Poseidon package specification

Table of contents

1 The Poseidon Standard v2.7.1

Poseidon is a solution for archaeogenetic genotype data organisation.

This standard defines the core components of the Poseidon package. Further details on genotype data, the .janno file and the .ssf file are documented on the Poseidon website.

A changelog for this standard is available on the website here.

1.1 The Poseidon package structure

A Poseidon package stores genotype data with context information for DNA samples from (ancient) (human) individuals. Packages are defined by the POSEIDON.yml file, which holds relative paths to all other files in a package.

A package therefore MUST contain:

- A `POSEIDON.yml` file to formally define the package
- Genotype data in PLINK or EIGENSTRAT format

It SHOULD additionally contain:

- A `.janno` file to store context information on spatiotemporal origin or sample quality
- A `.bib` file for literature references

It CAN also contain:

- A `README.md` file for arbitrary, additional context information
- A `CHANGELOG.md` file to document changes to the package
- A `.ssf` file with information on the underlying raw sequencing data

Here is an example of a package `Switzerland_LNBA_Roswita` in one directory:

```
Switzerland_LNBA_Roswita/POSEIDON.yml
Switzerland_LNBA_Roswita/Switzerland_LNBA.bed
Switzerland_LNBA_Roswita/Switzerland_LNBA.bim
Switzerland_LNBA_Roswita/Switzerland_LNBA.fam
Switzerland_LNBA_Roswita/Switzerland_LNBA.janno
Switzerland_LNBA_Roswita/Switzerland_LNBA.ssf
Switzerland_LNBA_Roswita/Switzerland_LNBA.bib
Switzerland_LNBA_Roswita/README.md
Switzerland_LNBA_Roswita/CHANGELOG.md
```

All text files in the package MUST be UTF-8 encoded.

1.2 The `POSEIDON.yml` file

The `POSEIDON.yml` file defines Poseidon packages by listing metainformation and relative paths in a standardised, machine-readable format.

- It MUST be a valid YAML file.
- Its mandatory and optional fields are documented in the POSEIDON_yml_fields.tsv file in this repository.

Here is an example for a `POSEIDON.yml` file:

```
poseidonVersion: 2.7.1
title: Switzerland_LNBA_Roswita
description: LNBA Switzerland genetic data not yet published
contributor:
  - name: Roswita Malone
    email: roswita.malone@example.org
    orcid: 1234-1234-1234-1234
  - name: Paul Panther
    email: paul.panther@example.edu
packageVersion: 1.1.2
lastModified: 2021-01-28
genotypeData:
  format: PLINK
  genoFile: Switzerland_LNBA_Roswita.bed
  genoFileChkSum: 95b093eefacc1d6499afcfe89b15d56c
  snpFile: Switzerland_LNBA_Roswita.bim
  snpFileChkSum: 6771d7c873219039ba3d5bdd96031ce3
  indFile: Switzerland_LNBA_Roswita.fam
  indFileChkSum: f77dc756666dbfef3bb35191ae15a167
  snpSet: 1240K
jannoFile : Switzerland_LNBA_Roswita.janno
jannoFileChkSum: 555d7733135ebcabd032d581381c5d6f
sequencingSourceFile: Switzerland_LNBA_Roswita.ssf
sequencingSourceFileChkSum: 19db1906240ee2f076e1a9659567dca4
bibFile: Switzerland_LNBA_Roswita.bib
bibFileChkSum: 70cd3d5801cee8a93fc2eb40a99c63fa
readmeFile: README.md
changelogFile: CHANGELOG.md
```

When a package is modified in any way (including updates of the context information in the `.janno` file), then the `packageVersion` field SHOULD be incremented and the `lastModified` field updated to the current date.

1.2.1 Package versioning

The `packageVersion` field is a mandatory entry of the `POSEIDON.yml` file. It denotes the version of the individual package, using a three-component versioning system derived from semantic versioning.

Each version number is comprised of three numbers, separated by a `.`. For example: `0.1.0`, `1.0.0` or `2.1.3`. The first number gives the `Major`, the second the `Minor` and the third the `Patch` component of the version number. For a Poseidon package these components SHOULD be incremented when the following changes occur:

- `Major` (e.g. `1.4.2` -> `2.0.0`)
  - When samples are added to a package.
  - When samples are removed from a package.
  - When the genotype data (i.e. the contents of the `.bed`/`.bim`/`.fam` or `.geno`/`.snp`/`.ind` files) for any number of samples is changed.
- `Minor` (e.g. `1.4.2` -> `1.5.0`)
  - When larger pieces of meta- or context information are added or modified in any package file, except the genotype data. For example:
    - An entire `.janno`, `.bib` or `.ssf` file is added or replaced.
    - Entire columns in the `.janno` or `.ssf` file are added or replaced.
    - Primary publications for samples in the `.janno` and `.bib` file are added or replaced.
- `Patch` (e.g. `1.4.2` -> `1.4.3`)
  - When smaller pieces of meta- or context information are added or modified in any package file, except the genotype data. For example:
    - Individual entries in the `.janno` or `.ssf` file are added or replaced.
    - Secondary publications for samples in the `.janno` and `.bib` file are added or replaced.
    - BibTeX entries in the `.bib` file are modified.
    - The package `description` changes in the `POSEIDON.yml` file.
    - The `CHANGELOG.md` file is modified with additional information on previous entries.

When the `packageVersion` is changed, then the `lastModified` date MUST be updated and an entry to the `CHANGELOG.md` file SHOULD be added summarising the changes made.

Packages SHOULD start at `packageVersion` `0.1.0`.

1.3 Genotype data

Genotype data in Poseidon packages is stored either in (binary) PLINK or EIGENSTRAT format.

	PLINK (binary)	EIGENSTRAT
genotype file	`.bed` (binary biallelic genotype table)	`.geno` (genotype file)
SNP file	`.bim` (extended MAP file)	`.snp` (snp file)
individual file	`.fam` (sample information)	`.ind` (indiv file)

In addition to these files (and optionally their checksums), the POSEIDON.yml file SHOULD also provide a `snpSet` entry which determines the shape of the genotype file.

1.4 The `.janno` file

The `.janno` file is a tab-separated text file with a header line. It holds context information (variables/columns) for each sample (objects/rows) in a package.

- A set of strictly defined core variables (defined by column name) and their possible content are documented here: janno_columns.tsv
- A `.janno` file CAN have all of these core variables, or only a subset of them.
- Only three columns MUST be present to make the file valid: **Poseidon_ID**, **Group_Name** and **Genetic_Sex**
- Arbitrary columns not defined here CAN be added as long as their column names do not clash with the defined ones.
- The column order is irrelevant.
- If information is unknown or a variable does not apply for a certain sample, then the respective cell(s) can be filled with the NULL value `n/a` or simply an empty string.
- The order of the samples (rows) in the `.janno` file MUST be equal to the order in the genetic data files (`.ind`, `.fam`) in the package.
- The values in the columns **Poseidon_ID**, **Group_Name** and **Genetic_Sex** MUST be equal to the terms used in the genetic data files (`.ind`, `.fam`).
- Multiple predefined columns of the `.janno` file are list columns that can hold multiple values (either strings or numerics) separated by `;`.
- The decimal separator for all floating point numbers MUST be `.`.

For a more extensive documentation of the columns and their interaction see https://poseidon-framework.github.io/#/janno_details.

1.5 The `.bib` file

A BibTeX file with all references listed in the `.janno` file. The entry keys MUST fit the ones used in the `.janno` file.

Example:

```
@article{CassidyPNAS2015,
    doi = {10.1073/pnas.1518445113},
    url = {https://doi.org/10.1073%2Fpnas.1518445113},
    year = 2015,
    month = {dec},
    publisher = {Proceedings of the National Academy of Sciences},
    volume = {113},
    number = {2},
    pages = {368--373},
    author = {Lara M. Cassidy and Rui Martiniano and Eileen M. Murphy and Matthew D.
    Teasdale and James Mallory and Barrie Hartwell and Daniel G. Bradley},
    title = {Neolithic and Bronze Age migration to Ireland and establishment of the
    insular Atlantic genome},
    journal = {Proceedings of the National Academy of Sciences}
}
```

To connect a sample in the package to this particular literature reference, the .janno file column `Publication` would have to be filled with `CassidyPNAS2015`.

1.6 The `README.md` file

A simple markdown file with informal, arbitrarily structured information accompanying the package.

Example:

```
This package contains a rather interesting set of samples relevant for the peopling of
the Territory of Christmas Island in the Indian Ocean. We consider this especially
relevant, because ...
```

1.7 The `CHANGELOG.md` file

A markdown file to document changes in the history of a package.

Example:

```
- V 1.2.0: Fixed a spelling mistake in the site name "Hosenacker"->"Rosenacker"
- V 1.1.1: Added mtDNA contamination estimation to .janno file
- V 1.1.0: The authors of @Gassenhauer_2021 made some previously restricted samples for
their publication available later and we added them
- V 1.0.0: Creation of the package
```

The structure with `- V X.X.X:` at the beginning of each line is not mandatory, but SHOULD be followed for reasons of interoperability.

1.8 The `.ssf` file

The `.ssf` file is another tab-separated text file with a header line. It stores sequencing source data, so metainformation about the raw sequencing data behind the genotypes in a Poseidon package. The primary entities in this table are sequencing entities, typically corresponding to DNA libraries or even multiple runs/lanes of the same library.

- The predefined columns are specified here: ssf_columns.tsv
- All columns of this schema are optional, so a `.ssf` CAN have all of these core variables, only a subset of them, or even none. It SHOULD have a `poseidon_IDs` column, though, to link the sequencing entities to the Poseidon package.
- The link to the individuals listed in the `.janno`-file (and therefore to the entire Poseidon package) is made through a many-to-many foreign-key relationship between the .janno column `Poseidon_ID` and the .ssf column `poseidon_IDs`. That means each entry in the .janno file can be linked to many rows in the .ssf file and vice versa.
- As in the `.janno` file arbitrary columns not defined here CAN be added to the `.ssf` file as long as their column names do not clash with the defined ones.
- The order of columns and rows is irrelevant.
- If information is unknown or a variable does not apply, then the respective cell(s) can be filled with the NULL value `n/a` or simply an empty string.
- Multiple predefined columns of the `.ssf` file are list columns that can hold multiple values (either strings or numerics) separated by `;`.
- The decimal separator for all floating point numbers MUST be `.`.