Package 'stellaR'

February 20, 2015

Title stellar evolution tracks and isochrones

Version 0.3-3

Date 2013-01-21

Description A package to manage and display stellar tracks and isochrones from Pisa low-mass database. Includes tools for isochrones construction and tracks interpolation.

Suggests lattice

URL The database is described at:

http://astro.df.unipi.it/stellar-models/

LazyData yes

License GPL (>= 2)

Maintainer Matteo Dell'Omodarme <mattdell@fastmail.fm>

Author Matteo Dell'Omodarme [aut, cre], Giada Valle [aut]

Repository CRAN

Date/Publication 2013-01-21 12:59:36

NeedsCompilation no

R topics documented:

`stellaR-package` *Stellar tracks and isochrones from Pisa database*

Description

The package is designed to manage and display stellar tracks and isochrones from Pisa low-mass database. Data are archived at CDS and are available for download (http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/540/A26).

The package includes tools to gather data from an active Internet connection, print and plot them with appropriate format.

Tools for make new isocrones at desired ages, and to interpolate the database for composition not already available are also provided.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

`compositions` *Compostions of computed models*

Description

The data sets gives the compositions of the computed models.

Usage

```
compositions
```

Format

A list containing 4 variables:

z the initial metallicities of the models.

y a data frame with the initial helium abundances of the models.

ml the mixing-lengths of the models.

afe the alpha-enhanchments of the models. It is coded `0` for [alpha/Fe] = 0.0 and `1` for [alpha/Fe] = 0.3.

Source

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

`getHbgrid` *HB models from ZAHB to thermal pulses*

Description

Import stellar track data for a grid of HB models starting from ZAHB and ending at the onset of thermal pulses.

Usage

```
getHbgrid(z, y, ml, afe,
          baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
```

Arguments

`z`	the initial metallicity of the model to get. Allowed values are 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01.
`y`	the initial helium abundance of the model to get. Allowed values are 0.25, 0.27, 0.33, 0.38, 0.42.
`ml`	the mixing-length of the model to get. Allowed values are 1.7, 1.8, 1.9.
`afe`	the alpha-enhanchment of the model to get. It can be one of `afe = 0` for [alpha/Fe] = 0.0 or `afe = 1` for [alpha/Fe] = 0.3.
`baseURL`	the URL of the base directory of the database from where get the models.

Details

As a sanity check, the function performs a consistency test on the parameters `z`, `y`, `ml`, `afe` by a call to `testComposition`.

By default the function requires an Internet access for data gathering. Tracks and isochrones are downloaded from the on-line repository CDS (`http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/540/A26`). It is also possible to gather data from a local download of the catalog, specifying as `baseURL` the root directory of the download. In this case the structure of the catalog and the names of files and directory must be preserved.

Value

`getHbgrid` returns an object of class `hbset`, i.e. a list with one element of class `hb` for each HB model.

If `baseURL` points to an invalid path or CDS ftp site is unavailable the function returns `NA` and issues a warning.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

See Also

getTrk, getHb, getIso, getTrkSet.

Examples

```
  ### slow!
  ## Not run:
  hbgrid <- getHbgrid(0.002, 0.25, 1.7, 0)

  ### get data from local directory /data
  hbgrid <- getHbgrid(0.002, 0.25, 1.7, 0, baseURL="/data/")
## End(Not run)
```

---

getIso *Import stellar isochrones data*

Description

Import a stellar isochrone.

Usage

```
getIso(age, z, y, ml, afe,
         baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
```

Arguments

age	the age (in Gyr) of the isochrone to get. Allowed values are 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0.
z	the initial metallicity of the model to get. Allowed values are 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01.
y	the initial helium abundance of the model to get. Allowed values are 0.25, 0.27, 0.33, 0.38, 0.42.
ml	the mixing-length of the model to get. Allowed values are 1.7, 1.8, 1.9.
afe	the alpha-enhanchment of the model to get. It can be one of `afe = 0` for [alpha/Fe] = 0.0 or `afe = 1` for [alpha/Fe] = 0.3.
baseURL	the URL of the base directory of the database from where get the models.

Details

As a sanity check, the function performs a consistency test on the parameters `z`, `y`, `ml`, `afe` by a call to `testComposition`.

By default the function requires an Internet access for data gathering. Tracks and isochrones are downloaded from the on-line repository CDS (`http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/540/A26`). It is also possible to gather data from a local download of the catalog, specifying as `baseURL` the root directory of the download. In this case the structure of the catalog and the names of files and directory must be preserved.

Value

`getIso` returns an object of class `iso`. More specifically this is a list with elements:

`age`	the age of the isochrone (in Gyr)
`z`	the metallicity of the model
`y`	the initial helium content of the model
`alpha.enh`	the mixing-length value of the model
`data`	the dataset of stellar isocrone, containing the following five variables: **logL** logarithm of the luminosity (L in unit of solar luminosity) **logTe** logarithm of the effective temperature (Te in K) **mass** mass of the star (in unit of solar mass) **radius** radius of the star (in unit of solar radius) **logg** logarithm of the surface gravity (g in cm s^-2)

`print`, and `plot` methods are available for the class.

If `baseURL` points to an invalid path or CDS ftp site is unavailable the function returns `NA` and issues a warning.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

See Also

`getZahb`, `getHbgrid`, `getTrk`, `getTrkSet`.

Examples

```
  ## Not run:
  iso <- getIso(12.0, 0.002, 0.25, 1.7, 0)

  ### get data from local directory /data
  iso <- getIso(12.0, 0.002, 0.25, 1.7, 0, baseURL="/data/")
## End(Not run)
```

getTrk *Import stellar track data*

Description

Import a stellar evolutionary track from pre-main sequence to He flash and from ZAHB to thermal pulses.

Usage

```
getTrk(m, z, y, ml, afe,
         baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
getHb(m, z, y, ml, afe,
        baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
```

Arguments

m	the mass, in unit of solar mass, of the model to get. Allowed values are 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10.
z	the initial metallicity of the model to get. Allowed values are 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01.
y	the initial helium abundance of the model to get. Allowed values are 0.25, 0.27, 0.33, 0.38, 0.42.
ml	the mixing-length of the model to get. Allowed values are 1.7, 1.8, 1.9.
afe	the alpha-enhanchment of the model to get. It can be one of `afe = 0` for [alpha/Fe] = 0.0 or `afe = 1` for [alpha/Fe] = 0.3.
baseURL	the URL of the base directory of the database from where get the models. See `Details` for further information.

Details

As a sanity check, the function performs a consistency test on the parameters `z`, `y`, `ml`, `afe` by a call to `testComposition`.

By default the function requires an Internet access for data gathering. Tracks and isochrones are downloaded from the on-line repository CDS (http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/540/A26). It is also possible to gather data from a local download of the catalog, specifying as `baseURL` the root directory of the download. In this case the structure of the catalog and the names of files and directory must be preserved.

Value

getTrk returns an object of class trk; getHb returns an object of class hb.

More specifically these objects are lists with common elements:

mass	the mass of the model (in unit of solar mass)
z	the metallicity of the model
y	the initial helium content of the model
ml	the mixing-length value of the model
alpha.enh	the alpha-enhanchment of the model. It can be one of alpha.enh = 0 for [alpha/Fe] = 0.0 or alpha.enh = 1 for [alpha/Fe] = 0.3.
data	the dataset of stellar track, containing the following variables: **mod** model number **time** logarithm of the evolutionary time (in yr) **logL** logarithm of the luminosity (L in unit of solar luminosity) **logTe** logarithm of the effective temperature (Te in K) **mass** mass of the star (in unit of solar mass) **Hc** central H/He abundance **logTc** logarithm of the central temperature (Tc in K) **logRHOc** logarithm of the central density (RHOc in g cm^-3) **MHEc** mass of the He core (in unit of solar mass) **Lpp** luminosity of pp chain (in unit of total luminosity L) **LCNO** luminosity of CNO cycle (in unit of total luminosity L) **L3a** luminosity of triple-alpha (in unit of total luminosity L) **Lg** gravitational luminosity (in unit of total luminosity L) **radius** radius of the star (in unit of solar radius) **logg** logarithm of the surface gravity (g in cm s^-2) Only time, logL, and logTe are mandatory, while all other variables are optional.

hb class contains also the element:

massRGB	the mass of the RGB progenitor

print, and plot methods are available for the two classes.

If baseURL points to an invalid path or CDS ftp site is unavailable the function returns NA and issues a warning.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

See Also

getZahb, getHbgrid, getIso, getTrkSet.

Examples

```
## Not run:
trk <- getTrk(0.9, 0.002, 0.25, 1.7, 0)

### get data from local directory /data
trk <- getTrk(0.9, 0.002, 0.25, 1.7, 0, baseURL="/data/")

###  multi-panel plot of the various quantities with time
track <- getTrk(0.80, 0.001, 0.25, 1.90, 0)
if(!is.na(track)[1]) {
trkdata <- within(stack(track$data), time <- rep(track$data$time,
    length.out=length(values)) )
require(lattice)
xyplot( values ~ time | ind, data=trkdata, type="l",
    scales=list(y=list(relation="free")))
}
## End(Not run)
```

`getTrkSet` *Import a set of data*

Description

Import a set of stellar evolutionary tracks from pre-main sequence to He flash, or a set of isochrones.

Usage

```
getTrkSet(m, z, y, ml, afe,
            baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
getIsoSet(age, z, y, ml, afe,
            baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
```

Arguments

`m`	a vector of masses, in unit of solar mass, of the models to get. Allowed values are 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10.
`age`	a vector of ages, in Gyr, of the models to get. Allowed values are 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0.
`z`	a vector of the initial metallicity of the models to get. Allowed values are 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01.
`y`	a vector of the initial helium abundance of the models to get. Allowed values are 0.25, 0.27, 0.33, 0.38, 0.42.

ml	a vector of the mixing-length of the models to get. Allowed values are 1.7, 1.8, 1.9.
afe	the alpha-enhanchment of the models to get. It can be one of `afe = 0` for [alpha/Fe] = 0.0 or `afe = 1` for [alpha/Fe] = 0.3.
baseURL	the URL of the base directory of the database from where get the models.

Details

As a sanity check, the function performs a consistency test on the parameters z, y, ml, afe by a call to testComposition.

By default the function requires an Internet access for data gathering. Tracks and isochrones are downloaded from the on-line repository CDS (http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/540/A26). It is also possible to gather data from a local download of the catalog, specifying as baseURL the root directory of the download. In this case the structure of the catalog and the names of files and directory must be preserved.

Value

getTrkSet returns an object of class trkset, i.e. a list of objects of class trk. getIsoSet returns an object of class isoset, i.e. a list of objects of class iso.

print, and plot methods are available for the two classes.

If baseURL points to an invalid path or CDS ftp site is unavailable the function returns NA and issues a warning.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

See Also

getZahb, getHbgrid, getHb, getIso.

Examples

```
  ## Not run:
  ### get two masses
  trkset <- getTrkSet(c(0.9, 1.0), 0.002, 0.25, 1.7, 0)

  ### get two masses at two metallicity, for a total of 4 objects
  trkset <- getTrkSet(c(0.9, 1.0), c(0.002, 0.01), 0.25, 1.7, 0)

  ### get data from local directory /data
  set <- getTrkSet(0.9, 0.002, c(0.25, 0.33), 1.7, 0, baseURL="/data/")
## End(Not run)
```

getZahb *Import stellar track data for ZAHB*

Description

Import a stellar ZAHB model.

Usage

```
getZahb(z, y, ml, afe,
          baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
```

Arguments

z	the initial metallicity of the model to get. Allowed values are 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01.
y	the initial helium abundance of the model to get. Allowed values are 0.25, 0.27, 0.33, 0.38, 0.42.
ml	the mixing-length of the model to get. Allowed values are 1.7, 1.8, 1.9.
afe	the alpha-enhanchment of the model to get. It can be one of `afe = 0` for [alpha/Fe] = 0.0 or `afe = 1` for [alpha/Fe] = 0.3.
baseURL	the URL of the base directory of the database from where get the models.

Details

As a sanity check, the function performs a consistency test on the parameters z, y, ml, afe by a call to testComposition.

By default the function requires an Internet access for data gathering. Tracks and isochrones are downloaded from the on-line repository CDS (http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/540/A26) It is also possible to gather data from a local download of the catalog, specifying as baseURL the root directory of the download. In this case the structure of the catalog and the names of files and directory must be preserved.

Value

getZahb returns an object of class zahb More specifically it is a list with elements:

z	the metallicity of the model
y	the initial helium content of the model
ml	the mixing-length value of the model
alpha.enh	the alpha-enhanchment of the model. It can be one of `alpha.enh = 0` for [alpha/Fe] = 0.0 or `alpha.enh = 1` for [alpha/Fe] = 0.3.
data	the dataset of stellar track, containing the following variables:

mass mass of the star (in unit of solar mass)

logTe logarithm of the effective temperature (Te in K)

logL logarithm of the luminosity (L in unit of solar luminosity)

`print`, and `plot` methods are available for the two classes.

If `baseURL` points to an invalid path or CDS ftp site is unavailable the function returns `NA` and issues a warning.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

See Also

`getTrk`, `getHbgrid`, `getIso`, `getTrkSet`.

Examples

```
  ## Not run:
  zahb <- getZahb(0.002, 0.25, 1.7, 0)

  ### get data from local directory /data
  zahb <- getZahb(0.002, 0.25, 1.7, 0, baseURL="/data/")
## End(Not run)
```

`interpTrk` *Interpolate stellar tracks data*

Description

Construct a set of interpolated tracks from the objects in the database.

Usage

```
interpTrk(z, y, ml, afe, vmass=seq(0.30,1.10, by=0.05),
          baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
```

Arguments

`z`	the initial metallicity of the models to generate. Allowed values are between `0.0001` and `0.01`.
`y`	the initial helium abundance of the models to generate. Allowed values are between `0.25` and `0.42`.
`ml`	the mixing-length of the model to generate. Allowed values are between `1.7` and `1.9`.

afe	the alpha-enhanchment of the model to get. It can be one of `afe = 0` for [alpha/Fe] = 0.0 or `afe = 1` for [alpha/Fe] = 0.3.
vmass	vector of masses of the stellar model to generate. Default to the whole set in the database. Allowed values are 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10.
baseURL	the URL of the base directory of the database from where get the models.

Details

A 3D interpolation is performed on the database to construct the required set of tracks. Interpolation on helium abundance and on mixing-length occur linearly on the variables, while the metallicity is log-transformed before the interpolation procedure.

Value

The function returns an object of class `trkset` containing the interpolated tracks.

Examples

```
  ### slow!
  ## Not run:
  iptrk <- interpTrk(0.002, 0.25, 1.74, 0)

  ### get data from local directory /data
  iptrk <- interpTrk(0.002, 0.25, 1.74, 0, baseURL="/data/")
## End(Not run)
```

---

keypoints *Extract relevant evolutionary points from stellar tracks or isochrones*

Description

Extract some relevant evolutionary points from stellar tracks or stellar isochrones.

Usage

```
## S3 method for class 'trk'
keypoints(x, ...)
## S3 method for class 'trkset'
keypoints(x, ...)
## S3 method for class 'iso'
keypoints(x, ...)
## S3 method for class 'isoset'
keypoints(x, ...)
```

Arguments

x	an object of classes trk, trkset, iso, or isoset.
...	further arguments, not implemented.

Details

keypoints.trk extracts from the object x$data the rows relative to the following evolutionary stages:

1. ZAMS: Zero-Age Main-Sequence, defined as the point for which the central H abundance drops below 99% of its initial value.
2. TO: Turn-Off, defined as the point for which the effective temperature reaches its maximum value. If multiple lines satisfy the constraint, the values of all the rows are averaged.
3. BTO: Brighter Turn-Off, defined as the point for which the effective temperature drops below the one of the TO minus 100 K. The points could not exist for low masses. See Chaboyer et al. (1996) for an analysis of the advantages of this evolutionary point with respect to TO.
4. exHc: Central H exhaustion, defined as the point for which the central H abundance is zero. For low masses the point could coincide with TO. This is the last point of the tracks with mass lower or equal to 0.50 Msun.
5. Heflash: Helium flash, the last point of the track for masses higher than 0.50 Msun.

keypoints.trkset performs the selection described above for all the set of objects.

The returned object contains the variable id, which labels the evolutionary phases with the following coding: 1 = ZAMS, 2 = TO, 3 = BTO, 4 = exHc, 5 = Heflash.

keypoints.iso extracts from the object x$data the rows relative to the following evolutionary stages:

1. TO: Turn-Off, defined as the point for which the effective temperature reaches its maximum value. If multiple lines satisfy the constraint, the values of all the rows are averaged.
2. BTO: Brighter Turn-Off, defined as the point for which the effective temperature drops below the one of the TO minus 100 K.

keypoints.isoset performs the selection described above for all the set of objects.

The returned object contains the variable id, which labels the evolutionary phases with the following coding: 1 = TO, 2 = BTO.

Value

A data frame containing the observations relative to the selected evolutionary phases. The columns relative to mass (or age for isochrones), metallicity, initial He value, mixing-length, alpha-enhancement, and phase identifier are inserted in the data frame.

References

B. Chaboyer, P. Demarque, P.J. Kernan, L.M. Krauss, and A. Sarajedini (1996). *An accurate relative age estimator for globular clusters*. MNRAS, 283, 683-689.

Examples

```
  ## Not run:
  trk <- getTrk(0.9, 0.002, 0.25, 1.7, 0)
  ### check return value from CDS
  if(!is.na(trk)[1]) keypoints(trk)

  is <- getIso(11, 0.002, 0.25, 1.7, 0)
  ### check return value from CDS
  if(!is.na(is)[1]) keypoints(is)

## End(Not run)
```

---

makeIso *Construct stellar isochrones from tracks*

Description

The function computes by interpolation stellar isochrones of given ages from evolutionary tracks.

Usage

```
makeIso(age, z=NULL, y=NULL, ml=NULL, afe=NULL, log=FALSE, linear=TRUE,
        tr=NULL, baseURL="ftp://cdsarc.u-strasbg.fr/pub/cats/J/A+A/540/A26/")
```

Arguments

age	the ages (in Gyr) of the isochrones to construct. A vector of ages is allowed.
z	the initial metallicity of the isochrones to build.
y	the initial helium abundance of the isochrones to build.
ml	the mixing-length of the isochrones to build.
afe	the alpha-enhanchment of the isochrone to build. It can be one of `afe = 0` for [alpha/Fe] = 0.0 or `afe = 1` for [alpha/Fe] = 0.3.
log	a logical value indicating whether the interpolation of the tracks should be performed on the logarithm of the evolutionary time.
linear	a logical value indicating whether linear interpolation should be performed. Only the option `linear=TRUE` is currently available.
tr	a set of track to be used for isocrones construction. See `Details` for further information.
baseURL	the URL of the base directory of the database from where get the models.

Details

Isochrones are obtained by mean of a standard interpolation procedure on the set of tracks. Let $S(m)$ be the set of tracks, parametrized by the value of the mass m. Let $t_i(m)$ be the evolutionary time for the ith point on the track of mass m. Let be k the point on the track of lower mass of $S(m)$ for which $t_k(m)$ is greater of the time required for the isochrone.

For each point $j >= k$ on $S(m)$, an interpolation of mass, logarithm of the effective temperature and logarithm of the luminosity is performed among tracks. These points define the required isochrone.

If a set of tracks is supplied by mean of the argument `tr`, the function interpolates among these tracks. The values of `z`, `y`, `ml`, `afe` are recovered from the supplied objects and a test of consistency is performed to assure that the tracks are homogeneous in these parameters.

Value

The function returns an object of class `isoset`.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

Examples

```
 ### slow!
 ## Not run:
  isoset <- makeIso(11.2, 0.002, 0.25, 1.7, 0)
## End(Not run)

  ### get data from local directory /data
  ## Not run: isoset <- makeIso(c(11.2, 12.4), 0.002, 0.25, 1.7, 0, baseURL="/data/")
```

`masshb` *Mass of stars in HB*

Description

These data sets give the stellar mass in HB for the progenitors in RGB.

Usage

```
masshb
masshbgrid
```

Format

The two data frames contain 6 variables:

mass the RGB mass of the model.

z the initial metallicity of the model.

y the initial helium abundance of the model.

ml the mixing-length of the model.

mix the alpha-enhanchment of the model. It is coded `mix=AS09a0` for [alpha/Fe] = 0.0 and `mix=AS09a3` for [alpha/Fe] = 0.3.

masshb the HB mass of the model. The dataset `masshbgrid` contains multiple values for `masshb` for each single RGB progenitor mass.

Source

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

---

plot.trk *Plot stellar track or isochrones objects*

Description

Plot one evolutionary track or one isochrone extracted from objects of classes `trk`, `hb`, `zahb`, `iso`.

Usage

```
## S3 method for class 'trk'
plot(x, add=FALSE, ...)
## S3 method for class 'hb'
plot(x, add=FALSE, ...)
## S3 method for class 'zahb'
plot(x, add=FALSE, ...)
## S3 method for class 'iso'
plot(x, add=FALSE, ...)
```

Arguments

`x`	an object of classes `trk`, `hb`, `zahb`, or `iso`.
`add`	a logical value indicating whether the plot should be added to an existing device.
`...`	further arguments passed to `plotAstro`.

Details

The contents of the vectors `x$data$logTe` and `x$data$logL` are plotted by a call to `plotAstro`. Sensible values for the axes major ticks and labels are chosen.

See Also

plotAstro.

Examples

```
  ## Not run:
  trk <- getTrk(0.9, 0.002, 0.25, 1.7, 0)

  ### check return value from CDS
  if(!is.na(trk)[1]) plot(trk)
## End(Not run)
```

---

plot.trkset *Plot a set stellar objects*

Description

Plot a set of evolutionary tracks from objects of classes trkset, and hbset, or a set of isochrones from object of class isoset.

Usage

```
## S3 method for class 'trkset'
plot(x, add=FALSE, col=1, lty=1, xlim=NULL, ylim=NULL, ...)
## S3 method for class 'hbset'
plot(x, add=FALSE, col=1, lty=1, xlim=NULL, ylim=NULL, ...)
## S3 method for class 'isoset'
plot(x, add=FALSE, col=1, lty=1, xlim=NULL, ylim=NULL, ...)
```

Arguments

x	an object of class trkset, hbset, or isoset.
add	a logical value indicating whether the plot should be added to an existing device.
col	the colors for lines. Multiple colors can be specified so that each track can be given its own color. If there are fewer colors than tracks they are recycled in the standard fashion.
lty	the line types. Multiple types can be specified so that each track can be given its own type. If there are fewer type than tracks they are recycled in the standard fashion.
xlim	the x limits (x1, x2) of the plot. The default value, NULL, indicates that the range of the finite values to be plotted should be used.
ylim	the y limits of the plot.
...	further arguments passed to plotAstro.

Details

For each object X contained in the input of class trkset, hbset, or isoset the contents of the vectors X$data$logTe and X$data$logL are plotted by a call to plotAstro. Sensible values for the axes ranges, major ticks and labels are chosen.

See Also

plotAstro.

Examples

```
  ## Not run:
  trkset <- getTrkSet(c(0.7, 0.9, 1.1), 0.002, 0.25, 1.7, 0)

  ### check return value from CDS
  if(!is.na(trkset)[1]) plot(trkset)
## End(Not run)
```

---

plotAstro *Plot function for stellar objects*

Description

Draw a scatter plot in the active graphics window.

Usage

```
plotAstro(x, y, type="l", xlab="", ylab="", xi=(max(x)-min(x))/5,
          yi=(max(y)-min(y))/5, xmt=3, ymt=3, revX=FALSE, revY=FALSE,
          xlim=NULL, ylim=NULL, cex=1.0, cex.axis=1.3, cex.lab=1.5,
          add=FALSE, ...)
```

Arguments

x, y	the x and y arguments provide the x and y coordinates for the plot.
type	1-character string giving the type of plot desired. The following values are possible: "p" for points, "l" for lines (default value), "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
xlab	a label for the x axis, default to empty string.
ylab	a label for the y axis, default to empty string.
xi	major ticks spacing for x axis.

yi	major ticks spacing for y axis.
xmt	number of minor ticks between two major ticks for x axis.
ymt	number of minor ticks between two major ticks for y axis.
revX	a logical value indicating whether the x axis should be reversed.
revY	a logical value indicating whether the y axis should be reversed.
xlim	the x limits (x1, x2) of the plot. The default value, NULL, indicates that the range of the finite values to be plotted should be used.
ylim	the y limits of the plot.
cex	the amount by which plotting text and symbols should be magnified relative to the default.
cex.axis	the magnification to be used for axis annotation relative to the current setting of cex.
cex.lab	the magnification to be used for x and y labels relative to the current setting of cex.
add	a logical value indicating whether the plot should be added to an existing device.
...	further arguments passed to low level plot functions.

See Also

plot.trk, plot.trkset.

Examples

```
  ## Not run:
  trk <- getTrk(0.9, 0.002, 0.25, 1.7, 0)

  ### check return value from CDS
  if(!is.na(trk)[1]) plot(trk)
## End(Not run)
```

---

print.trk *Print stellar track objects*

Description

Format and print the contents of the objects of classes trk, hb, zahb, iso.

Usage

```
## S3 method for class 'trk'
print(x, ...)
## S3 method for class 'hb'
print(x, ...)
## S3 method for class 'zahb'
print(x, ...)
## S3 method for class 'iso'
print(x, ...)
```

Arguments

x	An object of classes trk, hb, zahb, or iso.
...	further arguments passed to low level print functions.

Details

The contents of the variables mass (or age), z, y, ml, alpha.enh (and massRGB if appliable) from classes trk, hb, zahb, or iso are fomatted and printed. The function returns its argument invisibly (via invisible(x)).

Examples

```
  ## Not run:
  trk <- getTrk(0.9, 0.002, 0.25, 1.7, 0)
  trk

## End(Not run)
```

---

showComposition *Show the chemical and physical combinations in the database*

Description

Show the values of chemical composition (Z, Y, AFE) and mixing-length present in the database.

Usage

```
showComposition()
```

Value

Print the combinations of the inputs that exist in the database.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

See Also

testComposition.

Examples

```
showComposition()
```

---

testComposition *Check the existence of a record in the database*

Description

Check that the given chemical composition (Z, Y, [alpha/Fe]) and the mixing-length value exist in the stellar database.

Usage

```
testComposition(Z, Y, ML, AFE)
```

Arguments

Z	the initial metallicity of the model.
Y	the initial helium abundance of the model.
ML	the mixing-length of the model.
AFE	the alpha-enhanchment of the model. Allowed values are `AFE = 0` ([alpha/Fe] = 0.0) or `AFE = 1` ([alpha/Fe] = 0.3).

Value

The function returns TRUE if the combination of the inputs exists in the database.

References

M. Dell'Omodarme, G. Valle, S. Degl'Innocenti, and P.G. Prada Moroni (2012). *The Pisa Stellar Evolution Data Base for low-mass stars*. Astronomy and Astrophysics, 540, A26.

See Also

showComposition.

Examples

```
testComposition(0.002, 0.25, 1.7, 0)
```

Index