# genda Documentation

***Release 0.1***

**Jeffrey Hsu**

January 21, 2016

Contents

**Date**: January 21, 2016 **Version**: 0.1

**Code Repository:** http://github.com/jeffhsu3/genda

**genda** is Python package for working with genetic and genomic data using the python dataframe package pandas.

Contents:

CHAPTER 1

# Installing genda

Working on a package atm.

## 1.1 Installing from source

The source is available from github:: git clone https://github.com/jeffhsu3/genda.git python setup.py install

## 1.2 Binary Installers

Work in progress

## 1.3 Dependencies

- 'python <http://www.python.org>'_
- 'numpy <http://http://www.numpy.org/'_
- 'cython <http://www.cython.org/'_
- 'scipy <http://www.scipy.org/'_
- 'matplotlib] <http://matplotlib.org/>'_
- 'pandas <http://pandas.pydata.org/>'_
- 'pysam <https://code.google.com/p/pysam/>'_
- 'bx-python <https://bitbucket.org/james_taylor/bx-python/wiki/Home>'_

Installing genda is easy given you have all the dependencies. pySam is a must and is a great tool for working with BAM files in python.

If you have pip, all the dependencies will be installed for you. Just run python setup.py install.

CHAPTER 2

# Introduction

genda provides an easy way to load different formats of genetic data and analyze them.

CHAPTER 3

# Related Software

## 3.1 Biology

- biopython

CHAPTER 4

# Exporting Data to R

CHAPTER 5

# Working With VCFs

## 5.1 The VCF Class

**genda.formats.panVCF.VCF(vcf_file)**

**parameters:** vcf_file - The VCF file which you want to load into panVCF chunksize - pandas.read_csv chunksize DOESN'T WORK ATM as parsing info expects a pd.DataFrame

### 5.1.1 Data

**VCF.vcf** Pandas dataframe of vcf data, excluding headers

**VCF.geno** Pandas dataframe of genotype data

**VCF.head()** Returns the headers for the vcf

### 5.1.2 Hardy-Weinberg

VCF.hardyweinberg(snp, excludeNan=True)

### 5.1.3 Changing Reference Genome

:TODO switch to how NCBI does the first|second pass VCF.change_base(old_reference_file, new_reference_file, chrom)

Used when the reference genome of a vcf file must be changed. Must be done one chromosome at a time (vcf can have more than one, but the fasta files can only have one).

**parameters:** old_reference_file - The fasta file that contains the data for the chromosome of interest that serves as the reference for the current vcf

new_reference_file - The fasta file that contains the data for the chromosome of interest that is the desired reference for the vcf

chrom - The chromosome of interest. For example '22' or 'Y'

**output:** VCF - a new vcf object with a new VCF.vcf and VCF.geno

### 5.1.4 Indels

**isindel(line)**

**parameters:** line - the line of data to analyze

**output:** Boolean - True if line is an indel, if not, False

CHAPTER 6

# Running Aellic Expression Imbalance

Running aellic expression imbalance analysis on RNA-Sequencing data. Counts the alleles in the specified bam files from 'file_to_sample_mapping.txt' at locations specified in the genotype file. The script returns a pandas dataframe pickled to a file specified by OUTPUT. The columns of the aei dataframe is a multi-index with the first index being the sample id and the second index composing of base pairs in the order as specificed: [A, G, C, T].

:TODO Need to handle indels tag SNPs.

file_to_sample_mapping.txt should have the following tab-delimited format.

'" sample_name path_to_file Sample_1 /proj/data/sample_1.bam "'

genotype_data is a file specificying which SNPs to count at in the 1000G VCF type format.

At the moment the script has a lot of limitations. Tested on STAR and tophat aligned bams.

CHAPTER 7

# Working with SNP Arrays

## 7.1 The SNP Array class

SNP_array(zipfile, fileformat = 'one column', delim = ',', samp_col = None, encoding = None, heade_lines = 0, startatline = 0, readnrows = None):

**parameters:** zipfile - File with data in it. Will try gunzip, but if this fails, it will read it as plain text.

fileformat - Either 'one column' or 'two column' depending on wheter each individual's alleles are shown in one or two columsn. Defaults to 'one column'.

delim - What seperates the values in your data? Defaults to ',' but other common options are tabs ('t') or spaces (' ').

samp_col - What is the first column where the data for each sample starts. If no value is supplied, it defaults to 3 for one column data and 4 for two column data.

encoding - This arguement will take either an integer or a pandas series. If an integer is supplied, the encoder will be taken from the corresponding column (0 based of course). Alternatively a pandas series which contains the reference/alternate alleles (eg. 'A/G') for each position with ids that correspond to the file can be supplied. If no encoder is supplied, no genotype matrix will be created.

header_lines - The number of lines to be read as the reader.

startatline - What line to start reading the data at.

readnrows - How many lines of data to read. When None, the file is read until the end.

### 7.1.1 Data

**SNP_array.df** Data that was passed in parsed as a pandas dataframe.

**SNP_array.encoder** The encoder passed in on creation of the object.

**SNP_array.geno** A pandas dataframe of the genotype data. 0 represents homozygous for the reference allele, 1 is heterozygous, and 2 is homozygous for the alternate allele.

**SNP_array.apply_encoder(encoder)** If an encoder was not supplied upon creating the object, this is how you could still get a genotype matrix after the fact.

**parameters:** encoder - An integer that corresponds to the appropriate column or a pandas series which contains the reference/alternate alleles (eg. 'A/G') for each position with ids that correspond to the file.

**output:** A pandas dataframe of genotype data. 0 represents homozygous for the reference allele, 1 is heterozygous, and 2 is homozygous f or the alternate allele.

### 7.1.2 Hardy-Weinberg

SNP_array.hardyweinberg(snp, excludeNan=True)

CHAPTER 8

# Working with PED files

## 8.1 The PED class

PED(PED, MAP, encoder = None):

**parameters:** PED - The ped file containing the genotype data.

MAP - The map file which contains labels for the snps in the ped file.

encoder - A pandas series which contains the reference/alternate alleles (eg. 'A/G') for each position with ids that correspond to the file. Defaults to None. Must provide one in order to get a genotype matrix.

### 8.1.1 Data

**PED.PED** Data that was passed in parsed as a pandas dataframe.

**PED.MAP** The map file parsed as a pandas dataframe.

**PED.geno** A pandas dataframe of the genotype data. 0 represents homozygous for the reference allele, 1 is heterozygous, and 2 is homozygous for the alternate allele.

### 8.1.2 Hardy-Weinberg

PED.hardyweinberg(snp, excludeNan=True)

CHAPTER 9

# Genotype

Most of the analytic tools are built into the Genotype class, for which VCFs, SNP arrrays, and PED files are all subclasses of.

## 9.1 chi2_association

Uses the chi-squared statistic to determine the likelihood that each SNP determines a certain trait. Example.

**chi2_association(control, case, excludeNan = True)**

**parameters:** control - A pandas dataframe (SNP_array.geno for example) of samples that do not exhibit the desired trait.

case - A pandas dataframe (SNP_array.geno for example) of samples that do exhibit the desired trait.

excludeNan - Defaults to True. If True, means that Nans are not counted, otherwise they are counted as zeros.

**output:** (SNP_dict, ordered_p_values) SNP_dict - A dictionary of each SNP ID to its probability.

ordered_p_values - A list of all of the p-values in the order they were inputted. Useful for graphing.

## 9.2 Hierarcical Clustering

Creates a dendrogram showing the hierarchical clustering of the data. Example.

object.dendrogram()

## 9.3 Hardy-Weinberg

Determines if a given SNP is in Hardy-Weinberg equilibrium. Example.

**object.hardyweinberg(snp, excludeNan=True)**

**parameters:** snp - Name of the snp or locus to test.

excludeNans - When True, any calls of Nan will be excluded, when False, they will be treated as zeroes.

**output:** Boolean - True if snp is within Hardy-Weinberg, False, if it is not.

## 9.4 Fst

**Fst(subpopulations, loci, method = 'W', excludeNan = True, disable_warning = False)**

**parameters:** subpopulations - A list of genotype matricies (ex. VCF.geno, PED.geno, etc.) to be analyzed.

loci - A list of snps to be included in the analysis.

method - Which method of analysis to use. Options are 'WC' for the Weir & Cockerham method, 'W' for the Wright method, or 'R' for the Reich method (pairwise comparison). The default method is the Wright method because while testing it gave us by far the most accurate answers. For now, if you attempt to use the other two methods, a warning message will be printed to remind you of the possibility of an incorrect implimentation. There is more information about each method here.

excludeNan - When True, any calls of Nan will be excluded, when False, they will be treated as zeroes.

disable_warning - Defaults to False. Set to True to make the warning that is printed when using the WC or R methods disappear.

**Output:** Float - The Fst statistic.

CHAPTER 10

# Tutorials

CHAPTER 11

# Indices and tables

- genindex
- modindex
- search