Norwegian University
of Life Sciences

**Masters Thesis 2024 30p**
Faculty of Science and Technology

# A comparative study of soil temperature models, including machine learning models

Mats Hoem Olsen
Computer science

# 1. FORWORD

I would like to thank my advisors and friends. Also the Big Bang for happening.

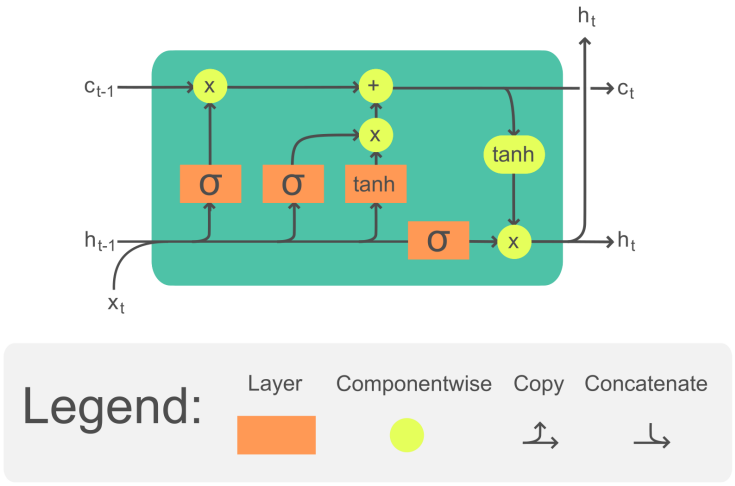



Fig. 1: LSTM cell Artist: chevalier$_e nglish_2$018

## 2. THEORY

### 2.1. Regression model

The regression model will be for the sake of convenience be expressed as the following expresion

$$\left(\vec{F} \circ \mathbf{A}\right) \vec{\beta} = \vec{y} + \vec{\varepsilon}$$

Where $\vec{F}$ is a vector function with following domain $\vec{F} : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times p}$ where $m, p \in \mathbb{N}$, $\mathbf{A}$ is the data in matrix form with dimentions $\mathbb{R}^{m \times n}$, $\vec{\beta}$ is the regression terms, $\vec{y}$ is the target (TJM), and $\vec{\varepsilon}$ is the error from modelling. The $\circ$ operator is the composition of $\vec{F}$ and $\mathbf{A}$, is a short way of writing $\vec{F}(\mathbf{A})$.

### 2.2. Long Short Term Memory model

# 3. METHOD

## 3.1. Source of data

For this comparative study the following data sourses will be used

1. Norwegian Institute of Bioeconomy Research LandbruksMeteorologisk service (LMT)
2. Xgeo
3. Norwegian Institute of Bioeconomy Research Kilden (Kilden)
4. The Norwegian Meteorological Institute (MET)

## 3.2. Dataset

The dataset is chosen from four regions in Norway; Innlandet, Vestfold, Trøndelag, and Østfold. From each region are four stations picked:

| | | | |
|---|---|---|---|
| Innlandet | 1. Kise | Trøndelag | 1. Kvithamar |
| | 2. Ilseng | | 2. Rissa |
| | 3. Apelsvoll | | 3. Frosta |
| | 4. Gausdal | | 4. Mære |
| Østfold | 1. Rygge | Vestfold | 1. Lier |
| | 2. Rakkestad | | 2. Ramnes |
| | 3. Tomb | | 3. Tjølling |
| | 4. Øsaker | | 4. Sande |

All stations are sampled from the date[1] 03-01 to 10-31 from 2016 to 2020. The features rain (RR), mean soil temperature at 10cm (TJM10), mean soil temperature at 20cm (TJM20), and air temperature at 2m (TM) are sampled from the LMT database. The snow parameter is sampled from MET via Xgeo for imputed values in areas where there are no messured values. The soil type, and soil texture is sampled from Kilden from Norwegian Institute of Bioeconomy Research.

## 3.3. Selection process

The selection process for finding these station can be compiled into these steps

1. Recommendation from Norwegian Institute of Bioeconomy Research
2. Compute the missing values in the data
3. Missing values analyse
4. Searching LMT database for alternative station candidates if current data is insufficient
5. If some station was replaced the repeat step 2

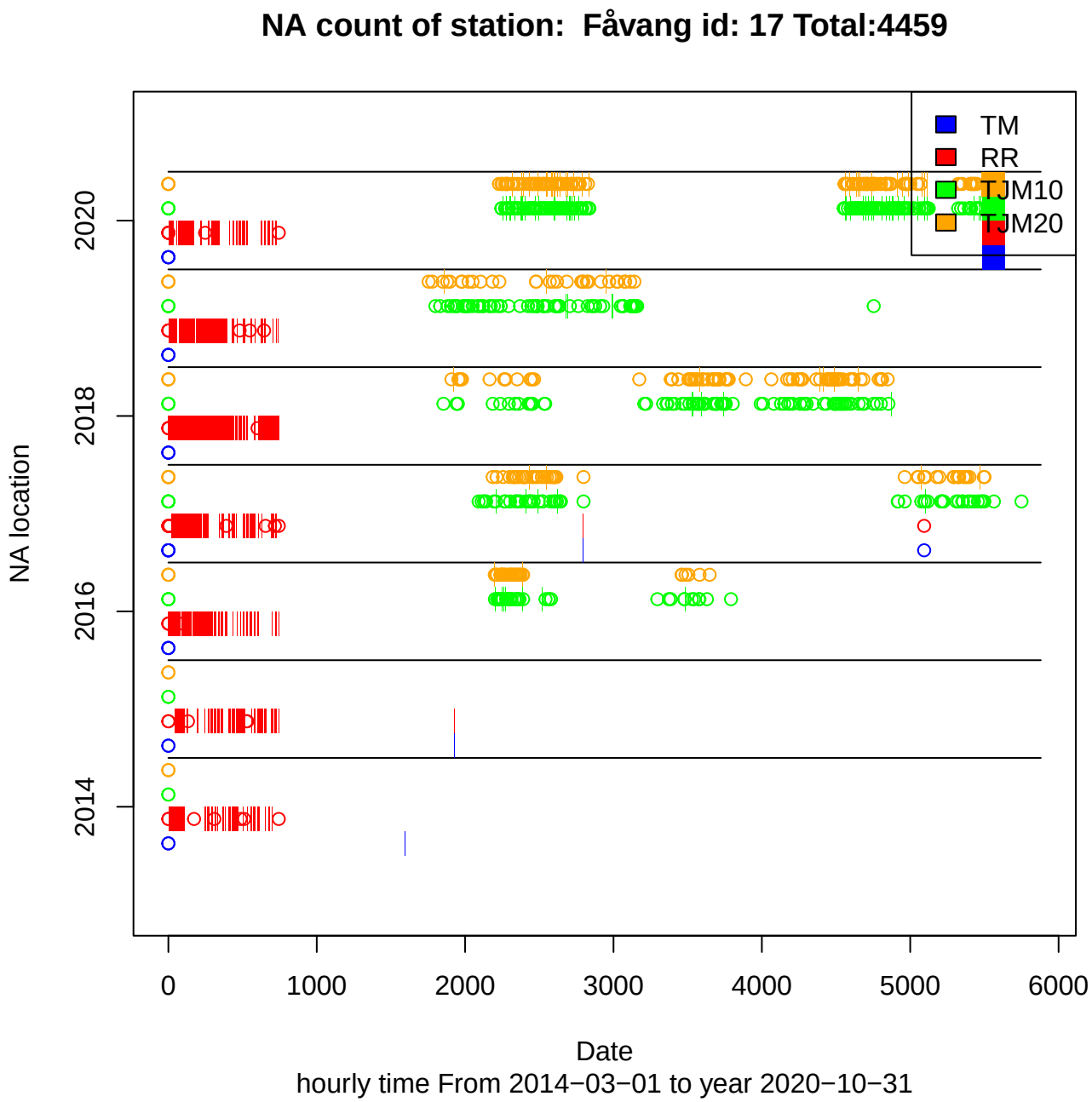



Fig. 2: Visual representation of missing values at station 17 from 2014 to 2020

The plots of stations follow a simple representation where the y-axis represent the year and the x-axis represent the index of the data as all tables are taken from the same period. A circle represent a singluar na values, while a band represent a series of 2 or more missing values. The colours represents the features used in this comperative study. This representation of the missing values will indicate sesonal, and systematic removal of data and give an overall indication of how much data is missing. To get further insight into the data a report is generated in parallel to the plots describing precisie date and time of all values and which other parameter values is also missing values in the same period. See appendix A.2 for the full detail of the report generation and appendix B for na-plots of the station chosen for this study.

### 3.4. Collection of data

The method used was a powershell[2] script that called the respective institutions servers using the "curl" program[3] to send an http request for the timeseries starting from 2014 to 2020 in the interval 1 of May to 31 of October. Code for data collection can be viewed in appendix A.1. The data is stores as an either a csv file or a json file for easy retrieval and manual control of values.

### 3.5. Setup of models

The models are set up in according to the relevant paper the model is fetched from, alternativly reuse the code made by the author. When importing the data to the model there will be modifying to the original code to facilitate for the model as far as it goes. Any modifications will be in the appendix under section A. The details of the models will be discused in section ??

[1]Format month-day
[2]Version 7.3.11
[3]curl 8.4.0 (Windows) libcurl/8.4.0 Schannel WinIDN

## A.1. Powershell

```
$baseUri = 'https://lmt.nibio.no/agrometbase/showweatherdata.php'
$datapath = "$($PSScriptRoot)/../../data/raw_data/nibio"

$line = Get-Content -Path "$($PSScriptRoot)/../../PRIVATE_FILES/frost_met_client.txt"
$FrostID = $line.Split(": ")[1]
$bases = @(
10 , 11 , 12 , 145 , 143 , 13 , 86 , 133 , 14 , 127 , 140 , 15 , 16 , 17 , 18 , 19 ,
)

$jobs = @()

foreach ($base in $bases) {
        foreach ($year in 2014..2022) {
                $full_path = "$($datapath)/weather_data_raw_hour_stID$($base)_y$($yea
                if(Test-Path $full_path -PathType Leaf){
                        continue
                }
    $jobs += Start-ThreadJob -Name "w$($base)-y$($year)" -ScriptBlock {
                        param($base, $baseUri, $year, $storage)
                        $form = @{
                                weatherstation=$base
                                loginterval=1
                                valuetype="value_raw"
                                date_start="$($year)-03-01"
                                date_end="$($year)-03-31"
                                format="csv"
                                separator="dot"
                        }

                        $Uri = "$($baseUri)?"

                        $Uri += "weatherstation=$($form["weatherstation"])&"

                        foreach ($el in @(1,297,6,7)) { # 1, 297 -< temp, nedbør
                                $Uri += "elementMeasurementTypes%5B%5D=$($el)&"
                        }

                        foreach ($key in @("loginterval","valuetype","date_start","da
                                $Uri += "$($key)=$($form[$key])&"
                        }
                        $Uri = $Uri.Substring(0,$Uri.length-1)
                        Write-Host $Uri
                        curl $Uri --output $storage --retry 3 --retry-delay 5
    } -ArgumentList $base, $baseUri, $year, $full_path
                Write-Host "Written w$($base)-y$($year)."
```

```
        }
}
if($jobs.length -eq 0) {
        Write-Host "No jobs"
} else {
        Write-Host "Downloads started..."

        Wait-Job -Job $jobs

        foreach ($job in $jobs) {
                        Receive-Job -Job $job
        }
}
```

A.2. R

```
## ------------------------------------------------------------------------
library(dplyr) # for data manipulation and transformation
library(tidyverse) # for a collection of packages for data manipulation and visualiza
library(stats) # for statistical functions and models
library(tsfeatures)
library(lubridate)
library(runner)

library(TSdist) # for calculating distance measures between time series
library(forecast) # for time series forecasting
library(TSA) # for time series analysis
library(tseries)
library(signal)
library(imputeTS)

library(ggplot2) # for creating beautiful and customizable visualizations
library(gridExtra) # for arranging multiple plots on a grid
library(RColorBrewer) # for creating color palettes for your plots
library(MLmetrics)
library(summarytools)


## ------------------------------------------------------------------------
# path definitions

ROOT <- "../../"

DATA_PATH <- paste0(ROOT,"data/")

DATA_INFO <- paste0(DATA_PATH,"info/")
DATA_INFO_NIBIO_FILE <- paste0(DATA_INFO ,"lmt.nibio.csv")
DATA_INFO_FROST_FILE <- paste0(DATA_INFO,"Frost_stations.csv")
DATA_FILE_SOIL_STATIONS <- paste0(DATA_INFO,"'Stasjonsliste␣jordtemperatur␣modellerin
```

```
DATA_COLLECTION <- paste0(DATA_PATH,"raw_data/")
DATA_COLLECTION_STAT <- paste0(DATA_COLLECTION,"Veret␣paa␣Aas␣2013-␣2017/") # pattern
DATA_COLLECTION_TIME <- paste0(DATA_COLLECTION,"Time␣2013-␣2023/") # pattern -> Time{
DATA_COLLECTION_NIBIO <- paste0(DATA_COLLECTION,"nibio/") # pattern -> weather_data_h

# ID definitions

station_names <- read.csv(DATA_INFO_NIBIO_FILE,
                          header=TRUE,
                          row.names="ID",
                          colClasses=c(ID="integer",Navn="character"))

nibio_id = list(
    Innlandet = c(11,17,18,26,27),
    Trøndelag = c(15,57,34,39,43),
    Østfold = c(37,41,52,118,5),
    SørVestlandet = c(14,29,32,48,22),
    Vestfold = c(30,38,42,50)
)

# function definitions

file_name.nibio <- function(station_id, year, path = NULL){
    if(is.null(path)){
        pattern = paste0(DATA_COLLECTION_NIBIO,"weather_data_hour_stID",station_id,"_
    } else {
        pattern = sprintf(path,station_id,year)
    }
    return(pattern)
}

data.nibio <- function(station_id,year, path = NULL){
    path <- file_name.nibio(station_id,year, path = path)
    data_nibio <- read.csv(path,
                           header=T, col.names = c("Time","TM","RR","TJM10","TJM20"))
    data_nibio <- mutate(data_nibio,across(
                                        "Time",
                                       str2date))
    data_nibio <- column_to_rownames(data_nibio, var = "Time")
    data_nibio <- mutate_at(data_nibio,c("TM","RR","TJM10","TJM20"), as.numeric)
    return(data_nibio)
}
na.interpol.cust <- function(data, maxgap = Inf, n.p,
                             s.window = 10, alg.option = "linear"){
    data.decomp <- stlplus::stlplus(data,n.p = n.p, s.window = s.window)
    data.new <- rep(0,length.out = length(data))
    for(part in c("seasonal", "trend", "remainder")){
        data.new <- data.new + na_interpolation(data.decomp$data[,part],
                                                maxgap=maxgap,
```

```
                                                    option = alg.option)
    }
    return(data.new)
}
str2date <- function(x) {
    return(as.POSIXlt(paste0(x,"00"),
                      format = "%Y-%m-%d %H:%M:%S%z",
                      tz="GMT"))
}

na.interplol.kal <-function(data, maxgap = Inf, n.p,
                            s.window = 10, alg.option = "StructTS"){
    data.decomp <- stlplus::stlplus(data,n.p = n.p, s.window = s.window)
    data.new <- rep(0,length.out = length(data))
    for(part in c("seasonal", "trend", "remainder")){
        data.new <- data.new + na_kalman(data.decomp$data[,part],
                                         maxgap=maxgap,
                                         model = alg.option,
                                    smooth = TRUE)
    }
    return(data.new)
}

find.na.index.length <- function(x){ # antar at x er bool vektor
    i <- 1 # starting index
    na.data <- data.frame()
    while(i <= length(x)){
        sample.data <- x[i:length(x)]
        first <- match(T, sample.data, nomatch = -1)
        if(first < 0) {
            break
        }
        last <- match(F, sample.data[first:length(sample.data)], nomatch = length(sam

        na.data <- rbind(na.data, data.frame(Length = c(last-first + 1), First = c(fi
        i <- i + last
    }
    return(na.data)
}


## ------------------------------------------------------------------------
blocks.index <- c()
len.na <- 8
len.val <- 12

data.check <- 1:5880
i <- 0
while(i < 5880){
```

```
    i <- i + len.val - 1
    blocks.index <- append(blocks.index,seq(i,i+len.na-1))
    i <- i + len.na
}
blocks.index <- blocks.index[blocks.index <= 5880]


## ------------------------------------------------------------------------
#library(moments)
data_nibio_no_na <- data.nibio(14,2019)
col.name <- "TM"

faulty.data <- data_nibio_no_na
faulty.data[blocks.index,col.name] <- NA

fixed.data <- na_interpolation(faulty.data[,col.name], option="spline", method = "per
abs.diff <- fixed.data - data_nibio_no_na[,col.name]
print(paste("μ",mean(abs.diff),"std:",sqrt(var(abs.diff)),"skewness:",skewness(abs.di
plot((abs.diff),xlim = c(0,5880))

fixed.data <- na.interpol.cust(faulty.data[,col.name], n.p = 21,alg.option="spline",
abs.diff <- fixed.data - data_nibio_no_na[,col.name]
print(paste("μ",mean(abs.diff),"std:",sqrt(var(abs.diff)),"skewness:",skewness(abs.di
plot((abs.diff),xlim = c(0,5880))


## ------------------------------------------------------------------------
# RR hadde ikke noe serlig, men hadde en rep ~= 31 (måned baser?)
# TM ~= 24?
# TJM10 ~= 24?
# TJM20 ~= 21?
perid <- c(TM = 24,TJM10 = 24, TJM20 = 24, RR = 31)

data.rle <- rle(is.na(data_nibio[,"TJM20"]))
data.max <- max(data.rle$lengths[data.rle$values])
indexes <- find.index.rle.bool(data.rle,data.max)
print(data.max)

for(col in c("TJM20")){
    imput <- as.ts(na.interpol.cust(data_nibio[,col],n.p=perid[col]))
    plot(imput,xlim = c(indexes[1]-100,indexes[2]+100))
    abline(v=indexes[1],col = "red")
    abline(v=indexes[2],col = "red")
    title(paste(col,"STL + naive"))
}

for(col in c("TJM20")){
    imput <- as.ts(na_interpolation(data_nibio[,col]))
    plot(imput,xlim = c(indexes[1]-100,indexes[2]+100))
```

```
    abline(v=indexes[1],col = "red")
    abline(v=indexes[2],col = "red")
    title(paste(col,"naive"))
}


## ------------------------------------------------------------------------

feature.name = c("TM","RR","TJM10","TJM20")
na.run.tables <- c()
full.count <- c()

notible_run <- 24*7
warning_run <- 8*2 # imputering fra begge ender

cat("Null count of data.",
            file = "data.txt",sep="\n")
cat(paste("notable runs,defined by nb length",notible_run,"and warning length",warnin
            file = "NB_data.txt",sep="\n")

station_names <- read.csv(DATA_INFO_NIBIO_FILE,
                          header=TRUE,
                          row.names="ID",
                          colClasses=c(ID="integer",Navn="character"))

na.run.station.year.feature <- list()

sub_set <- unlist(nibio_id)

all.id <- as.numeric(rownames(station_names))

for(id in all.id){
    # beginning plot
    pdf(file = paste0(ROOT,"plots/plot-",id,".pdf"))
    plot(NULL,
         sub = "hourly time From 2014-03-01 to year 2020-10-31",
         xlab="Date", ylab="NA location",
         xlim = c(0,5881), ylim = c(2013,2021))

    colours <- c(TM ="blue", RR = "red",TJM10 = "green",TJM20 = "orange")
    lev <- seq(-1/2,1/2,length.out=5)
    names(lev) <- feature.name

    numb <- 0
    denom <- 0
    na.run.count <- matrix(rep(0,length=5880*4),nrow = 5880, ncol = 4)
    colnames(na.run.count) <- feature.name
    na.count <- c()
    na.count.year <- c()
```

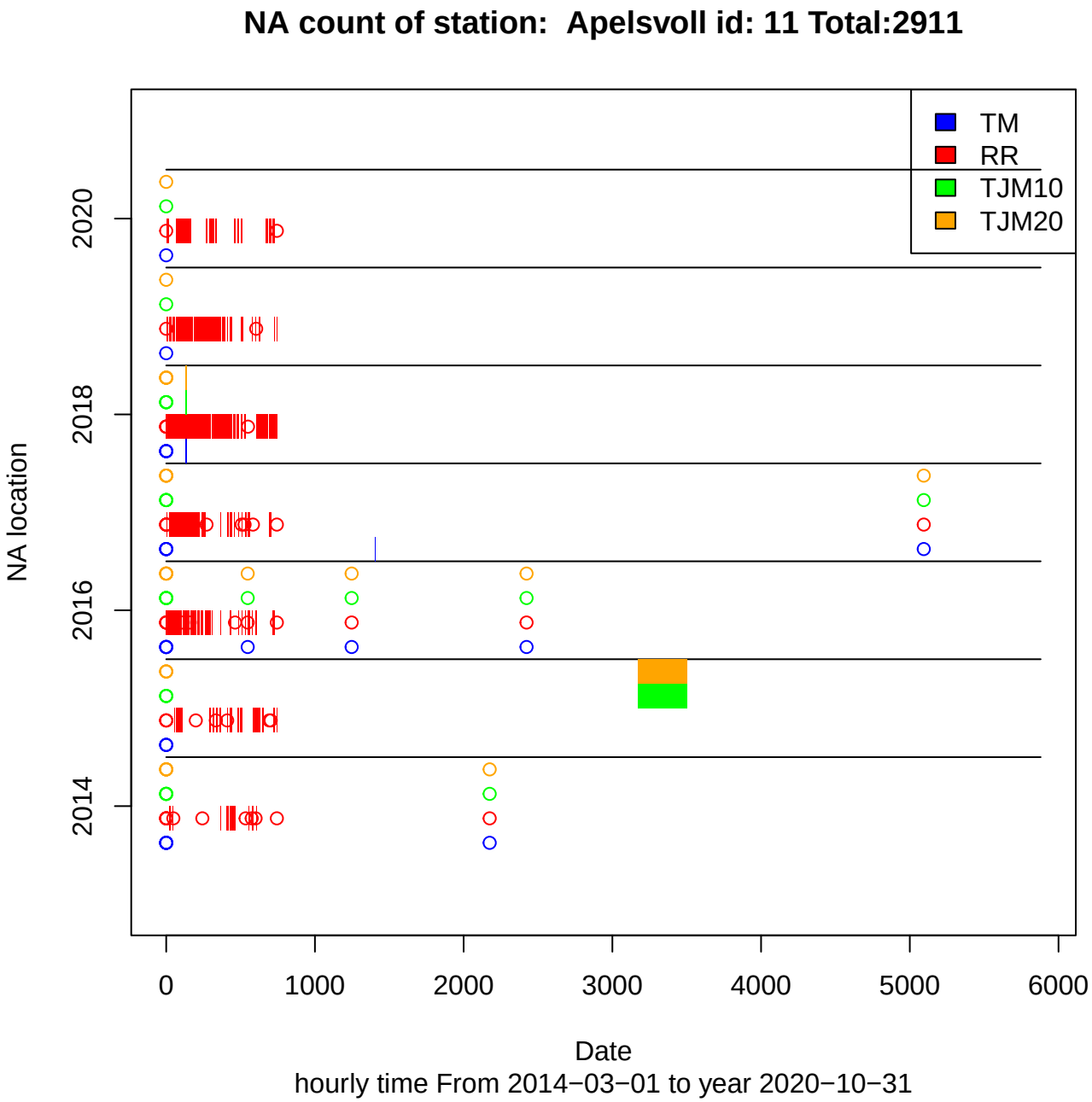
```
na.matrix.total <- NULL
#na.run.station.year.feature[[as.character(id)]] <- c()
#data_plot <- ggplot(title = paste("NA count of staion:",station_names[as.charact
na.plot <- FALSE
cat(paste("***************","station",id,"***************"),append=T,sep="\n",fil
for(year in seq(2014,2020)){

    # Drawing seperating lines

    lines(c(0,5880),c(year + 1/2,year + 1/2), col = "black")

    #lev <- seq(-1/2,1/2,length.out=5)
    #names(lev) <- c("TM","RR","TJM10","TJM20")
    #lev
    #lev["TJM20"]
    #lev[match("TJM20",names(lev))+1]
    cat(paste("::::::::year",year,"::::::::"),append=T,sep="\n",file = "NB_data.txt
    data_nibio <- suppressWarnings(data.nibio(id,year)) # henter data
    data_nibio <- data_nibio[rownames(data_nibio) ,]#> paste0(year,"-04-01"),]
    data_nibio_raw <- suppressWarnings(data.nibio(id,
                                    year,
                                    path=paste0(DATA_COLLECTION_NIBIO,
                                                "weather_data_raw_hour_stID%i_y%i.cs
                                                )
                                    ))

    data_nibio_raw[!is.na(data_nibio_raw["TM"]) & (data_nibio_raw["TM"] <= 0),"

    data_nibio[1:nrow(data_nibio_raw),"RR"] <- data_nibio_raw[1:nrow(data_nibio_r

    #na.run.station.year.feature[[as.character(id)]][[as.character(year)]] <- c()

    # Na analesys

    cat("----------Matrix representation, and pair NA's----------",append =T,sep="\n

    data.matrix <- as.matrix(ifelse(is.na(data_nibio),1,0))

    data.matrix.sq <- t(data.matrix)%*%data.matrix
    if(is.null(na.matrix.total)){
        na.matrix.total <- data.matrix.sq
    } else {
        na.matrix.total <- na.matrix.total + data.matrix.sq
    }

    cat("\t",append=T,file = "NB_data.txt",sep = "\t")
    suppressWarnings(write.table(data.matrix.sq,append =T,file = "NB_data.txt",se

    cat(paste("Total NA:",sum(diag(data.matrix.sq))),file = "NB_data.txt",append=
```

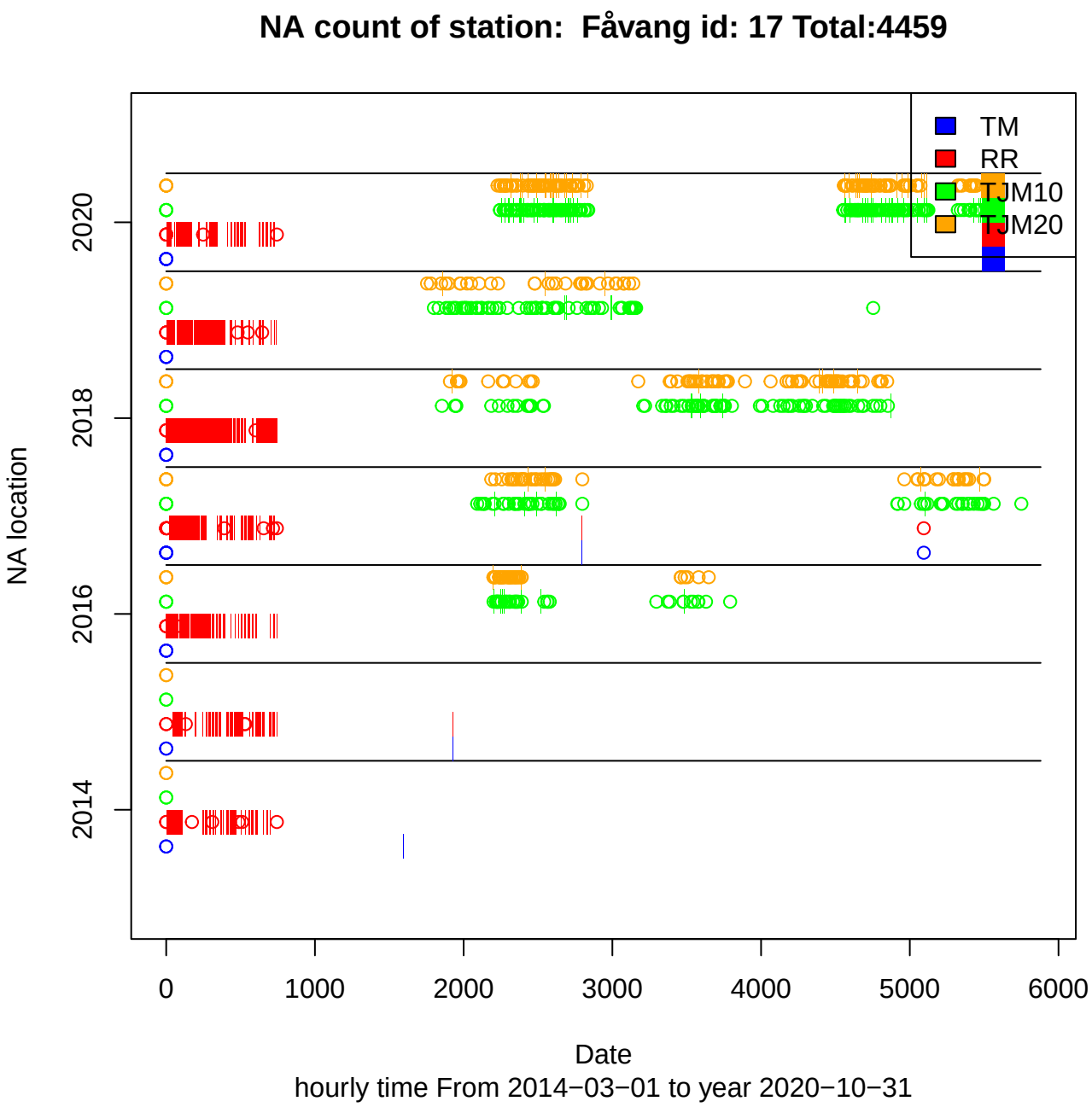
```
na.check <- is.na(data_nibio)
if(any(na.check)){
    if(length(na.count) == 0){
        na.count <- ifelse(na.check, 1, 0)
    } else {
        na.count <- na.count + ifelse(na.check, 1, 0)
    }
    #na.count.year[[as.character(year)]] <- sum(na.check)/(nrow(data_nibio)*4
    na.plot <- TRUE

    for(cols in feature.name){ # checker run for hver kolonne
        run_table <- table(NULL)
        cat(paste("\n----------station",id,"year",year,"feature",cols,"--
                  file = "NB_data.txt",append=T,sep="\n")
        if(sum(na.check[,cols]) > 0){
            run_na <- find.na.index.length(na.check[,cols])
            #na.run.station.year.feature[[as.character(id)]][[as.character(ye
            #print(paste("year:",year,"feature:",cols))
            #print(run_na)

            points(c(0,0,0,0),lev[1:4] + year + 1/8, col = colours)

            for(ind in 1:nrow(run_na)){
                c <- run_na[ind,"Length"]
                dates <- rownames(data_nibio)[c(run_na$First[ind],run_na$Last
                if(any(is.na(dates))){
                    print(dates)
                }
                cat(paste("\t-\t",dates[1],"|->",c,"run",ifelse(c != 1,paste(
                          file = "NB_data.txt",append=T,sep="")
                # plot conditions

                if(c == 1){
                    # plot dot
                    points(run_na$First[ind],year + lev[cols] + 1/8, col = co
                } else {
                    # plot rectangle
                    rect(run_na$First[ind],year + lev[cols],
                         run_na$Last[ind],year + lev[match(cols,names(lev))+1
                         col = colours[cols], border = NA
                        )
                }

                # Write condition

                 if(c >= notible_run){
                    cat("(NB!)",file = "NB_data.txt",append=T,sep="\n")
                } else if(c > warning_run) {
```

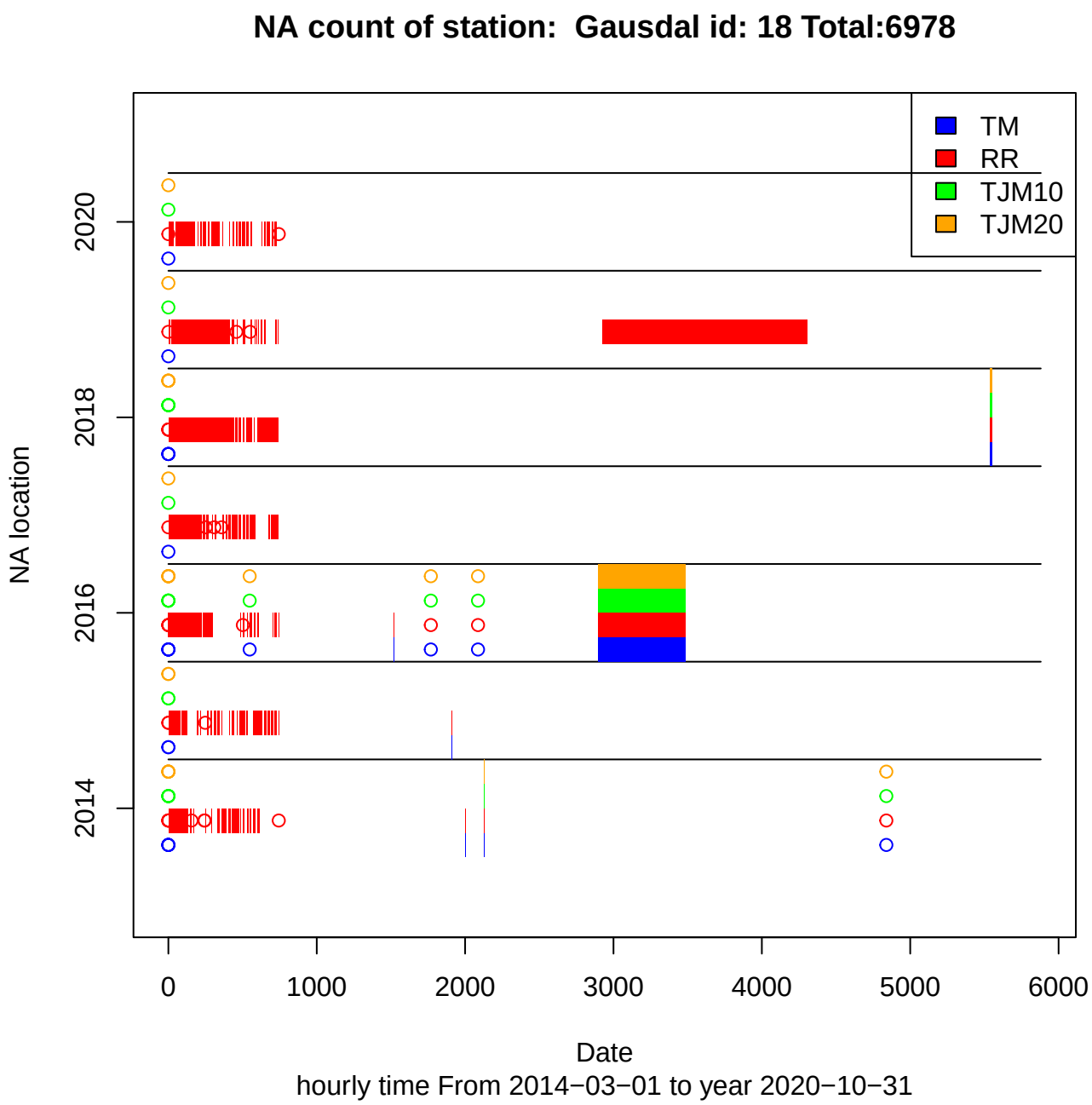
```
                        cat("(Warning)",
                            file = "NB_data.txt",append=T,sep="\n")
                    } else {
                        cat("",
                            file = "NB_data.txt",append=T,sep="\n")
                    }
                    na.run.count[c,cols] <- na.run.count[c,cols] + 1
                }
                run_table <- t(as.matrix(table(run_na$Length)))
            }

            cat(paste("\n----------Total for station",id,"year",year,"in feat
                      file = "NB_data.txt",append=T,sep="\n")
            cat("\t",append=T,file = "NB_data.txt",sep = "\t")
            suppressWarnings(write.table(run_table,file = "NB_data.txt",append=T,
            cat(paste("\t- total :\t",sum(na.check[,cols])),
                file = "NB_data.txt",append=T,sep="\n")
        }
    } else {
        cat(paste("\t- year",year,"without NA."),
                            file = "NB_data.txt",append=T,sep="\n")
        if(length(full.count[[as.character(id)]]) == 0){
            full.count[[as.character(id)]] <- 1/7
        } else {
            full.count[[as.character(id)]] <- full.count[[as.character(id)]] + 1/
        }
    }
    cat(paste("::::::: END year",year,"END :::::::"),append=T,sep="\n",file = "NB_d
}

legend(x = "topright",legend=feature.name, fill = colours)

if(na.plot){
    cat(paste("========== END station",id,"END ==========="),append=T,sep="\n
    cat(paste("Staion nr ",id),
        file = "data.txt",append=T,sep="\n")
    #suppressWarnings(write.table(bad_data,file = "data.txt",append=T)) # add lab
    cat(paste("prosent of",id,":",sum(na.count)/(nrow(data_nibio)*4)),
        file = "data.txt",append=T,sep="\n")
    cat(paste("prosent of",id,"for years:"),
        file = "data.txt",append=T,sep="\n")
    cat(paste0(unlist(na.count.year), collapse = "\n"),
        file = "data.txt",append=T,sep="\n")
    cat("\t",append=T,file = "NB_data.txt",sep = "\t")
    suppressWarnings(write.table(na.matrix.total,file = "NB_data.txt",append=T,se
    cat(paste("Total:",sum(diag(na.matrix.total))),file = "NB_data.txt", append =
}
title(main = paste0("NA count of station: ", station_names[as.character(id),],
                    " id: ",id,
```

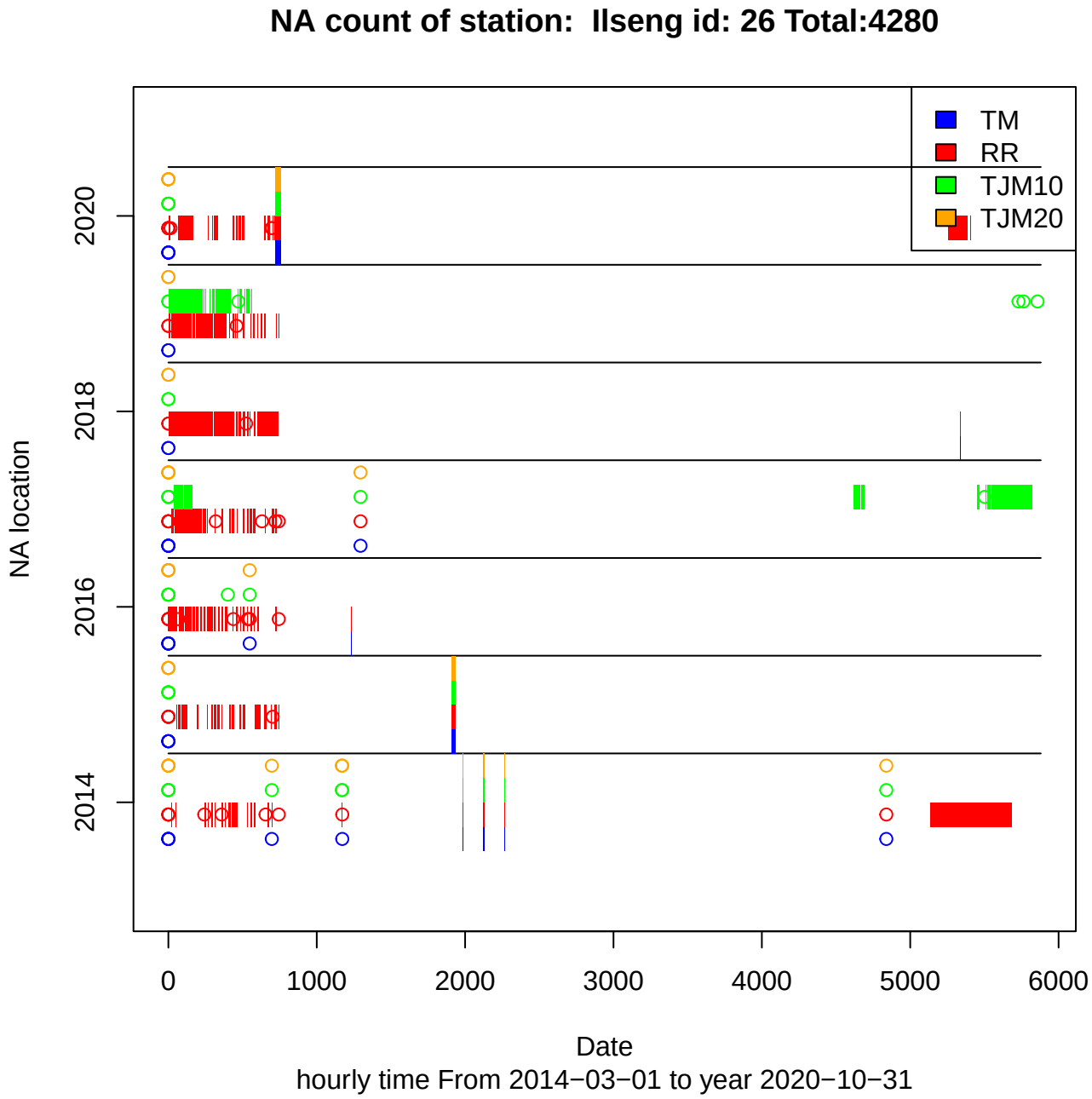
```
                    "␣Total:",sum(diag(na.matrix.total))))
    dev.off()
}


## ------------------------------------------------------------------------
plot(data.nibio(16,2017)[,"TM"],type="l")
plot(forecast(fit,h=24*7),xlim=c(5500,6000))


## ------------------------------------------------------------------------
imput_data <- na_interpolation(as.ts(data_nibio))


## ------------------------------------------------------------------------
# RR hadde ikke noe serlig, men hadde en rep ~= 31 (måned baser?)
# TM ~= 24?
# TJM10 ~= 24?
# TJM20 ~= 21?
for(col in c("TM","TJM10","TJM20")){
    acf(imput_data[,col])
    title(col)
    pacf(imput_data[,col])
    title(col)
}


## ------------------------------------------------------------------------
plot(stlplus::stlplus(imput_data[,"RR"],n.p = 31, s.window = 5,s.degree=2))


## ------------------------------------------------------------------------
data_stat_id = matrix()

for(id in nibio_id){
    csv_files <- list.files(path = DATA_COLLECTION_NIBIO,
                            pattern = regex(paste0(".*ID",id,"_y\\d{4}.csv")),
                                            full.names = TRUE)
    combined_data <- lapply(csv_files,
                            read.csv,
                            header=T,
                            col.names = c("Time","TM","RR","TJM10","TJM20")) %>% bind_row
    combined_data <- combined_data %>% column_to_rownames(., var = 'Time')
    combined_data <- mutate_at(combined_data,c("TM","RR","TJM10","TJM20"), as.numeric
}


## ------------------------------------------------------------------------
library( datasets )
```

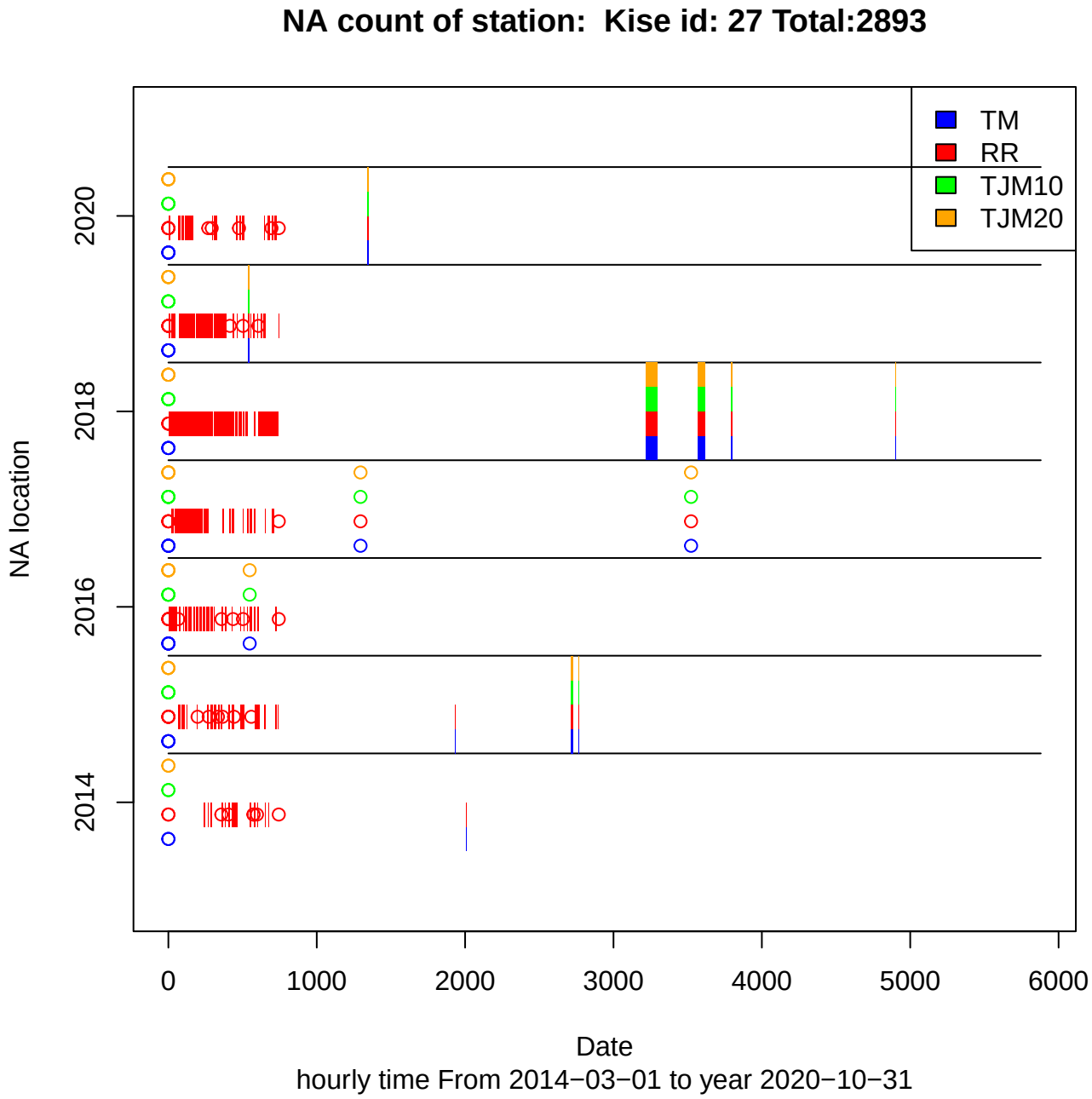
```
data("faithful")
# z - scores & Mahalanobis distance
z <- scale(imput_data) %>% as.data.frame()
mahalanobis(z , center = c(0 ,0) , cov = cov( imput_data,use = "all.obs" ) )
# DBSCAN & LOF
library( dbscan )
dbscan( imput_data , eps = 1)$cluster == 0
lof( imput_data , minPts = 5)
# Isolation forest
library( isotree )
iso_mod <- isolation.forest( imput_data )
predict( iso_mod , newdata = imput_data )
# one - class SVM
library( e1071 )
svm_mod <- svm ( imput_data , type = "one-classification")
print(sum(predict( svm_mod , newdata = imput_data )))


## ------------------------------------------------------------------------
adf.test(imputed.data[,"TJM10"])
kpss.test(imputed.data[,"TJM10"])
pp.test(imputed.data[,"TJM10"])
```

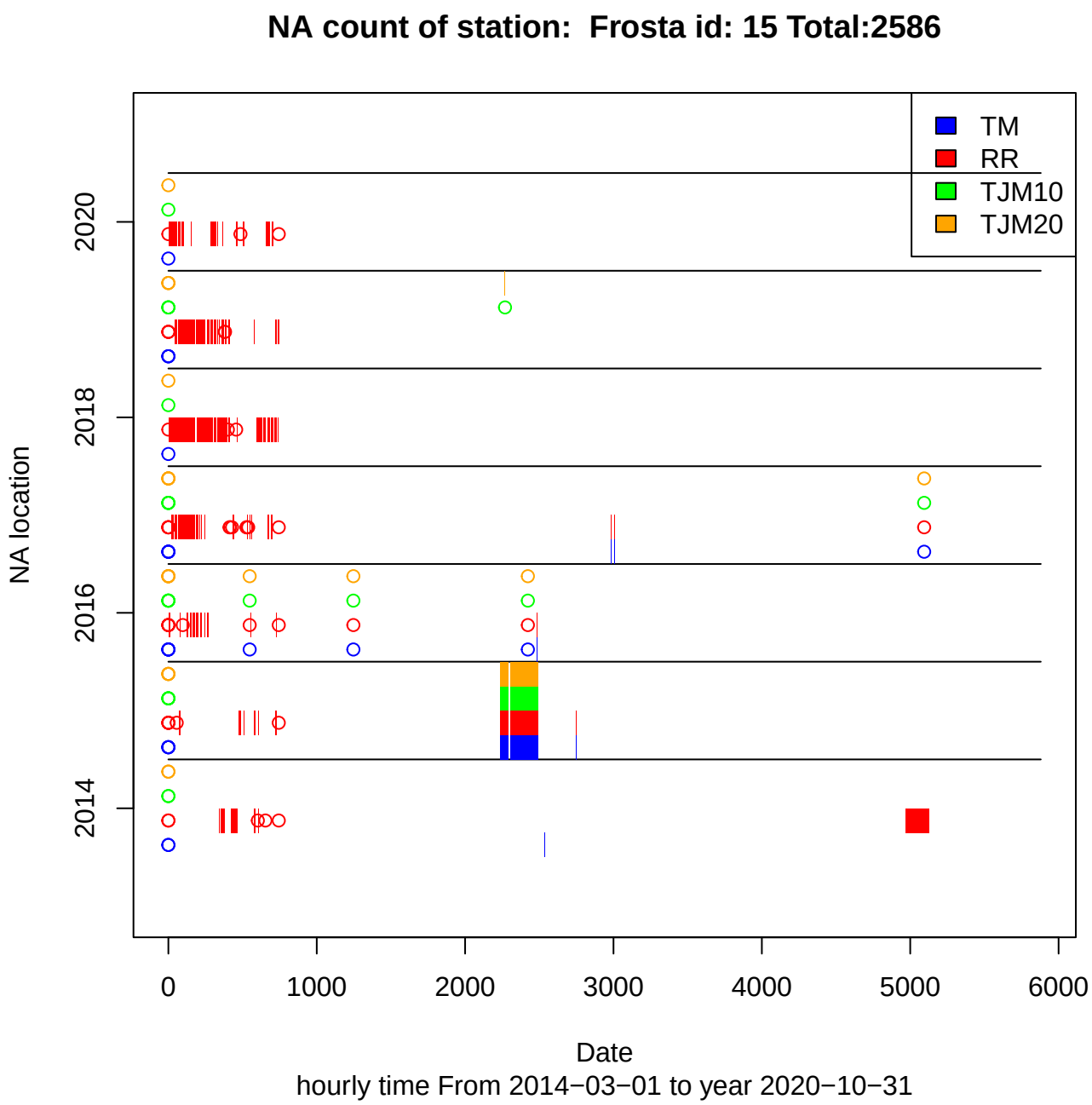

NA count of station: Apelsvoll id: 11 Total:2911
TM
RR
TJM10
TJM20
NA location
2014
2016
2018
2020
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

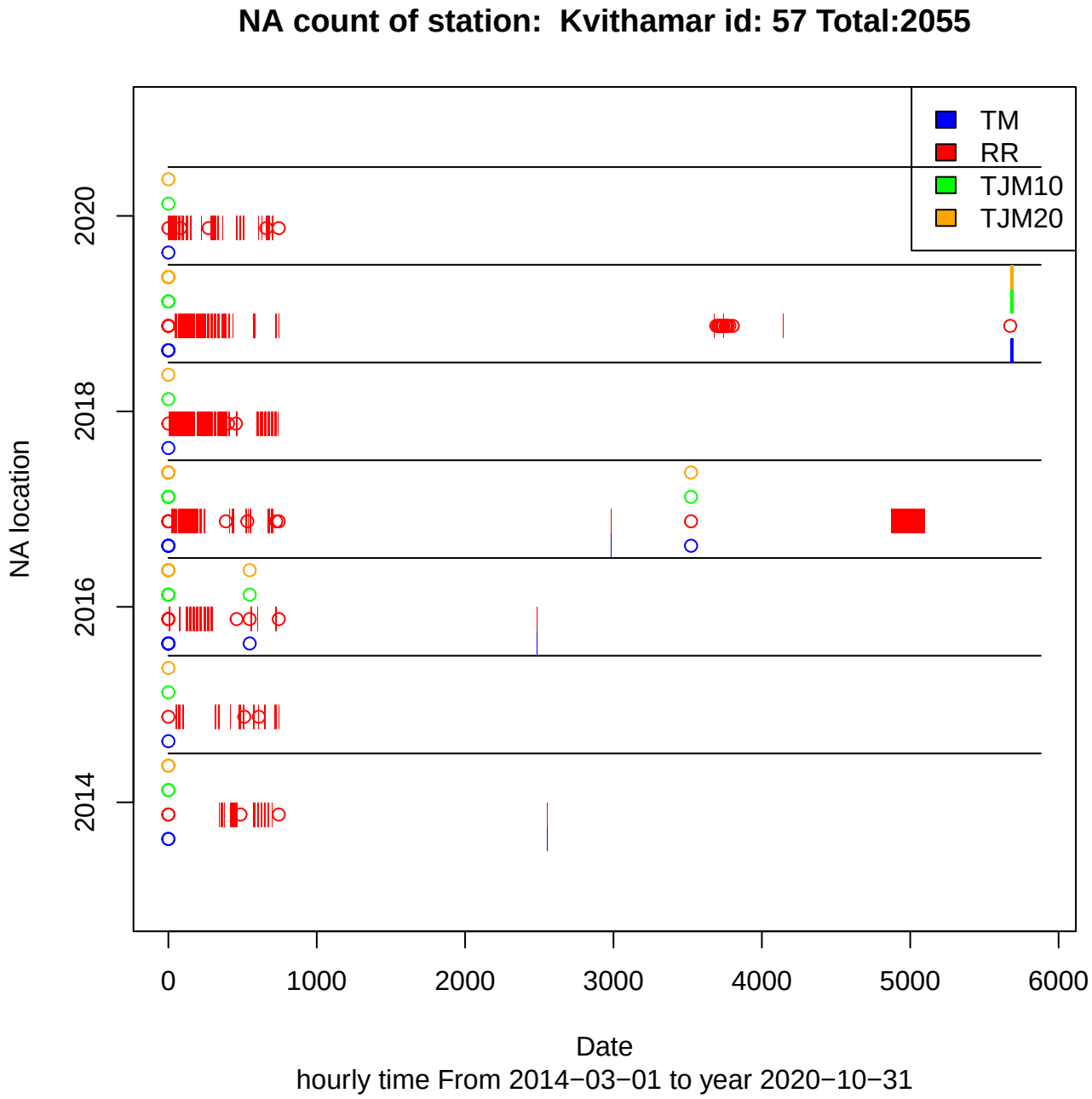

NA count of station: Fåvang id: 17 Total:4459
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

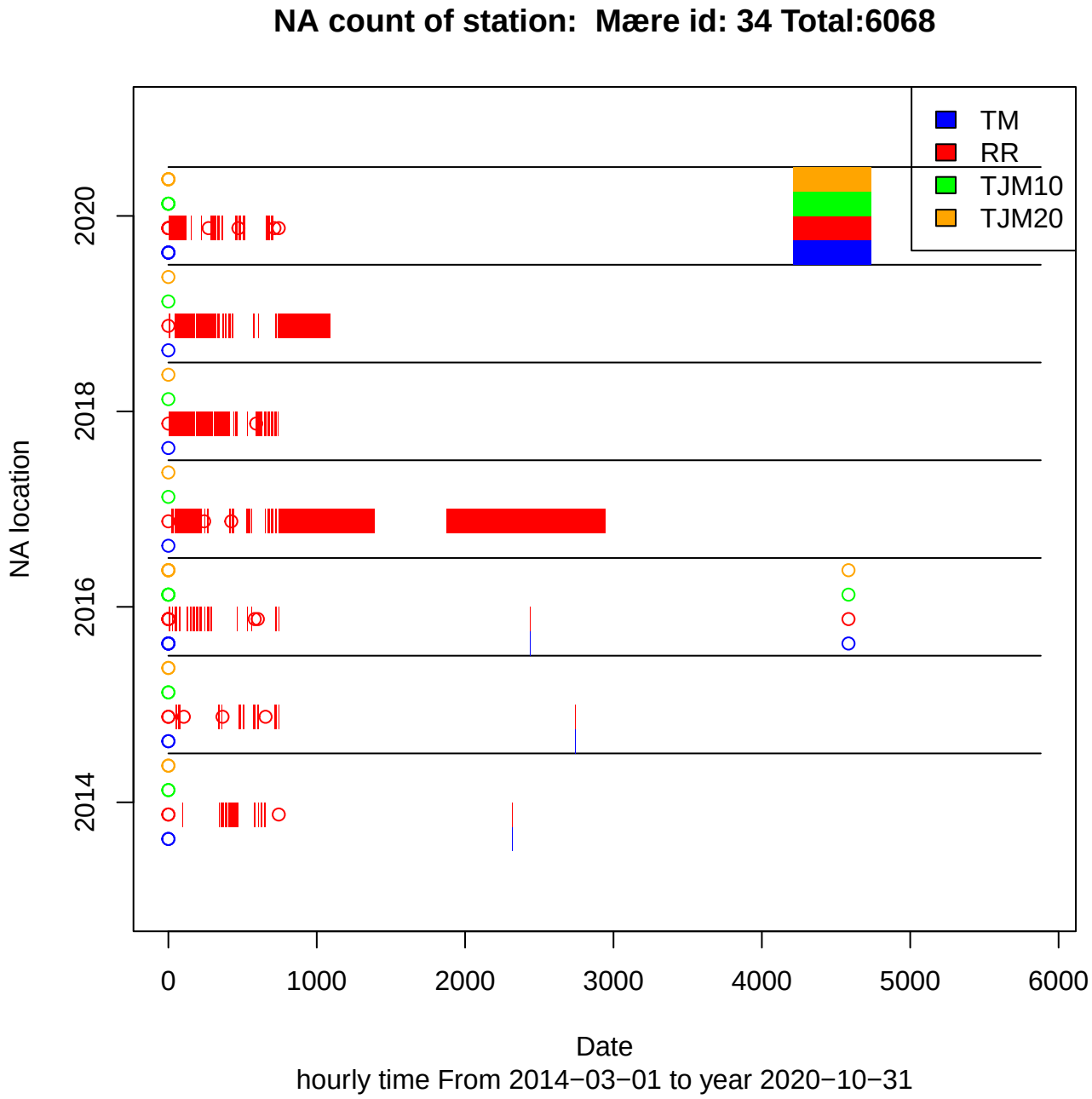

NA count of station: Gausdal id: 18 Total:6978
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

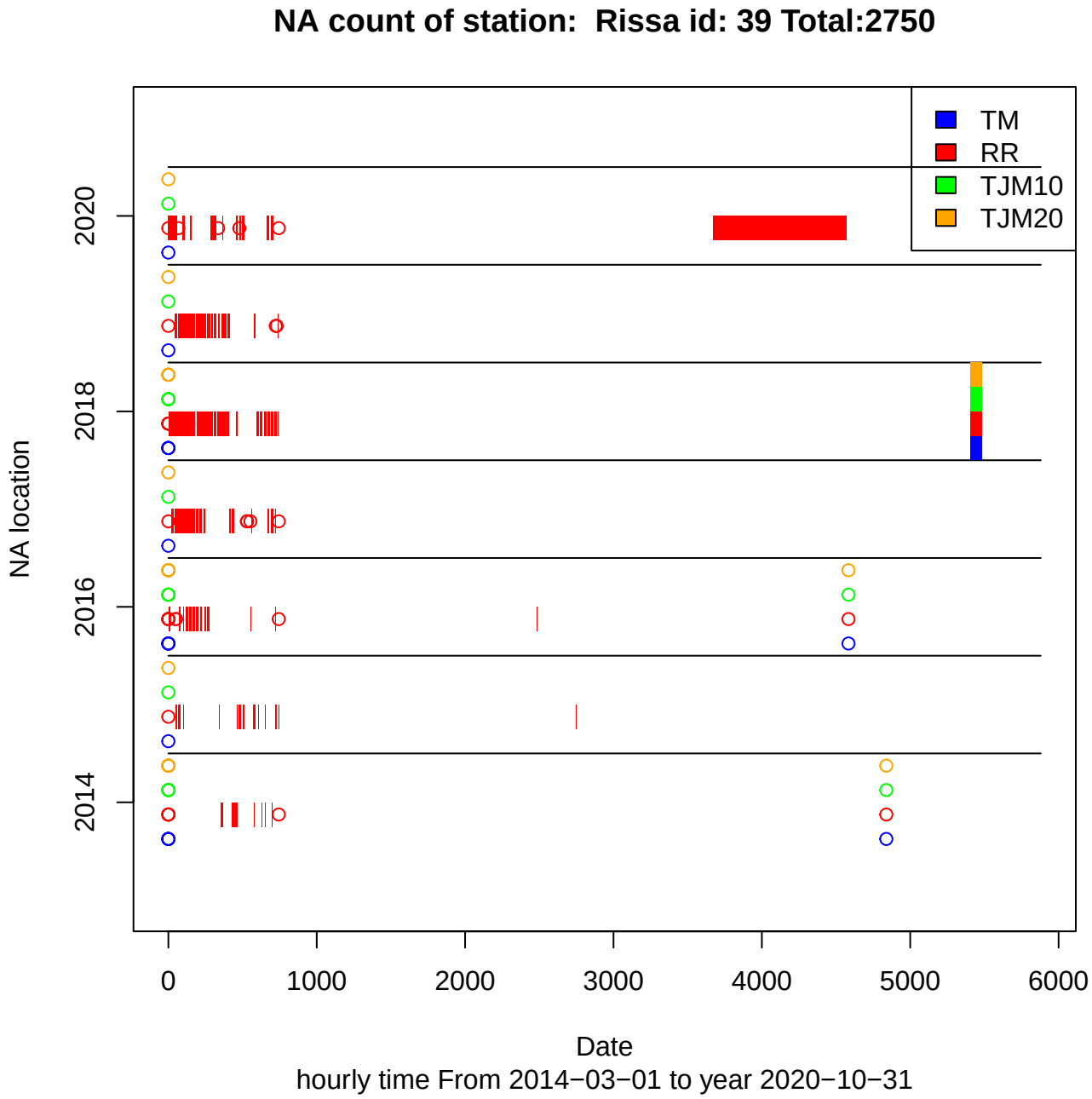

NA count of station: Ilseng id: 26 Total:4280
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

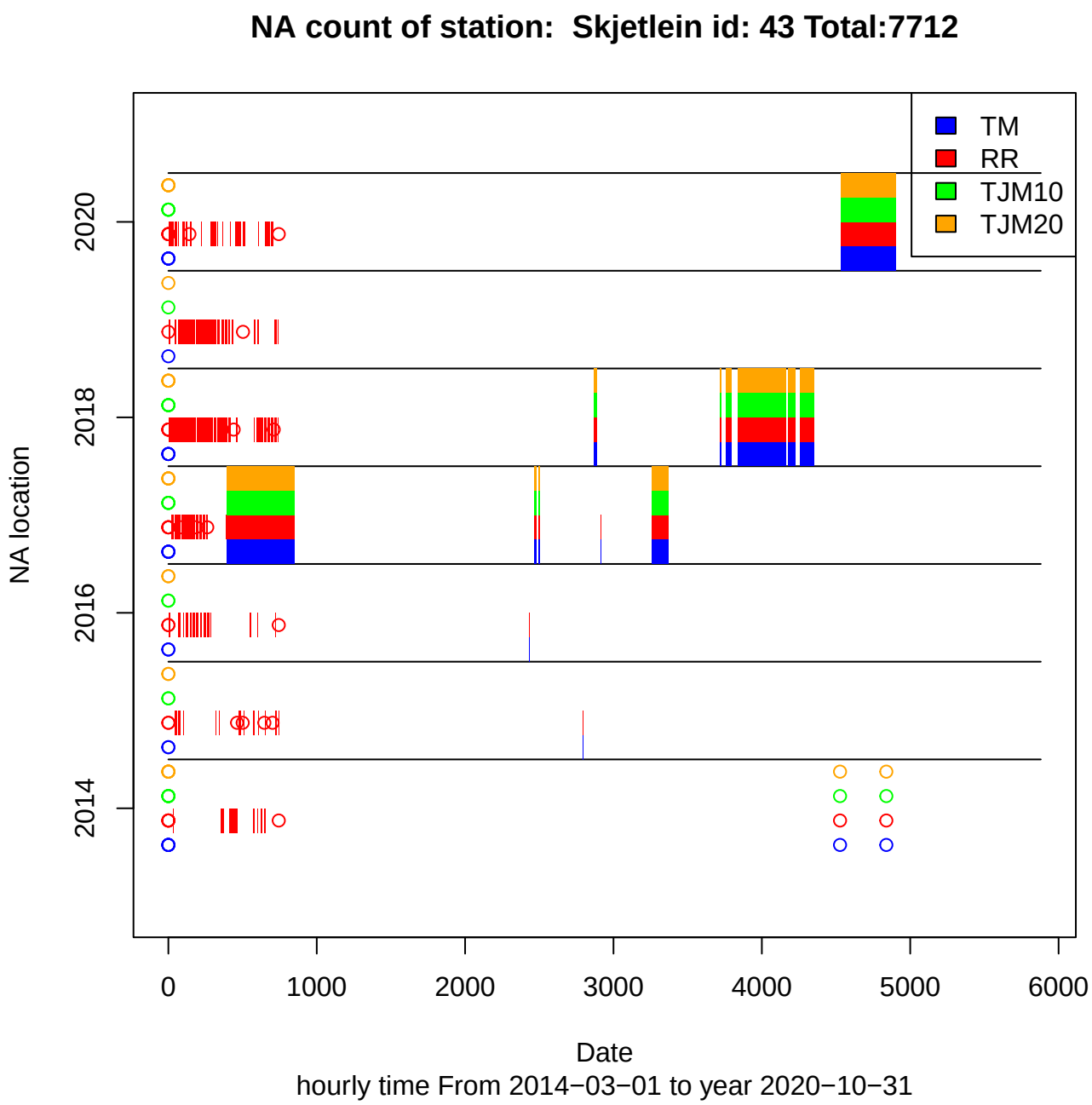

NA count of station:  Kise id: 27 Total:2893
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

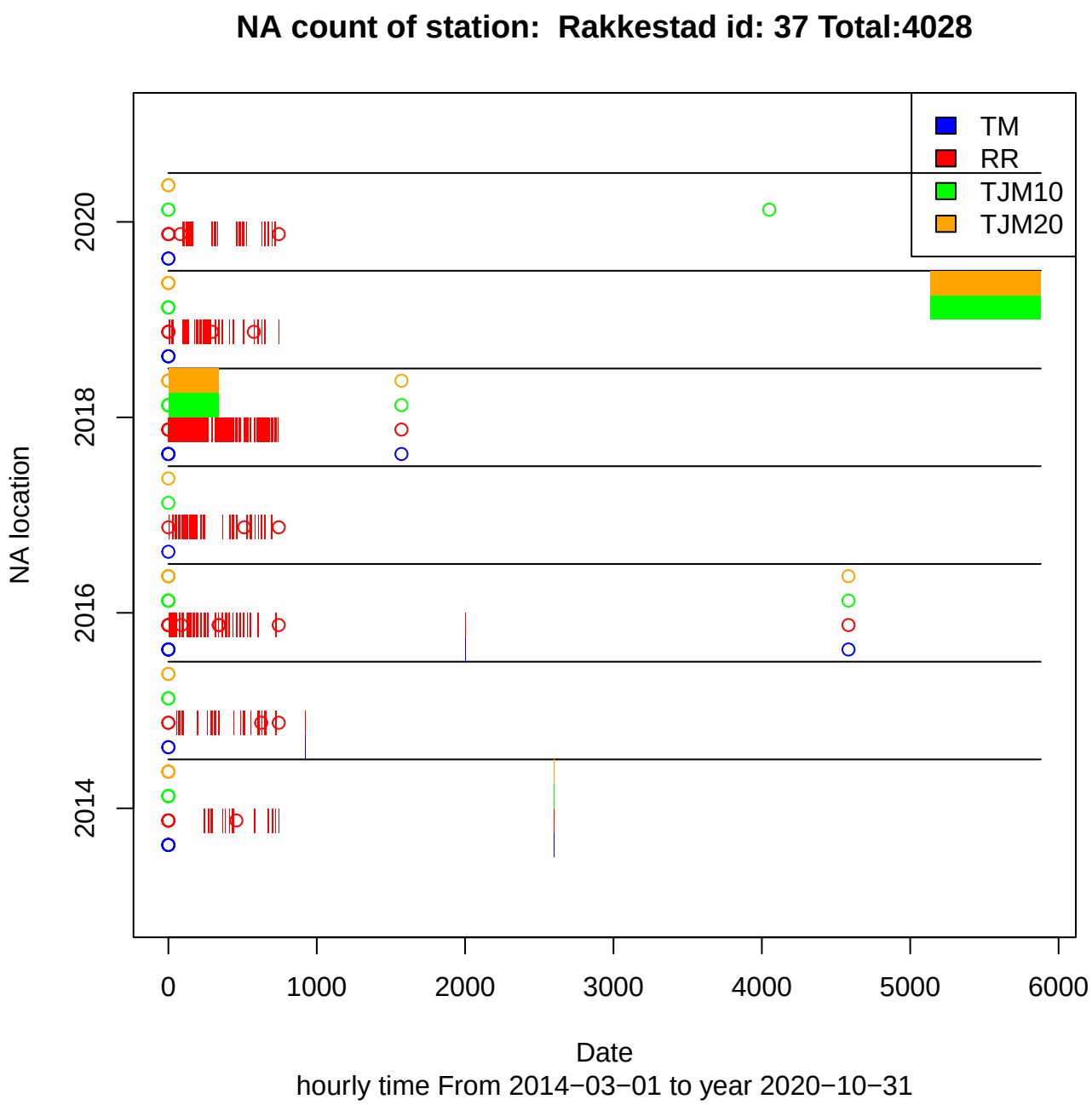

NA count of station: Frosta id: 15 Total:2586
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

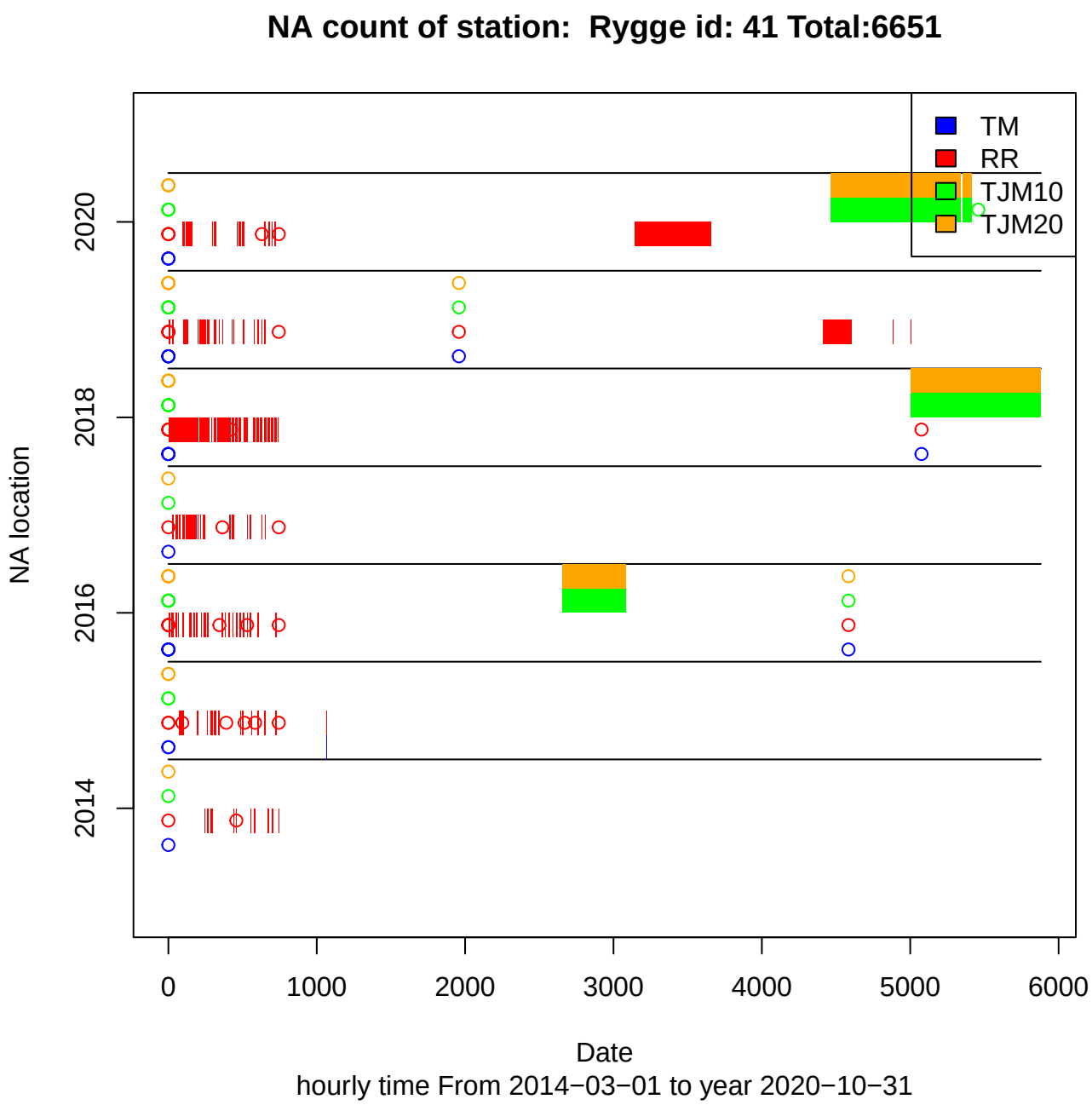

NA count of station: Kvithamar id: 57 Total:2055
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

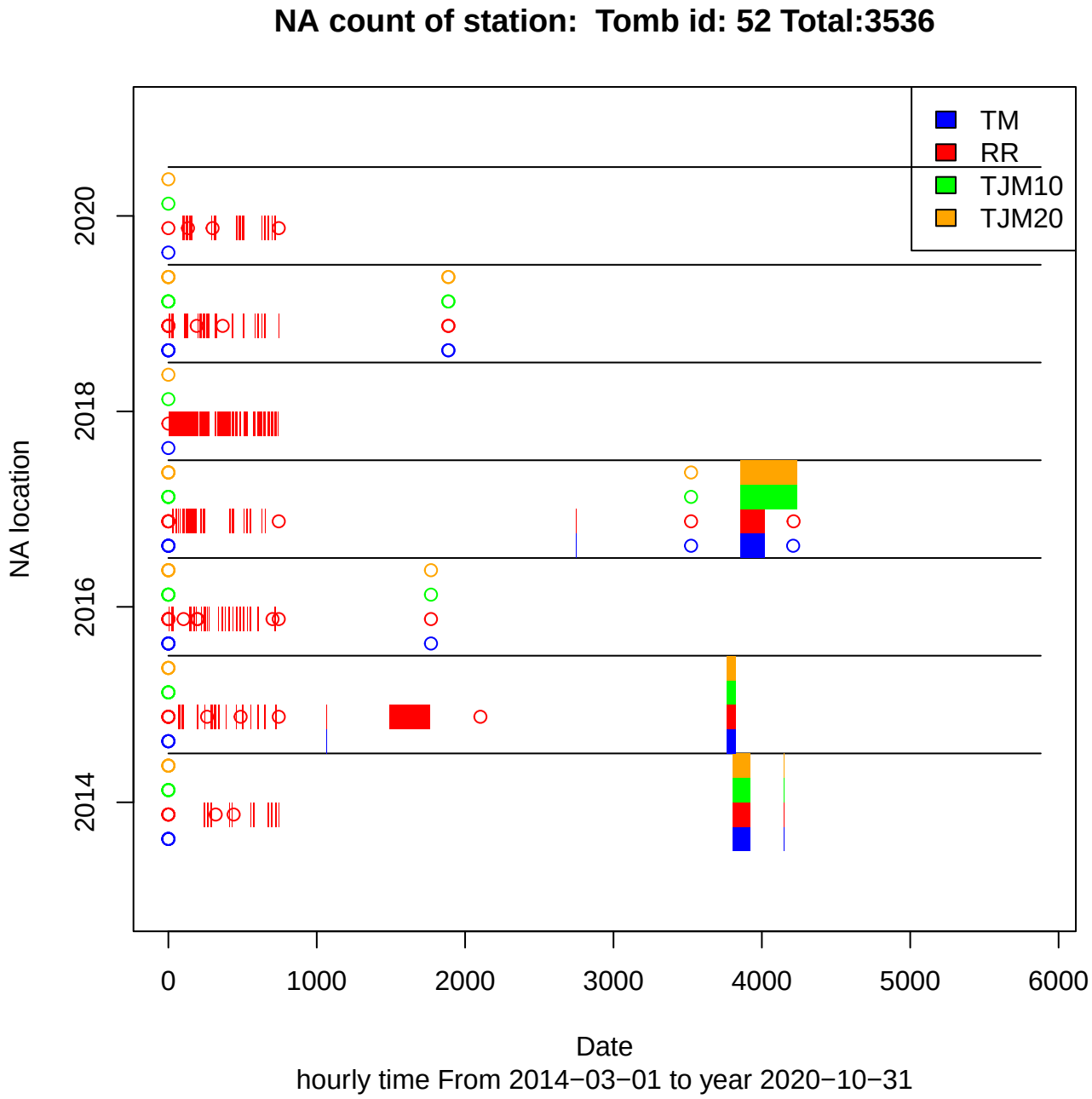

NA count of station: Mære id: 34 Total:6068
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

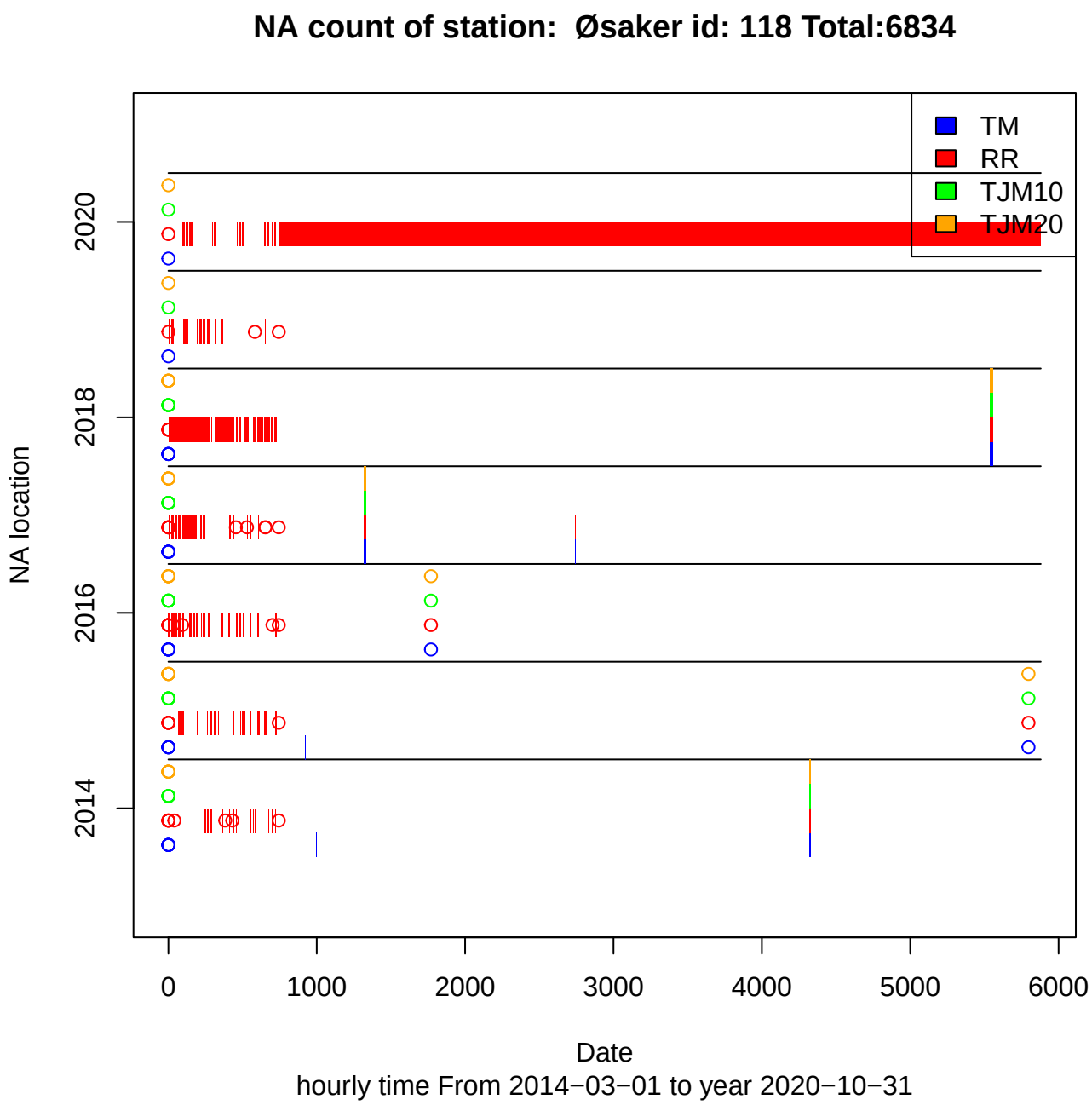

NA count of station: Rissa id: 39 Total:2750
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

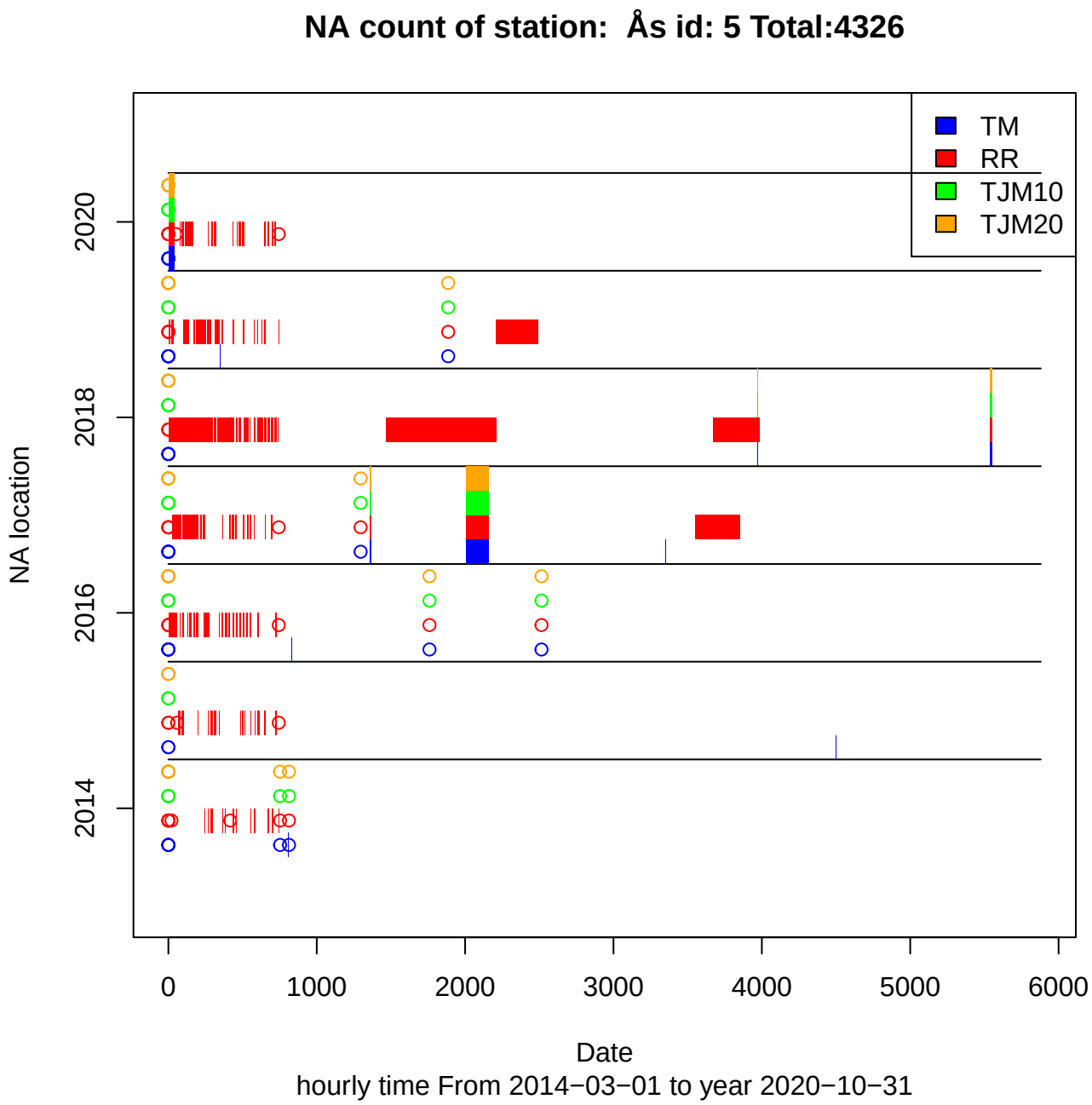

NA count of station: Skjetlein id: 43 Total:7712
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

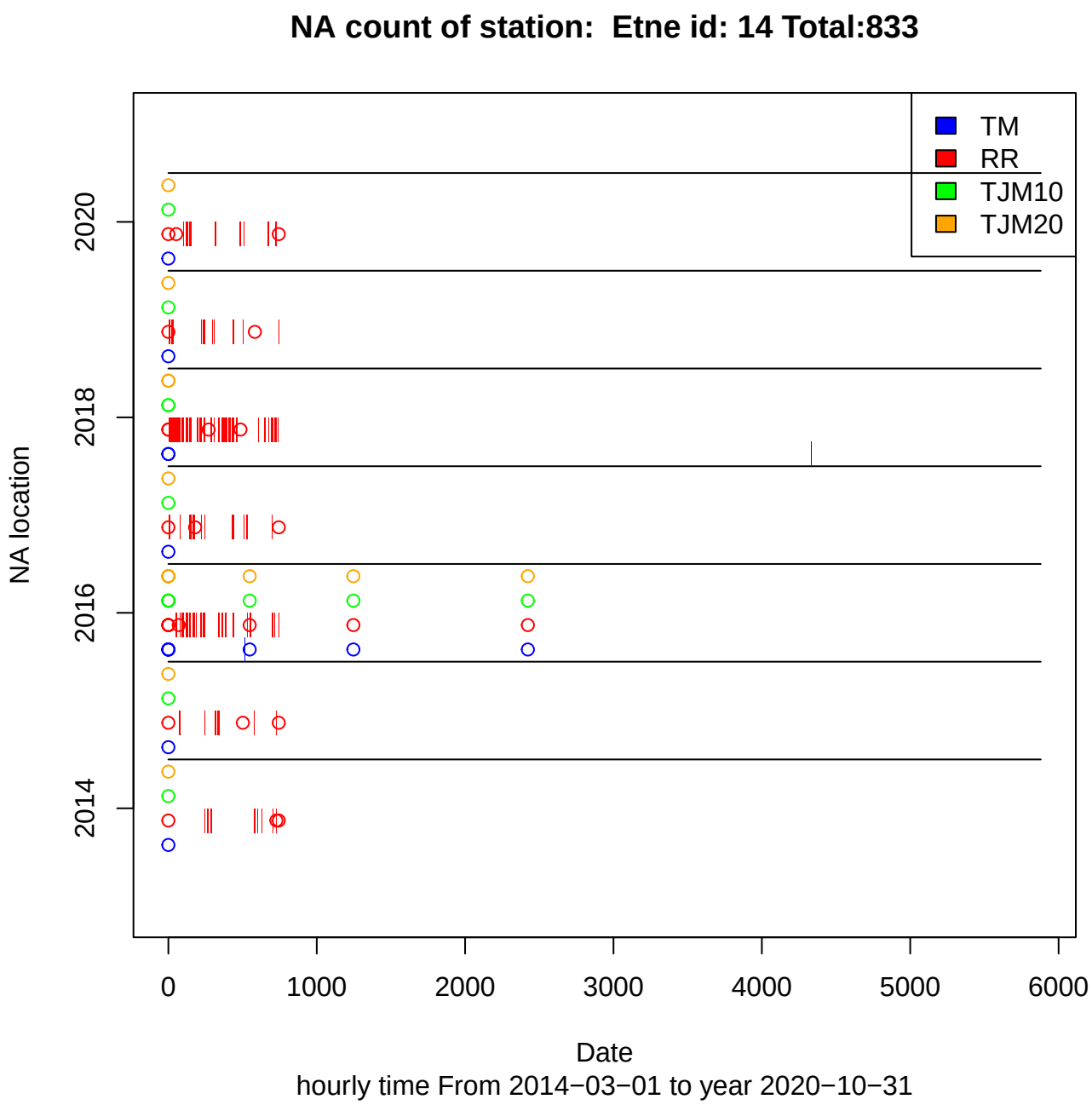

NA count of station: Rakkestad id: 37 Total:4028
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

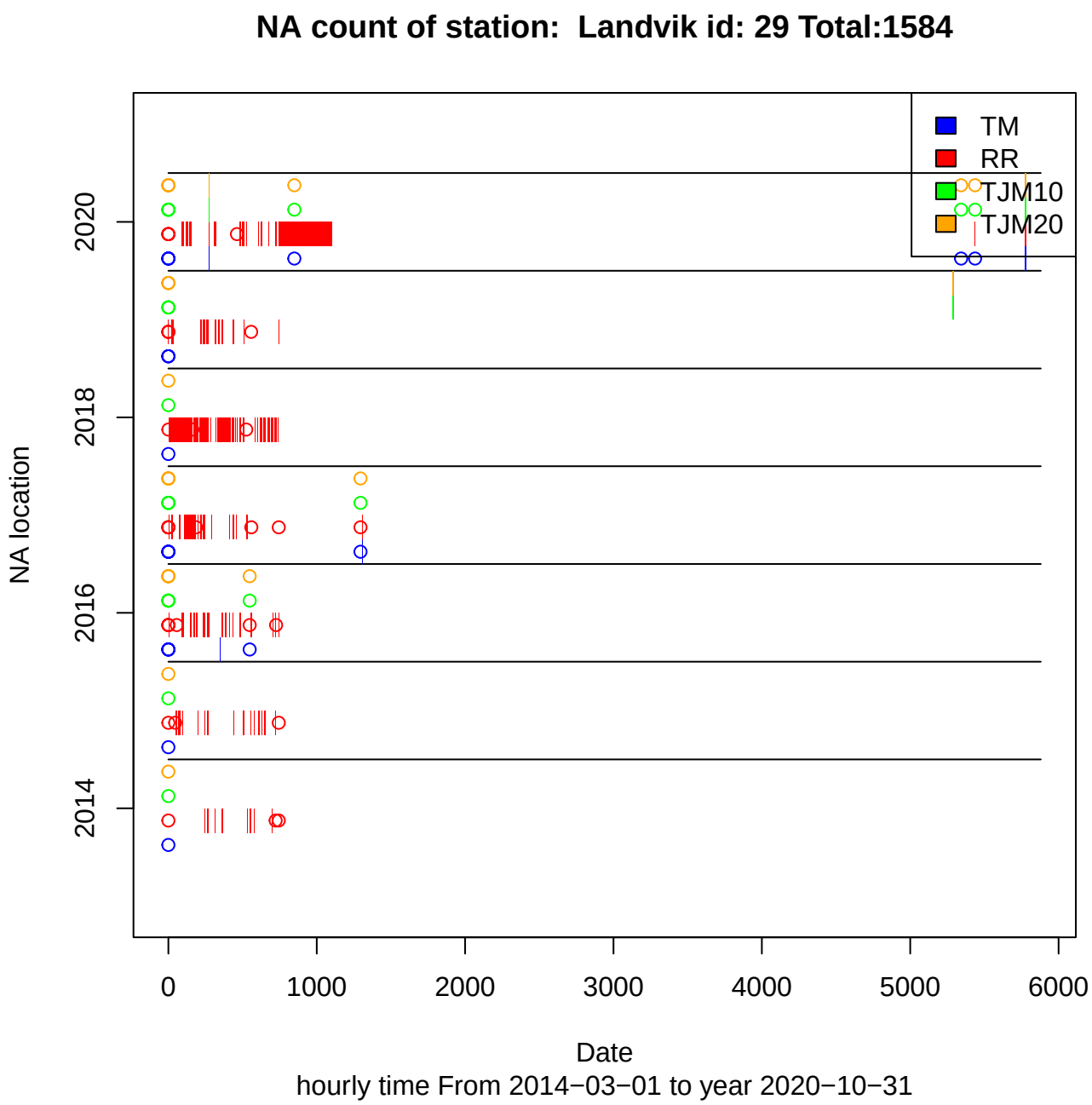

NA count of station: Rygge id: 41 Total:6651
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

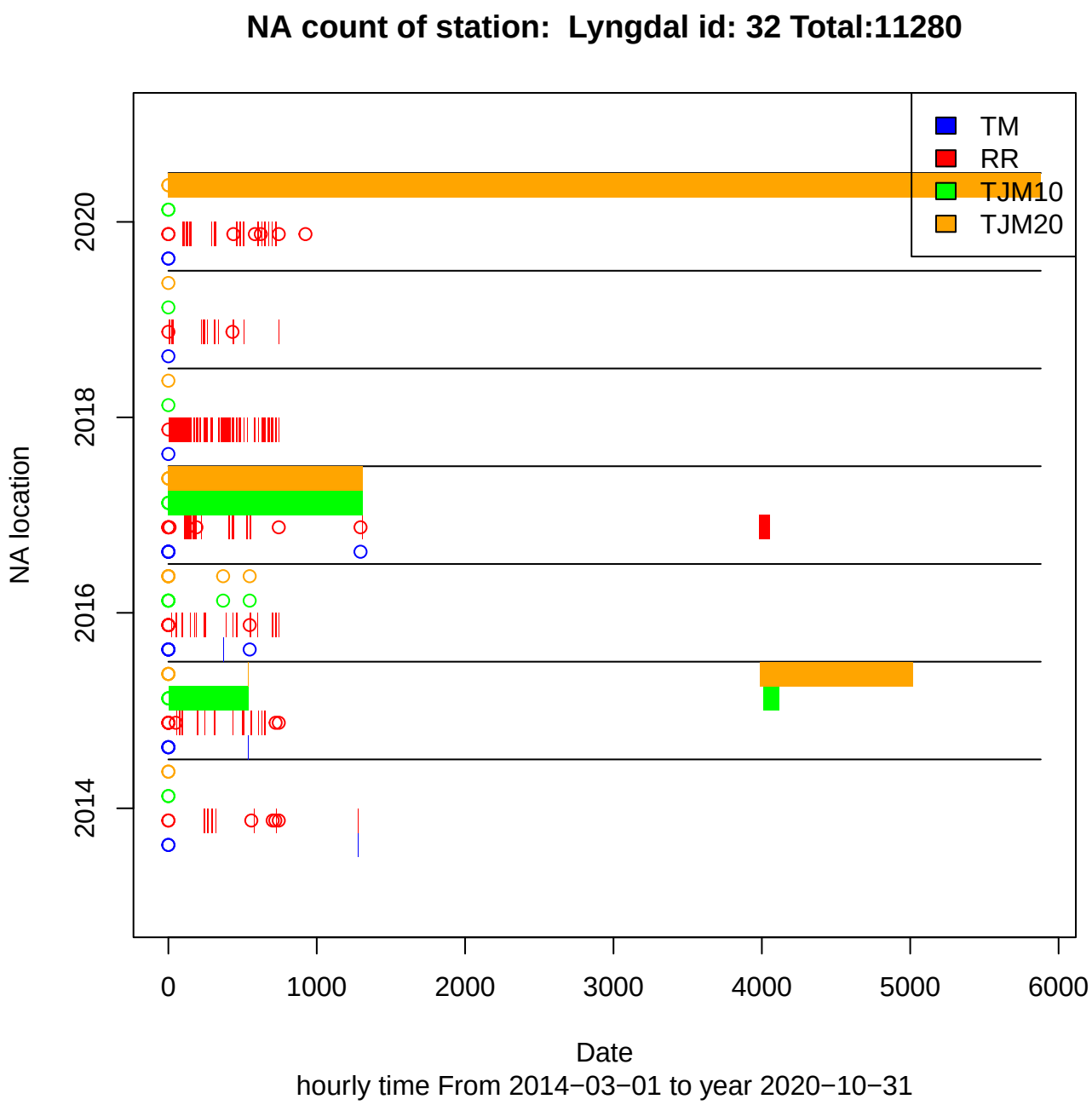

NA count of station: Tomb id: 52 Total:3536
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

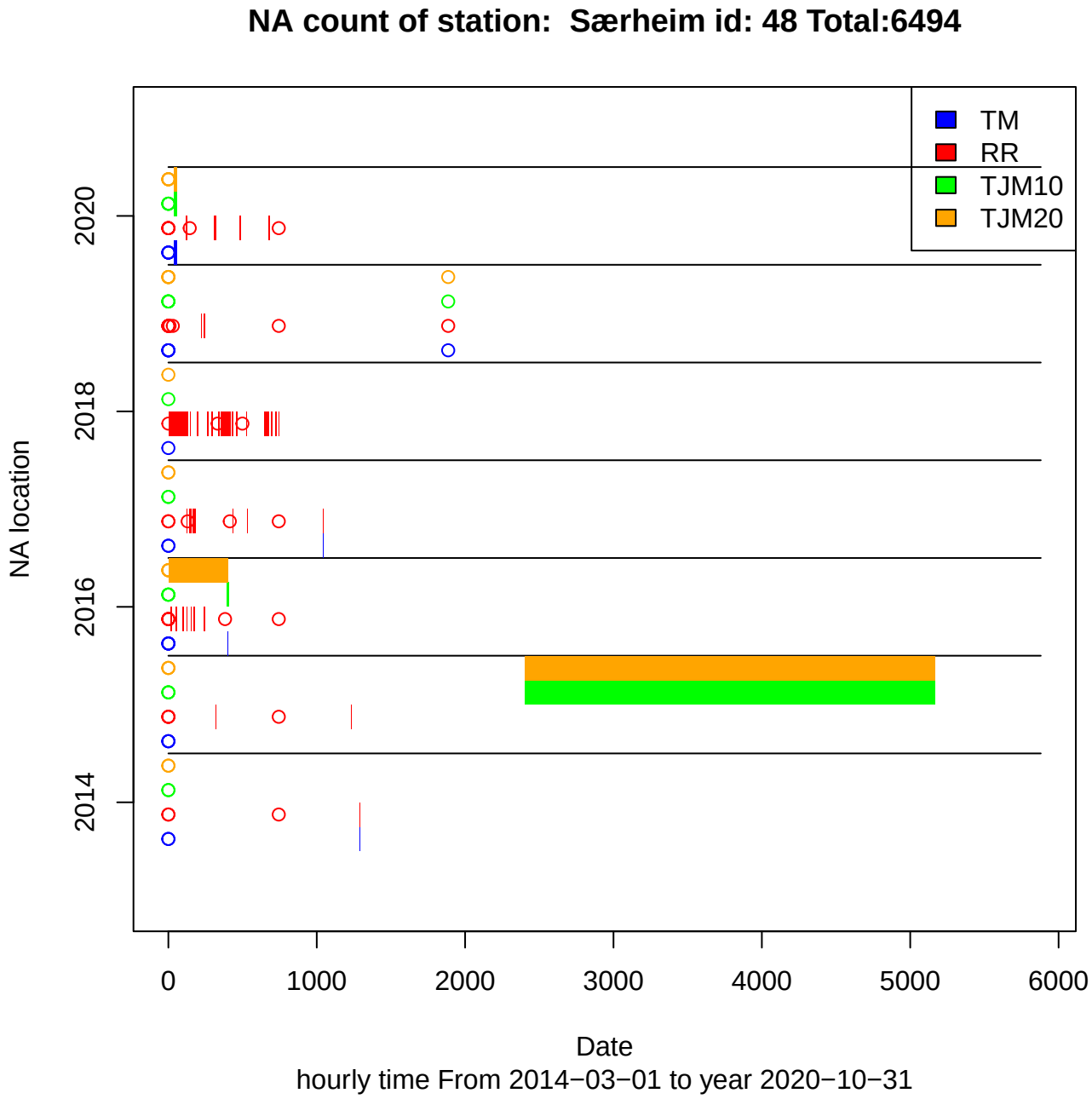

NA count of station: Øsaker id: 118 Total:6834
TM
RR
TJM10
TJM20
NA location
2014
2016
2018
2020
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

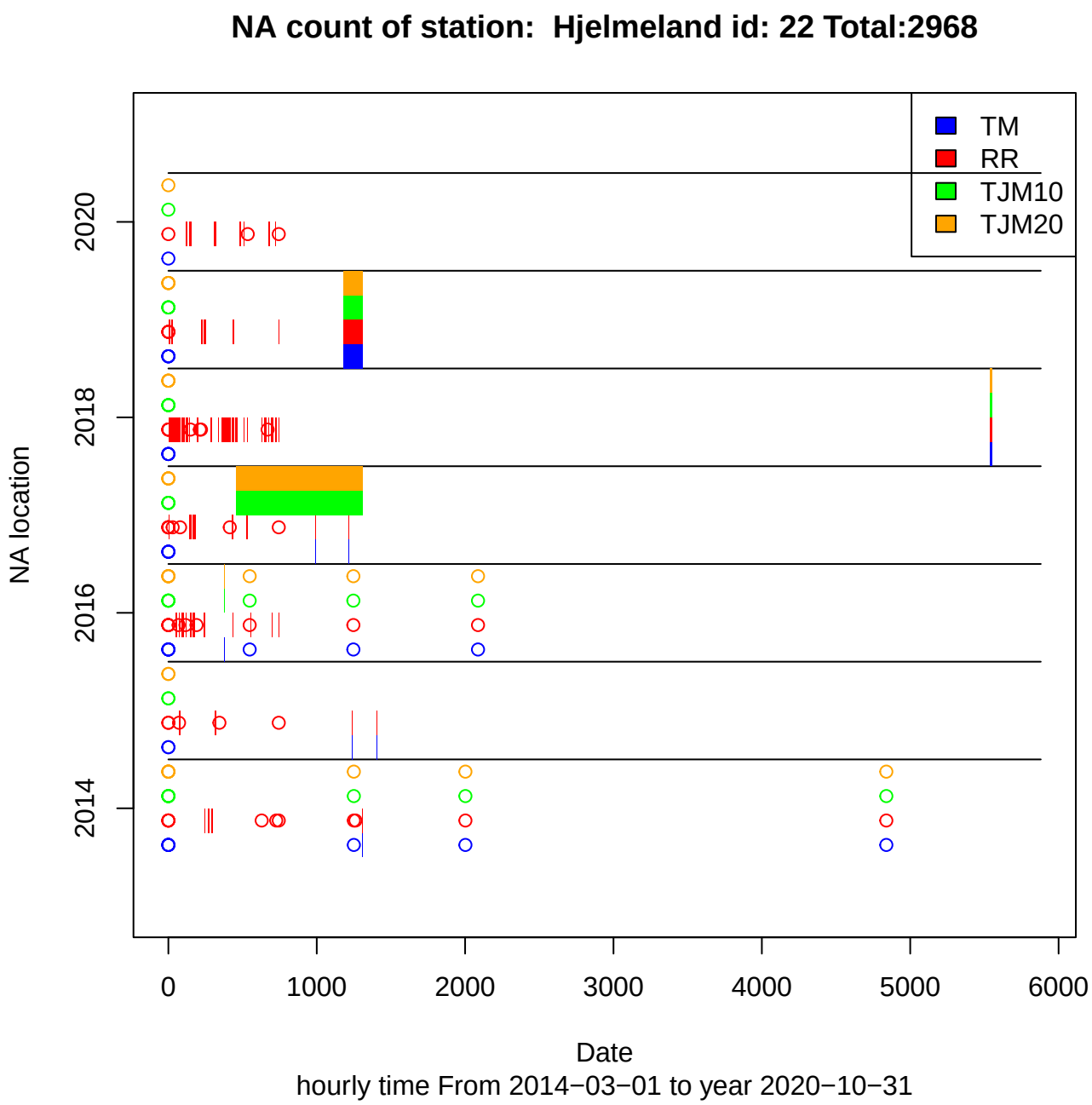

NA count of station: Ås id: 5 Total:4326
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

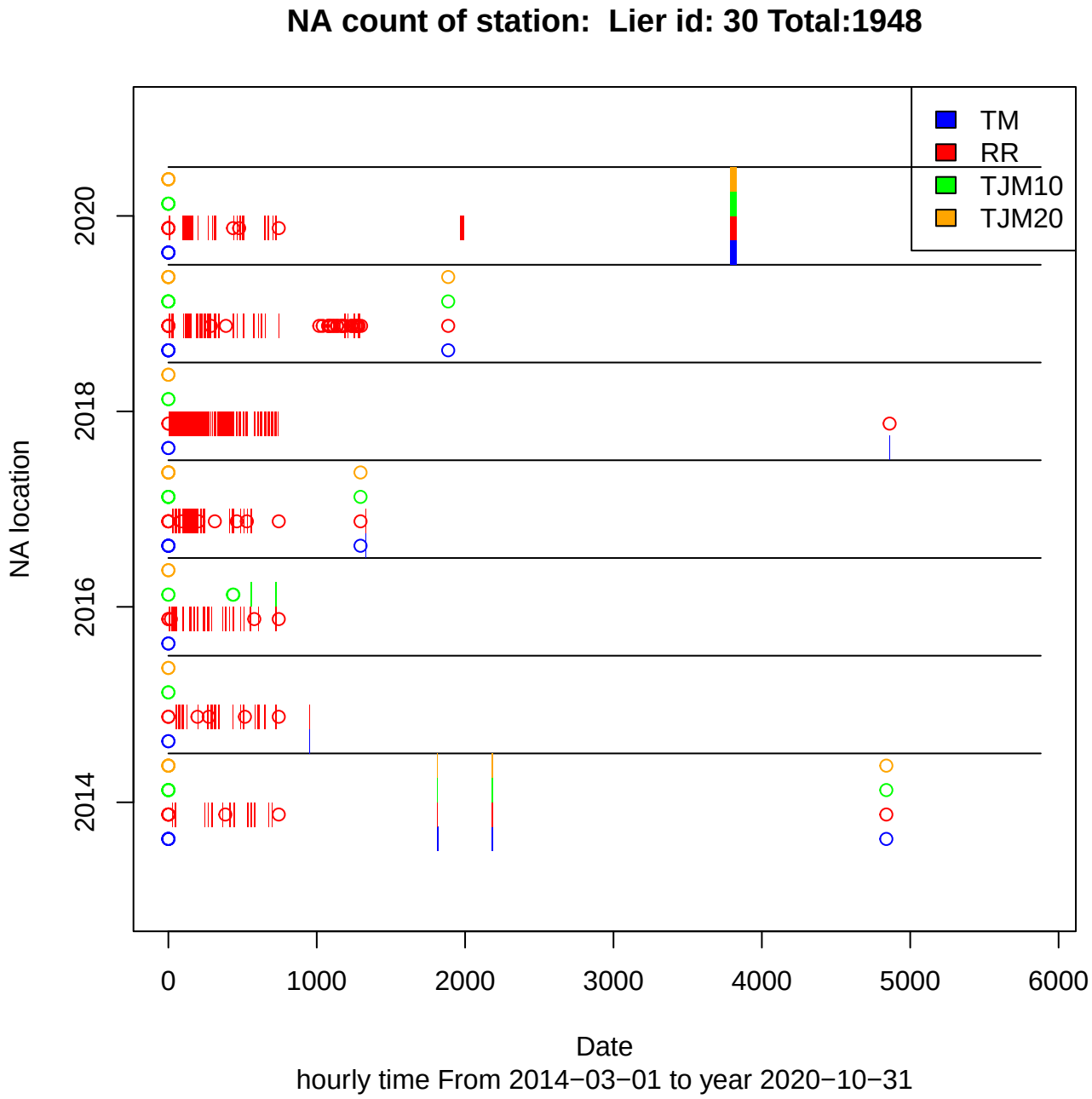

NA count of station:  Etne id: 14 Total:833
TM
RR
TJM10
TJM20
NA location
2014
2016
2018
2020
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

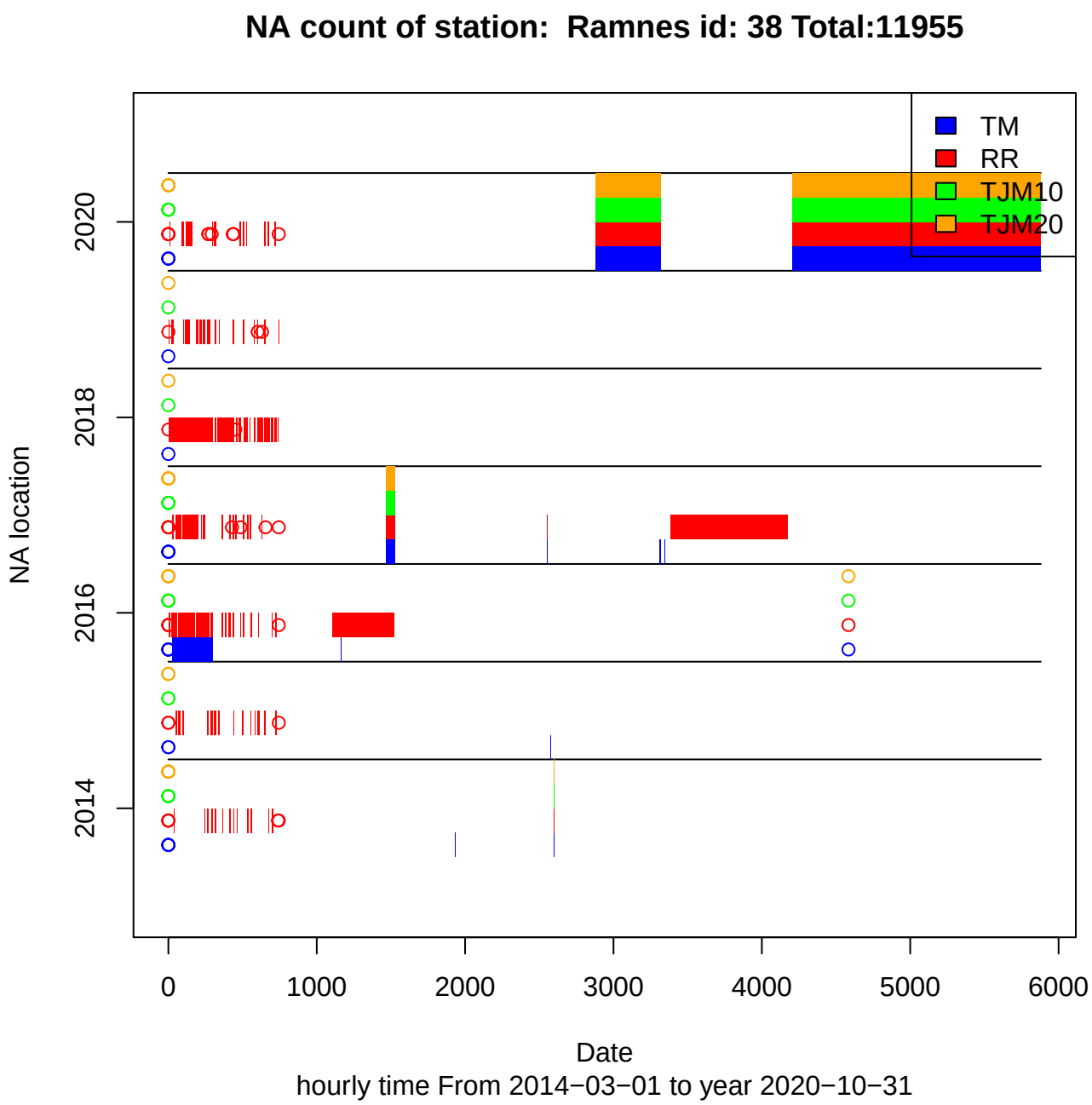

NA count of station: Landvik id: 29 Total:1584
TM
RR
TJM10
TJM20
NA location
2014
2016
2018
2020
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

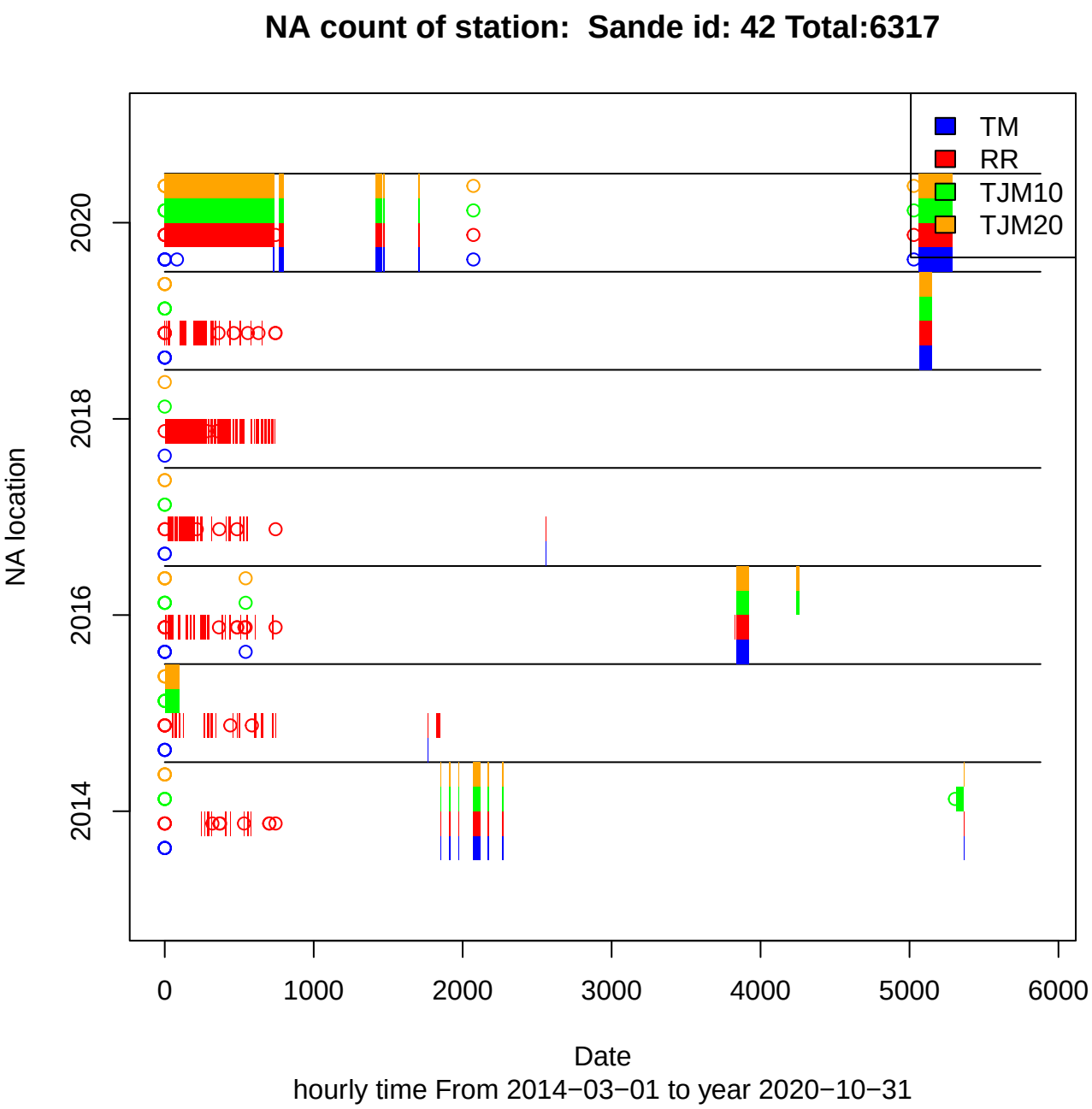

NA count of station: Lyngdal id: 32 Total:11280
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

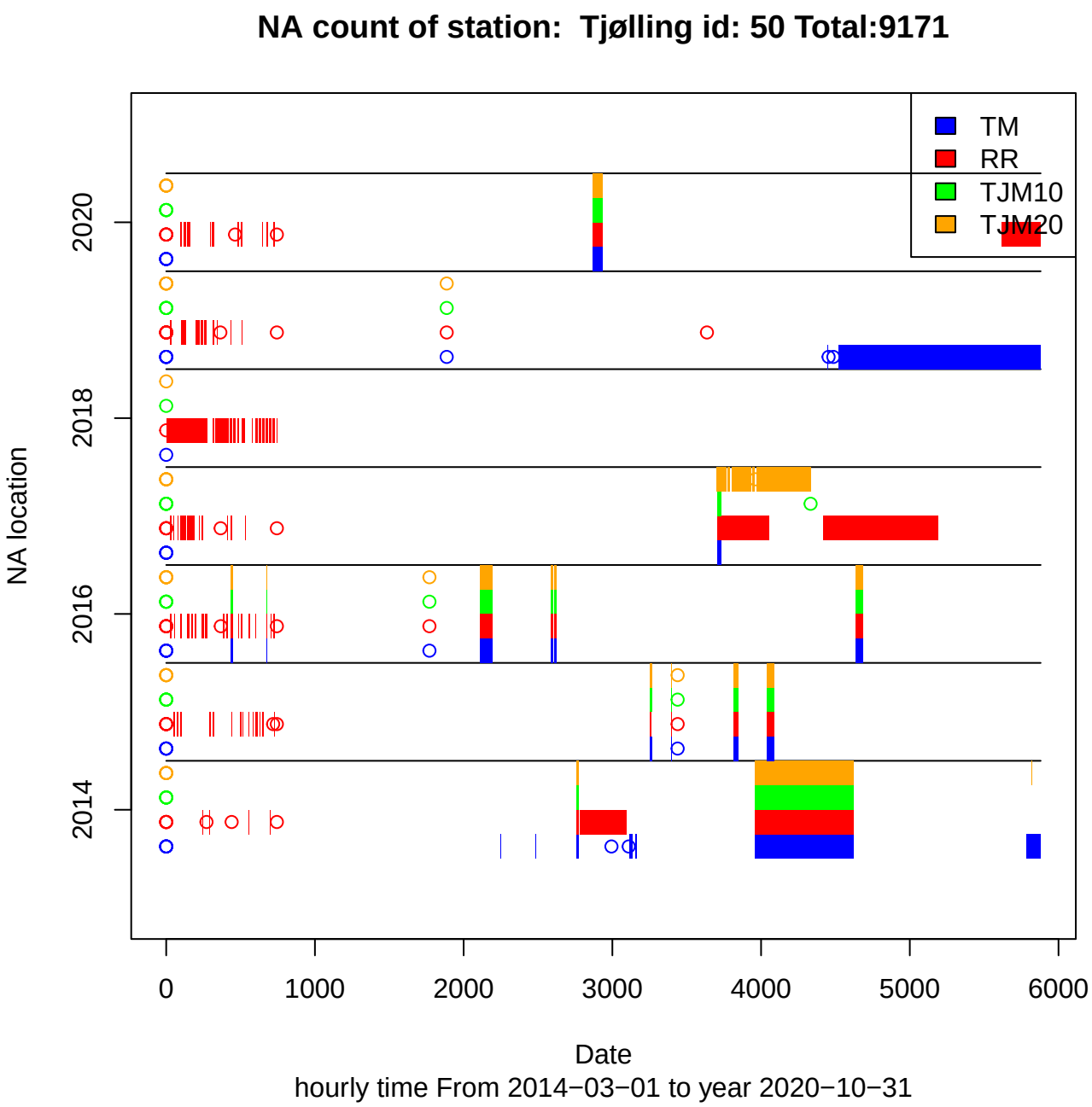

NA count of station: Særheim id: 48 Total:6494
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31

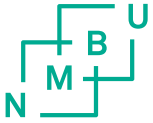

NA count of station: Hjelmeland id: 22 Total:2968
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31


NA count of station: Lier id: 30 Total:1948
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31


NA count of station: Ramnes id: 38 Total:11955
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31


NA count of station: Sande id: 42 Total:6317
TM
RR
TJM10
TJM20
NA location
2020
2018
2016
2014
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31


NA count of station: Tjølling id: 50 Total:9171
TM
RR
TJM10
TJM20
NA location
2014
2016
2018
2020
0
1000
2000
3000
4000
5000
6000
Date
hourly time From 2014−03−01 to year 2020−10−31




**Norges miljø- og biovitenskapelige universitet**
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway