Package 'Rquake'

December 16, 2020

Type Package

Title Seismic Hypocenter Determination

Version 2.4-4

Date 2020-12-13

Depends R (>= 2.12)

Imports RPMG, RSEIS, GEOmap, MBA, minpack.lm

Author Jonathan M. Lees [aut, cre],
Baptiste Auguie [ctb]

Maintainer Jonathan M. Lees <jonathan.lees@unc.edu>

Description Hypocenter estimation and analysis of seismic data collected continuously, or in trigger mode. The functions organize other functions from RSEIS and GEOmap to help researchers pick, locate, and store hypocenters for detailed seismic investigation.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-12-16 13:30:02 UTC

R topics documented:

`Rquake-package` *Seismic Analysis of Earthquake Hypocenter determination*

Description

Packages puts together codes from RSEIS, GEOmap, RFOC and others for a complete analysis of hypocenters estiamted using seismic data from field campaigns.

Details

Package:	Rquake
Type:	Package
Version:	2.0-2
Date:	2012-06-12
License:	GPL

Rquake is a package for analaysis of seismic data collected continuously, or in trigger mode. The functions organize other functions from RSEIS and GEOmap to help researchers pick, locate, and store hypocenters for detailed seismic investigation.

Note

Functions CONTPF EQXYresid INITpickfile NLSlocate PFoutput RQ SavePF UPdateEQLOC XYSETUP Y2Pphase chak contPFarrivals doAmap gMAP getregionals prepPDE viewCHAC

Author(s)

Jonathan M. Lees<jonathan.lees.edu> Maintainer:Jonathan M. Lees<jonathan.lees.edu>

References

Lee, W.H.K., and S.W. Stewart, Principles and Applications of Microearthquake Networks, Academic Press, New York, 1981.

See Also

`RSEIS`

Examples

```
## Not run:
```

```
lf = list.files(path=”/Users/lees/Site/CHAC/PIX”, pattern=”.RDATA”, full.names=TRUE)

for(i in 1:length(lf))
  {
fn = lf[i]


ret = chak(DBnov , gstas, gcomps , fn, stas, kind=2, Iendian=1, BIGLONG=FALSE)

  if(ret$but == ”QUIT”) break
  }




## End(Not run)
```

ASW.vel *1D Velocity Ecuador*

Description

1D Velocity Ecuador

Usage

```
data(ASW.vel)
```

Format

a list of velocities for hypocenter relocation

Source

Mario Ruiz

Examples

```
data(ASW.vel)
data(wu_coso.vel)
data(fuj1.vel)
data(LITHOS.vel)
```

```
RSEIS::Comp1Dvels(c("ASW.vel","wu_coso.vel",  "fuj1.vel" , "LITHOS.vel"  ))
```

`BLACKJACK` *Jackknife earthquake location*

Description

Perform jackknife on earthquake location by eliminating stations.

Usage

```
BLACKJACK(Ldat, vel)
```

Arguments

`Ldat`	event list
`vel`	velocity model

Details

stations are eliminated, not rows?

Value

event list with pseudo values

Note

events are located with P and S-wave arrivals, but code here should eliminate just stations.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Iversen, E. S., and J. M. Lees (1996), A statistical technique for validating velocity models, Bull. Seismol. Soc. Am. 86(6), 1853-1862.

See Also

Vlocate,HiJACK, plotJACKLLZ

Examples

```
## Not run:
COSOjack = HiJACK(lps, sta)


plotJACKLLZ(COSOjack, sta, proj)


## End(Not run)
```

---

chak *View Picked Data*

Description

Sequentially view data that has been picked previously using GPIX or PPIX and stored as wpx files.

Usage

```
chak(DBnov, gstas, gcomps, fn, stas, vel, kind = 2, Iendian = 1, BIGLONG = FALSE)
```

Arguments

DBnov	RSEIS data base
gstas	stations to extract
gcomps	components to extract
fn	wpx file name
stas	station list
vel	1D velocity model
kind	kind of data, 0=nativeR, 1=segy, 2=sac
Iendian	endian
BIGLONG	big long or short long

Details

Program is used for detailed picking. A wpx list exists and is read into memory, then the data is extracted from the disk, plotted with swig.

Value

Pickfile

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

viewCHAC

Examples

```
## Not run:
lf = list.files(path=”/Users/lees/Site/CHAC/PIX”, pattern=”.RDATA”, full.names=TRUE)

for(i in 1:length(lf))
  {
fn = lf[i]


ret = chak(DBnov , gstas, gcomps , fn, stas, kind=2, Iendian=1, BIGLONG=FALSE)

  if(ret$but == "QUIT”) break
  }


## End(Not run)
```

`checkLOCATEinput` *Check Location data*

Description

Check to see if location data has the minimally correct list components.

Usage

```
checkLOCATEinput(Ldat, EQ, vel = NULL)
```

Arguments

`Ldat`	list, must inlude: x,y,err, sec, cor (see details)
`EQ`	list, must inlude: x,y,z, t
`vel`	list, 1D velocity structure

Details

Input pick list must have at x,y,z, sec, cor, err elements for each station.

Value

logical: FALSE mean problem with data

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

XYlocate

Examples

```
## Not run:
library(RSEIS)
data(GH)

g1 = GH$pickfile
data(VELMOD1D)
vel= VELMOD1D



 w1 = which(!is.na(g1$STAS$lat))
        sec = g1$STAS$sec[w1]

        N = length(sec)
        Ldat =    list(
          name = g1$STAS$name[w1],
          sec = g1$STAS$sec[w1],
          phase = g1$STAS$phase[w1],
          lat=g1$STAS$lat[w1],
          lon = g1$STAS$lon[w1],
          z = g1$STAS$z[w1],
          err= g1$STAS$err[w1],
          yr = rep(g1$LOC$yr , times=N),
          jd = rep(g1$LOC$jd, times=N),
          mo = rep(g1$LOC$mo, times=N),
          dom = rep(g1$LOC$dom, times=N),
          hr =rep( g1$LOC$hr, times=N),
          mi = rep(g1$LOC$mi, times=N) )

 MLAT = median(Ldat$lat)
    MLON = median(Ldat$lon)

    proj = GEOmap::setPROJ(type=2, LAT0=MLAT, LON0=MLON)

####   get station X-Y values in km
    XY = GEOmap::GLOB.XY(Ldat$lat, Ldat$lon, proj)
###   add to Ldat list
    Ldat$x = XY$x
    Ldat$y = XY$y
       wstart = which.min(Ldat$sec)



EQ = list(x=XY$x[wstart], y=XY$y[wstart], z=6, t=Ldat$sec[wstart] )

checkLOCATEinput(Ldat, EQ)

## End(Not run)
```

clusterWPX *Cluster Analysis of Picks*

Description

Given a pick file in WPX format, break the picks apart clustered accoring to single link cluster analysis.

Usage

```
clusterWPX(twpx, tol = 200, PLOT = FALSE)
```

Arguments

twpx	WPX list
tol	tolerance in seconds - all pick distances less than tol will be set to zero to force these to be associated.
PLOT	logical, if TRUE, add verbose plotting

Details

If there is not significant separation of picks, only one cluster is returned. To avoid spurious clusters, increase the tolerance.

Value

list of WPX lists

Note

Cluster depends on what one considers a cluster.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

addWPX, catWPX, checkWPX,cleanWPX,PCsaveWPX, setWPX, repairWPX

Examples

```
## Not run:
s1 = setWPX(name="HI", yr=2011, jd=231, hr=4, mi=3, sec = runif(5))

s2 = setWPX(name="HI", yr=2011, jd=231, hr=5, mi=2, sec = runif(5))


s3 = catWPX(s1,s2)

L3 = clusterWPX(s3)


## End(Not run)
```

CONTPF *Button to Contour Pickfile Arrivals*

Description

Button to Contour Pickfile Arrivals, used internally in swig.

Usage

```
CONTPF(nh, g, idev = 3)
```

Arguments

nh	RSEIS list
g	swig parameters
idev	device for plotting

Details

Driver for contPFarrivals

Value

Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

contPFarrivals

Examples

```
## Not run:

buts = "CONTPF"
swig(GH, PADDLAB=buts)


## End(Not run)
```

---

contPFarrivals *Contour Pickfile Arrivals*

Description

Contour plot of arrival times recorded in a pickfile list.

Usage

```
contPFarrivals(PF, stas, proj=NULL, cont=TRUE, POINTS=TRUE, image=FALSE ,
             col=RSEIS::tomo.colors(50), gcol="black",   phase="P", add=TRUE)
```

Arguments

`PF`	Pickfile list in RSEIS format
`stas`	station list
`proj`	projection from GEOmap
`cont`	logical, add contour to plot
`POINTS`	logical, add mark up (stations) to plot
`image`	logical, add image to plot
`col`	color palette for image
`gcol`	color for contour lines
`phase`	character, phase to contour
`add`	logical, TRUE=add to existing plot

Details

Contours the arrival time. The earliest arrival is subtracted from each time pick. Uses only the phase indicated and there can be only one phase per station - default is earliest at each station.

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

doAmap

Examples

```
## Not run:
library(RSEIS)
data(GH)
data(coso_sta_LLZ)
sta = coso_sta_LLZ
g1 = GH$pickfile


proj = GEOmap::setPROJ(type=2, LAT0 =median(sta$lat) , LON0 = median(sta$lon))


grcol =  grey(seq(from=0.3, to=0.95, length=50))
  contPFarrivals(g1, sta, proj=proj,cont=TRUE, POINTS=TRUE,
                           image=TRUE , col=grcol,     phase="P",
add=FALSE )


## End(Not run)
```

---

convertseis2R *Convert seismic data*

Description

Convert seismic data (SAC or SEGY) to native R

Usage

```
convertseis2R(fn, destdir = ".", kind = 1, Iendian = 1, BIGLONG = FALSE)
```

Arguments

`fn`	character vector of file names
`destdir`	chracter, destination directory
`kind`	kind of data (1=SEGY, 2=SAC)
`Iendian`	integer, endianess
`BIGLONG`	Logical, TRUE=long is 8 bytes

Details

Program converts a list of files to native R format for fast loading.

Value

Side effects on system

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Mine.seis

Examples

```
## Not run:

lf = list.files(path="/Users/lees/Mydata", pattern="R0*", full.names=TRUE)
convertseis2R(lf, "/Users/lees/MyRdata",  kind = 1, Iendian = 1, BIGLONG = FALSE)

## End(Not run)
```

`coso_sta_LLZ` *Coso Station File*

Description

Coso Station Location file, 1989-1999

Usage

```
data(coso_sta_LLZ)
```

Format

Name, Lat, Lon, Z

Source

Personal Files

References

Wu, H. and J. M. Lees (1996). Attenuation Structure of Coso Geothermal Area, California, from P Wave Pulse Widths, Bull. Seismol. Soc. Am., 86, 1574-1590.

Lees, J. M. (1998), Multiplet analysis at Coso Geothermal,Bull. Seismol. Soc. Am. 88(5) 1127-1143.

`defaultVEL` *Default Velocity Function*

Description

Default Velocity Function is returned in the event no velocity function is available.

Usage

```
defaultVEL(kind = 1)
```

Arguments

`kind`	integer, 1=fuj1, 2=LITHOS

Details

A set of default velocity functions are available.

Value

velocity list, P and S waves

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

fuj1.vel

Examples

```
v = defaultVEL(1)
```

DistWeight *Distance wheighting*

Description

Distance weighting for non-linear earthquake location.

Usage

```
DistWeight(dist, err, distwt)
DistWeightLL(lat, lon, elat, elon, err, distwt)
DistWeightXY(x, y, ex, ey, err, distwt)
```

Arguments

`dist`	distance in km
`err`	sigma error in seconds
`distwt`	distance weighting parameter
`lat`	Latitude
`lon`	Longitude
`elat`	Event Latitude
`elon`	Event Longitude
`x`	station X(km)
`y`	station Y(km)
`ex`	event X (km)
`ey`	event Y (km)

Details

Based on Lquake scheme from University of Washington. If you need to reduce the effect of distance weighting, increase distwt.

Since the hypocenter moves between each iteration, the distance weighting is updated.

Value

vector of weights

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
DistWeight(1:10, .4, 20)
```

---

doAmap *Plot a map of station locations*

Description

Plot a map of station locations

Usage

```
doAmap(stas, doproj = TRUE)
```

Arguments

`stas`	station list
`doproj`	logical, if TRUE, project (UTM) the data so plot is in units of km with the median lat-lon as the center. If FALSE, use the lat-lon coordinates.

Details

The range of the plot is expanded by 10 percent prior to plotting.

Value

list, GEOmap projection

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

gMAP,expandbound,GLOB.XY

Examples

```
## Not run:
  fsta = "/Users/lees/Site/CHAC/staLLZ.txt"
stas = scan(file=fsta,what=list(name="", lat=0, lon=0, z=0))
stas$z = stas$z/1000

doAmap(stas, doproj = TRUE)


## End(Not run)
```

DoRLocate *Locate a set of picks in native R format*

Description

This is a script to apply Vlocate. After picking arrival times on several earthquake events and saving them with swig, the saved files can be located sequentially with this wrapper function.

Usage

```
DoRLocate(LF, stas, vel,  params=list(distwt = 100,
          lambdareg=20,
          REG = TRUE,
          WTS = TRUE,
          STOPPING = TRUE,
          tolx = 0.005,
          toly = 0.005,
          tolz = 0.01,  RESMAX = c(4,5),
 maxITER = c(7, 5, 7, 4)
))

DoUWLocate(LF, stas, vel,  params=list(distwt = 100,
          lambdareg=20,
          REG = TRUE,
          WTS = TRUE,
          STOPPING = TRUE,
          tolx = 0.005,
          toly = 0.005,
          tolz = 0.01))
```

Arguments

`LF`	List of file location to read (output of list.files)
`stas`	list, station location: name, lat, lon, z (and correction if available)
`vel`	list, velocity structure
`params`	list, parameters for Vlocate function

Details

Use swig and viewCHAC to pick P and S-wave arrivals, mostly via the PickWin button. After an event has been saved to disk in a native R format (suffix RDATA), these can be loaded and located.

The UW version is for files already picked and in the ascii-text UW-pickfile format.

Value

list of earthquake location and error ellipsoids.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Vlocate

Examples

```
## Not run:

##########  read in list of pick data
LF = list.files(path="./Detail_picks", pat="RDATA", full=TRUE)

###  read in station location file
fsta = "/home/lees/Site/CHAC/staLLZ.txt"

stas = scan(file=fsta,what=list(name="", lat=0, lon=0, z=0))
stas$z = stas$z/1000

############  set the velocity (this vel is for a geothermal field in Califormia)
data(VELMOD1D)
vel= VELMOD1D


#############   calculate the locations: (use default values)
KAM = DoRLocate(LF, stas, vel)


############  Done with earthquake locations....next pull data out of list

N = length(KAM)
```

```
H = list(lat=vector(length=N), lon=vector(length=N),
  z=vector(length=N), date=vector(length=N) ,   gap=vector(length=N),
herr =vector(length=N),
  zerr=vector(length=N),
  qual=vector(length=N))



for(i in 1:length(KAM))
  {
    zip = KAM[[i]]

    H$lat[i] = zip$EQ$lat
    H$lon[i] = zip$EQ$lon
    H$z[i] = zip$EQ$z
    H$date[i]  =     dateStamp(zip$EQ$Time)

    H$gap[i] = zip$ERR$gap
    H$herr[i] = zip$ERR$herr
    H$zerr[i] = zip$ERR$sterrz
    H$qual[i] = paste(zip$ERR$Q1, zip$ERR$Q2, sep="")

  }


data.frame(H)


MLAT = median(stas$lat)
MLON = median(stas$lon)

proj = GEOmap::setPROJ(type=2, LAT0=MLAT, LON0=MLON)

staxy = GEOmap::GLOB.XY(stas$lat, stas$lon, proj)
zq = GEOmap::GLOB.XY(H$lat, H$lon, proj)

plot(c(staxy$x, zq$x) , c(staxy$y, zq$y), type='n', xlab="E, km",
ylab="N, km", asp=1)

points(staxy, pch=6, col='red')

points(zq, pch=8, col='blue')

XYerror.bars(zq$x, zq$y, zq$y-H$herr/2, zq$y+H$herr/2,  zq$x-H$herr/2, zq$x+H$herr/2,
             pch = 1, col =1, barw = 0.05, add = TRUE )

#############  or: plot 95 percent confidence bounds


for(i in 1:length(KAM))
{
  zip = KAM[[i]]
 KOV = zip$ERR$cov[2:4, 2:4]

     eqlipse(zq$x[i], zq$y[i] , KOV,   wcols = c(1,2) , dof=zip$ERR$ndf, border="blue"  )



}

######################################
######################################   UW format data
######################################


setwd("/home/lees/Progs/R_stuff/EARTHQUAKE")


stafile =  "coso_sta.LLZ"

staf = stafile
stas = setstas(stafile )



pdir = "/home/lees/Progs/R_stuff/EARTHQUAKE/pfiles"
LF = list.files(path=pdir, pattern="p$", full.names=TRUE )

KAM = DoUWLocate(LF, stas, vel)





## End(Not run)
```

`eqlipse` *Error Elipse for Hypocenter Location*

Description

Error Elipse for Hypocenter Location

Usage

```
eqlipse(x, y, cov, wcols = c(1, 2), dof = 2, pct=0.05, ...)
```

Arguments

x	X-location for drawing
y	Y-location for drawing
cov	matrix, 3 by 3 Covariance matrix
wcols	vector, which columns to extract from cov, see details.
dof	Degrees of Freedom for 95 percent confidence
pct	Percent used for 2-sided confidence bounds, default=0.05
...	graphical parameters, par

Details

The 3 by 3 matrix is supplied and a 2 by 2 matrix is subtracted depending on which components are being drawn. For X-Y projections, use wcols=c(1,2). For vertical cross sections, rotate the cov matrix and then extract the columns.

Value

Side effects, graphical

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

eqwrapup

Examples

```
## Not run:
library(RSEIS)
data(GH)
data(wu_coso.vel)
vel = wu_coso.vel


gpf = GH$pickfile

w1 = which(gpf$STAS$phase=="P" | gpf$STAS$phase=="S" )

N = length(w1)

 Ldat =    list(
      name = gpf$STAS$name[w1],
      sec = gpf$STAS$sec[w1],
      phase = gpf$STAS$phase[w1],
      lat=gpf$STAS$lat[w1],
      lon = gpf$STAS$lon[w1],
      z = gpf$STAS$z[w1],
          err= gpf$STAS$err[w1],
          yr = rep(gpf$LOC$yr , times=N),
          jd = rep(gpf$LOC$jd, times=N),
          mo = rep(gpf$LOC$mo, times=N),
          dom = rep(gpf$LOC$dom, times=N),
          hr =rep( gpf$LOC$hr, times=N),
          mi = rep(gpf$LOC$mi, times=N) )

EQ = GH$pickfile$LOC

EQ$t = EQ$sec

kuality = eqwrapup(Ldat, EQ, vel, distwt = 20, verbose = TRUE )


 MLAT = median(Ldat$lat)
  MLON = median(Ldat$lon)
  proj = GEOmap::setPROJ(type=2, LAT0=MLAT, LON0=MLON)

  XYSTAS = GEOmap::GLOB.XY(Ldat$lat,  Ldat$lon , proj)


 eqxy = GEOmap::GLOB.XY(EQ$lat, EQ$lon, proj)


plot(range(c(XYSTAS$x, eqxy$x)), range(c(XYSTAS$y, eqxy$y)), type='n', asp=1, xlab="km", ylab="km" )
points(XYSTAS$x, XYSTAS$y, pch=6)
points(eqxy$x, eqxy$y, pch=8, col='red')


KOV = kuality$cov[2:4, 2:4]

eqlipse(eqxy$x, eqxy$y , KOV,   wcols = c(1,2) , dof=kuality$ndf, border="blue"  )


## End(Not run)
```

---

eqwrapup *Earthquake Wrap Uo*

Description

Calculate error and summary information on earthquake location.

Usage

```
eqwrapup(Ldat, EQ, vel, distwt=20, lambdareg = 0.0, verbose=FALSE)
```

Arguments

`Ldat`	List of station arrival times, lat-lon, and uncertainty
`EQ`	List of earthquake location: Lat-Lon-z-t
`vel`	velocity model
`distwt`	distance weight, default=20
`lambdareg`	numeric, regularization parameter (default=0)
`verbose`	logical, TRUE=print information to screen

Details

Earthquakes are located with a generalized inverse (SVD). covariance matrix is extracted and 95% confidence bounds are calculated. Quality factors Q1 and Q1 estimate the quality iof the location based on the gap, minimum distance and rms.

Value

List

`rms`	Root Mean Square Residual
`meanres`	Mean Residual
`sdres`	Standard Dev of residuals
`sdmean`	Standard error of mean residual
`sswres`	Sum squared weighted residuals
`ndf`	Number of Degrees of Freedom
`sterrx`	km, error in X (East-West)
`sterry`	km, error in Y (North-South)
`sterrz`	km, error in Z, (depth)
`sterrt`	s, Delta-time
`cov`	covariance matrix (used for error ellipsoids)
`lam`	lambda
`gap`	Spatial gap (max subtended angle)
`herr`	Horizontal error
`distmin`	Minimum distance to epicenter
`Q1`	Quality Factor based on Gap and RMS
`Q2`	Quality factor based on RMS, depth and min-Distance

Note

The Damping parameter (lambda) is set to zero. In the UW lquake program, lambda is set to 0.02.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Klocate, Glocate, getGAP

Examples

```
## Not run:
library(RSEIS)
data(GH)
data(wu_coso.vel)
vel = wu_coso.vel


gpf = GH$pickfile

w1 = which(gpf$STAS$phase=="P" | gpf$STAS$phase=="S" )

N = length(w1)

 Ldat =    list(
      name = gpf$STAS$name[w1],
      sec = gpf$STAS$sec[w1],
      phase = gpf$STAS$phase[w1],
      lat=gpf$STAS$lat[w1],
      lon = gpf$STAS$lon[w1],
      z = gpf$STAS$z[w1],
      err= gpf$STAS$err[w1],
      yr = rep(gpf$LOC$yr , times=N),
      jd = rep(gpf$LOC$jd, times=N),
      mo = rep(gpf$LOC$mo, times=N),
      dom = rep(gpf$LOC$dom, times=N),
      hr =rep( gpf$LOC$hr, times=N),
      mi = rep(gpf$LOC$mi, times=N) )

EQ = GH$pickfile$LOC

EQ$t = EQ$sec

kuality = eqwrapup(Ldat, EQ, vel, distwt = 20, verbose = TRUE )

names(kuality)

## End(Not run)
```

EQXYresid *Calculate Residuals*

Description

given an earthquake hypocenter and a list of station information, retrieve the station residuals.

Usage

```
EQXYresid(XY, vel = list(), h1 = c(0, 0, 0, 0), PLOT = FALSE)
```

Arguments

`XY`	matrix of station location and arrival times.
`vel`	list, RSEIS velocity model
`h1`	hypocenter location, c(x,y,z,t)
`PLOT`	logical, TRUE=plot the residuals

Details

The XY mtrix is in cartesian coordinates, i.e. it has been projected into units of km. Only 1D velocity models are used at this time. Only residuals of P and S wave arrivals are estimated.

Value

vector, right hand side of the least squares problem.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

travel.time1D,UPdateEQLOC

Examples

```
## Not run:
 XY = GEOmap::GLOB.XY(pstas$lat, pstas$lon, proj)
    elcor = rep(0, length(pstas$lat))

    DZ = pstas$z - mean(stas$z)

    elcor[pstas$phase==”P”] = DZ[pstas$phase==”P”]/v$vp[1]
    elcor[pstas$phase==”S”] = DZ[pstas$phase==”S”]/v$vs[1]

    XY$cor = elcor
    XY$phase = pstas$phase
    XY$sec = pstas$sec

    eqXY = GEOmap::GLOB.XY(sol[1], sol[2], proj)
    res =  EQXYresid(XY, vel=v , h1=c(eqXY$x, eqXY$y, sol[2],sol[4]) , PLOT=FALSE)
```

```
## End(Not run)
```

euler_passive *Euler Rotation Angles*

Description

Given three angles return rotation matrix.

Usage

```
euler_passive(phi, theta, psi)
```

Arguments

phi	angle with x-axis
theta	angle with y-axis
psi	angle with z-axis

Details

Code borrowed from cpp code in package cda. used in rgl.ellipsoid.

Value

3 by 3 rotation matrix.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>, Baptiste Auguie<baptiste.auguie@gmail.com>

See Also

rgl.ellipsoid

Examples

```
options(rgl.useNULL = TRUE)
phi=30*pi/180 ; theta= 20*pi/180; psi = 6*pi/180
rr = euler_passive(phi,theta,psi)
```

getEulers *Get Eulers Angles*

Description

Given a covariance matrix calculated with Vlocate, extract euler's angles for plotting in rgl

Usage

```
getEulers(R)
```

Arguments

R	covarince matrix

Details

Extract the euler angles for plotting an ellipsoid. psi about X-axis, theta about Y axis, phi about Z-axis.

Value

vector, phi theta psi

Note

Used in conjunction with ROTcovQUAKE

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

ROTcovQUAKE

Examples

```
options(rgl.useNULL = TRUE)
R = matrix( runif(9), ncol=3)

getEulers(R)
```

getGAP *Get Seismic Gap*

Description

Given an earthquake and a set of stations, return the maximum angle subtended between adjacent stations relative to the epicenter.

Usage

```
getGAP(EQ, Ldat, PLOT = FALSE)
```

Arguments

EQ	List, Earthequake location, elements (lat, lon) must be present
Ldat	List, station information, (lat, lon) must be present
PLOT	logical, plot the stations and show the gap

Details

Theangles are calculated in cartesian coordinates with the epicenter at the origin using a UTM projection.

Value

numeric, gap in degrees

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

eqwrapup

Examples

```
set.seed(0)

N = 10
snames = paste(sep="", "A", as.character(1:N))
stas = list(name=snames, lat=runif(N, 35.9823, 36.1414), lon=runif(N, -118.0031, -117.6213))

NEQ = 3
WEQ = list(lat=runif(NEQ, 35.9823, 36.1414), lon=runif(NEQ, -118.0031, -117.6213))
```

```
 MLAT = median(stas$lat)
  MLON = median(stas$lon)
  proj = GEOmap::setPROJ(type=2, LAT0=MLAT, LON0=MLON)

  XYSTAS = GEOmap::GLOB.XY(stas$lat,  stas$lon , proj)
  eqxy = GEOmap::GLOB.XY(WEQ$lat, WEQ$lon, proj)


plot(range(c(XYSTAS$x, eqxy$x)), range(c(XYSTAS$y, eqxy$y)), type='n', asp=1, xlab="km", ylab="km" )
points(XYSTAS$x, XYSTAS$y, pch=6)

for(i in 1:NEQ)
{
EQ = list(lat=WEQ$lat[i], lon=WEQ$lon[i])


g = getGAP(EQ, stas, PLOT=FALSE)

points(eqxy$x[i], eqxy$y[i], pch=8, col='red')
text(eqxy$x[i], eqxy$y[i], labels=paste("gap=", format(g)), pos=3)

}
```

GETpsTT *Get Pand S travel times and derivatives*

Description

Get Pand S travel times and derivatives

Usage

```
GETpsTT(phase, eqz = 6, staz = 0, delx = 1, dely = 1, deltadis = 6, vel)
```

Arguments

phase	character vector, phase
eqz	event depth
staz	station elevation
delx	km, delta X
dely	km, delta Y
deltadis	km, distance
vel	velocity models (P and S)

Details

Creates a vector of travel times, and a matrix and derivatives used for inversion.

Value

list:

`TT`	travel time vector
`Derivs`	matrix of derivatives, dtdx, dtdy, dtdz

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

many.time1D

Examples

```

## Not run:
library(RSEIS)
library(GEOmap)


data(GH)



data(VELMOD1D)
vel = VELMOD1D


p1 = GH$pickfile$STAS


loc = GH$pickfile$LOC


proj  = GEOmap::setPROJ(type = 2, LAT0 =loc$lat, LON0 =  loc$lon)


XYsta = GEOmap::GLOB.XY(p1$lat, p1$lon, proj)
XYq =   GEOmap::GLOB.XY(loc$lat, loc$lon, proj)

delx = XYq$x-XYsta$x
dely = XYq$y-XYsta$y
dists = sqrt(delx^2+dely^2)

G1 = GETpsTT(p1$phase, eqz=loc$z, staz=0, delx=delx, dely=dely,  deltadis=dists , vel)
```

```
## End(Not run)
```

getregionals *Extract regional events*

Description

Extract regional events from a hypocenter list (catalog)

Usage

```
getregionals(KAT, Mlat, Mlon, rad = 1000, t1 = 1, t2 = 2)
```

Arguments

KAT	catalog list
Mlat	central latitude
Mlon	central longitude
rad	radius (km)
t1	start time (julian days)
t2	end time (julian days)

Details

given an earthquake catalog from PDEs, for example, extract the events that are close to a network in a given time frame. The limited data set may be used to help predict arrival times for known hypocenter locations.

Value

Catalog

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Mine.seis

Examples

```
## Not run:
fpde = '/Users/lees/Site/CHAC/pde_catalog_NOV_2011.txt'

ipde = prepPDE(fpde)
Mlat = median(stas$lat)
Mlon = median(stas$lon)

Lt1 =  JtimL(upf$LOC)
t1 = Lt1-3/(24*6)
t2 = Lt1+3/(24*6)

localeqs = getregionals(KAT, Mlat, Mlon, rad=100 ,  t1=NULL, t2=NULL)

for(i in 1:length(localeqs))
  {
j = localeqs[i]
 at1 = KAT$jsec[j]
        at2 = at1+20/(24*60)
 GH = Mine.seis(at1, at2, DBnov , gstas, gcomps, kind=2, Iendian=1, BIGLONG=FALSE)

hord = which(GH$COMPS==”V”)

        gret = swig(GH, sel=hord, PADDLAB=buts)
####  save any picks to a file on disc for later use


  }



## End(Not run)
```

getresidTT *Travel time residuals*

Description

Given an earthquake location and a set of arrival times, return a vector of residuals.

Usage

```
getresidTT(Ldat, EQ, stas, vel)
```

Arguments

`Ldat`	List of arrival times
`EQ`	List of event location, (lat, lon, z, and time)
`stas`	station location list
`vel`	list, velocity structure

Details

1D travel time calculation.

Value

vector of residuals

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

travel.time1D

Examples

```
## Not run:

#########  LF is a vector of arrival time files
#####  KAM is a set of locations

 load(LF[i])

    zip = LeftjustTime(twpx)
    EQ = KAM[[i]]$EQ
    ERR = KAM[[i]]$ERR

    LOC = list(yr=EQ$Time$yr, mo=EQ$Time$mo, dom=EQ$Time$dom, hr=EQ$Time$hr,
       mi=EQ$Time$mi, sec=EQ$Time$sec, jd=EQ$Time$jd, lat=EQ$lat, lon=EQ$lon,
       z=EQ$z, mag=NA,gap=ERR$gap,delta=ERR$distmin,rms=ERR$rms,hozerr=ERR$herr   )

 resids = getresidTT(zip,LOC, stas , vel)


## End(Not run)
```

Gfirstguess *First gues from a pick file*

Description

Extract the lat lon from the pick file.

Usage

```
Gfirstguess(Ldat, type = "first")
```

Arguments

`Ldat`	
`type`	one of "first", "mean", or "median"

Details

Either the earliest arrival or the average station is returned. Used internally in the earthquake location program to provide a first guess.

Value

vector, lat, lon, z and tee

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Klocate

Examples

```
## Not run:

twpx  = latlonz2wpx(twpx, stas)

g1 = Gfirstguess(twpx, type = "first")



## End(Not run)
```

gMAP *Generic Map Button*

Description

Generic Map Button

Usage

```
gMAP(nh, g, idev = 3)
```

Arguments

nh	RSEIS structure
g	parameters used in swig
idev	device for plotting (not used)

Details

This is a button used internally in swig

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

swig

Examples

```
## Not run:

buts = "gMAP"
swig(GH, PADDLAB = buts )



## End(Not run)
```

GPIX *PICK Buttons for swig*

Description

defining functions for swig

Usage

```
GPIX(nh, g)
```

Arguments

nh	waveform list for RSEIS
g	plotting parameter list for interactive program

Details

Buttons can be defined on the fly.

GPIX Multiple picks on a panel

Value

The return value depends on the nature of the function as it is returned to the main code swig. Choices for returning to swig are: break, replot, revert, replace, donothing, exit.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

swig, XTR

Examples

```
## Not run:


STDLAB=c("DONE", "QUIT", "SELBUT" , "GPIX" )
data(GH)
JJ = swig(GH, sel=1:10, STDLAB=STDLAB)


## End(Not run)
```

HiJACK *Jackknife a list of events*

Description

Jackknife a list of events

Usage

```
HiJACK(lps, sta, vel)
```

Arguments

`lps`	vector of filenames with UW events
`sta`	staiton list
`vel`	velocity list

Details

Driver for BLACKJACK

Value

jackknife pseudovalues for each event

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Iversen, E. S., and J. M. Lees (1996), A statistical technique for validating velocity models, Bull. Seismol. Soc. Am. 86(6), 1853-1862.

See Also

BLACKJACK

Examples

```


## Not run:
COSOjack = HiJACK(lps, sta)


plotJACKLLZ(COSOjack, sta, proj)
```

```
## End(Not run)
```

imageINFLUENCE *Image Influence of stations*

Description

Plot contours/image of Influence scores.

Usage

```
imageINFLUENCE(B, sta, proj)
```

Arguments

B	Pseudovalue list
sta	station location list
proj	projection list

Details

Following jackknife - plot results. this function is called by plotJACKLLZ.

Value

side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Iversen, E. S., and J. M. Lees (1996), A statistical technique for validating velocity models, Bull. Seismol. Soc. Am. 86(6), 1853-1862.

See Also

plotJACKLLZ

`INITpickfile` *Initialize a pickfile*

Description

Initialize a pickfile

Usage

```
INITpickfile(stas = NULL, src = NULL, WPX = NULL)
```

Arguments

`stas`	station list
`src`	hypocenter location
`WPX`	GPIX or PPIX picks from swig

Details

Initialize a pickfile with a set of picks extracted from swig.

Value

list, pickfile

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

EmptyPickfile

Examples

```
## Not run:
PF =  INITpickfile(stas=stas, src=NULL, WPX=twpx)


## End(Not run)
```

Klocate *Earthquake Hypocenter Location*

Description

Earthquake Hypocenter Location

Usage

```
Klocate(Ldat, sol = c(0, 0, 0, 0), vel=defaultVEL(6),
distwt = 20, errtol = c(0.01, 0.01, 0.01), maxit = 20,
Lambda = 1, guessdepth = 6, APLOT = FALSE,
stas = list(name = "", lat = NA, lon = NA, z = NA))
```

Arguments

`Ldat`	swig pick list
`sol`	vector, initial solution
`vel`	velocity list
`distwt`	distance weight parameter
`errtol`	error tolerance
`maxit`	Maximum number of iterations
`Lambda`	damping parameter
`guessdepth`	initial depth for guess
`APLOT`	logical, plot intermediate solutions
`stas`	station list

Details

Inversion is done with SVD.

Value

Event location in Lat-Lon-Z-T.

Note

Damped least squares.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

swig, defaultVEL

Examples

```
## Not run:

LF = list.files(path=pdir, pattern=”p$”, full.names=TRUE )

GYPSY2 = vector(mode=”list”)

for(i in 1:length(LF))
  {
    g1 = getpfile( LF[i], sta=staf )
    ##  points(g1$H$lon, g1$H$lat, pch=8, col='red')

    w1 = which(!is.na(g1$STAS$lat))
    sec = g1$STAS$sec[w1]

    N = length(sec)
    Ldat =    list(
      name = g1$STAS$name[w1],
      sec = g1$STAS$sec[w1],
      phase = g1$STAS$phase[w1],
      lat=g1$STAS$lat[w1],
      lon = g1$STAS$lon[w1],
      z = g1$STAS$z[w1],
      err= g1$STAS$err[w1],
      yr = rep(g1$LOC$yr , times=N),
      jd = rep(g1$LOC$jd, times=N),
      mo = rep(g1$LOC$mo, times=N),
      dom = rep(g1$LOC$dom, times=N),
      hr =rep( g1$LOC$hr, times=N),
      mi = rep(g1$LOC$mi, times=N) )


    NEW = Klocate(Ldat, sol=c(MYLOC$y, MYLOC$x, 6, 0) )

    GYPSY2[[i]] = NEW


  }



## End(Not run)
```

lastPIX *Last Pix*

Description

RSEIS Button: Restore Last WPX file from memory. Function is used internally in swig.

Usage

```
lastPIX(nh, g)
editPIX(nh, g)
```

Arguments

nh	GH list from RSEIS
g	parameters from swig

Value

New WPX list attached to g

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

---

latlonz2wpx *Add Lat-Lon-Z to WPX list*

Description

Given an existing list of seismic picks, add Latitude, Longitude and Elevation associated with the indicated station.

Usage

```
latlonz2wpx(twpx, stas)
```

Arguments

twpx	List of picks from swig
stas	station list

Details

The names of the stations are matched to the station names int he station file.

Value

Pick file with LLZ added as list members.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Klocate

Examples

```
## Not run:

twpx  = latlonz2wpx(twpx, stas)


## End(Not run)
```

`LDATlist` *List location data*

Description

List location data

Usage

```
LDATlist(g1, w1)
```

Arguments

`g1`	loc list
`w1`	index

Value

side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

LeftjustTime *Adjust times relative to least minute.*

Description

Adjust times relative to least minute.

Usage

```
LeftjustTime(g1)
```

Arguments

g1	list with times, yr, jd, hr, mi, sec

Details

Reutrns the list with the times adjusted to the least minimum (left adjusted)

Value

list is returned.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

recdate

Examples

```
set.seed(0)

d1  = list(yr=rep(2005, 4), jd=rep(5, 4), hr=rep(6, 4), mi=c(1,1,2,3), sec=runif(4, 0, 60))
LeftjustTime(d1)
```

legitWPX *Legitimate Pix*

Description

Check WPX list for legitimate picks

Usage

```
legitWPX(twpx, quiet=TRUE)
```

Arguments

twpx	
quiet	logical, default=TRUE, FALSE generates an error message

Details

Used internall to test if a WPX list has legitimate picks. Initially a list is generated with NA and 0 values in the place holders. If no legitimate picks are added, the list still exists, but the picks are bogus, so this routine will return 0.

Value

integer: 0=not legitimate, 1=legitimate

Note

Currently only the name is tested for all(NA), but this might be changed int he future for a more sophisticated test.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

PCsaveWPX

Examples

```
###  test fails
## Not run:
library(RSEIS)
jk = RSEIS::cleanWPX()
legitWPX(jk)

data(GH)
```

```
gwpx = RSEIS::uwpfile2ypx(GH$pickfile)

legitWPX(gwpx)

## End(Not run)
```

`MeanStaDist` *Mean Station Distance*

Description

calculate the mean km distance of a set of Lat-lon pairs

Usage

```
MeanStaDist(Ldat)
```

Arguments

`Ldat`	station list with elements of Lat-Lon

Details

Given a list with elements named lat and lon, find the mean station distance.

Value

scalar

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

setPROJ, GLOB.XY, dist

Examples

```
## Not run:
library(RSEIS)
data(GH)
MeanStaDist(GH$pickfile$STAS)

## End(Not run)
```

NLSlocate *Nonlinear Least Squares Location*

Description

Nonlinear Least Squares Location using Gieger's method

Usage

```
NLSlocate(GH, vel = list(), init = c(0, 0, 0, 0), PLOT = FALSE)
```

Arguments

GH	List, RSEIS
vel	velocity model
init	initial guess for event location
PLOT	logical, TRUE=plot

Details

This is an adaptation of non-linear least squares inversion for earthquake location. A residual function is supplied, and iterations are performed until the location is determined.

Value

vector, new location

Note

At this stage there are no weighting mechanisms or code to eliminate data that has residuals that are too large.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Lee, W.H.K., and S.W. Stewart, Principles and Applications of Microearthquake Networks, Academic Press, New York, 1981.

See Also

swig

Examples

```
## Not run:

eqsol = NLSlocate(nh, vel=nh$vel,  PLOT=TRUE )



## End(Not run)
```

`OnePerSta` *One Phase Pick Per Station*

Description

Require only one pick per station of a specified phase.

Usage

```
OnePerSta(twpx, phase = "Y")
```

Arguments

`twpx`	WPX list
`phase`	character, specific phase

Details

This is used to reduce the number of picks for specific station and phase. The purpose is avoid multiple P-wave phases for each station in the earthquake location routines.

Value

WPX list

Note

For S-waves there may be multiple S-wave arrivals, as in the case for shear wave splitting. In that case it is probably best to name the phases differently, as in S1, S2, for example.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

cleanWPX, repairWPX

Examples

```
s1 = RSEIS::setWPX(name="HI", phase="P", yr=2011, jd=231, hr=4, mi=3, sec = runif(5))
s2 = RSEIS::setWPX(name="BYE", phase="P", yr=2011, jd=231, hr=4, mi=3, sec = runif(5))

s3 = RSEIS::catWPX(s1, s2)

s4 = OnePerSta(s3, phase = "P")
```

PCfiledatetime *Create a character string from a date*

Description

Create a character string from a date for naming unique output files.

Usage

```
PCfiledatetime(orgtim, tims)
```

Arguments

orgtim	time vector of length 5: c(yr, jd, hr, mi, sec)
tims	seconds to add to orgtim

Value

filename	character string

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
## Not run:
library(RSEIS)
data(GH)

g1 = getGHtime(GH)
g2 = unlist(g1)

PCfiledatetime(g2, 1)
```

```
## End(Not run)
```

PCsaveWPX *Save WPX list*

Description

Save a WPX list to a file on the local file system.

Usage

```
PCsaveWPX(twpx, destdir = ".")
```

Arguments

twpx	WPX list
destdir	character, destination directory, default=getwd()

Details

Creates a file with the list as in native binary format. This file can be loaded with the standard load function in R. The name of the file is created by using the minimum time extracted from the WPX list. The suffix on the file name is RDATA. When reading in, the object created is named "twpx" for further processing.

Value

Side effects on file system. The name of the output file is returned.

Note

User must have write access to the destination directory.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

addWPX, catWPX, checkWPX, cleanWPX, clusterWPX, repairWPX, setWPX

Examples

```
## Not run:
s1 = setWPX(name="HI", yr=2011, jd=231, hr=4, mi=3, sec = runif(5))
hh = PCsaveWPX(s1)

###   read in the data
load(hh)

data.frame(twpx)

## End(Not run)
```

PFoutput *Write a pickfile to disk*

Description

Write a pickfile to disk ina variety of formats.

Usage

```
PFoutput(PF, stas = NULL, sol = NULL, format = 0)
```

Arguments

PF	Pickfile list from RSEIS
stas	station list
sol	solution vector, (lat, lon, z, t0)
format	integer, 0=all formats, 1=native R, 2=UW, 3=csv)

Details

Writes files to disk in local directory.

Value

Side effects

Note

Creates a file name and writes to disk in a variety of formats.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

SavePF, RSEIS

Examples

```
## Not run:

PFoutput(nh$PF, stas = g$sta, sol = NULL, format = 0)


## End(Not run)
```

Pick3 *PICK Buttons for swig*

Description

Picking functions for swig

Usage

```
Pick3(nh, g)
```

Arguments

nh	waveform list for RSEIS
g	plotting parameter list for interactive program

Details

Buttons can be defined on the fly.

Pick3 Multiple picks on a panel

Value

The return value depends on the nature of the function as it is returned to the main code swig. Choices for returning to swig are: break, replot, revert, replace, donothing, exit.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

swig, PickWin

Examples

```
## Not run:
library(RSEIS)
MYFUNC<-function(nh, g)
  {
    print("pressed MYFUNC")
    d = data.frame(list(stations=nh$STNS, components=nh$COMPS))
print(d)
    g$action = "replot"
    invisible(list(global.vars=g))
  }

STDLAB=c("DONE", "QUIT", "SELBUT" , "MYFUNC" )
data(GH)
JJ = swig(GH, sel=1:10, STDLAB=STDLAB)


## End(Not run)
```

plotEQ *Plot Earthquake location*

Description

Plot Earthquake location

Usage

```
plotEQ(Ldat, AQ, add = FALSE, prep = FALSE,
TIT = "UTM Projected Stations", proj = NULL,
 xlim = NULL, ylim = NULL)
```

Arguments

Ldat	Data list
AQ	Earthquake solution (location)
add	logical, TRUE=add to plot
prep	preparation
TIT	title
proj	projection list
xlim	2-vector, x limits (km)
ylim	2-vector, y limits (km)

Details

used internally in RElocateEQ

Value

graphical side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

RElocateEQ

plotJACKLLZ *BoxPlot Jackknife of station locations*

Description

BoxPlot Jackknife of station locations

Usage

```
plotJACKLLZ(hjack, sta, proj = NULL,  PLOT=0, PS=FALSE,
    fbase="jack", width =c(10, 5) , height = c(8, 8))
```

Arguments

hjack	Output of hijack
sta	station location list
proj	projection list
PLOT	plotting flag, 0,1,2. If PLOT=0 plot both boxplots and map, if plot =1 plot only boxplot, if plot=2 plot only map. Default=0
PS	postscript output device, default=FALSE
fbase	basename for output png files
width	2-vector of width of plots output (inches)
height	2-vector of height of plots output (inches)

Details

takes the output of the HiJack function and extracts the pseudovalues and influence information for boxplots.

Value

Graphical side effects and

`X`	influence of lon
`Y`	influence of lat
`Z`	influence of depth
`files`	character vector of the names of the files output

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Iversen, E. S., and J. M. Lees (1996), A statistical technique for validating velocity models, Bull. Seismol. Soc. Am. 86(6), 1853-1862.

See Also

HiJACK, BLACKJACK,imageINFLUENCE

Examples

```
## Not run:

COSOjack = HiJACK(lps, sta)

plotJACKLLZ(COSOjack, sta, proj)




## End(Not run)
```

PostREQquake *Post Processing on EQrquake*

Description

Post Processing on EQrquake

Usage

```
PostREQquake(XQ, proj)
```

Arguments

XQ	List of Earthquakes
proj	projection list

Details

Following event locations, plot.

Value

graphical side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

PostVquake *Plotting error ellipsoids of many events*

Description

Plotting error ellipsoids of many events

Usage

```
PostVquake(MANYeq, GX, GY, XY, proj, add=FALSE, ...)
```

Arguments

`MANYeq`	List of earthquakes following Vlocate
`GX`	X-bounds for plot
`GY`	Y-bounds for plot
`XY`	station locations in km
`proj`	projection list
`add`	logical; if TRUE, add to existing plot (DEFAULT=FALSE)
`...`	graphical parameters for plotting (see par)

Details

Plots the event and the error ellipsoids

Value

Graphical side effects

Note

This is used to plot many event locations and their error ellipsoids

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

eqlipse

`prepPDE` *Read PDE file*

Description

Read PDE file output and prepare for use in Rquake. Creates an earthquake catalog.

Usage

```
prepPDE(fn)
```

Arguments

`fn`	file name and path

Details

Read in a PDE file that has been saved from the website, http://earthquake.usgs.gov/earthquakes/eqarchives/epic/epic_global.

Value

List (catalog) of earthquake hypocenters

Note

Take care if they change the format.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

getregionals

Examples

```
## Not run:
fpde = '/Users/lees/Site/CHAC/pde_catalog_NOV_2011.txt'
ipde = prepPDE(fpde)

## End(Not run)
```

`Qrangedatetime` *Range of Date Time*

Description

Return the range of dates and times for any list with a date/time list

Usage

```
Qrangedatetime(D)
```

Arguments

D	info list from RSEIS seismic data list

Value

min	date time list
max	date time list

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
## Not run:
library(RSEIS)
data(GH)

Qrangedatetime(GH$info)


## End(Not run)
```

RElocateEQ *Relocate earthquakes stored in UW format*

Description

Relocate earthquakes stored in UW format

Usage

```
RElocateEQ(lps, sta, vel, cont = TRUE,
sleep = 0.5, mapfun = NULL, PLOT = TRUE ,
 proj = NULL, xlim = NULL, ylim = NULL)
```

Arguments

`lps`	list of filenames save in UW format
`sta`	station location list
`vel`	velocity list
`cont`	logical, true=add contours
`sleep`	pause time between each event in seconds
`mapfun`	mapping function
`PLOT`	logical, TRUE=plot relocation sequence
`proj`	projection list
`xlim`	limits on X-direction (km)
`ylim`	limits on Y-direction (km)

Details

Cycles through all the events, plots them and contours the first arrival times. Uses P-wave arrivals for plotting.

Value

list of events with error elliposids and convergence path

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Vlocate

Examples

```
## Not run:

cosoqk = RElocateEQ(lps, sta, vel, cont=TRUE,
 mapfun=cosoPLOT, proj=proj, xlim=xlim, ylim=ylim )



## End(Not run)
```

ReSet *Button to reset the choices of station and component*

Description

Button to reset the choices of station and component in swig and Mine.seis

Usage

```
ReSet(nh, g)
```

Arguments

nh	RSEIS list
g	swig parameters

Details

Driver for SELstaDB

Value

Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

SELstaDB, Mine.seis

Examples

```
## Not run:

buts = "ReSet"
swig(GH, PADDLAB=buts)


## End(Not run)
```

---

ripper *Rip off Event location information*

Description

Extract Event location information following Vlocate

Usage

```
ripper(AQ)
```

Arguments

AQ	event location list

Details

Extract lat-lon from event locations to track intermediate solutions and convergence

Value

2 by N matrix, lat-lon

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

plotEQ

Examples

```
## Not run:
qtip = ripper(AQ)


## End(Not run)
```

`Rowz2Keep` *Rows to Keep for inversion*

Description

Selects which rows in the hypocenter determination to keep during non-linear itaerations based on robust rsidual elimination.

Usage

```
Rowz2Keep(Ldat, EQ, G1, RESMAX)
```

Arguments

Ldat	List of station arrivals
EQ	Earthquake location
G1	derivative and travel time estimates
RESMAX	2-vector for P and S-wave residual maxima

Details

This is a utility used internally.

Residuals greater than the respective maxima provided are eliminated in the svd inversion. If fewer than 4 remain, the smallest 4 rows are returned.

Value

Index of good rows

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

XYlocate

RQ *Rquake Button*

Description

Driver for NLSlocate

Usage

```
RQ(nh, g, idev = 3)
```

Arguments

`nh`	RSEIS list
`g`	parameters from swig
`idev`	device for plotting

Details

Button to be called from within swig after picking.

Value

new hypocenter

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

NLSlocate, EQXYresid, XYSETUP, swig,chak

Examples

```
## Not run:

 buts = c("GPIX","PPIX", "PickWin", "fspread", "gMAP", "RQ" , "CONTPF")

swig(GH, PADDLAB=buts)

## End(Not run)
```

SavePF *Save Pick File Button*

Description

Save a pick file from within swig

Usage

```
SavePF(nh, g)
```

Arguments

nh	RSEIS data list
g	list of parameters internal to swig

Details

Uses PFoutput to save a pickfile to disk.

Value

Side Effects

Note

Pickfile is saved as a native R file with wpx extension

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

PFoutput

Examples

```
## Not run:
buts = "SavePF"
swig(GH, PADDLAB=buts)


## End(Not run)
```

SELstaDB *Pick stations and components interactively*

Description

Pick stations and components interactively. This is a routine used in swig.

Usage

```
SELstaDB(IDB, sel=1,  newdev = TRUE, STAY = FALSE)
```

Arguments

IDB	list of stations and components
sel	vector of index to selected traces
newdev	logical, whether to create a new device.
STAY	logical, whether to keep device active.

Value

vector of index to list of stations and components

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

infoDB, makeDB

Examples

```
## Not run: 
fpath = "/Users/lees/Site/CHAC/DATA"
fpat = "201111"
DBnov = makeDB(fpath, fpat, kind=2, Iendian=1, BIGLONG=FALSE)
IDB = infoDB(DBnov)
SELstaDB(IDB)

## End(Not run)
```

`UPdateEQLOC` *Update an Earthquake location*

Description

Update an Earthquake location following a relocation.

Usage

```
UPdateEQLOC(PF, sol, vel, stas = NULL)
```

Arguments

`PF`	Pickfile List
`sol`	solution vector (lat, lon, z, t0)
`vel`	1D velocity model
`stas`	station list (name, lat, lon, z)

Details

After re-picking or changing the model or the station corrections, update the event location in the pickfile.

Value

Pickfile List

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

EQXYresid, NLSlocate,PFoutput

Examples

```

## Not run: 
upf = UPdateEQLOC(PF,  sol, stas=stas)



## End(Not run)
```

viewCHAC *View Continuous Data*

Description

Scroll through continuous data recorded in the field. Uses a database describing the locations and content of each file stored on disk.

Usage

```
viewCHAC(DBnov , gstas, gcomps,sched, stas, buts='GPIX', preFILT=list()
, replot=TRUE , kind=2, Iendian=1, BIGLONG=FALSE)
```

Arguments

`DBnov`	RSEIS Data Base (output of makeDB)
`gstas`	stations to extract
`gcomps`	components to extract
`sched`	schedule of start times for extraction
`stas`	station list
`buts`	buttons for swig
`preFILT`	Pre-Filter traces before plotting.
`replot`	logical, TRUE=rerun swig after done click
`kind`	kind of data, 0=nativeR, 1=segy, 2=sac
`Iendian`	endian
`BIGLONG`	big long or short long

Details

These are set up for the CHAC dataset.

Value

Graphics, and Side effects

Note

The preFILT argument is the standard way of assigning filters in RSEIS. For example,
preFILT = list(ON=TRUE, fl=5 , fh=15, type="BP", proto="BU")

will bandpass filter the traces between 5 and 15 Hz. If the logical ON flag of the filter is turned to FALSE, the filter is not applied.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

makeDB, Mine.seis

Examples

```
## Not run:

##########  set up data base:
fpath = "/home/lees/Site/CHAC/DATA"
fpat = "201111"
DBnov = makeDB(fpath, fpat, kind=2, Iendian=1, BIGLONG=FALSE)
###   get information:
IDB = infoDB(DBnov)

#####  select stations and components:
gstas = IDB$usta[-which(IDB$usta=="CHAC5")]
gcomps = IDB$ucomp[1:3]

####  extra buttons
buts = c("YPIX", "SPEC", "SGRAM", "WLET")
 fsta = "/home/lees/Site/CHAC/staLLZ.txt"
stas = scan(file=fsta,what=list(name="", lat=0, lon=0, z=0))
stas$z = stas$z/1000

###  set schedule
sched  =seq(from=325, to=335, by=1/24)

##########  open 2 windows
X11()
X11()

###  set main window to dev 2
dev.set(2)

###  set pre-filter (needs to be ON=TRUE to work

preFILT = list(ON=TRUE, fl=1/2 , fh=8, type="BP", proto="BU")

viewCHAC( DBnov , gstas, gcomps , sched, stas, buts =buts,
       preFILT = preFILT,kind = 2, Iendian = 1, BIGLONG = FALSE )



## End(Not run)
```

viewseis *View Continuous Data*

Description

Scroll through continuous data recorded in the field. Uses a database describing the locations and content of each file stored on disk.

Usage

```
viewseis(DBnov , gstas, gcomps,sched, stas,
buts='GPIX', replot=TRUE , kind=0, Iendian=1, BIGLONG=FALSE)
```

Arguments

`DBnov`	RSEIS Data Base
`gstas`	stations to extract
`gcomps`	components to extract
`sched`	schedule of start times for extraction
`stas`	station list
`buts`	buttons for swig
`replot`	logical, TRUE=rerun swig after done click
`kind`	kind of data, 0=nativeR, 1=segy, 2=sac
`Iendian`	endian
`BIGLONG`	big long or short long

Details

These are set up for the seis dataset

Value

Graphics, and Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

makeDB, Mine.seis

Examples

```
## Not run:
sched  =seq(from=325, to=335, by=1/24)
viewseis( DBnov , gstas, gcomps , sched, kind=2,
 Iendian=1, BIGLONG=FALSE)

## End(Not run)
```

Vlocate *Hypocenter Determination*

Description

Hypocenter Determination with error checking and adjustments.

Usage

```
Vlocate(Ldat,EQ,vel,
                  distwt = 10,
                  lambdareg =100,
                  REG = TRUE,
                  WTS = TRUE,
                  STOPPING = TRUE,
                  tolx = 0.1,
                  toly = 0.1,
                  tolz = 0.5,
                  RESMAX = c(.4,.5),
                  maxITER = c(7, 5, 7, 4),
                  PLOT=FALSE)
```

Arguments

`Ldat`	list, must inlude: lat, lon ,err, sec, cor (see details)
`EQ`	list, must inlude: lat,lon,z, t
`vel`	list, 1D velocity structure
`distwt`	distance weighting factor
`lambdareg`	regularization parameter for damping
`REG`	logical, TRUE=use regularization
`WTS`	logical, TRUE==use weighting
`STOPPING`	logical, TRUE=use stopping criteria
`tolx`	numeric, tolerance in km in x direction
`toly`	numeric, tolerance in km in y direction
`tolz`	numeric, tolerance in km in z direction
`RESMAX`	vector, residual max for P and S, default=c(4,5)
`maxITER`	vector, Maximum number of iterations for each section of the location routine, default=c(7,5,7,4)
`PLOT`	logical, plot results during iterations

Details

This is a wrapper for XYlocate, only here the lat-lon of the stations is passed and the code does the projection internally.

There are 3 main loops, each controled by differing input params: first event is located only in XY keeping the depth fixed (7 iterations). Then an initial free solution is estimated using robust elimination of residual based on RESMAX (5 iterations). Finally a set of 7 iterations is applied providing the final estimate, along with error bars, elliposids, etc.

In the event no good solution is derived, the regularization parameter is doubled and a loop with 4 iterations is applied, and the result returned.

Value

list:

`EQ`	Hypocenter lcoation
`ERR`	Error Analysis
`its`	number of iteration
`Ksolutions`	list of matrices, each with intermediate x,y,z,t locations

Note

The schedule may be adjusted by duplicating this function and changing the maxit parameters.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Lee and Stewart

See Also

XYlocate, Klocate, DoRLocate

Examples

```
## Not run:
library(RSEIS)
data(GH)

g1 = GH$pickfile

data(VELMOD1D)
vel= VELMOD1D
```

```
 w1 = which(!is.na(g1$STAS$lat))
         sec = g1$STAS$sec[w1]

         N = length(sec)
         Ldat =    list(
           name = g1$STAS$name[w1],
           sec = g1$STAS$sec[w1],
           phase = g1$STAS$phase[w1],
           lat=g1$STAS$lat[w1],
           lon = g1$STAS$lon[w1],
           z = g1$STAS$z[w1],
           err= g1$STAS$err[w1],
           yr = rep(g1$LOC$yr , times=N),
           jd = rep(g1$LOC$jd, times=N),
           mo = rep(g1$LOC$mo, times=N),
           dom = rep(g1$LOC$dom, times=N),
           hr =rep( g1$LOC$hr, times=N),
           mi = rep(g1$LOC$mi, times=N) )


wstart = which.min(Ldat$sec)
        EQ = list(lat=Ldat$lat[wstart], lon=Ldat$lon[wstart], z=6, t=Ldat$sec[wstart] )


  AQ = Vlocate(Ldat,EQ,vel,
      distwt = 10,
      lambdareg =100 ,
      REG = TRUE,
      WTS = TRUE,
      STOPPING = TRUE,
      tolx =   0.01,
      toly = 0.01 ,
      tolz = 0.05, maxITER = c(7,5,7,4) , RESMAX = c(0.1, 0.1),  PLOT=FALSE)

## End(Not run)
```

`XYerror.bars` *Error Bars in X and Y*

Description

Error Bars in X and Y

Usage

```
XYerror.bars(x, y, xlo = 0, xhi = 0, ylo = 0,
yhi = 0, pch = 1, col = 1, barw = 0.1, add = FALSE, ...)
```

Arguments

`x`	X-values
`y`	Y-values
`xlo`	X Lower limit of error bars
`xhi`	X Upper limit of error bars
`ylo`	Y Lower limit of error bars
`yhi`	Y Upper limit of error bars
`pch`	plotting character
`col`	color
`barw`	width of the bar (inches)
`add`	logical, add=FALSE starts a new plot
`...`	other plotting parameters

Value

graphical side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
set.seed(0)
zup = rnorm(10)

x = 1:10
y = 2*x+5+zup

ydown = rnorm(10)
ydown = ydown-min(ydown)+.2

yup = rnorm(10)
yup = yup-min(yup)+.2



zup = rnorm(10)
xup = zup-min(zup)+.5
xdown = rnorm(10)
xdown = xdown-min(xdown)+.2


####  example with different  error on either side:
XYerror.bars(x, y, y-ydown, y+yup, x-xdown, x+xup,
 pch = 1, col = 'brown' , barw = 0.1, add
= FALSE)
```

XYlocate *Locate Earthquake with UTM projection*

Description

Non-linear hypocenter location with UTM geographical projection. Used for locating earthquakes in local or regional settings.

Usage

```
XYlocate(Ldat, EQ, vel, maxITER = 10, distwt = 10,
lambdareg = 100, FIXZ
= FALSE, REG = TRUE, WTS = TRUE, STOPPING = TRUE,
RESMAX = c(.4,.5), tolx = 0.005, toly = 0.005,
 tolz = 0.01, PLOT = FALSE)
```

Arguments

`Ldat`	list, must inlude: x,y,err, sec, cor (see details)
`EQ`	list, must inlude: x,y,z, t
`vel`	list, 1D velocity structure
`maxITER`	Maximum number of iterations
`distwt`	distance weighting factor
`lambdareg`	regularization parameter for damping
`FIXZ`	logical, TRUE = fix depth, i.e. only calculate x,y,t
`REG`	logical, TRUE=use regularization
`WTS`	logical, TRUE==use weighting
`STOPPING`	logical, TRUE=use stopping criteria
`RESMAX`	vector, residual max for P and S, default=c(4,5)
`tolx`	numeric, tolerance in km in x direction
`toly`	numeric, tolerance in km in y direction
`tolz`	numeric, tolerance in km in z direction
`PLOT`	logical, plot results during iterations

Details

Input pick list must have at x,y,z, sec, cor, err elements for each station. If no station correction is available it is set to zero. If no uncertainty (err) is available, it is set to 0.05 sec. Each station must have a finite x-y coordinate and arrival time in seconds. Events are located relative to the minute.

Routine uses the svd in a sequence of linear inversions to estimate the nonlinear location.

Value

List:

`EQ`	list, Earthquake hypocenter and time
`its`	number of iterations
`rms`	rms residual
`wrms`	wheighted rms residual
`used`	vector, index of used equations
`guesses`	list of x,y,z,t intermediate locations when converging

Note

This routine should be called by a wrapper (Vlocate) that applies the algorithm several times and changes parameters based on the quality.

If RESMAX is used and the robust approach yields fewer than 4 equations, the best (smallest) four residuals will be used to determiine the event location.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Vlocate

Examples

```
## Not run:
library(RSEIS)
data(GH)

g1 = GH$pickfile
data(VELMOD1D)
vel= VELMOD1D



 w1 = which(!is.na(g1$STAS$lat))
         sec = g1$STAS$sec[w1]

         N = length(sec)
         Ldat =    list(
           name = g1$STAS$name[w1],
           sec = g1$STAS$sec[w1],
           phase = g1$STAS$phase[w1],
           lat=g1$STAS$lat[w1],
           lon = g1$STAS$lon[w1],
           z = g1$STAS$z[w1],
           err= g1$STAS$err[w1],
           yr = rep(g1$LOC$yr , times=N),
           jd = rep(g1$LOC$jd, times=N),
           mo = rep(g1$LOC$mo, times=N),
           dom = rep(g1$LOC$dom, times=N),
           hr =rep( g1$LOC$hr, times=N),
           mi = rep(g1$LOC$mi, times=N) )

 MLAT = median(Ldat$lat)
    MLON = median(Ldat$lon)

    proj = GEOmap::setPROJ(type=2, LAT0=MLAT, LON0=MLON)

####   get station X-Y values in km
    XY = GEOmap::GLOB.XY(Ldat$lat, Ldat$lon, proj)
###   add to Ldat list
    Ldat$x = XY$x
    Ldat$y = XY$y
       wstart = which.min(Ldat$sec)



EQ = list(x=XY$x[wstart], y=XY$y[wstart], z=6, t=Ldat$sec[wstart] )


 maxITER = 7
###print(EQ)
    AQ = XYlocate(Ldat,EQ,vel,
      maxITER = maxITER,
      distwt = 1,
      lambdareg =10 ,
      FIXZ = FALSE,
      REG = TRUE,
      WTS = TRUE,
      STOPPING = TRUE,
      RESMAX = c(0.1,0.1),
      tolx =   0.001,
      toly = 0.001 ,
      tolz = 0.5, PLOT=FALSE)

########  update the new location

AXY = GEOmap::XY.GLOB(AQ$EQ$x, AQ$EQ$y, proj)
AQ$EQ$lat = AXY$lat
AQ$EQ$lon = AXY$lon
if(AQ$EQ$lon>180) { AQ$EQ$lon = AQ$EQ$lon-360 }


plot(c(Ldat$x, AQ$EQ$x) , c(Ldat$y,AQ$EQ$y), type='n' , xlab="km",
ylab="km" )

points(Ldat$x, Ldat$y, pch=6)

points(AQ$EQ$x, AQ$EQ$y, pch=8, col='red')
points(EQ$x, EQ$y, pch=4, col='blue')


legend("topright", pch=c(8,4, 6), col=c("red", "blue", "black"),
 legend=c("Final location", "Initial guess", "Station"))


print(AQ)

EQ$x = 10
EQ$y = 2


## End(Not run)
```

XYSETUP *Set up matrix for hypocenter inversion*

Description

Set up matrix for hypocenter inversion

Usage

```
XYSETUP(STAS, init, vel)
```

Arguments

STAS	station information from pickfile
init	initial event location
vel	list, velocity

Details

This sets up the matrix used for nonlinear inversion. The code does not include information on the weighting. Station corrections are included.

The STAS are an internal component of the pickfile.

Value

matrix

Note

Need scheme for weighting according to errors in picks and distance weighting.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

setPROJ, GLOB.XY,NLSlocate

Examples

```
## Not run:
##  start with the location of the closest station
XY = XYSETUP(STAS, c(STAS$lat[w1],STAS$lon[w1], initz,  STAS$sec[w1]-t0a  ) , vel  )


## End(Not run)
```

Y2Pphase *Convert Y-phase to P-phase*

Description

Removes extraneous other-phase from a pick file. If Ypix were made initially as a rough pick, this removes them.

Usage

```
Y2Pphase(twpx, phase)
```

Arguments

twpx	WPX list
phase	character, phase to exchange to P

Details

Initially many events may be picked using GPIX button. These should be removed after the P-phases have been determined with PickWin.

Value

WPX returned without other-phases

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

PPIX, GPIX, YPIX, PickWin

Examples

```
## Not run:

newwpx = Y2Pphase(twpx, "G" )

## End(Not run)
```

Index