

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 “unpacked” 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     /...
35     /Reference_Genomes
36     /...
37     /...
38     /Archaic_Humans
39     /...
40     /...

```

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

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

```

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

```

46 2 Analysing your own dataset outside of the main repository

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

```

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

```

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

```

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58 description: Unpublished genetic data from my awesome project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

```

71 Two remarks: 1) all file paths are considered *relative* to the directory in which `POSEIDON.yml` resides. Here I
 72 assume that you put this file into the same directory as the three genotype files. 2) Besides the genotype data

73 files there are two (technically optional) files referenced by this example `POSEIDON.yml` file: `sources.bib`
74 and `my_project.janno`. Of course you can add them manually - `init` automatically creates empty dummy
75 versions.

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

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

81 3 Package creation and manipulation commands

82 3.1 Init command

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

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

```
86 Usage: trident init ((-p|--genoOne ARG) | (-r|--inFormat ARG)  
87 (-g|--genoFile ARG) (-s|--snpFile ARG) (-i|--indFile ARG))  
88 [--snpSet ARG] (-o|--outPackagePath ARG)  
89 [-n|--outPackageName ARG] [--minimal]  
90 Create a new Poseidon package from genotype data
```

91 Available options:

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

111 The command

```
112 trident init \  
113
```

```

113 -r EIGENSTRAT/PLINK \
114 -g path/to/geno_file \
115 -s path/to/snp_file \
116 -i path/to/ind_file \
117 --snpSet 1240K|HumanOrigins|Other \
118 -o path/to/new_package_name

```

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

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

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

127 3.2 Fetch command

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

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

```

130 Usage: trident fetch [-d|--baseDir DIR]
131                  (--downloadAll |
132                  (--fetchFile ARG | (-f|--fetchString ARG)))
133                  [--remoteURL ARG] [-u|--upgrade]

```

134 Download data from a remote Poseidon repository

135 Available options:

```

137 -h,--help          Show this help text
138 -d,--baseDir DIR   a base directory to search for Poseidon Packages
139                   (could be a Poseidon repository)
140 --downloadAll       download all packages the server is offering
141 --fetchFile ARG     A file with a list of packages. Works just as -f, but
142                   multiple values can also be separated by newline, not
143                   just by comma. -f and --fetchFile can be combined.
144 -f,--fetchString ARG List of packages to be downloaded from the remote
145                   server. Package names should be wrapped in asterisks:
146                   *package_title*. You can combine multiple values with
147                   comma, so for example: "*package_1*, *package_2*,
148                   *package_3*". fetchString uses the same parser as

```

```

149         forgeString, but does not allow excludes. If groups
150         or individuals are specified, then packages which
151         include these groups or individuals are included in
152         the download.
153     --remoteURL ARG        URL of the remote Poseidon server
154                             (default: "https://c107-224.cloud.gwdg.de")
155     -u,--upgrade           overwrite outdated local package versions

```

156 It works with

```

157 trident fetch -d ... -d ... \
158     -f "*package_title_1*,*package_title_2*,*package_title_3*,group_name,<Individual1>" \
159     --fetchFile path/to/forgeFile

```

160 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.

169 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.

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

173 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.

175 3.3 Forge command

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

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

```

179 Usage: trident forge [-d|--baseDir DIR]
180         [
181             ((-p|--genoOne ARG) | (-r|--inFormat ARG)
182             (-g|--genoFile ARG) (-s|--snpFile ARG)
183             (-i|--indFile ARG)) [--snpSet ARG]]
184         [--forgeFile ARG | (-f|--forgeString ARG)]
185         [--selectSnps ARG] [--intersect] [--outFormat ARG]
186         [--minimal] [--onlyGeno] (-o|--outPackagePath ARG)
187         [-n|--outPackageName ARG] [--no-extract]
188     Select packages, groups or individuals and create a new Poseidon package from
189     them

```

```

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

forge can be used with

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

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

Including one or multiple Poseidon packages with **-d** is not the only way to include data for a forge operation.

It is also possible to include unpackaged genotype data directly with `-r + -g + -s + -i (+ --snpSet)` or `-p (+ --snpSet)`. This makes the following example possible, where we merge data from one Poseidon package and two genotype datasets.

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

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 `ALAO26` becomes `<ALAO26>`.

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`:

```
# Packages
*package1*, *package2*

# Groups and individuals from other packages beyond package1 and package2
group1, <individual1>, group2, <individual2>, <individual3>

# group2 has two outlier individuals that should be ignored
-<bad_individual1> # This one has very low coverage
-<bad_individual2> # This one is from a different time period
```

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

```

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

```

379 With the default setting

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

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

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

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

```

389 trident genoconvert \
390   -p 2018_Mittnik_Baltic/Mittnik_Baltic.bed \
391   --outFormat EIGENSTRAT
392   -o my_directory

```

393 3.5 Update command

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

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

```

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

```

440 `update` if either

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

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

446 the following steps:

- 447 • If `--poseidonVersion` is different from the `poseidonVersion` field in the package, then that will be updated.
- 448 • 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](#)).
- 449 • The `lastModified` field will be updated to the current day (based on your computer's system time).
- 450 • The contributors in `--newContributors` will be added to the `contributor` field if they're not there already.
- 451 • 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`.
- 452 • 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.

453 :heavy_exclamation_mark: As `update` reads and rewrites POSEIDON.yml files, it may change their inner

454 order, layout or even content (e.g. if they have fields which are not in the [Poseidon package definition](#)). Create a

455 backup of the POSEIDON.yml file before running `update` if you are uncertain.

464 4 Inspection commands

465 4.1 List command

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

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

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

471 List packages, groups or individuals from local or remote Poseidon

472 repositories

473 Available options:

475 <code>-h,--help</code>	Show this help text
476 <code>-d,--baseDir DIR</code>	a base directory to search for Poseidon Packages (could be a Poseidon repository)
477 <code>--remote</code>	list packages from a remote server instead the local file system

```

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

```

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

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

495 This will yield a table like this

```

496 .------.------.------.
497 |                Title                |   Date   | Nr Individuals |
498 :=====:=====:=====:
499 | 2015_1000Genomes_1240K_haploid_pull | 2020-08-10 | 2535          |
500 | 2016_Mallick_SGDP1240K_diploid_pull | 2020-08-10 | 280           |
501 | 2018_BostonDatashare_modern_published | 2020-08-10 | 2772          |
502 | ...                                | ...       |               |
503 '------'------'-----'

```

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

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

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

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

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

```

511 trident list -d ... -d ... --groups
512 trident list -d ... -d ... --individuals

```

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

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

520 4.2 Summarise command

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

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

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

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

525 Available options:

527 -h,--help	Show this help text
528 -d,--baseDir DIR	a base directory to search for Poseidon Packages (could be a Poseidon repository)
530 --raw	output table as tsv without header. Useful for piping 531 into grep or awk

532 You can run it with

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

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

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

539 4.3 Survey command

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

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

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

543 Survey the degree of context information completeness for Poseidon packages

544 Available options:

546 -h,--help	Show this help text
547 -d,--baseDir DIR	a base directory to search for Poseidon Packages (could be a Poseidon repository)
549 --raw	output table as tsv without header. Useful for piping 550 into grep or awk

551 Running

552 `trident survey -d ... -d ...`

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

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

556 4.4 Validate command

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

558 Click here for command line details

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

560 Check one or multiple Poseidon packages for structural correctness

561 Available options:

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

568 You can run it with

569 trident validate -d ... -d ...

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

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

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

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