```
getwd()

library(EasyABC)
library(SimInf)

### take care that model was specified ok, otherwise error when applying ABC_mcmc
u0= data.frame(S=c(990), I=c(10), R=c(0))

### scale parameters
x= 1 ## such that beta is 0.1*1=0.1    # was:0.05
y= 1 ## such that gamma is 0.02*0.1=0.02     # was:0.2598


### example
# small example of how to pull the number infected from the output of a model run
model <- SIR(u0, 1:75, beta= 0.2, gamma=0.02)
#result <- run(model, threads = 1, seed=sample.int(1000000000,1))
result <- run(model, threads = 1)
plot(result)
abline(v=50)
str(result)
result@U[2,1] ### get number infected at time=1
### end example


### create function to use in ABC_mcmc
modelforABCmcmc2= function(parameters){
  library(SimInf)
  u0= data.frame(S=c(990), I=c(10), R=c(0))

  model <- SIR(u0, 1:75, beta= parameters[1], gamma=parameters[2])
  #result <- run(model, threads = 1, seed=sample.int(1000000000,1))
  result <- run(model, threads = 1)
  pop[1]=result@U[2,50]
  pop[2]=result@U[2,75]
  return(pop/sum(u0))
}



pop= numeric()
### try running it once, should return two population prevalence percentages
modelforABCmcmc2(c(0.2,0.02))



### set.seed for reproducability
set.seed(123)

### save the results from 10000 runs, take the means as the targets
saveres= matrix(c(0,0),10000,2)
for(i in 1:10000){
  saveres[i,]= modelforABCmcmc2(c(0.2,0.02))
}
hist(saveres[,1])
hist(saveres[,2])
mean(saveres[,1])
mean(saveres[,2])

### we call the target: truepop.prev
truepop.prev= c(round(mean(saveres[,1]),3), round(mean(saveres[,2]),3))
truepop.prev
plot(saveres[,1], saveres[,2], pch=16)
points(mean(saveres[,1]), mean(saveres[,2]), cex=1.5, col="red", pch=16)
```