

# *Stream Heatwaves Project: Metabolism Modeling*

*Michelle Catherine Kelly*

*2023-07-07*

## *1. Introduction*

This file walks through the process of data retrieval, processing, and metabolism analysis for 48 USGS streams, spanning January 1, 2017 through present day.

*As based on the methods defined by: Appling, A., et al. (2018). The metabolic regimes of 356 rivers in the United States. Scientific Data, 5(1). <https://doi.org/10.1038/sdata.2018.292>*

```
# Load packages
library(tidyverse) # v 1.3.2
library(dataRetrieval) # v 2.7.12

# Unit conversion helper functions
ft_m <- function(ft){
  ft/3.281
}
ft3_m3 <- function(ft3){
  ft3/35.315
}
```

## *2. Load site list*

Spencer has designated 48 USGS sites as suitable for the analysis. The main constraint here is that sites must have at least 20 years of temperature data in order to obtain accurate heatwave period estimates.

*This file contains just two columns: one for the numeric site ID codes (nwis\_id) and another for the site names (long\_name)*

```
# Load CSV file
sites <- read_csv("data/48USGS_sites.csv")
# Remove "nwis_" from the front of site numbers
sites$nwis_id <- str_extract(sites$nwis_id,
                             pattern = "\\d+")
```

## *3. Pull data*

For metabolism modeling, we need to obtain stream data from **USGS National Water Information System (NWIS)** (<https://waterdata.usgs.gov/nwis/rt>) and atmospheric data from **NASA Goddard Earth Sciences Data and Information Services Center (GES DISC)** (<https://disc.gsfc.nasa.gov/>).

### 3.1 USGS National Water Information System (NWIS)

**Variables:** Stream temperature, dissolved oxygen (DO), discharge

**URL:** <https://waterdata.usgs.gov/nwis/rt>

**R Package:** dataRetrieval (v2.7.12), written by USGS

**Parameter codes:**

USGS		
Code	Description	Units
00010	Water temperature	°C
00060	Discharge	ft <sup>3</sup> s <sup>-1</sup>
00065	Gage height	ft
00300	Dissolved oxygen	mg L <sup>-1</sup>

```
# Remove "nwis_" from the front of site ID strings
sites$nwis_id <- str_extract(sites$nwis_id,
                             pattern = "\\d+")

## Set parameters for data pull #####
# USGS parameter codes
parameters <-
  data.frame(code = c("00010", "00060", "00065", "00300"),
             description = c("Water temperature",
                             "Discharge", "Gage height",
                             "DO"),
             unit = c("degC", "ft3s", "ft", "mgL"))

# Start date
startD <- "2017-01-01"

# End date
endD <- "2023-06-01"

# Initialize empty list to store data
usgsVector <- vector(mode = "list",
                     length = length(sites$nwis_id))

# Initialize i
i <- 1

# For loop to pull site data from USGS API
for(i in seq_along(sites$nwis_id)){
  # Print i so user knows what segment we're on
  message("\n", i, ".\n-----")

  # Get current usgs site number & name
  siteNo <- sites$nwis_id[i]
  siteName <- sites$long_name[i]
```

```

# Pull instantaneous value (uv) data from USGS server
siteData <- readNWISuv(siteNumbers = siteNo,
                      parameterCd = parameters$code,
                      startDate = startD, endDate = endD)

# Check for errors in data pull before proceeding
if(nrow(siteData) == 0){
  # Tell user an informative error message - no
  # data present in the data frame
  message("ERROR: API returned no data for ", siteName,
          "\n(NWIS id: ", siteNo, "). This may indicate ",
          "that the station was\nretired, or this may ",
          "be due to an API error. \nPlease investigate.",
          "\n-----\n")
  # Exit and go on to next site as errors are present
  next
} else{
  # Rename columns into more human friendly versions
  siteData <- renameNWISColumns(siteData)

  # Save site Info for console output
  siteInfo <- attr(siteData, "siteInfo")
  # Tell user site data has finished downloading
  message("Data for ", siteInfo$station_nm,
          "\n(NWIS id: ", siteInfo$site_no,
          ") has been downloaded.\n")

  usgsData <-
    siteData %>%
      # Add time zone code to dateTime column
      mutate(dateTime =
        lubridate::ymd_hms(dateTime,
                           tz = unique(tz_cd))) %>%
      # Filter out unneeded columns
      select(-any_of(c("agency_cd", "tz_cd", "test")),
            -ends_with("_cd"))

  # Check - do we have all of the expected variables
  # in the dataset?
  varList <- c("site_no", "dateTime", "Wtemp_Inst",
              "Flow_Inst", "GH_Inst", "DO_Inst")
  # Are there any missing variables in the pulled data?
  missingVars <- varList[!(varList %in% names(usgsData))]

```

```

if(length(missingVars) != 0){
  # Return error to user
  message("ERROR: Data does not have all variables",
    " requested. Site\nis missing the following: ",
    paste0(missingVars, sep = ", "),
    "\nPlease investigate.",
    "\n-----\n")
  # Exit and go on to next site as errors are present
next
} else{
  # Rename columns to include units because I will
  # immediately forget them
  usgsData <-
    usgsData %>%
    # Convert from US units to Sci units
    mutate(Flow_Inst = ft3_m3(Flow_Inst),
      GH_Inst = ft_m(GH_Inst)) %>%
    # Filter to just 15-min data to sliiiiightly
    # decrease dataset size
    filter(lubridate::minute(dateTime)
      %in% c(0, 15, 30, 45)) %>%
    rename(Wtemp_C = Wtemp_Inst, Q_m3s = Flow_Inst,
      GageHt_m = GH_Inst, DO_mgL = DO_Inst)

  # Save variable info for console output
  varibInfo <- names(usgsData)
  # Report which variables were returned by data pull
  message("\nVariables returned by data pull:\n",
    paste0(varibInfo, sep = ", "),
    "\nRecord starts: ", first(usgsData$dateTime),
    "\nRecord ends: ", last(usgsData$dateTime),
    "\n-----\n")

  # Add site data to vector
  usgsVector[[i]] <- usgsData
}
}
}

```

#### Unsuitable sites for modeling from 2017 - present:.

Site 6. WATEREE R. BL EASTOVER, SC (NWIS id: 02148315).  
 Site was retired in 2014.

Site 18. DUCK RIVER AT SHELBYVILLE, TN (NWIS: 03597860).  
 Water temperature and dissolved oxygen sensors retired in 2012.

Site 19. WILSON PARK CK @ ST. LUKES HOSPITAL (NWIS: 040871488). Water temperature and dissolved oxygen sensors retired in 2017.

Site 46. SANTA YNEZ R NR SANTA YNEZ CA (NWIS id: 11126000)

Site 47. SACRAMENTO R A FREEPORT CA (NWIS id: 11447650)

```
# Unlist vector
```

```
# Save dataset
```

### 3.2 NASA GES DISC: Solar radiation, air pressure

Dataset: North American Land Data Assimilation System Phase 2 (NLDAS-2)

NASA dataset code: NLDAS\_FORA0125\_H.2.0 URL: [https://disc.gsfc.nasa.gov/datasets/NLDAS\\_FORA0125\\_H\\_002/summary](https://disc.gsfc.nasa.gov/datasets/NLDAS_FORA0125_H_002/summary) (DOI: 10.5067/6J5LHHOHZHN4)

1. Retrieved list of URLs for data pull using “Subset/Get data” button at link above
  - Time span: 2017-01-01 to 2023-06-01
2. Downloaded data files from GES DISC portal using the URL list and wget
  - Tutorial here: <https://disc.gsfc.nasa.gov/information/howto?title=How%20to%20Access%20GES%20DISC%20Data%20Using%20wget%20and%20curl>

Later steps: - unite with solar radiation and air pressure data - calculate DO saturation % based on temp & pressure, see mimsy eqns