GWAMAR user's manual

Release 1.9

Michal Wozniak

27 March 2015

Contents

1 About the software **1**
- 1.1 Background . . . 1
- 1.2 Availability . . . 1
- 1.3 About the authors . . . 2
- 1.4 Installation . . . 2
- 1.5 Design of GWAMAR . . . 2
- 1.6 Parameters . . . 3
- 1.7 Input data . . . 4
  - List of strains . . . 4
  - Phylogenetic tree . . . 5
  - Mutation profiles . . . 5
  - Drug resistance profiles . . . 5
  - Gold standard associations . . . 5
- 1.8 173 strains of M. tuberculosis . . . 6
- 1.9 Broad Institute dataset . . . 6

1 About the software

1.1 Background

Software to identify drug resistance-associated mutations in bacterial strains (published: BMC Genomics, 2014).

1.2 Availability

This is the project website:

`http://bioputer.mimuw.edu.pl/gwamar`

This is the project repository website:

`https://bitbucket.org/mimowo/gwamar`

Please don't hesitate to contact us with any comments and suggestion or if you are interested in co-developing this software.

1.3 About the authors

This software was implemented by Michal Wozniak. All authors contributed to design of the method and analysis of results. Project idea and guidance came from prof. Limsoon Wong (National University of Singapore) and prof. Jerzy Tiuryn (University of Warsaw).

Corresponding author: Michal Wozniak (`m.wozniak@mimuw.edu.pl`)

1.4 Installation

This software is written in Python. Python 2 or 3 is required to run GWAMAR. The software does not need any classical installation, you only need to download and extract the zip package (it works under Windows, Linux and Mac OS) from the project website:

`http://bioputer.mimuw.edu.pl/gwamar`

1.5 Design of GWAMAR

The software is designed as a set of executable scripts written in Python which can be run via the console script `gwamar.py`.

Figure 1 presents the hierarchy of folders in GWAMAR open in the Sublime text editor.

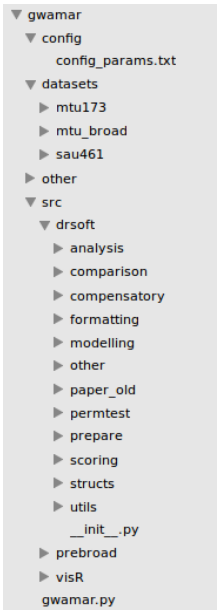

Figure 1: Hierarchy of folders in GWAMAR open in Sublime.

The hierarchy of folders in GWAMAR (including the dataset folders):

- `gwamar/` — the main GWAMAR folder with the executable files
  - `gwamar.py` — the main console script to run GWAMAR
  - `config/` — configuration files:
    * `config_params.txt` — configuration of the GWAMAR parameters

Parameter	Default	Example values	Description
a	–	p,s,a,cmp	action
w	1	1,2,3,...	# of threads (workers used)
d	mtu_broad	mtu173,mtu_broad	dataset
s	lh,ws,mi,or,r-tgh	lh,ws,mi,or,r-tgh	scores to compute
s	mi+or+lh,mi+or+lh+ws+r-tgh		combined scores to compute

Table 1: List of GWAMAR available parameters and their short descriptions.

- `datasets/` — input datasets
  - `dataset1/` — the folder with `dataset1` input files (for example `dataset1=mtu173`)
  - `dataset2/` — the folder with `dataset2` input files (for example `dataset2=mtu_broad`)
  - `dataset3/` — the folder with `dataset3` input files (for example `dataset3=sau461`)
- `src/`
  - `drsoft` — source code of GWAMAR
  - `drsoft/analysis` — scripts to analyze results of post-processing
  - `drsoft/comparison` — scripts to analyze comparison of different association scores based on gold standard
  - `drsoft/compensatory` — scripts for visualization of compensatory mutations
  - `drsoft/formatting` — scripts for visualization of drug resistance data
  - `drsoft/permtest` — scripts for computation of p-value scores
  - `drsoft/prepare` — scripts for preparing input for the experiments
  - `drsoft/scoring` — scripts running computations of different scores
  - `drsoft/structs` — structures used in GWAMAR
  - `drsoft/utils` — implementation of common methods used in GWAMAR
  - `prebroad` — preprocessing of Broad Institute data
  - `visR` — R tools for visualization of results

1.6 Parameters

There are two methods to specify parameters in GWAMAR:

- -key_value
- key=value

The two following commands (to compute association scores) are equivalent:

- `python gwamar.py -a s -w 4`
- `python gwamar.py a=s w=4`

Table 1 presents the list of GWAMAR available parameters and their short descriptions.

Table 2 presents the list of available actions, set by the `-a` parameter.

Action	Description
pb	preprocess the Broad Institute dataset
p	prepare data for further analysis
s	run scoring
a	output list of top-scoring mutations
cmp	run comparison of different scores
cmp1	save association lists for comparison
cmp2	perform sampling of negatives and AUC calculations
cmpF	generate the AUC barplots for different methods

Table 2: List of GWAMAR available options for the `-a` parameter.

1.7 Input data

Input files for GWAMAR, for a set of bacterial strains, consists of a set of mutation profiles and drug resistance profiles to associate. This data is specified in the following files:

- `dataset/input/strains_ordered.txt` - list of strains,
- `dataset/input/tree.txt` - phylogenetic tree of strains. Optional and necessary only for calculations of tree-aware statistics like: weighted support and TGH. It does not need to be provided to calculate odds ratio, mutual information or the standard hypergeometric test,
- `dataset/input/tree_bin.txt` - binary phylogenetic tree of strains. Optional, but useful to speed up computations of the TGH scores.
- `dataset/input/res_profiles.txt` - list of available drug resistance profiles,
- `dataset/input/point_mutations.txt` - list of point mutations,
- `dataset/input/gene_profiles.txt` - list of gene gain/loss profiles,
- `dataset/gold/gold_assocs_A.txt` - list of mutations associated with drug resistance which is used as for gold standard. For *M. tuberculosis* we retrieved this list from the TBDReamDB database (taking all the mutations in the database into account).
- `dataset/gold/gold_assocs_H.txt` - list of mutations associated with drug resistance which is used as for gold standard. For *M. tuberculosis* we retrieved this list from the TBDReamDB database (taking only high confidence mutations in the database into account).

In the following subsections we describe the input formats and provide small examples. Larger input datasets are integrated with the sources.

List of strains

File *dataset/input/strains_ordered.txt* provides the list of strains in the same order as the strains appear in the phylogenetic tree.

Example:

```
s1
s2
s3
s4
```

Phylogenetic tree

File *dataset/input/tree.txt* should provide the phylogenetic tree for the set of considered strains in the Nawick format.

Example:

```
((s1,s2),(s3.s4));
```

Mutation profiles

First line of the file `dataset/input/point_mutations.txt` contains an order list of strains. Positions of the strains on the list are used as the strain identifiers. All mutations are grouped by genes. Each block of mutations, for a given gene starts with a line in the following format:

```
>gene_id gene_cluster_id gene_name
```

After that line there is a series of lines which provide information on the set of mutations within the gene. Each of these lines has the following format:

```
position p/n mutation_profile
```

Here p and n stand for promoter (position < 0) and amino acid, respectively.

Example:

```
s1 s2 s3 s4
>Rv1484  1  inhA
-40 p TTCC
194 n I?TT
>Rv0667  2  rpoB
450 n SLLS
```

Drug resistance profiles

First line of this file contains an order list of strains. Positions of the strains on the list are used as the strain identifiers. In the letter lines drug resistance profiles are provided in the following format:

```
name_of_drug_resistance_profile drug_resistance_profile
```

Example:

```
s1 s2 s3 s4
drp1   RS?S
drp2   SRS?
```

Gold standard associations

File *dataset/gold/gold_assocs_H.txt* provides the list of gold standard associations which are used for assessment of the accuracy of different association methods. These associations are listed in the following format:

```
drp1 gene_name gene_id position reference_amino_acid mutated_amino_acid
```

Example:

```
drp1 rpoB Rv0667 450 S L
```

1.8 173 strains of M. tuberculosis

In order to preprocess the data we used eCAMBer. Dataset *mtu173*.

Preparation of intermediate files:
`python gwamar.py -a p -w 4 -d mtu173`

Computation of association scores:
`python gwamar.py -a s -w 4 -d mtu173 -s lh,ws,mi,or,r-tgh`

Generation of list of scored associations:
`python gwamar.py -a a -w 4 -d mtu173 -s lh,ws,mi,or,r-tgh`

Comparison of different association methods:
`python gwamar.py -a cmp -w 4 -d mtu173 -s lh,ws,mi,or,r-tgh`

1.9 Broad Institute dataset

In order to download and preprocess the Broad Institute data you can use the following command: `python gwamar.py -a pb -w 4 -d mtu_broad`

Here, the dataset *mtu_broad* is equivalent to the *mtu_broad* dataset from the paper.

Preparation of intermediate files:
`python gwamar.py -a p -w 4 -d mtu_broad`

Computation of the standard association scores:
`python gwamar.py -a s -w 4 -d mtu_broad -s lh,ws,mi,or,r-tgh`

Computation of the combined association scores:
`python gwamar.py -a s -w 4 -d mtu_broad -s mi+or+lh,mi+or+lh+ws+r-tgh`

Generation of list of scored associations:
`python gwamar.py -a a -w 4 -d mtu_broad -s lh,ws,mi,or,r-tgh`

Comparison of different association methods:
`python gwamar.py -a cmp -w 4 -d mtu_broad -s lh,ws,mi,or,r-tgh`