

Contents

1	Guide for trident v1.4.0.2 to v1.4.0.3	1
2	1.1 The trident CLI	1
3	1.1.1 General notes	3
4	1.2 Package creation and manipulation commands	4
5	1.2.1 Init command	4
6	1.2.2 Fetch command	5
7	1.2.3 Forge command	7
8	1.2.4 Genoconvert command	13
9	1.2.5 Rectify command	15
10	1.3 Inspection commands	16
11	1.3.1 List command	16
12	1.3.2 Summarise command	17
13	1.3.3 Survey command	18
14	1.3.4 Validate command	18
15		

1 Guide for trident v1.4.0.2 to v1.4.0.3

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

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

```

67 /path/to/poseidon/packages
68     /modern
69         /2019_poseidon_package1
70         /2019_poseidon_package2
71     /ancient
72         /...
73         /...
74     /Reference_Genomes
75         /...
76         /...

```

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

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

```

83 ~/my_project/my_project.geno
84 ~/my_project/my_project.snp
85 ~/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 1.1.1.1 General notes

114 **1.1.1.1.1 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 1.1.1.1.2 Package duplicates and versions

- 124 • For **trident** multiple packages in a set of base directories can share the same **title**, if they have different
125 **packageVersion** numbers. If the version numbers are identical or missing, then **trident** stops with an

exception.

- The **trident** subcommands **genoconvert**, **list**, **rectify**, **survey** and **validate** by default consider all versions of each Poseidon package in the given base directories. The **--onlyLatest** flag causes them to instead only consider the latest versions.
- **fetch** and **forge** generally consider all package versions and their selection language (see below) allows for detailed version handling.
- **summarize** always only shows results for the latest package versions.

1.1.1.3 Individual/sample duplicates

- Individual/sample names (**Poseidon_IDs**) within one package have to be unique, or **trident** will stop.
- We also discourage sample duplicates across packages in package repositories, but **trident** will generally continue with them. **validate** will fail though, if the **--ignoreDuplicates** flag is not set.
- **forge** offers a special mechanism to resolve sample duplicates within its selection language.

1.1.1.4 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.

1.1.1.5 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:

-h,--help	Show this help text
-p,--genoOne FILE	One of the input genotype data files. Expects .bed , .bim or .fam for PLINK and .geno , .snp or .ind for EIGENSTRAT. The other files must be in the same directory and must have the same base name.
--inFormat FORMAT	The format of the input genotype data: EIGENSTRAT or

```

165         PLINK. Only necessary for data input with --genoFile
166         + --snpFile + --indFile.
167     --genoFile FILE      Path to the input geno file.
168     --snpFile FILE       Path to the input snp file.
169     --indFile FILE       Path to the input ind file.
170     --snpSet SET         The snpSet of the package: 1240K, HumanOrigins or
171                          Other. Only relevant for data input with -p|--genoOne
172                          or --genoFile + --snpFile + --indFile, because the
173                          packages in a -d|--baseDir already have this
174                          information in their respective POSEIDON.yml files.
175                          (default: Other)
176     -o,--outPackagePath DIR Path to the output package directory.
177     -n,--outPackageName STRING
178                          The output package name. This is optional: If no name
179                          is provided, then the package name defaults to the
180                          basename of the (mandatory) --outPackagePath
181                          argument. (default: Nothing)
182     --minimal            Should the output data be reduced to a necessary
183                          minimum and omit empty scaffolding?

```

184 The command

```

185 trident init \
186   --inFormat EIGENSTRAT/PLINK \
187   --genoFile path/to/geno_file \
188   --snpFile path/to/snp_file \
189   --indFile path/to/ind_file \
190   --snpSet 1240K|HumanOrigins|Other \
191   -o path/to/new_package_name

```

192 requires the format (--inFormat) of your input data (either EIGENSTRAT or PLINK), the paths to the respective
193 files (--genoFile, --snpFile, --indFile), and optionally the “shape” of these files (--snpSet), so if they cover
194 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

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

198 1.2.2 Fetch command

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

201 Click here for command line details

```
202 Usage: trident fetch (-d|--baseDir DIR)
203             (--downloadAll |
204             (--fetchFile FILE | (-f|--fetchString DSL)))
205             [--remoteURL URL] [--archive STRING]
```

206
207 Download data from a remote Poseidon repository

208
209 Available options:

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

234 It works with

```
235 trident fetch -d ... -d ... \
236     -f "*package_title_1*,*package_title_2-1.0.1*,group_name,<individual1>"
```

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

240 Entities are specified using a special syntax (see also the documentation of `forge` below): packages are wrapped
241 in asterisks, with or without version appended after a dash (e.g. `*package_title*` or `*package_title-1.2.3*`),
242 group names are spelled as is, and individual names are wrapped in angular brackets (e.g. `<individual1>`).
243 Fetch will figure out which packages need to be downloaded to include all specified entities. `--downloadAll`,

which can be given instead of `-f` and `--fetchFile`, causes `fetch` to download all packages from the server. The downloaded packages are added in the first (!) `-d` directory (which gets created if it doesn't exist), but downloads are only performed if the respective packages are not already present in the latest version in any of the `-d` dirs.

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

`fetch` also has the optional arguments `--remote https://..."` to name an alternative Poseidon server and `--archive` to select a Poseidon archive on the server. Here is a list of the [archives available on the official Poseidon server](#).

1.2.3 Forge command

`forge` creates new Poseidon packages by extracting and merging packages, populations and individuals/samples from your Poseidon repositories.

[Click here for command line details](#)

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

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

Available options:

<code>-h,--help</code>	Show this help text
<code>-d,--baseDir DIR</code>	A base directory to search for Poseidon packages.
<code>-p,--genoOne FILE</code>	One of the input genotype data files. Expects .bed, .bim or .fam for PLINK and .geno, .snp or .ind 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> or <code>--genoFile</code> + <code>--snpFile</code> + <code>--indFile</code> , because the packages in a <code>-d --baseDir</code> already have this information in their respective POSEIDON.yml files.

(default: Other)

286

287 `--forgeFile FILE` A file with a list of packages, groups or individual

288 samples. Works just as `-f`, but multiple values can

289 also be separated by newline, not just by comma.

290 Empty lines are ignored and comments start with "#",

291 so everything after "#" is ignored in one line.

292 Multiple instances of `-f` and `--forgeFile` can be

293 given. They will be evaluated according to their

294 input order on the command line.

295 `-f,--forgeString DSL` List of packages, groups or individual samples to be

296 combined in the output package. Packages follow the

297 syntax `*package_title*`, populations/groups are simply

298 `group_id` and individuals `<individual_id>`. You can

299 combine multiple values with comma, so for example:

300 `"*package_1*, <individual_1>, <individual_2>, group_1"`. Duplicates are treated as one entry.

301 Negative selection is possible by prepending "-" to

302 the entity you want to exclude (e.g. `"*package_1*,`

303 `-<individual_1>, -group_1"`). `forge` will apply

304 excludes and includes in order. If the first entity

305 is negative, then `forge` will assume you want to merge

306 all individuals in the packages found in the `baseDirs`

307 (except the ones explicitly excluded) before the

308 exclude entities are applied. An empty `forgeString`

309 (and no `--forgeFile`) will therefore merge all

310 available individuals. If there are individuals in

311 your input packages with equal individual id, but

312 different main group or source package, they can be

313 specified with the special syntax

314 `"<package:group:individual>"`.

315

316 `--selectSnps FILE` To extract specific SNPs during this `forge` operation,

317 provide a Snp file. Can be either Eigenstrat (file

318 ending must be `'.snp'`) or Plink (file ending must be

319 `'.bim'`). When this option is set, the output package

320 will have exactly the SNPs listed in this file. Any

321 SNP not listed in the file will be excluded. If

322 option `'--intersect'` is also set, only the SNPs

323 overlapping between the SNP file and the forged

324 packages are output. (default: Nothing)

325 `--intersect` Whether to output the intersection of the genotype

326 files to be forged. The default (if this option is

327 not set) is to output the union of all SNPs, with

328 genotypes defined as missing in those packages which

329 do not have a SNP that is present in another package.

330 With this option set, the forged dataset will

331 typically have fewer SNPs, but less missingness.

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

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

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

338 `-o,--outPackagePath DIR` Path to the output package directory.

339 `-n,--outPackageName STRING`
 340 The output package name. This is optional: If no name
 341 is provided, then the package name defaults to the
 342 basename of the (mandatory) `--outPackagePath`
 343 argument. (default: Nothing)

344 `--packagewise` Skip the within-package selection step in forge. This
 345 will result in outputting all individuals in the
 346 relevant packages, and hence a superset of the
 347 requested individuals/groups. It may result in better
 348 performance in cases where one wants to forge entire
 349 packages or almost entire packages. Details: Forge
 350 conceptually performs two types of selection: First,
 351 it identifies which packages in the supplied base
 352 directories are relevant to the requested forge, i.e.
 353 whether they are either explicitly listed using
 354 `*PackageName*`, or because they contain selected
 355 individuals or groups. Second, within each relevant
 356 package, individuals which are not requested are
 357 removed. This option skips only the second step, but
 358 still performs the first.

359 `--outPlinkPopName MODE` Where to write the population/group name into the FAM
 360 file in Plink-format. Three options are possible:
 361 `asFamily` (default) | `asPhenotype` | `asBoth`. See also
 362 `--inPlinkPopName`.

363 `forge` can be used with

```
364 trident forge -d ... -d ... \
365   -f "*package_name*, group_id, <individual_id>" \
366   -o path/to/new_package_name
```

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

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

```

375 trident forge \
376   -d 2017_GonzalesFortesCurrentBiology \
377   -p 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \
378   --inFormat PLINK \
379   --genoFile 2017_HaberAJHG/2017_HaberAJHG.bed \
380   --snpFile 2017_HaberAJHG/2017_HaberAJHG.bim \
381   --indFile 2017_HaberAJHG/2017_HaberAJHG.fam \
382   -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \
383   -o testpackage \
384   --outFormat EIGENSTRAT \
385   --onlyGeno

```

386 **1.2.3.1 The forge selection language** The text in `--forgeString`, `--forgeFile` (and with limited syntax
387 also in `--fetchString` and `--fetchFile`) are parsed as a domain specific query language that describes precisely
388 which entities should be compiled in the output package of a given **forge** operation. The language has multiple
389 syntactic elements and a specific evaluation logic.

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

- 392 • Each package title is surrounded by `*`: `*package*`. That means if you want all individuals of the Poseidon
393 package 2019_Jeong_InnerEurasia in the output package you would add `*2019_Jeong_InnerEurasia*`
394 to the query.
- 395 • Groups/populations are not specially marked: `group`. So to get all individuals of the group
396 `Swiss_Roman_period`, you would simply add `Swiss_Roman_period`.
- 397 • Individuals/samples are surrounded by `<` and `>`: `<individual>`. ALA026 therefore becomes `<ALA026>`. A sec-
398 ond way to denote individuals is with the more verbose and specific syntax `<package:group:individual>`.
399 Such defined individuals take precedence over differently defined ones (so: directly with `<individual>` or
400 as a subset of `*package*` or `group`). This allows to resolve duplication issues precisely – at least in cases
401 where the duplicated individuals differ in source package or primary group.
- 402 • Package versions can be appended to package names, such as `*package-1.2.3*`, or `<package-1.2.3:group:individual>`.

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

```

405 # Packages
406 *package1*, *package2-1.2.3*
407
408 # Groups and individuals from other packages beyond package1 and package2
409 group1, <individual1>, group2, <individual2>, <pac1:group2:individual3>
410
411 # group2 has two outlier individuals that should be ignored
412 -<individual1> # This one has very low coverage
413 -<pac2:group3:individual4> # This one is from a different time period

```

414 By prepending `-` to entities, we can exclude them from the forged package (this feature is not avail-
415 able for `fetch`). `forge` figures out the final list of samples to include by executing all `forge`-entities in
416 order. So an entity list `*PackageA*,-<Individual1>,GroupA` may result in a different outcome than

417 ***PackageA*,GroupA,-<Individual1>**, depending on whether **<Individual1>** belongs to **GroupA** or not.

418 If the forge entity list starts with a negative entity, or if the entity list is empty, **forge** will implicitly assume
419 you want to include all individuals in all **latest** versions of packages found in the base directories (except the
420 ones explicitly excluded, of course).

421 The specific semantics of the various ways to include or exclude entities are:

422 1.2.3.1.1 Inclusion queries

- 423 • ***Pac1***: Select all individuals in the latest version of package “Pac1”
- 424 • ***Pac1-1.0.1***: Select all individuals in package “Pac1” with version “1.0.1”
- 425 • **Group1**: Select all individuals associated with “Group1” in all latest versions of all packages
- 426 • **<Ind1>**: Select the individual named “Ind1”, searching in all latest packages.
- 427 • **<Pac1:Group1:Ind1>**: Select the individual named “Ind1” associated with “Group1” in the latest version
428 of package “Pac1”
- 429 • **<Pac1-1.0.1:Group1:Ind1>**: Select the individual named “Ind1” associated with “Group1” in the package
430 “Pac1” with version “1.0.1”

431 1.2.3.1.2 Exclusion queries

- 432 • **-*Pac1***: Remove all individuals in all versions of package “Pac1”
- 433 • **-*Pac1-1.0.1***: Remove only individuals in package “Pac1” with version “1.0.1” (but leave other versions
434 in)
- 435 • **-Group1**: Remove all individuals associated with “Group1” in all versions of all packages (not just the
436 latest)
- 437 • **-<Ind1>**: Remove all individuals named “Ind1” in all versions of all packages (not just the latest).
- 438 • **-<Pac1:Group1:Ind1>**: Remove the individual named “Ind1” associated with “Group1”, searching in all
439 versions of package “Pac1”
- 440 • **-<Pac1-1.0.1:Group1:Ind1>**: Remove the individual named “Ind1” associated with “Group1”, but only
441 if they are in “Pac1” with version “1.0.1”

442 If a query results in multiple individuals with the same name, forge will throw an error.

443 **1.2.3.2 Treatment of the .janno file while merging** **forge** merges and subsets .janno files along with
444 the genotype data. If a package lacks a .janno file, then a basic one will be created internally based on the
445 information in the genotype data, and used for the output. Missing columns across packages will be filled with
446 **n/a**.

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

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

454 The following example illustrates the described behaviour:

455 **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.

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

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

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

`forge` has a an optional flag `--intersect`, that defines, if the genotype data from different packages should be merged with an `union` or an `intersect` operation. The default (if this option is not set) is to output the union of all SNPs, with genotypes defined as missing in samples from packages which do not have a SNP that is

present in another package. With this option set, on the other hand, the forged dataset will typically have fewer SNPs, but less missingness.

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

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

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

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

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

1.2.4 Genoconvert command

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

[Click here for command line details](#)

```
Usage: trident genoconvert ((-d|--baseDir DIR) |
    ((-p|--genoOne FILE) | --inFormat FORMAT
    --genoFile FILE --snpFile FILE --indFile FILE)
    [--snpSet SET]) --outFormat FORMAT [--onlyGeno]
    [-o|--outPackagePath DIR] [--removeOld]
    [--outPlinkPopName MODE] [--onlyLatest]
```

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

Available options:

<code>-h,--help</code>	Show this help text
<code>-d,--baseDir DIR</code>	A base directory to search for Poseidon packages.
<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.

```

511 --inFormat FORMAT      The format of the input genotype data: EIGENSTRAT or
512                        PLINK. Only necessary for data input with --genoFile
513                        + --snpFile + --indFile.
514 --genoFile FILE        Path to the input geno file.
515 --snpFile FILE          Path to the input snp file.
516 --indFile FILE          Path to the input ind file.
517 --snpSet SET            The snpSet of the package: 1240K, HumanOrigins or
518                        Other. Only relevant for data input with -p|--genoOne
519                        or --genoFile + --snpFile + --indFile, because the
520                        packages in a -d|--baseDir already have this
521                        information in their respective POSEIDON.yml files.
522                        (default: Other)
523 --outFormat FORMAT      the format of the output genotype data: EIGENSTRAT or
524                        PLINK.
525 --onlyGeno              Should only the resulting genotype data be returned?
526                        This means the output will not be a Poseidon package.
527 -o,--outPackagePath DIR Path to the output package directory. This is
528                        optional: If no path is provided, then the output is
529                        written to the directories where the input genotype
530                        data file (.bed/.geno) is stored. (default: Nothing)
531 --removeOld             Remove the old genotype files when creating the new
532                        ones.
533 --outPlinkPopName MODE  Where to write the population/group name into the FAM
534                        file in Plink-format. Three options are possible:
535                        asFamily (default) | asPhenotype | asBoth. See also
536                        --inPlinkPopName.
537 --onlyLatest            Consider only the latest versions of packages, or the
538                        groups and individuals within the latest versions of
539                        packages, respectively.
540
540 With the default setting
541
541 trident genoconvert -d ... -d ... --outFormat EIGENSTRAT|PLINK
542
542 all packages in -d will be converted to the desired --outFormat (either EIGENSTRAT or PLINK), if the data is
543 not already in this format. This includes updating the respective POSEIDON.yml files.
544
544 The “old” data is not deleted, but kept around. That means conversion can result in a package with both PLINK
545 and EIGENSTRAT data, but only one is linked in the POSEIDON.yml file, and that is what will be used by
546 trident. To delete the old data in the conversion you can add the --removeOld flag.
547
547 Instead of -d to change Poseidon packages, the -p (+ --snpSet) or --inFormat + --genoFile + --snpFile
548 + --indFile (+ --snpSet) allow to directly convert genotype data that is not wrapped in a Poseidon package
549 and store it to a directory given in -o. See this example:
550
550 trident genoconvert \
551     -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
552     --outFormat EIGENSTRAT
553     -o my_directory

```

554 1.2.5 Rectify command

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

557 Click here for command line details

```
558 Usage: trident rectify (-d|--baseDir DIR) [--ignorePoseidonVersion]
559                        [--poseidonVersion ?.??.?]
560                        [--packageVersion VPART [--logText STRING]]
561                        [--checksumAll | [--checksumGeno] [--checksumJanno]
562                        [--checksumSSF] [--checksumBib]]
563                        [--newContributors DSL] [--onlyLatest]
```

565 Adjust POSEIDON.yml files automatically to package changes

567 Available options:

568	-h,--help	Show this help text
569	-d,--baseDir DIR	A base directory to search for Poseidon packages.
570	--ignorePoseidonVersion	Read packages even if their poseidonVersion is not
571		compatible with trident.
572	--poseidonVersion ?.??.?	Poseidon version the packages should be updated to:
573		e.g. "2.5.3".
574	--packageVersion VPART	Part of the package version number in the
575		POSEIDON.yml file that should be updated: Major,
576		Minor or Patch (see https://semver.org).
577	--logText STRING	Log text for this version in the CHANGELOG file.
578	--checksumAll	Update all checksums.
579	--checksumGeno	Update genotype data checksums.
580	--checksumJanno	Update .janno file checksum.
581	--checksumSSF	Update .ssf file checksum
582	--checksumBib	Update .bib file checksum.
583	--newContributors DSL	Contributors to add to the POSEIDON.yml file in the
584		form "[Firstname Lastname](Email address);...".
585	--onlyLatest	Consider only the latest versions of packages, or the
586		groups and individuals within the latest versions of
587		packages, respectively.

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

```
589 trident rectify -d ... -d ... \
590     --poseidonVersion "X.X.X" \
591     --packageVersion Major|Minor|Patch \
592     --logText "short description of the update"
593     --checksumAll
594     --newContributors "[Firstname Lastname](Email address);..."
```

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

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

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

1.3 Inspection commands

1.3.1 List command

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

[Click here for command line details](#)

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

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

Available options:

<code>-h,--help</code>	Show this help text
<code>-d,--baseDir DIR</code>	A base directory to search for Poseidon packages.
<code>--remote</code>	List packages from a remote server instead the local file system.
<code>--remoteURL URL</code>	URL of the remote Poseidon server. (default: "https://server.poseidon-adna.org")
<code>--archive STRING</code>	The name of the Poseidon package archive that should be queried. If not given, then the query falls back to the default archive of the server selected with <code>--remoteURL</code> . See the archive documentation at https://www.poseidon-adna.org/#/archive_overview for a list of archives currently available from the official Poseidon Web API. (default: Nothing)
<code>--packages</code>	List all packages.
<code>--groups</code>	List all groups, ignoring any group names after the first as specified in the <code>.janno</code> -file.
<code>--individuals</code>	List all individuals/samples.
<code>-j,--jannoColumn COLNAME</code>	List additional fields from the janno files, using

the .janno column heading name, such as "Country", "Site", "Date_C14_Uncal_BP", etc..

`--raw` Return the output table as tab-separated values without header. This is useful for piping into `grep` or `awk`.

`--onlyLatest` Consider only the latest versions of packages, or the groups and individuals within the latest versions of packages, respectively.

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

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

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

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

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

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

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

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

Note that if you want a less fancy table, for example because you want to load this into Excel, or pipe into another command that cannot deal with the 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:

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

You can run it with

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

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

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

682 1.3.3 Survey command

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

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

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

686

687 Survey the degree of context information completeness for Poseidon packages

688

689 Available options:

690 <code>-h,--help</code>	Show this help text
691 <code>-d,--baseDir DIR</code>	A base directory to search for Poseidon packages.
692 <code>--raw</code>	Return the output table as tab-separated values 693 without header. This is useful for piping into <code>grep</code> 694 or <code>awk</code> .
695 <code>--onlyLatest</code>	Consider only the latest versions of packages, or the 696 groups and individuals within the latest versions of 697 packages, respectively.

698 Running

699 `trident survey -d ... -d ...`

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

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

703 1.3.4 Validate command

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

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

706 Usage: `trident validate ((-d|--baseDir DIR) [--ignoreGeno] [--fullGeno]
707 [--ignoreDuplicates] [-c|--ignoreChecksums]
708 [--ignorePoseidonVersion] |
709 --pym1 FILE | (-p|--genoOne FILE) | --inFormat FORMAT
710 --genoFile FILE --snpFile FILE --indFile FILE |
711 --janno FILE | --ssf FILE | --bib FILE) [--noExitCode]
712 [--onlyLatest]`

713

714 Check Poseidon packages or package components for structural correctness

715

```

716 Available options:
717   -h,--help           Show this help text
718   -d,--baseDir DIR    A base directory to search for Poseidon packages.
719   --ignoreGeno         Ignore snp and geno file.
720   --fullGeno           Test parsing of all SNPs (by default only the first
721                        100 SNPs are probed).
722   --ignoreDuplicates   Do not stop on duplicated individual names in the
723                        package collection.
724   -c,--ignoreChecksums Whether to ignore checksums. Useful for speedup in
725                        debugging.
726   --ignorePoseidonVersion Read packages even if their poseidonVersion is not
727                        compatible with trident.
728   --pym1 FILE          Path to a POSEIDON.yml file.
729   -p,--genoOne FILE    One of the input genotype data files. Expects .bed,
730                        .bim or .fam for PLINK and .geno, .snp or .ind for
731                        EIGENSTRAT. The other files must be in the same
732                        directory and must have the same base name.
733   --inFormat FORMAT    The format of the input genotype data: EIGENSTRAT or
734                        PLINK. Only necessary for data input with --genoFile
735                        + --snpFile + --indFile.
736   --genoFile FILE      Path to the input geno file.
737   --snpFile FILE       Path to the input snp file.
738   --indFile FILE       Path to the input ind file.
739   --janno FILE         Path to a .janno file.
740   --ssf FILE           Path to a .ssf file.
741   --bib FILE           Path to a .bib file.
742   --noExitCode         Do not produce an explicit exit code.
743   --onlyLatest         Consider only the latest versions of packages, or the
744                        groups and individuals within the latest versions of
745                        packages, respectively.

746 You can run it with

747 trident validate -d ... -d ...

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

749 Instead of validating entire packages with -d you can also apply it to individual files and package components: --pym1 (POSEIDON.yml), -p | --inFormat + --genoFile + --snpFile + --indFile (genotype
750 data), --janno (.janno file), --ssf (.ssf file) or --bib (.bib file). In this case validate attempts to read and
751 parse the respective files individually and reports any issues it encounters. Note that this considers the files in
752 isolation and does not include any cross-file consistency checks.

753

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

756   • Structural correctness of the POSEIDON.yml file.
757   • Presence of all files references in the POSEIDON.yml file.
758   • Full structural correctness of .janno, .ssf and .bib file.

```

- 759 • Superficial correctness of genotype data files by parsing the first 100 SNPs. A full check that parses all
760 SNPs can be triggered with the `--fullGeno` option. `--ignoreGeno`, on the other hand, causes `validate`
761 to ignore the genotype data entirely, which speeds up the validation significantly.
- 762 • Correspondence of BibTeX keys in `.bib` and `.janno`
- 763 • Correspondence of sample IDs in `.janno` and `.ssf`.
- 764 • Correspondence of sample and group IDs in `.janno` and genotype data files.

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

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