GDD related stuff

Victor, Lizzie

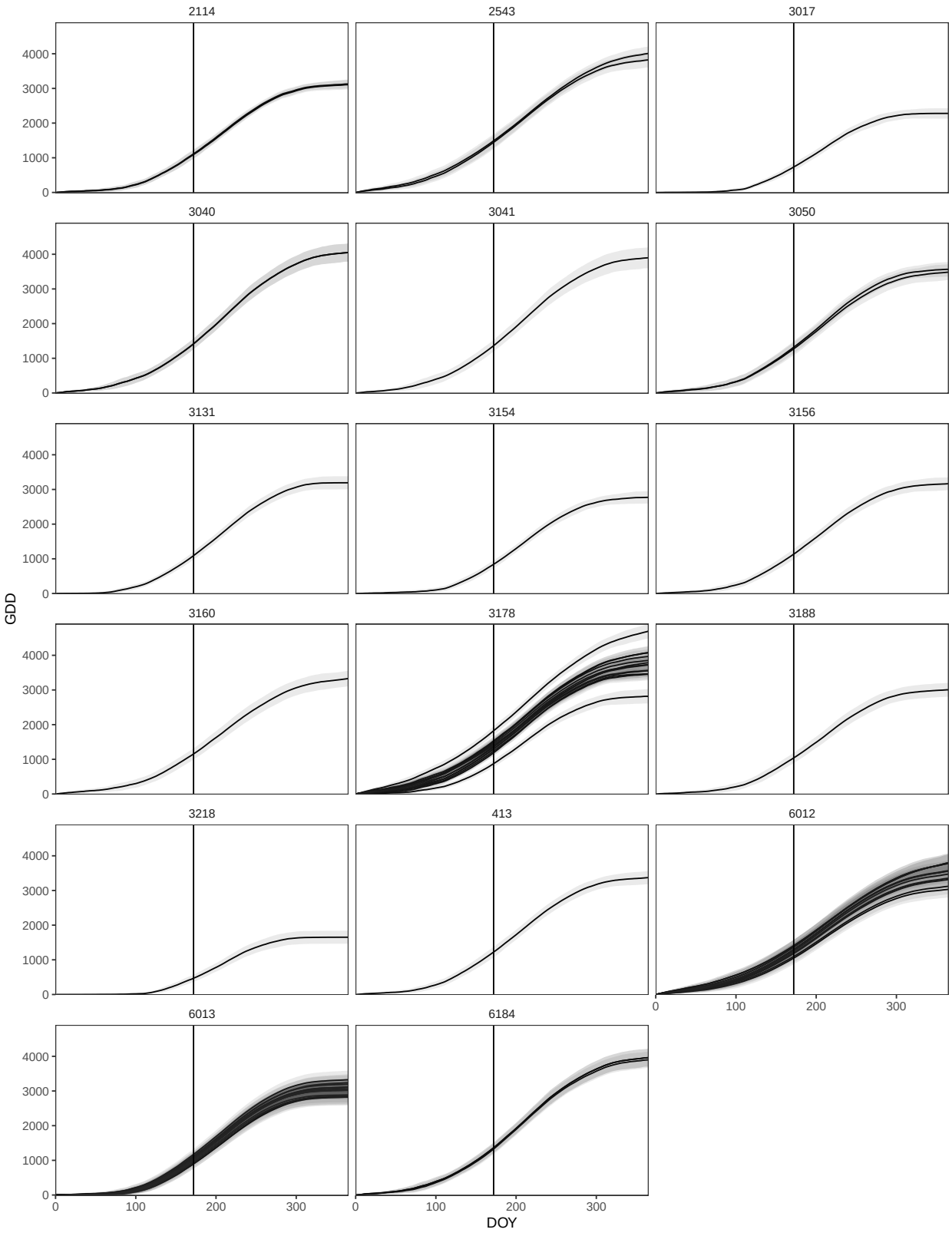
2024-08-26

```
process_data <- readRDS(file.path(wd, "data/processed", "journe2024_replicate.rds"))

gdd_data <- process_data %>%
  dplyr::group_by(sitenewname, Year) %>%
  dplyr::mutate(tmean_filt = ifelse(Tmean <= 35 & Tmean >=5, Tmean, 0),
                gdd = cumsum(tmean_filt)) %>%
  dplyr::select(-tmean_filt) %>%
  ungroup() %>%
  mutate(doy = as.numeric(strftime(Date, format = "%j")),
         source = stringr::str_split(sitenewname, "_", simplify = T)[, 1])
```

```
gdd_data %>%
  dplyr::group_by(source, sitenewname,doy) %>%
  summarise(gddm = mean(gdd), sd = sd(gdd)) %>%
  ggplot(aes(x = doy, y = gddm, group = sitenewname)) +
  facet_wrap(~ source, ncol = 3) +
  geom_line() +
  geom_ribbon(aes(ymin = gddm-sd, ymax = gddm+sd), alpha = 0.1) +
  theme_bw() +
  theme(legend.position = 'none',
        strip.background = element_blank(),
        panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
  labs(y = "GDD", x = "DOY")+
  geom_vline(xintercept = 172) +
  coord_cartesian(xlim = c(0,365), expand = FALSE)
```

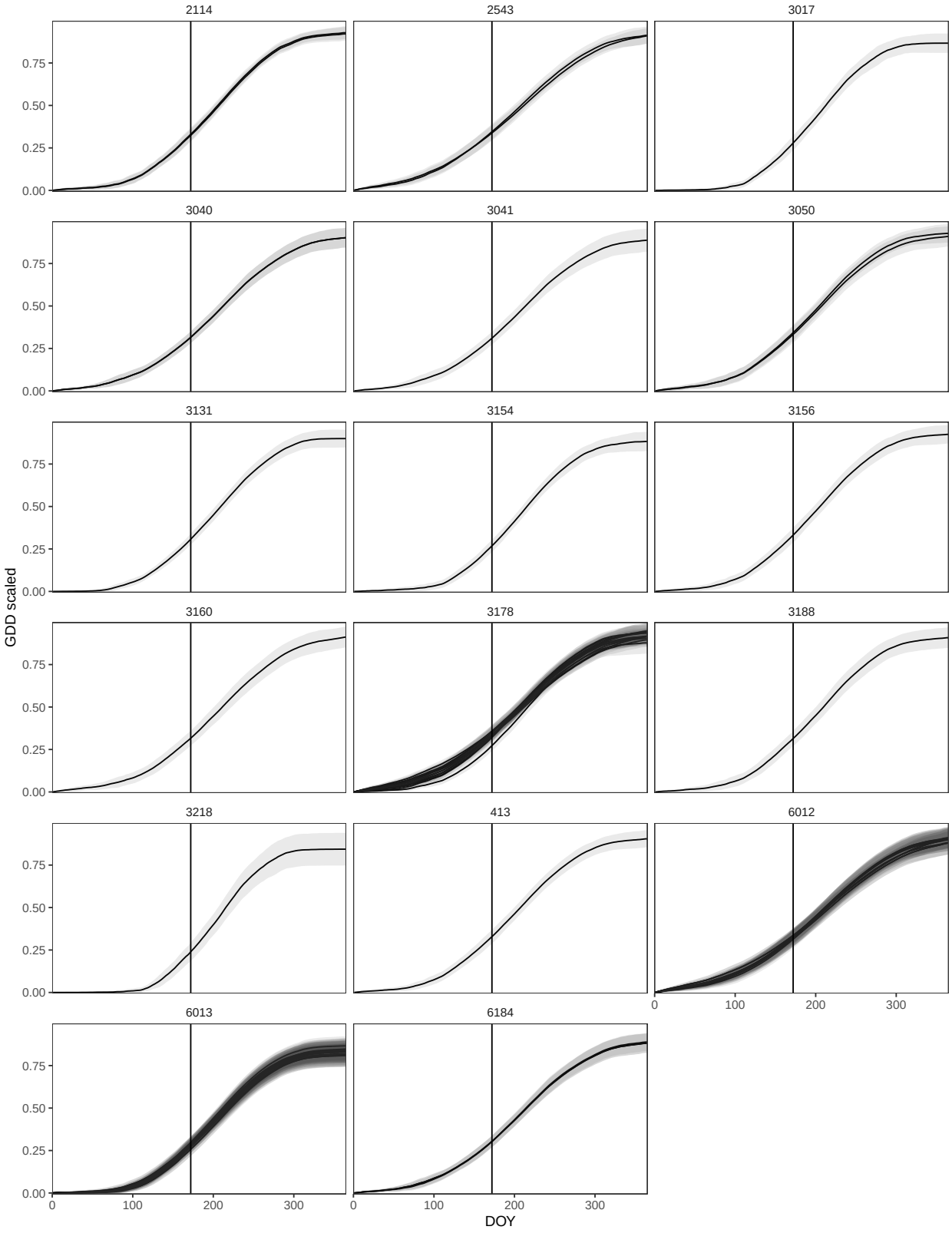
```
## `summarise()` has grouped output by 'source', 'sitenewname'. You can override
## using the `.groups` argument.
```




```
gdd_data %>%
  dplyr::group_by(sitenewname) %>%
```

```
  mutate(maxv = max(gdd), minv = min(gdd),
         gdd_sc = (gdd-minv)/(maxv-minv)) %>%
  dplyr::group_by(source,sitenewname,doy) %>%
  summarise(gddm = mean(gdd_sc), sd = sd(gdd_sc)) %>%
  ggplot(aes(x = doy, y = gddm, group = sitenewname)) +
  facet_wrap(~ source, ncol = 3) +
  geom_line() +
  geom_ribbon(aes(ymin = gddm-sd, ymax = gddm+sd), alpha = 0.1) +
  theme_bw() +
  theme(legend.position = 'none',
        strip.background = element_blank(),
        panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
  labs(y = "GDD scaled", x = "DOY")+
  geom_vline(xintercept = 172) +
  coord_cartesian(xlim = c(0,365), expand = FALSE)
```

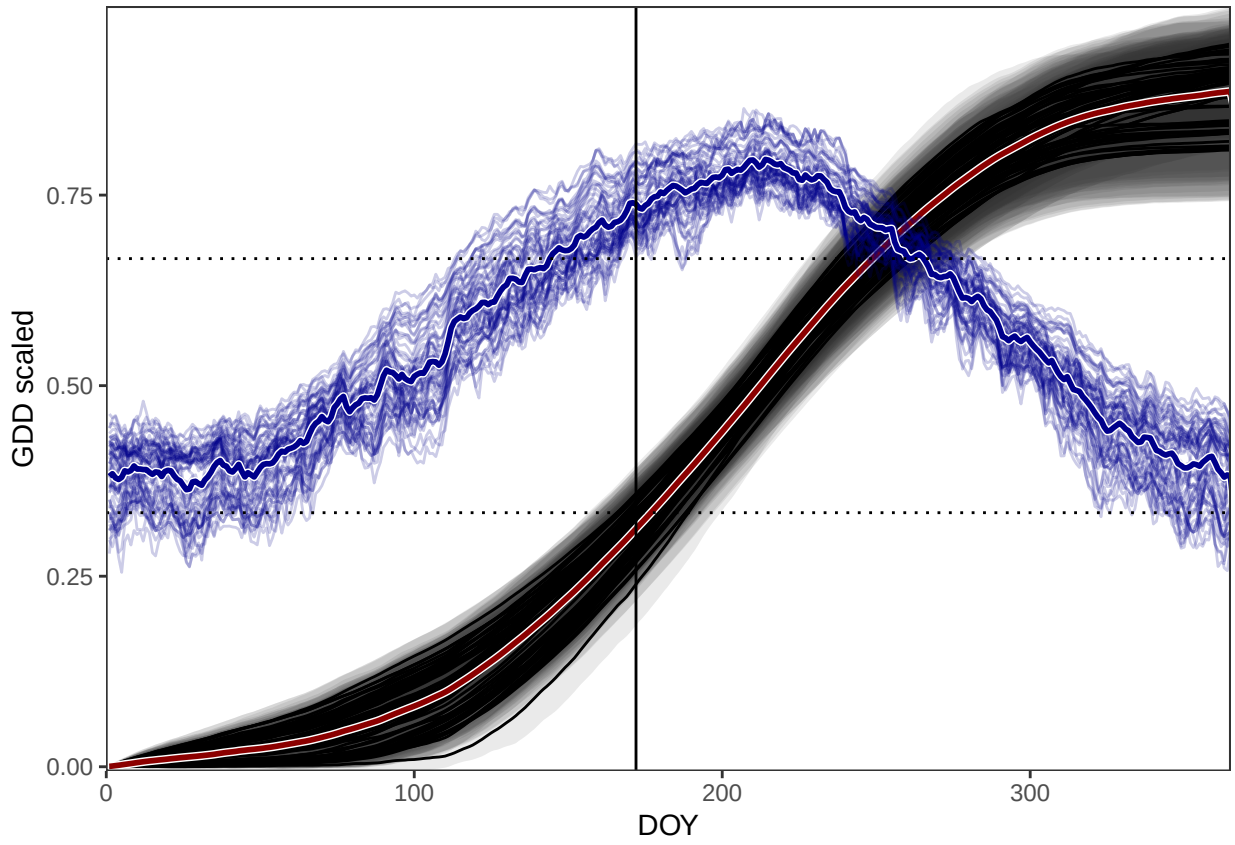
```
## `summarise()` has grouped output by 'source', 'sitenewname'. You can override
## using the `.groups` argument.
```




```
gdd_data %>%
  dplyr::group_by(sitenewname) %>%
```

```
  mutate(maxv = max(gdd), minv = min(gdd),
         gdd_sc = (gdd-minv)/(maxv-minv),
         maxt = max(Tmean), mint = min(Tmean),
         tmean_sc = (Tmean-mint)/(maxt-mint)) %>%
  dplyr::group_by(source,sitenewname,doy) %>%
  summarise(gddm = mean(gdd_sc), sd = sd(gdd_sc), tmean_sc = mean(tmean_sc)) %>%
  ggplot(aes(x = doy, y = gddm, group = sitenewname)) +
  geom_ribbon(aes(ymin = gddm-sd, ymax = gddm+sd), alpha = 0.1) +
  geom_line() +
  stat_summary(aes(group=1), fun=mean, geom="line", colour="white", linewidth = 1.5) +
  stat_summary(aes(group=1), fun=mean, geom="line", colour="darkred", linewidth = 1) +
  geom_line(aes(y = tmean_sc), color = "darkblue", alpha = 0.2) +
  stat_summary(aes(group=1, y = tmean_sc), fun=mean, geom="line", colour="white", linewidth = 1.5) +
  stat_summary(aes(group=1, y = tmean_sc), fun=mean, geom="line", colour="darkblue", linewidth = 1) +
  theme_bw() +
  theme(legend.position = 'none',
        strip.background = element_blank(),
        panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
  labs(y = "GDD scaled", x = "DOY")+
  geom_vline(xintercept = 172) +
  geom_hline(yintercept = c(1/3,2/3), linetype = "dotted") +
  coord_cartesian(xlim = c(0,365), expand = FALSE)
```

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



Get ERA5lAND P-PET balance

```
library("doFuture")
registerDoFuture()
plan(multisession, workers = 10)

clim_dir <- "D:/climate/ERA5-Land/phenofit_format/transformed"

rerun <- FALSE # switch because computation is too long

gdd_data$daily_wbal <- NA
foreach(s = unique(gdd_data$sitenewname), .export = "gdd_data") %dopar% {
  print(s)
  lat <- unique(gdd_data[gdd_data$sitenewname == s, "Latitude"]) %>% as.numeric()
  lon <- unique(gdd_data[gdd_data$sitenewname == s, "Longitude"]) %>% as.numeric()
  years <- unique(gdd_data[gdd_data$sitenewname == s, "Year"]) %>% unlist()

  if(s == "6012_2_FAGSYL"){
    lat = lat + 0.1
  }

  for(y in years){

    pr <- fread(file.path(clim_dir, paste0("ERA5LAND_pre_",y,"_dly.fit"))) %>% as.data.frame()
    pet <- fread(file.path(clim_dir, paste0("ERA5LAND_pet_",y,"_dly.fit"))) %>% as.data.frame()

    ind <- which(unlist(pr[,1]) == round(lat,1) & unlist(pr[,2]) == round(lon,1))

    wbal <- t(pr[ind, 3:ncol(pr)]-pet[ind, 3:ncol(pet)])

    gdd_data[gdd_data$sitenewname == s & gdd_data$Year == y, "daily_wbal"] <- wbal

  }

}









if(rerun){
  eobs_r <- rast(file.path(wd, "data/eobs", "tg_ens_mean_0.1deg_reg_v29.0e.nc"))

  # Loop on sites to get E-OBS daily temperature for the years recorded
  process_data <- data.frame()
  for(s in unique(journe_d$sitenewname)){

    lat <- unique(journe_d[journe_d$sitenewname == s, "Latitude"])
    lon <- unique(journe_d[journe_d$sitenewname == s, "Longitude"])
    years <- journe_d[journe_d$sitenewname == s, "Year"]
```

```
    years <- c(min(years)-2, min(years-1), years) # two years before seed fall

    sub_r <- subset(eobs_r, which(time(eobs_r, format = "years") %in% years))
    tmean <- terra::extract(
      sub_r,
      matrix(c(lon, lat), nrow = 1))

    process_data <- rbind(process_data, data.frame(Latitude = lat, Longitude = lon, sitenewname = s, Tm
                                                   Date = time(sub_r), Year = time(sub_r, format = "yea
  }

  # Add annual seed production (including one and two years later) -- not a very elegant way
  journe_T2 <- journe_d[,c("Longitude", "Latitude", "sitenewname", "Year", "Value")]
  journe_T2$Year <- journe_T2$Year-2
  names(journe_T2)[5] <- "Value2"
  journe_T1 <- journe_d[,c("Longitude", "Latitude", "sitenewname", "Year", "Value")]
  journe_T1$Year <- journe_T1$Year-1
  names(journe_T1)[5] <- "Value1"
  process_data <- process_data %>%
    dplyr::left_join(journe_d[c("Longitude", "Latitude", "sitenewname", "Year", "Value")],
                     by = c("Longitude", "Latitude", "sitenewname", "Year")) %>%
    dplyr::left_join(journe_T1[c("Longitude", "Latitude", "sitenewname", "Year", "Value1")],
                     by = c("Longitude", "Latitude", "sitenewname", "Year")) %>%
    dplyr::left_join(journe_T2[c("Longitude", "Latitude", "sitenewname", "Year", "Value2")],
                     by = c("Longitude", "Latitude", "sitenewname", "Year"))

  saveRDS(process_data, file = file.path(wd, "data/processed", "journe2024_replicate.rds"))

}else{

  process_data <- readRDS(file.path(wd, "data/processed", "journe2024_replicate.rds"))

}
```