Assumptions in global irrigation models

Seth Nathaniel Linga

2025-03-04

Contents

1 Number of assumptions

```r
data_countA <- read_excel("assumptions.xlsx", sheet = "total")

p_countA <- ggplot(data_countA, aes(y = fct_reorder(factor(Model), Total, .desc = FALSE), x = Total, fi
  geom_bar(stat = "identity") +
  labs(title = "a",
       x = "number of assumptions",
       y = "") +
  theme_bw() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
    axis.title.x = element_text(size = 14),
    axis.text.x = element_text(size = 12),
    axis.text.y = element_text(size = 14),
    legend.position = "none",
    panel.grid = element_blank()
  ) +
  scale_fill_manual(values = "grey50")

data_countB <- read_excel("assumptions.xlsx", sheet = "study") %>%
  mutate(Year = as.numeric(sub(".*(\\d{4}).*", "\\1", Study))) %>%
  filter(!is.na(Year))

model_order <- c("MPI-HM", "MATSIRO", "CLM4.5", "VIC", "WaterGAP", "WBMplus", "PCR-GLOBWB", "H08", "LPJm

data_countB <- data_countB %>%
  mutate(Model = factor(Model, levels = model_order))

p_countB <- ggplot(data_countB, aes(x = Year, y = Model, size = `Count of assumptions`, color = "grey50
  geom_point(alpha = 0.7) +
  theme_bw() +
  labs(title = "b", x = "year", y = "", size = "number of\nassumptions") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
    axis.title.x = element_text(size = 14),
    axis.text.x = element_text(size = 12),
    axis.text.y = element_blank(),
    legend.title = element_text(size = 14),
    legend.text = element_text(size = 12),
    panel.grid = element_blank()
  ) +
  guides(color = "none") +
  scale_size_continuous(range = c(3, 15)) +
  scale_color_manual(values = "grey50")

combined_count <- grid.arrange(p_countA, p_countB, ncol = 2, nrow = 1)
grid.draw(combined_count)
```
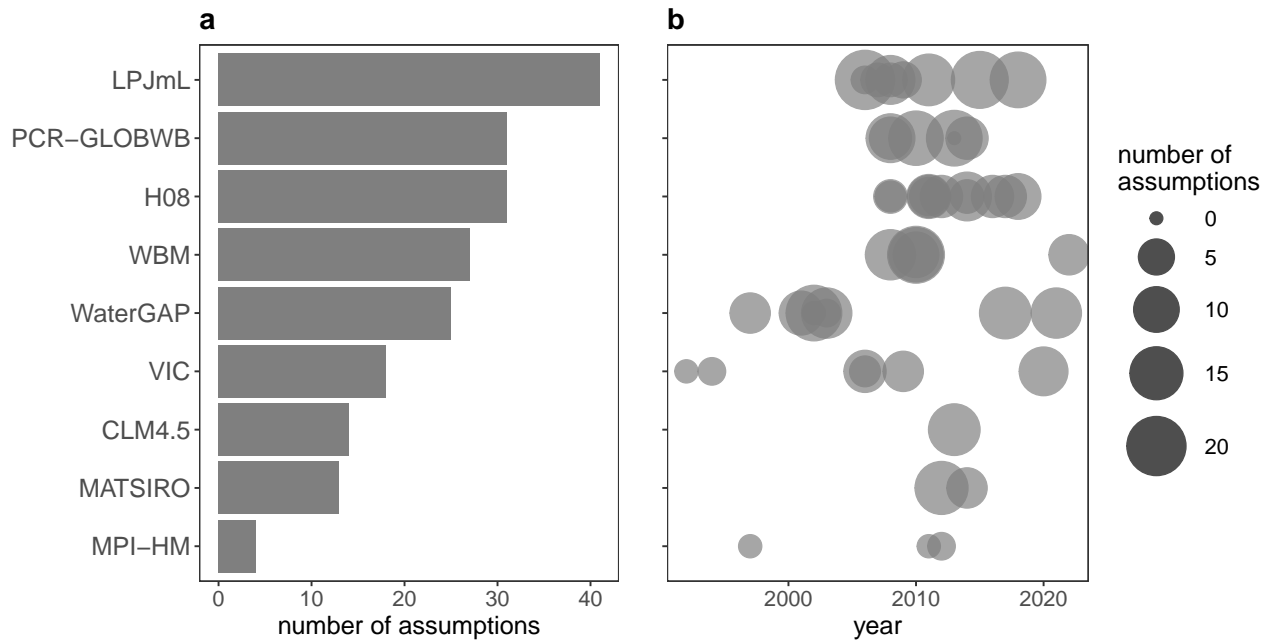
2 Nature of assumptions

```r
data_aspectA <- read_excel("assumptions.xlsx", sheet = "nature")

model_order_aspectA <- c("CLM4.5", "VIC", "MATSIRO", "PCR-GLOBWB", "LPJmL", "WaterGAP", "H08", "WBM",

data_aspectA <- data_aspectA %>%
  mutate(Total = Pragmatic + Empirical,
         Pragmatic_ratio = Pragmatic / Total,
         Empirical_ratio = Empirical / Total)

data_aspectA_long <- data_aspectA %>%
  pivot_longer(cols = c(Pragmatic, Empirical),
               names_to = "Nature", values_to = "Count") %>%
  mutate(
    Model = factor(Model, levels = model_order_aspectA),
    Ratio = Count / Total
  )

custom_colors <- c("Pragmatic" = "#1ac6c6", "Empirical" = "#f07167")

p_aspectA <- ggplot(data_aspectA_long, aes(x = Model, y = Ratio, fill = Nature)) +
  geom_bar(stat = "identity", position = "fill") +
  geom_text(aes(label = paste0(Count, "/", Total)),
            position = position_stack(vjust = 0.5), size = 3.5, color = "black") +
  labs(title = "a", x = "", y = "fraction") +
  scale_fill_manual(values = custom_colors, labels = c("Empirical", "Pragmatic")) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
    axis.text.x = element_text(size = 12, angle = 45, hjust = 1),
    axis.ticks.x = element_blank(),
```

3

```r
    axis.text.y = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    panel.grid = element_blank(),
    legend.position = "none"
  )

data_aspectB <- read_excel("assumptions.xlsx", sheet = "aspect")

data_aspectB_df <- data_aspectB %>%
  group_by(Model, Aspect) %>%
  mutate(Proportion = Count / sum(Count))

p_aspectB <- ggplot(data_aspectB_df, aes(x = Aspect, y = Proportion, fill = Nature)) +
  geom_bar(stat = "identity", position = "fill") +
  geom_point(aes(size = Count, fill = Nature),
             position = position_fill(vjust = 0.5),
             color = "black", shape = 21) +
  scale_size_continuous(name = "count", range = c(1, 8)) +
  labs(title = "b", x = "", y = "", fill = "Nature of Assumption") +
  facet_wrap(~ Model) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
    axis.text.x = element_text(size = 12, angle = 45, hjust = 1),
    axis.ticks.x = element_blank(),
    axis.text.y = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 12),
    panel.grid = element_blank(),
    strip.background = element_rect(fill = "white", color = "black"),
    strip.text = element_text(size = 12)
  ) +
  guides(fill = guide_legend(title = NULL),
         size = guide_legend(title = "number of\nassumptions"))

combined_aspect <- grid.arrange(p_aspectA, p_aspectB, ncol = 2, nrow = 1, widths = c(2, 3))
grid.draw(combined_aspect)
```
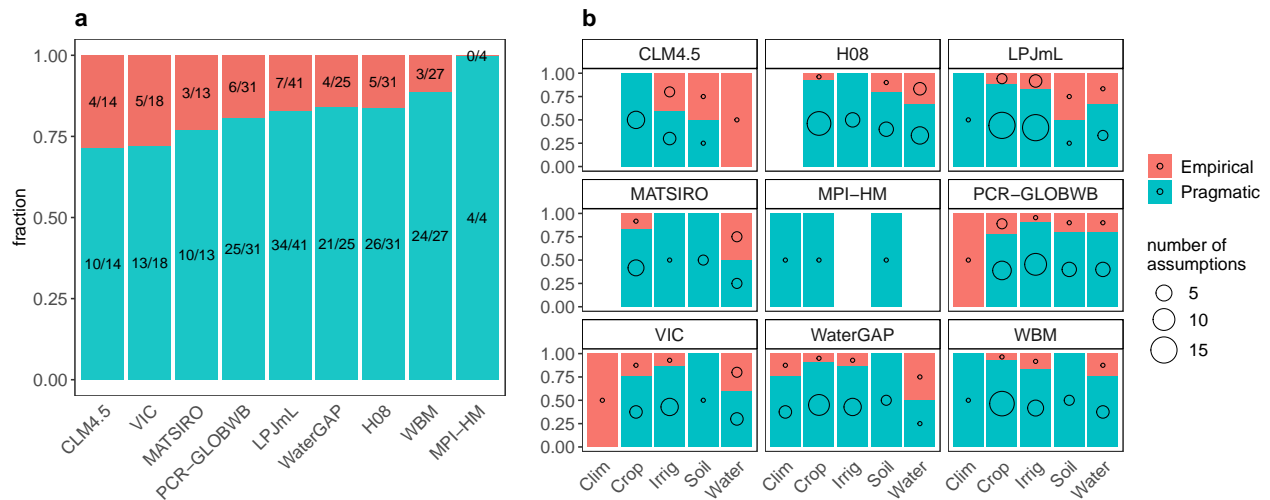
3 Evolution of assumptions (LPJmL)

```r
data_code <- read_excel("assumptions.xlsx", sheet = "code") %>%
  filter(Model == "LPJmL") %>%
  mutate(UniqueID = paste(Code, Author, sep = "_")) %>%
  arrange(Code, Year) %>%
  group_by(Code) %>%
  mutate(NextAuthor = lead(Author),
         NextYear = lead(Year),
         NextCode = lead(Code)) %>%
  ungroup()

edges_code <- data_code %>%
  filter(!is.na(NextAuthor) & Code == NextCode) %>%
  mutate(to = paste(NextCode, NextAuthor, sep = "_")) %>%
  select(from = UniqueID, to)

vertices_code <- data_code %>%
  distinct(UniqueID, Code, Author, Year)

facet_rects_code <- data.frame(
  xmin = c(1, 37, 73, 91, 95),
  xmax = c(36, 72, 90, 94, 105),
  facet = c("Crop", "Irrigation", "Soil", "Climate", "Water"),
  fill_color = c("lightgreen", "lavender", "peachpuff", "lightcoral", "lightblue"),
  label_x = c(18.5, 54.5, 81.5, 92.5, 100)
)

ellipse_data_code <- data.frame(
  x0 = c(17, 64),
  y0 = c(2011, 2015),
  a = c(12, 36),
  b = c(0.8, 0.8),
  angle = c(0, 0)
)

graph_code <- graph_from_data_frame(d = edges_code, vertices = vertices_code, directed = FALSE)
V(graph_code)$degree <- degree(graph_code, mode = "all")

custom_layout_code <- as.data.frame(vertices_code %>% select(Code, Year))
colnames(custom_layout_code) <- c("x", "y")

ggraph(graph_code, layout = custom_layout_code) +
  geom_rect(data = facet_rects_code, aes(xmin = xmin, xmax = xmax, ymin = -Inf, ymax = Inf, fill = facet
            alpha = 0.2, color = NA) +
  scale_fill_manual(values = setNames(facet_rects_code$fill_color, facet_rects_code$facet)) +
  geom_edge_link(arrow = arrow(length = unit(2, 'mm')),
                 end_cap = circle(2, 'mm'),
                 color = 'gray60') +
  geom_node_point(color = "gray40", size = 3) +
  scale_x_continuous(breaks = seq(min(vertices_code$Code), max(vertices_code$Code), by = 1), expand = c
  scale_y_reverse(breaks = seq(min(vertices_code$Year), max(vertices_code$Year), by = 2), expand = c(0.
  labs(x = "", y = "year") +
```
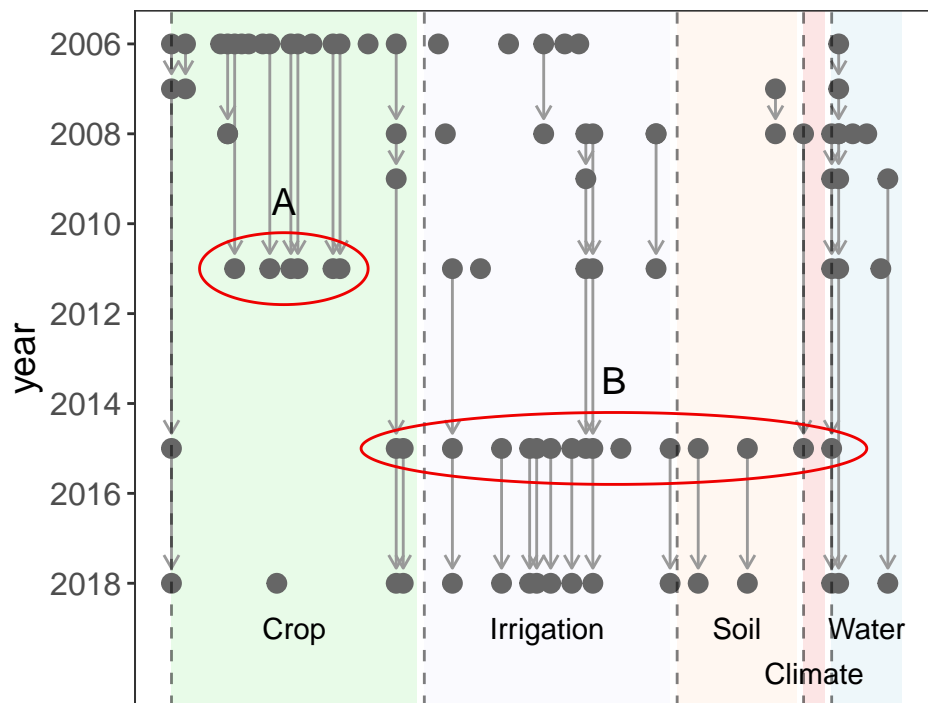
```r
  theme_bw() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
    axis.text.y = element_text(size = 12),
    panel.grid = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_text(size = 14),
    legend.position = "none"
  ) +
  geom_vline(data = facet_rects_code, aes(xintercept = xmin), color = "black", linetype = "dashed", size
  geom_text(data = facet_rects_code, aes(x = label_x,
                                   y = ifelse(facet == "Climate", max(vertices_code$Year) + 2.0, max(ve
                                   label = facet), vjust = 0.5, color = "black", size = 4) +
  ggforce::geom_ellipse(data = ellipse_data_code,
                    aes(x0 = x0, y0 = y0, a = a, b = b, angle = angle),
                    color = "red2", fill = NA, size = 0.5) +
  annotate("text", x = 17, y = 2009.5, label = "A", size = 5, color = "black") +
  annotate("text", x = 64, y = 2013.5, label = "B", size = 5, color = "black")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


4   Sharing of assumptions

```r
data_freq <- read_excel("assumptions.xlsx", sheet = "freq")
```

```r
filtered_freq <- data_freq %>%
  filter(Frequency >= 2) %>%
  group_by(Nature) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = Count / sum(Count) * 100)

p_hist <- ggplot(data_freq, aes(x = Frequency)) +
  geom_bar() +
  scale_y_continuous(breaks = seq(0, 80, by = 10)) +
  labs(title = "a", x = "number of sharing models", y = "number of assumptions") +
  theme_bw() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
    axis.text = element_text(size = 12),
    axis.title.x = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    panel.grid = element_blank(),
    legend.position = "none"
  )

assumption_summary <- data_freq %>%
  mutate(Group = ifelse(Frequency == 1, "unique \nto a model", "shared by \n  2 models")) %>%
  group_by(Group, Nature) %>%
  summarise(Count = n()) %>%
  group_by(Group) %>%
  mutate(Percentage = Count / sum(Count) * 100) %>%
  mutate(Group = factor(Group, levels = c("unique \nto a model", "shared by \n  2 models")))

## `summarise()` has grouped output by 'Group'. You can override using the
## `.groups` argument.

p_normalized <- ggplot(assumption_summary, aes(x = Group, y = Percentage, fill = Nature)) +
  geom_bar(stat = "identity", position = "stack") +
  scale_fill_manual(values = c("Empirical" = "#f07167", "Pragmatic" = "#1ac6c6")) +
  scale_y_continuous(breaks = seq(0, 100, by = 20)) +
  labs(title = "b", x = "", y = "proportion") +
  theme_bw() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
    axis.text = element_text(size = 12),
    axis.title.x = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    panel.grid = element_blank(),
    legend.position = "top",
    legend.title = element_blank(),
    legend.text = element_text(size = 10)
  )

combined_freq <- grid.arrange(p_hist, p_normalized, ncol = 2, widths = c(0.55, 0.45))
grid.draw(combined_freq)
```
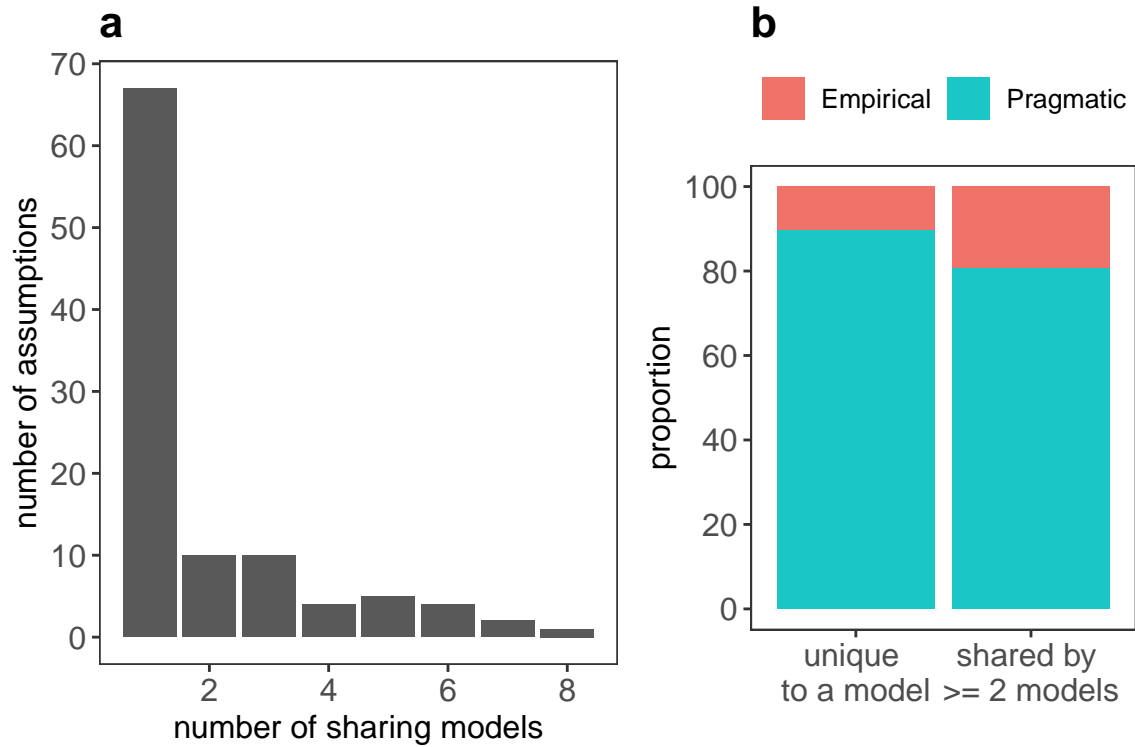
5 Network citation of pragmatic assumptions

```r
nodes_net <- read_excel("assumptions.xlsx", sheet = "cite_nodes")
edges_net <- read_excel("assumptions.xlsx", sheet = "cite_edges") %>%
  filter(!is.na(Source) & !is.na(Target))

nodes_net <- nodes_net %>% distinct(ID, .keep_all = TRUE)
model_colors_net <- setNames(scales::hue_pal()(length(unique(nodes_net$Model))), unique(nodes_net$Model))

size_range_net <- c(1, 10)
degree_limits_net <- range(nodes_net$degree, na.rm = TRUE)

titles_net <- c("a. Paddy vs nonpaddy", "b. Unlimited water supply")
plot_list_net <- list()

for (dataset in unique(nodes_net$Dataset)) {
  dataset_nodes <- nodes_net %>% filter(Dataset == dataset)
  dataset_edges <- edges_net %>% filter(Dataset == dataset & Source %in% dataset_nodes$ID & Target %in%

  graph_net <- graph_from_data_frame(dataset_edges, vertices = dataset_nodes, directed = TRUE)
  layout_net <- create_layout(graph_net, layout = 'mds')
  layout_net$x <- as.numeric(dataset_nodes$Year)
  layout_net$y <- runif(length(V(graph_net)), min = -10, max = 10)

  plot_net <- ggraph(layout_net) +
    geom_edge_link(arrow = arrow(length = unit(3, 'mm')), end_cap = circle(4, 'mm'), color = 'gray') +
    geom_node_point(aes(color = Model, size = degree), show.legend = FALSE) +
    geom_node_text(aes(label = Label), repel = TRUE, size = 3.5) +
```

```r
    scale_color_manual(values = model_colors_net) +
    scale_size_continuous(limits = degree_limits_net, range = size_range_net) +
    scale_x_continuous(breaks = seq(min(dataset_nodes$Year), max(dataset_nodes$Year), by = 5), expand =
    labs(title = titles_net[which(unique(nodes_net$Dataset) == dataset)], x = "") +
    theme_bw() +
    theme(
      plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
      axis.text.x = element_text(size = 12),
      panel.grid = element_blank(),
      axis.text.y = element_blank(),
      axis.ticks.y = element_blank(),
      axis.title.y = element_blank()
    )

  plot_list_net[[dataset]] <- plot_net
}

legend_plot_net <- ggplot(nodes_net, aes(x = 1, y = 1, color = Model, size = degree)) +
  geom_point(show.legend = TRUE) +
  scale_color_manual(values = model_colors_net, guide = guide_legend(nrow = 1)) +
  scale_size_continuous(name = "citation", limits = degree_limits_net, range = size_range_net, guide =
  theme_void() +
  theme(legend.position = "bottom", legend.direction = "horizontal",
        legend.box = "vertical", legend.box.spacing = unit(0.25, "lines"),
        plot.title = element_text(size = 16, face = "bold", hjust = 0, vjust = 1),
        legend.title = element_text(size = 14),  # Larger legend title
        legend.text = element_text(size = 12))   # Larger legend text

# Arrange plots with combined legend
final_plot <- grid.arrange(
  arrangeGrob(grobs = plot_list_net, nrow = 1),
  legend_plot_net,
  ncol = 1,
  heights = c(4, 0.6)
)

grid.draw(final_plot)
```
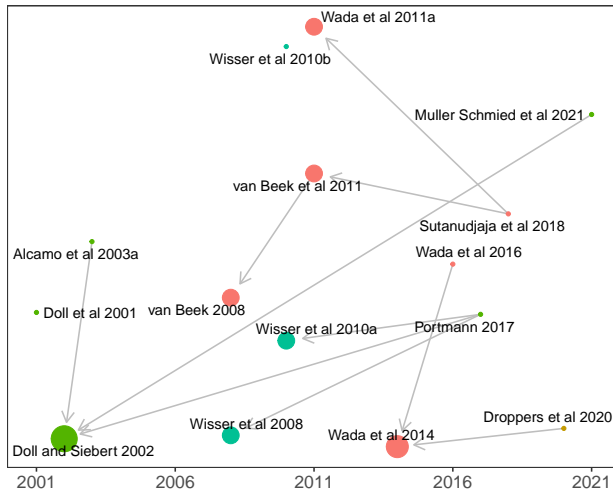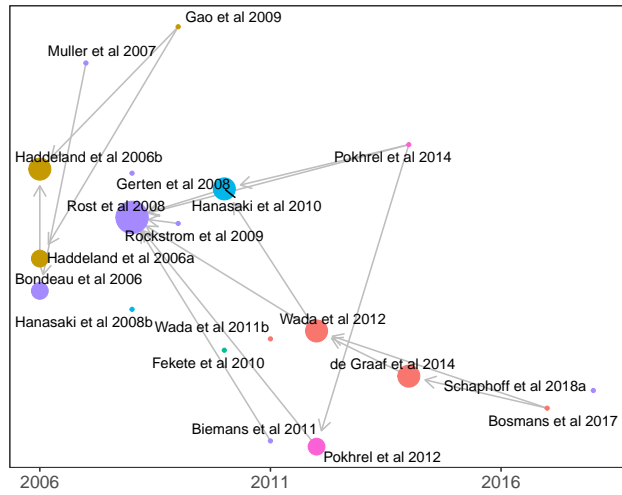
a. Paddy vs nonpaddy

Wada et al 2011a
Wisser et al 2010b
Muller Schmied et al 2021
van Beek et al 2011
Sutanudjaja et al 2018
Wada et al 2016
Alcamo et al 2003a
Doll et al 2001
van Beek 2008
Wisser et al 2010a
Portmann 2017
Wisser et al 2008
Wada et al 2014
Droppers et al 2020
Doll and Siebert 2002

2001   2006   2011   2016   2021

b. Unlimited water supply

Gao et al 2009
Muller et al 2007
Haddeland et al 2006b
Pokhrel et al 2014
Gerten et al 2008
Rost et al 2008
Hanasaki et al 2010
Rockstrom et al 2009
Haddeland et al 2006a
Bondeau et al 2006
Hanasaki et al 2008b
Wada et al 2011b
Wada et al 2012
Fekete et al 2010
de Graaf et al 2014
Schaphoff et al 2018a
Bosmans et al 2017
Biemans et al 2011
Pokhrel et al 2012

2006   2011   2016

citation   • 0   ● 1   ● 2   ● 3   ● 4   ● 5

Model   • H08   • LPJmL   • MATSIRO   • PCR–GLOBWB   • VIC   • WaterGAP   • WBMplus

6   Sankey diagram

The following is the code for the sankey diagram of shared assumptions. First column of umpersand can be removed to display the plot in html format.

```
##Load the nodes and links data from the Excel file
#
#nodes <- read_excel("assumptions.xlsx", sheet = "sankey_nodes")
#links <- read_excel("assumptions.xlsx", sheet = "sankey_links")
#
## Adjust link column names and add color groupings
#colnames(links) <- c("source", "target", "value")
#links$group <- as.factor(links$source)
#
## Use a custom, muted color palette for a professional look
#color_scale <- 'd3.scaleOrdinal().range(["#4E79A7", "#A0CBE8", "#F28E2B", "#FFBE7D", "#59A14F", "#8CD1
#
## Create Sankey diagram with adjusted color scale
#p <- sankeyNetwork(
#  Links = links,
#  Nodes = nodes,
#  Source = "source",
#  Target = "target",
#  Value = "value",
#  NodeID = "Node",
#  units = "TWh",
#  fontSize = 12,
#  nodeWidth = 1,
#  margin = list(left = 350, right = 20),
#  LinkGroup = "group",
#  colourScale = color_scale,
#  width = 800,
#  height = 800
```

```
#)
#
## Adjust label positioning in the diagram
#p <- htmlwidgets::onRender(p, "
#  function(el, x) {
#    d3.selectAll('.node text')
#       .style('text-anchor', 'start')
#       .attr('dx', '15');
#  }
#")
#
#p
```