The treatment of uncertainties in global water models

R code

Arnald Puy

Contents

1 Preliminary functions **2**

2 Models under study **4**

3 Bibliometric analysis **4**

4 Keywords analysis: "uncertainty" and "sensitivity" **6**

5 Arrange the data **6**

6 Descriptive analysis **7**

7 Study of n-tokens **16**

8 Session information **22**

1 Preliminary functions

```
# PRELIMINARY FUNCTIONS #######################################################

# Function to read in all required packages in one go
loadPackages <- function(x) {
  for(i in x) {
    if(!require(i, character.only = TRUE)) {
      install.packages(i, dependencies = TRUE)
      library(i, character.only = TRUE)
    }
  }
}

# Load the packages
loadPackages(c(
  "bibliometrix", "tidyverse", "data.table", "scales", "pdfsearch", "pdftools",
  "openxlsx", "cowplot", "wesanderson", "sjmisc", "ggpubr", "tm", "syuzhet",
  "qdapRegex", "tidytext", "igraph", "ggraph", "benchmarkme", "parallel"))

# Create custom theme
theme_AP <- function() {
  theme_bw() +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          legend.background = element_rect(fill = "transparent",
                                           color = NA),
          legend.key = element_rect(fill = "transparent",
                                    color = NA),
          strip.background = element_rect(fill = "white"),
          legend.margin = margin(0.5, 0.1, 0.1, 0.1),
          legend.box.margin = margin(0.2,-2,-7,-7))
}

# Set checkpoint
dir.create(".checkpoint")
library("checkpoint")

checkpoint("2022-05-30",
           R.version ="4.2.0",
           checkpointLocation = getwd())
```

```
# FUNCTION TO CLEAN TEXT ######################################################

# Function to remove punctuation, citations, numbers, stopwords in english,
# bring to lowercase and strip whitespace, and especial characters, etc...
clear_text <- function(x) {
  y <- gsub("-", "", x)
```

```r
  y <- rm_citation(y)
  y <- tm::removePunctuation(y)
  y <- tm::removeNumbers(y)
  y <- tm::removeWords(y, stopwords::stopwords(language = "en"))
  y <- tolower(y)
  y <- str_replace_all(y, "[[:punct:]]", "") # Remove punctuation characters
  y <- y <- str_remove_all(y, "[^[\\da-zA-Z ]]")# Remove all non-alphanumerical
  y <- stemDocument(y) # Stem the document and keep only the root of the word
  y <- tm::stripWhitespace(y)
  y <- str_squish(y)
  y <- tm::removeWords(y, c(" et ", "al", "table", "figure", "fig",
                            "figs", "can", "eg", "mm", "yr",
                            "last", "access", "see", "section"))
  y <- gsub(" ?doi\\w+ ?", "", y) # Remove words that start with doi
  y <- str_replace(y, "http", "") # Remove https
  y <- tm::removeWords(y, stopwords::stopwords(language = "en"))
  y <- trimws(y) # Remove leading/trailing white space
  y <- tm::stripWhitespace(y)
  y <- gsub("\\s[A-Za-z](?= )", "", y, perl = TRUE) # Remove isolated letters
  y <- gsub("\\s[A-Za-z]$", "", y, perl = TRUE) # Remove isolated letters end of string
  y <- str_squish(y)

  return(y)
}
```

2 Models under study

```r
# VECTOR WITH NAME OF MODELS ###################################################

models <- c("WaterGAP", "PCR-GLOBWB", "MATSIRO", "H08", "JULES-W1", "MPI-HM",
            "MHM", "LPJmL", "CWatM", "CLM", "DBHM", "ORCHIDEE", "GR4J",
            "SACRAMENTO", "PRMS")

models_vec <- paste(models, "_ref.bib", sep = "")
```

3 Bibliometric analysis

```r
# BIBLIOMETRIC ANALYSIS ########################################################

output <- results <- years <- journals <- dt <- dt.clean <- list()

selected_cols <- c("title", "abstract", "keywords", "keywords.plus")

for (i in 1:length(models_vec)) {

  output[[i]] <- convert2df(file = models_vec[i],
                            dbsource = "wos",
                            format = "bibtex")

  # Extract title --------------------------------------------------------------

  title <- output[[i]]$TI

  # Extract Authors, Countries and Universities -------------------------------

  # Authors
  tmp.authors <- output[[i]]$AU
  first.author <- sub(" *\\;.*", "", tmp.authors)
  last.author <- sub(".*\\;","", tmp.authors)

  # First author affiliation and country
  country.first <- sub(".*\\,", "", output[[i]]$RP)
  university.first <- sub(" *\\;.*", "", output[[i]]$affiliations)

  # Last author affiliation and country
  last.affiliation <- sub(".*\\;", "", output[[i]]$C1)
  country.last <- sub("\\.", "", sub(".*\\, ", "", last.affiliation))
  university.last <- sub(".*\\;", "", output[[i]]$affiliations)

  # Extract keywords ----------------------------------------------------------
  keywords <- gsub(";;", ";", output[[i]]$DE)
  keywords.plus <- gsub(";;", ";", output[[i]]$ID)

  # Create data.table ----------------------------------------------------------

  dt[[i]] <- data.table("WOS" = output[[i]]$UT,
                        "title" = title,
                        "year" = output[[i]]$PY,
                        "keywords" = keywords,
                        "keywords.plus" = keywords.plus,
                        "first.author" = first.author,
                        "last.author" = last.author,
                        "country.first" = country.first,
                        "country.last" = country.last,
                        "university.first" = university.first,
                        "university.last" = university.last,
                        "abstract" = output[[i]]$AB)

  dt.clean[[i]] <- copy(dt[[i]])

  dt.clean[[i]][, (selected_cols):= lapply(.SD, function(x)
    clear_text(x)), .SDcols = selected_cols]

  # Export data dirty and clean
  write.xlsx(dt[[i]], file = paste(models[i], "_bibliometric.xlsx", sep = ""))
  write.xlsx(dt.clean[[i]], file = paste(models[i], "_bibliometric_clean.xlsx", sep = ""))

  # Retrieve analysis bibliometrix ---------------------------------------------

  results[[i]] <- biblioAnalysis(output[[i]], sep = ";")
  years[[i]] <- data.table(results[[i]]$Years)
  journals[[i]] <- data.table(results[[i]]$Sources) %>%
    .[, SO:= str_to_title(SO)]
}

# Fill out affiliations erroneously labelled as NA -----------------------------

# Watergap (1)
for(i in c(1, 4, 5)) {
  output[[1]]$affiliations[[i]] <- "UNIVERSITAT KASSEL"
}


# Add names of models ----------------------------------------------------------
names(years) <- models
names(journals) <- models
names(dt.clean) <- models
names(dt.clean) <- models
```

4 Keywords analysis: “uncertainty” and “sensitivity”

```
# KEYWORDS ANALYSIS ############################################################

# Define vectors for search ---------------------------------------------------
directory <- "/Users/arnaldpuy/Documents/papers/ghms_bibliometric/"
directory_vec <- paste(directory, models, "_pdfs", sep = "")
filename_keywords <- paste(models, "keywords", sep = "_")

# Define vectors with keywords ------------------------------------------------
keywords_vec <- c("uncertainty", "sensitivity")
keywords_vec_stemmed <- stemDocument(keywords_vec)

# Loop -------------------------------------------------------------------------
dt.keyword <- dt.keyword.clean <- output <- list()
for (i in 1:length(directory_vec)) {

  output[[i]] <- keyword_directory(directory_vec[i],
                                   keyword = keywords_vec_stemmed,
                                   split_pdf = TRUE)

  dt.keyword[[i]] <- data.table("name" = output[[i]]$pdf_name,
                                "keyword" = output[[i]]$keyword,
                                "text" = output[[i]]$line_text)

  dt.keyword.clean[[i]] <- copy(dt.keyword[[i]])

  # Clean the text where the keywords are located
  dt.keyword.clean[[i]] <- dt.keyword.clean[[i]][, text:= clear_text(text)]

  # Write dirty and clean data
  fwrite(dt.keyword[[i]], file = paste(filename_keywords[i], ".csv", sep = ""))
  fwrite(dt.keyword.clean[[i]], file = paste(filename_keywords[i], "_clean.csv", sep = ""))
}

names(output) <- models
names(dt.keyword) <- models
names(dt.keyword.clean) <- models
```

5 Arrange the data

```
# ARRANGE DATA #################################################################
# Bibliometric analysis -------------------------

# Correct for USA and China
colsName <- c("country.first", "country.last")
full.dt <- rbindlist(dt.clean, idcol = "Model") %>%
  .[, (colsName):= lapply(.SD, function(x)
    ifelse(grepl("USA", x), "USA", x)), .SDcols = colsName] %>%
  .[, (colsName):= lapply(.SD, function(x)
    ifelse(grepl("CHINA", x), "CHINA", x)), .SDcols = colsName]

# Export
fwrite(full.dt, "full.dt.csv")

# Keywords analysis -----------------------------
full.keyword.dt <- rbindlist(dt.keyword.clean, idcol = "Model")

# Export
fwrite(full.keyword.dt, "full.keyword.dt.csv")
```

6 Descriptive analysis

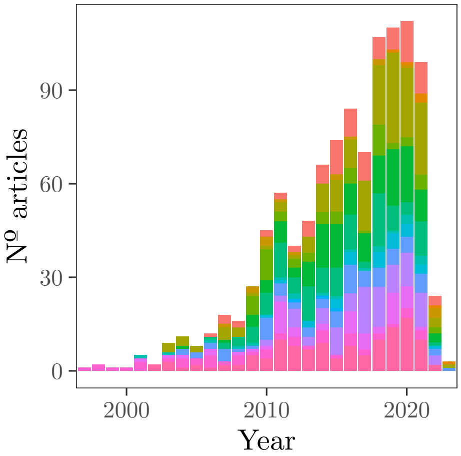
```
# DESCRIPTIVE STUDY ############################################################

# Total number of studies
total.n <- full.dt[, .(Model, WOS)] %>%
  .[, .(total.papers = .N), Model] %>%
  .[order(-total.papers)]

total.n
```

```
##          Model total.papers
##  1:       GR4J          167
##  2:   JULES-W1          136
##  3:   WaterGAP          126
##  4:      LPJmL          116
##  5: PCR-GLOBWB           95
##  6:        CLM           92
##  7:   ORCHIDEE           75
##  8:        H08           61
##  9:       PRMS           55
## 10: SACRAMENTO           52
## 11:        MHM           29
## 12:    MATSIRO           21
## 13:       DBHM           17
## 14:      CWatM            7
## 15:     MPI-HM            3
```

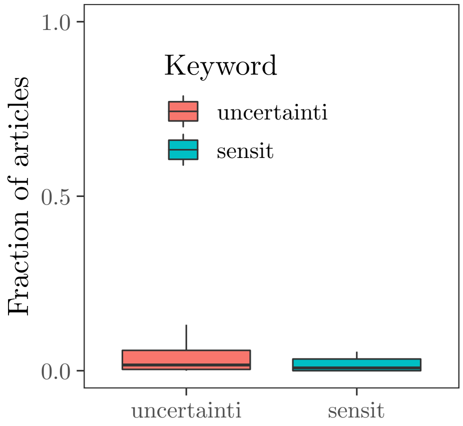
```
sum(total.n$total.papers)
```

```
## [1] 1052
```

```
# NUMBER OF STUDIES THROUGHT TIME ##############################################

plot.time <- rbindlist(years, idcol = "Model")[, .N, .(V1, Model)] %>%
  .[, V1:= as.factor(V1)] %>%
  ggplot(., aes(V1, N, fill = Model)) +
  geom_col() +
  scale_x_discrete(breaks = pretty_breaks(n = 3)) +
  labs(x = "Year", y = "Nº articles") +
  theme_AP() +
  theme(legend.position = "none")

plot.time
```



```
# FRACTION OF STUDIES WITH UNCERTAINTI AND SENSIT* AS KEYWORDS ################

full.dt <- full.dt[, `:=` (uncertainti = str_detect(keywords, keywords_vec_stemmed[1]),
                            sensit = str_detect(keywords, keywords_vec_stemmed[2]))]

plot.n.keywords <- full.dt[, lapply(.SD, function(x)
  sum(x) / .N), .SDcols = (keywords_vec_stemmed), Model] %>%
  melt(., measure.vars = keywords_vec_stemmed) %>%
  ggplot(., aes(variable, value, fill = variable)) +
  geom_boxplot() +
  labs(y = "Fraction of articles", x = "") +
  scale_y_continuous(breaks = pretty_breaks(n = 3),
                     limits = c(0, 1)) +
  scale_fill_discrete(name = "Keyword") +
  theme_AP() +
  theme(legend.position = c(0.45, 0.75))

plot.n.keywords
```

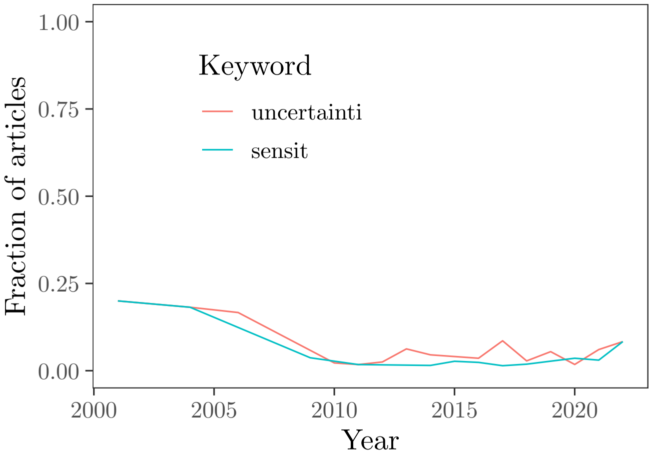


```
# Fraction of studies with both keywords in the abstract
full.dt[uncertainti == "TRUE" & sensit == "TRUE", .N] / full.dt[, .N]
```

```
## [1] 0.002851711
```

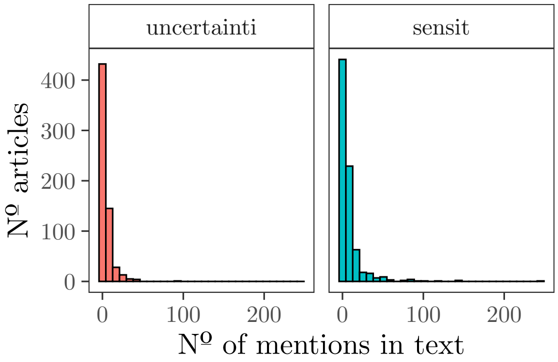
```
# FRACTION OF STUDIES WITH KEYWORDS THROUGH TIME ###############################

total.n.year <- rbindlist(years, idcol = "Model") %>%
  .[, .(total.n = .N), V1] %>%
  setnames(., "V1", "year")

plot.fraction.years <- full.dt[, .(WOS, uncertainti, sensit, year)] %>%
  melt(., measure.var = keywords_vec_stemmed) %>%
  .[value == TRUE, .N, .(year, variable)] %>%
  merge(., total.n.year, by = "year") %>%
  .[, fraction:= N / total.n] %>%
  ggplot(., aes(year, fraction, color = variable, group = variable)) +
  geom_line() +
  scale_color_discrete(name = "Keyword") +
  scale_y_continuous(limits = c(0, 1)) +
  labs(x = "Year", y = "Fraction of articles") +
  theme_AP() +
  theme(legend.position = c(0.35, 0.75))

plot.fraction.years
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



```
# PLOT HISTOGRAM ##############################################################

dist.plot <- full.keyword.dt[, .N, .(name, keyword)] %>%
  .[, keyword:= factor(keyword, levels = keywords_vec_stemmed)] %>%
  ggplot(., aes(N, fill = keyword)) +
  geom_histogram(color = "black") +
  labs(x = "Nº of mentions in text", y = "Nº articles") +
  facet_wrap(~keyword) +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  theme_AP() +
  theme(legend.position = "none")

dist.plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# How many articles have 1, 2, ... N mentions of uncertainti and sensit in the
# main text ---------------------------------------------------
full.keyword.dt[, .N, .(name, keyword)] %>%
  .[, .(number.articles = .N), .(N, keyword)] %>%
  .[.[order(N),.I[1:.N],"keyword"]$V1]
```

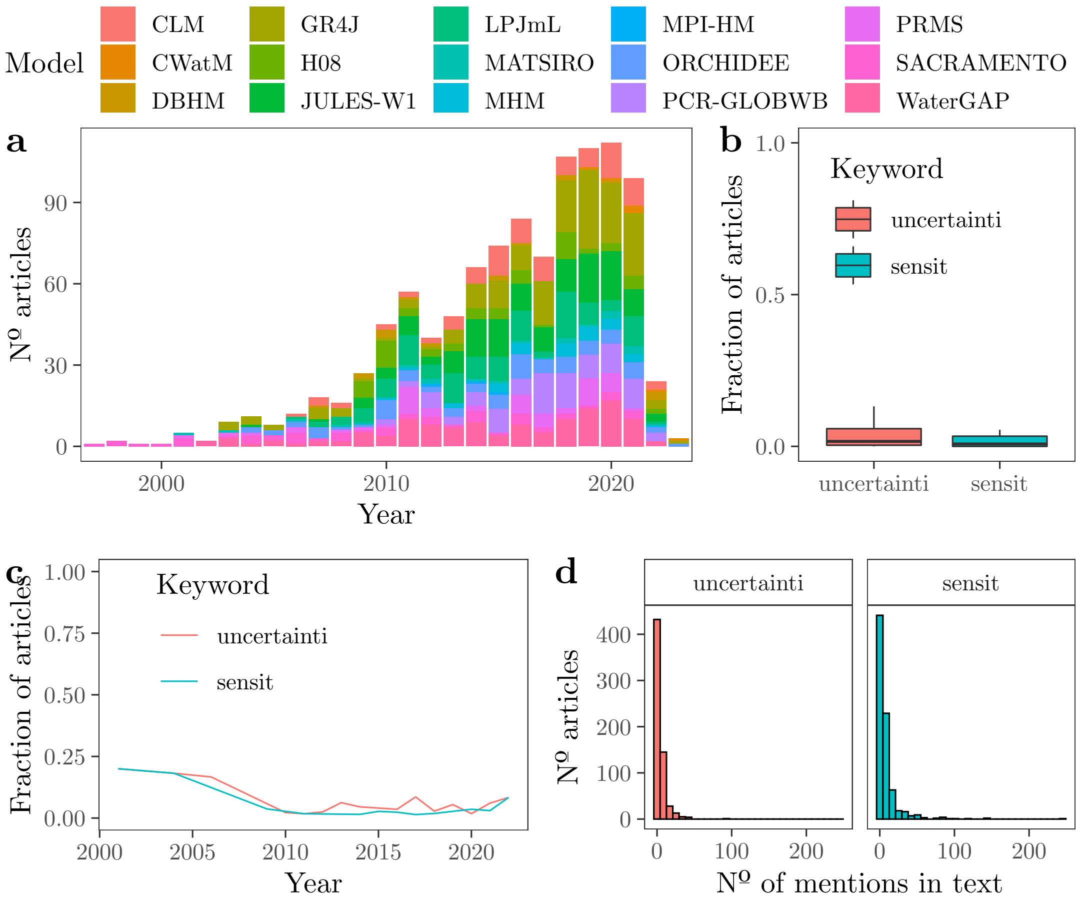
```
##        N     keyword number.articles
##   1:   1      sensit             157
##   2:   2      sensit             117
##  3:  3      sensit              96
##  4:  4      sensit              71
##  5:  5      sensit              58
##  6:  6      sensit              42
##  7:  7      sensit              39
##  8:  8      sensit              22
##  9:  9      sensit              25
## 10: 10      sensit              15
## 11: 11      sensit               9
## 12: 12      sensit              19
## 13: 13      sensit               9
## 14: 14      sensit              10
## 15: 15      sensit              10
## 16: 16      sensit               3
## 17: 17      sensit              10
## 18: 18      sensit               5
## 19: 19      sensit               4
## 20: 20      sensit               6
## 21: 21      sensit               6
## 22: 22      sensit               2
## 23: 23      sensit               4
## 24: 24      sensit               3
## 25: 25      sensit               5
## 26: 26      sensit               4
## 27: 30      sensit               3
## 28: 32      sensit               4
## 29: 33      sensit               1
## 30: 34      sensit               3
## 31: 35      sensit               1
## 32: 36      sensit               1
## 33: 37      sensit               1
## 34: 38      sensit               2
## 35: 39      sensit               1
## 36: 41      sensit               2
## 37: 43      sensit               1
## 38: 44      sensit               1
## 39: 45      sensit               1
## 40: 46      sensit               1
## 41: 47      sensit               1
## 42: 48      sensit               3
## 43: 51      sensit               1
## 44: 53      sensit               2
## 45: 54      sensit               2
## 46: 55      sensit               1
## 47: 59      sensit               1
## 48: 60      sensit               1
## 49: 78      sensit               1
## 50: 80      sensit               1
## 51:  81      sensit               1
## 52:  85      sensit               1
## 53:  86      sensit               1
## 54:  88      sensit               1
## 55:  95      sensit               1
## 56: 104      sensit               1
## 57: 119      sensit               1
## 58: 142      sensit               1
## 59: 143      sensit               1
## 60: 246      sensit               1
## 61:   1 uncertainti             187
## 62:   2 uncertainti             111
## 63:   3 uncertainti              78
## 64:   4 uncertainti              56
## 65:   5 uncertainti              36
## 66:   6 uncertainti              28
## 67:   7 uncertainti              19
## 68:   8 uncertainti              15
## 69:   9 uncertainti              15
## 70:  10 uncertainti              14
## 71:  11 uncertainti              10
## 72:  12 uncertainti               8
## 73:  13 uncertainti               6
## 74:  14 uncertainti               6
## 75:  15 uncertainti               1
## 76:  16 uncertainti               3
## 77:  17 uncertainti               3
## 78:  18 uncertainti               2
## 79:  19 uncertainti               1
## 80:  20 uncertainti               3
## 81:  21 uncertainti               3
## 82:  22 uncertainti               2
## 83:  23 uncertainti               2
## 84:  24 uncertainti               2
## 85:  25 uncertainti               2
## 86:  27 uncertainti               4
## 87:  28 uncertainti               1
## 88:  30 uncertainti               1
## 89:  32 uncertainti               3
## 90:  33 uncertainti               1
## 91:  39 uncertainti               2
## 92:  41 uncertainti               1
## 93:  46 uncertainti               1
## 94:  93 uncertainti               1
##       N     keyword number.articles
```

```
# MERGE DESCRIPTIVE PLOTS ######################################################
```

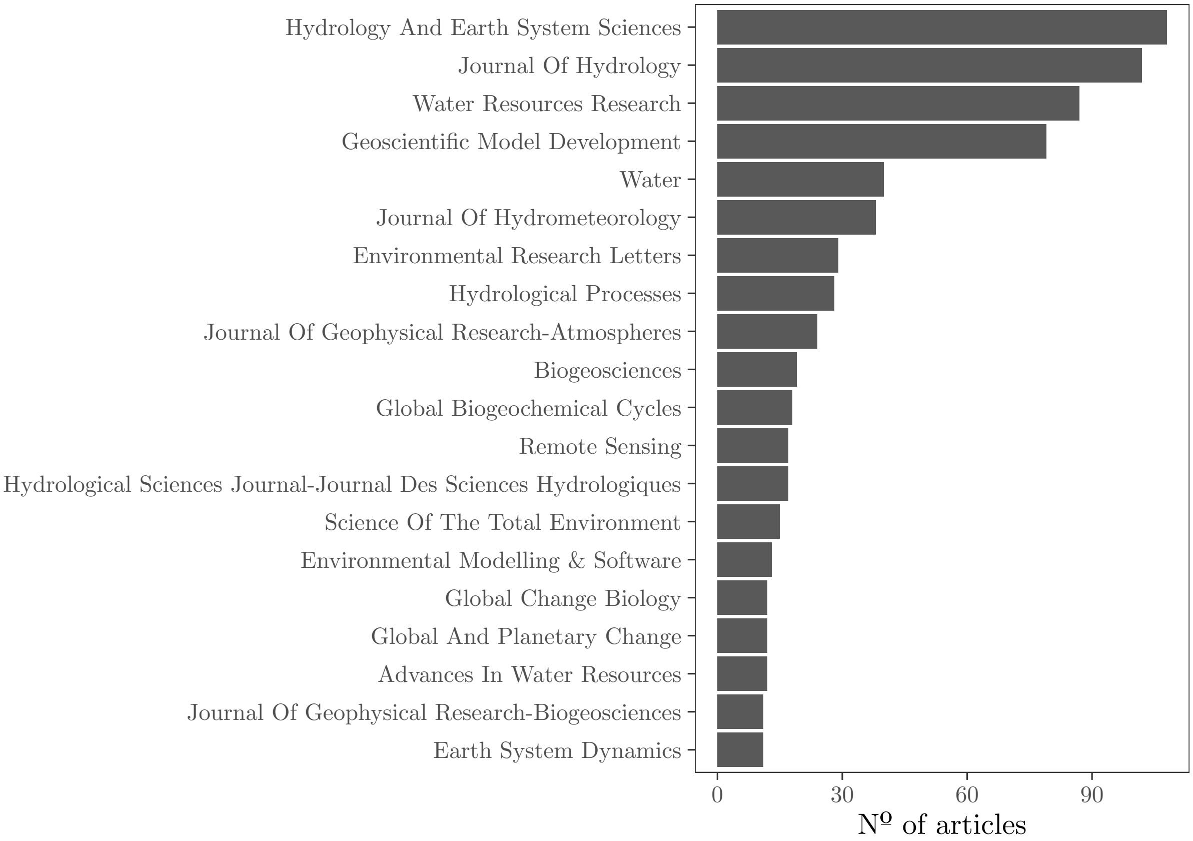
```
legend <- get_legend(plot.time + theme(legend.position = "top"))
top <- plot_grid(plot.time, plot.n.keywords, ncol = 2, labels = "auto",
                 rel_widths = c(0.65, 0.35))
all <- plot_grid(legend, top, ncol = 1, rel_heights = c(0.22, 0.78))
bottom <- plot_grid(plot.fraction.years, dist.plot, ncol = 2, labels = c("c", "d"),
                    rel_widths = c(0.5, 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
plot_grid(all, bottom,
          ncol = 1, labels = "", rel_heights = c(0.6, 0.4))
```



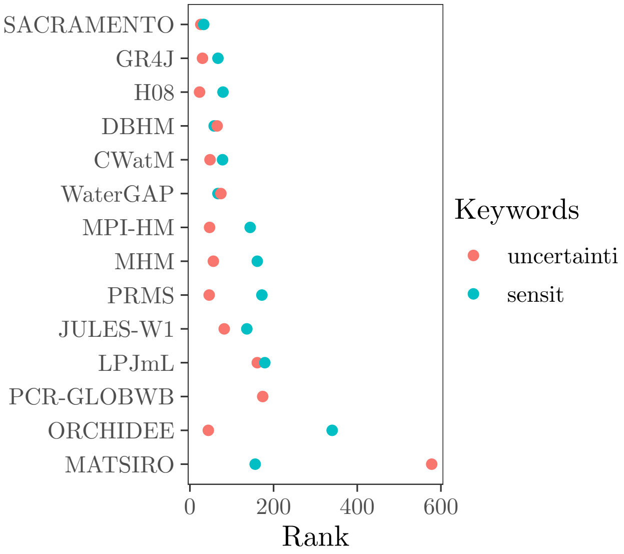
```
# PLOT JOURNALS ################################################################

rbindlist(journals, idcol = "Model") %>%
  .[, sum(N), SO] %>%
  .[order(-V1)] %>%
  .[1:20] %>%
  na.omit() %>%
  ggplot(., aes(reorder(SO, V1, sum), V1)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(x = "", y = "Nº of articles") +
  theme_AP()
```



```
# WORDCLOUD OF WORDS IN ABSTRACT ###############################################

tmp <- split(full.dt, full.dt$Model)
names(tmp) <- models

out <- dtm <- m <- v <- word.count <- list()
for (i in names(tmp)) {
  out[[i]] <- Corpus(VectorSource(tmp[[i]]$abstract))
  dtm[[i]] <- tm::TermDocumentMatrix(out[[i]])
  m[[i]] <- as.matrix(dtm[[i]])
  v[[i]] <- sort(rowSums(m[[i]]), decreasing=TRUE)
  word.count[[i]] <- data.table(word = names(v[[i]]), freq = v[[i]])
}

word.count.dt <- rbindlist(word.count, idcol = "Model")

# Plot wordcloud -------------------------------------------

plots.wordcloud <- list()

for(i in names(word.count)) {
  plots.wordcloud[[i]] <- word.count.dt[Model == i] %>%
    .[1:50] %>%
    ggplot(., aes(label = word, size = freq)) +
    ggwordcloud::geom_text_wordcloud_area(eccentricity = 1, shape = "square") +
    scale_size_area(max_size = 10) +
    theme_AP() +
    ggtitle(names(word.count[i]))
}

# Check rank of the terms "uncertainty" and "sensitivity" in the abstract -----

word.count.dt[, rank:= frank(-freq, ties.method = "first"), Model]

rank.keywords <- word.count.dt[word %chin% keywords_vec_stemmed] %>%
  merge(., total.n, by = "Model") %>%
  .[, word:= factor(word, levels = keywords_vec_stemmed)]

rank.keywords[order(word, rank)]
```

```
##          Model        word freq rank total.papers
##  1:        H08 uncertainti  137   23           61
##  2: SACRAMENTO uncertainti   37   26           52
##  3:       GR4J uncertainti   37   30          167
##  4:   ORCHIDEE uncertainti   52   44           75
##  5:       PRMS uncertainti   64   46           55
##  6:     MPI-HM uncertainti   68   47            3
##  7:      CWatM uncertainti   17   48            7
##  8:        MHM uncertainti   52   56           29
##  9:       DBHM uncertainti   34   65           17
## 10:   WaterGAP uncertainti   33   74          126
## 11:   JULES-W1 uncertainti   17   82          136
## 12:      LPJmL uncertainti    5  161          116
## 13: PCR-GLOBWB uncertainti    2  174           95
## 14:    MATSIRO uncertainti    1  578           21
## 15: SACRAMENTO      sensit   32   33           52
## 16:       DBHM      sensit   38   58           17
## 17:       GR4J      sensit   24   67          167
## 18:   WaterGAP      sensit   35   67          126
## 19:      CWatM      sensit   12   78            7
## 20:        H08      sensit   53   79           61
## 21:   JULES-W1      sensit   12  136          136
## 22:     MPI-HM      sensit   31  144            3
## 23:    MATSIRO      sensit    4  156           21
## 24:        MHM      sensit   26  161           29
## 25:       PRMS      sensit   26  172           55
## 26:      LPJmL      sensit    4  179          116
## 27:   ORCHIDEE      sensit   11  340           75
##          Model        word freq rank total.papers
```

```r
# Plot-------------------------------------------

ggplot(rank.keywords, aes(reorder(Model, -rank), rank, color = word)) +
  geom_point() +
  coord_flip() +
  labs(x = "", y = "Rank") +
  scale_color_discrete(name = "Keywords") +
  theme_AP() +
  theme(legend.position = "right")
```

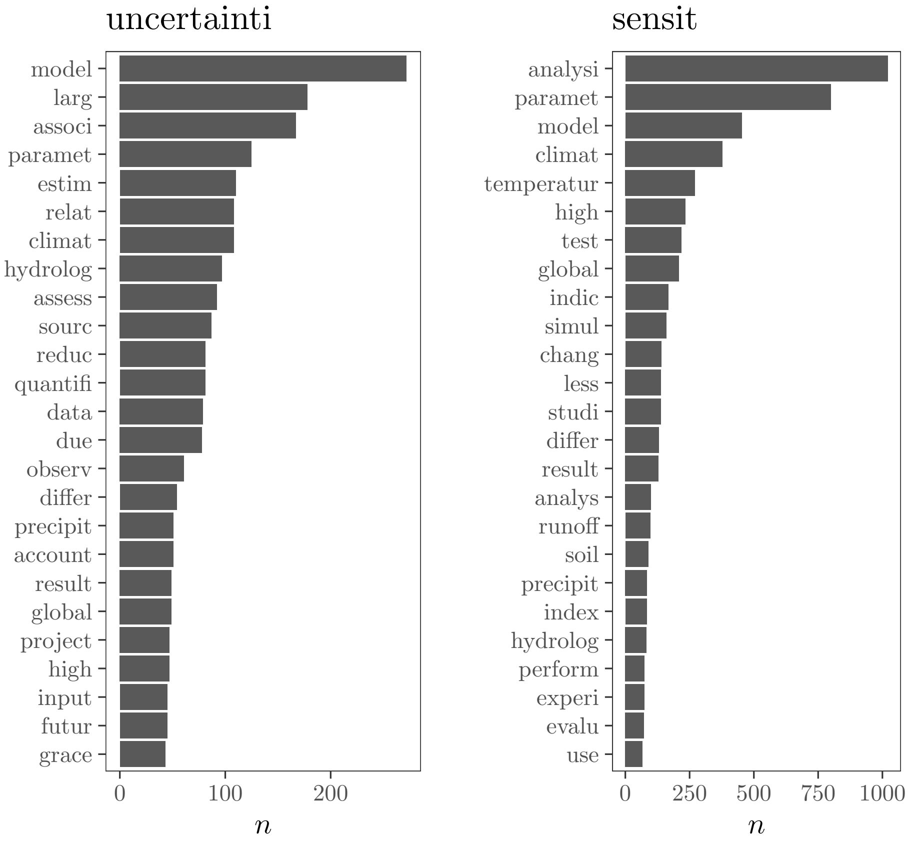


7 Study of n-tokens

```r
# STUDY OF N-TOKENS ###########################################################

# Number of tokens ------------------------------
N.tokens <- 2

# For loop --------------------------------------
output <- token.analysis <- vec <- plot.token <-
  plot.token.model <- graph_plot <- list()

for (i in 1:length(keywords_vec)) {

  output[[i]] <- full.keyword.dt %>%
    .[keyword == keywords_vec_stemmed[i]]

  # Token analysis ------------------------------------------------------------

  token.analysis[[i]] <- output[[i]] %>%
    unnest_tokens(bigram, text, token = "ngrams", n = N.tokens) %>%
    separate(bigram, into = c("word1", "word2"), sep = " ") %>%
    # We count the co-occurences of words without taking into account their order
    # within the n-token
    .[, `:=`(word1= pmin(word1, word2), word2 = pmax(word1, word2))] %>%
    count(word1, word2, Model, sort = TRUE) %>%
    unite(., col = "bigram", c("word1", "word2"), sep = " ")

  vec[[i]] <- token.analysis[[i]] %>%
    .[, str_detect(bigram, keywords_vec_stemmed[i])]

  plot.token[[i]] <-   token.analysis[[i]][vec[[i]]] %>%
    .[, sum(n), bigram] %>%
    .[order(-V1)] %>%
    .[, head(.SD, 25)] %>%
    .[, bigram:= str_remove(bigram, keywords_vec_stemmed[i])] %>%
    ggplot(., aes(reorder(bigram, V1, sum), V1)) +
    geom_bar(stat = "identity") +
    coord_flip() +
    theme_AP() +
    labs(y = "$n$", x = "") +
    theme(legend.position = "none") +
    ggtitle(keywords_vec_stemmed[i])

  plot.token.model[[i]] <- token.analysis[[i]][vec[[i]]] %>%
    .[, head(.SD, 5), Model] %>%
    .[, `:=` (bigram = str_remove(bigram, keywords_vec_stemmed[i]),
              Model = as.factor(Model))] %>%
    .[, bigram:= reorder_within(bigram, n, Model)] %>%
    ggplot(., aes(reorder(bigram, n, sum), n, fill = Model)) +
    geom_bar(stat = "identity") +
    coord_flip() +
    theme_AP() +
    labs(y = "$n$", x = "") +
    scale_x_reordered() +
    theme(legend.position = "none") +
    ggtitle(keywords_vec_stemmed[i]) +
    facet_wrap(~Model, scales = "free", ncol = 3)

  # Graph analysis -------------------------------------------------------------

  bigram_graph <- token.analysis[[i]] %>%
    separate(., col = "bigram", into = c("word1", "word2"), sep = " ") %>%
    .[n > 20] %>%
    graph_from_data_frame()

  set.seed(666)
```

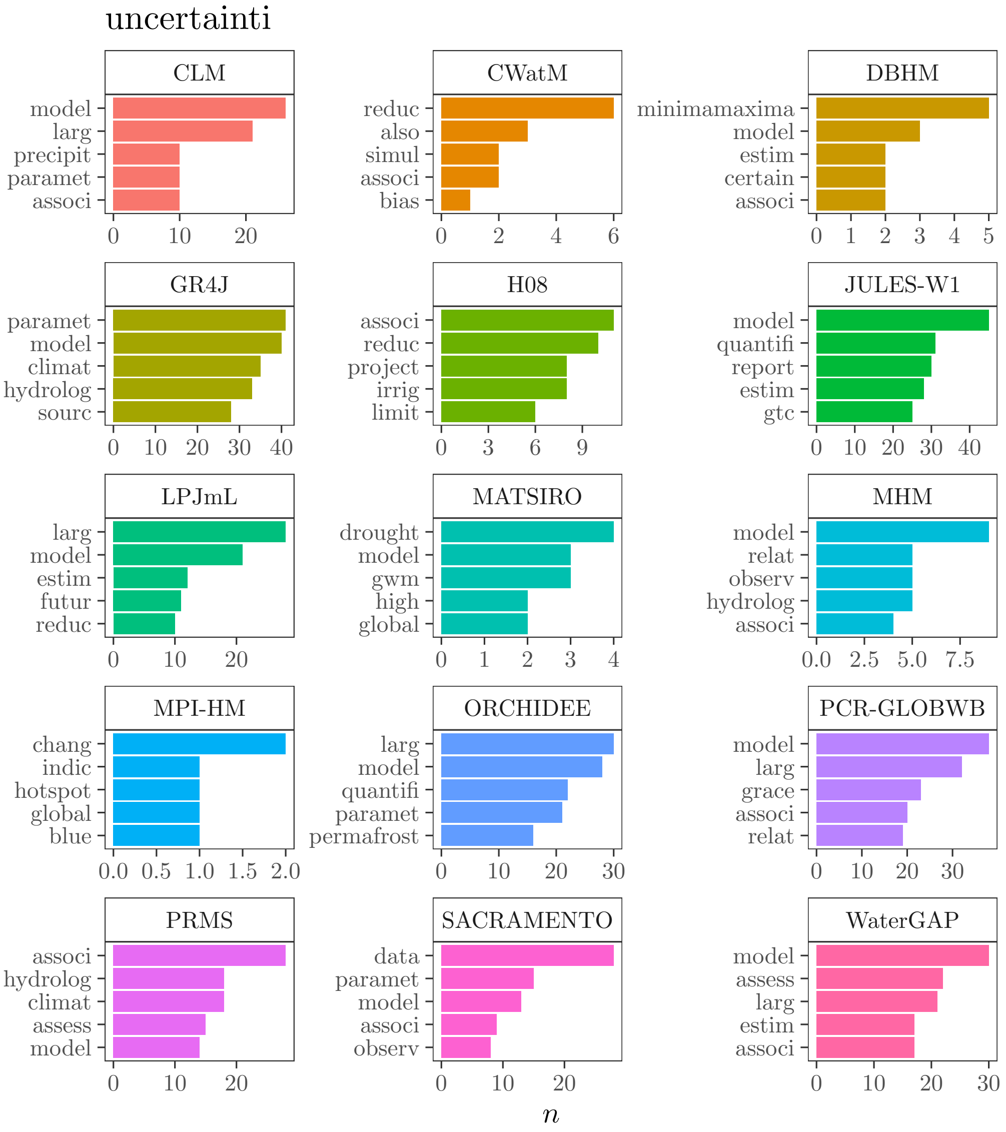
```
  a <- grid::arrow(type = "closed", length = unit(.08, "inches"))

  graph_plot[[i]] <- ggraph(bigram_graph, layout = "fr") +
    geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                   arrow = a, end_cap = circle(.02, 'inches')) +
    geom_node_point(color = "lightblue", size = 1.2) +
    geom_node_text(aes(label = name), vjust = 1, hjust = 1, size = 2.5) +
    labs(x = "", y = "") +
    theme_AP() +
    ggtitle(keywords_vec_stemmed[i])
}
```

```
# PLOT 25 MOST COMMON BIGRAMS ###################################################

plot_grid(plotlist = plot.token, ncol = 2, labels = "")
```

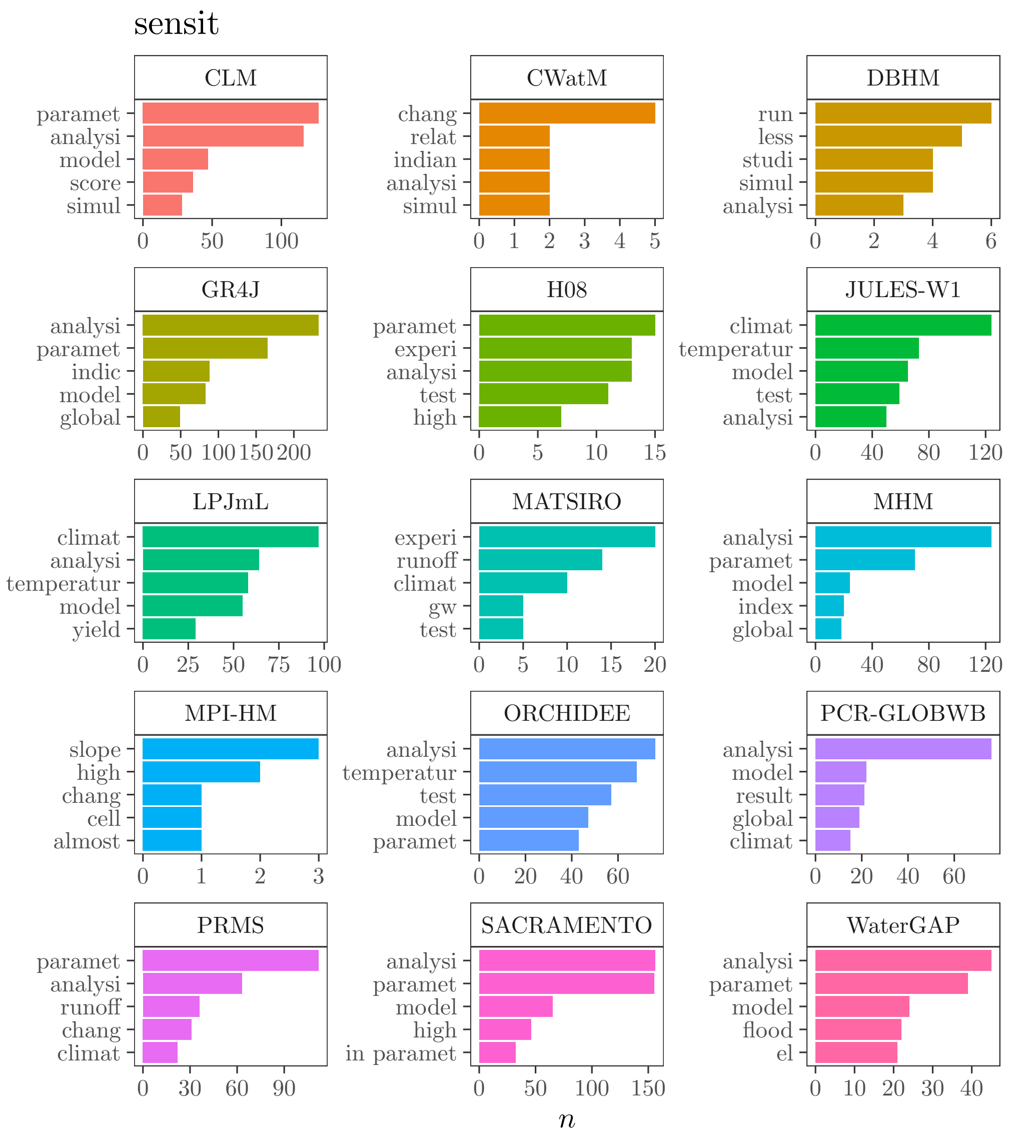


```
# PLOT 5 MOST BIGRAMS ###########################################################

sapply(1:2, function(i) plot.token.model[i])
```

```
## [[1]]
```

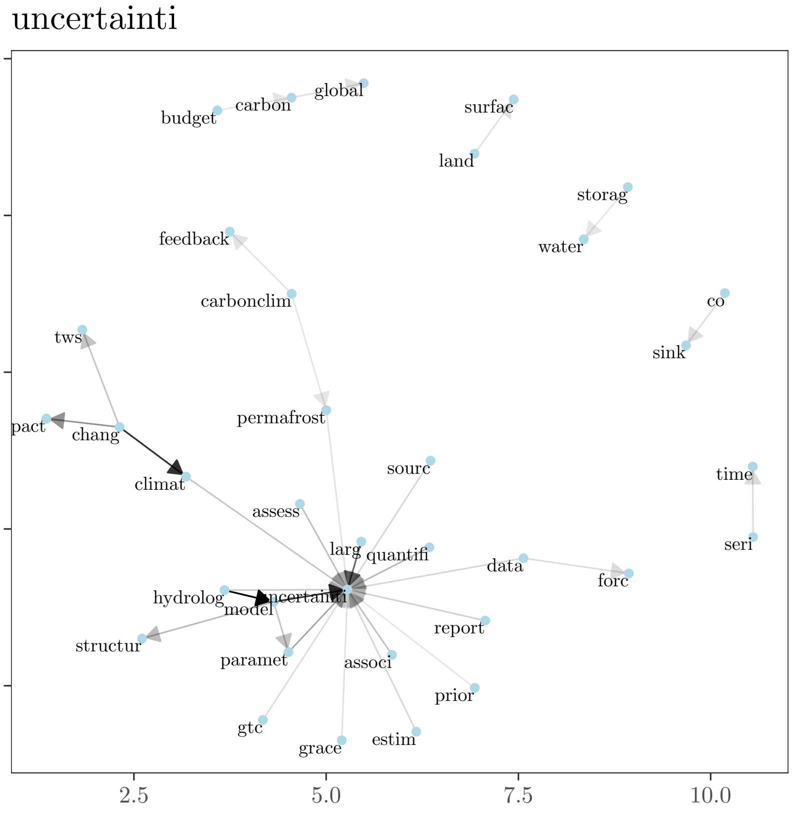

uncertainti
CLM
model
larg
precipit
paramet
associ
0 10 20
CWatM
reduc
also
simul
associ
bias
0 2 4 6
DBHM
minimamaxima
model
estim
certain
associ
0 1 2 3 4 5
GR4J
paramet
model
climat
hydrolog
sourc
0 10 20 30 40
H08
associ
reduc
project
irrig
limit
0 3 6 9
JULES-W1
model
quantifi
report
estim
gtc
0 10 20 30 40
LPJmL
larg
model
estim
futur
reduc
0 10 20
MATSIRO
drought
model
gwm
high
global
0 1 2 3 4
MHM
model
relat
observ
hydrolog
associ
0.0 2.5 5.0 7.5
MPI-HM
chang
indic
hotspot
global
blue
0.0 0.5 1.0 1.5 2.0
ORCHIDEE
larg
model
quantifi
paramet
permafrost
0 10 20 30
PCR-GLOBWB
model
larg
grace
associ
relat
0 10 20 30
PRMS
associ
hydrolog
climat
assess
model
0 10 20
SACRAMENTO
data
paramet
model
associ
observ
0 10 20
WaterGAP
model
assess
larg
estim
associ
0 10 20 30
n

```
##
## [[2]]
```

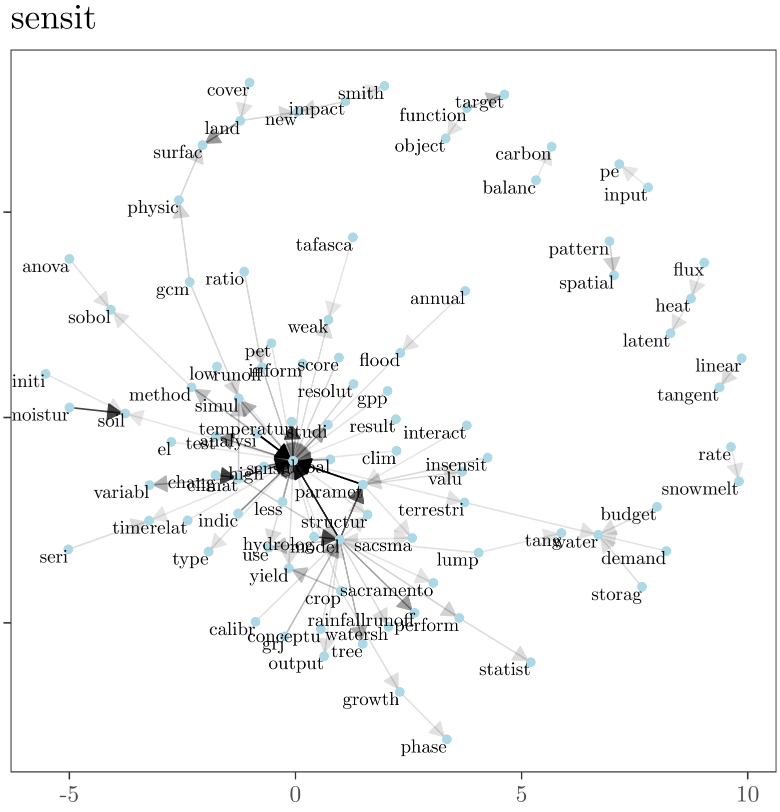


```
# PLOT NETWORKS OF BIGRAMS #####################################################

sapply(1:2, function(i) graph_plot[i])
```

```
## [[1]]
```


uncertainti
budget
carbon
global
surfac
land
storag
water
feedback
carbonclim
co
sink
tws
pact
chang
permafrost
climat
sourc
time
assess
larg
quantifi
data
seri
forc
hydrolog
uncertainti
model
report
structur
paramet
associ
prior
gtc
grace
estim
8
6
4
2
0
2.5
5.0
7.5
10.0

```
##
## [[2]]
```


sensit
cover
smith
impact
target
new
function
land
object
surfac
carbon
balanc
pe
input
physic
tafasca
anova
pattern
ratio
flux
gcm
annual
spatial
heat
sobol
weak
latent
pet
flood
low
runoff
inform
score
linear
initi
method
resolut
gpp
tangent
simul
moistur
soil
temperatur
studi
result
interact
el
test
analysi
clim
rate
insensit
sensit
global
valu
variabl
chang
climat
high
paramet
snowmelt
less
terrestri
budget
timerelat
indic
structur
seri
type
hydrolog
use
model
sacsma
tang
water
lump
demand
yield
sacramento
storag
crop
calibr
rainfallrunoff
perform
conceptu
watersh
grj
tree
output
statist
growth
phase
10
5
0
-5
0
5
10

8 Session information

```
# SESSION INFORMATION ##########################################################

sessionInfo()
```

```
## R version 4.2.0 (2022-04-22)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] benchmarkme_1.0.7  ggraph_2.0.5       igraph_1.3.1       tidytext_0.3.3
##  [5] qdapRegex_0.7.5    syuzhet_1.0.6      tm_0.7-8           NLP_0.2-1
##  [9] ggpubr_0.4.0       sjmisc_2.8.9       checkpoint_1.0.2   wesanderson_0.3.6
## [13] cowplot_1.1.1      openxlsx_4.2.5     pdftools_3.2.0     pdfsearch_0.3.0
## [17] scales_1.2.0       data.table_1.14.2  forcats_0.5.1      stringr_1.4.0
## [21] dplyr_1.0.9        purrr_0.3.4        readr_2.1.2        tidyr_1.2.0
## [25] tibble_3.1.7       ggplot2_3.3.6      tidyverse_1.3.1    bibliometrix_3.2.1
##
## loaded via a namespace (and not attached):
##   [1] readxl_1.4.0             backports_1.4.1        plyr_1.8.7
##   [4] lazyeval_0.2.2           SnowballC_0.7.0        digest_0.6.29
##   [7] foreach_1.5.2            htmltools_0.5.2        viridis_0.6.2
##  [10] fansi_1.0.3              magrittr_2.0.3         cluster_2.1.3
##  [13] doParallel_1.0.17        tzdb_0.3.0             graphlayouts_0.8.0
##  [16] modelr_0.1.8             tikzDevice_0.12.3.1    vroom_1.5.7
##  [19] askpass_1.1              colorspace_2.0-3       rvest_1.0.2
##  [22] ggrepel_0.9.1            pubmedR_0.0.3          haven_2.5.0
##  [25] xfun_0.31                crayon_1.5.1           jsonlite_1.8.0
##  [28] bibliometrixData_0.3.0 iterators_1.0.14       glue_1.6.2
##  [31] polyclip_1.10-0          stopwords_2.3          gtable_0.3.0
##  [34] car_3.0-13               rentrez_1.2.3          abind_1.4-5
##  [37] qpdf_1.1                 DBI_1.1.2              rstatix_0.7.0
##  [40] Rcpp_1.0.8.3             viridisLite_0.4.0      xtable_1.8-4
##  [43] dimensionsR_0.0.3        bit_4.0.4              flashClust_1.01-2
##  [46] DT_0.23                  htmlwidgets_1.5.4      httr_1.4.3
##  [49] ggwordcloud_0.5.0        RColorBrewer_1.1-3     ellipsis_0.3.2
##  [52] factoextra_1.0.7       pkgconfig_2.0.3        XML_3.99-0.9
##  [55] farver_2.1.0           dbplyr_2.1.1           utf8_1.2.2
##  [58] labeling_0.4.2         tidyselect_1.1.2       rlang_1.0.2
##  [61] later_1.3.0            munsell_0.5.0          cellranger_1.1.0
##  [64] tools_4.2.0            cli_3.3.0              generics_0.1.2
##  [67] sjlabelled_1.2.0       broom_0.8.0            evaluate_0.15
##  [70] fastmap_1.1.0          yaml_2.3.5             bit64_4.0.5
##  [73] knitr_1.39             fs_1.5.2               filehash_2.4-3
##  [76] tidygraph_1.2.1        zip_2.2.0              mime_0.12
##  [79] slam_0.1-50            leaps_3.1              xml2_1.3.3
##  [82] tokenizers_0.2.1       compiler_4.2.0         rstudioapi_0.13
##  [85] png_0.1-7              plotly_4.10.0          ggsignif_0.6.3
##  [88] reprex_2.0.1           tweenr_1.0.2           stringi_1.7.6
##  [91] highr_0.9              lattice_0.20-45        Matrix_1.4-1
##  [94] vctrs_0.4.1            stringdist_0.9.8       pillar_1.7.0
##  [97] lifecycle_1.0.1        rscopus_0.6.6          insight_0.17.1
## [100] httpuv_1.6.5           R6_2.5.1               promises_1.2.0.1
## [103] gridExtra_2.3          janeaustenr_0.1.5      codetools_0.2-18
## [106] benchmarkmeData_1.0.4  MASS_7.3-57            assertthat_0.2.1
## [109] withr_2.5.0            hms_1.1.1              grid_4.2.0
## [112] rmarkdown_2.14         carData_3.0-5          ggforce_0.3.3
## [115] scatterplot3d_0.3-41   shiny_1.7.1            lubridate_1.8.0
## [118] tinytex_0.39           FactoMineR_2.4
```

```
## Return the machine CPU
cat("Machine:     "); print(get_cpu()$model_name)
```

```
## Machine:
```

```
## [1] "Apple M1 Max"
```

```
## Return number of true cores
cat("Num cores:   "); print(detectCores(logical = FALSE))
```

```
## Num cores:
```

```
## [1] 10
```

```
## Return number of threads
cat("Num threads: "); print(detectCores(logical = FALSE))
```

```
## Num threads:
```

```
## [1] 10
```

“‘