Sample and Data Relationship Format for Proteomics (SDRF-Proteomics)

Version 1.1.0-dev, 2026-01-24

Table of Contents

Chapter 1. Status of this document

This document provides information to the proteomics community about a proposed standard for sample metadata annotations in public repositories called Sample and Data Relationship Format (SDRF)-Proteomics. Distribution is unlimited.

Version v1.1.0 - 2025-01

Chapter 2. Abstract

The Human Proteome Organisation (HUPO) Proteomics Standards Initiative (PSI) defines community standards for data representation in proteomics to facilitate data comparison, exchange, and verification. This document presents a specification for the Sample and Data Relationship Format (SDRF-Proteomics).

Further detailed information, including any updates to this document, implementations, and examples is available at SDRF GitHub Repository. The official PSI web page for the document is: HUPO-PSI SDRF.

Chapter 3. Motivation

Many resources have emerged that provide raw or integrated proteomics data in the public domain. If these are valuable individually, their integration through re-analysis represents a huge asset for the community [1].

Unfortunately, proteomics experimental design and sample related information are often missing in public repositories or stored in very diverse ways and formats. For example:

- The CPTAC Consortium provides for every dataset a set of Excel files with the information on each sample including tumor size, origin, but also how every sample is related to a specific raw file (e.g. instrument configuration parameters).
- As a resource routinely re-analysing public datasets, ProteomicsDB captures for each sample in the database a minimum number of properties to describe the sample and the related experimental protocol such as tissue, digestion method and instrument.

Such heterogeneity often prevents data interpretation, reproducibility, and integration of data from different resources. For every proteomics dataset we propose to capture at least three levels of metadata:

- (i) dataset description
- (ii) the sample metadata and data files acquisition metadata.
- (iii) The relation between the sample and the data files. The experimental design.

The general description includes minimum information to describe the study overall: title, description, date of publication, type of experiment. In ProteomeXchange partners this metadata is captured at the dataset level, in other omics resources this is captured as IDF file format (e.g. MAGE-TAB). Currently, all ProteomeXchange partners mandate this information for each dataset. However, the information regarding the sample and its relation to the data files (**Figure 1**) is mostly missing [1].

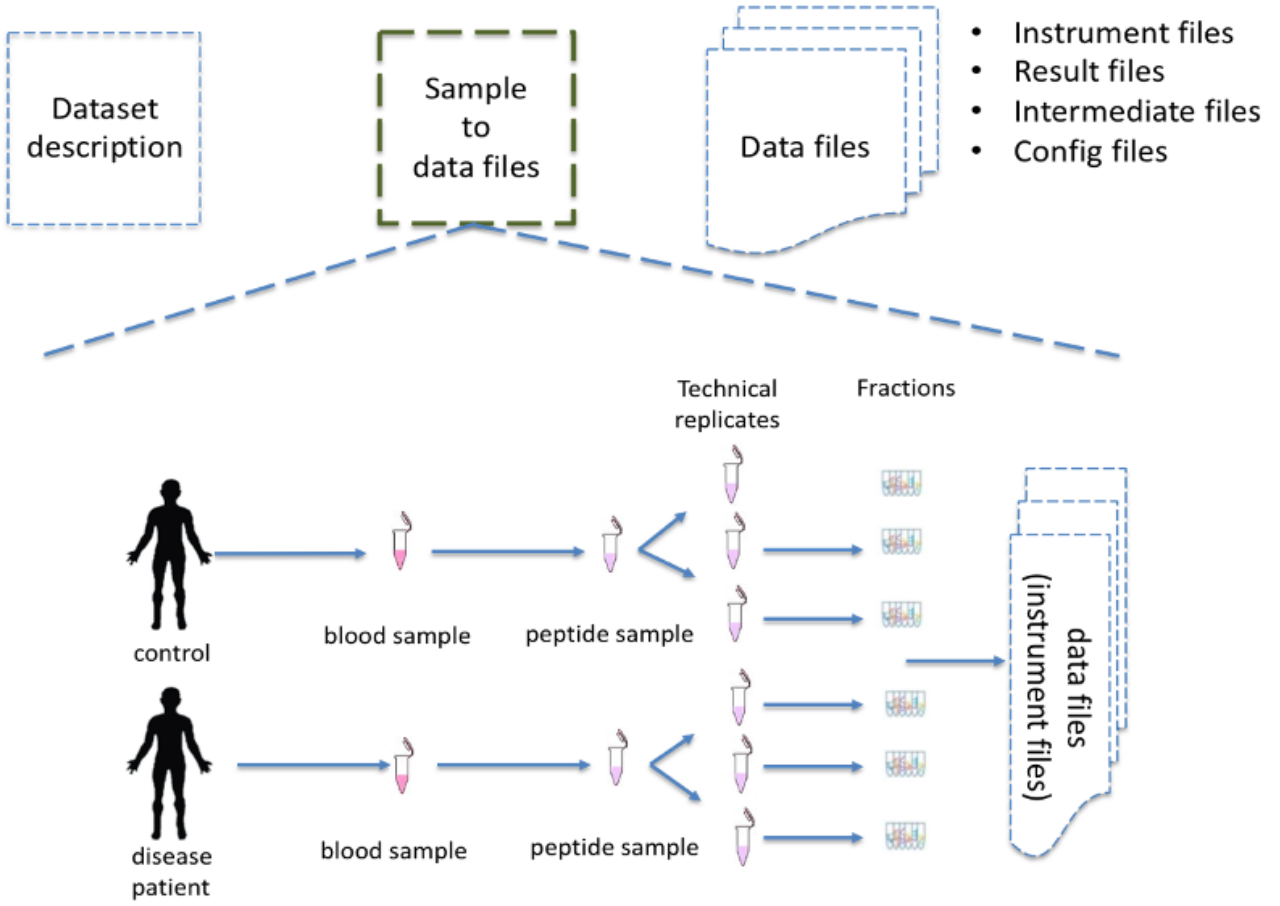

Figure 1: SDRF-Proteomics file format stores the information of the sample and its relation to the data files in the dataset. The file format includes not only information about the sample but also about how the data was acquired and processed.

Here, we introduced the Sample and Data Relationship Format (SDRF-Proteomics) to capture the sample metadata and its relation to the data files for proteomics experiments. The SDRF-Proteomics format is a tab-delimited file format that describes the sample characteristics and the relationships between samples and data files included in a dataset.

This specification, which is a community effort, aims to provide a standard for the proteomics community to annotate the sample metadata and its relation to the data files.

Chapter 4. Quick Start

If you're new to SDRF-Proteomics, here's a minimal example to get you started. An SDRF file is a tab-separated file where each row represents a sample-to-data-file relationship.

4.1. Minimal Example

source name	characteristics[organism]	characteristics[organism part]	characteristics[disease]	characteristics[biological replicate]	assay name	technology type	comment[proteomics data acquisition method]	comment[label]	comment[instrument]	comment[cleavage agent details]	comment[fraction identifier]	comment[technical replicate]	comment[data file]	factor value[disease]
sample_1	homo sapiens	liver	normal	1	run_1	proteomic profiling by mass spectrometry	Data-dependent acquisition	label free sample	Q Exactive HF	NT=Trypsin;AC=MS:1001251	1	1	sample_1.raw	normal
sample_2	homo sapiens	liver	hepatocellular carcinoma	1	run_2	proteomic profiling by mass spectrometry	Data-dependent acquisition	label free sample	Q Exactive HF	NT=Trypsin;AC=MS:1001251	1	1	sample_2.raw	hepatocellular carcinoma

Blue: Sample metadata | Green: Data file metadata | Orange: Factor values

NOTE

This minimal example shows a **mass spectrometry-based** proteomics experiment. For detailed requirements see the MS-Proteomics template. For **affinity-based proteomics** (Olink, SomaScan), different columns are required - see the Affinity-Proteomics template.

4.2. Key Concepts

1. **Sample metadata** uses `characteristics[…]` columns (e.g., organism, disease)
2. **Data file metadata** uses `comment[…]` columns (e.g., instrument, label)
3. **Factor values** use `factor value[…]` columns to indicate variables under study
4. Each row links one sample to one data file

4.3. Format Requirements

The SDRF-Proteomics format has the following core requirements:

- The SDRF file is a **tab-delimited format** where each row corresponds to a relationship between a Sample and a Data file.
- Each column MUST correspond to an attribute/property of the Sample or the Data file.
- Each cell value MUST be the property value for the corresponding Sample or Data file.
- The file MUST start with columns describing sample properties (e.g., organism, disease), followed by data file properties (e.g., label, fraction identifier, data file).
- Unknown values MUST be handled using reserved words: `not available` (value is unknown), `not applicable` (property doesn't apply), or `pooled` (value is a mixture from multiple samples).

4.4. Scope

The SDRF-Proteomics format aims to capture the **sample metadata** and its **relationship with data files** (e.g., raw files from mass spectrometers).

IMPORTANT	SDRF-Proteomics does **not** aim to capture downstream analysis details, including: which samples were compared to which other samples, how samples are combined into study variables, or analysis parameters such as FDR thresholds or p-value cutoffs.

4.5. Getting Started Steps

1. Choose a technology template based on your technology type: `ms-proteomics` for *proteomic profiling by mass spectrometry*, `affinity-proteomics` for *protein expression profiling by antibody/aptamer array* (Olink, SomaScan)
2. Add the appropriate sample template for your organism (Human, Vertebrates, Invertebrates, Plants, or Cell-lines)
3. Fill in sample metadata (characteristics columns)
4. Fill in data file metadata (comment columns)
5. Add factor values for your experimental variables
6. Validate your file using sdrf-pipelines

For detailed guidance, continue reading the full specification below.

Chapter 5. Validating SDRF Files

The official validator for SDRF-Proteomics files is **sdrf-pipelines**, a Python tool that checks your SDRF file for errors and compliance with the specification.

Installation:

```
pip install sdrf-pipelines
```

Basic Validation:

```
# Validate an SDRF file
parse_sdrf validate-sdrf --sdrf_file your_file.sdrf.tsv

# Validate with a specific template
parse_sdrf validate-sdrf --sdrf_file your_file.sdrf.tsv --template human
```

For more information, visit: sdrf-pipelines on GitHub

Chapter 6. Specification structure

This document describes the main specification of SDRF-Proteomics, the structure of the specification (**Figure 2**), how to contribute, and extend the specification. SDRF-Proteomics uses a three-tier system for organizing metadata requirements:

- The SDRF-Proteomics core specification: This document contains the main specification, requirements and rules for the SDRF-Proteomics format. It also includes the notational conventions and the relationship to other specifications.
- Templates: All templates are organized in the `templates/` directory. This includes:
  - **Core templates** (organism-based): human, vertebrates, invertebrates, plants
  - **Specialized templates** (experiment-type): cell-lines (sample), DDA acquisition (data), DIA acquisition (data), single-cell (mixed), affinity-proteomics (data), crosslinking (data), immunopeptidomics (mixed), metaproteomics (mixed)

Each template has its own directory containing a detailed README.adoc with checklists and examples, plus a template file ({name}-template.sdrf.tsv) with column headers. See the Templates section.

[Logo] | *images/sdrf-guidelines-structure.png*

Figure 2: SDRF-Proteomics specification structure. The main specification defines the core rules and is extended by technology templates (ms-proteomics, affinity-proteomics), sample templates (human, vertebrates, etc.), and specialized experiment-type templates.

> **NOTE** The main specification is in the `sdrf-proteomics` directory. All templates are organized in `sdrf-proteomics/templates/`. Templates are extensions of the core specification and should follow all the rules and requirements in the main specification. Each template includes its own detailed documentation with checklists, examples, and annotation guidelines specific to that template type.

The official website for SDRF-Proteomics project is https://github.com/bigbio/proteomics-metadata-standard. New use cases, changes to the specification and examples can be added by using Pull requests or issues in GitHub to reach the bigbio team.

A set of examples and annotated projects from ProteomeXchange can be found here: https://github.com/bigbio/proteomics-metadata-standard/tree/master/annotated-projects

Multiple tools have been implemented to validate, annotate and convert SDRF-Proteomics files. The official validator of SDRF-Proteomics is sdrf-pipelines (Python - https://github.com/bigbio/sdrf-pipelines). This tool allows to validate an SDRF-Proteomics file. In addition, it allows converting SDRF to other popular pipelines and software configure files such as MaxQuant or OpenMS.

6.1. Versioning

The SDRF-Proteomics specification is versioned using the Semantic Versioning 2.0.0 (https://semver.org/) scheme. The version number is in the format MAJOR.MINOR.PATCH, where:

- MAJOR version is incremented for incompatible changes to the specification, when major changes are done to the specification.
- MINOR version is incremented for new features that are backward compatible with the previous version. Guidelines and templates are added or modified.
- PATCH version is incremented for bug fixes and minor changes that do not affect the specification or the templates. This includes typos, formatting changes, and other minor updates.

Every change in the specification should be done in GitHub using pull requests into the dev branch. The pull request should include a description of the changes and the reason for the changes. The pull request will be reviewed by the community and merged into the main branch when approved. After the merge, the version number will be updated according to the changes made, the release will be performed, and the Zenodo record will be updated.

NOTE We added the prefix v to the version number to indicate that it is the version of the specification that was used to create the file. Examples: v1.1.0, v2.0.0, v3.0.0.

6.2. Notational Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMEND/RECOMMENDED”, “MAY”, “COULD BE”, and “OPTIONAL” are to be interpreted as described in RFC 2119 (https://www.rfc-editor.org/rfc/rfc2119).

6.3. Relationship to other specifications

SDRF-Proteomics is fully compatible with the SDRF file format part of MAGE-TAB. MAGE-TAB is the file format used to store metadata and sample information for transcriptomics experiments. When the ProteomeXchange project file is converted to idf file (project description in MAGE-TAB) and is combined with the SDRF-Proteomics a valid MAGE-TAB is obtained.

SDRF-Proteomics sample information can be embedded into mzTab metadata files. The sample metadata in mzTab contains properties as the columns in the SDRF-Proteomics and values as Sample cell values.

The SDRF-Proteomics aims to capture the sample metadata and its relationship with the data files (e.g. raw files from mass spectrometers). The SDRF-Proteomics do not aim to capture the downstream analysis part of the experimental design such as what samples should be compared, how they can be combined or parameters for the downstream analysis (FDR or p-values thresholds). The HUPO-PSI community will work in the future to include this information in other file formats such as mzTab or a new type of file format.

Chapter 7. SDRF-Proteomics specification

The SDRF-Proteomics file format describes the sample characteristics and the relationships between samples and data files. The file format is a tab-delimited one where each **ROW** corresponds to a relationship between a Sample and a Data file (in an ms proteomics experiment the data file containing the mass spectra), each **COLUMN** corresponds to an attribute/property of the Sample, the Data file, or the Factor values; and the value in each **CELL** is the specific value of the property for a given Sample/Data file/Factor value (**Figure 3**).

source name	characteristics[organism]	characteristics[…]	characteristics[biological replicate]	assay name	technology type	comment […]	comment [data file]	factor value[…]
sample_1	homo sapiens	…	1	run_1	proteomic profiling by mass spectrometry	…	sample_1.raw	…
sample_2	homo sapiens	…	2	run_2	proteomic profiling by mass spectrometry	…	sample_2.raw	…

Blue: Sample metadata (characteristics) | *Green: Data file metadata (comments)* | *Orange: Factor values*

Figure 3: SDRF-Proteomics in a nutshell. Each **row** links a sample to a data file. **Columns** represent sample properties (characteristics), data file properties (comments), or experimental variables (factor values).

The SDRF-Proteomics format contains three main sections:

- The first section contains the sample metadata.
- The second section contains the data file metadata.
- The third section contains the factor values properties.

7.1. Format rules

- **Case sensitivity**: Text values are case-insensitive, but **column names are case-sensitive**. Use lowercase for all column names (e.g., `source name`, `characteristics[organism]`, `comment[label]`). Incorrect casing like `Source Name` or `Characteristics[organism]` will cause validation failures.
- **Space sensitivity**: The SDRF is sensitive to spaces in column names (`sourcename` ¬ `source name`). Column names must include appropriate spaces (e.g., `source name`, not `sourcename`) but must NOT have a space before the bracket (e.g., `characteristics[organism]`, not `characteristics [organism]`).

- **Column order**: The SDRF columns follows some structure; first the sample metadata columns in Chapter 8; then the data file metadata columns in Chapter 9; followed by the factor values columns in Section 11.9.
- **Extension**: The extension of the SDRF file SHOULD be **sdrf.tsv (preferred)** or .txt.

7.2. Reserved words

There are general scenarios where cell values cannot be provided with actual data. The following reserved words MUST be used in these cases:

- **not available**: In some cases, the column is mandatory in the format, but for some samples the corresponding value is unknown or could not be determined. In those cases, users SHOULD use **not available**.
- **not applicable**: In some cases, the column is mandatory, but for some samples the corresponding value or concept does not apply. In those cases, users SHOULD use **not applicable**.
- **anonymized**: In some cases, the value exists but has been intentionally redacted for privacy protection (e.g., in clinical studies with de-identified patient data). In those cases, users SHOULD use **anonymized**.
- **pooled**: In some cases, the sample is a pool of multiple samples (e.g., TMT reference channels), and the value cannot be represented as a single value. In those cases, users SHOULD use **pooled**.

Table 1. Reserved words for SDRF cell values

Term	Meaning	Example	Use Case
not available	Value exists but is unknown or could not be determined	characteristics[age] = not available	Patient age was not recorded in the study
not applicable	Value or concept does not apply to this sample	characteristics[age] = not applicable	Synthetic peptide library has no age
anonymized	Value exists but is redacted for privacy protection	characteristics[age] = anonymized	Clinical study with de-identified patient data
pooled	Value represents a mixture of multiple samples	characteristics[biological replicate] = pooled	TMT reference channel pooled from multiple replicates

7.3. SDRF file-level metadata

Since version 1.1.0, SDRF-Proteomics supports optional file-level metadata using header comments at the beginning of the file. These header comments provide information about the SDRF file itself, such as the format version, template used, and validation status. This approach is inspired by other omics formats such as VCF (Variant Call Format) file headers and is fully compatible with pandas and other tabular data processing tools.

Header comments MUST:

- Start with `#` (single hash) followed by a key-value pair
- Appear at the very beginning of the file, before the column header row
- Use the format `#key=value`

The following header fields are supported:

Key	Description	Example	Requirement	Ontology Term
`file_format`	Identifier for the file format	SDRF	RECOMMENDED	PRIDE:0000831
`version`	SDRF-Proteomics specification version used	v1.1.0	RECOMMENDED	NCIT:C25714
`template`	Template(s) used, comma-separated. Only list leaf templates; parent templates are implied by inheritance	human,crosslinking	OPTIONAL	PRIDE:0000832
`template_version`	Version(s) of the template(s). Use single value if all same; comma-separated only when versions differ	v1.1.0	OPTIONAL	PRIDE:0000833
`source`	Tool or origin that generated the file. Replaces per-row `comment[tool metadata]` column	lesSDRF v0.1.0	OPTIONAL	NCIT:C25683
`validation_hash`	Hash from validator certification	sha256:abc123…	OPTIONAL	PRIDE:0000834

Example of an SDRF file with header comments (simplified example showing only select columns; see Chapter 11 for complete required columns):

```
#file_format=SDRF
#version=v1.1.0
#template=human
#template_version=v1.1.0
#source=lesSDRF v0.1.0
source name characteristics[organism]   characteristics[organism part]
characteristics[disease]    assay name  comment[data file]
sample_1    homo sapiens    liver   normal  run_1   sample_1.raw
```

7.3.1. Template Inheritance and Versioning Rules

Template inheritance: Only list leaf templates in the `#template` field. Parent templates are implied:

- `crosslinking` inherits from `ms-proteomics` ¬ use `#template=human,crosslinking` (not `ms-`

`proteomics,human,crosslinking`)

- `olink` inherits from `affinity-proteomics` ¬ use `#template=human,olink` (not `affinity-proteomics,olink,human`)

Version simplification:

- If all templates share the same version, use a single value: `#template_version=v1.1.0`
- Only use comma-separated versions when templates have different versions

```
# Same version for all templates - use single value
#template=human,crosslinking
#template_version=v1.1.0

# Different versions - list each (matching template order)
#template=human,crosslinking
#template_version=v1.1.0,v1.0.0
```

Source field: The `#source` header replaces the per-row `comment[tool metadata]` column. Use it to record the tool or origin that generated the SDRF file (e.g., `lesSDRF v0.1.0`, `PRIDE`, `user-generated`).

> **NOTE**
>
> Header comments are OPTIONAL. SDRF files without header comments are still valid. When present, header comments provide valuable provenance information and enable tools to handle version-specific features appropriately. Header property names use underscores (e.g., `file_format`, `template_version`) rather than spaces to maintain consistency with the tab-delimited nature of SDRF files and avoid ambiguity when parsing.

7.4. Table Column headers

Depending on each section the column headers (property names) will be prefixed with the following prefixes:

- `characteristics`: Sample metadata (e.g. *characteristics[organism]*)
- `comment`: Data file metadata (e.g. *comment[data file]*)
- `factor value`: Factor values properties (e.g. *factor value[disease]*)

Each property name MUST be a valid ontology term or a valid controlled vocabulary term. Each section will have some specific order for column headers.

> **NOTE**
>
> A list of all controlled vocabularies and ontologies supported are in the Chapter 13 section. On each section we also provide a list of properties that are supported.

7.5. Table Cell values

The value for each property, (e.g. characteristics, comment, factor value) corresponding to each sample or data file can be represented in multiple ways.

- **Free Text (Human readable)**: In the free text representation, the value is provided as text without Ontology support (e.g. colon or providing accession numbers). This is only RECOMMENDED when the text inserted in the table is the exact name of an ontology/CV term in EFO. If the term is not in EFO, other ontologies can be used.

source name	characteristics[organism]
sample 1	homo sapiens
sample 2	homo sapiens

- **Ontology url (Computer readable)**: Users can provide the corresponding URI (Uniform Resource Identifier) of the ontology/CV term as a value. This is recommended for enriched files where the user does not want to use intermediate tools to map from free text to ontology/CV terms.

source name	characteristics[organism]
Sample 1	http://purl.obolibrary.org/obo/NCBITaxon_9606
Sample 2	http://purl.obolibrary.org/obo/NCBITaxon_9606

- **Key=value representation (Human and Computer readable)**: The current representation aims to provide a mechanism to represent the complete information of the ontology/CV term including Accession, Name and other additional properties. In the key=value pair representation, the Value of the property is represented as an Object with multiple properties, where the key is one of the properties of the object and the value is the corresponding value for the particular key. An example of key value pairs is post-translational modification (see Protein Modifications):

```
NT=Glu->pyro-Glu;MT=fixed;PP=Anywhere;AC=Unimod:27;TA=E
```

Chapter 8. SDRF-Proteomics: Samples metadata

The Sample metadata section provides information about the samples of origin and their characteristics. Each sample contains a *source name* (unique identifier) and a set of *characteristics* columns. The first column of the file should be the *source name* and the following columns should be the characteristics of the sample. For example, for any proteomics experiment (human, vertebrate, cell line), the following characteristics should be provided:

- **source name**: Unique sample name (it can be present multiple times if the same sample is used several times in the same dataset)
- **characteristics[organism]**: The organism of the Sample of origin. Values MUST come from NCBI Taxonomy.
- **characteristics[organism part]**: The part of organism's anatomy or substance arising from an organism from which the biomaterial was derived (e.g., liver). Values SHOULD come from UBERON or BTO.
- **characteristics[disease]**: The disease under study in the Sample. Values SHOULD come from MONDO, EFO, or DOID. For healthy/control samples, use `normal` (PATO:0000461) - see Disease Annotation Guidelines.
- **characteristics[cell type]**: A cell type is a distinct morphological or functional form of cell (e.g., epithelial, glial). Values SHOULD come from Cell Ontology (CL) or BTO.

Example:

source name	characteristics[organism]	characteristics[organism part]	characteristics[disease]	characteristics[cell type]
sample_treat	homo sapiens	liver	liver cancer	not available
sample_control	homo sapiens	liver	liver cancer	not available

NOTE

Additional characteristics can be added depending on the type of the experiment and sample. The SDRF-Proteomics templates defines a set of templates and checklists of properties that should be provided depending on the proteomics experiment. In the core guidelines and templates, main document of SDRF-Proteomics, we explain the major sample properties for different experiments. However, SDRF-Proteomics can be extended using guidelines for specific experiments.

IMPORTANT

Each characteristic name in the column header SHOULD be a CV term from the EFO ontology. For example, the header `characteristics[organism]` corresponds to the ontology term Organism. However the values could be from EFO or other ontologies. For example, we RECOMMEND to use MONDO for diseases because it has better coverage than EFO. For healthy samples, use `normal` (PATO:0000461) - see Disease Annotation Guidelines.

IMPORTANT

Multiple values (columns) for the same `characteristics` term are allowed in SDRF-Proteomics (see Section 10.1). However, it is RECOMMENDED not to use the same column in the same file. If you have multiple phenotypes, you can

> specify what it refers to or use another more specific term, e.g., "immunophenotype".

8.1. BioSamples database integration

BioSamples provides persistent identifiers for biological samples that enable cross-database linking [5]. Use the optional *characteristics[biosample accession number]* column to link samples to BioSamples entries (EBI format: SAMEA*, NCBI format: SAMN*).

Column: `characteristics[biosample accession number]`

Examples: `SAMN12345678` (NCBI), `SAMEA12345678` (EBI)

Example usage:

source name	characteristics[biosample accession number]	characteristics[organism]	characteristics[organism part]	characteristics[disease]
sample_001	SAMN12345678	homo sapiens	liver	liver cancer
sample_002	SAMN12345679	homo sapiens	liver	normal
sample_003	SAMEA12345680	mus musculus	brain	normal

The BioSamples accession number enables connection to sample resources across databases, linking proteomics data with genomics, transcriptomics, and other omics datasets.

NOTE

> BioSample accession numbers from NCBI follow the format `SAMNxxxxxxxxxx` and from EBI follow the format `SAMEAxxxxxxxxx`, where `x` represents digits. Either NCBI or EBI BioSample accession numbers can be used depending on where the sample is registered. The *characteristics[biosample accession number]* column is OPTIONAL, but when available, providing BioSample accession numbers is RECOMMENDED to enhance data integration and reusability. Users must first request BioSample accession numbers from the appropriate service (NCBI or EBI) before including them in their SDRF files.

8.2. Encoding sample technical and biological replicates

SDRF-Proteomics uses two columns to track replicates [4]:

- **characteristics[biological replicate]**: Identifies independent biological samples within each experimental condition (factor value group). Replicate numbers are assigned per condition, so if you have 2 cancer samples and 2 healthy samples, both groups would have biological replicates numbered 1 and 2.
- **comment[technical replicate]**: Identifies repeated measurements of the same sample (e.g., multiple LC-MS/MS injections)

IMPORTANT

Biological replicate numbering restarts for each experimental condition (factor value). For example, in a disease study with factor value[disease], the cancer samples would be numbered 1, 2, 3… and the normal samples would independently be numbered 1, 2, 3… This establishes the relationship between each sample and its experimental condition.

The following example shows 2 biological replicates, each with 2 fractions and 2 technical replicates:

source name	characteristics[biological replicate]	assay name	comment[label]	comment[fraction identifier]	comment[technical replicate]	comment[data file]
patient_001_sample	1	run_01	label free sample	1	1	P001_F1_TR1.raw
patient_001_sample	1	run_02	label free sample	2	1	P001_F2_TR1.raw
patient_001_sample	1	run_03	label free sample	1	2	P001_F1_TR2.raw
patient_001_sample	1	run_04	label free sample	2	2	P001_F2_TR2.raw
patient_002_sample	2	run_05	label free sample	1	1	P002_F1_TR1.raw
patient_002_sample	2	run_06	label free sample	2	1	P002_F2_TR1.raw
patient_002_sample	2	run_07	label free sample	1	2	P002_F1_TR2.raw
patient_002_sample	2	run_08	label free sample	2	2	P002_F2_TR2.raw

In this example:

- **Biological replicates**: `patient_001_sample` and `patient_002_sample` are different biological samples (different source names), annotated with `characteristics[biological replicate]` values 1 and 2
- **Technical replicates**: Each biological sample is measured twice (`comment[technical replicate]` = 1 and 2)
- **Fractions**: Each technical replicate has 2 fractions (`comment[fraction identifier]` = 1 and 2)

IMPORTANT

Both *characteristics[biological replicate]* and *comment[technical replicate]* columns are REQUIRED. When no replicates are performed in a study, set both columns to 1 (i.e., each sample is biological replicate 1 and technical replicate 1). For **pooled samples** (e.g., TMT reference channels), use `pooled` for biological replicate since these samples are mixtures of multiple replicates and assigning a specific replicate number would be misleading.

Some examples with explicit annotation of the biological replicates can be found here:

- https://github.com/bigbio/proteomics-metadata-standard/blob/c3a56b076ef381280dfcb0140d2520126ace53ff/annotated-projects/PXD006401/PXD006401.sdrf.tsv

8.3. Pooled samples

When multiple samples are pooled into one (e.g., TMT/iTRAQ reference channels for normalization), use the *characteristics[pooled sample]* column to indicate pooling status. Allowed values:

- **not pooled**: Regular individual samples
- **pooled**: Sample is pooled but individual sources are unknown
- **SN=sample1;SN=sample2;…**: Lists source names of pooled samples when known

Example:

source name	characteristics[pooled sample]	characteristics[organism]	characteristics[age]	comment[label]	comment[data file]
sample_1	not pooled	homo sapiens	45Y	TMT126	file01.raw
sample_2	not pooled	homo sapiens	52Y	TMT127N	file01.raw
pooled_ref	SN=sample_1;SN=sample_2	homo sapiens	pooled	TMT131C	file01.raw

TIP For pooled samples, use `pooled` for individual-specific fields (biological replicate, age, sex) to indicate a mixture rather than a single sample.

8.4. Sample Metadata Guidelines

For detailed guidance on annotating sample metadata, refer to the following conventions documents:

- Sample Metadata Guidelines - Detailed guidelines for age, sex, disease, organism part, cell type, developmental stage, spiked-in samples, and other sample characteristics
- Human Sample Metadata Guidelines - Human-specific metadata including disease staging, treatment history, demographics, and lifestyle factors

Chapter 9. SDRF-Proteomics: data files metadata

The connection between samples and data files is done using properties annotated with the `comment` prefix. All properties referring to a data file (e.g., MS run file) are annotated with the category `comment`. This differentiates data file properties from sample properties (characteristics).

9.1. CV Term Format for Data File Metadata

For data file metadata (`comment` columns) that reference ontology terms, use the structured format: `NT={term name};AC={accession}`

Examples: `NT=HCD;AC=PRIDE:0000590`, `NT=Orbitrap;AC=MS:1000484`

This format enables automated validation and software extraction from raw files. Sample metadata (characteristics) can use simple term names since they are typically human-annotated.

The following properties MUST be provided for each data file in **mass spectrometry-based proteomics** experiments. For **affinity-based proteomics** (Olink, SomaScan), see the Affinity-Proteomics template for different required columns.

Column	Requirement	Description	Ontology
`assay name`	REQUIRED	Unique identifier for an MS run/data file	Free text
`technology type`	REQUIRED	Technology used to capture the data	Fixed values
`comment[proteomics data acquisition method]`	REQUIRED	DDA, DIA, PRM, SRM	PRIDE:0000659
`comment[label]`	REQUIRED	Label applied to sample (or "label free sample")	PRIDE - Labels
`comment[instrument]`	REQUIRED	Mass spectrometer model	PSI-MS - Instruments
`comment[cleavage agent details]`	REQUIRED	Enzyme information (use "not applicable" for top-down/undigested samples)	PSI-MS - Cleavage agents
`comment[fraction identifier]`	REQUIRED	Fraction number (1 if not fractionated)	Integer
`comment[technical replicate]`	REQUIRED	Technical replicate number (1 if none)	Integer
`comment[data file]`	REQUIRED	Name of the raw file	Free text

Example:

source name	assay name	technology type	comment[proteomics data acquisition method]	comment[label]	comment[instrument]	comment[data file]
sample_1	sample1_run1	proteomic profiling by mass spectrometry	data-dependent acquisition	label free sample	Q Exactive HF	sample1.raw

9.2. Sample Preparation and Fragmentation (MS-based only)

NOTE This section applies to **mass spectrometry-based proteomics** experiments only. For affinity-based proteomics, these properties do not apply.

For detailed documentation of sample preparation and MS/MS fragmentation properties, see the MS-Proteomics Template:

- **Sample preparation**: depletion, reduction reagent, alkylation reagent
- **Fractionation**: fractionation method (used with `comment[fraction identifier]`)
- **Fragmentation**: collision energy, dissociation method

9.3. Proteomics data acquisition method

Proteomics data acquisition method can happen in multiple ways: Data Dependent Acquisition (DDA), Data Independent Acquisition (DIA), and targeted approaches. The SDRF-Proteomics file format REQUIRES capturing the method used for the data acquisition in the *comment[proteomics data acquisition method]* column. The values MUST be children of the PRIDE ontology term proteomics data acquisition method (PRIDE:0000659). The following values are commonly used:

- data-dependent acquisition
- data-independent acquisition
  - diaPASEF
  - SWATH MS
- parallel reaction monitoring
- selected reaction monitoring

IMPORTANT The *comment[proteomics data acquisition method]* column is REQUIRED for all mass spectrometry-based SDRF files. This field must be explicitly specified and cannot be omitted or assumed.

You can find an example of a DIA experiment in the following link: DIA example

TIP For DIA experiments, additional properties like MS1 scan range can be captured. See DIA Scan Window Limits in the MS-Proteomics Template.

9.4. MS-Proteomics Template

For detailed guidance on data file metadata, refer to the conventions document:

- MS-Proteomics Template - Detailed guidelines for labels, instruments, modifications, cleavage agents, mass tolerances, RAW file URIs, and other data file properties

Chapter 10. Additional SDRF Rules

10.1. Column Cardinality

Some columns can appear multiple times for the same sample. The cardinality rules are:

- **Single (1)**: Column appears exactly once per sample (e.g., `characteristics[biological replicate]`)
- **Multiple (*)**: Column can appear multiple times (e.g., `comment[modification parameters]` can specify multiple post-translational modifications)

Example of multiple `comment[modification parameters]` columns:

source name	characteristics[…]	comment[modification parameters]	comment[modification parameters]	…
sample-1	…	NT=Carbamidomethyl;AC=UNIMOD:4;TA=C;MT=fixed;PP=Anywhere	NT=Oxidation;AC=UNIMOD:35;TA=M;MT=variable;PP=Anywhere	…

10.2. Row Uniqueness Requirements

SDRF files must satisfy specific uniqueness constraints to ensure data integrity and enable proper indexing by analysis tools.

Error-level constraint (validation fails): The combination of `source name` + `assay name` + `comment[label]` MUST be unique across all rows in the SDRF file. If two rows have identical values for all three columns, validation will fail with an error. This constraint ensures that each sample-run-label combination can be uniquely identified.

Warning-level constraint (validation warns): The combination of `source name` + `assay name` SHOULD be unique across all rows. Non-unique combinations will generate a warning during validation. This constraint helps identify potential issues where the same sample appears to have multiple entries for the same MS run without distinguishing labels.

Assay name uniqueness: Each distinct MS run/data file MUST have exactly one globally unique `assay name`, and no two different data files may share an assay name. To ensure uniqueness, it is RECOMMENDED to incorporate sample-specific information in assay names, such as sample IDs or replicate numbers (e.g., "sample1_run1", "sample1_run2", "patient001_fraction01").

> **NOTE** For multiplexed experiments (e.g., TMT, iTRAQ), multiple SDRF rows will share the same `assay name` because multiple samples are analyzed in a single MS run. In these cases, the `comment[label]` column distinguishes between different samples within the same run, and the combination of `source name` + `assay name` + `comment[label]` remains unique.

Example of valid multiplexed experiment:

source name	…	assay name	comment[label]	…	comment[data file]
sample_A	…	TMT_batch1_run1	TMT126	…	batch1_run1.raw
sample_B	…	TMT_batch1_run1	TMT127N	…	batch1_run1.raw
sample_C	…	TMT_batch1_run1	TMT127C	…	batch1_run1.raw
sample_D	…	TMT_batch1_run1	TMT128N	…	batch1_run1.raw

In this example, all four rows share the same `assay name` and `comment[data file]` because they represent different samples multiplexed in a single MS run. The combination of `source name` + `assay name` + `comment[label]` is unique for each row.

Chapter 11. Templates

11.1. What is a Template?

A **template** in SDRF-Proteomics is a predefined set of metadata columns (both required and recommended) that ensures consistent and complete annotation for a specific type of experiment or sample. Templates serve the same purpose as **metadata checklists**, **minimum information standards** (like MIAPE), or **validation schemas** in other data standards—they define what information must be captured to make a dataset FAIR (Findable, Accessible, Interoperable, Reusable).

Each template includes both **sample metadata** (characteristics columns describing the biological sample) and **data file metadata** (comment columns describing the MS data files)—everything needed to create a complete, validated SDRF file.

11.2. Template Layered Architecture

Templates in SDRF-Proteomics follow a **layered architecture** that separates concerns:

Layer	Purpose	Templates	Valid Alone?
BASE	Construction artifact with shared columns	base	No - never use directly
TECHNOLOGY	Minimum valid templates for data acquisition	ms-proteomics, affinity-proteomics	**Yes** - these are the minimum valid SDRFs
SAMPLE	Organism-specific sample metadata	human, vertebrates, invertebrates, plants	No - must combine with TECHNOLOGY
EXPERIMENT	Experiment-specific columns	cell-lines, single-cell, crosslinking, dda-acquisition, dia-acquisition	No - must combine with TECHNOLOGY

> **TIP** The **base** template is an internal construction artifact that defines shared columns inherited by all other templates. It should never be used directly by users. This specification document itself serves as the documentation for base template columns - all columns defined in the sample metadata and data file metadata sections are inherited by technology and sample templates.

Key concepts:

- **Technology templates** (`ms-proteomics` or `affinity-proteomics`) are **required** - you must use exactly one
- **Sample templates** (`human`, `vertebrates`, etc.) are **recommended** but optional
- **Experiment templates** provide additional methodology-specific columns
- Templates can be **extended**: add custom columns beyond the template requirements for study-specific metadata

When combining templates, include all required columns from each applicable template. The validator will check compliance with all specified templates.

11.3. Template Inheritance

Templates form a layered inheritance hierarchy. Child templates inherit all columns and validators from parent templates and can add new columns or strengthen requirements.

```
base (CONSTRUCTION ARTIFACT - not valid alone)
  ¬   Shared columns: source name, organism, organism part,
  ¬                   biological replicate, assay name, technology type,
  ¬                   instrument, technical replicate, data file
  ¬
  ¬¬¬ ms-proteomics (MINIMUM VALID for MS proteomics)
  ¬   ¬   Adds: acquisition method, cleavage agent, label, fraction identifier
  ¬   ¬
  ¬   ¬¬¬ dda-acquisition (extends ms-proteomics with DDA-specific columns)
  ¬   ¬¬¬ dia-acquisition (extends ms-proteomics with DIA-specific columns)
  ¬   ¬¬¬ crosslinking (extends ms-proteomics with XL-MS columns)
  ¬   ¬¬¬ single-cell (extends ms-proteomics with SCP columns)
  ¬   ¬¬¬ immunopeptidomics (extends ms-proteomics with MHC-MS columns)
  ¬
  ¬¬¬ affinity-proteomics (MINIMUM VALID for affinity proteomics)
      ¬   Adds: instrument, panel name, quantification unit, sample type
      ¬
      ¬¬¬ olink (extends affinity-proteomics with Olink-specific columns)
      ¬¬¬ somascan (extends affinity-proteomics with SomaScan-specific columns)

Sample templates (combine with TECHNOLOGY templates):
  ¬¬¬ human (adds: disease, age, sex, ancestry)
  ¬¬¬ vertebrates (adds: disease, developmental stage, strain)
  ¬¬¬ invertebrates (adds: disease, developmental stage, genotype)
  ¬¬¬ plants (adds: disease, developmental stage, growth conditions)

Experiment templates (organism-agnostic):
  ¬¬¬ cell-lines (adds: cell line name, Cellosaurus ID, passage number)
```

Inheritance rules:

1. Child templates inherit all columns from parent templates
2. Child templates may add new columns
3. Child templates may promote `optional` ¬ `recommended` ¬ `required`
4. Child templates **must not** weaken requirements (e.g., `required` ¬ `optional`)
5. Multiple templates can be combined (e.g., `ms-proteomics` + `human` + `crosslinking`)

Column requirement levels in templates:

Each column in a template has one of three requirement levels:

- **REQUIRED**: The column **MUST** appear in the SDRF file. Validation will fail if this column is missing. Template header files (.sdrf.tsv) always include these columns.

- **RECOMMENDED**: The column **SHOULD** appear in the SDRF file when applicable to the experiment. These columns are included in template header files and validation will generate warnings if missing.
- **OPTIONAL**: The column **MAY** be included if relevant to the study. These columns are NOT included in template header files but are documented in the template checklist. Users can add them as needed.

> **NOTE** When a column is marked as RECOMMENDED or OPTIONAL, it means users should evaluate whether the column applies to their specific experiment. For example, `characteristics[cell line]` is RECOMMENDED in the immunopeptidomics template because cell lines are commonly used, but tissue samples may use `not applicable`.

> **TIP** **For developers and maintainers:** For detailed information on YAML template structure, validators, and LinkML schema, see the Template Definitions Guidelines.

11.4. Specifying Templates in File Headers

When declaring templates in the `#template` header field, follow these rules:

Rule 1: Only list leaf templates - parent templates are implied

Since child templates inherit from parents, you only need to list the most specific (leaf) template. The parent is automatically implied:

```
# CORRECT: crosslinking inherits from ms-proteomics
#template=crosslinking

# INCORRECT: don't list parent when child is present
#template=ms-proteomics,crosslinking
```

Rule 2: List templates from different branches

When combining templates from different branches of the hierarchy (e.g., organism + experiment), list both:

```
# Human sample with crosslinking experiment
# - human is from the SAMPLE branch
# - crosslinking is from the EXPERIMENT branch (inherits ms-proteomics)
#template=human,crosslinking
```

Rule 3: Version simplification

Use a single version if all templates share the same version. Only use comma-separated versions when templates have different versions:

```
# All templates at v1.1.0 - use single value
#template=human,crosslinking
#template_version=v1.1.0
```

```
# Different versions - list each (matching template order)
#template=human,crosslinking
#template_version=v1.1.0,v1.0.0
```

Common examples:

Experiment Type	Templates	Header
Human MS proteomics	human (inherits ms-proteomics)	`#template=human`
Mouse MS proteomics	vertebrates (inherits ms-proteomics)	`#template=vertebrates`
Non-organism-specific MS	ms-proteomics only	`#template=ms-proteomics`
Human crosslinking	human + crosslinking	`#template=human,crosslinking`
Recombinant protein crosslinking	crosslinking only (no organism template)	`#template=crosslinking`
Human Olink	human + olink	`#template=human,olink`
Human DIA with cell lines	human + dia-acquisition + cell-lines	`#template=human,dia-acquisition,cell-lines`

11.5. Choosing Templates

First, determine your proteomics technology and select the appropriate technology template:

Template	Use For	Documentation	Valid Alone?
MS-Proteomics	All mass spectrometry-based proteomics (DDA, DIA, SRM/MRM/PRM)	Full documentation	**Yes** - minimum for MS
Affinity-Proteomics	Affinity-based proteomics (Olink, SomaScan)	Full documentation	**Yes** - minimum for affinity

11.6. Core Templates

Add the appropriate sample template based on your sample organism:

Template	Use For	Key Additional Columns	Documentation
Human	Human clinical samples	disease, age, sex, ancestry category, individual	Full documentation
Vertebrates	Mouse, rat, zebrafish, other vertebrates	disease, developmental stage, strain/breed, sex	Full documentation

Template	Use For	Key Additional Columns	Documentation
Invertebrates	Drosophila, C. elegans, insects	disease, developmental stage, strain/breed, genotype	Full documentation
Plants	Arabidopsis, crops, other plants	disease, developmental stage, strain/breed, growth conditions, treatment	Full documentation

For detailed explanations of each column, see sample metadata for sample properties and data file metadata for data file properties.

NOTE

The characteristics[cell type] column is RECOMMENDED across all templates when the cell type is known or can be determined. Use `not available` if the cell type cannot be determined (e.g., whole tissue samples, mixed cell populations). For cell line experiments, use the cell-lines template which provides more specific guidance.

11.7. Experiment-Type Templates

In addition to sample templates, SDRF-Proteomics provides specialized experiment-type templates. These templates extend the core templates with methodology-specific columns.

- **DDA Acquisition**: Data-dependent acquisition experiments. Includes columns for dissociation method, collision energy, fractionation method, modification parameters, and mass tolerances.
- **DIA Acquisition**: Data-independent acquisition experiments. Includes columns for scan window limits, isolation window width, DIA method, and spectral library information.
- **Cell Lines**: Experiments using cell line samples. Includes Cellosaurus integration for cell line identification.
- **Single-Cell Proteomics**: Single-cell proteomics experiments. Includes columns for cell isolation method, carrier proteome, and single-cell identifiers.
- **Immunopeptidomics**: MHC peptide immunopeptidomics. Includes columns for MHC class, HLA typing, and enrichment methods.
- **Crosslinking MS**: Cross-linking mass spectrometry experiments. Includes columns for crosslinking reagents and enrichment methods.
- **Metaproteomics**: Environmental and microbiome proteomics. Includes columns for environmental sample type and geographic location.
- **Affinity Proteomics**: Affinity-based proteomics (Olink, SomaScan). Includes columns specific to these platforms.

The template files can be downloaded from the templates folder.

11.8. Extending a Template

Templates define the minimum required and recommended columns for a given experiment type. However, SDRF-Proteomics is designed to be **extensible**:

1. **Add experiment-type templates**: If your experiment type has a specialized template (DDA, DIA, cell-lines, etc.), add it to your base configuration. See Experiment-Type Templates above.
2. **Add custom columns**: You can add any additional columns to capture study-specific metadata beyond what templates define. This section provides guidance on adding custom columns.

11.8.1. When to Add Custom Columns

Add custom columns when your experiment requires metadata that is:

- Not covered by existing templates but important for data interpretation
- Specific to your experimental design or technology
- Required for cross-study integration or data reuse
- Needed to comply with domain-specific standards (e.g., clinical trials, environmental studies)

11.8.2. Rules for Custom Columns

1. **Use appropriate column prefixes:**
   - `characteristics[…]` - For sample-related metadata (properties of the biological material)
   - `comment[…]` - For technical or protocol-related metadata (data acquisition, processing)
   - `factor value[…]` - For experimental variables (see Section 11.9)
2. **Follow naming conventions:**
   - Use lowercase for column names inside brackets
   - Use descriptive, specific names (e.g., `characteristics[tumor grade]` not `characteristics[grade]`)
   - Use ontology terms when they exist (e.g., `characteristics[patient bmi]` maps to EFO term)
3. **Use controlled vocabularies when available:**
   - Reference existing ontologies (EFO, MONDO, UBERON, etc.) for values
   - Prefer standardized terms over free text when possible
   - For new terms, follow the key=value format if ontology reference is needed

11.8.3. Common Additional Columns

Beyond template requirements, many commonly used columns can enhance your SDRF file. For a comprehensive list of additional columns for sample metadata (treatment, time point, dose, BMI, smoking status, etc.) and data metadata (sample preparation, enrichment method, etc.), see:

- Common Additional Columns in the Sample Metadata Guidelines

- SDRF Terms Reference for a comprehensive list of commonly used terms

11.8.4. Example: Adding Study-Specific Columns

For a drug treatment time-course study, you might add columns beyond the template requirements:

source name	characteristics[organism]	characteristics[disease]	characteristics[treatment]	characteristics[time point]	characteristics[dose]	…	factor value[treatment]	factor value[time point]
sample_ctrl_0h	homo sapiens	normal	vehicle control	0h	not applicable	…	vehicle control	0h
sample_drug_24h	homo sapiens	normal	dexamethasone	24h	100 nM	…	dexamethasone	24h

TIP When adding custom columns, check the SDRF Terms Reference and EBI Ontology Lookup Service (OLS) to find existing terms that match your metadata needs. Using standardized terms improves data interoperability.

11.9. Contributing New Templates

To contribute to template development or propose a new template:

1. Open an issue on GitHub describing the experiment type
2. Discuss requirements with the community
3. Submit a pull request following the established template structure

Chapter 12. Factor Values (Study Variables)

Factor values identify the experimental variables being studied - the conditions you want to compare in your analysis. They highlight which sample characteristics are the focus of your experiment.

12.1. Column Format

`factor value[{variable name}]`

12.2. When to Use Factor Values

Use factor values to indicate:

- The primary variable(s) under investigation
- Conditions being compared (e.g., disease vs. normal, treated vs. untreated)
- Variables that define experimental groups

> **NOTE** Use `normal` (not "control") in the disease field for healthy samples. "Control" is an experimental design concept, not a disease state. See Disease Annotation Guidelines for details.

12.3. Rules

- Factor value columns SHOULD appear after all characteristics and comment columns
- Multiple factor values can be used when studying multiple variables
- The value in a factor value column typically mirrors a characteristics column value

12.4. Example

In an experiment comparing tumor vs. normal tissue across different cancer stages:

source name	...	characteristics[disease]	characteristics[disease staging]	...	factor value[disease]	factor value[disease staging]
tumor_sample_1	…	breast carcinoma	stage II	…	breast carcinoma	stage II
normal_sample_1	…	normal	not applicable	…	normal	not applicable
tumor_sample_2	…	breast carcinoma	stage III	…	breast carcinoma	stage III

In this example, both `disease` and `disease staging` are factor values because the experiment aims to compare expression differences between disease states and across cancer stages.

Chapter 13. Ontologies and Controlled Vocabularies

SDRF-Proteomics uses ontologies and controlled vocabularies (CVs) to standardize metadata values. The following ontologies are supported:

Category	Ontology/CV	Description	Notes
General Purpose			
General	Experimental Factor Ontology (EFO)	General experimental metadata	
General	PATO	Phenotype and Trait Ontology	
General	NCI Thesaurus (NCIT)	Biomedical terminology	
General	PRIDE Controlled Vocabulary	Proteomics-specific terms	
Organism and Taxonomy			
Taxonomy	NCBI Taxonomy (NCBITaxon)	Organism classification	
Anatomy and Cell Types			
Anatomy	UBERON	Cross-species anatomy ontology	
Cell Type	Cell Ontology (CL)	Cell type classification	
Anatomy	BRENDA Tissue Ontology (BTO)	Tissues and cell lines	
Anatomy	Plant Ontology (PO)	Plant anatomy and development	For plant samples
Anatomy	FlyBase Anatomy (FBbt)	Drosophila anatomy	For Drosophila samples
Anatomy	WormBase Anatomy (WBbt)	C. elegans anatomy	For C. elegans samples
Anatomy	Zebrafish Anatomy (ZFA)	Zebrafish anatomy and development	For zebrafish samples
Disease (see Disease Annotation Guidelines)			
Disease	Mondo Disease Ontology (MONDO)	Unified disease ontology	RECOMMENDED
Disease	Experimental Factor Ontology (EFO)	Disease terms from EFO	
Healthy samples	Phenotype And Trait Ontology (PATO)	Use `normal` (PATO:0000461) for healthy samples	
Cell Lines			
Cell Lines	Cellosaurus	Cell line knowledge resource	RECOMMENDED

Category	Ontology/CV	Description	Notes
Cell Lines	Cell Line Ontology (CLO)	Cell line ontology	
Mass Spectrometry and Proteomics			
MS/Proteomics	PSI Mass Spectrometry CV (PSI-MS)	Instruments, methods, parameters	
Modifications	Unimod	Protein modifications database	
Modifications	PSI-MOD CV	Protein modifications ontology	
Other			
Chemistry	ChEBI	Chemical Entities of Biological Interest	
Environment	Environment Ontology (ENVO)	Environmental sample classification	For metaproteomics
Ancestry	Human Ancestry Ontology (HANCESTRO)	Human ancestry categories	For human samples

Chapter 14. Examples of Annotated Datasets

The following table provides links to example SDRF files for different experiment types. Click "View in Explorer" to open the SDRF file in the interactive viewer.

Experiment Type	Dataset	Description	View	Source
Label-free	PXD008934	Human proteome label-free quantification	View in Explorer	GitHub
TMT	PXD017710	TMT-labeled quantitative proteomics	View in Explorer	GitHub
SILAC	PXD000612	SILAC-based quantification	View in Explorer	GitHub
DIA	PXD018830	Data-independent acquisition	View in Explorer	GitHub
Phosphoproteomics	PXD000759	PTM enrichment study	View in Explorer	GitHub
Cell lines	PXD001819	Cell line proteomics	View in Explorer	GitHub

TIP Use the SDRF Explorer to browse all {total_datasets}+ annotated datasets with filtering, statistics, and interactive viewing.

A comprehensive collection of annotated projects is available at: Annotated Projects Repository

Chapter 15. Intellectual Property Statement

The PSI takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the PSI Chair.

The PSI invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the PSI Chair (see contacts information at PSI website).

Chapter 16. Copyright Notice

Chapter 17. How to cite

Please cite this document as:

Dai C, Füllgrabe A, Pfeuffer J, Solovyeva EM, Deng J, Moreno P, Kamatchinathan S, Kundu DJ, George N, Fexova S, Grüning B, Föll MC, Griss J, Vaudel M, Audain E, Locard-Paulet M, Turewicz M, Eisenacher M, Uszkoreit J, Van Den Bossche T, Schwämmle V, Webel H, Schulze S, Bouyssié D, Jayaram S, Duggineni VK, Samaras P, Wilhelm M, Choi M, Wang M, Kohlbacher O, Brazma A, Papatheodorou I, Bandeira N, Deutsch EW, Vizcaíno JA, Bai M, Sachsenberg T, Levitsky LI, Perez-Riverol Y. A proteomics sample metadata representation for multiomics integration and big data analysis. Nat Commun. 2021 Oct 6;12(1):5854. doi: 10.1038/s41467-021-26111-3. PMID: 34615866; PMCID: PMC8494749. [Manuscript - https://www.nature.com/articles/s41467-021-26111-3]

References

- [1] Y. Perez-Riverol, S. European Bioinformatics Community for Mass, Toward a Sample Metadata Standard in Public Proteomics Repositories, J Proteome Res 19(10) (2020) 3906-3909. doi:10.1021/acs.jproteome.0c00376
- [2] A. Gonzalez-Beltran, E. Maguire, S.A. Sansone, P. Rocca-Serra, linkedISA: semantic representation of ISA-Tab experimental metadata, BMC Bioinformatics 15 Suppl 14 (2014) S4. doi:10.1186/1471-2105-15-S14-S4
- [3] T.F. Rayner, P. Rocca-Serra, P.T. Spellman, et al., A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB, BMC Bioinformatics 7 (2006) 489. doi:10.1186/1471-2105-7-489
- [4] P. Blainey, M. Krzywinski, N. Altman, Points of significance: replication, Nat Methods 11(9) (2014) 879-80. doi:10.1038/nmeth.3091
- [5] D. Gupta, I. Liyanage, Y. Perez-Riverol, et al., BioSamples database: the global hub for sample metadata and multi-omics integration, Nucleic Acids Res (2025). doi:10.1093/nar/gkaf1133