Supplemental code file for manuscript: An open-source wavelet tool for improving prediction accuracy for natural system models

Ze Jiang, Md. Mamunur Rashid, Fiona Johnson, and Ashish Sharma

produced on *20:43:05 07 August, 2020*

Introduction

This reproducible and dynamic report was created using Bookdown (based on Rmarkdown and Knit) package, and summarizes the basic code and outputs (plots, tables, etc) produced during the course. The relative file paths indicated in the code below assume that your project working directly is structured as indicated here. If Knit PDF is problematic for you, switch the output to html.

The code below shows how the figures in the manuscript is reproduced.

Reproducibility and accessibility

In order to reproduce all steps listed below, the first thing is to install WASP package from https://github.com/zejiang-unsw/WASP (following the instruction and install all the dependencies). The data, demo and all code used in this analysis, including the Rmarkdown document used to compile this supplementary code file, are all available on GitHub. Once this GitHub repo has been downloaded, navigate to /WASP/vignettes to find the Rmarkdown document, and set this as your working directory for executing code.

Required R packages

A variety of R packages was used for this analysis. All graphics and data wrangling were handled using the tidyverse suite of packages. All packages used are available from the Comprehensive R Archive Network (CRAN) or Github.

```r
require(devtools)
# devtools::install_github("zejiang-unsw/WASP@devel", dependencies = TRUE)
# devtools::install_github("zejiang-unsw/NPRED")

library(WASP)
library(NPRED)
library(cowplot)
library(ggplot2)
library(raster)
library(overlapping) #PDF skill scores

library(plyr)
library(dplyr)
library(tidyr)
#library(tidyverse)
```

Run code

```
# if you want to knit with new results, change eval=TRUE
source("./code/EMS2020_Figure3-5.R")
source("./code/EMS2020_Figure4.R")
source("./code/EMS2020_Figure4S.R")
```

Figure 1: Flowchart of the proposed method

```
knitr::include_graphics('../inst/fig/Figure1.jpg')
```

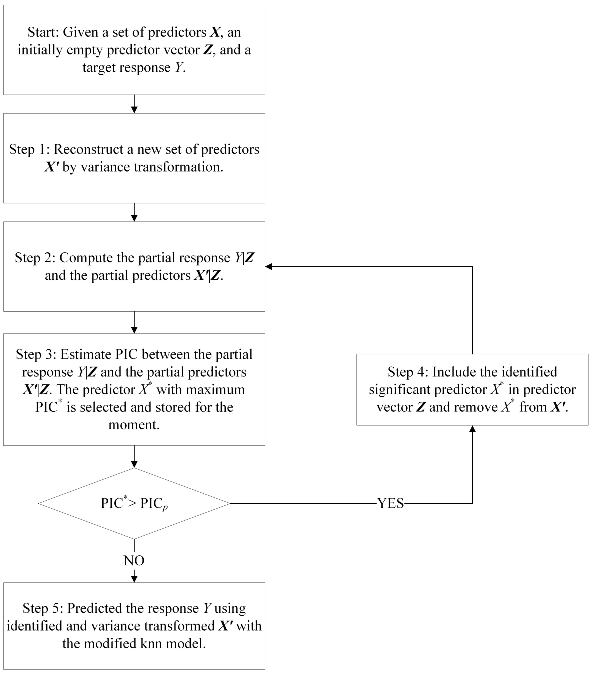

Figure 1: Flowchart of the proposed method

Figure 2: A demo of WASP package

Figure 2 is a screenshot of the sequence of R commands illustrating the usage of the WASP package to transform the potential predictors (see Fig. S1 in the Supporting Material for an example of predictor variables before and after variance transformation corresponding to the response), identify the significant predictors and predict the associated response. MODWT is adopted as the basis of wavelet transform in this case study since we are using observed data to predict target response and thus there is no dependence on future information.

Not that this will be empty figure in output directory since the figure is the code chunk itself.

```
#-------------------------------------------------------------------------------
#load response and predictor variables
data(SPI.12); data(data.CI); data(Ind_AWAP.2.5)
#study grids and period
Grid = sample(Ind_AWAP.2.5,1)
Grid = 149 # A sample grid
SPI.12.ts <- window(SPI.12, start=c(1910,1),end=c(2009,12))
data.CI.ts <- window(data.CI, start=c(1910,1),end=c(2009,12))
#partition into two folds
```

```r
folds <- cut(seq(1,nrow(SPI.12.ts)),breaks=2,labels=FALSE)
sub.cali <- which(folds==1, arr.ind=TRUE); sub.vali <- which(folds==2, arr.ind=TRUE)
#----------------------------------------------------------------------------------
###calibration and selection
data <- list(x=SPI.12.ts[sub.cali,Grid],dp=data.CI.ts[sub.cali,])

#variance transformation - calibration
dwt <- modwt.vt(data, wf=wf, J=8, boundary="periodic")

#stepwise PIC selection
sel <- NPRED::stepwise.PIC(dwt$x, dwt$dp.n)
#----------------------------------------------------------------------------------
###validation and prediction
data.val <- list(x=SPI.12.ts[sub.vali,Grid],dp=data.CI.ts[sub.vali,])

#variance transformation - validation
dwt.val <- modwt.vt.val(data.val, J=8, dwt)

#knn prediction
cpy <- sel$cpy; wt <- sel$wt
x=data$x; z=dwt$dp.n[,cpy]; zout=dwt.val$dp.n[,cpy]
mod <- knn(x, z, zout, k=5, pw=wt, extrap=T)
plot.new()
```

Figure 2: A demo of WASP package

Figure 3: The most significant climate indices identified over Australia

Figure 3 shows the most significant drivers (i.e. the predictor selected first in the PIC process from the set of variance-transformed climate indices) for both SPI12 and SPI36.

```r
#-----------------------------------------------------------
data(data.CI); data(Ind_AWAP.2.5)
Grids=Ind_AWAP.2.5
Ind_CI <- colnames(data.CI)

p0.list <- list(); tab.list <- list(); tab1.list<- list()
for(sc in c(12,36)){
    ###selection map
    load(paste0("./result/data.SPI.",sc,".selection_",mode,"_",wf,"_",k.folds,"folds.Rdata"))
    load(paste0("./result/data.SPI.",sc,".weights_",mode,"_",wf,"_",k.folds,"folds.Rdata"))

    #summary(sel.cv); head(sel.cv)
    sel <- vector("list",length=length(Grids))
    for(i in 1:length(Grids)){
      tmp1 <- NULL; tmp2 <- NULL
      for(j in 1:length(sel.cv)){
        tmp1 <- c(tmp1, sel.cv[[j]]$origin[[i]])
        tmp2 <- c(tmp2, sel.cv[[j]]$vt[[i]])
      }

      if(!is.null(tmp1)) sel[[i]]$origin <- data.frame(table(tmp1))[order(data.frame(table(tmp1))$Fre
      else sel[[i]]$origin <- NA
```

```
  if(!is.null(tmp2)) sel[[i]]$vt <- data.frame(table(tmp2))[order(data.frame(table(tmp2))$Freq, d
  else sel[[i]]$vt <- NA
}

#summary(sel); head(sel)
#---------------------------------------------------------
#selection map for the entire period
for(i in 1){
  for(j in c("origin","vt")){
    data <- vector("list",nrow(lat_lon.2.5))
    data[Grids] <- lapply(sel,function(ls) as.character(ls[[j]]))
    tab1.list[[length(tab1.list)+1]] <- data[Grid.sample]

    data <- matrix(NA,nrow=nrow(lat_lon.2.5))
    data[Grids] <- unlist(lapply(sel,function(ls) as.character(ls[[j]])[i]))
    freq = factor(data[Grids]); levels(freq) <- 1:4
    tab.list[[length(tab.list)+1]] <- table(freq)

    Sel.df <- data.frame(lat_lon.2.5, Driver=data); summary(Sel.df)
    levels(Sel.df$Driver) <- 1:4
    labels <- c("1" = "Nino3.4", "2" = "PDO", "3"="SAM", "4"="DMI")

    samp = data.frame(lat_lon.2.5[Grid.sample,], label=Grid.sample)
    miss = lat_lon.2.5[Ind_AWAP.2.5[which(is.na(Sel.df[Ind_AWAP.2.5,3]))],]


    #---------------------------------------
    p <- ggplot(data=na.omit(Sel.df), aes(x=lon, y=lat)) +
         geom_tile(aes(fill=Driver))+
         geom_polygon(data=Aus_map, aes(x = long, y = lat, group=group), color="grey", fill="NA")
         geom_text(data=samp, aes(x = lon, y = lat, label=label), color="red", size=2)+
         geom_point(data=miss, aes(x = lon, y = lat),shape=16, size=2)+
         #geom_tile(data=miss.na, aes(x = lon, y = lat),fill="white")+

         facet_wrap(~Driver, ncol=length(labels),labeller=labeller(Driver = labels), drop=FALSE)+
         coord_equal()+
         #scale_fill_manual(values=c("black","green","blue","purple"), labels=Ind_CI) +
         #scale_fill_manual(values=rep("darkgrey",length(labels)), labels=Ind_CI) +

         scale_y_continuous(breaks=seq(-44.75,-9.75,2.5), limits=c(-44.75,-9.75), expand = c(0,0)
         scale_x_continuous(breaks=seq(111.75,156.75,2.5),limits=c(111.75,156.75), expand = c(0,0
         theme_bw() +
         theme(text = element_text(size = 12),

              plot.margin=unit(c(0,1.5,0,1), "cm"),
              panel.spacing = unit(0, "lines"),
              panel.grid.major = element_blank(),
              panel.grid.minor = element_blank(),
              panel.background = element_rect(color = "black"),
              panel.border = element_rect(colour = "black"),

              legend.position="none",

              axis.ticks = element_blank(),
              axis.text = element_blank(),
              axis.title = element_blank()
```

```
                    )
        p
        p0.list[[length(p0.list)+1]] <- p
      }
    }

}
#-----------------------------------------------
#combine subplots - vt model
#cowplot::plot_grid(plotlist = p0.list, nrow=4, labels = c("(a)","(b)","(c)","(d)"), label_size = 12,
cowplot::plot_grid(plotlist = p0.list[c(2,4)], nrow=2, labels = c("(a)","(b)"), label_size = 12, hjus
```

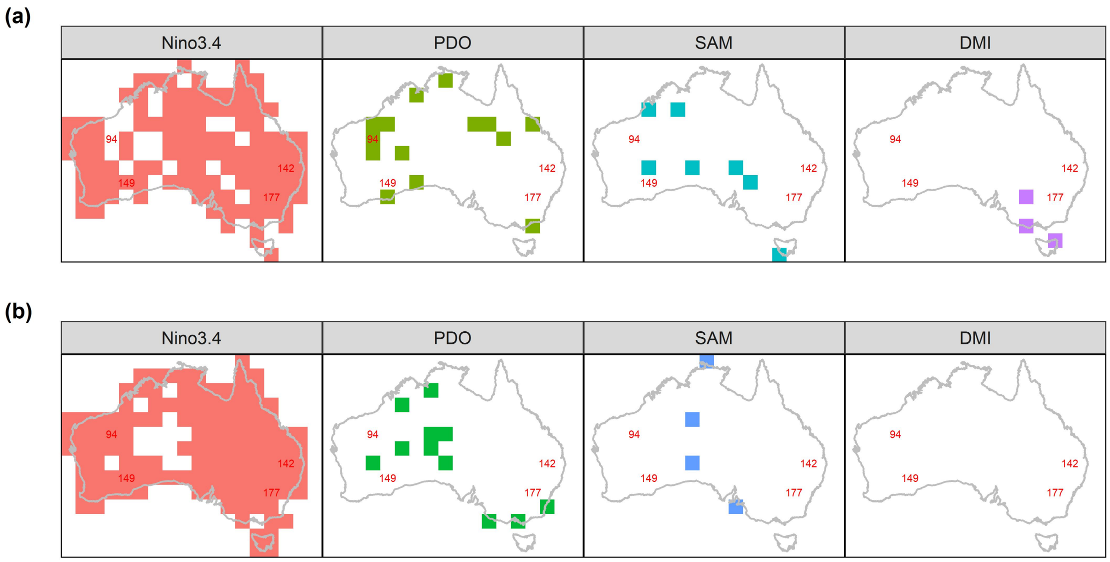

Figure 3: The most significant climate indices identified over Australia for different time scales of SPI. (a) SPI12; (b) SPI36.

```
#----------------------------------------------------------
#selection table for the entire region - Table 4
# tab.list
# Note that the number 1,2,3,4 here represnets the climate indices
# while in the Table 3 of manuscript it is their rank.
tab = cbind(c("SPI12","SPI12","SPI36","SPI36"),c("Original","VT","Original","VT"),t(sapply(tab.list,r
colnames(tab)=c("Indix","Model", Ind_CI,"Total")
tab1 <- tab[c(2,4,1,3),]

knitr::kable(
  tab1,
  booktabs = TRUE,
  align = "c",
  caption = 'Summary of climate indices selection.'
)%>% kable_styling(latex_options = "HOLD_position")
```

Table 1: Summary of climate indices selection.

Indix	Model	Nino34	PDO	SAM	DMI	Total
SPI12	VT	114	14	7	3	138
SPI36	VT	123	11	4	0	138
SPI12	Original	98	13	12	13	136
SPI36	Original	63	38	22	12	135

```
#selection table for the sampled grids
#tab1.list
for(i in 1:length(tab1.list)) print(lapply(tab1.list, function(ls) ls[[i]]))
#> [[1]]
#> [1] "1" "4" "2" "3"
#>
#> [[2]]
#> [1] "2" "3" "4" "1"
#>
#> [[3]]
#> [1] "1" "3" "4" "2"
#>
#> [[4]]
#> [1] "1" "2" "3" "4"
#>
#> [[1]]
#> [1] "1" "4"
#>
#> [[2]]
#> [1] "1" "3" "2" "4"
#>
#> [[3]]
#> [1] "1" "3" "2"
#>
#> [[4]]
#> [1] "1" "2" "3" "4"
#>
#> [[1]]
#> [1] "1" "2"
#>
#> [[2]]
#> [1] "1" "3" "4" "2"
#>
#> [[3]]
#> [1] "2" "4"
#>
#> [[4]]
#> [1] "1" "2" "3" "4"
#>
#> [[1]]
#> [1] "2" "1" "3"
#>
#> [[2]]
#> [1] "1" "2" "3" "4"
#>
#> [[3]]
```

```
#> [1] "4" "2" "3"
#>
#> [[4]]
#> [1] "1" "2" "3" "4"
```

Figure 4: Comparison of observed, predicted and predicted with variance transformation drought indices at four sampled grids

```
#-----------------------------------------------------------------------------
data(data.CI); data(Ind_AWAP.2.5)
Grid=Grid.sample #c(45,117,142,149)
Ind_CI <- colnames(data.CI)

#selection
p4.list <- list()
for(sc in c(12,36)){
    data.SPI.obs =  eval(parse(text=paste0("SPI.",sc)))

    ###model simulated response
    if(FALSE){
    load(paste0("./result/data.SPI.",sc,".selection_",mode,"_",wf,"_",k.folds,"folds.sample.Rdata"))
    load(paste0("./result/data.SPI.",sc,".weights_",mode,"_",wf,"_",k.folds,"folds.sample.Rdata"))
    load(paste0("./result/data.SPI.",sc,".mod_",mode,"_",wf,"_",k.folds,"folds.sample.Rdata"))
    load(paste0("./result/data.SPI.",sc,".ref_",mode,"_",wf,"_",k.folds,"folds.sample.Rdata"))
    } else { #cross check the result
    load(paste0("./result/data.SPI.",sc,".selection_",mode,"_",wf,"_",k.folds,"folds.Rdata"))
    load(paste0("./result/data.SPI.",sc,".weights_",mode,"_",wf,"_",k.folds,"folds.Rdata"))
    load(paste0("./result/data.SPI.",sc,".mod_",mode,"_",wf,"_",k.folds,"folds.Rdata"))
    load(paste0("./result/data.SPI.",sc,".ref_",mode,"_",wf,"_",k.folds,"folds.Rdata"))
    }

    #summary(sel.cv); head(sel.cv)
    sel <- vector("list",length=length(Grid))
    for(i in 1:length(Grid)){
      tmp1 <- NULL; tmp2 <- NULL
      for(j in 1:length(sel.cv)){
        tmp1 <- c(tmp1, sel.cv[[j]]$origin[[i]])
        tmp2 <- c(tmp2, sel.cv[[j]]$vt[[i]])
      }

      if(!is.null(tmp1)) sel[[i]]$origin <- data.frame(table(tmp1))[order(data.frame(table(tmp1))$Fre
      else sel[[i]]$origin <- NA
      if(!is.null(tmp2)) sel[[i]]$vt <- data.frame(table(tmp2))[order(data.frame(table(tmp2))$Freq, d
      else sel[[i]]$vt <- NA
    }

    #cross check with selection in Figure 3
    #summary(sel); head(sel)

    #-----------------------------
    ###density plot
    Ind <- Grid
    df.ref <- data.frame(Group=1, N=1:nrow(data.SPI.obs),
```

```r
                    rbind(cbind("Obs",data.SPI.obs[,Ind]),
                          cbind("Pred",data.SPI.ref[,Ind])))
colnames(df.ref) <- c("Group","N","Type",Ind)
df.ref.n <- gather(df.ref,"No","Value",4:(length(Ind)+3)) %>% spread("Type","Value")
df.ref.n$Obs <- as.numeric(df.ref.n$Obs)
df.ref.n$Pred <- as.numeric(df.ref.n$Pred)
df.ref.n$No <- as.numeric(df.ref.n$No)
#summary(df.ref.n)

df.mod <- data.frame(Group=2, N=1:nrow(data.SPI.obs),
                    rbind(cbind("Obs",data.SPI.obs[,Ind]),
                          cbind("Pred",data.SPI.mod[,Ind])))
colnames(df.mod) <- c("Group","N","Type",Ind)
df.mod.n <- gather(df.mod,"No","Value",4:(length(Ind)+3)) %>% spread("Type","Value")
df.mod.n$Obs <- as.numeric(df.mod.n$Obs)
df.mod.n$Pred <- as.numeric(df.mod.n$Pred)
df.mod.n$No <- as.numeric(df.mod.n$No)
#summary(df.mod.n)

data <- rbind(df.ref.n, df.mod.n)
limits.x <- c(-3.55,3.55); breaks.x <- seq(-3,3,1)
limits.y <- c(0,1); breaks.y <- seq(0,1,0.2)
Predictor.labs <- paste0("Sampled Grid: ",Ind)
names(Predictor.labs) <- Ind

p1 <- ggplot(data = data) +

  geom_density(aes(x=Obs, fill="Observed"),col="pink") +
  stat_density(aes(x=Pred, color=factor(Group)), geom="line",position="identity", lwd=1) +

  facet_wrap(No~., labeller = labeller(No = Predictor.labs)) +

  scale_x_continuous(breaks=breaks.x, limits=limits.x, expand = c(0,0)) +
  scale_y_continuous(breaks=breaks.y, limits=limits.y, expand = c(0,0)) +

  scale_color_manual(values=c("red","blue"), labels=c("Predicted","Predicted(VT)"))+
  scale_fill_manual(values=c("pink"))+

  theme_bw() +
  theme(text = element_text(size = 12),
        plot.margin = unit(c(1,1,1,1), "cm"),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_blank(),

        legend.title = element_blank(),
        #legend.position= c(0.9,0.1),
        legend.position="bottom",
        legend.key.width = unit(1,"cm"),

        #x and y axis
        axis.title.x = element_blank()
        #axis.title.y = element_blank()

  )
#p1
p4.list[[length(p4.list)+1]] <- p1
```

```
    #------------------------------
    #PDF skill scores
    SPI.ref.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
    SPI.mod.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))

    SPI.ref.PDF[,Grid] <- sapply(Grid, function(i) overlap(list(data.SPI.obs[,i],data.SPI.ref[,i]),na
    SPI.mod.PDF[,Grid] <- sapply(Grid, function(i) overlap(list(data.SPI.obs[,i],data.SPI.mod[,i]),na

    SPI.mod.PDF.OL <- (SPI.mod.PDF - SPI.ref.PDF)/SPI.ref.PDF*100
    print(SPI.mod.PDF.OL[,Grid])

    df1 <- cbind(Group=1,Grid, SPI.ref.PDF[,Grid])
    df2 <- cbind(Group=2,Grid, SPI.mod.PDF[,Grid])

    df.PDF <- data.frame(rbind(df1,df2))
    names(df.PDF) <- c("Group","No","PDF")

    p2<-ggplot(df.PDF,aes(x=factor(No),y=PDF)) +
      geom_bar(aes(fill=factor(Group)), position = "dodge", stat="identity") +

      scale_y_continuous(breaks=seq(0,1,0.2),limits=c(0,1),expand = c(0,0)) +
      scale_x_discrete(labels=paste0("Grid: ",sort(Grid))) +
      scale_fill_manual(values=c("red","blue"), labels=c("Predicted","Predicted(VT)"))+

      #xlab("Sampled Grid") +
      ylab(paste0("PDF Skill Score")) +

      theme_bw() +
      theme(text = element_text(size = 12),
            plot.margin = unit(c(1,1,1,1), "cm"),
            panel.grid.minor = element_blank(),
            panel.grid.major = element_blank(),
            legend.position="bottom",

            legend.key.width = unit(1,"cm"),
            legend.title = element_blank(),

            #x and y axis
            axis.title.x = element_blank()
            #axis.title.y = element_blank()

      )
    #p2
    p4.list[[length(p4.list)+1]] <- p2
}
#> [1] 35.239776  8.533196 20.170680 49.503104
#> [1] 58.75786 84.85021 47.61332 37.40719
cowplot::plot_grid(plotlist = p4.list, nrow=2, labels = c("(a)","(b)","(c)","(d)"), label_size = 12,
```

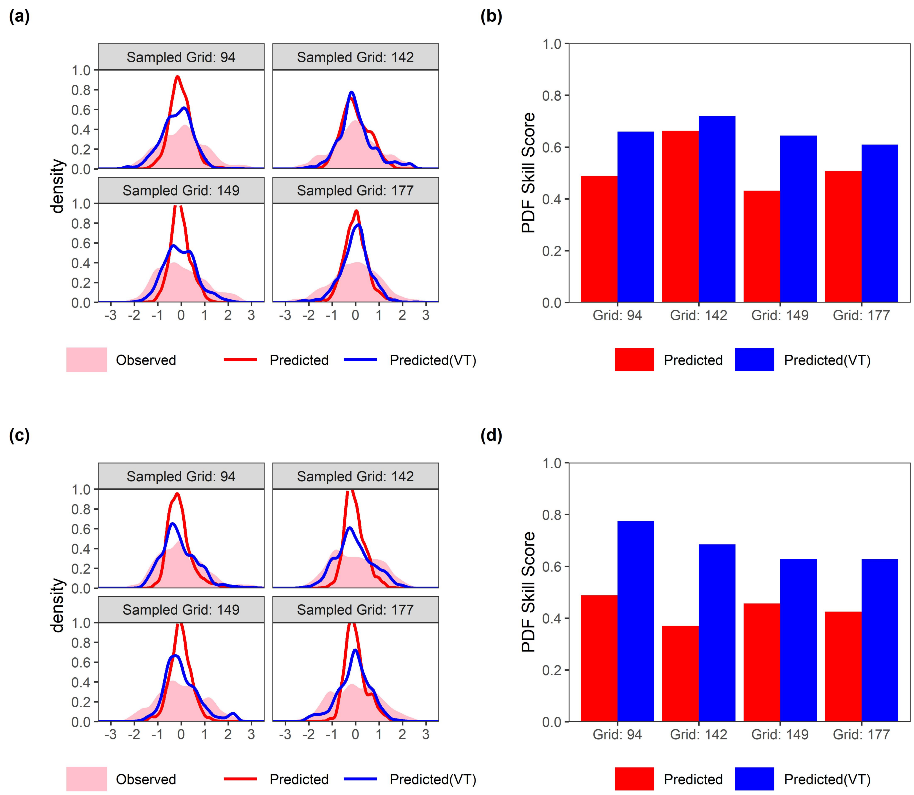

Figure 4: Comparison of observed, predicted and predicted with variance transformation drought indices at four sampled grids. SPI12: (a) Density plot (b) PDF skill scores; SPI36: (c) Density plot (d) PDF skill scores

Figure 5: The percent improvement of PDF skill scores compared to the non-wavelet model(biased).

```
#--------------------------------------------------------
data(data.CI); data(Ind_AWAP.2.5); data(lat_lon.2.5)
Grids=Ind_AWAP.2.5
Ind_CI <- colnames(data.CI)

p5.list <- list(); SPI.PDF.biased <- NULL
for(sc in c(12,36)){
    data.SPI.obs =  eval(parse(text=paste0("SPI.",sc)))

    ###model simulated response
    load(paste0("./result/data.SPI.",sc,".mod_",mode,"_",wf,"_",k.folds,"folds.Rdata"))
    load(paste0("./result/data.SPI.",sc,".ref_",mode,"_",wf,"_",k.folds,"folds.Rdata"))

    #---------------------------------------
    ###Improved skill scores spatial plot
    SPI.ref.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
    SPI.mod.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))

    SPI.ref.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.ref[,i]),
    SPI.mod.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.mod[,i]),
```

```r
SPI.mod.PDF.OL <- (SPI.mod.PDF - SPI.ref.PDF)/SPI.ref.PDF*100

summary(SPI.ref.PDF[,Grids])
summary(SPI.mod.PDF[,Grids])
summary(SPI.mod.PDF.OL[,Grids])

print(sum(SPI.mod.PDF.OL[,Grids]>0)/length(Grids)) # percentage of improvements of grids

#max.OL <- ifelse(max(SPI.mod.PDF.OL[,Grids])<60, 60, round_any(max(SPI.mod.PDF.OL[,Grids]), 10,
max.OL=60; min.OL=0
#---------------------------------------
for(data in c("SPI.mod.PDF.OL")){
  ###matrix to spatial points
  SPI.obs.mat <- data.frame(t(eval(parse(text = data))))

  SPI.obs.ras <- raster::rasterFromXYZ(cbind(lat_lon.2.5, SPI.obs.mat))
  SPI.obs.ras

  if(sum(na.omit(SPI.obs.mat)<0)){
  SPI.obs.p <- data.frame(lat_lon.2.5[which(SPI.obs.mat<0),], ID=0)
  coordinates(SPI.obs.p) = ~lon+lat
  proj4string(SPI.obs.p) = "+proj=longlat +datum=WGS84"
  #gridded(SPI.obs.p) <- TRUE
  sl1 <- list("sp.points",SPI.obs.p, first=FALSE, col="grey28", pch=19)
  } else sl1 <- list("sp.points",NULL, first=FALSE, col="grey28")

  sl2 <- list("sp.polygons",Aus_map, first=FALSE, col="grey")

  ## labels and color
  label <- seq(min.OL,max.OL,10);labelat = round(label,digits=2);
  labeltext = label; labeltext[length(label)]=paste0(">",max.OL)
  pal <-  colorRampPalette(c("lightblue", "blue"))
  my.palette <- pal(length(label))

  #df.ras.mask <- mask(SPI.obs.ras, Aus_map)
  p <- spplot(SPI.obs.ras, xlim=c(110,156), ylim=c(-45,-9),
              col.regions = my.palette,
              zlim=c(-20,max.OL),
              at = label, # colour breaks
              sp.layout = list(sl1,sl2),
              colorkey = list(space="bottom",
                              height = 0.5,
                              width = 1,
                              labels = list(at = labelat, labels = labeltext, cex=0.6)
              )
  )
  p
  p5.list[[length(p5.list)+1]] <- p

}

#-----------------------------------
# #boxplot - PDF skill scores
# par(mfrow=c(1,1),mar=c(3,4,2,2),mgp=c(1.5,0.6,0),ps=8, bg="transparent")
# boxplot(cbind(t(SPI.ref.PDF),t(SPI.mod.PDF)), ylim=c(0,1),xaxt="n", cex.main=0.8,
#         ylab="PDF Skill Score", col=c("red","blue"))
```

```r
    # axis(1, at= c(1,2), labels=c("Predicted","Predicted(VT)"))

    #----------------------------------
    #xyplot - PDF skill scores
    par(mfrow=c(1,1),mar=c(4,4,2,2),mgp=c(1.6,0.6,0),ps=10, pty="s", bg="transparent")
    plot(SPI.ref.PDF, SPI.mod.PDF, xlim=c(0,1), ylim=c(0,1),
         xlab="Predicted", ylab="Predicted(VT)", pch=19,col="blue")
    abline(a=0,b=1,col="grey")

    p5.list[[length(p5.list)+1]] <- recordPlot()

    # #--------------------------------------
    # ###RMSE
    # SPI.ref.RMSE <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
    # SPI.mod.RMSE <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
    #
    # SPI.ref.RMSE[,Grids] <- sapply(Grids, function(i) sqrt(mean((data.SPI.obs[,i]-data.SPI.ref[,i])
    # SPI.mod.RMSE[,Grids] <- sapply(Grids, function(i) sqrt(mean((data.SPI.obs[,i]-data.SPI.mod[,i])
    #
    # print(sum(SPI.ref.RMSE>SPI.mod.RMSE, na.rm=T)/length(Grids)) # percentage of improvements of gr
    #
    # #xyplot - RMSE
    # par(mfrow=c(1,1),mar=c(3,4,2,2),mgp=c(1.6,0.6,0),ps=10, bg="transparent")
    # plot(SPI.ref.RMSE, SPI.mod.RMSE, xlim=c(0,2), ylim=c(0,2),
    #      xlab="Predicted", ylab="Predicted(VT)", pch=19,col="blue")
    # abline(a=0,b=1,col="grey")
    #
    # p5.list[[length(p5.list)+1]] <- recordPlot()
    #
    # #--------------------------------------
    # ###Correlation
    # SPI.ref.cor <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
    # SPI.mod.cor <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
    #
    # SPI.ref.cor[,Grids] <- sapply(Grids, function(i) cor(data.SPI.obs[,i],data.SPI.ref[,i]))
    # SPI.mod.cor[,Grids] <- sapply(Grids, function(i) cor(data.SPI.obs[,i],data.SPI.mod[,i]))
    #
    # print(sum(SPI.ref.cor<SPI.mod.cor,na.rm=T)/length(Grids)) # percentage of improvements of grids
    #
    # #xyplot - Correlation
    # par(mfrow=c(1,1),mar=c(3,4,2,2),mgp=c(1.6,0.6,0),ps=10, bg="transparent")
    # plot(SPI.ref.cor, SPI.mod.cor, xlim=c(-0.5,1), ylim=c(-0.5,1),
    #      xlab="Predicted", ylab="Predicted(VT)", pch=19,col="blue")
    # abline(a=0,b=1,col="grey")
    #
    # p5.list[[length(p5.list)+1]] <- recordPlot()

    SPI.PDF.biased <- rbind(SPI.PDF.biased, cbind(sc, t(SPI.ref.PDF), t(SPI.mod.PDF)))
}
#> [1] 0.9927536
#> [1] 0.9710145
#----------------------------------------------
#combine subplots
cowplot::plot_grid(plotlist = p5.list, nrow=2, labels = c("(a)","(b)","(c)","(d)"), label_size = 12,
                   rel_widths = c(1, 1, 1, 1), hjust=0)
```

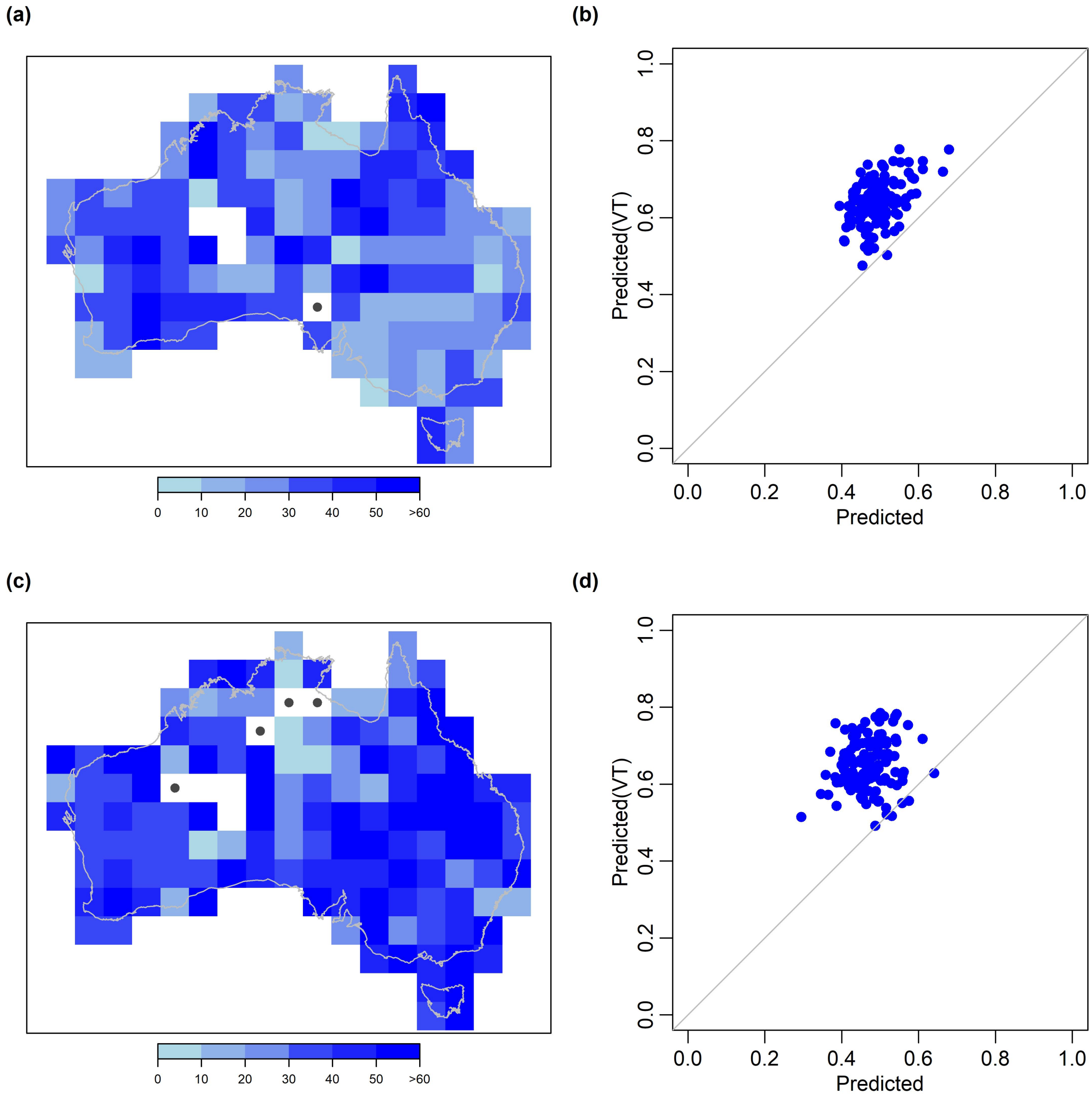

Figure 5: The most significant climate indices identified over Australia for different time scales of SPI. (a) SPI12; (b) SPI36.

Figure 6: Comparison between biased and unbiased variance transformation approach.

```
#----------------------------------------------------------
SPI.PDF.biased <- NULL; SPI.PDF.unbiased <- NULL
for(sc in c(12,36)){
      data.SPI.obs =  eval(parse(text=paste0("SPI.",sc)))
      path.biased <- paste0("./result/data.SPI.",sc,".mod_",mode,"_",wf,"_",k.folds,"folds.Rdata")
      path.unbiased <- paste0("./result/data.SPI.unbiased.",sc,".mod_",mode,"_",wf,"_",k.folds,"folds
      #-------------------------------------
      ###model simulated response- biased
      SPI.mod.biased<-get(load(path.biased))

      ###Improved skill scores spatial plot
```

```r
      #SPI.ref.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
      SPI.mod.PDF.biased <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))

      #SPI.ref.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.ref[,i
      SPI.mod.PDF.biased[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],SPI.mod.b

      SPI.PDF.biased <- rbind(SPI.PDF.biased, cbind(sc, wf, t(SPI.mod.PDF.biased)))

      #---------------------------------------
      ###model simulated response - unbiased
      SPI.mod.unbiased<-get(load(path.unbiased))
      #load(paste0("./result/data.SPI.unbiased.",sc,".ref_",mode,"_",wf,"_",k.folds,"folds.Rdata"))

      ###Improved skill scores spatial plot
      #SPI.ref.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
      SPI.mod.PDF.unbiased <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))

      #SPI.ref.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.ref[,i
      SPI.mod.PDF.unbiased[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],SPI.mod

      SPI.PDF.unbiased <- rbind(SPI.PDF.unbiased, cbind(sc, wf, t(SPI.mod.PDF.unbiased)))
}

summary(SPI.PDF.biased); summary(SPI.PDF.unbiased)
#>   sc         wf                       V3
#>  12:252   haar:504   0.475131428760356:  1
#>  36:252              0.491844154804781:  1
#>                      0.502746453482338:  1
#>                      0.51433833936426 :  1
#>                      0.514449815377981:  1
#>                      (Other)          :271
#>                      NA's             :228
#>   sc         wf                       V3
#>  12:252   haar:504   0.363243199610309:  1
#>  36:252              0.519722683082154:  1
#>                      0.525367011858875:  1
#>                      0.525933770891857:  1
#>                      0.526497365443261:  1
#>                      (Other)          :271
#>                      NA's             :228

if(TRUE){
df.PDF <- rbind(data.frame(Group="biased",SPI.PDF.biased), data.frame(Group="unbiased",SPI.PDF.unbias
colnames(df.PDF) <- c("Group","SPI","wf","VT")
df.PDF$VT <- as.numeric(as.character(df.PDF$VT))
#df.PDF$wf <- factor(df.PDF$wf, levels=c("haar","d8","d16"), labels = c("haar"="Haar(d2)","d8"="d8",'
#df.PDF <- gather(df.PDF, Model, PDF, 3:4)
summary(df.PDF)

labeller.labs <- c("SPI12","SPI36")
names(labeller.labs) <- c("12","36")

###Boxplot
p <-ggplot(df.PDF, aes(x=Group, y=VT))+
       geom_boxplot(aes(fill=wf)) +
       #geom_violin(aes(fill=Model)) +
```

```
       stat_summary(fun=mean, geom="point", shape=20, size=5, color="blue") +
       facet_grid(.~SPI,labeller = labeller(SPI = labeller.labs)) +
       labs(y="PDF skill scores")+
       theme_bw() +
       theme(text = element_text(size = 20,family="Serif"),
             plot.margin = unit(c(1,1,0.5, 1), "cm"),
             # panel.grid.minor = element_blank(),
             # panel.grid.major = element_blank(),
             # axis.text.y = element_text(angle = 90, hjust=0.5, size=12),
             # axis.title.x = element_text(size=16),
             # axis.title.y = element_text(size=16),
             axis.title.x = element_blank(),
             legend.position="none"
             )
print(p)
###Violin plot
p1 <-ggplot(df.PDF, aes(x=Group, y=VT))+
       #geom_boxplot(aes(fill=Model)) +
       geom_violin(aes(fill=wf)) +
       stat_summary(fun=mean, geom="point", shape=20, size=5, color="blue") +
       facet_grid(.~SPI, labeller = labeller(SPI = labeller.labs)) +
       labs(y="PDF skill scores")+
       theme_bw() +
       theme(text = element_text(size = 20,family="Serif"),
             plot.margin = unit(c(1,1,0.5, 1), "cm"),
             # panel.grid.minor = element_blank(),
             # panel.grid.major = element_blank(),
             # axis.text.y = element_text(angle = 90, hjust=0.5, size=12),
             # axis.title.x = element_text(size=16),
             # axis.title.y = element_text(size=16),
             axis.title.x = element_blank(),
             legend.position="none"
             )
p1
#ggsave(paste0(fig.path,"fig6-11.jpeg"),p,height=9,width=9,dpi=500)
#ggsave(paste0(fig.path,"fig6-12.jpeg"),p1,height=9,width=9,dpi=500)
}
```

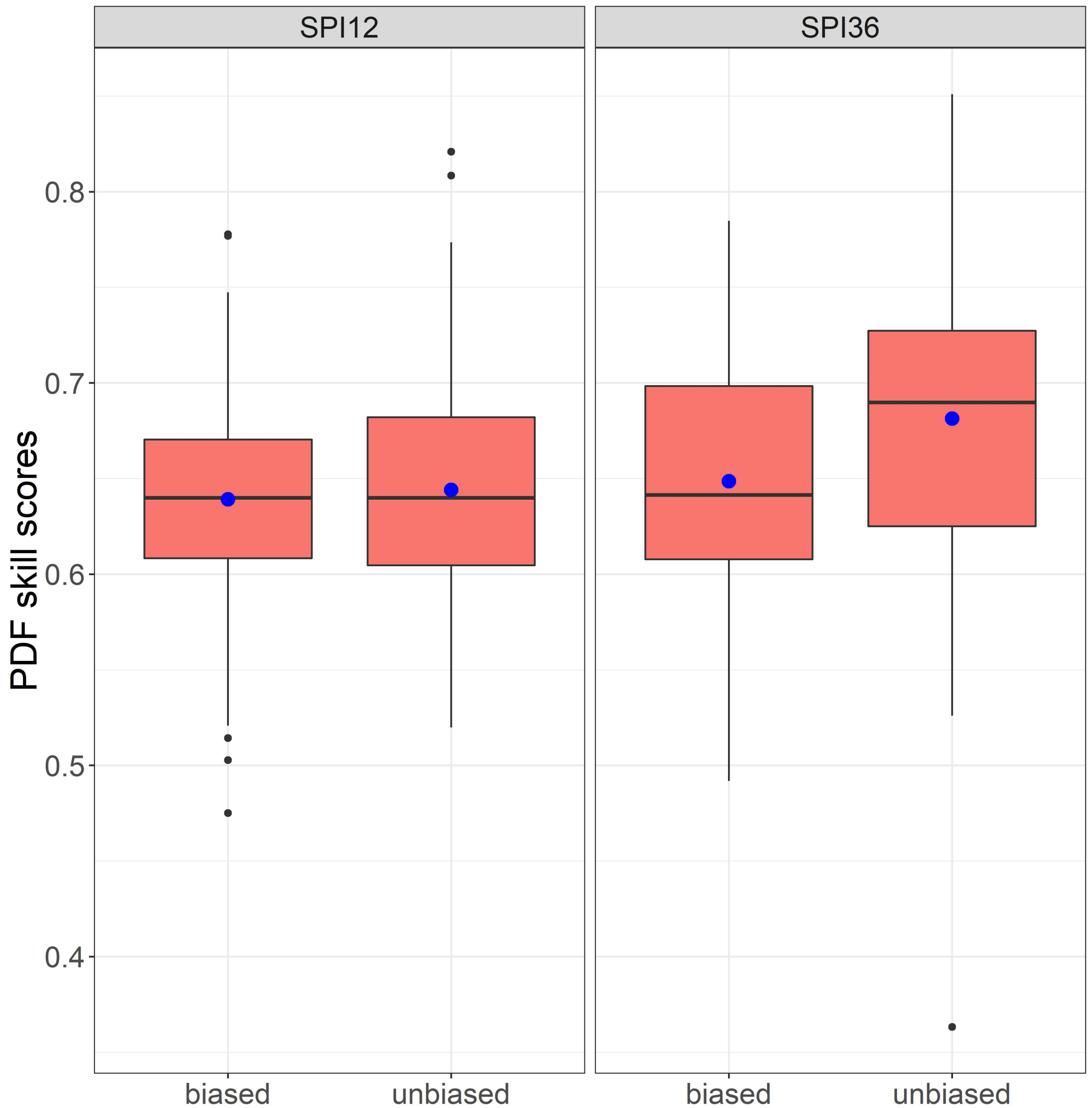

Figure 6: Comparison between biased and unbiased variance transformation approach.

```
Grids=Ind_AWAP.2.5

#----------------------------------------------------------
SPI.PDF.biased <- NULL; SPI.PDF.unbiased <- NULL
for(wf in c("haar","d8", "d16")){
  for(sc in c(12,36)){
      data.SPI.obs =  eval(parse(text=paste0("SPI.",sc)))
      path.biased <- paste0("./result/data.SPI.",sc,".mod_",mode,"_",wf,"_","2folds.Rdata")
      path.unbiased <- paste0("./result/data.SPI.unbiased.",sc,".mod_",mode,"_",wf,"_","2folds.Rdata"
      #----------------------------------------
      ###model simulated response- biased
      SPI.mod.biased<-get(load(path.biased))

      ###Improved skill scores spatial plot
      #SPI.ref.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
      SPI.mod.PDF.biased <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
```

```
  #SPI.ref.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.ref[,i
  SPI.mod.PDF.biased[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],SPI.mod.b

  SPI.PDF.biased <- rbind(SPI.PDF.biased, cbind(sc, wf, t(SPI.mod.PDF.biased)))

  #---------------------------------------
  ###model simulated response - unbiased
  SPI.mod.unbiased<-get(load(path.unbiased))
  #load(paste0("./result/data.SPI.unbiased.",sc,".ref_",mode,"_",wf,"_",k.folds,"folds.Rdata"))

  ###Improved skill scores spatial plot
  #SPI.ref.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
  SPI.mod.PDF.unbiased <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))

  #SPI.ref.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.ref[,i
  SPI.mod.PDF.unbiased[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],SPI.mod

  SPI.PDF.unbiased <- rbind(SPI.PDF.unbiased, cbind(sc, wf, t(SPI.mod.PDF.unbiased)))
 }
}

summary(SPI.PDF.biased); summary(SPI.PDF.unbiased)
#>   sc        wf                     V3
#>  12:756   d16 :504   0.467637430336387:  1
#>  36:756   d8  :504   0.485314832021182:  1
#>           haar:504   0.485512026782812:  1
#>                      0.4878217928411  :  1
#>                      0.489135455012063:  1
#>                      (Other)          :823
#>                      NA's             :684
#>   sc        wf                     V3
#>  12:756   d16 :504   0.467988706977782:  1
#>  36:756   d8  :504   0.477225594835066:  1
#>           haar:504   0.480354987920512:  1
#>                      0.484164332152402:  1
#>                      0.492057214014494:  1
#>                      (Other)          :823
#>                      NA's             :684

if(TRUE){
df.PDF <- rbind(data.frame(Group="biased",SPI.PDF.biased), data.frame(Group="unbiased",SPI.PDF.unbias
colnames(df.PDF) <- c("Group","SPI","wf","VT")
df.PDF$VT <- as.numeric(as.character(df.PDF$VT))
df.PDF$wf <- factor(df.PDF$wf, levels=c("haar","d8","d16"), labels = c("haar"="Haar(d2)","d8"="d8","d
#df.PDF <- gather(df.PDF, Model, PDF, 3:4)

summary(df.PDF)
tab.mean <- plyr::ddply(na.omit(df.PDF), .(Group,SPI,wf), summarize,  mean = round(mean(VT), 3))
tab.sd <- plyr::ddply(na.omit(df.PDF), .(Group,SPI,wf), summarize,  sd = round(sd(VT), 3))
tab.median <- plyr::ddply(na.omit(df.PDF), .(Group,SPI,wf), summarize,  median = round(median(VT), 3)

tab5 <- rbind(cbind(Metric="Mean",spread(tab.mean, Group, mean)),
              cbind(Metric="Median",spread(tab.median, Group, median)),
              cbind(Metric="SD",spread(tab.sd, Group, sd)))
tab5 <- cbind(mutate(tab5, dif=round((unbiased-biased),3)), J=rep(c(9,8,7),6))
```

```
tab5

kable(tab5[,c("Metric","SPI","wf","J","biased","unbiased","dif")], caption = "Comparision between bia
kable_styling("striped",position = "center") %>%
#add_header_above(c("Method"=1,"vwnd" = 3, "uwnd" = 3, "air" = 3, "hgt"=3, "shum"=3, "slp"=1)) #%>%
collapse_rows(columns = 1:2, valign = "middle") %>%
#save_kable(paste0(fig.path,"tab1.png"),zoom = 1.5)
kable_as_image(filename=paste0(fig.path,"tab1"), file_format = "png")
#as_image(file=paste0(fig.path,"tab1.png"))

knitr::include_graphics(paste0(fig.path,"tab1.png"))
}
```

Table 1: (#tab:tab5)Comparision between biased and unbiased estimator with varying wavelet filters

Metric	SPI	wf	J	biased	unbiased	dif
Mean	12	Haar(d2)	9	0.664	0.672	0.008
		d8	8	0.658	0.685	0.027
		d16	7	0.635	0.652	0.017
	36	Haar(d2)	9	0.679	0.699	0.020
		d8	8	0.716	0.718	0.002
		d16	7	0.697	0.719	0.022
Median	12	Haar(d2)	9	0.659	0.678	0.019
		d8	8	0.664	0.680	0.016
		d16	7	0.632	0.653	0.021
	36	Haar(d2)	9	0.684	0.703	0.019
		d8	8	0.703	0.724	0.021
		d16	7	0.698	0.726	0.028
SD	12	Haar(d2)	9	0.065	0.068	0.003
		d8	8	0.060	0.075	0.015
		d16	7	0.065	0.071	0.006
	36	Haar(d2)	9	0.069	0.072	0.003
		d8	8	0.064	0.068	0.004
		d16	7	0.068	0.070	0.002

Supplement Materical

This is the main feature of WASP package transforming predictor variables for both calibration and validation periods shown in Fig. S1.

```r
#----------------------------------------------------------------------------------------
#The transformation is done in Figure 1
start.cal <- c(1910,1); start.val <- c(1960,1)
p.list <- list()
Grid = 149 # A sample grid
#-----------------------------------------------
#plot before and after vt - calibration
if(TRUE){
  ndim = ncol(data.CI.ts); CI.names = colnames(data.CI.ts)
  x <- ts(dwt$x, start=start.cal, frequency = 12)
  dp <- ts(dwt$dp, start=start.cal, frequency = 12)
  dp.n <- ts(dwt$dp.n, start=start.cal, frequency = 12)

  par(mfrow=c(ndim+1,1),mar=c(2,4,2,2),bg = "white",pty="m",ps=8)
  ts.plot(x,xlab=NA, main=paste0("Sampled Grid: ", Grid), ylab=paste0("SPI",12), col=c("black"),lwd=c
  #ts.plot(x,xlab=NA, ylab=paste0("SPI",12), col=c("black"),lwd=c(1))
  for(nc in 1:ndim)
    ts.plot(dp[,nc],dp.n[,nc],xlab=NA,ylab=paste0(CI.names[nc]),
            col=c("red","blue"),lwd=c(1,1),lty=c(1,2))
  p.list[[length(p.list)+1]] <- grDevices::recordPlot()
}
#-----------------------------------------------
#plot before and after vt - validation
if(TRUE){
  ndim = ncol(data.CI.ts); CI.names = colnames(data.CI.ts)
  x <- ts(dwt.val$x, start=start.val, frequency = 12)
  dp <- ts(dwt.val$dp, start=start.val, frequency = 12)
  dp.n <- ts(dwt.val$dp.n, start=start.val, frequency = 12)
  par(mfrow=c(ndim+1,1),mar=c(2,4,2,2),bg = "white",pty="m",ps=8)
  ts.plot(x, xlab=NA, main=paste0("Sampled Grid: ",Grid), ylab=paste0("SPI",12), col=c("black"),lwd=c
  #ts.plot(x,xlab=NA, ylab=paste0("SPI",12), col=c("black"),lwd=c(1))
  for(nc in 1:ndim)
    ts.plot(dp[,nc],dp.n[,nc],xlab=NA,ylab=paste0(CI.names[nc]),
            col=c("red","blue"),lwd=c(1,1),lty=c(1,2))
  p.list[[length(p.list)+1]] <- grDevices::recordPlot()
}
#-----------------------------------------------
#combine two subplots
cowplot::plot_grid(plotlist = p.list, ncol=2, labels = c("(a)","(b)"), label_size = 12, hjust=0)
```

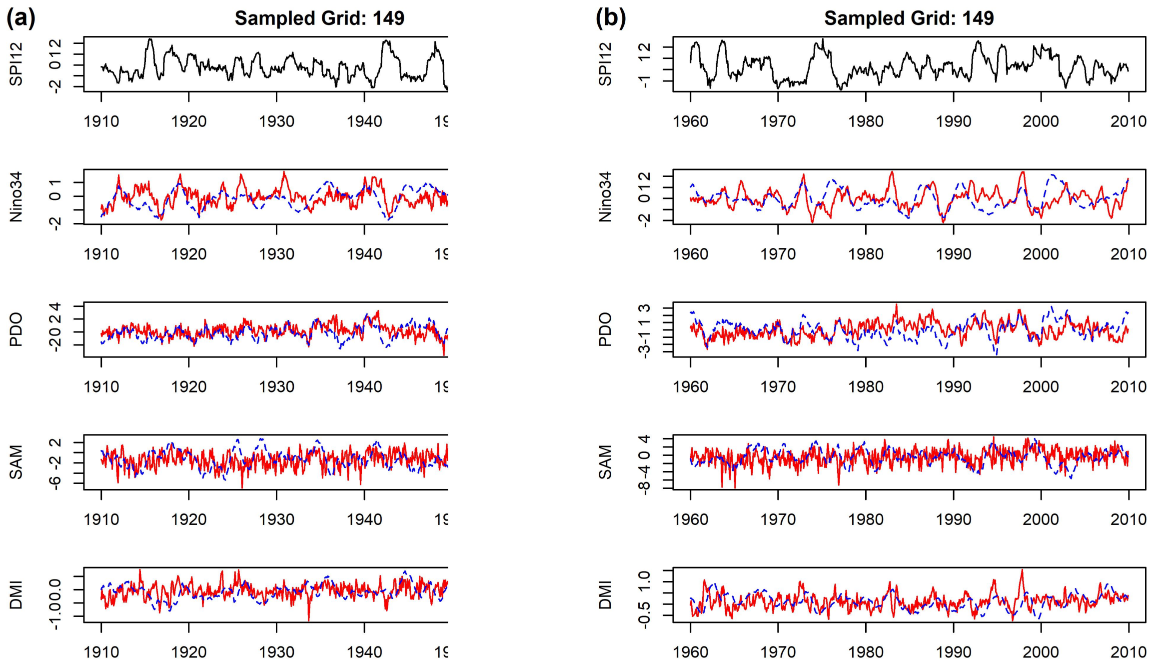

Figure S1: Example of predictor variables before (solid red line) and after (dashed blue line) variance transformation corresponding to the response SPI12 (solid black line) at a sampled grid. (a) Calibration (b) Validation

This is the map of Grid Index over Australia shown in Fig. S2. Grids with missing data (i.e., where 25% of rainfall values are zero or missing) are in white color, while the investigated four randomly sampled grids are highlighted in red color.

```
data(Aus_map)
data(lat_lon.2.5)
data(data.AWAP.2.5)

grids.ras <- raster::rasterFromXYZ(cbind(lat_lon.2.5, ID=1:nrow(lat_lon.2.5)))
#grids.ras
#----------------------------------------
sl1 <- list("sp.polygons",Aus_map, first=FALSE, col="grey")

#Missing grids and sampled grids
Ind_AWAP.2.5_NaN1 <- which(apply(data.AWAP.2.5, 2, function(m) sum(is.na(m)))==nrow(data.AWAP.2.5))
Ind_AWAP.2.5_NaN2 <- which(apply(data.AWAP.2.5, 2, function(m) sum(m==0))>=0.25*nrow(data.AWAP.2.5))
Ind_AWAP.2.5_NaN <- c(Ind_AWAP.2.5_NaN1,Ind_AWAP.2.5_NaN2)

grids.col=rep("green",nrow(lat_lon.2.5))
grids.col[Ind_AWAP.2.5_NaN] = "white"
grids.col[Grid.sample] = "red" #randomly sampled first and then fixed for investigation
sl2 <- list("sp.text", sp::coordinates(grids.ras), txt=1:nrow(lat_lon.2.5), cex=1, col=grids.col)

#extent(grids.ras)
p <- spplot(grids.ras, xlim=c(111.75,156.75), ylim=c(-44.75,-9.75),
            col.regions="white",
            border="black",
            sp.layout = list(sl1, sl2),
            colorkey=FALSE
            )
```

```
p
```

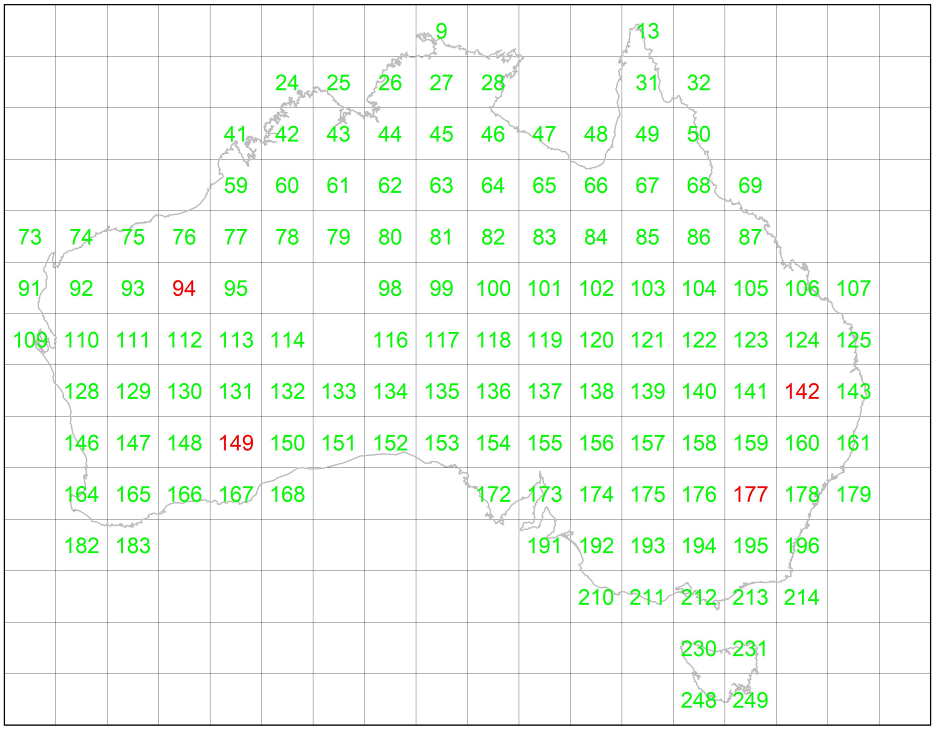

Figure S2: Map of grid index over Australia.

Fig. S3 shows the most significant drivers (i.e. the predictor selected first in the PIC process from the set of original climate indices) for both SPI12 and SPI36.

```
cowplot::plot_grid(plotlist = p0.list[c(1,3)], nrow=2, labels = c("(a)","(b)"), label_size = 12, hjus
```

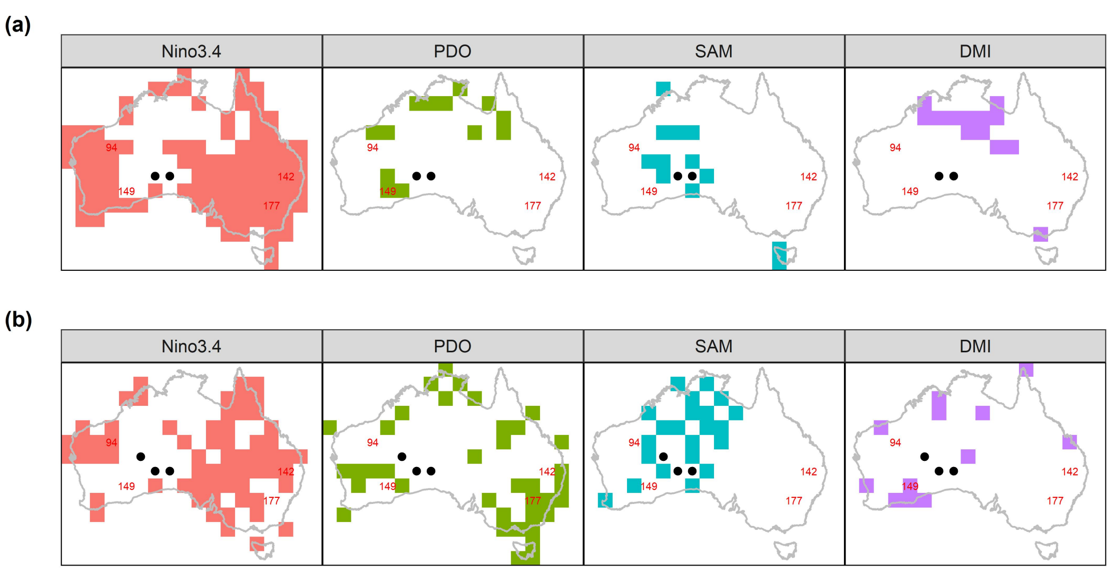

Figure S3: The most significant climate indices identified over Australia for different time scales of SPI. (a) SPI12; (b) SPI36.

Fig. S4 shows the percent improvement of PDF skill scores compared to the non-wavelet model(unbiased).

```
#---------------------------------------------------------
data(data.CI); data(Ind_AWAP.2.5); data(lat_lon.2.5)
Grids=Ind_AWAP.2.5; data(SPI.12); data(SPI.36)
Ind_CI <- colnames(data.CI)
wf="haar"; k.folds=4

p7.list <- list(); SPI.PDF.unbiased <- NULL
for(sc in c(12,36)){
    data.SPI.obs =  eval(parse(text=paste0("SPI.",sc)))

    ###model simulated response
    data.SPI.mod <- get(load(paste0("./result/data.SPI.unbiased.",sc,".mod_",mode,"_",wf,"_",k.folds,
    data.SPI.ref <- get(load(paste0("./result/data.SPI.unbiased.",sc,".ref_",mode,"_",wf,"_",k.folds,

    #---------------------------------------
    ###Improved skill scores spatial plot
    SPI.ref.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
    SPI.mod.PDF <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))

    SPI.ref.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.ref[,i]),
    SPI.mod.PDF[,Grids] <- sapply(Grids, function(i) overlap(list(data.SPI.obs[,i],data.SPI.mod[,i]),
    SPI.mod.PDF.OL <- (SPI.mod.PDF - SPI.ref.PDF)/SPI.ref.PDF*100

    summary(SPI.ref.PDF[,Grids])
    summary(SPI.mod.PDF[,Grids])
    summary(SPI.mod.PDF.OL[,Grids])

    print(sum(SPI.mod.PDF.OL[,Grids]>0)/length(Grids)) # percentage of improvements of grids

    #max.OL <- ifelse(max(SPI.mod.PDF.OL[,Grids])<60, 60, round_any(max(SPI.mod.PDF.OL[,Grids]), 10,
    max.OL=60; min.OL=0
    #---------------------------------------
    for(data in c("SPI.mod.PDF.OL")){
      ###matrix to spatial points
      SPI.obs.mat <- data.frame(t(eval(parse(text = data))))

      SPI.obs.ras <- raster::rasterFromXYZ(cbind(lat_lon.2.5, SPI.obs.mat))
      SPI.obs.ras

      if(sum(na.omit(SPI.obs.mat)<0)){
      SPI.obs.p <- data.frame(lat_lon.2.5[which(SPI.obs.mat<0),], ID=0)
      coordinates(SPI.obs.p) = ~lon+lat
      proj4string(SPI.obs.p) = "+proj=longlat +datum=WGS84"
      #gridded(SPI.obs.p) <- TRUE
      sl1 <- list("sp.points",SPI.obs.p, first=FALSE, col="grey28", pch=19)
      } else sl1 <- list("sp.points",NULL, first=FALSE, col="grey28")

      sl2 <- list("sp.polygons",Aus_map, first=FALSE, col="grey")

      ## labels and color
      label <- seq(min.OL,max.OL,10);labelat = round(label,digits=2);
      labeltext = label; labeltext[length(label)]=paste0(">",max.OL)
      pal <-  colorRampPalette(c("lightblue", "blue"))
      my.palette <- pal(length(label))
```

```r
  #df.ras.mask <- mask(SPI.obs.ras, Aus_map)
  p <- spplot(SPI.obs.ras, xlim=c(110,156), ylim=c(-45,-9),
              col.regions = my.palette,
              zlim=c(-20,max.OL),
              at = label, # colour breaks
              sp.layout = list(sl1,sl2),
              colorkey = list(space="bottom",
                              height = 0.5,
                              width = 1,
                              labels = list(at = labelat, labels = labeltext, cex=0.6)
              )
  )
  p
  p7.list[[length(p7.list)+1]] <- p

}

#-----------------------------------
# #boxplot - PDF skill scores
# par(mfrow=c(1,1),mar=c(3,4,2,2),mgp=c(1.5,0.6,0),ps=8, bg="transparent")
# boxplot(cbind(t(SPI.ref.PDF),t(SPI.mod.PDF)), ylim=c(0,1),xaxt="n", cex.main=0.8,
#         ylab="PDF Skill Score", col=c("red","blue"))
# axis(1, at= c(1,2), labels=c("Predicted","Predicted(VT)"))

#-----------------------------------
#xyplot - PDF skill scores
par(mfrow=c(1,1),mar=c(4,4,2,2),mgp=c(1.6,0.6,0),ps=10, pty="s", bg="transparent")
plot(SPI.ref.PDF, SPI.mod.PDF, xlim=c(0,1), ylim=c(0,1),
     xlab="Predicted", ylab="Predicted(VT)", pch=19,col="blue")
abline(a=0,b=1,col="grey")

p7.list[[length(p7.list)+1]] <- recordPlot()

# #--------------------------------------
# ###RMSE
# SPI.ref.RMSE <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
# SPI.mod.RMSE <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
#
# SPI.ref.RMSE[,Grids] <- sapply(Grids, function(i) sqrt(mean((data.SPI.obs[,i]-data.SPI.ref[,i])
# SPI.mod.RMSE[,Grids] <- sapply(Grids, function(i) sqrt(mean((data.SPI.obs[,i]-data.SPI.mod[,i])
#
# print(sum(SPI.ref.RMSE>SPI.mod.RMSE, na.rm=T)/length(Grids)) # percentage of improvements of gr
#
# #xyplot - RMSE
# par(mfrow=c(1,1),mar=c(3,4,2,2),mgp=c(1.6,0.6,0),ps=10, bg="transparent")
# plot(SPI.ref.RMSE, SPI.mod.RMSE, xlim=c(0,2), ylim=c(0,2),
#      xlab="Predicted", ylab="Predicted(VT)", pch=19,col="blue")
# abline(a=0,b=1,col="grey")
#
# p7.list[[length(p7.list)+1]] <- recordPlot()
#
# #--------------------------------------
# ###Correlation
# SPI.ref.cor <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
# SPI.mod.cor <- matrix(NA, nrow=1,ncol=ncol(data.SPI.obs))
```

```r
    #
    # SPI.ref.cor[,Grids] <- sapply(Grids, function(i) cor(data.SPI.obs[,i],data.SPI.ref[,i]))
    # SPI.mod.cor[,Grids] <- sapply(Grids, function(i) cor(data.SPI.obs[,i],data.SPI.mod[,i]))
    #
    # print(sum(SPI.ref.cor<SPI.mod.cor,na.rm=T)/length(Grids)) # percentage of improvements of grids
    #
    # #xyplot - Correlation
    # par(mfrow=c(1,1),mar=c(3,4,2,2),mgp=c(1.6,0.6,0),ps=10, bg="transparent")
    # plot(SPI.ref.cor, SPI.mod.cor, xlim=c(-0.5,1), ylim=c(-0.5,1),
    #      xlab="Predicted", ylab="Predicted(VT)", pch=19,col="blue")
    # abline(a=0,b=1,col="grey")
    #
    # p7.list[[length(p7.list)+1]] <- recordPlot()

    SPI.PDF.unbiased <- rbind(SPI.PDF.unbiased, cbind(sc, t(SPI.ref.PDF), t(SPI.mod.PDF)))
}
#> [1] 1
#> [1] 1
#-----------------------------------------------
#combine subplots
cowplot::plot_grid(plotlist = p7.list, nrow=2, labels = c("(a)","(b)","(c)","(d)"), label_size = 12,
                   rel_widths = c(1, 1, 1, 1), hjust=0)
```

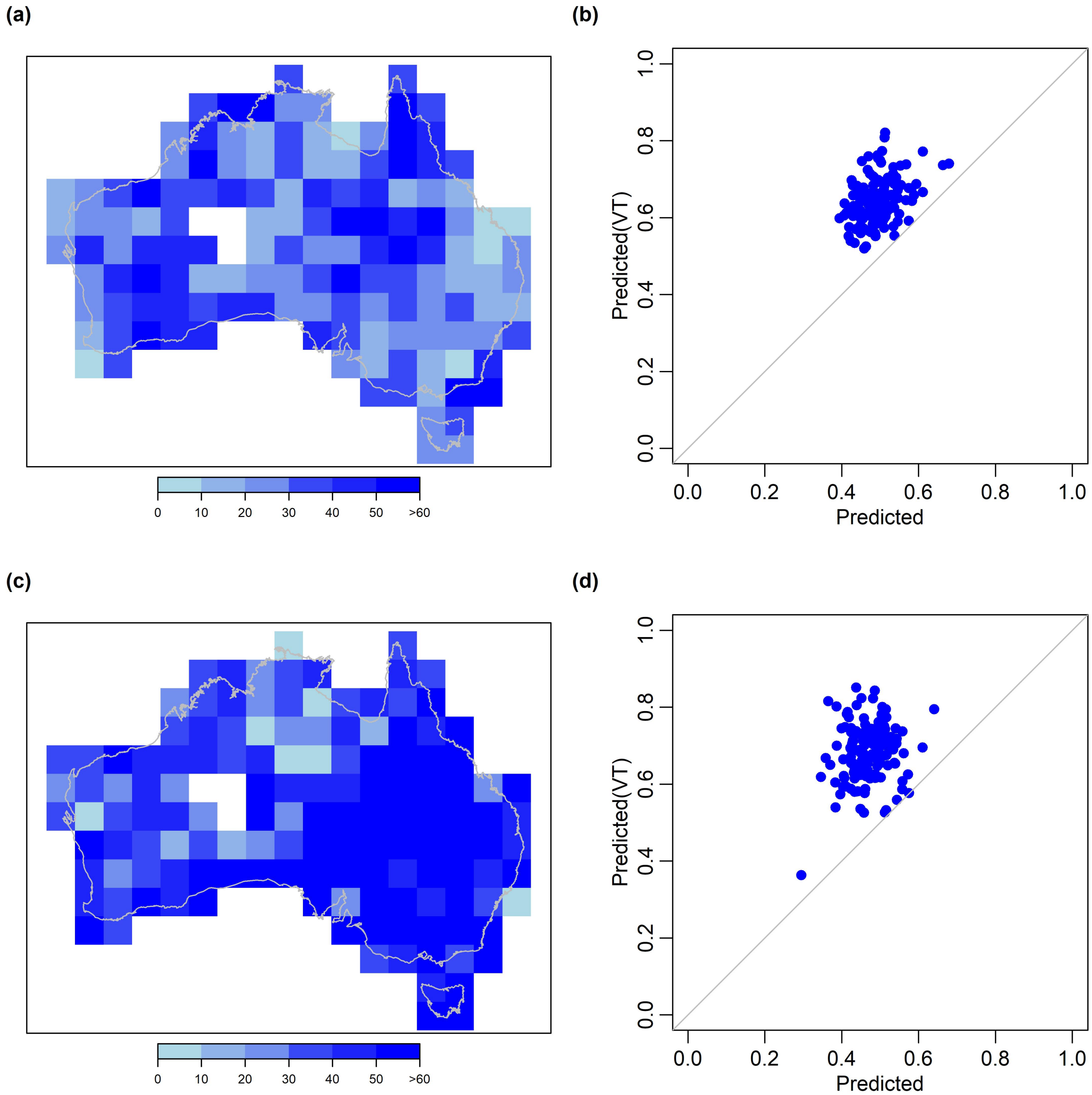

Figure S4: The percent improvement of PDF skill scores compared to the non-wavelet model using unbiased estimator

Reference

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962. doi:10.1029/2019wr026962

R session Info

```
sessionInfo()
#> R version 3.6.3 (2020-02-29)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 18363)
#>
#> Matrix products: default
```

```
#>
#> locale:
#> [1] LC_COLLATE=English_Australia.1252  LC_CTYPE=English_Australia.1252
#> [3] LC_MONETARY=English_Australia.1252 LC_NUMERIC=C
#> [5] LC_TIME=English_Australia.1252
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] tidyr_1.0.2        dplyr_0.8.3        plyr_1.8.5         overlapping_1.5.4
#>  [5] testthat_2.3.1     raster_3.0-12      sp_1.4-1           ggplot2_3.3.1
#>  [9] cowplot_1.0.0.9000 NPRED_1.0.3        WASP_1.3.1         devtools_2.2.1
#> [13] usethis_1.5.1      webshot_0.5.2      magick_2.3         bookdown_0.17
#> [17] kableExtra_1.1.0   knitr_1.28         rmarkdown_2.1
#>
#> loaded via a namespace (and not attached):
#>  [1] Rcpp_1.0.4.6       lattice_0.20-41    prettyunits_1.1.1  ps_1.3.0
#>  [5] assertthat_0.2.1   rprojroot_1.3-2    digest_0.6.23      R6_2.4.1
#>  [9] backports_1.1.5    evaluate_0.14      httr_1.4.1         pillar_1.4.3
#> [13] rlang_0.4.7        rstudioapi_0.11    callr_3.3.2        labeling_0.3
#> [17] desc_1.2.0         rgdal_1.4-8        readr_1.3.1        stringr_1.4.0
#> [21] munsell_0.5.0      compiler_3.6.3     xfun_0.12          gridGraphics_0.5-0
#> [25] pkgconfig_2.0.3    pkgbuild_1.0.6     htmltools_0.4.0    tidyselect_0.2.5
#> [29] tibble_3.0.3       codetools_0.2-16   fansi_0.4.1        viridisLite_0.3.0
#> [33] crayon_1.3.4       withr_2.1.2        grid_3.6.3         gtable_0.3.0
#> [37] lifecycle_0.2.0    magrittr_1.5       scales_1.1.0       cli_2.0.1
#> [41] stringi_1.4.5      farver_2.0.3       fs_1.3.1           remotes_2.1.0
#> [45] xml2_1.2.2         ellipsis_0.3.0     vctrs_0.3.2        tools_3.6.3
#> [49] glue_1.3.1         purrr_0.3.3        hms_0.5.2          processx_3.4.2
#> [53] pkgload_1.0.2      yaml_2.2.0         colorspace_1.4-1   sessioninfo_1.1.1
#> [57] rvest_0.3.5        waveslim_1.8.2     memoise_1.1.0
```