

Supplementary Information for *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](#)

Repository evidence accessed September 25, 2026

Abstract

This Supplementary Information documents CUBEDYNAMICS contracts, intermediate-state records, data sources, verification evidence, executable examples, and software-revision boundaries. It distinguishes public release candidate 0.1.0rc3 from later evidence accessed September 25, 2026, and separates maintained behavior from compatibility interfaces, reserved names, source candidates, and bounded post-release analyses.

Contents

S1	Purpose, scope, and version boundary	3
S2	Project origin, users, and contribution model	3
S3	Related numerical, workflow, and domain systems	4
S4	The inspectable grammar as a layered architecture	5
S4.1	Glossary and scope terms	6
S5	Semantic state and trace	7
S5.1	State vocabulary	7
S5.2	Trace boundary	8
S5.3	Branch and custom-callable boundaries	9
S6	Authored order and semantic constraints	9
S6.1	Order categories	9
S6.2	Conditional equivalence	10
S6.3	Event scope is explicit	10
S7	Condition, reduction, and metadata contracts	10
S7.1	Canonical condition representation	10
S7.2	Comparator and unresolved missing-value semantics	11

S7.3	Core verb contract matrix	11
S7.4	Units after transformations	13
S8	Alignment, interoperability, and visualization	13
S8.1	Alignment is part of the question	13
S8.2	Temporal labels are not observation support	13
S8.3	Ordinary scientific Python objects	14
S8.4	Plotting dispatch	14
S9	Source-qualified noun inventory	14
S9.1	Production-catalog nouns	14
S9.2	Installed bounded source candidates	15
S10	Source QA and evidence layers	16
S10.1	Evidence has distinct jobs	16
S10.2	Phase 1 production-source evidence	16
S10.3	Live-source status	17
S11	Executable vignette inventory	17
S12	Working Lands protocol and evidence boundary	17
S12.1	Question and observations	17
S12.2	Analytical statement	18
S12.3	Acceptance checks	20
S13	Verification strategy and RC1 triage	20
S13.1	Contract-compatible execution and informative rejection	20
S13.2	Finding classification	20
S13.3	Revision-specific evidence layers	20
S14	Status of secondary extensions and the wider namespace	22
S15	Reproduction and evidence commands	23
S16	Public rc3 and the resolved rc1 first-use findings	24
S17	Claim-to-evidence matrix	25
S18	Limitations and archival requirements	27

S1 Purpose, scope, and version boundary

This supplement provides the repository-level detail beneath the manuscript’s argument. Table S1 establishes the version boundary used throughout; subsequent sections document contracts, source records, examples, and validation at their stated scopes.

Four software states must remain distinct:

Table S1: Software states referenced by the manuscript and supplement.

State	What it establishes	What it does not establish
Historical 0.1.0rc1 artifact	The first public prerelease and the outside-user findings that motivated hardening	Inclusion of later semantic, packaging, documentation, or serialization repairs
Public 0.1.0rc3 artifact	An installable package and source archive were published on PyPI on September 2 from source commit <code>f777eb07d3ad</code> ; first-use repairs, reporting of the imported software revision, explicit time-interval checks, and defined event operations are included	Final 0.1.0 status or inclusion of later changes on <code>main</code>
Post-rc3 repository state	Later synchrony analysis limited to stated data and methods, associated diagnostics, and expanded validation evidence	Automatic inclusion in rc3, arbitrary many-dataset synthesis, national scalability, regime classification, or a stable release
Future final release	A possible archived and citable stable release after remaining scholarly and release gates pass	No final release DOI existed at the evidence date

The public rc3 artifact is the immutable release boundary; later repository evidence remains dated and mutable until archived at an exact commit. Developmental or experimental behavior is labeled directly rather than inferred from a roadmap or placeholder name (Tuff, 2026a,g,b,c).

S2 Project origin, users, and contribution model

CUBEDYNAMICS developed across climate, vegetation, fire, terrain, road, streamflow, synchrony, teaching, and decision projects. Their shared need was a small layer above ordinary scientific Python that named environmental inputs and transformations while retaining an inspectable result and evidence record. Table S2 summarizes the intended users and contribution routes; the package does not replace numerical arrays, storage systems, workflow schedulers, or scientific judgment.

Contribution is separated by scope. A project-specific transformation can remain an ordinary Python function and does not require a new data class or plugin registration. A project-specific data name should call a trusted loader and preserve dimensions, units, coordinate reference system, source identity, and records of origin and processing. Promotion into the shared vocabulary requires a stronger package-level record because public names and input and output rules will be reused outside the originating project. The maintained contribution path therefore joins implementation with direct and piped tests, tests of expected rejection, reviewed source data where applicable, generated reference material, and an executable example (Tuff, 2026a,i).

Table S2: User and contributor pathways in the current project architecture.

Role	Immediate use	Contribution pathway
Environmental researcher	Place a readable inspected sequence around existing xarray work	Report scientific contracts and boundaries through a vignette or project verb
Research software engineer	Maintain reusable workflows, tests, source and processing records, and release evidence	Improve core input and output rules, source adapters, quality assurance, documentation, or packaging
Source maintainer	Expose one vetted product through a scientific noun	Supply provider identity, standardized metadata, checked example datasets, quality assurance, and serving history
Domain-method developer	Keep project assumptions in a named operation without replacing the core	Publish ordinary Python functions, direct and piped tests, and interpretation guidance
Reviewer or collaborator	Descend from expression to the current result description, operation history, source, and evidence	Evaluate whether the recorded choices support the stated question

S3 Related numerical, workflow, and domain systems

The numerical and geospatial infrastructure beneath CUBEDYNAMICS is already mature. NumPy supplies general numerical arrays (Harris et al., 2020); xarray adds named dimensions, coordinate labels, and Dataset objects (Hoyer and Hamman, 2017); and Dask supports delayed and parallel computation (Rocklin, 2015). Rasterio and rioxarray handle gridded raster data (Rasterio contributors, 2013–present; Snow et al., 2026), while GeoPandas and Shapely represent vector features and geometry operations (Fleischmann et al., 2026; Shapely contributors, 2025). The SpatioTemporal Asset Catalog standard describes discoverable spatial and temporal assets (SpatioTemporal Asset Catalog community, 2024). Relational cubes originated as multidimensional aggregation operators (Gray et al., 1997), while Earth-observation cube systems organize sampled fields for multidimensional analysis (Giuliani et al., 2017; Lu et al., 2018; Appel and Pebesma, 2019). Analysis-ready and semantic-cube research further addresses preprocessing, comparability, and the meanings attached to cube contents (Dwyer et al., 2018; Augustin et al., 2019; van der Meer et al., 2022; Baumann, 2024).

Map algebra organized raster transformations by local, neighborhood, zonal, and global scope, while cubic and multidimensional map algebra extended those operations across space and time (Tomlin, 2012; Mennis et al., 2005; Mennis, 2010). Array databases and coverage algebra treat multidimensional fields as first-class values for subsetting, aggregation, transformation, and fusion (Baumann, 2021; Baumann et al., 2021). These traditions are the closest precedents for a library of array verbs. CUBEDYNAMICS does not claim their algorithms as new; its narrower focus is the environmental source, semantic transition, authored order, diagnostic, and evidence record surrounding such operations.

Other systems make sequence or execution structure explicit. Kepler and Pegasus exemplify scientific-workflow systems for composing or executing analytical pipelines (Ludäscher et al., 2006; Deelman et al., 2015); the Common Workflow Language describes portable command-line workflows (Crusoe et al., 2022); and WINGS uses domain descriptions and constraints to construct compatible computational experiments (Gil et al., 2011). Reviews of scientific workflows emphasize their role in computational reproducibility while also identifying substantial preservation and interoperability challenges (Cohen-Boulakia et al., 2017). ArcGIS ModelBuilder and GeoJModelBuilder provide related graphical geoprocessing workflows (Esri, 2026; Zhang et al., 2017). These are precedents for visible sequence, portable execution, or constrained composition, not direct substitutes for a package operating inside ordinary Python.

Environmental Python users are likewise more likely to use CUBEDYNAMICS alongside domain packages than instead of them. `xclim` supplies climate indicators on `xarray` objects (Bourgault et al., 2023); `clisops` performs climate-data operations (clisops contributors, 2026); `intake-esm` catalogs and loads Earth-system-model collections (intake-esm contributors, 2026); and `ESMValTool` organizes model-evaluation recipes (Righi et al., 2020). CUBEDYNAMICS contributes the source-qualified, ordered, and inspectable interface around such numerical work rather than reimplementing those methods.

Formal semantic systems address a different layer. Earth and environmental ontologies, observation models, and variable-description frameworks represent concepts and relationships in machine-interpretable form (Raskin and Pan, 2005; Madin et al., 2007; Haller et al., 2019; Coussot et al., 2024). CUBEDYNAMICS does not implement such an ontology. Its smaller runtime vocabulary records the package’s current interpretation and leaves richer cross-system semantic alignment to complementary standards and community vocabularies.

S4 The inspectable grammar as a layered architecture

In programming-language theory, a grammar defines symbols and rules for combining them into well-formed constructions (Aho et al., 2006). CUBEDYNAMICS uses the term operationally for an environmental vocabulary and documented composition rules, not for a complete formal language. Its public expression is ordinary Python syntax (Python Software Foundation, 2026):

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

The compact expression is grounded by source records, semantic contracts, state, trace, provenance, and evidence limited to stated queries and versions (Bowers, 2012; Tuff, 2026a).

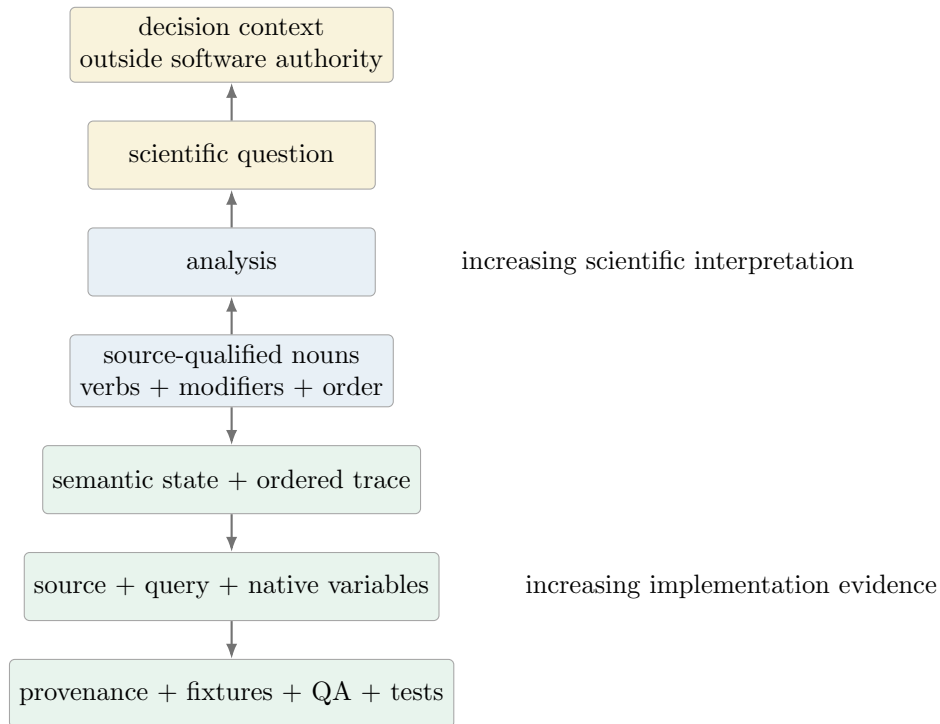


Figure S1: The grammar is an interface between scientific interpretation and supporting evidence. CUBEDYNAMICS exposes analytical choices but does not assume decision authority.

S4.1 Glossary and scope terms

The following project-specific definitions apply to the manuscript and supplement.

Environmental data product	A provider-published dataset with a particular measurement or estimation procedure, spatial and temporal coverage, resolution, revision history, and uncertainty. A product may contain direct measurements, interpolated fields, remotely retrieved quantities, modeled estimates, or derived variables.
Observation	A value supplied by an environmental data product. The term does not imply direct instrumental measurement; observation models and standards distinguish the observed property, feature of interest, procedure, and result (Madin et al., 2007; Janowicz et al., 2019; Haller et al., 2019; International Organization for Standardization, 2023).
Provider, product, and source qualifier	The <i>provider</i> distributes the data; the <i>product</i> is the particular dataset; and the <i>source qualifier</i> identifies that provider-product combination for a named environmental input. Two inputs called <code>temperature</code> therefore need not be equivalent measurements.
Metadata	Structured descriptive information attached to data, including variable names, units, dimensions, coordinates, source identity, and processing history. A coordinate reference system gives spatial coordinates a location on Earth. Metadata describe an object but do not prove that the description is correct.
Analysis and workflow	An <i>analysis</i> is the scientific sequence being interpreted. A <i>workflow</i> is the wider computational process, including preparation, execution environment, storage, and work before or after that sequence.
Grammar	A vocabulary together with rules for combining its elements. CUBEDYNAMICS uses the term operationally rather than claiming a complete formal language (Aho et al., 2006).
Expression	Python syntax that can be evaluated to produce a value (Python Software Foundation, 2026). Here an analytical expression begins with an environmental object and applies operations in written order.
Noun	A readable, stable name for a kind of environmental information, such as temperature or precipitation. It is intentionally less formal than an ontological variable description, which may separately encode the observed property, entity, context, and constraints (Coussot et al., 2024). A common noun does not make its source products interchangeable.
Verb or operation	A named transformation applied to the current object. A <i>configured verb</i> already has the parameter values needed for execution in a pipe.
Parameter	An explicit setting controlling an operation, such as dimension, threshold, comparison direction, quantile, or event duration.
Callable	A Python object that can be invoked with arguments; ordinary functions and methods are common examples (Python Software Foundation, 2026).
Pipe	The wrapper that passes the current object to each configured operation. The vertical bar <code> </code> denotes that passage. The pipe records local inspection information but is not a workflow scheduler.
Semantic	Concerning what a scientific object or operation is intended to mean, rather than only its numerical type or shape. The term does not claim implementation of a formal ontology.
Semantic contract	A documented rule stating accepted inputs, returned output, and the resulting change in the package’s interpretation of the object. The terminology is informed by software contracts built from preconditions, postconditions, and invariants,

	while extending the concern here to the declared scientific interpretation of an object (Meyer, 1992).
Semantic state	Metadata created or updated during execution that record the package’s declared or metadata-inferred interpretation and which information remains. It is an inspectable software claim, not independent scientific verification.
Trace	The ordered local record of operations, parameters, messages, and state changes completed inside one pipe. It excludes work before <code>pipe()</code> and after <code>unwrap()</code> .
Condition and summary	A <i>condition</i> is an xarray Dataset with a Boolean state variable indicating where or when a criterion is satisfied. A <i>summary</i> has reduced one or more dimensions and no longer retains all source variation.
Observation support	The spatial area or time interval represented by a value. A coordinate label such as a date or cell center may not specify that support completely (Hassell et al., 2017; Gotway and Young, 2002; Dungan et al., 2002).
Alignment	The relationship among dimensions, coordinate labels, and known observation support of objects intended for combination. Exact coordinate equality is a computational requirement here, not proof of comparability.
Provenance	Information about an object’s origin and processing history: its source, retrieval request, contributing data, and the activities producing later objects (Buneman et al., 2001; Moreau et al., 2011; World Wide Web Consortium, 2013; Davidson and Freire, 2008).
Fixture, bounded evidence, and quality assurance	A <i>fixture</i> is a fixed example dataset for repeatable offline tests. <i>Bounded evidence</i> applies only to an identified source, request, extent, period, revision, and test. <i>Quality assurance</i> is a documented set of checks with such a scope.
Vignette	An executable worked example combining identified data, code, output, interpretation, and reproduction instructions.
Guardrail	A check that reports or rejects a known mismatch or contract-incompatible operation instead of silently modifying the data or changing the question.

The design maxim follows directly from this architecture: *the shorter the analytical expression becomes, the stronger the evidence underneath it must become*. Concision is useful only if a reader can descend from a noun or verb into its source identity, parameters, state transition, trace, provenance, and validation boundary.

In this paper, *inspectability* is operational rather than perceptual. It means that the software exposes a recoverable record of the source qualifier and query, selected variables and units, explicit parameters, authored operation order, semantic states and transitions, trace messages, and bounded supporting evidence. Earlier work on auditing data analyses similarly began from records of executed statements and accessed or changed objects, using them for recreation and verification (Becker and Chambers, 1988). This definition does not establish that a source or method was scientifically appropriate, that a reader will understand the analysis more quickly, or that the result is suitable for a decision; repeatable work can still be wrong (Leek and Peng, 2015). Those questions require scientific review and, for usability claims, comparative study.

S5 Semantic state and trace

S5.1 State vocabulary

Semantic state is metadata containing the package’s declared or metadata-inferred interpretation of the current scientific object, not a replacement container hierarchy or verification of the object’s scientific

truth. Inference reads public metadata and coordinates and does not compute array values solely to infer meaning. Observation ontologies and the CF data model provide related precedents for distinguishing an observed property, result, procedure, units, and coordinate context (Madin et al., 2007; Haller et al., 2019; Hassell et al., 2017). The maintained runtime vocabulary is summarized in Table S3 (Tuff, 2026a).

Table S3: Maintained semantic-state vocabulary.

State	Interpretation
<code>observation</code>	A value whose more specific analytical role has not yet been identified
<code>continuous_field</code>	Numeric measurements distributed in space and/or time
<code>categorical_field</code>	Labels or categories distributed in space and/or time
<code>condition</code>	A true/false scientific state, optionally accompanied by meaningful magnitude and threshold fields
<code>event</code>	One or more identified intervals or occurrences
<code>feature</code>	A discrete geometry or observation feature
<code>relationship</code>	A comparison, association, overlap, or synchrony result
<code>summary</code>	A reduction that no longer retains all source variation
<code>network</code>	Connected features with graph or flow semantics

The state also records dimensions, shape, units, coordinate reference system (CRS), temporal and spatial status, time ordering, remaining temporal variation, source qualifier, and whether source provenance is present. These fields allow a verb contract to ask whether required information remains available. Absent, stale, or contradictory input metadata remain a limitation: contracts can inspect recorded metadata but cannot independently establish that a provider description is correct.

S5.2 Trace boundary

Each completed stage is appended to a trace object during pipe construction with its operation name, parameters, messages, and state transition. Database and workflow provenance models cover broader questions about source contribution, derivation, and exchange among systems (Buneman et al., 2001; Moreau et al., 2011; Davidson and Freire, 2008; World Wide Web Consortium, 2013). Guidance for computational geoscience makes a particularly useful distinction: a trace summarizes execution events, provenance describes the inputs, processing, and outputs of a completed result, and a workflow describes a reusable method (Gil et al., 2016). The CUBEDYNAMICS trace covers the inspected statement only. Preparation before `pipe` and work after `unwrap` remain outside it, so the trace complements rather than replaces environment capture, source snapshots, workflow provenance, revision control, or archival packaging (Soiland-Reyes et al., 2022; Leo et al., 2024).

```
print(analysis.explain())
print(analysis.semantic_state.as_dict())
print([step.as_dict() for step in analysis.semantic_trace])
report = analysis.validate()
result = analysis.unwrap()
```

The inspection methods do not rearrange the expression, add analysis stages, certify source observations, or compute lazy arrays merely to infer semantics. `unwrap` is a local boundary: it returns the wrapped value to ordinary Python without forcing computation or certifying the wider workflow.

Runtime identity is part of the inspection boundary. `version.info()` reports the package version, imported path, artifact kind, and Git revision when version-control metadata are available. This helps distinguish a public wheel from a later development checkout or a stale notebook kernel; it does not prove

that the wider environment or input data have been archived.

S5.3 Branch and custom-callable boundaries

Composition retains less than complete end-to-end lineage. In the Working Lands protocol, the locally warm and trace-or-no-rain branches each have their own state and trace before they are unwrapped. The final **overlap** stage records the condition labels, selected variables, and exact-alignment requirement, but its result does not embed both complete upstream traces and source records. Reconstructing the full analysis therefore requires retaining the branch records alongside the final pipe.

The **apply** stage is an explicit extension boundary. It forwards the current object to a user callable, whose code controls the output’s values, dimensions, and shape; CUBEDYNAMICS then infers semantic state from the returned public object. Pipe syntax and post-operation inspection remain available, but the grammar cannot infer or validate the callable’s scientific intent unless the project supplies an appropriate contract and metadata. The current trace does not by itself archive the callable’s source code, package version, or revision, so durable reconstruction requires those identifiers in the surrounding project provenance. Thus extension without subclassing is supported, but arbitrary callables do not automatically acquire maintained semantic guarantees.

S6 Authored order and semantic constraints

S6.1 Order categories

Order knowledge explains a sequence but never rewrites it. Semantic workflow systems provide precedent for constraining meaningful workflow composition without treating every executable chain as scientifically valid; geospatial workflow synthesis has likewise shown that richer semantic types can rule out many hard analytical errors while leaving scientific review necessary (Gil et al., 2011; Kruiger et al., 2021). The repository distinguishes the categories in Table S4 (Tuff, 2026f,g).

Table S4: Categories used to describe operation-order relationships.

Category	Meaning
REQUIRED_ORDER	A later operation requires the scientific object produced by an earlier operation
ORDER_CHANGES_MEANING	Both orders may execute but answer different questions
ORDER_REMOVES_REQUIRED_INFORMATION	An earlier operation can erase information required later
ORDER_EQUIVALENT_OR_NEAR_EQUIVALENT	Equivalence holds only under stated assumptions

The manuscript’s threshold/mean example is intentionally executable in both orders. Thresholding a field and then averaging asks for the proportion of observations satisfying the condition. Averaging and then thresholding asks whether the reduced value satisfies a newly defined condition. The current grammar preserves both statements and records that their order changes meaning.

Other sequences are rejected because information is absent or has been removed. Event detection requires a temporal condition with ordered time and remaining variation. A retained length-one time coordinate after reduction does not restore the removed variation. The grammar therefore rejects event detection after such a reduction and recommends detecting events earlier.

S6.2 Conditional equivalence

For a fixed climatology C , identical linear spatial weights, and identical masks,

$$\text{mean}_s (T(s, t) - C(s, t)) = \text{mean}_s T(s, t) - \text{mean}_s C(s, t). \quad (1)$$

The equality need not hold if climatologies, masks, interpolation, missingness, or weights differ between stages. The software records the assumptions associated with a conditional rule rather than treating a mathematical special case as universal permission to rearrange a workflow.

S6.3 Event scope is explicit

Event detection begins with a temporal condition that retains ordered variation. The resulting events are local to the indexed cells unless the analyst explicitly consolidates them. The post-rc1 event vocabulary therefore separates `detect_events`, `consolidate_events`, and `event_metrics`. Consolidation requires user-declared temporal-gap and spatial-connectivity rules; metrics retain whether they describe local-cell events or consolidated regional episodes. Seasonal gaps are not silently bridged, and a regional episode is not inferred merely because multiple cells are active (Tuff, 2026g).

S7 Condition, reduction, and metadata contracts

The word *contract* covers several kinds of software statement, which should not be conflated. In software engineering, design by contract associates operations with explicit preconditions, postconditions, and invariants (Meyer, 1992). CUBEDYNAMICS uses that precedent pragmatically: some rules are enforced by execution, while others diagnose or document changes in scientific interpretation. Table S5 identifies the status vocabulary used below. A final machine-readable contract-to-test manifest should bind each enforced or diagnosed rule to exact test identifiers at the archived submission revision.

Table S5: Status of rules described as semantic contracts.

Status	Meaning
Machine-enforced	Execution checks the precondition or postcondition and fails when it is violated
Diagnosed	Execution may continue or fail, but a maintained message exposes the relevant semantic distinction
Metadata-inferred	The package derives a state from public dimensions, coordinates, dtype, or attributes; incorrect input metadata remain possible
Documentary	The rule explains intended scientific use or evidence scope but is not independently verified by the runtime

S7.1 Canonical condition representation

The post-RC1 implementation uses one public condition representation across thresholding, quantile classification, and logical overlap. This removes an implementation-specific DataArray/Dataset branch from ordinary composition while retaining only fields with defined scientific meaning.

The key invariant is that semantic metadata must describe the result, not merely survive from the input. A condition mean is therefore a summary rather than a Boolean condition. Its state values represent the proportion of retained indexed positions for which the two-valued condition state was true. A scientist

Table S6: Canonical condition and summary contracts.

Operation	Public object	Semantic content
<code>threshold.state</code>	Condition Dataset	Boolean state ; may include meaningful magnitude and threshold
<code>quantile.state</code>	Condition Dataset	Boolean state ; empirical threshold and associated magnitude may be retained
<code>overlap</code>	Condition Dataset	Boolean state only; no fabricated magnitude or threshold
Condition $\rightarrow$ mean	Summary Dataset	Reduced floating-point state interpreted as occurrence proportion; auxiliary definition arrays are not averaged
Summary $\rightarrow$ threshold.state	New condition Dataset	A new explicit threshold defines a condition on the aggregate

who wants a reduction of magnitude or threshold arrays must request that question explicitly (Tuff, 2026f,g).

S7.2 Comparator and unresolved missing-value semantics

Threshold directions are inclusive: **above** uses greater-than-or-equal and **below** uses less-than-or-equal. The current condition contract is two-valued. When an input value is missing, the numerical comparison evaluates to false in the Boolean **state**, while a retained magnitude remains missing. A subsequent unweighted arithmetic mean of **state** therefore uses all indexed positions along the named dimension as its denominator and counts the missing-input position as a non-occurrence rather than excluding it as unknown. The resulting value must not be described as a proportion among valid observations unless the input is known to be fully finite.

This behavior can turn absence of evidence into apparent evidence of non-occurrence. Statistical missing-data theory makes the treatment of missingness part of the inferential assumptions rather than a neutral implementation detail (Rubin, 1976). The manuscript therefore treats missingness as an unresolved stable-release design decision. Candidate remedies include retaining an explicit validity mask, requiring a declared **missing=** policy, or introducing a three-valued condition. Until one is adopted, reductions of conditions containing missing inputs should be considered scientifically unsafe unless the analyst explicitly accepts false imputation and records the denominator. The package also lacks a minimum-valid-observation rule, area or exposure weighting, and a CUBEDYNAMICS-specific quantile interpolation parameter. `quantile.state` delegates the empirical quantile calculation to xarray’s default method; statistical conventions differ, so an archival analysis should record the exact method and dependency version when ties or short samples matter (Hyndman and Fan, 1996). The reviewed Working Lands fixture is reported as fully finite, so missingness does not affect that protocol’s recorded calculation.

S7.3 Core verb contract matrix

Table S7 summarizes the contracts most directly supporting the manuscript’s claims. Exact current signatures and the complete namespace belong in versioned generated API documentation rather than in this supplement.

Table S7: Core semantic contracts and their principal boundaries, part 1.

Operation	Required input	Result	Principal semantic or evidentiary boundary
anomaly	Numeric field with named dimension	Continuous field	Uses the mean along the named dimension; the comparison baseline remains an analytical choice
mean	Field, summary, or condition with named dimension	Summary	Uses skip-missing reduction for numeric fields; condition input reduces Boolean state to occurrence proportion
variance	Numeric field with named dimension	Summary	Squares known units; unknown units remain unknown
threshold_state	Numeric field or summary	Condition Dataset	Inclusive comparison; missing input becomes false in the two-valued state
quantile_state	Numeric field with ordered temporal variation	Condition Dataset	Cell-wise empirical threshold through xarray defaults; Dask execution gathers the selected time axis into one chunk while preserving spatial chunks
align_time	Two temporally indexed objects	Aligned value plus temporal report	Makes label and support choices explicit; does not resample or alter scientific values

Table S8: Core semantic contracts and their principal boundaries, part 2.

Operation	Required input	Result	Principal semantic or evidentiary boundary
overlap	Two condition Datasets	Condition Dataset	Requires exact coordinates and checks known temporal support; records composition metadata but not complete upstream branch histories
detect_events	Ordered temporal condition with remaining variation	Event result	Rejects continuous input or input whose temporal variation has been removed
consolidate_events	Local-cell event result plus declared connectivity	Regional-episode result	Temporal gap and spatial connectivity are user choices, not inferred scientific truth
event_metrics	Event result with explicit scope	Event-summary result	Retains whether metrics describe local cells or consolidated regional episodes
plot	Supported semantic object and dimensionality	Rendered view while retaining the value	Ambiguous general Datasets require explicit variable selection; optional backends remain environment dependent
apply	Any object accepted by the user callable	Callable-defined object	Output state is re-inferred, but scientific intent and shape changes are not validated by a maintained verb contract

S7.4 Units after transformations

Transformations update scientific metadata as well as numerical values. Dimensional consistency checking has long been proposed as a way to expose physically incoherent scientific computations, while unit ontologies address standardized, machine-interpretable descriptions for integration across resources (Petty, 2001; Gkoutos et al., 2012). CUBEDYNAMICS presently implements a narrower metadata contract rather than a complete unit algebra or unit ontology: variance deterministically squares known physical units, so `degC` becomes `degC^2`, `mm` becomes `mm^2`, and compound units are enclosed before squaring. Dimensionless variance remains dimensionless, while missing or explicitly unknown units are not invented. This contract preserves the source and provenance record while changing the interpretation of the statistic.

S8 Alignment, interoperability, and visualization

S8.1 Alignment is part of the question

Combining scientific operands requires explicit alignment. Logical overlap requires exact coordinate agreement, while harmless differences in dimension ordering can be normalized. Diagnostics distinguish incompatible dimension names, temporal coordinates, and spatial coordinates. The operation does not silently reproject, resample, shift, truncate, or intersect operands because those choices alter the analysis. Exact coordinates are necessary for this operation but are not evidence that the operands share observational footprints, temporal support, calendars, measurement processes, uncertainty, or scientific comparability. This boundary is consistent with the spatial change-of-support literature, which shows that apparently compatible data can still answer different questions when their scale or support differs (Gotway and Young, 2002; Dungan et al., 2002; Baumann, 2024).

The presence of an implemented `align_cube` stage does not weaken this principle. Alignment is a visible analytical operation with a chosen reference, not an implicit side effect of a later combining verb. Computational compatibility after explicit alignment still does not establish measurement equivalence.

S8.2 Temporal labels are not observation support

The current grammar distinguishes a timestamp from the interval represented by the observation. CF metadata provide related conventions for describing coordinates, bounds, and data-model relationships (Hassell et al., 2017). `TemporalSupport` records known support metadata, while `TemporalAlignmentReport` records whether labels and support are compatible. The `align_time` stage supports an explicit label-alignment mode and a stricter mode that requires exact known support. It does not resample values, shift coordinates, or invent support when metadata are absent. A support mismatch therefore remains a scientific choice to resolve, and unknown support remains visible as a review condition rather than being silently treated as agreement (Tuff, 2026a,g).

This distinction matters even for nominally daily products. PRISM and gridMET can assign the same date labels while representing different daily support conventions. Source adapters retain those conventions so coordinate equality alone cannot authorize combination. The Working Lands example avoids this cross-source issue because both branches use the same PRISM daily product family and the reviewed fixture places them on identical coordinates.

S8.3 Ordinary scientific Python objects

Nouns return ordinary xarray objects, conditions remain xarray Datasets, and `unwrap` returns the current value. NumPy- and Dask-backed execution behavior remains with the underlying scientific Python stack when an operation permits it. A lazy output does not by itself prove bounded remote access: transport behavior is validated separately from array evaluation.

S8.4 Plotting dispatch

The plotting interface selects a representation from semantic object and dimensionality rather than requiring the scientist to strip metadata or manually extract the common condition field. Current intended dispatch is summarized in Table S9.

Table S9: Semantic plotting dispatch in the post-RC1 repository state.

Input	Default interpretation
Three-dimensional spatiotemporal field	Interactive cube view
Two-dimensional spatial summary	Static map
One-dimensional temporal summary	Line plot
Condition Dataset	Canonical <code>state</code> field
Event Dataset	<code>event_active</code> where appropriate
General ambiguous Dataset	Explicit variable selection required

Lexcube, an optional interactive viewer for multidimensional data cubes, remains a visualization extra. The core architecture therefore places the semantic object before the rendering backend. The custom cube viewer and ordinary static plotting do not require Lexcube (Tuff, 2026f,g).

S9 Source-qualified noun inventory

S9.1 Production-catalog nouns

Table S10 reports the source flavors in the maintained production discovery catalog. A common noun is a common entry point, not a claim that its source flavors are observationally interchangeable (Tuff, 2026a,e).

PRISM and gridMET temperature illustrate the source-qualification rule. Table S11 preserves the detailed comparison moved from the main manuscript. The products use different construction methods and native grids (Daly et al., 2008; Abatzoglou, 2013).

Unit conversion or regridding may make these objects computationally compatible without demonstrating scientific equivalence. Observation identity and fitness for purpose still depend on the product, procedure, statistic, support, provenance, and intended analysis (Madin et al., 2007; Janowicz et al., 2019; Whitfield, 2012).

The gridMET names `wind`, `humidity`, and `radiation` are likewise umbrella entry points, not complete scientific definitions. The selected native variable, units, measurement height or aggregation where

Table S10: Implemented production-catalog nouns and source flavors.

Noun	Source flavors	Scope note
<code>temperature</code>	<code>prism</code> , <code>gridmet</code>	Source-specific statistics, units, grids, revisions, and provenance retained
<code>precipitation</code>	<code>prism</code> , <code>gridmet</code>	Daily precipitation access under source-specific contracts
<code>vpd</code>	<code>gridmet</code>	Variable-specific scientific QA remains necessary beyond catalog presence
<code>wind</code>	<code>gridmet</code>	Native interpretation and units retained
<code>humidity</code>	<code>gridmet</code>	Native interpretation and units retained
<code>radiation</code>	<code>gridmet</code>	Native interpretation and units retained
<code>surface_reflectance</code>	<code>sentinel2</code>	Spectral, spatial, acquisition, and cloud/scene constraints retained
<code>vegetation_index</code>	<code>sentinel2</code>	Derived index linked to its reflectance inputs and source record

Table S11: Selected distinctions retained beneath the shared `temperature` noun.

Property	PRISM flavor	gridMET flavor
Daily statistics	Maximum, minimum, and mean	Maximum and minimum; no native mean
Native variables and units	<code>tmax</code> , <code>tmin</code> , <code>tmean</code> ; degrees Celsius	<code>tmmx</code> , <code>tmmn</code> ; kelvin
Grid and documented span	Approximately 4 km; 1981–present	1/24°; 1979–present
Daily temporal support	Product-specific support retained in metadata	Different product-specific support retained in metadata
Revision behavior	Recent daily grids may be remodeled	Recent observations progress through maturity stages
Reviewed QA scope	Bounded temperature extract	Bounded maximum-temperature extract

applicable, query, and source revision must remain in the returned metadata and provenance. Catalog presence alone does not establish that every possible interpretation of the noun is implemented or reviewed.

S9.2 Installed bounded source candidates

The repository also contains three installed candidate projects that exercise different source shapes: streamflow, elevation, and roads. The roads project exposes two bounded providers, so Table S12 contains four source rows. In this section, *area of interest* (AOI) denotes the explicitly requested spatial extent. These projects are documented and vignetted, but remain outside production source discovery and have no production serving revisions (Tuff, 2026d).

Their source records cite the USGS Water Data for the Nation database, 3DEP accuracy assessment, and OpenStreetMap source and quality literature; Overture is tied to the tested release and schema (U.S. Geological Survey, 2026; Stoker and Miller, 2022; Haklay and Weber, 2008; Haklay, 2010). Candidate status does not imply national certification, cross-source ranking, or automatic promotion.

Table S12: Installed source candidates outside the production catalog.

Candidate	Implemented bounded scope	Explicit boundary
USGS streamflow	One station, continuous discharge, native status and units, time $\times$ station	Bounded dates, batches, observations, retries, and response bytes; not hydrologic decision validation
USGS 3DEP elevation	One fully covering CONUS 1/3 arc-second tile, native static raster	No mosaicking, silent clipping, value reprojection, or invented time axis
Overture roads	Explicit release with native road features and classes	Bounded AOI, partitions, row groups, features, attempts, and bytes
OpenStreetMap roads	Small Overpass way query with native tags and identifiers	Bounded AOI and response; not a production application backend or complete routable network

S10 Source QA and evidence layers

S10.1 Evidence has distinct jobs

Source evidence is separated because different checks answer different questions:

Table S13: Evidence layers and their claim boundaries.

Evidence layer	Supports	Does not establish
Retrieval and live health	The requested endpoint and access behavior were available at a recorded time	Continued availability or scientific correctness
Fixture and provenance	The reviewed bytes, query bounds, upstream identity, and checksum	All products, places, periods, or provider observations
Structural interpretation	Variables, dimensions, coordinates, units, CRS, masks, and schema	Observational accuracy or decision fitness
Numerical QA	Broad ranges, finiteness, ordering, and source-specific relationships	Independent ground truth or causal validity
Visual review	Detectable spatial, temporal, orientation, clipping, or rendering problems in reviewed outputs	Universal scientific validity
Semantic tests	State transitions, operation order, metadata updates, and expected rejection	Completeness of the scientific rule catalog
Publication checks	Executable notebooks, generated-reference freshness, documentation links, and rendering	User productivity or superiority to another workflow

A serving revision identifies a CubeDynamics adapter and interpretation contract, not the provider's scientific version. Schema fingerprints expose the interpreted structure without forcing value computation. Offline fixture certification and live endpoint health remain separate records.

S10.2 Phase 1 production-source evidence

The PRISM profile checks checksum and source flags, explicit CRS, increasing dates, finite nonempty arrays, broad physical bounds, coordinate overlap with the requested AOI, and the relationship that

Table S14: Bounded Phase 1 source evidence recorded by the repository.

Source evidence	Serving revision	Reviewed boundary
PRISM temperature	<code>temperature.prism@2026-08-26.1</code>	Checksum-controlled bounded temperature extract; pass with documented caveats
gridMET maximum temperature	<code>temperature.gridmet@2026-08-26.1</code>	One bounded maximum-temperature extract; annual-file retrieval remains a caveat
Sentinel-2 B04/B08 and normalized difference vegetation index (NDVI)	<code>surface_reflectance.sentinel2@2026-08-26.1</code>	One bounded South Dakota window; pixel-level cloud masking is not implemented in the baseline

minimum temperature does not exceed maximum temperature. The gridMET profile additionally checks its expected $1/24^\circ$ spacing, orientation, and kelvin range. The Sentinel-2 profile checks B04/B08 presence, ordered acquisition dates, finite pixels, 10 m spacing, native UTM coordinates, provider scaling, and NDVI bounds. Sentinel-2 mission characteristics are described by [Drusch et al. \(2012\)](#). These are bounded adapter checks, not certifications of every variable or observation ([Tuff, 2026e](#)).

S10.3 Live-source status

The live-source record accessed September 25, 2026 reports that a bounded PRISM request passed with documented caveats. Daymet is a separate credentialed live-certification target rather than a production-catalog noun or one of the installed candidates in Table S12; it is recorded as BLOCKED when the required Earthdata token is absent. Daymet’s data record includes product-specific uncertainty information, but a missing local credential remains an access boundary rather than an observational judgment ([Thornton et al., 2021](#)). Provider availability, authentication, package correctness, provenance retention, fixture validation, live health, and scientific suitability must remain distinguishable ([Tuff, 2026g](#)).

S11 Executable vignette inventory

The repository documents an offline notebook suite covering cube construction, grammar basics, state and event transitions, custom callables, lazy execution, native source shapes, and the Working Lands protocol ([Tuff, 2026i,g](#)). The exact notebook manifest is version-specific and therefore belongs with the archived validation report. Table S15 records the stable teaching roles relevant to this paper.

The vignette contract is narrative as well as computational: context, question, compact pipe, figure, interpretation, data record, and reproduction instructions remain visible. Random measurements are not silently substituted when a required observational fixture is unavailable.

S12 Working Lands protocol and evidence boundary

S12.1 Question and observations

The Working Lands vignette asks where, within a bounded central South Dakota window during July 2024, maximum temperature was at or above each grid position’s empirical 0.75 quantile while daily precipitation was at or below 0.1 mm. The manuscript calls these *locally warm* and *trace-or-no-rain* conditions rather than treating them as general definitions of heat or dryness. Both nouns use PRISM observations. These

Table S15: Roles represented in the executable vignette suite.

Vignette family	Purpose
Cube construction	Build labeled arrays from NumPy values, tidy tables, and multi-variable xarray Datasets
Grammar and transformations	Introduce source-qualified nouns, configured verbs, inspection methods, and supported reductions
States and events	Demonstrate transitions from continuous fields to conditions, events, and summaries
Extension and execution	Exercise project-owned callables and Dask-backed deferred execution within documented boundaries
Native source shapes	Retain the distinct semantics of terrain rasters, road features, and station time series
Working Lands	Compose two PRISM conditions into the bounded protocol documented in Section S12

are illustrative screening definitions rather than externally validated impact thresholds, and the area of interest was selected to keep acquisition and execution bounded rather than because it was a known impact hotspot (Tuff, 2026j).

Table S16: Repository-recorded Working Lands analysis input.

Field	Recorded value
Provider/product	PRISM Climate Group; AN81d/AN91d daily time series
AOI	$[-101.2, 43.7, -100.4, 44.3]$ in WGS84
Dates	July 1–31, 2024, daily
Variables	Daily maximum air temperature (degrees Celsius); daily precipitation total (mm)
Cube size	$31 \times 15 \times 19$ for each noun
Temperature range	21.922–40.386 °C
Precipitation range	0–40.260 mm/day
Finite cells	8,835 for each noun
Fixture	tests/fixtures/real_data/sd_working_lands_july_2024.nc
Fixture SHA-256	f9f3f0da6c621383b60d4895e661a185d3d58e7393b4f339ee91d36d83228a6a
Generated measurements	None

S12.2 Analytical statement

```

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"
    )
)

warm = warm_branch.unwrap()
trace_or_no_rain = trace_branch.unwrap()

analysis = (
    pipe(warm)
    | v.overlap(
        trace_or_no_rain,
        name="warm_and_trace_or_no_rain",
    )
    | v.mean(dim="time", keep_dim=False)
)
frequency = analysis.unwrap()["state"]
frequency_percent = frequency * 100 # presentation outside the trace

```

The two stages create canonical condition Datasets using inclusive comparators. Exact coordinates are required for overlap; the temporal mean then returns the unweighted proportion of 31 indexed days satisfying both definitions. The named branch objects preserve their separate traces after unwrapping, and multiplication by 100 remains an explicitly external presentation step. The final pipe does not embed both complete upstream histories. Table S17 identifies the records needed for reconstruction.

Table S17: Inspection records required to reconstruct the Working Lands statement. The table summarizes the documented record types; exact serialized outputs remain part of the archival evidence requirement.

Object	Recorded interpretation	Required accompanying record
warm_branch	PRISM maximum-temperature condition	Source query, quantile parameters, messages, and transition from continuous field to condition
trace_branch	PRISM precipitation condition	Source query, inclusive threshold parameters, messages, and transition from continuous field to condition
analysis	Unitless temporal occurrence-proportion summary	Exact-alignment rule, overlap definition, named reduction dimension, and final state
frequency_percent	Percentage used for plotting or reporting	Explicit post- unwrap multiplication by 100, recorded by the surrounding notebook rather than the pipe trace

The external reviewed repository record reports 6.5–22.6% co-occurrence, equivalent to 2–7 of 31 days, for the checksum identified in Table S16. “Warm” is a within-July position-wise rank computed with xarray’s default quantile method, so without ties approximately one quarter of days qualify by construction; it is not a long-term climatological anomaly (Hyndman and Fan, 1996). The output is an illustrative Boolean weather screen, not an impact, drought, causal, or risk classification (Leonard et al., 2014; Zscheischler et al., 2020). Because the binary fixture, generated map, and serialized inspection output have not yet been incorporated into the submission archive, the main paper treats this as a protocol and separately cited repository record rather than assembled empirical figure evidence.

S12.3 Acceptance checks

The fixture and notebook are rejected if source records indicate generated measurements; dimensions, dates, or units differ from the declared contract; cells are nonfinite or outside broad physical bounds; temperature and precipitation coordinates do not align exactly; required source-QA or decision figures are absent; named public operations are unavailable; direct and piped results disagree; or the documentation build and link checks fail. These checks establish the bounded analytical object and its reproducibility, not its fitness for a management decision.

The absent archival artifacts—the checksum-bound map, exact quantile method and dependency versions, marginal counts, tie diagnostics, and serialized pipe records—define the boundary of the present protocol. No presentation mock-up or statewide-looking substitute is empirical evidence.

S13 Verification strategy and RC1 triage

S13.1 Contract-compatible execution and informative rejection

Validation of a scientific grammar cannot be reduced to the fraction of arbitrary operation chains that run. Scientific software often lacks a complete test oracle, so invariants, relational checks, expected failures, and metamorphic relationships can provide evidence where one absolute expected output is unavailable (Kanewala and Bieman, 2014; Barr et al., 2015; Segura et al., 2016). Some apparent failures reveal genuine defects; others demonstrate that the grammar correctly refuses an expression that violates a maintained contract. The post-RC1 process therefore classifies findings before changing behavior (Tuff, 2026f).

Table S18: Representative behavioral checks underlying the semantic claims.

Expression or input	Recorded behavior	Interpretation
$\{0, 2, 4, 6\}^{\circ}\text{C} \rightarrow (T \leq 3) \rightarrow$ mean	0.5	Fraction of cells satisfying the condition
$\{0, 2, 4, 6\}^{\circ}\text{C} \rightarrow \text{mean} \rightarrow$ ($T \leq 3$)	TRUE	Condition evaluated on the regional mean
Boolean condition $\rightarrow \text{mean}(\text{time})$	Floating-point state only	Occurrence proportion; condition-definition arrays are not averaged
Variance of a field in degC	Units become degC²	Metadata follows the statistic
Overlap with unequal time or spatial coordinates	Rejected with axis-specific diagnostic	Alignment is explicit rather than silently inferred
Event detection after time variation is removed	Rejected despite a retained length-one coordinate	Coordinate presence is not confused with remaining variation

S13.2 Finding classification

S13.3 Revision-specific evidence layers

The earlier development log reported aggregate pass and skip totals, vignette execution, generated references, visual checks, and documentation gates (Tuff, 2026g). Those totals are deliberately omitted from the paper’s evidentiary claims until they can be reproduced at an exact archived revision and

Table S19: Selected RC1 findings and their treatment, part 1.

Finding	Classification	Maintained response
Condition mean retained condition semantics and reduced auxiliary definition fields	Semantic defect	Return a summary containing occurrence proportion on state ; retain the condition definition in metadata without averaging auxiliary arrays
Variance retained original units	Semantic defect	Square known physical units; retain dimensionless units and avoid inventing unknown units
Condition and two-dimensional summary plots did not compose	Implementation defect	Dispatch by semantic object and dimensionality
Overlap returned an inconsistent public form	Semantic/interface defect	Return the canonical condition Dataset with Boolean state only
Alignment error did not identify the mismatched axis class	Diagnostic defect	Distinguish dimension-name, temporal-coordinate, and spatial-coordinate mismatches
Mean followed by threshold was rejected although the order lesson taught it	Grammar defect	Support both orders and record that order changes meaning

Table S20: Selected RC1 findings and their treatment, part 2.

Finding	Classification	Maintained response
Exactly misaligned operands were rejected	Correct guardrail	Preserve fail-closed alignment; normalize only harmless dimension ordering
Event detection failed after temporal variation was removed	Correct guardrail	Preserve rejection even if a length-one coordinate remains
Randomly generated arbitrary verb chains failed	Test-design problem	Generate contract-compatible expressions or test expected rejection; do not make the grammar indiscriminately permissive
Fire environmental data did not overlap the event dates	Expected data mismatch	Require temporal overlap; do not substitute or extrapolate climate values
Missing optional Lexcube dependency	Optional-environment issue	Keep the integration optional and report the missing extra clearly

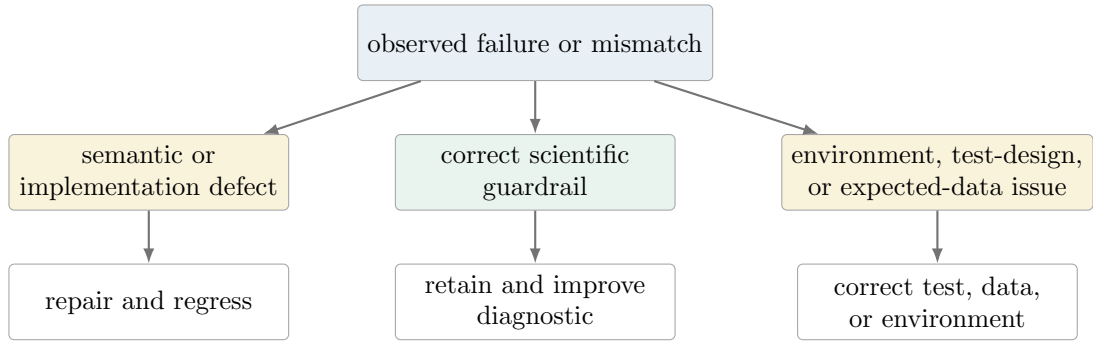


Figure S2: Validation triage distinguishes software defects from correct rejection and from limitations in the test, data, dependency, or environment.

Table S21: Revision-specific evidence boundary as of September 25, 2026.

Evidence object	Status in this manuscript
Public 0.1.0rc3 artifact	Published wheel and source distribution with attested source commit <code>f777eb07d3ad</code> (full identifier below); this establishes artifact identity, not all later development behavior
Working Lands repository record	Separately reviewed at commit <code>862a80aed8a2</code> (full identifier in the main paper); its binary fixture and generated outputs are not assembled in the submission archive
Later development validation	Internal dated records exist, but aggregate counts are not used as submission evidence because the exact commit, environment, machine-readable results, and skip reasons are not archived here
Final submission evidence bundle	Not yet available; it should bind code, environment, fixtures, generated figures, contract-to-test mapping, raw reports, and persistent identifiers

accompanied by every test identifier and skip reason. The claim-level scope retained here is summarized in Table S24 and its continuation.

S14 Status of secondary extensions and the wider namespace

Later synchrony analysis, Fire VASE, optional visualization integrations, compatibility aliases, and unimplemented reserved names are not evidence for the main paper’s core claims. Their complete callable inventory and exact signatures belong in versioned generated API documentation (Tuff, 2026h). Table S22 preserves only the distinctions needed to prevent later or project-specific capabilities from being attributed to public rc3.

Project-specific callables can participate through the **apply** boundary described in Section S5.3. They retain ordinary pipe syntax but do not automatically acquire maintained scientific contracts, implementation provenance, or validation evidence.

Table S22: Status and evidentiary role of capabilities outside the core demonstration.

Capability family	Status	Boundary
Post-rc3 synchrony diagnostics	Later bounded development work	Does not establish national scalability, regime classification, or rc3 behavior (Quian Quiroga et al., 2002; Donges et al., 2016; Wolf et al., 2020; Tuff, 2026g)
Fire VASE objects and helpers	Project-specific domain extension	Demonstrates an extension shape; it is not a standardized fire-science vocabulary or part of the minimal grammar
Optional visualization and remote-data integrations	Environment-dependent integrations	Availability and successful rendering depend on optional packages, services, and credentials
Compatibility aliases	Maintained for older workflows	Availability does not make them preferred teaching or publication interfaces
Reserved names such as <code>correlation_cube</code> and <code>fit_model</code>	Unimplemented	Must be removed or isolated experimentally before a stable release; their names are not evidence of an available analysis

S15 Reproduction and evidence commands

The public release boundary can be installed without a source checkout:

```
python -m pip install cubedynamics==0.1.0rc3
python -c "import cubedynamics as cd; print(cd.version_info())"
```

The commands below summarize the repository’s documented reproduction routes for later development evidence. They assume an exact source revision, Python 3.9 or later, and the relevant optional extras installed. The repository declares the MIT License, but this supplement does not provide a frozen operating-system image or resolved submission dependency lock; the commands are therefore not substitutes for an archived environment bundle.

```
# Deterministic offline tests
python -m pytest -m "not integration and not online" -q

# Publication validation and all supported vignettes
python scripts/run_validation.py --run-vignettes
python scripts/run_vignettes.py

# Deterministic decision evidence
python scripts/run_decision_qa.py

# Generated references and documentation
python scripts/build_reference_docs.py --check
python -m mkdocs build --strict
python scripts/check_site_links.py site
python scripts/check_repository_size.py --mode tracked
```

The validation runner writes machine-readable JSON, PNG evidence, a suite manifest, and a collated PDF under the ignored `artifacts/validation/` directory. The source-QA route is invoked separately with `python scripts/run_source_qa.py`; any credentialed or online result must be labeled with its date and environment rather than grouped with deterministic fixture evidence. Source QA writes to `artifacts/source_qa/`. Deterministic fixture tests cannot establish present endpoint health.

Table S23: Repository evidence map for manuscript claims.

Repository record	Primary use in the manuscript and supplement
PROMPT_LOG.md	Dated implementation decisions, exact validation results, and remaining caveats
docs/project/rc1_validation_triage.md	One-category-per-finding RC1 classification and maintained semantic decisions
docs/concepts/semantic_grammar.md	State vocabulary, verb contracts, order categories, inspection methods, and boundaries
docs/data/phase1_qa.md	Serving revisions, source profiles, bounded source QA, and remaining source limitations
docs/validation/	Publication validation modules, machine-readable evidence, and visual controls
docs/decision_vignettes/working_lands.ipynb	Executable two-noun analysis, code, plots, interpretation, and reproduction instructions
docs/decision_vignettes/validation.md	Working Lands input values, checksum, acceptance criteria, and dependency ledger
docs/reference/verbs/	Generated public callable status and signatures
tests/fixtures/real_data/	Small checked observational fixtures and provenance records

S16 Public rc3 and the resolved rc1 first-use findings

CubeDynamics 0.1.0rc1 was publicly released on August 31, 2026. Its outside-user acceptance exercise reached a **FIX BEFORE HUMAN OUTSIDE TEST** verdict after three first-use findings: core import failed in the tested Debian/aarch64 Python 3.11 image because the eager optional geospatial stack required `libexpat.so.1`; the initially encountered PRISM path implied an unsupported monthly request; and ordinary NetCDF serialization failed on Boolean metadata attributes (Tuff, 2026b; CubeDynamics naive-user testing project, 2026).

Those findings were addressed before the current public candidate. Core grammar and climate-noun imports no longer require the optional Sentinel-2/Rasterio compiled stack; in the exact affected arm64 image, project-run regression evidence records that core import and one bounded PRISM workflow pass while the optional Sentinel-2 path remains unsupported. PRISM nouns default to daily observations, maintained examples state frequency `D` explicitly, and unsupported monthly access fails rather than silently changing temporal meaning. Raw noun outputs and continuous summaries carry NetCDF-safe scalar metadata. For condition/state Datasets, `to_netcdf` encodes Boolean variables as flagged `int8` in a write-only copy and leaves the in-memory semantic object unchanged (Tuff, 2026g). The available record is not a repeated independent outside-user acceptance exercise for rc3; it should therefore be interpreted as internal release-regression evidence until an independent first-use rerun is archived.

The latest public artifact is 0.1.0rc3, uploaded to PyPI on September 2, 2026 as a wheel and source distribution through Trusted Publishing. PyPI records the following source attestation commit:

f777eb07d3ada8bd407b027f560727c90f6d3731

Rc3 is a release candidate rather than final 0.1.0, and later work on `main` is not part of that artifact unless separately archived (Tuff, 2026c).

S17 Claim-to-evidence matrix

Table S24: Claim-level evidence and explicit boundary, part 1.

Claim	Supporting evidence	Boundary
Authored order is preserved	Pipe semantics and order-rule tests; threshold/mean regressions	Does not prove every possible order relationship is cataloged
Both threshold/mean orders are meaningful and executable	Direct and piped examples plus <code>ORDER_CHANGES_MEANING</code> checks	Does not make the results equivalent
Condition objects compose canonically	Threshold, quantile, overlap, reduction, and plotting tests	Does not imply every relationship has magnitude or threshold fields
Condition mean represents an occurrence proportion	State-only reduction and direct-calculation agreement	Unweighted over the named dimension; missing numeric comparisons are false and remain in the denominator
Variance metadata follows the statistic	Unit-propagation regression tests	Unknown units remain unknown
Contract-inadmissible event transitions are rejected	Expected-failure tests after continuous input or removed time variation	Rule catalog is not complete
Local and regional event scopes remain distinct	Event detection, declared consolidation, and scope-retaining metric tests	Connectivity choices remain researcher-specified and do not establish ecological event identity

Table S25: Claim-level evidence and explicit boundary, part 2.

Claim	Supporting evidence	Boundary
Overlap does not silently harmonize operands	Exact-alignment, temporal-support, and axis-specific diagnostic tests	Coordinate or label equality does not establish scientific comparability; scientists must justify alignment explicitly
Nouns retain source identity	Adapter metadata, provenance records, fixtures, and QA profiles	Common noun does not establish source equivalence
Working Lands defines a two-noun protocol	Reviewed repository commit, documented fixture checksum, query, and acceptance criteria	The binary fixture, generated map, and serialized inspection output are not assembled in the submission archive; full lineage also requires separate branch records
Visualization follows semantic object and dimension	Plot-selection and rendering regressions	Optional backend availability remains environment dependent
Condition NetCDF export preserves in-memory semantics	Serialization regressions and rc3 first-use gate	Boolean state is encoded in the written copy; export does not certify downstream interpretation
Development validation records exist	Dated internal reports describe focused, offline, documentation, visual, and repository checks	Quantitative totals are withheld from submission claims until an exact commit, environment, raw report, and skip list are archived

S18 Limitations and archival requirements

The claim-to-evidence matrix above is the canonical summary of tested behavior and its boundaries. Cross-cutting limitations remain: the architecture is primarily linear; separately unwrapped branches do not merge complete lineage; semantic state depends on supplied metadata; conditions lack an explicit unknown state; alignment exposes but cannot resolve scientific comparability; arbitrary callables retain responsibility for their intent and implementation; uncertainty is not propagated; and source evidence ages. Environmental-modeling frameworks emphasize that uncertainty has multiple sources and must be identified and characterized throughout a study (Refsgaard et al., 2007); recording a semantic state is not a substitute for that work. Missing-as-false semantics, branch-lineage durability, and a frozen submission artifact are stable-release priorities rather than minor documentation issues. Domain extensions add further method-specific choices, such as event connectivity, that remain the researcher’s responsibility.

A mature submission supplement should freeze the exact repository commit used for every claim beyond public rc3, archive machine-readable validation outputs, enumerate skipped tests, and assign persistent identifiers. It should include the checksum-bound Working Lands map, exact inspection records, quantile method and dependency versions, a complete fixture and vignette manifest, rc3 first-use and serialization evidence, final source-certification timestamps, and a contract-to-test mapping that marks rules as enforced, diagnosed, inferred, or documentary. This is the role of a research compendium: to package the publication with the data, code, documentation, and computational environment needed to inspect and re-execute the analysis (Gil et al., 2016; Marwick et al., 2018; Stodden et al., 2016). If the journal permits file bundles, a checksum manifest should bind this PDF to the exact fixture provenance, test reports, generated references, figures, public artifact, and later source revision.

Language-model agents are a prospective use rather than an evaluated contribution. Research on data-driven scientific agents, constrained workflow generation, and tool-selection ambiguity provides relevant context (Gu et al., 2024; Balis et al., 2026; Liu et al., 2026). Explicit names, parameters, state, and diagnostics may provide a bounded tool vocabulary, but this manuscript reports no agent experiment, exported tool-schema evaluation, or improvement in tool selection. Accordingly, this supplement makes no claim of universal harmonization, source equivalence, complete workflow provenance, autonomous scientific judgment, improved human or agent performance, live-provider health from fixtures, or equivalence between public rc3 and later main. Its documented contribution is the tested connection among source-qualified nouns, authored transformations, evolving state, a local trace, ordinary scientific Python objects, and bounded evidence.

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>.
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 2 edition, 2006. ISBN 978-0-321-48681-3. URL <https://www.informit.com/store/compilers-principles-techniques-and-tools-9780321486813>.
- Marius Appel and Edzer Pebesma. On-demand processing of data cubes from satellite image collections with the gdalcubes library. *Data*, 4(3):92, 2019. doi: 10.3390/data4030092. URL <https://doi.org/10.3390/data4030092>.
- 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>.
- 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.
- 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 Hinsén, 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>.
- 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>.
- CubeDynamics naive-user testing project. CubeDynamics 0.1.0rc1 outside-user acceptance report. Black-box outside-user acceptance report, 2026. URL https://github.com/CU-ESIIL/cubedynamics_test_user/blob/main/reports/2026-08-31-rc1/naive_user_report.md. Dated August 31, 2026; verdict: fix before human outside test.
- 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>.
- Ewa Deelman, Karan Vahi, Gideon Juve, Mats Rynge, Scott Callaghan, Philip J. Maechling, Rajiv Mayani, Weiwei Chen, Rafael Ferreira da Silva, Miron Livny, and Kent Wenger. Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, 46:17–35, 2015. doi: 10.1016/j.future.2014.10.008. URL <https://doi.org/10.1016/j.future.2014.10.008>.
- Jonathan F. Donges, Carl-Friedrich Schleussner, Jonatan F. Siegmund, and Reik V. Donner. Event coincidence analysis for quantifying statistical interrelationships between event time series. *The European Physical Journal Special Topics*, 225(3):471–487, 2016. doi: 10.1140/epjst/e2015-50233-y. URL <https://doi.org/10.1140/epjst/e2015-50233-y>.
- Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Valérie Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, Aimé Meygret, Ferruccio Spoto, Olivier Sy, Francesca Marchese, and Paolo Bargellini. Sentinel-2: ESA’s optical high-resolution mission for GMES operational services. *Remote Sensing of Environment*, 120:25–36, 2012. doi: 10.1016/j.rse.2011.11.026. URL <https://doi.org/10.1016/j.rse.2011.11.026>.
- 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>.

- Esri. What is ModelBuilder? ArcGIS Pro documentation, 2026. URL <https://pro.arcgis.com/en/pro-app/latest/help/analysis/geoprocessing/modelbuilder/what-is-modelbuilder-.htm>. Accessed August 28, 2026.
- Martin Fleischmann, Joris Van den Bossche, Kelsey Jordahl, Matthew John Richards, James McBride, Jacob Wasserman, Brendan Ward, and Levi John Wolf. GeoPandas: Fundamental data structures for vector spatial data in Python. *Computers, Environment and Urban Systems*, 130:102495, 2026. doi: 10.1016/j.compenvurbsys.2026.102495. URL <https://doi.org/10.1016/j.compenvurbsys.2026.102495>.
- 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, Bruno Chatenoux, Andrea De Bono, Denisa Rodila, Jean-Philippe Richard, Karin Allenbach, Hy Dao, and Pascal Peduzzi. Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD). *Big Earth Data*, 1(1–2): 100–117, 2017. doi: 10.1080/20964471.2017.1398903. URL <https://doi.org/10.1080/20964471.2017.1398903>.
- Georgios V. Gkoutos, Paul N. Schofield, and Robert Hoehndorf. The units ontology: a tool for integrating units of measurement in science. *Database*, 2012:bas033, 2012. doi: 10.1093/database/bas033. URL <https://doi.org/10.1093/database/bas033>.
- 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>.
- Mordechai Haklay. How good is volunteered geographical information? a comparative study of OpenStreetMap and ordnance survey datasets. *Environment and Planning B: Planning and Design*, 37(4): 682–703, 2010. doi: 10.1068/b35097. URL <https://doi.org/10.1068/b35097>.

- Mordechai Haklay and Patrick Weber. OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008. doi: 10.1109/MPRV.2008.80. URL <https://doi.org/10.1109/MPRV.2008.80>.
- 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>.
- 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>.
- intake-esm contributors. intake-esm: Cataloging and loading earth system model data. Computer software, 2026. URL <https://github.com/intake/intake-esm>.
- International Organization for Standardization. ISO 19156:2023 Geographic information—Observations, measurements and samples. International standard, 2023. URL <https://www.iso.org/standard/82463.html>.
- 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>.
- Johannes F. Kruijer, 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, Alope 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>.
- 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>.
- Meng Lu, Marius Appel, and Edzer Pebesma. Multidimensional arrays for analysing geoscientific data. *ISPRS International Journal of Geo-Information*, 7(8):313, 2018. doi: 10.3390/ijgi7080313. URL <https://doi.org/10.3390/ijgi7080313>.
- 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, Serguei 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>.
- 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>.
- Bertrand Meyer. Applying “design by contract”. *Computer*, 25(10):40–51, 1992. doi: 10.1109/2.161279. URL <https://doi.org/10.1109/2.161279>.
- 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>.

- Grant W. Petty. Automated computation and consistency checking of physical dimensions and units in scientific programs. *Software: Practice and Experience*, 31(11):1067–1076, 2001. doi: 10.1002/spe.401. URL <https://doi.org/10.1002/spe.401>.
- Python Software Foundation. Python 3.14 glossary. Python language documentation, 2026. URL <https://docs.python.org/3/glossary.html>. Definitions of expression and callable; accessed September 25, 2026.
- Rodrigo Quian Quiroga, Thomas Kreuz, and Peter Grassberger. Event synchronization: A simple and fast method to measure synchronicity and time delay patterns. *Physical Review E*, 66(4):041904, 2002. doi: 10.1103/PhysRevE.66.041904. URL <https://doi.org/10.1103/PhysRevE.66.041904>.
- Rob Raskin and Michael Pan. Knowledge representation in the semantic web for earth and environmental terminology (SWEET). *Computers & Geosciences*, 31(9):1119–1125, 2005. doi: 10.1016/j.cageo.2004.12.004. URL <https://doi.org/10.1016/j.cageo.2004.12.004>.
- Rasterio contributors. Rasterio: Geospatial raster I/O for Python programmers. Computer software, 2013–present. URL <https://github.com/rasterio/rasterio>.
- 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>.
- Mattia Righi, Bouwe Andela, Veronika Eyring, Axel Lauer, Valeriu Predoi, Manuel Schlund, Javier Vegas-Regidor, Lisa Bock, Björn Brötz, Lee de Mora, Faruk Diblen, Laura Dreyer, Niels Drost, Paul Earnshaw, Birgit Hassler, Nikolay Koldunov, Bill Little, Saskia Loosveldt Tomas, and Klaus Zimmermann. Earth system model evaluation tool (ESMValTool) v2.0—technical overview. *Geoscientific Model Development*, 13:1179–1199, 2020. doi: 10.5194/gmd-13-1179-2020. URL <https://doi.org/10.5194/gmd-13-1179-2020>.
- 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>.
- Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.1093/biomet/63.3.581. URL <https://doi.org/10.1093/biomet/63.3.581>.
- 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>.
- Shapely contributors. Shapely, version 2.1.2. Computer software, 2025. URL <https://doi.org/10.5281/zenodo.17193310>.
- Alan D. Snow, Rémi Braun, RichardScottOZ, Martin Raspaud, David Brochart, Kirill Kouzoubov, Scott Henderson, Alessandro Amici, stefank0, Taher Chegini, Ray Bell, Mike Taves, Emmanuel Mathot, Daniel Lusk, Andrew Annex, Patrick Hoeffler, Peter Marsh, sangzi-liang, Seth Caldwell, Seth Miller, The Gitter Badger, Tom Augspurger, Tom Chudley, Valan Baptist Mathuranayagam, Xavier Nogueira, Yvonne Fröhlich, acoque, apiwat-chantawibul, and max k. rioarray, version 0.22.0. Computer software, 2026. URL <https://doi.org/10.5281/zenodo.18892731>.
- 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>.
- SpatioTemporal Asset Catalog community. SpatioTemporal Asset Catalog specification, version 1.1.0. Software specification, 2024. URL <https://github.com/radianteearth/stac-spec/releases/tag/v1.1.0>.
- Victoria Stodden, Marcia McNutt, David H. Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A. Heroux, John P. A. Ioannidis, and Michela Taufer. Enhancing reproducibility for computational methods. *Science*, 354(6317):1240–1241, 2016. doi: 10.1126/science.aah6168. URL <https://doi.org/10.1126/science.aah6168>.
- Jason M. Stoker and Barry Miller. The accuracy and consistency of 3D Elevation Program data: A systematic analysis. *Remote Sensing*, 14(4):940, 2022. doi: 10.3390/rs14040940. URL <https://doi.org/10.3390/rs14040940>.
- Peter E. Thornton, Rupesh Shrestha, Michele Thornton, Shih-Chieh Kao, Yaxing Wei, and Bradley E. Wilson. Gridded daily weather data for north america with comprehensive uncertainty quantification. *Scientific Data*, 8:190, 2021. doi: 10.1038/s41597-021-00973-0. URL <https://doi.org/10.1038/s41597-021-00973-0>.
- 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 0.1.0rc1. Public release candidate on PyPI and GitHub, 2026b. URL <https://pypi.org/project/cubedynamics/0.1.0rc1/>. Released August 31, 2026; tag v0.1.0rc1; source commit 4705ccfa33d57e00ca7ca219ce1acd14e774ae46.
- Ty Tuff. CubeDynamics 0.1.0rc3. Public release candidate on PyPI, 2026c. 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 source engineering and validation projects. Project documentation and bounded candidate adapters, 2026d. URL https://github.com/CU-ESIIL/cubedynamics/blob/main/docs/data/source_projects/production.md. Accessed September 25, 2026.
- Ty Tuff. CubeDynamics phase 1 source quality-assurance evidence. Project documentation and machine-readable evidence, 2026e. URL https://github.com/CU-ESIIL/cubedynamics/blob/main/docs/data/phase1_qa.md. Accessed September 25, 2026.
- Ty Tuff. CubeDynamics rc1 comprehensive validation triage. Project validation report, 2026f. 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, 2026g. URL https://github.com/CU-ESIIL/cubedynamics/blob/main/PROMPT_LOG.md. Accessed September 25, 2026.

- Ty Tuff. CubeDynamics public callable inventory. Generated software reference, 2026h. URL <https://github.com/CU-ESIIL/cubedynamics/blob/main/docs/reference/verbs/a-z.md>. Accessed September 25, 2026.
- Ty Tuff. CubeDynamics executable real-data vignette collection. Project documentation and Jupyter notebooks, 2026i. URL <https://github.com/CU-ESIIL/cubedynamics/tree/main/docs/vignettes>. Accessed September 25, 2026.
- Ty Tuff. Working lands: Read hot-and-dry weather as two nouns. Executable CubeDynamics worked environmental vignette, 2026j. URL https://github.com/CU-ESIIL/cubedynamics/blob/862a80aed8a2781b40e6e5293fd6cfbcba887aa4/docs/decision_vignettes/working_lands.ipynb. Reviewed at source commit 862a80aed8a2781b40e6e5293fd6cfbcba887aa4; accessed September 25, 2026.
- U.S. Geological Survey. USGS Water Data for the Nation: U.s. geological survey national water information system database. Data service, 2026. URL <https://doi.org/10.5066/F7P55KJN>.
- 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>.
- Frederik Wolf, Jonas Bauer, Niklas Boers, and Reik V. Donner. Event synchrony measures for functional climate network analysis: A case study on south american rainfall dynamics. *Chaos*, 30(3):033102, 2020. doi: 10.1063/1.5134012. URL <https://doi.org/10.1063/1.5134012>.
- World Wide Web Consortium. PROV-O: The PROV ontology. W3C Recommendation, 2013. URL <https://www.w3.org/TR/prov-o/>.
- Mingda Zhang, Xiaoqian Bu, and Peng Yue. GeoJModelBuilder: an open source geoprocessing workflow tool. *Open Geospatial Data, Software and Standards*, 2:8, 2017. doi: 10.1186/s40965-017-0022-7. URL <https://doi.org/10.1186/s40965-017-0022-7>.
- 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>.