**Supplementary material 1:** Bayesian analysis of qPCR data using MCMC.qpcr package.

#Bayesian analysis of qPCR data using MCMC.qpcr package

### see http://www.bio.utexas.edu/research/matz\_lab/matzlab/methods\_files/mcmc.qpcr.tutorial.pdf

#for a full tutorial

####Before starting: ####

# you should install the following R-package with:

# install.packages("MCMC.qpcr")

# install.packages("SLqPCR")

# Alternatively: source("https://bioconductor.org/biocLite.R"); biocLite("SLqPCR")

# install.packages("ggplot2")

library("MCMC.qpcr")

library("SLqPCR")

library("ggplot2")

#change the working directory if needed

#setwd("your/folder/path/here")

###you need a file with the efficiency tables and another with the Cq values in .csv or .txt format. Gene names in both tables needs to match

######Efficiency calculation######

efficienty.table = read.table("Efficiency\_tables\_allGenes.csv", sep=",", dec=".", h=T)

#create a table with efficiency values. Change plot=T if wish to print the plots

eff.allGene = PrimEff(efficienty.table, plot=F)

(E.values = eff.allGene[,1:2]) # table with the values

#Load the Ct values

Gene.table = read.table("Hfulgens\_Gill\_AllGenes\_MCMCformat.csv", sep=",", dec=".", h=T)

colnames(Gene.table)[3] = "sample"

####REFERENCE GENES SELECTION#######

#Actine, RP5 and E3UL are our target reference genes. Additional genes are included to test stability

#list the names of the genes to test

test.HKG = c("Actine", "RP5", "E3UL", "CS", "PK", "LCF","SOD", "COXIII", "PEPCK", "HK")

#Convert table into Genorm format

Gene.genorm = cq2genorm(data = Gene.table,

genes = test.HKG,

effic = E.values,

noamp = NA)

n.HKG = length(test.HKG)

#We start by ranking the selected reference/housekeeping genes. The function #selectHKgenes proceeds stepwise.That is, the

gene stability measure M of all candidate genes is computed and the gene with the highest

#M value is excluded. Then, the gene stability measure M for the remaining gene is

#calculated and so on. This procedure is repeated until two respectively minNrHK is reached.

###Select the reference genes

#minNrHK refers to the minimun numbers of references. default is two

minNrHK = 2

ref.genes = selectHKgenes(Gene.genorm, method ="Vandesompele",

minNrHK = minNrHK, geneSymbol = names(Gene.genorm), na.rm = TRUE)

reference = paste(ref.genes$ranking[1:minNrHK], collapse = '/')

ranks = data.frame(ranks = 1:(n.HKG-(minNrHK-1)), genes = c(reference, ref.genes$rank[-c(1:minNrHK)]),

M = rev(ref.genes$meanM))

ranks$genes = factor(ranks$genes, levels=ranks[order(-ranks$M), "genes"])

####GeNorm Results####

ranks; qplot(genes,M,data=ranks, xlab="Genes", ylab="M")

#RP5/E3UL are better suited as reference

####MCMC model. This if performed in the classic method i.e. using reference genes

G.classic = cq2log(

data = Gene.table,

genecols = c(6:28), #columns with the genes to analyse

condcols = c(3:5), #columns with the experimental conditions

effic = E.values,

noamp = 38)

#####Define Experiment:control at Temperature: 18.C as the global reference

G.classic$Experiment = relevel(G.classic$Experiment, ref="control")

G.classic$Temperature = relevel(G.classic$Temperature, ref="18.C")

#fit of a classic model with two-way design. "pr"and "pl" commands are included to test the model

G.classic.model = mcmc.qpcr.classic(

fixed="Temperature+Experiment+Temperature:Experiment",

#random = "sample",

data = G.classic,

controls=c("RP5","E3UL"),

pr = T,

pl =T)

diagnostic.mcmc(

model = G.classic.model,

col="grey50")

####Detect outliers

outs = outlierSamples(G.classic.model, G.classic)

###remove outliers from the list

G.classic = G.classic[!(G.classic$sample %in% outs), ]

#Repeat model witouth outliers

G.classic.model = mcmc.qpcr.classic(

fixed="Temperature+Experiment+Temperature:Experiment",

data = G.classic,

#random = "sample",

controls=c("RP5","E3UL")

)

summary(G.classic.model)

#change relative=T, to get the expression relative to the control (control not ploted)

G.Rel.classic = HPDsummary(model = G.classic.model,data = G.classic, xgroup="Temperature", relative=F, log.base=2)

trellisByGene(G.Rel.classic, xFactor="Temperature", groupFactor="Experiment", nrow=4,

facetScales="free" )

G.pairwise.table = G.Rel.classic$geneWise #pairwise table with the FC (above) and p values (below)

#save it as a CSV file:

#write.table(G.pairwise.table, "MCMC\_significance\_table.csv", sep=";", dec=",",row.names=T)

#Use: G.pairwise.table$xxxx to select a specific gene