Poseidon package specification

Table of contents

I 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 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.

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.

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`.

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.

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.

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`.

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 ...
```

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.

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 `.`.

II Appendix

1 POSEIDON.yml file fields

POSEIDON.yml file fields

Field	Description
poseidonVersion*	Poseidon package format version (e.g. 2.0.1). Should strictly follow the format X.Y.Z type: String
title*	title of the package type: String
description	some descriptive words about the package type: String
contributor	list of contributors to the package (not publication author, but the Poseidon package creator) type: Array
name*	name of one contributor subfield of: contributor type: String
email*	email of one contributor (must be a valid email address) subfield of: contributor type: String format: Email
orcid	orcid of one contributor (must be a valid orcid) subfield of: contributor type: String format: ORCID
packageVersion*	version of the package (should be changed/incremented when the package is changed). Should strictly follow the format X.Y.Z type: String
lastModified	date of last modification of the Poseidon package (should be updated when the package is changed) type: Date format: YYYY-MM-DD
genotypeData*	genotype file name section
format*	file format definition, allows EIGENSTRAT and PLINK subfield of: genotypeData type: String

POSEIDON.yml file fields *(continued)*

Field	Description
genoFile*	relative path to genoFile subfield of: genotypeData type: String format: Path
genoFileChkSum	md5 checksum of the genoFile subfield of: genotypeData type: String
snpFile*	relative path to snpFile subfield of: genotypeData type: String format: Path
snpFileChkSum	md5 checksum of the snpFile subfield of: genotypeData type: String
indFile*	relative path to indFile subfield of: genotypeData type: String format: Path
indFileChkSum	md5 checksum of the indFile subfield of: genotypeData type: String
snpSet	Can be either 1240K, HumanOrigins or Other depending on the list of SNPs used subfield of: genotypeData type: String format: (1240K|HumanOrigins|Other)
jannoFile	relative path to jannoFile type: String format: Path
jannoFileChkSum	md5 checksum of the jannoFile type: String
sequencingSourceFile	relative path to sequencingSourceFile type: String format: Path
sequencingSourceFileChkSum	md5 checksum of the sequencingSourceFile type: String
bibFile	relative path to bibFile type: String format: Path

POSEIDON.yml file fields *(continued)*

Field	Description
bibFileChkSum	md5 checksum of the bibFile type: String
readmeFile	relative path to readmeFile type: String format: Path
changelogFile	relative path to changelogFile type: String format: Path

2 .janno file variables

.janno file variables

Variable	Description
Poseidon_ID*	id as defined by the genetics laboratory, needs to be unique (e.g. I1234, BOT001), needs to fit to the values in the poseidon package .fam file, if multiple datasets exist for the same individual different IDs are required (e.g. loschbour_snpAD) type: String
Genetic_Sex*	"F", "M" or "U" because eigenstrat and plink formats only support these three, edge cases (XXY, XYY, X0) are undefined and should be grouped as F, M or U, with a note added type: Char allowed values: F; M; U
Group_Name*	ideally Eisenmann rule + underscore flags, e.g. to annotate relatives or outliers or low coverage, multiple entries separated by ; to accommodate different labels, value must equal the group name in the .fam file (in case of multiple entries the first one) list column type: String
Alternative_IDs	other identifiers for the same individual, e.g. IDs in other databases or popular names (e.g. Ötzi/Iceman) list column type: String
Relation_To	other individuals (by Poseidon_ID) that are related/identical to this individual, multiple entries separated by ; list column type: String

.janno file variables *(continued)*

Variable	Description
Relation_Degree	relationship degree for relatives mentioned in Related_To, multiple values separated by ; in the same order as Related_To in case of multiple relations list column type: String allowed values: identical; first; second; thirdToFifth; sixthToTenth; unrelated; other
Relation_Type	relationship type for relatives mentioned in Related_To as an arbitrary string (e.g. sister_of, child_of, nephew_of, ...), multiple values separated by ; in the same order as Related_To in case of multiple relations list column type: String
Relation_Note	arbitrary comments about the relations of this individual type: String
Collection_ID	id as defined by the provider/owner of a sample (e.g. grave 40 skeleton 2) type: String
Country	present-day political country type: String
Country_ISO	present-day political country expressed in ISO 3166-1 alpha-2 country codes type: String
Location	unspecified location information like administrative or topographic region or mountains/rivers/lakes/cities nearby type: String
Site	site name type: String
Latitude	latitude with up to 5 places after the decimal point type: Float allowed range: -90 to 90
Longitude	longitude with up to 5 places after the decimal point type: Float allowed range: -180 to 180

.janno file variables *(continued)*

Variable	Description
Date_Type	"C14" if there is a set of radiocarbon dates in the columns Date_C14_Labnr, Date_C14_Uncal_BP and Date_C14_Uncal_BP_Err whose post-calibration probability distribution is a meaningful prior for the individual's year of death, "contextual" for any other age information only given in Date_BC_AD_Start, Date_BC_AD_Median and Date_BC_AD_Stop, "modern" for present-day individuals type: String allowed values: C14; contextual; modern
Date_C14_Labnr	labnr of C14 date, multiple values separated by ; in case of multiple dates list column type: String
Date_C14_Uncal_BP	uncalibrated years BP (as in before 1950AD), as reported by C14 labs, multiple values separated by ; in the same order as Date_C14_Labnr in case of multiple dates, only relevant if Date_Type is "C14" list column type: Integer allowed range: 0 to Inf
Date_C14_Uncal_BP_Err	standard deviation (1 sigma ±), as reported by C14 labs, multiple values separated by ; in the same order as Date_C14_Labnr in case of multiple dates, only relevant if Date_Type is "C14" list column type: Integer allowed range: 0 to Inf
Date_BC_AD_Start	lower (older) bound for the age, negative numbers for BC, positive numbers for AD, in case of C14 dates 95% interval post calibration, 2000 for modern samples type: Integer allowed range: -Inf to 2050
Date_BC_AD_Median	calibrated median age for C14 dates, or simple mid-points for archaeological intervals, 2000 for modern samples type: Integer allowed range: -Inf to 2050
Date_BC_AD_Stop	upper (more recent) bound for the age, negative numbers for BC, positive numbers for AD, in case of C14 dates 95% interval post calibration, 2000 for modern samples type: Integer allowed range: -Inf to 2050
Date_Note	a free text field for arbitrary comments about the dating information type: String

.janno file variables *(continued)*

Variable	Description
MT_Haplogroup	mitochondrial haplogroup after phylotree.org as reported by Haplofind or Haplogrep type: String
Y_Haplogroup	Y-chromosome haplogroup reported as published, for internal data, please follow syntax with main branch + most terminal derived Y-SNP (e.g. R1b-P312) type: String
Source_Tissue	skeletal/tissue/source elements, specific bone name should be reported with an underscore (e.g. bone_phalanx), multiple values separated by ; list column type: String
Nr_Libraries	number of libraries type: Integer
Library_Names	identifiers of the libraries used to generate the genotype data, multiple values separated by ; list column type: String
Capture_Type	specifics of data generation method, multiple values separated by ; list column type: String allowed values: Shotgun; 1240K; ArborComplete; ArborPrimePlus; ArborAncestralPlus; TwistAncientDNA; OtherCapture; ReferenceGenome
UDG	UDG treatment, "mixed" in case multiple libraries with different UDG treatment were merged type: String allowed values: minus; half; plus; mixed
Library_Built	strandedness, "mixed" in case multiple libraries with different protocols were merged type: String allowed values: ds; ss; mixed
Genotype_Ploidy	ploidy of the genotypes type: String allowed values: diploid; haploid
Data_Preparation_Pipeline_URL	URL pointing to a description of the pipeline used to generate the genotype data from the source data type: String

.janno file variables *(continued)*

Variable	Description
Endogenous	% endogenous DNA as estimated from SG libraries (before capture), as for example estimated by EAGER, not on target and no quality filter, in case of multiple libraries report only the highest value type: Float allowed range: 0 to 100
Nr_SNPs	number of SNPs covered type: Integer
Coverage_on_Target_SNPs	average X-fold coverage across targeted SNP sites after quality filtering (internal data) type: Float
Damage	% damage on 5' end for the main shotgun library used for sequencing and/or capture, in case of multiple libraries report a value from the merged read alignment type: Float allowed range: 0 to 100
Contamination	(modern) contamination as measured by the method in Contamination_Meas, multiple values can be separated by ; (for different methods! In case of multiple libraries report a value from the merged read alignment), Contamination, Contamination_Err and Contamination_Meas must have the same number and order of (non-n/a) entries list column type: String
Contamination_Err	(modern) contamination estimate error list column type: String
Contamination_Meas	method to measure contamination, should be a software tool (ANGSD, Schmutzi, …) and the respective software versions, details should go to Contamination_Note list column type: String
Contamination_Note	arbitrary comments about the contamination estimate type: String
Genetic_Source_Accession_IDs	ENA or SRA Accession ID(s) pointing to the source data used to generate the genotyping data, if multiple are given they should be arranged by descending specificity (e.g. project id > sample id > sequencing run id) list column type: String
Primary_Contact	Project lead or first author type: String

.janno file variables *(continued)*

Variable	Description
Publication	bibtex key (e.g. "AuthorJournalYear") or "unpublished" list column type: String
Note	wildcard comments, e.g. note down aneuploidies here type: String
Keywords	arbitrary tags separated by ; (e.g. for custom sorting purposes) list column type: String

3 .ssf file variables

.ssf file variables

Variable	Description
poseidon_IDs	The Poseidon_IDs this sequencing entity corresponds to, from the Janno-file, multiple entries separated by ; list column type: String
udg	The kind of UDG treatment applied to the library for this sequencing entity type: String allowed values: minus; half; plus
library_built	The library preparation method applied for this sequencing entity (single- or double-stranded) type: String allowed values: ds; ss
sample_accession	The sample accession code as used in INSDC databases, including ENA and SRA (Example: SAMEA7050454) type: String
study_accession	The study accession code as used in INSDC databases, including ENA and SRA (Example: PRJEB39316) type: String
run_accession	The run accession code as used in INSDC databases, including ENA and SRA (Example: ERR4331996). This should be a unique identifier in most cases type: String
sample_alias	The sample alias defined by the submitter type: String

.ssf file variables *(continued)*

Variable	Description
secondary_sample_accession	A secondary sample accession, as used at the ENA for historical reasons (Example: ERS4811084) type: String
first_public	The date (YYYY-MM-DD) this sample was first made public type: Date
last_updated	The date (YYYY-MM-DD) this sample was last updated type: Date
instrument_model	The name of the instrument used (Example: Illumina HiSeq 2500) type: String
library_layout	The library layout of the sequencing entity (Example: SINGLE) type: String
library_source	The source of the DNA library (Example: GENOMIC) type: String
instrument_platform	The platform brand or type of the sequencer (Example: ILLUMINA) type: String
library_name	The ID of the library. Defaults to the library name the submitter has entered to the raw sequencing data repository. Data entries across which optical duplicates could exist should have matching library names. Can sometimes be useful to figure out which Poseidon_ID this entity belongs to. type: String
library_strategy	The strategy used to create the library (Example: WGS) type: String
fastq_ftp	The FTP-link(s) (URL) to the FASTQ file(s) (Example: ftp.sra.ebi.ac.uk/vol1/fastq/ERR433/009/ERR4332639/ERR4332639.fastq.gz) list column type: URL
fastq_aspera	The Aspera-link (URL) to the FASTQ-file(s). (Example: fasp.sra.ebi.ac.uk:/vol1/fastq/ERR433/009/ERR4332639/ERR4332639.fastq.gz) list column type: URL
fastq_bytes	The number of bytes of the FASTQ-file(s) in bytes list column type: Integer allowed range: 0 to Inf
fastq_md5	The MD5 hash(es) of the FASTQ-file(s) list column type: String

.ssf file variables *(continued)*

Variable	Description
read_count	The number of reads type: Integer allowed range: 0 to Inf
submitted_ftp	The URL(s) to the originally submitted file(s) before it got converted to FASTQ. This can sometimes be helpful for processing list column type: String