Package 'MFASTR'

January 25, 2018

Title The Multiple Filter Automatic Shear Wave Splitting Technique in R

Version 1.4

Date 2017-11-26

Description This is a port of the MFAST codes into R. The main functions do_station_simple and do_station_complex replicate MFAST's usability. Other functions in this package are documented to give advanced users more fexability. In contrast to MFAST, MFASTR (by default) uses zerophase filters and does not downsample.

URL http://mfast-package.geo.vuw.ac.nz/

BugReports https://github.com/shearwavesplitter/MFASTR/issues

Depends R (>= 3.0.0), RSEIS, TauP.R, circular

Imports signal, parallel

SystemRequirements GNU make, Linux

License

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

R topics documented:

`ak135_alp` *The ak135_alp velocity model*

Description

The ak135_alp velocity model

Usage

```
ak135_alp
```

Format

A TauP.R compatible velocity model

ak135_taupo *The ak135_taupo velocity model*

Description

The ak135_taupo velocity model

Usage

```
ak135_taupo
```

Format

A TauP.R compatible velocity model

all6plot *Create all6 plot*

Description

Create an all6 plot for a specific event

Usage

```
all6plot(path, cuspid, filter = 1, zerophase = TRUE, E = ".e", N = ".n",
  Z = ".z", auto = FALSE)
```

Arguments

path	Path to folder containing raw events and output folder
filter	Which filter to plot (e.g. 1 for fb1)
zerophase	Were the filters applied zero phase?
E	Vector signal of the east component
N	Vector signal of the north component
Z	Vector signal of the vertical component
auto	Select the first event if multiple are available?
display	Display plots within R?

all6_station *Plot all6*

Description

A wrapper for the all6plot function to plot all events from a station

Usage

```
all6_station(path, filter = c(1, 2, 3), zerophase = TRUE, E = ".e",
  N = ".n", Z = ".z")
```

Arguments

path	Path to folder containing raw events and output folder
filter	Which filter to plot (e.g. c(1,2,3) for fb1, fb2 & fb3)
zerophase	Were the filters applied zero phase?
E	Vector signal of the east component
N	Vector signal of the north component
Z	Vector signal of the vertical component

anginc *Angle of incidence*

Description

Determines the angle of incidence for an event

Usage

```
anginc(tvel, trip)
```

Arguments

tvel	Veloctity model read in by readtvel or a stored model (ak135_alp, ak135_taupo)
trip	Seismogram triplet (output of readtriplet)

Value

The angle of incidence at the surface (degrees) and the ray parameter

Examples

```
# Determine the angle of incidence for event 2002.054.09.47.lhor2
pathto <- "~/mfast/sample_data/raw_data"
write_sample(pathto)
event <- "2002.054.09.47.lhor2"
triplet <- readtriplet(event,path=pathto)
a <- anginc(ak135_alp,triplet)
```

`checkcomp` *Check components*

Description

Checks a folder to make sure all three components are present and moves those with missing components to a subdirectory

Usage

```
checkcomp(path, E = ".e", N = ".n", Z = ".z")
```

Arguments

`path`	Path to folder
`E`	Suffix of the east component
`N`	Suffix of the north component
`Z`	Suffix of the vertical component

`checkspick` *Check S-wave picks*

Description

Checks a folder to make sure all events have S-wave picks and moves those with missing picks to a subdirectory

Usage

```
checkspick(path, suffix = "E", header = "t0", E = ".e", N = ".n",
  Z = ".z")
```

Arguments

path	Path to folder
suffix	Which component to look for the S-pick in (E, N, or Z)
header	Header name of where the S-pick is stored
E	Suffix of the east component
N	Suffix of the north component
Z	Suffix of the vertical component

---

createini *Create .ini*

Description

Creates an MFAST .ini (paramter) file

Usage

```
createini(path, trip, filts, name, number = 3, E = ".e", N = ".n",
  Z = ".z", nwbeg = 5, fdmin = 0.3, fdmax = 8, t_win_freq = 3,
  tlagmax = 1, Ncmin = 5, Mmax = 15, zerophase = TRUE)
```

Arguments

path	Path to folder
trip	Seismogram triplet (output of readtriplet)
name	Event name (without suffix)
number	Number of best filters to use
nwbeg	number of start times tested
fdmin	Minimum allowed dominant frequency
fdmax	Maximum allowed dominant frequency
t_win_freq	Window to calculate the dominant frequency
tlagmax	Maximum allowed time delay (s)
Ncmin	Minimum number of points in an acceptable cluster
Mmax	maximum number ofclusters
A	dataframe of the best filters (output of filter_spread)
suffe	Suffix of east component
suffn	Suffix of north component
suffz	Suffix of vertical component
snrmax	Minimum snr allowed for a good filter
t_win_snr	Window for SNR (s)
t_err	Modification to t_win_snr to account for error in S-pick (s)

Value

A vector of dominant frequency in the S-wave (maxfreq) for each filter

Examples

```
# Create .ini file for event 2002.054.09.47.lhor2
pathto <- "~/mfast/sample_data/raw_data"
write_sample(pathto)
event <- "2002.054.09.47.lhor2"
triplet <- readtriplet(event,path=pathto)
bestfilt <- filter_spread(triplet)
maxfreq <- createini(pathto,triplet,bestfilt,event)
```

---

cut_simple *Simple cut*

Description

A simple routine to cuts out portion of a vector signal

Usage

```
cut_simple(x, dt, t1, t2, b = 0)
```

Arguments

x	vector signal
dt	sample interval
t1	Begin cut time
t2	End cut time

Value

A cut vector signal

`det.type` *Classify events*

Description

Determines which type an event can be classified as

Usage

```
det.type(summ, cutoff = 15, cutoff2 = 35, minper = 0.9,
  depthcutoff = 0.17, sww = 45, tvel = ak135_taupo, plot = TRUE)
```

Arguments

`summ`	A summary file (usually from a single station)
`cutoff`	Maximum propgation angle allowed for a Type 1 event
`cutoff2`	Maximum propgation angle allowed for a Type 2 event
`minper`	Percentage of the path that must be below the maximum to be classified into that type
`depthcutoff`	What percentage of the straight line path must be below cutoff2 to be classified as Type 2
`sww`	The shear wave window cutoff
`tvel`	The velocity model to be used (this must be the same as the one used to determine the ray parameters)
`plot`	Should the result be plotted?

Details

This function takes the events in a summary file and classifies each of them (based on their propogation angles) as Type 1 or Type 3 (full delay time and correct polarisation), Type 2 (correct polarisation) or Type NULL (low delay time and incorrect polarisation)

Value

A dataframe containing the summary file with the type for each event

do_all_complex *Run MFAST on multiple stations with more options*

Description

Run shear wave splitting measurements on multiple folders/stations

Usage

```
do_all_complex(path, sheader = "t0", nwbeg = 5, fdmin = 0.3, fdmax = 8,
  t_win_freq = 3, tlagmax = 1, Ncmin = 5, Mmax = 15, snrmax = 3,
  t_win_snr = 3, t_err = 0.02, filtnum = 3, type = "normal",
  filter = NULL, tvelpath = NULL, tvel = ak135_alp, suffe = ".e",
  suffn = ".n", suffz = ".z", zerophase = TRUE, no_threads = NULL)
```

Arguments

path	Path to folder containing folders with events
sheader	SAC header the S-wave pick is stored in
nwbeg	number of start times tested
fdmin	Minimum allowed dominant frequency
fdmax	Maximum allowed dominant frequency
t_win_freq	Window to calculate the dominant frequency (s)
tlagmax	Maximum allowed time delay (s)
Ncmin	Minimum number of points in an acceptable cluster
Mmax	maximum number of clusters
snrmax	Minimum snr allowed for a good filter
t_win_snr	Window for SNR (s)
t_err	Modification to t_win_snr to account for error in S-pick (s)
filtnum	Number of filters to test
type	Which of the MFAST default settings and filters to use. If a P-wave pick is present, type="verylocal" uses it to set t_win_snr
filter	User defined set of filters (this overrides the filter selected with type).
tvelpath	Path to a .tvel file containing the velocity model (overrides tvel)
tvel	A tvel file read with readtvel (ak135_alp and ak135_taupo are already loaded)
suffe	Suffix of east component
suffn	Suffix of north component
suffz	Suffix of vertical component
no_threads	Number of threads to run measurements on. Set to 1 for verbose mode. Defaults to the number of cores

Value

A dataframe containing a summary of all the stations

Examples

```
# Run on measurements three folders of the normal sample data
write_sample("~/mfast/sample_data/raw_data")
write_sample("~/mfast/sample_data/raw_data2")
write_sample("~/mfast/sample_data/raw_data3")
do_all_complex(path="~/mfast/sample_data")
```

---

do_all_simple *Run MFAST on multiple folders*

Description

Run shear wave splitting measurements on more than one folder/station

Usage

```
do_all_simple(path, sheader = "t0", type = "normal", filtnum = 3,
  tvelpath = NULL, tvel = ak135_alp, zerophase = TRUE,
  no_threads = NULL)
```

Arguments

path	Path to folder containing folders with events
sheader	SAC header the S-wave pick is stored in
type	Which of the MFAST default settings and filters to use
filtnum	Number of filters to test
tvelpath	Path to a .tvel file containing the velocity model (overrides tvel)
tvel	A tvel file read with readtvel (ak135_alp and ak135_taupo are already loaded)
no_threads	Number of threads to run measurements on. Set to 1 for verbose mode. Defaults to automatic selection

Details

Component suffixes are determined automatically

Value

A dataframe containing a summary of all the stations

Examples

```
# Run on measurements three folders of the normal sample data
write_sample("~/mfast/sample_data/raw_data")
write_sample("~/mfast/sample_data/raw_data2")
write_sample("~/mfast/sample_data/raw_data3")
do_all_simple(path="~/mfast/sample_data")
```

`do_station_complex` *Run MFAST with more options*

Description

Run shear wave splitting measurements on a folder of events with more options

Usage

```
do_station_complex(path, sheader = "t0", nwbeg = 5, fdmin = 0.3,
  fdmax = 8, t_win_freq = 3, tlagmax = 1, Ncmin = 5, Mmax = 15,
  snrmax = 3, t_win_snr = 3, t_err = 0.02, filtnum = 3,
  type = "normal", filter = NULL, tvelpath = NULL, tvel = ak135_alp,
  suffe = ".e", suffn = ".n", suffz = ".z", zerophase = TRUE,
  no_threads = NULL, mc.preschedule = TRUE, downsample = FALSE)
```

Arguments

`path`	Path to folder
`sheader`	SAC header the S-wave pick is stored in
`nwbeg`	number of start times tested
`fdmin`	Minimum allowed dominant frequency
`fdmax`	Maximum allowed dominant frequency
`t_win_freq`	Window to calculate the dominant frequency (s)
`tlagmax`	Maximum allowed time delay (s)
`Ncmin`	Minimum number of points in an acceptable cluster
`Mmax`	maximum number of clusters
`snrmax`	Minimum snr allowed for a good filter
`t_win_snr`	Window for SNR (s)
`t_err`	Modification to t_win_snr to account for error in S-pick (s)
`filtnum`	Number of filters to test
`type`	Which of the MFAST default settings and filters to use. If a P-wave pick is present, type="verylocal" uses it to set t_win_snr
`filter`	User defined set of filters (this overrides the filter selected with type).
`tvelpath`	Path to a .tvel file containing the velocity model (overrides tvel)

tvel	A tvel file read with readtvel (ak135_alp and ak135_taupo are already loaded)
suffe	Suffix of east component
suffn	Suffix of north component
suffz	Suffix of vertical component
no_threads	Number of threads to run measurements on. Set to 1 for verbose mode. Defaults to the number of cores
downsample	Downsample if sampling rate is less than 0.01s (Defaults to FALSE, originally used to decrease computational loads)

Value

A dataframe containing the summary file

Examples

```
# Run on measurements the normal sample data with defaults
write_sample("~/mfast/sample_data/raw_data")
do_station_complex(path="~/mfast/sample_data/raw_data")

# Run measurements with your own defined filters
filt_low <- c(0.1,0.2,0.5)
filt_high <- c(1,2,3)
filts <- cbind(filt_low,filt_high)
write_sample("~/mfast/sample_data/raw_data")
do_station_complex(path="~/mfast/sample_data/raw_data",filter=filts)
```

---

do_station_simple *Run MFAST*

Description

Run shear wave splitting measurements on a folder of events

Usage

```
do_station_simple(path, sheader = "t0", type = "normal", filtnum = 3,
  tvelpath = NULL, tvel = ak135_alp, zerophase = TRUE,
  no_threads = NULL, mc.preschedule = TRUE, downsample = FALSE)
```

Arguments

path	Path to folder
sheader	SAC header the S-wave pick is stored in
type	Which of the MFAST default settings and filters to use
filtnum	Number of filters to test
tvelpath	Path to a .tvel file containing the velocity model (overrides tvel)

tvel	A tvel file read with readtvel (ak135_alp and ak135_taupo are already loaded)
no_threads	Number of threads to run measurements on. Set to 1 for verbose mode. Defaults to the number of cores
downsample	Downsample if sampling rate is less than 0.01s (Defaults to FALSE, originally used to decrease computational loads)

Details

Component suffixes are determined automatically

Value

A dataframe containing the summary file

Examples

```
# Run on measurements the normal sample data
write_sample("~/mfast/sample_data/raw_data")
do_station_simple(path="~/mfast/sample_data/raw_data")

# Run on measurements the verylocal sample data where the S-pick is stored in the t5 header
write_sample("~/mfast/sample_data/raw_data",type="verylocal")
do_station_simple(path="~/mfast/sample_data/raw_data",type="verylocal",sheader="t5")
```

---

dt.weighted *Mean delay time*

Description

Determine the mean weighted delay time

Usage

```
dt.weighted(summ, weights = c(1, 2, 3))
```

Arguments

summ	Dataframe containing Castelazzi graded events (CZ_*.summ)
weights	A vector containing the weights with length equal to the number of filters used (usually 3) in order with the first corresponding to F1

Value

A list containing the weighted mean delay time, and mean delay time per kilometre (straightline) path length as well as their respective standard deviations and standard errors.

fast.weighted *Mean fast polarisation*

Description

Determine the mean weighted fast polarisation

Usage

```
fast.weighted(summ, weights = c(1, 2, 3))
```

Arguments

summ	Dataframe containing Castelazzi graded events (CZ_*.summ)
weights	A vector containing the weights with length equal to the number of filters used (usually 3) in order with the first corresponding to F1 or a weight for each measurement

Value

A list containing the weighted mean polarisation, its pythagorean length, and the (weighted) p-value from the Rayleigh test

filter_spread *Find best filters*

Description

Determines the best filters for an event

Usage

```
filter_spread(trip, type = "normal", filter = NULL, t_win_snr = 3,
  t_err = 0.05, snrmax = 3, zerophase = TRUE)
```

Arguments

trip	Seismogram triplet (output of readtriplet)
type	Which of the default filters to use. If a P-wave pick is present, type="verylocal" uses it to set t_win_snr
filter	User defined filters. Overrides filters selected by type (for "verylocal" the P-pick is still used)
t_win_snr	Window for SNR
t_err	Modification to t_win_snr to account for error in S-pick
snrmax	Minimum snr allowed for a good filter

Value

A dataframe of the filters sorted by SNR*bandwidth

Examples

```
# Define your own set of filters
filt_low <- c(0.1,0.2,0.5)
filt_high <- c(1,2,3)
filts <- cbind(filt_low,filt_high)
write_sample("~/mfast/sample_data/raw_data")
triplet <- readtriplet("2002.054.09.47.lhor2",path="~/mfast/sample_data/raw_data")
bestfilt <- filter_spread(triplet,filter=filts)
```

---

getevents *Get events*

Description

A handy function to retrieve specific events from a summary dataframe

Usage

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

Arguments

summ	Dataframe containing the summary file
events	A vector containing the rquired event names
station	Defaults to events on all stations

---

grade *Grade .summ file*

Description

Grades a .summ file (do_station automatically grades)

Usage

```
grade(path, minsnr = 3, tlagmax = 1, minl = 0, mfast = FALSE)
```

Arguments

path	Path to .summ file to be graded
minsnr	Minimum SNR allowed for an AB+ grade
tlagmax	Maximum time delay allowed for an AB+ grade
minl	Minimum lambdamax allowed for a AB+ grade
mfast	Set to TRUE to grade a .summ file produced by the original MFAST

Examples

```
# (Re)grade LHOR2.75.summ
write_sample("~/mfast/sample_data/raw_data")
do_station_simple(path="~/mfast/sample_data/raw_data")
pathto <- "~/mfast/sample_data/raw_data/LHOR2.summ_files/LHOR2.75.summ"
grade(pathto)
```

---

logfiles *Parse results*

Description

Parses output of shear wave splitting measurement for a set of filters (used to build .summ files)

Usage

```
logfiles(path, name, trip, filtlist, maxfreqv, comment = "MFASTR", anginc)
```

Arguments

path	Path to folder
name	Name of event
trip	Seismogram triplet (output of readtriplet)
filtlist	Dataframe of the best filters to be used (output of writesac_filt)
maxfreqv	Vector of dominant frequency in the S-wave (maxfreq) for each filter(output of create_ini)
comment	Optional comment
anginc	Angle of indidence (output of anginc)

Value

A dataframe containing the results for that event

Examples

```
# Run shear wave splitting measurement on event 2002.054.09.47.lhor2 and parse the results
pathto <- "~/mfast/sample_data/raw_data"
write_sample(pathto)
event <- "2002.054.09.47.lhor2"
triplet <- readtriplet(event,path=pathto)
a <- anginc(ak135_alp,triplet)
bestfilt <- filter_spread(triplet)
maxfreq <- createini(pathto,triplet,bestfilt,event)
f <- writesac_filt(pathto,triplet,event,bestfilt)
run_mfast(pathto,event,f)
res <- logfiles(pathto,event,triplet,f,maxfreq,anginc=a)
```

---

mclapply2 *Wrapper around mclapply to track progress*

Description

Based on http://stackoverflow.com/questions/10984556

Usage

```
mclapply2(X, FUN, ..., mc.preschedule = TRUE, mc.set.seed = TRUE,
  mc.silent = FALSE, mc.cores = getOption("mc.cores", 2L),
  mc.cleanup = TRUE, mc.allow.recursive = TRUE, mc.progress = TRUE,
  mc.style = 3)
```

Arguments

X	a vector (atomic or list) or an expressions vector. Other objects (including classed objects) will be coerced by 'as.list'
FUN	the function to be applied to
...	optional arguments to 'FUN'
mc.preschedule	see mclapply
mc.set.seed	see mclapply
mc.silent	see mclapply
mc.cores	see mclapply
mc.cleanup	see mclapply
mc.allow.recursive	see mclapply
mc.progress	track progress?
mc.style	style of progress bar (see txtProgressBar)

Examples

```
x <- mclapply2(1:1000, function(i, y) Sys.sleep(0.01))
x <- mclapply2(1:3, function(i, y) Sys.sleep(1), mc.cores=1)
```

meanaxial *Weighted axial mean*

Description

The mean of a weighted axial variable

Usage

```
meanaxial(vec, weights = NULL)
```

Arguments

vec	A vector of axis (degrees)
weights	A vector of weights of the same length as vec

Value

The mean axis (degrees) and the Pythagorean length

moving_dt *Delay time moving average*

Description

A moving average of delay time

Usage

```
moving_dt(summfile, windowlength, windowspeed, norm = FALSE)
```

Arguments

summfile	A dataframe containing a summary file (i.e. from readmfast)
windowlength	Size of the averaging window (in days)
windowspeed	Speed of advancing window (days per sample)
norm	Normalise by straight line path distance?

Value

A dataframe containing the end days of each window along with its mean, standard deviation (of the mean), median, upper and lower 95

moving_phi *Fast polarisation moving average*

Description

A moving average of fast polarisation

Usage

```
moving_phi(summfile, windowlength, windowspeed)
```

Arguments

summfile	A dataframe containing a summary file (i.e. from readmfast)
windowlength	Size of the averaging window (in days)
windowspeed	Speed of advancing window (days per sample)

Value

A dataframe containing the end days of each window along with its mean, median fast polarisation, and 95

moving_vpvs *vP/vS moving average*

Description

A moving average of vP/Vs

Usage

```
moving_vpvs(vpvs, year, doy_det, windowlength, windowspeed)
```

Arguments

vpvs	A vector of vP/vS ratios
year	A vector of years for each vP/vS
doy_det	A vector of julian days for each vP/vS
windowlength	Size of the averaging window (in days)
windowspeed	Speed of advancing window (days per sample)

Value

A dataframe containing the end days of each window along with its mean, standard deviation (of the mean), median, upper and lower 95

pathclus *Path cluster*

Description

Clusters measurements by their station to event paths

Usage

```
pathclus(summ, savepath, hvec = NULL, kmax = 7, runs = 20,
  minsample = 55, seed = NULL, plot = TRUE, rot = 180, palette = NULL)
```

Arguments

summ	An MFASTR summary file
savepath	Path to save plots and files
hvec	A vector of station elevations
kmax	Maximum number of clusters
runs	Number of runs for the clustering
seed	Random number seed
plot	Create plots?
rot	Degrees to rotate 3D lower hemisphere plot
palette	Vector of user defined colours for plotting if the number of clusters is greater than 12
minsamples	Minimum number of measurements for that station

Details

Uses the movMF package to fit mixtures of von Mises Fisher distributions to the station to event paths projected onto a unit hemisphere below each station.

Value

Creates folders containing the cuspids of events in each cluster along with the p-value of the Rayleigh test for polarisations in that cluster.

Examples

```
# Run for all stations and save to clustest folder
cz <- summ.cz("~/summfiles")
pathclus(cz,savepath="~/clustest",plot=TRUE)
```

perani *Weighted percentage anisotropy*

Description

Determine the weighted percentage anisotropy and shear wave anisotropy for each stations in a summary file

Usage

```
perani(summ, weights = NULL)
```

Arguments

summ	Dataframe containing MFAST summary file
weights	A vector containing the desired weights

Value

A dataframe containing each station and their corresponding percentage anistropy and shear wave anisotropy. As well as average values for all stations

plotrose *Plot rose diagram*

Description

Plots a rose diagram of data from a .summ file

Usage

```
plotrose(path, summ, name = "rose.eps", bins = 16, kd = FALSE, sym = 16,
  prop = 1.3, bincol = "darkgrey", antibincol = "lightgrey",
  cols = "blue", antipodal = "lightblue", axes = TRUE, arrow = TRUE,
  arwcol = "red", arwlty = 1, arwlwd = 2)
```

Arguments

path	Path to folder to save plots
name	Name of plot
bins	Number of bins
kd	Kernal density?
sym	Symbol for outer points
prop	Scale length of rose diagram bins

bincol	Colour of bins
cols	Colour of points
antipodal	Colour of antipodal points
axes	Plot axes?
arrow	Mean arrow? (Scaled by mean resultant length)
arwcol	Arrow colour
arwlty	Arrow line type (lty)
arwlwd	Arrow line thickness (lwd)
bincol	Colour of antipodal bins

`readmfast` *Read MFAST .summ file*

Description

Reads a .summ output from the original MFAST codes

Usage

```
readmfast(path, recuspid = FALSE, header = TRUE)
```

Arguments

path	The path the the summary file
recuspid	Regenerate unique cuspids from event names? (Useful if they have been truncated in MFAST)
header	Does the summary file have a one line header?

Details

This function is used with grade() to grade .summ files produced using the original MFAST codes (by setting mfast=TRUE).

Value

A dataframe containing the summary file

`readtriplet` *Read a SAC format siesmogram triplet*

Description

Reads, cuts, and loads S-wave pick into the t5 header using RSEIS/JSAC.seis as a workhorse

Usage

```
readtriplet(event, path = ".", E = ".e", N = ".n", Z = ".z",
  header = "t0", pheader = "a", downsample = FALSE)
```

Arguments

`event`	Event name
`path`	Path to folder
`header`	Name of header containing the S-wave pick
`pheader`	Name of header containing the P-wave pick
`downsample`	Downsample if sampling rate is less than 0.01s (Defaults to FALSE, originally used to decrease computational loads)
`suffe`	Suffix of east component
`suffn`	Suffix of north component
`suffz`	Suffix of vertical component

Details

The S-wave pick must be stored on at least the east component and the P-wave pick (if present) must be stored on the vertical component

Value

A list containing dataframes for each of the three components with signal and header information

Examples

```
# Read in 2002.054.09.47.lhor2
pathto <- "~/mfast/sample_data/raw_data"
write_sample(pathto)
event <- "2002.054.09.47.lhor2"
triplet <- readtriplet(event,path=pathto)
```

readtvel *Read .tvel*

Description

Reads a .tvel file and saves it in an RSEIS compatible format

Usage

```
readtvel(name)
```

Arguments

name	Name and path of .tvel file

Value

RSEIS compatible dataframe containing the velocity model

Examples

```
path <- "~/mfast/velocity/ak135_taupo.tvel"
model <- readtvel(path)
write_sample("~/mfast/sample_data/raw_data")
do_station_simple(path="~/mfast/sample_data/raw_data",tvel=model)
```

reanginc *Redetermine incidence angles*

Description

Redetermine incidence angles for events in summary file

Usage

```
reanginc(summpath, tvel = ak135_taupo, overwrite = FALSE, mfast = FALSE,
  mc.cores = NULL)
```

Arguments

summpath	Path to the .summ file
tvel	Veloctity model read in by readtvel or a stored model (ak135_alp, ak135_taupo)
overwrite	Should the original summfile be overwritten?
mfast	Is the summfile from the original MFAST?
mc.cores	Number of cores to run the calculations on (defaults to maximum)

Value

A summary file with redetermined incidence angles and ray parameters

Examples

```
# Redetermine the angle of incidences for a summary file
pathto <- "~/mfast/sample_data/summ_files/WPRZ.127.CZ.summ"
nsumm <- reanginc(pathto,tvel=ak135_alp)
```

---

rms *Root mean square*

Description

Simple routine to determine root mean square value of a signal

Usage

```
rms(x)
```

Arguments

x	Vector signal

Value

RMS value

---

run_mfast *Run splitting measurement*

Description

Runs shearwave splitting measurements on a set of filtered SAC files

Usage

```
run_mfast(path, name, filtlist)
```

Arguments

path	Path to folder
name	Name of event
filtlist	A dataframe of the best filters to be used (output of writesac_filt)

Examples

```
# Run shear wave splitting measurements on event 2002.054.09.47.lhor2
pathto <- "~/mfast/sample_data/raw_data"
write_sample(pathto)
event <- "2002.054.09.47.lhor2"
triplet <- readtriplet(event,path=pathto)
bestfilt <- filter_spread(triplet)
maxfreq <- createini(pathto,triplet,bestfilt,event)
f <- writesac_filt(pathto,triplet,event,bestfilt)
run_mfast(pathto,event,f)
```

`snr` *S-wave SNR*

Description

Determine the signal to noise ratio around the S-wave pick (workhorse of filter_spread)

Usage

```
snr(E, N, s, p = -12345, dt, t_win_snr = 3, t_err = 0.05, b = 0,
  type = "normal")
```

Arguments

`E`	Vector signal of the east component
`N`	Vector signal of the north component
`s`	S-wave pick time
`p`	P-wave pick time
`dt`	Sample interval
`t_win_snr`	Window for SNR (s)
`t_err`	Modification to t_win_snr to account for error in S-pick (s)
`type`	If type is set to "verylocal" then the P-wave pick (if present) is used to set t_win_snr

Value

Signal to noise ratio around the S-wave pick

stde.weighted *Weighted standard error*

Description

A bootstrapped weighted standard error for fast polarisations

Usage

```
stde.weighted(summ, weights = c(1, 2, 3), seed = NULL, iter = 9999)
```

Arguments

summ	Dataframe containing Castelazzi graded events (CZ_*.summ)
weights	A vector containing the weights with length equal to the number of filters used (usually 3) in order with the first corresponding to F1
seed	A random number seed
iter	Number of iterations

Details

This function can also be run with a custom weight for each measurement by setting them with weights. Or, for the unweighted version, set weights=rep(1,length(summ$fast)).

Value

The circular standard error in degrees

summ.ab *Read AB*

Description

Reads in multiple AB graded .summ files

Usage

```
summ.ab(path)
```

Arguments

path	The path to the folder containing the .summ files

Value

A dataframe containing all the .summ files

summ.cz *Read cz*

Description

Reads in multiple CZ graded .summ files

Usage

```
summ.cz(path)
```

Arguments

path	The path to the folder containing the .summ files

Value

A dataframe containing all the .summ files

summ.null *Read null*

Description

Reads in multiple null graded .summ files

Usage

```
summ.null(path)
```

Arguments

path	The path to the folder containing the .summ files

Value

A dataframe containing all the .summ files

writesac_filt *Write filtered SAC files*

Description

Writes out filtered waveforms ready to have shear wave splitting measured

Usage

```
writesac_filt(path, trip, name, filtlist, number = 3, E = ".e", N = ".n",
  Z = ".z", zerophase = TRUE)
```

Arguments

path	Path to folder
trip	Event triplet (output of readtriplet)
name	Name of the event
filtlist	Dataframe of the best filters (output of filter_spread)
number	Number of best filters to use
E	Suffix of the east component
N	Suffix of the north component
Z	Suffix of the vertical component #return A dataframe of the filters that have been written

Examples

```
# Write out three best filters for event 2002.054.09.47.lhor2
pathto <- "~/mfast/sample_data/raw_data"
event <- "2002.054.09.47.lhor2"
write_sample(pathto)
triplet <- readtriplet(event)
bestfilt <- filter_spread(triplet)
f <- writesac_filt(pathto,triplet,event,bestfilt)
```

writesac_filtsmp *Simple write*

Description

Write out an event with a chosen filter

Usage

```
writesac_filtsmp(path, trip, name, low, high, E = ".e", N = ".n",
  Z = ".z", n = 1, zerophase = TRUE)
```

Arguments

path	Path to folder
trip	Event triplet (output of readtriplet)
name	Name of the event
low	Low frequency cut-off
high	High frequency cut-off
E	Suffix of the east component
N	Suffix of the north component
Z	Suffix of the vertical component
n	Number for suffix .fbn (e.g .fb2)

---

writetessa *Write TESSA .summ file*

Description

Writes out a .summ file in the format required for TESSA

Usage

```
writetessa(summ, name)
```

Arguments

summ	Dataframe containing the summary file of measurements to be run in TESSA
name	Name of the file including path and .summ suffix (defaults to current working directory)

Examples

```
# Create a .summ file for TESSA from all F1, F2 and F3 graded measurements
cz <- summ.cz("~/path/to/summfiles")
writetessa(cz,"~/TESSA/summfiles/cz.summ")
```

write_sample *Sample data*

Description

Writes out MFAST sample data

Usage

```
write_sample(path, type = "normal")
```

Arguments

path	Path to folder
type	"normal" or "verylocal" sample data

Examples

```
# Write out MFAST sample events
write_sample("~/mfast/sample_data/raw_data")

# Write out MFAST verylocal sample events
write_sample("~/mfast/sample_data/raw_data",type="verylocal")
```

Index