Documentation: baghera$_{tool}$

Release 2.2.0

Viola Fanfani

Jan 28, 2022

Contents

By Viola Fanfani (v.fanfani@sms.ed.ac.uk)

The Bayesian Gene Heritability Analysis software (BAGHERA) estimates the contribution to the heritability of a trait/disease of all the SNPs in the genome (genome-wide heritability) and those nearby protein-coding genes (gene-level heritability).

BAGHERA requires only summary statistics from a Genome-wide Association Study (GWAS), LD scores calculated from a population matching the ethnicity of the GWAS study and a gene annotation file in GTF format.

CHAPTER 1

Workflow

Alongside BAGHERA, we are providing a snakemake workflow repository with sample data. workflow-baghera

CHAPTER 2

Getting Started

2.1 Installation

The easiest and fastest way to install BAGHERA using **conda**:

```
$ conda install -c stracquadaniolab -c bioconda -c conda-forge
```

We have prepared also a **docker** image that can be found at this link.

The image can be pulled as follows:

```
$ docker run docker.pkg.github.com/stracquadaniolab/baghera/baghera:latest
```

By downloading the docker image, you will download a virtual machine with the latest version of baghera and its requirements.

Once the image is downloaded, you can run baghera from the docker. For example, use the command below to be prompted baghera's help:

```
$ docker run docker.pkg.github.com/stracquadaniolab/baghera/baghera:latest -h
```

Here below is an example on how to use the *create-files* command:

```
$ docker run --rm -v "$PWD:$PWD" -w "$PWD" docker.pkg.github.com/stracquadaniolab/
↪baghera/baghera:latest create-files -l <ldscore_folder> -a <annotation.gtf> -s <ld_
↪annotated_snps> -g <genes_table>
```

2.2 Tutorial

A typical BAGHERA analysis consists of 3 steps, we briefly explain them here, more details can be found in the documentation and a practical example is in the snakemake workflow.

2.2.1 1- Build a SNP annotation file

Build a SNP annotation file, where SNPs are annotated to genes and LD scores are assigned. We use precomputed ld-score , from the set of variants for the European population of 1000 Genomes, and the genes in the Gencode v31 annotations , using only the protein coding terms. To cope with overlapping genes, we clustered them, obtaining a dataset of 15000 non-overlapping genes. For the annotation, we use a 50 kb window.

```
$ baghera-tool create-files -l <ldscore_folder> -a <annotation.gtf> -s <ld_annotated_
↪snps> -g <genes_table>
```

2.2.2 2- Annotate summary statistics

Annotate summary statistics with the SNP annotation built in step 1:

```
$ baghera-tool generate-snp-file -s <stats file> -i <input_type> -o <snps_file> -a
↪<ld_annotated_snps>
```

2.2.3 3- Run the regression

Run the regression:

```
$ baghera-tool gene-heritability <snps_file> <results_table> <summary_table> <log_
↪file> --sweeps <samples> --burnin <tuning> --n-chains <chains> --n-cores <cores> -m
↪<models>
```

Example

Running BAGHERA on the UK Biobank summary statistics for breast cancer, using EUR LD scores and the Gencode annotation.

```
$ baghera-tool create-files -l data/eur_w_ld_chr/ -a data/gencode.v31lift37.basic.
↪annotation.gtf -s data/ld_annotated_gencode_v31.csv -g data/genes_gencode_v31.csv
$ baghera-tool generate-snp-file -s data/C50.gwas.imputed_v3.both_sexes.tsv -i␣
↪position_ukbb -o data/c50.snps.csv -a data/ld_annotated_gencode_v31.csv
$ baghera-tool gene-heritability data/c50.snps.csv data/results_normal_c50.csv data/
↪summary_normal_c50.csv data/log_normal_c50.txt --sweeps 10000 --burnin 2500 --n-
↪chains 4 --n-cores 4 -m normal
```

Workflow

Alongside BAGHERA, we are providing a snakemake workflow repository with sample data.

CHAPTER 3

Usage

3.1 Creating the annotated files

3.1.1 Annotated variants

To run the GWAS analysis the variants in the study need to be annotated with the ld-score and to a gene. To create the SNPs dataset use the *baghera-tool create-files* command

We use precomputed ld-score , from the set of variants for the European population of 1000 Genomes (unzip the ld score files inside the downloaded folder), and the genes in the Gencode v31 annotations , only the protein coding ones. To cope with overlapping genes, we clustered them, obtaining a dataset of 15000 non-overlapping genes. For the annotation, we use a 50 kb window. The resulting dataset of annotated variants has around 1.3 millions SNPs, 55% of which are annotated with a gene.

Please note that this file has already been created, to process the data skip to the next section

It is possible to annotate a different set of variants, for example another reference panel, using the *create-files* function. For the moment it only supports .gtf files for the genes annotation and the LD-score folder with the structure in https://github.com/bulik/ldsc

The annotated ld scores table, return as an output, has the following structure:

chr	rs_id	position	cm	maf	l	gene
9	rs10123646	108998	0.090	0.499	3.155	FOXD4

While the gene table looks like the one below.

chrom	start	stop	name
1	65418	71585	OR4F5

3.1.2 Create the dataset

The BAGHERA core analysis uses a table like the one below

chr	rs_id	position	cm	maf	l	gene	sample_size	z
9	rs10123646	108998	0.090	0.499	3.155	FOXD4	361194	1.4

Such file is generated by merging a summary statistics file with the annotated ld file.

To create the SNPs dataset use the *baghera-tool generate-SNPs-file* command

The function uses a **tsv** table as input and merges it with the annotated ld score table.

SNPs input types

There are different input types managed by the code, specified in the parameter, use the *-i <type>* parameter.

We recommend the use of the **position** option, however we provide functions to directly process data that we have been using for this project.

Merge according to position

Once the user makes sure the genome build in use is consistent across all files, merging SNPs and genes according to their position is the safest. This way no rsId is taken into consideration, with the risk of a different naming.

Specifying the flag - *position*, BAGHERA expects to find the following columns in the input SNP file:

- chrom: chromosome
- pos: BP
- nCompleteSamples: number of samples
- tstat: beta/se stat

UKBB format

Since we processed all the data cancer data from the UKBB GWAS study available here (in the round two results), we provide an off-the-shelf flag to directly process these data. Using the flag *-i position_ukbb*, the tool automatically extracts the position from the *variant* field in the table. This function directly splits the variant column if those are not found an exception is raised THe tool is expecting the following fields:

- variant: a large variant
- nCompleteSamples: number of samples
- tstat: beta/se stat

Other input formats

The LD-score project has some available summary statistics that they have processed, use the *-i ldsc* to process the **sumstats** file (.sumstats.txt)

The LD-score project has some available summary statistics that they have processed, use *-i ukbb* to process the the old **UKBB** files, .assoc.tsv

3.2 Running the analysis

To date, the available code for the analysis allows to perform:

- a **gene-level Bayesian analysis**, that returns the probability of the single gene to affect the trait heritability with an higher probability than by chance
- a **genome-wide Bayesian regression**, able to estimate the heritability of the trait.

3.2.1 Gene-level analysis

The algorithm to perform a gene-level analysis is runnable as *baghera-tool gene-regression -h* here is the complete list of options of the function:

```
usage: baghera-tool gene-heritability [-h] [--sweeps SWEEPS] [-b BURNIN]
                                    [--n-chains N_CHAINS]
                                    [--n-cores N_CORES] [-N N_1KG]
                                    [-c CHROMOSOME] [--snp-thr SNP_THR]
                                    [--sep SEP] [-m MODEL] [-f]
                                    input-snp-filename output-genes-filename
                                    output-summary-filename logger-filename


    Performs bayesian gene-level heritability analysis.
    As input it needs the annotated snps file created with generate-snps-file.

    From command line one can specify all the parameters for the sampler
    (sweeps, burnin, chains and cores) and the parameters for the SNPs
    and genes filtering.

    Specify the gamma model by passing --model gamma


positional arguments:
input-snp-filename    Data Input, use the SNPs file from dataParse
output-genes-filename
                        output file for gene-level results, use .csv
output-summary-filename
                        output file for the genomewide results summary, use
                        .csv
logger-filename       file for the logger, use a txt

optional arguments:
-h, --help            show this help message and exit
--sweeps SWEEPS       number of samples for each chain (default: 1000)
-b BURNIN, --burnin BURNIN
                        number of burnin samples (default: 1000)
--n-chains N_CHAINS   number of chains of the sampler (default: 4)
--n-cores N_CORES     number of parallel cores to use (default: 4)
-N N_1KG, --N-1kG N_1KG
                        number of SNPs onwhich the LD-score is calculated
                        (default: 1290028)
-c CHROMOSOME, --chromosome CHROMOSOME
                        chromosome on which the analysis is run (default:
                        'all')
--snp-thr SNP_THR     threshold for the minimum number of SNPs in a gene
```

(continues on next page)

(continued from previous page)

```
                        (default: 10)
--sep SEP             separator for the input files, use t for tab separated
                        (not ) (default: ',')
-m MODEL, --model MODEL
                        specify the model for the regression, one betwenn
                        normal/gamma (default: 'normal')
-f, --fix-intercept   False
```

From the input annotated file

1. **gene_level_analysis.log** : is a log file that reports the parameters of the analysis and the output results. *This is just a synthesis of the analysis, not the run-time log file which can be found in the logs folder*

2. **summary.csv** : output summary statistics of the traces. *e* is the intercept and *mi* is the regression slope (heritability), *W* is the variance term and *herTOT* is the weighted summation of the signle gene term. For each of them the principal stats are reported.

3. **results.csv**: is the output file, where the stats for the single gene are reported.

Results Files format

The output file `results.csv` is a csv file, which contains all the results for each gene. Here is the a description of the format:

- *chrom* , *start* , *stop* , *name* : we are considering all the genes after the clustering, some of them have a composite name, and that are included in the analysis.
- *LDvariance*, variance of the ld-score within the gene
- *SNPs*, number of SNPs within the gene
- *StatsMax,StatsMean,StatsMin,StatsVariance*, for the chi-squared statistics within the gene max, min, average and variance values
- *P*, output probability
- *bg_mean, bg_var, bg_median, bg_5perc, bg_95perc*, are the stats for the single gene heritability weights.
- *mi_mean, mi_median*, genome-wide heritability posterior
- *h2g*, weighted sum of the single gene heritability

3.2.2 Genome-wide Heritability

The algorithm to get the heritability estimate, performing a genomewide regression on the SNPs

```
usage: baghera-tool gw-heritability [-h] [--sweeps SWEEPS] [-b BURNIN]
                                    [--n-chains N_CHAINS] [--n-cores N_CORES]
                                    [-N N_1KG] [-c CHROMOSOME] [--sep SEP]
                                    [-m MODEL] [-f]
                                    input-snp-filename output-summary-filename
                                    logger-filename

    Computes the genome-wide estimate heritability using Bayesian regression.


positional arguments:
input-snp-filename    Data Input, use the SNPs file from dataParse
```

(continues on next page)

(continued from previous page)

```
output-summary-filename
                        output file for the genomewide results summary, use
                        .csv
logger-filename       file for the logger, use a txt

optional arguments:
-h, --help            show this help message and exit
--sweeps SWEEPS       number of samples for each chain (default: 1000)
-b BURNIN, --burnin BURNIN
                        number of burnin samples (default: 1000)
--n-chains N_CHAINS   number of chains of the sampler (default: 4)
--n-cores N_CORES     number of parallel cores to use (default: 4)
-N N_1KG, --N-1kG N_1KG
                        number of SNPs onwhich the LD-score is calculates
                        (default: 1290028)
-c CHROMOSOME, --chromosome CHROMOSOME
                        chromosome on which the analysis is run (default:
                        'all')
--sep SEP             separator for the input files, use t for tab separated
                        (not ) (default: ',')
-m MODEL, --model MODEL
                        regression model (default: 'normal')
-f, --fix-intercept   False
```

As output, two files are created.

1. **Log file.txt** : is a log file that reports the parameters of the analysis and the output results. *This is just a synthesis of the analysis, not the run-time log file which can be found in the logs folder*

2. **Results_file.csv** : output summary statistics of the traces. *e* is the intercept and *mi* is the regression slope (heritability). For each of them the principal stats are reported.

This analysis is the bayesian version of the LDSC hertiability model.

CHAPTER 4

API

4.1 API

4.1.1 Snps

The SNP table is managed by the Snps class.

4.1.2 Genes

The gene table is managed by the Genes class.

4.1.3 Analysis

Gene-level heritability

All ananlysis functions are inside the *gene_regression.py* file

CHAPTER 5

Issues

BAGHERA is still under development, and a major refactoring will happen at some point. If you find a bug, please report it on GitHub.

CHAPTER 6

Changelog

6.1 Changelog

6.1.1 v2.1.11 21/01/2022

- Fixed bug to solve issue #5. Added back line that had been removed.

6.1.2 v2.0.0 - 25/04/2020

Major refactoring of the input/output formats and the classes structure. Documentation updated accordingly. No changes in the core behaviour.

6.1.3 v1.1.0 - 12/04/2019

- Updated documentation and minimal refactoring.

6.1.4 v1.0.1 - 12/04/2019

- Updated documentation with an example pipeline.

6.1.5 v1.0.0 - 02/04/2019

- Genome Wide Heritability analysis available.
- Files formatted according to PEP8 standards.
- Changed output strategy, now the code can be included in a snakemake pipelines.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search