pysumstats

Release 0.5.2

Jan 14, 2021

Contents

pysumstats, a python package for working with GWAS summary statistics

CHAPTER 1

SumStats

class pysumstats.**SumStats**(*path*, *phenotype=None*, *gwas_n=None*, *column_names=None*, *data=None*, *low_ram=False*, *tmpdir='sumstats_temporary'*, ***kwargs*)

Class for summary statistics of a single GWAS.

Parameters

- **path** (*str*) – Path to the file containing summary statistics. Should be a csv, or tab-delimited txt-file (.gz supported).
- **phenotype** (*str*) – Phenotype name
- **gwas_n** (*int*) – Optional N subjects in the GWAS, for N-based meta analysis (if there is no N column in the summary statistics)
- **column_names** (*dict*) – Optional dictionary of column names, if these are not automatically recognised. Keys should be: ['rsid', 'chr', 'bp', 'ea', 'oa', 'maf', 'b', 'se', 'p', 'hwe', 'info', 'n', 'eaf', 'oaf']
- **data** (*dict*) – Dataset for the new SumStats object, in general, don't specify this.
- **low_ram** (*bool*) – Whether to use the low_ram option for this SumStats object. Use this only when running into MemoryErrors. Enabling this option will read/write data from local storage rather then RAM. It will save lots of RAM, but it will significantly decrease processing speed.
- **tmpdir** (*str*) – Which directory to store the temporary files if low_ram is enabled.
- **kwargs** – other keyword arguments to be passed to pandas.read_csv() method

close()

Close connection to and HDF5 file if low_ram is specified

Returns None

copy()

Returns a deepcopy of the existing opject

describe (*columns=None*, *per_chromosome=False*)
Get a summary of the data.

Parameters

- **columns** (*list.*) – List of column names to print summary for (default: ['b', 'se', 'p'])
- **per_chromosome** (*bool.*) – Enable to return a list of summary dataframes per chromosome

Returns pd.Dataframe, or list

groupby (**args*, ***kwargs*)
Compatibility function to create pandas grouped object

Parameters

- **args** – arguments to be passed to pandas groupby function
- **kwargs** – keyword arguments to be passed to pandas groupby function

Returns a full grouped pandas dataframe object

head (*n=10*, *n_chromosomes=1*, ***kwargs*)
Prints (n_chromosomes) dataframes with the first n rows.

Parameters

- **n** (*int*) – number of rows to show
- **n_chromosomes** (*int*) – number of chromosomes to show.
- **kwargs** – keyword arguments to be passed to pandas head function

Returns None

manhattan (***kwargs*)
Generate a manhattan plot using this sumstats data

Parameters **kwargs** – keyworded arguments to be passed to *pysumstats.plot.manhattan()*

Returns None, or (fig, ax)

merge (*other*, *how='inner'*, *low_memory=False*)
Merge with other SumStats object(s).

Parameters

- **other** (`pysumstats.plot.SumStats` or list) – Other sumstats object, or list of other SumStats objects.
- **how** (*str*) – Type of merge.
- **low_memory** (*bool*) – Enable to use a more RAM-efficient merging method (WARNING: still untested)

Returns `pysumstats.plot.MergedSumStats` object

plot_all (*dest='.'*, *prefix='SumStatsPlots'*, *kwargs={}*)
Runs all attached plot functions

Parameters

- **dest** (*str*) – Folder to save resulting files to. File names will be: {prefix}_{plottype}_{YEAR-MONTH-DAY}.png
- **prefix** (*str*) – prefix to use when saving files.

- **kwargs** (*dict*) – Nested dictionary of other keyword arguments to be passed to each function (keys of top-level dictionary should be function names). Use the 'all' key the top level dictionary to pass keyword argument to every function.

Returns None

pzplot(***kwargs*)
Generate a PZ-plot using this sumstats data

Parameters **kwargs** – keyworded arguments to be passed to *pysumstats.plot.pzplot()*

Returns None, or (fig, ax)

qc(*maf=None*, *hwe=None*, *info=None*, ***kwargs*)
Basic GWAS quality control function.

Parameters

- **maf** (*float or None*) – Minor allele frequency cutoff, will drop SNPs where MAF < cutoff. Default: 0.1
- **hwe** (*float or None*) – Hardy-Weinberg Equilibrium cutoff, will drop SNPs where HWE < cutoff, if specified and HWE column is present in the data.
- **info** (*float or None*) – Imputation quality cutoff, will drop SNPs where Info < cutoff, if specified and Info column is present in the data.
- **kwargs** – Other columns to filter on, keyword should be column name, SNPs whill be dropped where the value < argument.

Returns None

qqplot(***kwargs*)
Generate a QQ-plot using this sumstats data

Parameters **kwargs** – keyworded arguments to be passed to *pysumstats.plot.qqplot()*

Returns None, or (fig, ax)

reset_index()
Reset the index of the data.

Returns None

save(*path*, *per_chromosome=False*, *per_phenotype=False*, *phenotype=None*, ***kwargs*)
Save the data held in this object to local storage.

Parameters

- **path** (*str*) – Relative or full path to the target file to store the data or object in. Paths ending in .pickle will save a pickled version of the full object. Note that with low_ram enabled this will **not** store the data. When per_phenotype is specified, add {} to the path where the phenotype name should be, if {} is not in the string, the filename will be prefixed with phenotype name.
- **per_chromosome** (*bool*) – Whether to save seperate files for each chromosome.
- **per_phenotype** – Set to True to create a separate file for each phenotype in MergedSumStats objects

:type per_phenotype :param phenotype: Only save a file for a specifici phenotype in MergedSumstats objects :type phenotype: str :param kwargs: keyword arguments to be passed to pandas to_csv() function. :return: None

sort_values(*by*, *inplace=True*, ***kwargs*)
Sorts values in the dataframe. Note: Sorting by chromosme (chr) will have no effect as data is already structured by chromosome.

Parameters

- **by** (*str*) – label of the column to sort values by
- **inplace** (*bool*) – Whether to return the sorted object or sort values within existing object. (Currently only inplace sorting is supported)
- **kwargs** – Other keyword arguments to be passed to pandas sort_values function

Returns Non

tail(*n=10*, *n_chromosomes=1*, ***kwargs*)
Prints (n_chromosomes) dataframes with the last n rows.

Parameters

- **n** (*int*) – number of rows to show
- **n_chromosomes** (*int*) – number of chromosomes to show.
- **kwargs** – keyword arguments to be passed to pandas tail function

Returns None

CHAPTER 2

MergedSumStats

class pysumstats.**MergedSumStats**(*data*, *phenotypes*, *merge_info*, *variables*, *xy*, *low_ram=False*, *tmpdir='sumstats_temporary'*, *allign=True*)

Class containing merged summary statistics. In general you will not create a MergedSumStats object manually.

Parameters

- **data** (*dict*) – dataset containing merged summary statistics
- **phenotypes** (*list*) – list of phenotype names.
- **merge_info** (*dict*) – Dict with information on the merge
- **variables** (*list*) – list of variables contained in the data.
- **xy** (*list*) – x and y suffixes (to be used in _allign)
- **low_ram** (*bool*) – Whether to use the low_ram option for this MergedSumStats object (passed down from SumStats). Use this only when running into MemoryErrors. Enabling this option will read/write data from local storage rather then RAM. It will save lots of RAM, but it will gratly decrease processing speed.
- **tmpdir** (*str*) – Which directory to store the temporary files if low_ram is enabled (passed down from SumStats).
- **allign** (*bool*) – Enable to auto-allign SNPs

afplot(*ref_phenotypes=None*, *other_phenotypes=None*, *filename=None*, *nrows=None*, *ncols=None*, *figsize=None*, *dpi=300*, ***kwargs*)

Generates AF comparison plots for merged GWAS data.

Parameters

- **ref_phenotypes** (*list*) – List of phenotypes to use as reference (defaults to all phenotypes)
- **other_phenotypes** (*list*) – List of phenotypes to compare reference to (defaults to all phenotypes, overlapping plots will be dropped)

- **filename** (*str*) – Target file to save the resulting figure to (if no name is specified, fig and axes are returned)
- **nrows** (*int*) – Specify number of rows in the figure (defaults to int(ceil(n_plots/ncols)))
- **ncols** (*int*) – Specify number of columns in the figure (defaults to int(sqrt(n_plots)))
- **figsize** (*(int, int)*) – Specify width and height of figure in inches (defaults to (ncols*5, nrows*5))
- **dpi** (*int*) – DPI setting to use when saving the figure.
- **kwargs** – Other keyword arguments to be passed to *pysumstats.plot.manhattan()*

Returns (fig, axes) or None.

close()
Close connection to and HDF5 file if low_ram is specified

Returns None

copy()

Returns a deepcopy of the existing opject

describe(*columns=None*, *per_chromosome=False*)
Get a summary of the data.

Parameters

- **columns** (*list*) – List of column names to print summary for (default: [‘b’, ‘se’, ‘p’])
- **per_chromosome** (*bool*) – Enable to return a list of summary dataframes per chromosome

Returns pd.Dataframe, or list of pd.Dataframes

groupby(**args*, ***kwargs*)
Compatibility function to create pandas grouped object

Parameters

- **args** – arguments to be passed to pandas groupby function
- **kwargs** – keyword arguments to be passed to pandas groupby function

Returns a full grouped pandas dataframe object

gwama(*cov_matrix=None*, *h2_snp=None*, *name='gwama'*)
Multivariate meta analysis as described in Baselmans, et al. 2019.

Parameters

- **cov_matrix** (*pd.Dataframe*) – Covariance matrix, defaults to generating a correlation matrix of Z-scores
- **h2_snp** (*dict*) – Dict of SNP heritabilities per GWAS, to use as additional weights. Defaults to all 1’s.
- **name** – New phenotype name to use in the new SumStats object (default: ‘gwama’)

Returns *pysumstats.SumStats* object

head(*n=10*, *n_chromosomes=1*, ***kwargs*)
Prints (n_chromosomes) dataframes with the first n rows.

Parameters

- **n** (*int*) – number of rows to show
- **n_chromosomes** (*int*) – number of chromosomes to show.
- **kwargs** – keyword arguments to be passed to pandas head function

Returns None

manhattan(*filename=None*, *phenotypes=None*, *nrows=None*, *ncols=None*, *figsize=None*, *dpi=300*, ***kwargs*)

Generates manhattan plots for each phenotype (or specified phenotypes) in merged GWAS data.

Parameters

- **filename** (*str*) – Target file to save the resulting figure to (if no name is specified, fig and axes are returned)
- **phenotypes** (*list*) – List of phenotype names to plot manhattans for (defaults to plotting all phenotypes)
- **nrows** (*int*) – Specify number of rows in the figure (defaults to int(ceil(len(phenotypes)/ncols)))
- **ncols** (*int*) – Specify number of columns in the figure (defaults to int(log2(len(phenotypes)/2)))
- **figsize** (*(int, int)*) – Specify width and height of figure in inches (defaults to (ncols*8, nrows*4))
- **dpi** (*int*) – DPI setting to use when saving the figure.
- **kwargs** – Other keyword arguments to be passed to *pysumstats.plot.manhattan()*

Returns (fig, axes) or None.

merge(*other*, *inplace=False*, *how='inner'*, *low_memory=False*)

Merge with other SumStats or MergedSumstats object(s).

Parameters

- **other** (*pysumstats.SumStats*, *pysumstats.MergedSumStats*, or list.) – *pysumstats.SumStats*, or *pysumstats.MergedSumStats* object, or a list of SumStats, MergedSumstats objects
- **inplace** (*bool*) – Enable to store the new data in the current MergedSumStats object. (currently not supported when low_ram is enabled)
- **how** (*str*) – Type of merge, for now only implemented for merges with *pysumstats.SumStats* objects
- **low_memory** (*bool*) – Enable to use a more RAM-efficient merging method (WARNING: still untested)

Returns None, or *pysumstats.MergedSumStats* object.

meta_analyze(*name='meta'*, *method='ivw'*, *debug=False*)

Meta analyze all GWAS summary statistics contained in this object. WARNING: There appears to be an error somewhere in this function that causes incorrect result. For now running .meta_analyze() will instead run .gwama() with an identity matrix (functionally identical to an ivw meta_analysis) :param name: New phenotype name to use for the new SumStats object (default: 'meta') :type name: str :param method: Meta-analysis method to use, should be one of ['ivw', 'samplesize'], default: 'ivw' :type method: str

:param debug: Run the meta_analyze function instead of .gwama() for debugging purposes :type debug: bool :return: `pysumstats.SumStats` object.

plot_all (*dest='.'*, *prefix='SumStatsPlots'*, *kwargs={}*)
Runs all attached plot functions

Parameters

- **dest** (*str*) – Folder to save resulting files to. File names will be: {prefix}_{plottype}_{YEAR-MONTH-DAY}.png
- **prefix** (*str*) – prefix to use when saving files.
- **kwargs** (*dict*) – Nested dictionary of other keyword arguments to be passed to each function (keys of top-level dictionary should be function names). Use the 'all' key the top level dictionary to pass keyword argument to every function.

Returns None

prep_for_mr (*exposure*, *outcome*, *filename=None*, *p_cutoff=None*, *bidirectional=False*, ***kwargs*)
Save a pre-formatted file to use with the MendelianRandomization package in R.

Parameters

- **exposure** (*str*) – phenotype name to use as exposure.
- **outcome** (*str*) – phenotype name to use as outcome.
- **filename** (*str, list or None*) – Path to where the resulting file(s) should be stored, or list of paths if bidirectional=True
- **p_cutoff** (*float*) – Optional p-value cut-off to apply. Will include SNPs where P > p_cutoff
- **bidirectional** (*bool*) – Enable to store two files (exposure=exposure, outcome=outcome), and (exposure=outcome, outcome=exposure)
- **kwargs** – Additional keyword arguments to be passed to pandas to_csv function.

Returns None

pzplot (*filename=None*, *phenotypes=None*, *nrows=None*, *ncols=None*, *figsize=None*, *dpi=300*, ***kwargs*)
Generates PZ-plots for each phenotype (or specified phenotypes) in merged GWAS data.

Parameters

- **filename** (*str*) – Target file to save the resulting figure to (if no name is specified, fig and axes are returned)
- **phenotypes** (*list*) – List of phenotype names to plot PZ-plots for (defaults to plotting all phenotypes)
- **nrows** (*int*) – Specify number of rows in the figure (defaults to int(sqrt(len(phenotypes))))
- **ncols** (*int*) – Specify number of columns in the figure (defaults to int(ceil(len(phenotypes)/nrows)))
- **figsize** (*(int, int)*) – Specify width and height of figure in inches (defaults to (ncols*5, nrows*5))
- **dpi** (*int*) – DPI setting to use when saving the figure.
- **kwargs** – Other keyword arguments to be passed to `pysumstats.plot.pzplot()`

Returns (fig, axes) or None.

qqplot (*filename=None*, *phenotypes=None*, *nrows=None*, *ncols=None*, *figsize=None*, *dpi=300*, ***kwargs*)
Generates QQ-plots for each phenotype (or specified phenotypes) in merged GWAS data.

Parameters

- **filename** (*str*) – Target file to save the resulting figure to (if no name is specified, fig and axes are returned)
- **phenotypes** (*list*) – List of phenotype names to plot QQ-plots for (defaults to plotting all phenotypes)
- **nrows** (*int*) – Specify number of rows in the figure (defaults to int(sqrt(len(phenotypes))))
- **ncols** (*int*) – Specify number of columns in the figure (defaults to int(ceil(len(phenotypes)/nrows)))
- **figsize** (*(int, int)*) – Specify width and height of figure in inches (defaults to (ncols*5, nrows*5))
- **dpi** (*int*) – DPI setting to use when saving the figure.
- **kwargs** – Other keyword arguments to be passed to *pysumstats.plot.qqplot()*

Returns (fig, axes) or None.

reset_index ()
Reset the index of the data.

Returns None

save (*path*, *per_chromosome=False*, *per_phenotype=False*, *phenotype=None*, ***kwargs*)
Save the data held in this object to local storage.

Parameters

- **path** (*str*) – Relative or full path to the target file to store the data or object in. Paths ending in .pickle will save a pickled version of the full object. Note that with low_ram enabled this will **not** store the data. When per_phenotype is specified, add {} to the path where the phenotype name should be, if {} is not in the string, the filename will be prefixed with phenotype name.
- **per_chromosome** (*bool*) – Whether to save seperate files for each chromosome.
- **per_phenotype** – Set to True to create a separate file for each phenotype in MergedSumStats objects

:type per_phenotype :param phenotype: Only save a file for a specifici phenotype in MergedSumstats objects :type phenotype: str :param kwargs: keyword arguments to be passed to pandas to_csv() function. :return: None

sort_values (*by*, *inplace=True*, ***kwargs*)
Sorts values in the dataframe. Note: Sorting by chromosme (chr) will have no effect as data is already structured by chromosome.

Parameters

- **by** (*str*) – label of the column to sort values by
- **inplace** (*bool*) – Whether to return the sorted object or sort values within existing object. (Currently only inplace sorting is supported)
- **kwargs** – Other keyword arguments to be passed to pandas sort_values function

Returns Non

tail (*n=10*, *n_chromosomes=1*, ***kwargs*)
Prints (n_chromosomes) dataframes with the last n rows.

Parameters

- **n** (*int*) – number of rows to show
- **n_chromosomes** (*int*) – number of chromosomes to show.
- **kwargs** – keyword arguments to be passed to pandas tail function

Returns None

zzplot (*ref_phenotypes=None*, *other_phenotypes=None*, *filename=None*, *nrows=None*, *ncols=None*, *figsize=None*, *dpi=300*, ***kwargs*)
Generates ZZ comparison plots for merged GWAS data.

Parameters

- **ref_phenotypes** (*list*) – List of phenotypes to use as reference (defaults to all phenotypes)
- **other_phenotypes** (*list*) – List of phenotypes to compare reference to (defaults to all phenotypes, overlapping plots will be dropped)
- **filename** (*str*) – Target file to save the resulting figure to (if no name is specified, fig and axes are returned)
- **nrows** (*int*) – Specify number of rows in the figure (defaults to int(ceil(n_plots/ncols)))
- **ncols** (*int*) – Specify number of columns in the figure (defaults to int(sqrt(n_plots)))
- **figsize** (*(int, int)*) – Specify width and height of figure in inches (defaults to (ncols*5, nrows*5))
- **dpi** (*int*) – DPI setting to use when saving the figure.
- **kwargs** – Other keyword arguments to be passed to *pysumstats.plot.zzplot()*

Returns (fig, axes) or None.

CHAPTER 3

Utilities

pysumstats.**cov_matrix_from_phenotype_file**(*dataframe*, *phenotypes=None*)

Function to generate a covariance (cov_Z) matrix from a phenotype file.

Parameters

- **dataframe** (*pd.Dataframe*) – pd.Dataframe containing phenotypic data
- **phenotypes** (*list.*) – list of phenotypes to include

Returns pd.Dataframe of covariance matrix.

CHAPTER 4

pysumstats.plot

pysumstats.plot, a subpackage for generating plots from summary statistics data

4.1 Manhattan plot

plot.**manhattan**(*fig=None, ax=None, filename=None, sigp=5e-08, sigcolor='black', sugp=1e-05, sugcolor='black', pointcolors=['midnightblue', 'goldenrod'], figsize=(12, 6), highlight=[], highlightcolors=['orange'], title=None, rainbow=False*)

Create a Manhattan plot.

Parameters

- **dataframe** – pd.Dataframe containing the following columns: ['rsid', 'chr', 'bp', 'p'], or *pysumstats.SumStats*
- **fig** – matplotlib.pyplot figure object to plot to (if not specified a new figure will be created)
- **ax** – matplotlib.pyplot axis to plot to (if not specified a new figure will be created)
- **filename** (*str.*) – Path to store the figure to (defaults to return fig, ax objects)
- **sigp** (*float*) – Where to plot significant line(set to a negative number to remove)
- **sigcolor** (*str*) – Color to use for significant line
- **sugp** (*float*) – Where to plot significant line (set to a negative number to remove)
- **sugcolor** (*str*) – Color to use for suggestive line
- **pointcolors** (*list*) – List of colors to cycle through for plotting SNPs
- **figsize** (*(float, float)*) – Figure size in inches (width, height)
- **highlight** (*list.*) – list of SNPs to highlight
- **highlightcolors** (*list.*) – List of colors to cycle through for highlighting SNPs
- **title** (*list.*) – Main figure title

- **rainbow** (*bool.*) – Enble rainbow colors

Returns None, or (fig, ax)

4.2 QQ plot

plot.**qqplot**(*fig=None*, *ax=None*, *filename=None*, *figsize=(8, 8)*, *pointcolor='black'*, *title=None*, *linecolor='red'*)

Function to generate a QQ-plot.

Parameters

- **pvector** – 1D-array of p-values
- **fig** – matplotlib.pyplot figure object to plot to (if not specified a new figure will be created)
- **ax** – matplotlib.pyplot axis to plot to (if not specified a new figure will be created)
- **filename** (*str.*) – Path to store the figure to (defaults to return fig, ax objects)
- **figsize** (*(float, float)*) – Figure size in inches (width, height)
- **pointcolor** (*str.*) – Color to use for points
- **title** (*str.*) – Main figure title.
- **linecolor** (*str.*) – Color for line x=y

Returns None, or (fig, ax)

4.3 PZ plot

plot.**pzplot**(*twotailed=True*, *difference_cutoff=0.1*, *fig=None*, *ax=None*, *filename=None*, *pointcolor='black'*, *differentcolor='red'*, *linecolor='black'*, *differentlinecolor='red'*, *title=None*, *figsize=(5, 5)*)

Generate a plot comparing the z-value as calculated from the p-value to the z-value as calculated from beta/se

Parameters

- **data** – 2D-array containing the columns ['b', 'se', 'p']
- **twotailed** (*bool*) – True if p-value was computed from both ends of the distribution.
- **difference_cutoff** (*None or float*) – Cut-off to use for highlighting SNPs with different z-values (to disable use None)
- **fig** – matplotlib.pyplot figure object to plot to (if not specified a new figure will be created)
- **ax** – matplotlib.pyplot axis to plot to (if not specified a new figure will be created)
- **filename** (*str*) – Path to store the figure to (defaults to return fig, ax objects)
- **pointcolor** (*str*) – Color to use for points
- **differentcolor** (*str*) – Color to use for points that deviate given difference_cutoff
- **linecolor** (*str*) – Color to use for the line x=y
- **differentlinecolor** (*str*) – Color to use for visualizing the difference_cutoff
- **title** (*str.*) – Main figure title.
- **figsize** (*(int, int)*) – Figure size

Returns None or (fig, ax)

4.4 AF plot

plot.**afplot**(*other_af*, *refname='ref_EAF'*, *othername='other_EAF'*, *difference_cutoff=0.1*, *fig=None*, *ax=None*, *filename=None*, *pointcolor='black'*, *differentcolor='red'*, *linecolor='black'*, *differentlinecolor='red'*, *title=None*, *figsize=(5, 5)*)

Generate a plot of (differences in) allele frequencies of two summary statistics.

Parameters

- **ref_af** – 1D-array of reference allele frequencies
- **other_af** – 1D-array of other allele frequencies
- **refname** (*str*) – Name to use for reference allele frequencies (x-axis label)
- **othername** (*str*) – Name to use for other allele frequencies (y-axis label)
- **difference_cutoff** (*None or float*) – Cut-off to use for highlighting SNPs with different allele frequency (to disable use None)
- **fig** – matplotlib.pyplot figure object to plot to (if not specified a new figure will be created)
- **ax** – matplotlib.pyplot axis to plot to (if not specified a new figure will be created)
- **filename** (*str*) – Path to store the figure to (defaults to return fig, ax objects)
- **pointcolor** (*str*) – Color to use for points
- **differentcolor** (*str*) – Color to use for points that deviate given difference_cutoff
- **linecolor** (*str*) – Color to use for the line x=y
- **differentlinecolor** (*str*) – Color to use for visualizing the difference_cutoff
- **title** (*str.*) – Main figure title.
- **figsize** ((*int, int*)) – Figure size

Returns None or (fig, ax)

4.5 ZZ plot

plot.**zzplot**(*data_y*, *xname='Z_x'*, *yname='Z_y'*, *twotailed=True*, *difference_cutoff=0.5*, *fig=None*, *ax=None*, *filename=None*, *pointcolor='black'*, *differentcolor='red'*, *linecolor='black'*, *differentlinecolor='red'*, *title=None*, *figsize=(5, 5)*)

Generate a plot comparing the z-values from two GWAS summary statistics

Parameters

- **data_x** – 2D-array containing the column 'z', the columns ['b', 'se'] or the column 'p' (this is priority order)
- **data_y** – 2D-array containing the column 'z', the columns ['b', 'se'] or the column 'p' (this is priority order)
- **twotailed** (*bool*) – True if p-value was computed from both ends of the distribution.
- **xname** (*str*) – Name to use for z_values of data_x (x-axis label)
- **yname** (*str*) – Name to use for z_values of data_y (y-axis label)

- **difference_cutoff** (*None or float*) – Cut-off to use for highlighting SNPs with different z-values (to disable use None)
- **fig** – matplotlib.pyplot figure object to plot to (if not specified a new figure will be created)
- **ax** – matplotlib.pyplot axis to plot to (if not specified a new figure will be created)
- **filename** (*str*) – Path to store the figure to (defaults to return fig, ax objects)
- **pointcolor** (*str*) – Color to use for points
- **differentcolor** (*str*) – Color to use for points that deviate given difference_cutoff
- **linecolor** (*str*) – Color to use for the line x=y
- **differentlinecolor** (*str*) – Color to use for visualizing the difference_cutoff
- **title** (*str.*) – Main figure title.
- **figsize** (*(int, int)*) – Figure size

Returns None or (fig, ax)

CHAPTER 5

Guides

5.1 Opening and saving sumstats files

5.1.1 Opening sumstats files

From csv/txt/tsv

pysumstats will automatically recognize csv/txt/tsv extensions and read files appropriately. It assumes a .txt or .tsv files is separated by tabs, and .csv files are separated by commas. If your file is separated by a character other then the default, please specify *sep* when opening SumStats. For example, a European csv file can be opened with `pysumstats.SumStats` *('myfile.csv', sep=';')*

From gzipped files

Since pysumstats uses `pandas.read_csv()` to open files the *.gz* extension is automatically recognized and will be opened appropriately. If you are using a different type of compression see the documentation of `pandas.read_csv()` function to check wether your comprression is supported.

From pickled files

When you have saved your `pysumstats.SumStats` object as a *.pickle* file you can open it as you would any other file. Loading `pysumstats.MergedSumStats` objects from a *.pickle* file will not work by running *pysumstats.SumStats('mymerged.pickle')*. To open `pysumstats.MergedSumStats` objects stored as *.pickle* please run the following:

```
import pickle
with open('mymerged.pickle', 'rb') as f:
    my_merged_obj = pickle.load(f)
```

Warning: When a *pysumstats.SumStats* with *low_ram* enabled is storted as a *.pickle* file the data is **not** included in the file, only the reference to the temporary file!

When *low_ram* is enabled I highly recommend to save your file as a .csv or something similar.

From other files

Any file extension other then the following: *['.txt.', '.txt.gz', '.tsv', '.tsv.gz', '.csv', '.csv.gz', '.pickle']* will be passed straight to `pandas.read_csv()` hence you will have to supply the appropriate keyworded arguments to enable reading.

5.1.2 Saving sumstats files

Compatibility method

If you want to use the saved data in any other package or program you should save it in plaintext format (or gzipped plain text). Both *pysumstats.SumStats* and *pysumstats.MergedSumStats* objects support saving to *.csv, .txt(.gz), .tsv(.gz)* using the same default separators as for reading (commas for .csv, tabs for .txt and .tsv). Supplying the *sep* argument will override these defaults. Attempting to save with any other file extension will raise a *KeyError* .. note:

```
The `index` and `header` arguments are used internally and cannot be set.
```

Fastest method

If you want to continue working on your file with the *pysumstats* package later it is much faster to save your object as a pickled file. Both *pysumstats.SumStats* and *pysumstats.MergedSumStats* objects support saving to a *.pickle* file from their respective *.save()* methods. Alternatively files can be pickled in the standard way:

```
import pickle
with open('mypickledsumstats.pickle', 'wb') as f:
    pickle.load(f, my_sumstats_obj)
```

Warning: When a *pysumstats.SumStats* or *pysumstats.MergedSumStats* with *low_ram* enabled is storted as a *.pickle* file the data is **not** included in the file, only the reference to the temporary file!

When *low_ram* is enabled I highly recommend to save your file as a .csv or something similar.

Per chromosome

Alternatively, data can be stored in 1 file per chromosome by passing *per_chromsome=True* to *pysumstats.SumStats.save()* or *pysumstats.SumStats.save()*. By default files will be stored as: *chr1_[path]*, *chr2_[path]*, etc. Alternatively you can add *{}* to the path where you want the chromsomenumbers to be in the files. For example: *my_sumstats_obj.save('my_data_chr{}.csv', per_chromosome=True)*.

Python Module Index

p

Index