

Network Citation Analysis

R code

Arnald Puy

Contents

1 Preliminary	2
----------------------	----------

1 Preliminary

```
# PRELIMINARY FUNCTIONS #####

sensobol::load_packages(c("sensobol", "data.table", "tidyverse", "janitor",
                          "igraph", "ggraph", "tidygraph", "cowplot", "viridis",
                          "wesanderson", "parallel", "doParallel", "tm",
                          "benchmarkme"))

# Custom theme for plots
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, -4, -7, -7),
          plot.margin = margin(3, 4, 0, 4),
          legend.text = element_text(size = 8),
          axis.title = element_text(size = 8.5),
          legend.key.width = unit(0.4, "cm"),
          legend.key.height = unit(0.4, "cm"),
          legend.title = element_text(size = 8.5))
}

# Define color palette
selected_wesanderson <- "Chevalier1"

# DIMENSIONS DATA #####

water.models <- c("WaterGAP", "PCR-GLOBWB", "LPJmL", "CLM4.5", "DBHM",
                  "TOPMODEL", "HO8", "JULES-W1", "MPI-HM", "VIC", "SWAT",
                  "GR4J", "HYPE", "HBV", "MATSIRO", "SACRAMENTO", "MHM",
                  "CWatM", "ORCHIDEE")

dt <- list()

for (i in 1:length(water.models)) {

  dt[[i]] <- fread(paste(water.models[[i]], ".csv", sep = ""), skip = 1) %>%
    clean_names() %>%
    data.table()

}
```

```

names(dt) <- water.models
dt.water <- rbindlist(dt, idcol = "Model")
rm(dt)

wos.dt <- fread("final.dt.csv")
wos.titles <- wos.dt[Model %in% water.models]

# REMOVE DUPLICATED REFERENCES #####

# Number of papers in more than one model
n_occur <- data.frame(table(dt.water$publication_id))
papers_repeated <- data.table(n_occur[n_occur$Freq > 1,])
length(papers_repeated$Var1) # number of repeated papers

## [1] 2323

# Fraction of repeated papers over the total
length(papers_repeated$Var1) / nrow(dt.water)

## [1] 0.07791903

# How many papers are repeated twice, three times, etc...
papers_repeated[, .(N.repeated.papers = .N), Freq]

##      Freq N.repeated.papers
##      <int>          <int>
## 1:      2             1798
## 2:      4             106
## 3:      6              18
## 4:      3             348
## 5:      5              38
## 6:      8               5
## 7:      7               6
## 8:      9               1
## 9:     11               3

# Extract which papers are repeated for which model
dt.sample.repeated <- dt.water[publication_id %in% papers_repeated$Var1] %>%
  .[, .(publication_id, Model, title, source_title_anthology_title)] %>%
  .[order(publication_id)]

dt.sample.repeated

##      publication_id      Model
##      <char>          <char>
## 1: pub.1000120678    TOPMODEL
## 2: pub.1000120678    SACRAMENTO
## 3: pub.1000226548    WaterGAP
## 4: pub.1000226548    PCR-GLOBWB

```

```

##      5: pub.1000226548      HBV
##      ---
## 5482: pub.1167654662 PCR-GLOBWB
## 5483: pub.1167736853 PCR-GLOBWB
## 5484: pub.1167736853      MHM
## 5485: pub.1167835489      CLM4.5
## 5486: pub.1167835489      TOPMODEL
##
##
##      1:                      Temporal dynamics of model parameter sensitivity
##      2:                      Temporal dynamics of model parameter sensitivity
##      3:                      Multiscale
##      4:                      Multiscale
##      5:                      Multiscale
##      ---
## 5482: Scenario setup and forcing data for impact model evaluation and impact attribution wi
## 5483:      Tradeoffs Between Temporal and Spatial Pattern Calibration and Their Impacts o
## 5484:      Tradeoffs Between Temporal and Spatial Pattern Calibration and Their Impacts o
## 5485:                      Development of inter-g
## 5486:                      Development of inter-g
##
##      source_title_anthology_title
##
##      <char>
##      1:      Water Resources Research
##      2:      Water Resources Research
##      3:      Journal of Hydrometeorology
##      4:      Journal of Hydrometeorology
##      5:      Journal of Hydrometeorology
##      ---
## 5482: Geoscientific Model Development
## 5483:      Water Resources Research
## 5484:      Water Resources Research
## 5485: Geoscientific Model Development
## 5486: Geoscientific Model Development

```

```

# Randomly retrieve only one of the repeated studies per model
set.seed(6)
dt.no.repeated <- dt.sample.repeated[,.SD[sample(.N, min(1,.N))], publication_id]

# Setkey to filter and retrieve
res <- setkey(dt.water, publication_id, Model) %>%
  .[J(dt.no.repeated$publication_id, dt.no.repeated$Model)]

# Make the dataset without repeated papers across models
final.dt <- rbind(res, dt.water[!publication_id %in% papers_repeated$Var1])

# Check which papers do not have cited bibliography metadata and exclude them
final.dt <- final.dt[, empty_cited_references:= grepl("^\\s*$", cited_references)] %>%
  .[empty_cited_references == FALSE] %>%

```

```

# Filter dataset to ensure all titles use a water model
.[tolower(.$title) %in% was.titles$title.large] %>%
setnames(., "authors", "from.authors")

# Check the WOS and the Dimensions dataset
was.dimensions <- merge(was.dt[Model %in% water.models] %>%
  .[, .(WOS = .N), Model],
  final.dt[, .(Dimensions = .N), Model],
  by = "Model")

was.dimensions[order(-Dimensions)]

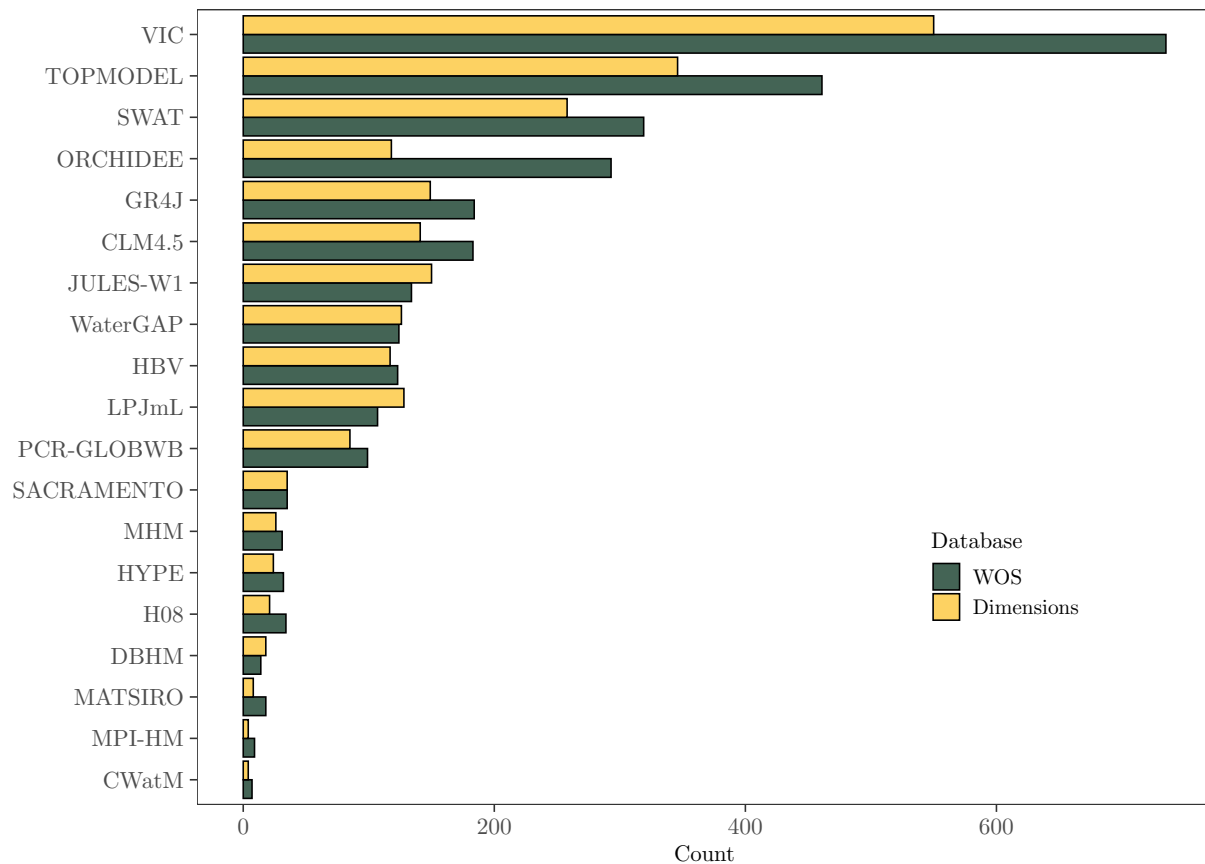
##           Model    WOS Dimensions
##           <char> <int>      <int>
## 1:         VIC    735         550
## 2:    TOPMODEL    461         346
## 3:         SWAT    319         258
## 4:    JULES-W1    134         150
## 5:         GR4J    184         149
## 6:     CLM4.5    183         141
## 7:     LPJmL    107         128
## 8:   WaterGAP    124         126
## 9:   ORCHIDEE    293         118
## 10:         HBV    123         117
## 11: PCR-GLOBWB     99          85
## 12: SACRAMENTO     35          35
## 13:         MHM     31          26
## 14:         HYPE     32          24
## 15:         H08     34          21
## 16:         DBHM     14          18
## 17:    MATSIRO     18           8
## 18:     CWatM       7           4
## 19:    MPI-HM       9           4

# PLOT DIFFERENCES BETWEEN WOS AND DIMENSIONS #####

plot.models <- was.dimensions %>%
  melt(., measure.vars = c("WOS", "Dimensions")) %>%
  ggplot(., aes(reorder(Model, value), value, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  coord_flip() +
  scale_fill_manual(name = "Database",
                    values = wes_palette(name = selected_wesanderson, 2)) +
  labs(y = "Count", x = "") +
  theme_AP() +
  theme(legend.position = c(0.8, 0.3))

plot.models

```



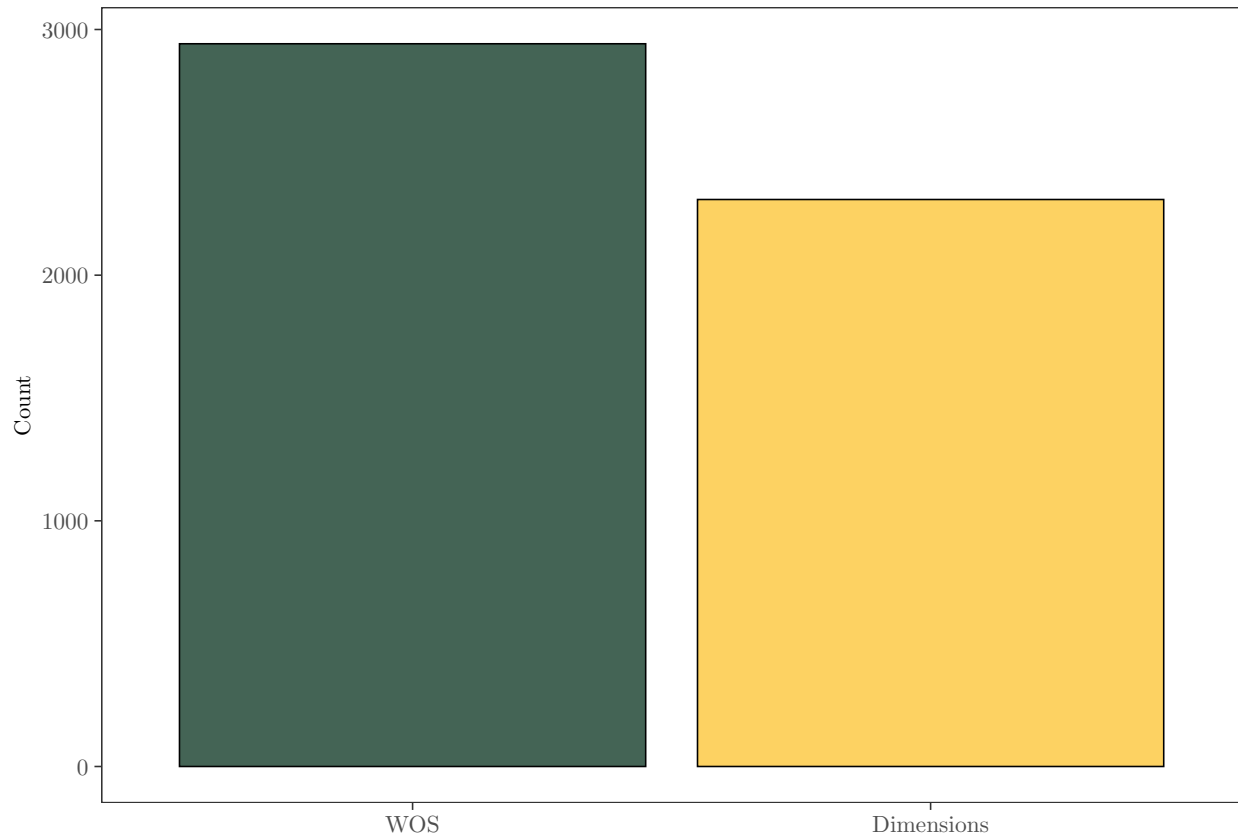
```
wos.dimensions.dt <- wos.dimensions %>%
  melt(., measure.vars = c("WOS", "Dimensions"), variable.name = "dataset") %>%
  .[, .(total = sum(value)), dataset]
```

```
wos.dimensions.dt
```

```
##      dataset total
##      <fctr> <int>
## 1:      WOS  2942
## 2: Dimensions 2308
```

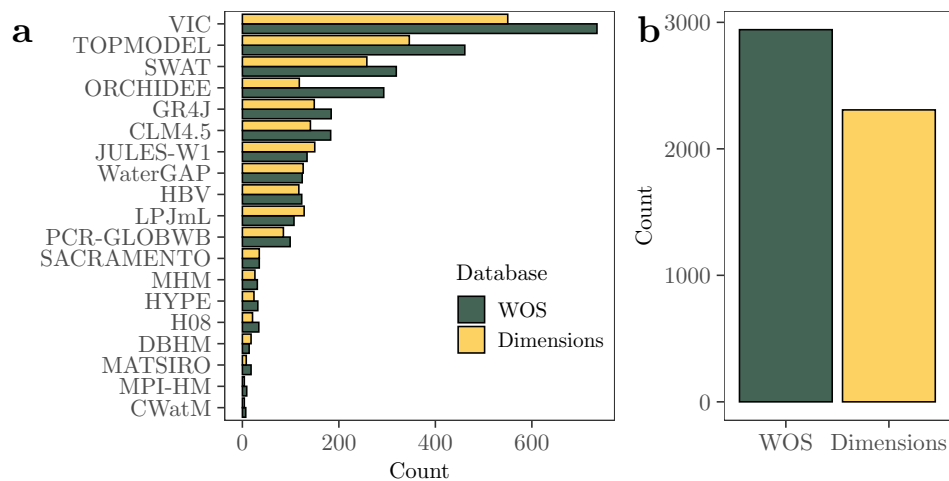
```
plot.databases <- wos.dimensions.dt %>%
  ggplot(., aes(dataset, total, fill = dataset)) +
  geom_bar(stat = "identity", color = "black") +
  scale_fill_manual(name = "Database",
                    values = wes_palette(name = selected_wesanderson, 2)) +
  labs(x = "", y = "Count") +
  theme_AP() +
  theme(legend.position = "none")
```

```
plot.databases
```



MERGE PLOTS

```
plot_grid(plot.models, plot.databases, ncol = 2, rel_widths = c(0.65, 0.35),
          labels = "auto")
```



EXTRACT REFERENCES

```
setnames(final.dt, c("Model", "publication_id"), c("citing_model", "citing_id"))

column_names <- c("authors",
                  "author_id",
```

```

        "source",
        "year",
        "volume",
        "issue",
        "pagination",
        "to.doi",
        "publication_id",
        "times_cited")

direct.citation <- final.dt %>%
  .[, .(citing_id, cited_references, citing_model, title, doi, pub_year, from.authors)] %>%
  separate_rows(cited_references, sep = ";(?:=\\[\\])") %>%
  separate(col = cited_references, into = column_names, sep = "\\|") %>%
  data.table() %>%
  setnames(., "authors", "to.authors")

## Warning: Expected 10 pieces. Additional pieces discarded in 16 rows [8733, 9603, 9819,
## 13673, 26551, 28818, 30722, 42918, 76098, 81210, 81628, 83529, 98793, 100483,
## 104944, 106015].

# ARRANGE DATA FOR NETWORK CITATION ANALYSIS #####

# Create a directed graph from the dataset
edges <- data.table(from = direct.citation$citing_id,
  to = direct.citation$publication_id,
  from.model = direct.citation$citing_model,
  year = direct.citation$year,
  n.citations = direct.citation$times_cited,
  to.doi = direct.citation$to.doi,
  to.authors = direct.citation$to.authors)

# Merge data from citing papers with data from cited papers
network.dt <- merge(edges, final.dt[, .(citing_id, doi, pub_year, times_cited, from.authors)],
  by.x = "from", by.y = "citing_id")

colnames(network.dt)

## [1] "from"          "to"            "from.model"    "year"          "n.citations"
## [6] "to.doi"        "to.authors"    "doi"           "pub_year"      "times_cited"
## [11] "from.authors"

# Change column names to clarify
new_colnames <- c("from", "to", "from.model", "to.year", "to.n.citations",
  "to.doi", "to.authors", "from.doi", "from.year", "from.n.citations", "from.a

setnames(network.dt, colnames(network.dt), new_colnames)

# Reorder columns
setcolorder(network.dt, c("from", "to", "from.year", "to.year", "from.authors",

```



```

        "to.authors", "from.n.citations", "to.n.citations",
        "from.doi", "to.doi", "from.model"))

# Remove square brackets from the to.authors column
network.dt[, to.authors:= gsub("\\[|\\]", "", to.authors)]

# Identify the model of the cited paper
tmp <- network.dt[, .(from.model, from)] %>%
  unique()
setkey(tmp, "from")

# Define parallel computing
cl <- makeCluster(floor(detectCores() * 0.75))
registerDoParallel(cl)

# Search in parallel
to.model <- foreach(i=1:nrow(network.dt),
  .packages = "data.table",
  .combine = "c") %dopar%
{
  tmp[network.dt[[i, "to"]]]$from.model
}

# Stop parallel cluster
stopCluster(cl)

# Add vector of model names
network.dt[, to.model:= to.model]

# Turn some columns into numeric
columns_to_modify <- grep("citation", names(network.dt), value = TRUE)
network.dt[, (columns_to_modify):= lapply(.SD, as.numeric), .SDcols = columns_to_modify]

## Warning in lapply(.SD, as.numeric): NAs introduced by coercion

# Export dataset
fwrite(network.dt, "network.dt.csv")

# PLOT DESCRIPTIVE FIGURES #####

plot.n.citing <- network.dt[, unique(from.n.citations), from] %>%
  ggplot(., aes(V1)) +
  geom_histogram(color = "black", fill = "grey") +
  scale_x_log10(breaks = scales::trans_breaks("log10", function(x) 10^(2 * x)),
    labels = scales::trans_format("log10", scales::math_format(10^.x))) +
  theme_AP() +
  labs(x = "Citations", y = "N° papers") +
  ggtitle("Citing")

```

```
plot.n.citing
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 71 rows containing non-finite values (`stat_bin()`).
## Warning in (function (texString, cex = 1, face = 1, engine =
## getOption("tikzDefaultEngine"), : Attempting to calculate the width of a
## Unicode string using the pdftex engine. This may fail! See the Unicode section of
## ?tikzDevice for more information.

## Warning in (function (texString, cex = 1, face = 1, engine =
## getOption("tikzDefaultEngine"), : Attempting to calculate the width of a
## Unicode string using the pdftex engine. This may fail! See the Unicode section of
## ?tikzDevice for more information.

## Warning in (function (texString, cex = 1, face = 1, engine =
## getOption("tikzDefaultEngine"), : Attempting to calculate the width of a
## Unicode string using the pdftex engine. This may fail! See the Unicode section of
## ?tikzDevice for more information.

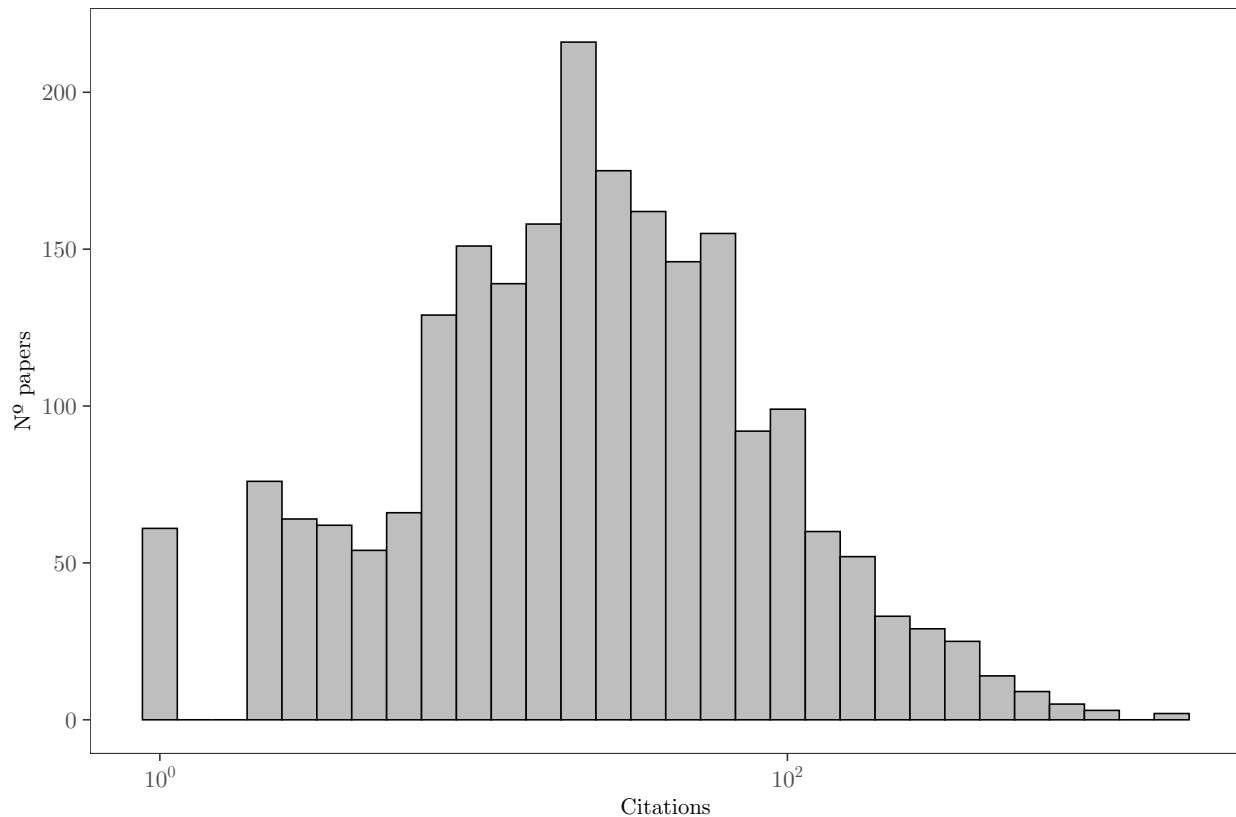
## Warning in (function (texString, cex = 1, face = 1, engine =
## getOption("tikzDefaultEngine"), : Attempting to calculate the width of a
## Unicode string using the pdftex engine. This may fail! See the Unicode section of
## ?tikzDevice for more information.

## Warning in (function (texString, cex = 1, face = 1, engine =
## getOption("tikzDefaultEngine"), : Attempting to calculate the width of a
## Unicode string using the pdftex engine. This may fail! See the Unicode section of
## ?tikzDevice for more information.

## Warning in (function (texString, cex = 1, face = 1, engine =
## getOption("tikzDefaultEngine"), : Attempting to calculate the width of a
## Unicode string using the pdftex engine. This may fail! See the Unicode section of
## ?tikzDevice for more information.

## Warning in (function (texString, cex = 1, face = 1, engine =
## getOption("tikzDefaultEngine"), : Attempting to calculate the width of a
## Unicode string using the pdftex engine. This may fail! See the Unicode section of
## ?tikzDevice for more information.
```

Citing

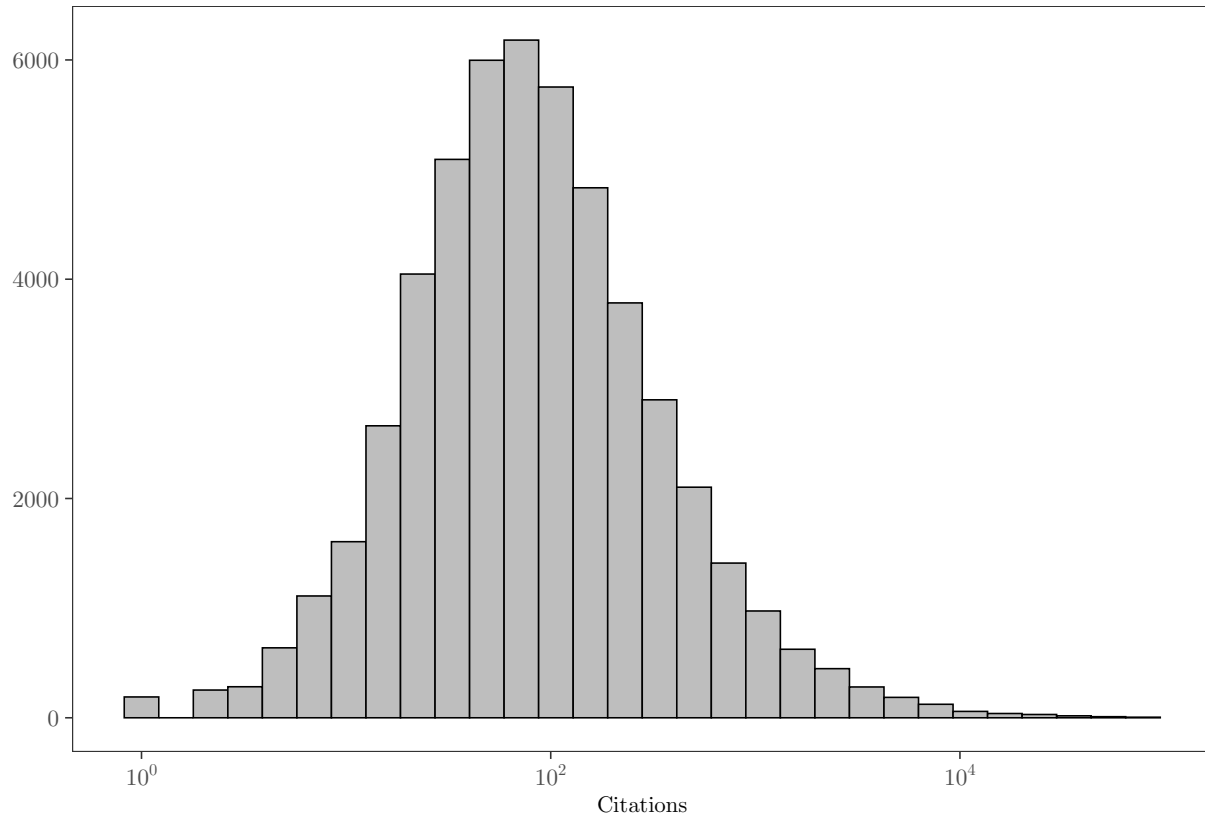


```
plot.n.cited <- network.dt[, unique(to.n.citations), to] %>%
  ggplot(., aes(V1)) +
  geom_histogram() +
  scale_x_log10(breaks = scales::trans_breaks("log10", function(x) 10^(2 * x)),
               labels = scales::trans_format("log10", scales::math_format(10^.x))) +
  geom_histogram(color = "black", fill = "grey") +
  theme_AP() +
  labs(x = "Citations", y = "") +
  ggtitle("Cited")
```

```
plot.n.cited
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 17 rows containing non-finite values (`stat_bin()`).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 17 rows containing non-finite values (`stat_bin()`).
```

Cited



CHECK WHETHER WATER MODEL COMMUNITIES CITE THEIR OWN MODEL THE MOST

```
da <- network.dt[, .N, .(from.model, to.model)] %>%
  .[!is.na(to.model)]
```

```
keycol <-c("from.model", "N")
setorderv(da, keycol, order = -1)
dcast(da, from.model~to.model, value.var = "N")
```

Key: <from.model>

##	from.model	CLM4.5	CWatM	DBHM	GR4J	H08	HBV	HYPE	JULES-W1	LPJmL
##	<char>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
## 1:	CLM4.5	144	NA	1	NA	NA	NA	NA	30	32
## 2:	CWatM	1	NA	NA	NA	NA	NA	NA	NA	1
## 3:	DBHM	2	NA	29	3	NA	NA	NA	2	9
## 4:	GR4J	NA	NA	3	208	2	23	4	4	6
## 5:	H08	NA	NA	1	3	7	NA	NA	2	14
## 6:	HBV	NA	NA	2	47	1	60	12	2	NA
## 7:	HYPE	NA	NA	1	4	NA	3	50	NA	2
## 8:	JULES-W1	15	NA	NA	5	NA	2	1	224	40
## 9:	LPJmL	1	NA	1	5	1	NA	NA	23	388
## 10:	MATSIRO	NA	NA	NA	NA	NA	NA	NA	1	2
## 11:	MHM	NA	NA	2	17	NA	10	4	9	1
## 12:	MPI-HM	NA	NA	1	NA	NA	NA	NA	1	3

## 13:	ORCHIDEE	15	NA	1	NA	1	NA	NA	45	58
## 14:	PCR-GLOBWB	4	1	NA	10	5	3	3	3	38
## 15:	SACRAMENTO	NA	NA	2	8	NA	1	1	NA	NA
## 16:	SWAT	1	NA	3	16	1	12	14	7	5
## 17:	TOPMODEL	3	NA	NA	32	NA	10	5	17	2
## 18:	VIC	5	NA	24	36	NA	35	7	16	14
## 19:	WaterGAP	6	NA	6	NA	2	1	NA	2	45
##	MATSIRO	MHM	MPI-HM	ORCHIDEE	PCR-GLOBWB	SACRAMENTO	SWAT	TOPMODEL	VIC	
##	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	
## 1:	NA	NA	NA	24	3	NA	1	26	31	
## 2:	NA	1	NA	NA	4	NA	1	NA	13	
## 3:	NA	1	1	7	5	2	2	5	39	
## 4:	NA	8	NA	NA	7	19	7	264	82	
## 5:	NA	NA	2	2	5	NA	NA	2	3	
## 6:	NA	1	NA	3	13	8	13	65	42	
## 7:	NA	2	NA	1	2	1	3	7	11	
## 8:	NA	8	1	51	9	9	2	89	17	
## 9:	NA	NA	NA	26	22	1	1	8	22	
## 10:	4	NA	NA	NA	NA	NA	NA	12	4	
## 11:	NA	58	NA	3	19	4	3	9	24	
## 12:	NA	NA	1	NA	NA	NA	NA	3	2	
## 13:	NA	3	NA	194	5	1	1	25	15	
## 14:	NA	14	2	8	139	4	1	12	61	
## 15:	NA	2	NA	2	2	25	2	36	18	
## 16:	NA	2	NA	5	8	3	116	45	81	
## 17:	1	1	2	26	8	23	9	1170	127	
## 18:	NA	17	2	7	42	23	27	120	2450	
## 19:	NA	4	3	50	80	NA	NA	11	47	
##	WaterGAP									
##	<int>									
## 1:	11									
## 2:	6									
## 3:	13									
## 4:	13									
## 5:	10									
## 6:	14									
## 7:	3									
## 8:	12									
## 9:	67									
## 10:	2									
## 11:	11									
## 12:	4									
## 13:	24									
## 14:	119									
## 15:	1									
## 16:	18									
## 17:	3									
## 18:	38									

```
## 19:      447
```

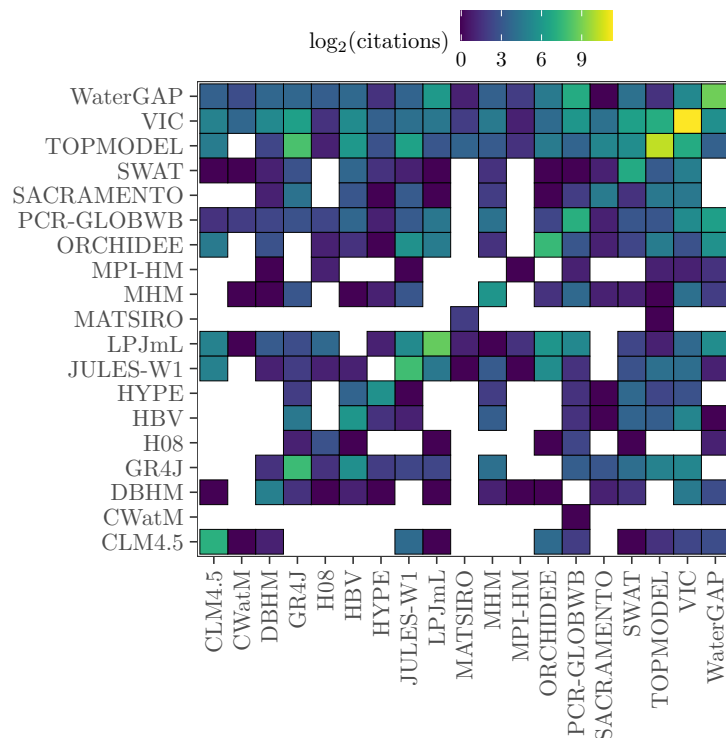
```
# Identify which model is the most cited
largest.citation <- da[da[, .I[N == max(N)], from.model]$V1]
sum(largest.citation$from.model == largest.citation$to.model) / nrow(largest.citation)
```

```
## [1] 0.5789474
```

```
da[, N:= log2(N)]
```

```
# Tile plot
```

```
ggplot(da, aes(x = from.model, y = to.model, fill = N))+
  geom_tile(col = "black") +
  scale_fill_viridis(name = "log$_2$(citations)") +
  theme_AP() +
  theme(legend.position = "top") +
  labs(x = "", y = "") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



```
# DISTRIBUTION OF CITATIONS PER MODEL #####
```

```
from.vec <- unique(network.dt$from)
to.vec <- unique(network.dt$to)

total.per.model <- network.dt[, unique(from), from.model] %>%
  .[, .(total = .N), from.model]

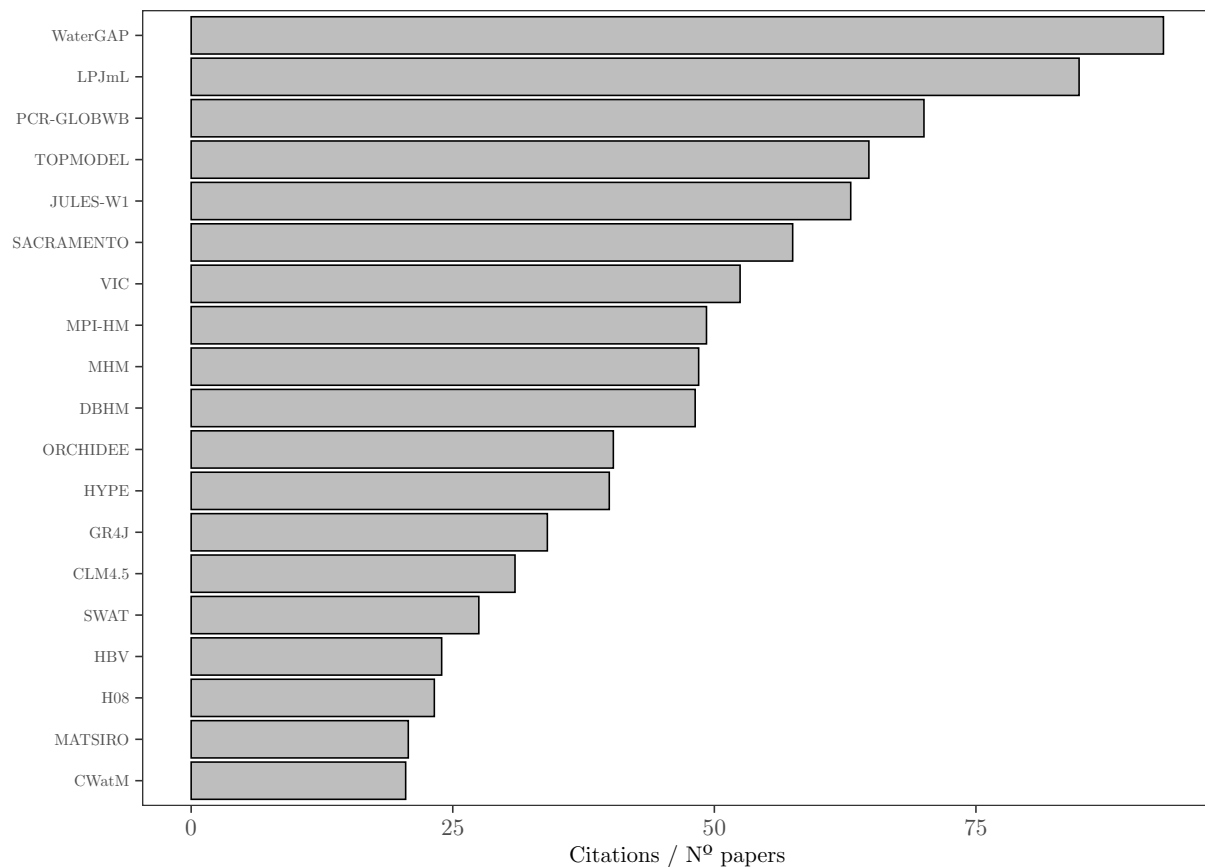
plot.total <- network.dt[, unique(from), .(from.model, from.n.citations)] %>%
```

```

[, sum(from.n.citations), from.model] %>%
merge(., total.per.model, by = "from.model") %>%
[, normalized:= V1 / total] %>%
ggplot(., aes(reorder(from.model, normalized), normalized)) +
geom_bar(stat = "identity", color = "black", fill = "grey") +
coord_flip() +
theme_AP() +
labs(y = "Citations / N° papers", x = "") +
theme(axis.text.y = element_text(size = 6))

```

plot.total



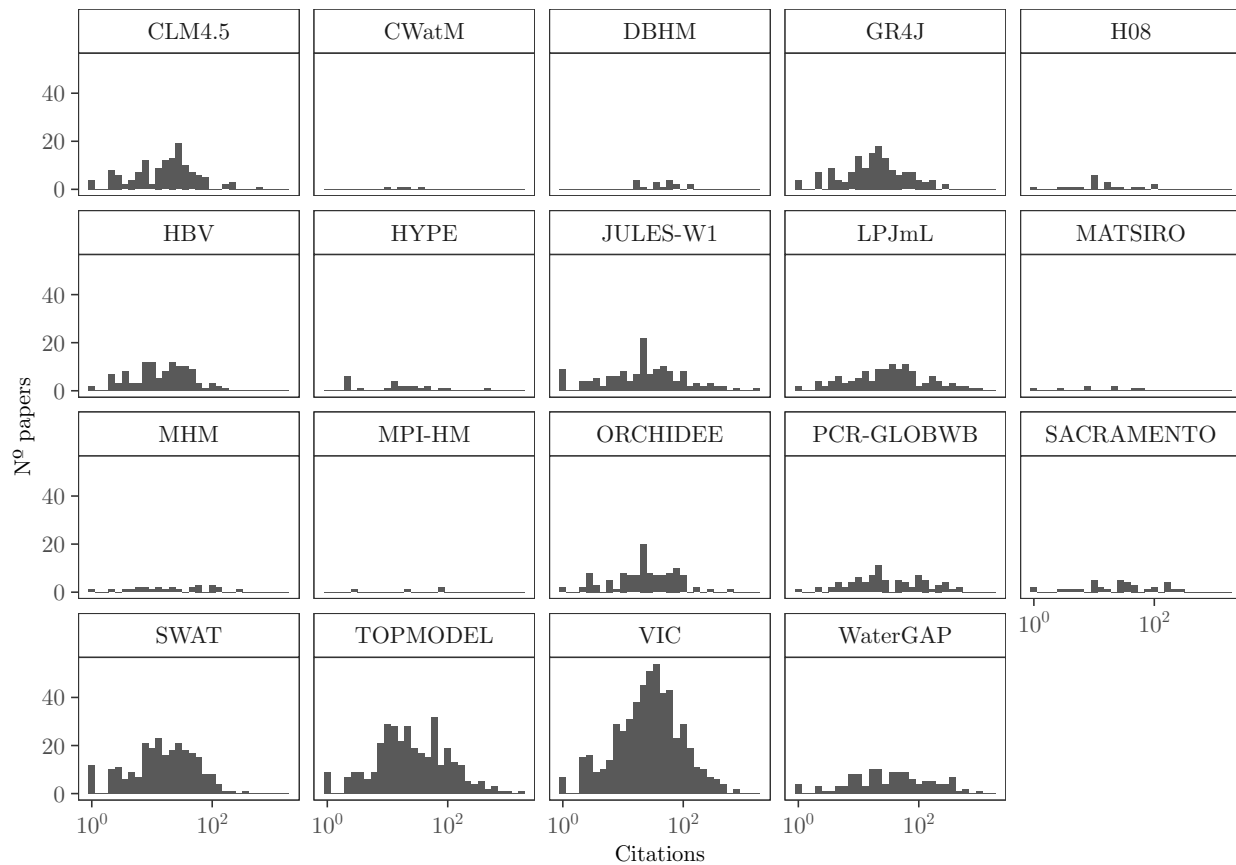
```

plot.total.model <- network.dt[, unique(from), .(from.model, from.n.citations)] %>%
ggplot(., aes(from.n.citations)) +
geom_histogram() +
scale_x_log10(breaks = scales::trans_breaks("log10", function(x) 10^(2 * x)),
              labels = scales::trans_format("log10", scales::math_format(10^.x))) +
facet_wrap(~from.model) +
labs(x = "Citations", y = "N° papers") +
theme_AP()

```

plot.total.model

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



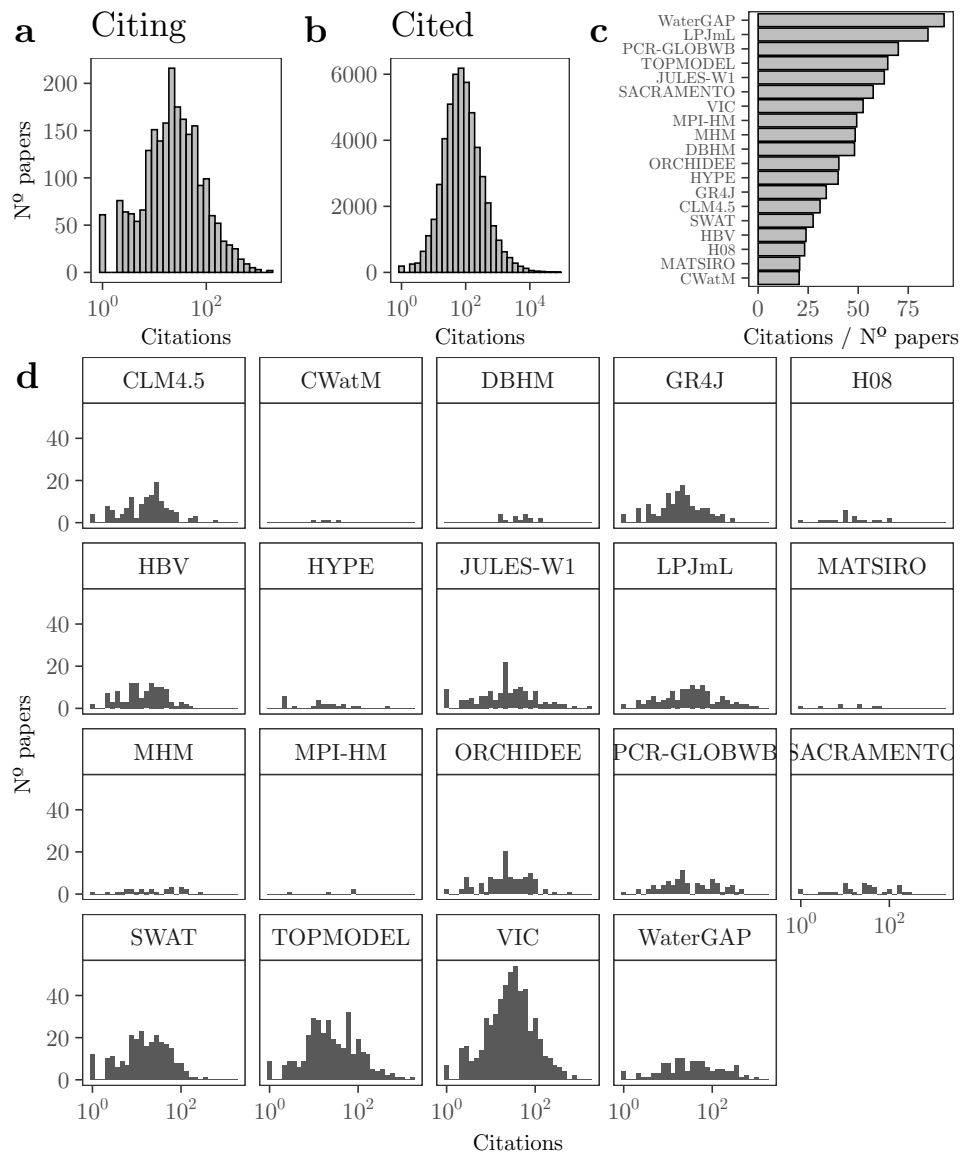
```
# MERGE PLOTS #####
```

```
top <- plot_grid(plot.n.citing, plot.n.cited, plot.total,
  ncol = 3, labels = "auto", rel_widths = c(0.3, 0.3, 0.4))
```

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

```
plot_grid(top, plot.total.model, ncol = 1, rel_heights = c(0.3, 0.7),
  labels = c("", "d"))
```

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

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

```
sessionInfo()
```

```
## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.2.1
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-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
##
## time zone: Europe/London
## tzcode source: internal
##
## attached base packages:
## [1] parallel stats graphics grDevices utils datasets methods
## [8] base
##
## other attached packages:
## [1] benchmarkme_1.0.8 tm_0.7-11 NLP_0.2-1 doParallel_1.0.17
## [5] iterators_1.0.14 foreach_1.5.2 wesanderson_0.3.6 viridis_0.6.4
## [9] viridisLite_0.4.2 cowplot_1.1.1 tidygraph_1.3.0 ggraph_2.1.0
## [13] igraph_1.6.0 janitor_2.2.0 lubridate_1.9.2 forcats_1.0.0
## [17] stringr_1.5.1 dplyr_1.1.4 purrr_1.0.2 readr_2.1.4
## [21] tidyr_1.3.0 tibble_3.2.1 ggplot2_3.4.4 tidyverse_2.0.0
## [25] data.table_1.14.99 sensobol_1.1.4
##
## loaded via a namespace (and not attached):
## [1] benchmarkmeData_1.0.4 gtable_0.3.4 xfun_0.39
## [4] ggrepel_0.9.5 lattice_0.22-5 tzdb_0.3.0
## [7] vctrs_0.6.5 tools_4.3.3 Rdpack_2.6
## [10] generics_0.1.3 fansi_1.0.6 pkgconfig_2.0.3
## [13] Matrix_1.6-5 filehash_2.4-5 lifecycle_1.0.4
## [16] compiler_4.3.3 farver_2.1.1 munsell_0.5.0
## [19] ggforce_0.4.1 graphlayouts_1.0.2 codetools_0.2-19
## [22] snakecase_0.11.1 htmltools_0.5.5 yaml_2.3.7
## [25] pillar_1.9.0 MASS_7.3-60.0.1 tidyselect_1.2.0
## [28] digest_0.6.34 slam_0.1-50 stringi_1.8.3
## [31] polyclip_1.10-6 fastmap_1.1.1 grid_4.3.3
## [34] colorspace_2.1-0 cli_3.6.2 magrittr_2.0.3
## [37] utf8_1.2.4 withr_3.0.0 scales_1.3.0
## [40] timechange_0.2.0 httr_1.4.5 rmarkdown_2.21
## [43] gridExtra_2.3 hms_1.1.3 evaluate_0.20
```

```
## [46] knitr_1.42          rbibutils_2.2.16      rlang_1.1.3
## [49] Rcpp_1.0.12          glue_1.7.0           xml2_1.3.3
## [52] tweenr_2.0.2         rstudioapi_0.15.0    tikzDevice_0.12.4
## [55] R6_2.5.1
```

```
## 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
```