Bioinformatics metadata

Jeffrey Szymanski

January 14, 2026

- Elements
  - Immuntable reference files
    - Possible PHI
    - Provenence
  - Mutable "sample sheet"
    - xlsx or series of tsvs
    - No PHI
    - Source of truth
    - Validation input
- No PHI
- With Libreoffice calc
  - Use Excel A1 formula syntax (Tools -> Options -> LibreOffice Calc -> Formula -> Formula Options -> Formula syntax)

Logical data representation, Common Data Model schema

Documentation

Good Best Practices for Bioinformatics Metadata Schema with Frictionless

- Components
  - Excel as the user interface: non-programmers enter data, schema validates it. Hierarchical data model: organize by biological and technical relationships
  - schema yaml
    Self-documenting: schemas generate data dictionaries programmatically
- Schema Structure
  - Version control by repo-level versioning
  - Define Shared Elements Once, Use YAML anchors for repeated definitions:
  - Basic Resource Structure
  - Field Definition Principles
    - Every field should include:
      - name: Machine-readable identifier
      - title: Human-readable label
      - description: Clear explanation including format requirements and units
      - example: Representative value
      - type: Data type
      - constraints: Validation rules

        - Example:

- Primary Key Conventions
    - Use descriptive prefix plus zero-padded four-digit numbers:
        - Enables lexicographic sorting
        - Format: ^[prefix]4$
        - Examples: subj0001, samp0001, lib0001, run0001
        - Always include pattern constraint to catch typos.
- Foreign Key Conventions
    - Foreign key fields should be minimal:
    - Include pattern constraint to validate format before checking referential integrity.
    - Detailed metadata lives in the parent table only - avoid redundancy.
- Data Type Decision Principles
    - Prefer Enums Over Booleans. Use enums for domain-specific concepts that require explanation:
    - Booleans are acceptable only when semantics are universally obvious and require no explanation.
    - Document Enum Values
        - Use custom enumDescriptions property for programmatic data dictionary generation:
    - Standardize Units in Field Names: Include units in field name and description:

- Table Design Principles
  - Hierarchical Relationships
    - Organize tables by biological and technical hierarchy:

      `subjects → samples → libraries → fastqs`
    - Use foreign keys to enforce referential integrity.
  - When to Split Tables
    - Split into separate tables when entities have:
      - Many divergent fields (>5 unique per subtype)
      - Fundamentally different semantics
      - Example: human participants vs mouse subjects
  - Keep entities in one table when:
    - Few optional fields (2-3)
    - Structural similarity between subtypes
    - Add discriminator field for subtype and leave subtype-specific fields optional:
- Schema Evolution Principles
  - Never use "other" catch-all categories or allow arbitrary values - this defeats validation.
  - Constraint Defaults
    - Frictionless defaults are:
      - required: false (field is optional)

        - unique: false (duplicates allowed)
        - No enum restrictions
    - Only declare constraints when restricting behavior.
- Field Ordering
    - Primary key first
    - Foreign keys second
    - Required fields
    - Optional fields
    - Group related fields together

Preamble

Subjects

Participants

Mice

Samples

Libraries

Sequencing runs

FASTQs

#+end_src