Package 'SiteAdapt'

December 3, 2020

Type Package

Title Site Adaptation of Solar Irradiance Modeled Series

Version 1.1.0

Author Carlos M. Fernandez-Peruchena

Maintainer Carlos M. Fernandez-Peruchena <cfernandez@cener.com>

Description The SiteAdapt procedure improves the accuracy of modeled solar irradiance series through site-adaptation with coincident ground-based measurements relying on the use of a regression preprocessing followed by an empirical quantile mapping (eQM) correction. Fernandez-Peruchena et al (2020) <doi:10.3390/rs12132127>.

Imports glmulti, solaR, hyfo, hydroGOF, RColorBrewer, ggplot2, ggpubr, stats

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2020-12-03 15:00:02 UTC

R topics documented:

`adapt_process` *Site adaptation of solar irradiance modeled series with coincident ground measurements*

Description

Site adaptation of solar irradiance modeled series with coincident ground measurements

Usage

```
adapt_process(
  subset_target_period,
  latitude_target,
  z_target,
  subset_calibrating_period,
  latitude_calibrat,
  z_calibrat,
  GHI_threshold,
  DNI_threshold
)
```

Arguments

`subset_target_period`	Dataframe object with solar radiation series to be adapted including time (with same time zone as subset_calibrating_period), the solar irradiance modeled series to be site adapted, along with their clear sky index and solar elevation (degrees)
`latitude_target`	Site latitude of solar radiation series to be adapted (degrees, +N)
`z_target`	Site elevation above sea level of solar radiation series to be adapted (m)
`subset_calibrating_period`	Dataframe object with solar radiation series for calibrating including time (with same time zone as subset_target_period), solar irradiance modeled and measured series, along with modeled clear sky index and solar elevation (degrees)
`latitude_calibrat`	Site latitude of solar radiation series for calibrating (degrees, +N)
`z_calibrat`	Site elevation above sea level of solar radiation series for calibrating (m)
`GHI_threshold`	Upper limit of GHI series (same units that Target). For automatic calulation from observed data, set it to -99
`DNI_threshold`	Upper limit of DNI series (same units that Target). For automatic calulation from observed data, set it to -99

Value

Dataframe object including time and site adapted solar irradiance series

References

Fernández-Peruchena, C.M.; Polo, J.; Martín, L.; Mazorra, L. Site-Adaptation of Modeled Solar Radiation Data: The SiteAdapt Procedure. Remote Sens. 2020, 12, 2127.

---

calibration_2016 *Dataframe object with solar radiation series for calibration.*

Description

The Dataframe includes time, modeled and measured solar irradiance, modeled clear sky index and solar elevation (degrees).

Usage

```
data(calibration_2016)
```

Format

An object of class `"data.frame"`.

Source

BSRN and CAMSRad service

References

Fernández-Peruchena, C.M.; Polo, J.; Martín, L.; Mazorra, L. Site-Adaptation of Modeled Solar Radiation Data: The SiteAdapt Procedure. Remote Sens. 2020, 12, 2127.

---

m_Kasten *Calculation of relative air mass based on Kasten parametrization*

Description

Calculation of relative air mass based on Kasten parametrization

Usage

```
m_Kasten(Sun_elev, z)
```

Arguments

Sun_elev	Sun elevation angle (degrees) above horizon
z	Site elevation above sea level (m)

Value

Relative air mass based on Kasten parametrization

References

Kasten, F. A simple parameterization of the phyrheliometric formula for determining the Linke turbidity factor. Meteorol. Rdsch. 1980, 33, 124–127.

---

observed_2013_2016 *Dataframe object with ground measured solar radiation.*

Description

The Dataframe includes time and solar irradiance measured series.

Usage

```
data(observed_2013_2016)
```

Format

An object of class "data.frame".

Source

BSRN

---

post_process *Postprocessing of adapted solar irradiance*

Description

Postprocessing of adapted solar irradiance

Usage

```
post_process(
  df_daytime,
  subset_target_period_high_elev,
  GHI_threshold,
  DNI_threshold
)
```

Arguments

df_daytime	Dataframe object with daytime adapted solar radiation series including time (with same time zone as subset_target_period_high_elev)
subset_target_period_high_elev	Dataframe object with daytime modeled solar radiation series including time (with same time zone as df_daytime)
GHI_threshold	GHI threshold value, in the same units that modeled and adapted datasets. Default value is -99
DNI_threshold	GHI threshold value, in the same units that modeled and adapted datasets. Default value is -99

Value

Dataframe object including time and site adapted solar irradiance series without inconsistencies

References

Fernández-Peruchena, C.M.; Polo, J.; Martín, L.; Mazorra, L. Site-Adaptation of Modeled Solar Radiation Data: The SiteAdapt Procedure. Remote Sens. 2020, 12, 2127.

---

pre_process *Preprocessing of solar irradiance series Site adaptation*

Description

Preprocessing of solar irradiance series Site adaptation

Usage

```
pre_process(
  subset_target_period,
  latitude_target,
  z_target,
  subset_calibrating_period,
  latitude_calibrat,
  z_calibrat,
  GHI_threshold,
  DNI_threshold
)
```

Arguments

subset_target_period	Dataframe object with solar radiation series to be adapted including time (with same time zone as subset_calibrating_period), the solar irradiance modeled series to be site adapted, along with their clear sky index and solar elevation (degrees)

latitude_target
: Site latitude of solar radiation series to be adapted (degrees, +N)

z_target
: Site elevation above sea level of solar radiation series to be adapted (m)

subset_calibrating_period
: Dataframe object with solar radiation series for calibrating including time (with same time zone as subset_target_period), solar irradiance modeled and measured series, along with modeled clear sky index and solar elevation (degrees)

latitude_calibrat
: Site latitude of solar radiation series for calibrating (degrees, +N)

z_calibrat
: Site elevation above sea level of solar radiation series for calibrating (m)

GHI_threshold
: Upper limit of GHI series (same units that Target). For automatic calulation from observed data, set it to -99

DNI_threshold
: Upper limit of DNI series (same units that Target). For automatic calulation from observed data, set it to -99

Value

Dataframe object including time and site adapted solar irradiance series

References

Fernández-Peruchena, C.M.; Polo, J.; Martín, L.; Mazorra, L. Site-Adaptation of Modeled Solar Radiation Data: The SiteAdapt Procedure. Remote Sens. 2020, 12, 2127.

---

site_adapt *Site Adaptation of Solar Irradiance Modeled Series with Coincident Ground Measurements*

Description

Site Adaptation of Solar Irradiance Modeled Series with Coincident Ground Measurements

Usage

```
site_adapt(
  Target,
  latitude_target,
  z_target,
  Calibration,
  latitude_calibrat,
  z_calibrat,
  timezone,
  GHI_threshold,
  DNI_threshold
)
```

Arguments

Target	Dataframe object with solar radiation series to be adapted including time (with same time zone as subset_calibrating_period), the solar irradiance modeled series to be site adapted, along with their clear sky index and solar elevation (degrees)
latitude_target	Site latitude of solar radiation series to be adapted (degrees, +N)
z_target	Site elevation above sea level of solar radiation series to be adapted (m)
Calibration	Dataframe object with solar radiation series for calibrating including time (with same time zone as subset_target_period), solar irradiance modeled and measured series, along with modeled clear sky index and solar elevation (degrees)
latitude_calibrat	Site latitude of solar radiation series for calibrating (degrees, +N)
z_calibrat	Site elevation above sea level of solar radiation series for calibrating (m)
timezone	Time zone specification of the calibration_period and target_period datasets
GHI_threshold	Upper limit of GHI series (same units that Target). For automatic calulation from observed data, set it to -99
DNI_threshold	Upper limit of DNI series (same units that Target). For automatic calulation from observed data, set it to -99

Value

Dataframe object including time and site adapted solar irradiance series

References

Fernández-Peruchena, C.M.; Polo, J.; Martín, L.; Mazorra, L. Site-Adaptation of Modeled Solar Radiation Data: The SiteAdapt Procedure. Remote Sens. 2020, 12, 2127.

Examples

```
# A site located in the the Namib Desert of Namibia (Gobabeb, GOB) is selected

# - latitude:   23.5614 S
# - Longitude:  15.0420 E
# - Elevation:  407.0 m asl

 # Load calibration and modeled datasets
   data(calibration_2016) # Measured from BSRN
   data(target_2013_2016) # Provided by CAMS-RAD service

target_2013_2016$time = as.POSIXct(
paste(target_2013_2016$Year, "-",
target_2013_2016$Month, "-",
target_2013_2016$Day, " ",
target_2013_2016$Hour, ":",
target_2013_2016$Minute, sep=""),
tz ="UTC")
```

```
 # Apply the site adaptation procedure
   site_adapted_series = site_adapt(
   Target = target_2013_2016,
   latitude_target = -23.5614, # Latitude of target site
   z_target = 407.0, # Elevation of target site
   Calibration = calibration_2016,
   latitude_calibrat = -23.5614, # Same location of target period
   z_calibrat = 407.0, # Same location of target period
   timezone = "UTC",
   GHI_threshold = -99, # The threshold is calculated from observed data
   DNI_threshold = -99) # The threshold is calculated from observed data


# Load measured data for evaluating the site adaptation performance:
   data(observed_2013_2016)

# Merge datasets

site_adapted_series$time = as.POSIXct(
paste(site_adapted_series$Year, "-",
site_adapted_series$Month, "-",
site_adapted_series$Day, " ",
site_adapted_series$Hour, ":",
site_adapted_series$Minute, sep=""),
tz ="UTC")

observed_2013_2016$time = as.POSIXct(
paste(observed_2013_2016$Year, "-",
observed_2013_2016$Month, "-",
observed_2013_2016$Day, " ",
observed_2013_2016$Hour, ":",
observed_2013_2016$Minute, sep=""),
tz ="UTC")


meas_model = merge(observed_2013_2016[,6:9],
target_2013_2016[,c(6:9,11)],
by = "time", all = FALSE )

meas_model_adapt = merge(meas_model,
site_adapted_series[,6:10],
by = "time", all = FALSE )


# Display scatterplots
library(RColorBrewer)
pal <- rev(brewer.pal(11,"Spectral"))
pal=pal[2:11]

library(ggplot2)
```

```
scatter_DNI.obs = ggplot() +
geom_hex(data=meas_model_adapt,aes(x=DNI.obs, y = DNI.mod),bins = 125, alpha = 1) +
scale_fill_gradientn(colours = pal)+ theme_light() +
xlab(expression(paste("Measured DNI (W / ", m^2, " )", sep=""))) +
 ylab(expression(paste("Modeled DNI (W / ", m^2, " )", sep=""))) +
 theme(legend.position = "none") +
 xlim(100, 1120) + ylim(100,1120) +
geom_abline(intercept = 0, slope = 1, color="purple", linetype="solid", size=1.5, alpha = 0.5)

plot_DNI_adapt = ggplot() +
geom_hex(data=meas_model_adapt,aes(x=DNI.obs, y = DNI_adapted),bins = 125, alpha = 1) +
 scale_fill_gradientn(colours = pal)+ theme_light() +
 xlab(expression(paste("Measured DNI (W / ", m^2, " )", sep=""))) +
 ylab(expression(paste("Adapted DNI (W / ", m^2, " )", sep=""))) +
 theme(legend.position = "none") + xlim(100, 1120) + ylim(100,1120) +
 geom_abline(intercept = 0, slope = 1, color="purple", linetype="solid", size=1.5, alpha = 0.5)

library(ggpubr)
ggarrange(scatter_DNI.obs, plot_DNI_adapt)

scatter_GHI.obs = ggplot() +
geom_hex(data=meas_model_adapt,aes(x=GHI.obs, y = GHI.mod),bins = 125, alpha = 1) +
 scale_fill_gradientn(colours = pal)+ theme_light() +
 xlab(expression(paste("Measured GHI (W / ", m^2, " )", sep=""))) +
 ylab(expression(paste("Modeled GHI (W / ", m^2, " )", sep=""))) +
 theme(legend.position = "none") + xlim(100, 1180) + ylim(100,1180) +
 geom_abline(intercept = 0, slope = 1, color="purple", linetype="solid", size=1.5, alpha = 0.5)

plot_GHI_adapt = ggplot() +
geom_hex(data=meas_model_adapt,aes(x=GHI.obs, y = GHI_adapted),bins = 125, alpha = 1) +
 scale_fill_gradientn(colours = pal) + theme_light() +
 xlab(expression(paste("Measured GHI (W / ", m^2, " )", sep=""))) +
 ylab(expression(paste("Adapted GHI (W / ", m^2, " )", sep=""))) +
 theme(legend.position = "none") + xlim(100, 1180) + ylim(100,1180) +
 geom_abline(intercept = 0, slope = 1, color="purple", linetype="solid", size=1.5, alpha = 0.5)
ggarrange(scatter_GHI.obs, plot_GHI_adapt)

scatter_DHI.obs = ggplot() +
geom_hex(data=meas_model_adapt,aes(x=DHI.obs, y = DHI.mod),bins = 125, alpha = 1) +
 scale_fill_gradientn(colours = pal)+ theme_light() +
 xlab(expression(paste("Measured DHI (W / ", m^2, " )", sep=""))) +
 ylab(expression(paste("Modeled DHI (W / ", m^2, " )", sep=""))) +
 theme(legend.position = "none") + xlim(25, 700) + ylim(25, 700) +
 geom_abline(intercept = 0, slope = 1, color="purple", linetype="solid", size=1.5, alpha = 0.5)

plot_DHI_adapt = ggplot() +
geom_hex(data=meas_model_adapt,aes(x=DHI.obs, y = DHI_adapted),bins = 125, alpha = 1) +
 scale_fill_gradientn(colours = pal)+ theme_light() +
 xlab(expression(paste("Measured DHI (W / ", m^2, " )", sep=""))) +
 ylab(expression(paste("Adapted DHI (W / ", m^2, " )", sep=""))) +
 theme(legend.position = "none") + xlim(25, 700) + ylim(25, 700) +
 geom_abline(intercept = 0, slope = 1, color="purple", linetype="solid", size=1.5, alpha = 0.5)
ggarrange(scatter_DHI.obs, plot_DHI_adapt)
```

```
 # Display ECDF plots
plot_ECDF_DNI = ggplot(data=meas_model_adapt[which(meas_model_adapt$Elev > 0),])+
 stat_ecdf(aes(DNI.obs), col="firebrick", lwd = 0.75) +
 stat_ecdf(aes(DNI.mod), col="dodgerblue", lwd = 0.75) +
 stat_ecdf(aes(DNI_adapted), col="purple", lwd = 0.75) +
 theme_light() + xlab(expression(paste("DNI (W / ", m^2, " )", sep="")))+ ylab("ECDF ( - )")+
 annotate("text", x = 50, y = 0.9, label = "Measured", col="firebrick1", size = 4)+
 annotate("text", x = 50, y = 0.8, label = "Modeled", col="dodgerblue", size = 4)+
 annotate("text", x = 50, y = 0.7, label = "Adapted", col="purple", size = 4)
plot_ECDF_DNI

plot_ECDF_GHI = ggplot(data=meas_model_adapt[which(meas_model_adapt$Elev > 0),])+
 stat_ecdf(aes(GHI.obs), col="firebrick", lwd = 0.75) +
 stat_ecdf(aes(GHI.mod), col="dodgerblue", lwd = 0.75) +
 stat_ecdf(aes(GHI_adapted), col="purple", lwd = 0.75) +
 theme_light() + xlab(expression(paste("GHI (W / ", m^2, " )", sep="")))+ ylab("ECDF ( - )")+
 annotate("text", x = 50, y = 0.9, label = "Measured", col="firebrick1", size = 4)+
 annotate("text", x = 50, y = 0.8, label = "Modeled", col="dodgerblue", size = 4)+
 annotate("text", x = 50, y = 0.7, label = "Adapted", col="purple", size = 4)
plot_ECDF_GHI

plot_ECDF_DHI = ggplot(data=meas_model_adapt[which(meas_model_adapt$Elev > 0),])+
 stat_ecdf(aes(DHI.obs), col="firebrick", lwd = 0.75) +
 stat_ecdf(aes(DHI.mod), col="dodgerblue", lwd = 0.75) +
 stat_ecdf(aes(DHI_adapted), col="purple", lwd = 0.75) +
 theme_light() + xlab(expression(paste("DHI (W / ", m^2, " )", sep="")))+ylab("ECDF ( - )")+
 annotate("text", x = 25, y = 0.9, label = "Measured", col="firebrick1", size = 4)+
 annotate("text", x = 25, y = 0.8, label = "Modeled", col="dodgerblue", size = 4)+
 annotate("text", x = 25, y = 0.7, label = "Adapted", col="purple", size = 4)
plot_ECDF_DHI




# Statistical indicators
library(hydroGOF)
pbias(meas_model_adapt$GHI.mod,meas_model_adapt$GHI.obs)
pbias(meas_model_adapt$GHI_adapted,meas_model_adapt$GHI.obs)

pbias(meas_model_adapt$DNI.mod,meas_model_adapt$DNI.obs)
pbias(meas_model_adapt$DNI_adapted,meas_model_adapt$DNI.obs)

pbias(meas_model_adapt$DHI.mod,meas_model_adapt$DHI.obs)
pbias(meas_model_adapt$DHI_adapted,meas_model_adapt$DHI.obs)

rmse(meas_model_adapt$GHI.mod,meas_model_adapt$GHI.obs)
rmse(meas_model_adapt$GHI_adapted,meas_model_adapt$GHI.obs)

rmse(meas_model_adapt$DNI.mod,meas_model_adapt$DNI.obs)
```

```
rmse(meas_model_adapt$DNI_adapted,meas_model_adapt$DNI.obs)

rmse(meas_model_adapt$DHI.mod,meas_model_adapt$DHI.obs)
rmse(meas_model_adapt$DHI_adapted,meas_model_adapt$DHI.obs)
```

SOLIS *DNI under clear sky conditions (Solar Energy, 82(8), 758-762)*

Description

DNI under clear sky conditions (Solar Energy, 82(8), 758-762)

Usage

```
SOLIS(top, Sun_elev, aod_380, aod_500, w, site_elevation)
```

Arguments

top	Solar irradiance at the top of atosphere
Sun_elev	Sun elevation angle (degrees) above horizon
aod_380	Aerosol optical depth at 380 nm (dimensionless)
aod_500	Aerosol optical depth at 500 nm (dimensionless)
w	Atmospheric water vapor content (cm)
site_elevation	Site elevation above sea level (m)

Value

Dataframe object including time and site adapted solar irradiance series

References

Ineichen, P. A broadband simplified version of the Solis clear sky model. Sol. Energy 2008, 82, 758–762.

target_2013_2016 *Dataframe object with modeled solar radiation series.*

Description

The Dataframe includes time, modeled solar irradiance, modeled clear sky index and solar elevation (degrees).

Usage

```
data(target_2013_2016)
```

Format

An object of class "data.frame".

Source

BSRN

References

Fernandez-Peruchena, C. M. et al (2020). Site-adaptation of modeled solar radiation data: The SiteAdapt procedure. Remote Sensing.

TOA *Calculation of the Top of Atmosphere (TOA) solar irradiance on a horizontal plane*

Description

Calculation of the Top of Atmosphere (TOA) solar irradiance on a horizontal plane

Usage

```
TOA(latitude, Sun_elev, Time_Stamp)
```

Arguments

latitude	Site latitude (degrees, +N)
Sun_elev	Sun elevation angle (degrees) above horizon
Time_Stamp	Time series (object of class "POSIXct")

Value

Top of Atmosphere (TOA) Solar irradiance on a horizontal plane

Index