Contents

1 Guide for xerxes v0.2.0.0 1
1.1 Fstats command 1
1.1.1 Allowed statistics 3
1.1.2 Defining statistics directly via `--stat` 3
1.1.3 Defining statistics in a simple text file 3
1.1.4 Input via a configuraton file 4
1.1.5 Output 5
1.1.6 Whitepaper 5
1.2 RAS (in development) 6

1 Guide for xerxes v0.2.0.0

1.1 Fstats command

Xerxes allows you to analyse genotype data across poseidon packages, including your own, as explained above by “hooking” in your own package via a `--baseDir` (or `-d`) parameter. This has the advantage that you can compute arbitrary F-Statistics across groups and individuals distributed in many packages, without the need to explicitly merge the data first. Xerxes also takes care of merging PLINK and EIGENSTRAT data on the fly. It also takes care of different genotype base sets, like Human-Origins vs. 1240K. It also flips alleles automatically across genotype files, and throws an error if the alleles in different packages are incongruent with each other. Xerxes is also smart enough to select only the packages relevant for the statistics that you need, and then streams through only those genotype data.

Here is an example command for computing several F-Statistics:

```
xerxes fstats -d ... -d ... \
  --stat "F4(<Chimp.REF>, <Altai_published.DG>, Yoruba, French)" \
  --stat "F3(<Chimp.REF>, <Altai_snpAD.DG>, Spanish)" \
  --statFile fstats.txt
  --statConfig fstats.yaml
  -f outputfile.txt
```

First, the two options `-d ...` exemplify that you need to provide at least one base directory for poseidon packages, but can also give multiple. Second, F-Statistics can be entered in three different ways:

1. Directly via the command line using `--stat`.
2. Using a simple text file using `--statFile`
3. Using a powerful configuration file that allows more options.

These three input ways can be mixed and matched, and given multiple times. They are explained below.

Last, option `-f` can be used to write the output table into a tab-separated text file, beyond just printing a table into the standard out when the program finishes. Note that there are more options, which you can view using `xerxes fstats --help`:

```
Usage: xerxes fstats (-d|--baseDir DIR) [-j|--jackknife ARG]
                     [-e|--excludeChroms ARG]
                     (--stat ARG | --statConfig ARG | --statFile ARG)
                     [--noTransitions] [-f|--tableOutFile ARG]
                     [--blockTableFile ARG]

  Compute f-statistics on groups and invidiuals within and across Poseidon
  packages

Available options:
  -h,--help                Show this help text
  -d,--baseDir DIR         A base directory to search for Poseidon packages.
  -j,--jackknife ARG       Jackknife setting. If given an integer number, this
                           defines the block size in SNPs. Set to "CHR" if you
                           want jackknife blocks defined as entire chromosomes.
                           The default is at 5000 SNPs
  -e,--excludeChroms ARG   List of chromosome names to exclude chromosomes,
                           given as comma-separated list. Defaults to X, Y, MT,
                           chrX, chrY, chrMT, 23,24,90
  --stat ARG               Specify a summary statistic to be computed. Can be
                           given multiple times. Possible options are: F4(a, b,
                           c, d), F3(a, b, c), F3star(a, b, c), F2(a, b), PWM(a,
                           b), FST(a, b), Het(a) and some more special options
                           described at
                           https://poseidon-framework.github.io/#/xerxes?id=fstats-command.
                           Valid entities used in the statistics are group names
                           as specified in the *.fam, *.ind or *.janno failes,
                           individual names using the syntax "<Ind_name>", so
                           enclosing them in angular brackets, and entire
                           packages like "*Package1*" using the Poseidon package
                           title. You can mix entity types, like in
                           "F4(<Ind1>,Group2,*Pac*,<Ind4>)". Group or individual
                           names are separated by commas, and a comma can be
                           followed by any number of spaces.
  --statConfig ARG         Specify a yaml file for the Fstatistics and group
                           configurations
  --statFile ARG           Specify a file with F-Statistics specified similarly
                           as specified for option --stat. One line per
                           statistics, and no new-line at the end
  --maxSnps ARG            Stop after a maximum nr of snps has been processed.
                           Useful for short test runs
  --noTransitions          Skip transition SNPs and use only transversions
  -f,--tableOutFile ARG    a file to which results are written as tab-separated
                           file
  --blockTableFile ARG     a file to which the per-Block results are written as
                           tab-separated file
```

1.1.1 Allowed statistics

The following statistics are allowed in the `--stat`, `--statFile` and `--statConfig` options. In all of the following, symbols `a`, `b`, `c` or `d` stand for arbitrary entities allowed in Poseidon, so groups (such as `French`), individuals (such as `<MA1.SG>`) or packages (such as `*2012_PattersonGenetics*`).

- `F2vanilla(a, b)`: F2-Statistics - Vanilla version. Computed using `F2vanilla(a, b) = (a-b)^2` across the genome.
- `F2(a, b)`: F2-Statistics (bias-corrected version). Computed as `F2(a, b) = F2vanilla(a, b) - hA/sA - hB/sB`, where where `sA` is the number of non-missing alleles in entity A, and `hA = nA * nA' / sA * (sA - 1)` is an estimator of half the heterozygosity (see `Het(a)`), and likewise for `sB` and `nB` etc.
- `F3vanilla(a,b,c)`: F3-Statistics - Vanilla version, recommended if used as Outgroup-F3 statistics or with group c being pseudo-haploid: Are computed as `F3(a, b, c) = (c-a)(c-b)` across all SNPs.
- `F3(a,b,c`: F3-statistics (bias-corrected version). Computed as `F3(a, b, c) = F3vanilla(a, b) - hC/sC`.
- `F3star(a,b,c)`: F3-Statistics as defined in Patterson et al. 2012 - normalised and bias-corrected version, recommended for Admixture-F3 tests. Are computed by i) first substracting per SNP from the vanilla-F3 statistic a bias-correction term hC/sC, as above for F2, and ii) then normalising the genome-wide estimate by a genome-wide estimate of the heterozygosity of entity C (`Het(c)`), in order to make results comparable between different groups C (see Patterson et al., Genetics, 2012)
- `F4(a,b,c,d)`: F4 statistics. Are computed by averageing the quantity (a-b)(c-d) across all SNPs. No bias correction is necessary for this statistic.
- `Het(a)`: An estimate of the heterozygosity across all SNPs, computed as `2*hA`, with `hA` defined as above in `F2`
- `FST(a, b)`: An estimate of FST across the genome, following the formular from Appendix A in Patterson et al. 2012, which is a ratio of two terms, with numerator being `F2(a, b)` including bias correction, and the denominator being `F2(a, b) + hA + hB` including bias correction and `hA` and `hB` defined as above.
- `PWM(a, b)`: The pairwise mismatch rate between entities a and b, computed from allele frequencies as `a (1 - b) + (1 - a) b`.

All of these equations are from Patterson, Nick, Priya Moorjani, Yontao Luo, Swapan Mallick, Nadin Rohland, Yiping Zhan, Teri Genschoreck, Teresa Webster, and David Reich. 2012. “Ancient Admixture in Human History.” Genetics 192 (3): 1065–93. See also Appendix A of this paper for the unbiased estimators used above.

For each of the “slots” A, B, C or D, you can enter: * Individuals, using the syntax `<Individual_Name>` * Groups, using no special syntax “Group_Name” * Packages, using syntax `*Package_Name*` (This can be useful if you happen to have a homogenous set of individuals from multiple groups in one package and want to consider all of these as one group.)

1.1.2 Defining statistics directly via `--stat`

This is the simples option to instruct the program to compute a specified statistic. Each statistic requires a separate input using `--stat` using this input method. Example:

```
xerxes fstats -d ... -d ... --stat "F3(French, Spanish, <Chimp.REF>) --stat "FST(French,
Spanish)"
```

1.1.3 Defining statistics in a simple text file

You can prepare a text file, into which you write the above statistics, one statistics per line. Example:

```
F4(<Chimp.REF>, <Altai_published.DG>, Yoruba, French)
F4(<Chimp.REF>, <Altai_snpAD.DG>, Spanish, French)
F4(Mbuti,Nganasan,Saami.DG,Finnish)
```

you can then load these statistics using the option `--statFile fstats.txt`.

1.1.4 Input via a configuraton file

This is the most powerful way to input F-Statistics. Here is an example:

```
groupDefs:
  CEU2: ["CEU.SG", "-<NA12889.SG>", "-<NA12890.SG>"]
  FIN2: ["FIN.SG", "-<HG00383.SG>", "-<HG00384.SG>"]
  GBR2: ["GBR.SG", "-<HG01791.SG>", "-<HG02215.SG>"]
  IBS2: ["IBS.SG", "-<HG02238.SG>", "-<HG02239.SG>"]
fstats:
- type: F2
  a: ["French", "Spanish"]
  b: ["Han", "CEU2"]
  # Ascertainment is optional
- type: F3 # This will create 3x2x1 = 6 Statistics
  a: ["French", "Spanish", "Mbuti"]
  b: ["Han", "CEU2"]
  c: ["<Chimp.REF>"]
  ascertainment:
    outgroup: "<Chimp.REF>" # ascertaining on outgroup-polarised derived allele frequency
    reference: "CEU2"
    lower: 0.05
    upper: 0.95
- type: F4 # This will create 5x5x4x1 = 100 Statistics
  a: ["<I0156.SG>", "<I0157.SG>", "<I0159.SG>", "<I0160.SG>", "<I0161.SG>"]
  b: ["<I0156.SG>", "<I0157.SG>", "<I0159.SG>", "<I0160.SG>", "<I0161.SG>"]
  c: ["CEU2", "FIN2", "GBR2", "IBS2"]
  d: ["<Chimp.REF>"]
  ascertainment:
    # A missing outgroup means: ascertain on minor allele frequency
    reference: "CEU.SG"
    lower: 0.00
    upper: 0.10
```

The top level structure of this YAML file is an object with two fields: `groupDefs` (which is optional) and `fstats` (which is mandatory).

1.1.4.1 Group Definitions You can specify adhoc group definitions using the syntax above. Every group consists of a name (used as object key) and then a JSON- or YAML-list of signed entities, following the same syntax of `trident forge` (see trident). Briefly: Individuals, Groups and Packages can be added or excluded (prefixed by a `-`) in order. In the example above, two individuals are removed from each group.

Note that currently, groups can be defined only independently, so not incremental to each other. That means, you cannot currently use an already defined new group name in the entity list of a following group name.

1.1.4.2 Statistic input using YAML

Each statistic defined in the `fstats` section of the YAML file, actually defines a loop over multiple populations in each statistic. In the example above, there are 6 F3-Statistics, each using a different combination of the input groups defined in each of the `a:`, `b:` and `c:` slots. There are also 100 (!) F4 statistics, following all combinations of 5x5x4x1 slots defined in `a:`, `b:`, `c:` and `d:`. This makes it very convenient to loop over statistics.

1.1.4.3 Ascertainment (experimental feature)

In addition, every statistic section allows for a definition of an ascertainment specification, using a special key `ascertainment:`, which is optional. If given, you can specify an optional `outgroup`, a `reference` group in which to ascertain SNPs, and `lower` and `upper` allele frequency bounds. If specified, only SNPs for which the `reference` group has an allele frequency within the given bounds are used to compute the statistic (note that normalisation is still using all non-missing SNPs for that given statistic). If an `outgroup` is defined, then the outgroup-polarised derived allele frequency is used. If no `outgroup` is defined, then the minor allele frequency is used instead. If an outgroup is defined, any sites where the outgroup is polymorphic are treated as missing.

You can save this into a text file, for example named `fstats_config.yaml`, and load it via `--statConfig fstats_config.yaml`.

1.1.5 Output

The final output of the `fstats` command looks like this:

```
.-----------.-----------.---------------.--------.-------.---------.----------------.------------------
| Statistic |     a     |       b       |   c    |   d   | NrSites | Estimate_Total | Estimate_Jackknif
:===========:===========:===============:========:=======:=========:================:==================
| F3        | French    | Italian_North | Mbuti  |       | 593124  | 5.9698e-2      | 5.9698e-2
| F3        | French    | Han           | Mbuti  |       | 593124  | 5.0233e-2      | 5.0233e-2
| F3        | Sardinian | Pima          | French |       | 593124  | -1.2483e-3     | -1.2483e-3
| F4        | French    | Russian       | Han    | Mbuti | 593124  | -1.6778e-3     | -1.6778e-3
| F4        | Sardinian | French        | Pima   | Mbuti | 593124  | -1.4384e-3     | -1.4384e-3
'-----------'-----------'---------------'--------'-------'---------'----------------'------------------
```

which lists each statistic, the slots a, b, c and d, the number of sites with non-missing data for that statistic, Ascertainment information (outgroup, reference, lower and upper bound, if given), the genome-wide estimate, its standard error and its Z-score. If you specify an output file using option `--tableOutFile` or `-f`, these results are also written as tab-separated file.

Additionally, an option `--blockOutFile` can be specified, to which then a table with estimates per Jackknife block is written.

1.1.6 Whitepaper

The repository comes with a detailed whitepaper that describes some more mathematica details of the methods implemented here.

1.2 RAS (in development)

The RAS command computes pairwise RAS statistics between a collection of “left” entities, and a collection of “right” entities. Every Entity is either a group name or an individual, with the similar syntax as in F-statistics above, so `French` is a group, and `<IND001>` is an individual.

The input of left-pops and right-pops uses a YAML file via `--popConfigFile`. Here is an example:

```
groupDefs:
  group1: a,b,-c,-<d>
  group2: e,f,-<g>
popLefts:
- <I13721>
- <I14000>
- <I13722>
- <Iceman.SG>
popRights:
- Mbuti
- Mixe
- Spanish
outgroup: <Chimp.REF>
```

In this case, two groups are defined on the fly: `group1` comprises groups `a` and `b`, but excludes group `c` and individual `d`. Note that inclusions and exclusions are executed in order. `group2` comprises of group `e` and group `f`, but excludes individual `<g>`.

As in RAScalculator, the allele frequency ascertainment is done across right populations only.

The are a couple of optons, as specified in the CLI help (`xerxes ras --help`):

```
Usage: xerxes ras (-d|--baseDir DIR) [-j|--jackknife ARG]
                  [-e|--excludeChroms ARG] --popConfigFile ARG
                  [-k|--maxAlleleCount ARG] [-m|--maxMissingness ARG]
                  (-f|--tableOutFile ARG)
  Compute RAS statistics on groups and individuals within and across Poseidon
  packages

Available options:
  -h,--help                Show this help text
  -d,--baseDir DIR         a base directory to search for Poseidon Packages
                           (could be a Poseidon repository)
  -j,--jackknife ARG       Jackknife setting. If given an integer number, this
                           defines the block size in SNPs. Set to "CHR" if you
                           want jackknife blocks defined as entire chromosomes.
                           The default is at 5000 SNPs
  -e,--excludeChroms ARG   List of chromosome names to exclude chromosomes,
                           given as comma-separated list. Defaults to X, Y, MT,
                           chrX, chrY, chrMT, 23,24,90
  --popConfigFile ARG      a file containing the population configuration
  -k,--maxAlleleCount ARG  define a maximal allele-count cutoff for the RAS
                           statistics. (default: 10)
  -m,--maxMissingness ARG  define a maximal missingness for the right
                           populations in the RAS statistics. (default: 0.1)
  -f,--tableOutFile ARG    the file to which results are written as
                           tab-separated file
```

The output gives both cumulative (up to allele-count k) and and per-allele-frequency RAS (for allele count k) for every pair of left and rights. The standard out contains a pretty-printed table, and in adition, a tab-separated file is written to the file specified using option `-f`.

`xerxes ras` makes a few important assumptions: 1) It assumes that the Right Populations are "nearly" completely non-missing. Any allele that is actually missing from the rights is in fact treated as homozygous-reference! A different approach would be to compute the actual frequencies on the non-missing right alleles, but then we cannot anymore nicely accumulate over different ascertainment allele counts. 2) If no outgroup is specified, the ascertainment operates on minor-allele frequency (as in fstats) 3) If an outgroup is specified and missing from a SNP, or if the SNP is polymorphic, the SNP is skipped as missing