Create Package Data Sets

By Jason C. Fisher

Contents

Introduction

This *package vignette* explains the processing steps for creating R data sets in the **wrv** package. Data sets are processed at two levels: *Level 1* is raw data mapped on a uniform space-time grid scale; and *Level 2* results from analyses of Level 1 data.

Software

Extend the capabilities of R by loading the following user-contriubted packages:

```
library(rgdal)   # bindings for the geospatial data abstraction library
library(raster)  # gridded spatial data toolkit
library(RCurl)   # general network client interface
library(rgeos)   # interface to geometry engine
```

Set option to prevent standardization of raster names:

```
rasterOptions(standardnames = FALSE)
```

Input/Output Paths

Package data sets are primarily created from raw data files located on GitHub in the **wrv**-package repository. The top-level uniform resource locator (URL) for these files is specified as:

```
url.git <- "https://raw.githubusercontent.com/jfisher-usgs/wrv/master/inst/extdata/"
```

Raw data files containing land-surface elevations were deemed too large in file size (about 500 MB) to be placed in the package repository. These elevations are part of the National Elevation Dataset (NED) 1/3-arc-second raster and available in a Esri ArcGRID file format. The URL used to retrieve these files is specified as:

```
url.ned <- paste0("http://gisdata.usgs.gov/TDDS/DownloadFile.php",
                  "?TYPE=ned3g_zip&FNAME=n44w115.zip&ORIG=TNM&dlpre=2-2")
```

Specify the directory path names for output files created from R commands in this pacakge vignette:

```
dir.create(dir.dat <- file.path(getwd(), "data"), showWarnings = FALSE)
```

Space-Time Grid Scale

The length and time dimensions for data sets are in units of meters and days, respectively. Conversion factors are listed with an explanation of how they are used:

```
mm.to.m              <- 0.001          # millimeters to meters
ft.to.m              <- 0.3048         # feet to meters
mi2.to.m2            <- 2589990        # square miles to square meters
af.to.m3             <- 1233.48185532  # acre-feet to cubic meters
ft.per.mo.to.m.per.d <- 0.0100141769   # feet per month to meters per day
af.per.y.to.m3.per.d <- 3.377          # acre-feet per year to cubic meters per day
cfs.to.m3.per.d      <- 2446.57555     # cubic feet per second to cubic meters per day
```

The common coordinate reference system (CRS) applied to all spatial data sets is the Idaho Transverse Mercator projection (IDTM). PROJ.4 projection arguments are used to specify a CRS in R. The CRS that all raw data files are converted into is specified as:

```
crs <- CRS(paste("+proj=tmerc +lat_0=42 +lon_0=-114 +k=0.9996 +x_0=2500000 +y_0=1200000",
                 "+datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0"))
```

The common spatial grid applied to all gridded data sets is composed of 565 rows and 429 columns, and has a constant cell size at 100 meters by 100 meters.

```
ext <- extent(2453200, 2496100, 1344139, 1400639)  # xmin, xmax, ymin, ymax in IDTM
spatial.grid <- raster(crs = crs, ext = ext, resolution = 100)
```

Something...

```
tr.interval <- as.Date(c("1995-01-01", "2010-12-31")) # transient
tr.stress.periods <- seq(tr.interval[1] , tr.interval[2], "month")
yr.mo <- format(tr.stress.periods, "%Y%m")
```

Level 1 Data

Data sets processed at Level 1 are raw data (read from files or specified in this vignette) mapped on a uniform space-time grid scale. Raw data variables that duplicate data, or are unnecessary for model processing or quality assurance purposes, are removed. A few of the Level 1 data sets require supplemental processing steps; descriptions of these steps are included alongside the relevent 'code chunks'. Variable names, for the most part, are maintained between raw and processed data sets.

Tables

Canal Seepage (canal.seep)

Canal seepage in the Big Wood River (fig. 1).

```
file <- wrv::DownloadFile(paste0(url.git, "canal/canal.seep.csv"))
canal.seep <- read.csv(file, strip.white = TRUE)
save(canal.seep, file = file.path(dir.dat, "canal.seep.rda"), compress = "xz")
```

Combined Surface Water Irrigation Diversions (comb.sw.irr)

Something

```
file <- wrv::DownloadFile(paste0(url.git, "div/comb.sw.irr.csv"))
comb.sw.irr <- read.csv(file, strip.white = TRUE)
comb.sw.irr$Pdate <- as.Date(comb.sw.irr$Pdate, format = "%m/%d/%Y")
comb.sw.irr$MaxDivRate <- comb.sw.irr$MaxDivRate * cfs.to.m3.per.d
save(comb.sw.irr, file = file.path(dir.dat, "comb.sw.irr.rda"), compress = "xz")
```

Evapotranspiration Methods (et.method)

Evapotranspiration method per season (fig. 4).

```
file <- wrv::DownloadFile(paste0(url.git, "et/et.method.csv"))
et.method <- read.csv(file, strip.white = TRUE)
et.method$YearMonth <- as.character(et.method$YearMonth)
save(et.method, file = file.path(dir.dat, "et.method.rda"), compress = "xz")
```

Groundwater Diversions (div.gw)

Groundwater diversions from wells (fig. 5).

```
file <- wrv::DownloadFile(paste0(url.git, "div/div.gw.csv"))
div.gw <- read.csv(file, strip.white = TRUE)
div.gw$YearMonth <- as.factor(div.gw$YearMonth)
div.gw$GWDiv <- div.gw$GWDiv_af * af.to.m3
div.gw$GWDiv_af <- NULL
div.gw[is.na(div.gw$GWDiv), "GWDiv"] <- 0
save(div.gw, file = file.path(dir.dat, "div.gw.rda"), compress = "xz")
```

Irrigation Efficiency (efficiency)

Something

```
file <- wrv::DownloadFile(paste0(url.git, "irr/efficiency.csv"))
efficiency <- read.csv(file, strip.white = TRUE)
save(efficiency, file = file.path(dir.dat, "efficiency.rda"), compress = "xz")
```

Irrigation Lands for Year (irr.lands.year)

Something

```
file <- wrv::DownloadFile(paste0(url.git, "irr/irr.lands.year.csv"))
irr.lands.year <- read.csv(file, strip.white = TRUE, colClasses = "character")
save(irr.lands.year, file = file.path(dir.dat, "irr.lands.year.rda"), compress = "xz")
```

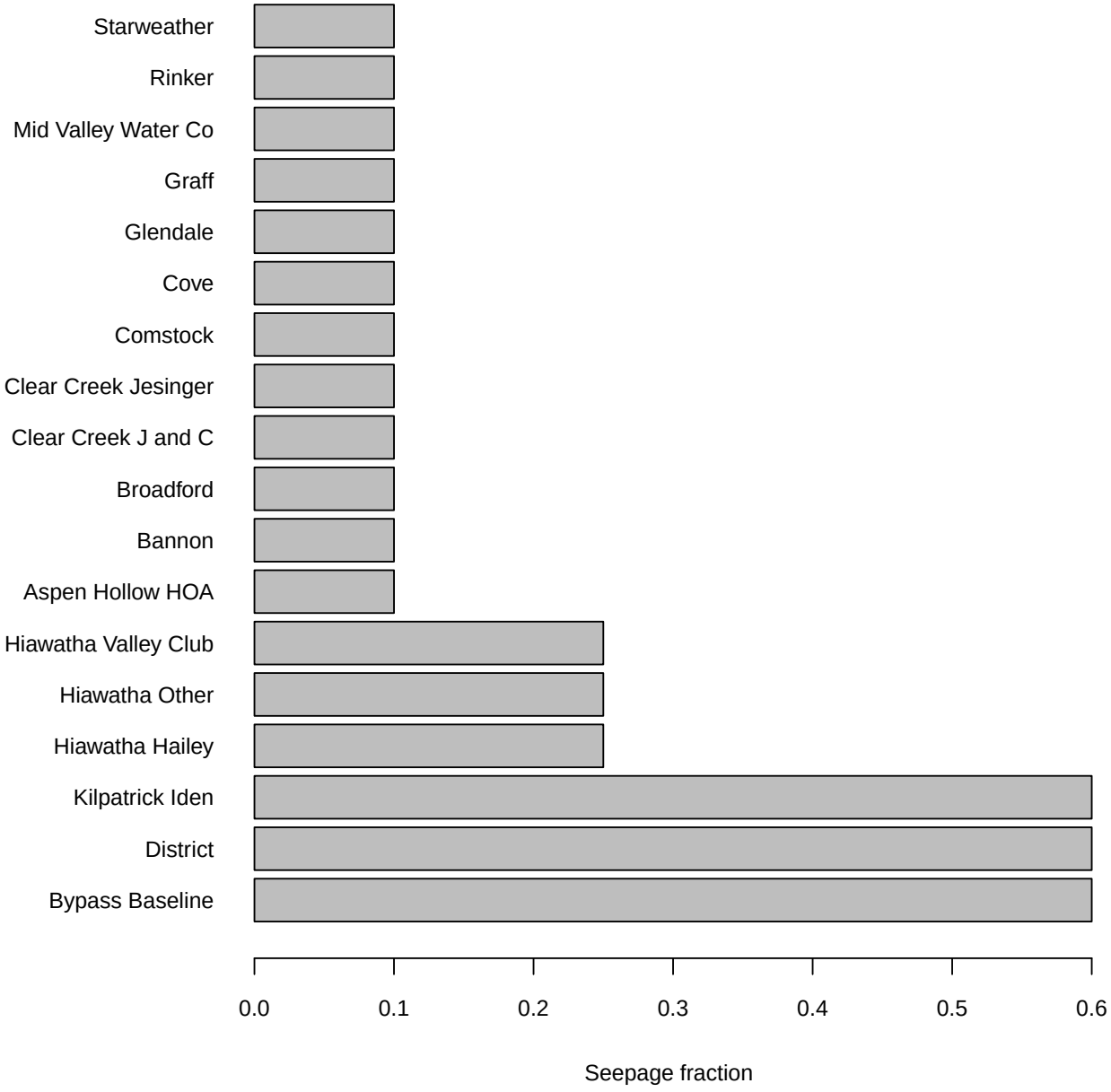

Figure 1: Seepage fraction for canals in the Wood River Valley.

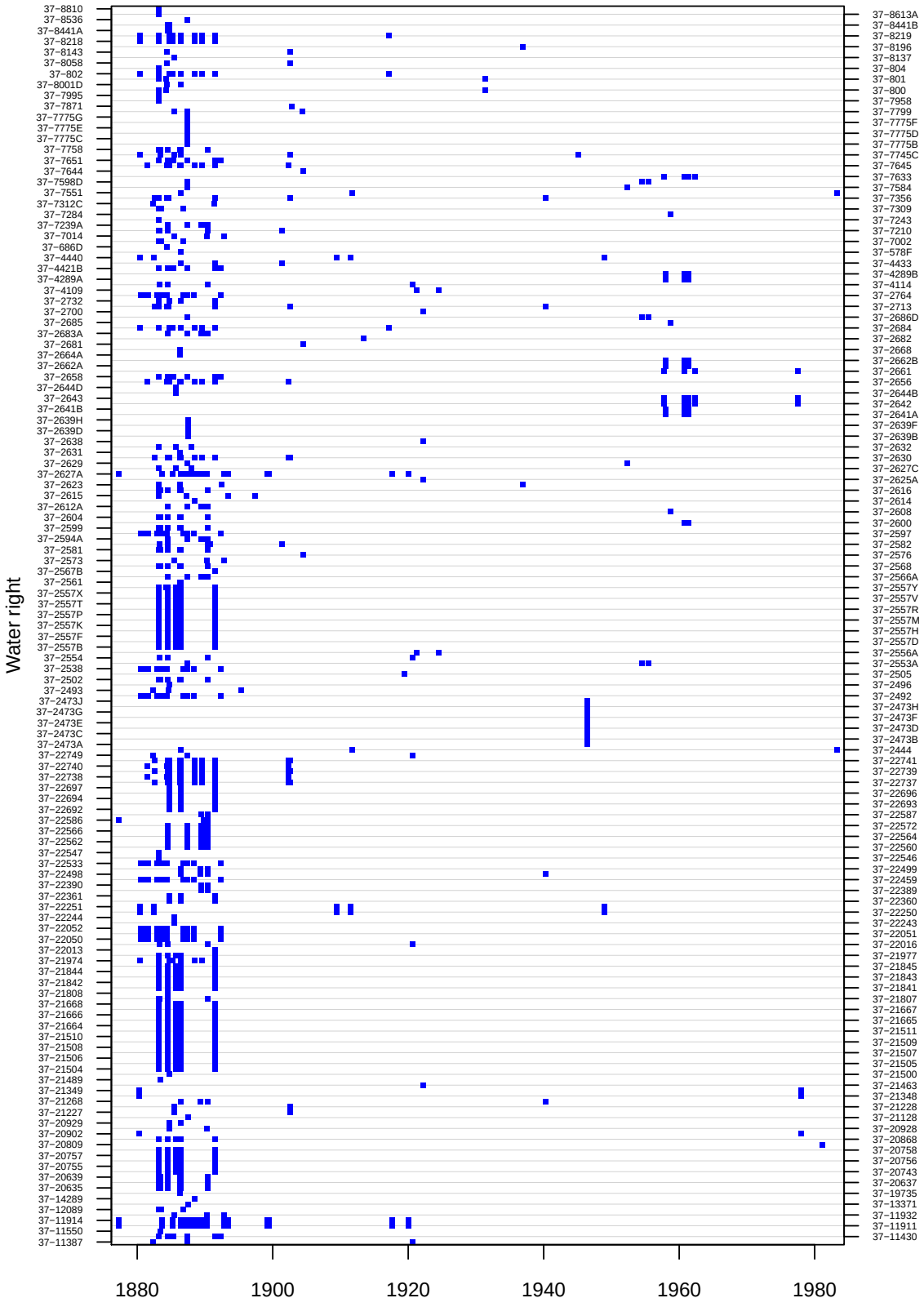

Figure 2: Priority date for surface-water irrigation.

Figure 3: Maximum diversion rate for surface-water irrigation in cubic meters per day.

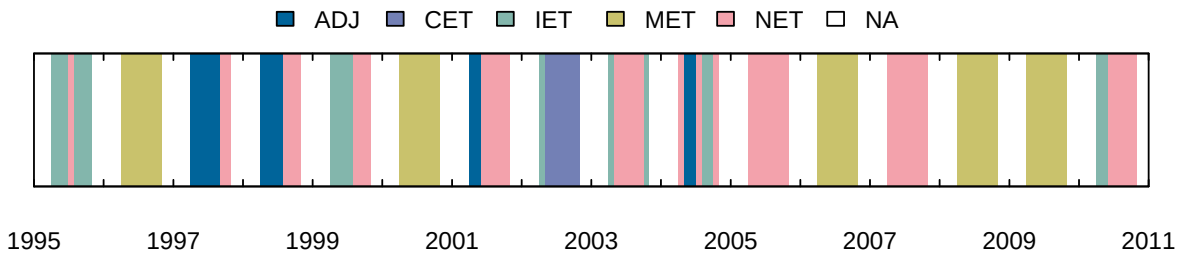

Figure 4: Evapotranspiration method per season.

Precipitation (precipitation)

Adjusted precipitation in the Wood River Valley (fig. 6).

```
file <- wrv::DownloadFile(paste0(url.git, "precip/precipitation.csv"))
precipitation <- read.csv(file, strip.white = TRUE)
precipitation$YearMonth <- as.factor(precipitation$YearMonth)
precipitation$Precip <- precipitation$Precip_ft * ft.to.m
precipitation$Precip_ft <- NULL
save(precipitation, file = file.path(dir.dat, "precipitation.rda"), compress = "xz")
```

Priority Cuts (priority.cuts)

Something

```
file <- wrv::DownloadFile(paste0(url.git, "div/priority.cuts.csv"))
priority.cuts <- read.csv(file, strip.white = TRUE)
priority.cuts$YearMonth <- as.factor(priority.cuts$YearMonth)
priority.cuts$Pdate_BWR <- as.Date(priority.cuts$Pdate_BWR, format = "%m/%d/%Y")
priority.cuts$Pdate_SC <- as.Date(priority.cuts$Pdate_SC, format = "%m/%d/%Y")
save(priority.cuts, file = file.path(dir.dat, "priority.cuts.rda"), compress = "xz")
```

Surface Water Diversions (div.sw)

Something

```
file <- wrv::DownloadFile(paste0(url.git, "div/div.sw.csv"))
div.sw <- read.csv(file, strip.white = TRUE)
div.sw$YearMonth <- as.factor(div.sw$YearMonth)
div.sw$SWDiv <- div.sw$SWDiv_af * af.to.m3
div.sw$SWDiv_af <- NULL
div.sw[is.na(div.sw$SWDiv), "SWDiv"] <- 0
save(div.sw, file = file.path(dir.dat, "div.sw.rda"), compress = "xz")
```

Wastewater Treatment Plant Diversions (div.ww)

Something

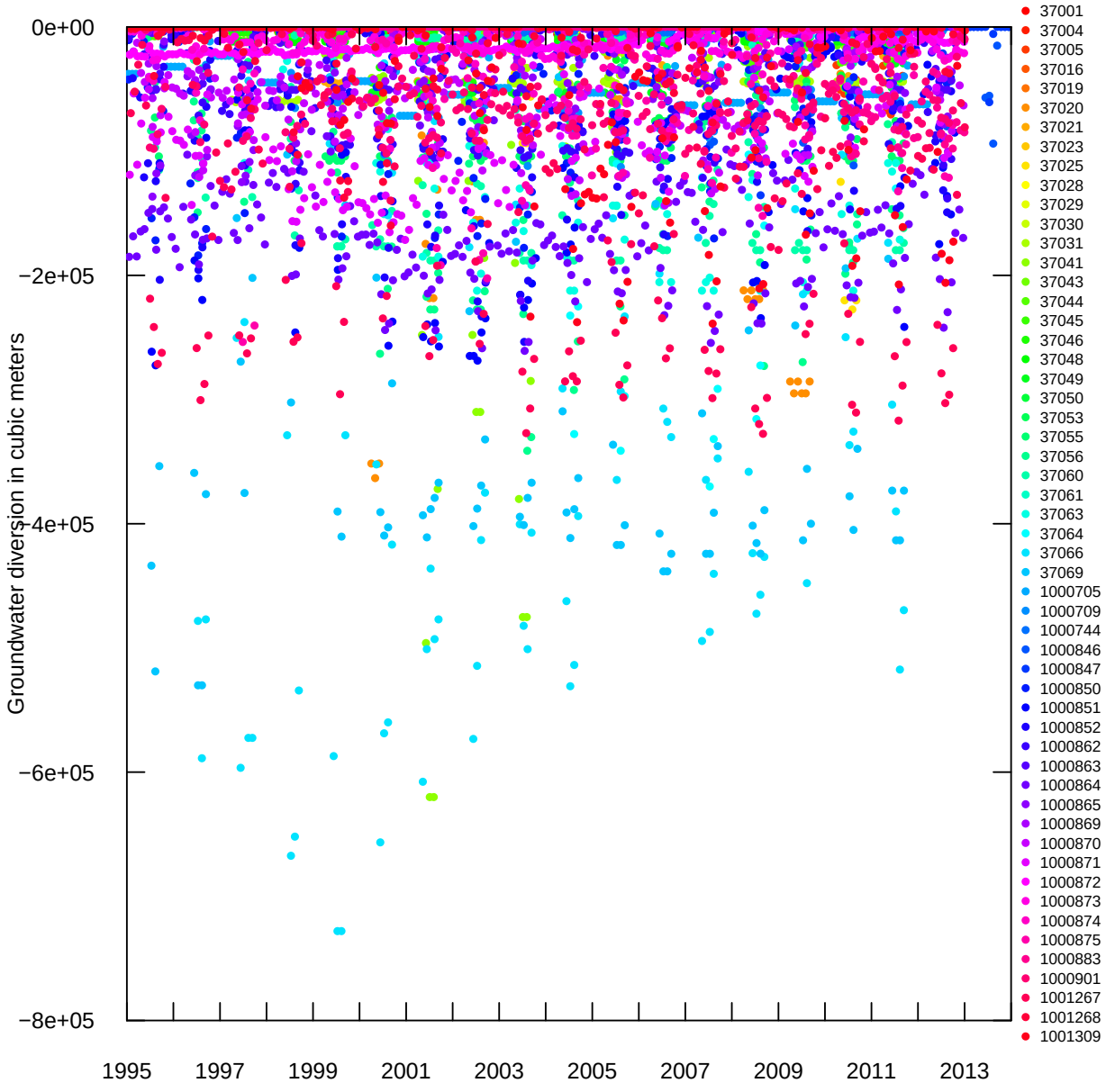

Figure 5: Groundwater diversions from wells. Color indicates a well's WMIS number.

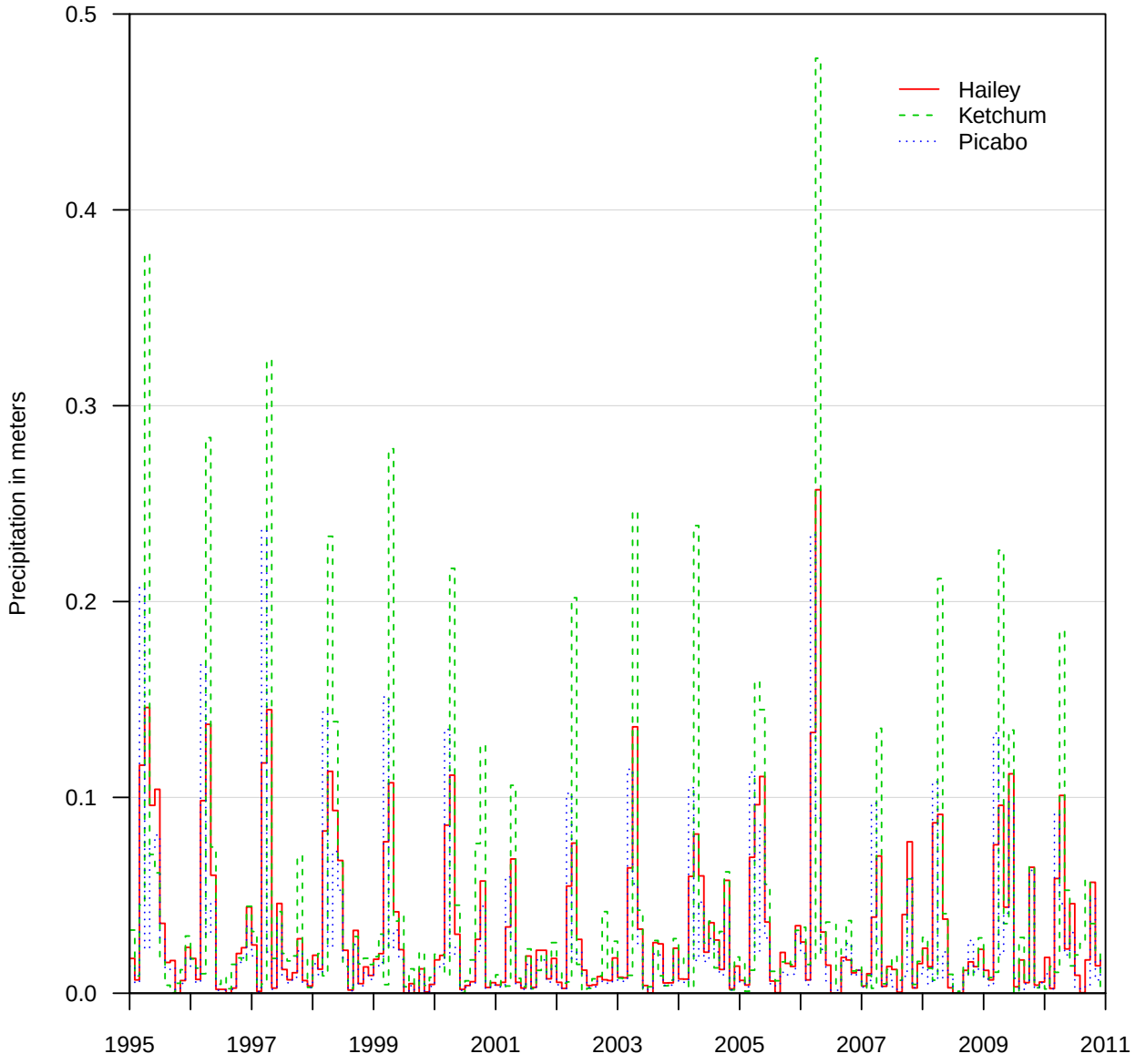

Figure 6: Adjusted precipitation in the Wood River Valley.

```
file <- wrv::DownloadFile(paste0(url.git, "div/div.ww.csv"))
div.ww <- read.csv(file, strip.white = TRUE)
div.ww$YearMonth <- as.factor(div.ww$YearMonth)
div.ww$WWDiv <- div.ww$WWTP_af * af.to.m3
div.ww$WWTP_af <- NULL
div.ww[is.na(div.ww$WWDiv), "WWDiv"] <- 0
save(div.ww, file = file.path(dir.dat, "div.ww.rda"), compress = "xz")
```

Daily Mean Discharge at the Hailey Gaging Station (hailey.discharge)

Daily mean discharge in the Big Wood River at Hailey Idaho (fig. 7).

```
file <- wrv::DownloadFile(paste0(url.git, "hailey.discharge.csv"))
d <- read.csv(file, strip.white = TRUE)
d$Date <- as.Date(as.character(d$Date), format = "%Y%m%d")
d$Disch <- d$Disch_cfs * cfs.to.m3.per.d
d$Disch_cfs <- NULL
hailey.discharge <- d[, c("Date", "Disch", "Code")]
save(hailey.discharge, file = file.path(dir.dat, "hailey.discharge.rda"), compress = "xz")
```

Tributary Groundwater Flows (tributaries)

Average groundwater flows in the tributary canyons and upper Wood River Valley (WRV) (fig. 8).

```
file <- wrv::DownloadFile(paste0(url.git, "tributaries.csv"))
d <- read.csv(file, strip.white = TRUE, stringsAsFactors = FALSE)
d$BasinArea <- d$BasinArea_mi2 * mi2.to.m2
d$BasinArea_mi2 <- NULL
d$Flow <- d$AnnualFlow_af * af.per.y.to.m3.per.d
d$AnnualFlow_af <- NULL
tributaries <- d
save(tributaries, file = file.path(dir.dat, "tributaries.rda"), compress = "xz")
```

Points of Diversion for Groundwater (pod.gw)

Points of diversion for groundwater.

```
file <- wrv::DownloadFile(paste0(url.git, "div/pod.gw.csv"))
d <- read.csv(file, strip.white = TRUE, stringsAsFactors = FALSE)
d$Pdate <- as.Date(d$PriorityDa, format = "%m/%d/%Y")
d$IrrRate <- d$IRRcfs * cfs.to.m3.per.d
cols <- c("WMISNumber", "WaterRight", "EntityName", "EntitySrce", "Pdate", "IrrRate")
pod.gw <- d[, cols]
save(pod.gw, file = file.path(dir.dat, "pod.gw.rda"), compress = "xz")
```

Recharge from Infiltration Basins (infiltration)

Recharge from infiltration basins in the WRV.

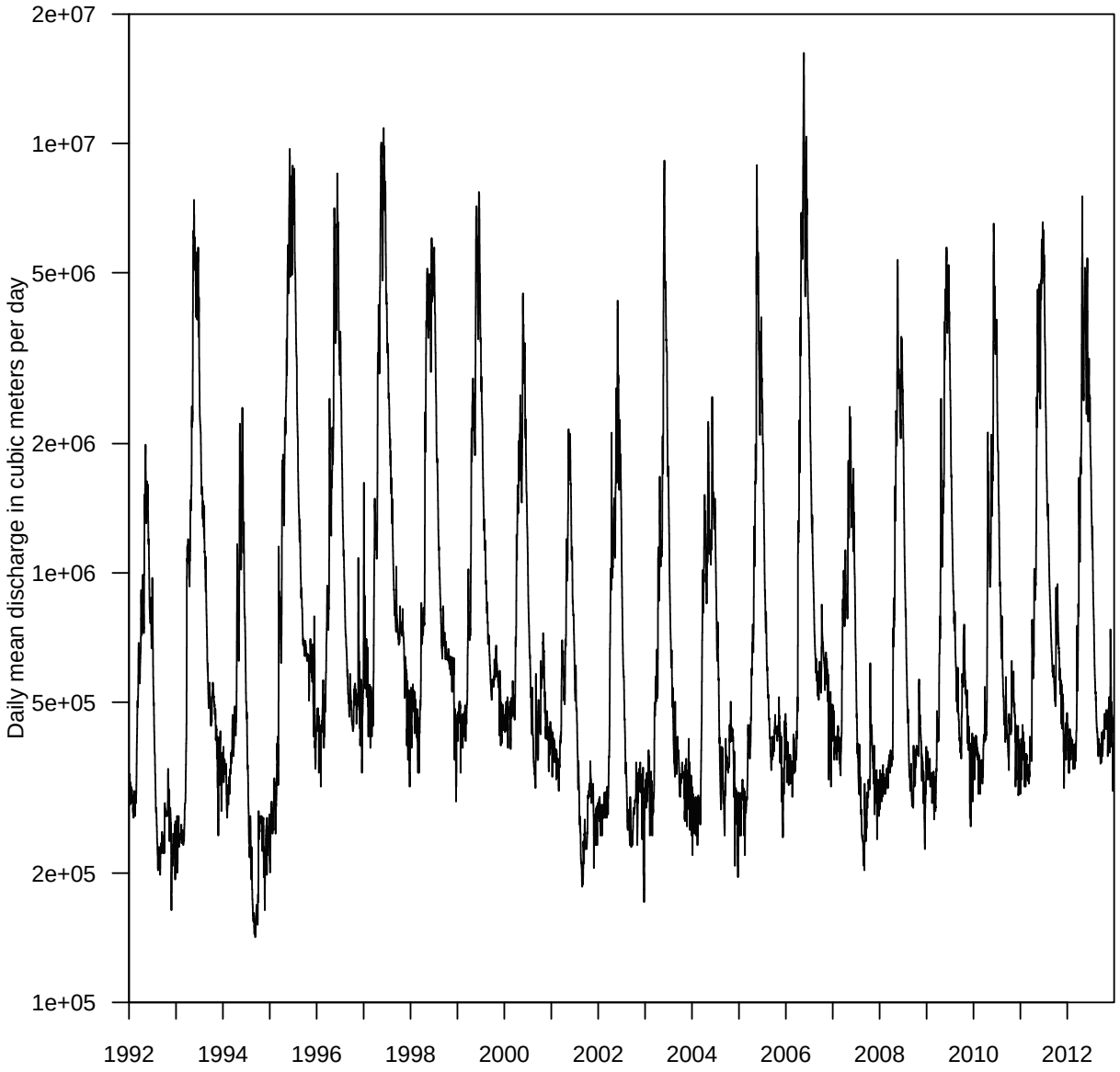

Figure 7: Daily mean discharge in the Big Wood River, Hailey, Idaho.

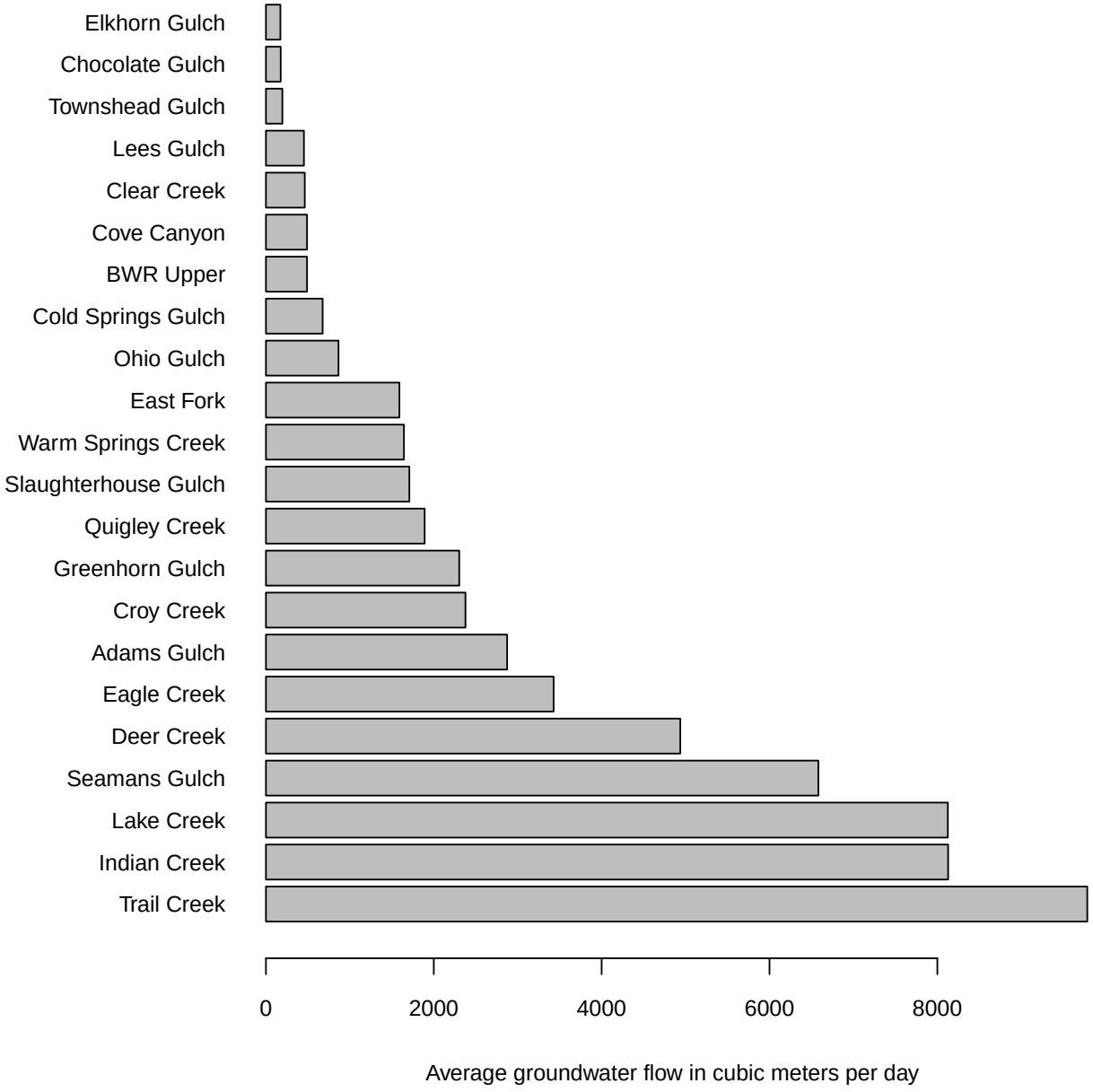

Figure 8: Average groundwater flows in the tributary canyons and upper Wood River Valley.

```
file <- wrv::DownloadFile(paste0(url.git, "div/infiltration.csv"))
infiltration <- read.csv(file, strip.white = TRUE)
infiltration$YearMonth <- as.factor(infiltration$YearMonth)
infiltration$Rech <- infiltration$Rech_af * af.to.m3
infiltration$Rech_af <- NULL
save(infiltration, file = file.path(dir.dat, "infiltration.rda"), compress = "xz")
```

Points

Cities and Towns (cities)

Cities and towns in the WRV and surrounding areas (fig. 9).

```
files <- wrv::DownloadFile(paste0(url.git, "cities.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
cities <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
cities <- spTransform(cities, crs)
save(cities, file = file.path(dir.dat, "cities.rda"), compress = "xz")
```

Map Labels (map.labels)

Map labels in the WRV and surrounding areas (fig. 10).

```
file <- wrv::DownloadFile(paste0(url.git, "map.labels.csv"))
map.labels <- read.csv(file, strip.white=TRUE, stringsAsFactors = FALSE)
map.labels$label <- sub("\\\\n", "\\\n", map.labels$label)
coordinates(map.labels) <- 1:2
colnames(map.labels@coords) <- c("x", "y")
proj4string(map.labels) <- CRS("+init=epsg:4326")
map.labels <- spTransform(map.labels, crs)
save(map.labels, file = file.path(dir.dat, "map.labels.rda"), compress = "xz")
```

Groundwater Points of Diversion, Wells (pod.wells)

Groundwater points of diversion, wells (fig. 11).

```
files <- wrv::DownloadFile(paste0(url.git, "div/pod.wells.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
pod.wells <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
pod.wells <- spTransform(pod.wells, crs)
d <- pod.wells@data
cols <- c("TopOpen1", "BotOpen1", "TopOpen2", "BotOpen2")
d[, cols] <- d[, cols] * ft.to.m
d[d$TopOpen1 == 0 | d$BotOpen1 == 0, c("TopOpen1", "BotOpen1")] <- NA
d[d$TopOpen2 == 0 | d$BotOpen2 == 0, c("TopOpen2", "BotOpen2")] <- NA
```

Something

```
is.pred <- is.na(d$TopOpen1)
dists <- as.matrix(dist(coordinates(pod.wells)))
dists <- dists[!is.pred & d$WellUse %in% "Irrigation", ]
nearest.well <- as.integer(apply(dists, 2, function(i) names(which.min(i))))
```

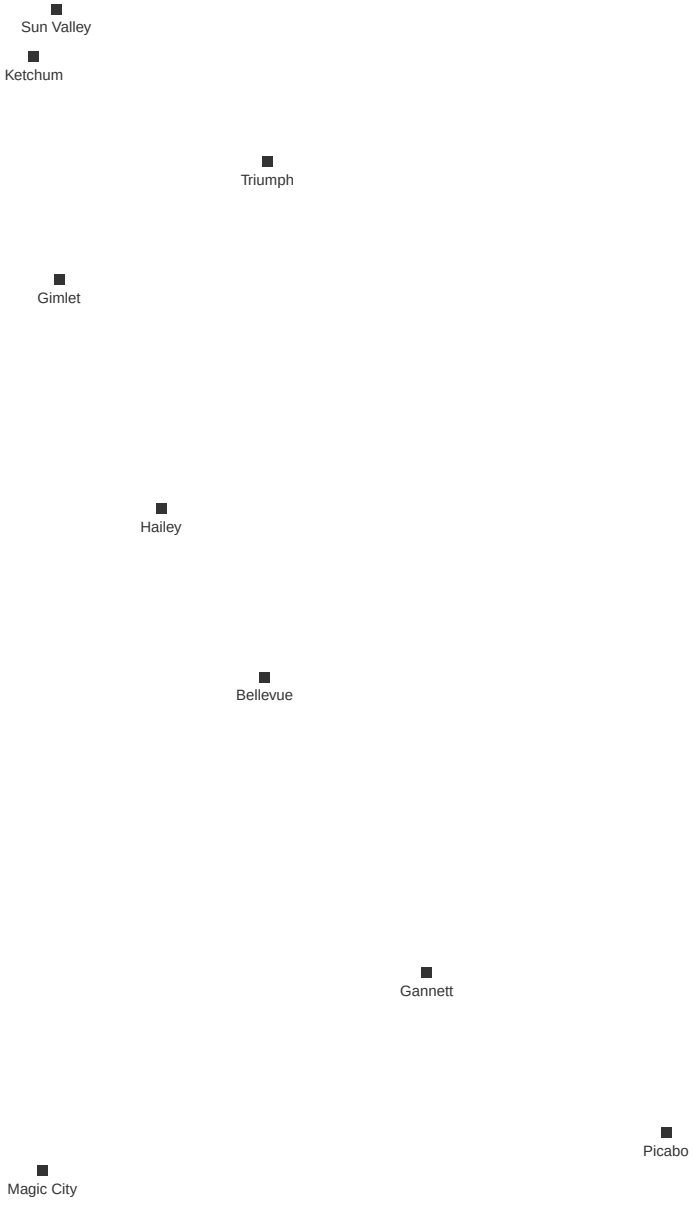

Figure 9: Cities and towns in the Wood River Valley and surrounding areas.

Figure 10: Map labels in the Wood River Valley and surrounding areas.

```
d$TopOpen1[is.pred] <- d$TopOpen1[nearest.well[is.pred]]
d$BotOpen1[is.pred] <- d$BotOpen1[nearest.well[is.pred]]
d$Pred <- is.pred
cols <- c("WMISNumber", "WellUse", "TopOpen1", "BotOpen1", "TopOpen2", "BotOpen2", "Pred")
pod.wells@data <- d[, cols]
save(pod.wells, file = file.path(dir.dat, "pod.wells.rda"), compress = "xz")
```

Lines

Canals (canals)

Canals of the WRV (fig. 12).

```
files <- wrv::DownloadFile(paste0(url.git, "canal/canals.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
canals <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
canals <- spTransform(canals, crs)
canals@data <- canals@data[, "EntityName", drop = FALSE]
save(canals, file = file.path(dir.dat, "canals.rda"), compress = "xz")
```

Reaches of the Big Wood River and Silver Creek (bwr.sc)

Reaches of the Big Wood River and Silver Creek (fig. 13).

```
files <- wrv::DownloadFile(paste0(url.git, "bwr.sc.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
bwr.sc <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
bwr.sc <- spTransform(bwr.sc, crs)
save(bwr.sc, file = file.path(dir.dat, "bwr.sc.rda"), compress = "xz")
```

Rivers and Streams (rivers)

Rivers and streams of the WRV and surrounding areas (fig. 14).

```
files <- wrv::DownloadFile(paste0(url.git, "rivers.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
rivers <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
rivers <- spTransform(rivers, crs)
save(rivers, file = file.path(dir.dat, "rivers.rda"), compress = "xz")
```

Polygons

Extent of Aquifer System (aquifer.extent)

The estimated extent of the WRV aquifer system (fig. 15).

```
files <- wrv::DownloadFile(paste0(url.git, "aquifer.extent.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
aquifer.extent <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
aquifer.extent <- spTransform(aquifer.extent, crs)
save(aquifer.extent, file = file.path(dir.dat, "aquifer.extent.rda"), compress = "xz")
```

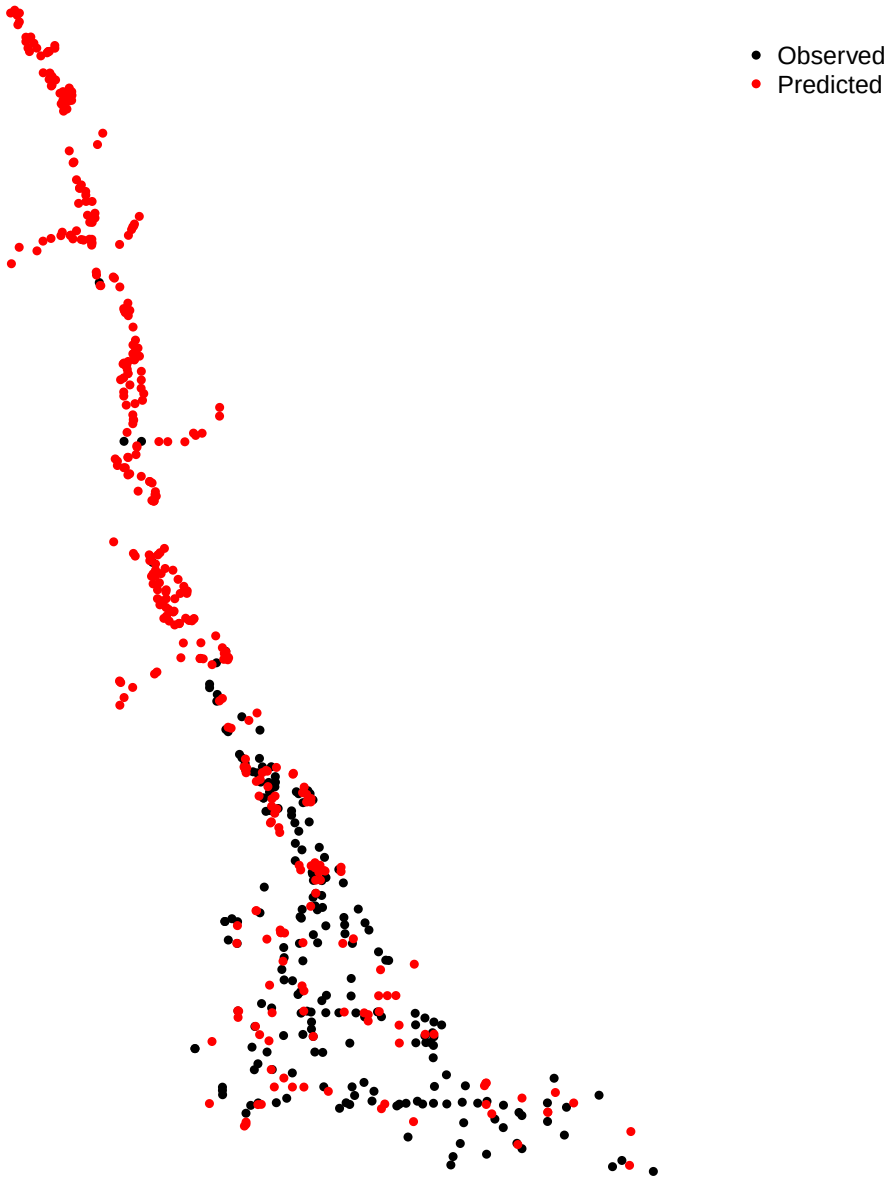

Figure 11: Groundwater points of diversion, wells. Color indicates whether well opening is a predicted value.

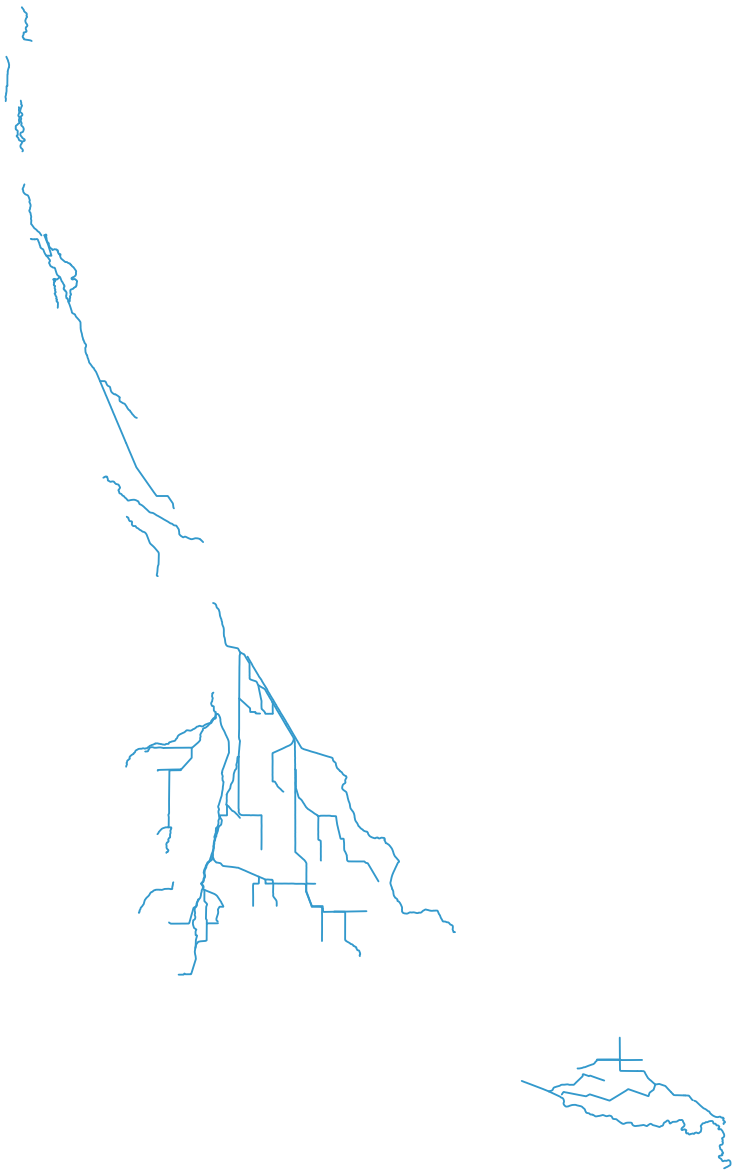

Figure 12: Canals of the Wood River Valley.

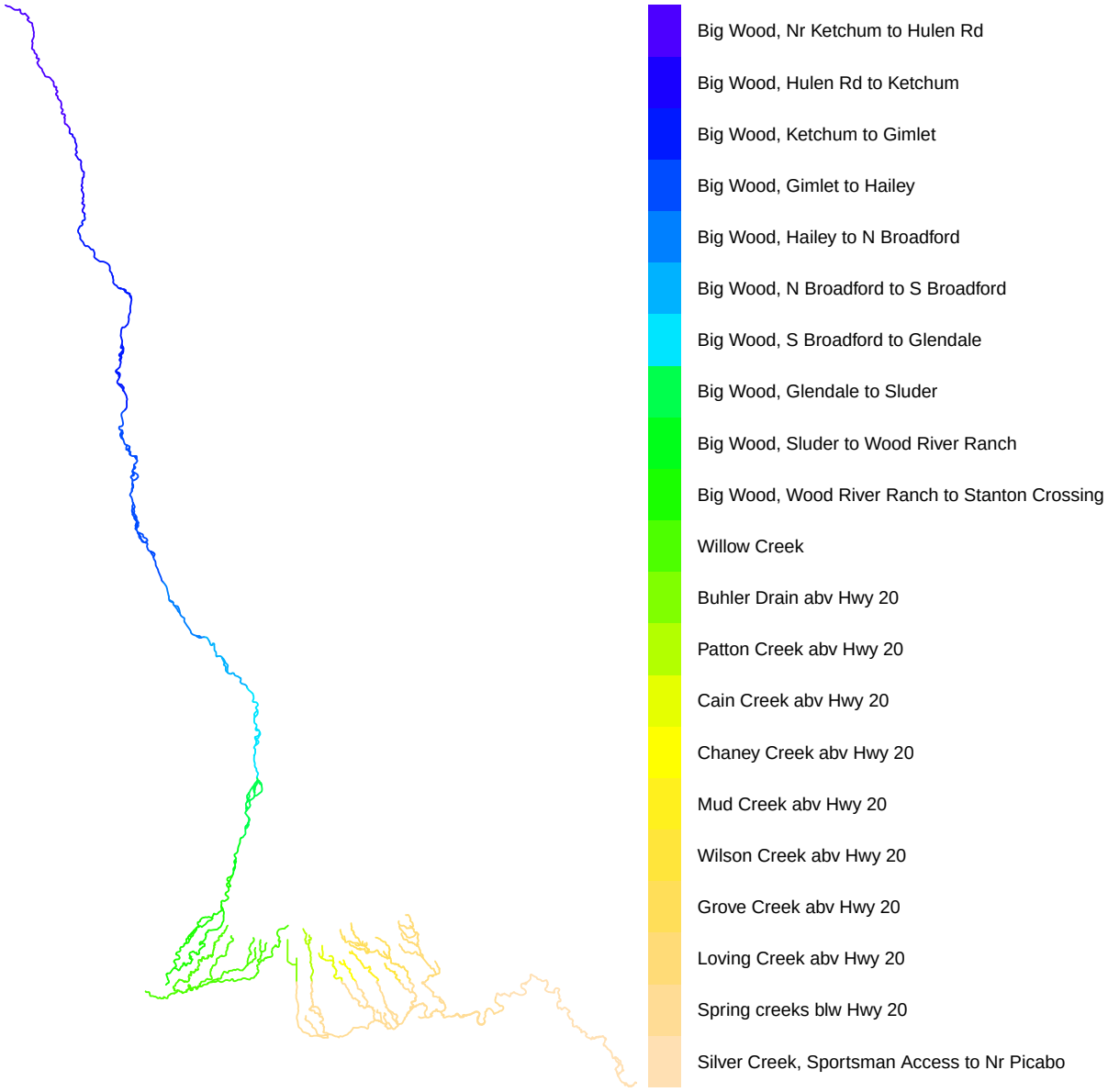

Figure 13: Reaches of the Big Wood River and Silver Creek.

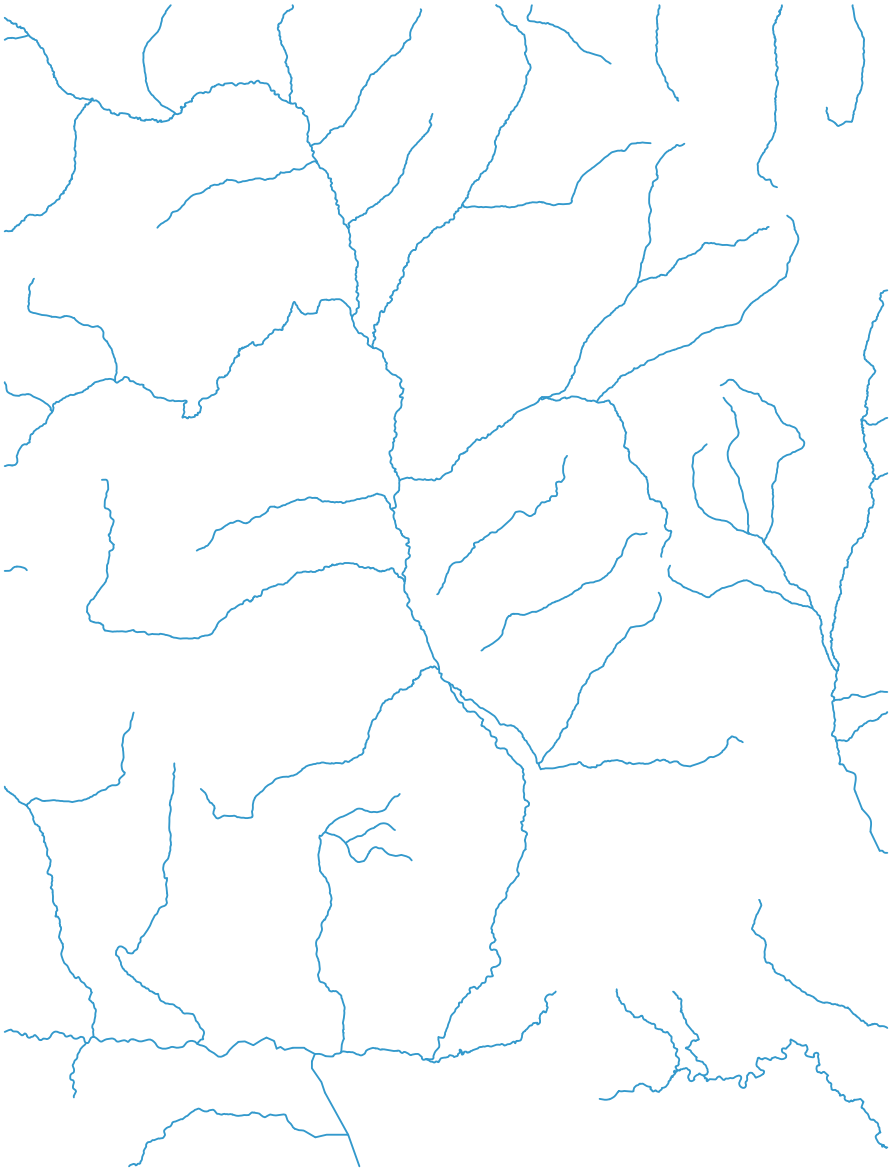

Figure 14: Rivers and streams of the Wood River Valley and surrounding areas.

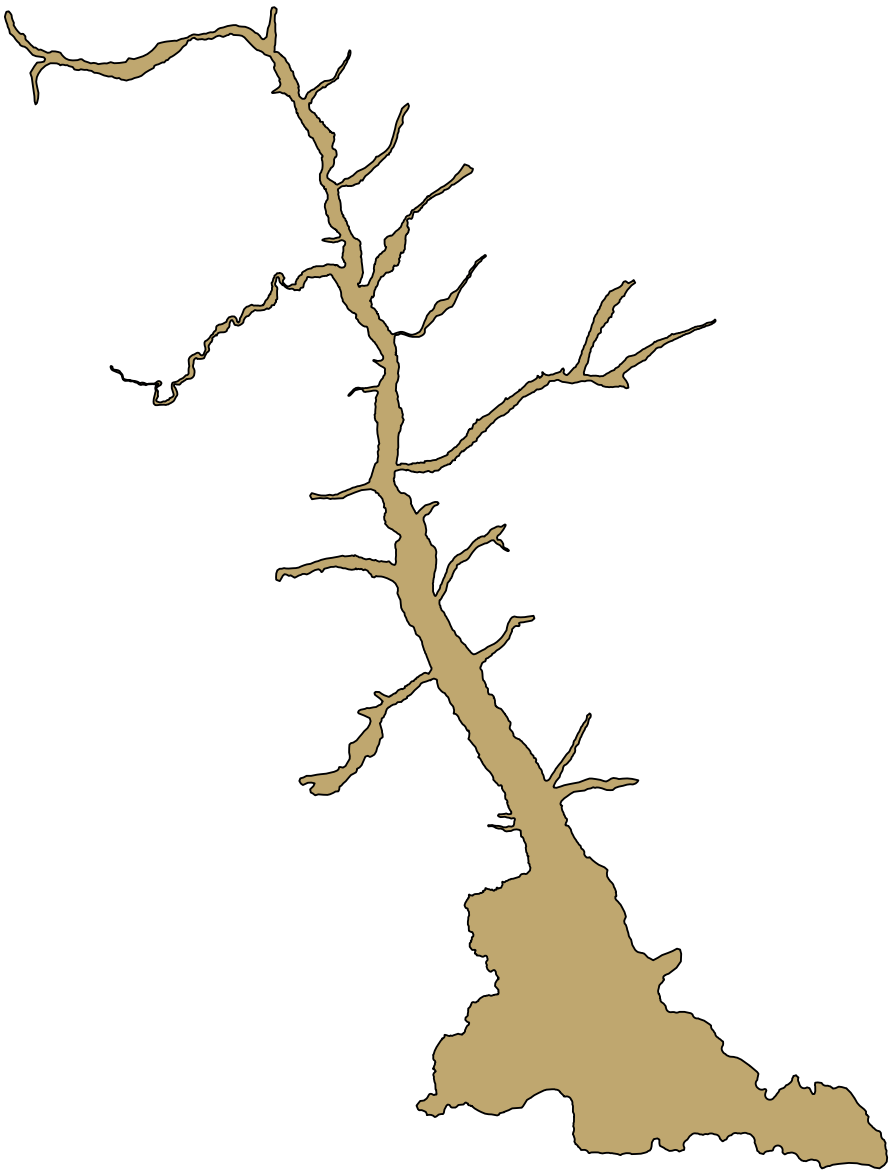

Figure 15: Estimated extent of the Wood River Valley aquifer system.

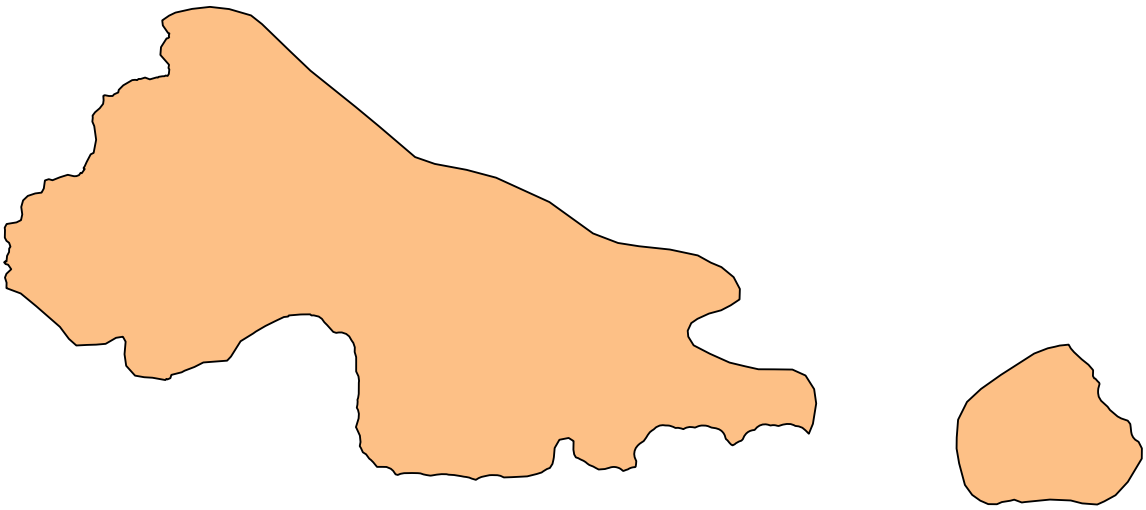

Figure 16: Estimated extent of the confining unit separating the unconfined aquifer from the underlying confined aquifer in the Wood River Valley.

Extent of Aquitard (aquitard.extent)

The estimated extent of the confining unit (aquitard) separating the unconfined aquifer from the underlying confined aquifer in the WRV (fig. 16).

```
files <- wrv::DownloadFile(paste0(url.git, "aquitard.extent.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
aquitard.extent <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
aquitard.extent <- spTransform(aquitard.extent, crs)
save(aquitard.extent, file = file.path(dir.dat, "aquitard.extent.rda"), compress = "xz")
```

Extent of Basalt (basalt.extent)

The estimated extent of the basalt underlying the alluvial WRV aquifer system (fig. 17).

```
files <- wrv::DownloadFile(paste0(url.git, "basalt.extent.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
basalt.extent <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
basalt.extent <- spTransform(basalt.extent, crs)
save(basalt.extent, file = file.path(dir.dat, "basalt.extent.rda"), compress = "xz")
```

Drain Locations at Stanton Crossing and Silver Creek (drains)

Polygons used to define the location of drain boundary conditions in the model domain (fig. 18). The polygons clip the line segments along the aquifer boundary (see aquifer.extent), and model cells intersecting these clipped-line segments are defined as boundary cells.

```
file <- wrv::DownloadFile(paste0(url.git, "drains.kml"))
drains <- suppressWarnings(readOGR(file, basename(file), verbose = FALSE))
drains@data$Description <- NULL
drains <- spTransform(drains, crs)
save(drains, file = file.path(dir.dat, "drains.rda"), compress = "xz")
```

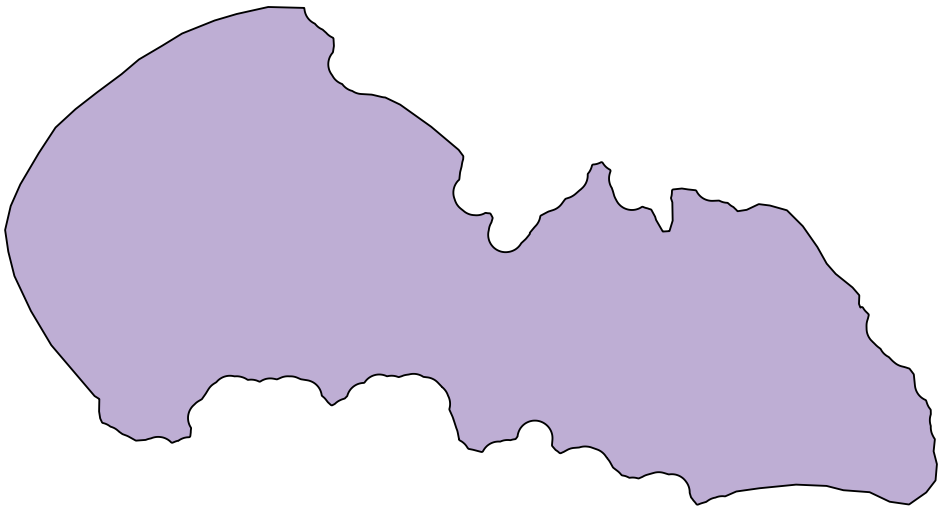

Figure 17: Estimated extent of the basalt underlying the alluvial Wood River Valley aquifer system.

Figure 18: Polygons used to define the location of drain boundary conditions in the model domain.

Groundwater Flow Locations in Tributaries (specified.flows)

Polygons used to define the location of specified flow boundaries in the model domain (fig. 19). Model cells intersecting a polygon line segment are defined as specified flow cells. Cells within the body of the polygon but not a specified flow cell are made inactive.

```
file <- wrv::DownloadFile(paste0(url.git, "specified.flows.kml"))
specified.flows <- suppressWarnings(readOGR(file, basename(file), verbose = FALSE))
specified.flows@data$Description <- NULL
specified.flows <- spTransform(specified.flows, crs)
save(specified.flows, file = file.path(dir.dat, "specified.flows.rda"), compress = "xz")
```

Precipitation Zones (precip.zones)

Precipitation zones of the WRV and surrounding areas (fig. 20).

```
zone.names <- c("Ketchum", "Hailey", "Picabo")
sep <- c(43.438, 43.592)  # northing zone seperators in decimal degrees
ext <- c(-115, -114, 43, 44)  # west, east, south, and north extent in decimal degrees
p1 <- rbind(c(ext[1], sep[2]), c(ext[2], sep[2]), c(ext[2], ext[4]), c(ext[1], ext[4]))
p2 <- rbind(c(ext[1], sep[1]), c(ext[2], sep[1]), c(ext[2], sep[2]), c(ext[1], sep[2]))
p3 <- rbind(c(ext[1], ext[3]), c(ext[2], ext[3]), c(ext[2], sep[1]), c(ext[1], sep[1]))
p1 <- Polygons(list(Polygon(rbind(p1, p1[1, ]))), ID = 1)
p2 <- Polygons(list(Polygon(rbind(p2, p2[1, ]))), ID = 2)
p3 <- Polygons(list(Polygon(rbind(p3, p3[1, ]))), ID = 3)
p <- SpatialPolygons(list(p1, p2, p3), proj4string = CRS("+init=epsg:4326"))
p <- spTransform(p, crs)
p <- SpatialPolygonsDataFrame(p, data.frame(ID = 1:3, PrecipZone = zone.names))
precip.zones <- p
save(precip.zones, file = file.path(dir.dat, "precip.zones.rda"), compress = "xz")
```

Irrigation Entities (irr.entities)

Irrigation entities of the WRV and surrounding areas (fig. 21).

```
files <- wrv::DownloadFile(paste0(url.git, "irr/irr.entities.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
irr.entities <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
irr.entities <- spTransform(irr.entities, crs)
irr.entities <- gBuffer(irr.entities, width = 0, byid = TRUE)  # self-intersecting rings
d <- irr.entities@data
d$EntitySrce <- as.factor(paste(d$EntityName, d$Source))
d$PrecipZone <- over(gCentroid(irr.entities, byid = TRUE), precip.zones)$PrecipZone
irr.entities@data <- d[, c("EntityName", "Source", "EntitySrce", "PrecipZone")]
save(irr.entities, file = file.path(dir.dat, "irr.entities.rda"), compress = "xz")
```

Irrigation Lands (irr.lands)

Irrigated and semi-irrigated lands of the WRV. For example, lands irrigated in 1996 are shown in figure 22.

```
yr <- c(1996, 2000, 2002, 2006, 2008, 2009, 2010)
urls <- paste0(url.git, "irr/irr.lands.", yr, ".zip")
irr.lands <- list()
```

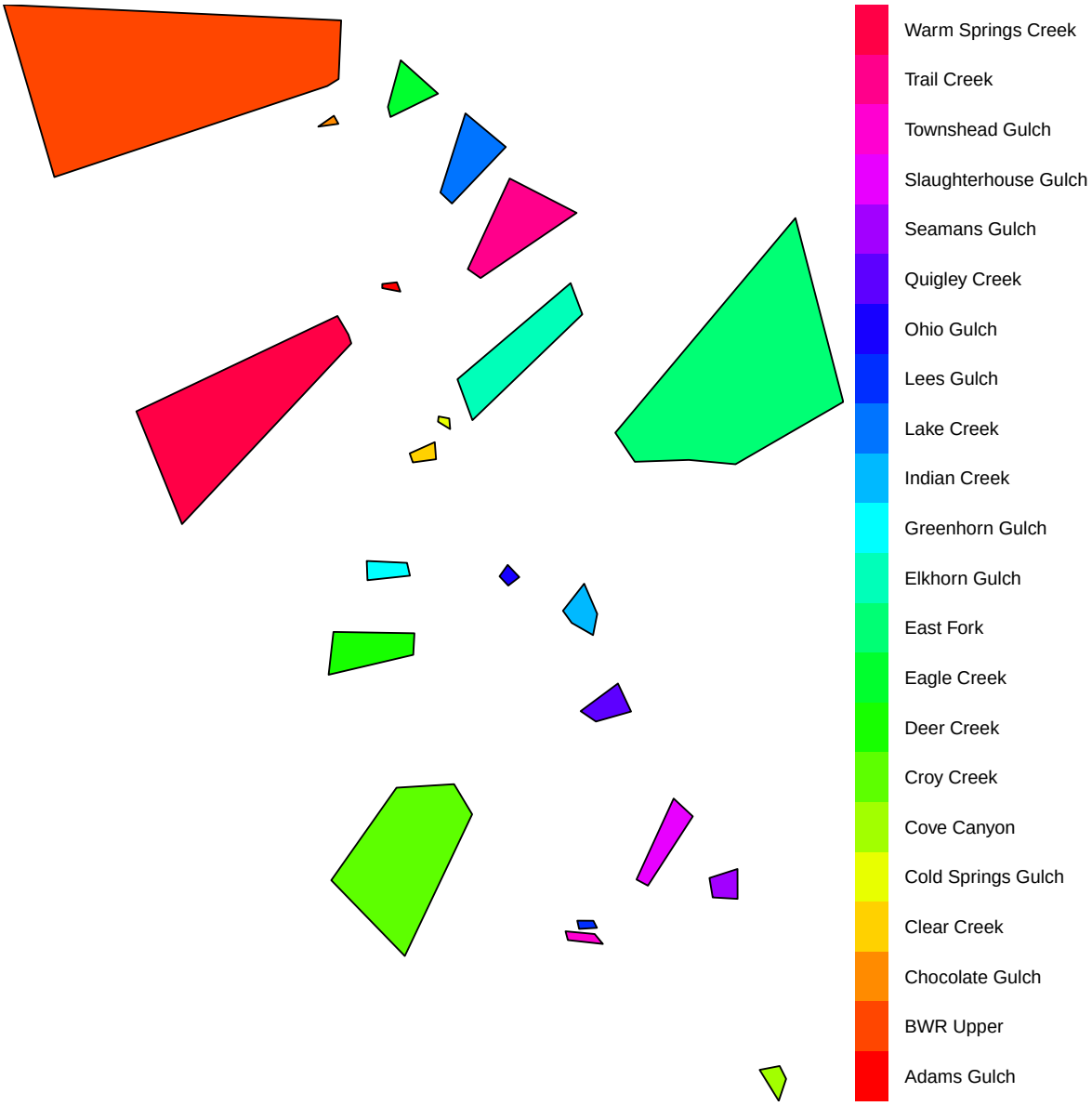

Figure 19: Polygons used to define boundary conditions at source locations in the model domain.

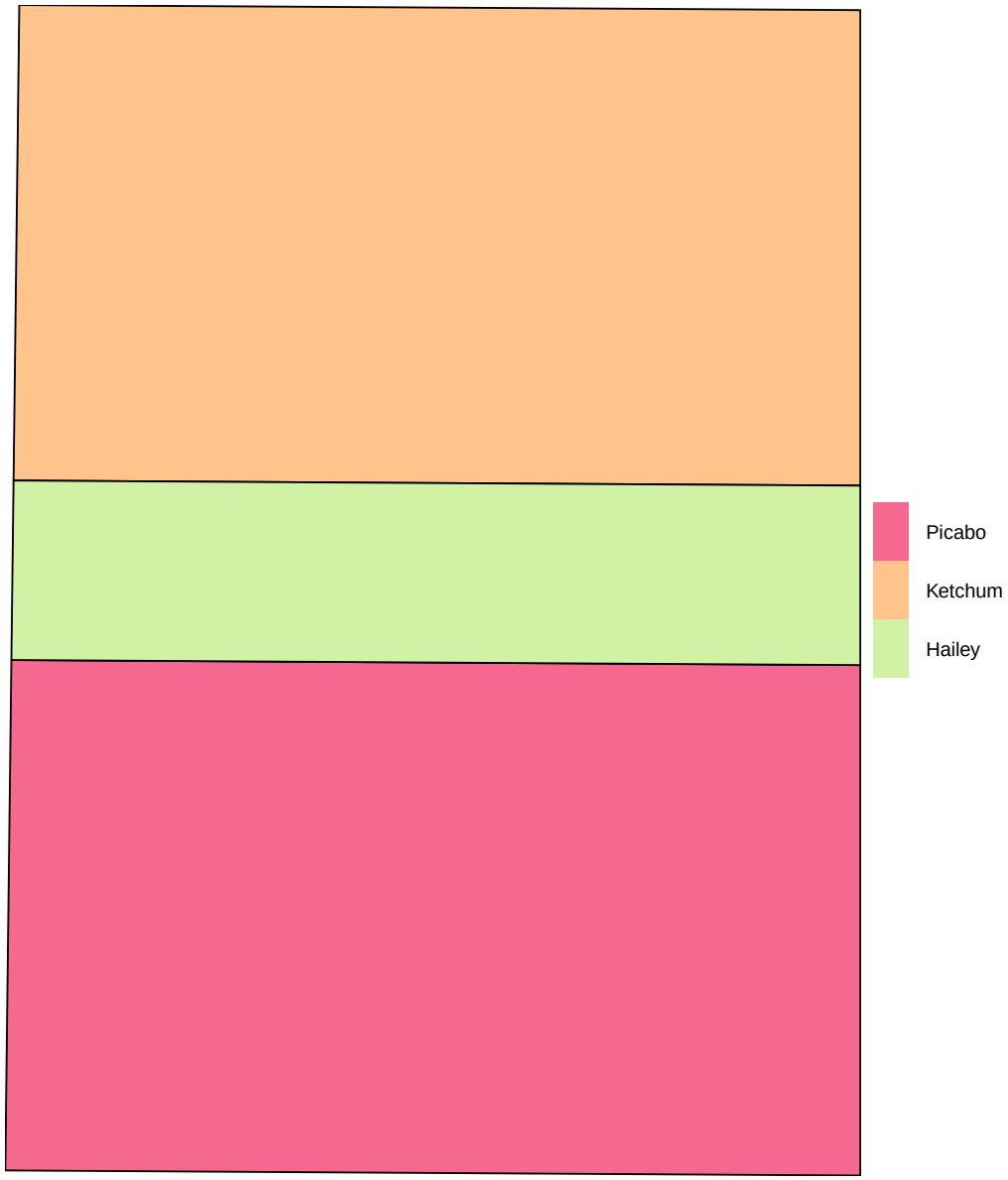

Figure 20: Precipitation zones of the Wood River Valley and surrounding areas.

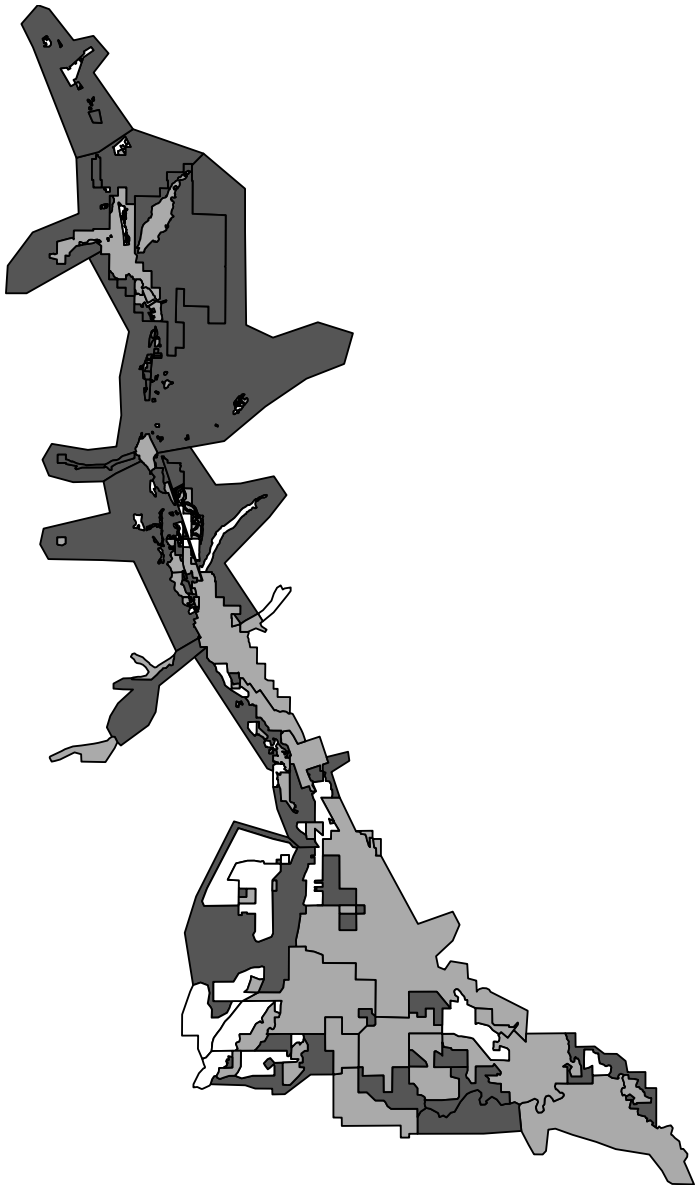

Figure 21: Irrigation entities of the Wood River Valley and surrounding areas.

```
for (i in seq_along(urls)) {
  files <- wrv::DownloadFile(urls[i])
  layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
  p <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
  p <- spTransform(p, crs)
  p@data <- p@data[, paste0("STATUS_", substr(yr[i], 1, 3)), drop = FALSE]
  names(p@data) <- "Status"
  p <- p[p@data[, "Status"] != "non-irrigated", ]
  p <- gBuffer(p, width = 0, byid = TRUE)  # self-intersecting rings
  p@data <- droplevels(p@data)
  irr.lands[[i]] <- p
}
names(irr.lands) <- as.character(yr)
save(irr.lands, file = file.path(dir.dat, "irr.lands.rda"), compress = "xz")
```

Lakes and Reservoirs (lakes)

Lakes and reservoirs of the WRV and surrounding areas (fig. 23).

```
files <- wrv::DownloadFile(paste0(url.git, "lakes.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
lakes <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
lakes <- spTransform(lakes, crs)
save(lakes, file = file.path(dir.dat, "lakes.rda"), compress = "xz")
```

Public Land Parcels (public.parcels)

Public land parcels for areas north of Bellevue (fig. 24).

```
files <- wrv::DownloadFile(paste0(url.git, "irr/public.parcels.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
public.parcels <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
public.parcels <- spTransform(public.parcels, crs)
public.parcels <- as(public.parcels, "SpatialPolygons")
save(public.parcels, file = file.path(dir.dat, "public.parcels.rda"), compress = "xz")
```

Soil Infiltration (soils)

Soil infiltration rates of the WRV and surrounding areas (fig. 25).

```
files <- wrv::DownloadFile(paste0(url.git, "soil/soils.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
soils <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
soils <- spTransform(soils, crs)
d <- soils@data
d$InfRate <- d$INF * ft.per.mo.to.m.per.d
soils@data <- d[, c("SoilLayer", "InfRate")]
save(soils, file = file.path(dir.dat, "soils.rda"), compress = "xz")
```

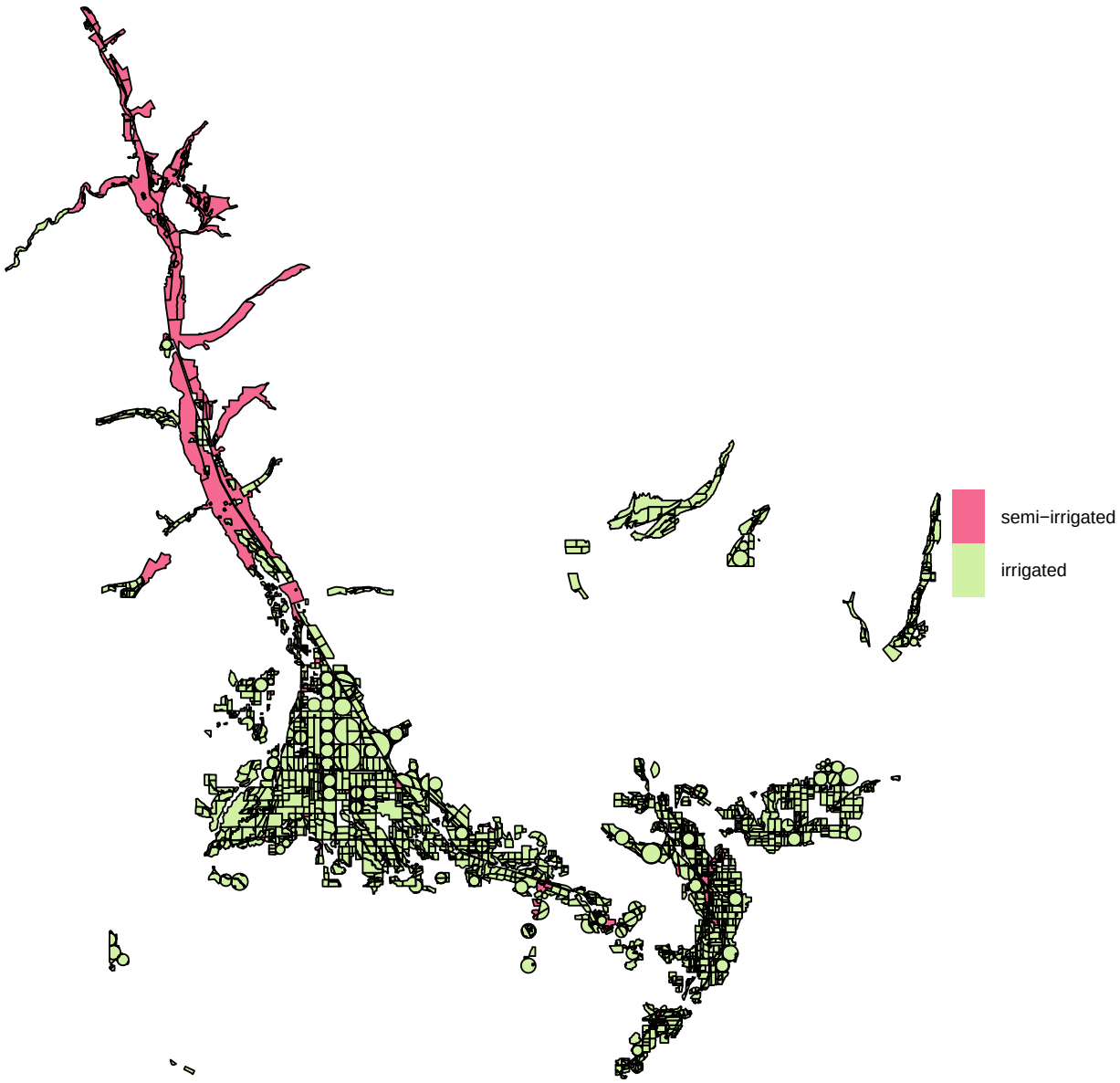

Figure 22: Irrigated and semi-irrigated lands of the Wood River Valley in 1996.

Figure 23: Lakes and reservoirs of the Wood River Valley and surrounding areas.

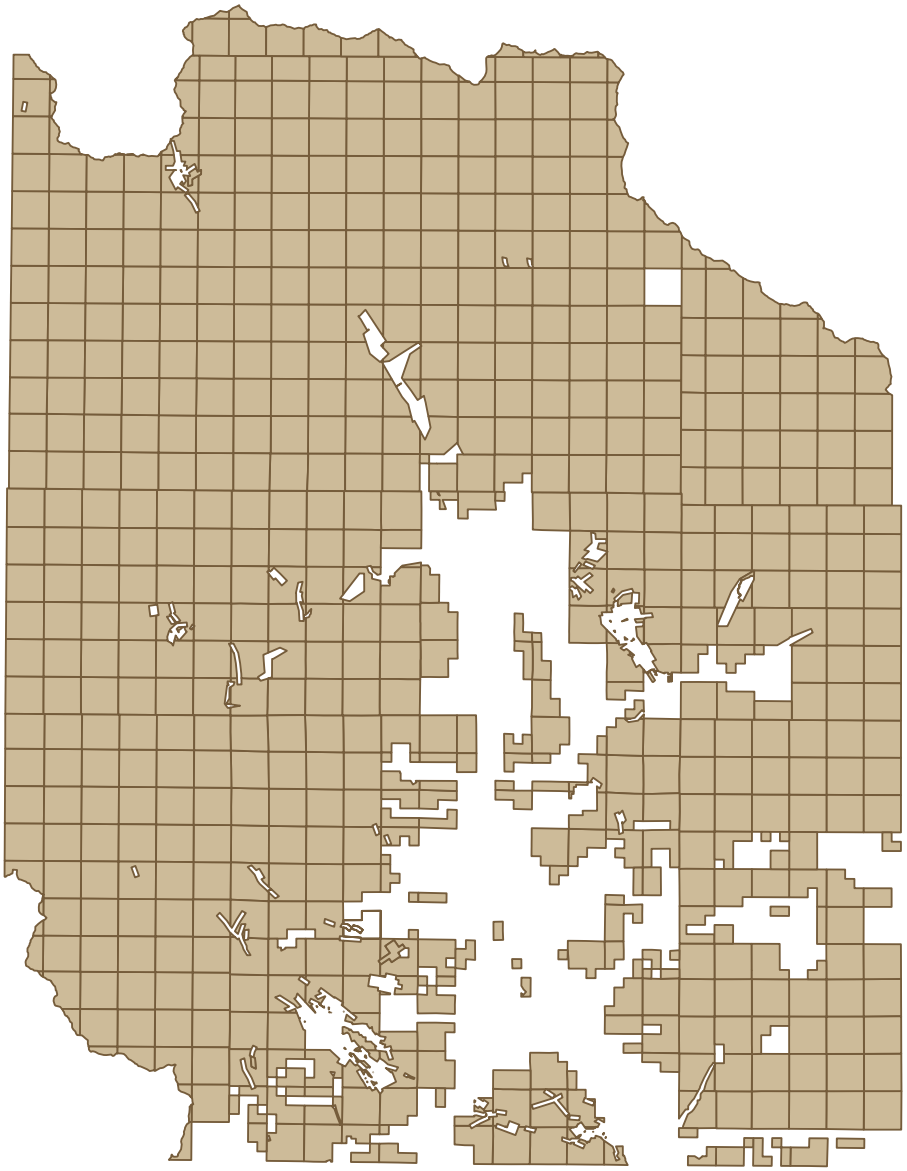

Figure 24: Public land parcels for areas north of Bellevue.

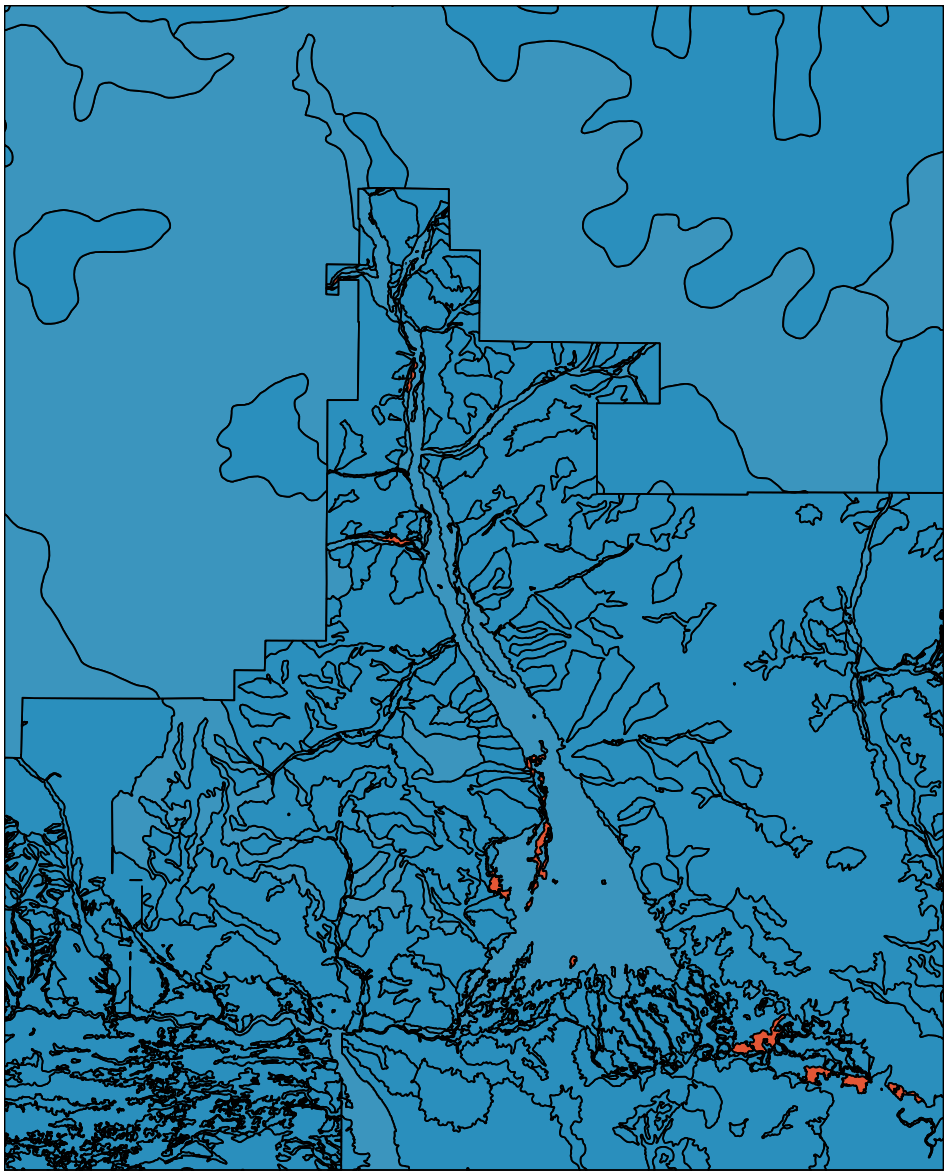

Figure 25: Soil infiltration rates of the Wood River Valley and surrounding areas.

Wetlands (wetlands)

Wetlands of the WRV and surrounding areas (fig. 26).

```
files <- wrv::DownloadFile(paste0(url.git, "irr/wetlands.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
wetlands <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
wetlands <- spTransform(wetlands, crs)
wetlands <- as(wetlands, "SpatialPolygons")
save(wetlands, file = file.path(dir.dat, "wetlands.rda"), compress = "xz")
```

Infiltration Basins (infil.basins)

Infiltration basins in the WRV (fig. 27).

```
files <- wrv::DownloadFile(paste0(url.git, "div/infil.basins.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
infil.basins <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
infil.basins <- spTransform(infil.basins, crs)
save(infil.basins, file = file.path(dir.dat, "infil.basins.rda"), compress = "xz")
```

Grids

Something

```
high.res.spatial.grid <- disaggregate(spatial.grid, fact = 5L)
```

Pre-Quaternary Bedrock Surface and Top of Quaternary Basalt (alluvium.bottom)

The estimated elevation of the pre-Quaternary bedrock surface and top of Quaternary basalt in the the WRV aquifer system (fig. 28).

```
file <- wrv::DownloadFile(paste0(url.git, "alluvium.bottom.tif"))
alluvium.bottom <- readGDAL(file, band = 1, silent = TRUE)
alluvium.bottom <- projectRaster(raster(alluvium.bottom), high.res.spatial.grid)
alluvium.bottom <- aggregate(alluvium.bottom, fact = 5L)
names(alluvium.bottom) <- "alluvium.bottom"
save(alluvium.bottom, file = file.path(dir.dat, "alluvium.bottom.rda"), compress = "xz")
```

Topography of Land Surface (land.surface)

The topography of the land surface in the WRV and vicinity.

```
wrv::DownloadFile(url.ned)
file <- file.path(tempdir(), "n44w115", "grdn44w115_13")
r <- raster(readGDAL(file, band = 1, silent = TRUE))
land.surface <- projectRaster(r, high.res.spatial.grid)
land.surface <- aggregate(land.surface, fact = 5L)
land.surface[is.na(alluvium.bottom)] <- NA
names(land.surface) <- "land.surface"
save(land.surface, file = file.path(dir.dat, "land.surface.rda"), compress = "xz")
```

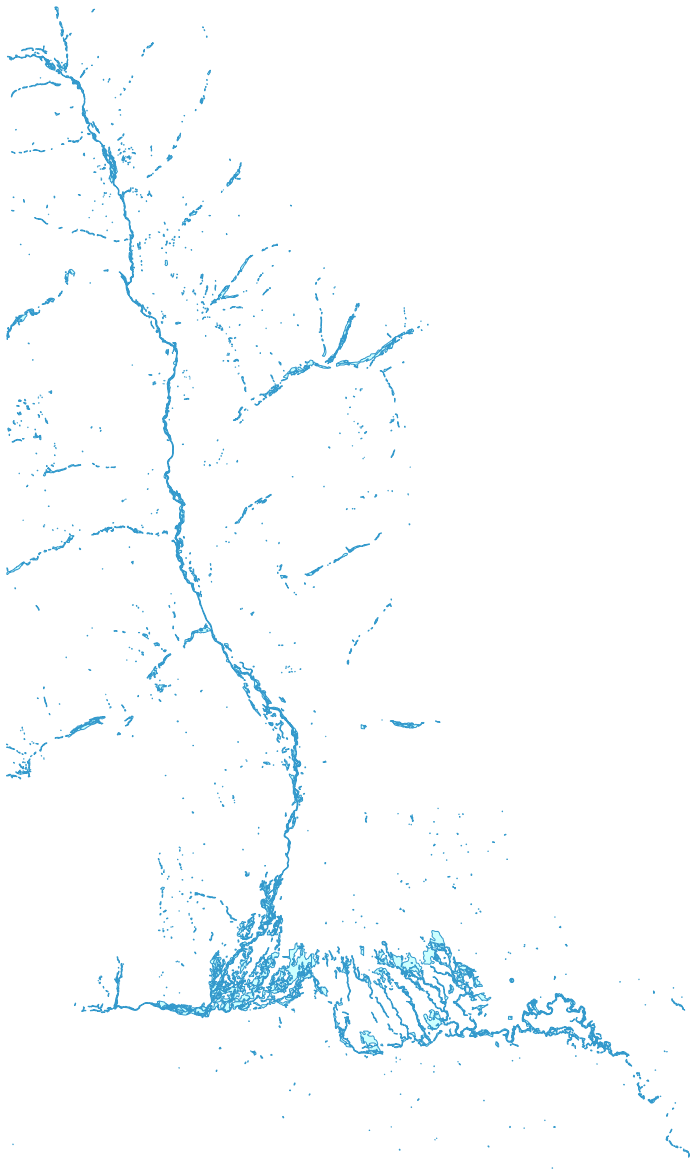

Figure 26: Wetlands of the Wood River Valley and surrounding areas.

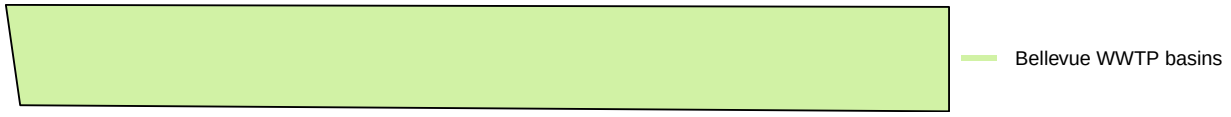

Figure 27: Infiltration basins in the Wood River Valley.

Hillshading based on the slope and aspect of land-surface elevations (fig. 29).

```
ext <- extent(spatial.grid)
ext <- extent(c(extendrange(c(ext@xmin, ext@xmax), f = 0.05),
                extendrange(c(ext@ymin, ext@ymax), f = 0.05)))
nrows <- (ext@ymax - ext@ymin) / 20
ncols <- (ext@xmax - ext@xmin) / 20
r <- projectRaster(r, raster(ext, nrows = nrows, ncols = ncols, crs = crs))
r[] <- r[] * 2
r <- hillShade(slope = terrain(r, opt = "slope"), aspect = terrain(r, opt = "aspect"))
r.range <- range(r[], na.rm = TRUE)
r[] <- findInterval(r[], seq(r.range[1], r.range[2], length.out = 255)) / 255
r[] <- round(r[], digits = 6)
hill.shading <- r
save(hill.shading, file = file.path(dir.dat, "hill.shading.rda"), compress = "xz")
```

Evapotranspiration (et)

Average monthly evapotranspiration in the WRV and surrounding areas. For example, the average evapotranspiration during October 2008 is shown in figure 30.

```
urls <- paste0(url.git, "et/et.", yr.mo, ".tif")
FUN <- function(i) {
  file <- wrv::DownloadFile(urls[i])
  r <- readGDAL(file, band = 1, silent = TRUE)
  r[[1]] <- r[[1]] * mm.to.m
  return(r)
}
et.raw <- lapply(seq_along(urls), FUN)
names(et.raw) <- as.character(yr.mo)
```

Something...

```
FUN <- function(i) {
  r <- aggregate(projectRaster(raster(i), high.res.spatial.grid), fact = 5L)
  r[is.na(alluvium.bottom)] <- NA
  upper.limit <- mean(r[], na.rm = TRUE) + sd(r[], na.rm = TRUE) * 3
  r[r > upper.limit] <- upper.limit
  r[] <- round(r[], digits = 6)
  return(r)
```

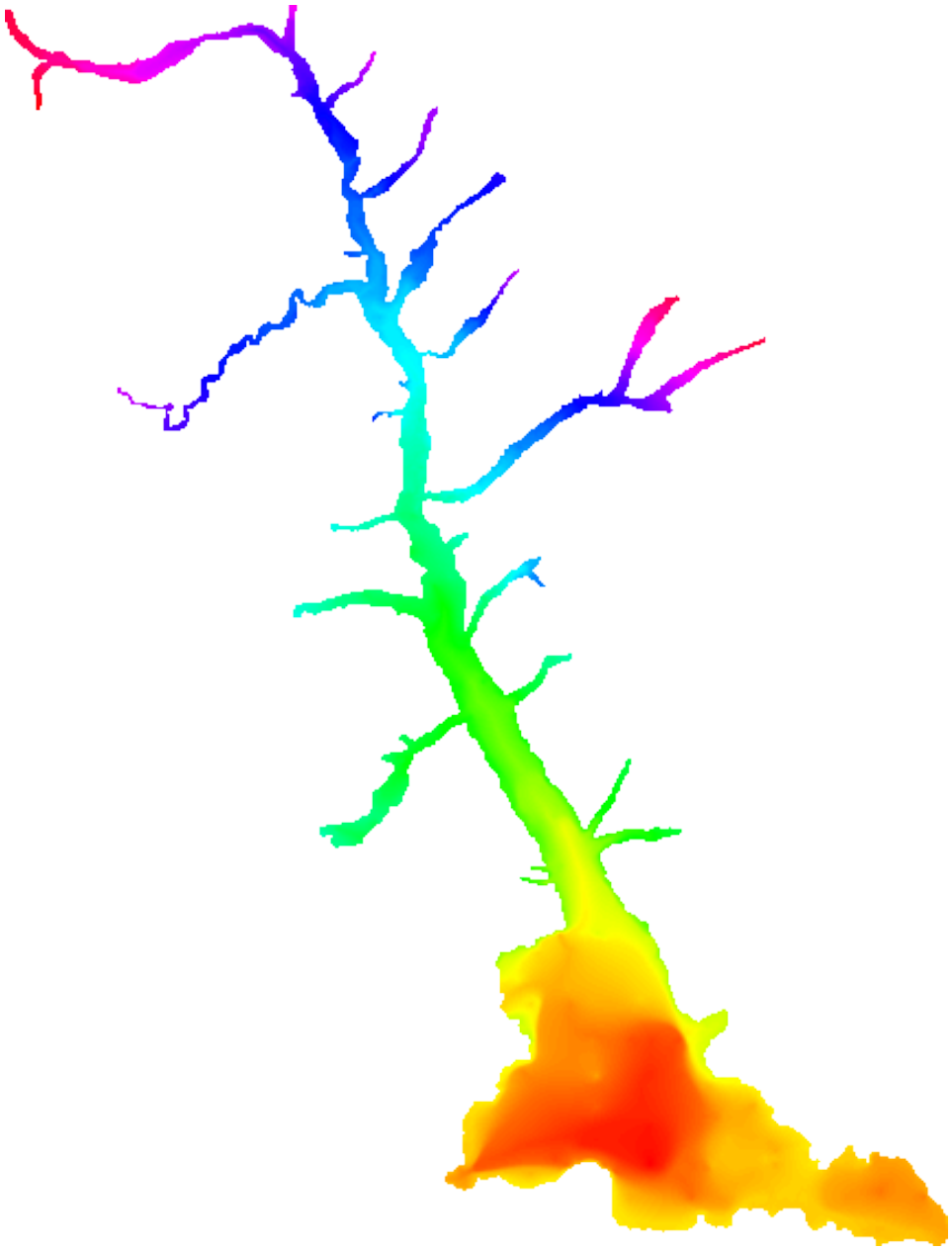

Figure 28: Estimated elevation of the pre-Quaternary bedrock surface and top of Quaternary basalt in the the Wood River Valley aquifer system.

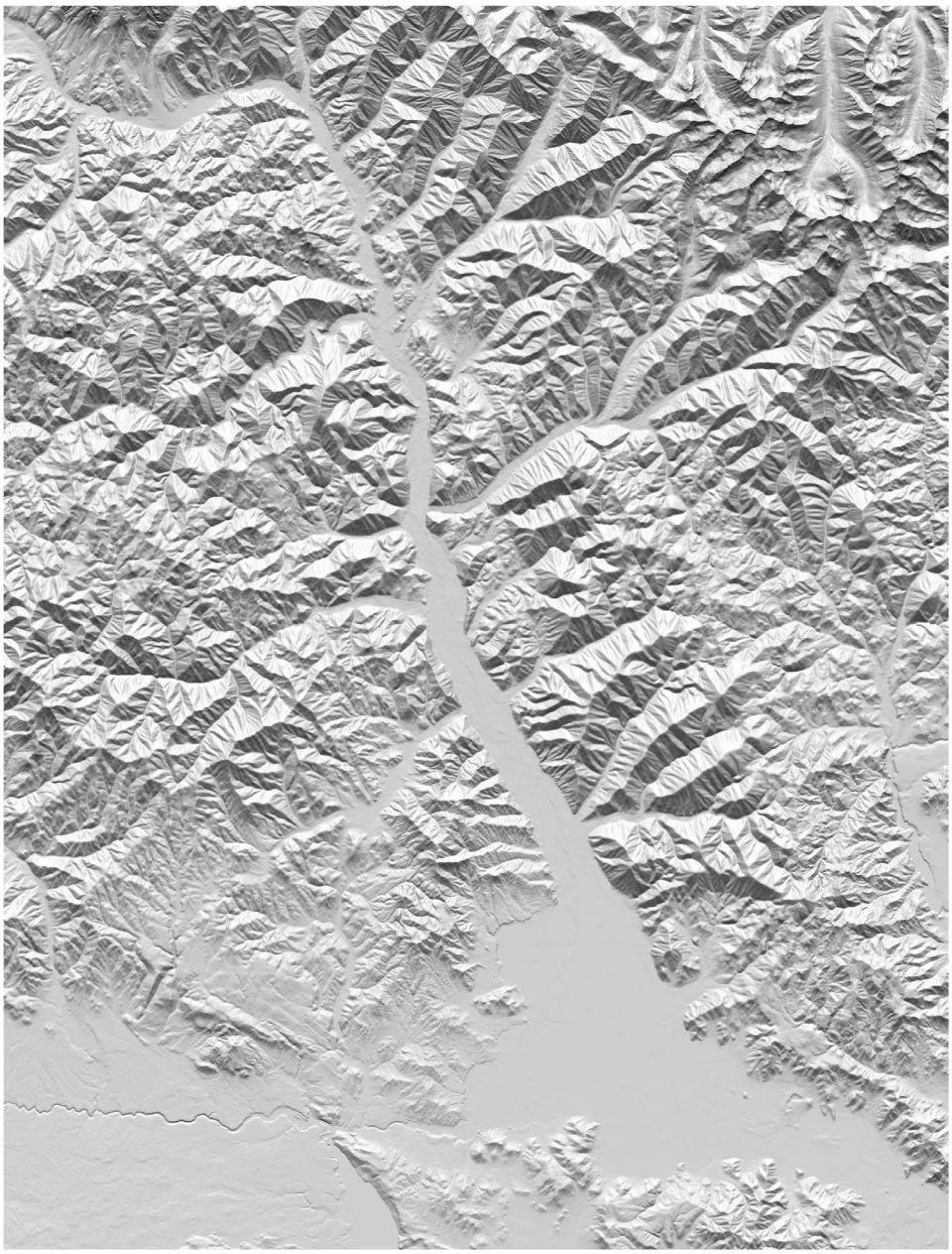

Figure 29: Hillshading based on the slope and aspect of land-surface elevations.

```
}
et <- stack(lapply(et.raw, FUN), quick = TRUE)
names(et) <- as.character(yr.mo)
save(et, file = file.path(dir.dat, "et.rda"), compress = "xz")
```

Level 2 Data

Data sets processed at Level 2 result from analyses of Level 1 data.

Polygons

Monthly Irrigation Entity Components (entity.components)

Something...

```
p <- irr.lands
p <- lapply(p, function(i) wrv::SetPolygons(i, wetlands, "gDifference", 0.001))
p <- lapply(p, function(i) wrv::SetPolygons(i, public.parcels, "gDifference", 0.001))
p <- lapply(p, function(i) wrv::SetPolygons(irr.entities, i, "gIntersection", 0.001))
for (i in seq_along(p)) p[[i]]@data$area <- gArea(p[[i]], byid = TRUE)
irr.by.entity <- p
```

Something...

```
FUN <- function(i) {
  d <- aggregate(i@data$area, by = list(i@data$EntitySrce), sum)
  names(d) <- c("EntitySrce", "area")
  FUN <- function(j) as.character(i@data$PrecipZone[i@data$EntitySrce == j][1])
  d$PrecipZone <- as.factor(vapply(d$EntitySrce, FUN, ""))
  return(d)
}
area.by.entity <- lapply(irr.by.entity, FUN)
```

Something...

```
FUN <- function(i) {
  yr <- irr.lands.year$IL_Year[irr.lands.year$Year %in% substr(i, 1, 4)]
  p <- irr.by.entity[[yr]]
  unique.sources <- sort(unique(as.character(p@data$EntitySrce)))
  FUN <- function (j) {
    x <- gUnaryUnion(p[p@data$EntitySrce == j, ])@polygons[[1]]
    slot(x, "ID") <- j
    return(x)
  }
  sp <- SpatialPolygons(lapply(unique.sources, FUN), proj4string = p@proj4string)
  et.pts <- spTransform(as(et.raw[[i]], "SpatialPointsDataFrame"), crs)
  mean.et <- over(sp, et.pts, fn = mean, na.rm = TRUE)
  d <- as.data.frame(list(EntitySrce = rownames(mean.et), mean.et = mean.et[, 1]))
  d <- merge(d, area.by.entity[[yr]], by = "EntitySrce", all.x = TRUE)
  d$et.vol <- d$mean.et * d$area
  d$precip.vol <- NA
  for (j in levels(d$PrecipZone)) {
```

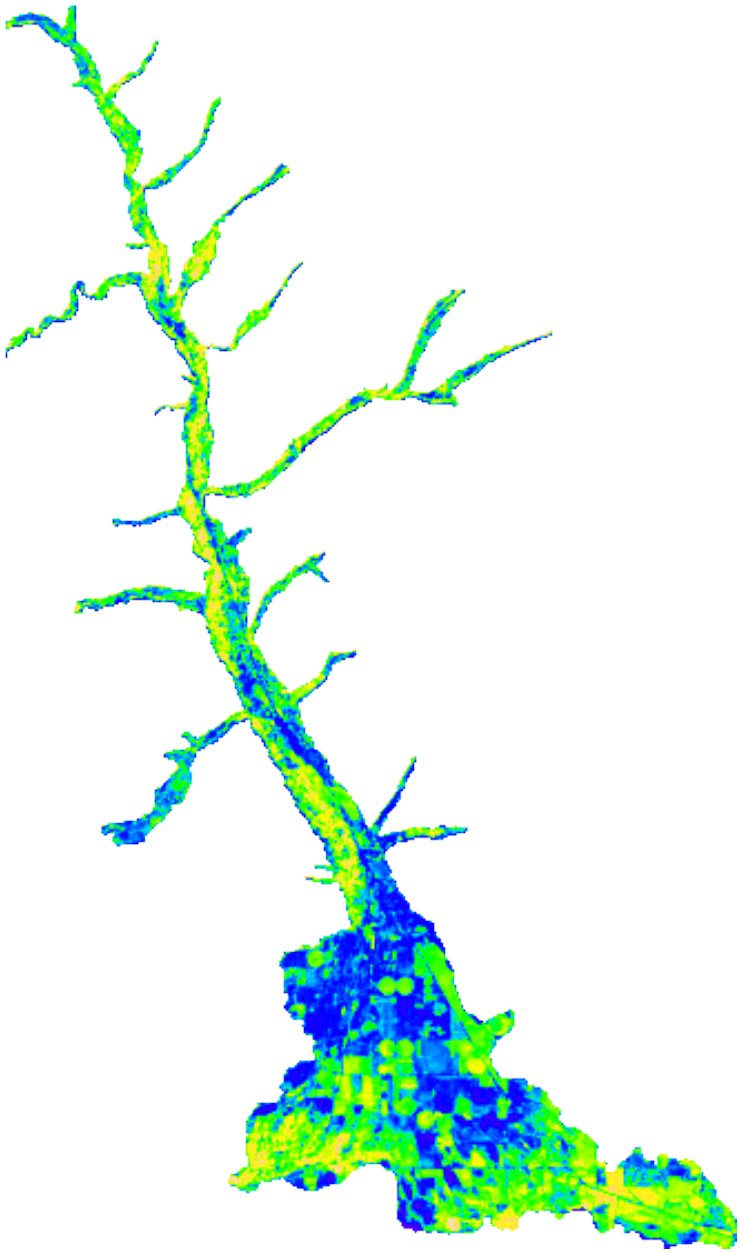

Figure 30: Evapotranspiration during October 2008 in the the Wood River Valley.

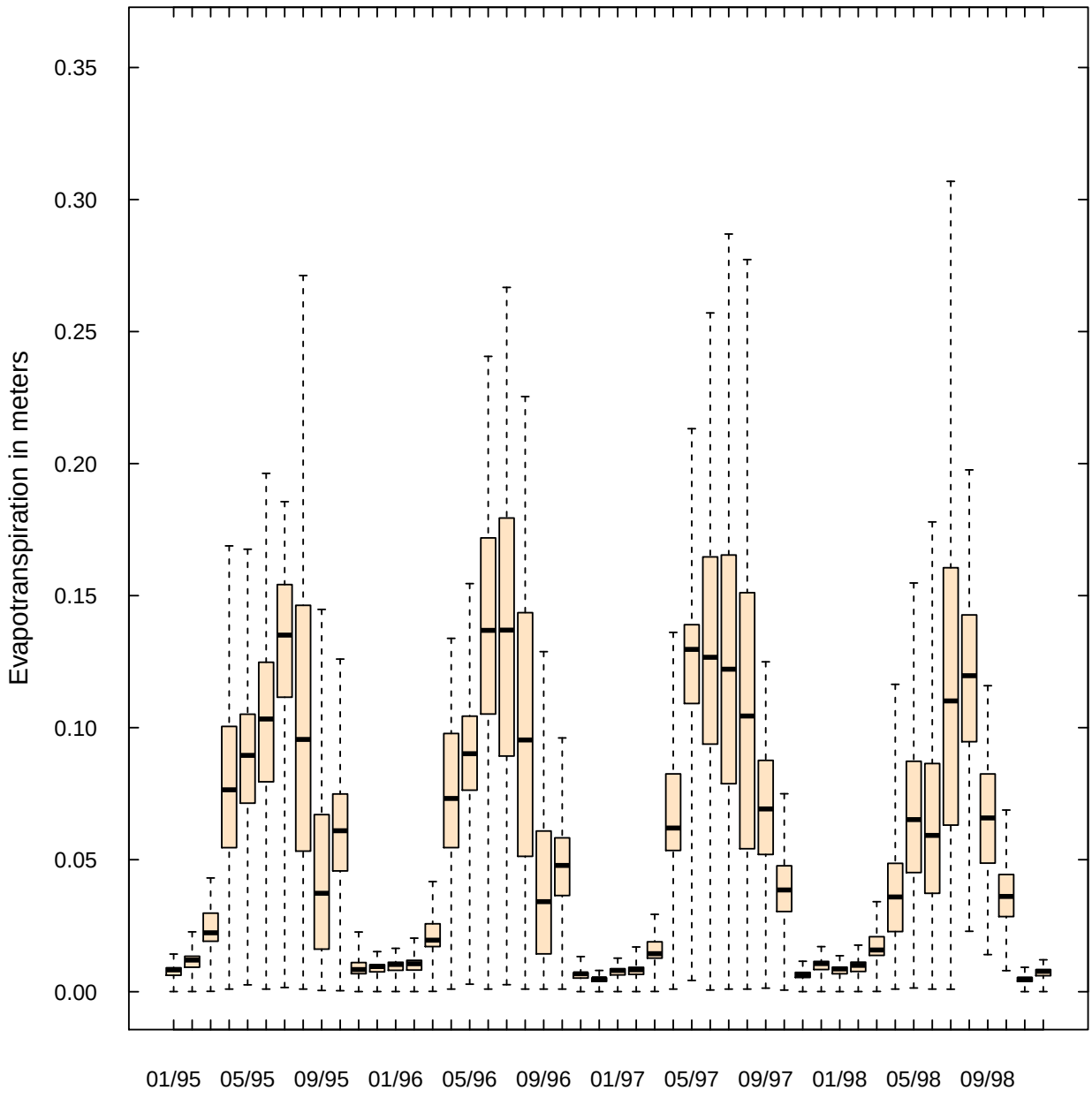

Figure 31: Evapotranspiration from 1995 through 1998 in the the Wood River Valley.

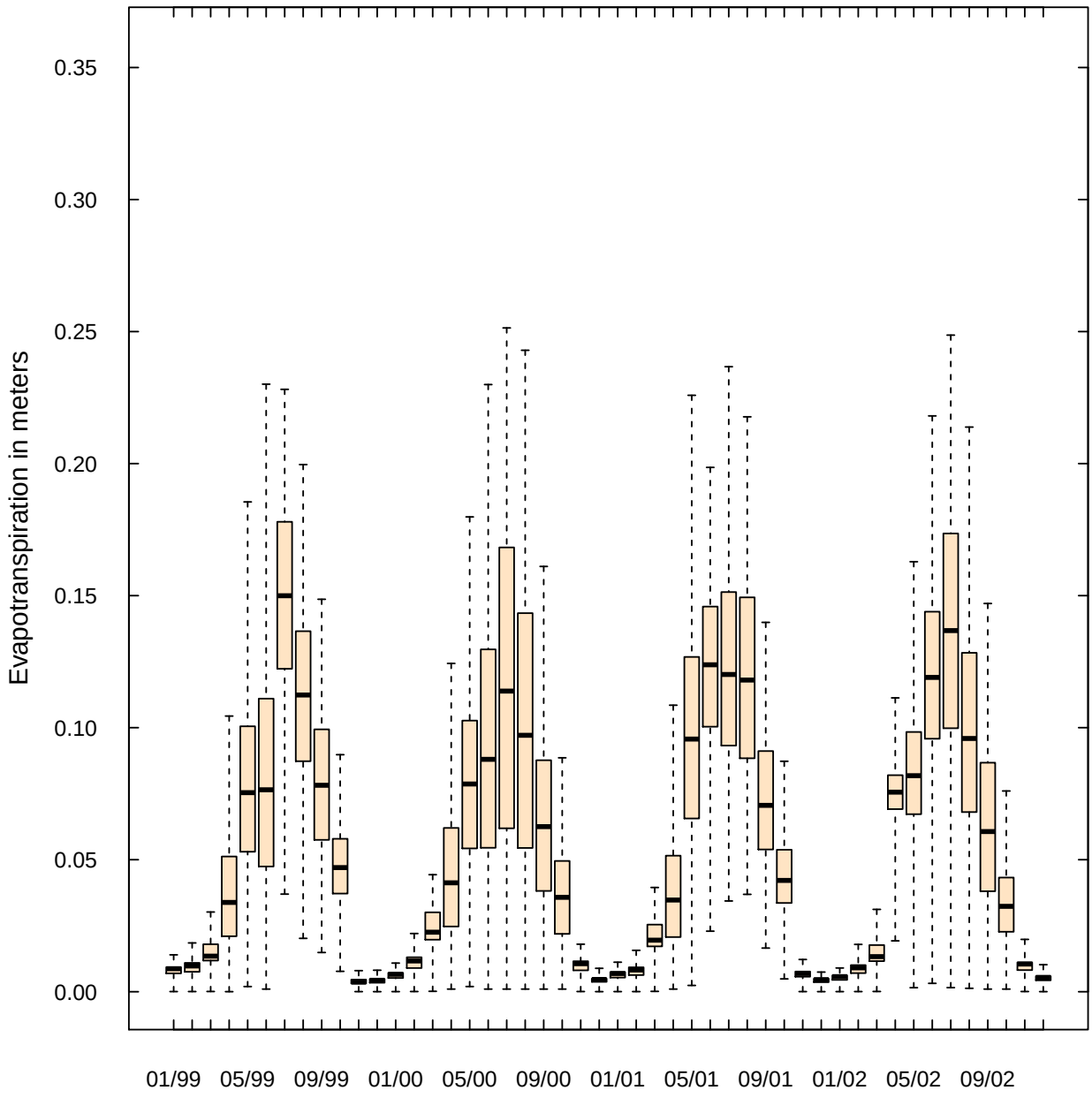

Figure 32: Evapotranspiration from 1999 through 2002 in the the Wood River Valley.

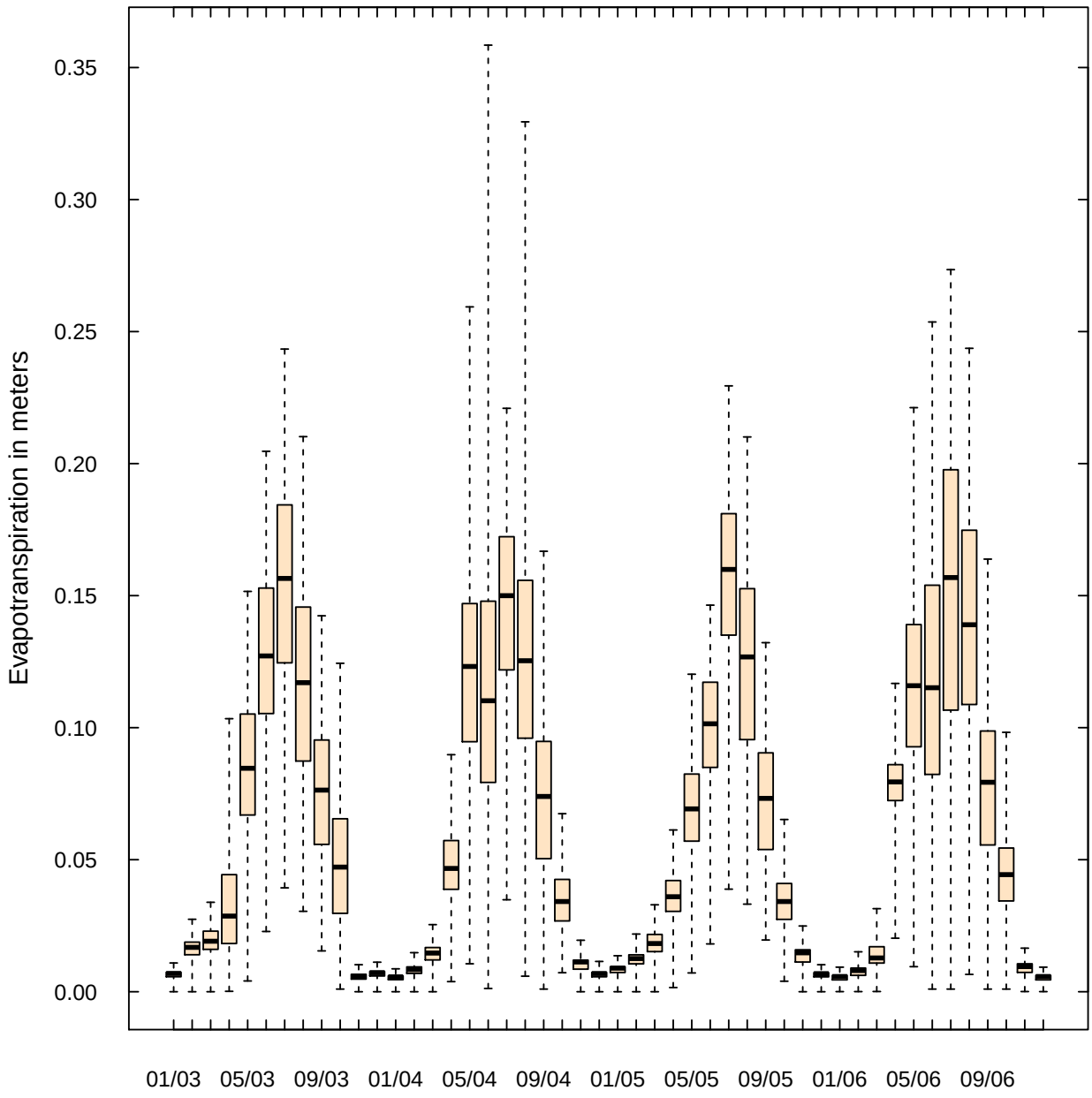

Figure 33: Evapotranspiration from 2003 through 2006 in the the Wood River Valley.

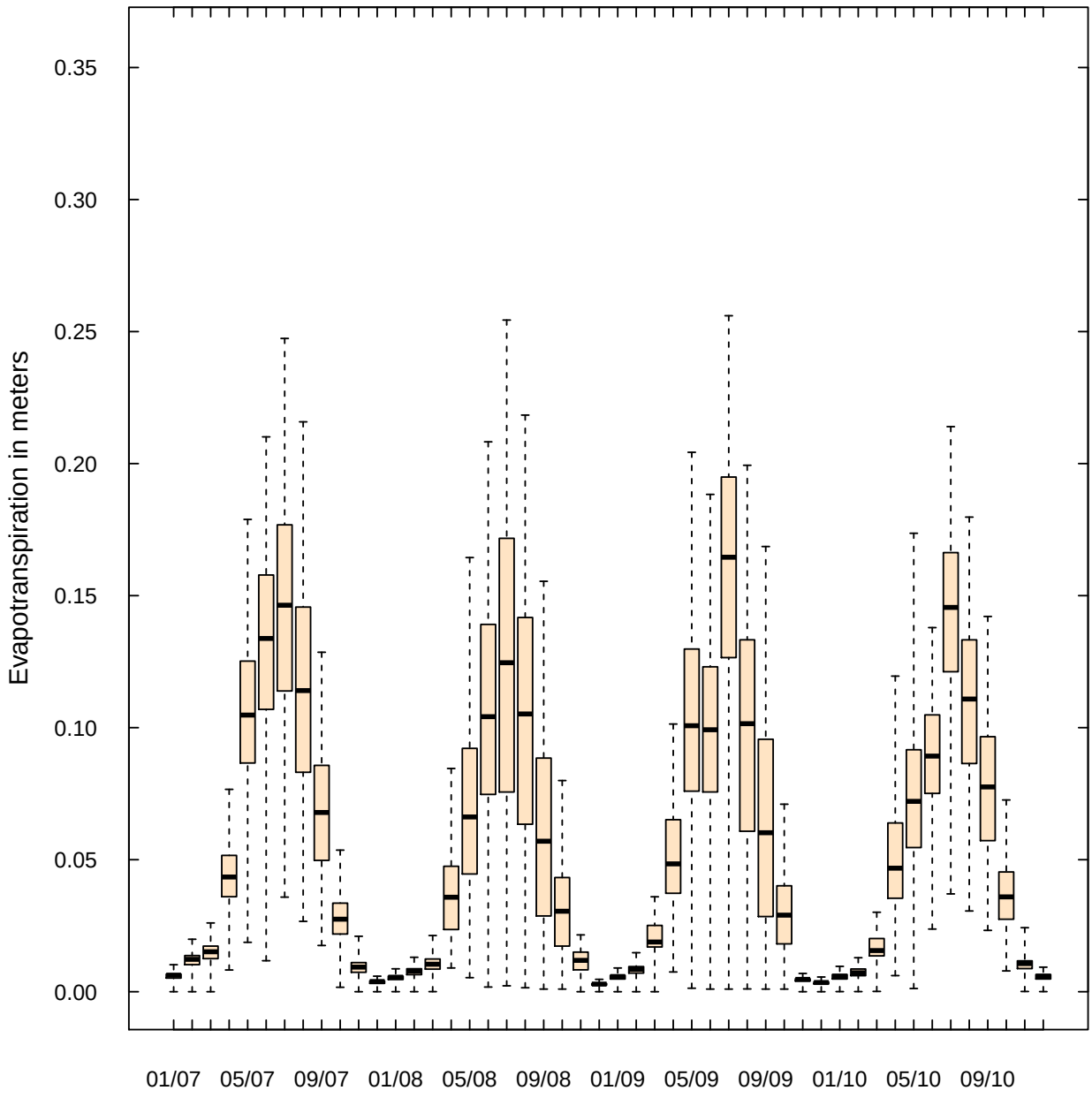

Figure 34: Evapotranspiration from 2007 through 2010 in the the Wood River Valley.

```
    is.in.zone <- d$PrecipZone == j
    idx <- which(precipitation$YearMonth == i & precipitation$PrecipZone == j)
    d$precip.vol[is.in.zone] <- d$area[is.in.zone] * precipitation$Precip[idx]
  }
  d$cir.vol <- d$et.vol - d$precip.vol
  idxs <- match(d$EntitySrce, irr.entities@data$EntitySrce)
  d[, c("EntityName", "Source")] <- irr.entities@data[idxs, c("EntityName", "Source")]
  rownames(d) <- d$EntitySrce
  return(SpatialPolygonsDataFrame(sp, d))
}
entity.components <- lapply(yr.mo, FUN)
names(entity.components) <- yr.mo
save(entity.components, file = file.path(dir.dat, "entity.components.rda"),
     compress = "xz")
```

Grids

Rasterized Canals (r.canals)

Something...

```
r <- rasterize(canals, land.surface, "EntityName", silent = TRUE)
r <- ratify(r, count = TRUE)
rat <- levels(r)[[1]]
rat$EntityName <- levels(canals@data$EntityName)[rat$ID]
levels(r) <- rat
r.canals <- r
save(r.canals, file = file.path(dir.dat, "r.canals.rda"), compress = "xz")
```

Rasterized Monthly Irrigation Entities (rs.entities)

Something...

```
FUN <- function(i) {
  r <- rasterize(entity.components[[i]], land.surface, "EntityName", silent = TRUE)
  r <- ratify(r, count = TRUE)
  rat <- levels(r)[[1]]
  rat$EntityName <- levels(entity.components[[i]]@data$EntityName)[rat$ID]
  levels(r) <- rat
  return(r)
}
rs.entities <- stack(lapply(yr.mo, FUN), quick = TRUE)
names(rs.entities) <- yr.mo
save(rs.entities, file = file.path(dir.dat, "rs.entities.rda"), compress = "xz")
```

Rasterized Monthly Recharge on Non-Irrigated Lands (rs.rech.non.irr)

Something...

```
r.zones <- ratify(rasterize(precip.zones, land.surface, "ID", silent = TRUE))
levels(r.zones) <- cbind(levels(r.zones)[[1]], att = precip.zones@data$PrecipZone)
```

```
r.soils <- rasterize(soils, land.surface, "InfRate", silent = TRUE)
FUN <- function(i) {
  p <- precipitation[precipitation$YearMonth == i, c("PrecipZone", "Precip")]
  p <- p[match(p$PrecipZone, levels(r.zones)[[1]]$att), "Precip"]
  names(p) <- levels(r.zones)[[1]]$att
  r <- r.zones
  levels(r) <- cbind(levels(r)[[1]], Precip = p)
  r <- deratify(r, "Precip") - et[[i]]
  is.pos <- r > r.soils
  r[is.pos] <- r.soils[is.pos]
  r[!is.na(rs.entities[[i]])] <- NA
  r[] <- round(r[], digits = 6)
  return(r)
}
rs.rech.non.irr <- stack(lapply(yr.mo, FUN), quick = TRUE)
names(rs.rech.non.irr) <- yr.mo
save(rs.rech.non.irr, file = file.path(dir.dat, "rs.rech.non.irr.rda"), compress = "xz")
```

Reproducibility

To recreate the **wrv**-package data sets, evaluate R code extracted from this vignette using the following command:

```
source(system.file("doc", "wrv-datasets.R", package = "wrv"), echo = TRUE)
list.files(dir.dat, full.names = TRUE)  # path names of output files
```

Version information about R and attached or loaded packages is as follows:

- R version 3.1.1 (2014-07-10), `x86_64-w64-mingw32`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: RCurl 1.95-4.3, bitops 1.0-6, raster 2.3-0, rgdal 0.8-16, rgeos 0.3-6, sp 1.0-15
- Loaded via a namespace (and not attached): evaluate 0.5.5, formatR 1.0, grid 3.1.1, highr 0.3, igraph 0.7.1, knitr 1.6, lattice 0.20-29, stringr 0.6.2, tools 3.1.1, wrv 0.1-5

Total processing time for this vignette was 9.3 hours, built on September 09, 2014.