

Contents

1	Guide for trident v1.1.10.2	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 v1.1.10.2

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.
- 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).

127 **1.1.1.3 Group names in .fam files** The .fam file of Plink-formatted genotype data is used inconsistently
 128 across different popular aDNA software tools to store group/population name information. The (global) option
 129 `--inPlinkPopName` with the arguments `asFamily` (default), `asPhenotype` and `asBoth` allows to control the
 130 reading of the population name from Plink .fam files. The subcommands that write genotype data (`forge`,
 131 `genoconvert`) have a corresponding option `--outPlinkPopName` to specify this for the output.

132 **1.1.1.4 Whitespaces in the .janno file** While reading the .janno file `trident` trims all leading and
 133 trailing whitespaces around individual cells. Also all instances of the `No-Break Space` unicode character will be
 134 removed. This means these whitespaces will not be preserved when a package is `forged`.

135 1.2 Package creation and manipulation commands

136 1.2.1 Init command

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

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

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

144 Create a new Poseidon package from genotype data

146 Available options:

147 <code>-h,--help</code>	Show this help text
148 <code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects .bed or
149	.bim or .fam for PLINK and .geno or .snp or .ind for
150	EIGENSTRAT. The other files must be in the same
151	directory and must have the same base name
152 <code>--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or
153	PLINK (only necessary for data input with <code>--genoFile</code>
154	+ <code>--snpFile</code> + <code>--indFile</code>)
155 <code>--genoFile ARG</code>	the input geno file path
156 <code>--snpFile ARG</code>	the input snp file path
157 <code>--indFile ARG</code>	the input ind file path
158 <code>--snpSet ARG</code>	the snpSet of the package: 1240K, HumanOrigins or
159	Other. (only relevant for data input with
160	<code>-p --genoOne</code> or <code>--genoFile</code> + <code>--snpFile</code> + <code>--indFile</code> ,
161	because the packages in a <code>-d --baseDir</code> already have
162	this information in their respective <code>POSEIDON.yml</code>
163	files) Default: Other
164 <code>-o,--outPackagePath ARG</code>	the output package directory path
165 <code>-n,--outPackageName ARG</code>	the output package name - this is optional: If no
166	name is provided, then the package name defaults to
167	the basename of the (mandatory) <code>--outPackagePath</code>
168	argument

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

170 The command

```
171 trident init \  
172   --inFormat EIGENSTRAT/PLINK \  
173   --genoFile path/to/geno_file \  
174   --snpFile path/to/snp_file \  
175   --indFile path/to/ind_file \  
176   --snpSet 1240K|HumanOrigins|Other \  
177   -o path/to/new_package_name
```

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

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

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

185 1.2.2 Fetch command

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

188 Click here for command line details

```
189 Usage: trident fetch (-d|--baseDir DIR)  
190           (--downloadAll |  
191           (--fetchFile ARG | (-f|--fetchString ARG)))  
192           [--remoteURL ARG] [-u|--upgrade]
```

193 Download data from a remote Poseidon repository

194
195 Available options:

196 -h,--help	Show this help text
197 -d,--baseDir DIR	a base directory to search for Poseidon Packages 198 (could be a Poseidon repository)
199 --downloadAll	download all packages the server is offering
200 --fetchFile ARG	A file with a list of packages. Works just as -f, but 201 multiple values can also be separated by newline, not 202 just by comma. -f and --fetchFile can be combined.
203 -f,--fetchString ARG	List of packages to be downloaded from the remote

204 server. Package names should be wrapped in asterisks:
 205 `*package_title*`. You can combine multiple values with
 206 comma, so for example: `"*package_1*, *package_2*,`
 207 `*package_3*"`. `fetchString` uses the same parser as
 208 `forgeString`, but does not allow excludes. If groups
 209 or individuals are specified, then packages which
 210 include these groups or individuals are included in
 211 the download.

212 `--remoteURL ARG` URL of the remote Poseidon server
 213 (default: "https://c107-224.cloud.gwdg.de")

214 `-u,--upgrade` overwrite outdated local package versions

215 It works with

216 `trident fetch -d ... -d ... \`
 217 `-f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<Individual1>"`

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

221 Entities are specified using a special syntax (see also the documentation of `forge` below): Package titles are
 222 wrapped in asterisks: `package_title`, group names are spelled as is, and individual names are wrapped in angular
 223 brackets, like `<Individual1>`. Fetch will figure out which packages need to be downloaded to include all specified
 224 entities. `--downloadAll`, which can be given instead of `-f` and `--fetchFile`, causes fetch to download all
 225 packages from the server. The downloaded packages are added in the first (!) `-d` directory (which gets created
 226 if it doesn't exist), but downloads are only performed if the respective packages are not already present in an
 227 up-to-date version in any of the `-d` dirs.

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

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

232 To overwrite outdated package versions with `fetch`, the `-u/--upgrade` flag has to be set. Note that many file
 233 systems do not offer a way to recover overwritten files. So be careful with this switch.

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

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

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

424 `--intersect` also influences the automatic determination of the `snpSet` field in the `POSEIDON.yml` file for the
 425 resulting package. If the `snpSets` of all input packages are identical, then the resulting package will just inherit
 426 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

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

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

433 The `--onlyGeno` command specifies that only genotype data should be output, not an entire Poseidon package.
 434 With `--packagewise` the within-package selection step in forge can be skipped. This will result in outputting
 435 all individuals in the relevant packages, and hence a superset of the requested individuals/groups. It may result
 436 in better performance in cases where one wants to forge entire packages.

437 1.2.4 Genoconvert command

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

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

```
441 Usage: trident genoconvert ((-d|--baseDir DIR) |
442                             ((-p|--genoOne ARG) | --inFormat ARG --genoFile ARG
443                             --snpFile ARG --indFile ARG) [--snpSet ARG])
444                             --outFormat ARG [--onlyGeno]
445                             [-o|--outPackagePath ARG] [--removeOld]
446                             [--outPlinkPopName ARG]
```

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

448
 449 Available options:

450	<code>-h,--help</code>	Show this help text
451	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
452		
453	<code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects .bed or .bim or .fam for PLINK and .geno or .snp or .ind for EIGENSTRAT. The other files must be in the same directory and must have the same base name
454		
455	<code>--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or PLINK (only necessary for data input with --genoFile + --snpFile + --indFile)
456		
457	<code>--genoFile ARG</code>	the input geno file path
458	<code>--snpFile ARG</code>	the input snp file path
459	<code>--indFile ARG</code>	the input ind file path
460	<code>--snpSet ARG</code>	the snpSet of the package: 1240K, HumanOrigins or Other. (only relevant for data input with -p --genoOne or --genoFile + --snpFile + --indFile, because the packages in a -d --baseDir already have this information in their respective POSEIDON.yml files) Default: Other
461		
462	<code>--outFormat ARG</code>	the format of the output genotype data: EIGENSTRAT or PLINK.
463		
464	<code>--onlyGeno</code>	should only the resulting genotype data be returned? This means the output will not be a Poseidon package
465		
466	<code>-o,--outPackagePath ARG</code>	the output package directory path - this is optional: If no path is provided, then the output is written to the directories where the input genotype data file
467		
468		
469		
470		
471		
472		
473		
474		
475		

476 (.bed/.geno) is stored

477 --removeOld Remove the old genotype files when creating the new

478 ones

479 --outPlinkPopName ARG Where to write the population/group name into the FAM

480 file in Plink-format. Three options are possible:

481 asFamily (default) | asPhenotype | asBoth. See also

482 --inPlinkPopName.

483 With the default setting

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

485 all packages in -d will be converted to the desired --outFormat (either EIGENSTRAT or PLINK), if the data is

486 not already in this format. This includes updating the respective POSEIDON.yml files.

487 The “old” data is not deleted, but kept around. That means conversion can result in a package with both PLINK

488 and EIGENSTRAT data, but only one is linked in the POSEIDON.yml file, and that is what will be used by

489 trident. To delete the old data in the conversion you can add the --removeOld flag.

490 Instead of -d to change Poseidon packages, the -p (+ --snpSet) or --inFormat + --genoFile + --snpFile

491 + --indFile (+ --snpSet) allow to directly convert genotype data that is not wrapped in a Poseidon package

492 and store it to a directory given in -o. See this example:

493 trident genoconvert \

494 -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \

495 --outFormat EIGENSTRAT

496 -o my_directory

497 1.2.5 Update command

498 update automatically harmonizes POSEIDON.yml files of one or multiple packages if the packages were changed.

499 This is not an automatic update from one Poseidon version to the next!

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

501 Usage: trident update (-d|--baseDir DIR) [--poseidonVersion ARG]

502 [--ignorePoseidonVersion] [--versionComponent ARG]

503 [--noChecksumUpdate] [--newContributors ARG]

504 [--logText ARG] [--force]

505 Update POSEIDON.yml files automatically

506

507 Available options:

508 -h,--help Show this help text

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

510 (could be a Poseidon repository)

511 --poseidonVersion ARG Poseidon version the packages should be updated to:

512 e.g. "2.5.3" (default: Nothing)

513 --ignorePoseidonVersion Read packages even if their poseidonVersion is not

514 compatible with the trident version. The assumption

515 is, that the package is already structurally adjusted

516 to the trident version and only the version number is

```

517         lagging behind.
518 --versionComponent ARG    Part of the package version number in the
519                             POSEIDON.yml file that should be updated: Major,
520                             Minor or Patch (see https://semver.org)
521                             (default: Patch)
522 --noChecksumUpdate        Should update of checksums in the POSEIDON.yml file
523                             be skipped
524 --ignoreGeno              ignore SNP and GenoFile
525 --newContributors ARG     Contributors to add to the POSEIDON.yml file in the
526                             form "[Firstname Lastname](Email address);..."
527 --logText ARG             Log text for this version jump in the CHANGELOG file
528                             (default: "not specified")
529 --force                   Normally the POSEIDON.yml files are only changed if
530                             the poseidonVersion is adjusted or any of the
531                             checksums change. With --force a package version
532                             update can be triggered even if this is not the case.

```

533 It can be called with a lot of optional arguments

```

534 trident update -d ... -d ... \
535     --poseidonVersion "X.X.X" \
536     --versionComponent Major/Minor/Patch \
537     --noChecksumUpdate
538     --ignoreGeno
539     --newContributors "[Firstname Lastname](Email address);..."
540     --logText "short description of the update"
541     --force

```

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

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

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

- 552 • If `--poseidonVersion` is different from the `poseidonVersion` field in the package, then that will be
553 updated.
- 554 • The `packageVersion` will be incremented. If `--versionComponent` is not set, then it falls back to `Patch`,
555 so a change in the last position of the three digit version number. `Minor` increments the middle, and `Major`
556 the first position (see [semantic versioning](#)).
- 557 • The `lastModified` field will be updated to the current day (based on your computer's system time).
- 558 • The contributors in `--newContributors` will be added to the `contributor` field if they're not there already.
- 559 • If any checksums changed, then they will be updated. If certain checksums are not set yet, then they will

560 be added. The checksum update can be skipped with `--noChecksumUpdate` or partially skipped for the
 561 genotype data with `--ignoreGeno`.

- 562 • The CHANGELOG.md file will be updated with a new row for the new version and the text in `--logText`
 563 (default: “not specified”), which will be appended as the first line of the file. If no CHANGELOG.md file
 564 exists, then it will be created and referenced in the POSEIDON.yml file.

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

568 1.3 Inspection commands

569 1.3.1 List command

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

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

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

575 List packages, groups or individuals from local or remote Poseidon
 576 repositories

577 Available options:

579 <code>-h,--help</code>	Show this help text
580 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
582 <code>--remote</code>	list packages from a remote server instead the local file system
584 <code>--remoteURL ARG</code>	URL of the remote Poseidon server (default: "https://c107-224.cloud.gwdg.de")
586 <code>--packages</code>	list all packages
587 <code>--groups</code>	list all groups, ignoring any group names after the first as specified in the Janno-file
589 <code>--individuals</code>	list individuals
590 <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, ...
594 <code>--raw</code>	output table as tsv without header. Useful for piping into grep or awk
596 <code>--ignoreGeno</code>	ignore SNP and GenoFile

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

```
598 trident list -d ... -d ... --packages
```

599 This will yield a table like this

```
600 .-----.
```

```

601 |           Title           |      Date      | Nr Individuals |
602 :=====:=====:=====:
603 | 2015_1000Genomes_1240K_haploid_pullndown | 2020-08-10 | 2535          |
604 | 2016_Mallick_SGDP1240K_diploid_pullndown | 2020-08-10 | 280           |
605 | 2018_BostonDatashare_modern_published    | 2020-08-10 | 2772          |
606 | ...                                     | ...         |               |
607 '-----'-----'-----'

```

so 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

```
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 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 `--jannoColumn Country` `--jannoColumn 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.

1.3.2 Summarise command

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

[Click here for command line details](#)

```
Usage: trident summarise (-d|--baseDir DIR) [--raw]
```

```
  Get an overview over the content of one or multiple Poseidon packages
```

```
Available options:
```

```

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

```

You can run it with

```
trident summarise -d ... -d ...
```

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.

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

641 1.3.3 Survey command

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

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

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

645 Survey the degree of context information completeness for Poseidon packages

646

647 Available options:

648	<code>-h,--help</code>	Show this help text
649	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
650		
651	<code>--raw</code>	output table as tsv without header. Useful for piping into <code>grep</code> or <code>awk</code>
652		

653 Running

654 `trident survey -d ... -d ...`

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

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

658 1.3.4 Validate command

659 `validate` checks poseidon datasets for structural correctness.

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

661 Usage: `trident validate (-d|--baseDir DIR)`

662 Check one or multiple Poseidon packages for structural correctness

663

664 Available options:

665	<code>-h,--help</code>	Show this help text
666	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
667		
668	<code>--ignoreGeno</code>	ignore SNP and GenoFile
669	<code>--fullGeno</code>	test parsing of all SNPs (by default only the first 100 SNPs are probed)
670		
671	<code>--noExitCode</code>	do not produce an explicit exit code
672	<code>--ignoreDuplicates</code>	do not stop on duplicated individual names in the package collection
673		

674 You can run it with

675 `trident validate -d ... -d ...`

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

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

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

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

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