Pharmacophore

ISSN-2229-5402

Journal home page: http://www.pharmacophorejournal.com

A COMMON PROVENANCE LANGUAGE FOR TRACEABLE PHARMACEUTICAL DATA, REPRODUCIBLE BIOINFORMATICS, AND AUDITABLE COMPUTATIONAL DISCOVERY

Ahmed Al-Jubouri[1]*, Emma Carter[2], Thomas Hoffmann[3], Noor Al-Hassan[1]

1. ***Department of Data Provenance and Computational Reproducibility, College of Pharmacy, University of Baghdad, Baghdad, Iraq.***
2. ***Department of Bioinformatics Traceability and Auditable Workflows, Faculty of Pharmacy, University of London, London, United Kingdom.***
3. ***Department of Computational Discovery and Data Governance, Faculty of Pharmaceutical Sciences, Heidelberg University, Heidelberg, Germany.***

ARTICLE INFO

Received:
20 May 2024
Received in revised form:
05 October 2024
Accepted:
07 October 2024
Available online:
28 October 2024

Keywords: Pharmaceutical provenance, Computational reproducibility, Bioinformatics workflows, Ontology engineering, Evidence lineage, FAIR data

ABSTRACT

Pharmaceutical computation increasingly depends on heterogeneous molecular, biological, experimental, clinical, and computational resources that are repeatedly transformed across data acquisition, preprocessing, modeling, interpretation, and decision-support activities. Although workflow systems, metadata standards, FAIR data practices, machine-learning reporting recommendations, and biomedical knowledge graphs address important parts of this landscape, they do not provide a shared language for describing how pharmaceutical evidence is produced, modified, interpreted, and reused. This article proposes a Common Pharmaceutical Provenance Language as an original ontological contribution for representing the entities, activities, agents, relationships, versions, decisions, and evidential boundaries that constitute computational pharmaceutical discovery. The proposed language distinguishes source observations from derived data, transformations from their outputs, computational predictions from interpretations, and evidence lineage from claims of mechanism, utility, or therapeutic validity. Its architecture organizes provenance across four linked analytical zones: acquisition, preprocessing, modeling, and interpretation. It further specifies minimum metadata requirements, typed relationships, version-aware transformation records, human-decision objects, and interfaces with FAIR data systems, executable workflows, and knowledge graphs. The contribution is conceptual rather than empirically validated. It does not establish that provenance completeness guarantees reproducibility, that transparent lineage ensures data suitability, or that auditable computation demonstrates biological mechanism, clinical value, or regulatory acceptability. Instead, it provides a structured vocabulary through which these distinctions can be recorded, evaluated, and communicated. Future research must test semantic coverage, cross-platform interoperability, annotation burden, governance processes, domain extensibility, and the relationship between provenance quality and reproducible pharmaceutical inference. A shared provenance language may thereby support more traceable computational research while preserving the limits of what computational records can establish.

To Cite This Article: Al-Jubouri A, Carter E, Hoffmann T, Al-Hassan N. A Common Provenance Language for Traceable Pharmaceutical Data, Reproducible Bioinformatics, and Auditable Computational Discovery. Pharmacophore. 2024;15(5):69-79. https://doi.org/10.51847/L0e6kRVTHx

Introduction

Pharmaceutical discovery increasingly operates through computational chains rather than isolated analyses. Molecular structures may be assembled from public and proprietary repositories, omics measurements may be normalized and integrated, assay records may be filtered or reannotated, and machine-learning models may generate predictions that are subsequently interpreted through biological pathways, chemical knowledge, or expert judgment. Across life-science computation and biopharmaceutical research, reproducibility and reuse depend not only on accessible outputs but also on preserved metadata, executable context, and shared data practices [1–3]. Yet the record accompanying a pharmaceutical result often remains fragmented across database documentation, workflow files, scripts, notebooks, parameter logs, model cards, version-control

Corresponding Author: Ahmed Al-Jubouri; Department of Data Provenance and Computational Reproducibility, College of Pharmacy, University of Baghdad, Baghdad, Iraq. E-mail: ahmed.aljubouri@uobaghdad.edu.iq.

repositories, laboratory records, and informal analyst explanations. Consequently, a result may appear technically reproducible while its scientific lineage remains incomplete.

The problem is intensified by the heterogeneity of drug-discovery evidence. Pharmaceutical computation integrates chemical entities, targets, genes, proteins, pathways, phenotypes, diseases, assays, publications, safety observations, and therapeutic hypotheses. These objects differ in granularity, certainty, provenance, and intended use. Reviews of biomedical data resources from a knowledge-graph perspective show that drug-discovery datasets vary substantially in identifiers, data models, coverage, curation practices, and semantic relationships [4]. Integration can therefore obscure rather than resolve provenance when source distinctions, transformation histories, evidence types, or contextual restrictions are collapsed into a unified representation. A shared representation must preserve both connectivity and difference: it should enable objects to be linked without implying that all links carry equivalent evidential weight.

Machine-learning applications add further complexity because a pharmaceutical prediction is shaped by more than the final algorithm. Dataset composition, label construction, preprocessing, feature generation, hyperparameter selection, validation design, software dependencies, and reporting decisions all influence what a model output means. Reproducibility standards for life-science machine learning consequently emphasize documentation across data, model, software, evaluation, and reporting domains [5]. However, compliance with reporting recommendations does not automatically produce an interoperable provenance record. Information may be available to a reader yet remain difficult for machines to interpret, compare, or propagate across workflows. Conversely, a machine-readable record may be technically complete while failing to express why a transformation was chosen, which alternative was rejected, or how uncertainty limits downstream interpretation.

This article develops a proposed Common Pharmaceutical Provenance Language, hereafter CPPL, as an original pharmaceutical ontology contribution. CPPL is conceived as a shared semantic layer for representing computational pharmaceutical objects, transformations, agents, versions, decisions, evidence relations, and interpretive boundaries across the discovery lifecycle. Its central claim is not that provenance alone makes computational research valid. Rather, explicit provenance can make the conditions, dependencies, and limitations of a result more inspectable. The proposed language therefore separates traceability from reproducibility, reproducibility from validity, prediction from explanation, association from mechanism, and computational evidence from experimental or clinical confirmation. The article defines the problem that CPPL addresses, specifies its core objects and relationships, and establishes the boundaries within which such a language may support auditable computational discovery.

Why Pharmaceutical Computation Lacks a Shared Provenance Vocabulary

The absence of a common pharmaceutical provenance vocabulary is not primarily caused by a lack of technical tools. Workflow systems already capture portions of computational execution, including inputs, outputs, parameters, software dependencies, and task ordering. Nevertheless, comparative work on scientific workflows, workflow managers, and interoperable workflow provenance shows that these descriptions remain distributed across platform-specific formats, execution models, and portability conventions [6–8]. A workflow engine can record that a process occurred, but it may not express the pharmaceutical identity of the objects involved, the evidential status of an input, the scientific rationale for a preprocessing decision, or the interpretive boundary attached to an output. The unresolved problem is therefore semantic coordination across systems rather than the simple absence of execution records.

Pharmaceutical computation also lacks a shared vocabulary because its analytical objects are described according to the priorities of different communities. Chemoinformatics emphasizes molecular representation and structure–activity relationships; bioinformatics emphasizes sequences, expression profiles, variants, pathways, and analytical pipelines; computational pharmacology emphasizes targets, mechanisms, exposures, effects, and therapeutic contexts; machine learning emphasizes datasets, features, labels, models, and evaluation procedures. Biomedical reproducibility recommendations support more rigorous reporting of research design and computational practice, but they do not define a unified pharmaceutical ontology spanning these domains [9]. Terms such as *dataset*, *feature*, *evidence*, *model*, *version*, *decision*, and *interpretation* may therefore refer to different objects or levels of abstraction in different analytical settings.

A second source of fragmentation is the tendency to treat performance reporting as a substitute for provenance. A model may report discrimination, ranking, calibration, or prediction-error measures without preserving how examples were selected, how labels were constructed, which records were excluded, or which model version produced the reported output. Structured recommendations for biological machine-learning validation organize documentation around data, optimization, model, and evaluation components [10]. Such recommendations are essential, but the CPPL proposal extends the problem beyond validation reporting. It asks how each documented element should be represented as an identifiable object, how relationships among objects should be typed, and how downstream interpretations should remain connected to the data and transformations from which they arose. A performance estimate is thus represented as one provenance-bearing evaluation object, not as a general certificate of pharmaceutical usefulness.

The proposed ontology responds by defining provenance as a structured account of what existed, what changed, who or what acted, under which conditions, for what stated purpose, and with which evidential consequence. This definition includes computational execution but is not limited to it. It encompasses source acquisition, identifier resolution, curation, exclusion, normalization, feature construction, model training, parameter selection, validation, interpretation, and claim formation. The resulting vocabulary must also encode negative and limiting information, including unavailable metadata, unresolved mappings, incompatible versions, unsupported transformations, and uncertain interpretations. It should not imply that every

provenance element is equally important in every context; instead, it should support context-specific minimum profiles while maintaining a stable semantic core. **Table 1** organizes the evidence, constructs, and interpretation boundaries needed to develop why pharmaceutical computation lacks a shared provenance vocabulary within the article's central argument.

Table 1. Evidence domains, core questions, scientific requirements, and interpretation boundaries for Why pharmaceutical computation lacks a shared provenance vocabulary

Construct or Evidence Domain	Core Question	Scientific Basis	Role in the Article	Failure or Overclaiming Risk	Boundary Statement
Source Identity	What data, knowledge resource, assay record, publication, or model output entered the computation?	Pharmaceutical analyses integrate heterogeneous objects with different identifiers, curation histories, and evidence status	Establishes the origin of each computational input	Treating a named source as proof that the extracted data are suitable or correct	Source identification supports traceability but does not establish data quality or fitness for purpose
Acquisition Context	When, how, and under which access or selection conditions was an object obtained?	Databases and repositories evolve, while queries, access dates, filters, and releases affect retrieved content	Connects a local analytical object to a specific source state	Assuming that another user will retrieve identical content from a changing resource	Acquisition provenance can document retrieval conditions but cannot guarantee future source stability
Transformation History	Which operations converted an input into a derived object?	Cleaning, normalization, mapping, aggregation, feature construction, and modeling alter meaning and comparability	Provides the backbone for tracing derived pharmaceutical evidence	Describing transformations only by informal labels or software names	A transformation record identifies an operation but does not demonstrate that the operation was scientifically appropriate
Execution Environment	Which software, dependencies, parameters, hardware conditions, and workflow versions were used?	Computational reproducibility depends on executable context as well as code and data	Supports technical reconstruction of analytical steps	Equating environment capture with full scientific reproducibility	A preserved environment may permit rerunning an analysis without validating its design or interpretation
Agent and Responsibility	Which person, organization, software agent, or automated process performed or authorized an action?	Human and automated contributions affect curation, modeling, interpretation, and approval	Makes responsibility and attribution explicit	Converting attribution into an assumption of authority or correctness	Recorded responsibility identifies participation but does not certify competence, independence, or validity
Decision Provenance	Why was one analytical action, threshold, mapping, model, or interpretation selected?	Pharmaceutical workflows contain discretionary choices that are rarely fully encoded in execution logs	Preserves rationale, alternatives, and decision context	Retrospectively presenting choices as inevitable or objectively optimal	A recorded rationale makes a decision inspectable but does not prove that it was the best decision
Evidence Lineage	Which observations, transformations, and interpretive steps support a downstream claim?	Drug-discovery conclusions may combine experimental records, curated assertions, predictions, and expert interpretation	Prevents loss of connection between claims and their supporting evidence	Collapsing all contributing links into a single undifferentiated confidence statement	Lineage supports auditability but does not convert association or prediction into mechanism or therapeutic validation
Semantic Interoperability	Can different systems interpret objects and relationships consistently?	Platform-specific records limit exchange, comparison, and machine actionability	Motivates shared classes and typed relationships	Assuming that common serialization alone creates shared meaning	Technical exchange is insufficient unless concepts and relation semantics are aligned
Minimum Metadata	Which information must always accompany a provenance object, and which information is context-dependent?	Excessive requirements increase annotation burden, whereas insufficient metadata weakens traceability	Supports tiered conformity profiles	Treating one universal checklist as appropriate for every pharmaceutical workflow	Minimum metadata should be purpose-specific while remaining compatible with a common semantic core
Interpretive Boundary	What may and may not be concluded from a recorded computational result?	Transparent computation does not alone establish mechanism, safety, clinical utility, or regulatory acceptability	Embeds evidential caution into the ontology	Using provenance completeness as a proxy for scientific validity	Provenance quality concerns the inspectability of evidence production, not the truth or usefulness of the resulting claim

Provenance Objects Across Data Acquisition, Preprocessing, Modeling, and Interpretation

Within CPPL, a provenance object is any identifiable entity, activity, agent, decision, or assertion whose state or relationship is necessary to understand how a computational pharmaceutical result was produced. The acquisition zone begins with a source object, such as a molecular database release, omics repository record, assay table, curated drug–target assertion, publication-derived statement, or locally generated dataset. An acquisition activity then records the query, access condition, filter, import

procedure, extraction rule, timestamp, and source version through which a local object was created. Nextflow demonstrates how processes, dependencies, inputs, outputs, and execution configurations can be made explicit within reproducible computational workflows [11]. CPPL generalizes this logic beyond a particular workflow language by distinguishing the external source, the act of retrieval, the locally instantiated object, and the pharmaceutical context in which that object is intended to be used.

The preprocessing zone represents operations that alter the structure, representation, composition, or interpretation of acquired objects before modeling. These include identifier reconciliation, duplicate removal, missing-data handling, normalization, batch correction, chemical standardization, feature generation, label construction, ontology mapping, aggregation, exclusion, and train–validation partitioning. Galaxy illustrates the value of retaining histories that connect datasets, tools, parameters, and workflow steps in collaborative biomedical analysis [12]. CPPL treats each transformation as an activity linked to input and output entities through typed relations such as *used*, *generated*, *derived from*, *mapped to*, *excluded from*, and *replaced by*. The output of preprocessing is therefore not merely a "processed dataset"; it is a versioned entity whose lineage identifies the transformations and decisions that shaped its analytical meaning.

The modeling zone extends provenance from datasets to model-bearing objects. It includes model specifications, architectures, training activities, parameter configurations, software environments, learned parameter states, checkpoints, evaluation datasets, metric definitions, and evaluation outputs. Community-curated pipeline frameworks demonstrate how standardized components, testing practices, software containers, version control, and reusable workflow structures can increase the consistency of bioinformatics analysis [13]. CPPL nevertheless distinguishes reusable pipeline quality from model validity. A model version should be connected to the exact data versions, feature representations, labels, optimization procedure, randomization conditions, software components, and evaluation design that produced it. A retrained model, recalibrated model, fine-tuned model, or model evaluated under a new endpoint must therefore be represented as a distinct version or derivation rather than silently replacing an earlier state.

The interpretation zone captures the transition from computational output to scientific meaning. It includes ranked candidates, predicted interactions, inferred pathways, explanatory attributions, uncertainty statements, analyst annotations, expert assessments, and claim objects. RO-Crate-based workflow-run provenance shows how data, software, actions, agents, and contextual metadata can be packaged as linked research objects [14]. CPPL builds on this object-centered logic by requiring that an interpretive claim remain connected to the specific output, model version, transformation chain, and decision context from which it emerged. An interpretation may state that a molecule is prioritized for further investigation, but it must not be represented as equivalent to verified binding, demonstrated mechanism, developability, safety, or therapeutic benefit. The provenance record should therefore preserve both the supporting lineage and the explicit boundary of the claim.

Core Ontology Classes, Relationships, and Minimum Metadata

The proposed CPPL organizes pharmaceutical provenance around seven upper-level classes: Entity, Activity, Agent, Context, Decision, Assertion, and Constraint. An Entity is a persistent or versioned object, including a source record, dataset, molecular representation, feature matrix, model, parameter state, workflow definition, evaluation result, or interpretation. An Activity is an acquisition, transformation, computation, evaluation, curation, or interpretive operation. An Agent is a person, organization, software component, automated service, or instrument responsible for an activity. Context records the scientific, computational, organizational, and intended-use conditions under which an object was produced or applied. Decision captures a choice among alternatives; Assertion represents a proposition supported, contradicted, or qualified by evidence; and Constraint records uncertainty, applicability, access, licensing, or interpretive boundaries. FAIR assessment frameworks demonstrate why digital objects require identifiable and evaluable properties rather than general declarations of good practice [15].

These classes are connected by typed, directional relationships. Core relations include *was generated by*, *used*, *was derived from*, *was attributed to*, *was informed by*, *is version of*, *preceded*, *selected*, *excluded*, *supports*, *contradicts*, *qualifies*, *interprets*, and *is bounded by*. Relationship typing is essential because two objects may be connected without carrying the same scientific meaning. A model output may be *derived from* a processed dataset, whereas a mechanistic statement may merely be *informed by* that output. Similarly, an analyst may *interpret* an attribution map without the map itself *demonstrating* a biological mechanism. Community-governed FAIR maturity frameworks support explicit, machine-actionable criteria and reveal why semantic conformity should be assessed through transparent tests rather than asserted through self-description [16]. CPPL therefore separates relation existence, relation type, evidential strength, and interpretive scope.

Minimum metadata should be layered rather than imposed through one exhaustive universal record. The proposed core profile requires a persistent or locally unique identifier, object class, human-readable label, creation or acquisition time, responsible agent, version, source or parent object, generating activity, applicable context, and access condition. An activity profile additionally records inputs, outputs, software or method identity, parameters, environment, execution status, and relevant decisions. A claim profile records the proposition, supporting and conflicting objects, responsible interpreter, uncertainty statement, and explicit boundary. Domain extensions may add molecular-standardization rules, assay conditions, omics-processing metadata, pharmacological endpoints, or model-specific descriptors. Metadata-template research shows how community requirements can be converted into structured, reusable, and machine-actionable forms while retaining domain specificity [17].

CPPL itself must also be treated as a versioned research object. Every class and relationship should have a stable identifier, definition, scope note, examples, exclusions, provenance, status, and change history. Deprecated terms should remain resolvable, while mappings to external vocabularies should state whether equivalence is exact, broader, narrower, or context-dependent. FAIR vocabulary guidance emphasizes accessibility, persistent identification, reuse, explicit semantics, maintenance, and community governance [18]. These principles inform the proposed ontology but do not validate its pharmaceutical coverage. **Table 2** organizes the evidence, constructs, and interpretation boundaries needed to develop core ontology classes, relationships, and minimum metadata within the article's central argument.

Table 2. Components, relationships, uncertainties, and validation needs within Core ontology classes, relationships, and minimum metadata

Component or Layer	Inputs	Core Function	Expected Output	Uncertainty or Limitation	Validation Requirement
Identity Layer	Source identifiers, local identifiers, database accessions, workflow object identifiers	Assigns stable identity to provenance-bearing objects and separates versions or local instantiations	Resolvable, distinguishable object records	External identifiers may be absent, unstable, duplicated, or semantically inconsistent	Identifier-resolution tests, collision checks, and version-retrieval tests
Entity Layer	Data records, datasets, molecular representations, models, software, outputs, interpretations	Represents persistent or versioned objects that participate in pharmaceutical computation	Typed provenance entities with source and state information	Entity boundaries may differ across platforms or scientific communities	Competency-question testing across representative pharmaceutical use cases
Activity Layer	Inputs, methods, parameters, execution context, decisions	Represents acquisition, preprocessing, modeling, evaluation, curation, and interpretation operations	Traceable activities linked to inputs and outputs	Activity granularity may be too coarse for audit or too detailed for practical annotation	Reconstruction studies assessing whether material steps can be identified and interpreted
Agent Layer	People, teams, organizations, instruments, software services, automated systems	Records participation, responsibility, delegation, and attribution	Agent–activity associations with defined roles	Attribution does not establish competence, independence, or correctness	Role-consistency checks and review of responsibility assignments
Relationship Layer	Typed links among entities, activities, agents, decisions, assertions, and constraints	Preserves derivation, use, generation, attribution, versioning, support, contradiction, and qualification	Machine-interpretable provenance graph	Similar relations may be interpreted differently across domains	Semantic mapping tests and expert review of relation definitions
Context Layer	Scientific purpose, disease area, assay setting, computational environment, intended use	Qualifies the conditions under which an object or activity should be interpreted	Context-bounded provenance record	Context cannot be exhaustively represented and may change over time	Cross-context transfer tests and assessment of omitted material conditions
Decision Layer	Alternatives, selected option, responsible agent, rationale, supporting evidence	Represents discretionary human or automated choices	Inspectable decision record connected to downstream effects	Rationales may be incomplete, retrospective, or affected by undocumented constraints	Prospective recording studies and reviewer assessment of decision sufficiency
Assertion And Evidence Layer	Computational outputs, curated evidence, publications, expert judgments, uncertainty statements	Links claims to supporting, conflicting, or qualifying evidence	Evidence-bearing assertions with explicit scope	Evidence relations do not themselves establish causality or truth	Independent evidence-lineage audits and claim-to-source traceability tests
Minimum Metadata Profiles	Core fields, activity-specific fields, claim-specific fields, domain extensions	Balances interoperability with context-specific information needs	Conformant records at a declared profile level	Minimal profiles may omit decisive details; extensive profiles may create excessive burden	Field-completeness, usability, annotation-burden, and information-loss assessments
Boundary and Constraint Layer	Uncertainty, applicability, access, licensing, validation status, known limitations	Prevents provenance records from being interpreted beyond their scientific or operational scope	Explicit machine- and human-readable boundary statements	Boundaries may be ignored by downstream systems or users	User studies and automated tests of boundary preservation during exchange and reuse

Representing Transformations, Model Versions, Human Decisions, and Evidence Lineage

A transformation in CPPL is represented as a first-class activity rather than as an implicit difference between two files. Its record identifies the input state, output state, method, parameterization, software or instrument, responsible agent, execution context, and declared purpose. Transformations may be deterministic, stochastic, manual, hybrid, reversible, or information-losing. These properties matter because rerunning the same nominal operation may not reproduce the same output when software, reference data, randomization, or execution conditions change. Dataset-versioning work demonstrates the need to identify evolving datasets through formal version relationships and recorded changes rather than silently overwriting prior states [19]. CPPL accordingly distinguishes correction, extension, reduction, reannotation, reformatting, and analytical derivation.

Model provenance requires more than a model name or stored parameter file. A model specification describes the architecture or algorithmic form; a model-building activity links the specification to training data, feature representations, labels, optimization settings, software, and execution conditions; and a model state represents the resulting learned parameters or fitted configuration. Evaluation activities should identify the exact model state, evaluation data version, endpoint, metric definition, aggregation procedure, and any post-processing. Provenance-driven scientific data pipelines illustrate how data, software, annotations, and computational steps can be linked so that outputs remain traceable to their generating conditions [20]. CPPL extends this principle by requiring recalibration, fine-tuning, retraining, endpoint substitution, and threshold changes to generate explicit derivation records.

Human decisions must remain visible because pharmaceutical computation contains choices that cannot be reconstructed from execution logs alone. Examples include selecting a database release, resolving conflicting identifiers, defining inclusion criteria, choosing a molecular representation, excluding an assay, selecting a model, interpreting an attribution, or deciding that evidence is sufficient for escalation. A decision object should identify the available alternatives, selected option, responsible agent, stated rationale, supporting information, time, and downstream activities affected. Ontology-grounded knowledge-graph ecosystems show how heterogeneous sources and semantic choices can be assembled into configurable computational resources [21]. CPPL adds a decision-provenance layer so that the construction and interpretation of such resources are not represented as fully automatic or epistemically neutral. **Figure 1** presents the pharmaceutical provenance spine, showing how the article's key components and boundaries are connected within representing transformations, model versions, human decisions, and evidence lineage.

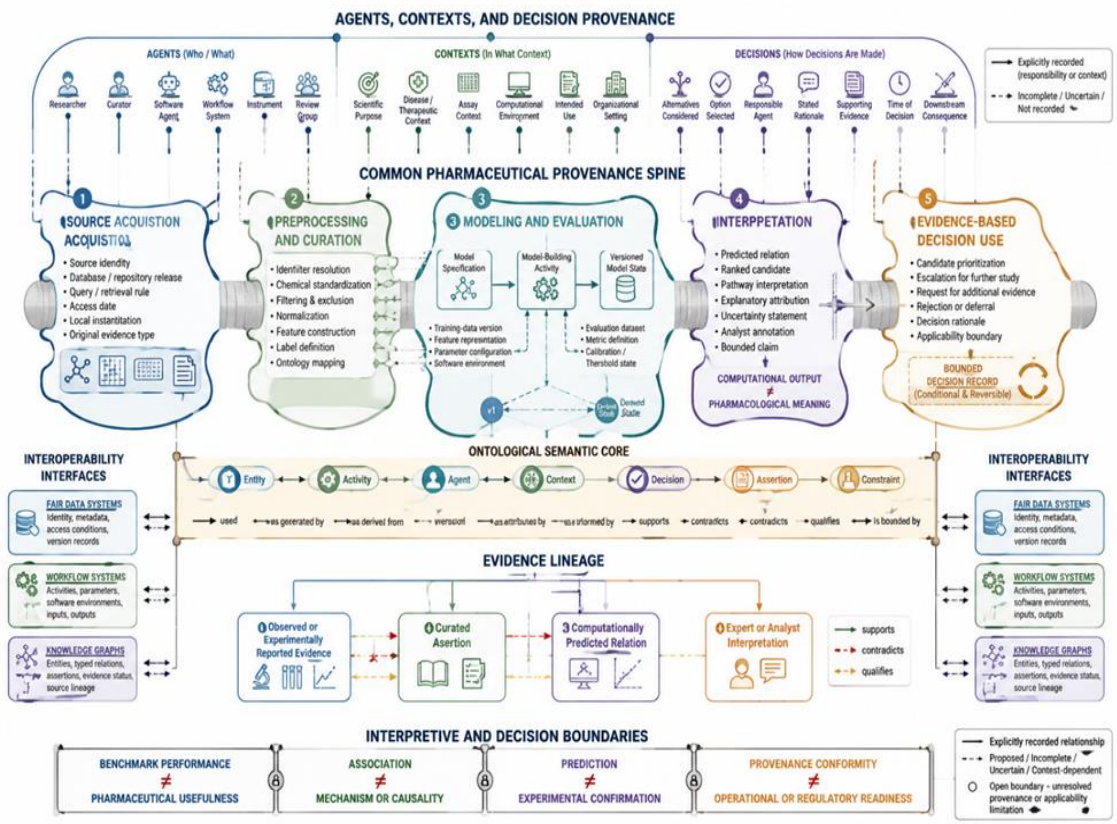

Figure 1. Pharmaceutical Provenance Spine

The figure is an original conceptual synthesis that organizes the article's central contribution across pharmaceutical computation lacks a shared provenance vocabulary, provenance objects across data acquisition, preprocessing, modeling, and interpretation, provenance objects across data acquisition, core ontology classes, relationships, and minimum metadata, core ontology classes, minimum metadata. Arrows and grouping indicate proposed or evidence-supported relationships rather than measured effect sizes. The figure does not represent a validated predictive model, regulatory determination, clinical recommendation, or deployment-ready system.

Evidence lineage connects an assertion to the entities, activities, decisions, and prior assertions that materially support or qualify it. This lineage should retain evidence type, source identity, direction of support, contradictions, uncertainty, and applicability context. The Open Targets Platform demonstrates the pharmaceutical value of integrating multiple evidence sources for systematic target identification and prioritization [22]. Nevertheless, an integrated evidence score or prioritization record is not equivalent to causal proof, experimental replication, clinical benefit, or regulatory acceptability. CPPL therefore represents a prioritization as a bounded computational assertion whose lineage can be inspected, not as a validated therapeutic

conclusion. Its purpose is to expose how evidence was assembled and interpreted while preserving the scientific work still required.

Interoperability with FAIR Data, Workflow Systems, and Knowledge Graphs

Interoperability requires CPPL to connect with existing infrastructures without replacing their specialized functions. FAIR data systems address discovery, access, semantic interpretation, and reuse; workflow systems represent executable process logic; and knowledge graphs connect heterogeneous entities and relations. CPPL is proposed as a provenance bridge among these layers. Systematic integration of biomedical knowledge has shown that heterogeneous evidence can be connected to prioritize drug-repurposing hypotheses [23]. Such integration is scientifically useful, but it also illustrates a boundary: a graph-derived prioritization inherits the provenance, uncertainty, incompleteness, and biases of its component sources. Interoperability should therefore preserve source-specific evidence and derivation paths rather than flatten them into one apparently homogeneous knowledge base.

At the knowledge-graph interface, CPPL classes can be represented as nodes or typed resources, while provenance relationships become explicit edges or predicates. Biomedical knowledge-graph construction depends on entity normalization, ontology alignment, relation modeling, source integration, and decisions about graph scope [24]. CPPL contributes a distinct layer by describing how each node, relation, and assertion entered the graph, which transformation produced it, which version it belongs to, and whether it was curated, observed, computed, or inferred. This distinction is necessary because a graph edge representing an experimentally reported drug–target interaction should not be semantically interchangeable with an edge predicted by a model or inferred from a rule.

Within computational drug discovery, knowledge graphs may support search, candidate prioritization, link prediction, representation learning, and evidence integration [25]. CPPL does not prescribe one graph-learning method or assume that graph connectivity establishes pharmacological meaning. Instead, it specifies provenance requirements for graph-derived objects. A predicted relation should identify the graph version, source subgraph, preprocessing activities, model state, training objective, evaluation context, and interpretation. Where a relation is subsequently reviewed or experimentally investigated, the new evidence should be added as a distinct provenance object rather than overwriting its earlier predicted status. This preserves the difference between a computational hypothesis and later confirmation, contradiction, or qualification.

Interoperability with curated pharmaceutical databases requires mappings among identifiers, entity definitions, relationship vocabularies, source citations, and release histories. DrugBank exemplifies a resource that integrates drug identities, chemical properties, targets, enzymes, transporters, interactions, and associated knowledge within an evolving curated database [26]. CPPL would not reproduce such domain content. It would record which resource version, record, field, mapping, and extraction activity contributed to a local computational object. Mappings should be directional and qualified where definitions differ. A local “drug” entity, for example, may represent an active ingredient, salt, mixture, investigational compound, approved product, or computational structure. Interoperability is therefore achieved through explicit semantic alignment and preserved provenance, not through identifier matching alone.

Adoption, Governance, Extensibility, and Conformity Assessment

Adoption depends on whether CPPL improves traceability without imposing disproportionate annotation and infrastructure burdens. Pharmaceutical organizations differ in data maturity, platform architecture, scientific focus, intellectual-property constraints, and quality-management practices. Work on FAIR implementation in biopharma emphasizes that data value depends on both FAIR properties and fitness-related quality considerations [27]. CPPL should therefore be introduced through bounded use cases rather than enterprise-wide semantic replacement. Initial applications might focus on high-value handoffs, such as transferring a processed omics dataset into target-prioritization modeling, reproducing a molecular-property model, or tracing evidence used in a candidate-ranking decision.

Implementation priorities should be determined by scientific value, reuse potential, risk, and feasibility. Pharmaceutical FAIRification research shows that selecting which datasets to prioritize requires attention to intended use, stakeholder needs, effort, and expected benefit [28]. The same principle applies to provenance. Records supporting consequential, frequently reused, or difficult-to-reconstruct analyses may warrant deeper representation than temporary exploratory outputs. CPPL should consequently offer tiered profiles: a foundational identity-and-derivation profile, a reproducibility profile containing executable context, and an audit profile adding decision rationale, evidence lineage, uncertainty, and review history. These profiles are proposed implementation levels, not validated maturity stages.

Governance must specify ownership, contribution, review, release, deprecation, extension, and dispute-resolution procedures. A pharmaceutical provenance council could include computational scientists, data stewards, ontology specialists, domain experts, quality personnel, and intended users. Its role would be semantic rather than regulatory: maintaining definitions, reviewing extension requests, documenting mappings, and publishing change histories. Cost–benefit frameworks for FAIR pharmaceutical research illustrate why implementation choices should consider expected reuse, decision value, labor, infrastructure, and organizational readiness [29]. CPPL governance should likewise evaluate whether a proposed metadata obligation materially improves traceability or merely increases documentation burden.

Conformity assessment should test declared CPPL profiles rather than issue a general label of compliance. Assessment may examine identifier resolvability, required-field completion, relationship validity, version consistency, lineage continuity, decision linkage, boundary preservation, and interoperability with declared systems. Practical FAIR assessment for research

software demonstrates how executable and evolving software objects can be evaluated through explicit indicators rather than treated as static supplementary materials [30]. CPPL conformity would remain narrower: it would indicate that a provenance record meets a specified semantic profile, not that the underlying data, model, interpretation, or pharmaceutical decision is scientifically valid. **Table 3** organizes the evidence, constructs, and interpretation boundaries needed to develop adoption, governance, extensibility, and conformity assessment within the article's central argument.

Table 3. Evaluation, implementation, and research priorities arising from A Common Provenance Language for Traceable Pharmaceutical Data, Reproducible Bioinformatics, and Auditable Computational Discovery

Evaluation Dimension	What Must be Tested	Suitable Evidence or Assessment	Failure Signal	Interpretive Limitation	Decision Relevance
Semantic Coverage	Whether the ontology can represent material objects, activities, decisions, agents, and constraints across pharmaceutical use cases	Competency questions, expert mapping exercises, missing-concept analysis, cross-domain case studies	Important steps require informal notes or misleading class assignments	Coverage in selected cases does not establish universal applicability	Determines whether CPPL is sufficiently expressive for a proposed implementation
Relationship Precision	Whether users and systems apply provenance relations consistently	Annotation comparison, relation-definition review, graph-consistency testing	The same relation is used for derivation, support, interpretation, and causality	Consistent annotation does not establish that a relation is scientifically correct	Identifies relations requiring clarification, subdivision, or restriction
Metadata Sufficiency	Whether declared profiles preserve information needed for tracing and interpretation	Reconstruction exercises, expert review, missing-field analysis	Analysts cannot identify sources, transformations, versions, or material decisions	More metadata may improve documentation while increasing burden	Supports selection of minimum profile requirements
Workflow Interoperability	Whether provenance can move between workflow systems without losing material meaning	Export–import tests, cross-platform execution records, semantic-difference analysis	Objects are exchanged but relation types, versions, or parameters are lost	Successful exchange does not guarantee reproducible execution	Informs platform integration and mapping priorities
Knowledge-Graph Interoperability	Whether provenance distinctions remain visible after graph integration	Graph transformation tests, source-lineage queries, assertion-status checks	Observed, curated, predicted, and inferred relations become indistinguishable	Queryability does not establish evidential validity	Supports bounded reuse of graph-derived evidence
Version Integrity	Whether revisions, retraining, recalibration, and data updates remain distinguishable	Version-chain audits and change-history reconstruction	Later objects silently replace or become confused with earlier states	Version traceability does not determine whether the change was beneficial	Supports comparison, rollback, and reproducibility decisions
Evidence-Lineage Continuity	Whether claims can be traced to material supporting and conflicting objects	Claim-to-source audits, lineage-completeness review, contradiction checks	A claim lacks a recoverable path to the evidence and decisions that shaped it	Complete lineage does not demonstrate mechanism or truth	Helps reviewers judge what evidence was actually used
Human-Decision Provenance	Whether consequential choices and alternatives are recorded prospectively and intelligibly	Decision-record review, user studies, retrospective reconstruction comparison	Selection rationales are absent, generic, or created only after outcomes are known	A documented rationale may still be biased or mistaken	Supports accountable review and identification of discretionary steps
Annotation Burden	Whether provenance capture is feasible within routine scientific work	Time-on-task studies, user interviews, automation-coverage assessment	Users bypass fields, enter low-quality defaults, or maintain separate unofficial records	Low burden does not guarantee adequate semantic depth	Determines which elements should be mandatory, automated, or optional
Governance Performance	Whether changes, extensions, mappings, and disputes are handled transparently	Governance audits, change-log review, extension-case analysis	Conflicting local variants proliferate without documented alignment	Effective governance in one organization may not transfer directly to another	Supports decisions about stewardship and release management
Conformity Assessment	Whether a record satisfies a declared CPPL profile	Machine-actionable validation plus targeted expert review	A system claims compliance without meeting required semantic or lineage conditions	Conformity concerns record structure, not scientific validity	Enables bounded acceptance, exchange, and remediation decisions
Decision-Use Boundary	Whether downstream users understand what provenance completeness can and cannot establish	User interpretation studies and misuse scenario testing	Traceability is treated as proof of reproducibility, mechanism, clinical utility, or regulatory acceptability	Boundary awareness may differ across stakeholder groups	Determines whether records are safe to reuse in consequential contexts

Limitations and Boundary Conditions

CPPL is a proposed conceptual ontology, not an empirically validated standard. Its upper-level classes and relationships are designed to organize recurring provenance needs, but pharmaceutical domains differ in terminology, evidential practice, confidentiality, and analytical granularity. A representation suitable for molecular-property modeling may omit conditions essential to pharmacometrics, bioinformatics, formulation modeling, or clinical data analysis. The ontology may therefore underrepresent domain-specific objects or become unwieldy if every local distinction is elevated into the common core.

Evaluation must determine whether the proposed abstractions improve communication without erasing scientifically important differences.

Provenance completeness also cannot guarantee reproducibility or validity. Missing software, inaccessible data, stochastic computation, external services, undocumented human knowledge, and evolving reference resources may prevent reconstruction even when a structured record exists. Conversely, an analysis may be repeatable while remaining biased, poorly designed, biologically implausible, or unsuitable for its intended pharmaceutical use. Validation recommendations for biological machine learning reinforce the need to evaluate data, optimization, model, and evaluation design separately [10]. CPPL can expose these elements and their relationships, but it cannot determine automatically whether the associated scientific choices were appropriate.

The greatest misuse risk is epistemic inflation: treating a well-documented computational lineage as evidence that a prediction is causal, experimentally confirmed, clinically useful, safe, synthesizable, developable, or regulatorily acceptable. Knowledge integration can generate valuable hypotheses while inheriting source bias, missingness, and uncertain relation semantics [23]. Accordingly, CPPL records should carry explicit claim types and boundary statements. Conformity with the ontology would indicate only that specified provenance information is represented according to a declared profile. It would not certify a dataset, workflow, model, candidate, therapeutic hypothesis, or organizational decision.

Research Agenda and Implementation Priorities

The first research priority is semantic evaluation. Representative pharmaceutical cases should be converted into competency questions such as whether a user can determine which source release supplied a molecular record, which preprocessing step altered a label, which model state generated a prediction, which evidence informed an interpretation, and which limitations apply to reuse. Comparative annotation studies should examine whether independent experts assign the same classes and relations, while error analyses should identify ambiguous definitions, missing concepts, and excessive granularity. These studies should evaluate both human interpretability and machine-actionable consistency without assuming that agreement alone demonstrates scientific correctness.

The second priority is technical and organizational validation across systems. Pilot implementations should test CPPL mappings to workflow histories, FAIR metadata records, research-software descriptions, model registries, and biomedical knowledge graphs. Evaluation should examine information preservation during export, import, merging, version updates, and graph transformation. Prospective case studies should compare provenance captured during an analysis with provenance reconstructed afterward, because retrospective documentation may omit alternatives, failed steps, or informal decisions. Annotation burden, automation opportunities, access-control requirements, and the treatment of proprietary evidence must be assessed alongside semantic completeness.

Implementation should proceed in stages. A foundational release should define stable upper-level classes, essential relationships, identity rules, version semantics, and a minimal profile. Subsequent domain modules could address chemical standardization, omics preprocessing, molecular modeling, computational pharmacology, and evidence integration. Governance should include public change histories, extension criteria, mapping policies, deprecation procedures, and independent review of conformity tests. Broader adoption should depend on evidence that CPPL improves traceability, reconstruction, and interpretation in defined settings. Until such evidence is available, the ontology should be presented as an extensible research proposal and evaluated through bounded pilots rather than treated as an operational or regulatory standard.

Conclusion

The proposed Common Pharmaceutical Provenance Language provides a structured way to represent the entities, activities, agents, contexts, versions, decisions, assertions, constraints, and evidence relationships through which computational pharmaceutical knowledge is produced. Its original contribution is the pharmaceutical provenance spine: a version-aware and decision-aware semantic connection across acquisition, preprocessing, modeling, interpretation, and downstream evidence use. The proposal is intended to make computational lineage more inspectable while preserving distinctions that are often lost when data, workflows, models, and knowledge graphs are integrated. It does not claim that transparent provenance guarantees reproducibility, scientific validity, biological mechanism, therapeutic usefulness, clinical benefit, regulatory acceptability, or deployment readiness. Its value must be established through semantic testing, cross-platform implementation, reconstruction studies, burden assessment, governance evaluation, and domain-specific validation. A common provenance language may support more auditable computational discovery only when its records remain technically interoperable, scientifically contextualized, and explicit about the boundaries of the conclusions they can support.

Acknowledgments: None

Conflict of interest: None

Financial support: None

Ethics statement: None

References

1. Grüning BA, Chilton J, Köster J, Dale R, Soranzo N, van den Beek M, et al. Practical computational reproducibility in the life sciences. Cell Syst. 2018;6(6):631-5. doi:10.1016/j.cels.2018.03.014
2. Leipzig J, Nüst D, Hoyt CT, Ram K, Greenberg J. The role of metadata in reproducible computational research. Patterns (N Y). 2021;2(9):100322. doi:10.1016/j.patter.2021.100322
3. Wise J, Grebe de Barron A, Splendiani A, Balali-Mood B, Vasant D, Little E, et al. Implementation and relevance of FAIR data principles in biopharmaceutical R&D. Drug Discov Today. 2019;24(4):933-8. doi:10.1016/j.drudis.2019.01.008
4. Bonner S, Barrett IP, Ye C, Swiers R, Engkvist O, Bender A, et al. A review of biomedical datasets relating to drug discovery: A knowledge graph perspective. Brief Bioinform. 2022;23(6). doi:10.1093/bib/bbac404
5. Heil BJ, Hoffman MM, Markowetz F, Lee SI, Greene CS, Hicks SC. Reproducibility standards for machine learning in the life sciences. Nat Methods. 2021;18(10):1132-5. doi:10.1038/s41592-021-01256-7
6. Cohen-Boulakia S, Belhajjame K, Collin O, Chopard J, Froidevaux C, Gaignard A, et al. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. Future Gener Comput Syst. 2017;75:284-98. doi:10.1016/j.future.2017.01.012
7. Khan FZ, Soiland-Reyes S, Sinnott RO, Lonie A, Goble C, Crusoe MR. Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv. Gigascience. 2019;8(11). doi:10.1093/gigascience/giz095
8. Wratten L, Wilm A, Göke J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nat Methods. 2021;18(10):1161-8. doi:10.1038/s41592-021-01254-9
9. Brito JJ, Li J, Moore JH, Greene CS, Nogoy NA, Garmire LX, et al. Recommendations to enhance rigor and reproducibility in biomedical research. Gigascience. 2020;9(6). doi:10.1093/gigascience/giaa056
10. Walsh I, Fishman D, Garcia-Gasulla D, Titma T, Pollastri G, ELIXIR Machine Learning Focus Group, et al. DOME: Recommendations for supervised machine learning validation in biology. Nat Methods. 2021;18(10):1122-7. doi:10.1038/s41592-021-01205-4
11. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017;35(4):316-9. doi:10.1038/nbt.3820
12. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018;46(W1). doi:10.1093/nar/gky379
13. Ewels PA, Peltzer A, Fillinger S, Patel H, Alneberg J, Wilm A, et al. The nf-core framework for community-curated bioinformatics pipelines. Nat Biotechnol. 2020;38(3):276-8. doi:10.1038/s41587-020-0439-x
14. Leo S, Crusoe MR, Rodríguez-Navas L, Sirvent R, Kanitz A, De Geest P, et al. Recording provenance of workflow runs with RO-Crate. PLoS One. 2024;19(9). doi:10.1371/journal.pone.0309210
15. Wilkinson MD, Sansone SA, Schultes E, Doorn P, Santos LOBDS, Dumontier M. A design framework and exemplar metrics for FAIRness. Sci Data. 2018;5:180118. doi:10.1038/sdata.2018.118
16. Wilkinson MD, Dumontier M, Sansone SA, Bonino da Silva Santos LO, Prieto M, Batista D, et al. Evaluating FAIR maturity through a scalable, automated, community-governed framework. Sci Data. 2019;6(1):174. doi:10.1038/s41597-019-0184-5
17. Musen MA, O'Connor MJ, Schultes E, Martinez-Romero M, Hardi J, Graybeal J. Modeling community standards for metadata as templates makes data FAIR. Sci Data. 2022;9(1):696. doi:10.1038/s41597-022-01815-3
18. Cox SJD, Gonzalez-Beltran AN, Magagna B, Marinescu MC. Ten simple rules for making a vocabulary FAIR. PLoS Comput Biol. 2021;17(6). doi:10.1371/journal.pcbi.1009041
19. González-Cebrián A, Bradford M, Chis AE, González-Vélez H. Standardised versioning of datasets: A FAIR-compliant proposal. Sci Data. 2024;11(1):358. doi:10.1038/s41597-024-03153-y
20. Mitchell SN, Lahiff A, Cummings N, Hollocombe J, Boskamp B, Field R, et al. FAIR data pipeline: Provenance-driven data management for traceable scientific workflows. Philos Trans A Math Phys Eng Sci. 2022;380(2233):20210300. doi:10.1098/rsta.2021.0300
21. Callahan TJ, Tripodi IJ, Stefanski AL, Cappelletti L, Taneja SB, Wyrwa JM, et al. An open source knowledge graph ecosystem for the life sciences. Sci Data. 2024;11(1):363. doi:10.1038/s41597-024-03171-w
22. Ochoa D, Hercules A, Carmona M, Suveges D, Gonzalez-Uriarte A, Malangone C, et al. Open Targets Platform: Supporting systematic drug-target identification and prioritisation. Nucleic Acids Res. 2021;49(D1). doi:10.1093/nar/gkaa1027
23. Himmelstein DS, Lizee A, Hessler C, Brueggeman L, Chen SL, Hadley D, et al. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. Elife. 2017;6. doi:10.7554/eLife.26726
24. Nicholson DN, Greene CS. Constructing knowledge graphs and their biomedical applications. Comput Struct Biotechnol J. 2020;18:1414-28. doi:10.1016/j.csbj.2020.05.017
25. Zeng X, Tu X, Liu Y, Fu X, Su Y. Toward better drug discovery with knowledge graph. Curr Opin Struct Biol. 2022;72:114-26. doi:10.1016/j.sbi.2021.09.003

26. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, et al. DrugBank 5.0: A major update to the DrugBank database for 2018. Nucleic Acids Res. 2018;46(D1). doi:10.1093/nar/gkx1037
27. Harrow I, Balakrishnan R, Küçük-McGinty H, Plasterer T, Romacker M. Maximizing data value for biopharma through FAIR and quality implementation: FAIR plus Q. Drug Discov Today. 2022;27(5):1441-7. doi:10.1016/j.drudis.2022.01.006
28. Alharbi E, Gadiya Y, Henderson D, Zaliani A, Delfin-Rossaro A, Cambon-Thomsen A, et al. Selection of data sets for FAIRification in drug discovery and development: Which, why, and how? Drug Discov Today. 2022;27(8):2184-90. doi:10.1016/j.drudis.2022.05.010
29. Alharbi E, Skeva R, Juty N, Jay C, Goble C. A FAIR-Decide framework for pharmaceutical R&D: FAIR data cost-benefit assessment. Drug Discov Today. 2023;28(4):103510. doi:10.1016/j.drudis.2023.103510
30. Martín del Pico E, Gelpí JL, Capella-Gutiérrez S. FAIRsoft—a practical implementation of FAIR principles for research software. Bioinformatics. 2024;40(8). doi:10.1093/bioinformatics/btae464