S1$_A RD$ *Documentation*

Release 0.9

John Truckenbrodt, Tom Jones

Jan 18, 2022

Contents

CHAPTER 1

API Documentation

clc_legend	read clc meta data from the dedicated CSV file available here.
clc_prep	resample and crop the corine product to the resolution and extent of a reference image.
clc_prepare	create a CLC subset resampled to a reference image.
commonextent	compute the common extent of multiple extent dictionaries.
dem_aspect	compute the aspect of a DEM.
dem_degree2meter	compute the spatial resolution in meters for a DEM with WGS84 degree coordinates.
dem_distribution	create a polar slope-aspect DEM plot superimposed with the area visible to a SAR sensor.
dem_slope	compute the slope of a DEM.
dev_max	compute the maximum deviation from the median of all array values and the corresponding ID.
extent2patch	create a matplotlib rectangle patch from an extent dictionary.
parallel_apply_along_axis	Like `numpy.apply_along_axis()`, but takes advantage of multiple cores.
sampler	central function to select random samples from arrays.
scatter	general function for creating scatter plots.
uzh_prepare	create an UZH incident angle subset resampled to a reference image.
visible_sar_angle_map	create a SAR sensor slope-aspect visibility mask; used by *dem_distribution()*.
wkt2shp	convert a well-known text string geometry to a shapefile.

S1_ARD.util.**clc_legend**(*filename*)

read clc meta data from the dedicated CSV file available here.

Parameters **filename** (*str*) – the CSV file to be read

Returns the CSV values in a dictionary

Return type dict

S1_ARD.util.**clc_prep**(*clc*, *reference*, *outname*)
resample and crop the corine product to the resolution and extent of a reference image.

Parameters

- **clc** (*str*) – the name of the CLC input file
- **reference** (*str*) – the name of the reference file
- **outname** (*str*) – the named of the output image

S1_ARD.util.**clc_prepare**(*reference*, *outdir*, *source*)
create a CLC subset resampled to a reference image.

Parameters

- **reference** (*str*) – the reference file with the target CRS and extent
- **outdir** (*str*) – the directory to write the new file to; new files are named clc{index}.tif, e.g. clc1.tif.
- **source** (*str*) – the original product to be subsetted

Returns the name of the file written to *outdir*

Return type str

S1_ARD.util.**commonextent**(**args*)
compute the common extent of multiple extent dictionaries.

Parameters **args** (*dict*) – an extent dictionary, see e.g. spatialist.vector.Vector.extent

Returns the common extent

Return type dict

S1_ARD.util.**dem_aspect**(*img*)
compute the aspect of a DEM.

Parameters **img** (*numpy.ndarray*) – the DEM array

Returns the computed aspect array

Return type numpy.ndarray

S1_ARD.util.**dem_degree2meter**(*demfile*)
compute the spatial resolution in meters for a DEM with WGS84 degree coordinates.

Parameters **demfile** (*str*) – the DEM file

Returns (posting_east, posting_north)

Return type tuple

See also:

spatialist.auxil.haversine()

S1_ARD.util.**dem_distribution**(*slope*, *aspect*, *head_angle*, *inc_angle*, *look_dir='right'*, *nsamples=1000*, *title=''*, *mask=None*)
create a polar slope-aspect DEM plot superimposed with the area visible to a SAR sensor.

Parameters

- **slope** (*numpy.ndarray*) –
- **aspect** (*numpy.ndarray*) –
- **head_angle** (*float*) – the SAR sensor heading
- **inc_angle** (*float*) – the SAR sensor's incident angle
- **look_dir** (*str*) – the SAR sensor look direction; either *left* or *right*
- **nsamples** (*int*) – the number of samples to select from the *slope* and *aspect* arrays using function *sampler()*
- **title** (*str*) – the plot's title
- **mask** (*numpy.ndarray*) – an additional binary array to mask the slope and aspect values

See also:

visible_sar_angle_map()

S1_ARD.util.**dem_slope**(*img*, *xres_m*, *yres_m*)
compute the slope of a DEM.

Parameters

- **img** (*numpy.ndarray*) – the input DEM
- **xres_m** (*int or float*) – the x resolution of the DEM in same units as the height values
- **yres_m** (*int or float*) – the y resolution of the DEM in same units as the height values

S1_ARD.util.**dev_max**(*arr*)
compute the maximum deviation from the median of all array values and the corresponding ID.

Parameters **arr** (*numpy.ndarray*) – the 1D array

Returns (maximum deviation, ID)

Return type tuple

S1_ARD.util.**extent2patch**(*extent*, *edgecolor='r'*)
create a matplotlib rectangle patch from an extent dictionary.

Parameters

- **extent** (*dict*) – an extent dictionary, see e.g. spatialist.vector.Vector.extent
- **edgecolor** (*str*) – the edge color of the path

Returns

Return type matplotlib.patches.Rectangle

S1_ARD.util.**parallel_apply_along_axis**(*func1d*, *axis*, *arr*, *cores=4*, **args*, ***kwargs*)
Like numpy.apply_along_axis(), but takes advantage of multiple cores. Adapted from here.

Parameters

- **func1d** (*function*) – the function to be applied
- **axis** (*int*) – the axis along which to apply *func1d*
- **arr** (*numpy.ndarray*) – the input array
- **cores** (*int*) – the number of parallel cores

- **args** (*any*) – Additional arguments to *func1d*.
- **kwargs** (*any*) – Additional named arguments to *func1d*.

Returns

Return type numpy.ndarray

S1_ARD.util.**sampler**(*nanmask*, *nsamples=None*, *seed=42*)

central function to select random samples from arrays.

Parameters

- **nanmask** (*numpy.ndarray*) – a mask to limit the sample selection
- **nsamples** (*int*) – the number of samples to select
- **seed** (*int*) – seed used to initialize the pseudo-random number generator

Returns the generated random samples

Return type numpy.ndarray

See also:

`numpy.random.seed()`, `numpy.random.choice()`

S1_ARD.util.**scatter**(*x*, *y*, *z=None*, *xlab=''*, *ylab=''*, *title=''*, *nsamples=1000*, *mask=None*, *measures=None*, *regline=False*, *o2o=False*, *denscol=False*, *grid=False*, *xlim=None*, *ylim=None*, *sort_z=False*, *legend=False*, *regline_label='regression'*, *o2o_label='1-to-1'*)

general function for creating scatter plots.

Parameters

- **x** (*numpy.ndarray*) – dataset I
- **y** (*numpy.ndarray*) – dataset II
- **z** (*numpy.ndarray*) – dataset III for coloring the data points; overrides parameter *denscol*
- **xlab** (*str*) – the x-axis label
- **ylab** (*str*) – the y-axis label
- **title** (*str*) – the plot title
- **nsamples** (*int*) – the number of data samples to plot
- **mask** (*numpy.ndarray*) – an optional array for masking the datasets
- **measures** (*list*) –

  additional measures to be printed in a text box; current options:

  - *eq*: the linear regression equation
  - *rmse*
  - *r2*
  - *n*: the number of samples
  - *cv_x*, *cv_y*: the coefficient of variation of either *x* or *y*
  - *mean_x*, *mean_y*: the mean value of either *x* or *y*
- **regline** (*bool*) – draw a linear regression line?

- **o2o** (*bool*) – draw a data one-to-one line?
- **denscol** (*bool*) – color the points by Gaussian density?; overridden by parameter z
- **grid** (*bool*) – add a mesh grid to the plot?
- **xlim** (*tuple*) – the x-axis limits
- **ylim** (*tuple*) – the y-axis limits
- **sort_z** (*bool*) – if z is not None, sort its values so that points with high z values are plotted last?
- **legend** (*bool*) – add a legend for the regression line and one-to-one line if they exist?
- **regline_label** (*str*) – the legend label for the regression line
- **o2o_label** (*str*) – the legend label for the one-to-one line

S1_ARD.util.**uzh_prepare**(*reference*, *outdir*, *source*)
create an UZH incident angle subset resampled to a reference image.

Parameters

- **reference** (*str*) – the reference file with the target extent
- **outdir** (*str*) – the directory to write the new file to; new files are named uzh_{epsg}_{index}.tif, e.g. uzh_4326_1.tif.
- **source** (*str*) – the original product to be subsetted

Returns the content of the file written to *outdir*

Return type numpy.ndarray

S1_ARD.util.**visible_sar_angle_map**(*head_angle*, *inc_angle*, *look_dir='right'*)
create a SAR sensor slope-aspect visibility mask; used by *dem_distribution()*.

Parameters

- **head_angle** (*float*) – the SAR sensor heading
- **inc_angle** (*float*) – the SAR sensor's incident angle
- **look_dir** (*str*) – the SAR sensor look direction; either *left* or *right*

Returns the binary map with aspect-slope coordinates

Return type numpy.ndarray

S1_ARD.util.**wkt2shp**(*wkt*, *srs*, *outname*)
convert a well-known text string geometry to a shapefile.

Parameters

- **wkt** (*str*) – the well-known text description
- **srs** (*int, str*) – the spatial reference system; see spatialist.auxil.crsConvert() for options.
- **outname** (*str*) – the name of the shapefile to write

CHAPTER 2

Indices and tables

- genindex
- search

Python Module Index

S

Index