A critique of irrigation efficiency modeling

R code

Arnald Puy

Contents

1 Read in data **3**

2 The model **7**

2.1 Function to create sample matrix . . . 7
2.2 Define distributions . . . 8
2.3 Uncertainty in the proportion of large-scale irrigated areas . . . 10
2.4 Function to create sample matrix and transfrom to appropriate distributions . . . 10
2.5 Run the model . . . 11
2.6 Define settings . . . 11
2.7 Run model . . . 12
2.8 Extract model output . . . 12

3 Uncertainty analysis **18**

4 Sensitivity analysis **20**

```r
# 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("data.table", "tidyverse", "sensobol", "wesanderson",
               "cowplot", "parallel", "foreach", "doParallel",
               "countrycode", "ggridges", "scales", "overlapping"))

# 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),
          legend.position = "top",
          strip.background = element_rect(fill = "white"),
          plot.margin = margin(t = 0, r = 0.3, b = 0, l = 0.3, unit ="cm"))
}

# Set checkpoint

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

checkpoint("2021-08-02",
           R.version ="4.0.3",
           checkpointLocation = getwd())
```

1 Read in data

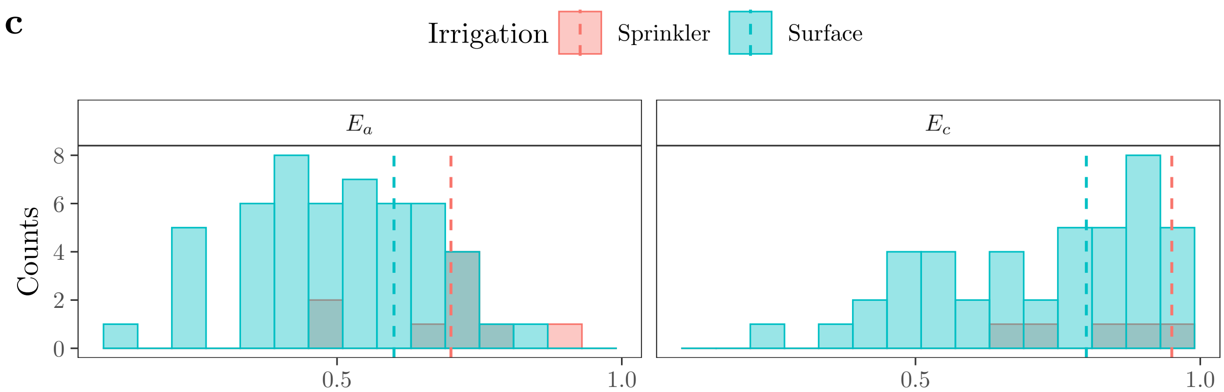
```
# READ IN DATA ----------------------------------------------------------------------

# Rohwer data
rohwer <- fread("rohwer_data_all.csv")
rohwer[rohwer == ""] <- NA
rohwer <- rohwer[, Large_fraction:= Large_fraction / 100]

# Bos data
bos <- fread("bos_data.csv")
bos <- bos[, Scale := ifelse(Irrigated_area < 10000, "<10.000 ha", ">10.000 ha")]

# Solley data (USA)
usa.dt <- fread("usa_efficiency.csv")
usa.dt <- usa.dt[, Efficiency:= consumptive.use / total.withdrawal]

# FAO 1997 data (Irrigation potential in Africa)
fao_dt <- fread("fao_1997.csv")
fao_dt <- fao_dt[, Efficiency:= Efficiency / 100]

# Create data set with E_a values as defined by Rohwer
bos.rohwer.ea <- data.table("Irrigation" = c("Surface", "Sprinkler"),
                            "Value" = c(0.6, 0.7),
                            "variable" = "E_a")

# Create data set with E_c values as defined by Rohwer
bos.rohwer.ec <- data.table("Irrigation" = c("Surface", "Sprinkler"),
                            "Value" = c(0.8, 0.95),
                            "variable" = "E_c")

bos.rohwer.all <- rbind(bos.rohwer.ec, bos.rohwer.ea)

# As a function of scale
bos.rohwer.mf.ec <- data.table("Scale" = c("<10.000 ha", ">10.000 ha"),
                               "Value" = c(0.85, 0.59),
                               "variable" = "E_c")

bos.rohwer.mf.ed <- data.table("Scale" = c("<10.000 ha", ">10.000 ha"),
                               "Value" = c(0.81, 0.72),
                               "variable" = "E_d")

bos.rohwer.mf.all <- rbind(bos.rohwer.mf.ec, bos.rohwer.mf.ed)
```

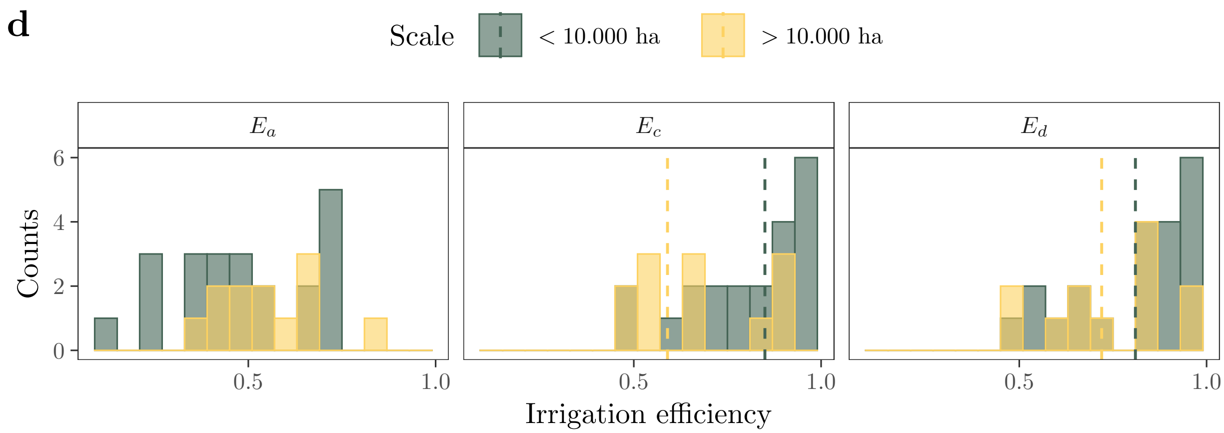
```
# PLOT ------------------------------------------------------------------------------

# Field and conveyance efficiency ----------------
```

```
efficiencies_labeller <- c("E_c" = "$E_c$",
                           "E_a" = "$E_a$")

a <- bos %>%
  melt(., measure.vars = c("E_a", "E_c")) %>%
  ggplot(., aes(value, fill = Irrigation, color = Irrigation)) +
  geom_histogram(position = "identity", alpha = 0.4, bins = 15) +
  facet_wrap(~variable, labeller = as_labeller(efficiencies_labeller)) +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  geom_vline(data = bos.rohwer.all, aes(xintercept = Value,
                                         color = Irrigation,
                                         group = variable),
             lty = 2,
             size = 1) +
  labs(x = "", y = "Counts") +
  theme_AP()

# As a function of scale -----------------------

efficiencies_labeller <- c("E_c" = "$E_c$",
                           "E_a" = "$E_a$",
                           "E_d" = "$E_d$")

b <- melt(bos, measure.vars = c("E_c", "E_a", "E_d")) %>%
  na.omit() %>%
  ggplot(., aes(value, fill = Scale, color = Scale)) +
  geom_histogram(bins = 15, position = "identity", alpha = 0.6) +
  labs(x = "Irrigation efficiency", y = "Counts") +
  facet_wrap(~ variable, labeller = as_labeller(efficiencies_labeller)) +
  geom_vline(data = bos.rohwer.mf.all, aes(xintercept = Value,
                                           color = Scale,
                                           group = variable),
             lty = 2,
             size = 1) +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  scale_color_manual(values = wes_palette(2, name = "Chevalier1"),
                     name = "Scale",
                     labels = c("$<10.000$ ha", "$>10.000$ ha")) +
  scale_fill_manual(values = wes_palette(2, name = "Chevalier1"),
                    name = "Scale",
                    labels = c("$<10.000$ ha", "$>10.000$ ha")) +
  theme_AP()

bottom <- plot_grid(a, b, ncol = 1, labels = c("c", "d"))
```

```
## Warning: Removed 74 rows containing non-finite values (stat_bin).
```

```
bottom
```

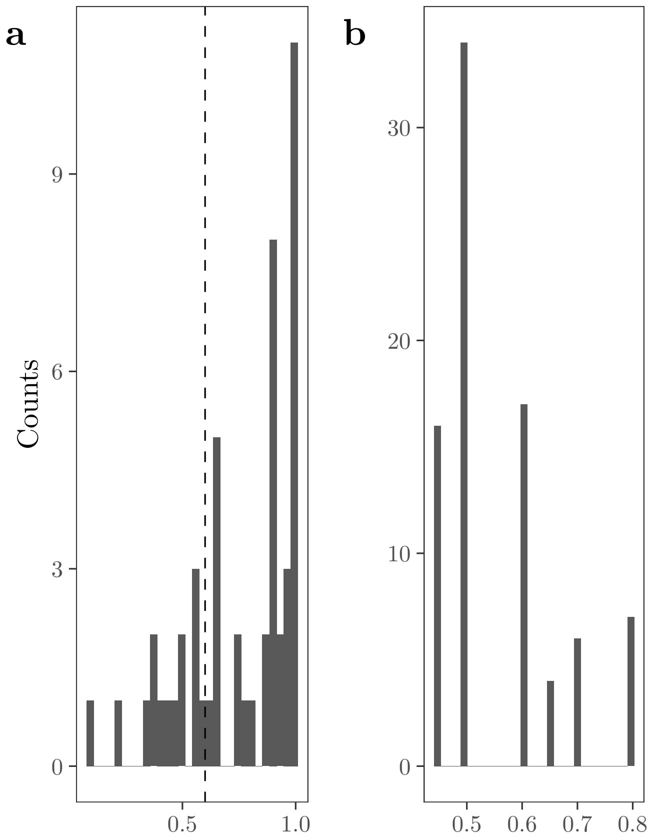




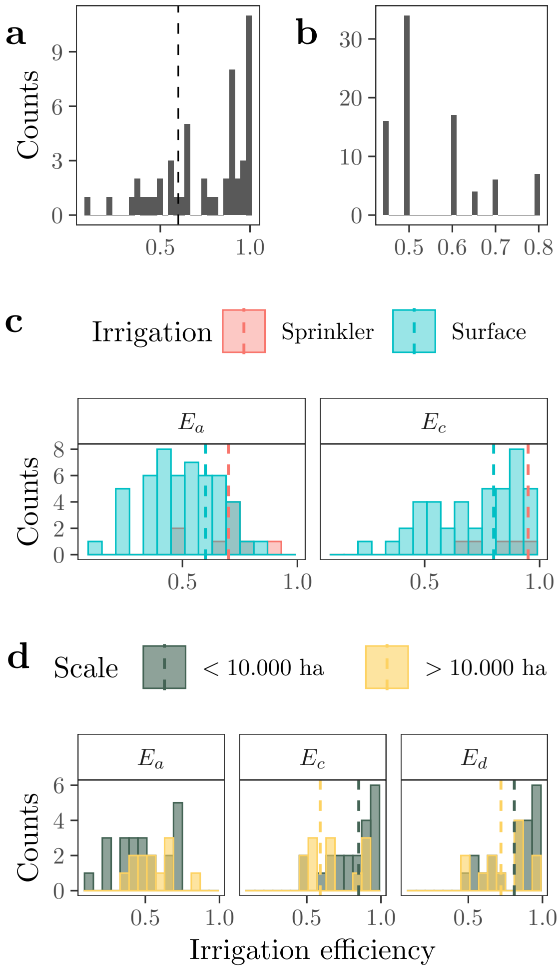

```
# PLOT USA AND AFRICA ----------------------------------------------------------

c1 <- ggplot(usa.dt, aes(Efficiency)) +
  geom_histogram() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  geom_vline(xintercept = 0.6, lty = 2) +
  labs(x = "", y = "Counts") +
  theme_AP()

d1 <- ggplot(fao_dt, aes(Efficiency)) +
  geom_histogram() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  labs(x = "", y = "") +
  theme_AP()

top <- cowplot::plot_grid(c1, d1, ncol = 2, labels = "auto")
```

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

## Warning: Removed 3 rows containing non-finite values (stat_bin).

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

```
top
```




```
# PLOT MERGED -----------------------------------------------------------------------

plot_grid(top, bottom, ncol = 1, rel_heights = c(0.3, 0.7))
```



2 The model

2.1 Function to create sample matrix

```
# CREATE FUNCTION TO DESIGN SAMPLE MATRIX ------------------------------------------

params_algo <- list(
  "Surface" = c("Ea_surf", "Ec_surf", "Proportion_large", "m", "r_L"),
  "Sprinkler" = c("Ea_sprinkler", "Ec_sprinkler", "m"),
  "Micro" = c("Ea_micro", "Ec_micro", "m"),
  "Mixed" = c("Ea_surf", "Ea_sprinkler", "Ec_surf", "Ec_sprinkler",
              "Proportion_large", "m", "r_L")
)

params_fun <- function(IFT) {
  out <- params_algo[[IFT]]
  return(out)
}
```

```
sample_matrix_fun <- function(IFT) {
  params <- params_fun(IFT = IFT)
  mat <- sensobol::sobol_matrices(N = N, params = params)
  out <- list(params, mat)
  names(out) <- c("parameters", "matrix")
  return(out)
}
```

2.2 Define distributions

```
# DEFINE TRUNCATED DISTRIBUTIONS ---------------------------------------------------

# EA SURFACE ---------

Ea.surface <- bos[Irrigation == "Surface"][, .(min = min(E_a, na.rm = TRUE),
                                               max = max(E_a, na.rm = TRUE))]
shape <- 3.502469
scale <- 0.5444373
minimum <- Ea.surface$min
maximum <- Ea.surface$max
weibull_dist <- sapply(c(minimum, maximum), function(x)
  pweibull(x, shape = shape, scale = scale))

# EC SURFACE --------

Ec.surface <- bos[Irrigation == "Surface"][, .(min = min(E_c, na.rm = TRUE),
                                               max = max(E_c, na.rm = TRUE))]
shape1 <- 5.759496
shape2 <- 1.403552
minimum.beta <- Ec.surface$min
maximum.beta <- Ec.surface$max
beta_dist <- sapply(c(minimum.beta, maximum.beta), function(x)
  pbeta(x, shape1 = shape1, shape2 = shape2))

# EA SPRINKLER --------

Ea.sprinkler <- bos[Irrigation == "Sprinkler"][, .(min = min(E_a, na.rm = TRUE),
                                               max = max(E_a, na.rm = TRUE))]
shape.spr <- 6.9913711
scale.spr <- 0.7451178
minimum.spr <- Ea.sprinkler$min
maximum.spr <- Ea.sprinkler$max
weibull_dist_spr <- sapply(c(minimum.spr, maximum.spr), function(x)
  pweibull(x, shape = shape.spr, scale = scale.spr))

# MANAGEMENT FACTOR (m) -----
```

```r
shape1.m <- 5.759496
shape2.m <- 1.403552
minimum.m <- 0.65
maximum.m <- 1
beta_dist.m <- sapply(c(minimum.m, maximum.m), function(x)
  pbeta(x, shape1 = shape1.m, shape2 = shape2.m))
```

```r
# FUNCTION TO TRANSFORM TO APPROPRIATE DISTRIBUTIONS -------------------------------

distributions_fun <- list(

  # SURFACE IRRIGATION
  # --------------------------

  "Ea_surf" = function(x) {

    out <- qunif(x, weibull_dist[[1]], weibull_dist[[2]])
    out <- qweibull(out, shape, scale)
  },

  "Ec_surf" = function(x)  {

    out <- qunif(x, beta_dist[[1]], beta_dist[[2]])
    out <- qbeta(out, shape1, shape2)
  },

  # SPRINKLER IRRIGATION
  # -------------------------

  "Ea_sprinkler" = function(x) {

    out <- qunif(x, weibull_dist_spr[[1]], weibull_dist_spr[[2]])
    out <- qweibull(out, shape.spr, scale.spr)
  },

  "Ec_sprinkler" = function(x) qunif(x, 0.64, 0.96),

  # MICRO (DRIP) IRRIGATION
  # -------------------------

  "Ea_micro" = function(x) out <- qunif(x, 0.75, 0.95),
  "Ec_micro" = function(x) out <- qunif(x, 0.9, 0.95),

  # PROPORTION LARGE
  # ------------------------
  "Proportion_large" = function(x) x,
```

```
  # MANAGEMENT FACTOR
  # -------------------------

  "m" = function(x) {
    out <- qunif(x, beta_dist.m[[1]], beta_dist.m[[2]])
    out <- qbeta(out, shape1.m, shape2.m)
    },

  # REDUCTION IN MANAGEMENT FACTOR DUE TO LARGE-SCALE
  # ---------------------------
  "r_L" = function(x) qunif(x, 0.1, 0.5)
)
```

2.3 Uncertainty in the proportion of large-scale irrigated areas

```
# DEFINE THE UNCERTAINTY IN THE LARGE FRACTION AT THE COUNTRY LEVEL ----------------

rohwer.frac <- rohwer[, .(Country, Large_fraction)]
rohwer.frac[, `:=` (min = Large_fraction, max = Large_fraction + 0.1)]

countries.list <- split(rohwer.frac, seq(nrow(rohwer.frac)))
names(countries.list) <- rohwer$Country
```

2.4 Function to create sample matrix and transfrom to appropriate distributions

```
# FULL ALGORITHM TO CREATE SAMPLE MATRIX ------------------------------------------

full_sample_matrix <- function(IFT, Country) {
  tmp <- sample_matrix_fun(IFT = IFT)
  mat <- tmp[["matrix"]]
  temp <- colnames(mat)
  mat <- sapply(seq_along(temp), function(x) distributions_fun[[temp[x]]](mat[, x]))
  colnames(mat) <- temp
  countries.frac <- countries.list[[Country]]

  if(IFT == "Surface" | IFT == "Mixed") {

  mat[, "Proportion_large"] <- qunif(mat[, "Proportion_large"],
                                     countries.frac$min, countries.frac$max)
  }
  out <- list(tmp$parameters, mat)
  names(out) <- c("parameters", "matrix")
  return(out)
}
```

2.5 Run the model

```r
# FULL MODEL -----------------------------------------------------------------

full_model <- function(IFT, Country, sample.size, R) {

  tmp <- full_sample_matrix(IFT = IFT, Country = Country)
  mat <- tmp$matrix

  if(IFT == "Surface") {

    Mf <- mat[, "m"] - mat[, "r_L"] * mat[, "Proportion_large"]
    y <- mat[, "Ea_surf"] * mat[, "Ec_surf"] * Mf

  } else if(IFT == "Sprinkler") {

    Mf <- mat[, "m"]
    y <- mat[, "Ea_sprinkler"] * mat[, "Ec_sprinkler"] * Mf

  } else if(IFT == "Mixed") {

    Mf.surf <- mat[, "m"] - mat[, "r_L"] * mat[, "Proportion_large"]
    y.surf <- mat[, "Ea_surf"] * mat[, "Ec_surf"] * Mf.surf

    Mf.sprink <- mat[, "m"]
    y.sprink <- mat[, "Ea_sprinkler"] * mat[, "Ec_sprinkler"] * Mf.sprink

    y <- 0.5 * y.surf + 0.5 * y.sprink

  } else {
    Mf <- mat[, "m"]
    y <- mat[, "Ea_micro"] * mat[, "Ec_micro"] * Mf
  }

  ind <- sobol_indices(N = sample.size, Y = y, params = tmp$parameters,
                       boot = TRUE, R = R)
  out <- list(y, ind)
  names(out) <- c("output", "indices")
  return(out)
}
```

2.6 Define settings

```r
# DEFINE SETTINGS ------------------------------------------------------------

N <- 2^13
R <- 10^2
```

2.7 Run model

```
# RUN MODEL -------------------------------------------------------------------------

y <- mclapply(1:nrow(rohwer), function(x)
  full_model(IFT = rohwer[[x, "IFT"]],
             Country = rohwer[[x, "Country"]],
             sample.size = N,
             R = R),
  mc.cores = detectCores() * 0.75)
```

2.8 Extract model output

```
# EXTRACT MODEL OUTPUT -----------------------------------------------------------

output <- lapply(y, function(x) x[["output"]][1:(2 * N)])
names(output) <- rohwer$Country
tmp <- lapply(output, data.table) %>%
  rbindlist(., idcol = "Country") %>%
  merge(., rohwer[, .(Country, IFT)], all.x = TRUE)

tmp <- tmp[, Continent:= countrycode(tmp[, Country], origin = "country.name",
                                     destination = "continent")] %>%
  .[, IFT:= factor(IFT, levels = c("Surface", "Sprinkler", "Micro", "Mixed"))]
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest,
```

```
# COMPUTE RANGES ------------------------------------------------------------------

calc <- tmp[, .(min = min(V1), max = max(V1)), .(Continent, Country)] %>%
  .[, .(range = max - min), .(Continent, Country)] %>%
  .[order(range)]

print(calc, n = Inf)
```

```
##      Continent                 Country     range
##   1:      Asia                  Cyprus 0.4427170
##   2:      Asia                  Israel 0.4427170
##   3:      Asia                  Jordan 0.4427170
##   4:      Asia                    Oman 0.4427170
##   5:      Asia    United Arab Emirates 0.4427170
##   6:      Asia                    Iraq 0.5217747
##   7:    Europe                   Italy 0.5217747
##   8:    Europe             Netherlands 0.5217747
##   9:    Europe                 Albania 0.5408817
##  10:    Africa                 Algeria 0.5408817
##  11:  Americas                  Brazil 0.5408817
##  12:  Americas                  Canada 0.5408817
##  13:  Americas                    Cuba 0.5408817
```

```
## 14:    Europe                     France 0.5408817
## 15:      Asia                 Kazakhstan 0.5408817
## 16:    Africa                 Mozambique 0.5408817
## 17:      Asia               Saudi Arabia 0.5408817
## 18:      Asia                  Sri Lanka 0.5408817
## 19:  Americas              United States 0.5408817
## 20:    Africa                      Benin 0.5593234
## 21:    Africa                   Botswana 0.5593234
## 22:      Asia                     Brunei 0.5593234
## 23:    Africa               Burkina Faso 0.5593234
## 24:    Africa                Ivory Coast 0.5593234
## 25:      Asia                     Kuwait 0.5593234
## 26:      Asia                    Lebanon 0.5593234
## 27:    Africa                      Libya 0.5593234
## 28:    Africa                    Namibia 0.5593234
## 29:      Asia                      Qatar 0.5593234
## 30:    Europe                   Slovenia 0.5593234
## 31:    Africa               South Africa 0.5593234
## 32:    Europe                      Spain 0.5593234
## 33:    Africa                  Swaziland 0.5593234
## 34:    Africa                   Tanzania 0.5593234
## 35:    Africa                    Tunisia 0.5593234
## 36:    Africa                     Zambia 0.5593234
## 37:    Africa                   Zimbabwe 0.5593234
## 38:    Europe                    Austria 0.5928658
## 39:    Europe                    Belgium 0.5928658
## 40:      <NA>                   Byelarus 0.5928658
## 41:    Europe                    Croatia 0.5928658
## 42:    Europe             Czech Republic 0.5928658
## 43:    Europe                    Denmark 0.5928658
## 44:    Europe                    Finland 0.5928658
## 45:    Europe                    Germany 0.5928658
## 46:    Europe                     Greece 0.5928658
## 47:    Europe                    Hungary 0.5928658
## 48:    Europe                     Latvia 0.5928658
## 49:    Europe                  Lithuania 0.5928658
## 50:    Europe                 Luxembourg 0.5928658
## 51:    Europe                  Macedonia 0.5928658
## 52:    Africa                     Malawi 0.5928658
## 53:    Europe                    Romania 0.5928658
## 54:    Europe                     Russia 0.5928658
## 55:    Europe                   Slovakia 0.5928658
## 56:    Europe                     Sweden 0.5928658
## 57:    Europe                Switzerland 0.5928658
## 58:    Europe                    Ukraine 0.5928658
## 59:    Europe             United Kingdom 0.5928658
## 60:      Asia                 Azerbaijan 0.6754616
## 61:  Americas                      Chile 0.6754616
```

```
##  62:  Americas                     Mexico 0.6754616
##  63:      Asia                   Pakistan 0.6754616
##  64:    Africa                      Sudan 0.6754616
##  65:      Asia                      Syria 0.6754616
##  66:      Asia                 Tajikistan 0.6754616
##  67:      Asia                   Thailand 0.6754616
##  68:      Asia                 Uzbekistan 0.6754616
##  69:  Americas                  Venezuela 0.6754616
##  70:      Asia                    Vietnam 0.6754616
##  71:      Asia                Afghanistan 0.7043999
##  72:  Americas                  Argentina 0.7043999
##  73:   Oceania                  Australia 0.7043999
##  74:      Asia                 Bangladesh 0.7043999
##  75:    Europe                   Bulgaria 0.7043999
##  76:      Asia                      China 0.7043999
##  77:  Americas                   Colombia 0.7043999
##  78:  Americas                    Ecuador 0.7043999
##  79:    Africa                      Egypt 0.7043999
##  80:    Africa                   Ethiopia 0.7043999
##  81:      Asia                    Georgia 0.7043999
##  82:      Asia                      India 0.7043999
##  83:      Asia                  Indonesia 0.7043999
##  84:      Asia                       Iran 0.7043999
##  85:      Asia                      Japan 0.7043999
##  86:      Asia                 Kyrgyzstan 0.7043999
##  87:    Europe                    Moldova 0.7043999
##  88:      Asia                      Nepal 0.7043999
##  89:    Africa                    Nigeria 0.7043999
##  90:      Asia                North Korea 0.7043999
##  91:      Asia                Philippines 0.7043999
##  92:    Europe                   Portugal 0.7043999
##  93:      Asia                     Turkey 0.7043999
##  94:      Asia               Turkmenistan 0.7043999
##  95:    Africa                     Angola 0.7527662
##  96:      Asia                    Armenia 0.7527662
##  97:  Americas                     Belize 0.7527662
##  98:      Asia                     Bhutan 0.7527662
##  99:  Americas                    Bolivia 0.7527662
## 100:    Europe   Bosnia and Herzegovina 0.7527662
## 101:      Asia                      Burma 0.7527662
## 102:    Africa                    Burundi 0.7527662
## 103:      Asia                   Cambodia 0.7527662
## 104:    Africa                   Cameroon 0.7527662
## 105:    Africa Central African Republic 0.7527662
## 106:    Africa                       Chad 0.7527662
## 107:    Africa                      Congo 0.7527662
## 108:  Americas                 Costa Rica 0.7527662
## 109:    Africa                   Djibouti 0.7527662
```

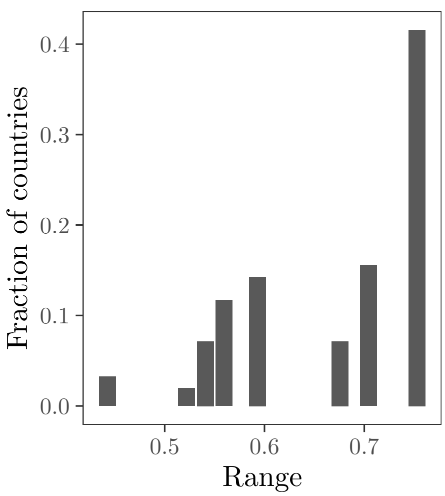
```
## 110:  Americas       Dominican Republic 0.7527662
## 111:  Americas              El Salvador 0.7527662
## 112:    Africa        Equatorial Guinea 0.7527662
## 113:    Africa                  Eritrea 0.7527662
## 114:  Americas            French Guiana 0.7527662
## 115:    Africa                    Gabon 0.7527662
## 116:    Africa                   Gambia 0.7527662
## 117:    Africa                    Ghana 0.7527662
## 118:  Americas                Guatemala 0.7527662
## 119:    Africa                   Guinea 0.7527662
## 120:    Africa            Guinea-Bissau 0.7527662
## 121:  Americas                   Guyana 0.7527662
## 122:  Americas                    Haiti 0.7527662
## 123:  Americas                 Honduras 0.7527662
## 124:  Americas                  Jamaica 0.7527662
## 125:    Africa                    Kenya 0.7527662
## 126:      Asia                     Laos 0.7527662
## 127:    Africa                  Lesotho 0.7527662
## 128:    Africa                  Liberia 0.7527662
## 129:    Africa               Madagascar 0.7527662
## 130:      Asia                 Malaysia 0.7527662
## 131:    Africa                     Mali 0.7527662
## 132:    Africa               Mauritania 0.7527662
## 133:      Asia                 Mongolia 0.7527662
## 134:    Africa                  Morocco 0.7527662
## 135:   Oceania              New Zealand 0.7527662
## 136:  Americas                Nicaragua 0.7527662
## 137:    Africa                    Niger 0.7527662
## 138:    Europe                   Norway 0.7527662
## 139:  Americas                   Panama 0.7527662
## 140:   Oceania         Papua New Guinea 0.7527662
## 141:  Americas                 Paraguay 0.7527662
## 142:  Americas                     Peru 0.7527662
## 143:    Europe                   Poland 0.7527662
## 144:  Americas              Puerto Rico 0.7527662
## 145:    Africa                   Rwanda 0.7527662
## 146:    Africa                  Senegal 0.7527662
## 147:    Europe                   Serbia 0.7527662
## 148:    Africa             Sierra Leone 0.7527662
## 149:    Africa                  Somalia 0.7527662
## 150:      Asia              South Korea 0.7527662
## 151:  Americas                 Suriname 0.7527662
## 152:    Africa                     Togo 0.7527662
## 153:  Americas                 Trinidad 0.7527662
## 154:    Africa                   Uganda 0.7527662
## 155:  Americas                  Uruguay 0.7527662
## 156:    Africa           Western Sahara 0.7527662
## 157:      Asia                    Yemen 0.7527662
```

```
## 158:    Africa                          Zaire 0.7527662
##      Continent                       Country     range
```

```
rang <- calc[, .(total = .N), range] %>%
  .[, N.countries:= 154] %>%
  .[, fraction:= total / N.countries]

print(rang)
```

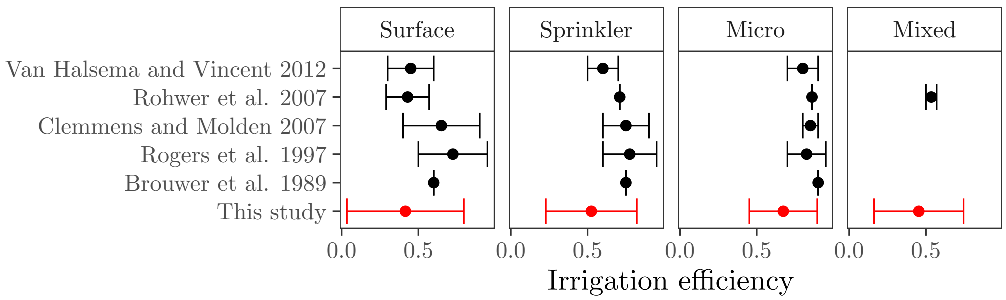
```
##        range total N.countries   fraction
## 1: 0.4427170     5         154 0.03246753
## 2: 0.5217747     3         154 0.01948052
## 3: 0.5408817    11         154 0.07142857
## 4: 0.5593234    18         154 0.11688312
## 5: 0.5928658    22         154 0.14285714
## 6: 0.6754616    11         154 0.07142857
## 7: 0.7043999    24         154 0.15584416
## 8: 0.7527662    64         154 0.41558442
```

```
ggplot(rang, aes(range, fraction)) +
  geom_bar(stat = "identity") +
  labs(x = "Range", y = "Fraction of countries") +
  theme_AP()
```




```
# COMPARE RANGES ------------------------------------------------------------------

ranges_empirical <- tmp[, .(higher = max(V1), lower = min(V1)), IFT] %>%
  .[, Study:= "This study"]

ranges_efficiencies <- fread("ranges_efficiencies.csv")

rbind(ranges_empirical, ranges_efficiencies)[, mean.value:= (higher + lower) / 2] %>%
  .[, Study:= factor(Study, levels = c("This study",
                                       "Brouwer et al. 1989",
                                       "Rogers et al. 1997",
```

```
                                              "Clemmens and Molden 2007",
                                              "Rohwer et al. 2007",
                                              "Van Halsema and Vincent 2012"))] %>%
  ggplot(., aes(mean.value, Study, color = ifelse(Study == "This study", "red", "black"))) +
  geom_point() +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  geom_errorbar(aes(xmin = lower, xmax = higher)) +
  scale_color_identity() +
  facet_wrap(~IFT, ncol = 4) +
  labs(x = "Irrigation efficiency", y = "") +
  theme_AP()
```




```
# CHECK OVERLAP ------------------------------------------------------------------

dd <- tmp[!Continent == "Oceania"] %>%
  split(., .$Continent, drop = TRUE)

overlap.dt <- lapply(dd, function(x) split(x, x$IFT, drop = TRUE)) %>%
  lapply(., function(x) lapply(x, function(y) y[, V1])) %>%
  lapply(., function(x) overlap(x)$OV)

overlap.dt
```

```
## $Africa
## Surface-Sprinkler     Surface-Mixed   Sprinkler-Mixed
##         0.3287201         0.5081015         0.5222135
##
## $Americas
## Surface-Mixed
##     0.4933317
##
## $Asia
## Surface-Micro Surface-Mixed   Micro-Mixed
##    0.05230668    0.43345515    0.07617371
##
## $Europe
## Surface-Sprinkler     Surface-Mixed   Sprinkler-Mixed
##         0.3123871         0.5342126         0.4591130
```

3 Uncertainty analysis

```r
# PLOT UNCERTAINTY -------------------------------------------------------------

plot_ggridges <- function(dt, Cont) {
  pp <- ggplot(dt[Continent == Cont], aes(x = V1, y = fct_reorder(Country, V1),
                                          fill = IFT), alpha = 0.5) +
    geom_density_ridges(scale = 2) +
    labs(x = "Irrigation efficiency", y = "") +
    facet_wrap(~Continent) +
    theme_AP() +
    theme(legend.position = "none")
  return(pp)
}

a <- plot_ggridges(dt = tmp, Cont = "Africa") +
  scale_fill_manual(labels = c("Surface", "Sprinkler", "Mixed"),
                    values = c("#D8B70A", "#02401B", "#81A88D"),
                    name = "Irrigation")

b <- plot_ggridges(dt = tmp, Cont = "Americas") +
  scale_fill_manual(labels = c("Surface", "Mixed"),
                    values = c("#D8B70A", "#81A88D"),
                    name = "Irrigation")

c <- plot_ggridges(dt = tmp, Cont = "Asia") +
  scale_fill_manual(labels = c("Surface", "Micro", "Mixed"),
                    values = c("#D8B70A", "#A2A475", "#81A88D"),
                    name = "Irrigation")

d <- plot_ggridges(dt = tmp, Cont = "Europe") +
  scale_fill_manual(labels = c("Surface", "Sprinkler", "Mixed"),
                    values = c("#D8B70A", "#02401B", "#81A88D"),
                    name = "Irrigation")


legend.africa <- get_legend(a + theme(legend.position = "top"))
```

```
## Picking joint bandwidth of 0.014
```

```r
legend.asia <- get_legend(c + theme(legend.position = "top"))
```

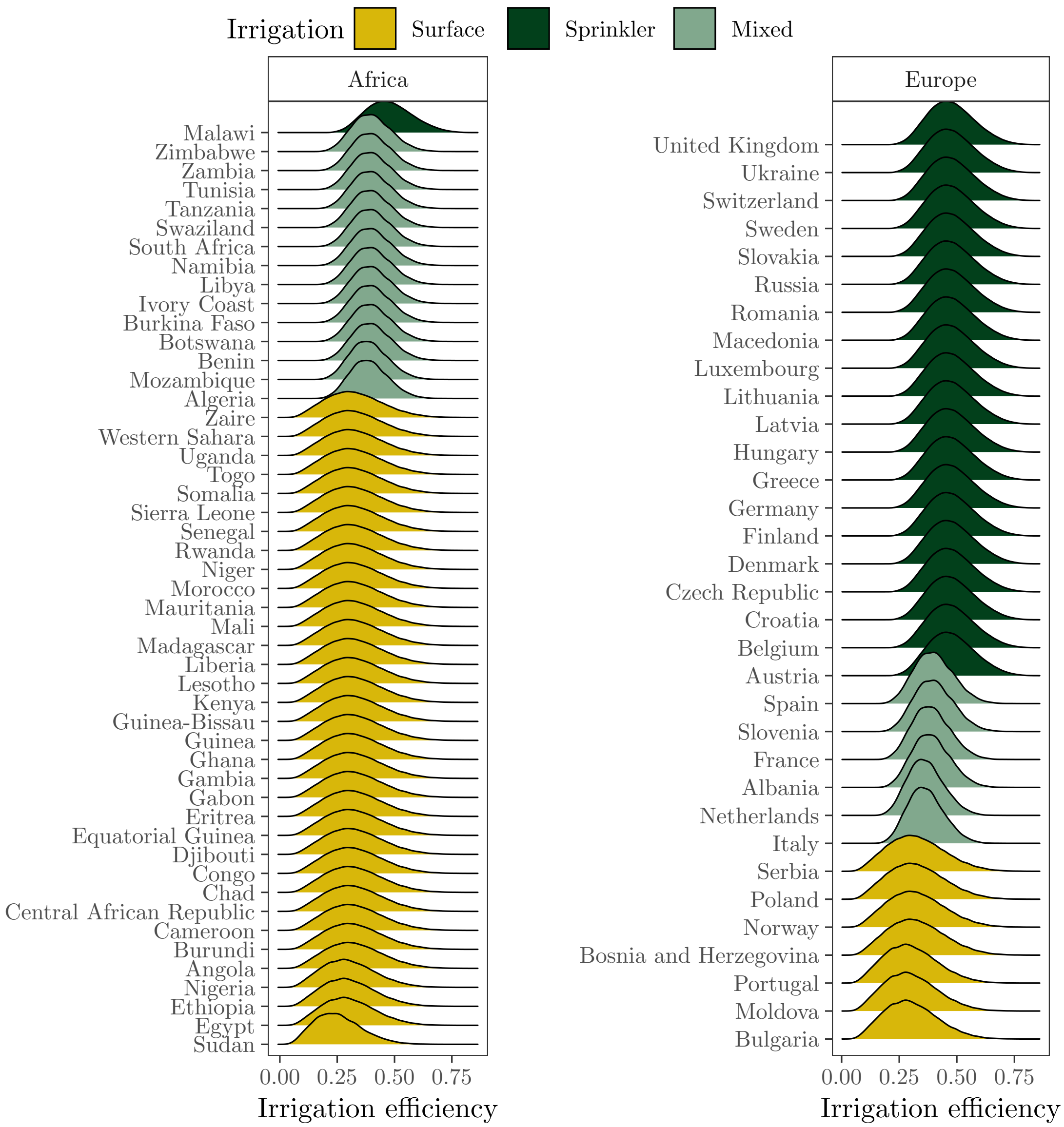
```
## Picking joint bandwidth of 0.0134
```

```r
# MERGE PLOTS ------------------------------------------------------------------

bottom <- plot_grid(a, d, ncol = 2)
```

```
## Picking joint bandwidth of 0.014
```

```
## Picking joint bandwidth of 0.0129
```

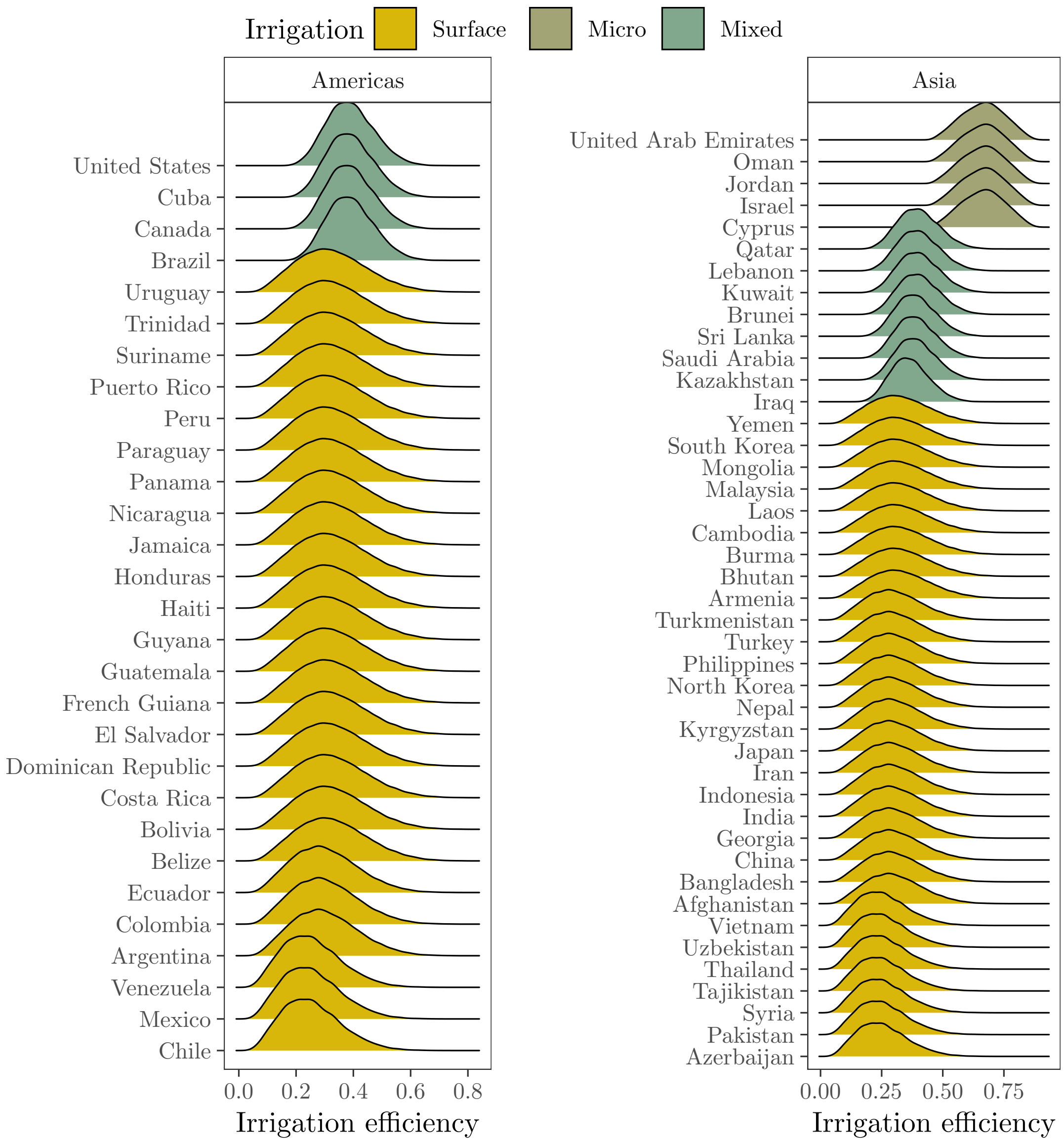
```
plot_grid(legend.africa, bottom, ncol = 1, rel_heights = c(0.05, 0.95))
```




```
bottom <- plot_grid(b, c, ncol = 2)
```

```
## Picking joint bandwidth of 0.0145
## Picking joint bandwidth of 0.0134
```

```
plot_grid(legend.asia, bottom, ncol = 1, rel_heights = c(0.05, 0.95))
```



4 Sensitivity analysis

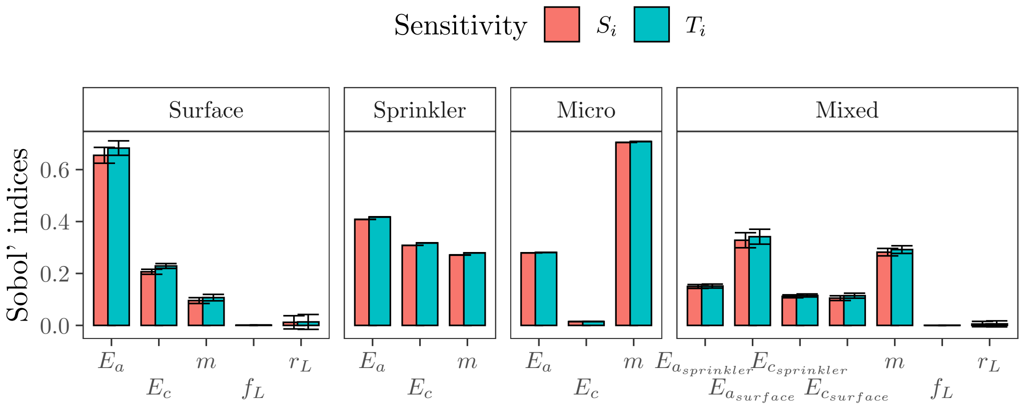
```
# SAMPLE MATRIX DISTRIBUTIONS --------------------------------------------------

# Define labels
label_facets <- c("Ea_surf" = "$E_{a_{\\text{surface}}}$",
                  "Ec_surf" = "$E_{c_{\\text{surface}}}$",
                  "Ea_sprinkler" = "$E_{a_{\\text{sprinkler}}}$",
                  "Ec_sprinkler" = "$E_{c_{\\text{sprinkler}}}$",
                  "Ea_micro" = "$E_{a_{\\text{micro}}}$",
                  "Ec_micro" = "$E_{c_{\\text{micro}}}$",
                  "Proportion_large" = "$f_L$",
                  "m" = "$m$",
                  "r_L" = "$r_L$")
```

```r
# EXTRACT SOBOL' INDICES ---------------------------------------------------------

ind <- lapply(y, function(x) x[["indices"]]$results)
names(ind) <- rohwer$Country
ind <- rbindlist(ind, idcol = "Country")

ind[, Continent:= countrycode(ind[, Country], origin = "country.name",
                              destination = "continent")]
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest,
```

```r
tmp.ift <- split(rohwer, rohwer$IFT)

out <- list()
for(i in names(tmp.ift)) {
  out[[i]] <- ind[Country %in% tmp.ift[[i]][, Country]]
}
```

```r
# PLOT SOBOL' INDICES ------------------------------------------------------------

ind.dt <- rbindlist(out, idcol = "IFT") %>%
  .[, IFT:= factor(IFT, levels = c("Surface", "Sprinkler", "Micro", "Mixed"))]

tmp <- ind.dt[, .(mean = mean(original), sd = sd(original)),
              .(sensitivity, parameters, IFT)]

tmp2 <- tmp[!IFT == "Mixed"][, parameters:= ifelse(parameters == "Ea_surf", "$E_a$",
                                                   ifelse(parameters == "Ec_surf", "$E_c$",
                                                          ifelse(parameters == "Ea_sprinkler",
                                                                 ifelse(parameters == "Ec_sprin
                                                                        ifelse(parameters == "
                                                                               ifelse(paramete

rbind(tmp[IFT == "Mixed"], tmp2) %>%
  ggplot(., aes(parameters, mean, fill = sensitivity), color = black) +
  geom_bar(stat = "identity", position = position_dodge(0.6), color = "black") +
  geom_errorbar(aes(ymin = mean - sd, ymax = mean + sd), position = position_dodge(0.6)) +
  scale_x_discrete(labels = label_facets,
                   guide = guide_axis(n.dodge = 2)) +
  scale_fill_discrete(name = "Sensitivity", labels = c("$S_i$", "$T_i$")) +
  labs(x = "", y = "Sobol' indices") +
  facet_grid(~IFT, space = "free_x", scale = "free_x") +
  theme_AP()
```



Sensitivity
S_i
T_i
Surface
Sprinkler
Micro
Mixed
Sobol' indices
0.0
0.2
0.4
0.6
E_a
E_c
m
f_L
r_L
$E_{a_{sprinkler}}$
$E_{a_{surface}}$
$E_{c_{sprinkler}}$
$E_{c_{surface}}$