ZANARDI

an open-source pipeline for multiple-species genomic analysis using SNP array data

Latest update: December 9th, 2015

Marras G., Rossoni A., Schwarzenbacher H., Biffani S., Biscarini F., Stella A., Nicolazzi E.L.

Created with the support of:

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement n° 289592

GENE2FARM

Upgraded with the support of:

genHome

climgen

DISCLAIMER

Zanardi is a free software tool that uses proprietary software that is publicly available online. Zanardi is distributed in the hope that it will be useful WITHOUT ANY WARRANTY, and WITHOUT ANY IMPLIED WARRANTY of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. You can redistribute our pipeline and/or modify it, but entirely at your own risk. See the GNU General Public License for more details: http://www.gnu.org/licenses/. This tool is intended for research, with no commercial intent, and it can be freely used by any organization. Our only goal is to help researchers to carry out genomic analyses more easily.

Alessandro (Alex) Zanardi was not involved in anything related to this tool. We named this tool after him only as a tribute to a great man with an inspiring life. Therefore, neither Alex Zanardi nor his foundation bear any responsibility for any aspect of this tool. Some of the authors of this tool were funded by the Gene2Farm, ClimGen and GenHome projects, which were financed with support from the European Commission and the Italian Government. However, neither the European Commission, the Italian Government nor the partners of the projects mentioned can be held responsible for ANY information contained in this website, or outcome from using the tool or any modification of it. Likewise, the authors of this tool are not responsible for ANY output or result obtained from it, or any modification of it.

For bug reports, feedback and questions contact *ezequiel.nicolazzi@ptp.it*, but PLEASE read this manual carefully before sending in your bug report or question.

Content

CHAPTER 1. INTRODUCTION

1.A. WHAT IS ZANARDI?

Zanardi is a tool written with the objective of making life easier for people to deal with genomic data. Usually, format conversions are necessary and challenging, and many softwares have different requirements and dependencies. This, together with a large number of other difficulties, makes software integration unstable and linking multiple softwares within a pipeline a big headache. Zanardi is an all-in-one tool, user-friendly and completely open-source that tries to tackle all the above.

This tool heavily relies on multiple-purpose software coded by other researchers, so please make sure you acknowledge their work while running Zanardi. See the "How to Cite Zanardi" section for full information and links to the original software. As a user, you're expected to be perfectly aware of your objectives when running the tool, and what you expect as a result. Running Zanardi without a good knowledge of the underlying software is a great risk, so please use this tool wisely and read this manual. Although we hope using Zanardi helps to make life easier for you, you are strongly encouraged to read the manuals for each of the softwares integrated into Zanardi.

1.B. WHY "ZANARDI"?

The name of this tool is inspired by our personal hero: Alex Zanardi (http://en.wikipedia.org/wiki/Alex_Zanardi), an outstanding person, a racing champion, a Paralympic gold medalist and a true *ironman*, among other qualities. Although his life is a great inspiration *per se*, this tool was named after him because we wanted something that was fast, robust and that could help people overcome the struggle of ... genomic analyses. Pretty close to someone like him! If you are interested in knowing more about his inspiring life, you should read the book: "Alex Zanardi: My Sweetest Victory". You won't regret it.

1.C. HOW TO CITE ZANARDI

We're working on it… if you wish to cite this tool, please cite the github address and its authors.

1.D. WHO ARE WE?

Gabriele Marras *is a post-doc researcher at Parco Tecnologico Padano (PTP), Italy.*
ResearchGate: https://www.researchgate.net/profile/Gabriele_Marras

Attilio Rossoni *is the head of R&D at ANARB (Italian Brown Swiss Breeders Association), Italy.*
LinkedIn: https://www.linkedin.com/pub/attilio-rossoni/1a/146/3a0
ResearchGate: https://www.researchgate.net/profile/Attilio_Rossoni

Hermann Schwarzenbacher *is a reserarcher at the R&D office of ZuchtData GmbH, Austria.*
ResearchGate: https://www.researchgate.net/profile/Hermann_Schwarzenbacher

Stefano Biffani *is a Research fellow (Geneticist/Biostatistician) at IBBA-CNR National Research Council*
LinkedIn: https://www.linkedin.com/pub/stefano-biffani/19/a95/aa6
ResearchGate: https://www.researchgate.net/profile/Stefano_Biffani

Filippo Biscarini *is a Principal investigator in Bioinformatics, Bioistatistics and Biomedicine at Parco Tecnologico Padano (PTP), Italy.*
LinkedIn: https://www.linkedin.com/pub/filippo-biscarini/65/279/600
ResearchGate: https://www.researchgate.net/profile/Filippo_Biscarini

Ezequiel L. Nicolazzi *is the Technical manager of the Bioinformatics core facility of Parco Tecnologico Padano (PTP), Italy.*
LinkedIn: https://it.linkedin.com/pub/ezequiel-luis-nicolazzi/30/34/792
ResearchGate: https://www.researchgate.net/profile/Ezequiel_Nicolazzi

1.E. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European's Union Seventh Framework Programme for research, technological development and demonstration under grant agreement 289592 - **Gene2Farm** (www.gene2farm.eu), from the FACCE-ERA-NET ClimGen project and from the Italian research project "GenHome".

Authors wish to acknowledge the contribution of John Woolliams (Roslin, Edinburgh) for English and understanding revision.

The flag picture in the header page was downloaded from www.psdgraphics.com .

CHAPTER 2. ZANARDI SOFTWARE - GENERAL

Zanardi is, essentially, a modular software that performs a wide variety of genomic analyses. Its modular nature makes it versatile, and easy to debug, update and upgrade. Its main purpose is to being able to perform and stream multiple genomic analyses starting from a bunch of input files containing data. We expect this to greatly improve the quality of life (and life expectation) of any researcher. We plan to integrate as many softwares as possible. To help us achieve this ambitious goal, we're not only releasing the full code, fully open-source, but we're also strongly encouraging users to provide feedback, ideas, new modules, etc.

The normal user will only interact with the parameter file (PARAMETER.txt) and a simplified command line that tells the software which analysis to run and, if required, stream further analyses. In order to facilitate the usually tedious download-uncompress-install-link process of dependencies (i.e. Zanardi uses many third-party software) we provide a routine that will do all that for you automatically. You wish everything in life was *that* simple, right?
Once launched, Zanardi interprets the list of analyses to be run, reads and thoroughly checks the parameter file, and starts the required analyses, following a specific order (see Chapter 3 for further information).
A very appealing functionality of Zanardi is the possibility to include an infinite number of input files. In fact, multiple files of multiple formats can be used at the same time. Zanardi will first transform everything to PLINK (ped and map) format, and then automatically merge the files using default PLINK -- merge functionalities. Please note that *all the downstream analyses will be performed on this merged file*.
Since many analyses can be streamed to run sequentially, the general behavior when multiple analyses are requested is that the output of the first analysis is used as the input for the second… and so on as needed.
With only very few exceptions, all analyses contained in Zanardi can be called simultaneously, with the result that a *large* number of analyses can be streamed with almost no effort. The output of each step will be saved in the associated *output* folder, in order to make everything easily accessible.
Since clarity is essential when many analyses are run simultaneously, detailed reports are given for each step, on screen and simultaneouly in a Zanardi.log file. Beware, this log file will be overwritten each time Zanardi is run.
Please remember that the "[GOOD NEWS]:" message means the step or the analysis was successful, whereas the "[BAD NEWS]:" message indicates that an error was found and the program will terminate.

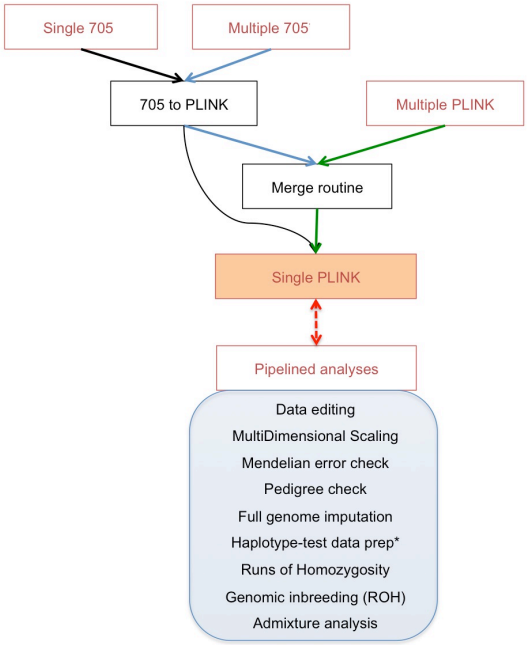
2.B. (A FEW MORE) TECHNICAL ASPECTS

Zanardi is almost completely written in Python 2.7+. We strongly advise using Python 2.7+ when running Zanardi as earlier versions are not officially supported. In addition, Zanardi uses a few R scripts, mainly to obtain nice graphs using the *ggplot2* library (the default *plot* library was too basic), and a single bash script for the download of third-party software. Except for these few exceptions, everything in Zanardi can be read or modified python programmer with average skills. Java 1.7+ is a further requirement for any analysis using Beagle.
A deliberate design choice was to use only Python's built-in libraries to reduce the dependencies of the whole structure to a minimum and so enhance user-friendliness. Therefore we ask all contributors to follow this design choice when contributing their own code.
Zanardi was developed under CentOS, but it should work with any Linux/Unix and Mac distribution.
To report a bug in any aspect of Zanardi, please contact ezequiel.nicolazzi@ptp.it .

CHAPTER 3. ZANARDI SOFTWARE – USAGE

3.A. STRUCTURE AND STREAMLINE

Zanardi has a specific order in which analyses are run, which cannot be modified (unless you are able to code in python, but doing it at your own risk!). The general scheme is:



In short:

- **Red boxes** indicate Zanardi's possible input files (single or multiple PLINK or 705 files – see Chapter 3.B for further info)
- **Black boxes** indicate Zanardi's way to convert the data: 705 files are first transformed to PLINK and then merged if required, multiple files are merged into a single PLINK file over which all analyses are run
- **Blue boxes** indicate options (analyses available – note this graph gets outdated fast!).
- **Straight arrows** indicate an (intermediate) step, which is performed by default if the corresponding input file(s) is/are provided. For example: if a single Interbull 705 file is provided, it is converted to PLINK format (PED). If multiple Interbull 705 files are provided (or more than one SNP array is contained in a single 705 file), Zanardi produces first multiple PEDs (as many as SNP arrays are present), and then the files are merged (using PLINK) to create a single PLINK FILE.
- **Dashed arrows** are the analyses streamlined by Zanardi, thus optional and user defined. A user can choose to run just one, a few or all of them. See Chapter 3.D for further information.
- **Asterisk** means the option is standalone. Since the dataset is heavily modified, the output of this analysis will terminate the program.

3.B. INSTALLING ZANARDI & GENERAL REQUIREMENTS

Zanardi is for users running Linux/Unix or Mac operative systems only, and only runs over **64bit** infrastructures. As stated in the previous section, the only pre-requirements to run this pipeline are: **Python 2.7+**, **Java** 1.7+ (to run Beagle) and **R 3+** (+ *ggplot2* installed for some options), which are ~default in any

up-to-date computer used for scientific research.
The fastest and most clever way of getting this pipeline and all accessory files is installing Git and cloning this repository. Further information on how to install Git can be found at http://git-scm.com/book/en/v2/Getting-Started-Installing-Git. An example of cloning command using command line is:

```
git clone --recursive https://github.com/bioinformatics-ptp/Zanardi.git
```

Important warning to Windows users: Our advice for Windows users is to install a virtual machine (e.g. VirtualBox) + Linux (Ubuntu) and run Zanardi in there. PLEASE DO NOT TRY USING Zanardi under Cygwin, as it will NOT work. Cygwin uses a twisted way of building linux-like (?) paths.

3.C. SETTING THE PARAMETER FILE

Zanardi parameter file (PARAMETER.txt) is key to obtain the desired results. Users are advised to set up this file carefully. The parameter file can be divided in 3 main sections:

1) **Software paths** (e.g. path to where third-party software is installed. See the *--download* option if you don't have the required software installed).
2) **Input files and general info** (e.g. input files for Zanardi: genotypes, phenotypes, pedigree files, plus general information).
3) **Program options** (e.g. specific for each option)

3.C.1. SOFTWARE PATHS

Like the section name suggests, this is where paths to third-party software should be specified. If you have one or more of the third-party software already installed in your computer, just provide the link to the executable. If not, please see the *--download* option in Chapter 3. Note that if you have already Beagle v.3 and/or Beagle v.4 in your computer, the .jar executable should be renamed as *beagle3.jar* and *beagle4.jar* for Zanardi to recognize them.

Currently, Zanardi handles the following third-party software (more to come):

- Plink v1.9 – soon to be Plink v.2 (variable **PGM_PLINK**),
- FCgene (variable **PGM_FCGENE**),
- Beagle v.3 (variable **PGM_BEAGLE3**),
- Beagle v.4 (variable **PGM_BEAGLE4**),
- Admixture v1.23 (variable **PGM_ADMIXTURE**).

Please note that Zanardi checks only the paths required for the analyses required by the user. E.g. if you choose an option that uses only plink (see Chapter 3 for further information on the software used by each option, and Chapter 4 for citing each software used), only the variable PGM_PLINK will be checked.

3.C.2. INPUT FILES AND GENERAL INFO

Here users should provide the full path to the input files and a few – important – general specifications. Following, a list (and a brief explanation) of currently available variables.

- SPECIES AREA

- Variable **SPECIES**: This variable specifies the species to be analyzed. This is used to set the number of maximum chromosomes available, which are autosomal and which are not, etc.. Currently available options are (users are asked to state the *name* of the species; the number of chromosomes are reported to help users setting the right chromosome names/numbers on each of the species):
    - **COW**: 29 AU + X (or 30),Y (or 31), XY (or 32), MT (or 33)

- GOAT: 29 AU + X (or 30),Y (or 31), XY (or 32), MT (or 33)
- CHICKEN: 38 AU + Z (or 39), W (or 40), ZW (or 41), MT (or 42)
- DOG: 38 AU + X (or 39),Y (or 40), XY (or 41), MT (or 42)
- HORSE: 31 AU + X (or 32),Y (or 33), XY (or 34), MT (or 35)
- MOUSE: 19 AU + X (or 20),Y (or 21), XY (or 22), MT (or 23)
- SHEEP: 26 AU + X (or 27),Y (or 28), XY (or 29), MT (or 30)
- HUMAN: 29 AU + X (or 30),Y (or 31), XY (or 32), MT (or 33)
- PIG*: 18 AU + X (or 24),Y (or 25), XY (or 26), MT (or 27)
- ALL: 59 chromosomes (including X,Y,XY,MT that must be coded with letter code. This is the max number of chromosomes that can be provided)

***NOTE: SINCE PIG SPECIES IS NOT OFFICIALLY SUPPORTED BY PLINK, WE INCLUDED THIS OPTION TO FACILITATE WORK FOR RESEARCHERS WORKING ON THIS SPECIES. OUR ADVICE IS TO USE X,Y AND MT INSTEAD OF NUMBERS FOR CHROMOSOME CODING.**

- INPUT FILE AREA

PLINK FILES (PLINK or INTERBULL files are required)

- Variable **INPUT_PED:** One or more PLINK PED format file(s) - with full path, if desired. See next paragraph 3.D.1 or http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#ped for file format and for more information. If multiple files are provided they should be comma separated (","). No blank spaces are allowed. See the provided PARAMFILE.txt for an example.

- Variable **INPUT_MAP:** One or more PLINK MAP format file(s) - with full path, if desired. See next paragraph 3.D.1 or http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#map for file format and for more information. If multiple files are provided they should be comma separated (","), **and they have to follow the same order as the PLINK PED files provided** (e.g. if 2 PED files are provided, the first with 50 SNP and the second with 10 SNP, then 2 MAP files are expected, containing 50 rows the first, and 10 rows the second). No blank spaces are allowed. See the provided PARAMFILE.txt for an example.

The general (important) rule is that Zanardi will read each value in the INPUT_PED and link it to the value in the same position in INPUT_MAP. Some simple examples:

- *Single PLINK file couple*: One PED and one MAP file couple are specified in the variables INPUT_PED and INPUT_MAP of the parameter file. Imagine you have 2 files: *geno.ped* and *geno.map* in a folder named */home/myplink_files*. The parameter file should look like this:

```
INPUT_PED=/home/myplink_files/geno.ped
INPUT_MAP=/home/myplink_files/geno.map
```

- *Multiple PLINK file couples*: If more than one PLINK file couples are analysed together, filenames should simply be comma separated. An infinite number of [PED/MAP] files are allowed. It is important that A) the number of files in INPUT_PED and INPUT_MAP are the same; B) each PED file has its corresponding MAP file in the same position (see example below). For example, following the above example, if the same folder also contains a *genoHD.ped* and *genoHD.map* file couple, and the user wishes to analyse all geno[.ped/.map] and genoHD[.ped/.map] files together, the parameter file should now look like this:

```
INPUT_PED=/home/myplink_files/geno.ped,/home/myplink_files/genoHD.ped
INPUT_MAP=/home/myplink_files/geno.map,/home/myplink_files/genoHD.map
```

When multiple files are provided, Zanardi will automatically merge all files using default PLINK merge functionalities (--merge-mode 1). This means only consensus calls are retained, whereas all

conflicting calls are set to missing (see PLINK --*merge* option for further information).

NOTE: Zanardi does not check for consistency of SNPs, as PLINK already takes care of that. Zanardi will only automatically check PLINK log file(s) and stop if an error message is given. Thus, it is up to the user (e.g. you!) to check if alleles are coded consistently, and if ped/map files are correct!

INTERBULL FILES (PLINK or INTERBULL files are required)

- Variable **INPUT_705:** One or more 705 format file(s) - with full path, if desired. If you don't know what a "705 file" is, see next paragraph 3.D.2 for file format info (in any case, we strongly advise you to use PLINK format, which is much more versatile!). No blank spaces are allowed. See the provided PARAMFILE.txt for an example.

- Variable **INPUT_705_MAP:** The map for each chip contained in the 705 files provided. Note that, conversely to what expected for PLINK format files, the number of files in the genotype and map in here may differ. However, similarly to PLINK, the order of SNPs in these files **must** be the same as in the genotype file *(Zanardi will not reorder SNPs based on SNP index; on the contrary it will assume the order is exactly as shown in the map, just as PLINK map files).* No blank spaces are allowed. See next paragraph 3.D.2 for file format and for more information. See also the provided PARAMFILE.txt for an example.

When a single Interbull 705 file is provided: A single 705 file can contain individuals genotyped with one or more arrays. Imagine you have a genotype 705 file: *geno705.txt*, which contains individuals genotyped with 3 different arrays. Therefore, 3 "index" or "map" files are required in the INPUT_705_MAP variable (one for each array, let's say: *Index1.txt*, *Index2.txt* and *Index3.txt*). All files are stored in a folder named /home/my705_files. The parameter file should look like this:

```
INPUT_705=/home/my705_files/geno705.txt
INPUT_705_MAP=/home/my705_files/Index1.txt,/home/my705_files/Index2.txt,/home/my705_files/Index3.txt
```

When multiple Interbull 705 files are provided: Similarly to what described for multiple PLINK files, multiple 705 files are accepted, by comma-separating the names of the files to be analysed.

PEDIGREE FILES (OPTIONAL)

- Variable **INPUT_PEDIG:** A single pedigree file. No blank spaces are allowed. All individuals in the genotype file(s) (and their ancestors) are expected to be in this file. ***The pedigree should be sorted from old to young. Note that all individuals not coded as missing (e.g. "0") are expected to be present as individuals (e.g. 1st column) in this file***. Zanardi will check the consistency of this file each time a pedigree-related option is called. See next paragraph 3.D.3 for file format and more information.

PHENOTYPE FILES (OPTIONAL)

- Variable **INPUT_PHENO:** A single phenotype (e.g. EBV) file. All individuals in the genotype file(s) are expected to be in this file. No blank spaces are allowed. See next paragraph 3.D.4 for file format and more information.

OUTPUT FILE NAME (OPTIONAL)

- Variable **OUTPUT_NAME:** The (user-defined) output name suffix. No blank spaces are allowed. Each analysis produces an output file with a *fixed* prefix (see Chapter 3.F for further info) and a user-defined suffix (optional).

3.C.3. PROGRAM OPTIONS

These will be thoroughly described in Section 3.F.

3.D. INPUT FILES FORMATS

3.D.1. PLINK PED/MAP FILE FORMATS:

Zanardi accepts single or multiple PLINK ped/map file couples, which should be included in the INPUT_PED and INPUT_MAP variables of the parameter file. For specifics about PLINK [ped/map] format, please visit PLINK software web page
http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#ped.
Note that [transposed/long/binary] PLINK formats are not currently accepted as input file in Zanardi.

3.D.2. 705 FILE FORMATS:

Generally, this is a format useful to cattle breeder associations. "Interbull 705" format is a file format for the International Genomic Evaluation. It is a known and common format for most major cattle breeder associations (at least those participating to the Gene2farm project). Zanardi accepts single or multiple "Interbull 705" files, which should be included in the INPUT_705 and INPUT_705_MAP variables of the parameter file. Please note that, conversely to PLINK, it is not required to have the same number of INPUT_705 and INPUT_705_MAP files, since multiple SNP arrays (e.g. densities) could be present in a single 705 file.

Genotypic files (the INPUT_705 files) contain general information of individuals and numerical coding for genotypes (see notes of the following table for more information).

Field	Start. pos.	Content	Format	Example
1	1	Record type	Int. 3	705
2	4	Country sending the info	Char. 3	ITA
3	7	Breed of evaluation	Char. 3	BSW
4	11	Breed of animal **(1)**	Char. 3	BSW
5	14	Country of first registration **(1)**	Char. 3	AUS
6	17	Sex **(1)**	Char. 1	M
7	18	ID number of individual **(1)**	Char. 12	00Z000000000
8	31	Number of SNPs in array	Int. 10	54001
9	42	Genotype for SNPs in Index	Int. X **(2)**	0/1/2.. **(3)**

Notes: 1) These are the elements of the International ID (Char. 19) 2) The number of integers has to be exactly the same as the number of SNPs in array (Field n.8) 3) Fields here are 1 number for each genotype, as follows: 0="BB",1="AB" or "BA", 2="AA", 5="Missing", 7=Imputed "BB" ,8=Imputed "AB" or "BA", 9=Imputed "AA"

Index – or MAP - files (INPUT_705_MAP) contain information regarding the SNPs (position, name, chromosome number, etc). **These files need to be sorted by SNP index for array.**

Field	Start. pos.	Content	Format	Example
1	1	SNP name	Char. 53	UA-IFASA-9433
2	54	SNP index for array **(1)**	Int. 6	777909
3	60	Chromosome	Int. 10	20
4	70	Physical position	Int. 15	15478658
5	85	Overall SNP index **(2)**	Int. 10	53947

Notes: 1) This field indicates the position of the SNP in Field n.9 of the genotype file 2) This is an across-array index used by Interbull. If you're not using Interbull Index files, set to "0" all this column (Zanardi will not use it!)

3.D.2BIS. COMBINATION OF PLINK AND 705 FILE FORMATS:

Zanardi can contemporarily handle both PLINK and 705 type of files. All the above applies for each of the file format, so please carefully follow the instructions given. **However, since Zanardi will convert Interbull 705 genotype coding into "A" and "B" alleles, PLINK files provided should have A/B coding as well, otherwise the program will stop!** If you need help doing this, please see chapter 3.F, as we also provide a "satellite" open-source software able to do all this for you! ☺

If both types of files are used contemporarily, Zanardi will first convert the 705 files in PLINK format (one PLINK file for each array), and then merge all files together. You do not need to specify a thing, Zanardi will do everything automatically – your only concern should be in providing the right input files!

3.D.3. PEDIGREE FILE FORMAT:

Pedigree files are required for some Zanardi options (e.g. *--pedigchk*, --gsprep, etc..).
The pedigree file must:

- be sorted from old to young,
- Include all individuals provided in the genotype file.
- if using --gsprep or --optiprep options, it must include all the individuals in the pedigree (e.g. all male/female parents that are not missing should be present as individuals with parents in the pedigree).

If any of the above condition is not met, Zanardi will stop (telling you what is wrong with your file). In addition, special requirements for some options will be explained in the options section in Chapter 3.E.
The pedigree file should contain 5 fields, semicolon (";") separated, for: Individual ID, Male parent ID, Female parent ID, Date of birth and sex (see table below). This file is "free" format, e.g. column length is not needed to be fixed. Note that although Zanardi accepts IDs with blank spaces, PLINK does not, so we strongly suggest you to avoid it! Missing/Unknown individuals should be coded as "0" or five or more "U" letters (e.g. "UUUUUUUUUUUUUUUUUU" is ok, "UUUU" is not).

Field	Content	Format	Example
1	Individual ID	ANY	Luke
2	Sire/Male parent ID	ANY	Anakin
3	Dam/Female parent ID	ANY	Padme
4	Date of birth	YYYYMMDD **(1)**	19820324
5	Sex (M or F)	Char. 1	M

Notes: 1) Depending on the option chosen, Zanardi will run a lenient or strict pedigree check. Pedigree is, in any case, assumed to be correct.

3.D.4. PHENOTYPE FILE FORMAT:

Similarly to the pedigree file, the phenotype file is required only for some Zanardi options (e.g. --gsprep, etc..). The only condition required for this file is that all individuals in the genotype file must be present in the phenotype file, and that it must contain 3 fields, semicolon (";") separated, for: Individual ID, Phenotype (e.g. EBV, Deregressed Proof, Yield deviation, etc…), Accuracy (currently used only by *--gsprep* option. It can be set to 0 for all other options).

3.E. ZANARDI USAGE

Once the parameter file is fully set, the only thing left to do is to run Zanardi from command line, choosing the appropriate analyses. The general usage is (a list of options can be found below):

```
python Zanardi.py [options]
```

To have a quick reminder of the options available, you can use an internal help routine:

```
python Zanardi.py -h
```

or

```
python Zanardi.py –help
```

Please remember that the user is expected to interact only with Zanardi's *parameter file* (PARAMFILE.txt), and its in/outputs.

3.F. OPTIONS

Except for a few cases, Zanardi can be run using single or multiple options contemporarily. When multiple options are chosen, Zanardi will streamline the work using the output of the first step as input of the second step, and so on.

Notation:

- **.(xxx)** = A set of file extensions that depend on the software used
- [< NAME >] = A user-defined < name > for folders / filename suffix, provided in the command line (options --outdir --tempdir) or in the parameter file (OUTPUT_NAME variable), respectively.

-h or --help options

MEANING:
This option produces a quick reference to the list of options available on screen.

SOFTWARE USED:
N/A

PARAMETER FILE OPTIONS:
N/A

INPUT FILE(S) REQUIRED:
N/A

OUTPUT FILE(S) PRODUCED:
N/A

--download option

MEANING:
This option is a stand-alone option. This means that, when invoked, it will terminate the program after running (even if other options are present). This option usually is run very few times (once?) to download the required software. The option calls a small bash script that automatically downloads the required software, uncompress/installs it and updates the link in the parameter file. Single or multiple software downloads can be required contemporarily. For multiple-software download, provide a comma-separated list of available software. Currently available options are: plink, beagle3, beagle4, fcgene and admixture. For example, the following command:

```
python Zanardi.py --download=beagle3,beagle4,fcGENE,PLINK,ADMIXTURE
```

will download the required software (all 5 programs), update the path for each software in the parameter file and quit Zanardi. Note the software names are is **not** case sensitive.

SOFTWARE USED:
Own code

PARAMETER FILE OPTIONS:
N/A

INPUT FILE(S) REQUIRED:
N/A

OUTPUT FILE(S) PRODUCED:
Each software downloaded is placed in a separated folder (except for BEAGLE v.3 and v.4, which are in the same "BEAGLE" folder) under the "UTILS" folder. Note that beagle .jar executables are automatically renamed to *beagle3.jar* and *beagle4.jar*.

!!! WARNING!!!
PLINK v1.9 is currently under development. This means the link provided with the current version of Zanardi may be (most probably is) broken and download app won't work correctly. If this happens, don't panic. Just go to PLINK v1.9 download page, copy the link of the Linux/Mac 64-bit (STABLE) download button (right click your mouse over the "download" word, and select "copy link" from the menu), and use that link to modify the variable PLINK (currently, row 24) in ***UTILS/ZANARDI_UTILS/download_app.sh***. This tedious procedure won't be necessary once PLINK v1.9 becomes PLINK v2.0.

--plinkqc option

MEANING:
This option runs a quality control over the data **after the merge step** (if more than one genotype file is provided). Actually it can also allow to streamline and integrate any PLINK functionality within Zanardi (*only for advanced users*, see PLINK_OTHOPT parameter variable).

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)

PARAMETER FILE OPTIONS:
QCCRATE_IND: call rate for individuals - values range from 0 to 1;
QCCRATE_SNP: call rate for SNPs - values range from 0 to 1;
QCMAF: Minor allele frequency - values range from 0 to 1;
QCHWE: Hardy-Weinberg Equilibrium - values range from 0 to 1;
QC_OTHOPT: (apply *any* other PLINK option using plink syntax – See PLINK manual for further info)

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:
Quality controlled PLINK file – *after merge* (PLINK format):

`[<OUTDIR>]/PLINK_OUT_[<FILENAME_SUFFIX>].(ped/map/log/...)`

--mds option

MEANING:
This option produces a Multi-dimensional Scaling plot (MDS) over all genotype samples provided.

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
R (with *ggplot2* package installed)

PARAMETER FILE OPTIONS:
MDSGROUPop: Group individuals based on the "FID" information of the PLINK file (e.g. first column of the PED file; usually used to provide breed information) – Accepted values Y or N. If Y is chosen, MDS values from all individuals with same FID are averaged together (resulting in a single "dot" for each FID. If N is chosen, then all individuals are plotted, irrespectively of their FID.

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:
PLINK MDS step (PLINK format):

`[<TEMPDIR>]/MDSPLOT.(ped/map/log)`

MDS R script (plain text):

`[<TEMPDIR>]/mds_plot.R`

MDS plot (in .pdf format):

`[<OUTDIR>]/MDS_PLOT_(Inds/Pop)_[<FILENAME_SUFFIX>].pdf`

--pedigchk option

MEANING:
This option runs a mendelian inheritance check (e.g. opposing homozygous in close relatives) among all genotyped samples, following the pedigree file provided by the user.

****IMPORTANT NOTE**** The total number of SNPs considered and, as a consequence, also the final % of mendelian errors, are those autosomal SNPs that are homozygous in BOTH individuals. For example, if:

individual (1) genotypes are : AA AB 00 (hom AA for SNP1, het on SNP2, and missing on SNP3)
individual (2) genotypes are : BB BB BB (hom BB for SNPs1,2 and 3)

, then the % of mendelian error will be 100%, as only the first SNP is considered for calculation and SNP1 is an opposing homozygote (AA in Ind1 and BB in Ind2).

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
Own code for pedigree check.

PARAMETER FILE OPTIONS:
PDSKIPCOUPLE: To avoid useless calculations, samples already controlled can be skipped from the pedigree check, using this variable. The format required is the same as the PEDIGCHK_pass.txt file (see OUTPUT FILES PRODUCED) section below for more information.
PDMEND_THRES: This is a required parameter, values ranging between 0 and 1, indicating the mendelian inheritance error rate threshold (e.g. a value of .02 means that all individuals with more than 2% of mendelian inheritance error rate will be considered as failing samples).
PDBESTALL:This is a required parameter, and must be "Y" or "N". "Y" means that the best match for all individuals failing the pedigree check will be searched for in the full genotype file (WARNING: *highly time consuming if large number of samples genotyped or large number of individuals fail!*). "N" means that the search will be restricted to all individuals of the same sex of the failing parent (e.g. if the sire fails, all female individuals will not be checked) and to all individuals born *before* the target individual itself. Therefore, if you're not so sure of the accuracy of your pedigree file, use "Y".

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)
A pedigree file (INPUT_PEDIG)

OUTPUT FILE(S) PRODUCED:

- Samples passing pedigree check:
  `[<OUTDIR>]/PEDIGCHK_pass.txt`

Field	Content	Example
1	ID son (individual)	LukeSkywalker
2	ID parent	Anakin
3	Type of relationship [SIRE/DAM]	SIRE
4	Mend. Errors/Total nonmissing SNPs	0/1138
5	Mendelian error (%)	0.00000%
6	Sample similarity (%)	0.00000%

Semicolon-separated file, with header.
NOTE: this file (or files with an identical trace) can be used in subsequent runs to skip already checked couples (PDSKIPCOUPLE variable in parameter file).

- Samples failing pedigree check:
  `[<OUTDIR>]/PEDIGCHK_fail.txt`
  Trace is identical to PEDIGCHK_pass.txt (except for last column)

- (if at least 1 failing sample) Best matching (e.g. potential) parents for failing samples:
  `[<OUTDIR>]/PEDIGCHK_bestmatch.txt`

Field	Content	Example
1	Candidate # / Total candidates	1/1
2	ID son (individual)	LukeSkywalker
3	ID *POTENTIAL* parent	Anakin
4	Type of relationship [SIRE/DAM]	SIRE
5	Mendelian error (%)	0.00000%
6	Sample similarity (%)	0.00000%

NOTE: If no plausible best match is found, field 3 will be set to "---" and field 5 will be set to "-999"

--mendchk option

MEANING:
This option runs an UNINFORMATIVE (e.g. all genotyped samples are cross-checked) mendelian inheritance check (e.g. opposing homozygous in close relatives). Usually, many users need to check their genotypes in search of unexpected replicates or close-relatives (e.g. sampling/lab errors). This option allows to cross-check all individuals, and report only those below a certain user-defined threshold (see PARAMETER FILE OPTIONS).

IMPORTANT NOTE The total number of SNPs considered and, as a consequence, also the final % of mendelian errors, are those autosomal SNPs that are homozygous in BOTH individuals. For example, if:

individual (1) genotypes are : AA AB 00 (hom AA for SNP1, het on SNP2, and missing on SNP3)
individual (2) genotypes are : BB BB BB (hom BB for SNPs1,2 and 3)

, then the % of mendelian error will be 100%, as only the first SNP is considered for calculation and SNP1 is an opposing homozygote (AA in Ind1 and BB in Ind2).

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015);
Own code for mendelian error check.

PARAMETER FILE OPTIONS:
MENDERR_THRES: This is a required parameter, values ranging between 0 and 1, indicating the mendelian inheritance error rate threshold (e.g. a value of .02 means that all individuals with more than 2% of mendelian inheritance error rate will be considered as non related samples, and thus will not be reported).

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:
Samples passing pedigree check: `[<OUTDIR>]/MENDCHK_lessTHRESHOLD.txt`

Field	Content	Example
1	ID 1	LukeSkywalker
2	ID 2	Anakin
3	Mend. Errors/Total nonmissing SNPs	0/1138
4	Mendelian error (%)	0.00000%
5	Sample similarity (%)	0.00000%

Semicolon-separated file, with header.

--beagle3 option

MEANING:
This option will run an imputation step using Beagle v.3.

SOFTWARE USED:
FCgene (Roshyara and Scholz, 2014)
Beagle v.3 (Browning and Browning, 2007)

PARAMETER FILE OPTIONS:
`BGMEMORY:` virtual memory allocated to the process, in MB - default "2000" (e.g. 2Gb)
`BG3_MISSING:` Missing allele coding - default "0"
`BG_OTHOPT:` (OPTIONAL) similarly to PLINK, apply other BEAGLE options (using beagle v.3 syntax. See Beagle v.3 manual for further info).

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:

- Conversion from PLINK to BEAGLEv.3 format:
  `[<TEMPDIR>]/`**`PLINK_beagle.`**`(bgl/_fcgene.log/..)`
- Beagle run output (BEAGLE v.3 OUTPUT FILE FORMAT):
  `[<OUTDIR>]/`**`BEAGLE_OUT.`**`(dose.gz/gprobs.gz/phased.gz/..)`
- Conversion from BEAGLEv.3 to PLINK format:
  `[<OUTDIR>]/`**`BEAGLE_OUT_`**`[<FILENAME_SUFFIX>].(ped/map/log)`

--beagle4 option

MEANING:
This option will run an imputation step using Beagle v.4.
IMPORTANT: If *phase* information is required by the user, convert Beagle v.4 format file output on your own. PLINK conversion (e.g. form VCF to PLINK PED/MAP) does not maintain the phase information from Beagle files.

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
Beagle v.4 (Browning and Browning, 2009)

PARAMETER FILE OPTIONS:
BGMEMORY: virtual memory allocated to the process, in MB - default "2000" (e.g. 2Gb)
BG_OTHOPT: (OPTIONAL) similarly to PLINK, apply other BEAGLE options (using beagle v.4 syntax. See Beagle v.4 manual for further info).

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)
NOTE: beagle v.4 allows the use of pedigree information (which speeds things up). This option can be included using BG_OTHOPT variable.

OUTPUT FILE(S) PRODUCED:

- Conversion from PLINK to VCF format:
  `[<TMPDIR>]/beagle4_infile.vcf`
- Beagle run output:
  `[<TMPDIR>]/result_beagle4.vcf.gz`
- Conversion from VCF to PLINK format:
  `[<OUTDIR>]/BEAGLE_OUT_[<FILENAME_SUFFIX>].(ped/map/log)`

--fimpute option

WARNING: FImpute is open-access for all *research* use, whereas it is NOT for commercial use. Please keep this in mind when using this option!

MEANING:
This option will run an imputation step using FImpute.
IMPORTANT: If *phase* information is required by the user, convert FImpute format file output on your own. PLINK conversion (e.g. form PLINK PED/MAP to FImpute format) does not maintain the phase information from FImpute files.

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
FImpute (Sargolzaei et al., 2014)

PARAMETER FILE OPTIONS:
FMP_NJOB: Number of jobs to be run in parallel - default "1"
FMP_OTHOPT: (OPTIONAL) similarly to PLINK, apply other FImpute options. Any other option except for the inclusion of numbers of jobs and the presence of input files are allowed here (using FImpute syntax separated by ";". See FImpute manual for further info).

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)
NOTE: FImpute allows the use of pedigree information (which speeds things up). This option can't be included using FMP_OTHOPT variable. If the field INPUT_PEDIG filled, the pedigree is loaded automatically.

OUTPUT FILE(S) PRODUCED:

- Conversion in 12 PLINK format:
  `[<TMPDIR>]/FIMPUTE_recode12.(ped/map/log)`
- Conversion from PLINK to FImpute format:
  `[<TMPDIR>]/genotype_[<FILENAME_SUFFIX>].FM`
  `[<TMPDIR>]/snp_info_[<FILENAME_SUFFIX>].FM`
- Parameter file for FImpute:
  `[<TMPDIR>]/param_FImpute_[<FILENAME_SUFFIX>].FM`
- Allele frequency using PLINK:
  `[<TMPDIR>]/ freqACGT.(frq/nosex/log)`
- FImpute run output folder:
  `[<OUTDIR>]/output_FImpute_[<FILENAME_SUFFIX>]`
- Conversion from FImpute to PLINK format:
  `[<OUTDIR>]/FIMPUTE_[<FILENAME_SUFFIX>].(ped/map/log)`

--haprep [BSW/FLK] option

MEANING:
This option is standalone (program exits after running), it is intended for 2 COW breeds only: Brown Swiss (BSW) and Flekvieh (FLK). This option will prepare the input file for a web service able to predict if individuals are carriers (or not) of a breed-specific haplotype linked to reduced fertility. This option works only with BovineSNP50 v.2 array (it automatically selects BTA19) and requires raw genotypes, as it selects the SNPs used in the training model (~300 for BSW, ~1100 for FLK). An imputation step (only on BTA19) is run, as the machine learning algorithm used on the post-hoc analysis does not accept missing genotypes. The output file produced by Zanardi is the input file for the web-app that runs the analysis: https://stebif68.shinyapps.io/EzeApp

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
Beagle v.4 (Browning and Browning, 2009)
Own code

PARAMETER FILE OPTIONS:
N/A

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:

- Reduction of dimension of genotypes (only BTA19):
  `[<TMPDIR>]/PLINK_HAPREP.(ped/map/log)`
- Conversion from PLINK to VCF format:
  `[<TMPDIR>]/beagle4_infile.vcf`
- Beagle run output:
  `[<TMPDIR>]/result_beagle4.vcf.gz`
- Conversion from VCF to PLINK format:
  `[<OUTDIR>]/BEAGLE_OUT[<FILENAME_SUFFIX>].(ped/map/log)`
- Reduce number of SNPs required in output:
  `[<TMPDIR>]/HAPLO_small.txt`
  `[<TMPDIR>]/PLINK_HAPREP.ped`
- Final output:
  `[<OUTDIR>]/HAPREP[<FILENAME_SUFFIX>].txt`

--roh option

MEANING:
This option will search for Runs of Homozygosity individual- and chromosome-wise. Conversely to PLINK --roh option, this is a variable length ROH procedure (e.g. avoids the fixed sliding window procedure). A plot by FID (e.g. first column in PLINK PED file, usually used to identify the breed) and by chromosome is produced using R (+ggplot2 package).

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
Own code (Marras et al., 2015)

PARAMETER FILE OPTIONS:
ROH_SNP: Minimum number of SNP for each ROH (e.g. if a ROH has less than this number is not accounted for).
ROH_MAXMIS: Maximum number of missing SNP per ROH (e.g. tolerance for this number of missing SNPs in a ROH).
ROH_MAXHET: Maximum number of heterozygous SNP per ROH (e.g. tolerance for this number of heterozygous SNPs – usually used to account for genotyping call error).
ROH_MINLEN: Minimum length - in Mb - of ROH

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:

- Conversion to 1/2 allele format (PLINK v1.9):
  `[<TMPDIR>]/ROH_recode12.(ped/map)`
- ROH reads output:
  `[<OUTDIR>]/ROH_reads_[<FILENAME_SUFFIX>].txt`

Field	Content	Example
1	FID (or breed)	Jedi
2	ID individual	LukeSkywalker
3	Chromosome	1
4	Count (# SNPs in ROH)	500
5	Start (ROH bp start)	1000
6	End (ROH bp end)	50000
7	Length (ROH length)	49000

- ROH R script (PLAIN TEXT):
  `[<TMPDIR>]/roh_plot.R`
- ROH plot:
  `[<OUTDIR>]/ROH_plot_[<FILENAME_SUFFIX>].pdf`

--froh option

MEANING:
This option will search for Runs of Homozygosity individual- and chromosome-wise with the objective of obtaining ROH-based inbreeding coefficients. A file including total inbreeding and chromosome-wise inbreeding indexes is provided (by individual), as long as all output files from --roh option.

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
Own code (Marras et al., 2015)

PARAMETER FILE OPTIONS:
ROH_SNP: Minimum number of SNP for each ROH (e.g. if a ROH has less than this number is not accounted for).
ROH_MAXMIS: Maximum number of missing SNP per ROH (e.g. tolerance for this number of missing SNPs in a ROH).
ROH_MAXHET: Maximum number of heterozygous SNP per ROH (e.g. tolerance for this number of heterozygous SNPs – usually used to account for genotyping call error).
ROH_MINLEN: Minimum length - in Mb - of ROH

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:

- Conversion to 1/2 allele format (PLINK v1.9):
  `[<TMPDIR>]/ROH_recode12.(ped/map)`
- ROH-based inbreeding coefficients:
  `[<OUTDIR>]/ROH_inbreeding_[<FILENAME_SUFFIX>].txt`

Field	Content	Example
1	FID (or breed)	Jedi
2	ID individual	LukeSkywalker
3	Total inbreeding coeff. (FROH)	0.0001
4-33	Chrom-wise FROH (1-29)	0.1; ... ; 0.1

--admixture option

MEANING:
This option runs an ADMIXTURE analysis over all the gentoyped individuals (e.g. similar to STRUCTURE but optimized for whole-genome SNPs). For further information on ADMIXTURE: https://www.genetics.ucla.edu/software/admixture/admixture-manual.pdf

SOFTWARE USED:
PLINK v1.9 (Chang et al., 2015)
Admixture (Alexander and Lange, 2011)
R (with *ggplot2* package installed)

PARAMETER FILE OPTIONS:
ROH_SNP: Minimum number of SNP for each ROH (e.g. if a ROH has less than this number is not accounted for).
ADM_KVALUE: This refers to the number of maximum K runs. K is, in (extremely) simple words, what Admixture uses to cluster individuals assuming K populations. Zanardi will run all K's from 2 to the desired K. Broadly, if number of population to be analysed is known, then choose a *K=known_pops+3*. In order to chose the most suitable K for your dataset, see the lowest CV value in the CV plot (created automatically in Zanardi).
ADM_CORE: number of processors used in the calculation (the higher, the lower the processing time.
ADM_CV: number of cross-validations (general rule: the higher, the better. However, this value is proportional to the processing time.

INPUT FILE(S) REQUIRED:
Any genotype and map input file (INPUT_PED+INPUT_MAP and/or INPUT_705+INPUT_705_MAP)

OUTPUT FILE(S) PRODUCED:

- Conversion to 1/2 allele format (PLINK v1.9):
  `[<TMPDIR>]/Admixture_[<FILENAME_SUFFIX>]_K.(ped/map)`
- ADMIXTURE run (for i=2,K)
  `[<OUTDIR>]/Admixture_[<FILENAME_SUFFIX>]_K.[i].(Q/P)`
- ADMIXTURE CV plot
  `[<OUTDIR>]/Admixture_CVplot_[<FILENAME_SUFFIX>].pdf`
- ADMIXTURE BAR plot (one file, multiple pages, one for each K)
  `[<OUTDIR>]/Admixture_BARplot_[<FILENAME_SUFFIX>].pdf`

3.G. "SATELLITE" SOFTWARE

Here we cite a few self developed software that you might find useful. We cite these not because we like being auto-referencing, but because both software are specifically linked to the PLINK format and can save you a LOT of time and problems solving issues using tools already developed.

- SNPConvert. Availablility: https://github.com/nicolazzie/SNPConvert . This is a suite of tools, provided both as source codes (any OS, dependency: Python 2.7) or as GUI (Windows and Mac, 64bit).
    - *PEDDA_ROW.* This program is an open-source python program able to convert the Illumina FinalReport file in ROW format into a ped/map (PLINK) file format. Only 2 input files (the original FinalReport files and the SNP map file) and the setting of 8 simple parameters are required to run the program.
    - *PEDDA_MATRIX.* This program is an open-source python program able to convert the Illumina FinalReport file in MATRIX format into a ped/map (PLINK) file format. Only 2 input files (the original FinalReport files and the SNP map file) and the setting of 5 simple parameters are required to run the program.
    - *iConvert*. This program is an open-source python program able to convert FORWARD, TOP and A/B allele coding formats and, if required, update SNP map information. Essentially, users are required to download a SNPchimp v.3 (Nicolazzi et al., 2015; http://bioinformatics.tecnoparco.org/SNPchimp) file with the allele coding required (e.g. input and output, for example: forward and top) and, after setting simple preferences in the parameter file, the program will convert the allele coding as desired. Input file is PLINK PED/MAP format, so it is fully compatible with Zanardi.
- *AffyPipe.* Availability: https://github.com/nicolazzie/AffyPipe.git (Nicolazzi et al., 2014). This program is for Affymetrix users and, again, it is an open-source python program able to perform the extraction of genotypes from raw Affymetrix files, providing as output a PLINK PED/MAP file. Therefore, again, this is fully compatible with Zanardi.

Contributions from the community for these two softwares and for Zanardi itself are highly encouraged!

CHAPTER 5. REFERENCES

Alexander, D.H., and K. Lange. 2011. Enhancements to the ADMIXTURE algorithm for individual ancestry estimation. *BMC Bioinformatics*. 12:246. doi:10.1186/1471-2105-12-246.

Browning, B.L., and S.R. Browning. 2009. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *Am. J. Hum. Genet.* 84:210–23. doi:10.1016/j.ajhg.2009.01.005.

Browning, S.R., and B.L. Browning. 2007. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *Am. J. Hum. Genet.* 81:1084–97. doi:10.1086/521987.

Chang, C.C., C.C. Chow, L.C. Tellier, S. Vattikuti, S.M. Purcell, and J.J. Lee. 2015. Second-generation PLINK: rising to the challenge of larger and richer datasets. *Gigascience*. 4:7. doi:10.1186/s13742-015-0047-8.

Marras, G., G. Gaspa, S. Sorbolini, C. Dimauro, P. Ajmone-Marsan, A. Valentini, J.L. Williams, and N.P.P. Macciotta. 2015. Analysis of runs of homozygosity and their relationship with inbreeding in five cattle breeds farmed in Italy. *Anim. Genet.* 46:110–21. doi:10.1111/age.12259.

Nicolazzi, E.L., A. Caprera, N. Nazzicari, P. Cozzi, F. Strozzi, C. Lawley, A. Pirani, C. Soans, F. Brew, H. Jorjani, G. Evans, B. Simpson, G. Tosser-Klopp, R. Brauning, J.L. Williams, and A. Stella. 2015. SNPchiMp v.3: integrating and standardizing single nucleotide polymorphism data for livestock species. *BMC Genomics*. 16:1–6. doi:10.1186/s12864-015-1497-1.

Nicolazzi, E.L., D. Iamartino, and J.L. Williams. 2014. AffyPipe: an open-source pipeline for Affymetrix Axiom genotyping workflow. *Bioinformatics*. btu486–.

Roshyara, N.R., and M. Scholz. 2014. fcGENE: a versatile tool for processing and transforming SNP datasets. *PLoS One*. 9:e97589. doi:10.1371/journal.pone.0097589.

Sargolzaei, M., J.P. Chesnais, and F.S. Schenkel. 2014. A new approach for efficient genotype imputation using information from relatives. *BMC Genomics*. 15:478. doi:10.1186/1471-2164-15-478.