

# Analysis of HPC results

Willem Vervoort, Michaela Dolk & Floris van Ogtrop

2017-05-15

```
# root dir
knitr::opts_knit$set(root.dir = "C:/Users/rver4657/ownCloud/Virtual Experiments/VirtExp")
knitr::opts_chunk$set(echo = TRUE)
# LOAD REQUIRED PACKAGES # #####
library(pander)
library(tidyverse)
library(xts)
library(zoo)
library(ggplot2)
library(reshape2)
library(Rcpp)
library(hydromad)
```

This rmarkdown document and the resulting pdf are stored on github. All directories (apart from the root working directory) refer to the directories in this repository.

## Introduction

This document is related to the manuscript “Disentangling climate change trends in Australian streamflow” (vervoort et al.), submitted to Journal of Hydrology. This is the sixth part of the series that analyses the results from the GR4J and SimHyd model fitting on the High Performance computer Artemis at the University of Sydney.

In particular, this part extracts the best parameters of the modelling, plots the performance distributions and extracts the residuals to be analysed in a further script using Mann Kendall (this is separated because this takes quite long to run). Finally a comparison between the non-parametric epsilon (sensitivity) and the model based epsilon is plotted for both gridded and non-gridded rainfall.

This is ultimately related to *Figure 3*, *Figure 4* and *Figure 9* in the manuscript. *Figure 9* is generated in a different script, as this is the Mann Kendall analysis of the residuals of the models.

To recap, we have 4 different High Performance Computing results for each of the 13 catchments. In each of these the two models (GR4J and SimHyd) were fitted to 10 years of climate data (1970 - 1980). However, there are results for both station rainfall and gridded rainfall to make sure there is no difference between these. As we have seen in the non-parametric epsilon calculation, there is a major difference between the two rainfall data sets, and it is not necessarily clear, which one of these is a better data set.

The model fitting is based on the shuffled complex evolution optimisation in Hydromad as described in more detail in the paper. Essentially the model was fitted 10 times using the `FitBySCE()` function in the Hydromad package. The scripts related to the high performance computing and the HPC scripts are stored in the Rcode/HPC folder.

The HPC results are not stored on Github, as the files are too large, but are stored in the Cloudstor data directory.

## 1. Load basic data and define storage

This loads all the basic climate data and the catchment characteristics.  
It also compiles the SimHyd model code.

```
load("data/DailyDataIncludingGridded.Rdata")
load("data/ClimCh_project_MD.Rdata")
# correct the column name of maxT in GridRainAllDataout
colnames(GridRainAllDataout)[5] <- "MaxT"
# change to tibble
GridAlldata <- as_tibble(cbind(GridRainAllDataout,
                              Date=rep(time(flow_zoo),nrow(Stations))))
# now use spread
GridAlldata_wide <- spread(GridAlldata[,c("Station","gridRain","Date")],
                           key=Station, value=gridRain)
Gridrain_zoo <- zoo(GridAlldata_wide[,2:14],order.by=time(flow_zoo))

# compile SimHyd
rcode_dir <- paste(getwd(),"RCode/HPC",sep="/")
source(paste(rcode_dir,"Simhyd.r",sep="/"))
```

Define the beginning and end date for the modelling

```
start.date <- as.Date("1981-01-01")
end.date <- as.Date("2010-12-31")
```

As a first step define storage for the results of the modelling. These will get renamed for each of the individual modelling results

```
sum_Res <- list()
Chiew_Res <- list()

mod_Res <- list()

Chiew <- data.frame(station=character(length=10),eta_p=numeric(length=10),
                   eta_e=numeric(length=10))

Results <- data.frame(station=character(length=10),
                      Mod.r.sq=numeric(length=10),
                      Mod.bias=numeric(length=10))

Residuals <- list()

# some other auxillary data frames
pred_results <- data.frame(Pmin15ET0=numeric(length=nrow(flow_zoo)),
                           Pmin10ET0=numeric(length=nrow(flow_zoo)),
                           POET0=numeric(length=nrow(flow_zoo)),
                           Pplus10ET0=numeric(length=nrow(flow_zoo)),
                           Pmin15ETplus5=numeric(length=nrow(flow_zoo)),
                           Pmin10ETplus5=numeric(length=nrow(flow_zoo)),
                           POETplus5=numeric(length=nrow(flow_zoo)),
                           Pplus10ETplus5=numeric(length=nrow(flow_zoo)),
                           Pmin15ETplus10=numeric(length=nrow(flow_zoo)),
                           Pmin10ETplus10=numeric(length=nrow(flow_zoo)),
                           POETplus10=numeric(length=nrow(flow_zoo)),
                           Pplus10ETplus10=numeric(length=nrow(flow_zoo)))
```

## 2. GR4J model results with station rainfall

Extract the modelling results, rerun the model, do the Chiew (2006) amplification analysis and write away the results.

```
# find the list of files with GR4J results
filelist <- dir("../Projectdata/HPCResults", pattern = "GR4JCalibOutput")

for (i in seq_along(filelist)) {
  #i <- 1
  # load the rainfall, ET and flow data
  pred_data <- window(merge(flow_zoo[,i], rain_zoo[,i], maxT_zoo[,i]),
                      start=start.date, end=end.date)
  colnames(pred_data) <- c("Q","P","E")
  # # another storage data frame for the residuals
  resid_out <- data.frame(matrix(0,ncol=10,nrow=nrow(pred_data)))
  # load the relevant output
  load(paste("../Projectdata/HPCResults/",filelist[i],sep=""))
  # extract the model and update with the parameters
  Mod <- Output$mod

  mod_Res[[i]] <- Output$Store
  Chiew[,1] <- Stations[i,1]
  Results[,1] <- Stations[i,1]

  # run through all iterations
  for (j in 1:(nrow(Output$Store))) {
    # testing
    #j <- 1
    # update the model with the final fitted parameters, for each iteration
    Mod <- update(Mod, x1=Output$Store[j,8],x2=Output$Store[j,5],
                  x3=Output$Store[j,6],x4=Output$Store[j,7],
                  etmult=Output$Store[j,9],
                  return_state=F)

    # now predict the model output and use pred_data
    pred_mod <- predict(Mod,newdata=pred_data, all=T,na.rm=F)

    # store the residuals
    resid_out[,j] <- pred_data$Q-pred_mod

    Results[j,2:3] <- c(summary(Mod)$r.squared,summary(Mod)$rel.bias)

    # Now run the Chiew 2006 simulations on all the data
    mu <- cbind(rep(c(-15,-10,0,10),3),c(rep(0,4),rep(5,4),rep(10,4)))
    # Create the precipitation and ET data variations
    # station data
    RAIN <- rain_zoo
    test <- list()
```

```

for (k in 1:nrow(mu)) {
  temp <- as.data.frame(cbind((1+mu[k,1]/100)*RAIN[,i],
                              (1+mu[k,2]/100)*maxT_zoo[,i]))
  test[[k]] <- do.call(cbind,apply(temp,2,function(x) aggregate(x,
                                                                  list(year=format(time(flow_zoo),"%Y")),sum,na.rm=T)))
  test[[k]] <- test[[k]][,-3]
}
clim_adj <- do.call(rbind,test)

# now run the different pred results
for (k in 1:ncol(pred_results)) {
  # run the model over all data
  pred_data2 <- window(merge(flow_zoo[,i],
                              (1+mu[k,1]/100)*RAIN[,i],
                              (1+mu[k,2]/100)*maxT_zoo[,i]))
  colnames(pred_data2) <- c("Q","P","E")

  pred_results[,k] <- predict(Mod,newdata=pred_data2, all=T,na.rm=F)
}

# summarise the data annually
pred_ann <- apply(pred_results,2,
                  function(x) aggregate(x,list(year=format(time(flow_zoo),"%Y")),
                                          sum,na.rm=T))

ann_flow <- rep(pred_ann[[1]][,2],6)
pred_t <- do.call(rbind,pred_ann)
# Now add the ET and precipitation data
pred_ann <- data.frame(pred_t,rain=clim_adj[,2],maxT=clim_adj[,3])
# summarise base rain and temp
ann_rain <- rep(aggregate(RAIN[,i],list(year=format(time(flow_zoo),"%Y")),
                          sum,na.rm=T),6)
ann_maxT <- rep(aggregate(maxT_zoo[,i],
                          list(year=format(time(flow_zoo),"%Y")),
                          sum,na.rm=T),6)

# Now calculate the difference
pred_diff <- pred_ann
pred_diff[,2] <- pred_diff[,2] - ann_flow
pred_diff[,3] <- pred_diff[,3] - ann_rain
pred_diff[,4] <- pred_diff[,4] - ann_maxT
# Now fit a linear model (least squares (Chiew, 2006))
fit <- lm(x~rain + maxT,data=pred_diff)
# store the results
Chiew[j,2:3] <- coef(fit)[2:3]

}
Chiew_Res[[i]] <- Chiew
sum_Res[[i]] <- Results
Residuals[[i]] <- resid_out
}

```

We can now temporary write away the results and the residuals and make some initial plots. Further plots will be generated later when comparing to the non-parametric epsilon ( $\epsilon$ ).

```

# write away the results
OutputTrends <- do.call(rbind,sum_Res)
write.csv(OutputTrends,
          file="../ProjectData/GR4JHPC_modelperformance.csv",
          row.names=F)

OutputChiew <- do.call(rbind,Chiew_Res)
write.csv(OutputChiew,file="../ProjectData/GR4JHPC_ChiewAnalysis.csv",
          row.names=F)

save(Residuals,file="../ProjectData/residuals/GR4JHPCresiduals.Rdata")

OutputMod_GR4J <- do.call(rbind,mod_Res)
save(OutputMod_GR4J,file="../ProjectData/GR4JHPCModelResults.Rdata")

OutputMod_GR4J <- OutputMod_GR4J[,c(1:4,ncol(OutputMod_GR4J))]

OutputMod_GR4J$model <- "GR4JStationData"

p <- ggplot(OutputMod_GR4J,aes(station,r.squared)) + geom_boxplot()
p <- p + stat_summary(fun.y=mean, geom="point", shape=16,
                     size=5,aes(col=rel.bias))
print(p)

```

### 3. GR4J model results with gridded rainfall

Extract the modelling results

```

# find the list of files with GR4J results
filelist <- dir("../Projectdata/HPCResults", pattern = "GR4JGridCalibOutput")

for (i in seq_along(filelist)) {
  # load the rainfall, ET and flow data
  pred_data <- window(merge(flow_zoo[,i], Gridrain_zoo[,i], maxT_zoo[,i]),
                     start=start.date, end=end.date)
  colnames(pred_data) <- c("Q","P","E")
  # # another storage data frame for the residuals
  resid_out <- data.frame(matrix(0,ncol=10,nrow=nrow(pred_data)))
  # load the relevant output
  load(paste("../Projectdata/HPCResults/",filelist[i],sep=""))
  # extract the model and update with the parameters
  Mod <- Output$mod

  mod_Res[[i]] <- Output$Store
  Chiew[,1] <- Stations[i,1]
  Results[,1] <- Stations[i,1]

  # run through all iterations
  for (j in 1:(nrow(Output$Store))) {
    # testing
    #j <- 1

```

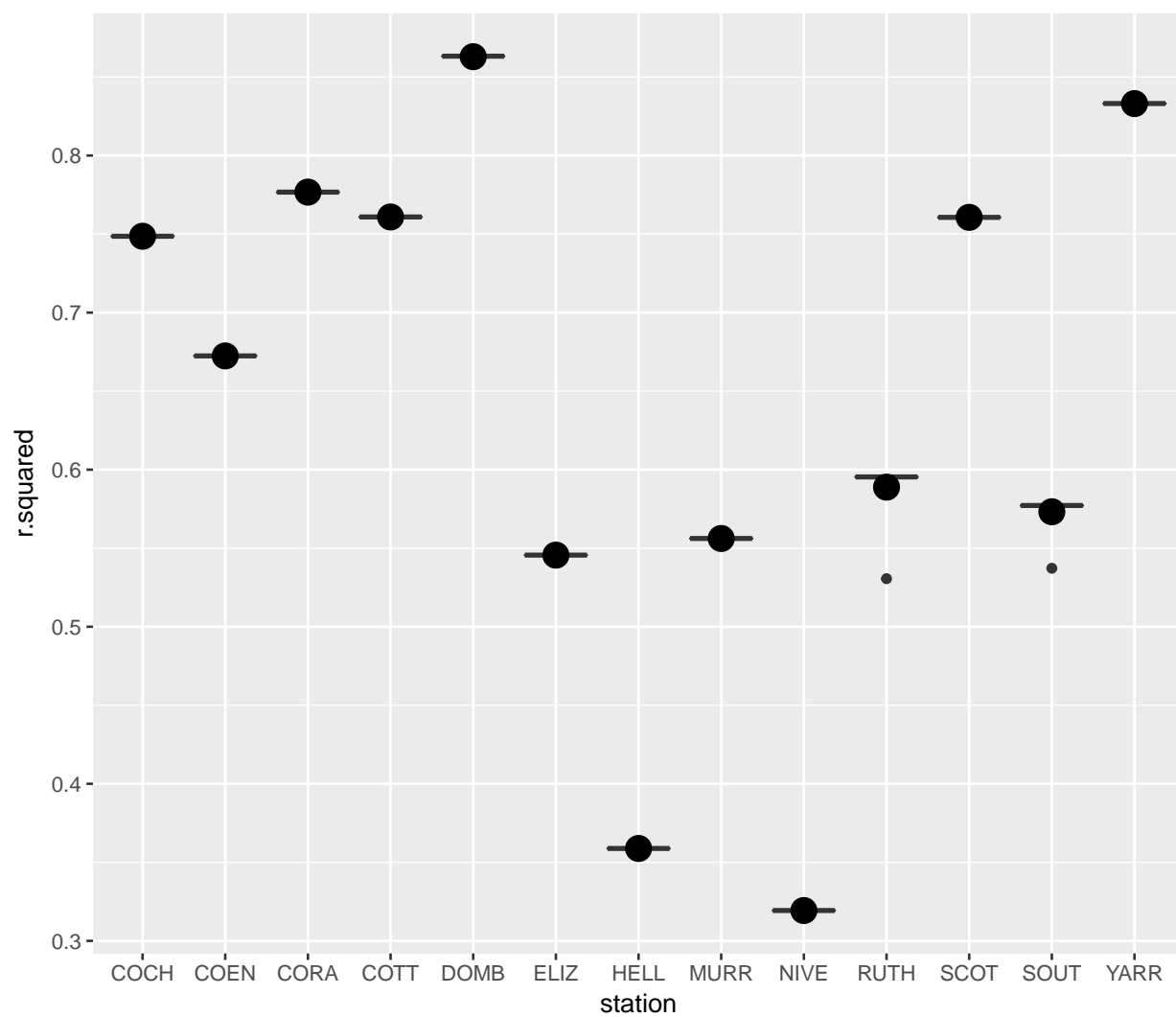


Figure 1: Calibration results for the GR4J model across 10 iterations using station rainfall data.

```

# update the model with the final fitted parameters, for each iteration
Mod <- update(Mod, x1=Output$Store[j,8],x2=Output$Store[j,5],
              x3=Output$Store[j,6],x4=Output$Store[j,7],
              etmult=Output$Store[j,9],
              return_state=F)

# now predict the model output and use pred_data
pred_mod <- predict(Mod,newdata=pred_data, all=T,na.rm=F)

# store the residuals
resid_out[,j] <- pred_data$Q-pred_mod

Results[j,2:3] <- c(summary(Mod)$r.squared,summary(Mod)$rel.bias)

# Now run the Chiew 2006 simulations on all the data
mu <- cbind(rep(c(-15,-10,0,10),3),c(rep(0,4),rep(5,4),rep(10,4)))
# Create the precipitation and ET data variations
# station data
RAIN <- Gridrain_zoo
test <- list()
for (k in 1:nrow(mu)) {
  temp <- as.data.frame(cbind((1+mu[k,1]/100)*RAIN[,i],
                              (1+mu[k,2]/100)*maxT_zoo[,i]))
  test[[k]] <- do.call(cbind,apply(temp,2,function(x) aggregate(x,
                                                                list(year=format(time(flow_zoo),"%Y")),sum,na.rm=T)))
  test[[k]] <- test[[k]][,-3]
}
clim_adj <- do.call(rbind,test)

# now run the different pred results
for (k in 1:ncol(pred_results)) {
  # run the model over all data
  pred_data2 <- window(merge(flow_zoo[,i],
                             (1+mu[k,1]/100)*RAIN[,i],
                             (1+mu[k,2]/100)*maxT_zoo[,i]))
  colnames(pred_data2) <- c("Q", "P", "E")

  pred_results[,k] <- predict(Mod,newdata=pred_data2, all=T,na.rm=F)
}

# summarise the data annually
pred_ann <- apply(pred_results,2,
                  function(x) aggregate(x,list(year=format(time(flow_zoo),"%Y")),
                                         sum,na.rm=T))

ann_flow <- rep(pred_ann[[1]][,2],6)
pred_t <- do.call(rbind,pred_ann)
# Now add the ET and precipitation data
pred_ann <- data.frame(pred_t,rain=clim_adj[,2],maxT=clim_adj[,3])
# summarise base rain and temp
ann_rain <- rep(aggregate(RAIN[,i],list(year=format(time(flow_zoo),"%Y")),
                          sum,na.rm=T),6)
ann_maxT <- rep(aggregate(maxT_zoo[,i],
                          list(year=format(time(flow_zoo),"%Y")),

```

```

sum,na.rm=T),6)

# Now calculate the difference
pred_diff <- pred_ann
pred_diff[,2] <- pred_diff[,2] - ann_flow
pred_diff[,3] <- pred_diff[,3] - ann_rain
pred_diff[,4] <- pred_diff[,4] - ann_maxT
# Now fit a linear model (least squares (Chiew, 2006))
fit <- lm(x~rain + maxT,data=pred_diff)
# store the results
Chiew[j,2:3] <- coef(fit)[2:3]

}
Chiew_Res[[i]] <- Chiew
sum_Res[[i]] <- Results
Residuals[[i]] <- resid_out
}

```

We can now temporary write away the results and the residuals and make some initial plots. Further plots will be generated later when comparing to the non-parametric epsilon ( $\epsilon$ ).

```

# write away the results
OutputTrends <- do.call(rbind,sum_Res)
write.csv(OutputTrends,
          file="../ProjectData/GR4JGridHPC_modelperformance.csv",
          row.names=F)

OutputChiew <- do.call(rbind,Chiew_Res)
write.csv(OutputChiew,file="../ProjectData/GR4JGridHPC_ChiewAnalysis.csv",
          row.names=F)

save(Residuals,file="../ProjectData/residuals/GR4JGridHPCresiduals.Rdata")

OutputMod_GridGR4J <- do.call(rbind,mod_Res)
save(OutputMod_GridGR4J,file="../ProjectData/GR4JgridHPCModelResults.Rdata")

OutputMod_GridGR4J <- OutputMod_GridGR4J[,c(1:4,ncol(OutputMod_GridGR4J))]
OutputMod_GridGR4J$model <- "GR4JGriddedData"

p <- ggplot(OutputMod_GridGR4J,aes(station,r.squared)) + geom_boxplot()
p <- p + stat_summary(fun.y=mean, geom="point", shape=16,
                     size=5,aes(col=rel.bias))
print(p)

```

## 4. SimHyd model results with station rainfall

Extract the modelling results from the SimHyd model, rerun the model, do the Chiew (2006) amplification analysis and write away the results.

```

# find the list of files with SimHyd results
filelist <- dir("../Projectdata/HPCResults", pattern = "SimhydCalibOutput")

```



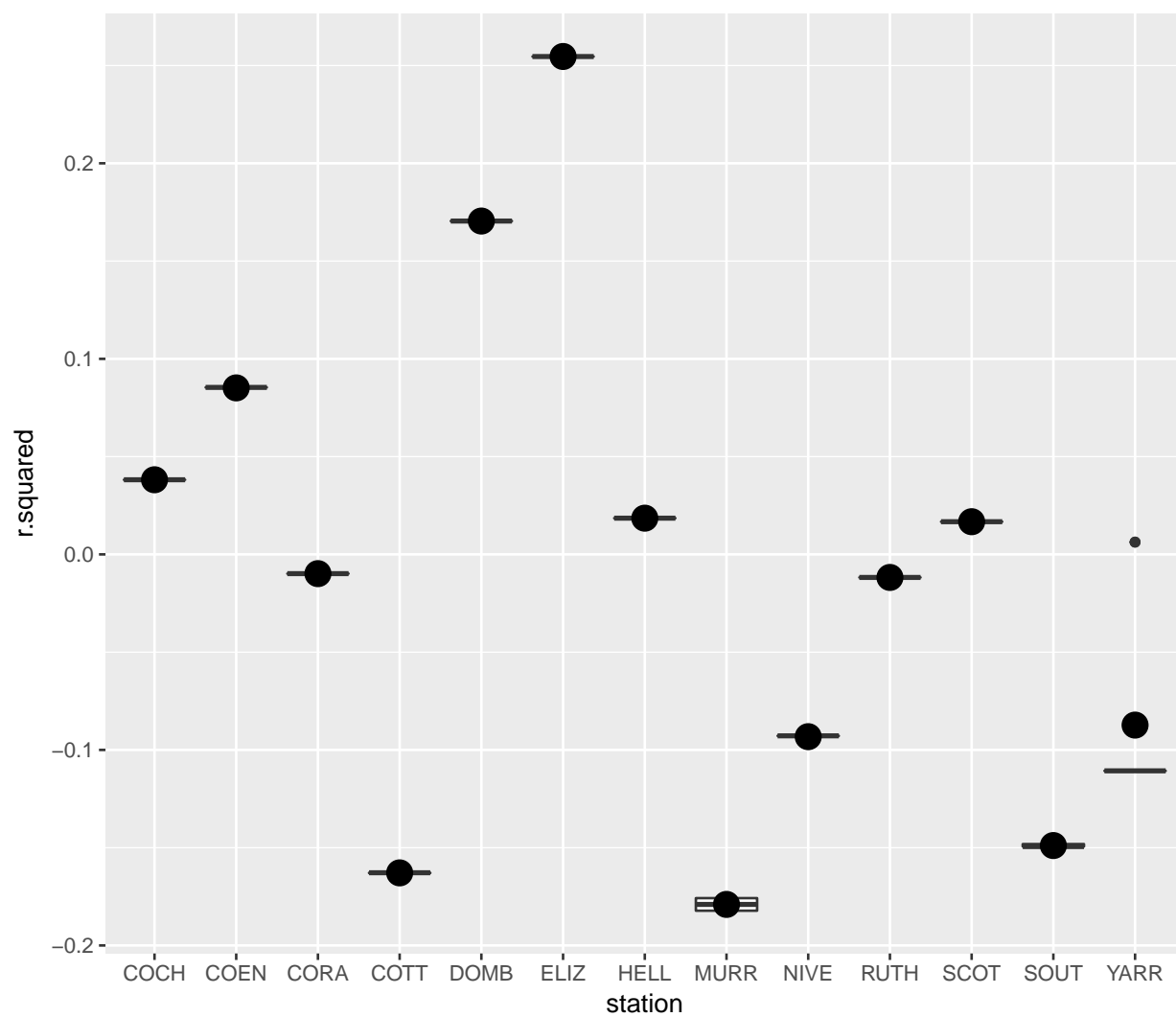


Figure 2: Calibration results for the GR4J model across 10 iterations with gridded rainfall

```

for (i in seq_along(Stations[,1])) {
  # load the rainfall, ET and flow data
  pred_data <- window(merge(flow_zoo[,i], rain_zoo[,i], maxT_zoo[,i]),
    start=start.date, end=end.date)
  colnames(pred_data) <- c("Q","P","E")
  # # another storage data frame for the residuals
  resid_out <- data.frame(matrix(0,ncol=10,nrow=nrow(pred_data)))
  # load the relevant output
  load(paste("../Projectdata/HPCResults/",filelist[i],sep=""))
  # extract the model and update with the parameters
  Mod <- Output$mod

  mod_Res[[i]] <- Output$Store
  Chiew[,1] <- Stations[i,1]
  Results[,1] <- Stations[i,1]

  # run through all iterations
  for (j in 1:(nrow(Output$Store))) {
    # testing
    #j <- 1
    # update the model with the final fitted parameters, for each iteration
    Mod <- update(Mod, INSC=Output$Store[j,7],COEFF=Output$Store[j,8],
      SQ=Output$Store[j,9],SMSC=Output$Store[j,10],
      SUB=Output$Store[j,11],CRAK=Output$Store[j,12],
      K=Output$Store[j,13],
      etmult=Output$Store[j,14], DELAY=Output$Store[j,5],
      X_m = Output$Store[j,6],
      return_state=F)

    # now predict the model output and use pred_data
    pred_mod <- predict(Mod,newdata=pred_data, all=T,na.rm=F)

    # store the residuals
    resid_out[,j] <- pred_data$Q-pred_mod

    Results[j,2:3] <- c(summary(Mod)$r.squared,summary(Mod)$rel.bias)

    # Now run the Chiew 2006 simulations on all the data
    mu <- cbind(rep(c(-15,-10,0,10),3),c(rep(0,4),rep(5,4),rep(10,4)))
    # Create the precipitation and ET data variations
    # station data
    RAIN <- rain_zoo
    test <- list()
    for (k in 1:nrow(mu)) {
      temp <- as.data.frame(cbind((1+mu[k,1]/100)*RAIN[,i],
        (1+mu[k,2]/100)*maxT_zoo[,i]))
      test[[k]] <- do.call(cbind,apply(temp,2,function(x) aggregate(x,
        list(year=format(time(flow_zoo),"%Y")),sum,na.rm=T)))
      test[[k]] <- test[[k]][,-3]
    }
    clim_adj <- do.call(rbind,test)
  }
}

```

```

# now run the different pred results
for (k in 1:ncol(pred_results)) {
  # run the model over all data
  pred_data2 <- window(merge(flow_zoo[,i],
                             (1+mu[k,1]/100)*RAIN[,i],
                             (1+mu[k,2]/100)*maxT_zoo[,i]))
  colnames(pred_data2) <- c("Q", "P", "E")

  pred_results[,k] <- predict(Mod,newdata=pred_data2, all=T,na.rm=F)
}
# summarise the data annually
pred_ann <- apply(pred_results,2,
  function(x) aggregate(x,list(year=format(time(flow_zoo),"%Y")),
                        sum,na.rm=T))

ann_flow <- rep(pred_ann[[1]][,2],6)
pred_t <- do.call(rbind,pred_ann)
# Now add the ET and precipitation data
pred_ann <- data.frame(pred_t,rain=clim_adj[,2],maxT=clim_adj[,3])
# summarise base rain and temp
ann_rain <- rep(aggregate(RAIN[,i],list(year=format(time(flow_zoo),"%Y")),
                        sum,na.rm=T),6)
ann_maxT <- rep(aggregate(maxT_zoo[,i],
                        list(year=format(time(flow_zoo),"%Y")),
                        sum,na.rm=T),6)

# Now calculate the difference
pred_diff <- pred_ann
pred_diff[,2] <- pred_diff[,2] - ann_flow
pred_diff[,3] <- pred_diff[,3] - ann_rain
pred_diff[,4] <- pred_diff[,4] - ann_maxT
# Now fit a linear model (least squares (Chiew, 2006))
fit <- lm(x~rain + maxT,data=pred_diff)
# store the results
Chiew[j,2:3] <- coef(fit)[2:3]

}
Chiew_Res[[i]] <- Chiew
sum_Res[[i]] <- Results
Residuals[[i]] <- resid_out
}

```

We can now temporary write away the results and the residuals and make some initial plots. Further plots will be generated later when comparing to the non-parametric epsilon ( $\epsilon$ ).

```

# write away the results
OutputTrends <- do.call(rbind,sum_Res)
write.csv(OutputTrends,
  file="../ProjectData/SimHydHPC_modelperformance.csv",
  row.names=F)

OutputChiew <- do.call(rbind,Chiew_Res)
write.csv(OutputChiew,file="../ProjectData/SimHydHPC_ChiewAnalysis.csv",
  row.names=F)

```

```

save(Residuals,file="../ProjectData/residuals/SimHydHPCresiduals.Rdata")

OutputMod_SimHyd <- do.call(rbind,mod_Res)
save(OutputMod_SimHyd,file="../ProjectData/SimHydHPCModelResults.Rdata")

OutputMod_SimHyd <- OutputMod_SimHyd[,c(1:4,ncol(OutputMod_SimHyd))]
OutputMod_SimHyd$model <- "SimHydStationData"

p <- ggplot(OutputMod_SimHyd,aes(station,r.squared)) + geom_boxplot()
p <- p + stat_summary(fun.y=mean, geom="point", shape=16,
                      size=5,aes(col=rel.bias))
print(p)

```

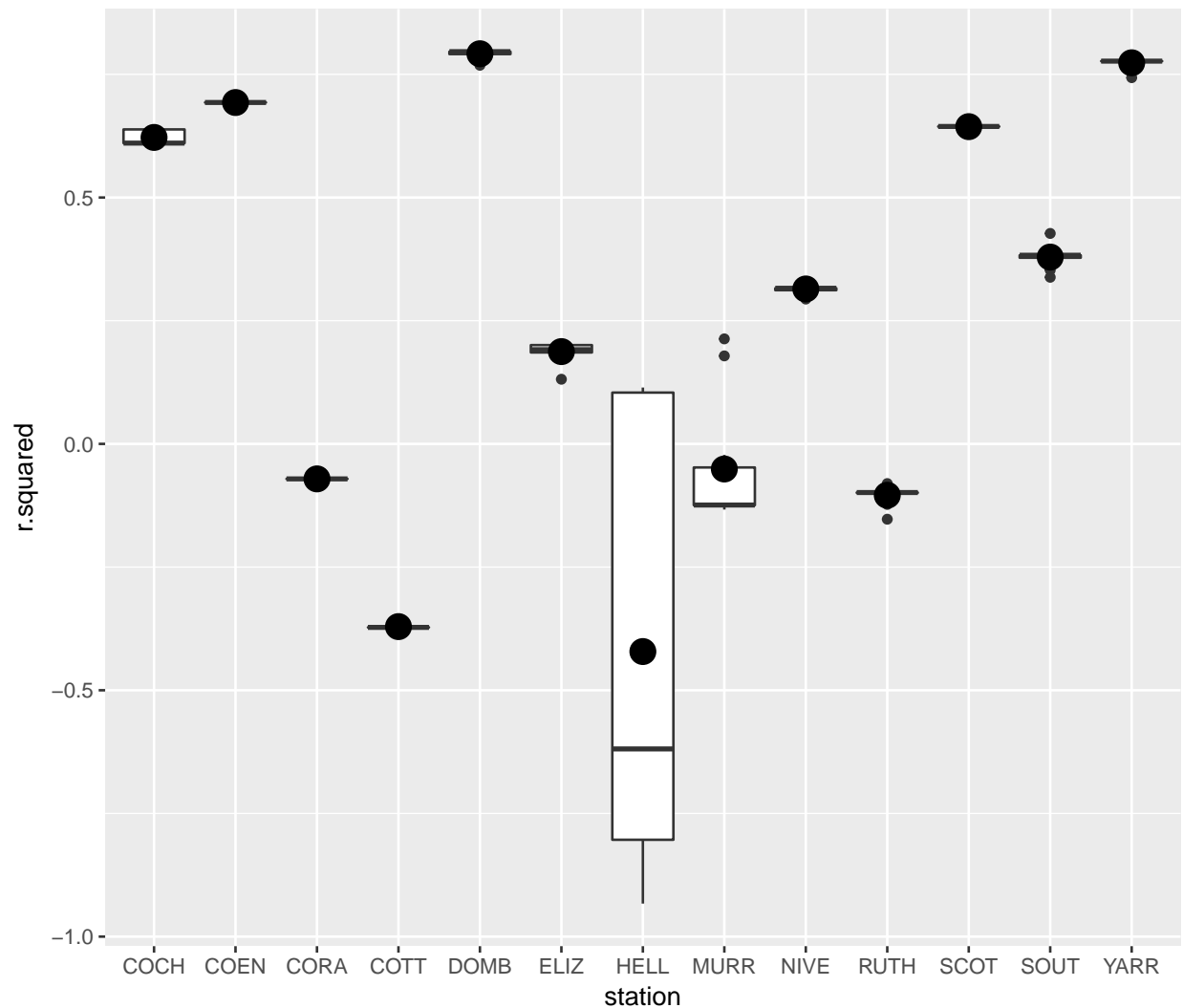


Figure 3: Calibration results for the SimHyd model across 10 iterations using station rainfall data.

## 5. SimHyd model results with gridded rainfall

Extract the modelling results

```
# find the list of files with GridSimHyd results
filelist <- dir("../Projectdata/HPCResults", pattern = "gridSimhydCalibOutput")

for (i in seq_along(filelist)) {
  # load the rainfall, ET and flow data
  pred_data <- window(merge(flow_zoo[,i], Gridrain_zoo[,i], maxT_zoo[,i]),
                      start=start.date, end=end.date)
  colnames(pred_data) <- c("Q","P","E")
  # another storage data frame for the residuals
  resid_out <- data.frame(matrix(0,ncol=10,nrow=nrow(pred_data)))
  # load the relevant output
  load(paste("../Projectdata/HPCResults/",filelist[i],sep=""))
  # extract the model and update with the parameters
  Mod <- Output$mod

  mod_Res[[i]] <- Output$Store
  Chiew[,1] <- Stations[i,1]
  Results[,1] <- Stations[i,1]

  # run through all iterations
  for (j in 1:(nrow(Output$Store))) {
    # testing
    #j <- 1
    # update the model with the final fitted parameters, for each iteration
    Mod <- update(Mod, INSC=Output$Store[j,7],COEFF=Output$Store[j,8],
                  SQ=Output$Store[j,9],SMSC=Output$Store[j,10],
                  SUB=Output$Store[j,11],CRAK=Output$Store[j,12],
                  K=Output$Store[j,13],
                  etmult=Output$Store[j,14], DELAY=Output$Store[j,5],
                  X_m = Output$Store[j,6],
                  return_state=F)

    # now predict the model output and use pred_data
    pred_mod <- predict(Mod,newdata=pred_data, all=T,na.rm=F)

    # store the residuals
    resid_out[,j] <- pred_data$Q-pred_mod

    Results[j,2:3] <- c(summary(Mod)$r.squared,summary(Mod)$rel.bias)

    # Now run the Chiew 2006 simulations on all the data
    mu <- cbind(rep(c(-15,-10,0,10),3),c(rep(0,4),rep(5,4),rep(10,4)))
    # Create the precipitation and ET data variations
    # station data
    RAIN <- Gridrain_zoo
    test <- list()
    for (k in 1:nrow(mu)) {
      temp <- as.data.frame(cbind((1+mu[k,1]/100)*RAIN[,i],
```

```

        (1+mu[k,2]/100)*maxT_zoo[,i]))
test[[k]] <- do.call(cbind,apply(temp,2,function(x) aggregate(x,
        list(year=format(time(flow_zoo),"%Y")),sum,na.rm=T)))
test[[k]] <- test[[k]][,-3]
}
clim_adj <- do.call(rbind,test)

# now run the different pred results
for (k in 1:ncol(pred_results)) {
  # run the model over all data
  pred_data2 <- window(merge(flow_zoo[,i],
        (1+mu[k,1]/100)*RAIN[,i],
        (1+mu[k,2]/100)*maxT_zoo[,i]))
  colnames(pred_data2) <- c("Q","P","E")

  pred_results[,k] <- predict(Mod,newdata=pred_data2, all=T,na.rm=F)
}
# summarise the data annually
pred_ann <- apply(pred_results,2,
  function(x) aggregate(x,list(year=format(time(flow_zoo),"%Y")),
    sum,na.rm=T))

ann_flow <- rep(pred_ann[[1]][,2],6)
pred_t <- do.call(rbind,pred_ann)
# Now add the ET and precipitation data
pred_ann <- data.frame(pred_t,rain=clim_adj[,2],maxT=clim_adj[,3])
# summarise base rain and temp
ann_rain <- rep(aggregate(RAIN[,i],list(year=format(time(flow_zoo),"%Y")),
  sum,na.rm=T),6)
ann_maxT <- rep(aggregate(maxT_zoo[,i],
  list(year=format(time(flow_zoo),"%Y")),
  sum,na.rm=T),6)

# Now calculate the difference
pred_diff <- pred_ann
pred_diff[,2] <- pred_diff[,2] - ann_flow
pred_diff[,3] <- pred_diff[,3] - ann_rain
pred_diff[,4] <- pred_diff[,4] - ann_maxT
# Now fit a linear model (least squares (Chiew, 2006))
fit <- lm(x~rain + maxT,data=pred_diff)
# store the results
Chiew[j,2:3] <- coef(fit)[2:3]

}
Chiew_Res[[i]] <- Chiew
sum_Res[[i]] <- Results
Residuals[[i]] <- resid_out
}

```

We can now temporary write away the results and the residuals and make some initial plots. Further plots will be generated later when comparing to the non-parametric epsilon ( $\epsilon$ ).

```

# write away the results
OutputTrends <- do.call(rbind,sum_Res)

```

```

write.csv(OutputTrends,
          file="../ProjectData/SimHydGridHPC_modelperformance.csv",
          row.names=F)

OutputChiew <- do.call(rbind,Chiew_Res)
write.csv(OutputChiew,file="../ProjectData/SimHydGridHPC_ChiewAnalysis.csv",
          row.names=F)

save(Residuals,file="../ProjectData/residuals/SimHydGridHPCresiduals.Rdata")

OutputMod_GridSimHyd <- do.call(rbind,mod_Res)
save(OutputMod_GridSimHyd,file="../ProjectData/SimHydHPCModelResults.Rdata")

OutputMod_GridSimHyd <- OutputMod_GridSimHyd[,c(1:4,ncol(OutputMod_GridSimHyd))]
OutputMod_GridSimHyd$model <- "SimHydGridStationData"

p <- ggplot(OutputMod_GridSimHyd,aes(station,r.squared)) + geom_boxplot()
p <- p + stat_summary(fun.y=mean, geom="point", shape=16,
                     size=5,aes(col=rel.bias))
print(p)

```

## 6. Final plot comparing performance of all models

```

OutputMod <- rbind(OutputMod_GR4J,OutputMod_GridGR4J, OutputMod_SimHyd,
                  OutputMod_GridSimHyd)
p <- ggplot(OutputMod,aes(station,r.squared)) + geom_boxplot()
p <- p + stat_summary(fun.y=mean, geom="point", shape=16,
                     size=5,aes(colour=rel.bias))
p <- p + facet_wrap(~model)
print(p)

# Write away the data
save(OutputMod,file="../ProjectData/ModelResults.Rdata")

```

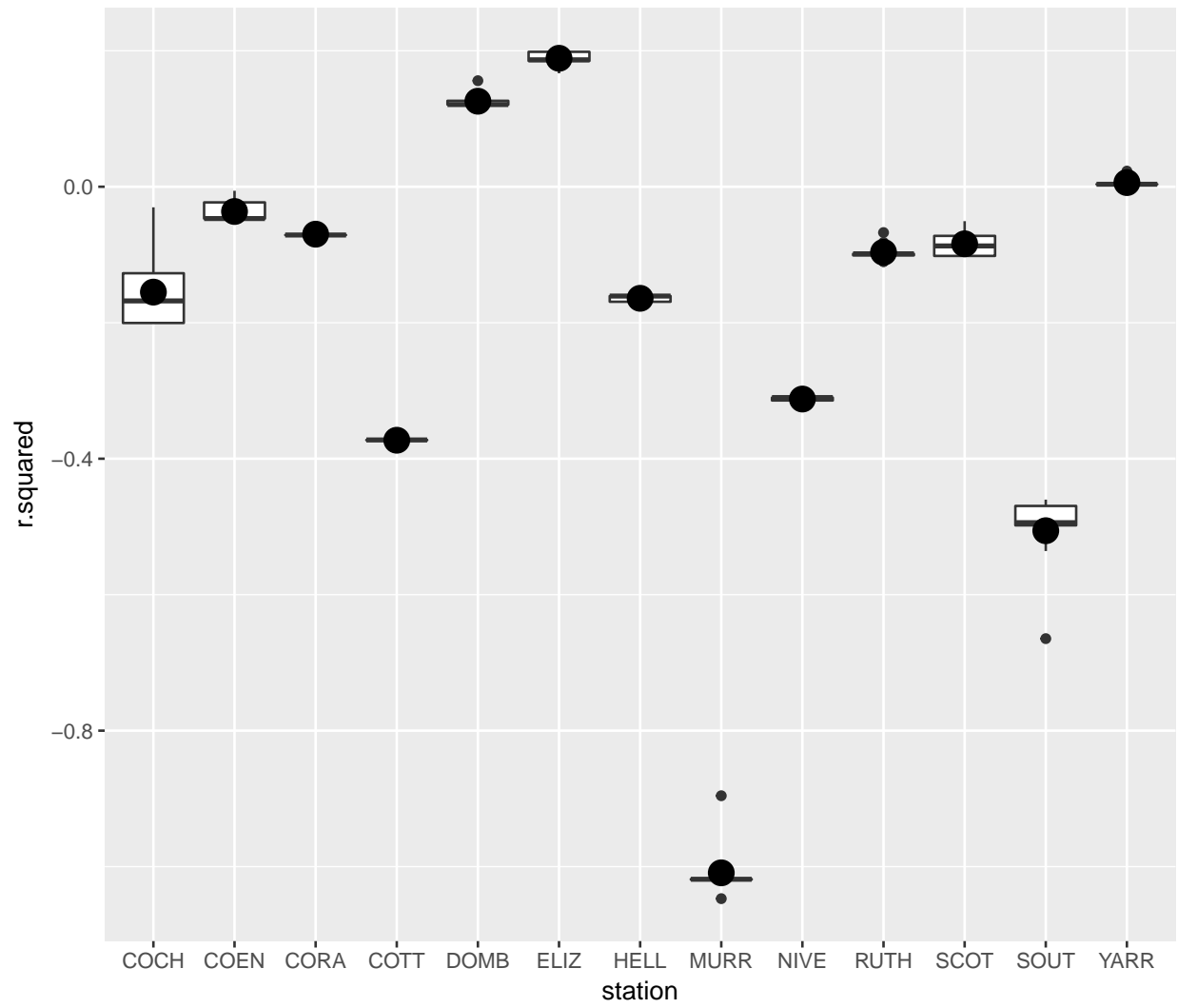


Figure 4: Calibration results for the SimHyd model across 10 iterations with gridded rainfall



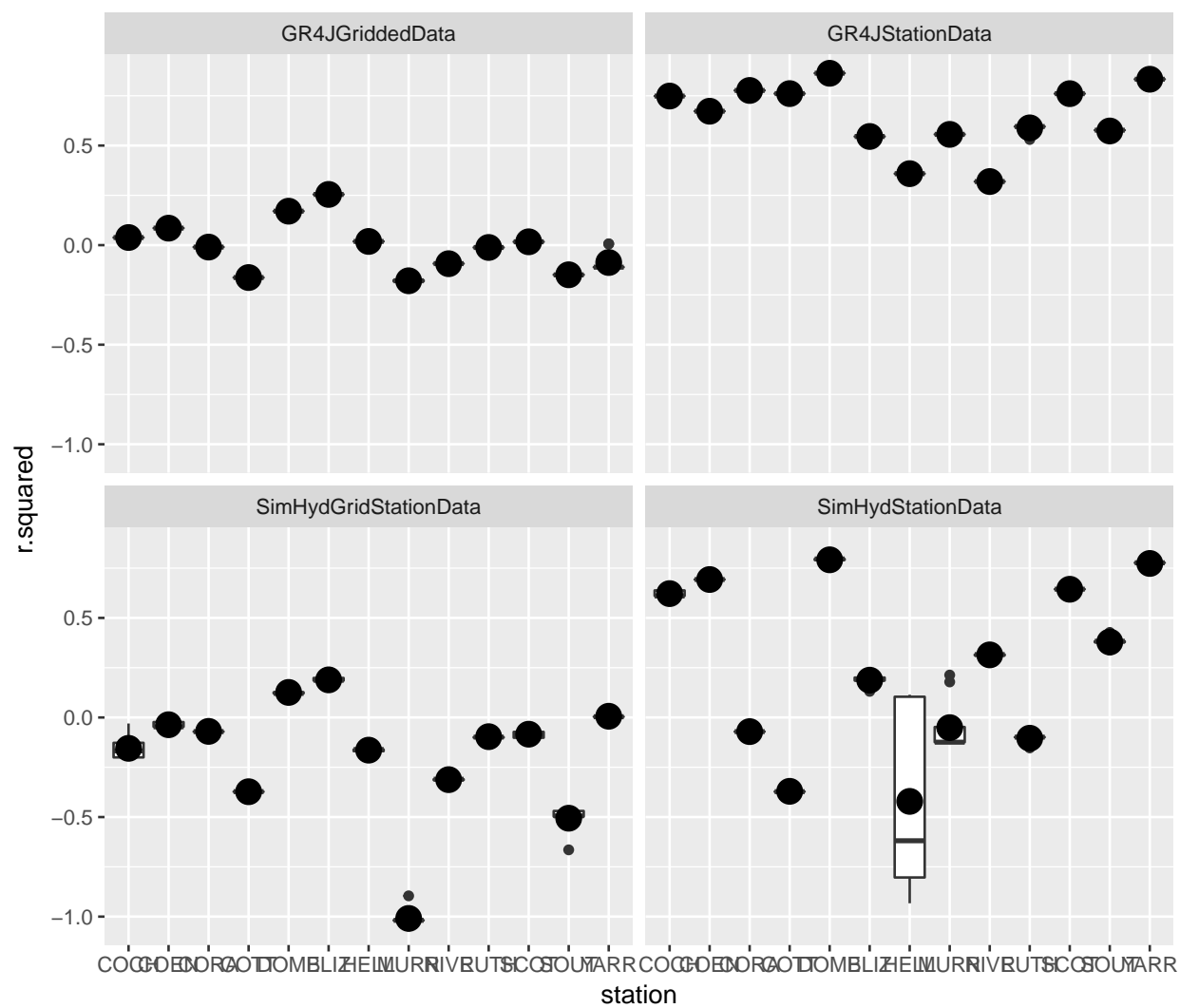


Figure 5: Comparing the performance of different models