

# Contents

1	<b>1 Guide for trident v0.29.0</b>	<b>1</b>
2	1.1 Poseidon package repositories . . . . .	1
3	1.2 Analysing your own dataset outside of the main repository . . . . .	2
4	1.3 Package creation and manipulation commands . . . . .	2
5	1.3.1 Init command . . . . .	2
6	1.3.2 Fetch command . . . . .	4
7	1.3.3 Forge command . . . . .	5
8	1.3.4 Genoconvert command . . . . .	9
9	1.3.5 Update command . . . . .	10
10	1.4 Inspection commands . . . . .	12
11	1.4.1 List command . . . . .	12
12	1.4.2 Summarise command . . . . .	13
13	1.4.3 Survey command . . . . .	13
14	1.4.4 Validate command . . . . .	14
15		

## 1 Guide for trident v0.29.0

### 1.1 Poseidon package repositories

Trident generally requires Poseidon “packages” to work with (since version 0.28.0 it also supports direct interaction with “unpackaged” genotype data – see `-p` below). Most trident subcommands therefore have a central parameter, called `--baseDir` or simply `-d` to specify one or more base directories to look for packages. For example, if all Poseidon packages live inside a repository at `/path/to/poseidon/packages` you would simply say `trident <subcommand> -d /path/to/poseidon/dirs/` and trident would automatically search all subdirectories inside of the repository for valid poseidon packages (as identified by valid `POSEIDON.yml` files).

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

```
/path/to/poseidon/packages
  /modern
    /2019_poseidon_package1
    /2019_poseidon_package2
  /ancient
    /...
    /...
  /Reference_Genomes
    /...
    /...
  /Archaic_Humans
    /...
    /...
```

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

Let’s use the `list` command to list all packages in the `modern` and `Reference_Genomes`:

```

41 trident list -d /path/to/poseidon/packages/modern \
42   -d /path/to/poseidon/packages/ReferenceGenomes --packages

```

## 43 1.2 Analysing your own dataset outside of the main repository

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

```

48 ~/my_project/my_project.geno
49 ~/my_project/my_project.snp
50 ~/my_project/my_project.ind

```

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

```

53 poseidonVersion: 2.5.0
54 title: My_awesome_project
55 description: Unpublished genetic data from my awesome project
56 contributor:
57   - name: Stephan Schiffels
58     email: schiffels@institute.org
59 packageVersion: 0.1.0
60 lastModified: 2020-10-07
61 genotypeData:
62   format: EIGENSTRAT
63   genoFile: my_project.geno
64   snpFile: my_project.snp
65   indFile: my_project.ind
66   jannoFile: my_project.janno
67   bibFile: sources.bib

```

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

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

```

74 trident list -d /path/to/poseidon/packages/modern \
75   -d /path/to/poseidon/packages/ReferenceGenomes
76   -d ~/my_project --packages

```

## 77 1.3 Package creation and manipulation commands

### 78 1.3.1 Init command

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

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

```
82 Usage: trident init ((-p|--genoOne ARG) | (-r|--inFormat ARG)
83                (-g|--genoFile ARG) (-s|--snpFile ARG) (-i|--indFile ARG))
84                [--snpSet ARG] (-o|--outPackagePath ARG)
85                [-n|--outPackageName ARG] [--minimal]
```

86 Create a new Poseidon package from genotype data

87  
88 Available options:

```
89  -h,--help                Show this help text
90  -p,--genoOne ARG         one of the input genotype data files. Expects .bed or
91                          .bim or .fam for PLINK and .geno or .snp or .ind for
92                          EIGENSTRAT. The other files must be in the same
93                          directory and must have the same base name
94  -r,--inFormat ARG        the format of the input genotype data: EIGENSTRAT or
95                          PLINK
96  -g,--genoFile ARG        the input geno file path
97  -s,--snpFile ARG         the input snp file path
98  -i,--indFile ARG         the input ind file path
99  --snpSet ARG             the snpSet of the new package: 1240K, HumanOrigins or
100                          Other. Default: Other
101  -o,--outPackagePath ARG  the output package directory path
102  -n,--outPackageName ARG  the output package name - this is optional: If no
103                          name is provided, then the package name defaults to
104                          the basename of the (mandatory) --outPackagePath
105                          argument
106  --minimal                should only a minimal output package be created?
```

107 The command

```
108 trident init \
109   -r EIGENSTRAT/PLINK \
110   -g path/to/geno_file \
111   -s path/to/snp_file \
112   -i path/to/ind_file \
113   --snpSet 1240K|HumanOrigins|Other \
114   -o path/to/new_package_name
```

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

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

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

### 122 1.3.2 Fetch command

123 `fetch` allows to download poseidon packages from a remote poseidon server.

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

```
125 Usage: trident fetch (-d|--baseDir DIR) [-f|--fetchString ARG] [--fetchFile ARG]
126                [--remoteURL ARG] [-u|--upgrade] [--downloadAll]
```

127 Download data from a remote Poseidon repository

128  
129 Available options:

130	<code>-h,--help</code>	Show this help text
131	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
133	<code>-f,--fetchString ARG</code>	List of packages to be downloaded from the remote server. Package names should be wrapped in asterisks: 135 <code>*package_title*</code> . You can combine multiple values with 136 comma, so for example: <code>"*package_1*, *package_2*,</code> 137 <code>*package_3*"</code> . <code>fetchString</code> uses the same parser as 138 <code>forgeString</code> , but does not allow excludes. If groups 139 or individuals are specified, then packages which 140 include these groups or individuals are included in 141 the download.
142	<code>--fetchFile ARG</code>	A file with a list of packages. Works just as <code>-f</code> , but 143 multiple values can also be separated by newline, not 144 just by comma. <code>-f</code> and <code>--fetchFile</code> can be combined.
145	<code>--remoteURL ARG</code>	URL of the remote Poseidon 146 server (default: <code>"https://c107-224.cloud.gwdg.de"</code> )
147	<code>-u,--upgrade</code>	overwrite outdated local package versions
148	<code>--downloadAll</code>	download all packages the server is offering

149 It works with

```
150 trident fetch -d ... -d ... \  
151 -f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<Individual1>" \  
152 --fetchFile path/to/forgFile
```

153 and the entities you want to download must be listed either in a simple string with comma-separated values  
154 (`-f/--fetchString`) or in a text file (`--fetchFile`). Entities are specified using a special syntax: Package titles  
155 are wrapped in asterisks: `package_title` (see also the documentation of `forge` below), group names are spelled  
156 as is, and individual names are wrapped in angular brackets, like `<Individual1>`. Fetch will figure out which  
157 packages need to be downloaded to include all specified entities. `--downloadAll` causes fetch to ignore `-f` and  
158 download all packages from the server. The downloaded packages are added in the first (!) `-d` directory, but  
159 downloads are only performed if the respective packages are not already present in an up-to-date version in any  
160 of the `-d` dirs.

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

163 `fetch` also has the optional arguments `--remote https://..."` do name an alternative poseidon server. The  
164 default points to the [DAG server](#).

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

### 167 1.3.3 Forge command

168 `forge` creates new poseidon packages by extracting and merging packages, populations and individuals from  
169 your poseidon repositories.

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

171 Usage: `trident forge [-d|--baseDir DIR]`

```
172     [
173         ((-p|--genoOne ARG) | (-r|--inFormat ARG)
174         (-g|--genoFile ARG) (-s|--snpFile ARG)
175         (-i|--indFile ARG)) [--snpSet ARG]]
176     [--forgeFile ARG | (-f|--forgeString ARG)]
177     [--selectSnps ARG] [--intersect] [--outFormat ARG]
178     [--minimal] [--onlyGeno] (-o|--outPackagePath ARG)
179     [-n|--outPackageName ARG] [-w|--warnings] [--no-extract]
```

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

182  
183 Available options:

184 -h,--help	Show this help text
185 -d,--baseDir DIR	a base directory to search for Poseidon Packages 186 (could be a Poseidon repository)
187 -p,--genoOne ARG	one of the input genotype data files. Expects .bed or 188 .bim or .fam for PLINK and .geno or .snp or .ind for 189 EIGENSTRAT. The other files must be in the same 190 directory and must have the same base name
191 -r,--inFormat ARG	the format of the input genotype data: EIGENSTRAT or 192 PLINK
193 -g,--genoFile ARG	the input geno file path
194 -s,--snpFile ARG	the input snp file path
195 -i,--indFile ARG	the input ind file path
196 --snpSet ARG	the snpSet of the new package: 1240K, HumanOrigins or 197 Other. Default: Other
198 --forgeFile ARG	A file with a list of packages, groups or individual 199 samples. Works just as -f, but multiple values can 200 also be separated by newline, not just by comma. 201 Empty lines are ignored and comments start with "#", 202 so everything after "#" is ignored in one line.

203 `-f,--forgeString ARG` List of packages, groups or individual samples to be  
204 combined in the output package. Packages follow the  
205 syntax `*package_title*`, populations/groups are simply  
206 `group_id` and individuals `<individual_id>`. You can  
207 combine multiple values with comma, so for example:  
208 `"*package_1*, <individual_1>, <individual_2>,"`  
209 `group_1"`. Duplicates are treated as one entry.  
210 Negative selection is possible by prepending "-" to  
211 the entity you want to exclude (e.g. `"*package_1*,`  
212 `-<individual_1>, -group_1"`). `forge` will apply  
213 excludes and includes in order. If the first entity  
214 is negative, then `forge` will assume you want to merge  
215 all individuals in the packages found in the `baseDirs`  
216 (except the ones explicitly excluded) before the  
217 exclude entities are applied. An empty `forgeString`  
218 will therefore merge all available individuals.

219 `--selectSnps ARG` To extract specific SNPs during this `forge` operation,  
220 provide a Snp file. Can be either Eigenstrat (file  
221 ending must be `'.snp'`) or Plink (file ending must be  
222 `'.bim'`). When this option is set, the output package  
223 will have exactly the SNPs listed in this file. Any  
224 SNP not listed in the file will be excluded. If  
225 option `'--intersect'` is also set, only the SNPs  
226 overlapping between the SNP file and the forged  
227 packages are output.

228 `--intersect` Whether to output the intersection of the genotype  
229 files to be forged. The default (if this option is  
230 not set) is to output the union of all SNPs, with  
231 genotypes defined as missing in those packages which  
232 do not have a SNP that is present in another package.  
233 With this option set, the forged dataset will  
234 typically have fewer SNPs, but less missingness.

235 `--outFormat ARG` the format of the output genotype data: EIGENSTRAT or  
236 PLINK. Default: PLINK

237 `--minimal` should only a minimal output package be created?

238 `--onlyGeno` should only the resulting genotype data be returned?  
239 This means the output will not be a Poseidon package

240 `-o,--outPackagePath ARG` the output package directory path

241 `-n,--outPackageName ARG` the output package name - this is optional: If no  
242 name is provided, then the package name defaults to  
243 the basename of the (mandatory) `--outPackagePath`  
244 argument

245 `-w,--warnings` Show all warnings for merging genotype data

246 `--no-extract` Skip the selection step in `forge`. This will result in  
247 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 or almost entire packages. Note that this will also ignore any ordering in the output groups/individuals. With this option active, individuals from the relevant packages will just be written in the order that they appear in the original packages.

forge can be used with

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

where the entities (packages, groups/populations, individuals/samples) you want in the output package can be denoted either as a simple string with comma-separated values (-f/--forgeString) or in a text file (--forgeFile).

Including one or multiple Poseidon packages with -d is not the only way to include data for a forge operation. It is also possible to include unpackaged genotype data directly with -r + -g + -s + -i (+ --snpSet) or -p (+ --snpSet). This makes the following example possible, where we merge data from one Poseidon package and two genotype datasets.

```
trident forge \
  -d 2017_GonzalesFortesCurrentBiology \
  -r PLINK \
  -g 2017_HaberAJHG/2017_HaberAJHG.bed \
  -s 2017_HaberAJHG/2017_HaberAJHG.bim \
  -i 2017_HaberAJHG/2017_HaberAJHG.fam \
  -r PLINK \
  -g 2018_VeeramahPNAS/2018_VeeramahPNAS.bed \
  -i 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \
  -s 2018_VeeramahPNAS/2018_VeeramahPNAS.bim \
  -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \
  -o testpackage \
  --onlyGeno
```

**1.3.3.1 The forge selection language** Entities in the --forgeString or the --forgeFile have to be marked in a certain way:

- Each package is surrounded by \*, so if you want all individuals of 2019\_Jeong\_InnerEurasia in the output package you would add \*2019\_Jeong\_InnerEurasia\* to the list.
- Groups/populations are not specially marked. So to get all individuals of the group Swiss\_Roman\_period, you would simply add Swiss\_Roman\_period.
- Individuals/samples are surrounded by < and >, so ALA026 becomes <ALA026>.

Do not forget to wrap the forgeString in quotes.

290 You can either use `-f/--forgeString` or `--forgeFile`. In the file each line is treated as a separate `forgeString`,  
 291 empty lines are ignored and `#`s start comments. So this is a valid `forgeFile`:

```
292 # Packages
293 *package1*, *package2*
294
295 # Groups and individuals from other packages beyond package1 and package2
296 group1, <individual1>, group2, <individual2>, <individual3>
297
298 # group2 has two outlier individuals that should be ignored
299 -<bad_individual1> # This one has very low coverage
300 -<bad_individual2> # This one is from a different time period
```

301 By prepending `-` to the bad individuals, we can exclude them from the forged package. `forge` figures out the final list of samples to include by executing all `forge`-entities in order. So an entity list  
 302 `*PackageA*, -<Individual1>, GroupA` may result in a different outcome than `*PackageA*, GroupA, -<Individual1>`,  
 303 depending on whether `<Individual1>` belongs to `GroupA` or not. If the `forge` entity list starts with a negative  
 304 entity, or if the entity list is empty, `forge` will implicitly assume you want to include all individuals in all  
 305 packages found in the `baseDirs` (except the ones explicitly excluded, of course). An empty `forgeString` will  
 306 therefore merge all available individuals.

308 **1.3.3.2 Other options** Just as for `init` the output package of `forge` is created as a new directory `-o`. The  
 309 title can also be explicitly defined with `-n`.

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

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

318 `--intersect` also influences the automatic determination of the `snpSet` field in the `POSEIDON.yml` file for the  
 319 resulting package. If the `snpSets` of all input packages are identical, then the resulting package will just inherit  
 320 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

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



325 Merging genotype data across different data sources and file formats is tricky. `forge` is more verbose about  
326 potential issues, if the `-w/--warnings` flag is set.

### 327 1.3.4 Genoconvert command

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

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

```
331 Usage: trident genoconvert [-d|--baseDir DIR]
332                               [
333                               ((-p|--genoOne ARG) | (-r|--inFormat ARG)
334                               (-g|--genoFile ARG) (-s|--snpFile ARG)
335                               (-i|--indFile ARG)) [--snpSet ARG]]
336                               --outFormat ARG [--onlyGeno]
337                               [-o|--outPackagePath ARG] [--removeOld]
338   Convert the genotype data in a Poseidon package to a different file format
```

340 Available options:

341	<code>-h,--help</code>	Show this help text
342	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
344	<code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects .bed or 345 .bim or .fam for PLINK and .geno or .snp or .ind for 346 EIGENSTRAT. The other files must be in the same 347 directory and must have the same base name
348	<code>-r,--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or 349 PLINK
350	<code>-g,--genoFile ARG</code>	the input geno file path
351	<code>-s,--snpFile ARG</code>	the input snp file path
352	<code>-i,--indFile ARG</code>	the input ind file path
353	<code>--snpSet ARG</code>	the snpSet of the new package: 1240K, HumanOrigins or 354 Other. Default: Other
355	<code>--outFormat ARG</code>	the format of the output genotype data: EIGENSTRAT or 356 PLINK.
357	<code>--onlyGeno</code>	should only the resulting genotype data be returned? 358 This means the output will not be a Poseidon package
359	<code>-o,--outPackagePath ARG</code>	the output package directory path - this is optional: 360 If no path is provided, then the output is written to 361 the directories where the input genotype data file 362 (.bed/.geno) is stored
363	<code>--removeOld</code>	Remove the old genotype files when creating the new 364 ones

365 With the default setting

```
366 trident genoconvert -d ... -d ... --outFormat EIGENSTRAT|PLINK
```

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

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

Instead of `-d` to change Poseidon packages, the combination `-r + -g + -s + -i (+ --snpSet)` or `-p (+ --snpSet)` allows to directly convert genotype data that is not wrapped in a Poseidon package and store it to a directory given in `-o`. See this example:

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

### 1.3.5 Update command

`update` automatically updates POSEIDON.yml files of one or multiple packages if the packages were changed.

[Click here for command line details](#)

```
Usage: trident update (-d|--baseDir DIR) [--poseidonVersion ARG]  
        [--ignorePoseidonVersion] [--versionComponent ARG]  
        [--noChecksumUpdate] [--newContributors ARG]  
        [--logText ARG] [--force]
```

Update POSEIDON.yml files automatically

#### Available options:

<code>-h,--help</code>	Show this help text
<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
<code>--poseidonVersion ARG</code>	Poseidon version the packages should be updated to: e.g. "2.5.3" (default: Nothing)
<code>--ignorePoseidonVersion</code>	Read packages even if their poseidonVersion is not compatible with the trident version. The assumption is, that the package is already structurally adjusted to the trident version and only the version number is lagging behind.
<code>--versionComponent ARG</code>	Part of the package version number in the POSEIDON.yml file that should be updated: Major, Minor or Patch (see <a href="https://semver.org">https://semver.org</a> ) (default: Patch)
<code>--noChecksumUpdate</code>	Should update of checksums in the POSEIDON.yml file be skipped
<code>--ignoreGeno</code>	ignore SNP and GenoFile
<code>--newContributors ARG</code>	Contributors to add to the POSEIDON.yml file in the form "[Firstname Lastname](Email address);..."
<code>--logText ARG</code>	Log text for this version jump in the CHANGELOG

```

409         file (default: "not specified")
410     --force           Normally the POSEIDON.yml files are only changed if
411                       the poseidonVersion is adjusted or any of the
412                       checksums change. With --force a package version
413                       update can be triggered even if this is not the case.

```

414 It can be called with a lot of optional arguments

```

415 trident update -d ... -d ... \
416   --poseidonVersion "X.X.X" \
417   --versionComponent Major/Minor/Patch \
418   --noChecksumUpdate
419   --ignoreGeno
420   --newContributors "[Firstname Lastname](Email address);..."
421   --logText "short description of the update"
422   --force

```

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

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

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

- 433 • If `--poseidonVersion` is different from the `poseidonVersion` field in the package, then that will be  
 434 updated.
- 435 • The `packageVersion` will be incremented. If `--versionComponent` is not set, then it falls back to `Patch`,  
 436 so a change in the last position of the three digit version number. `Minor` increments the middle, and `Major`  
 437 the first position (see [semantic versioning](#)).
- 438 • The `lastModified` field will be updated to the current day (based on your computer's system time).
- 439 • The contributors in `--newContributors` will be added to the `contributor` field if they're not there already.
- 440 • If any checksums changed, then they will be updated. If certain checksums are not set yet, then they will  
 441 be added. The checksum update can be skipped with `--noChecksumUpdate` or partially skipped for the  
 442 genotype data with `--ignoreGeno`.
- 443 • The CHANGELOG.md file will be updated with a new row for the new version and the text in `--logText`  
 444 (default: "not specified"), which will be appended as the first line of the file. If no CHANGELOG.md file  
 445 exists, then it will be created and referenced in the POSEIDON.yml file.

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

## 1.4 Inspection commands

### 1.4.1 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 ARG])
                (--packages | --groups | --individuals
                [-j|--jannoColumn JANNO_HEADER]) [--raw]
List packages, groups or individuals from local or remote Poseidon
repositories
```

Available options:

```
-h,--help          Show this help text
-d,--baseDir DIR   a base directory to search for Poseidon Packages
                   (could be a Poseidon repository)
--remote           list packages from a remote server instead the local
                   file system
--remoteURL ARG    URL of the remote Poseidon
                   server (default: "https://c107-224.cloud.gwdg.de")
--packages         list all packages
--groups           list all groups, ignoring any group names after the
                   first as specified in the Janno-file
--individuals      list individuals
-j,--jannoColumn JANNO_HEADER
                   list additional fields from the janno files, using
                   the Janno column heading name, such as Country, Site,
                   Date_C14_Uncal_BP, Endogenous, ...
--raw             output table as tsv without header. Useful for piping
                   into grep or awk
--ignoreGeno       ignore SNP and GenoFile
```

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

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

This will yield a table like this

```
.------.------.------.
|          Title          |    Date    | Nr Individuals |
:=====:=====:=====:
| 2015_1000Genomes_1240K_haploid_pulldown | 2020-08-10 | 2535          |
| 2016_Mallick_SGDP1240K_diploid_pulldown | 2020-08-10 | 280           |
| 2018_BostonDatashare_modern_published   | 2020-08-10 | 2772          |
| ...                                     | ...         |               |
'------'------'-----'
```

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

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

```
trident list --packages --remote
```

will result in a view of all published packages in our public online repository.

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

```
trident list -d ... -d ... --groups
```

```
trident list -d ... -d ... --individuals
```

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

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

### 1.4.2 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 (could be a Poseidon repository)
<code>--raw</code>	output table as tsv without header. Useful for piping into <code>grep</code> or <code>awk</code>

You can run it with

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

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

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

### 1.4.3 Survey command

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

[Click here for command line details](#)

527 Usage: trident survey (-d|--baseDir DIR) [--raw]  
 528 Survey the degree of context information completeness for Poseidon packages  
 529  
 530 Available options:  
 531 -h,--help Show this help text  
 532 -d,--baseDir DIR a base directory to search for Poseidon Packages  
 533 (could be a Poseidon repository)  
 534 --raw output table as tsv without header. Useful for piping  
 535 into grep or awk

536 Running

537 trident survey -d ... -d ...

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

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

541 **1.4.4 Validate command**

542 validate checks poseidon datasets for structural correctness.

543 Click here for command line details

544 Usage: trident validate (-d|--baseDir DIR) [--verbose]  
 545 Check one or multiple Poseidon packages for structural correctness  
 546

547 Available options:  
 548 -h,--help Show this help text  
 549 -d,--baseDir DIR a base directory to search for Poseidon Packages  
 550 (could be a Poseidon repository)  
 551 --verbose print more output to the command line  
 552 --ignoreGeno ignore SNP and GenoFile  
 553 --noExitCode do not produce an explicit exit code

554 You can run it with

555 trident validate -d ... -d ...

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

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

- 560 • Presence of the necessary files
- 561 • Full structural correctness of .bib and .janno file
- 562 • Superficial correctness of genotype data files. A full check would be too computationally expensive
- 563 • Correspondence of BibTeX keys in .bib and .janno
- 564 • Correspondence of individual and group IDs in .janno and genotype data files

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