

Guide for trident v1.0.0.0

Contents

1	Poseidon package repositories	1
2	Analysing your own dataset outside of the main repository	2
3	Package creation and manipulation commands	3
3.1	Init command	3
3.2	Fetch command	4
3.3	Forge command	5
3.3.1	The forge selection language	8
3.3.2	Other options	9
3.4	Genoconvert command	9
3.5	Update command	10
4	Inspection commands	12
4.1	List command	12
4.2	Summarise command	14
4.3	Survey command	14
4.4	Validate command	15

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
  /...
```

```

33     /...
34 /Reference_Genomes
35     /...
36     /...
37 /Archaic_Humans
38     /...
39     /...

```

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

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

```

43 trident list -d /path/to/poseidon/packages/modern \
44 -d /path/to/poseidon/packages/ReferenceGenomes --packages

```

45 2 Analysing your own dataset outside of the main repository

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

```

50 ~/my_project/my_project.geno
51 ~/my_project/my_project.snp
52 ~/my_project/my_project.ind

```

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

```

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57 description: Unpublished genetic data from my awesome project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

```

70 Two remarks: 1) all file paths are considered *relative* to the directory in which `POSEIDON.yml` resides. Here I
 71 assume that you put this file into the same directory as the three genotype files. 2) Besides the genotype data
 72 files there are two (technically optional) files referenced by this example `POSEIDON.yml` file: `sources.bib` and

73 `my_project.janno`. Of course you can add them manually - `init` automatically creates empty dummy versions.

74 Once you have set up your own “Poseidon” package (which is really only a skeleton so far), you can add it to

75 your `trident` analysis, by simply adding your project directory to the command using `-d`:

```
76 trident list -d /path/to/poseidon/packages/modern \
77     -d /path/to/poseidon/packages/ReferenceGenomes
78     -d ~/my_project --packages
```

79 3 Package creation and manipulation commands

80 3.1 Init command

81 `init` creates a new, valid poseidon package from genotype data files. It adds a valid `POSEIDON.yml` file, a dummy

82 `.janno` file for context information and an empty `.bib` file for literature references.

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

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

88 Create a new Poseidon package from genotype data

89 Available options:

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

109 The command

```
110 trident init \
111     -r EIGENSTRAT/PLINK \
112     -g path/to/geno_file \
```

```

113 -s path/to/snp_file \
114 -i path/to/ind_file \
115 --snpSet 1240K|HumanOrigins|Other \
116 -o path/to/new_package_name

```

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

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

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

124 3.2 Fetch command

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

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

```

127 Usage: trident fetch [-d|--baseDir DIR]
128                 (--downloadAll |
129                 (--fetchFile ARG | (-f|--fetchString ARG)))
130                 [--remoteURL ARG] [-u|--upgrade]

```

131 Download data from a remote Poseidon repository

132 Available options:

134 <code>-h,--help</code>	Show this help text
135 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
136 <code>--downloadAll</code>	download all packages the server is offering
137 <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.
138 <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: "*package_1*, *package_2*, *package_3*". <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

```

149                                     the download.
150  --remoteURL ARG                     URL of the remote Poseidon server
151                                     (default: "https://c107-224.cloud.gwdg.de")
152  -u,--upgrade                        overwrite outdated local package versions

```

153 It works with

```

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

```

157 and the entities you want to download must be listed either in one or more simple strings with comma-separated values, which can be passed via one or multiple options `-f/--fetchString`, or in one or more text files (`--fetchFile`). Entities are then combined from these sources. Entities are specified using a special syntax: Package titles are wrapped in asterisks: *package_title* (see also the documentation of `forge` below), group names are spelled as is, and individual names are wrapped in angular brackets, like `<Individual1>`. 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 an up-to-date version in any of the `-d` dirs.

156 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.

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

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

172 3.3 Forge command

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

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

```

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

```

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

187 Available options:

```

188  -h,--help                        Show this help text

```

190	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages
191		(could be a Poseidon repository)
192	<code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects .bed or
193		.bim or .fam for PLINK and .geno or .snp or .ind for
194		EIGENSTRAT. The other files must be in the same
195		directory and must have the same base name
196	<code>-r,--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or
197		PLINK
198	<code>-g,--genoFile ARG</code>	the input geno file path
199	<code>-s,--snpFile ARG</code>	the input snp file path
200	<code>-i,--indFile ARG</code>	the input ind file path
201	<code>--snpSet ARG</code>	the snpSet of the new package: 1240K, HumanOrigins or
202		Other. Default: Other
203	<code>--forgeFile ARG</code>	A file with a list of packages, groups or individual
204		samples. Works just as -f, but multiple values can
205		also be separated by newline, not just by comma.
206		Empty lines are ignored and comments start with "#",
207		so everything after "#" is ignored in one line.
208		Multiple instances of -f and --forgeFile can be
209		given. They will be evaluated according to their
210		input order on the command line.
211	<code>-f,--forgeString ARG</code>	List of packages, groups or individual samples to be
212		combined in the output package. Packages follow the
213		syntax *package_title*, populations/groups are simply
214		group_id and individuals <individual_id>. You can
215		combine multiple values with comma, so for example:
216		"*package_1*, <individual_1>, <individual_2>,"
217		group_1". Duplicates are treated as one entry.
218		Negative selection is possible by prepending "-" to
219		the entity you want to exclude (e.g. "*package_1*,"
220		-<individual_1>, -group_1"). forge will apply
221		excludes and includes in order. If the first entity
222		is negative, then forge will assume you want to merge
223		all individuals in the packages found in the baseDirs
224		(except the ones explicitly excluded) before the
225		exclude entities are applied. An empty forgeString
226		(and no --forgeFile) will therefore merge all
227		available individuals.
228	<code>--selectSnps ARG</code>	To extract specific SNPs during this forge operation,
229		provide a Snp file. Can be either Eigenstrat (file
230		ending must be '.snp') or Plink (file ending must be
231		'.bim'). When this option is set, the output package
232		will have exactly the SNPs listed in this file. Any
233		SNP not listed in the file will be excluded. If
234		option '--intersect' is also set, only the SNPs

235 overlapping between the SNP file and the forged
236 packages are output.

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

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

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

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

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

250 `-n,--outPackageName ARG` the output package name - this is optional: If no
251 name is provided, then the package name defaults to
252 the basename of the (mandatory) `--outPackagePath`
253 argument

254 `--no-extract` Skip the selection step in forge. This will result in
255 outputting all individuals in the relevant packages,
256 and hence a superset of the requested
257 individuals/groups. It may result in better
258 performance in cases where one wants to forge entire
259 packages or almost entire packages. Note that this
260 will also ignore any ordering in the output
261 groups/individuals. With this option active,
262 individuals from the relevant packages will just be
263 written in the order that they appear in the original
264 packages.

265 `forge` can be used with

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

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

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

```

279 trident forge \
280   -d 2017_GonzalesFortesCurrentBiology \
281   -r PLINK \
282   -g 2017_HaberAJHG/2017_HaberAJHG.bed \
283   -s 2017_HaberAJHG/2017_HaberAJHG.bim \
284   -i 2017_HaberAJHG/2017_HaberAJHG.fam \
285   -r PLINK \
286   -g 2018_VeeramahPNAS/2018_VeeramahPNAS.bed \
287   -i 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \
288   -s 2018_VeeramahPNAS/2018_VeeramahPNAS.bim \
289   -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \
290   -o testpackage \
291   --onlyGeno

```

292 3.3.1 The forge selection language

293 Entities in the `--forgeString` or the `--forgeFile` have to be marked in a certain way:

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

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

300 You can either use `-f/--forgeString` or `--forgeFile`. In the file each line is treated as a separate `forgeString`,

301 empty lines are ignored and `#`s start comments. So this is a valid `forgeFile`:

```

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

```

311 By prepending `-` to the bad individuals, we can exclude them from the forged package. `forge` fig-

312 ures out the final list of samples to include by executing all `forge`-entities in order. So an entity list

313 `*PackageA*, -<Individual1>, GroupA` may result in a different outcome than `*PackageA*, GroupA, -<Individual1>`,

314 depending on whether `<Individual1>` belongs to `GroupA` or not. If the `forge` entity list starts with a negative

315 entity, or if the entity list is empty, `forge` will implicitly assume you want to include all individuals in all

316 packages found in the `baseDirs` (except the ones explicitly excluded, of course). An empty `forgeString` will

317 therefore merge all available individuals.

3.3.2 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 might be 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 `-w/--warnings` flag is set.

3.4 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 ARG) | (-r|--inFormat ARG)
    (-g|--genoFile ARG) (-s|--snpFile ARG)
    (-i|--indFile ARG)) [--snpSet ARG]]
--outFormat ARG [--onlyGeno]
[-o|--outPackagePath ARG] [--removeOld]

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

Available options:
```

```

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

```

376 With the default setting

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

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

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

```

386 trident genoconvert \
387   -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
388   --outFormat EIGENSTRAT
389   -o my_directory

```

390 3.5 Update command

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

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

```

393 Usage: trident update (-d|--baseDir DIR) [--poseidonVersion ARG]
394             [--ignorePoseidonVersion] [--versionComponent ARG]
395             [--noChecksumUpdate] [--newContributors ARG]
396             [--logText ARG] [--force]
397 Update POSEIDON.yml files automatically
398
399 Available options:
400 -h,--help          Show this help text
401 -d,--baseDir DIR   a base directory to search for Poseidon Packages
402                   (could be a Poseidon repository)
403 --poseidonVersion ARG Poseidon version the packages should be updated to:
404                   e.g. "2.5.3" (default: Nothing)
405 --ignorePoseidonVersion Read packages even if their poseidonVersion is not
406                   compatible with the trident version. The assumption
407                   is, that the package is already structurally adjusted
408                   to the trident version and only the version number is
409                   lagging behind.
410 --versionComponent ARG Part of the package version number in the
411                   POSEIDON.yml file that should be updated: Major,
412                   Minor or Patch (see
413                   https://semver.org) (default: Patch)
414 --noChecksumUpdate   Should update of checksums in the POSEIDON.yml file
415                   be skipped
416 --ignoreGeno         ignore SNP and GenoFile
417 --newContributors ARG Contributors to add to the POSEIDON.yml file in the
418                   form "[Firstname Lastname](Email address);..."
419 --logText ARG        Log text for this version jump in the CHANGELOG
420                   file (default: "not specified")
421 --force              Normally the POSEIDON.yml files are only changed if
422                   the poseidonVersion is adjusted or any of the
423                   checksums change. With --force a package version
424                   update can be triggered even if this is not the case.
425
426 It can be called with a lot of optional arguments
427
428 trident update -d ... -d ... \
429     --poseidonVersion "X.X.X" \
430     --versionComponent Major/Minor/Patch \
431     --noChecksumUpdate
432     --ignoreGeno
433     --newContributors "[Firstname Lastname](Email address);..."
434     --logText "short description of the update"
435     --force
436
437 By default update will not edit a package's POSEIDON.yml file, even when arguments like --versionComponent,
438 --newContributors or --logText are explicitly set. This default exists to run the function on a large set of
439 packages where only few of them were edited and need an active update. A package will only be modified by

```

437 **update** if either

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

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

- 444 • If `--poseidonVersion` is different from the `poseidonVersion` field in the package, then that will be updated.
- 445 • The `packageVersion` will be incremented. If `--versionComponent` is not set, then it falls back to `Patch`, so a change in the last position of the three digit version number. `Minor` increments the middle, and `Major` 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 be added. The checksum update can be skipped with `--noChecksumUpdate` or partially skipped for the genotype data with `--ignoreGeno`.
- 449 • 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.

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

460 4 Inspection commands

461 4.1 List command

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

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

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

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

469 Available options:

471 -h,--help	Show this help text
472 -d,--baseDir DIR	a base directory to search for Poseidon Packages 473 (could be a Poseidon repository)
474 --remote	list packages from a remote server instead the local 475 file system
476 --remoteURL ARG	URL of the remote Poseidon

```

477         server (default: "https://c107-224.cloud.gwdg.de")
478 --packages      list all packages
479 --groups        list all groups, ignoring any group names after the
480                  first as specified in the Janno-file
481 --individuals    list individuals
482 -j,--jannoColumn JANNO_HEADER
483                  list additional fields from the janno files, using
484                  the Janno column heading name, such as Country, Site,
485                  Date_C14_Uncal_BP, Endogenous, ...
486 --raw           output table as tsv without header. Useful for piping
487                  into grep or awk
488 --ignoreGeno     ignore SNP and GenoFile

```

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

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

491 This will yield a table like this

```

492 .------.------.------.
493 |           Title           |   Date   | Nr Individuals |
494 :=====:=====:=====:
495 | 2015_1000Genomes_1240K_haploid_pull | 2020-08-10 | 2535          |
496 | 2016_Mallick_SGDP1240K_diploid_pull | 2020-08-10 | 280           |
497 | 2018_BostonDatashare_modern_published | 2020-08-10 | 2772          |
498 | ...                         | ...       |               |
499 '-----'-----'-----'

```

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

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

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

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

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

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

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

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

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

516 4.2 Summarise command

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

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

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

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

521

522 Available options:

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

528 You can run it with

529 trident summarise -d ... -d ...

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

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

535 4.3 Survey command

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

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

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

539 Survey the degree of context information completeness for Poseidon packages

540

541 Available options:

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

547 Running

548 trident survey -d ... -d ...

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

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

552 4.4 Validate command

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

554 Click here for command line details

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

556 Check one or multiple Poseidon packages for structural correctness

557

558 Available options:

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

565 You can run it with

566 **trident validate -d ... -d ...**

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

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

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

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