FAIR DATA MANAGEMENT USING pISA-TREE: STANDARD PROJECT DIRECTORY TREE

A. Blejec, Ž. Ramšak, M. Petek, Š. Baebler,
A. Coll, M. Zagorščak, K. Gruden

2018-11-12

https://github.com/NIB-SI/pISA-tree

Table of Contents

Introduction

This protocol describes a system for organisation of your experiments in the **F**indable, **A**ccessible, **I**nteroperable and **R**eusable (FAIR) manner - thus allowing integrative multiscale and multilevel analyses. It is set in accordance with ISA-tab standard and is compatible with FAIRDOM, using SEEK and JERM (Just Enough Results Model) frameworks as a basis.

To properly manage and annotate the data within the project one needs to design pISA project before the samples are collected. Thus, data management plan should be prepared when designing the experiments in parallel with the wet-lab experimental setup. This allows proper management of data storage resources, allocation of sufficient infrastructure for analyses, definition of vocabularies used for data annotation and exposes problems related with interoperability. So far this was not done in a systematic way as experiments were, in general, less complex and included mainly only one type of variables measured per one sample (e.g. only microarray analysis was done combined with limited qPCR analysis...). When dealing with complex experiments with data collected with multiple technologies (NGS, microarrays, qPCR, LC MS/MS, ...), for different molecular levels (mRNA, miRNA, proteins, metabolites, …), together with structural information (from Bright-field microscopes, EM, CCD cameras, MRI, Micro-CT, Cryo-soft X-ray, …) the data can only be properly analysed if organised in the way described below.

Data management is described in the **project data management plan** (prepared at the beginning of the project or when preparing proposal) and should be followed throughout the project. An example is given in the Annex 3.

Definitions

Sample – material collected in the experiment

The definition of a sample is a complex matter. It starts with the sample collection itself (Fig.1).

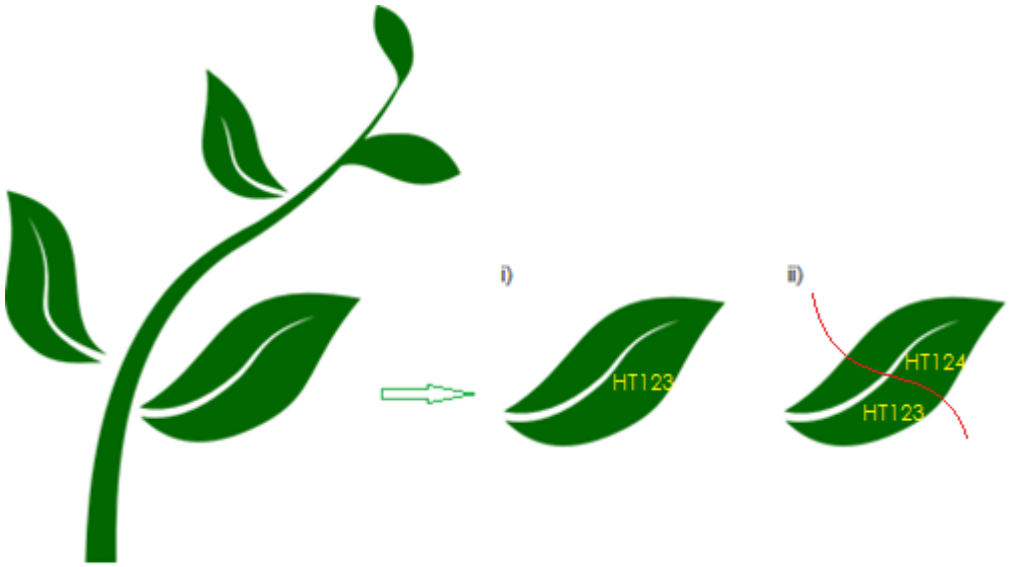

Fig.1: Sample collection example, showing assignment of sample IDs

i) If a full leaf would be homogenized and used in two techniques, transcriptomics and metabolomics, it would get the same ID (e.g. HT123).

ii) However, if the leaf would be first cut in half and then one part of the leaf would be used for transcriptomics and other part for metabolomics, leaf parts would get different identifiers (e.g. HT123 and HT124), meaning that one leaf may have more than one unique identifier, each of which identifies it for a different purpose. The system still has to allow us to extract the information that these two samples are from the same experiment, same plant and same leaf. This information is captured in phenodata file (see below).

Analyte – molecules extracted from the sample and analysed in an assay

Samples are further processed, e.g. RNA, DNA or proteins are isolated. If the same sample is used for isolation of different molecules or if something was wrong with procedure and the sample(s) need(s) to be processed again, then these **sample ID's** would be repeated several times and the traceability of our analysis would be lost (see Fig.2).

This is why we introduce the term **analyte**, for which we can create additional unique IDs that combine both the sample ID and the substance that was produced during analysis (e.g. HT123__RNA, HT123__cDNA, HT123__cDNA50x, see Fig.2). These additional analyte IDs are automatically created for wet lab assays of specific type (see subchapters Phenodata files and Assay).

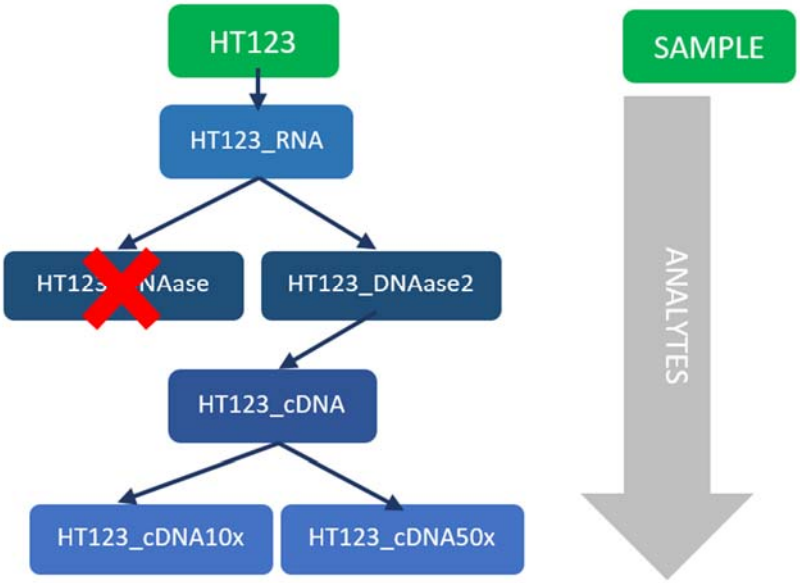

Fig.2: Example of Sample and Analyte IDs. From sample HT123, RNA was extracted, producing analyte HT123_RNA. This was treated with DNAse to produce HT123_DNAse. As the reaction was not successful, the step was repeated (HT123_DNAse2). The latter was used for dilutions (HT123_cDNA10x and HT123_cDNA50x), while HT123_DNAse was discarded.

Phenodata – master sample file, document with sample descriptions (see Fig.3)

Feature – measured variable, e.g. pH, microarray probe, gene expression, …

Featuredata – list of features measured in the experiment, with some descriptions etc (see Fig.3)

Sample ID – an unique identifier determining a sample, usually in a short alphanumerical form e.g. HT123 (see Fig.2 and Fig.3)

Analyte ID – an unique identifier determining a substance analysed in the assay, usually in a short alphanumerical form e.g. HT123_RNA (see Fig.2)

Feature ID – a measured variable identifier, e.g. probe ID, gene ID, qPCR amplicon ID, … (see Fig.3)

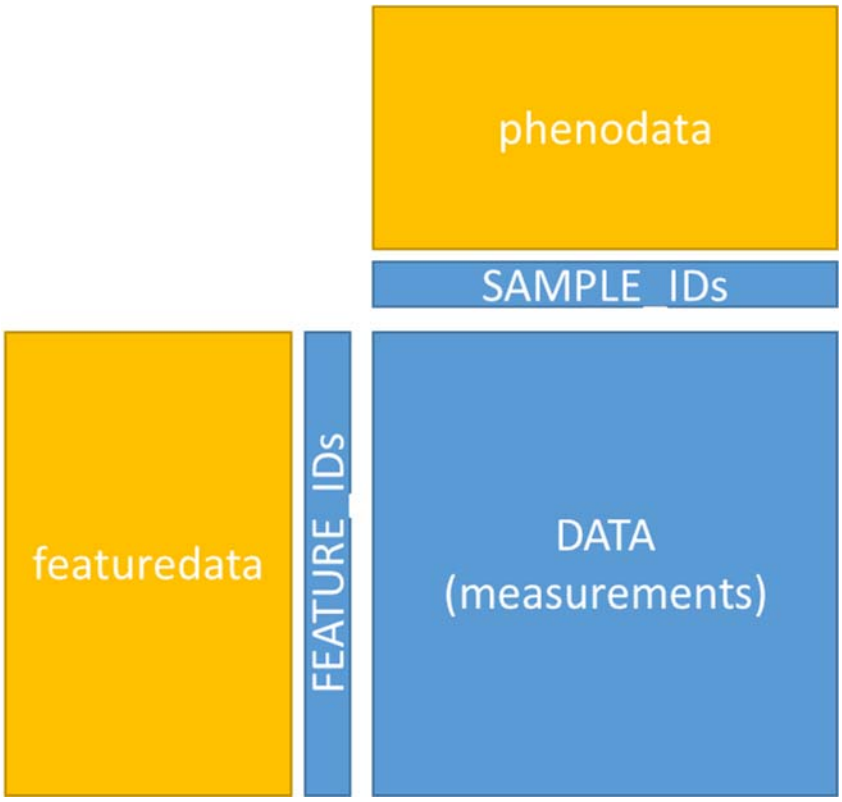

Fig.3: Schematic overview of relationship between phenodata and featuredata in high-throughput experiments.

Minimimal information about experiment – description of general information on how experiment (assay) was performed that provides us enough information for reproduction of experiment

Metadata – information related to pISA-tree layers. Metadata about samples is encompassed in phenodata and metadata about features in featuredata. Metadata of assay should be structured according to recommended 'minimal information about experiment' standard.

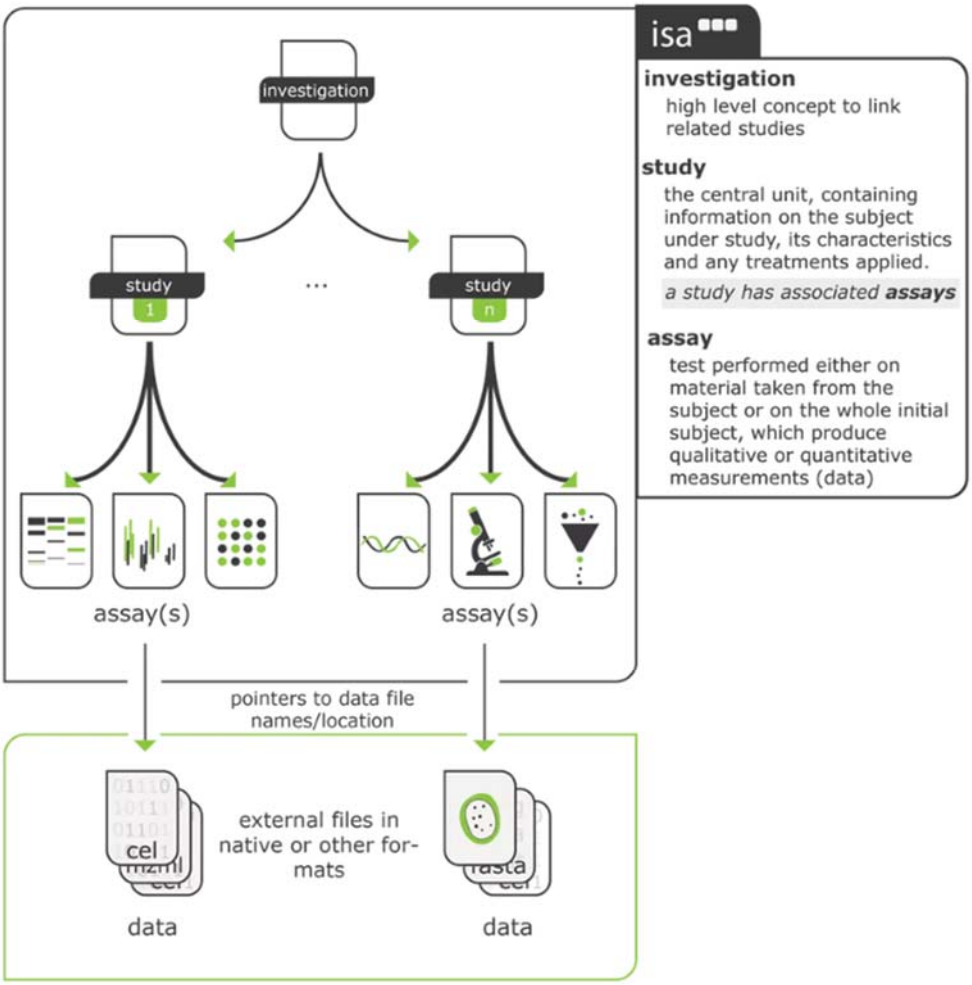

Fig.4: Schematic overview of the ISA-tab standard layers (Sansone *et al.,* Nat Genetics 2012) that are also used as levels of the pISA directory tree.

ISA layers – investigation/study/assay layers. A hierarchical data structure for linking and storing experimental data and metadata (see Fig.4).

pISA-tree layer ID – a short name of project/investigation/study/assay layers; usually acronyms or abbreviations are used.

What is pISA-tree?

pISA-tree provides a set of batch files (script files with extension .bat) that are used to create a standard directory tree for research projects. Batch scripts are executable on Microsoft Windows operating systems (OS) via Command Prompt (cmd). For Linux/Unix-like OS first install Wine. Command-line access in Wine is similar to Windows cmd and is invoked by typing wine cmd in the terminal.

Detailed instructions for installation are given in README.MD file on GitHub (https://github.com/NIB-SI/pISA).

pISA layers

You have to create a local folder (root directory), which will serve as the top pISA-tree layer and will contain your future projects (on Fig.5 it is called **pISA_projects**). The root directory contains the **makeProject.bat** batch file whereas batch files for creating other levels are stored in the Templates folder (see Fig.5). Appropriate batch files are automatically copied into newly created layers.

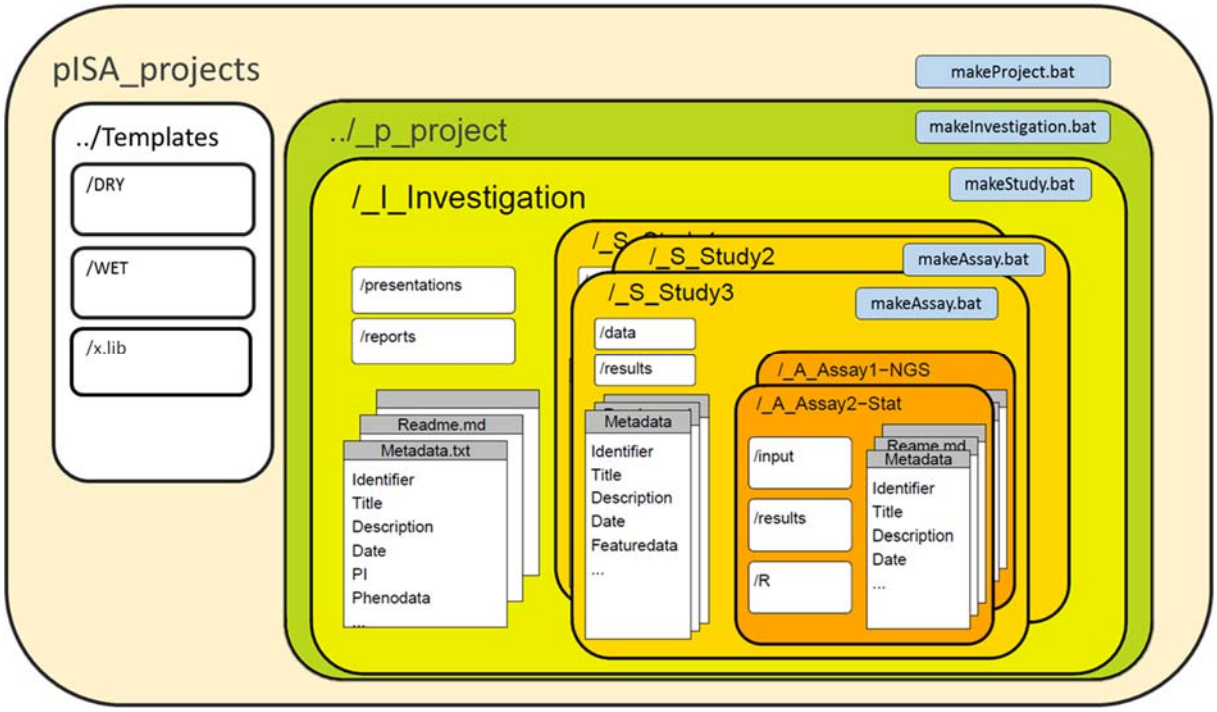

Fig.5: pISA-tree layers with corresponding subfolders and files that are automatically generated by batch scripts.

Project is organised as a collection of one or more **investigations**. An **investigation** is similarly organised as a collection of one or more **studies**. Each **study** has its own collection of one or more **assays**. Assays, either wet-lab or dry-lab, can be of specific type (e.g. MicroArray, NGS, Modelling, Statistical Analysis, ...) and are structured accordingly.

Here are some examples for each level of the pISA directory tree:

pISA-tree	Description	Layer ID example
Project		Lesions
Investigation	*The high-level concept to bring together related studies.* Contains the Master sample table (named phenodata)	HT *(stands for Hormonal treatments)*
Study	*The central layer, containing information on the subject under study, its characteristics and any treatments applied.* One biological experiment; e.g. batch of plants in the growth chamber, batch of fermentations performed in parallel, … Exceptionally, a new study should be defined also when you are integrating data between studies. If the integration is within a study itself, then new assays are generated.	ser1_treat1_stu ser1_ treat2_stu ser2_treat1_ath
Assay	*Test performed either on material taken from the subject or on the overall initial subject, which produces qualitative or quantitative measurements (data).* One test; it can be a batch of chips, qPCR plates…. This can be organised according to the researcher's preference. Wet-lab and dry-lab assays have different features to assist the researcher and consequently different structure.	PLT001 May18 ANOVA3 linRegM

Main batch files

- **makeproject.bat** - makes a new **project** directory tree
- **makeInvestigation.bat** - makes a new **investigation** directory tree (subdirectory tree within the **project**)
- **makeStudy.bat** - makes a new **study** (subdirectory tree within the **investigation**)
- **makeAssay.bat** - makes a new **assay** (subdirectory tree within the **study**)

Creation of the directory tree

The directory tree is a way to enforce the subordination of pISA layers. To emphasize the layer type, directory names are constructed automatically using the standard prefix and short layer ID. Standard prefixes are:

- _p_ for project
- _I_ for investigation
- _S_ for study
- _A_ for assay

project

To create a new project, run (double click) the file **makeProject.bat** and enter the project ID (short project name without spaces and special characters; see Annex 2). This will make a directory tree, metadata files and a local copy of **makeInvestigation.bat**. Short project name (ID), automatically prefixed with _p_, is used as the name of the directory. For example, if you set the project ID as blah the project directory name will automatically become _p_blah.

Note: If a Windows Defender warning message appears while executing batch files, click 'More info' and then 'Run anyway'. You have to do this only once for each batch file.

Investigation

To create a new investigation, run the file **makeInvestigation.bat** and enter the investigation ID (short investigation name without spaces and special characters; see Annex 2). This will make a directory tree, metadata files and a local copy of **makeStudy.bat**. Short investigation name (ID), automatically prefixed with _I_, is used as the name of the directory. The investigation directory name for the investigation bleh will be _I_bleh.

Study

To create a new study, run the **makeStudy.bat** and enter the study ID (short study name without spaces and special characters; see Annex 2). This will make a directory tree with several standard folders, metadata and auxiliary files and a local copy of **makeAssay.bat**. The study folder name will be _S_bloh for a study with short name (ID) bloh.

Note: New study should be initiated for each new batch of samples that is going to be collected!

Assay

Analyses for each study are stored in the folder of that study. To make a new assay, run the **makeAssay.bat** file.

First, you will be asked to choose between the assay wet- or dry-lab **Class**:

- Wet-lab e.g. measurements on the biological material (MicroArray, RNA-seq, qPCR, ...)
- Dry-lab e.g. process data (Statistics, Modelling, data integration, ...)

Second, you will enter the assay **Type** (i.e. RNAisol, qPCR, RNA-seq, GC-MS) and assay ID (Short name, for example RNA1). Short assay name and type (separated by '-' and prefixed by _A_) are used as the name of the assay directory tree (for example: _A_RNA1-RNAisol). The structure of subfolders and automatically generated files in shown in Fig.5.

Note: defining the assay type can bring in some additional item descriptors that can help you with data management of particular experiment. Each assay type, however, has to be carefully designed. For wet-lab, RNA isolation coupled with DNase treatment and reverse transcription (RNAisol) is available. The user can manually create new assay types as explained in Annex 4 or create them on-the-fly by selecting Other from the batch script menu.

Third, you will be is asked to choose the phenodata file that will be used in the particular assay (see Phenodata).

When creating either of these levels a certain folder structure is created. Descriptions of generated subfolders and required files are given in the table below:

Folder	INVESTIGATION	STUDY	ASSAY wet-lab	ASSAY dry-lab
Level-specific files	**_INVESTIGATION_METADATA.TXT** master sample table: **phenodata_yyyymmdd.txt** feature summary table e.g. **FST.txt** **README.MD**	**_STUDY_METADATA.TXT** **README.MD**	**_ASSAY_METADATA.TXT** following specific minimal standards feature table e.g. **featuredata.txt** analyte table e.g. **analytes.txt** **README.MD**	**_ASSAY_METADATA.TXT** following specific minimal standards and paths to phenodata_yyyymmdd.txt, analytes.txt and featuredata.txt **README.MD**
/input	NG	NG	NG	input data for the analysis
/reports	project files (e.g. applications) which make it easier to understand the investigation or to report to the outside world	protocols for sample handling (e.g. treatments, plant growth) prior to any assay	assay specific protocols	Protocol, documented procedure of the analysis and tools used in the assay (e.g. Markdown reports with embedded code)
/scripts	NG	NG	NG	scripts, algorithmic pipelines, macros (series of commands), … used in the assay
/output	NG	NG	results of the assay	results of the assay
/output/raw	NG	NG	original data collected from the source (machine)	NG
/presentations	all presentation prepared summarising results and experimental design within investigation	NG	NG	NG
/other	NG	NG	for anything not included elsewhere	for anything not included elsewhere

NG – directory not generated on this level

Metadata files

Several metadata files need to be prepared by users or are automatically generated by pISA-tree.

pISA level metadata files

Each level has a ***_LEVEL_METADATA.TXT*** file, a file with additional information needed to describe the experiment with enough information to be reproducible. This metadata files are tabulator-delimited text that list informative items for specific pISA levels in two columns:

1. item name (ended by a colon)
2. item value

Item value can be some text, for example investigator's name or longer study description, analysis description etc., or path to the phenodata file. Each item pair in the metadata should be typed in one line. Be careful if the metadata contains prime symbol (', as in 5'), it is better to spell it out, like 5-prime. For other unfavoured characters see Annex 2.

Two examples of the metadata entries are given below (tab character is shown as right arrow →):

Investigator: →Miha Mihav
Phenodata: →./phenodata_20181010.txt

When starting new project, investigation, study or assay, pISA-tree will guide you through the questionnaire to collect the required metadata.

README.MD files

At each pISA level a dummy **readme.md** file is automatically generated. These are free-form text files and can be used to make notes that explain the content of the directory, changes made to files etc.

Common.ini files

When running batch file to create a new project, investigation, study or assay, the user is asked to enter basic metadata (as described above). Some metadata are however identical for all studies and assays within one investigation (or similar for other pISA-tree layers). To avoid multiple entry of these metadata with every new study, user can enter such information into **common.ini** file. This file is created as a dummy file in pISA-tree root directory and will be automatically copied to newly created pISA-tree levels. This file contains following content:

Principal investigator:→*
License:→Creative Commons Attribution 4.0
Sharing permission:→Private
Upload to FAIRDOMHub:→Yes

The last three lines are related to synchronisation with FAIRDOMHub and need to be filled in if you plan to synchronise your pISA-tree with FAIRDOMHub automatically. License options are listed here: https://docs.seek4science.org/help/user-guide/licenses.html.

The **common.ini** files should be modified by the user to enter fields and metadata that are fixed for a particular project/investigation/study/assay (e.g. the principal investigator name, contact address etc). Information in this files will be automatically appended to metadata files for all subordinated pISA-tree levels.

*Note: In computer sciences *.ini files usually contain initial values and settings, thus here this file extension is used.*

_LEVEL_METADATA.txt files and **common.ini** file are plain text files. You can open and edit them with any text editor (e.g. Notepad++, WordPad, ConTEXT, Nano, …), Excel, OpenOffice Calc, … at any time (not just when starting a new level in pISA-tree). In some text editors the tab character that is separating item names and item values might be invisible. You can visualise it by enabling the "show symbols" or "show all characters" option. If you use Excel, the file will be presented in two columns and might be more readable and easier to edit. In this case, do not forget to save the file opened in Excel as Text (Tab delimited) file and do not change its name nor extension (.txt or .ini).

Phenodata files

Phenodata files (the name of the file originates from the golden age of microarrays) are tabulator delimited text files that describe your **samples**. Sometimes they are also referred to as Master sample files. Phenodata files are created so that they contain date of creation (eg. phenodata_20181010. txt; see Note2) and are **stored in the Investigation folder**. Every start of new Study is related to the collection of new samples in wet lab. Already before starting the real experiment (e.g. growing plants), one should create a phenodata file together with the basic pISA-tree structure.

Column headers of the phenodata file are partially prescribed, but any additional columns that might help to better describe collected samples can be added:

All samples used in an investigation must have unique sample IDs (unique keys), which are a combination of the two-letter study acronym (e.g. HT for hormonal treatments) and a three-digit number, e.g. HT001–HT999. By definition, unique sample ID means that within the same phenodata file there will not be two distinct rows that have the same values of sample ID.

Besides sample ID, which is always in the first column, phenodata file should contain Sample Name (longer and more descriptive). Further columns should contain sample descriptions, like for example: time after start (day 1, 2, 3, ...), treatment data (mock, PVY, …), genotype (NT, coi1, NahG, ...), position of the sample on the plant (upper leaf, ...) and any further information you consider relevant for analyses and reproducibility. When creating these descriptions you should not use any spaces nor special characters (see Annex 2).

An empty phenodata file is automatically generated when you create a new pISA-tree investigation.

Note1: to allow computer readability of phenodata, allowing for easy automatic integration of results, standard vocabulary should be used when filling in the phenodata file. Standard vocabulary means that you always use the same word for the same description. The 'same word' here means to be consistent using the uppercase/lowercase combinations, hyphens, underscores and other appropriate characters. Information on minimal information to be entered can be found in various standards (see Annex 1).

*Note2: sometimes **corrections of the phenodata files** are needed, for example when a sample is misannotated or is misclassified. To allow traceability of all dry lab analyses, when the phenodata is changed, it should be saved with the new date. In this way we can trace which phenodata file was used for which assay and repeat the analysis, if necessary. Note that opening a new study and adding new samples to the phenodata is not considered as a correction.*

*Note3: for any type of analyses, especially dry-lab assays, it is very practical to add info in which assays which samples should be used. Prior to the creation of the assay level (using **makeAssay.bat**) the user should add a column (to the right of the last existing column) into the phenodata file marking the samples that will be analysed in that particular assay. The column name should be*

AssayID of the planned assay and the samples should be marked by adding '1' into the corresponding cell.

An additional feature is offered in pISA-tree helping you in analyses/reporting on the level of assays. When **makeAssay.bat** *file is executed, user will be asked to enter the standard and assay specific items (metadata). The* **Analytes.txt** *file (see example in the table below) will be generated at the Assay level combining the information about the samples in the phenodata file and information entered as metadata:*

SampleID	Sample Name	Homogenisation_ protocol	Operator	Date_Homo genisation	RNA_ID	ng/ul	…
SMP_001	Leaf 1	Rneasy_Plant	Bob	24. 04. 2018	RNA_SMP_001		…
SMP_003	Leaf 3	Rneasy_Plant	Bob	24. 04. 2018	RNA_SMP_003		..
SMP_007	Leaf 7	Rneasy_Plant	Bob	24. 04. 2018	RNA_SMP_007		..

Table3: Example of Analytes.txt file for the RNA isolation Assay. Columns 1-6 were generated automatically by the script, whereas the other content (results of the analysis) has to be entered by the user.

Featuredata files

Featuredata file (i.e. annotation file) lists and describes the features (e.g. gene, metabolite) measured in a particular assay (biological experiment). Besides the unique IDs (e.g. geneID, metaboliteID, …), the file that describes the features also provides additional information about that features (e.g. short name, description, Gene Ontology terms, EC, MapMan Bin, …), any technical issue (e.g. specificity problem, quantification problems), etc.

The file should be created or downloaded (**.gal** file in the case of microarrays, **.gff** file in case of RNASeq, …). The file should be prepared in a tab delimited format where the first column contains list of all features and is named *featureID* (see also Note 3 below), followed by any number of columns that give improved knowledge and understanding of the feature.

Note 1: Although the annotation files can be quite complex, they have to contain at least two columns: featureID and Description.

Note 2: For microarray analysis this file normally includes also information on feature positions on the microarray which are provided by the manufacturer of the microarrays.

Note 3: For all transcriptomics (microarrays, NGS, qPCR) and proteomics experiments we will link the features to corresponding genes. Consequently, first column in the Annotation file should list GeneIDs and be named "geneID".

Auxiliary batch files

These files can help you with your data management issues but are not obligatory for FAIRness of your data:

showMetadata.bat

Collects all metadata files in a tree below the current level. Descriptions are typeset in either **METADATA.TXT** (plain text file) or **METADATA.MD** (plain text file in a markdown format; all text files can be edited by any text editor, e.g. Notepad, Wordpad or Excel and Word as long as they are saved as the text files. Use 'Open with' option to select the non-default program to open such data).

xcheckMetadata.bat

Checks all metadata files for missing required information (*) in a tree below the current level. Produces the file named **xCheckMetadata.md** which is similar to the one produced by showMetadata.bat but lists only lines with asterisks (*).

showTree.bat

List a directory tree below the current level in the file **TREE.TXT**.

update.bat

Replaces batch files in existing folder tree (all existing projects, investigations, studies and assays) with the updated versions from the root and x.lib subdirectory. After downloading an updated version of pISA-tree from GitHub, extract and replace all files in root and Templates directory. Run the update.bat file to update all batch files in all existing layers.

Annex 1: standards helping in setting up appropriate metadata files

Plenty of various platform dependent standards exist for the description of experimental data; consequently, all these standards are assay dependent (e.g. qPCR assay that involves sample preparation for it).

- ISA-TAB creator allows us to modify existing templates to suit our purposes or create new ones
- some of the existing templates: default ISA-TAB templates, MIAPPE template that improves on the phenotyping, for metabolomics: CIMR-MTBLS, MetaboLights; ScientificData templates
- pISA tree templates are stored in folder /Templates/WET/

- relevant minimum information standards (more info on FAIRsharing webpage):

Standard	ExperimentType	Description
MINSEQE	sequence reads	minimum information about a high throughput sequencing experiment
CIMR	metabolomics	core information for metabolomics; see MIAPE-MS too
MIQE	transcriptomics	minimum information for qPCR experiments
MIASE	models	minimum information about a simulation experiment
MIRIAM	models	minimum information required in the annotation of models
MIAPA	sequences	minimum information about a phylogenetic analysis
MIACA	-	minimum information about a cellular assay; high-throughput cell biological analyses (cells in culture); extension of minimum information captured by primary nucleotide sequence archives
MINI	electrophysiology	minimum information about a neuroscience investigation; electrophysiology
STRENDA	-	standards for reporting enzymology data guidelines
MIAPPE	Phenotype features	Minimum information about plant phenotyping

Annex 2: characters to avoid in directories, filenames, tables and IDs

Do not use any of these common illegal characters/symbols:

- # pound (hashtag)
- < left angle bracket
- $ dollar sign
- + plus sign
- % percent
- > right angle bracket
- ! exclamation point
- ^ circumflex accent
- & ampersand
- * asterisk
- ‘ single quotes
- | pipe
- { left bracket
- ? question mark
- “ double quotes
- = equal sign
- } right bracket
- / forward slash
- : colon
- \ backslash
- blank spaces
- @ at sign

Keep these rules in mind:

1. do not start or end your filename with a space, period, hyphen, or underline.
2. keep your filenames to a reasonable length
3. operating systems are case sensitive
4. avoid spaces and special characters (e.g. + and − are mathematical symbols).
5. do not use data types and keywords for table or column names, also do not pick names that will change meaning

Appropriate folder names:

- myDocuments
- my_Documents
- MyDocuments

Annex 3: An example of data management plan

ERA CoBioTech project INDIE data management plan

The FAIRDOM Research Infrastructure provides software, consultancy, and training to support research projects with the on-going management, retention and dissemination of their data, models and other assets. The FAIRDOM Platform provides a complete data management solution from raw data to publication that can be deployed in part or whole. FAIRDOM operates the FAIRDOMHub.org, a public, centrally managed resource with project controlled spaces, a guarantee of retention until 2029, integration with modelling tools and support for reproducible publications with leading journals.

We will be subcontracting our data management services to FAIRDOM, specifically to HITZ which is one of the partners in FAIRDOM Association e.V.. In Agreement with FAIRDOM we will run parts of the data management at Ljubljana subcontracting only small parts to FAIRDOM, mainly in regards to implementation of additional features into the infrastructure if required for the project (e.g setup assays tailored for synthetic biology experiments, allow easy transfer of data from local data management system to FAIRDOM). Also HITZ will provide training for project members and advice on structuring and annotating the data types, and optimising data handling pipelines within the project; and model curation services. Such combined data management activity is part of an effort of NIB to become a member of the FAIRDOM infrastructure team.

The role of DMP Responsible Person in this project will be taken by Tjaša Stare, who has extensive modelling and transcriptomics experience and experience in data management according to FAIR principles. Moreover, the role as PAL (project and area liaison) will be taken by Maria Suarez Diez (SSB), Katarina Cankar (WPR) and the newly employed postdocs at partners Axxence and UNIBI to actively involve all partners generating data within the project. DMP Responsible Person will represent the project at the annual DMP Responsible Person meetings and PALs will join her on at least one semi-annual PAL meeting. Funding for travel to the three DMP Responsible Person meetings (1 night each) and to the six PAL meetings (2 nights each) is covered in the budget of each partner hosting DMP responsible person or PAL. Each partner has 0.5 months whilst NIB as coordinator has 3 months per year dedicated to data management.

During the kick-off meeting of this project and during each of the regular project meetings, a session on data management will be organized by PALs and DMP Responsible Person to coordinate curation and uploading of selected data sets to FAIRDOMHub. Additional virtual conferences will be organized every 6 months in between face-to-face meetings to check if DMP needs updating and if data curation is proceeding as planned.
This project will generate experimental data (fermentation data, transcriptomics data, metabolomics data and productivity data) which will be used to iteratively refine the genome-scale metabolic model and regulatory model of *Corynebacterium*. Simulation of the models will be performed to predict the effect of planned metabolic engineering. Models for construction and optimisation of orthogonal circuits for indole production will additionally be generated and tested through simulations.
Experimental data and models will be generated and stored in accordance with ISA format (investigation-study-assay levels), provided within FAIRDOMHub. Sample and experiment metadata file will be recorded with the sample collection/experiments using unique identifiers, standardized across the consortium or using pre-existing standards to enable easy sharing and allow interlinking of datasets (via SEEK templates and other means). Similarly, models and model simulation results will be stored and linked with the corresponding experimental data used for simulations and validation following the ISA structure. All SOPs and scripts, as well as data

annotation files, will be catalogued on the fly locally, within the local data management system pISA developed to provide easy data management for local experimentalists and modelers. Local servers will be used to store raw data at place of source, i.e. where they are generated. All partners have servers with sufficient storage capacity and computation power to perform planed tasks. Data uploading to FAIRDOMHub will be performed by each partner immediately after the analysis is done to allow interlinking of all generated datasets and exchange of data between partners. Detailed information describing the basics data management including the formats and ontologies used for dataset description are available in table below:

	Metabolic models	Regulatory models	Orthogonal circuits models	Simulations	Constructs/ bricks	Novel strains	Transcriptomics	Metabolomics	Fermentation data
generated by	SSB	NIB	SSB	SSB,NIB	WPR, UNIBI	WPR, UNIBI	UNIBI	WPR	UNIBI, AXXENCE
format of generated data	n.a.	n.a.	n.a.	n.a.	fasta	txt	fastq		txt
raw data storage at	SSB, GitLab	NIB	SSB, GitLab	SSB,NIB	WPR, UNIBI	WPR, UNIBI	UNIBI,GEO,SSR	WPR, Metabo Lights	UNIBI, AXXENCE
expected data size	<1GB	<1GB	<1GB	<1GB	<1GB	<1GB	<500GB	100 GB	<1GB
analysed by	SSB,NIB	NIB	SSB	all	WPR, UNIBI	WPR, UNIBI	UNIBI; NIB	WPR	all
minimal information requirements	MIRIAM	MIRIAM	MIRIAM	MIASE	MIRIAM		MIAME	CIMR	
standards, formats	SBML	SBGN,SBML	SBGN,SBML	SEDML	SBOL data and visual		MAGE-ML		
ontolologies and vocabularies used	SBO	SBO	SBO		SBOL		GO, KEGG		ENVO
scripts stored at	GitLab, FAIRDOM	FAIRDOM	GitLab, FAIRDOM	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.

This table emphasizes that we are going to use MIBBI standards for data exchange. Details in using the MIBBI Standards will be defined in the first 6 months of the project. The **raw** data will be stored locally at the partners and catalogued at the FAIRDOMHub, the **processed, analyzed** data, together with SOPs will be stored in the FAIRDOMHub.

Data transfer: We have agreed to transfer all data together with all needed metadata between partners. The same applies to the transfer into FAIRDOMHub. We expect some of the raw datasets to be too large to be effectively stored into FAIRDOMHub. These data will be *catalogued, i.e.* links to local servers hosting the data will be stored in the FAIRDOMHub. Data transfer integrity will be assessed using MD5 checksums.

Versioning of models, scripts and SOPs will be performed according to the data management policies of each partner and catalogued within FAIRDOMHub.

Long term storage is for the analysed data together with metadata, SOPs and scripts in interlinked form planned in FAIRDOMHub. Appropriate raw data will be deposited to public databases such as GEO and SSR for transcriptomics data and MetaboLights for metabolomics data when decided according to IP requirements.

Data curation: DMP responsible person will create INDIE project within FAIRDOMHub and coordinate upload and curation of assets

Data accessibility: Different levels of data accessibility will be managed through FAIRDOMHub. The project consortium will vote on data clearing taking into account all relevant legal, ethical and

IP issues in a Data Management session of a project meeting. For all data that needs ethical / legal clearance, the take responsibility for the data created on their premises. Afterwards both PAL and DMP Responsible Person serve as gatekeepers to manage accessibility of data in FAIRDOMHub in a two-step fashion. Unlocking by PALs guarantees that all technical standards for DOI-assignable storage and publication at FAIRDOMHub have been taken care of, while unlocking by DMP Responsible Person guarantees that all legal, ethical and IP requirements relevant to the data sets to be published has been fulfilled.
Data set updating and curation will be handled continuously, but decisions on update releases will be taken as described before for data uploading to FAIRDOMHub.

Annex 4: Developer tools

Here some additional features of pISA-tree app are listed which are not applicable for the standard user, but more for the ones that would like to extend it.

Adding new Assay Classes and Types

Subdirectories within assay directory trees, for different Classes, differ slightly, according to the need of the specific **Class**. Assay classes and types are defined as subdirectories of the Templates directory. An example is "../Templates/Wet/RNAisol". For this example, the directory name defines the assay **Class** as "Wet" and the subdirectory name assay **Type** as "RNAisol". To add another class, create directory *myclass* within Template directory: ../Templates/myclass. To add another type of, for example *Wet*-lab assay (here named *mytype*), create it on-the-fly by selecting Other from the batch script menu or create a new subdirectory within appropriate Class directory with the name *mytype*: ../Templates/Wet/mytype .

In addition to the basic items, one can also use assay specific items, depending on the assay type. The assay specific items are pre-specified in the **meta_AType_Template.txt** and **Analytes_Template.txt** files, placed within the appropriate *Class/Type* subdirectories. The **makeAssay.bat** batch file will accordingly generate questions (if any) to add information to assay metadata file. The **meta_AType_Template.txt** and **Analytes_Template.txt** files are specific for each used assay Type in your system.

The **meta_AType_Template.txt** and **Analytes_Template.txt** files are plain text files. Each line represents one "Item name - Item value" pair, separated by the tabulator character (illustrated below as the right arrow →). The first of the pair – "Item name", will appear as the assay specific question during the assay creation. The second of the pair – "Item value", will be either offered or has to be entered manually.

An example of the **Analytes_Template.txt** file:

Item name →Item value
Isolation Protocol →Rneasy_Plant/ZymoRNA
Operator →John/Bob/Katja/Anna
Date Homogenisation→%today%
RNA ID → RNA_$
ng/ml → Blank

The **meta_AType_Template.txt** or **Analytes_Template.txt** will not need to be tackled with by standard user of pISA-tree application.

The user will be asked about the assay specific items (defined by assay specific **Analytes_Template.txt** file) when running **makeAssay.bat** and those will be included in the **_ASSAY_METADATA** file. In addition, they are used as the assay specific description of samples used in an assay and are automatically added as the assay specific extension to the phenodata file. Assay specific metadata will be copied into columns of the **Analytes.txt** file, which contains information about the samples used in the assay.

Syntax rules in item value part are used for support of choices in menu-like data entry. This reduces errors in spelling, spacing, and use of the character case.

Fields with one or more choices

Item value choices, if more than one, are separated by the slash (/) character. See the example above for the items named Isolation Protocol and Operator. To select the operator name, a simple menu will be presented to the user:

1 John
2 Bob
3 Katja
4 Anna
5 Other

User will use numbers (1 to 5) to select the name to use. The last line ("Other") is automatically added and enables ad-hoc addition of any new choice. If the choice is likely to occur in future, it can be added into the analytes.ini file.

Date field

Date fields are considered in the same way as ordinary choice fields. Special bookmark %today% will be replaced by current date in a data entry menu.

Sample ID replacement

New sample related identification codes are sometimes needed. Sample ID can be automatically inserted in the place of a dollar character ($) to form new IDs. In example above for the field RNA ID and Sample names HT123, HT124, HT125 one would get new IDs: HT123_RNA, HT124_RNA and HT125_RNA.

Blank fields

The word Blank as item value signals the column that has to be left blank in the Analytes.txt file.

	File name	Add/edit items	Entry	Used for	Included in:
Assay type specific	**meta_AType_Template.txt** in AssayType folder	Yes	Typed during creation	Assays of this type	Metadata file
Analytes specific	**Analytes_Template.txt** in AssayType folder	Yes	Typed during creation	Assays of this type	Metadata file and Analytes.txt

Annex 5: Upload to FAIRDOMHub

Assay levels include **upload.bat** batch file and **ignore.txt**.