Documentation For CT_Analysis_Library

Nathan Hughes

April 9, 2018

Contents

Installation instructions:

To install globally run with sudo, else on a virtualenv just:

```
pip3 install .
```

Requirements

- Python3
- pandas
- numpy
- matplotlib
- seaborn
- scipy
- sklearn
- statsmodels
- pymc3
- xlrd

ct_data.py

NoDataFoundException

A custom exception for when there is no data in any of the scanned folders

CTData

This is a custom object made for holding CT Data

init

This is the initialise function for the CTData object, this will only need called once.

- **param** folder the folder to look for data in
- **param** rachis a boolean to decide to load in rachis data or not

get_data

Grabs the dataframe inside the class

- **returns** a copy of the dataframe used in this class

gather_data

this function gathers together all the data of interest

- **param** folder is a starting folder
- **returns** tuple of (seed files, rachis files)

create_dimensions_ratio

This is an additional phenotype which is of someuse for finding out the relationship between the dimension variales

make_dataframe

this function **returns** a dataframe of grain *param*eters and optionally of the rachis top and bottom

- **param** grain_files is the output from gather_data
- **param** rachis_files is an optional output from gather_data also
- **returns** a dataframe of the information pre-joining

clean_data

Following *param*eters outlined in the CT software documentation I remove outliers which are known to be errors

- **param** remove_small a boolean to remove small grains or not
- **param** remove_large a boolean to remove larger grains or not

get_files

Grabs a tuple of grain files and rachis

- **returns** a tuple of grain files and rachis files

fix_colnames

Because Biologists like to give data which are not normalised to any degree this function exists to attempt to correct the grouping columns, after standardisation https://github.com/SirSharpest/CT_Analysing_Library/issues/2 this shouldn't be needed anymore, but kept for legacy issues that could arise!

join_spikes_by_rachis

So important part of this function is that we accept that the data is what it is that is to say: rtop, rbot and Z are all orientated in the proper direction

It's main purpose is to join split spikes by rachis nodes identified in the analysis process

- **param** grain_df is the grain dataframe to take on-board

remove_percentile

This function is targeted at removing a percentile of a dataframe it uses a column to decide which to measure against. By default this will remove everything above the percentile value

- **param** df is the dataframe to manipulate
- **param** column is the attribute column to base the removal of
- **param** target_percent is the percentage to aim for
- **param** bool_below is a default **param** which if set

to True will remove values below rather than above percentage

get_spike_info

This function should do something akin to adding additional information to the data frame

- note there is some confusion in the NPPC about whether to use

folder name or file name as the unique id when this is made into end-user software, a toggle should be added to allow this

- **param** excel_file a file to attach and read data from
- **param** join_column if the column for joining data is different then it should be stated

look_up

gather_data

aggregate_spike_averages

This will aggregate features (specified by attributes) into their medians on a per-spike basis.

Makes direct changes to the dataframe (self.df)

- **param** attributes list of features to average
- **param** groupby how the data should be aggregated

find_troublesome_spikes

This will attempt to identify spikes which are not performing as expected

The default criteria for this is simply a count check So it requires that aggregate_spike_averages has been run

- **returns** a dataframe with candidates for manual investigation

data_transforms.py

box_cox_data

The powers or Box_Cox transform

Appears to be something which could really help with the kind of skewed data which this library seeks to assist with.

- **param** values_array a numpy array of numbers to be transformed

standarise_data

This is to conform with the likes of PCA, the following text is borrowed from: https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60 Some code also also heavily borrowed from this page and I take minimal credit for it.

PCA is effected by scale so you need to scale the features in your data before applying PCA. Use StandardScaler to help you standardize the datasets features onto unit scale (mean = 0 and variance = 1) which is a requirement for the optimal performance of many machine learning algorithms. If you want to see the negative effect not scaling your data can have, scikit-learn has a section on the effects of not standardizing your data.

uses: $\frac{x_i mean(x)}{stdev(x)} assumes Normal Distribution$

To try and fit a normal distribution I am applying log scales of log_2

- **param** df the data to be standarised
- **param** features the list of features to standardise
- **param** groupby how the columns should be grouped
- **returns** scaled values

perform_pca

This function will perform a PCA and return the principle components as a dataframe.

Read this for more information

- **param** n_components components to check form
- **param** df dataframe of the data to analyse
- **param** features features from the dataframe to use
- **param** groupby the column in the df to use
- **param** standardise=False asks whether to standardise the data prior to PCA
- **returns** a dataframe of the data, the pca object and the scaled data for reference

pca_to_table

Creates a dataframe of the PCA weights for each attribute

https://stackoverflow.com/questions/22984335/recovering-features-names-of-explained-variance-ratio-in-p

- **returns** a pca table

graphing.py

InvalidPlot

Except to trigger when a graph is given wrong args

plot_difference_of_means

Plots a difference of means graph

- **param** trace a trace object
- **param** **kwargs keyword arguments for matplotlib
- **returns** a plot axes with the graph plotted

plot_forest_plot

Plots a forest plot

- **param** trace a trace object
- **param** name1 the name of the first group
- **param** name2 the name of the second group
- **returns** a forestplot on a gridspec

plot_boxplot

This should just create a single boxplot and return the figure and an axis, useful for rapid generation of single plots Rather than the madness of the plural function

Accepts Kwargs for matplotlib and seaborn

- **param** data a CTData object or else a dataframe
- **param** attribute the attribute to use in the boxplot
- **param** **kwargs keyword arguments for matplotlib
- **returns** a figure and axes

plot_qqplot

What's a QQ plot? https://stats.stackexchange.com/questions/139708/qq-plot-in-python

- **param** vals the values to use in the qqplot
- **param** plot the plot to place this on

plot_histogram

Simple histogram function which accepts seaborn and matplotlib kwargs **returns** a plot axes

- **param** data a CTData object or else a dataframe
- **param** attribute the attribute to use in the histogram
- **param** **kwargs keyword arguments for matplotlib
- **returns** an axes

plot_pca

Plots the PCA of the data given in a 2D plot

- **param** pca the pca object
- **param** dataframe the dataframe from the pca output
- **param** groupby the variable to group by in the plot
- **param** single_plot a boolean to decide to multiplot or not
- return a seaborn plot object

check_var_args

Helper function to fix bad arguments before they get used in evaluations

- **param** arg arguments to check if fine or not

statistical_tests.py

baysian_hypothesis_test

Implements and uses the hypothesis test outlined as a robust replacement for the t-test

for reference http://www.indiana.edu/~kruschke/BEST/BEST.pdf

- **param** group1 a numpy array to test
- **param** group2 a numpy array to test
- **param** group1_name the name of the first group
- **param** group2_name the name of the second group
- **returns** a summary dataframe

check_normality

https://stackoverflow.com/a/12839537

Null Hypothesis is that X came from a normal distribution

which means: If the p-val is very small, it means it is unlikely that the data came from a normal distribution

As for chi-square: chi or ttest?

- **param** vals the values to test for normality
- **returns** a boolean to indicate if normal or not

perform_t_test

Performs the standard t-test and **returns** a p-value

- **param** group1 the first group to compare
- **param** group2 the second group to compare
- **returns** a p-value of the ttest