,] 0.9262166 0.9251540 0.9272226 0.9246590 0.9246819 0.9242851
```