Analysis of HPC results

Willem Vervoort, Michaela Dolk & Floris van Ogtrop

2017-08-28

```
# 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(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.

```r
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

```r
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

```r
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),pvalue_eta_p=numeric(length=10),
                    pvalue_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)),
                           P0ET0=numeric(length=nrow(flow_zoo)),
                           Pplus10ET0=numeric(length=nrow(flow_zoo)),
                           Pmin15ETplus5=numeric(length=nrow(flow_zoo)),
                           Pmin10ETplus5=numeric(length=nrow(flow_zoo)),
                           P0ETplus5=numeric(length=nrow(flow_zoo)),
                           Pplus10ETplus5=numeric(length=nrow(flow_zoo)),
                           Pmin15ETplus10=numeric(length=nrow(flow_zoo)),
                           Pmin10ETplus10=numeric(length=nrow(flow_zoo)),
                           P0ETplus10=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.

```r
# 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[grep(Stations[i,1],filelist)],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,2:5] <- c(coef(fit)[2:3],summary(fit)$coefficients[2:3,4])


  }
  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 (ϵ).

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

```r
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)
pander(aggregate(OutputChiew[,2:5],list(Station=OutputChiew$station),
                 mean),
       caption="Results GR4J epsilon fit with significance")
```

Table 1: Results GR4J epsilon fit with significance

Station	eta_p	eta_e	pvalue_eta_p	pvalue_eta_e
COCH	0.7442	-0.01141	2.49e-192	0.2201
COEN	0.8159	-0.01613	1.375e-212	3.583e-05
CORA	0.7546	-0.01969	1.318e-149	4.499e-05
COTT	0.4678	-0.01504	1.631e-97	0.01108
DOMB	0.5981	-0.03821	6.462e-254	9.181e-25
ELIZ	0.7626	-0.05314	6.745e-136	4.964e-11
HELL	1.581	-0.08282	1.259e-279	2.459e-21
MURR	1.016	-0.06884	4.758e-236	9.471e-43
NIVE	1.052	-0.04456	4.625e-246	1.687e-06
RUTH	0.678	-0.02417	2.424e-114	7.499e-05
SCOT	0.4478	-0.01888	1.234e-182	1.708e-14
SOUT	1.412	-0.1233	1.985e-260	2.591e-44
YARR	0.1021	-0.005532	1.468e-147	9.643e-11

```r
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

```r
# 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
```

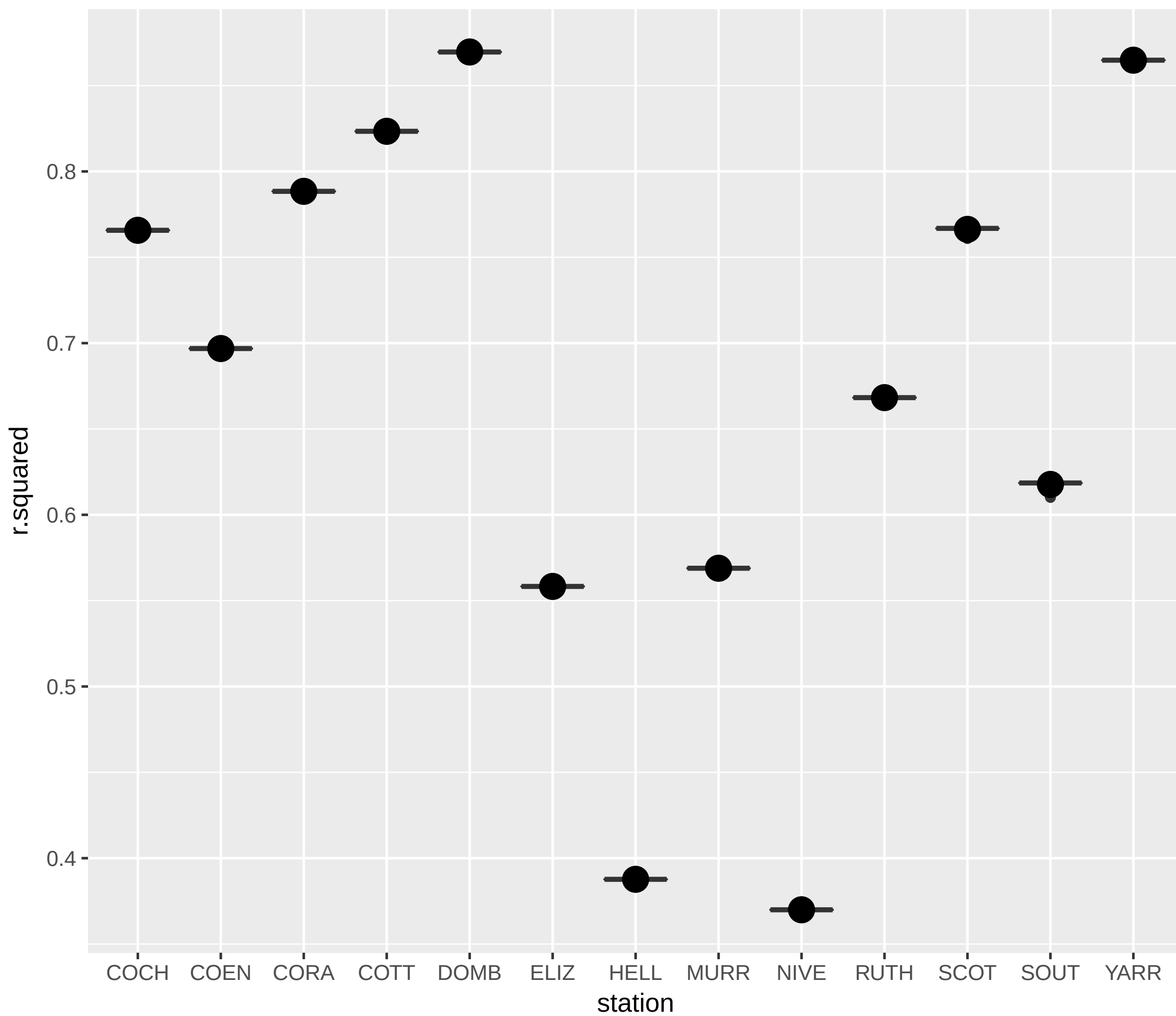

Figure 1: Calibration results for the GR4J model across 10 iterations using station rainfall data.

```r
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[grep(Stations[i,1],filelist)],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 <- 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,2:5] <- c(coef(fit)[2:3],summary(fit)$coefficients[2:3,4])


  }
  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 (ϵ).

```
# 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)
pander(aggregate(OutputChiew[,2:5],list(Station=OutputChiew$station),
                 mean),
       caption="Results GR4JGrid epsilon fit with significance")
```

Table 2: Results GR4JGrid epsilon fit with significance

Station	eta_p	eta_e	pvalue_eta_p	pvalue_eta_e
COCH	0.9275	-0.007995	4.463e-222	0.3807
COEN	0.8056	-0.006384	8.296e-196	0.1288
CORA	0.3075	-0.04092	3.263e-09	0.003896
COTT	0.1181	-0.01446	9.734e-35	5.417e-05
DOMB	0.3664	-0.003243	2.981e-266	0.1115
ELIZ	0.7229	-0.02572	2.662e-265	4.164e-08
HELL	0.7255	-0.005686	7.445e-264	0.5841
MURR	0.4732	-0.005268	6.32e-97	0.09043
NIVE	0.5074	-0.1125	5.85e-135	5.386e-26
RUTH	0.1025	-0.009539	3.182e-12	0.01797
SCOT	0.2772	-0.006771	7.788e-161	0.03633
SOUT	0.2859	-0.01555	3.902e-58	0.02851
YARR	0.05668	-0.003763	1.234e-89	0.01758

```r
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.

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

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[grep(paste(Stations[i,1],"SimhydCalibOutput",sep=""),
                          filelist)],sep=""))
  # extract the model and update with the parameters
  Mod <- Output$mod
```

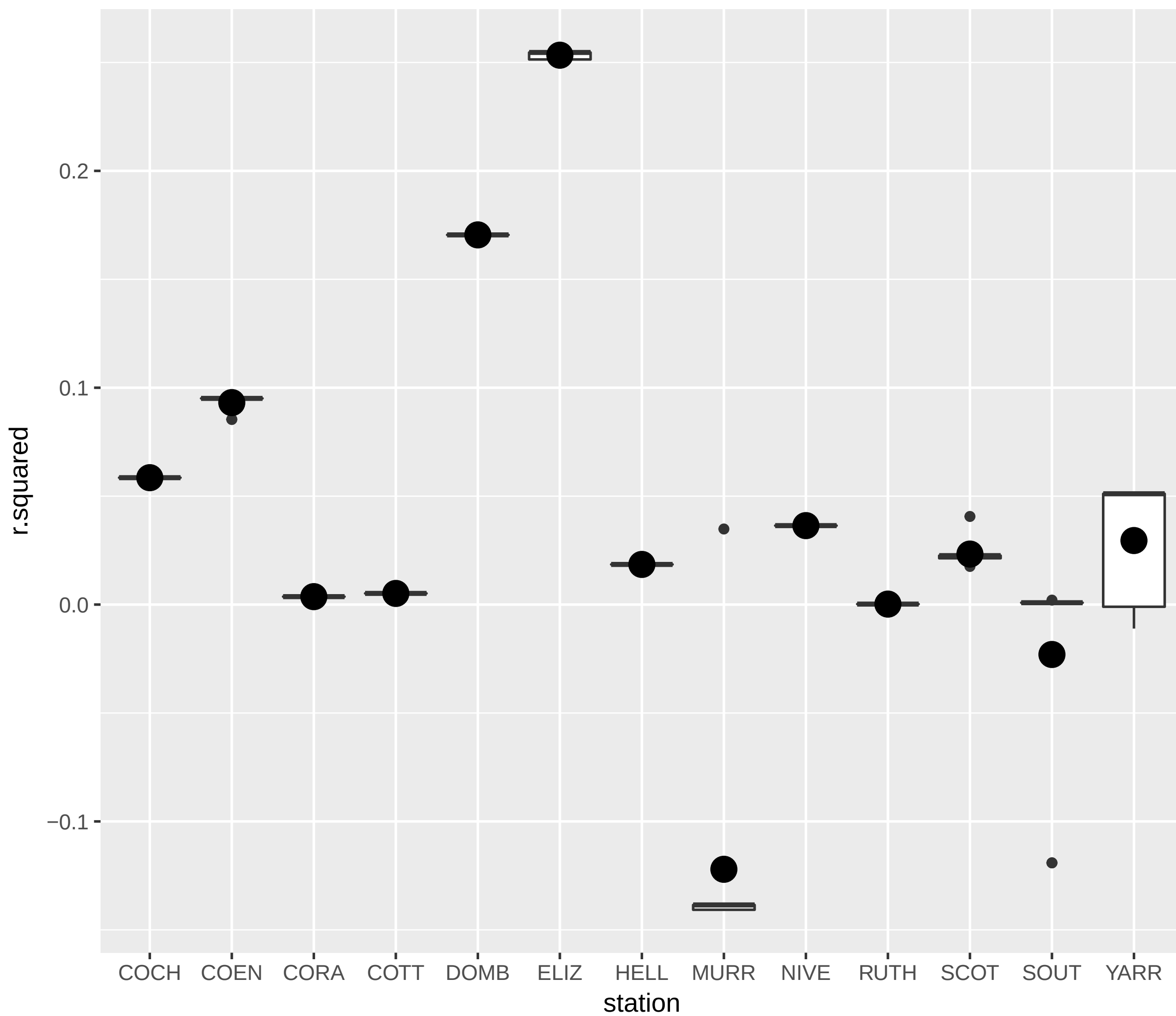

Figure 2: Calibration results for the GR4J model across 10 iterations with gridded rainfall

```
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,2:5] <- c(coef(fit)[2:3],summary(fit)$coefficients[2:3,4])


  }
  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 (ϵ).

```
# 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)
pander(aggregate(OutputChiew[,2:5],list(Station=OutputChiew$station),
                 mean),
       caption="Results SimHyd epsilon fit with significance")
```

Table 3: Results SimHyd epsilon fit with significance

Station	eta_p	eta_e	pvalue_eta_p	pvalue_eta_e
COCH	0.8255	-0.006084	2.018e-197	0.6458
COEN	0.8772	0.5293	3.147e-07	0.5172
CORA	-0.3519	555.6	0.08604	0.5569
COTT	0.7642	-0.005388	2.242e-98	0.5698

Station	eta_p	eta_e	pvalue_eta_p	pvalue_eta_e
DOMB	0.89	-0.01538	1.15e-297	0.3015
ELIZ	0.894	-0.001224	7.254e-166	0.883
HELL	0.8888	-0.002064	5.144e-258	0.6936
MURR	0.9206	-0.003866	1.267e-224	0.579
NIVE	0.7699	-0.007339	1.309e-07	0.617
RUTH	0.7995	-0.007393	2.109e-128	0.3145
SCOT	0.7585	-0.002837	7.074e-248	0.6671
SOUT	0.9314	-0.002338	3.821e-237	0.673
YARR	0.8992	-0.003147	2.647e-285	0.4761

```
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)
```

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[grep(Stations[i,1],filelist)],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))) {
```

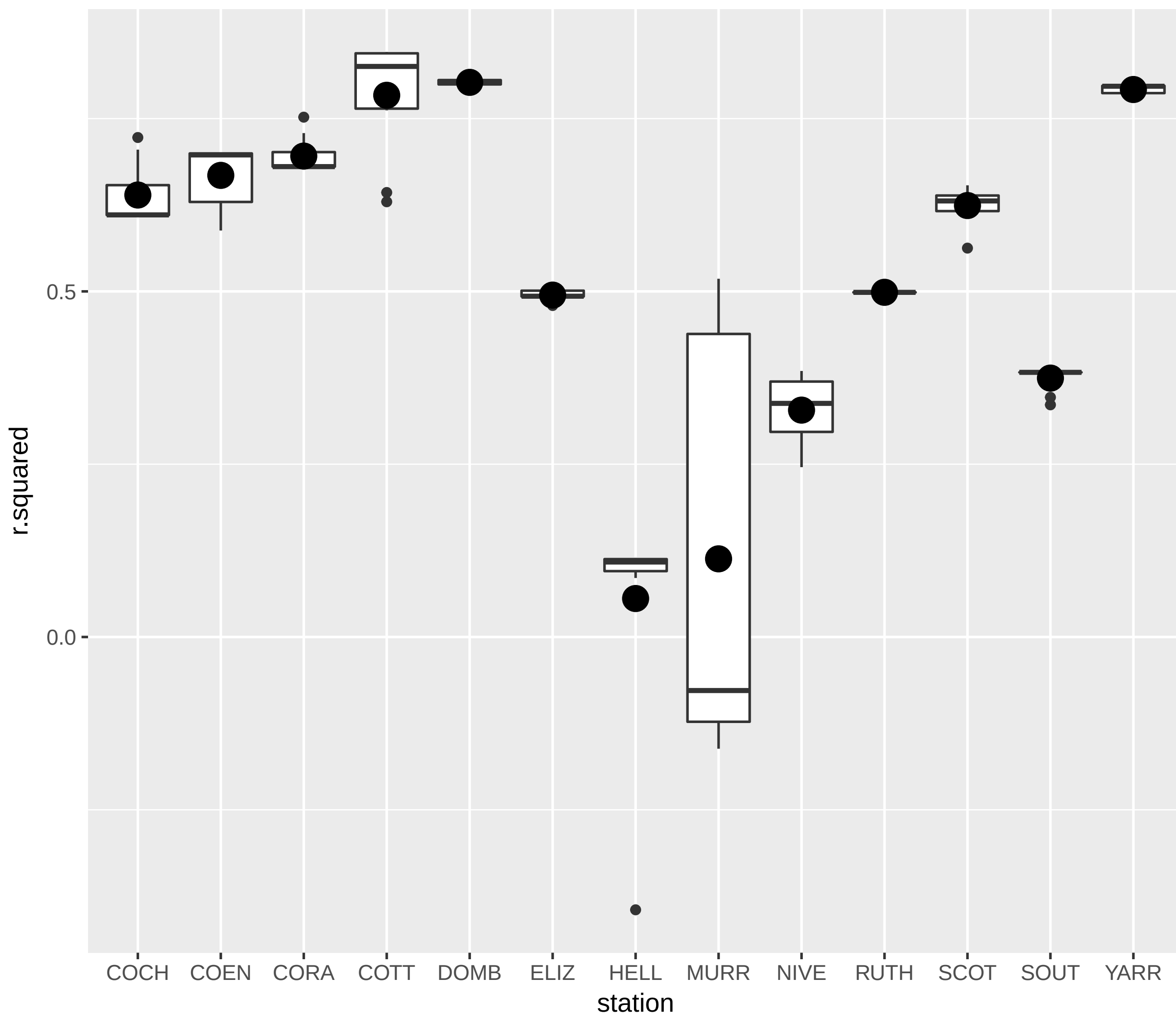

Figure 3: Calibration results for the SimHyd model across 10 iterations using station rainfall data.

```r
# 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,2:5] <- c(coef(fit)[2:3],summary(fit)$coefficients[2:3,4])


  }
  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 (ϵ).

```
# 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)
pander(aggregate(OutputChiew[,2:5],list(Station=OutputChiew$station),
                 mean),
       caption="Results SimHydGrid epsilon fit with significance")
```

Table 4: Results SimHydGrid epsilon fit with significance

Station	eta_p	eta_e	pvalue_eta_p	pvalue_eta_e
COCH	0.9332	0.0005496	2.872e-220	0.9526
COEN	0.8849	0.0008553	8.505e-211	0.839
CORA	0.7566	-0.001316	1.176e-140	0.6303
COTT	0.8247	-0.007401	4.957e-191	0.662
DOMB	0.8099	-0.004452	3.428e-267	0.8289
ELIZ	0.9015	-5.458e-05	3.363e-261	0.9894
HELL	0.7612	-0.004885	1.999e-16	0.7081
MURR	0.8154	-0.001546	3.8e-264	0.8239
NIVE	0.7553	-0.01118	1.133e-139	0.6108
RUTH	0.898	-0.001279	7.845e-156	0.8402

Station	eta_p	eta_e	pvalue_eta_p	pvalue_eta_e
SCOT	0.7685	7.173e-05	1.726e-211	0.9638
SOUT	0.9449	0.0002668	9.124e-279	0.9632
YARR	0.8428	-0.002979	1.843e-246	0.798

```
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 in the manuscript

The code below simply makes Figure 4 in the manuscript, which is only based on the station data and not on the gridded data.

```
OutputMod <- rbind(OutputMod_GR4J, OutputMod_SimHyd)
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, ncol=1)
print(p)
```

```
# save the plot
tiff("../Manuscript/Figure4_ModelCalStats.tif",res=600,
     compression="lzw",
     width=10*480,height=10*480)
print(p)
dev.off()
```

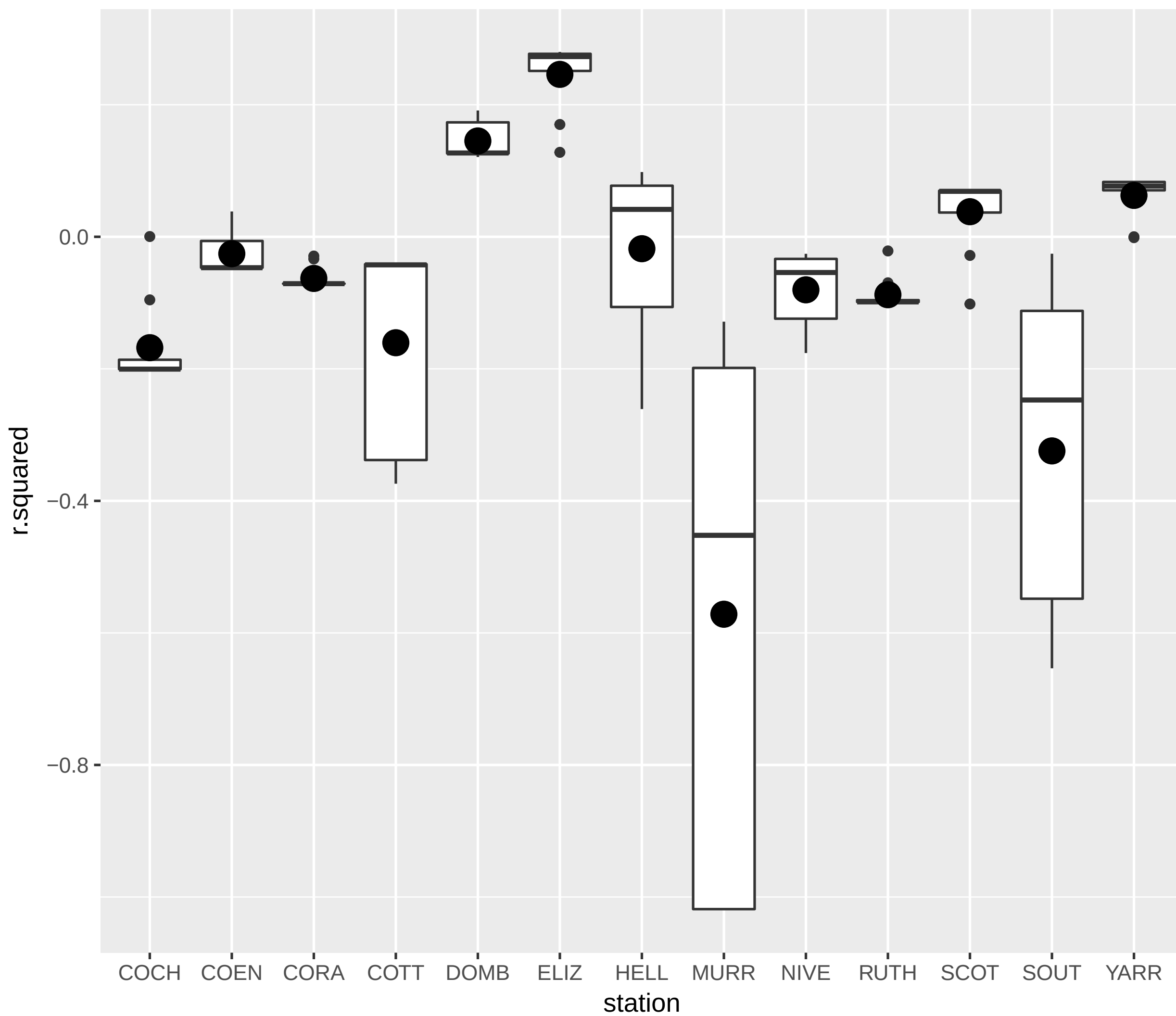

Figure 4: Calibration results for the SimHyd model across 10 iterations with gridded rainfall

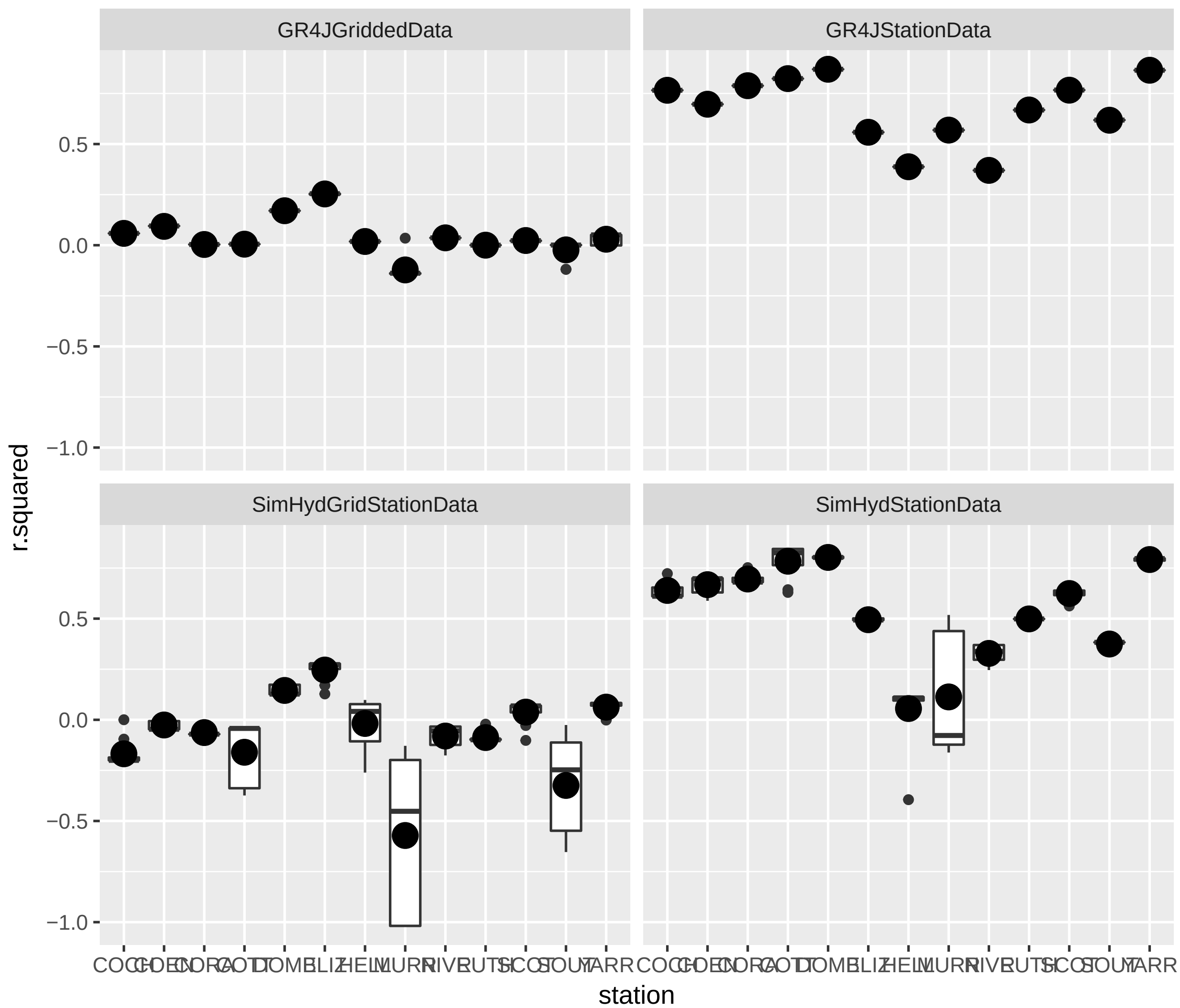

Figure 5: Comparing the performance of different models

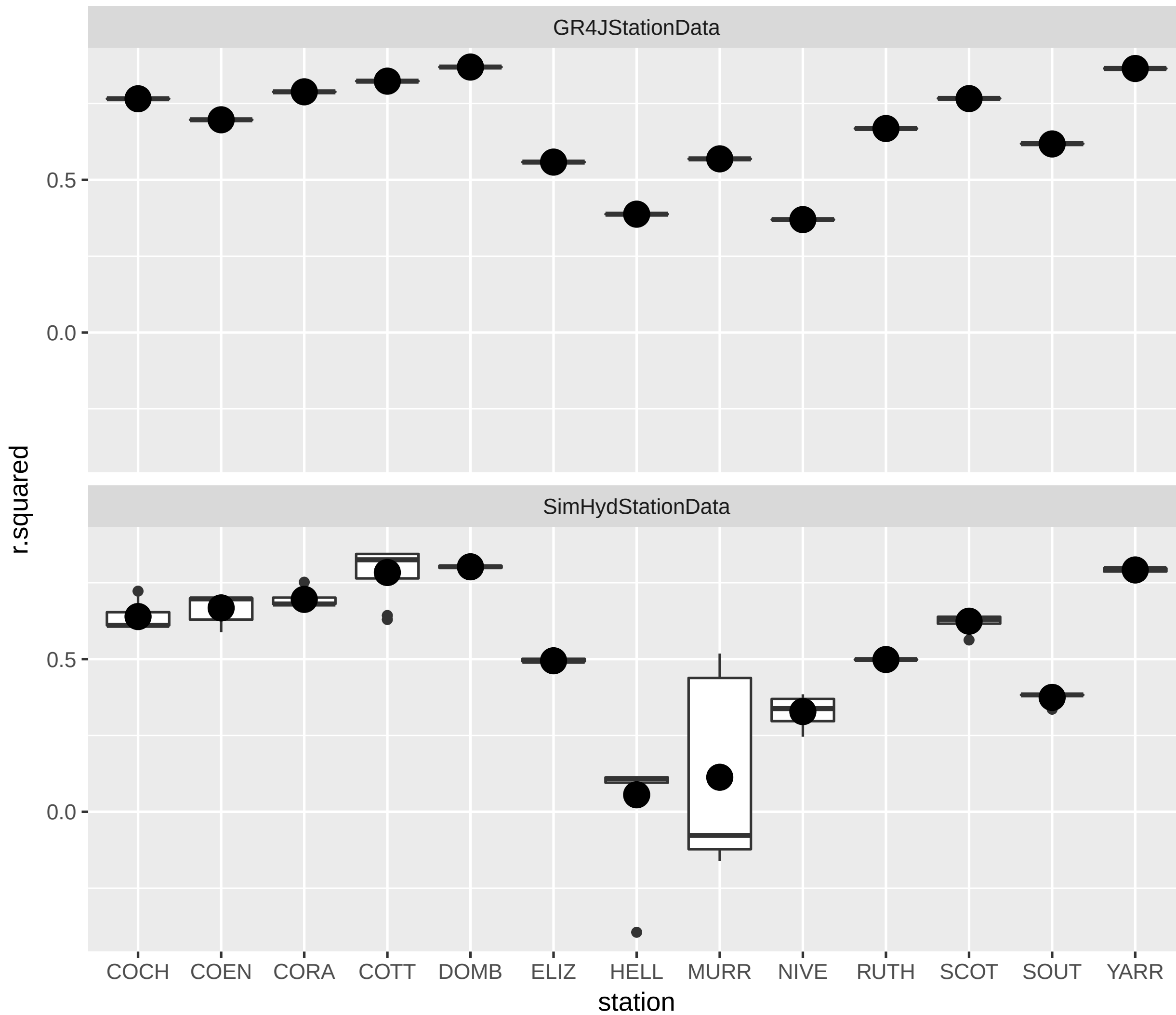

Figure 6: Manuscript Figure 4, Comparing the performance of different models

```
## pdf
##   2
```