CubeDynamics: An Inspectable Grammar for Environmental Data Analysis

Ty Tuff
Environmental Data Science Innovation & Impact Lab
Cooperative Institute for Research in Environmental Sciences
University of Colorado Boulder, Boulder, Colorado, USA
ORCID 0000-0001-5249-5197

September 26, 2026

Abstract

Environmental questions that sound simple often compress consequential choices about data source, spatial and temporal support, transformations, missing values, and operation order into short scripts. Re-running a script supports computational repeatability but does not by itself reveal the scientific statement encoded by those choices. CubeDynamics is an open-source Python package that places an inspectable interface around ordinary xarray objects. Source-qualified environmental nouns retain provider and product identity; configured verbs expose parameters and execute in written order; semantic state describes the current result; and a local trace records operations completed inside each pipe. Its contribution is this combination of source identity, semantic transitions, order-aware diagnostics, ordinary-array interoperability, and contract-focused verification—not a new numerical array, workflow scheduler, or data-cube store. A bounded South Dakota protocol illustrates source selection, condition construction, composition, and the limits of branch lineage. Evidence from public release candidate 0.1.0rc3 and later development records remains revision-specific. The principal unresolved design issue is that missing numerical inputs currently become false conditions rather than explicit unknowns. The objective is to make environmental complexity readable without concealing the assumptions beneath concise code.

1 Introduction

The question *Where did locally warm days coincide with trace or no rain?* sounds modest. An analyst could load temperature and precipitation, turn each variable into a stated true/false criterion—called a *condition* in this paper—combine the conditions, and make a map. Yet each step opens another question. Which provider-published datasets, called *products* here, supplied the values? Which dates and grids are represented, and what area or time interval does each value summarize? Is "warm" a fixed cutoff, a rank within the selected sample, or a departure from a longer-term reference climate? Does "trace or no rain" mean a daily precipitation total below a stated value? How are missing observations handled? Were products placed on a common grid or matched in time, and in what order were they transformed and summarized? A short question can require substantial computational machinery.

There is a deeper problem. Even perfectly rerunnable code may not make the question inspectable. Consider the same 2×2 temperature field, cutoff, and spatial mean—the arithmetic average across the four cells—in Figure 1. Applying the cutoff first asks what fraction of cells satisfy a condition. Averaging first asks whether the regional mean satisfies it. Both sequences can be computationally repeatable; they are different scientific statements.

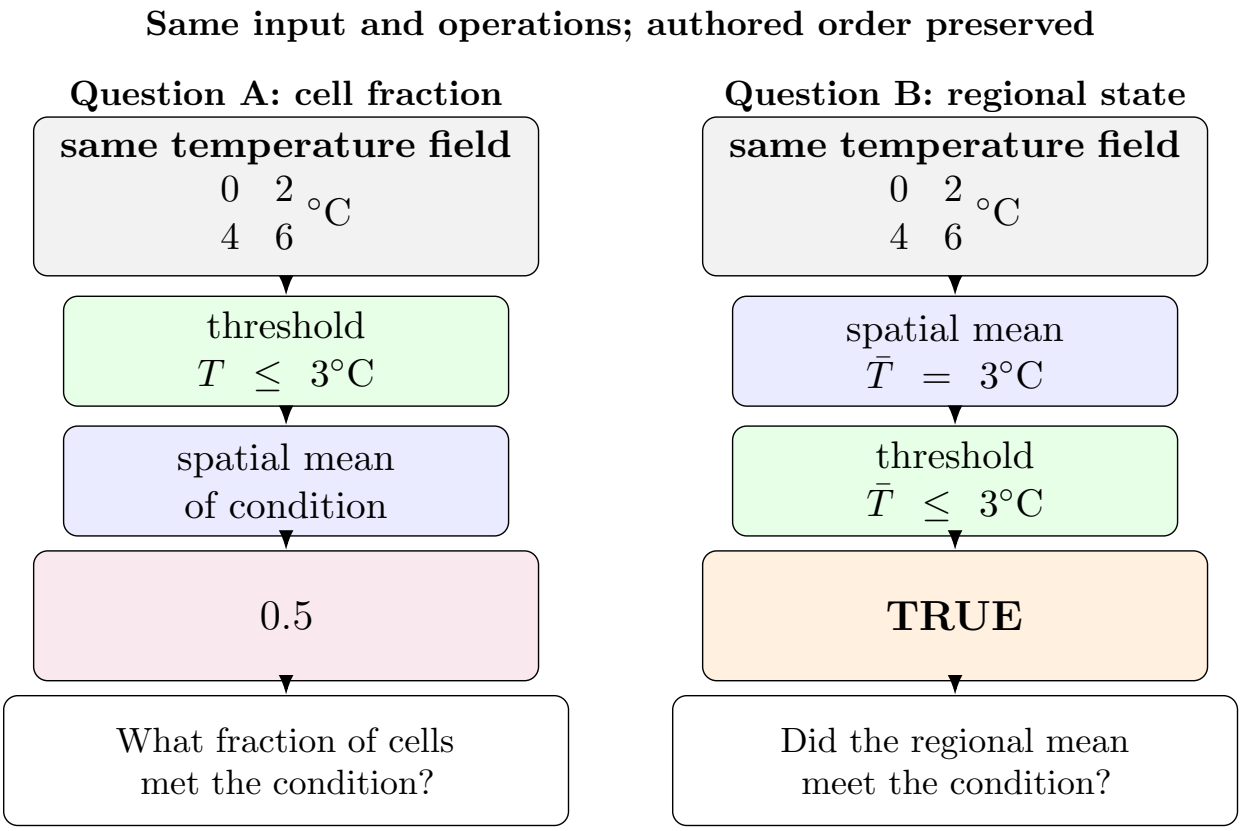

Figure 1: The order written changes scientific meaning. For the same field $\{0, 2, 4, 6\}$ °C and inclusive cutoff $T \leq 3$ °C, applying the cutoff before averaging produces the fraction of cells satisfying it, 0.5. Averaging first produces a regional mean of 3 °C, which then satisfies the cutoff and produces one true/false result, `TRUE`. CubeDynamics executes both statements and preserves rather than equates them.

Reproducibility terminology varies among fields. In this paper, *computational repeatability* means that the same code can be executed again with the same inputs and software environment to obtain the same result. Broader uses of *reproducibility* may instead ask whether another team can obtain a consistent result with independently assembled code, data, or methods (Peng, 2011; Goodman et al., 2016; National Academies of Sciences, Engineering, and Medicine, 2019; Essawy et al., 2020). A repeatable analysis can still encode a mistaken question or inappropriate choice (Leek and Peng, 2015). *Scientific inspectability* denotes a different and narrower property: a reviewer can recover the provider and product, exact data request, parameter values, order of operations, changes in what intermediate results represent, record of completed steps, and evidence limited to what was actually tested. This emphasis has an older precedent in systems for auditing data analyses by recording executed statements and the objects they accessed or changed (Becker and Chambers, 1988). Inspectability does not establish that an analysis is scientifically appropriate or suitable for a

decision. It makes the choices available for examination.

CubeDynamics is intended for environmental researchers and research software engineers who already use xarray, a Python library for multidimensional arrays with named dimensions and coordinates (Hoyer and Hamman, 2017), together with notebooks and geospatial libraries. The package adds shared names for environmental inputs and transformations and records what the current result represents after each stage. It does not introduce another numerical data structure or schedule jobs. Its central question is:

> *What exactly did this analysis ask the observations?*

1.1 Definitions and scope

Several project-specific terms are essential. An *observation* is any value supplied by an environmental product; it may be measured, interpolated, remotely retrieved, modeled, or derived. A *noun* is a readable name for environmental information, and a required *source qualifier* identifies the provider and product supplying it. A *verb* is a configured transformation applied in written order. *Semantic state* records the package's current interpretation of the object, while the *trace* records operations completed inside one pipe. A Boolean *condition* states where or when a criterion is satisfied; a *summary* has reduced one or more dimensions. *Observation support* is the area or interval represented by a value, not merely its coordinate label. These records expose software claims for review; they do not independently verify metadata, scientific suitability, or environmental truth. The complete glossary and contracts are provided in the Supplementary Information (SI).

This paper makes four bounded contributions: (1) a source-qualified vocabulary for environmental inputs; (2) configured transformations whose declared semantic transitions follow authored order; (3) a local state and trace that remain attached to ordinary xarray-compatible work; and (4) a verification framework linking documented contracts to reference cases, invariants, controlled comparisons, and expected rejection. Claims of improved comprehension, scientific correctness, productivity, or decision quality are outside the present evaluation.

2 From narrative pipelines to a shared grammar

Ben Fry's computational information-design work is relevant because it named a sequence—acquire, parse, filter, mine, represent, refine, and interact—that joined data handling, analysis, and presentation into one readable process (Fry, 2004, 2007). The contribution was not that every analysis must follow seven stages, but that naming consequential stages can make a complex computational argument easier to inspect. The same design sensibility appears in ggplot2's layered implementation of the Grammar of Graphics and in the tidyverse's named operations for tabular work (Wilkinson, 2005; Wickham, 2010, 2014; Wickham et al., 2019).

When my work moved from R data frames—table-like objects organized by rows and columns—to larger environmental arrays in Python, xarray provided arrays whose dimensions and coordinates have names (Hoyer and Hamman, 2017), while Dask allowed those arrays to be evaluated in parallel or only when their values were requested (Rocklin, 2015). I missed the narrative continuity of the

earlier R workflow. The analysis still had a sequence, yet its data sources and steps that changed a field into a condition or summary were easily dispersed across loading routines, project helper functions, array expressions, and metadata. CubeDynamics began as an attempt to recover that readable progression while addressing a different problem: environmental data need their source and spatial and temporal meaning retained as they are transformed, together with evidence that generic data-manipulation operations do not supply by themselves.

That route is personal, but the design problem is shared. Information-visualization research has long represented analysis as transformations among data states (Chi, 2000). Literate programming argued that programs should be organized for human explanation as well as machine execution (Knuth, 1984). Computational notebooks place executable work beside narrative but expose recurring tensions between exploration, explanation, execution order, and reproducibility (Rule et al., 2018; Pimentel et al., 2019). More general workflow scholarship treats analysis as a progression of consequential choices rather than a single calculation (Stoudt et al., 2021). CubeDynamics belongs to this broader lineage. Its particular focus is what environmental data and intermediate results mean as they move through an executable analysis: what was observed, from which product, over what spatial area or time interval, through which transformations, and with what evidence.

The idea then grew through workflows involving climate and vegetation observations, fire objects, terrain, roads, streamflow, methods for comparing timing across locations, teaching notebooks, and worked environmental vignettes. These projects differed scientifically, but repeatedly needed the same connective layer: a recognizable name for the environmental information entering an analysis, explicit transformations written in analytical order, and a way to inspect what object existed after each stage. The need was not another storage system. It was a shared interface between project methods and the numerical infrastructure beneath them.

The term *data cube* has several technical lineages. Database research formalized cubes as multidimensional aggregation operators over relational data (Gray et al., 1997); an Earth-system data cube instead represents sampled fields with named spatial, temporal, and possibly vertical, variable, or ensemble axes (Baumann, 2021; Mahecha et al., 2020; Montero et al., 2024). Its analytical advantage is not that calculations become mathematically impossible elsewhere, but that histories, neighborhoods, cross-variable relationships, and reductions over named axes become native operations on one object. Work on analysis-ready and semantic Earth-observation cubes further shows that organization, preprocessing, and semantic labels remain consequential to what downstream analysis can support (Dwyer et al., 2018; Augustin et al., 2019; van der Meer et al., 2022; Baumann, 2024). Map algebra classified raster operations by whether they act locally, over neighborhoods, within zones, or globally; cubic and multidimensional map algebra extended this logic across space and time (Tomlin, 2012; Mennis et al., 2005; Mennis, 2010). Array and coverage algebras similarly treat multidimensional fields as first-class values that can be subset, transformed, aggregated, and combined (Baumann, 2021; Baumann et al., 2021). CubeDynamics does not store cubes or invent these algorithms. It adds environmental source identity, declared semantic transitions, order diagnostics, and an inspection path around established array work.

The linguistic framing is a design aid, not a claim that Python code is natural language. Environmental nouns identify what enters an analysis; verbs identify transformations; parameters specify how those transformations are performed; and written order supplies part of the syntax. The package does not ask users to write a separate language, generate every valid program, or claim that its composition rules are complete.

Table 1: Closest technical traditions and the narrower role of CubeDynamics.

Tradition or system	Primary role	Relation to CubeDynamics
NumPy, xarray, and Dask	Numerical arrays, labeled dimensions, and lazy or parallel execution (Harris et al., 2020; Hoyer and Hamman, 2017; Rocklin, 2015)	Reused directly; CubeDynamics does not replace their computation
Map, array, and coverage algebra	Operator vocabularies over spatial and multidimensional fields (Tomlin, 2012; Mennis et al., 2005; Baumann et al., 2021)	Precedent for composable whole-field operations; source and semantic records are the added focus
Domain libraries such as xclim and clisops	Maintained climate indicators and data operations (Bourgault et al., 2023; clisops contributors, 2026)	Candidate numerical methods to call, not methods reimplemented by the grammar
Workflow and process-graph systems	Portable execution, planning, and constrained composition (Ludäscher et al., 2006; Gil et al., 2011; Crusoe et al., 2022; Cohen-Boulakia et al., 2017; Schramm et al., 2021)	Broader workflow scope; CubeDynamics retains only a local in-Python state and trace

Different contributors can work at different layers: project methods may remain ordinary Python functions, while promotion into the shared vocabulary requires documented meaning, evidence, tests, and teaching material. The fuller comparison and contribution pathways are reported in the SI.

3 Design: a readable expression and an inspectable foundation

Using the definitions in Section 1.1, the public pattern can be written as:

```
analysis = pipe(noun) | verb(parameters) | verb(parameters)
```

Here `analysis` names the resulting pipe object. `pipe(noun)` wraps the starting scientific Python object and initializes local state and trace. Each vertical bar invokes the next configured verb once, from left to right; this statement concerns Python operation order, not how often a lazy computational graph may later evaluate. The expression remains ordinary Python rather than a separate language.

3.1 Source-qualified nouns

A noun provides a stable scientific entry point; its source qualifier preserves the provider–product identity that gives the values context. This follows observation and metadata traditions that distinguish the observed property, procedure, feature of interest, result, units, and coordinates (Madin et al., 2007; Janowicz et al., 2019; Haller et al., 2019; Hassell et al., 2017). Formal variable-description frameworks go further by decomposing environmental variable names into properties, observed

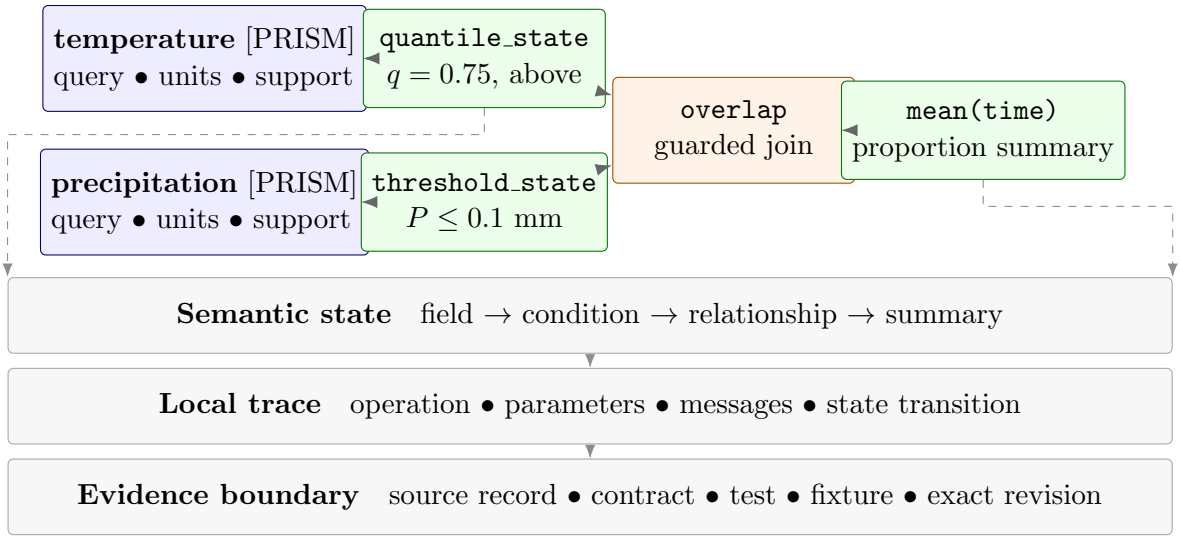

Figure 2: One analytical statement with its inspection layers. Source-qualified inputs enter separate transformations before an exact-coordinate combination and temporal reduction. Semantic state records the declared interpretation after each operation; each branch retains a local trace; and supporting evidence remains tied to identified contracts, tests, fixtures, and revisions. The figure describes the architecture rather than an empirical environmental result.

entities, contexts, and constraints for human and machine interoperability; implementation experience also shows that complex variables can admit more than one plausible decomposition (Coussot et al., 2024). CubeDynamics nouns are intentionally lighter-weight entry points rather than ontology terms. Thus the `prism` and `gridmet` forms of `temperature` share a vocabulary without becoming equivalent measurements. Their native variables, statistical definitions, units, grids, support, revisions, requests, provenance, and quality-assurance records remain source-specific (Daly et al., 2008; Abatzoglou, 2013; Tuff, 2026a).

A common noun supports discovery and shared code; the qualifier prevents that convenience from silently becoming a substitution decision. Computational compatibility is not scientific interchangeability, and provenance remains essential to fitness for purpose (Whitfield, 2012).

3.2 Verbs, order, state, and trace

A configured verb has a documented input and output rule. A threshold changes a continuous observation into a Boolean condition; reducing that condition over a dimension creates a summary. Because these transformations change what the object means, semantic state must describe the result rather than merely preserve input labels. The current two-valued condition model has an important limitation: comparison with a missing numerical value produces `false`, and a later condition mean includes that position in its denominator. A condition frequency therefore means a proportion over all indexed positions, not a proportion over valid observations, unless the input is known to be fully finite. The SI treats a validity mask, explicit missingness policy, or three-valued condition as an unresolved stable-release design decision rather than a cosmetic limitation.

The order in which verbs are authored is part of the scientific statement. CubeDynamics calls

each stage once, from left to right, and does not rearrange the expression. Some orders are both valid but meaning-changing, as in Figure 1; some remove information required later; and some are conditionally equivalent only under explicit assumptions. Semantic workflow research provides close precedent for using scientific concepts and constraints to distinguish meaningful compositions from invalid ones (Gil et al., 2011; Bowers, 2012; Kruiger et al., 2021).

After each stage, semantic state updates the package's interpretation of the object and the information believed to remain. The trace records operations and transitions completed inside the current pipe. Provenance scholarship distinguishes questions about which source data contributed to a result, where those data originated, and which activities produced later objects (Buneman et al., 2001; Moreau et al., 2011). Geoscience publication guidance likewise distinguishes an execution trace, the provenance of one completed run, and a reusable workflow or method (Gil et al., 2016). A local trace answers only part of that larger provenance problem: preparation, separately unwrapped branches, arbitrary callables, and downstream use require complementary records (Davidson and Freire, 2008; World Wide Web Consortium, 2013; Soiland-Reyes et al., 2022; Leo et al., 2024). Neither state nor trace verifies incoming metadata. The complete fields and branch boundary are listed in the SI.

Alignment follows the same principle. Equal coordinate labels do not prove equal spatial footprints or temporal support. The change-of-support literature shows why differences in scale, aggregation, and support can alter analysis even when data appear computationally compatible (Gotway and Young, 2002; Dungan et al., 2002; Baumann, 2024). CubeDynamics therefore requires exact coordinates for direct composition and reports known support differences. It does not silently *reproject* coordinates into another spatial reference system, *resample* values onto another grid, *truncate* unmatched coordinates, or invent missing support. Researchers remain responsible for *harmonization*: the scientifically justified transformations used to make differing products comparable for a stated question.

3.3 Ordinary objects and project extensions

The adoption path is reversible: xarray object → `pipe` → inspected operations → `unwrap` → xarray object. NumPy- and Dask-backed values retain their established behavior where operations permit it. `unwrap()` returns the current value; it does not force computation, certify the result, or complete the wider workflow.

Project extensions do not require a new data class or registration through a plugin system. An ordinary Python function that accepts the current object and returns the next object can participate in a pipe. That low barrier is important for research groups developing domain methods. Promotion into the shared package remains stricter because a public scientific name and its documented input and output rules will be reused outside the originating project. The repository therefore encourages project-owned vocabularies first, followed by broader promotion only when meaning, evidence, tests, and teaching material are ready together (Tuff, 2026a).

3.4 Evidence beneath the abstraction

Shorter code does not make the underlying data work disappear. It can instead hide retrieval, variable selection, coordinate interpretation, provenance, and validation behind a brief noun call. To

remain inspectable, the abstraction must provide a path back to those details. Metadata integration itself is scientific work rather than a cost-free administrative step (Edwards et al., 2011).

This motivates the design maxim:

> *The shorter the analytical expression becomes, the stronger the evidence underneath it must become.*

A reader should be able to descend from a noun to its source record and from a verb to its parameters, change in meaning, trace entry, and behavioral evidence. Provider availability, fixture identity, metadata interpretation, numerical plausibility, and scientific suitability answer different questions; the SI separates their evidence.

4 Worked protocol: locally warm and trace-or-no-rain days

The Working Lands protocol translates the opening question into an explicit, deliberately narrow statement: within a central South Dakota bounding box during July 2024, identify dates when PRISM daily maximum temperature is at or above each grid position's empirical 0.75 quantile and PRISM daily precipitation is at or below 0.1 mm. Both inputs use the same PRISM daily product family, and the reviewed fixture places them on identical coordinates, so this protocol avoids rather than solves cross-product harmonization. The source request is visible before modification:

```
from cubedynamics import data, pipe, verbs as v

query = dict(
    source="prism", bbox=[-101.2, 43.7, -100.4, 44.3],
    start="2024-07-01", end="2024-07-31",
)
temperature = data.temperature(**query, statistic="maximum")
precipitation = data.precipitation(**query)

warm_branch = pipe(temperature) | v.quantile_state(
    quantile=0.75, direction="above", name="warm_july_day"
)
trace_branch = pipe(precipitation) | v.threshold_state(
    threshold=0.1, direction="below", name="trace_or_no_rain"
)

analysis = (
    pipe(warm_branch.unwrap())
    | v.overlap(
        trace_branch.unwrap(),
        name="warm_and_trace_or_no_rain",
    )
    | v.mean(dim="time", keep_dim=False)
)
```

Temperature becomes a locally ranked condition, precipitation becomes an inclusive trace-or-no-rain condition, `overlap` applies logical conjunction, and the temporal mean returns an unweighted

proportion over the 31 indexed days. In this protocol, `direction="above"` and `direction="below"` are inclusive. The reviewed fixture is reported as fully finite, so the package's missing-as-false behavior does not affect this particular calculation. "Warm" is a rank within one July sample, not a climatological anomaly. Sample-quantile definitions vary (Hyndman and Fan, 1996). The reviewed implementation delegates to xarray's default method rather than recording an explicit CubeDynamics method choice. The result is not a drought, impact, causal, or risk classification (Leonard et al., 2014; Zscheischler et al., 2020).

Table 2: What the Working Lands protocol demonstrates and what remains outside the final pipe.

Object	Retained inspection record	Boundary
`temperature`, `precipitation`	PRISM source qualifier, request, units, coordinates, and support metadata	A common provider does not validate the thresholds or intended use
`warm_branch`, `trace_branch`	Separate parameters, state transitions, messages, and traces	Unwrapping does not merge both histories into the later pipe
`analysis`	Exact-alignment requirement, logical overlap, temporal reduction, and summary state	Complete reconstruction requires retaining both branch records beside the final trace

The SI records the reviewed fixture checksum, query, acceptance checks, and repository-reported output range. Because the binary fixture and generated inspection bundle are not part of the present archive, the protocol is used here to demonstrate analytical expression and evidence boundaries rather than to support a new environmental result.

5 Implementation verification and semantic consistency testing

The worked example raises a practical question: what evidence shows that the software performed the stated analysis and described the result correctly?

The word *validation* can refer to different questions. In computational modeling, *verification* commonly asks whether software correctly implements the intended model, whereas *validation* asks how well that model represents the physical world for an intended use (Oberkampf and Trucano, 2002). CubeDynamics is not itself a physical model, and the evaluation reported here does not compare PRISM, gridMET, or any threshold with independent environmental truth. *Implementation verification* asks whether an operation performs its documented computation. *Semantic consistency testing* asks whether returned values, dimensions, units, state, trace, and diagnostics agree with the documented software contract. This terminology avoids implying validation of a data product, scientific hypothesis, or decision rule.

Scientific software is difficult to test because the correct result may be unknown or too expensive to calculate independently. The mechanism that decides whether a test output is correct is called a *test oracle*; its absence or weakness is the *oracle problem* (Barr et al., 2015). A systematic review identified oracle problems, large input spaces, floating-point behavior, limited real-world test data, and difficulties separating code from the scientific model as recurring challenges (Kanewala and Bieman, 2014). No single test therefore establishes correctness. CubeDynamics combines four kinds of evidence:

1. *Reference examples and direct comparisons.* Small fixed arrays and checksum-controlled fixtures have expected results, while piped calculations are compared with direct xarray calculations. Reference datasets are a long-standing approach to black-box testing of scientific numerical software (Cox and Harris, 1999). Such examples contribute to software trustworthiness but do not prove complete correctness (Hook and Kelly, 2009).
2. *Properties and invariants.* Tests check statements that must remain true across examples: a condition contains Boolean state, a reduction removes the named variation, variance changes known units appropriately, and composition does not silently alter coordinates. This follows the broader property-based-testing idea of checking general behavior rather than only individual answers (Claessen and Hughes, 2000).
3. *Relations among controlled runs.* Metamorphic testing provides a general strategy for checking known relationships among outputs when a complete expected result is unavailable (Segura et al., 2016; Lin et al., 2020). In this paper, the threshold/mean pair is described more narrowly as an order-semantics regression: both expressions execute, but the tests require their different scientific objects and questions to remain distinguishable.
4. *Expected rejection and regression.* Inputs that violate a contract should fail with a useful explanation. Once a defect is repaired, a regression test preserves the repaired behavior. Semantic workflow systems and concept-aware geospatial workflow synthesis provide related precedents for constraining meaningful composition (Gil et al., 2011; Kruiger et al., 2021). Execution-context ontologies separately address whether a workflow rerun can be validated against an earlier execution (Miksa and Rauber, 2017).

The semantic layer matters because a numerical value can be correct while its description is wrong. For example, averaging a Boolean condition may correctly produce 0.5, yet retaining the label "Boolean condition" would misdescribe a proportion; similarly, a variance may be numerically correct while incorrectly retaining the input units. RC1 testing exposed errors of this kind as well as correct guardrails. Repairs covered condition summaries, unit propagation, canonical condition outputs, plotting dispatch, and the two valid threshold/mean orders. Exact-coordinate enforcement and rejection of event detection after temporal variation was removed were retained as intended behavior (Tuff, 2026c,d).

Table 3: Principal software claims, supporting checks, and explicit boundaries.

Claim	Supporting checks	Boundary
Authored order is preserved	Hand-calculated threshold/mean examples and order regressions	The catalog does not cover every possible pair of operations
Condition reduction changes meaning	Direct calculation and state-transition checks	Missing inputs remain false in the present two-valued condition model
Composition does not silently harmonize	Exact-coordinate and known-support diagnostics	Coordinate equality does not establish scientific comparability
Invalid event transitions are rejected	Expected-failure tests after required temporal variation is removed	Rejection validates a documented contract, not environmental event identity

The public rc3 artifact, the separately checked Working Lands repository revision, and later development records are not one evaluated object. Aggregate development counts are therefore omitted from the main evidence claim. The SI reports the revision boundaries and the archival materials required before post-rc3 behavior can support a unified release-level claim.

6 Discussion

6.1 What the present evidence establishes

The implemented contribution is narrower than a universal environmental language. The current evidence supports source-qualified inputs, authored execution order, documented state transitions, local traces, exact-coordinate composition, selected support diagnostics, and informative rejection for cataloged contract violations. It does not establish usability gains, complete contract coverage, scientific appropriateness, uncertainty propagation, end-to-end workflow reconstruction, or improved decisions.

Three issues are stable-release priorities. First, missing numerical inputs currently become false conditions rather than explicit unknowns. Second, separately unwrapped branches retain separate records but do not automatically contribute complete histories to the final trace. Third, the manuscript's release and development evidence must be frozen as one reproducible submission bundle before later behavior can support archival claims. These are substantive boundaries of the present design.

6.2 A community of environmental vocabularies

The immediate use of CubeDynamics is to make an existing xarray analysis readable and reviewable. Scientists can inspect a familiar object without abandoning ordinary Python; domain-method developers can publish project vocabularies; source contributors can pair a noun with its evidence; and reviewers can follow a concise expression into the records beneath it.

This contribution model also prevents one universal vocabulary from absorbing every scientific assumption. Core grammar behavior should remain small and general. Shared nouns and verbs should be reusable across projects. Domain assumptions belong first in project packages where their scientific community can evaluate them. The built-in fire, synchrony, terrain, road, and streamflow work demonstrates possible extension shapes; it does not make every project method part of the minimal grammar.

6.3 Toward larger environmental synthesis

Earth-system research increasingly organizes heterogeneous observations into multivariate data cubes (Mahecha et al., 2020; Montero et al., 2024). Interoperability among those systems remains an active problem (Giuliani et al., 2019), and whether data are ready for analysis depends on the task, scale, metadata, and intended interpretation (Baumann, 2024). CubeDynamics' long-term design question is therefore not merely how to load many datasets. It is whether the analytical argument of a large synthesis can remain readable while each abstraction stays connected to source identity, support, transformations, provenance, and evidence.

The aspirational endpoint may be a page-sized specification rather than a five-line script: temperature, precipitation, vegetation, fire history, terrain, hydrology, roads, soils, species, and infrastructure entering distinct analytical paths; some summarized, some combined, and some deliberately kept

separate. A scientist should be able to read that argument, a computer to execute it, and a reviewer to inspect its components. The current implementation demonstrates primarily linear pipes and a bounded combination of two inputs. It does not automatically reconcile heterogeneous products, provide a complete language for arbitrary branching, select sources autonomously, or exercise scientific judgment.

Longer-term, the explicit vocabulary may also be useful when bounded software agents select tools from language descriptions. Recent work evaluates agents on data-driven scientific analysis, constrains language interpretation through structured workflow generation, and studies ambiguity in tool selection (Gu et al., 2024; Balis et al., 2026; Liu et al., 2026). For CubeDynamics, however, this remains a design hypothesis: no agent evaluation, exported tool-schema study, or comparative error analysis is reported. Any such use would still require restricted tools, pinned revisions, preserved records, and human scientific responsibility.

State remains a compact software interpretation rather than verification of metadata; support records expose but do not resolve comparability; arbitrary Python callables retain responsibility for their own intent; uncertainty is not propagated; and source records age. Environmental-modeling scholarship treats uncertainty identification, characterization, and communication as work spanning the modeling process, not as a metadata field that can be added at the end (Refsgaard et al., 2007). Complete reconstruction still requires archived environments, inputs, revisions, and provenance outside the pipe (Sandve et al., 2013; Wilson et al., 2017; Wilkinson et al., 2016).

7 Conclusion

CubeDynamics places source identity, authored transformations, changing interpretation, and local inspection records around ordinary xarray objects. The Working Lands protocol demonstrates the present two-input scope while exposing the incomplete branch-lineage boundary. The extension model allows communities to add domain methods without claiming a universal language or new numerical algorithms.

The goal is not to make difficult environmental science look simple. It is to make the complexity readable, executable, extensible, and inspectable without losing the ability to verify what lies underneath.

8 Data, code, and release status

Source code and documentation are available at `https://github.com/CU-ESIIL/cubedynamics` and `https://cu-esiil.github.io/cubedynamics/` under the MIT License. The latest public artifact reviewed here is release candidate 0.1.0rc3, uploaded to PyPI on September 2, 2026 from source commit `f777eb07d3ada8bd407b027f560727c90f6d3731` (Tuff, 2026b). The Working Lands repository evidence was separately reviewed at commit `862a80aed8a2781b40e6e5293fd6cfbcba887aa4`; the SI records its fixture path and SHA-256 checksum. The fixture is not yet included in the submission archive, and its redistribution terms should be confirmed in the final data-availability record. Later development evidence remains mutable and is not treated as part of rc3 or as archival submission evidence. No final-release DOI or unified submission archive has been assigned. A

final evidence package should bind the manuscript, exact software revision, environment, fixtures, licenses, generated figures, machine-readable reports, and skip reasons. This follows geoscience provenance guidance and the research-compendium model of packaging data, code, documentation, and environment together, as well as software-citation and FAIR research-software principles (Gil et al., 2016; Marwick et al., 2018; Smith et al., 2016; Barker et al., 2022).

Generative artificial intelligence transparency statement

The author used OpenAI Codex, accessed through September 26, 2026, for LaTeX diagnostics, repository and documentation checks, structural and stylistic editing, literature review, and conceptual figure design. The author supplied and reviewed the scientific and software content; Codex did not generate observational data, empirical environmental figures, or validation outcomes. The author verified retained claims and citations and accepts responsibility for the manuscript.

Author contributions

Ty Tuff: Conceptualization, Methodology, Software, Validation, Data curation, Visualization, Writing—original draft, Writing—review and editing, and Project administration.

Competing interests

The author declares no competing interests.

Acknowledgements

The author acknowledges the maintainers of the scientific Python libraries on which CubeDynamics depends and the organizations that provide the environmental data products used in its validation examples.

References

John T. Abatzoglou. Development of gridded surface meteorological data for ecological applications and modelling. *International Journal of Climatology*, 33(1):121–131, 2013. doi: 10.1002/joc.3413. URL https://doi.org/10.1002/joc.3413.

Hannah Augustin, Martin Sudmanns, Dirk Tiede, Stefan Lang, and Andrea Baraldi. Semantic earth observation data cubes. *Data*, 4(3):102, 2019. doi: 10.3390/data4030102. URL https://doi.org/10.3390/data4030102.

Bartosz Balis, Michal Orzechowski, Piotr Kica, Michal Dygas, and Michal Kuszewski. From research question to scientific workflow: Leveraging agentic AI for science automation. arXiv preprint arXiv:2604.21910, 2026. URL https://doi.org/10.48550/arXiv.2604.21910.

Michelle Barker, Neil P. Chue Hong, Daniel S. Katz, Anna-Lena Lamprecht, Carlos Martinez-Ortiz, Fotis E. Psomopoulos, Jennifer Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, and Tom Honeyman. Introducing the FAIR principles for research software. *Scientific Data*, 9:622, 2022. doi: 10.1038/s41597-022-01710-x. URL https://doi.org/10.1038/s41597-022-01710-x.

Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering*, 41(5):507–525, 2015. doi: 10.1109/TSE.2014.2372785. URL https://doi.org/10.1109/TSE.2014.2372785.

Peter Baumann. A general conceptual framework for multi-dimensional spatio-temporal data sets. *Environmental Modelling & Software*, 143:105096, 2021. doi: 10.1016/j.envsoft.2021.105096. URL https://doi.org/10.1016/j.envsoft.2021.105096.

Peter Baumann. On the analysis-readiness of spatio-temporal earth data and suggestions for its enhancement. *Environmental Modelling & Software*, 176:106017, 2024. doi: 10.1016/j.envsoft.2024.106017. URL https://doi.org/10.1016/j.envsoft.2024.106017.

Peter Baumann, Dimitar Misev, Vlad Merticariu, and Bang Pham Huu. Array databases: Concepts, standards, implementations. *Journal of Big Data*, 8:28, 2021. doi: 10.1186/s40537-020-00399-2. URL https://doi.org/10.1186/s40537-020-00399-2.

Richard A. Becker and John M. Chambers. Auditing of data analyses. *SIAM Journal on Scientific and Statistical Computing*, 9(4):747–760, 1988. doi: 10.1137/0909049. URL https://doi.org/10.1137/0909049.

Pascal Bourgault, David Huard, Trevor James Smith, Travis Logan, Abel Aoun, Juliette Lavoie, Éric Dupuis, Gabriel Rondeau-Genesse, Raquel Alegre, Clair Barnes, Alexis Beaupré Laperrière, Sébastien Biner, David Caron, Carsten Ehbrecht, Jeremy Fyke, Tom Keel, Marie-Pier Labonté, Ludwig Lierhammer, Jwen-Fai Low, Jamie Quinn, Philippe Roy, Dougie Squire, Ag Stephens, Maliko Tanguy, and Christopher Whelan. xclim: xarray-based climate data analytics. *Journal of Open Source Software*, 8(85):5415, 2023. doi: 10.21105/joss.05415. URL https://doi.org/10.21105/joss.05415.

Shawn Bowers. Scientific workflow, provenance, and data modeling challenges and approaches. *Journal on Data Semantics*, 1:19–30, 2012. doi: 10.1007/s13740-012-0004-y. URL https://doi.org/10.1007/s13740-012-0004-y.

Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and where: A characterization of data provenance. In *Database Theory—ICDT 2001*, volume 1973 of *Lecture Notes in Computer Science*, pages 316–330. Springer, 2001. doi: 10.1007/3-540-44503-X_20. URL https://doi.org/10.1007/3-540-44503-X_20.

Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Visualization 2000*, pages 69–75, 2000. doi: 10.1109/INFVIS.2000.885092. URL https://doi.org/10.1109/INFVIS.2000.885092.

Koen Claessen and John Hughes. QuickCheck: A lightweight tool for random testing of Haskell programs. In *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming*, pages 268–279, 2000. doi: 10.1145/351240.351266. URL https://doi.org/10.1145/351240.351266.

clisops contributors. clisops: Climate simulation operations. Computer software and documentation, 2026. URL https://clisops.readthedocs.io/.

Sarah Cohen-Boulakia, Khalid Belhajjame, Olivier Collin, Jacek Chopard, Christine Froidevaux, Alban Gaignard, Konrad Hinsen, Pierre Larmande, Yvan Le Bras, Frédéric Lemoine, Fabien Mareuil, Hervé Ménager, Christophe Pradal, and Christophe Blanchet. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. *Future Generation Computer Systems*, 75:284–298, 2017. doi: 10.1016/j.future.2017.01.012. URL https://doi.org/10.1016/j.future.2017.01.012.

Charly Coussot, Isabelle Braud, Véronique Chaffard, Brice Boudevillain, and Sylvie Galle. Implementing a new research data alliance recommendation, the I-ADOPT framework, for the naming of environmental variables of continental surfaces. *Earth Science Informatics*, 17(5): 4261–4277, 2024. doi: 10.1007/s12145-024-01373-9. URL https://doi.org/10.1007/s12145-024-01373-9.

Maurice G. Cox and Peter M. Harris. Design and use of reference data sets for testing scientific software. *Analytica Chimica Acta*, 380(2–3):339–351, 1999. doi: 10.1016/S0003-2670(98)00481-4. URL https://doi.org/10.1016/S0003-2670(98)00481-4.

Michael R. Crusoe, Sanne Abeln, Alexandru Iosup, Peter Amstutz, John Chilton, Nebojša Tijanić, Hervé Ménager, Stian Soiland-Reyes, Bogdan Gavrilović, Carole Goble, and CWL Community. Methods included: Standardizing computational reuse and portability with the common workflow language. *Communications of the ACM*, 65(6):54–63, 2022. doi: 10.1145/3486897. URL https://doi.org/10.1145/3486897.

Christopher Daly, Michael Halbleib, Joseph I. Smith, Wayne P. Gibson, Matthew K. Doggett, George H. Taylor, Jan Curtis, and Phillip P. Pasteris. Physiographically sensitive mapping of climatological temperature and precipitation across the conterminous United States. *International Journal of Climatology*, 28(15):2031–2064, 2008. doi: 10.1002/joc.1688. URL https://doi.org/10.1002/joc.1688.

Susan B. Davidson and Juliana Freire. Provenance and scientific workflows: Challenges and opportunities. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1345–1350, 2008. doi: 10.1145/1376616.1376772. URL https://doi.org/10.1145/1376616.1376772.

Jennifer L. Dungan, Joe N. Perry, Mark R. T. Dale, Pierre Legendre, Sandra Citron-Pousty, Marie-Josée Fortin, Agnieszka Jakomulska, Maria N. Miriti, and Michael S. Rosenberg. A balanced view of scale in spatial statistical analysis. *Ecography*, 25(5):626–640, 2002. doi: 10.1034/j.1600-0587.2002.250510.x. URL https://doi.org/10.1034/j.1600-0587.2002.250510.x.

John L. Dwyer, David P. Roy, Brian Sauer, Calli B. Jenkerson, Hankui K. Zhang, and Leo Lymburner. Analysis ready data: Enabling analysis of the landsat archive. *Remote Sensing*, 10 (9):1363, 2018. doi: 10.3390/rs10091363. URL https://doi.org/10.3390/rs10091363.

Paul N. Edwards, Matthew S. Mayernik, Archer L. Batcheller, Geoffrey C. Bowker, and Christine L. Borgman. Science friction: Data, metadata, and collaboration. *Social Studies of Science*, 41(5): 667–690, 2011. doi: 10.1177/0306312711413314. URL https://doi.org/10.1177/0306312711413314.

Bakinam T. Essawy, Jonathan L. Goodall, Daniel Voce, Mohamed M. Morsy, Jeffrey M. Sadler, Young Don Choi, David G. Tarboton, and Tanu Malik. A taxonomy for reproducible and replicable research in environmental modelling. *Environmental Modelling & Software*, 134:104753, 2020. doi: 10.1016/j.envsoft.2020.104753. URL https://doi.org/10.1016/j.envsoft.2020.104753.

Ben Fry. *Visualizing Data: Exploring and Explaining Data with the Processing Environment*. O'Reilly Media, 2007. ISBN 978-0-596-51455-6. URL https://www.oreilly.com/library/view/visualizing-data/9780596514556/.

Benjamin Jotham Fry. *Computational Information Design*. PhD thesis, Massachusetts Institute of Technology, 2004. URL https://www.benfry.com/phd/.

Yolanda Gil, Varun Ratnakar, Jihie Kim, Pedro Antonio Gonzalez-Calero, Paul Groth, Joshua Moody, and Ewa Deelman. Wings: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems*, 26(1):62–72, 2011. doi: 10.1109/MIS.2010.9. URL https://doi.org/10.1109/MIS.2010.9.

Yolanda Gil, Cédric H. David, Ibrahim Demir, Bakinam T. Essawy, Robinson W. Fulweiler, Jonathan L. Goodall, Leif Karlstrom, Huikyo Lee, Heath J. Mills, Ji-Hyun Oh, Suzanne A. Pierce, Allen Pope, Mimi W. Tzeng, Sandra R. Villamizar, and Xuan Yu. Toward the geoscience paper of the future: Best practices for documenting and sharing research from data to software to provenance. *Earth and Space Science*, 3(10):388–415, 2016. doi: 10.1002/2015EA000136. URL https://doi.org/10.1002/2015EA000136.

Gregory Giuliani, Joan Masó, Paolo Mazzetti, Stefano Nativi, and Alaitz Zabala. Paving the way to increased interoperability of earth observations data cubes. *Data*, 4(3):113, 2019. doi: 10.3390/data4030113. URL https://doi.org/10.3390/data4030113.

Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. What does research reproducibility mean? *Science Translational Medicine*, 8(341):341ps12, 2016. doi: 10.1126/scitranslmed.aaf5027. URL https://doi.org/10.1126/scitranslmed.aaf5027.

Carol A. Gotway and Linda J. Young. Combining incompatible spatial data. *Journal of the American Statistical Association*, 97(458):632–648, 2002. doi: 10.1198/016214502760047140. URL https://doi.org/10.1198/016214502760047140.

Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow, and Hamid Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29–53, 1997. doi: 10.1023/A:1009726021843. URL https://doi.org/10.1023/A:1009726021843.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A. Merrill, Jeffrey Heer, and Tim Althoff. BLADE: Benchmarking language model agents for data-driven science. In *Findings of the Association for Computational Linguistics: EMNLP 2024*,

pages 13936–13971. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.815. URL https://doi.org/10.18653/v1/2024.findings-emnlp.815.

Armin Haller, Krzysztof Janowicz, Simon J. D. Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10(1):9–32, 2019. doi: 10.3233/SW-180320. URL https://doi.org/10.3233/SW-180320.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

David Hassell, Jonathan Gregory, Jon Blower, Bryan N. Lawrence, and Karl E. Taylor. A data model of the climate and forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1). *Geoscientific Model Development*, 10:4619–4646, 2017. doi: 10.5194/gmd-10-4619-2017. URL https://doi.org/10.5194/gmd-10-4619-2017.

Daniel Hook and Diane Kelly. Testing for trustworthiness in scientific software. In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, pages 59–64. IEEE, 2009. doi: 10.1109/SECSE.2009.5069163. URL https://doi.org/10.1109/SECSE.2009.5069163.

Stephan Hoyer and Joe Hamman. xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, 5(1):10, 2017. doi: 10.5334/jors.148. URL https://doi.org/10.5334/jors.148.

Rob J. Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365, 1996. doi: 10.1080/00031305.1996.10473566. URL https://doi.org/10.1080/00031305.1996.10473566.

Krzysztof Janowicz, Armin Haller, Simon J. D. Cox, Danh Le Phuoc, and Maxime Lefrançois. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, 56:1–10, 2019. doi: 10.1016/j.websem.2018.06.003. URL https://doi.org/10.1016/j.websem.2018.06.003.

Upulee Kanewala and James M. Bieman. Testing scientific software: A systematic literature review. *Information and Software Technology*, 56(10):1219–1232, 2014. doi: 10.1016/j.infsof.2014.05.006. URL https://doi.org/10.1016/j.infsof.2014.05.006.

Donald E. Knuth. Literate programming. *The Computer Journal*, 27(2):97–111, 1984. doi: 10.1093/comjnl/27.2.97. URL https://doi.org/10.1093/comjnl/27.2.97.

Johannes F. Kruiger, Vedran Kasalica, Rogier Meerlo, Anna-Lena Lamprecht, Enkhbold Nyamsuren, and Simon Scheider. Loose programming of GIS workflows with geo-analytical concepts. *Transactions in GIS*, 25(1):424–449, 2021. doi: 10.1111/tgis.12692. URL https://doi.org/10.1111/tgis.12692.

Jeffrey T. Leek and Roger D. Peng. Reproducible research can still be wrong: Adopting a prevention approach. *Proceedings of the National Academy of Sciences*, 112(6):1645–1646, 2015. doi: 10.1073/pnas.1421412111. URL https://doi.org/10.1073/pnas.1421412111.

Simone Leo, Michael R. Crusoe, Laura Rodríguez-Navas, Raúl Sirvent, Alexander Kanitz, Paul De Geest, Rudolf Wittner, Luca Pireddu, Daniel Garijo, José M. Fernández, Iacopo Colonnelli, Matej Gallo, Tazro Ohta, Hirotaka Suetake, Salvador Capella-Gutierrez, Renske de Wit, Bruno P. Kinoshita, and Stian Soiland-Reyes. Recording provenance of workflow runs with RO-Crate. *PLOS ONE*, 19(9):e0309210, 2024. doi: 10.1371/journal.pone.0309210. URL https://doi.org/10.1371/journal.pone.0309210.

Michael Leonard, Seth Westra, Aloke Phatak, Martin Lambert, Bart van den Hurk, Kathleen McInnes, James Risbey, Sandra Schuster, Doerte Jakob, and Mark Stafford-Smith. A compound event framework for understanding extreme impacts. *Wiley Interdisciplinary Reviews: Climate Change*, 5(1):113–128, 2014. doi: 10.1002/wcc.252. URL https://doi.org/10.1002/wcc.252.

Xuanyi Lin, Michelle Simon, and Nan Niu. Exploratory metamorphic testing for scientific software. *Computing in Science & Engineering*, 22(2):78–87, 2020. doi: 10.1109/MCSE.2018.2880577. URL https://doi.org/10.1109/MCSE.2018.2880577.

Marianne Menglin Liu, Daniel Garcia, Fjona Parllaku, Vikas Upadhyay, Syed Fahad Allam Shah, and Dan Roth. ToolScope: Enhancing LLM agent tool use through tool merging and context-aware filtering. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34095–34119. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1573. URL https://doi.org/10.18653/v1/2026.acl-long.1573.

Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, and Yang Zhao. Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065, 2006. doi: 10.1002/cpe.994. URL https://doi.org/10.1002/cpe.994.

Joshua Madin, Shawn Bowers, Mark Schildhauer, Sergeui Krivov, Deana Pennington, and Ferdinando Villa. An ontology for describing and synthesizing ecological observation data. *Ecological Informatics*, 2(3):279–296, 2007. doi: 10.1016/j.ecoinf.2007.05.004. URL https://doi.org/10.1016/j.ecoinf.2007.05.004.

Miguel D. Mahecha, Fabian Gans, Gunnar Brandt, Rune Christiansen, Sarah E. Cornell, Normann Fomferra, Guido Kraemer, Jonas Peters, Paul Bodesheim, Gustau Camps-Valls, Jonathan F. Donges, Wouter Dorigo, Lina M. Estupinan-Suarez, Victor H. Gutierrez-Velez, Martin Gutwin, Martin Jung, Maria C. Londoño, Diego G. Miralles, Phillip Papastefanou, and Markus Reichstein. Earth system data cubes unravel global multivariate dynamics. *Earth System Dynamics*, 11: 201–234, 2020. doi: 10.5194/esd-11-201-2020. URL https://doi.org/10.5194/esd-11-201-2020.

Ben Marwick, Carl Boettiger, and Lincoln Mullen. Packaging data analytical work reproducibly using R (and friends). *The American Statistician*, 72(1):80–88, 2018. doi: 10.1080/00031305.2017.1375986. URL https://doi.org/10.1080/00031305.2017.1375986.

Jeremy Mennis. Multidimensional map algebra: Design and implementation of a spatio-temporal GIS processing language. *Transactions in GIS*, 14(1):1–21, 2010. doi:

10.1111/j.1467-9671.2009.01179.x. URL https://doi.org/10.1111/j.1467-9671.2009.01179.x.

Jeremy Mennis, Roland Viger, and C. Dana Tomlin. Cubic map algebra functions for spatio-temporal analysis. *Cartography and Geographic Information Science*, 32(1):17–32, 2005. doi: 10.1559/1523040053270765. URL https://doi.org/10.1559/1523040053270765.

Tomasz Miksa and Andreas Rauber. Using ontologies for verification and validation of workflow-based experiments. *Journal of Web Semantics*, 43:25–45, 2017. doi: 10.1016/j.websem.2017.01.002. URL https://doi.org/10.1016/j.websem.2017.01.002.

David Montero, Guido Kraemer, Anca Anghelea, César Aybar, Gunnar Brandt, Gustau Camps-Valls, Felix Cremer, Ida Flik, Fabian Gans, Sarah Habershon, Chaonan Ji, Teja Kattenborn, Laura Martínez-Ferrer, Francesco Martinuzzi, Martin Reinhardt, Maximilian Söchting, Khalil Teber, and Miguel D. Mahecha. Earth system data cubes: Avenues for advancing earth system research. *Environmental Data Science*, 3:e27, 2024. doi: 10.1017/eds.2024.22. URL https://doi.org/10.1017/eds.2024.22.

Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric G. Stephan, and Jan Van den Bussche. The open provenance model core specification (v1.1). *Future Generation Computer Systems*, 27(6):743–756, 2011. doi: 10.1016/j.future.2010.07.005. URL https://doi.org/10.1016/j.future.2010.07.005.

National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*. National Academies Press, Washington, DC, 2019. doi: 10.17226/25303. URL https://doi.org/10.17226/25303.

William L. Oberkampf and Timothy G. Trucano. Verification and validation in computational fluid dynamics. *Progress in Aerospace Sciences*, 38(3):209–272, 2002. doi: 10.1016/S0376-0421(02)00005-2. URL https://doi.org/10.1016/S0376-0421(02)00005-2.

Roger D. Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011. doi: 10.1126/science.1213847. URL https://doi.org/10.1126/science.1213847.

João Felipe Pimentel, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. A large-scale study about quality and reproducibility of jupyter notebooks. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories*, pages 507–517, 2019. doi: 10.1109/MSR.2019.00077. URL https://doi.org/10.1109/MSR.2019.00077.

Jens Christian Refsgaard, Jeroen P. van der Sluijs, Anker Lajer Højberg, and Peter A. Vanrolleghem. Uncertainty in the environmental modelling process—a framework and guidance. *Environmental Modelling & Software*, 22(11):1543–1556, 2007. doi: 10.1016/j.envsoft.2007.02.004. URL https://doi.org/10.1016/j.envsoft.2007.02.004.

Matthew Rocklin. Dask: Parallel computation with blocked algorithms and task scheduling. In *Proceedings of the 14th Python in Science Conference*, pages 126–132, 2015. doi: 10.25080/Majora-7b98e3ed-013. URL https://doi.org/10.25080/Majora-7b98e3ed-013.

Adam Rule, Aurélien Tabard, and James D. Hollan. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018. doi: 10.1145/3173574.3173606. URL https://doi.org/10.1145/3173574.3173606.

Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, and Eivind Hovig. Ten simple rules for reproducible computational research. *PLoS Computational Biology*, 9(10):e1003285, 2013. doi: 10.1371/journal.pcbi.1003285. URL https://doi.org/10.1371/journal.pcbi.1003285.

Michael Schramm, Edzer Pebesma, Milutin Milenković, Luca Foresta, Jeroen Dries, Alexander Jacob, Wolfgang Wagner, Matthias Mohr, Markus Neteler, Matej Kadunc, Tomasz Miksa, Pieter Kempeneers, Jan Verbesselt, Bernd Gößwein, Christian Navacchi, Sander Lippens, and Johannes Reiche. The openEO API—harmonising the use of earth observation cloud services using virtual data cube functionalities. *Remote Sensing*, 13(6):1125, 2021. doi: 10.3390/rs13061125. URL https://doi.org/10.3390/rs13061125.

Sergio Segura, Gordon Fraser, Ana B. Sánchez, and Antonio Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on Software Engineering*, 42(9):805–824, 2016. doi: 10.1109/TSE.2016.2532875. URL https://doi.org/10.1109/TSE.2016.2532875.

Arfon M. Smith, Daniel S. Katz, Kyle E. Niemeyer, and FORCE11 Software Citation Working Group. Software citation principles. *PeerJ Computer Science*, 2:e86, 2016. doi: 10.7717/peerj-cs.86. URL https://doi.org/10.7717/peerj-cs.86.

Stian Soiland-Reyes, Peter Sefton, Mercè Crosas, Leyla Jael Castro, Frederik Coppens, José M. Fernández, Daniel Garijo, Björn Grüning, Marco La Rosa, Simone Leo, Eoghan Ó Carragáin, Marc Portier, Ana Trisovic, RO-Crate Community, Paul Groth, and Carole Goble. Packaging research artefacts with RO-Crate. *Data Science*, 5(2):97–138, 2022. doi: 10.3233/DS-210053. URL https://doi.org/10.3233/DS-210053.

Sara Stoudt, Váleri N. Vásquez, and Ciera C. Martinez. Principles for data analysis workflows. *PLoS Computational Biology*, 17(3):e1008770, 2021. doi: 10.1371/journal.pcbi.1008770. URL https://doi.org/10.1371/journal.pcbi.1008770.

C. Dana Tomlin. *GIS and Cartographic Modeling*. Esri Press, Redlands, California, 2012. ISBN 9781589483095. URL https://www.esri.com/en-us/esri-press/browse/gis-and-cartographic-modeling. Revised edition of the 1990 work that established map algebra.

Ty Tuff. CubeDynamics: A composable grammar for spatiotemporal science. Computer software and accompanying documentation, 2026a. URL https://github.com/CU-ESIIL/cubedynamics. Development repository and documentation; accessed September 25, 2026.

Ty Tuff. CubeDynamics 0.1.0rc3. Public release candidate on PyPI, 2026b. URL https://pypi.org/project/cubedynamics/0.1.0rc3/. Released September 2, 2026; wheel and source distribution published through Trusted Publishing; attested source commit f777eb07d3ada8bd407b027f560727c90f6d3731.

Ty Tuff. CubeDynamics rc1 comprehensive validation triage. Project validation report, 2026c. URL https://github.com/CU-ESIIL/cubedynamics/blob/main/docs/project/rc1_validation_triage.md. Accessed September 25, 2026.

Ty Tuff. CubeDynamics prompt log: semantic, release, and validation record. Project validation record, 2026d. URL https://github.com/CU-ESIIL/cubedynamics/blob/main/PROMPT_LOG.md. Accessed September 25, 2026.

Lucas van der Meer, Martin Sudmanns, Hannah Augustin, Andrea Baraldi, and Dirk Tiede. Semantic querying in earth observation data cubes. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XLVIII-4/W1-2022, pages 503–510, 2022. doi: 10.5194/isprs-archives-XLVIII-4-W1-2022-503-2022. URL https://doi.org/10.5194/isprs-archives-XLVIII-4-W1-2022-503-2022.

Paul H. Whitfield. Why the provenance of data matters: Assessing fitness for purpose for environmental data. *Canadian Water Resources Journal*, 37(1):23–36, 2012. doi: 10.4296/cwrj3701866. URL https://doi.org/10.4296/cwrj3701866.

Hadley Wickham. A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28, 2010. doi: 10.1198/jcgs.2009.07098. URL https://doi.org/10.1198/jcgs.2009.07098.

Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10):1–23, 2014. doi: 10.18637/jss.v059.i10. URL https://doi.org/10.18637/jss.v059.i10.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the Tidyverse. *Journal of Open Source Software*, 4 (43):1686, 2019. doi: 10.21105/joss.01686. URL https://doi.org/10.21105/joss.01686.

Leland Wilkinson. *The Grammar of Graphics*. Springer, 2 edition, 2005. doi: 10.1007/0-387-28695-0. URL https://doi.org/10.1007/0-387-28695-0.

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3: 160018, 2016. doi: 10.1038/sdata.2016.18. URL https://doi.org/10.1038/sdata.2016.18.

Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, and Tracy K. Teal. Good enough practices in scientific computing. *PLoS Computational Biology*, 13(6):e1005510, 2017. doi: 10.1371/journal.pcbi.1005510. URL https://doi.org/10.1371/journal.pcbi.1005510.

World Wide Web Consortium. PROV-O: The PROV ontology. W3C Recommendation, 2013. URL https://www.w3.org/TR/prov-o/.

Jakob Zscheischler, Olivia Martius, Seth Westra, Emanuele Bevacqua, Colin Raymond, Radley M. Horton, Bart van den Hurk, Amir AghaKouchak, Aglaé Jézéquel, Miguel D. Mahecha, Douglas Maraun, Alexandre M. Ramos, Nina N. Ridder, Wim Thiery, and Edoardo Vignotto. A typology

of compound weather and climate events. *Nature Reviews Earth & Environment*, 1:333–347, 2020. doi: 10.1038/s43017-020-0060-z. URL https://doi.org/10.1038/s43017-020-0060-z.