FAIRy: Reproducible Datasets by Default

Jennifer Slotnick
FAIRy / Datadabra
🔗linkedin.com/in/jenniferslotnick/

Local-first preflight checks and shareable report artifacts

Dataset: checks + report + rerun.

Dataset handoff issues

Structure / Format

- Missing metadata/ wrong formats
- Inconsistent sample IDs across files
- Ambiguous dates / provenance

Workflow impact

- Blocked by schema/ validation
- Dumped on curator to fix by hand

Reproducibility issues come before analysis

Analysis you can rerun and share

- You revisit a notebook 3 weeks later and can’t recreate the same results
- A teammate can’t merge/join because keys changed or IDs aren’t stable
- The same dataset updated and your outputs silently changed

Using TidyTuesday for Katas

- Data Science Learning Community / TidyTuesday
- Great demo sandbox (no sensitive data)
- Shows that “clean” data still has assumptions
  - N/A markers
  - Inconsistent categories
  - Ambiguous dates

The dataset preflight concept

```
Dataset → Preflight → PASS/WARN/FAIL report → Fix
→ Re-run (see what cleared)
```

So analysis doesn’t depend on hidden cleanup steps

What FAIRy Already Does

Fix → re-run —> tells you what cleared

```
=== FAIRy Preflight ===
Rulepack:         GEO-SEQ-BULK@0.1.0
FAIRy version:    0.1.0
Run at (UTC):     2025-10-29T05:57:07.345059+00:00
FAIL findings:    1
WARN findings:    0
submission_ready: False
Report JSON:      out/report.json

Example finding:
  [FAIL] GEO.BIO.CONTEXT_MISSING @ row 1
    why: Each sample needs biological source info (tissue / cell_line / cell_type).
    fix: Provide at least one of tissue, cell_line, or cell_type for each row.

Resolved since last run:
  ✔CORE.DATE.INVALID_ISO8601
  ✔CORE.ID.UNMATCHED_SAMPLE
```

- Runs locally (offline)
- PASS/FAIL/WARN with why and how to fix
- Write JSON and Markdown report with timestamp, rulepack version you can share
- On re-run highlights what cleared (fast feedback loop)

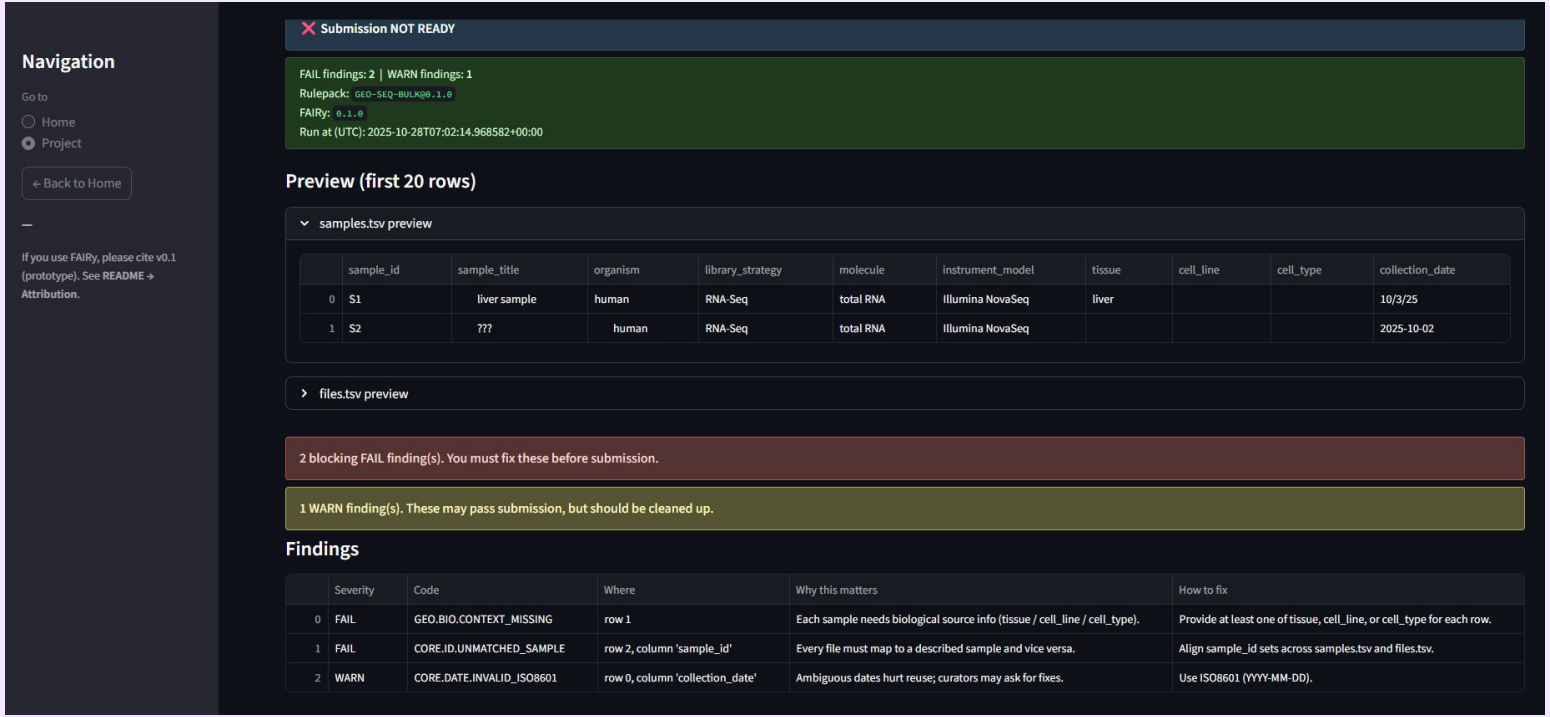
Like unit tests for datasets and a shareable QA report

Rulepacks

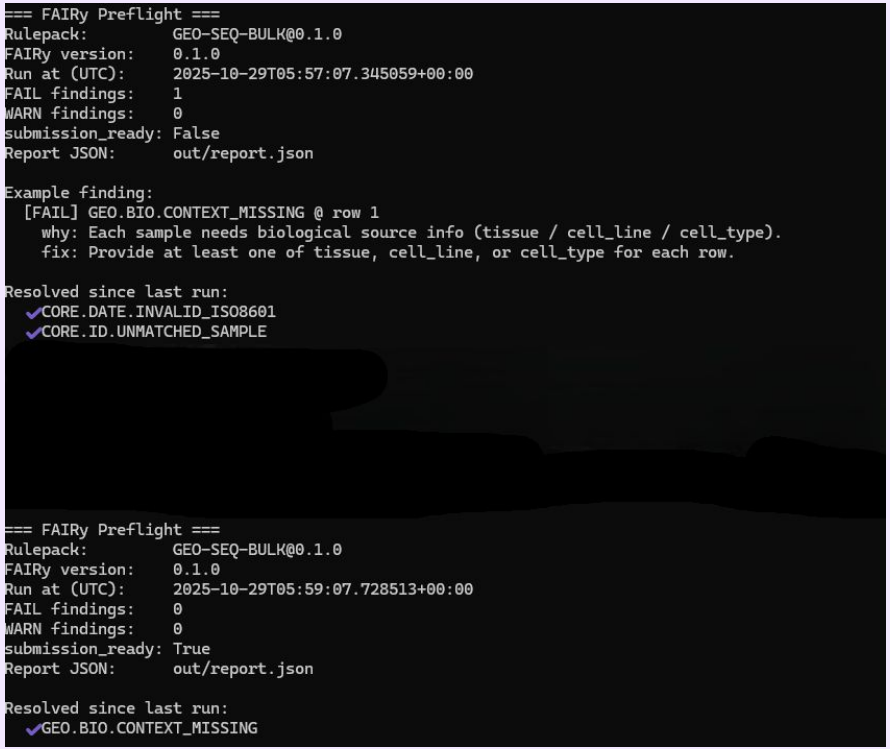

```
"rulepack_id": "GEO-SEQ-BULK",
"rulepack_version": "0.1.0",
"repository": "GEO",
"assay_type": ["RNA-seq", "ATAC-seq", "ChIP-seq", "methyl-seq"],
"mode": "bulk",
"description": "Bulk (pooled) sequencing submissions to GEO. One row per biological sample; excludes single-cell and spatial.",
"rules": [
  {
    "code": "GEO.REQ.MISSING_FIELD",
    "severity": "FAIL",
    "where": "samples.tsv (library_strategy, molecule, instrument_model, sample_title, organism)",
    "why": "GEO requires these fields for each sample.",
    "how_to_fix": "Add the missing column(s); fill every row with allowed values.",
    "check": {
      "type": "require_columns",
      "target_file": "samples.tsv",
      "required_columns": [
        "sample_title",
        "organism",
        "library_strategy",
        "molecule",
        "instrument_model"
      ]
    }
  },
```


- Rulepack = versioned bundle of expectations
- Portable across datasets in a domain
- Makes checks shareable and repeatable

FAIRy: First Run



FAIRy: After fixes



What makes datasets reproducible?

- Reproducibility starts before data analysis
- “Default” means reproducing the dataset doesn’t require tribal knowledge
- A dataset should come with reproducible artifacts like:
  - README or data dictionary (what columns mean, units, categories)
  - Schema (what columns exist, types, required fields)
  - Provenance (source, version/date, how it was produced)
  - Checksums (to make sure you are using the same file)
  - Processing script or pipeline (how it was cleaned or transformed)

Looking for collaborators

QR for FAIRy-Core repository on Github

Scan QR and skim the Penguins quickstart

If you have Python, run the Penguins validate command and look at the JSON report

Long-term hope: compare pilots and find the pieces that travel across domains.