
AltamISA Documentation

Release 0.2.9+0.g2b0ade8.dirty

Core Unit Bioinformatics

Apr 29, 2021

1 Quick Example	3
2 Features	5
3 Special Extensions	7
4 Frequently Asked Questions	9
Python Module Index	49
Index	51

AltamISA is a Python 3 library for representing the [ISA-tools data model](#) and reading and writing [ISA-Tab file format](#). The documentation is split into three parts (accessible through the navigation on the left):

Installation & Getting Started Instructions for the installation of the module and some examples to get you started.

API Documentation This section contains the API documentation for the module

Project Info More information on the project, including the changelog, list of contributing authors, and contribution instructions.

CHAPTER 1

Quick Example

Start parsing an ISA-Tab dataset by reading and validating an [investigation file](#) (download it to your working directory first):

```
from altamisa.isatab import *

with open("i_investigation.txt", "rt") as investigation_file:
    investigation = InvestigationReader.from_stream(investigation_file).read()

InvestigationValidator(investigation).validate()
```

For more inspiration on how to use AltamISA, see [Examples](#).

CHAPTER 2

Features

The main features are

- immutable data structures (named tuples) for representing records,
- simple implementation as directed acyclic graph (DAG) between ISA material and process nodes,
- strictly validating parser with various sanity checks,
- well-tested code, and well-documented API,
- example applications, e.g., conversion of ISA-tab to Graphviz dot.

CHAPTER 3

Special Extensions

In addition to the original ISA-Tab format specifications, AltamISA supports the following special modifications to improve specific use cases:

- **List of values** in `Characterics` or `Parameter Value` fields by using semicolon-separators (“;”). Note, for ontology terms the same number of splits is expected in the associated field `Term Source REF` and `Term Accession Number`.
- **Material name** `Library Name` for improved library annotation in nucleotide sequencing assays.

Note: Although these modifications have been discussed by the ISA community ([list of values](#); [library name](#)), they are not supported by official ISA software, yet.

Frequently Asked Questions

Why another Python library for ISA-tab? Attempting to use the official Python `isa-api` package led to quite some frustration on our side. Even the official ISA-tab examples parsed into non-expected graph structures. Further, the official Python API has several features that were irrelevant for us, e.g., conversion from and to various other formats.

Is validation a big deal? Yes it is. A lot of ISA-tab files that we found out in the wild while exploring the model and format were not validating. (We're looking at you, MetaboLights). The aim of the projects that we are using ISA-tab are not just describing experiments so humans can read the experiment descriptions. Rather, we want to have machine readable (and interpretable) data formats. Here, strict syntax and ideally semantic validation is key.

What's the state? The ISA standard and ISA-Tab file I/O is fully implemented, tested, and we're actively using it in other applications. The validation is also working stably yet we are planning several extensions and additional checks.

What's the aim? The aim is to have a stable and correct library for parsing, representing, and writing ISA-tab.

What's on the roadmap? Mostly fine-tuning, stabilization, and additional validations. At some point we might want to add support for ISA-JSON but that is a secondary aim at the moment. The advantage of ISA-Tab is that you can edit it with spreadsheet application.

4.1 Installation

4.1.1 Using pip

To install AltamISA, run this command in your terminal:

```
$ pip install altamisa
```

This is the preferred method to install AltamISA, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

4.1.2 Using conda

If you like [conda](#) as much as we do, you can install AltamISA from the [Bioconda](#) channel. This assumes that you have already setup conda and the Bioconda channel [as described in the Bioconda manual](#).

```
$ conda install altamisa
```

4.1.3 From sources

The sources for AltamISA can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/bihealth/altamisa
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/bihealth/altamisa/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

4.2 Getting Started

After installation, you can use AltamISA in your project simply by importing the module.

```
from altamisa.isatab import *
```

That's all, continue and look at the list of examples.

4.3 Examples

To run the examples, download a full [test dataset](#) (all files) from the AltamISA repository to your working directory.

Import AltamISA (if not already done) and other required modules.

```
from altamisa.isatab import *
import os
import warnings
```

4.3.1 Parsing and validation

AltamISA provides separate functions to parse, validate and export ISA-tab investigation, study or assay files, respectively. Investigation, study and assay files may be parsed independently from each other. Reading and validating a single investigation file is performed as follows:

```
# Parse and validate an investigation file
with open("i_investigation.txt", "rt") as investigation_file:
    investigation = InvestigationReader.from_stream(investigation_file).read()

InvestigationValidator(investigation).validate()
```

For study and assay parsing, unique ids need to be set to enable unambiguous identification of study , assay and, in particular, their nodes in later applications (such as a complete graph creation):

```
# Parse ad study and an assay file
with open("s_BII-S-1.txt", "rt") as inputf:
    study = StudyReader.from_stream("S1", inputf).read()

with open("a_transcriptome.txt", "rt") as inputf:
    assay = AssayReader.from_stream("S1", "A1", inputf).read()
```

However, in real use cases an ISA-Tab dataset contains related investigation, study and assay files and thus should be handled as unit. In particular, validation of studies and assays requires information from parent elements such as the investigation or study, respectively. Thus, joint parsing and validation of a complete ISA-Tab dataset may look like this:

```
# Read investigation
with open("i_investigation.txt", "rt") as investigation_file:
    investigation = InvestigationReader.from_stream(investigation_file).read()

# Validate investigation
InvestigationValidator(investigation).validate()

# Read studies and assays
path_in = os.path.normpath(os.path.dirname("i_investigation.txt"))
studies = {}
assays = {}
for s, study_info in enumerate(investigation.studies):
    if study_info.info.path:
        with open(os.path.join(path_in, study_info.info.path), "rt") as inputf:
            studies[s] = StudyReader.from_stream("S{}".format(s + 1), inputf).read()
    if study_info.assays:
        assays[s] = {}
        for a, assay_info in enumerate(study_info.assays):
            if assay_info.path:
                with open(os.path.join(path_in, assay_info.path), "rt") as inputf:
                    assays[s][a] = AssayReader.from_stream(
                        "S{}".format(s + 1), "A{}".format(a + 1), inputf
                    ).read()

# Validate studies and assays
for s, study_info in enumerate(investigation.studies):
    if study_info.info.path:
        StudyValidator(investigation, study_info, studies[s]).validate()
    for a, assay_info in enumerate(study_info.assays):
        if assay_info.path:
            AssayValidator(investigation, study_info, assay_info, assays[s][a]).
                validate()
```

4.3.2 Writing

Having a set of AltamISA investigation, studies and assays available as parsed above, the models can be given out as ISA-Tab files again as follows (make sure to not use the same path for the investigation, as files might be overwritten otherwise):

```
# Write investigation
path_out = "/tmp/altamisa_example/"
os.makedirs(path_out, exist_ok=True)
with open(
    os.path.join(path_out, "i_investigation.txt"), "wt", newline=""
) as output_investigation_file:
    InvestigationWriter.from_stream(investigation, output_investigation_file).write()

# Write studies and assays
for s, study_info in enumerate(investigation.studies):
    if study_info.info.path:
        with open(os.path.join(path_out, study_info.info.path), "wt", newline="") as _
        ↳outputf:
            StudyWriter.from_stream(studies[s], outputf).write()
        for a, assay_info in enumerate(study_info.assays):
            if assay_info.path:
                with open(os.path.join(path_out, assay_info.path), "wt", newline="") as _
                ↳outputf:
                    AssayWriter.from_stream(assays[s][a], outputf).write()
```

4.3.3 Working with AltamISA warnings

Parsing, validating and writing of ISA-Tab files may result in AltamISA warnings, if format or data is not conform to the specifications (except for AltamISA's *Special Extensions*). Warnings will not stop AltamISA from parsing an ISA-Tab dataset into a technically valid model. However, any AltamISA warning should be reported to the user to allow him to improve or correct his ISA-Tab files. Furthermore, we discourage from accepting and working with ISA-Tab datasets which result in warnings of the category *CriticalIsaValidationWarning*.

Warnings may be collected as follows, to enable joint notification or evaluation:

```
# Show all warnings of same type and content
warnings.simplefilter("always")

# Collect warnings
with warnings.catch_warnings(record=True) as records:
    # Work with ISA-Tab files here, e.g.:
    InvestigationValidator(investigation).validate()

# Print warnings
for record in records:
    warnings.showwarning(
        record.message, record.category, record.filename, record.lineno, record.line
    )
```

4.3.4 Manual ISA model creation

A minimal AltamISA ISA model may be created as follows. To create more comprehensive models with, for instance, more filled investigation sections, material or process information, please refer to the description of the *Models*.

```

def create_and_write(out_path):
    """Create an investigation with a study and assay and write to ``output_path``."""

    # Prepare one or more study sections
    # Prepare basic study information
    study_info = models.BasicInfo(
        path="s_minimal.txt",
        identifier="s_minimal",
        title="Germline Study",
        description=None,
        submission_date=None,
        public_release_date=None,
        comments=(
            models.Comment(name="Study Grant Number", value=None),
            models.Comment(name="Study Funding Agency", value=None),
        ),
        headers=[],
    )

    # Create one or more assays
    assay_01 = models.AssayInfo(
        measurement_type=models.OntologyTermRef(
            name="exome sequencing assay",
            accession="http://purl.obolibrary.org/obo/OBI_0002118",
            ontology_name="OBI",
        ),
        technology_type=models.OntologyTermRef(
            name="nucleotide sequencing",
            accession="http://purl.obolibrary.org/obo/OBI_0000626",
            ontology_name="OBI",
        ),
        platform=None,
        path="a_minimal.txt",
        comments=(),
        headers=[],
    )

    # Prepare one or more protocols
    protocol_01 = models.ProtocolInfo(
        name="sample collection",
        type=models.OntologyTermRef(name="sample collection"),
        description=None,
        uri=None,
        version=None,
        parameters={},
        components={},
        comments=(),
        headers=[],
    )
    protocol_02 = models.ProtocolInfo(
        name="nucleic acid sequencing",
        type=models.OntologyTermRef(name="nucleic acid sequencing"),
        description=None,
        uri=None,
        version=None,
        parameters={},
        components={},
    )

```

(continues on next page)

(continued from previous page)

```

        comments=(),
        headers=[],
    )

    # Create study object
    study_01 = models.StudyInfo(
        info=study_info,
        designs=(),
        publications=(),
        factors={},
        assays=(assay_01,),
        protocols={protocol_01.name: protocol_01, protocol_02.name: protocol_02},
        contacts=(),
    )

    # Prepare other investigation section
    # Prepare one or more ontology term source references
    onto_ref_01 = models.OntologyRef(
        name="OBI",
        file="http://data.bioontology.org/ontologies/OBI",
        version="31",
        description="Ontology for Biomedical Investigations",
        comments=(),
        headers=[],
    )

    # Prepare basic investigation information
    invest_info = models.BasicInfo(
        path="i_minimal.txt",
        identifier="i_minimal",
        title="Minimal Investigation",
        description=None,
        submission_date=None,
        public_release_date=None,
        comments=(),
        headers=[],
    )

    # Create investigation object
    investigation = models.InvestigationInfo(
        ontology_source_refs={onto_ref_01.name: onto_ref_01},
        info=invest_info,
        publications=(),
        contacts=(),
        studies=(study_01,),
    )

    # Validate investigation
    InvestigationValidator(investigation).validate()

    # Write the investigation as ISA-Tab txt file
    with open(os.path.join(out_path, investigation.info.path), "wt", newline="") as _
↪outputf:
        InvestigationWriter.from_stream(investigation=investigation, output_
↪file=outputf).write()

    # Create a corresponding Study graph

```

(continues on next page)

(continued from previous page)

```

# Create at least on source, one sample and one collection process
# Unique names are required for unambiguous node identification
source_01 = models.Material(
    type="Source Name",
    unique_name="S1-source-0815",
    name="0815",
    extract_label=None,
    characteristics=(),
    comments=(),
    factor_values=(),
    material_type=None,
    headers=[table_headers.SOURCE_NAME],
)

sample_01 = models.Material(
    type="Sample Name",
    unique_name="S1-sample-0815-N1",
    name="0815-N1",
    extract_label=None,
    characteristics=(),
    comments=(),
    factor_values=(),
    material_type=None,
    headers=[table_headers.SAMPLE_NAME],
)

process_01 = models.Process(
    protocol_ref="sample collection",
    unique_name="S1-sample collection-2-1",
    name=None,
    name_type=None,
    date=None,
    performer=None,
    parameter_values=(),
    comments=(),
    array_design_ref=None,
    first_dimension=None,
    second_dimension=None,
    headers=[table_headers.PROTOCOL_REF],
)

# Create the arcs to connect the material and process nodes, referenced by the_
↪ unique name
arc_01 = models.Arc(tail="S1-source-0815", head="S1-sample collection-2-1")
arc_02 = models.Arc(tail="S1-sample collection-2-1", head="S1-sample-0815-N1")

# Create the study graph object
study_graph_01 = models.Study(
    file=investigation.studies[0].info.path,
    header=None,
    materials={source_01.unique_name: source_01, sample_01.unique_name: sample_01}
↪,
    processes={process_01.unique_name: process_01},
    arcs=(arc_01, arc_02),
)

```

(continues on next page)

(continued from previous page)

```

# Validate study graph
StudyValidator(
    investigation=investigation, study_info=investigation.studies[0], study=study_
↪graph_01
    ).validate()

# Write the study as ISA-Tab txt file
with open(
    os.path.join(out_path, investigation.studies[0].info.path), "wt", newline=""
) as outputf:
    StudyWriter.from_stream(study_or_assay=study_graph_01, output_file=outputf).
↪write()

# Create a corresponding Assay graph

# Create at least on samples, one output material and one collection process
# Unique names are required for unambiguous node identification
# Explicit header definition per node is currently required to enable export to_
↪ISA-Tab
sample_01 = models.Material(
    type="Sample Name",
    unique_name="S1-sample-0815-N1",
    name="0815-N1",
    extract_label=None,
    characteristics=(),
    comments=(),
    factor_values=(),
    material_type=None,
    headers=[table_headers.SAMPLE_NAME],
)

data_file_01 = models.Material(
    type="Raw Data File",
    unique_name="S1-A1-0815-N1-DNA1-WES1_L???_???_R1.fastq.gz-COL4",
    name="0815-N1-DNA1-WES1_L???_???_R1.fastq.gz",
    extract_label=None,
    characteristics=(),
    comments=(),
    factor_values=(),
    material_type=None,
    headers=[table_headers.RAW_DATA_FILE],
)

data_file_02 = models.Material(
    type="Raw Data File",
    unique_name="S1-A1-0815-N1-DNA1-WES1_L???_???_R2.fastq.gz-COL5",
    name="0815-N1-DNA1-WES1_L???_???_R2.fastq.gz",
    extract_label=None,
    characteristics=(),
    comments=(),
    factor_values=(),
    material_type=None,
    headers=[table_headers.RAW_DATA_FILE],
)

process_01 = models.Process(
    protocol_ref="nucleic acid sequencing",

```

(continues on next page)

(continued from previous page)

```

        unique_name="S1-A1-0815-N1-DNA1-WES1-3",
        name="0815-N1-DNA1-WES1",
        name_type="Assay Name",
        date=None,
        performer=None,
        parameter_values=(),
        comments=(),
        array_design_ref=None,
        first_dimension=None,
        second_dimension=None,
        headers=[table_headers.PROTOCOL_REF, table_headers.ASSAY_NAME],
    )

    # Create the arcs to connect the material and process nodes, referenced by the
    ↪unique name
    arcs = (
        models.Arc(tail="S1-sample-0815-N1", head="S1-A1-0815-N1-DNA1-WES1-3"),
        models.Arc(
            tail="S1-A1-0815-N1-DNA1-WES1-3",
            head="S1-A1-0815-N1-DNA1-WES1_L???_???_R1.fastq.gz-COL4",
        ),
        models.Arc(
            tail="S1-A1-0815-N1-DNA1-WES1_L???_???_R1.fastq.gz-COL4",
            head="S1-A1-0815-N1-DNA1-WES1_L???_???_R2.fastq.gz-COL5",
        ),
    )

    # Create the assay graph object
    assay_graph_01 = models.Assay(
        file=investigation.studies[0].assays[0].path,
        header=None,
        materials={
            sample_01.unique_name: sample_01,
            data_file_01.unique_name: data_file_01,
            data_file_02.unique_name: data_file_02,
        },
        processes={process_01.unique_name: process_01},
        arcs=arcs,
    )

    # Validate assay graph
    AssayValidator(
        investigation=investigation,
        study_info=investigation.studies[0],
        assay_info=investigation.studies[0].assays[0],
        assay=assay_graph_01,
    ).validate()

    # Create output path, if not existing
    if not os.path.exists(out_path):
        os.makedirs(out_path, exist_ok=True)

    # Write the assay as ISA-Tab txt file
    with open(
        os.path.join(out_path, investigation.studies[0].assays[0].path), "wt",
        ↪newline=""
    ) as outputf:

```

(continues on next page)

(continued from previous page)

```
AssayWriter.from_stream(study_or_assay=assay_graph_01, output_file=outputf) .
↪ write()
```

4.4 Parsers

Classes to read data from ISA-Tab investigation, study and assay files and create the corresponding AltamISA models. Note that parsers can issue warnings, see [Validators](#) on how to capture them properly.

Contents

- *Parsers*
 - *altamisa.isatab.InvestigationReader*
 - *altamisa.isatab.AssayReader*
 - *altamisa.isatab.AssayRowReader*
 - *altamisa.isatab.StudyReader*
 - *altamisa.isatab.StudyRowReader*

4.4.1 altamisa.isatab.InvestigationReader

class altamisa.isatab.**InvestigationReader** (*input_file: TextIO, filename=None*)

Main class to read an investigation file into an InvestigationInfo object.

Parameters *input_file (TextIO)* – ISA-Tab investigation file

classmethod **from_stream** (*input_file: TextIO, filename=None*)

Construct from file-like object

read () → altamisa.isatab.models.InvestigationInfo

Read the investigation file

Return type models.InvestigationInfo

Returns Investigation model including all information from the investigation file

4.4.2 altamisa.isatab.AssayReader

class altamisa.isatab.**AssayReader** (*study_id: str, assay_id: str, input_file: TextIO, filename=None*)

Read an ISA-TAB assay file (a_*.txt) into a Assay object.

This is the main facade class for reading assay objects. Prefer it over using the more low-level code.

Parameters

- **study_id** (*str*) – Unique identifier for the study, needed to disambiguate nodes between files.
- **assay_id** (*str*) – Unique identifier for the assay, needed to disambiguate nodes between files.

- **input_file** (*TextIO*) – ISA-Tab assay file

classmethod from_stream (*study_id: str, assay_id: str, input_file: TextIO, filename=None*)
Construct from file-like object

read()
Parse the assay file

Return type `models.Assay`

Returns Assay model including graph of material and process nodes

4.4.3 altamisa.isatab.AssayRowReader

class `altamisa.isatab.AssayRowReader` (*study_id: str, assay_id: str, input_file: TextIO, filename: str*)
Read an ISA-TAB assay file (`a_*.txt`) into a tabular/object representation.

This is a more low-level part of the interface. Please prefer using `:py:AssayReader:` over using this class.

Parameters

- **study_id** (*str*) – Unique identifier for the study, needed to disambiguate nodes between files.
- **assay_id** (*str*) – Unique identifier for the assay, needed to disambiguate nodes between files.
- **input_file** (*TextIO*) – ISA-Tab assay file

classmethod from_stream (*study_id: str, assay_id: str, input_file: TextIO, filename: str = None*)
Construct from file-like object

read()
Read assays rows

Returns Nodes per row of the assay file

4.4.4 altamisa.isatab.StudyReader

class `altamisa.isatab.StudyReader` (*study_id: str, input_file: TextIO, filename=None*)
Read an ISA-TAB study file (`s_*.txt`) into a Study object.

This is the main facade class for reading study objects. Prefer it over using the more low-level code.

Parameters

- **study_id** (*str*) – Unique identifier for the study, needed to disambiguate nodes between files.
- **input_file** (*TextIO*) – ISA-Tab study file

classmethod from_stream (*study_id: str, input_file: TextIO, filename=None*)
Construct from file-like object

read()
Parse the study file

Return type `models.Study`

Returns Study model including graph of material and process nodes

4.4.5 altamisa.isatab.StudyRowReader

class altamisa.isatab.StudyRowReader (*study_id: str, input_file: TextIO, filename: str*)

Read an ISA-TAB study file (*s_*.txt*) into a tabular/object representation.

This is a more low-level part of the interface. Please prefer using `:py:StudyReader:` over using this class.

Parameters

- **study_id** (*str*) – Unique identifier for the study, needed to disambiguate nodes between files.
- **input_file** (*TextIO*) – ISA-Tab study file

classmethod from_stream (*study_id: str, input_file: TextIO, filename: str = None*)

Construct from file-like object

read ()

Read the study rows

Returns Nodes per row of the study file

4.5 Writers

Classes to write AltamISA model files to ISA-Tab investigation, study and assay files. Note that writers can issue warnings, see [Writers](#) on how to capture them properly.

Contents

- [Writers](#)
 - [altamisa.isatab.InvestigationWriter](#)
 - [altamisa.isatab.AssayWriter](#)
 - [altamisa.isatab.StudyWriter](#)

4.5.1 altamisa.isatab.InvestigationWriter

class altamisa.isatab.InvestigationWriter (*investigation: altamisa.isatab.models.InvestigationInfo, output_file: TextIO, quote=None, lineterminator=None*)

Main class to write an investigation file from an `InvestigationInfo` object.

Parameters

- **investigation** (*models.InvestigationInfo*) – The investigation model to write
- **output_file** (*TextIO*) – Output ISA-Tab investigation file
- **quote** (*str*) – Optional quoting character (none by default)
- **lineterminator** (*str*) – Optional line terminator (OS specific by default)

```
classmethod from_stream(investigation: altamisa.isatab.models.InvestigationInfo, output_file:
                        TextIO, quote=None, lineterminator=None)
    Construct from file-like object

write()
    Write investigation file
```

4.5.2 altamisa.isatab.AssayWriter

```
class altamisa.isatab.AssayWriter(study_or_assay: NamedTuple, output_file: TextIO,
                                quote=None, lineterminator=None)
    Class that writes a file from an Assay object.
```

Parameters

- **study_or_assay** (*models.Assay*) – The assay model to write
- **output_file** (*TextIO*) – Output ISA-Tab assay file
- **quote** (*str*) – Optional quoting character (none by default)
- **lineterminator** (*str*) – Optional line terminator (OS specific by default)

4.5.3 altamisa.isatab.StudyWriter

```
class altamisa.isatab.StudyWriter(study_or_assay: NamedTuple, output_file: TextIO,
                                quote=None, lineterminator=None)
    Class that writes a file from an Study object.
```

Parameters

- **study_or_assay** (*models.Study*) – The study model to write
- **output_file** (*TextIO*) – Output ISA-Tab study file
- **quote** (*str*) – Optional quoting character (none by default)
- **lineterminator** (*str*) – Optional line terminator (OS specific by default)

4.6 Validators

Classes to validate the integrity of the investigation model as well as study and assay graphs with respect to ISA-Tab format specifications.

AltamISA uses the [Python warnings module](#) for reporting validation warnings.

```
# Parse investigation and validate, capture warnings in `ws`.
with open("investigation.tsv", "rt") as inputf:
    with warnings.catch_warnings(record=True) as warnings:
        isa_inv = InvestigationReader.from_stream(input_file=inputf).read()
        InvestigationValidator(isa_inv).validate()

# Iterate over all captured warnings and handle them.
for warning in warnings:
    somehow_handle(warning)
```

Note: You can use the *IsaWarning* class hierarchy for getting severity information.

Contents

- *Validators*
 - *altamisa.isatab.InvestigationValidator*
 - *altamisa.isatab.AssayValidator*
 - *altamisa.isatab.StudyValidator*

4.6.1 altamisa.isatab.InvestigationValidator

class altamisa.isatab.InvestigationValidator (*investigation:* *altamisa.isatab.models.InvestigationInfo*)

Validator for Investigation

Parameters *investigation* (*models.InvestigationInfo*) – The investigation model to validate

validate ()
Validate the investigation

4.6.2 altamisa.isatab.AssayValidator

class altamisa.isatab.AssayValidator (*investigation:* *altamisa.isatab.models.InvestigationInfo*,
study_info: *altamisa.isatab.models.StudyInfo*, *assay_info:* *altamisa.isatab.models.AssayInfo*, *assay:*
altamisa.isatab.models.Assay, *parent_study:* *altamisa.isatab.models.Study = None*)

Validator for Assay

Parameters

- **investigation** (*models.InvestigationInfo*) – The corresponding investigation model
- **study_info** (*models.StudyInfo*) – The corresponding study information
- **assay_info** (*models.AssayInfo*) – The corresponding assay information
- **assay** (*models.Assay*) – The Assay model to validate
- **parent_study** (*models.Study*) – Optional: The parent Study of the current Assay (for extended validation)

validate ()
Validate the assay

4.6.3 altamisa.isatab.StudyValidator

```
class altamisa.isatab.StudyValidator (investigation: altamisa.isatab.models.InvestigationInfo,
                                     study_info: altamisa.isatab.models.StudyInfo, study:
                                     altamisa.isatab.models.Study)
```

Validator for Study

Parameters

- **investigation** (*models.InvestigationInfo*) – The corresponding investigation model
- **study_info** (*models.StudyInfo*) – The corresponding study information
- **study** (*models.Study*) – The Study model to validate

4.7 Models

Class models for storing and representing ISA data, with particular focus on ISA-Tab compatibility. The modeling follows the structure of the specifications with different classes for each file type (investigation, study, assay), investigation sections, the different study and assay column types etc. In particular, study and assay data (i.e. corresponding materials and processes) are represented by use of directed acyclic graphs.

Note that all models are *immutable* after construction. Here is a common pattern for getting a *copy* with modifying select members.

```
>>> import attr
>>> from altamisa.isatab import Comment
>>> c1 = Comment(name="Name", value="the value")
>>> c1
Comment(name='Name', value='value')
>>> c2 = Comment(**{**attr.asdict(c1), "name": "Another Name"})
>>> c2
Comment(name='Another Name', value='value')
```

Contents

- *Models*
 - *altamisa.isatab.AnnotatedStr*
 - *altamisa.isatab.OntologyTermRef*
 - *altamisa.isatab.Comment*
 - *altamisa.isatab.OntologyRef*
 - *altamisa.isatab.BasicInfo*
 - *altamisa.isatab.PublicationInfo*
 - *altamisa.isatab.ContactInfo*
 - *altamisa.isatab.DesignDescriptorsInfo*
 - *altamisa.isatab.FactorInfo*
 - *altamisa.isatab.AssayInfo*

```
– altamisa.isatab.ProtocolComponentInfo
– altamisa.isatab.ProtocolInfo
– altamisa.isatab.StudyInfo
– altamisa.isatab.InvestigationInfo
– altamisa.isatab.Characteristics
– altamisa.isatab.FactorValue
– altamisa.isatab.ParameterValue
– altamisa.isatab.Material
– altamisa.isatab.Process
– altamisa.isatab.Arc
– altamisa.isatab.Study
– altamisa.isatab.Assay
```

4.7.1 altamisa.isatab.AnnotatedStr

class altamisa.isatab.**AnnotatedStr** (*value*, ***kwargs*)

A str that can be flagged with values.

Example Usage

```
>>> x = AnnotatedStr("text", key1="value1", key2=2)
>>> x
"text"
>>> x.key1
"value1"
>>> x.key2
2
```

4.7.2 altamisa.isatab.OntologyTermRef

class altamisa.isatab.**OntologyTermRef** (*name: str = None*, *accession: str = None*, *ontology_name: str = None*)

Reference to a term into an ontology.

Can be either initialized with

- all three of a *name*, an *accession*, and an *ontology_name*
- only a *name*
- nothing (empty)

name = None

Human-readable name of the term

accession = None

The accession of the referenced term

ontology_name = None

Name of the ontology (links to `OntologyRef.name`)

4.7.3 altamisa.isatab.Comment

```
class altamisa.isatab.Comment (name: str, value: str)
    Representation of a Comment [*] cell.

    name = None
        Comment name

    value = None
        Comment value
```

4.7.4 altamisa.isatab.OntologyRef

```
class altamisa.isatab.OntologyRef (name: str, file: str, version: str, description: str, com-
                                   ments: Tuple[altamisa.isatab.models.Comment], headers:
                                   List[str])
    Description of an ontology term source, as used for investigation file.

    name = None
        The name of the ontology (e.g., CEBI)

    file = None
        Path to file or URI to ontology

    version = None
        Version of the ontology

    description = None
        Description of the ontology

    comments = None
        Comments

    headers = None
        Headers from/for ISA-tab parsing/writing
```

4.7.5 altamisa.isatab.BasicInfo

```
class altamisa.isatab.BasicInfo (path: pathlib.Path, identifier: str, title: str, description: str,
                                   submission_date: datetime.date, public_release_date: date-
                                   time.date, comments: Tuple[altamisa.isatab.models.Comment],
                                   headers: List[str])
    Basic metadata for an investigation or study (INVESTIGATION or STUDY).

    path = None
        Path to the investigation or study file

    identifier = None
        Investigation/Study identifier

    title = None
        Investigation/Study title

    description = None
        Investigation/Study description

    submission_date = None
        Investigation/Study submission date
```

public_release_date = None
Investigation/Study public release date

comments = None
Comments

headers = None
Headers from/for ISA-tab parsing/writing

4.7.6 altamisa.isatab.PublicationInfo

```
class altamisa.isatab.PublicationInfo (pubmed_id: str, doi: str, authors: str, title: str, status:
                                     Union[altamisa.isatab.models.OntologyTermRef, str],
                                     comments: Tuple[altamisa.isatab.models.Comment],
                                     headers: List[str])
```

Information regarding an investigation publication (INVESTIGATION PUBLICATIONS).

pubmed_id = None
Publication PubMed ID

doi = None
Publication DOI

authors = None
Publication author list string

title = None
Publication title

status = None
Publication status

comments = None
Comments

headers = None
Headers from/for ISA-tab parsing/writing

4.7.7 altamisa.isatab.ContactInfo

```
class altamisa.isatab.ContactInfo (last_name: str, first_name: str, mid_initial: str, email: str,
                                   phone: str, fax: str, address: str, affiliation: str, role:
                                   Union[altamisa.isatab.models.OntologyTermRef, str], com-
                                   ments: Tuple[altamisa.isatab.models.Comment], headers:
                                   List[str])
```

Investigation contact information

last_name = None
Last name of contact

first_name = None
First name of contact

mid_initial = None
Middle initial of contact

email = None
Email of contact

phone = None
Phone of contact

fax = None
Fax no. of contact

address = None
Address of contact

affiliation = None
Affiliation of contact

role = None
Role of contact

comments = None
Comments

headers = None
Headers from/for ISA-tab parsing/writing

4.7.8 altamisa.isatab.DesignDescriptorsInfo

```
class altamisa.isatab.DesignDescriptorsInfo (type: Union[altamisa.isatab.models.OntologyTermRef,
                                                         str],      comments:      Tuple[altamisa.isatab.models.Comment],
                                                         headers: List[str])
```

Study design descriptors information

type = None
Design descriptors type

comments = None
Comments

headers = None
Headers from/for ISA-tab parsing/writing

4.7.9 altamisa.isatab.FactorInfo

```
class altamisa.isatab.FactorInfo (name: str, type: Union[altamisa.isatab.models.OntologyTermRef,
                                                         str],  comments:  Tuple[altamisa.isatab.models.Comment],
                                                         headers: List[str])
```

Study factor information

name = None
Factor name

type = None
Factor type

comments = None
Comments

headers = None
Headers from/for ISA-tab parsing/writing

4.7.10 altamisa.isatab.AssayInfo

```
class altamisa.isatab.AssayInfo (measurement_type: Union[altamisa.isatab.models.OntologyTermRef,  
                                str], technology_type: Union[altamisa.isatab.models.OntologyTermRef,  
                                str], platform: str, path: pathlib.Path, comments: Tu-  
                                ple[altamisa.isatab.models.Comment], headers: List[str])  
  
    Study assay information  
  
    measurement_type = None  
        Assay measurement type  
  
    technology_type = None  
        Assay technology type  
  
    platform = None  
        Assay platform  
  
    path = None  
        Path to assay file  
  
    comments = None  
        Comments  
  
    headers = None  
        Headers from/for ISA-tab parsing/writing
```

4.7.11 altamisa.isatab.ProtocolComponentInfo

```
class altamisa.isatab.ProtocolComponentInfo (name: str, type: Union[altamisa.isatab.models.OntologyTermRef,  
                                             str])  
  
    Protocol component information  
  
    name = None  
        Protocol component name  
  
    type = None  
        Protocol component type
```

4.7.12 altamisa.isatab.ProtocolInfo

```
class altamisa.isatab.ProtocolInfo (name: str, type: Union[altamisa.isatab.models.OntologyTermRef,  
                                str], description: str, uri: str, version: str, parameters:  
                                Dict[str, Union[altamisa.isatab.models.OntologyTermRef,  
                                str]], components: Dict[str, alt-  
                                amisa.isatab.models.ProtocolComponentInfo], com-  
                                ments: Tuple[altamisa.isatab.models.Comment], headers:  
                                List[str])  
  
    Protocol information  
  
    name = None  
        Protocol name  
  
    type = None  
        Protocol type  
  
    description = None  
        Protocol
```

```

uri = None
    Protocol URI

version = None
    Protocol version

parameters = None
    Protocol parameters

components = None
    Protocol components

comments = None
    Comments

headers = None
    Headers from/for ISA-tab parsing/writing

```

4.7.13 altamisa.isatab.StudyInfo

```

class altamisa.isatab.StudyInfo(info: altamisa.isatab.models.BasicInfo, designs: Tu-
                                ple[altamisa.isatab.models.DesignDescriptorsInfo], pub-
                                lications: Tuple[altamisa.isatab.models.PublicationInfo],
                                factors: Dict[str, altamisa.isatab.models.FactorInfo], as-
                                says: Tuple[altamisa.isatab.models.AssayInfo], protocols:
                                Dict[str, altamisa.isatab.models.ProtocolInfo], contacts:
                                Tuple[altamisa.isatab.models.ContactInfo])

```

The full metadata regarding one study

```

info = None
    Basic study information

designs = None
    Study designs by name

publications = None
    Publication list for study

factors = None
    Study factors by name

assays = None
    Study assays

protocols = None
    Study protocols by name

contacts = None
    Study contact list

```

4.7.14 altamisa.isatab.InvestigationInfo

```

class altamisa.isatab.InvestigationInfo(ontology_source_refs: Dict[str, al-
                                          tamisa.isatab.models.OntologyRef], info: al-
                                          tamisa.isatab.models.BasicInfo, publications: Tu-
                                          ple[altamisa.isatab.models.PublicationInfo], con-
                                          tacts: Tuple[altamisa.isatab.models.ContactInfo],
                                          studies: Tuple[altamisa.isatab.models.StudyInfo])

```

Representation of an ISA investigation

ontology_source_refs = None
Ontologies defined for investigation

info = None
Basic information on investigation

publications = None
List of investigation publications

contacts = None
Contact list for investigation

studies = None
List of studies in this investigation

4.7.15 altamisa.isatab.Characteristics

```
class altamisa.isatab.Characteristics (name: str, value:
                                     List[Union[altamisa.isatab.models.OntologyTermRef,
                                     str]], unit: Union[altamisa.isatab.models.OntologyTermRef,
                                     str])
    Representation of a Characteristics[*] cell.

    name = None
        Characteristics name

    value = None
        Characteristics value

    unit = None
        Characteristics unit
```

4.7.16 altamisa.isatab.FactorValue

```
class altamisa.isatab.FactorValue (name: str, value: Union[altamisa.isatab.models.OntologyTermRef,
                                                         str], unit: Union[altamisa.isatab.models.OntologyTermRef,
                                                         str])
    Representation of a Factor Value[*] cell.

    name = None
        Factor name

    value = None
        Factor value

    unit = None
        Factor value unit
```

4.7.17 altamisa.isatab.ParameterValue

```
class altamisa.isatab.ParameterValue (name: str, value: List[Union[altamisa.isatab.models.OntologyTermRef,
                                                                    str]], unit: Union[altamisa.isatab.models.OntologyTermRef,
                                                                    str])
    Representation of a Parameter Value[*] cell.

    name = None
        Parameter name
```

value = None
Parameter value

unit = None
Parameter value unit

4.7.18 altamisa.isatab.Material

```
class altamisa.isatab.Material(type: str, unique_name: str, name: str, extract_label:
    Union[altamisa.isatab.models.OntologyTermRef, str], char-
    acteristics: Tuple[altamisa.isatab.models.Characteristics], com-
    ments: Tuple[altamisa.isatab.models.Comment], factor_values:
    Tuple[altamisa.isatab.models.FactorValue], material_type:
    Union[altamisa.isatab.models.OntologyTermRef, str], headers:
    List[str])
```

Representation of a Material or Data node.

type = None
The type of node (i.e. column name)

unique_name = None
The unique name of the material node. This is usually created with respect to study/assay and column.
The unique name is necessary to distinguish materials of different type with potential overlaps in names.
Otherwise graph representation might be incorrect (ambiguous arcs, loops) and the original relation of material and process not conclusively reproducible.

name = None
Original name of a material or data file

extract_label = None
The label of a Labeled Extract

characteristics = None
Material characteristics

comments = None
Material comments

factor_values = None
Material factor values

material_type = None
Material type

headers = None
Columns headers from/for ISA-tab parsing/writing

4.7.19 altamisa.isatab.Process

```
class altamisa.isatab.Process(protocol_ref: str, unique_name: str, name: str, name_type:
    str, date: datetime.date, performer: str, parame-
    ter_values: Tuple[altamisa.isatab.models.ParameterValue],
    comments: Tuple[altamisa.isatab.models.Comment],
    array_design_ref: str, first_dimension:
    Union[altamisa.isatab.models.OntologyTermRef, str], sec-
    ond_dimension: Union[altamisa.isatab.models.OntologyTermRef,
    str], headers: List[str])
```

Representation of a Process or Assay node.

protocol_ref = None

Referenced to protocol name from investigation

unique_name = None

The unique name of the process node. This is usually created with respect to study/assay and column. The unique name is necessary to distinguish processes of different protocols with potential overlaps in names. Otherwise graph representation might be incorrect (ambiguous arcs, loops) and the original relation of material and process not conclusively reproducible. When “Protocol REF” is given without a further qualifying name, this is generated from the protocol reference with an auto-incrementing number.

name = None

Original name of a process (e.g. from Assay Name etc.)

name_type = None

Type of original name (e.g. Assay Name)

date = None

Process date

performer = None

Performer of process

parameter_values = None

Tuple of parameters values

comments = None

Tuple of process comments

array_design_ref = None

Array design reference (special case annotation)

Technology types: “DNA microarray”, “protein microarray”

Protocol types: “nucleic acid hybridization”, “hybridization”

first_dimension = None

First dimension (special case annotation, INSTEAD of Gel Electrophoresis Assay Name)

Technology types: “gel electrophoresis”

Protocol types: “electrophoresis”

second_dimension = None

Second dimension (special case annotation, INSTEAD of Gel Electrophoresis Assay Name)

Technology types: “gel electrophoresis”

Protocol types: “electrophoresis”

headers = None

Columns headers from/for ISA-tab parsing/writing

4.7.20 altamisa.isatab.Arc

class altamisa.isatab.Arc (tail: str, head: str)

Representation of an arc between two Material and/or Process nodes.

tail = None

The arc’s tail name

head = None

The arc’s head name

4.7.21 altamisa.isatab.Study

```
class altamisa.isatab.Study (file:          pathlib.Path,      header:          Tuple,      materials:
                             Dict[str,      altamisa.isatab.models.Material], processes:
                             Dict[str,      altamisa.isatab.models.Process], arcs:          Tu-
                             ple[altamisa.isatab.models.Arc])
```

Representation of an ISA study.

file = None

Path to ISA study file

header = None

The study's header

materials = None

A mapping from material name to `Material` object (Data is a kind of material)

processes = None

A mapping from process name to `Process` object

arcs = None

The processing arcs

4.7.22 altamisa.isatab.Assay

```
class altamisa.isatab.Assay (file:          pathlib.Path,      header:          Tuple,      materials:
                             Dict[str,      altamisa.isatab.models.Material], processes:
                             Dict[str,      altamisa.isatab.models.Process], arcs:          Tu-
                             ple[altamisa.isatab.models.Arc])
```

Representation of an ISA assay.

file = None

Path to ISA assay file

header = None

The assay's header

materials = None

A mapping from material name to `Material` object (Data is a kind of material)

processes = None

A mapping from process name to `Process` object

arcs = None

The processing arcs

4.8 Constants

Constants are used internally to ensure consistent use of keywords when parsing and writing investigation, study and assay files (in particular in the headers). They can also be used when creating a model from scratch, i.e. to provide a desired header order (used for writing) when creating study/assay nodes or investigation sections (or section entries).

Contents

- [Constants](#)

- *Investigation Headers*
- *Table Headers*
- *Table Restrictions*
- *Table Tokens*

4.8.1 Investigation Headers

Constants of valid vertical row headers for investigation parsing and writing, as specified in the [ISA-Tab investigation specifications](#).

```
altamisa.constants.investigation_headers.ONTOLOGY_SOURCE_REFERENCE = 'ONTOLOGY SOURCE REFERENCE'
altamisa.constants.investigation_headers.INVESTIGATION = 'INVESTIGATION'
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATIONS = 'INVESTIGATION PUBLICATIONS'
altamisa.constants.investigation_headers.INVESTIGATION_CONTACTS = 'INVESTIGATION CONTACTS'
altamisa.constants.investigation_headers.STUDY = 'STUDY'
altamisa.constants.investigation_headers.STUDY_DESIGN_DESCRIPTOR = 'STUDY DESIGN DESCRIPTOR'
altamisa.constants.investigation_headers.STUDY_PUBLICATIONS = 'STUDY PUBLICATIONS'
altamisa.constants.investigation_headers.STUDY_FACTORS = 'STUDY FACTORS'
altamisa.constants.investigation_headers.STUDY_ASSAYS = 'STUDY ASSAYS'
altamisa.constants.investigation_headers.STUDY_PROTOCOLS = 'STUDY PROTOCOLS'
altamisa.constants.investigation_headers.STUDY_CONTACTS = 'STUDY CONTACTS'
altamisa.constants.investigation_headers.TERM_SOURCE_NAME = 'Term Source Name'
altamisa.constants.investigation_headers.TERM_SOURCE_FILE = 'Term Source File'
altamisa.constants.investigation_headers.TERM_SOURCE_VERSION = 'Term Source Version'
altamisa.constants.investigation_headers.TERM_SOURCE_DESCRIPTION = 'Term Source Description'
altamisa.constants.investigation_headers.ONTOLOGY_SOURCE_REF_KEYS = ('Term Source Name', 'Term Source File', 'Term Source Version', 'Term Source Description')
    Collected header keys of ONTOLOGY SOURCE REFERENCE section
altamisa.constants.investigation_headers.INVESTIGATION_IDENTIFIER = 'Investigation Identifier'
altamisa.constants.investigation_headers.INVESTIGATION_TITLE = 'Investigation Title'
altamisa.constants.investigation_headers.INVESTIGATION_DESCRIPTION = 'Investigation Description'
altamisa.constants.investigation_headers.INVESTIGATION_SUBMISSION_DATE = 'Investigation Submission Date'
altamisa.constants.investigation_headers.INVESTIGATION_PUBLIC_RELEASE_DATE = 'Investigation Public Release Date'
altamisa.constants.investigation_headers.INVESTIGATION_INFO_KEYS = ('Investigation Identifier', 'Investigation Title', 'Investigation Description', 'Investigation Submission Date', 'Investigation Public Release Date')
    Collected header keys of INVESTIGATION section
altamisa.constants.investigation_headers.INVESTIGATION_PUBMED_ID = 'Investigation PubMed ID'
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATION_DOI = 'Investigation Publication DOI'
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATION_AUTHOR_LIST = 'Investigation Publication Author List'
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATION_TITLE = 'Investigation Publication Title'
```

```
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATION_STATUS = 'Investigation
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATION_STATUS_TERM_ACCESSION_NUM
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATION_STATUS_TERM_SOURCE_REF =
altamisa.constants.investigation_headers.INVESTIGATION_PUBLICATIONS_KEYS = ('Investigation
    Collected header keys of INVESTIGATION PUBLICATIONS
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_LAST_NAME = 'Investigation Pe
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_FIRST_NAME = 'Investigation P
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_MID_INITIALS = 'Investigation
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_EMAIL = 'Investigation Person
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_PHONE = 'Investigation Person
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_FAX = 'Investigation Person P
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_ADDRESS = 'Investigation Pers
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_AFFILIATION = 'Investigation
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_ROLES = 'Investigation Person
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_ROLES_TERM_ACCESSION_NUMBER =
altamisa.constants.investigation_headers.INVESTIGATION_PERSON_ROLES_TERM_SOURCE_REF = 'Inve
altamisa.constants.investigation_headers.INVESTIGATION_CONTACTS_KEYS = ('Investigation Pers
    Collected header keys of INVESTIGATION CONTACTS section
altamisa.constants.investigation_headers.STUDY_IDENTIFIER = 'Study Identifier'
altamisa.constants.investigation_headers.STUDY_TITLE = 'Study Title'
altamisa.constants.investigation_headers.STUDY_DESCRIPTION = 'Study Description'
altamisa.constants.investigation_headers.STUDY_SUBMISSION_DATE = 'Study Submission Date'
altamisa.constants.investigation_headers.STUDY_PUBLIC_RELEASE_DATE = 'Study Public Release
altamisa.constants.investigation_headers.STUDY_FILE_NAME = 'Study File Name'
altamisa.constants.investigation_headers.STUDY_INFO_KEYS = ('Study Identifier', 'Study Titl
    Collected header keys of STUDY section
altamisa.constants.investigation_headers.STUDY_DESIGN_TYPE = 'Study Design Type'
altamisa.constants.investigation_headers.STUDY_DESIGN_TYPE_TERM_ACCESSION_NUMBER = 'Study D
altamisa.constants.investigation_headers.STUDY_DESIGN_TYPE_TERM_SOURCE_REF = 'Study Design
altamisa.constants.investigation_headers.STUDY_DESIGN_DESCR_KEYS = ('Study Design Type', 'S
    Collected header keys of STUDY DESIGN DESCRIPTORS section
altamisa.constants.investigation_headers.STUDY_PUBMED_ID = 'Study PubMed ID'
altamisa.constants.investigation_headers.STUDY_PUBLICATION_DOI = 'Study Publication DOI'
altamisa.constants.investigation_headers.STUDY_PUBLICATION_AUTHOR_LIST = 'Study Publication
altamisa.constants.investigation_headers.STUDY_PUBLICATION_TITLE = 'Study Publication Title
altamisa.constants.investigation_headers.STUDY_PUBLICATION_STATUS = 'Study Publication Stat
altamisa.constants.investigation_headers.STUDY_PUBLICATION_STATUS_TERM_ACCESSION_NUMBER =
```

```
altamisa.constants.investigation_headers.STUDY_PUBLICATION_STATUS_TERM_SOURCE_REF = 'Study
altamisa.constants.investigation_headers.STUDY_PUBLICATIONS_KEYS = ('Study PubMed ID', 'St
    Collected header keys of STUDY PUBLICATIONS section
altamisa.constants.investigation_headers.STUDY_FACTOR_NAME = 'Study Factor Name'
altamisa.constants.investigation_headers.STUDY_FACTOR_TYPE = 'Study Factor Type'
altamisa.constants.investigation_headers.STUDY_FACTOR_TYPE_TERM_ACCESSION_NUMBER = 'Study
altamisa.constants.investigation_headers.STUDY_FACTOR_TYPE_TERM_SOURCE_REF = 'Study Factor
altamisa.constants.investigation_headers.STUDY_FACTORS_KEYS = ('Study Factor Name', 'Study
    Collected header keys of STUDY FACTORS section
altamisa.constants.investigation_headers.STUDY_ASSAY_FILE_NAME = 'Study Assay File Name'
altamisa.constants.investigation_headers.STUDY_ASSAY_MEASUREMENT_TYPE = 'Study Assay Measur
altamisa.constants.investigation_headers.STUDY_ASSAY_MEASUREMENT_TYPE_TERM_ACCESSION_NUMBER
altamisa.constants.investigation_headers.STUDY_ASSAY_MEASUREMENT_TYPE_TERM_SOURCE_REF = 'St
altamisa.constants.investigation_headers.STUDY_ASSAY_TECHNOLOGY_TYPE = 'Study Assay Techno
altamisa.constants.investigation_headers.STUDY_ASSAY_TECHNOLOGY_TYPE_TERM_ACCESSION_NUMBER
altamisa.constants.investigation_headers.STUDY_ASSAY_TECHNOLOGY_TYPE_TERM_SOURCE_REF = 'St
altamisa.constants.investigation_headers.STUDY_ASSAY_TECHNOLOGY_PLATFORM = 'Study Assay Tec
altamisa.constants.investigation_headers.STUDY_ASSAYS_KEYS = ('Study Assay File Name', 'St
    Collected header keys of STUDY ASSAYS section
altamisa.constants.investigation_headers.STUDY_PROTOCOL_NAME = 'Study Protocol Name'
altamisa.constants.investigation_headers.STUDY_PROTOCOL_TYPE = 'Study Protocol Type'
altamisa.constants.investigation_headers.STUDY_PROTOCOL_TYPE_TERM_ACCESSION_NUMBER = 'Study
altamisa.constants.investigation_headers.STUDY_PROTOCOL_TYPE_TERM_SOURCE_REF = 'Study Proto
altamisa.constants.investigation_headers.STUDY_PROTOCOL_DESCRIPTION = 'Study Protocol Descr
altamisa.constants.investigation_headers.STUDY_PROTOCOL_URI = 'Study Protocol URI'
altamisa.constants.investigation_headers.STUDY_PROTOCOL_VERSION = 'Study Protocol Version'
altamisa.constants.investigation_headers.STUDY_PROTOCOL_PARAMETERS_NAME = 'Study Protocol
altamisa.constants.investigation_headers.STUDY_PROTOCOL_PARAMETERS_NAME_TERM_ACCESSION_NUM
altamisa.constants.investigation_headers.STUDY_PROTOCOL_PARAMETERS_NAME_TERM_SOURCE_REF =
altamisa.constants.investigation_headers.STUDY_PROTOCOL_COMPONENTS_NAME = 'Study Protocol
altamisa.constants.investigation_headers.STUDY_PROTOCOL_COMPONENTS_TYPE = 'Study Protocol
altamisa.constants.investigation_headers.STUDY_PROTOCOL_COMPONENTS_TYPE_TERM_ACCESSION_NUM
altamisa.constants.investigation_headers.STUDY_PROTOCOL_COMPONENTS_TYPE_TERM_SOURCE_REF =
altamisa.constants.investigation_headers.STUDY_PROTOCOLS_KEYS = ('Study Protocol Name', 'St
    Collected header keys of STUDY PROTOCOLS section
altamisa.constants.investigation_headers.STUDY_PERSON_LAST_NAME = 'Study Person Last Name'
altamisa.constants.investigation_headers.STUDY_PERSON_FIRST_NAME = 'Study Person First Name'
```

```

altamisa.constants.investigation_headers.STUDY_PERSON_MID_INITIALS = 'Study Person Mid Initials'
altamisa.constants.investigation_headers.STUDY_PERSON_EMAIL = 'Study Person Email'
altamisa.constants.investigation_headers.STUDY_PERSON_PHONE = 'Study Person Phone'
altamisa.constants.investigation_headers.STUDY_PERSON_FAX = 'Study Person Fax'
altamisa.constants.investigation_headers.STUDY_PERSON_ADDRESS = 'Study Person Address'
altamisa.constants.investigation_headers.STUDY_PERSON_AFFILIATION = 'Study Person Affiliation'
altamisa.constants.investigation_headers.STUDY_PERSON_ROLES = 'Study Person Roles'
altamisa.constants.investigation_headers.STUDY_PERSON_ROLES_TERM_ACCESSION_NUMBER = 'Study Person Roles Term Accession Number'
altamisa.constants.investigation_headers.STUDY_PERSON_ROLES_TERM_SOURCE_REF = 'Study Person Roles Term Source Ref'
altamisa.constants.investigation_headers.STUDY_CONTACTS_KEYS = ('Study Person Last Name',
    Collected header keys of STUDY CONTACTS section

```

4.8.2 Table Headers

Constants of valid column headers for study and assay table parsing and writing, as specified in the [ISA-Tab study and assay specifications](#).

```

altamisa.constants.table_headers.EXTRACT_NAME = 'Extract Name'
altamisa.constants.table_headers.LABELED_EXTRACT_NAME = 'Labeled Extract Name'
altamisa.constants.table_headers.LIBRARY_NAME = 'Library Name'
    Special material (see Special Extensions)
altamisa.constants.table_headers.SAMPLE_NAME = 'Sample Name'
altamisa.constants.table_headers.SOURCE_NAME = 'Source Name'
altamisa.constants.table_headers.ARRAY_DATA_FILE = 'Array Data File'
altamisa.constants.table_headers.ARRAY_DATA_MATRIX_FILE = 'Array Data Matrix File'
altamisa.constants.table_headers.ARRAY_DESIGN_FILE = 'Array Design File'
altamisa.constants.table_headers.DERIVED_ARRAY_DATA_FILE = 'Derived Array Data File'
altamisa.constants.table_headers.DERIVED_ARRAY_DATA_MATRIX_FILE = 'Derived Array Data Matrix File'
altamisa.constants.table_headers.DERIVED_DATA_FILE = 'Derived Data File'
altamisa.constants.table_headers.DERIVED_SPECTRAL_DATA_FILE = 'Derived Spectral Data File'
altamisa.constants.table_headers.IMAGE_FILE = 'Image File'
altamisa.constants.table_headers.METABOLITE_ASSIGNMENT_FILE = 'Metabolite Assignment File'
altamisa.constants.table_headers.PEPTIDE_ASSIGNMENT_FILE = 'Peptide Assignment File'
altamisa.constants.table_headers.POST_TRANSLATIONAL_MODIFICATION_ASSIGNMENT_FILE = 'Post Translational Modification Assignment File'
altamisa.constants.table_headers.PROTEIN_ASSIGNMENT_FILE = 'Protein Assignment File'
altamisa.constants.table_headers.RAW_DATA_FILE = 'Raw Data File'
altamisa.constants.table_headers.RAW_SPECTRAL_DATA_FILE = 'Raw Spectral Data File'
altamisa.constants.table_headers.SPOT_PICKING_FILE = 'Spot Picking File'
altamisa.constants.table_headers.ASSAY_NAME = 'Assay Name'

```

```
altamisa.constants.table_headers.DATA_TRANSFORMATION_NAME = 'Data Transformation Name'
altamisa.constants.table_headers.GEL_ELECTROPHORESIS_ASSAY_NAME = 'Gel Electrophoresis Assay Name'
altamisa.constants.table_headers.HYBRIDIZATION_ASSAY_NAME = 'Hybridization Assay Name'
altamisa.constants.table_headers.MS_ASSAY_NAME = 'MS Assay Name'
altamisa.constants.table_headers.NORMALIZATION_NAME = 'Normalization Name'
altamisa.constants.table_headers.PROTOCOL_REF = 'Protocol REF'
altamisa.constants.table_headers.SCAN_NAME = 'Scan Name'
altamisa.constants.table_headers.DATE = 'Date'
altamisa.constants.table_headers.LABEL = 'Label'
altamisa.constants.table_headers.MATERIAL_TYPE = 'Material Type'
altamisa.constants.table_headers.PERFORMER = 'Performer'
altamisa.constants.table_headers.TERM_SOURCE_REF = 'Term Source REF'
altamisa.constants.table_headers.TERM_ACCESSION_NUMBER = 'Term Accession Number'
altamisa.constants.table_headers.UNIT = 'Unit'
altamisa.constants.table_headers.ARRAY_DESIGN_REF = 'Array Design REF'
altamisa.constants.table_headers.FIRST_DIMENSION = 'First Dimension'
altamisa.constants.table_headers.SECOND_DIMENSION = 'Second Dimension'
altamisa.constants.table_headers.CHARACTERISTICS = 'Characteristics'
altamisa.constants.table_headers.COMMENT = 'Comment'
altamisa.constants.table_headers.FACTOR_VALUE = 'Factor Value'
altamisa.constants.table_headers.PARAMETER_VALUE = 'Parameter Value'
altamisa.constants.table_headers.DATA_FILE_HEADERS = ('Array Data File', 'Array Data Matrix')
    Collected header keys indicating a data file.
altamisa.constants.table_headers.MATERIAL_NAME_HEADERS = ('Extract Name', 'Labeled Extract')
    Collected header keys indicating a material/data name.
altamisa.constants.table_headers.PROCESS_NAME_HEADERS = ('Assay Name', 'Data Transformation')
    Collected header keys indicating a process name.
```

4.8.3 Table Restrictions

Constants used for validation of data and annotation restrictions in study and assay tables.

```
altamisa.constants.table_restrictions.PROTEIN_EXPRESSION_PROFILING = 'protein expression profiling'
altamisa.constants.table_restrictions.PROTEIN_IDENTIFICATION = 'protein identification'
altamisa.constants.table_restrictions.METABOLITE_PROFILING = 'metabolite profiling'
altamisa.constants.table_restrictions.DNA_MICROARRAY = 'dna microarray'
altamisa.constants.table_restrictions.GEL_ELECTROPHORESIS = 'gel electrophoresis'
altamisa.constants.table_restrictions.PROTEIN_MICROARRAY = 'protein microarray'
altamisa.constants.table_restrictions.NUCLEOTIDE_SEQUENCING = 'nucleotide sequencing'
```

```

altamisa.constants.table_restrictions.MASS_SPECTROMETRY = 'mass spectrometry'
altamisa.constants.table_restrictions.PT_DATA_COLLECTION = 'data collection'
altamisa.constants.table_restrictions.PT_DATA_NORMALIZATION = 'data normalization'
altamisa.constants.table_restrictions.PT_DATA_TRANSFORMATION = 'data transformation'
altamisa.constants.table_restrictions.PT_ELECTROPHORESIS = 'electrophoresis'
altamisa.constants.table_restrictions.PT_HYBRIDIZATION = 'hybridization'
altamisa.constants.table_restrictions.PT_MASS_SPECTROMETRY = 'mass spectrometry'
altamisa.constants.table_restrictions.PT_NUCLEIC_ACID_HYBRIDIZATION = 'nucleic acid hybridization'
altamisa.constants.table_restrictions.RESTRICTED_MATERIALS_ATECH = {'Library Name': {'nucleic acid hybridization', 'mass spectrometry', 'data collection', 'data normalization', 'data transformation', 'electrophoresis', 'hybridization'}}
    Materials restricted to certain assay technologies
altamisa.constants.table_restrictions.RESTRICTED_MATERIALS_AMEAS = {}
    Materials restricted to certain assay measurements
altamisa.constants.table_restrictions.RESTRICTED_FILES_ATECH = {'Array Data File': {'dna microarray', 'mass spectrometry', 'data collection', 'data normalization', 'data transformation', 'electrophoresis', 'hybridization'}}
    Data files restricted to certain assay technologies
altamisa.constants.table_restrictions.RESTRICTED_FILES_AMEAS = {'Metabolite Assignment File': {'dna microarray', 'mass spectrometry', 'data collection', 'data normalization', 'data transformation', 'electrophoresis', 'hybridization'}}
    Data files restricted to certain assay measurements
altamisa.constants.table_restrictions.RESTRICTED_PROTO_NAMES_ATECH = {'Gel Electrophoresis': {'dna microarray', 'mass spectrometry', 'data collection', 'data normalization', 'data transformation', 'electrophoresis', 'hybridization'}}
    Protocol names restricted to certain assay technologies
altamisa.constants.table_restrictions.RESTRICTED_PROTO_NAMES_PTYPE = {'Data Transformation': {'dna microarray', 'mass spectrometry', 'data collection', 'data normalization', 'data transformation', 'electrophoresis', 'hybridization'}}
    Protocol names restricted by certain protocol types (ignored if ref is UNKNOWN)
altamisa.constants.table_restrictions.RESTRICTED_PROTO_ANNOS_ATECH = {'Array Design REF': {'dna microarray', 'mass spectrometry', 'data collection', 'data normalization', 'data transformation', 'electrophoresis', 'hybridization'}}
    Protocol special case annotations restricted to certain assay technologies
altamisa.constants.table_restrictions.RESTRICTED_PROTO_ANNOS_PTYPE = {'Array Design REF': {'dna microarray', 'mass spectrometry', 'data collection', 'data normalization', 'data transformation', 'electrophoresis', 'hybridization'}}
    Protocol special case annotations restricted to certain protocol types

```

4.8.4 Table Tokens

Token constants for study and assay table parsing and writing.

```

altamisa.constants.table_tokens.TOKEN_ANONYMOUS = 'Anonymous'
    Used for marking anonymous/unnamed Processes
altamisa.constants.table_tokens.TOKEN_EMPTY = 'Empty'
    Used for marking empty/unnamed Data
altamisa.constants.table_tokens.TOKEN_UNKNOWN = 'Unknown'
    Used for named Processes without protocol reference

```

4.9 Headers

This module contains code for the representation of headers from study and assay files and parsing thereof.

```

class altamisa.isatab.headers.ColumnHeader(column_type, col_no, span)
    Column header in a study or assay file

```

```
column_type = None
    The type of this header

col_no = None
    The column number this header refers to

span = None
    Number of columns this header spans

term_source_ref_header = None
    Link to the TermSourceRefHeader to use

unit_header = None
    Link to the UnitHeader to use

get_simple_string() → List[str]
    Return a list of simple string representations of the column types

class altamisa.isatab.headers.SimpleColumnHeader(col_no)
    Base class for simple column headers.

    column_type = None
        The value to use for the type argument.

class altamisa.isatab.headers.ExtractHeader(col_no)
    Extract header in an assay

    column_type = 'Extract Name'

class altamisa.isatab.headers.LabeledExtractHeader(col_no)
    Labeled Extract header in an assay

    column_type = 'Labeled Extract Name'

class altamisa.isatab.headers.LibraryHeader(col_no)
    Library header in an assay

    column_type = 'Library Name'

class altamisa.isatab.headers.SampleHeader(col_no)
    Sample header in a study or assay

    column_type = 'Sample Name'

class altamisa.isatab.headers.SourceHeader(col_no)
    Source header in a study

    column_type = 'Source Name'

class altamisa.isatab.headers.ArrayDataFileHeader(col_no)
    ArrayData header in an assay

    column_type = 'Array Data File'

class altamisa.isatab.headers.ArrayDataMatrixFileHeader(col_no)
    ArrayData Matrix File header in an assay

    column_type = 'Array Data Matrix File'

class altamisa.isatab.headers.ArrayDesignFileHeader(col_no)
    ArrayDesignFile header in an assay

    column_type = 'Array Design File'

class altamisa.isatab.headers.DerivedArrayDataFileHeader(col_no)
    DerivedArrayData header in an assay
```

```
    column_type = 'Derived Array Data File'
class altamisa.isatab.headers.DerivedArrayDataMatrixFileHeader(col_no)
    DerivedArrayData header in an assay
    column_type = 'Derived Array Data Matrix File'
class altamisa.isatab.headers.DerivedDataFileHeader(col_no)
    Derived Data File header in an assay
    column_type = 'Derived Data File'
class altamisa.isatab.headers.DerivedSpectralDataFileHeader(col_no)
    DerivedSpectralData header in an assay
    column_type = 'Derived Spectral Data File'
class altamisa.isatab.headers.ImageFileHeader(col_no)
    Image File header in an assay
    column_type = 'Image File'
class altamisa.isatab.headers.MetaboliteAssignmentFileHeader(col_no)
    PeptideAssignment header in an assay
    column_type = 'Metabolite Assignment File'
class altamisa.isatab.headers.PeptideAssignmentFileHeader(col_no)
    PeptideAssignment header in an assay
    column_type = 'Peptide Assignment File'
class altamisa.isatab.headers.PostTranslationalModificationAssignmentFileHeader(col_no)
    PostTranslationalModificationAssignment header in an assay
    column_type = 'Post Translational Modification Assignment File'
class altamisa.isatab.headers.ProteinAssignmentFileHeader(col_no)
    ProteinAssignment header in an assay
    column_type = 'Protein Assignment File'
class altamisa.isatab.headers.RawDataFileHeader(col_no)
    Raw Data header in an assay
    column_type = 'Raw Data File'
class altamisa.isatab.headers.RawSpectralDataFileHeader(col_no)
    Raw Spectral Data header in an assay
    column_type = 'Raw Spectral Data File'
class altamisa.isatab.headers.SpotPickingFileHeader(col_no)
    SpotPickingFile header in an assay
    column_type = 'Spot Picking File'
class altamisa.isatab.headers.AssayNameHeader(col_no)
    Assay Name header in an assay
    column_type = 'Assay Name'
class altamisa.isatab.headers.DataTransformationNameHeader(col_no)
    DataTransformationName header in an assay
    column_type = 'Data Transformation Name'
```

```
class altamisa.isatab.headers.GelElectrophoresisAssayNameHeader (col_no)
    GelElectrophoresisAssayName header in an assay

    column_type = 'Gel Electrophoresis Assay Name'

class altamisa.isatab.headers.HybridizationAssayNameHeader (col_no)
    HybridizationAssayName header in an assay

    column_type = 'Hybridization Assay Name'

class altamisa.isatab.headers.MsAssayNameHeader (col_no)
    MsAssayName header in an assay

    column_type = 'MS Assay Name'

class altamisa.isatab.headers.NormalizationNameHeader (col_no)
    Normalization Name header in a assay

    column_type = 'Normalization Name'

class altamisa.isatab.headers.ProtocolRefHeader (col_no)
    Protocol REF header in a study or assay

    column_type = 'Protocol REF'

class altamisa.isatab.headers.ScanNameHeader (col_no)
    ScanName header in assay

    column_type = 'Scan Name'

class altamisa.isatab.headers.ArrayDesignRefHeader (col_no)
    ArrayDesignRef header in an assay

    column_type = 'Array Design REF'

class altamisa.isatab.headers.DateHeader (col_no)
    Date annotation header in a study or assay

    column_type = 'Date'

class altamisa.isatab.headers.FirstDimensionHeader (col_no)
    First Dimension header in an assay

    column_type = 'First Dimension'

class altamisa.isatab.headers.LabelHeader (col_no)
    Label header in an assay

    column_type = 'Label'

class altamisa.isatab.headers.MaterialTypeHeader (col_no)
    Material Type header in an assay

    column_type = 'Material Type'

class altamisa.isatab.headers.PerformerHeader (col_no)
    Performer header in an assay

    column_type = 'Performer'

class altamisa.isatab.headers.SecondDimensionHeader (col_no)
    Second Dimension header in an assay

    column_type = 'Second Dimension'

class altamisa.isatab.headers.TermRefAnnotationHeader (col_no)
    Term reference annotation header
```

```

    get_simple_string() → List[str]
        Return a list of simple string representations of the column types

class altamisa.isatab.headers.UnitHeader(col_no)
    Unit annotation header in a study or assay

    column_type = 'Unit'

class altamisa.isatab.headers.LabeledColumnHeader(col_no, label)
    Base class for labeled column headers.

    column_type = None
        The value to use for the type argument.

    get_simple_string()
        Return a list of simple string representations of the column types

class altamisa.isatab.headers.CharacteristicsHeader(col_no, label)
    Material Characteristics[*] header in a study or assay

    column_type = 'Characteristics'

class altamisa.isatab.headers.CommentHeader(col_no, label)
    Comment header in a study or assay

    column_type = 'Comment'

class altamisa.isatab.headers.FactorValueHeader(col_no, label)
    Factor Value[*] header in a study or assay

    column_type = 'Factor Value'

class altamisa.isatab.headers.ParameterValueHeader(col_no, label)
    Protocol Parameter Value[*] header in a study or assay

    column_type = 'Parameter Value'

class altamisa.isatab.headers.HeaderParserBase(tokens)
    Helper base class for parsing a header from a study or assay file.

    Parameters tokens (list) – List of strings, e.g. a split line read from a tsv/cvs file.

    allowed_headers = None
        Names of the allowed headers

    simple_headers = {'Array Data File': <class 'altamisa.isatab.headers.ArrayDataFileHeader'>}
        Headers that are mapped to SimpleColumnHeader

    labeled_headers = {'Characteristics': <class 'altamisa.isatab.headers.CharacteristicsHeader'>}
        Labeled headers

    run() → Iterator[altamisa.isatab.headers.ColumnHeader]
        Parse the header

class altamisa.isatab.headers.StudyHeaderParser(tokens)
    Helper class for parsing header of a study or assay.

    file_type = 'study'

    allowed_headers = ('Sample Name', 'Source Name', 'Protocol REF', 'Date', 'Performer',

class altamisa.isatab.headers.AssayHeaderParser(tokens)
    Helper class for parsing header of a assay file.

    file_type = 'assay'

```

```
allowed_headers = ('Extract Name', 'Labeled Extract Name', 'Library Name', 'Sample Name')
```

4.10 Exceptions

Exceptions and Warnings used in the AltamISA library.

exception `altamisa.exceptions.IsaException`

Base class for exceptions raised by Altamisa.

exception `altamisa.exceptions.ParseIsatabException`

Exception raised on problems parsing ISA-TAB.

exception `altamisa.exceptions.WriteIsatabException`

Exception raised on problems writing ISA-TAB.

exception `altamisa.exceptions.IsaWarning`

Base class for warnings raised by Altamisa.

exception `altamisa.exceptions.ParseIsatabWarning`

Warning raised on problems parsing ISA-TAB.

exception `altamisa.exceptions.WriteIsatabWarning`

Warning raised on problems writing ISA-TAB.

exception `altamisa.exceptions.IsaValidationWarning`

Warning raised on problems validating ISA models or objects.

exception `altamisa.exceptions.AdvisoryIsaValidationWarning`

Warning raised on uncritical problems when validating ISA models or objects.

exception `altamisa.exceptions.ModerateIsaValidationWarning`

Warning raised on moderate problems when validating ISA models or objects.

exception `altamisa.exceptions.CriticalIsaValidationWarning`

Warning raised on critical problems when validating ISA models or objects.

4.11 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.11.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/bihealth/altamisa/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

AltamISA could always use more documentation, whether as part of the official AltamISA docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/bihealth/altamisa/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.11.2 Get Started!

Ready to contribute? Here’s how to set up *altamisa* for local development.

1. Fork the *altamisa* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/altamisa.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv altamisa
$ cd altamisa/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you’re done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 altamisa tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.11.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/bihealth/altamisa/pull_requests and make sure that the tests pass for all supported Python versions.

4.11.4 Tips

To run a subset of tests:

```
$ py.test tests.test_altamisa
```

4.11.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.md). Then run:

```
$ git tag vMAJOR.MINOR.PATCH
$ git push
$ git push --tags
```

This will create a tag on Github. Next, build with sdist:

```
$ rm -rf dist
$ python setup.py sdist
```

Ensure that this builds a clean package without hash and -dirty. Now, release to PyPI using twine:

```
$ twine upload --repository-url https://test.pypi.org/legacy/ dist/altamisa-*.tar.gz
$ twine upload dist/altamisa-*.tar.gz
```

4.12 Credits

- Dieter Beule
- Manuel Holtgrewe
- Jennifer Kirwan

- Mathias Kuhring
- Mikko Nieminen

4.13 History

4.13.1 v0.2.8

- Mostly meta adjustments.

4.13.2 v0.2.7

- Adding exception for duplicate node annotations

4.13.3 v0.2.6

- Minor fixes regarding investigation file names and comments.

4.13.4 v0.2.5

- Minor fixes of validation and warnings.
- Fixes optional parameter `filename` of `AssayReader`.

4.13.5 v0.2.4

- Ensuring that input order is output order. This is true except for the corner case where materials are not located in “blocks”. Such corner cases would require storing the tabular representation (and keeping it in sync) at all times and does not yield to a robustly usable implementation. NB: the input is also not sorted the test adjusted with this patch shows.
- Adding optional parameter `filename` to the various readers.
- Exposing `RefTableBuilder` class with slightly changed interface.

4.13.6 v0.2.3

- Minor fixes and additions with focus on improving the export.

4.13.7 v0.2.2

- Updating documentation for JOSS.

4.13.8 v0.2.1

- Adding JOSS paper draft.
- Fixing problem with writing empty lines on Windows (#52).
- Update documentation with examples for manual model creation.
- Fixing authorship documentation.
- Fixing package (#58).

4.13.9 v0.2.0

- Switching to `attrs` instead of using `Namedtuple`. This gets rid of some warts regarding constructor overriding but should offer the same functionality otherwise.
- Various updates to the documentation.

4.13.10 v0.1.0

First public release.

- Started out with ISA-TAB parser and `NamedTuple`-based data model.

4.14 License

You can find the License of AltamISA below.

MIT License

Copyright (c) 2018–2019, Berlin Institute of Health

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

a

`altamisa.constants.investigation_headers`,
34
`altamisa.constants.table_headers`, 37
`altamisa.constants.table_restrictions`,
38
`altamisa.constants.table_tokens`, 39
`altamisa.exceptions`, 44
`altamisa.isatab.headers`, 39

A

- accession (*altamisa.isatab.OntologyTermRef* attribute), 24
- address (*altamisa.isatab.ContactInfo* attribute), 27
- AdvisoryIsaValidationWarning, 44
- affiliation (*altamisa.isatab.ContactInfo* attribute), 27
- allowed_headers (*altamisa.isatab.headers.AssayHeaderParser* attribute), 43
- allowed_headers (*altamisa.isatab.headers.HeaderParserBase* attribute), 43
- allowed_headers (*altamisa.isatab.headers.StudyHeaderParser* attribute), 43
- altamisa.constants.investigation_headers (module), 34
- altamisa.constants.table_headers (module), 37
- altamisa.constants.table_restrictions (module), 38
- altamisa.constants.table_tokens (module), 39
- altamisa.exceptions (module), 44
- altamisa.isatab.headers (module), 39
- AnnotatedStr (class in *altamisa.isatab*), 24
- Arc (class in *altamisa.isatab*), 32
- arcs (*altamisa.isatab.Assay* attribute), 33
- arcs (*altamisa.isatab.Study* attribute), 33
- ARRAY_DATA_FILE (in module *altamisa.constants.table_headers*), 37
- ARRAY_DATA_MATRIX_FILE (in module *altamisa.constants.table_headers*), 37
- ARRAY_DESIGN_FILE (in module *altamisa.constants.table_headers*), 37
- array_design_ref (*altamisa.isatab.Process* attribute), 32
- ARRAY_DESIGN_REF (in module *altamisa.constants.table_headers*), 38
- ArrayDataFileHeader (class in *altamisa.isatab.headers*), 40
- ArrayDataMatrixFileHeader (class in *altamisa.isatab.headers*), 40
- ArrayDesignFileHeader (class in *altamisa.isatab.headers*), 40
- ArrayDesignRefHeader (class in *altamisa.isatab.headers*), 42
- Assay (class in *altamisa.isatab*), 33
- ASSAY_NAME (in module *altamisa.constants.table_headers*), 37
- AssayHeaderParser (class in *altamisa.isatab.headers*), 43
- AssayInfo (class in *altamisa.isatab*), 28
- AssayNameHeader (class in *altamisa.isatab.headers*), 41
- AssayReader (class in *altamisa.isatab*), 18
- AssayRowReader (class in *altamisa.isatab*), 19
- assays (*altamisa.isatab.StudyInfo* attribute), 29
- AssayValidator (class in *altamisa.isatab*), 22
- AssayWriter (class in *altamisa.isatab*), 21
- authors (*altamisa.isatab.PublicationInfo* attribute), 26

B

BasicInfo (class in *altamisa.isatab*), 25

C

- characteristics (*altamisa.isatab.Material* attribute), 31
- Characteristics (class in *altamisa.isatab*), 30
- CHARACTERISTICS (in module *altamisa.constants.table_headers*), 38
- CharacteristicsHeader (class in *altamisa.isatab.headers*), 43
- col_no (*altamisa.isatab.headers.ColumnHeader* attribute), 40
- column_type (*altamisa.isatab.headers.ArrayDataFileHeader* attribute), 40

`column_type (altamisa.isatab.headers.ArrayDataMatrixFileHeader attribute), 40`
`column_type (altamisa.isatab.headers.ArrayDesignFileHeader attribute), 40`
`column_type (altamisa.isatab.headers.ArrayDesignRefHeader attribute), 42`
`column_type (altamisa.isatab.headers.AssayNameHeader attribute), 41`
`column_type (altamisa.isatab.headers.CharacteristicsHeader attribute), 43`
`column_type (altamisa.isatab.headers.ColumnHeader attribute), 39`
`column_type (altamisa.isatab.headers.CommentHeader attribute), 43`
`column_type (altamisa.isatab.headers.DataTransformationNameHeader attribute), 41`
`column_type (altamisa.isatab.headers.DateHeader attribute), 42`
`column_type (altamisa.isatab.headers.DerivedArrayDataFileHeader attribute), 40`
`column_type (altamisa.isatab.headers.DerivedArrayDataMatrixFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.DerivedDataFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.DerivedSpectralDataFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.ExtractHeader attribute), 40`
`column_type (altamisa.isatab.headers.FactorValueHeader attribute), 43`
`column_type (altamisa.isatab.headers.FirstDimensionHeader attribute), 42`
`column_type (altamisa.isatab.headers.GelElectrophoresisAssayNameHeader attribute), 42`
`column_type (altamisa.isatab.headers.HybridizationAssayNameHeader attribute), 42`
`column_type (altamisa.isatab.headers.ImageFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.LabeledColumnHeader attribute), 43`
`column_type (altamisa.isatab.headers.LabeledExtractHeader attribute), 40`
`column_type (altamisa.isatab.headers.LabelHeader attribute), 42`
`column_type (altamisa.isatab.headers.LibraryHeader attribute), 40`
`column_type (altamisa.isatab.headers.MaterialTypeHeader attribute), 42`
`column_type (altamisa.isatab.headers.MetaboliteAssignmentFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.MsAssayNameHeader attribute), 42`
`column_type (altamisa.isatab.headers.NormalizationNameHeader attribute), 42`
`column_type (altamisa.isatab.headers.ParameterValueHeader attribute), 43`
`column_type (altamisa.isatab.headers.PeptideAssignmentFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.PerformerHeader attribute), 42`
`column_type (altamisa.isatab.headers.PostTranslationalModificationAssayNameHeader attribute), 41`
`column_type (altamisa.isatab.headers.ProteinAssignmentFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.ProtocolRefHeader attribute), 42`
`column_type (altamisa.isatab.headers.RawDataFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.RawSpectralDataFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.SampleHeader attribute), 40`
`column_type (altamisa.isatab.headers.ScanNameHeader attribute), 42`
`column_type (altamisa.isatab.headers.SecondDimensionHeader attribute), 42`
`column_type (altamisa.isatab.headers.SimpleColumnHeader attribute), 40`
`column_type (altamisa.isatab.headers.SourceHeader attribute), 40`
`column_type (altamisa.isatab.headers.SpotPickingFileHeader attribute), 41`
`column_type (altamisa.isatab.headers.UnitHeader attribute), 43`
`ColumnHeader (class in altamisa.isatab.headers), 39`
`Comment (class in altamisa.isatab), 25`
`constants (altamisa.constants.table_headers), 38`
`comments (altamisa.isatab.AssayInfo attribute), 28`
`comments (altamisa.isatab.BasicInfo attribute), 26`
`comments (altamisa.isatab.ContactInfo attribute), 27`
`comments (altamisa.isatab.DesignDescriptorsInfo attribute), 27`
`comments (altamisa.isatab.FactorInfo attribute), 27`
`comments (altamisa.isatab.Material attribute), 31`
`comments (altamisa.isatab.OntologyRef attribute), 25`
`comments (altamisa.isatab.Process attribute), 32`
`comments (altamisa.isatab.ProtocolInfo attribute), 29`
`comments (altamisa.isatab.PublicationInfo attribute), 26`
`components (altamisa.isatab.ProtocolInfo attribute), 29`
`ContactInfo (class in altamisa.isatab), 26`
`contacts (altamisa.isatab.InvestigationInfo attribute), 30`
`critical_isa_validation_warnings (altamisa.isatab.StudyInfo attribute), 29`
`CriticalIsaValidationWarning, 44`

D

DATA_FILE_HEADERS (in module *altamisa.constants.table_headers*), 38

DATA_TRANSFORMATION_NAME (in module *altamisa.constants.table_headers*), 37

DataTransformationNameHeader (class in *altamisa.isatab.headers*), 41

date (*altamisa.isatab.Process* attribute), 32

DATE (in module *altamisa.constants.table_headers*), 38

DateHeader (class in *altamisa.isatab.headers*), 42

DERIVED_ARRAY_DATA_FILE (in module *altamisa.constants.table_headers*), 37

DERIVED_ARRAY_DATA_MATRIX_FILE (in module *altamisa.constants.table_headers*), 37

DERIVED_DATA_FILE (in module *altamisa.constants.table_headers*), 37

DERIVED_SPECTRAL_DATA_FILE (in module *altamisa.constants.table_headers*), 37

DerivedArrayDataFileHeader (class in *altamisa.isatab.headers*), 40

DerivedArrayDataMatrixFileHeader (class in *altamisa.isatab.headers*), 41

DerivedDataFileHeader (class in *altamisa.isatab.headers*), 41

DerivedSpectralDataFileHeader (class in *altamisa.isatab.headers*), 41

description (*altamisa.isatab.BasicInfo* attribute), 25

description (*altamisa.isatab.OntologyRef* attribute), 25

description (*altamisa.isatab.ProtocolInfo* attribute), 28

DesignDescriptorsInfo (class in *altamisa.isatab*), 27

designs (*altamisa.isatab.StudyInfo* attribute), 29

DNA_MICROARRAY (in module *altamisa.constants.table_restrictions*), 38

doi (*altamisa.isatab.PublicationInfo* attribute), 26

E

email (*altamisa.isatab.ContactInfo* attribute), 26

extract_label (*altamisa.isatab.Material* attribute), 31

EXTRACT_NAME (in module *altamisa.constants.table_headers*), 37

ExtractHeader (class in *altamisa.isatab.headers*), 40

F

FACTOR_VALUE (in module *altamisa.constants.table_headers*), 38

factor_values (*altamisa.isatab.Material* attribute), 31

FactorInfo (class in *altamisa.isatab*), 27

factors (*altamisa.isatab.StudyInfo* attribute), 29

FactorValue (class in *altamisa.isatab*), 30

FactorValueHeader (class in *altamisa.isatab.headers*), 43

fax (*altamisa.isatab.ContactInfo* attribute), 27

file (*altamisa.isatab.Assay* attribute), 33

file (*altamisa.isatab.OntologyRef* attribute), 25

file (*altamisa.isatab.Study* attribute), 33

file_type (*altamisa.isatab.headers.AssayHeaderParser* attribute), 43

file_type (*altamisa.isatab.headers.StudyHeaderParser* attribute), 43

first_dimension (*altamisa.isatab.Process* attribute), 32

FIRST_DIMENSION (in module *altamisa.constants.table_headers*), 38

first_name (*altamisa.isatab.ContactInfo* attribute), 26

FirstDimensionHeader (class in *altamisa.isatab.headers*), 42

from_stream() (*altamisa.isatab.AssayReader* class method), 19

from_stream() (*altamisa.isatab.AssayRowReader* class method), 19

from_stream() (*altamisa.isatab.InvestigationReader* class method), 18

from_stream() (*altamisa.isatab.InvestigationWriter* class method), 20

from_stream() (*altamisa.isatab.StudyReader* class method), 19

from_stream() (*altamisa.isatab.StudyRowReader* class method), 20

G

GEL_ELECTROPHORESIS (in module *altamisa.constants.table_restrictions*), 38

GEL_ELECTROPHORESIS_ASSAY_NAME (in module *altamisa.constants.table_headers*), 38

GelElectrophoresisAssayNameHeader (class in *altamisa.isatab.headers*), 41

get_simple_string() (*altamisa.isatab.headers.ColumnHeader* method), 40

get_simple_string() (*altamisa.isatab.headers.LabeledColumnHeader* method), 43

get_simple_string() (*altamisa.isatab.headers.TermRefAnnotationHeader* method), 42

H

head (*altamisa.isatab.Arc* attribute), 32

header (*altamisa.isatab.Assay* attribute), 33

header (*altamisa.isatab.Study* attribute), 33

HeaderParserBase (class in *altamisa.isatab.headers*), 43

headers (*altamisa.isatab.AssayInfo* attribute), 28
 headers (*altamisa.isatab.BasicInfo* attribute), 26
 headers (*altamisa.isatab.ContactInfo* attribute), 27
 headers (*altamisa.isatab.DesignDescriptorsInfo* attribute), 27
 headers (*altamisa.isatab.FactorInfo* attribute), 27
 headers (*altamisa.isatab.Material* attribute), 31
 headers (*altamisa.isatab.OntologyRef* attribute), 25
 headers (*altamisa.isatab.Process* attribute), 32
 headers (*altamisa.isatab.ProtocolInfo* attribute), 29
 headers (*altamisa.isatab.PublicationInfo* attribute), 26
 HYBRIDIZATION_ASSAY_NAME (in module *altamisa.constants.table_headers*), 38
 HybridizationAssayNameHeader (class in *altamisa.isatab.headers*), 42

I

identifier (*altamisa.isatab.BasicInfo* attribute), 25
 IMAGE_FILE (in module *altamisa.constants.table_headers*), 37
 ImageFileHeader (class in *altamisa.isatab.headers*), 41
 info (*altamisa.isatab.InvestigationInfo* attribute), 30
 info (*altamisa.isatab.StudyInfo* attribute), 29
 INVESTIGATION (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_CONTACTS (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_CONTACTS_KEYS (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_DESCRIPTION (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_IDENTIFIER (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_INFO_KEYS (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_PERSON_ADDRESS (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_AFFILIATION (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_EMAIL (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_FAX (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_FIRST_NAME (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_LAST_NAME (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_MID_INITIALS (in module *altamisa.constants.investigation_headers*), 35

tamisa.constants.investigation_headers), 35
 INVESTIGATION_PERSON_PHONE (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_ROLES (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_ROLES_TERM_ACCESSION_NUMBER (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PERSON_ROLES_TERM_SOURCE_REF (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PUBLIC_RELEASE_DATE (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_PUBLICATION_AUTHOR_LIST (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_PUBLICATION_DOI (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_PUBLICATION_STATUS (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_PUBLICATION_STATUS_TERM_ACCESSION_NUMBER (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PUBLICATION_STATUS_TERM_SOURCE_REF (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PUBLICATION_TITLE (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_PUBLICATIONS (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_PUBLICATIONS_KEYS (in module *altamisa.constants.investigation_headers*), 35
 INVESTIGATION_PUBMED_ID (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_SUBMISSION_DATE (in module *altamisa.constants.investigation_headers*), 34
 INVESTIGATION_TITLE (in module *altamisa.constants.investigation_headers*), 34
 InvestigationInfo (class in *altamisa.isatab*), 29
 InvestigationReader (class in *altamisa.isatab*), 18
 InvestigationValidator (class in *altamisa.isatab*), 18

tamisa.isatab), 22
 InvestigationWriter (class in *altamisa.isatab*), 20
 IsaException, 44
 IsaValidationWarning, 44
 IsaWarning, 44

L

LABEL (in module *altamisa.constants.table_headers*), 38
 LABELED_EXTRACT_NAME (in module *altamisa.constants.table_headers*), 37
 labeled_headers (in *altamisa.isatab.headers.HeaderParserBase* attribute), 43
 LabeledColumnHeader (class in *altamisa.isatab.headers*), 43
 LabeledExtractHeader (class in *altamisa.isatab.headers*), 40
 LabelHeader (class in *altamisa.isatab.headers*), 42
 last_name (*altamisa.isatab.ContactInfo* attribute), 26
 LIBRARY_NAME (in module *altamisa.constants.table_headers*), 37
 LibraryHeader (class in *altamisa.isatab.headers*), 40

M

MASS_SPECTROMETRY (in module *altamisa.constants.table_restrictions*), 38
 Material (class in *altamisa.isatab*), 31
 MATERIAL_NAME_HEADERS (in module *altamisa.constants.table_headers*), 38
 material_type (*altamisa.isatab.Material* attribute), 31
 MATERIAL_TYPE (in module *altamisa.constants.table_headers*), 38
 materials (*altamisa.isatab.Assay* attribute), 33
 materials (*altamisa.isatab.Study* attribute), 33
 MaterialTypeHeader (class in *altamisa.isatab.headers*), 42
 measurement_type (*altamisa.isatab.AssayInfo* attribute), 28
 METABOLITE_ASSIGNMENT_FILE (in module *altamisa.constants.table_headers*), 37
 METABOLITE_PROFILING (in module *altamisa.constants.table_restrictions*), 38
 MetaboliteAssignmentFileHeader (class in *altamisa.isatab.headers*), 41
 mid_initial (*altamisa.isatab.ContactInfo* attribute), 26
 ModerateIsaValidationWarning, 44
 MS_ASSAY_NAME (in module *altamisa.constants.table_headers*), 38
 MsAssayNameHeader (class in *altamisa.isatab.headers*), 42

N

name (*altamisa.isatab.Characteristics* attribute), 30
 name (*altamisa.isatab.Comment* attribute), 25
 name (*altamisa.isatab.FactorInfo* attribute), 27
 name (*altamisa.isatab.FactorValue* attribute), 30
 name (*altamisa.isatab.Material* attribute), 31
 name (*altamisa.isatab.OntologyRef* attribute), 25
 name (*altamisa.isatab.OntologyTermRef* attribute), 24
 name (*altamisa.isatab.ParameterValue* attribute), 30
 name (*altamisa.isatab.Process* attribute), 32
 name (*altamisa.isatab.ProtocolComponentInfo* attribute), 28
 name (*altamisa.isatab.ProtocolInfo* attribute), 28
 name_type (*altamisa.isatab.Process* attribute), 32
 NORMALIZATION_NAME (in module *altamisa.constants.table_headers*), 38
 NormalizationNameHeader (class in *altamisa.isatab.headers*), 42
 NUCLEOTIDE_SEQUENCING (in module *altamisa.constants.table_restrictions*), 38

O

ontology_name (*altamisa.isatab.OntologyTermRef* attribute), 24
 ONTOLOGY_SOURCE_REF_KEYS (in module *altamisa.constants.investigation_headers*), 34
 ONTOLOGY_SOURCE_REFERENCE (in module *altamisa.constants.investigation_headers*), 34
 ontology_source_refs (*altamisa.isatab.InvestigationInfo* attribute), 29
 OntologyRef (class in *altamisa.isatab*), 25
 OntologyTermRef (class in *altamisa.isatab*), 24

P

PARAMETER_VALUE (in module *altamisa.constants.table_headers*), 38
 parameter_values (*altamisa.isatab.Process* attribute), 32
 parameters (*altamisa.isatab.ProtocolInfo* attribute), 29
 ParameterValue (class in *altamisa.isatab*), 30
 ParameterValueHeader (class in *altamisa.isatab.headers*), 43
 ParseIsatabException, 44
 ParseIsatabWarning, 44
 path (*altamisa.isatab.AssayInfo* attribute), 28
 path (*altamisa.isatab.BasicInfo* attribute), 25
 PEPTIDE_ASSIGNMENT_FILE (in module *altamisa.constants.table_headers*), 37
 PeptideAssignmentFileHeader (class in *altamisa.isatab.headers*), 41
 performer (*altamisa.isatab.Process* attribute), 32

PERFORMER (in module *altamisa.constants.table_headers*), 38
 PerformerHeader (class in *altamisa.isatab.headers*), 42
 phone (*altamisa.isatab.ContactInfo* attribute), 26
 platform (*altamisa.isatab.AssayInfo* attribute), 28
 POST_TRANSLATIONAL_MODIFICATION_ASSIGNMENT_FILE (in module *altamisa.constants.table_headers*), 37
 PostTranslationalModificationAssignmentFileHeader (class in *altamisa.isatab.headers*), 41
 Process (class in *altamisa.isatab*), 31
 PROCESS_NAME_HEADERS (in module *altamisa.constants.table_headers*), 38
 processes (*altamisa.isatab.Assay* attribute), 33
 processes (*altamisa.isatab.Study* attribute), 33
 PROTEIN_ASSIGNMENT_FILE (in module *altamisa.constants.table_headers*), 37
 PROTEIN_EXPRESSION_PROFILING (in module *altamisa.constants.table_restrictions*), 38
 PROTEIN_IDENTIFICATION (in module *altamisa.constants.table_restrictions*), 38
 PROTEIN_MICROARRAY (in module *altamisa.constants.table_restrictions*), 38
 ProteinAssignmentFileHeader (class in *altamisa.isatab.headers*), 41
 protocol_ref (*altamisa.isatab.Process* attribute), 31
 PROTOCOL_REF (in module *altamisa.constants.table_headers*), 38
 ProtocolComponentInfo (class in *altamisa.isatab*), 28
 ProtocolInfo (class in *altamisa.isatab*), 28
 ProtocolRefHeader (class in *altamisa.isatab.headers*), 42
 protocols (*altamisa.isatab.StudyInfo* attribute), 29
 PT_DATA_COLLECTION (in module *altamisa.constants.table_restrictions*), 39
 PT_DATA_NORMALIZATION (in module *altamisa.constants.table_restrictions*), 39
 PT_DATA_TRANSFORMATION (in module *altamisa.constants.table_restrictions*), 39
 PT_ELECTROPHORESIS (in module *altamisa.constants.table_restrictions*), 39
 PT_HYBRIDIZATION (in module *altamisa.constants.table_restrictions*), 39
 PT_MASS_SPECTROMETRY (in module *altamisa.constants.table_restrictions*), 39
 PT_NUCLEIC_ACID_HYBRIDIZATION (in module *altamisa.constants.table_restrictions*), 39
 public_release_date (*altamisa.isatab.BasicInfo* attribute), 25
 PublicationInfo (class in *altamisa.isatab*), 26
 publications (*altamisa.isatab.InvestigationInfo* attribute), 30
 publications (*altamisa.isatab.StudyInfo* attribute), 29
 pubmed_id (*altamisa.isatab.PublicationInfo* attribute), 26

R

RAW_DATA_FILE (in module *altamisa.constants.table_headers*), 37
 RAW_SPECTRAL_DATA_FILE (in module *altamisa.constants.table_headers*), 37
 RawDataFileHeader (class in *altamisa.isatab.headers*), 41
 RawSpectralDataFileHeader (class in *altamisa.isatab.headers*), 41
 read() (*altamisa.isatab.AssayReader* method), 19
 read() (*altamisa.isatab.AssayRowReader* method), 19
 read() (*altamisa.isatab.InvestigationReader* method), 18
 read() (*altamisa.isatab.StudyReader* method), 19
 read() (*altamisa.isatab.StudyRowReader* method), 20
 RESTRICTED_FILES_AMEAS (in module *altamisa.constants.table_restrictions*), 39
 RESTRICTED_FILES_ATECH (in module *altamisa.constants.table_restrictions*), 39
 RESTRICTED_MATERIALS_AMEAS (in module *altamisa.constants.table_restrictions*), 39
 RESTRICTED_MATERIALS_ATECH (in module *altamisa.constants.table_restrictions*), 39
 RESTRICTED_PROTO_ANNOS_ATECH (in module *altamisa.constants.table_restrictions*), 39
 RESTRICTED_PROTO_ANNOS_PTYPE (in module *altamisa.constants.table_restrictions*), 39
 RESTRICTED_PROTO_NAMES_ATECH (in module *altamisa.constants.table_restrictions*), 39
 RESTRICTED_PROTO_NAMES_PTYPE (in module *altamisa.constants.table_restrictions*), 39
 role (*altamisa.isatab.ContactInfo* attribute), 27
 run() (*altamisa.isatab.headers.HeaderParserBase* method), 43

S

SAMPLE_NAME (in module *altamisa.constants.table_headers*), 37
 SampleHeader (class in *altamisa.isatab.headers*), 40
 SCAN_NAME (in module *altamisa.constants.table_headers*), 38
 ScanNameHeader (class in *altamisa.isatab.headers*), 42
 second_dimension (*altamisa.isatab.Process* attribute), 32
 SECOND_DIMENSION (in module *altamisa.constants.table_headers*), 38
 SecondDimensionHeader (class in *altamisa.isatab.headers*), 42

simple_headers	(<i>altamisa.isatab.headers.HeaderParserBase</i> attribute), 43	<i>tamisa.constants.investigation_headers</i> , 34	
SimpleColumnHeader	(class in <i>altamisa.isatab.headers</i>), 40	STUDY_CONTACTS_KEYS	(in module <i>altamisa.constants.investigation_headers</i>), 37
SOURCE_NAME	(in module <i>altamisa.constants.table_headers</i>), 37	STUDY_DESCRIPTION	(in module <i>altamisa.constants.investigation_headers</i>), 35
SourceHeader	(class in <i>altamisa.isatab.headers</i>), 40	STUDY_DESIGN_DESCR_KEYS	(in module <i>altamisa.constants.investigation_headers</i>), 35
span	(<i>altamisa.isatab.headers.ColumnHeader</i> attribute), 40	STUDY_DESIGN_DESCRIPTOR	(in module <i>altamisa.constants.investigation_headers</i>), 34
SPOT_PICKING_FILE	(in module <i>altamisa.constants.table_headers</i>), 37	STUDY_DESIGN_TYPE	(in module <i>altamisa.constants.investigation_headers</i>), 35
SpotPickingFileHeader	(class in <i>altamisa.isatab.headers</i>), 41	STUDY_DESIGN_TYPE_TERM_ACCESSION_NUMBER	(in module <i>altamisa.constants.investigation_headers</i>), 35
status	(<i>altamisa.isatab.PublicationInfo</i> attribute), 26	STUDY_DESIGN_TYPE_TERM_SOURCE_REF	(in module <i>altamisa.constants.investigation_headers</i>), 35
studies	(<i>altamisa.isatab.InvestigationInfo</i> attribute), 30	STUDY_FACTOR_NAME	(in module <i>altamisa.constants.investigation_headers</i>), 36
Study	(class in <i>altamisa.isatab</i>), 33	STUDY_FACTOR_TYPE	(in module <i>altamisa.constants.investigation_headers</i>), 36
STUDY	(in module <i>altamisa.constants.investigation_headers</i>), 34	STUDY_FACTOR_TYPE_TERM_ACCESSION_NUMBER	(in module <i>altamisa.constants.investigation_headers</i>), 36
STUDY_ASSAY_FILE_NAME	(in module <i>altamisa.constants.investigation_headers</i>), 36	STUDY_FACTOR_TYPE_TERM_SOURCE_REF	(in module <i>altamisa.constants.investigation_headers</i>), 36
STUDY_ASSAY_MEASUREMENT_TYPE	(in module <i>altamisa.constants.investigation_headers</i>), 36	STUDY_FACTORS	(in module <i>altamisa.constants.investigation_headers</i>), 36
STUDY_ASSAY_MEASUREMENT_TYPE_TERM_ACCESSION_NUMBER	(in module <i>altamisa.constants.investigation_headers</i>), 36	STUDY_FACTORS_KEYS	(in module <i>altamisa.constants.investigation_headers</i>), 36
STUDY_ASSAY_MEASUREMENT_TYPE_TERM_SOURCE_REF	(in module <i>altamisa.constants.investigation_headers</i>), 36	STUDY_FILE_NAME	(in module <i>altamisa.constants.investigation_headers</i>), 35
STUDY_ASSAY_TECHNOLOGY_PLATFORM	(in module <i>altamisa.constants.investigation_headers</i>), 36	STUDY_IDENTIFIER	(in module <i>altamisa.constants.investigation_headers</i>), 35
STUDY_ASSAY_TECHNOLOGY_TYPE	(in module <i>altamisa.constants.investigation_headers</i>), 36	STUDY_INFO_KEYS	(in module <i>altamisa.constants.investigation_headers</i>), 35
STUDY_ASSAY_TECHNOLOGY_TYPE_TERM_ACCESSION_NUMBER	(in module <i>altamisa.constants.investigation_headers</i>), 36	STUDY_PERSON_ADDRESS	(in module <i>altamisa.constants.investigation_headers</i>), 35
STUDY_ASSAY_TECHNOLOGY_TYPE_TERM_SOURCE_REF	(in module <i>altamisa.constants.investigation_headers</i>), 36		
STUDY_ASSAYS	(in module <i>altamisa.constants.investigation_headers</i>), 34		
STUDY_ASSAYS_KEYS	(in module <i>altamisa.constants.investigation_headers</i>), 36		
STUDY_CONTACTS	(in module <i>altamisa.constants.investigation_headers</i>), 36		

37	(in module al-	(in module al-	al-
STUDY_PERSON_AFFILIATION (in module al-	tamisa.constants.investigation_headers), 37	tamisa.constants.investigation_headers), 36	
STUDY_PERSON_EMAIL (in module al-	tamisa.constants.investigation_headers), 37	STUDY_PROTOCOL_TYPE (in module al-	al-
		tamisa.constants.investigation_headers), 36	
STUDY_PERSON_FAX (in module al-	tamisa.constants.investigation_headers), 37	STUDY_PROTOCOL_TYPE_TERM_ACCESSION_NUMBER	
		(in module al-	al-
STUDY_PERSON_FIRST_NAME (in module al-	tamisa.constants.investigation_headers), 36	tamisa.constants.investigation_headers), 36	
STUDY_PERSON_LAST_NAME (in module al-	tamisa.constants.investigation_headers), 36	STUDY_PROTOCOL_TYPE_TERM_SOURCE_REF	
		(in module al-	al-
STUDY_PERSON_MID_INITIALS (in module al-	tamisa.constants.investigation_headers), 36	tamisa.constants.investigation_headers), 36	
STUDY_PERSON_PHONE (in module al-	tamisa.constants.investigation_headers), 37	STUDY_PROTOCOL_URI (in module al-	al-
		tamisa.constants.investigation_headers), 36	
STUDY_PERSON_ROLES (in module al-	tamisa.constants.investigation_headers), 37	STUDY_PROTOCOL_VERSION (in module al-	al-
		tamisa.constants.investigation_headers), 36	
STUDY_PERSON_ROLES_TERM_ACCESSION_NUMBER		STUDY_PROTOCOLS (in module al-	al-
(in module al-		tamisa.constants.investigation_headers), 34	
tamisa.constants.investigation_headers), 37		STUDY_PROTOCOLS_KEYS (in module al-	al-
STUDY_PERSON_ROLES_TERM_SOURCE_REF		tamisa.constants.investigation_headers), 36	
(in module al-		STUDY_PUBLIC_RELEASE_DATE (in module al-	al-
tamisa.constants.investigation_headers), 37		tamisa.constants.investigation_headers), 35	
STUDY_PROTOCOL_COMPONENTS_NAME (in module	altamisa.constants.investigation_headers), 36	STUDY_PUBLICATION_AUTHOR_LIST (in module	altamisa.constants.investigation_headers), 35
STUDY_PROTOCOL_COMPONENTS_TYPE (in module	altamisa.constants.investigation_headers), 36	STUDY_PUBLICATION_DOI (in module al-	al-
STUDY_PROTOCOL_COMPONENTS_TYPE_TERM_ACCESSION_NUMBER		tamisa.constants.investigation_headers), 35	
(in module al-		STUDY_PUBLICATION_STATUS (in module al-	al-
tamisa.constants.investigation_headers), 36		altamisa.constants.investigation_headers), 35	
STUDY_PROTOCOL_COMPONENTS_TYPE_TERM_SOURCE_REF		STUDY_PUBLICATION_STATUS_TERM_ACCESSION_NUMBER	
(in module al-		(in module al-	al-
tamisa.constants.investigation_headers), 36		tamisa.constants.investigation_headers), 35	
STUDY_PROTOCOL_DESCRIPTION (in module al-	tamisa.constants.investigation_headers), 36	STUDY_PUBLICATION_STATUS_TERM_SOURCE_REF	
STUDY_PROTOCOL_NAME (in module al-	tamisa.constants.investigation_headers), 36	(in module al-	al-
		tamisa.constants.investigation_headers), 35	
STUDY_PROTOCOL_PARAMETERS_NAME (in module	altamisa.constants.investigation_headers), 36	STUDY_PUBLICATION_TITLE (in module al-	al-
STUDY_PROTOCOL_PARAMETERS_NAME_TERM_ACCESSION_NUMBER		tamisa.constants.investigation_headers), 35	
(in module al-		STUDY_PUBLICATIONS (in module al-	al-
tamisa.constants.investigation_headers), 36		tamisa.constants.investigation_headers), 34	
STUDY_PROTOCOL_PARAMETERS_NAME_TERM_SOURCE_REF		STUDY_PUBLICATIONS_KEYS (in module al-	al-
		altamisa.constants.investigation_headers), 36	
STUDY_PROTOCOL_PARAMETERS_NAME_TERM_SOURCE_REF		STUDY_PUBMED_ID (in module al-	al-
		tamisa.constants.investigation_headers), 35	
STUDY_PROTOCOL_PARAMETERS_NAME_TERM_SOURCE_REF		STUDY_PUBSUBMISSION_DATE (in module al-	al-

tamisa.constants.investigation_headers),
35
STUDY_TITLE (in module
tamisa.constants.investigation_headers),
35
StudyHeaderParser (class in
tamisa.isatab.headers), 43
StudyInfo (class in *altamisa.isatab*), 29
StudyReader (class in *altamisa.isatab*), 19
StudyRowReader (class in *altamisa.isatab*), 20
StudyValidator (class in *altamisa.isatab*), 23
StudyWriter (class in *altamisa.isatab*), 21
submission_date (*altamisa.isatab.BasicInfo* at-
tribute), 25

T

tail (*altamisa.isatab.Arc* attribute), 32
technology_type (*altamisa.isatab.AssayInfo* at-
tribute), 28
TERM_ACCESSION_NUMBER (in module
tamisa.constants.table_headers), 38
TERM_SOURCE_DESCRIPTION (in module
tamisa.constants.investigation_headers), 34
TERM_SOURCE_FILE (in module
tamisa.constants.investigation_headers),
34
TERM_SOURCE_NAME (in module
tamisa.constants.investigation_headers),
34
TERM_SOURCE_REF (in module
tamisa.constants.table_headers), 38
term_source_ref_header (al-
tamisa.isatab.headers.ColumnHeader at-
tribute), 40
TERM_SOURCE_VERSION (in module
tamisa.constants.investigation_headers),
34
TermRefAnnotationHeader (class in al-
tamisa.isatab.headers), 42
title (*altamisa.isatab.BasicInfo* attribute), 25
title (*altamisa.isatab.PublicationInfo* attribute), 26
TOKEN_ANONYMOUS (in module al-
tamisa.constants.table_tokens), 39
TOKEN_EMPTY (in module al-
tamisa.constants.table_tokens), 39
TOKEN_UNKNOWN (in module al-
tamisa.constants.table_tokens), 39
type (*altamisa.isatab.DesignDescriptorsInfo* attribute),
27
type (*altamisa.isatab.FactorInfo* attribute), 27
type (*altamisa.isatab.Material* attribute), 31
type (*altamisa.isatab.ProtocolComponentInfo* at-
tribute), 28
type (*altamisa.isatab.ProtocolInfo* attribute), 28

U

unique_name (*altamisa.isatab.Material* attribute), 31
unique_name (*altamisa.isatab.Process* attribute), 32
unit (*altamisa.isatab.Characteristics* attribute), 30
unit (*altamisa.isatab.FactorValue* attribute), 30
unit (*altamisa.isatab.ParameterValue* attribute), 31
UNIT (in module *altamisa.constants.table_headers*), 38
unit_header (*altamisa.isatab.headers.ColumnHeader*
attribute), 40
UnitHeader (class in *altamisa.isatab.headers*), 43
uri (*altamisa.isatab.ProtocolInfo* attribute), 28

V

validate() (*altamisa.isatab.AssayValidator* method),
22
validate() (*altamisa.isatab.InvestigationValidator*
method), 22
value (*altamisa.isatab.Characteristics* attribute), 30
value (*altamisa.isatab.Comment* attribute), 25
value (*altamisa.isatab.FactorValue* attribute), 30
value (*altamisa.isatab.ParameterValue* attribute), 30
version (*altamisa.isatab.OntologyRef* attribute), 25
version (*altamisa.isatab.ProtocolInfo* attribute), 29

W

write() (*altamisa.isatab.InvestigationWriter* method),
21
WriteIsatabException, 44
WriteIsatabWarning, 44