

# Contents

1	<b>1 Guide for trident v1.0.0.0</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 . . . . .	14
14	1.4.4 Validate command . . . . .	14
15		

## 1 Guide for trident v1.0.0.0

### 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]
126                 (--downloadAll |
127                 (--fetchFile ARG | (-f|--fetchString ARG)))
128                 [--remoteURL ARG] [-u|--upgrade]
129 Download data from a remote Poseidon repository
```

131 Available options:

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	overwrite outdated local package versions

151 It works with

```
152 trident fetch -d ... -d ... \  
153 -f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<Individual1>" \  
154 --fetchFile path/to/forgeFile
```

155 and the entities you want to download must be listed either in one or more simple strings with comma-separated  
156 values, which can be passed via one or multiple options `-f/--fetchString`, or in one or more text files  
157 (`--fetchFile`). Entities are then combined from these sources. Entities are specified using a special syntax:  
158 Package titles are wrapped in asterisks: `package_title` (see also the documentation of `forge` below), group names  
159 are spelled as is, and individual names are wrapped in angular brackets, like `<Individual1>`. Fetch will figure  
160 out which packages need to be downloaded to include all specified entities. `--downloadAll`, which can be given

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

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

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

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

### 170 1.3.3 Forge command

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

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

```
174 Usage: trident forge [-d|--baseDir DIR]
175                [
176                ((-p|--genoOne ARG) | (-r|--inFormat ARG)
177                (-g|--genoFile ARG) (-s|--snpFile ARG)
178                (-i|--indFile ARG)) [--snpSet ARG]]
179                [--forgeFile ARG | (-f|--forgeString ARG)]
180                [--selectSnps ARG] [--intersect] [--outFormat ARG]
181                [--minimal] [--onlyGeno] (-o|--outPackagePath ARG)
182                [-n|--outPackageName ARG] [--no-extract]
```

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

186 Available options:

187 <code>-h,--help</code>	Show this help text
188 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
190 <code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects .bed or .bim or .fam for PLINK and .geno or .snp or .ind for EIGENSTRAT. The other files must be in the same directory and must have the same base name
194 <code>-r,--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or PLINK
196 <code>-g,--genoFile ARG</code>	the input geno file path
197 <code>-s,--snpFile ARG</code>	the input snp file path
198 <code>-i,--indFile ARG</code>	the input ind file path
199 <code>--snpSet ARG</code>	the snpSet of the new package: 1240K, HumanOrigins or Other. Default: Other
201 <code>--forgeFile ARG</code>	A file with a list of packages, groups or individual samples. Works just as <code>-f</code> , but multiple values can

also be separated by newline, not just by comma.  
Empty lines are ignored and comments start with "#",  
so everything after "#" is ignored in one line.  
Multiple instances of -f and --forgeFile can be  
given. They will be evaluated according to their  
input order on the command line.

-f,--forgeString ARG List of packages, groups or individual samples to be  
combined in the output package. Packages follow the  
syntax \*package\_title\*, populations/groups are simply  
group\_id and individuals <individual\_id>. You can  
combine multiple values with comma, so for example:  
"\*package\_1\*, <individual\_1>, <individual\_2>,  
group\_1". Duplicates are treated as one entry.  
Negative selection is possible by prepending "-" to  
the entity you want to exclude (e.g. "\*package\_1\*,  
-<individual\_1>, -group\_1"). forge will apply  
excludes and includes in order. If the first entity  
is negative, then forge will assume you want to merge  
all individuals in the packages found in the baseDirs  
(except the ones explicitly excluded) before the  
exclude entities are applied. An empty forgeString  
(and no --forgeFile) will therefore merge all  
available individuals.

--selectSnps ARG To extract specific SNPs during this forge operation,  
provide a Snp file. Can be either Eigenstrat (file  
ending must be '.snp') or Plink (file ending must be  
'.bim'). 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  
option '--intersect' is also set, only the SNPs  
overlapping between the SNP file and the forged  
packages are output.

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

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

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

--onlyGeno should only the resulting genotype data be returned?

This means the output will not be a Poseidon package

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

248 `-n,--outPackageName ARG` the output package name - this is optional: If no  
 249 name is provided, then the package name defaults to  
 250 the basename of the (mandatory) `--outPackagePath`  
 251 argument  
 252 `--no-extract` Skip the selection step in forge. This will result in  
 253 outputting all individuals in the relevant packages,  
 254 and hence a superset of the requested  
 255 individuals/groups. It may result in better  
 256 performance in cases where one wants to forge entire  
 257 packages or almost entire packages. Note that this  
 258 will also ignore any ordering in the output  
 259 groups/individuals. With this option active,  
 260 individuals from the relevant packages will just be  
 261 written in the order that they appear in the original  
 262 packages.

263 `forge` can be used with

```
264 trident forge -d ... -d ... \  

265   -f "*package_name*, group_id, <individual_id>" \  

266   --forgeFile path/to/forgeFile \  

267   -o path/to/new_package_name
```

268 where the entities (packages, groups/populations, individuals/samples) you want in the output package can be  
 269 denoted either as one or more simple strings with comma-separated values via one or more (`-f/--forgeString`)  
 270 options, or in one or more text files (`--forgeFile`). Because the order in which inclusions and exclusions  
 271 are given, the order strictly follows the order as these strings are given via options `-f/--forgeString` and  
 272 `--forgeFile`.

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

```
277 trident forge \  

278   -d 2017_GonzalesFortesCurrentBiology \  

279   -r PLINK \  

280   -g 2017_HaberAJHG/2017_HaberAJHG.bed \  

281   -s 2017_HaberAJHG/2017_HaberAJHG.bim \  

282   -i 2017_HaberAJHG/2017_HaberAJHG.fam \  

283   -r PLINK \  

284   -g 2018_VeeramahPNAS/2018_VeeramahPNAS.bed \  

285   -i 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \  

286   -s 2018_VeeramahPNAS/2018_VeeramahPNAS.bim \  

287   -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \  

288   -o testpackage \  

289   --onlyGeno
```

290 **1.3.3.1 The forge selection language** Entities in the `--forgeString` or the `--forgeFile` have to be  
291 marked in a certain way:

- 292 • Each package is surrounded by `*`, so if you want all individuals of `2019_Jeong_InnerEurasia` in the  
293 output package you would add `*2019_Jeong_InnerEurasia*` to the list.
- 294 • Groups/populations are not specially marked. So to get all individuals of the group `Swiss_Roman_period`,  
295 you would simply add `Swiss_Roman_period`.
- 296 • Individuals/samples are surrounded by `<` and `>`, so `ALA026` becomes `<ALA026>`.

297 Do not forget to wrap the `forgeString` in quotes.

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

```
300 # Packages
301 *package1*, *package2*
302
303 # Groups and individuals from other packages beyond package1 and package2
304 group1, <individual1>, group2, <individual2>, <individual3>
305
306 # group2 has two outlier individuals that should be ignored
307 -<bad_individual1> # This one has very low coverage
308 -<bad_individual2> # This one is from a different time period
```

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

316 **1.3.3.2 Other options** Just as for `init` the output package of `forge` is created as a new directory `-o`. The  
317 title can also be explicitly defined with `-n`.

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

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

326 `--intersect` also influences the automatic determination of the `snpSet` field in the `POSEIDON.yml` file for the  
327 resulting package. If the `snpSets` of all input packages are identical, then the resulting package will just inherit  
328 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

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

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

### 335 1.3.4 Genoconvert command

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

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

339 Usage: trident genoconvert [-d|--baseDir DIR]  
340 [  
341 ((-p|--genoOne ARG) | (-r|--inFormat ARG)  
342 (-g|--genoFile ARG) (-s|--snpFile ARG)  
343 (-i|--indFile ARG)) [--snpSet ARG]]  
344 --outFormat ARG [--onlyGeno]  
345 [-o|--outPackagePath ARG] [--removeOld]  
346 Convert the genotype data in a Poseidon package to a different file format

347 Available options:

348 `-h,--help` Show this help text  
349 `-d,--baseDir DIR` a base directory to search for Poseidon Packages  
350 (could be a Poseidon repository)  
351 `-p,--genoOne ARG` one of the input genotype data files. Expects `.bed` or  
352 `.bim` or `.fam` for PLINK and `.geno` or `.snp` or `.ind` for  
353 EIGENSTRAT. The other files must be in the same  
354 directory and must have the same base name  
355 `-r,--inFormat ARG` the format of the input genotype data: EIGENSTRAT or  
356 PLINK  
357 `-g,--genoFile ARG` the input geno file path  
358 `-s,--snpFile ARG` the input snp file path  
359 `-i,--indFile ARG` the input ind file path  
360 `--snpSet ARG` the snpSet of the new package: 1240K, HumanOrigins or  
361 Other. Default: Other  
362 `--outFormat ARG` the format of the output genotype data: EIGENSTRAT or  
363 PLINK.  
364 `--onlyGeno` should only the resulting genotype data be returned?

```

366                                     This means the output will not be a Poseidon package
367  -o,--outPackagePath ARG the output package directory path - this is optional:
368                                     If no path is provided, then the output is written to
369                                     the directories where the input genotype data file
370                                     (.bed/.geno) is stored
371  --removeOld Remove the old genotype files when creating the new
372                                     ones
373
374 With the default setting
375
376 trident genoconvert -d ... -d ... --outFormat EIGENSTRAT|PLINK
377
378 all packages in -d will be converted to the desired --outFormat (either EIGENSTRAT or PLINK), if the data is
379 not already in this format. This includes updating the respective POSEIDON.yml files.
380
381 The “old” data is not deleted, but kept around. That means conversion can result in a package with both PLINK
382 and EIGENSTRAT data, but only one is linked in the POSEIDON.yml file, and that is what will be used by
383 trident. To delete the old data in the conversion you can add the --removeOld flag.
384
385 Instead of -d to change Poseidon packages, the combination -r + -g + -s + -i (+ --snpSet) or -p (+
386 --snpSet) allows to directly convert genotype data that is not wrapped in a Poseidon package and store it to a
387 directory given in -o. See this example:
388
389 trident genoconvert \
390   -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
391   --outFormat EIGENSTRAT
392   -o my_directory

```

### 387 1.3.5 Update command

388 **update** automatically updates POSEIDON.yml files of one or multiple packages if the packages were changed.  
 389 [Click here for command line details](#)

```

390 Usage: trident update (-d|--baseDir DIR) [--poseidonVersion ARG]
391               [--ignorePoseidonVersion] [--versionComponent ARG]
392               [--noChecksumUpdate] [--newContributors ARG]
393               [--logText ARG] [--force]
394   Update POSEIDON.yml files automatically
395

```

#### 396 Available options:

```

397  -h,--help Show this help text
398  -d,--baseDir DIR a base directory to search for Poseidon Packages
399                  (could be a Poseidon repository)
400  --poseidonVersion ARG Poseidon version the packages should be updated to:
401                        e.g. "2.5.3" (default: Nothing)
402  --ignorePoseidonVersion Read packages even if their poseidonVersion is not
403                          compatible with the trident version. The assumption
404                          is, that the package is already structurally adjusted
405                          to the trident version and only the version number is
406                          lagging behind.

```

```

407 --versionComponent ARG    Part of the package version number in the
408                             POSEIDON.yml file that should be updated: Major,
409                             Minor or Patch (see
410                             https://semver.org) (default: Patch)
411 --noChecksumUpdate         Should update of checksums in the POSEIDON.yml file
412                             be skipped
413 --ignoreGeno               ignore SNP and GenoFile
414 --newContributors ARG      Contributors to add to the POSEIDON.yml file in the
415                             form "[Firstname Lastname](Email address);..."
416 --logText ARG              Log text for this version jump in the CHANGELOG
417                             file (default: "not specified")
418 --force                    Normally the POSEIDON.yml files are only changed if
419                             the poseidonVersion is adjusted or any of the
420                             checksums change. With --force a package version
421                             update can be triggered even if this is not the case.

```

422 It can be called with a lot of optional arguments

```

423 trident update -d ... -d ... \
424   --poseidonVersion "X.X.X" \
425   --versionComponent Major/Minor/Patch \
426   --noChecksumUpdate
427   --ignoreGeno
428   --newContributors "[Firstname Lastname](Email address);..."
429   --logText "short description of the update"
430   --force

```

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

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

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

- 441 • If --poseidonVersion is different from the poseidonVersion field in the package, then that will be  
442 updated.
- 443 • The packageVersion will be incremented. If --versionComponent is not set, then it falls back to Patch,  
444 so a change in the last position of the three digit version number. Minor increments the middle, and Major  
445 the first position (see [semantic versioning](#)).
- 446 • The lastModified field will be updated to the current day (based on your computer's system time).
- 447 • The contributors in --newContributors will be added to the contributor field if they're not there already.
- 448 • If any checksums changed, then they will be updated. If certain checksums are not set yet, then they will  
449 be added. The checksum update can be skipped with --noChecksumUpdate or partially skipped for the

450 genotype data with `--ignoreGeno`.

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

454 :heavy\_exclamation\_mark: As `update` 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 package definition](#)). Create a backup of the POSEIDON.yml file before running `update` if you are uncertain.

## 457 1.4 Inspection commands

### 458 1.4.1 List command

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

460 Click here for command line details

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

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

467 Available options:

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

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

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

488 This will yield a table like this

```
489 .----- .----- .-----
490 |           Title           |      Date      | Nr Individuals |
```

```

491 :=====;=====;=====;
492 | 2015_1000Genomes_1240K_haploid_pulldown | 2020-08-10 | 2535 |
493 | 2016_Mallick_SGDP1240K_diploid_pulldown | 2020-08-10 | 280 |
494 | 2018_BostonDatashare_modern_published | 2020-08-10 | 2772 |
495 | ... | ... | |
496 '-----'-----'-----'

```

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:

```

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

```

You can run it with

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

which will show you context information like – among others – the number of individuals in the dataset, their sex distribution, the mean age of the samples (for ancient data) or the mean coverage on the 1240K SNP array

529 in a table. **summarise** depends on complete .janno files and will silently ignore missing information for some  
530 statistics.

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

### 532 1.4.3 Survey command

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

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

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

536 Survey the degree of context information completeness for Poseidon packages

537

538 Available options:

539 -h,--help	Show this help text
540 -d,--baseDir DIR	a base directory to search for Poseidon Packages 541 (could be a Poseidon repository)
542 --raw	output table as tsv without header. Useful for piping 543 into grep or awk

544 Running

545 **trident survey -d ... -d ...**

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

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

### 549 1.4.4 Validate command

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

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

552 Usage: **trident validate** (-d|--baseDir DIR) [--verbose]

553 Check one or multiple Poseidon packages for structural correctness

554

555 Available options:

556 -h,--help	Show this help text
557 -d,--baseDir DIR	a base directory to search for Poseidon Packages 558 (could be a Poseidon repository)
559 --verbose	print more output to the command line
560 --ignoreGeno	ignore SNP and GenoFile
561 --noExitCode	do not produce an explicit exit code

562 You can run it with

563 **trident validate -d ... -d ...**

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

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

- 568 • Presence of the necessary files
- 569 • Full structural correctness of .bib and .janno file
- 570 • Superficial correctness of genotype data files. A full check would be too computationally expensive
- 571 • Correspondence of BibTeX keys in .bib and .janno
- 572 • Correspondence of individual and group IDs in .janno and genotype data files

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