

Contents

1	Guide for trident v1.2.0.0 to v1.2.1.0	1
1.1	The trident CLI	1
1.1.1	General notes	3
1.2	Package creation and manipulation commands	4
1.2.1	Init command	4
1.2.2	Fetch command	5
1.2.3	Forge command	6
1.2.4	Genoconvert command	12
1.2.5	Update command	13
1.3	Inspection commands	15
1.3.1	List command	15
1.3.2	Summarise command	16
1.3.3	Survey command	17
1.3.4	Validate command	17

1 Guide for trident v1.2.0.0 to v1.2.1.0

1.1 The trident CLI

Trident is a command line software tool structured in multiple subcommands. If you installed it properly you can call it on the command line by typing `trident`. This will show an overview of the general options and all subcommands, which are explained in detail below.

```
Usage: trident [--version] [--logMode ARG] [--errLength ARG]
        [--inPlinkPopName ARG] (COMMAND | COMMAND)
trident is a management and analysis tool for Poseidon packages. Report issues
here: https://github.com/poseidon-framework/poseidon-hs/issues
```

Available options:

<code>-h, --help</code>	Show this help text
<code>--version</code>	Show version number
<code>--logMode ARG</code>	How information should be reported: NoLog, SimpleLog, DefaultLog, ServerLog or VerboseLog (default: DefaultLog)
<code>--errLength ARG</code>	After how many characters should a potential error message be truncated. "Inf" for no truncation. (default: CharCount 1500)
<code>--inPlinkPopName ARG</code>	Where to read the population/group name from the FAM file in Plink-format. Three options are possible: asFamily (default) asPhenotype asBoth.

Package creation and manipulation commands:

<code>init</code>	Create a new Poseidon package from genotype data
<code>fetch</code>	Download data from a remote Poseidon repository
<code>forge</code>	Select packages, groups or individuals and create a

```

43         new Poseidon package from them
44     genoconvert      Convert the genotype data in a Poseidon package to a
45                       different file format
46     update           Update POSEIDON.yml files automatically
47
48 Inspection commands:
49     list              List packages, groups or individuals from local or
50                       remote Poseidon repositories
51     summarise         Get an overview over the content of one or multiple
52                       Poseidon packages
53     summarize         Synonym for summarise
54     survey            Survey the degree of context information completeness
55                       for Poseidon packages
56     validate          Check one or multiple Poseidon packages for
57                       structural correctness

```

Trident allows to work directly with genotype data (see `-p` below), but its optimized for the interaction with [Poseidon packages](#), which wrap and contextualize the data. Most trident subcommands therefore have a central parameter, called `--baseDir` or simply `-d` to specify one or more base directories to look for packages. For example, if all Poseidon packages live inside a repository at `/path/to/poseidon/packages` you would simply say `trident <subcommand> -d /path/to/poseidon/dirs/` and `trident` would automatically search all subdirectories inside of the repository for valid Poseidon packages (as identified by valid `POSEIDON.yml` files).

You can arrange a poseidon repository in a hierarchical way. For example:

```

65 /path/to/poseidon/packages
66     /modern
67         /2019_poseidon_package1
68         /2019_poseidon_package2
69     /ancient
70         /...
71         /...
72     /Reference_Genomes
73         /...
74         /...

```

You can use this structure to select only the level of packages you're interested in, even individual ones, and you can make use of the fact that `-d` can be given multiple times.

Being able to specify one or multiple repositories is often not enough, as you may have your own data to co-analyse with the main repository. This is easy to do, as you simply need to provide your own genotype data as yet another Poseidon package to be added to your `trident` command. For example, let's say you have genotype data in `EIGENSTRAT` format (`trident` supports `EIGENSTRAT` and `PLINK` as formats.):

```

81 ~/my_project/my_project.geno
82 ~/my_project/my_project.snp
83 ~/my_project/my_project.ind

```

then you can make that to a skeleton Poseidon package with the `init` command. You can also do it manually by simply adding a `POSEIDON.yml` file, with for example the following content:

```

86 poseidonVersion: 2.5.0
87 title: My_awesome_project
88 description: Unpublished genetic data from my awesome project
89 contributor:
90   - name: Stephan Schiffels
91     email: schiffels@institute.org
92 packageVersion: 0.1.0
93 lastModified: 2020-10-07
94 genotypeData:
95   format: EIGENSTRAT
96   genoFile: my_project.geno
97   snpFile: my_project.snp
98   indFile: my_project.ind
99   jannoFile: my_project.janno
100  bibFile: sources.bib

```

Two remarks: 1) all file paths are considered *relative* to the directory in which POSEIDON.yml resides. Here we assume that you put this file into the same directory as the three genotype files. 2) Besides the genotype data files there are two (technically optional) files referenced by this example POSEIDON.yml file: **sources.bib** and **my_project.janno**. Of course you can add them manually - **init** automatically creates empty dummy versions.

Once you have set up your own “Poseidon” package (which is really only a skeleton so far), you can add it to your **trident** analysis, by simply adding your project directory to the command using **-d**, for example:

```

107 trident list -d /path/to/poseidon/packages/modern \
108   -d /path/to/poseidon/packages/ReferenceGenomes
109   -d ~/my_project --packages

```

1.1.1.1 General notes

1.1.1.1.1 Logging and command line output For all subcommands the general argument **--logMode** defines how trident reports messages (to stderr) on the command line:

- *NoLog*: Hides all messages.
- *SimpleLog*: Plain and simple output to stderr.
- *DefaultLog*: Adds severity indicators before each message. (default setting)
- *ServerLog*: Additionally adds timestamps before each message.
- *VerboseLog*: Shows not just messages on the log levels **Info**, **Warning** and **Error** like the other modes, but also on the more verbose level **Debug**. Use this for debugging.

1.1.1.2 Duplicates

- If multiple packages in a package repository share the same **title**, then trident will try to select the one with the highest version number. If this is not sufficient to resolve the conflict, trident will stop. An exception for that is the **list** subcommand, which will read and report all packages/groups/individuals in all versions.
- Individual/sample names (**Poseidon_IDs**) within one package have to be unique, or trident will stop.
- We generally also discourage ID duplicates across packages in package repositories, but trident will generally continue with them after printing a warning. This does not apply for **validate**, by default (you can

change this behaviour with `--ignoreDuplicates`), and `forge`. `forge` offers a special mechanism to resolve duplicates within its selection language (see below).

1.1.1.3 Group names in .fam files The `.fam` file of Plink-formatted genotype data is used inconsistently across different popular aDNA software tools to store group/population name information. The (global) option `--inPlinkPopName` with the arguments `asFamily` (default), `asPhenotype` and `asBoth` allows to control the reading of the population name from Plink `.fam` files. The subcommands that write genotype data (`forge`, `genoconvert`) have a corresponding option `--outPlinkPopName` to specify this for the output.

1.1.1.4 Whitespaces in the .janno file While reading the `.janno` file `trident` trims all leading and trailing whitespaces around individual cells. Also all instances of the `No-Break Space` unicode character will be removed. This means these whitespaces will not be preserved when a package is `forged`.

1.2 Package creation and manipulation commands

1.2.1 Init command

`init` creates a new, valid Poseidon package from genotype data files. It adds a valid `POSEIDON.yml` file, a dummy `.janno` file for context information and an empty `.bib` file for literature references.

[Click here for command line details](#)

```
Usage: trident init ((-p|--genoOne ARG) | --inFormat ARG --genoFile ARG
                  --snpFile ARG --indFile ARG) [--snpSet ARG]
                  (-o|--outPackagePath ARG) [-n|--outPackageName ARG]
                  [--minimal]
```

Create a new Poseidon package from genotype data

Available options:

<code>-h,--help</code>	Show this help text
<code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects <code>.bed</code> or <code>.bim</code> or <code>.fam</code> for PLINK and <code>.geno</code> or <code>.snp</code> or <code>.ind</code> for EIGENSTRAT. The other files must be in the same directory and must have the same base name
<code>--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or PLINK (only necessary for data input with <code>--genoFile</code> + <code>--snpFile</code> + <code>--indFile</code>)
<code>--genoFile ARG</code>	the input geno file path
<code>--snpFile ARG</code>	the input snp file path
<code>--indFile ARG</code>	the input ind file path
<code>--snpSet ARG</code>	the snpSet of the package: 1240K, HumanOrigins or Other. (only relevant for data input with <code>-p --genoOne</code> or <code>--genoFile</code> + <code>--snpFile</code> + <code>--indFile</code> , because the packages in a <code>-d --baseDir</code> already have this information in their respective <code>POSEIDON.yml</code> files) Default: Other
<code>-o,--outPackagePath ARG</code>	the output package directory path
<code>-n,--outPackageName ARG</code>	the output package name - this is optional: If no

168 name is provided, then the package name defaults to
 169 the basename of the (mandatory) `--outPackagePath`
 170 argument
 171 `--minimal` should only a minimal output package be created?

172 The command

```
173 trident init \  

  174 --inFormat EIGENSTRAT/PLINK \  

  175 --genoFile path/to/geno_file \  

  176 --snpFile path/to/snp_file \  

  177 --indFile path/to/ind_file \  

  178 --snpSet 1240K|HumanOrigins|Other \  

  179 -o path/to/new_package_name
```

180 requires the format (`--inFormat`) of your input data (either `EIGENSTRAT` or `PLINK`), the paths to the respective
 181 files (`--genoFile`, `--snpFile`, `--indFile`), and optionally the “shape” of these files (`--snpSet`), so if they cover
 182 the 1240K, the `HumanOrigins` or an `Other` SNP set. A simpler interface added in trident 0.29.0 is available with
 183 `-p (+ --snpSet)`.

	EIGENSTRAT	PLINK
genoFile	.geno	.bed
snpFile	.snp	.bim
indFile	.ind	.fam

184 The output package of `init` is created as a new directory `-o`, which should not already exist, and gets the
 185 package `title` corresponding to the basename of `-o`. You can also set the title explicitly with `-n`. The `--minimal`
 186 flag causes `init` to create a minimal package with a very basic `POSEIDON.yml` and no `.bib` and `.janno` files.

187 1.2.2 Fetch command

188 `fetch` allows to download Poseidon packages from a remote Poseidon server. Read more about this repository
 189 [here](#).

190 Click here for command line details

```
191 Usage: trident fetch (-d|--baseDir DIR)  

  192             (--downloadAll |  

  193             (--fetchFile ARG | (-f|--fetchString ARG)))  

  194             [--remoteURL ARG]
```

195
 196 Download data from a remote Poseidon repository

197
 198 Available options:

```
199 -h,--help          Show this help text  

  200 -d,--baseDir DIR   a base directory to search for Poseidon Packages  

  201                   (could be a Poseidon repository)  

  202 --downloadAll       download all packages the server is offering
```

```

203  --fetchFile ARG          A file with a list of packages. Works just as -f, but
204                           multiple values can also be separated by newline, not
205                           just by comma. -f and --fetchFile can be combined.
206  -f,--fetchString ARG     List of packages to be downloaded from the remote
207                           server. Package names should be wrapped in asterisks:
208                           *package_title*. You can combine multiple values with
209                           comma, so for example: "*package_1*, *package_2*,
210                           *package_3*". fetchString uses the same parser as
211                           forgeString, but does not allow excludes. If groups
212                           or individuals are specified, then packages which
213                           include these groups or individuals are included in
214                           the download.
215  --remoteURL ARG          URL of the remote Poseidon server
216                           (default: "https://server.poseidon-adna.org")

```

217 It works with

```

218 trident fetch -d ... -d ... \
219   -f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<individual1>"

```

220 and the entities you want to download must be listed either in a simple string of comma-separated values, which
221 can be passed via -f/--fetchString, or in a text file (--fetchFile). Entities are then combined from these
222 sources.

223 Entities are specified using a special syntax (see also the documentation of **forge** below): Package titles are
224 wrapped in asterisks: ***package_title***, group names are spelled as is, and individual names are wrapped in
225 angular brackets, so **<individual1>**. Fetch will figure out which packages need to be downloaded to include all
226 specified entities. --downloadAll, which can be given instead of -f and --fetchFile, causes fetch to download
227 all packages from the server. The downloaded packages are added in the first (!) -d directory (which gets created
228 if it doesn't exist), but downloads are only performed if the respective packages are not already present in the
229 latest version in any of the -d dirs.

230 Note that **trident fetch** makes most sense in combination with **trident list --remote**: First one can inspect
231 what is available on the server, then one can create a custom fetch command.

232 **fetch** also has the optional arguments --remote https://..." to name an alternative poseidon server. The
233 default points to the **DAG server**.

234 1.2.3 Forge command

235 **forge** creates new Poseidon packages by extracting and merging packages, populations and individuals from
236 your Poseidon repositories.

237 Click here for command line details

```

238 Usage: trident forge ((-d|--baseDir DIR) |
239                      ((-p|--genoOne ARG) | --inFormat ARG --genoFile ARG
240                      --snpFile ARG --indFile ARG) [--snpSet ARG])
241                      [--forgeFile ARG | (-f|--forgeString ARG)]
242                      [--selectSnps ARG] [--intersect] [--outFormat ARG]
243                      [--minimal] [--onlyGeno] (-o|--outPackagePath ARG)

```

```

244         [-n|--outPackageName ARG] [--packagewise]
245         [--outPlinkPopName ARG]
246     Select packages, groups or individuals and create a new Poseidon package from
247     them
248
249     Available options:
250     -h,--help                Show this help text
251     -d,--baseDir DIR          a base directory to search for Poseidon Packages
252                               (could be a Poseidon repository)
253     -p,--genoOne ARG          one of the input genotype data files. Expects .bed or
254                               .bim or .fam for PLINK and .geno or .snp or .ind for
255                               EIGENSTRAT. The other files must be in the same
256                               directory and must have the same base name
257     --inFormat ARG            the format of the input genotype data: EIGENSTRAT or
258                               PLINK (only necessary for data input with --genoFile
259                               + --snpFile + --indFile)
260     --genoFile ARG            the input geno file path
261     --snpFile ARG             the input snp file path
262     --indFile ARG             the input ind file path
263     --snpSet ARG              the snpSet of the package: 1240K, HumanOrigins or
264                               Other. (only relevant for data input with
265                               -p|--genoOne or --genoFile + --snpFile + --indFile,
266                               because the packages in a -d|--baseDir already have
267                               this information in their respective POSEIDON.yml
268                               files) Default: Other
269     --forgeFile ARG           A file with a list of packages, groups or individual
270                               samples. Works just as -f, but multiple values can
271                               also be separated by newline, not just by comma.
272                               Empty lines are ignored and comments start with "#",
273                               so everything after "#" is ignored in one line.
274                               Multiple instances of -f and --forgeFile can be
275                               given. They will be evaluated according to their
276                               input order on the command line.
277     -f,--forgeString ARG      List of packages, groups or individual samples to be
278                               combined in the output package. Packages follow the
279                               syntax *package_title*, populations/groups are simply
280                               group_id and individuals <individual_id>. You can
281                               combine multiple values with comma, so for example:
282                               "*package_1*, <individual_1>, <individual_2>,
283                               group_1". Duplicates are treated as one entry.
284                               Negative selection is possible by prepending "-" to
285                               the entity you want to exclude (e.g. "*package_1*,
286                               -<individual_1>, -group_1"). forge will apply
287                               excludes and includes in order. If the first entity
288                               is negative, then forge will assume you want to merge

```

289 all individuals in the packages found in the baseDirs
 290 (except the ones explicitly excluded) before the
 291 exclude entities are applied. An empty forgeString
 292 (and no --forgeFile) will therefore merge all
 293 available individuals. If there are individuals in
 294 your input packages with equal individual id, but
 295 different main group or source package, they can be
 296 specified with the special syntax
 297 "<package:group:individual>".

298 --selectSnps ARG To extract specific SNPs during this forge operation,
 299 provide a Snp file. Can be either Eigenstrat (file
 300 ending must be '.snp') or Plink (file ending must be
 301 '.bim'). When this option is set, the output package
 302 will have exactly the SNPs listed in this file. Any
 303 SNP not listed in the file will be excluded. If
 304 option '--intersect' is also set, only the SNPs
 305 overlapping between the SNP file and the forged
 306 packages are output.

307 --intersect Whether to output the intersection of the genotype
 308 files to be forged. The default (if this option is
 309 not set) is to output the union of all SNPs, with
 310 genotypes defined as missing in those packages which
 311 do not have a SNP that is present in another package.
 312 With this option set, the forged dataset will
 313 typically have fewer SNPs, but less missingness.

314 --outFormat ARG the format of the output genotype data: EIGENSTRAT or
 315 PLINK. Default: PLINK

316 --minimal should only a minimal output package be created?

317 --onlyGeno should only the resulting genotype data be returned?
 318 This means the output will not be a Poseidon package

319 -o,--outPackagePath ARG the output package directory path

320 -n,--outPackageName ARG the output package name - this is optional: If no
 321 name is provided, then the package name defaults to
 322 the basename of the (mandatory) --outPackagePath
 323 argument

324 --packagewise Skip the within-package selection step in forge. This
 325 will result in outputting all individuals in the
 326 relevant packages, and hence a superset of the
 327 requested individuals/groups. It may result in better
 328 performance in cases where one wants to forge entire
 329 packages or almost entire packages. Details: Forge
 330 conceptually performs two types of selection: First,
 331 it identifies which packages in the supplied base
 332 directories are relevant to the requested forge, i.e.
 333 whether they are either explicitly listed using


```

334         *PackageName*, or because they contain selected
335         individuals or groups. Second, within each relevant
336         package, individuals which are not requested are
337         removed. This option skips only the second step, but
338         still performs the first.
339     --outPlinkPopName ARG    Where to write the population/group name into the FAM
340                             file in Plink-format. Three options are possible:
341                             asFamily (default) | asPhenotype | asBoth. See also
342                             --inPlinkPopName.
343
344     forge can be used with
345
346     trident forge -d ... -d ... \
347         -f "*package_name*, group_id, <individual_id>" \
348         -o path/to/new_package_name
349
350     where the entities (packages, groups/populations, individuals/samples) you want in the output package can be
351     denoted either as a string on the command line (-f/--forgeString), or in an input text file (--forgeFile).
352     See the section below for the syntax of this selection language. Do not forget to wrap the --forgeString query
353     in quotes.
354
355     Including one or multiple Poseidon packages with -d is not the only way to include data for a forge operation.
356     It is also possible to consider unpackaged genotype data directly with -p (+ --snpSet) or --inFormat +
357     --genoFile + --snpFile + --indFile (+ --snpSet). This makes the following example possible, where we
358     merge data from one Poseidon package and two genotype datasets to get a new EIGENSTRAT dataset.
359
360     trident forge \
361         -d 2017_GonzalesFortesCurrentBiology \
362         -p 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \
363         --inFormat PLINK \
364         --genoFile 2017_HaberAJHG/2017_HaberAJHG.bed \
365         --snpFile 2017_HaberAJHG/2017_HaberAJHG.bim \
366         --indFile 2017_HaberAJHG/2017_HaberAJHG.fam \
367         -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \
368         -o testpackage \
369         --outFormat EIGENSTRAT \
370         --onlyGeno

```

1.2.3.1 The forge selection language The text in --forgeString and --forgeFile are parsed as a domain specific query language that describes precisely which entities should be compiled in the output package of a given forge operation. The language has multiple syntactic elements and a specific evaluation logic.

In general a --forgeString query consists of multiple entities, separated by ,. The main entities are Poseidon packages, groups/populations and individuals/samples:

- Each package title is surrounded by *: *package*. That means if you want all individuals of the Poseidon package 2019_Jeong_InnerEurasia in the output package you would add *2019_Jeong_InnerEurasia* to the query.
- Groups/populations are not specially marked: group. So to get all individuals of the group Swiss_Roman_period, you would simply add Swiss_Roman_period.

376 • Individuals/samples are surrounded by < and >: <individual>. ALA026 therefore becomes <ALA026>. A sec-
 377 ond way to denote individuals is with the more verbose and specific syntax <package:group:individual>.
 378 Such defined individuals take precedence over differently defined ones (so: directly with <individual> or
 379 as a subset of *package* or group). This allows to resolve duplication issues precisely – at least in cases
 380 where the duplicated individuals differ in source package or primary group.

381 In the --forgeFile each line is treated as a separate forgeString, empty lines are ignored and #s start comments.
 382 So this is a valid forgeFile:

```
383 # Packages
384 *package1*, *package2*
385
386 # Groups and individuals from other packages beyond package1 and package2
387 group1, <individual1>, group2, <individual2>, <individual3>
388
389 # group2 has two outlier individuals that should be ignored
390 -<bad_individual1> # This one has very low coverage
391 -<bad_individual2> # This one is from a different time period
```

392 By prepending - to the bad individuals, we can exclude them from the forged package. forge fig-
 393 ures out the final list of samples to include by executing all forge-entities in order. So an entity list
 394 *PackageA*, -<Individual1>, GroupA may result in a different outcome than *PackageA*, GroupA, -<Individual1>,
 395 depending on whether <Individual1> belongs to GroupA or not. If the forge entity list starts with a negative
 396 entity, or if the entity list is empty, forge will implicitly assume you want to include all individuals in all
 397 packages found in the baseDirs (except the ones explicitly excluded, of course).

398 An empty forgeString will therefore merge all available individuals.

399 **1.2.3.2 Treatment of the .janno file while merging** forge merges and subsets .janno files along with
 400 the genotype data. If a package lacks a .janno file, then a basic one will be created internally based on the
 401 information in the genotype data, and used for the output. Missing columns across packages will be filled with
 402 n/a.

403 For merging two .janno files **A** and **B** the following rules apply regarding undefined, arbitrary additional columns:

- 404 • If **A** has an additional column which is not in **B** then empty cells in the rows imported from **B** are filled
 405 with n/a.
- 406 • If **A** and **B** share additional columns with identical column name, then they are treated as semantically
 407 identical units and merged accordingly.
- 408 • In the resulting .janno file, all additional columns from both **A** and **B** are sorted alphabetically and
 409 appended after the normal, specified variables.

410 The following example illustrates the described behaviour:

411 **A.janno**

Poseidon_ID	Group_Name	Genetic_Sex	AdditionalColumn1	AdditionalColumn2
XXX011	POP1	M	A	D
XXX012	POP2	F	B	E
XXX013	POP1	M	C	F

412 B.janno

Poseidon_ID	Group_Name	Genetic_Sex	AdditionalColumn3	AdditionalColumn2
YYY022	POP5	F	G	J
YYY023	POP5	F	H	K
YYY024	POP5	M	I	L

413 A.janno + B.janno

Poseidon_ID	Group_Name	Genetic_Sex	AdditionalColumn1	AdditionalColumn2	AdditionalColumn3
XXX011	POP1	M	A	D	n/a
XXX012	POP2	F	B	E	n/a
XXX013	POP1	M	C	F	n/a
YYY022	POP5	F	n/a	J	G
YYY023	POP5	F	n/a	K	H
YYY024	POP5	M	n/a	L	I

414 **1.2.3.3 Treatment of the .ssf file while merging** The Sequencing Source File (short .ssf file) is forged in
 415 exactly the same way as the janno file. SSF files that are present are included in the forge product in the way
 416 that the user expects, following selection of those entities which are listed in the `poseidon_IDS` columns of the
 417 SSF files. Columns that are only present in some packages, including those not defined by our [Schema] are also
 418 included in the forged product in the same way as described for Janno above.

419 **1.2.3.4 Other options** Just as for `init` the output package of `forge` is created as a new directory `-o`. The
 420 title can also be explicitly defined with `-n`.

421 `--minimal` allows for the creation of a minimal output package without `.bib` and `.janno`. This is especially
 422 useful for data analysis pipelines, where only the genotype data is required. Even more basic output comes with
 423 `--onlyGeno`, which means that only the genotype data is returned without any Poseidon package.

424 `forge` has a an optional flag `--intersect`, that defines, if the genotype data from different packages should
 425 be merged with an **union** or an **intersect** operation. The default (if this option is not set) is to output the
 426 union of all SNPs, with genotypes defined as missing in samples from packages which do not have a SNP that is
 427 present in another package. With this option set, on the other hand, the forged dataset will typically have fewer
 428 SNPs, but less missingness.

429 `--intersect` also influences the automatic determination of the `snpSet` field in the POSEIDON.yml file for the
 430 resulting package. If the `snpSets` of all input packages are identical, then the resulting package will just inherit
 431 this configuration. Otherwise `forge` applies the following pairwise merging logic:

Input snpSet A	Input snpSet B	<code>--intersect</code>	Ouput snpSet
Other	*	*	Other
1240K	HumanOrigins	True	HumanOrigins
1240K	HumanOrigins	False	1240K

432 `--selectSnps` allows to provide `forge` with a SNP file in EIGENSTRAT (.snp) or PLINK (.bim) format to
 433 create a package with a specific selection. When this option is set, the output package will have exactly the
 434 SNPs listed in this file. Any SNP not listed in the file will be excluded. If `--intersect` is also set, only the
 435 SNPs overlapping between the SNP file and the forged packages are output.

436 Merging genotype data across different data sources and file formats is tricky. `forge` is more verbose about
 437 potential issues, if the `--logMode` flag is set to `VerboseLog`.

438 The `--onlyGeno` command specifies that only genotype data should be output, not an entire Poseidon package.

439 With `--packagewise` the within-package selection step in `forge` can be skipped. This will result in outputting
 440 all individuals in the relevant packages, and hence a superset of the requested individuals/groups. It may result
 441 in better performance in cases where one wants to forge entire packages.

442 1.2.4 Genoconvert command

443 `genoconvert` converts the genotype data in a Poseidon package to a different file format. The respective entries
 444 in the POSEIDON.yml file are changed accordingly.

445 [Click here for command line details](#)

```
446 Usage: trident genoconvert ((-d|--baseDir DIR) |
447                             ((-p|--genoOne ARG) | --inFormat ARG --genoFile ARG
448                             --snpFile ARG --indFile ARG) [--snpSet ARG])
449                             --outFormat ARG [--onlyGeno]
450                             [-o|--outPackagePath ARG] [--removeOld]
451                             [--outPlinkPopName ARG]
```

452 Convert the genotype data in a Poseidon package to a different file format

454 Available options:

455	<code>-h,--help</code>	Show this help text
456	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
457		
458	<code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects .bed or 459 .bim or .fam for PLINK and .geno or .snp or .ind for 460 EIGENSTRAT. The other files must be in the same 461 directory and must have the same base name
462	<code>--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or 463 PLINK (only necessary for data input with <code>--genoFile</code> 464 + <code>--snpFile</code> + <code>--indFile</code>)
465	<code>--genoFile ARG</code>	the input geno file path
466	<code>--snpFile ARG</code>	the input snp file path
467	<code>--indFile ARG</code>	the input ind file path
468	<code>--snpSet ARG</code>	the snpSet of the package: 1240K, HumanOrigins or 469 Other. (only relevant for data input with 470 <code>-p --genoOne</code> or <code>--genoFile</code> + <code>--snpFile</code> + <code>--indFile</code> , 471 because the packages in a <code>-d --baseDir</code> already have 472 this information in their respective POSEIDON.yml 473 files) Default: Other

```

474 --outFormat ARG          the format of the output genotype data: EIGENSTRAT or
475                          PLINK.
476 --onlyGeno              should only the resulting genotype data be returned?
477                          This means the output will not be a Poseidon package
478 -o,--outPackagePath ARG  the output package directory path - this is optional:
479                          If no path is provided, then the output is written to
480                          the directories where the input genotype data file
481                          (.bed/.geno) is stored
482 --removeOld              Remove the old genotype files when creating the new
483                          ones
484 --outPlinkPopName ARG    Where to write the population/group name into the FAM
485                          file in Plink-format. Three options are possible:
486                          asFamily (default) | asPhenotype | asBoth. See also
487                          --inPlinkPopName.

```

488 With the default setting

```

489 trident genoconvert -d ... -d ... --outFormat EIGENSTRAT|PLINK

```

490 all packages in -d will be converted to the desired --outFormat (either EIGENSTRAT or PLINK), if the data is
491 not already in this format. This includes updating the respective POSEIDON.yml files.

492 The “old” data is not deleted, but kept around. That means conversion can result in a package with both PLINK
493 and EIGENSTRAT data, but only one is linked in the POSEIDON.yml file, and that is what will be used by
494 trident. To delete the old data in the conversion you can add the --removeOld flag.

495 Instead of -d to change Poseidon packages, the -p (+ --snpSet) or --inFormat + --genoFile + --snpFile
496 + --indFile (+ --snpSet) allow to directly convert genotype data that is not wrapped in a Poseidon package
497 and store it to a directory given in -o. See this example:

```

498 trident genoconvert \
499   -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
500   --outFormat EIGENSTRAT
501   -o my_directory

```

502 1.2.5 Update command

503 **update** automatically harmonizes POSEIDON.yml files of one or multiple packages if the packages were changed.
504 This is not an automatic update from one Poseidon version to the next!

505 [Click here for command line details](#)

```

506 Usage: trident update (-d|--baseDir DIR) [--poseidonVersion ARG]
507                          [--ignorePoseidonVersion] [--versionComponent ARG]
508                          [--noChecksumUpdate] [--newContributors ARG]
509                          [--logText ARG] [--force]
510 Update POSEIDON.yml files automatically

```

512 Available options:

```

513 -h,--help          Show this help text
514 -d,--baseDir DIR   a base directory to search for Poseidon Packages

```

```

515         (could be a Poseidon repository)
516 --poseidonVersion ARG    Poseidon version the packages should be updated to:
517                          e.g. "2.5.3" (default: Nothing)
518 --ignorePoseidonVersion  Read packages even if their poseidonVersion is not
519                          compatible with the trident version. The assumption
520                          is, that the package is already structurally adjusted
521                          to the trident version and only the version number is
522                          lagging behind.
523 --versionComponent ARG    Part of the package version number in the
524                          POSEIDON.yml file that should be updated: Major,
525                          Minor or Patch (see https://semver.org)
526                          (default: Patch)
527 --noChecksumUpdate        Should update of checksums in the POSEIDON.yml file
528                          be skipped
529 --ignoreGeno              ignore SNP and GenoFile
530 --newContributors ARG     Contributors to add to the POSEIDON.yml file in the
531                          form "[Firstname Lastname](Email address);..."
532 --logText ARG             Log text for this version jump in the CHANGELOG file
533                          (default: "not specified")
534 --force                   Normally the POSEIDON.yml files are only changed if
535                          the poseidonVersion is adjusted or any of the
536                          checksums change. With --force a package version
537                          update can be triggered even if this is not the case.

```

538 It can be called with a lot of optional arguments

```

539 trident update -d ... -d ... \
540   --poseidonVersion "X.X.X" \
541   --versionComponent Major/Minor/Patch \
542   --noChecksumUpdate
543   --ignoreGeno
544   --newContributors "[Firstname Lastname](Email address);..."
545   --logText "short description of the update"
546   --force

```

547 By default `update` will not edit a package's POSEIDON.yml file, even when arguments like `--versionComponent`,
548 `--newContributors` or `--logText` are explicitly set. This default exists to run the function on a large set of
549 packages where only few of them were edited and need an active update. A package will only be modified by
550 `update` if either

- 551 • any of the files with checksums (e.g. the genotype data) in it were modified,
- 552 • the `--poseidonVersion` argument differs from the `poseidonVersion` in the package's POSEIDON.yml
553 file
- 554 • or the `--force` flag was set in `update`.

555 If any of these applies to a package in the search directory (`--baseDir/-d`), it will be updated. This includes
556 the following steps:

- 557 • If `--poseidonVersion` is different from the `poseidonVersion` field in the package, then that will be

558 updated.

- 559 • The `packageVersion` will be incremented. If `--versionComponent` is not set, then it falls back to `Patch`,
560 so a change in the last position of the three digit version number. Minor increments the middle, and Major
561 the first position (see [semantic versioning](#)).
- 562 • The `lastModified` field will be updated to the current day (based on your computer's system time).
- 563 • The contributors in `--newContributors` will be added to the `contributor` field if they're not there already.
- 564 • If any checksums changed, then they will be updated. If certain checksums are not set yet, then they will
565 be added. The checksum update can be skipped with `--noChecksumUpdate` or partially skipped for the
566 genotype data with `--ignoreGeno`.
- 567 • The `CHANGELOG.md` file will be updated with a new row for the new version and the text in `--logText`
568 (default: "not specified"), which will be appended as the first line of the file. If no `CHANGELOG.md` file
569 exists, then it will be created and referenced in the `POSEIDON.yml` file.

570 :heavy_exclamation_mark: As `update` reads and rewrites `POSEIDON.yml` files, it may change their inner order,
571 layout or even content (e.g. if they have fields which are not in the [Poseidon package definition](#)). Create a backup
572 of the `POSEIDON.yml` file before running `update` if you are uncertain.

573 1.3 Inspection commands

574 1.3.1 List command

575 `list` lists packages, groups and individuals of the datasets you use, or of the packages available on the server.

576 [Click here for command line details](#)

```
577 Usage: trident list ((-d|--baseDir DIR) | --remote [--remoteURL ARG])
578                 (--packages | --groups | --individuals
579                 [-j|--jannoColumn JANNO_HEADER]) [--raw]
```

581 List packages, groups or individuals from local or remote Poseidon
582 repositories

583 Available options:

585 <code>-h,--help</code>	Show this help text
586 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
587 <code>--remote</code>	list packages from a remote server instead the local file system
588 <code>--remoteURL ARG</code>	URL of the remote Poseidon server (default: "https://server.poseidon-adna.org")
589 <code>--packages</code>	list all packages
590 <code>--groups</code>	list all groups, ignoring any group names after the first as specified in the Janno-file
591 <code>--individuals</code>	list individuals
592 <code>-j,--jannoColumn JANNO_HEADER</code>	list additional fields from the janno files, using the Janno column heading name, such as Country, Site, Date_C14_Uncal_BP, Endogenous, ...

```

600     --raw                output table as tsv without header. Useful for piping
601                          into grep or awk

```

To list packages from your local repositories, as seen above you can run

```

603 trident list -d ... -d ... --packages

```

This will yield a nicely formatted table of all packages, their last update and the number of individuals in it.

To view packages on the remote server, instead of using directories to specify the locations of repositories on your system, you can use `--remote` to show packages on the remote server. For example

```

607 trident list --packages --remote

```

will result in a view of all published packages in our [public online repository](#).

You can also list groups, as defined in the third column of EIGENSTRAT `.ind` files (or the first/last column of a PLINK `.fam` file), and individuals with `--groups` and `--individuals` instead of `--packages`.

The `--individuals` flag provides a way to immediately access information from the `.janno` files on the command line. This works with the `-j/--jannoColumn` option. For example adding `-j Country -j Date_C14_Uncal_BP` to the commands above will add the `Country` and the `Date_C14_Uncal_BP` columns to the respective output tables.

Note that if you want a less fancy table, for example because you want to load this into Excel, or pipe into another command that cannot deal with the neat table layout, you can use the `--raw` option to output that table as a simple tab-delimited stream.

618 1.3.2 Summarise command

619 `summarise` prints some general summary statistics for a given poseidon dataset taken from the `.janno` files.

620 [Click here for command line details](#)

```

621 Usage: trident summarise (-d|--baseDir DIR) [--raw]
622     Get an overview over the content of one or multiple Poseidon packages
623
624 Available options:
625     -h,--help                Show this help text
626     -d,--baseDir DIR         a base directory to search for Poseidon Packages
627                             (could be a Poseidon repository)
628     --raw                    output table as tsv without header. Useful for piping
629                             into grep or awk

```

630 You can run it with

```

631 trident summarise -d ... -d ...

```

632 which will show you context information like – among others – the number of individuals in the dataset, their sex distribution, the mean age of the samples (for ancient data) or the mean coverage on the 1240K SNP array in a table. `summarise` depends on complete `.janno` files and will silently ignore missing information for some statistics.

636 You can use the `--raw` option to output the summary table in a simple, tab-delimited layout.

637 1.3.3 Survey command

638 **survey** tries to indicate package completeness (mostly focused on .janno files) for poseidon datasets.

639 [Click here for command line details](#)

640 Usage: trident survey (-d|--baseDir DIR) [--raw]

641 Survey the degree of context information completeness for Poseidon packages

642 Available options:

644 -h,--help	Show this help text
645 -d,--baseDir DIR	a base directory to search for Poseidon Packages (could be a Poseidon repository)
647 --raw	output table as tsv without header. Useful for piping 648 into grep or awk

649 Running

650 trident survey -d ... -d ...

651 will yield a table with one row for each package. See trident survey -h for a legend which cell of this table
652 means what.

653 Again you can use the --raw option to output the survey table in a tab-delimited format.

654 1.3.4 Validate command

655 **validate** checks poseidon datasets for structural correctness.

656 [Click here for command line details](#)

657 Usage: trident validate (-d|--baseDir DIR)

658 Check one or multiple Poseidon packages for structural correctness

659 Available options:

661 -h,--help	Show this help text
662 -d,--baseDir DIR	a base directory to search for Poseidon Packages 663 (could be a Poseidon repository)
664 --ignoreGeno	ignore SNP and GenoFile
665 --fullGeno	test parsing of all SNPs (by default only the first 666 100 SNPs are probed)
667 --noExitCode	do not produce an explicit exit code
668 --ignoreDuplicates	do not stop on duplicated individual names in the 669 package collection

670 You can run it with

671 trident validate -d ... -d ...

672 and it will either report a success (Validation passed) or failure with specific error messages to simplify fixing
673 the issues.

674 **validate** tries to ensure that each package in the dataset adheres to the [schema definition](#). Here is a list of
675 what is checked:

676 • Presence of the necessary files
677 • Full structural correctness of .bib and .janno file
678 • Superficial correctness of genotype data files by parsing the first 100 SNPs. A full check that parses all
679 SNPs can be run with the `--fullGeno` option
680 • Correspondence of BibTeX keys in .bib and .janno
681 • Correspondence of individual and group IDs in .janno and genotype data files

682 In fact much of this validation already runs as part of the general package reading pipeline invoked for many
683 trident subcommands (e.g. `forge`). `validate` is meant to be more thorough, though, and will explicitly fail if
684 even a single package is broken.

685 Remember to run it with `--logMode VerboseLog` to get more information if the output is not sufficient to debug
686 an issue.