

Contents

0.1	Guide for trident v1.1.0.0 to v1.1.4.2	1
0.1.1	Poseidon package repositories	1
0.1.2	Analysing your own dataset outside of the main repository	1
0.1.3	Package creation and manipulation commands	2
0.1.4	Inspection commands	11

0.1 Guide for trident v1.1.0.0 to v1.1.4.2

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`:

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

0.1.2 Analysing your own dataset outside of the main repository

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

```
~/my_project/my_project.geno
```

40 ~/my_project/my_project.snp

41 ~/my_project/my_project.ind

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

44 poseidonVersion: 2.5.0

45 title: My_awesome_project

46 description: Unpublished genetic data from my awesome project

47 contributor:

48 - name: Stephan Schiffels

49 email: schiffels@institute.org

50 packageVersion: 0.1.0

51 lastModified: 2020-10-07

52 genotypeData:

53 format: EIGENSTRAT

54 genoFile: my_project.geno

55 snpFile: my_project.snp

56 indFile: my_project.ind

57 jannoFile: my_project.janno

58 bibFile: sources.bib

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

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

65 **trident list -d /path/to/poseidon/packages/modern **

66 **-d /path/to/poseidon/packages/ReferenceGenomes**

67 **-d ~/my_project --packages**

68 0.1.3 Package creation and manipulation commands

69 **0.1.3.1 Init command** **init** creates a new, valid poseidon package from genotype data files. It adds a valid
70 POSEIDON.yml file, a dummy .janno file for context information and an empty .bib file for literature references.

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

72 Usage: **trident init** ((-p|--genoOne ARG) | --inFormat ARG --genoFile ARG

73 --snpFile ARG --indFile ARG) [--snpSet ARG]

74 (-o|--outPackagePath ARG) [-n|--outPackageName ARG]

75 [--minimal]

76 Create a new Poseidon package from genotype data

77 Available options:

79 -h,--help Show this help text

80 -p,--genoOne ARG one of the input genotype data files. Expects .bed or

```

81         .bim or .fam for PLINK and .geno or .snp or .ind for
82         EIGENSTRAT. The other files must be in the same
83         directory and must have the same base name
84     --inFormat ARG         the format of the input genotype data: EIGENSTRAT or
85                             PLINK (only necessary for data input with --genoFile
86                             + --snpFile + --indFile)
87     --genoFile ARG         the input geno file path
88     --snpFile ARG          the input snp file path
89     --indFile ARG          the input ind file path
90     --snpSet ARG           the snpSet of the new package: 1240K, HumanOrigins or
91                             Other. Default: Other
92     -o,--outPackagePath ARG the output package directory path
93     -n,--outPackageName ARG the output package name - this is optional: If no
94                             name is provided, then the package name defaults to
95                             the basename of the (mandatory) --outPackagePath
96                             argument
97     --minimal              should only a minimal output package be created?

```

The command

```

99 trident init \
100   --inFormat EIGENSTRAT/PLINK \
101   --genoFile path/to/geno_file \
102   --snpFile path/to/snp_file \
103   --indFile path/to/ind_file \
104   --snpSet 1240K|HumanOrigins|Other \
105   -o path/to/new_package_name

```

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

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

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

0.1.3.2 Fetch command `fetch` allows to download poseidon packages from a remote poseidon server.

Click here for command line details

```

115 Usage: trident fetch (-d|--baseDir DIR)
116         (--downloadAll |

```

```

117         (--fetchFile ARG | (-f|--fetchString ARG)))
118         [--remoteURL ARG] [-u|--upgrade]
119 Download data from a remote Poseidon repository
120
121 Available options:
122 -h,--help                Show this help text
123 -d,--baseDir DIR         a base directory to search for Poseidon Packages
124                           (could be a Poseidon repository)
125 --downloadAll            download all packages the server is offering
126 --fetchFile ARG          A file with a list of packages. Works just as -f, but
127                           multiple values can also be separated by newline, not
128                           just by comma. -f and --fetchFile can be combined.
129 -f,--fetchString ARG     List of packages to be downloaded from the remote
130                           server. Package names should be wrapped in asterisks:
131                           *package_title*. You can combine multiple values with
132                           comma, so for example: "*package_1*, *package_2*,
133                           *package_3*". fetchString uses the same parser as
134                           forgeString, but does not allow excludes. If groups
135                           or individuals are specified, then packages which
136                           include these groups or individuals are included in
137                           the download.
138 --remoteURL ARG          URL of the remote Poseidon server
139                           (default: "https://c107-224.cloud.gwdg.de")
140 -u,--upgrade             overwrite outdated local package versions
141
142 It works with
143
144 trident fetch -d ... -d ... \
145   -f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<Individual1>" \
146   --fetchFile path/to/forgeFile

```

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.

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

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

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.

160 **0.1.3.3 Forge command** `forge` creates new poseidon packages by extracting and merging packages,
161 populations and individuals from your poseidon repositories.

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

163 Usage: `trident forge ((-d|--baseDir DIR) |`
164 `((-p|--genoOne ARG) | --inFormat ARG --genoFile ARG`
165 `--snpFile ARG --indFile ARG) [--snpSet ARG])`
166 `[--forgeFile ARG | (-f|--forgeString ARG)]`
167 `[--selectSnps ARG] [--intersect] [--outFormat ARG]`
168 `[--minimal] [--onlyGeno] (-o|--outPackagePath ARG)`
169 `[-n|--outPackageName ARG] [--no-extract]`
170 Select packages, groups or individuals and create a new Poseidon package from
171 them

173 Available options:

174	<code>-h,--help</code>	Show this help text
175	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
177	<code>-p,--genoOne ARG</code>	one of the input genotype data files. Expects .bed or 178 .bim or .fam for PLINK and .geno or .snp or .ind for 179 EIGENSTRAT. The other files must be in the same 180 directory and must have the same base name
181	<code>--inFormat ARG</code>	the format of the input genotype data: EIGENSTRAT or 182 PLINK (only necessary for data input with --genoFile 183 + --snpFile + --indFile)
184	<code>--genoFile ARG</code>	the input geno file path
185	<code>--snpFile ARG</code>	the input snp file path
186	<code>--indFile ARG</code>	the input ind file path
187	<code>--snpSet ARG</code>	the snpSet of the new package: 1240K, HumanOrigins or 188 Other. Default: Other
189	<code>--forgeFile ARG</code>	A file with a list of packages, groups or individual 190 samples. Works just as -f, but multiple values can 191 also be separated by newline, not just by comma. 192 Empty lines are ignored and comments start with "#", 193 so everything after "#" is ignored in one line. 194 Multiple instances of -f and --forgeFile can be 195 given. They will be evaluated according to their 196 input order on the command line.
197	<code>-f,--forgeString ARG</code>	List of packages, groups or individual samples to be 198 combined in the output package. Packages follow the 199 syntax *package_title*, populations/groups are simply 200 group_id and individuals <individual_id>. You can 201 combine multiple values with comma, so for example: 202 "*package_1*, <individual_1>, <individual_2>," 203 group_1". Duplicates are treated as one entry. 204 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

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

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

250 packages.

251 forge can be used with

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

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

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

```
265 trident forge \  
266 -d 2017_GonzalesFortesCurrentBiology \  
267 -p 2018_VeeramahPNAS/2018_VeeramahPNAS.fam \  
268 --inFormat PLINK \  
269 --genoFile 2017_HaberAJHG/2017_HaberAJHG.bed \  
270 --snpFile 2017_HaberAJHG/2017_HaberAJHG.bim \  
271 --indFile 2017_HaberAJHG/2017_HaberAJHG.fam \  
272 -f "<STR241.SG>,<ERS1790729.SG>,Iberia_HG.SG" \  
273 -o testpackage \  
274 --outFormat EIGENSTRAT \  
275 --onlyGeno
```

276 **0.1.3.3.1 The forge selection language** Entities in the --forgeString or the --forgeFile have to be
277 marked in a certain way:

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

283 Do not forget to wrap the forgeString in quotes.

284 You can use both -f/--forgeString and --forgeFile and even combine multiple of each. They are evaluated
285 in order.

286 In the file each line is treated as a separate forgeString, empty lines are ignored and #s start comments. So this
287 is a valid forgeFile:

```
288 # Packages  
289 *package1*, *package2*
```

290

291 # Groups and individuals from other packages beyond package1 and package2

292 group1, <individual1>, group2, <individual2>, <individual3>

293

294 # group2 has two outlier individuals that should be ignored

295 -<bad_individual1> # This one has very low coverage

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

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

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

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

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

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

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

303 therefore merge all available individuals.

304 **0.1.3.3.2 Other options** Just as for **init** the output package of **forge** is created as a new directory -o.

305 The title can also be explicitly defined with -n.

306 --minimal allows for the creation of a minimal output package without .bib and .janno. This might be

307 especially useful for data analysis pipelines, where only the genotype data is required. Even more basic output

308 comes with --onlyGeno, which means that only the genotype data is returned without any Poseidon package.

309 **forge** has a an optional flag --intersect, that defines, if the genotype data from different packages should

310 be merged with an **union** or an **intersect** operation. The default (if this option is not set) is to output the

311 union of all SNPs, with genotypes defined as missing in samples from packages which do not have a SNP that is

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

313 SNPs, but less missingness.

314 --intersect also influences the automatic determination of the snpSet field in the POSEIDON.yml file for the

315 resulting package. If the snpSets of all input packages are identical, then the resulting package will just inherit

316 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

317 --selectSnps allows to provide **forge** with a SNP file in EIGENSTRAT (.snp) or PLINK (.bim) format to

318 create a package with a specific selection. When this option is set, the output package will have exactly the

319 SNPs listed in this file. Any SNP not listed in the file will be excluded. If --intersect is also set, only the

320 SNPs overlapping between the SNP file and the forged packages are output.

321 Merging genotype data across different data sources and file formats is tricky. **forge** is more verbose about

322 potential issues, if the --logMode flag is set to VerboseLog.

323 **0.1.3.4 Genoconvert command** **genoconvert** converts the genotype data in a Poseidon package to a

324 different file format. The respective entries in the POSEIDON.yml file are changed accordingly.

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

```
326 Usage: trident genoconvert ((-d|--baseDir DIR) |
327                             ((-p|--genoOne ARG) | --inFormat ARG --genoFile ARG
328                             --snpFile ARG --indFile ARG) [--snpSet ARG])
329                             --outFormat ARG [--onlyGeno]
330                             [-o|--outPackagePath ARG] [--removeOld]
```

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

332 Available options:

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

359 With the default setting

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

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

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

366 Instead of -d to change Poseidon packages, the -p (+ --snpSet) or --inFormat + --genoFile + --snpFile
367 + --indFile (+ --snpSet) allow to directly convert genotype data that is not wrapped in a Poseidon package

368 and store it to a directory given in -o. See this example:

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

373 **0.1.3.5 Update command** update automatically harmonizes POSEIDON.yml files of one or multiple
374 packages if the packages were changed. This is not an automatic update from one Poseidon version to the next!

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

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

380 Update POSEIDON.yml files automatically

381
382 Available options:

383 -h,--help	Show this help text
384 -d,--baseDir DIR	a base directory to search for Poseidon Packages 385 (could be a Poseidon repository)
386 --poseidonVersion ARG	Poseidon version the packages should be updated to: 387 e.g. "2.5.3" (default: Nothing)
388 --ignorePoseidonVersion	Read packages even if their poseidonVersion is not 389 compatible with the trident version. The assumption 390 is, that the package is already structurally adjusted 391 to the trident version and only the version number is 392 lagging behind.
393 --versionComponent ARG	Part of the package version number in the 394 POSEIDON.yml file that should be updated: Major, 395 Minor or Patch (see https://semver.org) 396 (default: Patch)
397 --noChecksumUpdate	Should update of checksums in the POSEIDON.yml file 398 be skipped
399 --ignoreGeno	ignore SNP and GenoFile
400 --newContributors ARG	Contributors to add to the POSEIDON.yml file in the 401 form "[Firstname Lastname](Email address);..."
402 --logText ARG	Log text for this version jump in the CHANGELOG file 403 (default: "not specified")
404 --force	Normally the POSEIDON.yml files are only changed if 405 the poseidonVersion is adjusted or any of the 406 checksums change. With --force a package version 407 update can be triggered even if this is not the case.

408 It can be called with a lot of optional arguments

```
409 trident update -d ... -d ... \  
410
```

```

410 --poseidonVersion "X.X.X" \
411 --versionComponent Major/Minor/Patch \
412 --noChecksumUpdate
413 --ignoreGeno
414 --newContributors "[Firstname Lastname](Email address);..."
415 --logText "short description of the update"
416 --force

```

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

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

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

- If `--poseidonVersion` is different from the `poseidonVersion` field in the package, then that will be updated.
- 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](#)).
- The `lastModified` field will be updated to the current day (based on your computer's system time).
- The contributors in `--newContributors` will be added to the `contributor` field if they're not there already.
- 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`.
- 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.

: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.

0.1.4 Inspection commands

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

Click here for command line details

```

447 Usage: trident list ((-d|--baseDir DIR) | --remote [--remoteURL ARG])
448                (--packages | --groups | --individuals
449                [-j|--jannoColumn JANNO_HEADER]) [--raw]
450 List packages, groups or individuals from local or remote Poseidon

```

```

451 repositories
452
453 Available options:
454 -h,--help          Show this help text
455 -d,--baseDir DIR    a base directory to search for Poseidon Packages
456                     (could be a Poseidon repository)
457 --remote            list packages from a remote server instead the local
458                     file system
459 --remoteURL ARG      URL of the remote Poseidon server
460                     (default: "https://c107-224.cloud.gwdg.de")
461 --packages          list all packages
462 --groups            list all groups, ignoring any group names after the
463                     first as specified in the Janno-file
464 --individuals       list individuals
465 -j,--jannoColumn JANNO_HEADER
466                     list additional fields from the janno files, using
467                     the Janno column heading name, such as Country, Site,
468                     Date_C14_Uncal_BP, Endogenous, ...
469 --raw              output table as tsv without header. Useful for piping
470                     into grep or awk
471 --ignoreGeno        ignore SNP and GenoFile
472
473 To list packages from your local repositories, as seen above you can run
474
475 trident list -d ... -d ... --packages
476
477 This will yield a table like this
478
479 .------.------.------.
480 |           Title           |   Date   | Nr Individuals |
481 :=====:=====:=====:
482 | 2015_1000Genomes_1240K_haploid_pulldown | 2020-08-10 | 2535          |
483 | 2016_Mallick_SGDP1240K_diploid_pulldown | 2020-08-10 | 280           |
484 | 2018_BostonDatashare_modern_published   | 2020-08-10 | 2772          |
485 | ...                                     | ...       |               |
486 '-----'-----'-----'
487
488 so a nicely formatted table of all packages, their last update and the number of individuals in it.
489
490 To view packages on the remote server, instead of using directories to specify the locations of repositories on
491 your system, you can use --remote to show packages on the remote server. For example
492
493 trident list --packages --remote
494
495 will result in a view of all published packages in our public online repository.
496
497 You can also list groups, as defined in the third column of EIGENSTRAT .ind files (or the first column of a
498 PLINK .fam file), and individuals:
499
500 trident list -d ... -d ... --groups
501 trident list -d ... -d ... --individuals

```

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

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

499 **0.1.4.2 Summarise command** `summarise` prints some general summary statistics for a given poseidon
500 dataset taken from the `.janno` files.

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

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

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

504 Available options:

506 <code>-h,--help</code>	Show this help text
507 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages 508 (could be a Poseidon repository)
509 <code>--raw</code>	output table as tsv without header. Useful for piping 510 into <code>grep</code> or <code>awk</code>

511 You can run it with

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

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

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

518 **0.1.4.3 Survey command** `survey` tries to indicate package completeness (mostly focused on `.janno` files)
519 for poseidon datasets.

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

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

522 Survey the degree of context information completeness for Poseidon packages

523 Available options:

525 <code>-h,--help</code>	Show this help text
526 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages 527 (could be a Poseidon repository)
528 <code>--raw</code>	output table as tsv without header. Useful for piping 529 into <code>grep</code> or <code>awk</code>

530 Running

531 `trident survey -d ... -d ...`

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

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

535 **0.1.4.4 Validate command** `validate` checks poseidon datasets for structural correctness.

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

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

538 Check one or multiple Poseidon packages for structural correctness

539

540 Available options:

541	<code>-h,--help</code>	Show this help text
542	<code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
543		
544	<code>--verbose</code>	print more output to the command line
545	<code>--ignoreGeno</code>	ignore SNP and GenoFile
546	<code>--noExitCode</code>	do not produce an explicit exit code

547 You can run it with

548 `trident validate -d ... -d ...`

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

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

- 553 • Presence of the necessary files
- 554 • Full structural correctness of `.bib` and `.janno` file
- 555 • Superficial correctness of genotype data files. A full check would be too computationally expensive
- 556 • Correspondence of BibTeX keys in `.bib` and `.janno`
- 557 • Correspondence of individual and group IDs in `.janno` and genotype data files

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