

Contents

1	Guide for trident v1.3.0.4	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	7
1.2.4	Genoconvert command	12
1.2.5	Rectify command	14
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 v1.3.0.4

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 MODE | --debug] [--errLength INT]
        [--inPlinkPopName MODE] (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 MODE</code>	How information should be reported: NoLog, SimpleLog, DefaultLog, ServerLog or VerboseLog. (default: DefaultLog)
<code>--debug</code>	Short for <code>--logMode VerboseLog</code> .
<code>--errLength INT</code>	After how many characters should a potential error message be truncated. "Inf" for no truncation. (default: CharCount 1500)
<code>--inPlinkPopName MODE</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
-------------------	--

```

43  fetch          Download data from a remote Poseidon repository
44  forge          Select packages, groups or individuals and create a
45                  new Poseidon package from them
46  genoconvert    Convert the genotype data in a Poseidon package to a
47                  different file format
48  rectify        Adjust POSEIDON.yml files automatically to package
49                  changes
50
51  Inspection commands:
52  list           List packages, groups or individuals from local or
53                  remote Poseidon repositories
54  summarise      Get an overview over the content of one or multiple
55                  Poseidon packages
56  survey         Survey the degree of context information completeness
57                  for Poseidon packages
58  validate       Check Poseidon packages or package components for
59                  structural correctness
60
61  Trident allows to work directly with genotype data (see -p below), but its optimized for the interaction with
62  Poseidon packages, which wrap and contextualize the data. Most trident subcommands therefore have a central
63  parameter, called --baseDir or simply -d to specify one or more base directories to look for packages. For example,
64  if all Poseidon packages live inside a repository at /path/to/poseidon/packages you would simply say trident
65  <subcommand> -d /path/to/poseidon/dirs/ and trident would automatically search all subdirectories inside
66  of the repository for valid Poseidon packages (as identified by valid POSEIDON.yml files).
67
68  You can arrange a poseidon repository in a hierarchical way. For example:
69
70  /path/to/poseidon/packages
71      /modern
72          /2019_poseidon_package1
73          /2019_poseidon_package2
74      /ancient
75          /...
76          /...
77      /Reference_Genomes
78          /...
79          /...
80
81  You can use this structure to select only the level of packages you're interested in, even individual ones, and you
82  can make use of the fact that -d can be given multiple times.
83
84  Being able to specify one or multiple repositories is often not enough, as you may have your own data to
85  co-analyse with the main repository. This is easy to do, as you simply need to provide your own genotype data as
86  yet another Poseidon package to be added to your trident command. For example, let's say you have genotype
87  data in EIGENSTRAT format (trident supports EIGENSTRAT and PLINK as formats.):
88
89  ~/my_project/my_project.geno
90  ~/my_project/my_project.snp
91  ~/my_project/my_project.ind

```

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

```
88 poseidonVersion: 2.7.1
89 title: My_awesome_project
90 description: Unpublished genetic data from my awesome project
91 contributor:
92   - name: Stephan Schiffels
93     email: schiffels@institute.org
94 packageVersion: 0.1.0
95 lastModified: 2020-10-07
96 genotypeData:
97   format: EIGENSTRAT
98   genoFile: my_project.geno
99   snpFile: my_project.snp
100   indFile: my_project.ind
101   jannoFile: my_project.janno
102   bibFile: sources.bib
```

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

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

```
110 trident list -d /path/to/poseidon/packages/modern \
111   -d /path/to/poseidon/packages/ReferenceGenomes
112   -d ~/my_project --packages
```

113 1.1.1 General notes

114 **1.1.1.1 Logging and command line output** For all subcommands the general argument **--logMode**
115 defines how trident reports messages (to stderr) on the command line:

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

122 **--debug** is short for **--logMode VerboseLog** to activate this important log level more easily.

123 1.1.1.2 Duplicates

- 124 • If multiple packages in a package repository share the same **title**, then trident will try to select the
125 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 FILE) | --inFormat FORMAT --genoFile FILE
                  --snpFile FILE --indFile FILE) [--snpSet SET]
                  (-o|--outPackagePath DIR) [-n|--outPackageName STRING]
                  [--minimal]
```

Create a new Poseidon package from genotype data

Available options:

<code>-h,--help</code>	Show this help text
<code>-p,--genoOne FILE</code>	One of the input genotype data files. Expects <code>.bed</code> , <code>.bim</code> or <code>.fam</code> for PLINK and <code>.geno</code> , <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 FORMAT</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 FILE</code>	Path to the input geno file.
<code>--snpFile FILE</code>	Path to the input snp file.
<code>--indFile FILE</code>	Path to the input ind file.
<code>--snpSet SET</code>	The snpSet of the package: 1240K, HumanOrigins or Other. Only relevant for data input with <code>-p --genoOne</code>

167 or --genoFile + --snpFile + --indFile, because the
 168 packages in a -d|--baseDir already have this
 169 information in their respective POSEIDON.yml files.
 170 (default: Other)
 171 -o,--outPackagePath DIR Path to the output package directory.
 172 -n,--outPackageName STRING
 173 The output package name. This is optional: If no name
 174 is provided, then the package name defaults to the
 175 basename of the (mandatory) --outPackagePath
 176 argument. (default: Nothing)
 177 --minimal Should the output data be reduced to a necessary
 178 minimum and omit empty scaffolding?

179 The command

```
180 trident init \
181   --inFormat EIGENSTRAT/PLINK \
182   --genoFile path/to/geno_file \
183   --snpFile path/to/snp_file \
184   --indFile path/to/ind_file \
185   --snpSet 1240K|HumanOrigins|Other \
186   -o path/to/new_package_name
```

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

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

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

193 1.2.2 Fetch command

194 `fetch` allows to download Poseidon packages from a remote Poseidon server via a [Web API](#). Read more about
 195 the data available with it [here](#).

196 Click here for command line details

```
197 Usage: trident fetch (-d|--baseDir DIR)
198       (--downloadAll |
199       (--fetchFile FILE | (-f|--fetchString DSL)))
200       [--remoteURL URL] [--archive STRING]
201
```

202 Download data from a remote Poseidon repository

203

204 Available options:

205	<code>-h,--help</code>	Show this help text
206	<code>-d,--baseDir DIR</code>	A base directory to search for Poseidon packages.
207	<code>--downloadAll</code>	Download all packages the server is offering.
208	<code>--fetchFile FILE</code>	A file with a list of packages. Works just as <code>-f</code> , but
209		multiple values can also be separated by newline, not
210		just by comma. <code>-f</code> and <code>--fetchFile</code> can be combined.
211	<code>-f,--fetchString DSL</code>	List of packages to be downloaded from the remote
212		server. Package names should be wrapped in asterisks:
213		<code>*package_title*</code> . You can combine multiple values with
214		comma, so for example: <code>"*package_1*, *package_2*,</code>
215		<code>*package_3*"</code> . <code>fetchString</code> uses the same parser as
216		<code>forgeString</code> , but does not allow excludes. If groups
217		or individuals are specified, then packages which
218		include these groups or individuals are included in
219		the download.
220	<code>--remoteURL URL</code>	URL of the remote Poseidon server.
221		(default: <code>"https://server.poseidon-adna.org"</code>)
222	<code>--archive STRING</code>	The name of the Poseidon package archive that should
223		be queried. If not given, then the query falls back
224		to the default archive of the server selected with
225		<code>--remoteURL</code> . See the archive documentation at
226		https://www.poseidon-adna.org/#/archive_overview for
227		a list of archives currently available from the
228		official Poseidon Web API. (default: Nothing)

229 It works with

230 `trident fetch -d ... -d ... \`

231 `-f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<individual1>"`

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

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

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

244 `fetch` also has the optional arguments `--remote https://...` to name an alternative Poseidon server and

245 --archive to select a Poseidon archive on the server. Here is a list of the [archives available on the official](#)
246 [Poseidon server](#).

247 1.2.3 Forge command

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

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

```
251 Usage: trident forge ((-d|--baseDir DIR) |  
252                      ((-p|--genoOne FILE) | --inFormat FORMAT --genoFile FILE  
253                      --snpFile FILE --indFile FILE) [--snpSet SET])  
254                      [--forgeFile FILE | (-f|--forgeString DSL)]  
255                      [--selectSnps FILE] [--intersect] [--outFormat FORMAT]  
256                      [--minimal] [--onlyGeno] (-o|--outPackagePath DIR)  
257                      [-n|--outPackageName STRING] [--packagewise]  
258                      [--outPlinkPopName MODE]
```

260 Select packages, groups or individuals and create a new Poseidon package from
261 them

263 Available options:

264 -h,--help	Show this help text
265 -d,--baseDir DIR	A base directory to search for Poseidon packages.
266 -p,--genoOne FILE	One of the input genotype data files. Expects .bed, 267 .bim or .fam for PLINK and .geno, .snp or .ind for 268 EIGENSTRAT. The other files must be in the same 269 directory and must have the same base name.
270 --inFormat FORMAT	The format of the input genotype data: EIGENSTRAT or 271 PLINK. Only necessary for data input with --genoFile 272 + --snpFile + --indFile.
273 --genoFile FILE	Path to the input geno file.
274 --snpFile FILE	Path to the input snp file.
275 --indFile FILE	Path to the input ind file.
276 --snpSet SET	The snpSet of the package: 1240K, HumanOrigins or 277 Other. Only relevant for data input with -p --genoOne 278 or --genoFile + --snpFile + --indFile, because the 279 packages in a -d --baseDir already have this 280 information in their respective POSEIDON.yml files. 281 (default: Other)
282 --forgeFile FILE	A file with a list of packages, groups or individual 283 samples. Works just as -f, but multiple values can 284 also be separated by newline, not just by comma. 285 Empty lines are ignored and comments start with "#", 286 so everything after "#" is ignored in one line. 287 Multiple instances of -f and --forgeFile can be

288 given. They will be evaluated according to their
289 input order on the command line.

290 `-f,--forgeString DSL` List of packages, groups or individual samples to be
291 combined in the output package. Packages follow the
292 syntax `*package_title*`, populations/groups are simply
293 `group_id` and individuals `<individual_id>`. You can
294 combine multiple values with comma, so for example:
295 `"*package_1*, <individual_1>, <individual_2>,"`
296 `group_1"`. Duplicates are treated as one entry.
297 Negative selection is possible by prepending "-" to
298 the entity you want to exclude (e.g. `"*package_1*,`
299 `-<individual_1>, -group_1"`). `forge` will apply
300 excludes and includes in order. If the first entity
301 is negative, then `forge` will assume you want to merge
302 all individuals in the packages found in the `baseDirs`
303 (except the ones explicitly excluded) before the
304 exclude entities are applied. An empty `forgeString`
305 (and no `--forgeFile`) will therefore merge all
306 available individuals. If there are individuals in
307 your input packages with equal individual id, but
308 different main group or source package, they can be
309 specified with the special syntax
310 `"<package:group:individual>"`.

311 `--selectSnps FILE` To extract specific SNPs during this `forge` operation,
312 provide a Snp file. Can be either Eigenstrat (file
313 ending must be `'.snp'`) or Plink (file ending must be
314 `'.bim'`). When this option is set, the output package
315 will have exactly the SNPs listed in this file. Any
316 SNP not listed in the file will be excluded. If
317 option `'--intersect'` is also set, only the SNPs
318 overlapping between the SNP file and the forged
319 packages are output. (default: Nothing)

320 `--intersect` Whether to output the intersection of the genotype
321 files to be forged. The default (if this option is
322 not set) is to output the union of all SNPs, with
323 genotypes defined as missing in those packages which
324 do not have a SNP that is present in another package.
325 With this option set, the forged dataset will
326 typically have fewer SNPs, but less missingness.

327 `--outFormat FORMAT` The format of the output genotype data: EIGENSTRAT or
328 PLINK. (default: PLINK)

329 `--minimal` Should the output data be reduced to a necessary
330 minimum and omit empty scaffolding?

331 `--onlyGeno` Should only the resulting genotype data be returned?
332 This means the output will not be a Poseidon package.


```

333  -o,--outPackagePath DIR  Path to the output package directory.
334  -n,--outPackageName STRING
335                          The output package name. This is optional: If no name
336                          is provided, then the package name defaults to the
337                          basename of the (mandatory) --outPackagePath
338                          argument. (default: Nothing)
339  --packagewise            Skip the within-package selection step in forge. This
340                          will result in outputting all individuals in the
341                          relevant packages, and hence a superset of the
342                          requested individuals/groups. It may result in better
343                          performance in cases where one wants to forge entire
344                          packages or almost entire packages. Details: Forge
345                          conceptually performs two types of selection: First,
346                          it identifies which packages in the supplied base
347                          directories are relevant to the requested forge, i.e.
348                          whether they are either explicitly listed using
349                          *PackageName*, or because they contain selected
350                          individuals or groups. Second, within each relevant
351                          package, individuals which are not requested are
352                          removed. This option skips only the second step, but
353                          still performs the first.
354  --outPlinkPopName MODE   Where to write the population/group name into the FAM
355                          file in Plink-format. Three options are possible:
356                          asFamily (default) | asPhenotype | asBoth. See also
357                          --inPlinkPopName.

```

358 forge can be used with

```

359  trident forge -d ... -d ... \
360    -f "*package_name*, group_id, <individual_id>" \
361    -o path/to/new_package_name

```

362 where the entities (packages, groups/populations, individuals/samples) you want in the output package can be
363 denoted either as a string on the command line (-f/--forgeString), or in an input text file (--forgeFile).
364 See the section below for the syntax of this selection language. Do not forget to wrap the --forgeString query
365 in quotes.

366 Including one or multiple Poseidon packages with -d is not the only way to include data for a forge operation.
367 It is also possible to consider unpackaged genotype data directly with -p (+ --snpSet) or --inFormat +
368 --genoFile + --snpFile + --indFile (+ --snpSet). This makes the following example possible, where we
369 merge data from one Poseidon package and two genotype datasets to get a new EIGENSTRAT dataset.

```

370  trident forge \
371    -d 2017_GonzalesFortesCurrentBiology \
372    -p 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \
373    --inFormat PLINK \
374    --genoFile 2017_HaberAJHG/2017_HaberAJHG.bed \
375    --snpFile 2017_HaberAJHG/2017_HaberAJHG.bim \

```

```

376 --indFile 2017_HaberAJHG/2017_HaberAJHG.fam \
377 -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \
378 -o testpackage \
379 --outFormat EIGENSTRAT \
380 --onlyGeno

```

381 **1.2.3.1 The forge selection language** The text in `--forgeString` and `--forgeFile` are parsed as a
382 domain specific query language that describes precisely which entities should be compiled in the output package
383 of a given `forge` operation. The language has multiple syntactic elements and a specific evaluation logic.

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

- 386 • Each package title is surrounded by `*`: `*package*`. That means if you want all individuals of the Poseidon
387 package `2019_Jeong_InnerEurasia` in the output package you would add `*2019_Jeong_InnerEurasia*`
388 to the query.
- 389 • Groups/populations are not specially marked: `group`. So to get all individuals of the group
390 `Swiss_Roman_period`, you would simply add `Swiss_Roman_period`.
- 391 • Individuals/samples are surrounded by `<` and `>`: `<individual>`. `ALA026` therefore becomes `<ALA026>`. A sec-
392 ond way to denote individuals is with the more verbose and specific syntax `<package:group:individual>`.
393 Such defined individuals take precedence over differently defined ones (so: directly with `<individual>` or
394 as a subset of `*package*` or `group`). This allows to resolve duplication issues precisely – at least in cases
395 where the duplicated individuals differ in source package or primary group.

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

```

398 # Packages
399 *package1*, *package2*
400
401 # Groups and individuals from other packages beyond package1 and package2
402 group1, <individual1>, group2, <individual2>, <individual3>
403
404 # group2 has two outlier individuals that should be ignored
405 -<bad_individual1> # This one has very low coverage
406 -<bad_individual2> # This one is from a different time period

```

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

413 An empty `forgeString` will therefore merge all available individuals.

414 **1.2.3.2 Treatment of the .janno file while merging** `forge` merges and subsets `.janno` files along with
415 the genotype data. If a package lacks a `.janno` file, then a basic one will be created internally based on the

information in the genotype data, and used for the output. Missing columns across packages will be filled with n/a.

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

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

The following example illustrates the described behaviour:

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

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

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

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

434 **1.2.3.4 Treatment of the .bib file while merging** In the forge process all relevant samples for the output
 435 package are determined. This includes their .janno entries and therefore the information on the publication keys
 436 documented for them in the .janno **Publication** column. The output .bib file compiles only the relevant references
 437 for the samples in the output package. It includes the references exactly once and is sorted alphabetically (by
 438 key).

439 **1.2.3.5 Other options** Just as for **init** the output package of **forge** is created as a new directory **-o**. The
 440 title can also be explicitly defined with **-n**.

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

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

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

Input snpSet A	Input snpSet B	--intersect	Ouput snpSet
Other	*	*	Other
1240K	HumanOrigins	True	HumanOrigins
1240K	HumanOrigins	False	1240K

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

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

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

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

462 1.2.4 Genoconvert command

463 **genoconvert** converts the genotype data in a Poseidon package to a different file format. The respective entries
 464 in the POSEIDON.yml file are changed accordingly.

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

466 Usage: trident genoconvert ((-d|--baseDir DIR) |

```

467         ((-p|--genoOne FILE) | --inFormat FORMAT
468         --genoFile FILE --snpFile FILE --indFile FILE)
469         [--snpSet SET]) --outFormat FORMAT [--onlyGeno]
470         [-o|--outPackagePath DIR] [--removeOld]
471         [--outPlinkPopName MODE]
472
473     Convert the genotype data in a Poseidon package to a different file format
474
475     Available options:
476     -h,--help                Show this help text
477     -d,--baseDir DIR         A base directory to search for Poseidon packages.
478     -p,--genoOne FILE        One of the input genotype data files. Expects .bed,
479                               .bim or .fam for PLINK and .geno, .snp or .ind for
480                               EIGENSTRAT. The other files must be in the same
481                               directory and must have the same base name.
482     --inFormat FORMAT        The format of the input genotype data: EIGENSTRAT or
483                               PLINK. Only necessary for data input with --genoFile
484                               + --snpFile + --indFile.
485     --genoFile FILE          Path to the input geno file.
486     --snpFile FILE           Path to the input snp file.
487     --indFile FILE           Path to the input ind file.
488     --snpSet SET             The snpSet of the package: 1240K, HumanOrigins or
489                               Other. Only relevant for data input with -p|--genoOne
490                               or --genoFile + --snpFile + --indFile, because the
491                               packages in a -d|--baseDir already have this
492                               information in their respective POSEIDON.yml files.
493                               (default: Other)
494     --outFormat FORMAT        the format of the output genotype data: EIGENSTRAT or
495                               PLINK.
496     --onlyGeno               Should only the resulting genotype data be returned?
497                               This means the output will not be a Poseidon package.
498     -o,--outPackagePath DIR  Path to the output package directory. This is
499                               optional: If no path is provided, then the output is
500                               written to the directories where the input genotype
501                               data file (.bed/.geno) is stored. (default: Nothing)
502     --removeOld              Remove the old genotype files when creating the new
503                               ones.
504     --outPlinkPopName MODE    Where to write the population/group name into the FAM
505                               file in Plink-format. Three options are possible:
506                               asFamily (default) | asPhenotype | asBoth. See also
507                               --inPlinkPopName.
508
509     With the default setting
510     trident genoconvert -d ... -d ... --outFormat EIGENSTRAT|PLINK
511
512     all packages in -d will be converted to the desired --outFormat (either EIGENSTRAT or PLINK), if the data is

```

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

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

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

```
518 trident genoconvert \
519   -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
520   --outFormat EIGENSTRAT
521   -o my_directory
```

522 1.2.5 Rectify command

523 **rectify** automatically harmonizes POSEIDON.yml files of one or multiple packages. This is not an automatic
524 update from one Poseidon version to the next, but rather a clean-up wizard after manual modifications.

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

```
526 Usage: trident rectify (-d|--baseDir DIR) [--ignorePoseidonVersion]
527                [--poseidonVersion ??.?]
528                [--packageVersion VPART [--logText STRING]]
529                [--checksumAll | [--checksumGeno] [--checksumJanno]
530                [--checksumSSF] [--checksumBib]]
531                [--newContributors DSL]
```

532 Adjust POSEIDON.yml files automatically to package changes

533 Available options:

536	<code>-h,--help</code>	Show this help text
537	<code>-d,--baseDir DIR</code>	A base directory to search for Poseidon packages.
538	<code>--ignorePoseidonVersion</code>	Read packages even if their poseidonVersion is not 539 compatible with trident.
540	<code>--poseidonVersion ??.?</code>	Poseidon version the packages should be updated to: 541 e.g. "2.5.3".
542	<code>--packageVersion VPART</code>	Part of the package version number in the 543 POSEIDON.yml file that should be updated: Major, 544 Minor or Patch (see https://semver.org).
545	<code>--logText STRING</code>	Log text for this version in the CHANGELOG file.
546	<code>--checksumAll</code>	Update all checksums.
547	<code>--checksumGeno</code>	Update genotype data checksums.
548	<code>--checksumJanno</code>	Update .janno file checksum.
549	<code>--checksumSSF</code>	Update .ssf file checksum
550	<code>--checksumBib</code>	Update .bib file checksum.
551	<code>--newContributors DSL</code>	Contributors to add to the POSEIDON.yml file in the 552 form "[Firstname Lastname](Email address);...".

553 It can be called with a lot of optional arguments:

```
554 trident rectify -d ... -d ... \  
555     --poseidonVersion "X.X.X" \  
556     --packageVersion Major|Minor|Patch \  
557     --logText "short description of the update"  
558     --checksumAll  
559     --newContributors "[Firstname Lastname](Email address);..."
```

560 These arguments determine which fields of the POSEIDON.yml file should be modified.

- 561 • `--poseidonVersion` allows a simple change of the `poseidonVersion` field in the POSEIDON.yml file.
- 562 • `--packageVersion` increments the package version number in the first, the second or the third position.
563 It can optionally be called with `--logText`, which appends an entry to the CHANGELOG file for the
564 respective package version update. `--logText` also creates a new CHANGELOG file if it does not exist
565 yet.
- 566 • `--checksumGeno`, `--checksumJanno`, `--checksumSSF` and `--checksumBib` add or modify the respective
567 checksum fields in the POSEIDON.yml file. `--checksumAll` is a wrapper to call all of them at once.
- 568 • `--newContributors` adds new contributors.

569 :warning: As `rectify` reads and rewrites POSEIDON.yml files, it may change their inner order, layout or
570 even content (e.g. if they have fields which are not in the **POSEIDON.yml definition**). Create a backup of the
571 POSEIDON.yml file before running `rectify` if you are uncertain if this might affect you negatively.

572 1.3 Inspection commands

573 1.3.1 List command

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

575 Click here for command line details

```
576 Usage: trident list ((-d|--baseDir DIR) | --remote [--remoteURL URL]  
577                 [--archive STRING])  
578                 (--packages | --groups | --individuals  
579                 [-j|--jannoColumn COLNAME]) [--raw]
```

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

584 Available options:

585 -h,--help	Show this help text
586 -d,--baseDir DIR	A base directory to search for Poseidon packages.
587 --remote	List packages from a remote server instead the local 588 file system.
589 --remoteURL URL	URL of the remote Poseidon server. 590 (default: "https://server.poseidon-adna.org")
591 --archive STRING	The name of the Poseidon package archive that should 592 be queried. If not given, then the query falls back 593 to the default archive of the server selected with

```

594         --remoteURL. See the archive documentation at
595         https://www.poseidon-adna.org/#/archive_overview for
596         a list of archives currently available from the
597         official Poseidon Web API. (default: Nothing)
598     --packages      List all packages.
599     --groups        List all groups, ignoring any group names after the
600                     first as specified in the .janno-file.
601     --individuals    List all individuals/samples.
602     -j,--jannoColumn COLNAME List additional fields from the janno files, using
603                     the .janno column heading name, such as "Country",
604                     "Site", "Date_C14_Uncal_BP", etc..
605     --raw           Return the output table as tab-separated values
606                     without header. This is useful for piping into grep
607                     or awk.

```

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

```

609 trident list -d ... -d ... --packages

```

610 This will yield a nicely formatted table of all packages, their version and the number of individuals in them.

611 You can use `--remote` to show packages on the remote server. For example

```

612 trident list --packages --remote --archive "community-archive"

```

613 will result in a view of all packages available in one of the [public online archives](#). Just as for `fetch`, the `--archive`
614 flag allows to choose which public archive to query.

615 Independent of whether you query a local or an online archive, you can not just list packages, but also groups,
616 as defined in the third column of EIGENSTRAT `.ind` files (or the first/last column of a PLINK `.fam` file), and
617 individuals with the flags `--groups` and `--individuals` (instead of `--packages`).

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

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

625 1.3.2 Summarise command

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

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

```

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

```

629

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

631

632 Available options:

```

633 -h,--help          Show this help text

```



```

634  -d,--baseDir DIR          A base directory to search for Poseidon packages.
635  --raw                    Return the output table as tab-separated values
636                          without header. This is useful for piping into grep
637                          or awk.

```

638 You can run it with

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

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

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

644 1.3.3 Survey command

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

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

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

```
649     Survey the degree of context information completeness for Poseidon packages
```

```
650 Available options:
```

```

652  -h,--help                Show this help text
653  -d,--baseDir DIR         A base directory to search for Poseidon packages.
654  --raw                    Return the output table as tab-separated values
655                          without header. This is useful for piping into grep
656                          or awk.

```

657 Running

```
658 trident survey -d ... -d ...
```

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

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

662 1.3.4 Validate command

663 `validate` checks Poseidon packages and individual package components for structural correctness.

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

```

665 Usage: trident validate ((-d|--baseDir DIR) [--ignoreGeno] [--fullGeno]
666                        [--ignoreDuplicates] [-c|--ignoreChecksums]
667                        [--ignorePoseidonVersion] |
668                        --pym1 FILE | (-p|--genoOne FILE) | --inFormat FORMAT
669                        --genoFile FILE --snpFile FILE --indFile FILE |
670                        --janno FILE | --ssf FILE | --bib FILE) [--noExitCode]

```

671

672 Check Poseidon packages or package components for structural correctness

673

674 Available options:

675	-h,--help	Show this help text
676	-d,--baseDir DIR	A base directory to search for Poseidon packages.
677	--ignoreGeno	Ignore snp and geno file.
678	--fullGeno	Test parsing of all SNPs (by default only the first
679		100 SNPs are probed).
680	--ignoreDuplicates	Do not stop on duplicated individual names in the
681		package collection.
682	-c,--ignoreChecksums	Whether to ignore checksums. Useful for speedup in
683		debugging.
684	--ignorePoseidonVersion	Read packages even if their poseidonVersion is not
685		compatible with trident.
686	--pyml FILE	Path to a POSEIDON.yml file.
687	-p,--genoOne FILE	One of the input genotype data files. Expects .bed,
688		.bim or .fam for PLINK and .geno, .snp or .ind for
689		EIGENSTRAT. The other files must be in the same
690		directory and must have the same base name.
691	--inFormat FORMAT	The format of the input genotype data: EIGENSTRAT or
692		PLINK. Only necessary for data input with --genoFile
693		+ --snpFile + --indFile.
694	--genoFile FILE	Path to the input geno file.
695	--snpFile FILE	Path to the input snp file.
696	--indFile FILE	Path to the input ind file.
697	--janno FILE	Path to a .janno file.
698	--ssf FILE	Path to a .ssf file.
699	--bib FILE	Path to a .bib file.
700	--noExitCode	Do not produce an explicit exit code.

701 You can run it with

702 `trident validate -d ... -d ...`

703 to check packages and it will either report a success (`Validation passed`) or failure with specific error messages.

704 Instead of validating entire packages with `-d` you can also apply it to individual files and package components: `--pyml` (POSEIDON.yml), `-p` | `--inFormat` + `--genoFile` + `--snpFile` + `--indFile` (genotype data), `--janno` (.janno file), `--ssf` (.ssf file) or `--bib` (.bib file). In this case `validate` attempts to read and parse the respective files individually and reports any issues it encounters. Note that this considers the files in isolation and does not include any cross-file consistency checks.

709 When applied to packages, `validate` tries to ensure that each package adheres to the `schema definition`. Here is a list of what is checked:

- 711 • Structural correctness of the POSEIDON.yml file.
- 712 • Presence of all files references in the POSEIDON.yml file.
- 713 • Full structural correctness of .janno, .ssf and .bib file.
- 714 • Superficial correctness of genotype data files by parsing the first 100 SNPs. A full check that parses all

715 SNPs can be triggered with the `--fullGeno` option. `--ignoreGeno`, on the other hand, causes `validate`
716 to ignore the genotype data entirely, which speeds up the validation significantly.

- 717 • Correspondence of BibTeX keys in `.bib` and `.janno`
- 718 • Correspondence of sample IDs in `.janno` and `.ssf`.
- 719 • Correspondence of sample and group IDs in `.janno` and genotype data files.

720 In fact much of this validation already runs as part of the general package reading pipeline invoked for other
721 trident subcommands (e.g. `forge`). `validate` is meant to be more thorough/brittle, though, and will explicitly
722 fail if even a single package is broken. For special cases more flexibility can be enabled with the options
723 `--ignoreDuplicates`, `--ignoreChecksums` and `--ignorePoseidonVersion`.

724 Remember to run `validate` it with `--debug` to get more information in case the default output is not sufficient
725 to analyse an issue.