

Contents

1	Guide for trident v0.28.0	1
1.1	Poseidon package repositories	1
1.2	Analysing your own dataset outside of the main repository	2
1.3	Package creation and manipulation commands	2
1.3.1	Init command	2
1.3.2	Fetch command	4
1.3.3	Forge command	5
1.3.4	Genoconvert command	8
1.3.5	Update command	9
1.4	Inspection commands	11
1.4.1	List command	11
1.4.2	Summarise command	13
1.4.3	Survey command	13
1.4.4	Validate command	14
1.5	Analysis commands	14

1 Guide for trident v0.28.0

1.1 Poseidon package repositories

Trident generally requires Poseidon datasets to work with. Most trident subcommands therefore have a central parameter, called `--baseDir` or simply `-d` to specify one or more base directories to look for Poseidon 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 (-r|--inFormat ARG) (-g|--genoFile ARG) (-s|--snpFile ARG)
83             (-i|--indFile ARG) [--snpSet ARG] (-o|--outPackagePath ARG)
84             [-n|--outPackageName ARG] [--minimal]
```

85 Create a new Poseidon package from genotype data

86 Available options:

```
87 -h,--help          Show this help text
88 -r,--inFormat ARG  the format of the input genotype data: EIGENSTRAT or
89                   PLINK
90 -g,--genoFile ARG  the input geno file path
91 -s,--snpFile ARG   the input snp file path
92 -i,--indFile ARG   the input ind file path
93 --snpSet ARG       the snpSet of the new package: 1240K, HumanOrigins or
94                   Other. Default: Other
95 -o,--outPackagePath ARG the output package directory path
96 -n,--outPackageName ARG the output package name - this is optional: If no
97                   name is provided, then the package name defaults to
98                   the basename of the (mandatory) --outPackagePath
99                   argument
100 --minimal          should only a minimal output package be created?
```

102 The command

```
103 trident init \
104   -r EIGENSTRAT/PLINK \
105   -g path/to/geno_file \
106   -s path/to/snp_file \
107   -i path/to/ind_file \
108   --snpSet 1240K|HumanOrigins|Other \
109   -o path/to/new_package_name
```

110 requires the format `-r` (`--inFormat`) of your input data (either `EIGENSTRAT` or `PLINK`), the paths to the
111 respective files in `-g` (`--genoFile`), `-s` (`--snpFile`), and `-i` (`--indFile`), and optionally the “shape” of these
112 files (`--snpSet`), so if they cover the 1240K, the `HumanOrigins` or an `Other` SNP set.

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

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

116 1.3.2 Fetch command

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

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

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

121 Download data from a remote Poseidon repository

122 Available options:

124	<code>-h,--help</code>	Show this help text
125	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
127	<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 the download.
136	<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.
139	<code>--remoteURL ARG</code>	URL of the remote Poseidon server (default: "https://c107-224.cloud.gwdg.de")
141	<code>-u,--upgrade</code>	overwrite outdated local package versions
142	<code>--downloadAll</code>	download all packages the server is offering

143 It works with

```
144 trident fetch -d ... -d ... \  
145     -f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<Individual1>" \  
146     --fetchFile path/to/forgeFile
```

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

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

157 `fetch` also has the optional arguments `--remote https://...` to name an alternative poseidon server. The

158 default points to the [DAG server](#).

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

161 1.3.3 Forge command

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

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

```
165 Usage: trident forge [-d|--baseDir DIR]
166                   [(-r|--inFormat ARG) (-g|--genoFile ARG) (-s|--snpFile ARG)
167                   (-i|--indFile ARG) [--snpSet ARG]]
168                   [--forgeFile ARG | (-f|--forgeString ARG)]
169                   [--selectSnps ARG] [--intersect] [--outFormat ARG]
170                   [--minimal] [--onlyGeno] (-o|--outPackagePath ARG)
171                   [-n|--outPackageName ARG] [-w|--warnings] [--no-extract]
```

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

174
175 Available options:

176	<code>-h,--help</code>	Show this help text
177	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
178		
179	<code>-r,--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or 180 PLINK
181	<code>-g,--genoFile ARG</code>	the input geno file path
182	<code>-s,--snpFile ARG</code>	the input snp file path
183	<code>-i,--indFile ARG</code>	the input ind file path
184	<code>--snpSet ARG</code>	the snpSet of the new package: 1240K, HumanOrigins or 185 Other. Default: Other
186	<code>--forgeFile ARG</code>	A file with a list of packages, groups or individual 187 samples. Works just as <code>-f</code> , but multiple values can 188 also be separated by newline, not just by comma. 189 Empty lines are ignored and comments start with "#", 190 so everything after "#" is ignored in one line.
191	<code>-f,--forgeString ARG</code>	List of packages, groups or individual samples to be 192 combined in the output package. Packages follow the 193 syntax <code>*package_title*</code> , populations/groups are simply 194 <code>group_id</code> and individuals <code><individual_id></code> . You can 195 combine multiple values with comma, so for example: 196 <code>"*package_1*, <individual_1>, <individual_2>,"</code> 197 <code>group_1"</code> . Duplicates are treated as one entry. 198 Negative selection is possible by prepending "-" to 199 the entity you want to exclude (e.g. <code>"*package_1*,</code> 200 <code>-<individual_1>, -group_1"</code>). <code>forge</code> will apply

201 excludes and includes in order. If the first entity
 202 is negative, then forge will assume you want to merge
 203 all individuals in the packages found in the baseDirs
 204 (except the ones explicitly excluded) before the
 205 exclude entities are applied. An empty forgeString
 206 will therefore merge all available individuals.

207 `--selectSnps ARG` To extract specific SNPs during this forge operation,
 208 provide a Snp file. Can be either Eigenstrat (file
 209 ending must be '.snp') or Plink (file ending must be
 210 '.bim'). When this option is set, the output package
 211 will have exactly the SNPs listed in this file. Any
 212 SNP not listed in the file will be excluded. If
 213 option '--intersect' is also set, only the SNPs
 214 overlapping between the SNP file and the forged
 215 packages are output.

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

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

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

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

227 This means the output will not be a Poseidon package

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

229 `-n,--outPackageName ARG` the output package name - this is optional: If no
 230 name is provided, then the package name defaults to
 231 the basename of the (mandatory) --outPackagePath
 232 argument

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

234 `--no-extract` Skip the selection step in forge. This will result in
 235 outputting all individuals in the relevant packages,
 236 and hence a superset of the requested
 237 individuals/groups. It may result in better
 238 performance in cases where one wants to forge entire
 239 packages or almost entire packages. Note that this
 240 will also ignore any ordering in the output
 241 groups/individuals. With this option active,
 242 individuals from the relevant packages will just be
 243 written in the order that they appear in the original
 244 packages.

245 `forge` can be used with

```

246 trident forge -d ... -d ... \
247   -f "*package_name*, group_id, <individual_id>" \
248   --forgeFile path/to/forgeFile \
249   -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)`. This makes the following example possible, where we merge data from one Poseidon package and two genotype datasets.

```

257 trident forge \
258   -d 2017_GonzalesFortesCurrentBiology \
259   -r PLINK \
260   -g 2017_HaberAJHG/2017_HaberAJHG.bed \
261   -s 2017_HaberAJHG/2017_HaberAJHG.bim \
262   -i 2017_HaberAJHG/2017_HaberAJHG.fam \
263   -r PLINK \
264   -g 2018_VeeramahPNAS/2018_VeeramahPNAS.bed \
265   -i 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \
266   -s 2018_VeeramahPNAS/2018_VeeramahPNAS.bim \
267   -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \
268   -o testpackage \
269   --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.

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

```

280 # Packages
281 *package1*, *package2*
282
283 # Groups and individuals from other packages beyond package1 and package2
284 group1, <individual1>, group2, <individual2>, <individual3>
285
286 # group2 has two outlier individuals that should be ignored
287 -<bad_individual1> # This one has very low coverage

```

288 `-<bad_individual2> # This one is from a different time period`

289 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
 290 `*PackageA*, -<Individual1>, GroupA` may result in a different outcome than `*PackageA*, GroupA, -<Individual1>`,
 291 depending on whether `<Individual1>` belongs to `GroupA` or not. If the `forge` entity list starts with a negative
 292 entity, or if the entity list is empty, `forge` will implicitly assume you want to include all individuals in all
 293 packages found in the `baseDirs` (except the ones explicitly excluded, of course). An empty `forgeString` will
 294 therefore merge all available individuals.

296 **1.3.3.2 Other options** Just as for `init` the output package of `forge` is created as a new directory `-o`. The
 297 title can also be explicitly defined with `-n`.

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

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

306 `--intersect` also influences the automatic determination of the `snpSet` field in the `POSEIDON.yml` file for the
 307 resulting package. If the `snpSets` of all input packages are identical, then the resulting package will just inherit
 308 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

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

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

315 1.3.4 Genoconvert command

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

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

319 Usage: `trident genoconvert [-d|--baseDir DIR]`
 320 `[(-r|--inFormat ARG) (-g|--genoFile ARG)`
 321 `(-s|--snpFile ARG) (-i|--indFile ARG)`


```

322         [--snpSet ARG]] --outFormat ARG [--onlyGeno]
323         [--removeOld]

```

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

325
326 Available options:

```

327     -h,--help                Show this help text
328     -d,--baseDir DIR         a base directory to search for Poseidon Packages
329                               (could be a Poseidon repository)
330     -r,--inFormat ARG        the format of the input genotype data: EIGENSTRAT or
331                               PLINK
332     -g,--genoFile ARG        the input geno file path
333     -s,--snpFile ARG         the input snp file path
334     -i,--indFile ARG         the input ind file path
335     --snpSet ARG             the snpSet of the new package: 1240K, HumanOrigins or
336                               Other. Default: Other
337     --outFormat ARG          the format of the output genotype data: EIGENSTRAT or
338                               PLINK.
339     --onlyGeno               should only the resulting genotype data be returned?
340                               This means the output will not be a Poseidon package
341     --removeOld              Remove the old genotype files when creating the new
342                               ones

```

343 With the default setting

```

344 trident genoconvert -d ... -d ... --outFormat EIGENSTRAT|PLINK

```

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

347 Instead of -d to change Poseidon packages, the combination -r + -g + -s + -i (+ --snpSet) allows to
348 directly convert genotype data that is not wrapped in a Poseidon package. See this example:

```

349 trident genoconvert \
350     -r PLINK \
351     -g 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
352     -s 2018_Mittnik_Baltic/Mittnik_Baltic.bim \
353     -i 2018_Mittnik_Baltic/Mittnik_Baltic.fam \
354     --outFormat EIGENSTRAT

```

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

358 Remember that the POSEIDON.yml file can also be edited by hand if you want to replace the genotype data in
359 a package.

360 1.3.5 Update command

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

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

```

363 Usage: trident update (-d|--baseDir DIR) [--poseidonVersion ARG]
364             [--ignorePoseidonVersion] [--versionComponent ARG]
365             [--noChecksumUpdate] [--newContributors ARG]
366             [--logText ARG] [--force]
367 Update POSEIDON.yml files automatically
368
369 Available options:
370 -h,--help          Show this help text
371 -d,--baseDir DIR   a base directory to search for Poseidon Packages
372                   (could be a Poseidon repository)
373 --poseidonVersion ARG Poseidon version the packages should be updated to:
374                   e.g. "2.5.3" (default: Nothing)
375 --ignorePoseidonVersion Read packages even if their poseidonVersion is not
376                   compatible with the trident version. The assumption
377                   is, that the package is already structurally adjusted
378                   to the trident version and only the version number is
379                   lagging behind.
380 --versionComponent ARG Part of the package version number in the
381                   POSEIDON.yml file that should be updated: Major,
382                   Minor or Patch (see
383                   https://semver.org) (default: Patch)
384 --noChecksumUpdate   Should update of checksums in the POSEIDON.yml file
385                   be skipped
386 --ignoreGeno         ignore SNP and GenoFile
387 --newContributors ARG Contributors to add to the POSEIDON.yml file in the
388                   form "[Firstname Lastname](Email address);..."
389 --logText ARG        Log text for this version jump in the CHANGELOG
390                   file (default: "not specified")
391 --force              Normally the POSEIDON.yml files are only changed if
392                   the poseidonVersion is adjusted or any of the
393                   checksums change. With --force a package version
394                   update can be triggered even if this is not the case.

```

395 It can be called with a lot of optional arguments

```

396 trident update -d ... -d ... \
397   --poseidonVersion "X.X.X" \
398   --versionComponent Major/Minor/Patch \
399   --noChecksumUpdate
400   --ignoreGeno
401   --newContributors "[Firstname Lastname](Email address);..."
402   --logText "short description of the update"
403   --force

```

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

407 **update** if either

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

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

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

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

430 1.4 Inspection commands

431 1.4.1 List command

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

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

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

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

439
440 Available options:

441 <code>-h,--help</code>	Show this help text
442 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages 443 (could be a Poseidon repository)
444 <code>--remote</code>	list packages from a remote server instead the local 445 file system
446 <code>--remoteURL ARG</code>	URL of the remote Poseidon 447 server (default: "https://c107-224.cloud.gwdg.de")

```

448 --packages          list all packages
449 --groups            list all groups, ignoring any group names after the
450                      first as specified in the Janno-file
451 --individuals        list individuals
452 -j,--jannoColumn JANNO_HEADER
453                      list additional fields from the janno files, using
454                      the Janno column heading name, such as Country, Site,
455                      Date_C14_Uncal_BP, Endogenous, ...
456 --raw                output table as tsv without header. Useful for piping
457                      into grep or awk
458 --ignoreGeno         ignore SNP and GenoFile

```

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

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

461 This will yield a table like this

```

462 .------.------.------.
463 |           Title           |    Date    | Nr Individuals |
464 :=====:=====:=====:
465 | 2015_1000Genomes_1240K_haploid_pull | 2020-08-10 | 2535          |
466 | 2016_Mallick_SGDP1240K_diploid_pull | 2020-08-10 | 280           |
467 | 2018_BostonDatashare_modern_published | 2020-08-10 | 2772          |
468 | ...                          | ...        |               |
469 '-----'-----'-----'

```

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

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

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

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

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

```

477 trident list -d ... -d ... --groups
478 trident list -d ... -d ... --individuals

```

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

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

486 1.4.2 Summarise command

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

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

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

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

491

492 Available options:

493 -h,--help	Show this help text
494 -d,--baseDir DIR	a base directory to search for Poseidon Packages (could be a Poseidon repository)
495 --raw	output table as tsv without header. Useful for piping 497 into grep or awk

498 You can run it with

499 trident summarise -d ... -d ...

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

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

505 1.4.3 Survey command

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

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

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

509 Survey the degree of context information completeness for Poseidon packages

510

511 Available options:

512 -h,--help	Show this help text
513 -d,--baseDir DIR	a base directory to search for Poseidon Packages 514 (could be a Poseidon repository)
515 --raw	output table as tsv without header. Useful for piping 516 into grep or awk

517 Running

518 trident survey -d ... -d ...

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

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

522 1.4.4 Validate command

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

524 Click here for command line details

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

526 Check one or multiple Poseidon packages for structural correctness

527

528 Available options:

529 -h,--help	Show this help text
530 -d,--baseDir DIR	a base directory to search for Poseidon Packages
531	(could be a Poseidon repository)
532 --verbose	print more output to the command line
533 --ignoreGeno	ignore SNP and GenoFile
534 --noExitCode	do not produce an explicit exit code

535 You can run it with

536 **trident validate -d ... -d ...**

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

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

- 541 • Presence of the necessary files
- 542 • Full structural correctness of .bib and .janno file
- 543 • Superficial correctness of genotype data files. A full check would be too computationally expensive
- 544 • Correspondence of BibTeX keys in .bib and .janno
- 545 • Correspondence of individual and group IDs in .janno and genotype data files

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

549 1.5 Analysis commands

550 All analysis commands (e.g. **trident fstats**) have been moved from trident to the analysis tool **xerxes**.