R tools for MFAST

Stefan Mroczek

January 27, 2017

Chapter 1

R tools for MFAST

1.1 Background

R is a free software environment designed for statistical computing (R Core Team 2013). Due to its wide usage (especially among the statistics community), there many packages are available for various applications most of which are statistical in nature.

To analyse the shear wave splitting data we have used almost exclusively R. To achieve this we have written a number of functions and scripts designed specifically to read and analyse the data. These are available on github (http://github.com/shearwavesplitter/R-tools-for-MFAST) and are documented here. The eventual aim is to be build these functions into a package written to a standard high enough to submit to CRAN (The Comprehensive R Archive Network).

All functions and scripts are written in R version 3.1.2. Compatibility depends mostly on packages being available for a particular release.

1.1.1 Required packages

These packages are available on the CRAN website (https://cran.r-project.org) or can be directly downloaded from within R.

- circular - For dealing with circular quantities (in our case axial polarisations) (Agostinelli and Lund 2013). Pewsey, Neuhäuser, and Ruxton (2013) provides a good overview of using circular statistics in R using this package.
- movMF - (Hornik and Grün 2014)
- RColorBrewer - Colour palettes for plots (Neuwirth 2014)
- fields Curve, surface and function fitting for spatial data (Nychka et al. 2015)
- ggplot2 - For producing plots (Wickham 2009). Used for plots in this thesis
- ggmap - For producing maps (Kahle and Wickham 2013). Used for maps in this thesis

1.2 Functions

1.2.1 sm.createTESSA

Description

Creates a TESSA input file

Usage

sm.createTESSA(summfile, name="out.summ", path="~")

Arguments

summfile	
name	Name of the output file
path	Location to save output

Details

sm.createTESSA creates an input file for the TESSA codes using a summfile that has been read (and graded etc.) within R.

Examples

```
## Create input file for events from all stations with a final
    grade of F3
dat <- sm.read('~/summaryfiles')
subset(dat, finalgrade == "F3")
sm.createTESSA(dat)
```

1.2.2 sm.getevents

Description

Return specific events from dataframe

Usage

```
sm.getevents(summ, events, station=NULL )
```

Arguments

summ	.summ dataframe (output of sm.read)
events	a vector containing event cuspid
station	Station, NULL for all stations

Details

sm.getevents returns a dataframe of measurements with cuspids listed in the events vector.

Examples

```
## Return measurements for events with cuspids stored in text
    file
dat <- sm.read('~/summaryfiles')
cs <- read.table('~/R_MFAST/cluster1') #Read a table of cuspids
sm.getvents(dat,cs$V1,station="WPRZ")
```

1.2.3 sm.getevents

Description

Return specific events from dataframe

Usage

sm.getevents(summ, events, station=NULL)

Arguments

summ	.summ dataframe (output of sm.read)
events	a vector containing event cuspid
station	Station, NULL for all stations

Details

sm.getevents returns a dataframe of measurements with cuspids listed in the events vector.

Examples

```
## Return measurements for events with cuspids stored in text
    file
dat <- sm.read('~/summaryfiles')
cs <- read.table('~/R_MFAST/cluster1') #Read a table of cuspids
sm.getvents(dat,cs$V1,station="WPRZ")
```

1.2.4 sm.pathclus

Description

Clustering of station to event paths

Usage

sm.pathclus(data, hvec=NULL, kmax=7, path="~", plotextra=TRUE, rot=180)

Arguments

data	.summ dataframe (output of sm.read)
hvec	Vector of station elevations (in order of unique(data$stat))
kmax	Maximum number of clusters for each station
path	Path to save clusters and cluster figures
plotextra	Include 2D and 3D plots (TRUE/FALSE)
rot	Rotation of 3D plot of piece points

Details

sm.pathclus uses movMF (Section **??**) to cluster pierce points on a unit hemisphere below each station. Pierce points are calculated for straight line station-event paths intersect the unit hemisphere. This function automatically divides the stations up so it does not have to be run individually.

Output

Vectors of the cuspids for each cluster are saved in path along with 3D figure of the clustered pierce points, a simple 2D map of the clusters and the ros diagrams of each cluster. The name of cuspid text files contain the station and the p-value returned from a test (i.e. WPRZ_cluster5_p-val_0.196)

Examples

```
## Cluster all stations (individually)table
dat <- sm.read('~/summaryfiles')
sm.pathclus(dat,path='~/R_MFAST/clusters')
```

1.2.5 sm.plot

Description

Rose plots of of axial data (e.g. shear wave splitting polarisations)

Usage

sm.proj(data, name1="raw.eps", name2="double.eps", path="~", cols="blue", antipodal="lightblue", bins=16, kd=TRUE, arrow=TRUE, medarrow=FALSE)

Arguments

data	Vector of axial values (polarisations)
name1	Name of the output file
name2	Name of the output file (double angled plot)
path	Location to save output
cols	Colour of points
antipodal	Colour of antipodal points
bins	Number of bins for rose plot
kd	Option to plot kernel density estimate (with a smoothing bandwidth of 10)
arrow	Option to plot average arrow
medarrow	Option to plot median arrow

Details

sm.plot creates rose plots for visualisation of of axial data

Examples

```
## Rose plot of polarisations for station WPRZ
dat <- sm.read('~/summaryfiles', station='WPRZ')
```

```
sm.plot(dat$fast)
```

1.2.6 sm.proj

Description

Projection map of shear wave splitting measurements

Usage

sm.proj(summf, lm=2, pierce=1.5, savpath="~", savnam="proj.png", mlat=NULL, mlon=NULL, zoom=13, xvec=NULL, yvec=NULL, hvec=NULL)

Arguments

summf	.summ dataframe (output of sm.read)
lm	Length multiplier
pierce	Pierce depth of projection (km from sea level)
savpath	Location to save output map
savnam	Name of output map
mlat	Latitude to centre map on
mlon	Longitude to centre map on
zoom	Zoom of getgooglemap() function (see **ggmap** documentation)
xvec	Vector longitude of perturbations to adjust station name locations (°)
yvec	Vector latitude perturbations to station name locations (°)
hvec	Vector of station elevations (in order of unique(summf$stat))

Details

sm.proj projects shear wave splitting measurements onto a map. The point where the station-event path pierces the given depth (pierce) is projected to

the surface and a vector is drawn. The vector is oriented in the direction of the fast azimuth with its length scaled with delay time (multiplied by lm).

Examples

```
## Projection map of shear wave splitting measurements for
    station WPRZ. Centring of the map is automatic.
dat <- sm.read('~/summaryfiles', station='WPRZ')
sm.proj(dat)
## Projection map of all stations with location of 3rd station's
     label moved 0.02 degrees.
dat <- sm.read('~/summaryfiles')
latpet <- rep(0,length(dat$fast))
latrep <- rep(0,length(dat$fast))
lonrep <- rep(0,length(dat$fast))
lonrep[3] <- 0.02
latrep[3] <- 0.02
sm.proj(dat,mlat=-38.57,mlon=176.2,xvec=lonrep,yvec=latrep)
```

1.2.7 sm.read

Description

Reads and grades a raw .summ file

Usage

sm.read(path, tlagmax=1, station=NULL, minl=0, minsnr=3, type=2)

sm.read_l(path, tlagmax=0.4, station=NULL, minl=0, minsnr=3, type=2)

sm.read_vl(path, tlagmax=0.2, station=NULL, minl=0, minsnr=3 ,type=2)

Arguments

path	Path to folder containing .summ files
tlagmax	Maximum time delay
station	Station(s) to read in. NULL reads all stations
minl	Minimum value of lambdamax allowed
minsnr	Minimum signal to noise ratio allowed
type	Grading method (see **details**)

Details

sm.read and its variants read in raw .summ files (ungraded), grades the measurements and then returns a dataframe of the AB grade measurements (depending on the grading process). Grading is either type=1 where the AB measurement with the best filter from a unique event are chosen or the default type=2 where the measurement with the lowest error from an earthquake is chosen (detailed in Castellazzi et al. 2015). type=2 has an additional column finalgrade where measurements are assigned F1 (where only one of the three best filters for a unique event is AB grade), F2 (where two are AB and are within 10° of each other), and F3 (where three are high grade and within 10° of each other). A third category, F2b, indicates that there are three high grade measurements but only two are within 10° of each other. sm.read_l and sm.read_vl are for the local and very local versions of the MFAST code respectively.

Format

The output of sm.read is a dataframe containing all the columns of a .summ file. Column names match those in the .summ files except they do not contain any symbols or numbers except for underscores (i.e. "10dist_(ev-stat)" would

"dist_evstat"). Column 1 "event" records the filter number used. All values remain identical **except** fast (fast azimuth) which is converted to radians and given the data type "circular".

Examples

```
## Read station WPRZ and calculate its mean orientation
dat <- sm.read('~/summaryfiles', station='WPRZ')
mean <- mean.circular(dat$fast*2)/2
## Read multiple stations
dat <- sm.read('~/summary files', station=c('WPRZ', 'ARAZ'))
```

Notes

IMPORTANT: There is a bug in MFAST whereby, if you have periods in your event names, an incorrect cuspid (unique event identifier) is saved. This prevents sm.read from working. To fix this navigate to your mfast/bin folder and add cuspid=${fn%.*-*.fb*}; cuspid=${cuspid%.0} to line 154 in mfast_logfiles (as of MFAST v2.2). This should work for all cases *unless* your names end in ".0".

1.2.8 sm.readraw

Description

Reads and returns .summ file

Usage

sm.readraw(path, tlagmax=1, station=NULL, minsnr=3)

Arguments

path	Path to folder containing .summ files
tlagmax	Maximum time delay
station	Station(s) to read in. NULL reads all stations
minsnr	Minimum signal to noise ratio allowed

Details

sm.readraw read in raw .summ files (ungraded), grades the measurements and then returns a dataframe of the all the measurements in the summ file (including all three filters for each event).

Format

See **sm.read** (Section 1.2.7).

Examples

```
## Read station WPRZ
dat <- sm.readraw('~/summaryfiles', station='WPRZ')
## Read multiple stations
dat <- sm.readraw('~/summary files', station=c('WPRZ', 'ARAZ'))
```

Notes

See **sm.read** (Section 1.2.7).

1.2.9 sm.stde

Description

Calculates standard error of a mean orientation using a bootstrap method

Usage

sm.stde(data, seed=NULL, iter=9999)

Arguments

data	Vector of fast azimuths (radians, undoubled)
seed	Random number generator state. Set to allow reproducibility
iter	Number of iterations for bootstrapping

Details

Standard error is a measure of how close the sample mean is to the mean of the underlying population of fast azimuths. sm.stde calculates the standard error by re-sampling (with replacement) the doubled fast azimuths and calculating the mean of the re-sampled values. This is repeated iter times with the mean calculated each time. The standard error is the standard deviation of these means. Angles are doubled and then halved within the function.

Examples

```
## Calculate standard error of station WPRZ orientation
dat <- sm.read('~/summaryfiles', station='WPRZ')
se <- sm.stde(dat$fast)
## For the same seed the calculated standard error will be
    identical every time it is run
dat <- sm.read('~/summaryfiles', station='WPRZ')
se <- sm.stde(dat$fast, seed=2016)
```

1.3 Scripts

Bibliography

Agostinelli, C. and U. Lund (2013). *R package* `circular`*: Circular Statistics (version 0.4-7)*. CA: Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University, Venice, Italy. UL: Department of Statistics, California Polytechnic State University, San Luis Obispo, California, USA.

Castellazzi, Claire et al. (2015). "Shear wave automatic picking and splitting measurements at Ruapehu volcano, New Zealand". In: *Journal of Geophysical Research. Solid Earth* 120.5, pp. 3363–3384.

Hornik, Kurt and Bettina Grün (2014). "movMF: An R Package for Fitting Mixtures of von Mises-Fisher Distributions". In: *Journal of Statistical Software* 58.10, pp. 1–31.

Kahle, David and Hadley Wickham (2013). "ggmap: Spatial Visualization with ggplot2". In: *The R Journal* 5.1, pp. 144–161.

Neuwirth, Erich (2014). *RColorBrewer: ColorBrewer Palettes*. R package version 1.1-2.

Nychka, Douglas et al. (2015). *fields: Tools for spatial data*. R package version 8.4-1. Boulder, CO, USA: University Corporation for Atmospheric Research.

Pewsey, Arthur, Markus Neuhäuser, and Graeme D Ruxton (2013). *Circular statistics in R*. Oxford University Press.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.

Wickham, Hadley (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.