Title

Practical Systems for Formalizing Scientific Terminology and Protocols

Permalink

https://escholarship.org/uc/item/8dj010g6

Author

Gillespie, Charles Thomas Henley

Publication Date

2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Practical Systems for Formalizing Scientific Terminology and Protocols

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Neurosciences (with a specialization in Multiscale Biology)

by

Charles Thomas Henley Gillespie

Committee in charge:

Maryann Martone, Chair
Tom Bartol
Mark Ellisman
Andrew McCulloch
Terry Sejnowski
Bradley Voytek

2023

The dissertation of Charles Thomas Henley Gillespie is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

To Mother who gave me her love of the living world.
To Father who gave me his love of the metaphysical.
To Sohini who gives me a vision of what yet may be.
To Amy and Leon Kass who gave me *D'Aulaires' Greek Myths* that I might love mother Earth and father Sky.
To Leigh van Valen who saw farther riding on giants than most men would see riding spaceships in orbit.
To the countless other mentors, friends, and guides who saw and nurtured in me the spirit of science. I owe to you a debt that can never be repaid.

EPIGRAPH

in this way, if they do not exactly reach the point they desire,
they will come at least in the end to some place that will
probably be preferable to the middle of a forest.
—Rene Descartes

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

Thank you to all of my co-authors. The kind of science we do can only be accomplished as a team. Without each and every one of you this dissertation would not have been possible.

Chapter 1 of this dissertation is a reprint that appeared in full as ”The Neuron Phenotype Ontology: A FAIR Approach to Proposing and Classifying Neuronal Types,” Thomas H. Gillespie, Shreejoy J. Tripathy, Mohameth François Sy, Maryann E. Martone, and Sean L. Hill, Neuroinformatics 20:793-809 2022 [29]. The dissertation author was the primary investigator and was the first author of this paper.

Chapter 2 of this dissertation is a reprint that appeared in full as ”AtOM, an ontology model to standardize use of brain atlases in tools, workflows, and data infrastructures,” Heidi Kleven, Thomas H. Gillespie, Lyuba Zehl, Tim Dickscheid, Jan G. Bjaalie, Maryann E. Martone, and Trygve B. Leergaard, Scientific Data 10:486 2023 [55]. The dissertation author was the primary investigator and co-first author of this paper along with Heidi Kleven.

Thank you to my curators, Jessica Pshoulias, and Gabi Pine. May you never see a three in your boolean columns.

Thank you to all the members of the FAIR Data Informatics lab at UCSD and all of the members of CNL-S at Salk. Having two laborties to call home is a rare gift. You have enriched everything that I do.

Thank you to my many collaborators across countless projects. Building infrastructure for science is often an invisible task, but to us it is a lively cosmos of scientific activity.

Thank you to Mathew Abrams for your long-standing support in facilitating the work that we do, weekly dives into world of methodology and ontology, and for your vision for what neuroinformatics could be.

Thank you to Anita Bandrowski, and Jeffery Grethe for years of sustenance, mentorship, and the occasional reality check.

Thank you to the members of my committe Maryann Martone, Terry Sejnowski, Andrew

McCulloch, Bradley Voytek, Mark Ellisman, and Tom Bartol for allowing me to take on such an ambitious project for a dissertation.

Thank you to Terry Sejnowski for giving me a place to land when I was on my way out of experimental science.

Finally, thank you to my advisor Maryann Martone for your patience that is rivaled only by your demand for excellence. Science as a whole is better because of the stand you take.

VITA

2010 Batchelor of Arts, University of Chicago

2016 Master of Science, University of California San Diego

2023 Doctor of Philosophy, University of California San Diego

PUBLICATIONS

Kleven H, **Gillespie TH**, Zehl L, Dickscheid T, Bjaalie JG, Martone ME, Leergaard TB. AtOM, an ontology model to standardize use of brain atlases in tools, workflows, and data infrastructures. Scientific Data. 2023 July 26;10(1):486. `https://doi.org/10.1038/s41597-023-02389-4`

Hawrylycz MJ, Martone ME, et al. A guide to the BRAIN Initiative Cell Census Network data ecosystem. PLOS Biology. 2023 June 03. `https://doi.org/10.1371/journal.pbio.3002133`

Tan SZ, Kir H, Aevermann BD, **Gillespie T**, Harris N, Hawrylycz MJ, Jorstad NL, Lein ES, Matentzoglu N, Miller JA, Mollenkopf TS, et al. Brain Data Standards-A method for building data-driven cell-type ontologies. Scientific Data. 2023 Jan 24;10(1):50. `https://doi.org/10.1038/s41597-022-01886-2`

Surles-Zeigler MC, Sincomb T, **Gillespie TH**, de Bono B, Bresnahan J, Mawe GM, Grethe JS, Tappan S, Heal M, Martone ME. Extending and using anatomical vocabularies in the stimulating peripheral activity to relieve conditions project. Frontiers in Neuroinformatics. 2022 Aug 24;16:819198. `https://doi.org/10.3389/fninf.2022.819198`

Gillespie TH, Tripathy SJ, Sy MF, Martone ME, Hill SL. The Neuron Phenotype Ontology: A FAIR Approach to Proposing and Classifying Neuronal Types. Neuroinformatics. 2022 Jul;20(3):793-809. `https://doi.org/10.1007/s12021-022-09566-7`

de Bono B, **Gillespie T**, Surles-Zeigler MC, Kokash N, Grethe JS, Martone M. Representing normal and abnormal physiology as routes of flow in ApiNATOMY. Frontiers in Physiology. 2022 Apr 25;13:795303. `https://doi.org/10.3389/fphys.2022.795303`

BRAIN Initiative Cell Census Network (BICCN). A multimodal cell census and atlas of the mammalian primary motor cortex. Nature. 2021 Oct 7;598(7879):86-102. `https://doi.org/10.1038/s41586-021-03950-0`

Osanlouy M, Bandrowski AE, de Bono B, Brooks D, Cassarà AM, Christie R, Ebrahimi N, **Gillespie TH**, Grethe JS, Guercio LA, Heal M, Lin M, Kuster N, Martone ME, Neufeld E, Nickerson DP, Soltani EG, Tappan S Wagenaar JB, Zhuang K, Hunter PJ. The SPARC DRC: Building a Resource for the Autonomic Nervous System Community. Frontiers in Physiology. 2021 June 24;12:693735. `https://doi.org/10.3389/fphys.2021.693735`

Hsu C, Bandrowski AE, **Gillespie TH**, Udel J, Lin K, Ozyurt IB, Grethe JS Martone ME. Comparing the Use of Research Resource Identifiers and Natural Language Processing for Citation of Databases, Software and Other Digital Artifacts. Computing in Science & Engineering. Online 2019 Nov 12. March/April 2020. `https://doi.org/10.1109/mcse.2019.2952838`

Kennedy DN, Abraham SA, Bates JF, Ghosh SS, **Gillespie TH**, Goncalves M, Grethe JS, Halchenko YO, Hanke M, Haselgrove C, Hodge SM. Everything Matters: The ReproNim Perspective on Reproducible Neuroimaging. Frontiers in Neuroinformatics. 2019 Feb;13:1. `https://doi.org/10.3389/fninf.2019.00001`

Babic Z, Capes-Davis A, Martone ME, Bairoch A, Ozyurt IB, **Gillespie TH**, Bandrowski AE. Meta-Research: Incidences of problematic cell lines are lower in papers that use RRIDs to identify cell lines. eLife. 2019 Jan 29;8:e41676. `https://doi.org/10.7554/eLife.41676.001`

Bandrowski AE, **Gillespie TH**, Martone ME. Research Resource IDentifiers for Key Biological Resources. Abstract accepted at: Workshop on Research Objects at IEEE eScience; 2018 Oct 29; Amsterdam, NL. `https://doi.org/10.5281/zenodo.1412731`

ABSTRACT OF THE DISSERTATION

Practical Systems for Formalizing Scientific Terminology and Protocols

by

Charles Thomas Henley Gillespie

Doctor of Philosophy in Neurosciences (with a specialization in Multiscale Biology)

University of California San Diego, 2023

Maryann Martone, Chair

This dissertation presents the results of three efforts to build practical systems for formalizing concepts in science: neuron types, brain regions, and experimental protocols. Neuron types and brain regions are foundational concepts in neuroscience, and protocols are foundational for all scientific results and the concepts we build from them.

Chapter One presents the Neuron Phenotype Ontology and supporting tools and their application to model common types generally known in the field and to experimental types defined by the exact techniques employed in a single lab. The goal: to provide a common method to name and communicate about neuron types by composing types as collections of phenotypes with each phenotype being a pair of a term for the value of the phenotype drawn from existing

shared terminology and a relationship that captures the data modality of and methodology used to determine that value.

Chapter Two presents the AtOM ontology model for anatomical atlases and the results of applying it to model a wide range of extant atlases. The goal: to establish standard ways of identifying atlases and their versions and enabling their use in digital infrastructure to facilitate a wide variety of use cases. Examples include clear communication about the exact spatial and semantic brain regions from which experimental data are collected and linking those regions to the methodologically defined criteria used to delineate their boundaries.

Chapter Three presents `protc/ur`, a domain specific language for specifying protocols, and presents the results of applying it to extract structured data from experimental protocols. The goals: to validate the `protc/ur` domain model and curation workflows, show that `protc/ur` enables queries over complex relationships to find quantitative data extracted from natural language, and, ultimately, demonstrate that `protc/ur` and the system as a whole are an effective way to formalize protocols and make the details of methodology visible in information systems.

Taken together these chapters show the effectiveness of using experimental methodology as an organizing principle in scientific information systems and the potential that it has as a guiding principle for building practical tools for working scientists.

Introduction

Of the many possible perspectives on the work presented in this dissertation, there is one unifying perspective that stands out among the rest. That is formalization, the process of taking informal, colloquial, or even technical but nonetheless idiosyncratic or specific jargon, and translating it into an internally consistent representation that follows a set of explicit rules.

In particular, this dissertation is about building practical systems for formalizing terminology in neuroscience for cell types and brain regions, and about building practical systems for formalizing something that applies to all science – experimental protocols.

As exemplified in the first two chapters, there is a virtuous cycle between the formalization of experimental protocols and formalization of terminology. In chapter one this is seen in the way that the methodological consistency underlying the definition for neuron types within an individual paper made it possible to construct formal representations for those neurons. In chapter two this is seen in the opportunities for data integration that are enabled by the use of formal identifiers for brain regions and proper referencing of atlases. Formal protocols, and effective communication via semi-formal protocols, cannot proceed without at least having formal identifiers for things, for example, the periodic table, IUPAC nomenclature, or the binomial name system[1, 2]. Likewise, the meaning of formal identifiers cannot become sufficiently pre-

[1] Though it is not included as part of this dissertation, our work on Research Resource Identifiers (RRIDs) is of particular interest in understanding the vital importance of formal identifiers for reproducibility and effective communication about critical reagents such as antibodies and cell lines [5].

[2] Some might ask given this statement, how science proceeded before such formal systems came into place. The answer is rooted in part in the difference in scale between early modern science and the current global scientific enterprise, and in the distinction between formalization and communication with others who do not already share

cise for use in science without the formalization of the protocols used to create or classify their referents[3, 4]

In the past it was possible for a single individual to develop an internal understanding that was sufficiently consistent that they were able to make scientific progress without having to communicate extensively with many others[5]. As the phenomena we study have become increasingly complex and as we have had to divide up the labor of understanding nature, communication between individuals and between groups that do not share the same jargon has become unavoidable in order for us to make progress. With it has come the need to formalize our understanding of everything from the meaning of `one inch` to the meaning of `acetylcholine` to the meaning of `Python`. When trying to apply or interpret scientific results, or to follow a protocol, informal communication can lead and has led at best to time wasted failing to get a program to run on the wrong version of a runtime, or, at worst, to disasters such as the capsizing of a ship on its maiden voyage[6].

Buildings systems to enable formalization is particularly challenging [90]. The majority

the same jargon.

[3] Traditionally, this formalization has been embodied in the corporate entities of modern industrial society. The fact that the formalization is not crystallized into a version controlled file written in a formal language does not mean that no such formalization exists. Rather, it means that it is hard to access, understand, and assess the formalization of that knowledge when it is embodied as part of a process carried out by a corporate entity.

[4] Even formal identifiers that seem like they might not need an underlying protocol often do. For example, the ORCID identifier system [41] for research contributors was designed from inception with a protocol defining how it identified contributors that is different from similar systems such as ISNI [64]. Specifically, the ORCID system requires that that the referent of an ORCID identifier must explicitly registered for the ORCID identifier themselves. This means that bulk uploads of contributors from publishers that might contain mistakes and introduce non-existent or duplicate persons have not been used to populate the ORCID registry, but it also means that it is not possible for Aristotle to ever be identified by an ORCID [40].

[5] Consider for example Newton, who certainly corresponded with other scientists at the time, but who was able to develop the foundations for modern physics and calculus while socially isolating in the countryside to avoid the plague in London [63].

[6] This is not to say that creativity and accident do not have a critical role to play in science, nor to say that jargon is informal. Jargon employed by small groups of experts can be extremely formal vital for effective and rapid communication. Issue arise when the formal meaning is implicit [90] or assumed and when two pieces of jargon with the same name but different meaning collide. For example, in the case of the capsized ship, the term ”inch” was exchanged without explicit formalization or standardization and the result was the confusion of the Swedish inch and Amsterdam inch. Both the Swedish foot and Amsterdam foot were the same length, but the Swedish foot had 12 inches and the Amsterdam foot had 11 inches. The difference compounded over many timbers was sufficient in combination with other factors to result in the capsizing of the Vasa [45, 13].

of the practical work in this dissertation focuses on formal languages of one kind or another. While formal languages can provide excellent interfaces to a domain [6, 101, 23], they are also notorious for being a source of enormous frustration and friction for domain experts who are not necessarily experts in the technical systems. The risk as outlined in [90] is that users either work around the formal part of the system, use it incorrectly, or bail out via an escape hatch if one is present.

One conclusion of the work on protocols in chapter three is that supporting gradual formalization is one way to empower users to slowly increase the formality of a protocol or parts of a protocol as their audience changes and grows, or as certain parts of that protocol are identified to be particularly critical for achieving a desired result.

Formalization has a cost. Therefore, it is critical to provide tools that empower formalization when it is necessary without forcing it when it is not. As such, the work underlying this dissertation has attempted, although perhaps not always successfully, to create interfaces that allow users to do practical work in an efficient way. We have found that in the balance of trying to build software systems, it is more practical and cost effective to target the needs of users who can be taught to apply the formalism in collaboration with domain experts, namely curators. Attempting to build systems for formalization that can be used directly by working scientists in the lab is currently beyond the capabilities of most research groups, and as noted, probably not actually a good use of resources.

Despite these challenges, the opportunities created by formalization are immense, as seen in chapters one and two, especially in light of the increasing informaticization of science. One might argue that the need for formalization is particularly acute not only for the new things that can be done with systems that are more than mere stores of natural language text, but also because the neural network based tools that have recently drawn attention are exquisitely bad at dealing with even the most basic types of scientific evidence [7, 36].

Yet despite the obstacles and opportunities, formalization is not an end in and of itself.

There must be some reason to expend such effort.

In all three chapters formalization is perused in service of evidence. For neurons it is evidence tying neuron types to their constituent phenotypes and the phenotypes to the original type of measurement that was made. For brain regions it is evidence tying the location for data collection to specific versions of brain atlases, and evidence tying the regions defined in those atlases to delineation criteria. For protocols it is evidence tying data back to methodology, the details of and constrains on the experimental processes that produced it.

Further, it is not just to link to evidence in principle, but to build systems that link to evidence in practice that can make it possible find whatever evidence we do have to support a given scientific claim, from the existence of a particular type of neuron in a particular brain region, to the validity of a protocol for collecting interpretable data.

The dissertation is laid out into three chapters.

Chapter one focuses on the creation of an ontology for neuron types. The key result is a flexible representation for neuron types that can integrate both existing knowledge from the literature and experimental results into a single system.

Chapter two focuses on the creation of an ontology for anatomical atlases. The key result is an ontology model for anatomical atlases and their parts that can be applied to any type of atlas and that provides clear guidance for both creators and users of atlases with regard to versioning and referencing of atlases.

Chapter three focuses on the creation of a formal language for scientific protocols. The key result is the use of the language to annotate natural language protocols with high coverage of the text and reasonable curation efficiency, and the ability to use the information extracted from those protocols to find for datasets.

Chapters one and two represent the two different phases of the cycle between formal identifiers and experimental protocols. Chapter one shows how methodological consistency makes it possible to create formal identifiers for experimental neuron types. Chapter two provides a way to formalize the identification of atlases and their versions so that they can be used to more precisely name regions used in protocols and communicated in results. Conceptually, chapter three connects the first two together in its focus on the formalization of experimental protocols which are what ultimately provide the definitions of neuron types and brain regions.

Chapter 1

The Neuron Phenotype Ontology: A FAIR Approach to Proposing and Classifying Neuronal Types

Abstract

Most of the work underlying Chapter One of this dissertation was conducted nearly half a decade ago. In the time since then, the underlying system has continued to evolve as it has been applied to represent and manage information about populations of neurons in the peripheral nervous system for the SPARC Connectivity Knowledge base of the Autonomic Nervous System (SCKAN) [30]. As a result of this work we have identified three key additions that need to be made to the system.

The first is the simplified RDF representation that was considered at the time but not implemented. The representation of neuron types as OWL classes is excellent for automatically classifying individual neurons, but it is profoundly bad as a way to work with the types those classes represent. It was not implemented at the time because the Python representation already

provided similar functionality; however, as use cases developed, the types also needed to be more easily accessible in the knowledge base.

The second is the representation of phenotype values as unions and intersections of OWL classes. Without this, the number of terms needed for cross products between, e.g., anatomical regions and layers would be difficult to manage and maintain. Further, forcing users to create such terms defeats the purpose of a composable language for neuron types that can reuse existing terminology.

The third is the representation of the order in which neurites pass through anatomical regions. This has already been implemented [1] as it is particularly important in the peripheral nervous system where a single neuron can pass through many regions via a circuitous route, and where supporting ontologies may lack the adjacency axioms needed to infer the order or where that order would be ambiguous even if those axioms were present, and more practically because specifying the order on the neuron itself keeps the information in one place, is easier to verify, and can in principle be used to cross check adjacency information from supporting ontologies.

Finally, there are two improvements to the system that validate the use of the `neurondm` Python representation as an interchange format for neuron types. The first is that the `neurondm` Python library has been decoupled from the NIF-Ontology and can be used much more easily[2]. The second is that the `neurondm` representation is being used as a common format for alignment and exchange between multiple groups and multiple systems for curating neuron types from the literature, creating visual representations body maps, and representing anatomical flows[3].

[1] `https://github.com/tgbugs/pyontutils/blob/master/neurondm/neurondm/orders.py`

[2] See, e.g., `https://github.com/SciCrunch/sparc-curation/blob/master/docs/sckan-python.ipynb` for an example of how to get started using `neurondm` to work with populations from SCKAN.

[3] See `https://github.com/tgbugs/pyontutils/blob/master/neurondm/docs/composer.py` for examples the application of `neurondm` to these use cases.

Neuroinformatics (2022) 20:793–809
https://doi.org/10.1007/s12021-022-09566-7

The Neuron Phenotype Ontology: A FAIR Approach to Proposing and Classifying Neuronal Types

Thomas H. Gillespie[1] · Shreejoy J. Tripathy[2,3,4] · Mohameth François Sy[5] · Maryann E. Martone[1] · Sean L. Hill[2,3,4,5]

Accepted: 19 January 2022 / Published online: 10 March 2022

Abstract

The challenge of defining and cataloging the building blocks of the brain requires a standardized approach to naming neurons and organizing knowledge about their properties. The US Brain Initiative Cell Census Network, Human Cell Atlas, Blue Brain Project, and others are generating vast amounts of data and characterizing large numbers of neurons throughout the nervous system. The neuroscientific literature contains many neuron names (e.g. parvalbumin-positive interneuron or layer 5 pyramidal cell) that are commonly used and generally accepted. However, it is often unclear how such common usage types relate to many evidence-based types that are proposed based on the results of new techniques. Further, comparing different types across labs remains a significant challenge. Here, we propose an interoperable knowledge representation, the Neuron Phenotype Ontology (NPO), that provides a standardized and automatable approach for naming cell types and normalizing their constituent phenotypes using identifiers from community ontologies as a common language. The NPO provides a framework for systematically organizing knowledge about cellular properties and enables interoperability with existing neuron naming schemes. We evaluate the NPO by populating a knowledge base with three independent cortical neuron classifications derived from published data sets that describe neurons according to molecular, morphological, electrophysiological, and synaptic properties. Competency queries to this knowledge base demonstrate that the NPO knowledge model enables interoperability between the three test cases and neuron names commonly used in the literature.

Keywords Neurons · Cell types · Ontology · Knowledge base · Interoperability · Knowledge integration · FAIR principles

Introduction

The modern description and classification of neurons and the diversity of their properties began with the work of Santiago Ramon y Cajal over 100 years ago. Cajal benefitted from a newly discovered technique, the Golgi stain, to reveal neurons as individual entities of remarkably different shapes, which he described as the "butterflies of the soul". Our knowledge of neuron types (as with cell types) has continued to evolve as new experimental techniques emerge. For this reason, a centerpiece of the US Brain Initiative is to re-examine what constitutes a cell type in light of new ways of probing the nervous system. Through the BRAIN Initiative Cell Census Network (BICCN) researchers are generating large pools of data using cutting edge methods that are being integrated across data types through the use of standards such as common spatial and semantic mappings (Ecker et al., 2017). The BICCN joins several other large initiatives such as the Blue Brain Project (Markram, 2006), Human Cell Atlas (Regev et al., 2017), and SPARC (https://sparc.science/) which also seek to provide foundational knowledge on the types of cells that make up the nervous system. As these data are analyzed and synthesized, new ways to distinguish

✉ Sean L. Hill
sean.hill@epfl.ch

[1] Department of Neuroscience, University of California, San Diego, CA, USA

[2] Department of Psychiatry, University of Toronto, Toronto, ON, Canada

[3] Department of Physiology, University of Toronto, Toronto, ON, Canada

[4] Krembil Centre for Neuroinformatics, Centre for Addiction and Mental Health, Toronto, ON, Canada

[5] Blue Brain Project, École Polytechnique Fédérale de Lausanne (EPFL), Campus Biotech, 1202 Geneva, Switzerland

among different classes of neurons are being proposed and published.

One of the end goals of these large projects is to integrate and analyze large quantities of cellular data to derive new taxonomic classification of neurons across neural structures and to arrive at a new understanding of what constitutes a cell type in the nervous system. To manage this process, some have called for a consistent naming scheme for neurons, so that as new types are discovered, their findings can be reported and compared in an organized way (DeFelipe et al., 2013; Hamilton et al., 2012; Shepherd et al., 2019). Biology has a long history of successfully developing and deploying taxonomies and naming conventions for new entities, e.g., species, enzymes. The process usually involves the commissioning of an authoritative body that comes up with a regularized method and vocabulary for distinguishing among different types and applying an appropriate nomenclature. This approach has been attempted for neuron types. For example, the Petilla terminology proposed a set of criteria and controlled terminology for naming cortical interneurons based on traditional electrophysiological and morphological measurements (Petilla Interneuron Nomenclature Group et al., 2008). However, developing taxonomies and naming conventions pre-supposes that we understand the key dimensions across which neurons should be classified and the foundations of what constitutes a cell type. If the methodological foundations for the classification have not yet reached something universally agreed upon as foundational, such as a nucleotide or amino acid sequence, then the classification remains technique dependent. Thus, as new technologies enable further characterization of additional dimensions,including some that may be foundational, our concept of cell types is likely to evolve. While we know that existing techniques for determining cell type have not yet been able to measure something as foundational as a nucleotide sequence, recent large integrative data gathering exercises have tended to refine our current concepts rather than replace them (Osumi-Sutherland, 2017). A single cell transcriptomic analysis of retinal bipolar cells, (Shekhar et al., 2016), detected 17 different types of RBC, 15 of which had been previously described. The challenge remains to define a knowledge representation that can readily adapt to and integrate results from new data-driven taxonomic efforts but which still supports references to classical naming schemes to ensure integration with the large amount of historical published knowledge. Further, even when foundational techniques can be routinely deployed at scale, not all experiments and certainly not all clinical use cases will be able to employ those techniques directly. Thus, our knowledge management systems need to explicitly account for the techniques that are required to perform such classification so that mappings to other techniques can be developed.

Most proposed schemes, to date, comprise a hierarchical method based on various phenotypic properties for their foundation, i.e., key molecular, physiological, and connectivity signatures that distinguish a neuron type. Phenotypic properties are typically properties of a neuron which are consistent across a variety of measurements, although many phenotypic properties can only be consistently reproduced with a specific experimental technique or protocol. Given the multiple dimensions across which neurons can be differentiated, a phenotype-based approach for classification could effectively generate an almost infinite number of ways to categorize neurons, depending on the granularity at which the distinctions are expressed. A single taxonomy that effectively organizes neurons across these dimensions is unlikely. The recent proposal for naming cortical neurons by (Shepherd et al., 2019) shows how quickly the number of phenotypes can explode, particularly when trying to address the results of dense phenotypic sampling such as array expression. Thus for neuronal cell types, given the complexity and variety of potentially distinguishing features and the likely evolution of these over time, any system for communicating and comparing across phenotypes will require a firm computational foundation.

Traditionally, such proposed classifications are communicated through the research paper, where any taxonomy proposed is presented in the form a table, dendrogram or some other figure (e.g., Paul et al., 2017, Table S7; Markram et al., 2015, Table 1). The problem with our traditional way of constructing and communicating these taxonomies is that they require a human being to understand, compare, and reconcile them (Petilla Interneuron Nomenclature Group et al., 2008). Anyone who has attempted to read through multiple articles, each with their own proposal for classifying cell types within a region understands the difficulties in trying to reconcile the different schemes, even when they are based on limited numbers of data dimensions. The multiplicity of papers proposing classification schemes just for cortical interneurons illustrates this point (Cauli et al., 1997). With the BICCN and other large scale consortia tasked to map the cellular landscape of the brain and body, the potential number of these taxonomies is likely to explode beyond the current already unmanageable number, as researchers apply new types of analytics to understand the data. For neuroscience to move beyond paper-based forums for discussion and integration, we need to treat taxonomies and names as computable artifacts that comply with the FAIR data principles, FAIR = Findable, Accessible, Interoperable and Reusable; (Wilkinson et al., 2016).

Towards that end, we have developed an ontology-based data model, the Neuron Phenotype Ontology (NPO). The NPO aims to provide an interoperable representation of cell

Table 1 The current Phenotypic Dimensions of the NPO and the associated ontologies/vocabularies used to populate the data model. When NIFSTD appears in this table the terms were nearly always added to support the NPO. Examples are drawn from Fig. 2

Phenotypic dimension	Definition	Vocabularies/ontologies
Taxonomic Example: Species	The species or taxon rank in which the phenotype inheres	NCBI taxonomy[1]
Anatomical Example: Brain Region	The regions of the nervous system containing parts of the neuron. Primary location is indicated by the location of the cell soma, but anatomical location may be assigned to any cell part through a series of predicates	UBERON; various brain atlases via NIFSTD parcellation[2]
Morphological	Distinguishing morphological characteristics	NIFSTD[3]
Molecular Example: Expression	Distinguishing molecular constituents	NCBI Gene[4], CHEBI[5], Protein Ontology[6]
Physiological	Expresses a relationship between a neuron type and an electrophysiological phenotype concept. This should be used when a neuron type is described using a high level electrophysiological concept class, e.g., bursting	NIFSTD Petilla Conventions (Petilla Interneuron Nomenclature Group, 2008)
Connection	Indicates a synaptic relationship between cell types. Further elaborated into connectivity determined by different techniques, e.g., physiology, electron microscopy	Gene Ontology[7]
Circuit role Example: Projection	Indicates whether the neuron is an Intrinsic neuron (local circuit neuron), projection neuron, or sensory neuron	NIFSTD (Bug et al., 2008)
Projection targets Example: Projection	Expresses a relationship between a neuron type and a brain region to which it sends axons. Synaptic relationships are represented through the connection relationship	UBERON (Mungall et al., 2012)/various atlases/NIF Gross Anatomy (Bug et al., 2008)

[1] https://www.ncbi.nlm.nih.gov/taxonomy

[2] https://github.com/SciCrunch/NIF-Ontology/blob/master/docs/brain-regions.org

[3] https://github.com/SciCrunch/NIF-Ontology

[4] https://www.ncbi.nlm.nih.gov/gene

[5] https://www.ebi.ac.uk/chebi/

[6] https://proconsortium.org/

[7] http://geneontology.org/

types that can evolve as our phenotypic knowledge evolves, from initial data gathering to modeling and synthesis (Fig. 1). The NPO provides a computable representation of cell types defined by collections of phenotypic properties, designed to enable interoperability between neuronal taxonomies. It is designed to enable scientists to discover which cell types (or potential cell types) share similar properties and to help scientists understand when the cell types they observe are the same or similar to other cell types described in the literature or from other laboratories. Here, we show how the NPO can be used to express taxonomies proposed by different research groups using modern techniques, enable comparisons between them, and enable queries with commonly used neuron types from the literature.

Methods

Overview of NPO

The NPO as well as all data and code referenced below are available for reuse under open licenses (see Data and Code availability statement).

The NPO is composed of two parts. A set of core ontology files that define a data model for neuron types, and the NPOKB, the collection of neuron types defined using the NPO core ontology data model. See supplemental methods for details.

The NPO provides a data model for modeling a neuron type as a "bag of key phenotypes", that is, neurons are represented as a collection of phenotypic properties (Fig. 2) formalized as Web Ontology Language (OWL) classes. These properties can then be used to communicate about and compare phenotypes across laboratories, species, and experimental techniques. This approach has been demonstrated previously in the context of text-based queries of neuron type mentions (Richardet et al., 2015). The original set of object properties for the ontology were sourced from the existing NeuroLex (RRID:SCR_005402) model for neurons (Larson & Martone, 2013). As we developed the CUTs and EBTs we added new properties as needed based on the phenotypes that were measured in particular experiments.

Each of these dimensions is linked to a formal vocabulary or ontology, which is used to provide the descriptors for qualitative phenotypic attributes (Table 1). When possible,

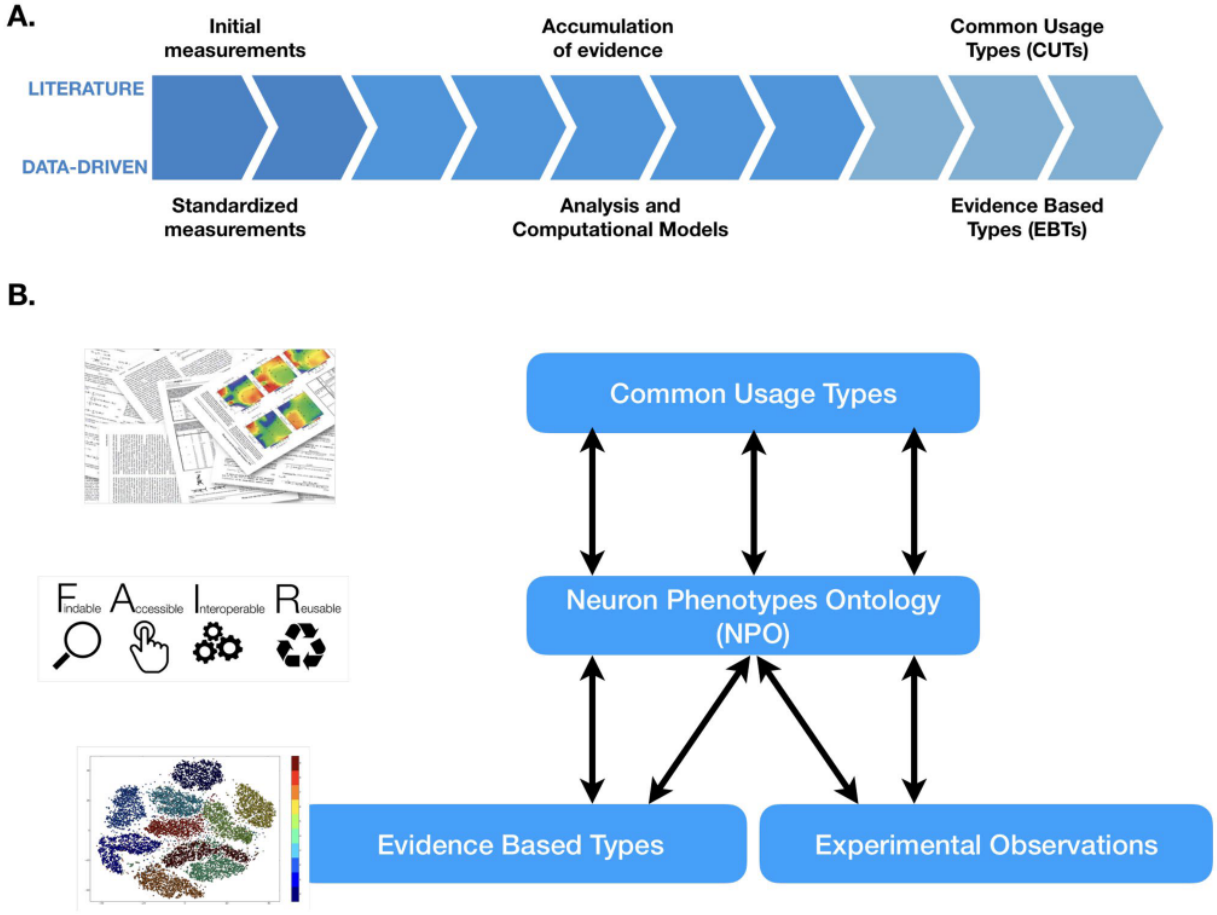

A.
Initial measurements
Accumulation of evidence
Common Usage Types (CUTs)
LITERATURE
DATA-DRIVEN
Standardized measurements
Analysis and Computational Models
Evidence Based Types (EBTs)
B.
Findable Accessible Interoperable Reusable
Common Usage Types
Neuron Phenotypes Ontology (NPO)
Evidence Based Types
Experimental Observations

Fig. 1 Evolution of neuron knowledge **A** Common usage types (CUTs) emerge in the literature as evidence accumulated for generally accepted neuron types with implicitly known properties. Data-driven studies generate evidence-based types (EBTs) based on explicitly measured standardized properties **B** The Neuron Phenotype Ontology (NPO) provides interoperability between the CUTs from the literature, the EBTs from data-driven studies, and new experimental observations from individual laboratories

the vocabularies are drawn from community ontologies/vocabularies in broad use across biomedicine to aid in interoperability. Those dimensions that were not covered by specific community ontologies were added as classes to the appropriate branches of the NIFSTD ontology. NIFSTD is a harmonized set of neuroscience relevant ontologies developed and maintained by the Neuroscience Information Framework (Bug et al., 2008). These dimensions are further elaborated in a set of predicates that capture more granular aspects of phenotypes. For example, *hasMolecularPhenotype* can be further divided into *hasNeurotransmitterPhenotype*, *hasEpigeneticPhenotype*, and *hasExpressionPhenotype* (Fig. 3). *hasExpressionPhenotype* is further broken down into a set of predicates that captures the methodology used to reveal the phenotype. In the current version (v1) of the NPO, we have not made use of the full set of relationships to simplify the reasoning. Relationships that have not been used in the current version of the NPO or that are not in the NPO core but are planned for inclusion in the future are grayed out in Fig. 3.

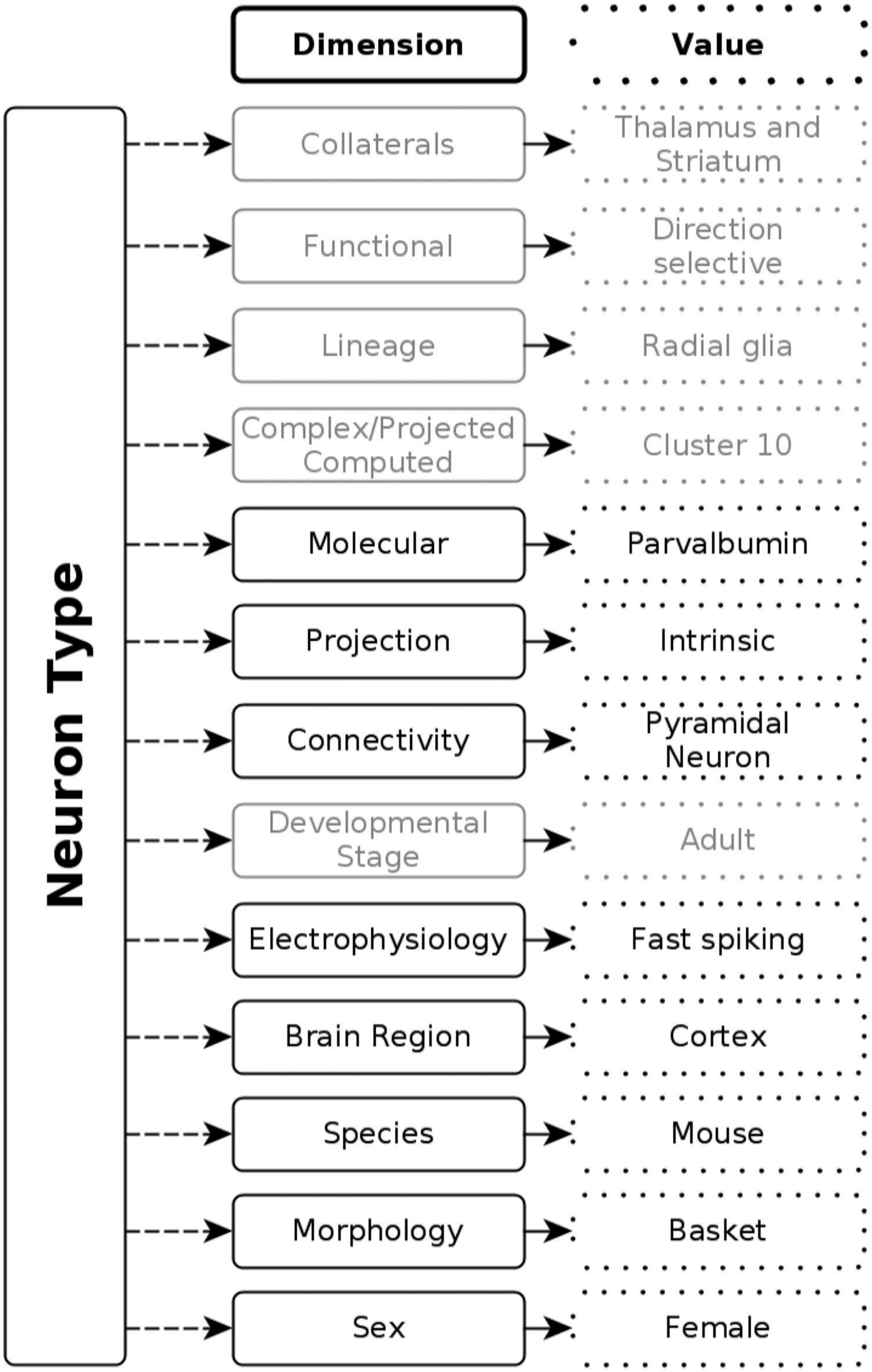
For negative phenotypes, that is, where the lack of a particular phenotype is considered to be a distinguishing feature between neuron types, we use negation in OWL semantics, e.g., a parvalbumin negative neuron would be modeled as "not (*hasExpressionPhenotype* some 'parvalbumin alpha')".

We have also included disjointness axioms[1] in cases where the strength of the assertions from the EBTs were not as definitive as full negation.

For evaluation purposes, we have used the NPO data model to construct a knowledge base of neuronal phenotypes

[1] For an introduction to disjointness axioms in ontologies see Disjointness Between Classes in an Ontology (Stevens & Sattler, 2012) http://ontogenesis.knowledgeblog.org/1260/.

Fig. 2 High level data model for neuron phenotypes. The Neuron Phenotype Ontology characterizes neuron types as bundles of normalized phenotypic properties. Dimensions that have not been used in the current version of the NPO or are planned for the future are grayed out



comprising two branches: 1. Phenotypic representations of common usage types (CUTs) from classical morphological and physiological studies over the past 100 years; 2. Classification models arising from newer experimental techniques tied to individual projects,laboratories or initiatives, termed evidence based types (EBTs). The data model is supported by computational tools that enable individual researchers to compose the complex phenotype of a neuron out of any number of individual phenotypes that are tightly linked to individual data sets and analyses (Fig. 4). We have created

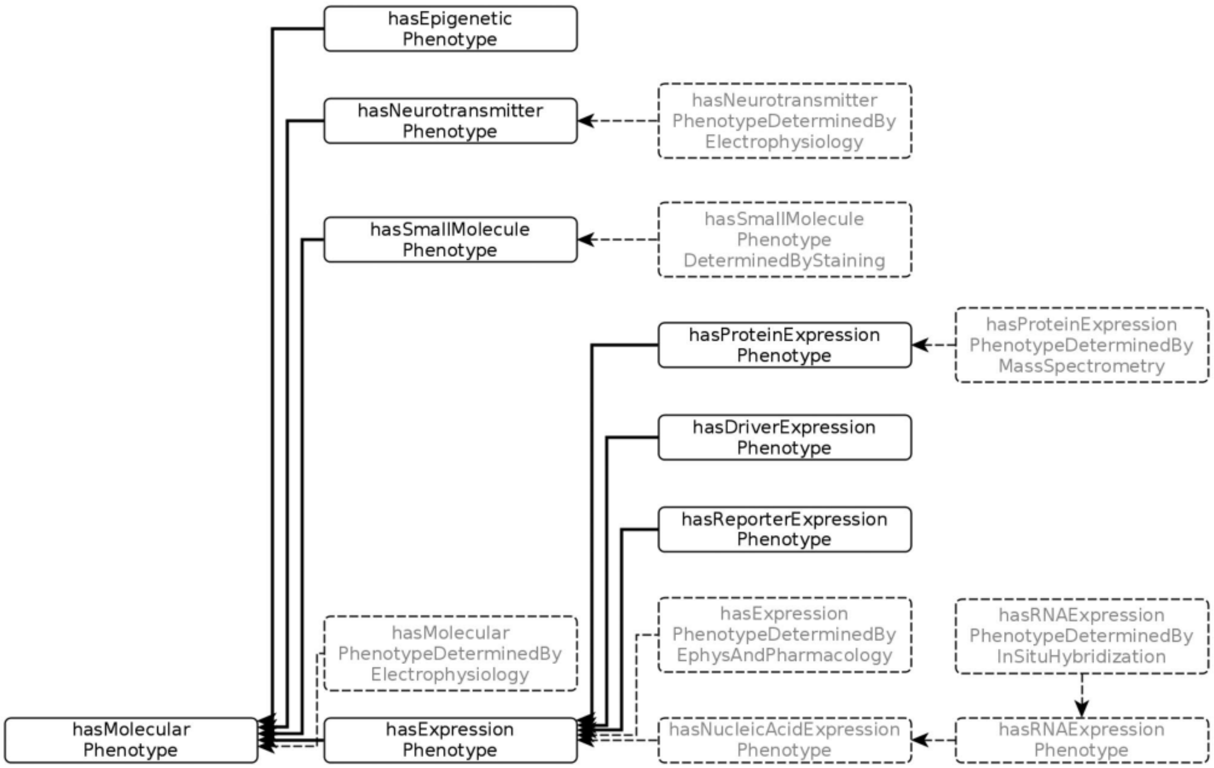

hasEpigenetic Phenotype
hasNeurotransmitter Phenotype
hasNeurotransmitter PhenotypeDeterminedBy Electrophysiology
hasSmallMolecule Phenotype
hasSmallMolecule Phenotype DeterminedByStaining
hasProteinExpression Phenotype
hasProteinExpression PhenotypeDeterminedBy MassSpectrometry
hasDriverExpression Phenotype
hasReporterExpression Phenotype
hasMolecular PhenotypeDeterminedBy Electrophysiology
hasExpression PhenotypeDeterminedBy EphysAndPharmacology
hasRNAExpression PhenotypeDeterminedBy InSituHybridization
hasMolecular Phenotype
hasExpression Phenotype
hasNucleicAcidExpression Phenotype
hasRNAExpression Phenotype

Fig. 3 The set of predicates employed to define molecular phenotypes. Relationships that have not been used in the current version of the NPO or are planned for the future are grayed out

a python library called neurondm that implements Neuron Lang, a domain specific language (DSL) for specifying neuron types. Neuron Lang was created as part of this project to provide a compact representation that expands into more verbose OWL2. The neurondm library provides tools for generating human readable neuron names based on these OWL semantics as well as tools for mapping to and from collections of local names for phenotypes by using ontology identifiers as the common language underlying all local naming. The tools allow us to automatically generate names for neurons in a regular and consistent way using a set of rules operating on the neurons' constituent phenotypes. Neuron types created using neurondm can be exported to Python or to any serialization supported by rdflib, however deterministic turtle[2] (ttl) is preferred. When neurondm generates an OWL ontology it tracks provenance by inserting the exact path and git commit hash for the source python file in the owl:Ontology section via the prov:wasGeneratedBy predicate.

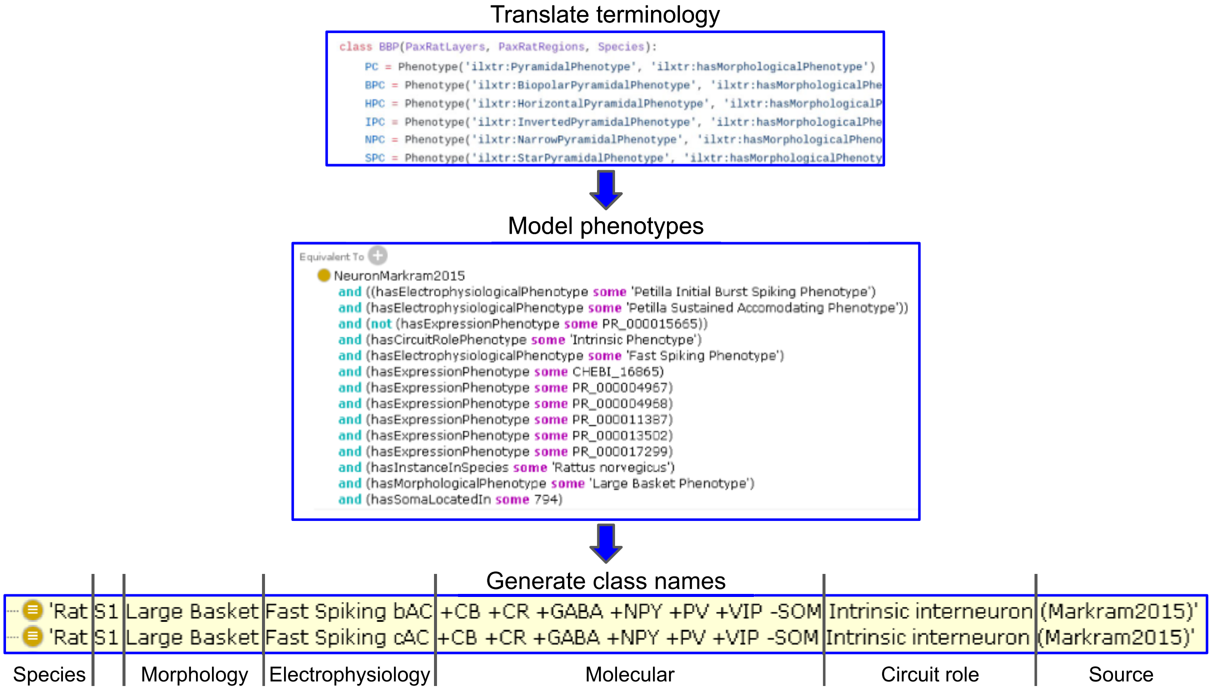
Modeling Decisions

Neuron Class Names

Each neuron in the NPO is identified by a full uniform resource identifier (URI) and a compact identifier for ease of reference. The compact identifier has the prefix npokb and the ontology is registered in BioPortal[3] (RRID:SCR_002713) using the NPOKB prefix as NPO prefix was taken. Each class has multiple human readable labels assigned as annotation properties. Neurons are named according to the phenotypic properties they display. These labels are generated automatically based on the collection of phenotypic properties reported for each cell type using the neurondm Python library. Phenotypes are expressed as OWL2.0 restrictions, and neuron types as equivalent to the intersection of those restrictions (Fig. 4). NPO provides two versions of these names. *Local label* records molecular properties in the native form in which they were measured, e.g., genes, proteins, transgenes, while the *rdfs:label* contains a normalized view where molecules are assigned a common

[2] https://github.com/tgbugs/pyontutils/blob/cc538d9c790d607cbc8c2af8a3c25f1bfa3bfc0b/ttlser/docs/ttlser.md

[3] https://bioportal.bioontology.org/



```
class BBP(PaxRatLayers, PaxRatRegions, Species):
    PC = Phenotype('ilxtr:PyramidalPhenotype', 'ilxtr:hasMorphologicalPhenotype')
    BPC = Phenotype('ilxtr:BiopolarPyramidalPhenotype', 'ilxtr:hasMorphologicalPhe
    HPC = Phenotype('ilxtr:HorizontalPyramidalPhenotype', 'ilxtr:hasMorphologicalP
    IPC = Phenotype('ilxtr:InvertedPyramidalPhenotype', 'ilxtr:hasMorphologicalPhe
    NPC = Phenotype('ilxtr:NarrowPyramidalPhenotype', 'ilxtr:hasMorphologicalPheno
    SPC = Phenotype('ilxtr:StarPyramidalPhenotype', 'ilxtr:hasMorphologicalPhenoty
```

```
NeuronMarkram2015
  and ((hasElectrophysiologicalPhenotype some 'Petilla Initial Burst Spiking Phenotype')
  and (hasElectrophysiologicalPhenotype some 'Petilla Sustained Accomodating Phenotype'))
  and (not (hasExpressionPhenotype some PR_000015665))
  and (hasCircuitRolePhenotype some 'Intrinsic Phenotype')
  and (hasElectrophysiologicalPhenotype some 'Fast Spiking Phenotype')
  and (hasExpressionPhenotype some CHEBI_16865)
  and (hasExpressionPhenotype some PR_000004967)
  and (hasExpressionPhenotype some PR_000004968)
  and (hasExpressionPhenotype some PR_000011387)
  and (hasExpressionPhenotype some PR_000013502)
  and (hasExpressionPhenotype some PR_000017299)
  and (hasInstanceInSpecies some 'Rattus norvegicus')
  and (hasMorphologicalPhenotype some 'Large Basket Phenotype')
  and (hasSomaLocatedIn some 794)
```

```
'Rat S1 Large Basket Fast Spiking bAC +CB +CR +GABA +NPY +PV +VIP -SOM Intrinsic interneuron (Markram2015)'
'Rat S1 Large Basket Fast Spiking cAC +CB +CR +GABA +NPY +PV +VIP -SOM Intrinsic interneuron (Markram2015)'
```

Fig. 4 Process used to translate local terminology into ontology-based representations and machine-generated names. Using neurondm, phenotypes are first mapped by a user into ontology identifiers (top panel). Neuron types are constructed and neurondm automatically translates these mappings into OWL equivalence statements (middle panel). From the same internal representation of these restrictions neurondm generates a set of human readable labels (bottom panel)

molecular abbreviation regardless of the form in which it was measured (see below). For ease of reference we also preserve the common name for the CUT and the original name assigned by the investigator for EBTs if it was provided. These can be found under *origLabel*, and they also appear as *skos:prefLabel* when they are present, otherwise *skos:prefLabel* is populated from *rdfs:label* so there are no neurons missing a preferred label.

For the NPOKB, we generally follow the ordering recommended by Hamilton et al. (2012) and Ecker et al. (2017). In both papers, the recommendation was to create an ordered taxonomy based on key phenotypic features, arranged roughly hierarchically, starting from the highest level, species, followed by anatomical regions, then a set of standardized names for morphological, physiological, molecular or connectional phenotypes (Fig. 4, lower panel). In this way, as proposed originally by Hamilton et al., (2012), it is easy to generate a human readable list of neurons from a given species or brain region and to compare across complex phenotypes. In addition, while we are still sorting out what constitutes a cell type, we define the local environment in which the neuron resides.

Molecular Indicators

For EBTs, NPO preserves the means by which molecular phenotypes are determined. If gene expression is measured, we use the identifier for the gene; if the expression of a transgene is measured, we include the transgene; if the protein is measured, we include the protein. For CUTs, we only use the protein, peptide or small molecules that are thought to define the class. In order to tie together these different measurements, we created a class called phenotype indicator (PhenotypeIndicator) that groups together the different forms of molecular entities, e.g., a somatostatin indicator is equivalent to Sst, SST, SOM, Sst-IRES-Cre, Sst-IRES-FLpO. A somatostatin neuron is then defined as equivalent to any neuron that has some somatostatin indicator as a molecular phenotype. In this way, we simplify the reasoning required to retrieve all somatostatin neurons, but we also clearly preserve the statements made by investigators in their instances or model assertions as preserved in the *localLabel*. In addition, to translate all of the different representations of a particular molecular entity into a consistent human readable label, we have assembled a set of short names that represent each class based largely on common conventions or the names used in NCBI for mouse genes. These short names are used in the *skos:hiddenLabel* for each class and are suffixed with " (indicator)" to create the *rdfs:label*. For example, when generating a label, phenotype indicators for parvalbumin are shown as PV. These labels are available through the “hidden label” annotation property under the *ilxtr:PhenotypeIndicator* class.

Knowledge Base Construction

The basic process for constructing CUTs and EBTs from external sources has four steps where the first three can be done in any order. First we identify the names for the cell types in the source material. For example, in the Paul et al. (2017) paper the header of Table S7 contains the names of 6 neuron types PVBC, CHC, CCK, MNC, ISC, and LPC. Second we identify the phenotype values that are associated with those cell types. For example, in the Markram et al. (2015) paper these include values such as CB, PV, CR, NPY, VIP, SOM, bNAC, cAC, dNAC, bAC, cIR, fast spiking, non-accommodating, non-adapting, late spiking, etc. Third we identify the local names for the phenotypic dimensions that are being used and map those dimensions to and existing object property or create a new one if we determine that the dimensions is determined to be new and not captured by an existing object property, or is a specialization of some more general dimension. For example, for the Allen cell types these are sex_full_name, transgenic_line type,structure, hemisphere, cell_soma_location, and dendrite type. Finally we convert all phenotype values to ontology identifiers, match the values to the dimensions, construct the owl restrictions, and bag them into neuron types with a local name that matches the one provided in the original source.

For EBTs terms are selected for use as a phenotype value as follows. If a term exists and in one of the community ontologies listed in Table 1 we use it, but in some cases (e.g. during development) it is easier to create "new" terms (mint new identifiers) that match the local nomenclature used in the paper to simplify matching the EBTs to the original source. Those "new" terms are then replaced or mapped e.g. with NCBIGene identifiers. For anatomical phenotypes we use terms from species specific anatomical atlases (e.g. the Allen Mouse Reference Atlas Ontology (RRID:SCR_021000)) whenever the original source data mentions them as a reference. Sometimes papers use specific terminology that is not contained in a community ontology in which case we created a new term. There are three areas where new terms were created specifically for the NPO: phenotype indicators which are used to subsume multiple different types of evidence for a phenotype, neuron morphology (which has been upstreamed to PATO), and Petilla electrophysiological classification terms.

For CUTs phenotype indicators are used for molecular phenotypes and Uberon identifiers are used for anatomical phenotypes. Other phenotypic dimensions such as morphology do not have values with the same diversity of indicators and techniques, and thus phenotype classes are chosen in the same way as described for EBTs.

Each set of EBTs was constructed from the original source using a different approach. For the Allen cell types we retrieve the input data in a computationally accessible form from the Allen REST API. Since the original source is computationally accessible there is no issue validating reproducibility of an individual conversion. For the Markram and Huang models the original sources are opaque and computationally inaccessible. As a result we created computationally accessible representations of the original figures and tables. For Markram we converted a figure into a text representation that could be parsed and then manually checked that the accessible version was consistent with the opaque version. For Huang we did not attempt to convert the original source into a visually similar representation and instead encoded the information direction in the Python source file because the underlying sources were rasterized images and a table in a pdf. The effort needed to write custom code for parsing and converting directly from the underlying source could not be justified. See the supplemental methods for details.

When modeling CUTs curators try to the best of their ability to follow the consensus in the literature if there is one. Validation that a CUT is correct is derived from whether the neuron types that are inferred to be subclasses of the CUT include EBTs that should classify under the CUT, and similarly EBTs that should not classify as a CUT are excluded.

While developing the NPO we routinely checked for unexpected classifications, and the competency queries have been developed in part to detect such cases. In principle careful construction of disjointness axioms could be used to cause reasoning errors if an EBT does not classify under the expected CUT, however this has not been implemented.

It is not currently possible to run the code to regenerate the common usage types from the archive reference in this paper because neurondm is configured to pull data from the google sheets API v4. Even if we were to make a copy of the sheet available publicly users would need to configure API access to google sheets which is a significant stumbling block. The neurondm code could be updated to transparently switch between google sheets and an archival source, however this has not been done at this time.

Data and Code Availability

A docker image that captures the environment and code for this paper is available at https://hub.docker.com/layers/156844166/tgbugs/musl/npo-1.0-neurondm-build/images/sha256-c64fef99a0315184b604d20a571e6881de17c4da201edd74830b6169ee0d276a and an archive of that image has been archived on Zenodo at https://doi.org/10.5281/zenodo.5033493. The docker files specifying the image are part of https://github.com/tgbugs/dockerfiles/blob/d942371dc399510914d039022d2b4f92303bc120/source.org#npo-10-neurondm-build and the archive is on Zenodo at http://doi.org/10.5281/zenodo.5068491. See supplemental methods for how to use the image.

The NPO can be viewed by loading the.ttl file available at https://raw.githubusercontent.com/SciCrunch/NIF-Ontology/npo-1.0/ttl/npo.ttl into the Protégé Ontology Tool (RRID:SCR_003299) v5.5.0 or higher. Note that WebProtégé is not capable of running the reasoners required by the NPO. As described in the supplemental methods, npo.ttl is the "light" version of the full ontology that makes it less reliant on the full import chain. Additional information about the structure of the NPO and working with the NPO can be found in the supplemental methods. The NPO is distributed under a CC-BY 4.0 Attribution license, but it imports community ontologies that may be covered under different licenses.

The work here describes v1.0 of the NPO which can be accessed at https://raw.githubusercontent.com/SciCrunch/NIF-Ontology/npo-1.0/ttl/npo.ttl. In the import closure of npo.ttl there are no external imports except for http://purl.obolibrary.org/obo/bfo.owl which had versionIri http://purl.obolibrary.org/obo/bfo/2019-08-26/bfo.owl at the time npo 1.0 was released. All other ontology iris resolve to the neurons branch of the NIF-Ontology except for http://ontology.neuinfo.org/NIF/ttl/generated/parcellation-artifacts.ttl. As a result, importing npo.ttl directly in Protégé will result in the newest version of the imports on the neurons branch being used, which may lead to some small differences in the results compared to what are presented here. However, it is possible to use the NIF-Ontology catalog file to load an exact view of version 1.0 of npo.ttl by cloning the git repository and checking out the npo-1.0 tag. In the event that the npo-1.0 tag is somehow lost at some point in the future, it names the sha1 commit hash 7bb15aa5fda9391809032a6765419dfb2486b2fa which is a merge commit with parents d6615f8 and cdffa6e. The NIF-Ontology repository can identified by root commit hash sha1 ba8482cfccd934b45591e6bbfd6378ef165d0e31 and/or 4f3e0493d926a2c42459b8622dda4de148cf2c5d.

The NPOKB is available on BioPortal at https://bioportal.bioontology.org/ontologies/NPOKB. A loaded graph that can be used with SciGraph, a neo4J-based database for serving ontologies, is available at https://github.com/SciCrunch/NIF-Ontology/releases/tag/npo-1.0.

The content of the NPO is also accessible via the UCSD SciCrunch SciGraph API at https://scicrunch.org/api/1/sparc-scigraph/. Documentation for access can be found at http://ontology.neuinfo.org/docs/NIF-Ontology/README.html#using-nifstd.

The neurondm git repo is https://github.com/tgbugs/pyontutils/tree/master/neurondm. The pyontutils repository can be identified by the root commit hash sha1 6d96945e85d4e949215910f13f3e620495b5e165.

All python code bears an MIT license and is available on the Python Package Index (PyPI) https://pypi.org/project/neurondm/. It can be installed via `pip install neurondm`. Additional instructions are available in the README.[4]

An archive of the code corresponding to this publication is also available on Zenodo at https://doi.org/10.5281/zenodo.4005727. Additional release artifacts are also available on the GitHub release page https://github.com/tgbugs/pyontutils/releases/tag/neurondm-0.1.3.

The full list of CUTs is available at: https://github.com/tgbugs/pyontutils/releases/download/neurondm-0.1.3/data-bundle-2020-08-28.zip.

The full datasets produced for the competency queries (see Results) are available at: Gillespie et al. (2020) https://zenodo.org/record/4007065#.X03TD2dKiAZ.

Results

Common Usage Types

Common usage types represent neuron types that have been reliably identified over many years by multiple groups using multiple techniques. The criteria we used to identify CUTs is provided in Supplementary Table S1. Any type that meets these criteria can and (given sufficient resources) will ultimately be included as a CUT. A master spreadsheet was created in Google Spreadsheets and populated with a list of neuron "stubs" that were created automatically by taking the list of major brain regions in the UBERON ontology and creating two classes per region: Region X projection neuron and Region X intrinsic neuron. These anatomical regions were at a fairly coarse level and comprised the major brain and spinal cord regions, but generally not subregions, for example, cerebral cortex and not motor

[4] https://github.com/tgbugs/pyontutils/blob/master/neurondm/README.md

cortex. Individual brain regions were then augmented with the list of neuron types extracted from online knowledge bases. We started with the list of approximately 300 mammalian neurons from Neurolex Wiki (RRID:SCR_005402) (Larson & Martone, 2013) that had been compiled through expert input via the Neuron Registry Task Force of the INCF (Hamilton et al., 2012), as well as by community contributions. This list was then cross referenced to NeuroElectro (RRID:SCR_006274), BAMS Cells (RRID:SCR_003531), Hippocampome.org (RRID:SCR_009023), NeuroMorpho.org (RRID:SCR_002145) and Blue Brain Project (RRID:SCR_002994). All of these sources were accessed via the Neuroscience Information Framework (RRID:SCR_002894) project to find a set of cells that were referenced in multiple databases. As NeuroElectro maps their nomenclature to the Neurolex names, we used this database to examine representation of these cell types in the neurophysiology literature. We selected all neurons that were referenced in more than one paper.

This procedure resulted in a working list of ~ 350 neurons (for full list see Data Availability Statement). From this list, we then selected ~ 100 neurons for which we had basic morphological and molecular properties available. We also included the neurotransmitter for the majority. We elected to focus in v1.0 primarily on molecular and morphological phenotypes, rather than the full complexity available in the NPO (Fig. 2), as these are the most well known for CUTs and are the most frequent types encountered in the EBTs (Zeng & Sanes, 2017). We also elected in the modeling to take a minimalist approach, that is, our representation is meant not to represent an exhaustive list of every molecule that has been identified within a neuron, but the minimum set of molecules and morphological features that are characteristic for that type. This decision allowed us to construct OWL equivalence statements for each CUT that defined the necessary and sufficient conditions that would allow EBTs to classify under these CUTs. Additional phenotypes were still recorded but added through the Subclassof axiom. Subclassof represents a weaker form of restriction, representing a necessary but not sufficient condition for membership in a class. In order to avoid logical inconsistencies that would interfere with classification, we only included positive phenotypes in necessary and sufficient conditions for CUTs. If distinguishing negative phenotypes were present, they were modeled as entailments rather than OWL restrictions.

Following (Larson et al., 2007), the primary anatomical location of a neuron is assigned based on the brain region in which the soma is located, e.g., cerebellar neuron is equivalent to a neuron with a cell soma in any part of the cerebellum.

Evidence-based Types

EBTs represent cell types and taxonomies proposed by a single group based on an analysis of experimental evidence. In an ideal world the experimental types for every paper ever published and every database involving neurons would be part of the NPO. For this version of the NPO and for the purposes of evaluating our phenotype model, we focused on 3 projects that have generated cortical classifications based on large amounts of experimental data:

A. Cortical cell types proposed by the Blue Brain Project (Markram et al., 2015), as elaborated in the text and Table 1. In this study, 56 total types across 9 morphological types are identified and physiologically characterized from cells in cortical area S1 of rats ranging from P11-P15 from which they recorded physiological properties. Cell-specific molecular markers were confirmed by immunohistochemistry and RT-PCR. (Markram et al., 2015) utilize a nomenclature aligned to the Petilla conventions (Petilla Interneuron Nomenclature Group et al., 2008) to annotate their physiological properties. For NPO V1.0, we included the molecular, morphological and electrophysiological phenotypic dimensions.
B. The classification of proposed cortical GABAergic cell types from Josh Huang and colleagues as summarized in Table S7 of Paul et al. (2017) supplemented with additional information from Fig. 1. The latter was used primarily to create disjointness axioms (see Fig. 1b). For NPO v1.0, we concentrated primarily on the gene expression phenotypes presented in this table, supplemented with information from the rest of the paper, e.g., disjointness axioms based on Fig. 1b. Synaptic and physiological phenotypes will be included in a later version.
C. The ~ 800 cell classes contained in the Allen Cell Types database (RRID:SCR_014806), a database of experimental electrophysiological, morphological and transcriptomic data derived from single cell data. In the Cell Types database, no classification scheme was proposed; rather the records represent statistical summaries of properties measured from these classes of cells identified in transgenic lines. We therefore include this as an EBT. For this version, we focused on molecular measurements from mouse cortex.

Competency Queries

The NPO was designed to classify neurons according to phenotype dimensions, regardless of whether they represent EBTs or CUTs. To test the integrity of the knowledge base and the structure of the ontology, we developed a set of competency queries (CQ):

Table 2 Examples of EBT and CUT neurons returned from Competency query CQ1: Find all examples of parvalbumin containing neurons. The form of the parvalbumin indicator is highlighted in red. Only one example is provided from the Allen EBT (total 59). Full results are available in Gillespie et al. (2020). The compact identifier for each class is prefixed (in bold) to the localLabel for ease of reference. The local label preserves the form in which the molecule was measured. The Common/original name represents the common name from the superclass for all of the physiological subtypes for the Markram cells. However, for the local label we provide a subtype as the superclass does not include the full molecular profile in the name

Type	#	Common/original name	NPO localLabel
CUT	6	**nifext:56**: Neocortex basket cell	**nifext:56**: Mammalia neocortex L2/3 Basket + **PV** + GABA intrinsic neuron
EBT Markram	16	**npokb:112**: Nest basket cell	**npokb:112**: Rattus norvegicus S1 Nest basket (intersectionOf AC b) Fast spiking + GABA + calbindin + CR + NPY + **PV** + VIP -SST intrinsic neuron (Markram2015)
EBT Huang	2	**npokb:43**: PVBC cortical neuron	**npokb:43**: Mus musculus neocortex Basket + GABA + **PV-cre** intrinsic neuron (Huang2017)
EBT Allen	59	none	**npokb:434**: Mus musculus female left cerebral hemisphere VISrl2_3 -Apical Dendrite -Spiny + **Pvalb-T2A-FlpO** + Vipr2-IRES2-Cre + Ai65(RCFL-tdT) neuron (AllenCT)

1. Find all parvalbumin + neurons
   Description Logic (DL) Query: hasPhenotype some 'parvalbumin (indicator)'.
2. Find all cortical neurons that contain somatostatin
   DL Query: hasPhenotype some 'somatostatin (indicator)' and hasSomaLocatedIn some (neocortex or 'part of' some neocortex).
3. How do basket cells described in Paul et al. (2017) and Markram et al. (2015) compare on key dimensions?
   DL Query: (NeuronHuang2017 or NeuronMarkram2015) and hasPhenotype some 'Basket phenotype'.
4. What EBTs are related to the Martinotti cell?
   Determine which neurons classify under the CUT Neocortex Martinotti cell
   DL Query: NeuronEBM and hasPhenotype some 'Martinotti phenotype'

All of the results presented below were produced by issuing OWL DL queries as specified above in Protégé v5.5.0 on a MacBook Pro using the ELK 0.4.3 reasoner unless otherwise noted. More information on loading the ontology into Protégé can be found in the Supplemental Methods.

CQ1: Find All Examples of Parvalbumin Neurons

This query should return all neurons that have a phenotype associated with parvalbumin, regardless of exactly what molecule was measured (DNA, RNA, protein) or how it was measured. In this version of the NPO, we achieve this by creating phenotype indicators without specifying the relationships between these measures through the npokb:parvalbumin (indicator) class. The results of this query are summarized in Table 2. A total of 86 neurons are returned, including EBTs (Huang, N = 2, Markram, N = 16 and Allen; N = 59) and CUTs (N = 9). To aid in comparison across these classes, we illustrate with one example each from the Markram EBTs and Allen data. The complete list of neurons is provided in Gillespie et al., (2020). The original label is provided for each EBT and the common name for the CUT. These are followed by the *localLabel* names that preserve the form of molecule upon which the classifications were based to illustrate how the NPO can be used to compare across different assertions about molecular identity (Markram2015, Huang2017, AllenCT). Related phenotypic values are color coded to aid in comparison. In this case, we use the *localLabel* that preserves the original type of molecule upon which the classifications were based. For a complete list of abbreviations, see Table S2.

Three of the neuron classes indicate that the parvalbumin cells are basket cells, while the Allen data does not specify morphology beyond noting that these cells lack an apical dendrite and dendritic spines.

CQ2: Find All Cortical Neurons That Contain Somatostatin

This query should return all cortical neurons that contain somatostatin regardless of cortical subregion or atlas brain region. Details about how atlas brain regions are handled are provided in the supplemental methods. This query returns a total of 100 neurons, including the neocortex Martinotti cell from the CUT and EBTs from the three classification schemes (Table 3). For Markram, we show only one subtype from each of the 3 main types. For Allen, we selected a few representative examples. Note that Allen neurons are returned from retrosplenial cortex (RSPd2/3) and two areas of primary visual cortex (VISal6a, VISl5) while Markram is returned for primary somatosensory cortex (S1). Both Huang and Allen cells use the same transgenic line for Sst expression, however it is extremely difficult to tell by looking at the laboratory nomenclatures (as demonstrated by Table 3) because they are called SST by Huang and Sst-IRES-FlpO by Allen. Thus, while the local labels preserve the nomenclature

Table 3 Results for CQ2: Find all cortical neurons containing somatostatin. Full results are available in Gillespie et al. (2020). The compact identifier for each class is prefixed (in bold) to the local label for ease of reference. The local label preserves the form in which the molecule was measured. The Common/original name represents the common name from the superclass for all of the physiological subtypes for the Markram cells. However, for the local label we provide a subtype as the superclass does not include the full molecular profile in the name. Similar entities across cell types are color coded. Brain region = blue; somatostatin indicator = red

Type	#	Common/original name	NPO localLabel
CUT	1	**nifext:55**: Neocortex Martinotti cell	**nifext:55**: Mammalia **neocortex** (unionOf EGL L3 L5) (with-axon-in cortical layer I) Martinotti + **Sst** + GABAR + GluR + GABA intrinsic neuron'
EBT Markram	31	• **npokb:114:** Small basket neuron • **npokb:111**: Martinotti neuron • **npokb:109**: Double bouquet neuron	• **npokb:75**: Rattus norvegicus **S1** Small basket (intersectionOf NAC d) Fast spiking + GABA + calbindin + NPY + **SST** + VIP -CR -PV intrinsic neuron (Markram2015) • **npokb:89**: Rattus norvegicus **S1** Martinotti (intersectionOf AC b) Regular spiking non pyramidal + GABA + calbindin + NPY + **SST** -CR -PV -VIP intrinsic neuron (Markram2015) • **npokb:87**: Rattus norvegicus **S1** Double bouquet (intersectionOf IR c) Regular spiking non pyramidal + GABA + calbindin + CR + **SST** + VIP -NPY -PV intrinsic neuron (Markram2015)
EBT Huang	4	• **npokb:42**: MNC neuron • **npokb:45**: LPC neuron	• **npokb:42**: Mouse **Neocortex** Martinotti + GABA (intersectionOf + Adcy2 + Calb2 + Grin3a + Inhbb + Nppc + Pde2a + Rgs6 + Rgs7 + Sst + Zip1 + Znt3) + CR + **SST** interneuron (Huang2017) • **npokb:45**: Mouse **Neocortex** + GABA (intersectionOf + Calca + Chrm2 + Cort + Gpr88 + Gucy1a3 + Gucy1b3 + Hcrtr1 + Kcnmb4 + Nos1 + Opn3 + Oxtr + Pde1a + Penk + Prkg2 + Ptn + Rln1 + Slc7a3 + **Sst** + Syt4 + Syt5 + Syt6 + Tacr1 + Trpc6 + Unc5d + Wnt2) + **SST** + NOS1 projection (Huang, 2017)
EBT Allen	64	none	• **npokb:296**: Mus musculus female right cerebral hemisphere **RSPd2_3** -Apical Dendrite (intersectionOf Spiny sparse) + **Sst-IRES-FlpO** + Nos1-CreERT2 + Ai65(RCFL-tdT) neuron (AllenCT) • **npokb:415**: Mus musculus female left cerebral hemisphere **VISl5** -Apical Dendrite -Spiny + **Sst-IRES-Cre** + Ai14(RCL-tdT) neuron (AllenCT) • **npokb:412**: Mus musculus female right cerebral hemisphere **VISp6a** -Apical Dendrite (intersectionOf Spiny sparse) + **Sst-IRES-Cre** + Ai14(RCL-tdT) neuron (AllenCT)

used in the source (Paul et al., 2017 and Allen Cell Types Database respectively), they are difficult or impossible to use for alignment. The NPO resolves this issue by mapping to identifier systems wherever possible by reviewing the source to see what the local nomenclature actually means. The default labels for neurons (not shown in Table 3) are generated from the underlying identifier which makes it possible to see that Huang and Allen use the same transgenic line (JAX:028579) developed by the Huang lab, regardless of the different local nomenclature. In the NPO, if a transgene is involved, and it was derived from a transgenic mouse line, we use the Jackson lab stock number to represent transgenic phenotype when it is available.

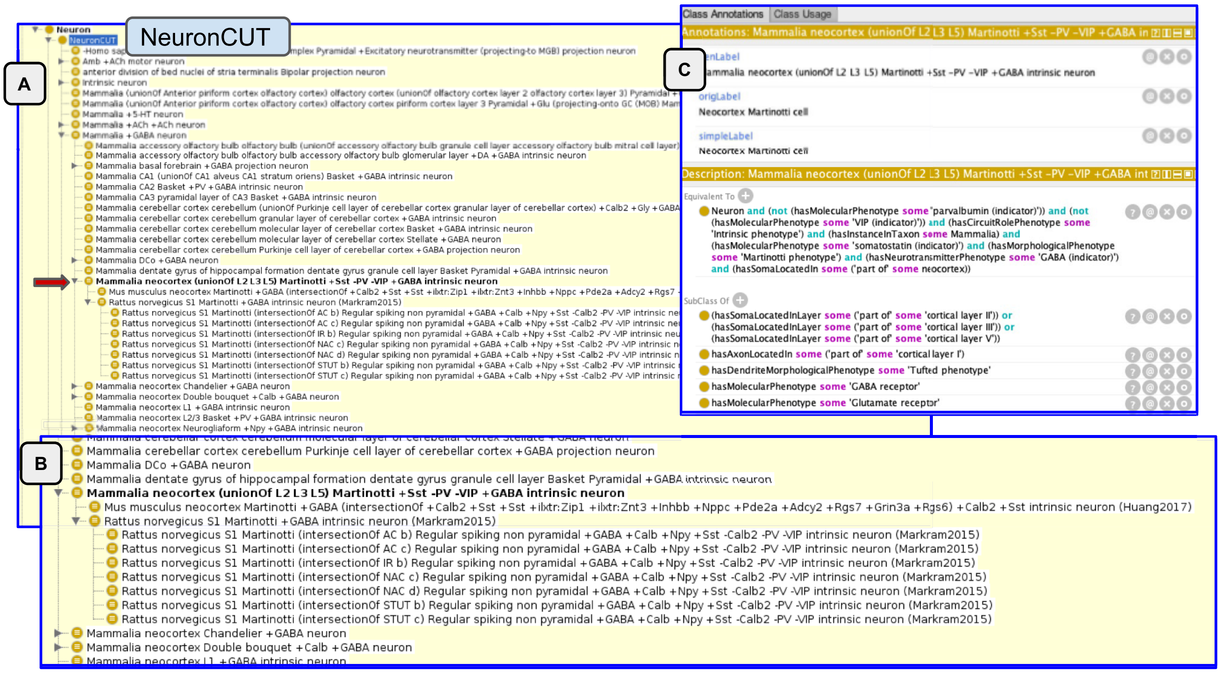
CQ3:How do Basket Cells Described in Paul et al. (2017) and Markram et al. (2015) Compare on Key Dimensions?

This query returned EBT cells from the two groups that were assigned the morphological phenotype "basket". A total of 22 neurons were returned, 20 from Markram and two from Huang. A subset are illustrated in Table 4 and related phenotypes are color coded across the different types for ease of comparison. For the Markram cells, we only show one subtype for each main class.

Two classes of basket neurons are returned for Huang, while three are returned for Markram. Each of the three Markram classes are distinguished by distinct basket morphologies: small basket phenotype, large basket phenotype,

Table 4 Neurons that have a basket phenotype. Similar entities across the cell are color coded to aid in comparison. The full results list is available in Gillespie et al (2020). Similar entities are color coded across cell types: blue = brain region; green = morphology; purple = neurotransmitter; dark red = parvalbumin indicator; red = somatostatin indicator

Original name	NPO ID	NPO Label
PVBC Neuron (Huang2017)	npokb:43	Mus musculus **neocortex Basket + GABA** (intersectionOf + Adm + Cckbr + PV + ilxtr:Kv3 + Rspo2 + Adcy8 + Cox6c + Gabra1 + Gabra4 + Gabrd + Gria1 + Gria4 + Mef2c + Pparg + Ppargc1a + Rgs4 + Slit2 + Slit3 + Tac1 + Arhgef10 + Esrrg + Nefh + Adcy1 + Rasl11b) + **PV** intrinsic neuron (Huang2017)
CCKC Neuron (Huang2017)	npokb:40	Mus musculus **neocortex Basket + GABA** (intersectionOf + Crh + Cck + Cck + Cnr1 + Edn3 + Htr3a + Igf1 + VIP + VIP + Vipr1 + Adcy9 + Chrm3 + Cplx2 + Htr2c + Pnoc + Npy1r + Tac2 + Cplx3 + Pde7b + Prok2 + Hs6st3 + Syt10 + Rgs12) + Cck + **VIP** intrinsic neuron (Huang2017)
Large basket cell (Markram2015): subtype	npokb:59	'Rattus norvegicus **S1 Large Basket** (intersectionOf AC b) Fast Spiking + **GABA** + Calb + Calb2 + Npy + **PV + VIP** -Sst interneuron (Markram2015)'
Nest basket cell (Markram2015): subtype	npokb:65	'Rattus norvegicus **S1 Nest Basket** (intersectionOf AC b) Fast Spiking + **GABA** + Calb + Calb2 + Npy + **PV + VIP** -Sst interneuron (Markram2015)'
Small basket cell (Markram2015): subtype	npokb:73	'Rattus norvegicus **S1 Small Basket** (intersectionOf AC c) Fast Spiking + **GABA** + Calb + Npy + Sst + **VIP** -Calb2 **-PV** interneuron (Markram2015)'



Neuron
NeuronCUT
-Homo sap… mplex Pyramidal +Excitatory neurotransmitter (projecting-to MGB) projection neuron
Amb +ACh motor neuron
anterior division of bed nuclei of stria terminalis Bipolar projection neuron
Intrinsic neuron
Mammalia (unionOf Anterior piriform cortex olfactory cortex) olfactory cortex (unionOf olfactory cortex layer 2 olfactory cortex layer 3) Pyramidal +
Mammalia (unionOf Anterior piriform cortex olfactory cortex) olfactory cortex piriform cortex layer 3 Pyramidal +Glu (projecting-onto GC (MOB) Mam
Mammalia +5-HT neuron
Mammalia +ACh +ACh neuron
Mammalia +GABA neuron
Mammalia accessory olfactory bulb olfactory bulb (unionOf accessory olfactory bulb granule cell layer accessory olfactory bulb mitral cell layer)
Mammalia accessory olfactory bulb olfactory bulb accessory olfactory bulb glomerular layer +DA +GABA intrinsic neuron
Mammalia basal forebrain +GABA projection neuron
Mammalia CA1 (unionOf CA1 alveus CA1 stratum oriens) Basket +GABA intrinsic neuron
Mammalia CA2 Basket +PV +GABA intrinsic neuron
Mammalia CA3 pyramidal layer of CA3 Basket +GABA intrinsic neuron
Mammalia cerebellar cortex cerebellum (unionOf Purkinje cell layer of cerebellar cortex granular layer of cerebellar cortex) +Calb2 +Gly +GABA
Mammalia cerebellar cortex cerebellum granular layer of cerebellar cortex +GABA intrinsic neuron
Mammalia cerebellar cortex cerebellum molecular layer of cerebellar cortex Basket +GABA intrinsic neuron
Mammalia cerebellar cortex cerebellum molecular layer of cerebellar cortex Stellate +GABA neuron
Mammalia cerebellar cortex cerebellum Purkinje cell layer of cerebellar cortex +GABA projection neuron
Mammalia DCo +GABA neuron
Mammalia dentate gyrus of hippocampal formation dentate gyrus granule cell layer Basket Pyramidal +GABA intrinsic neuron
Mammalia neocortex (unionOf L2 L3 L5) Martinotti +Sst -PV -VIP +GABA intrinsic neuron
Mus musculus neocortex Martinotti +GABA (intersectionOf +Calb2 +Sst +Sst +ilxtr:Zip1 +ilxtr:Znt3 +Inhbb +Nppc +Pde2a +Adcy2 +Rgs7
Rattus norvegicus S1 Martinotti +GABA intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf AC b) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic ne
Rattus norvegicus S1 Martinotti (intersectionOf AC c) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic ne
Rattus norvegicus S1 Martinotti (intersectionOf IR b) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neu
Rattus norvegicus S1 Martinotti (intersectionOf NAC c) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic n
Rattus norvegicus S1 Martinotti (intersectionOf NAC d) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic n
Rattus norvegicus S1 Martinotti (intersectionOf STUT b) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic
Rattus norvegicus S1 Martinotti (intersectionOf STUT c) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic
Mammalia neocortex Chandelier +GABA neuron
Mammalia neocortex Double bouquet +Calb +GABA neuron
Mammalia neocortex L1 +GABA intrinsic neuron
Mammalia neocortex L2/3 Basket +PV +GABA intrinsic neuron
Mammalia neocortex Neurogliaform +Npy +GABA intrinsic neuron

Mammalia cerebellar cortex cerebellum molecular layer of cerebellar cortex Stellate +GABA neuron
Mammalia cerebellar cortex cerebellum Purkinje cell layer of cerebellar cortex +GABA projection neuron
Mammalia DCo +GABA neuron
Mammalia dentate gyrus of hippocampal formation dentate gyrus granule cell layer Basket Pyramidal +GABA intrinsic neuron
Mammalia neocortex (unionOf L2 L3 L5) Martinotti +Sst -PV -VIP +GABA intrinsic neuron
Mus musculus neocortex Martinotti +GABA (intersectionOf +Calb2 +Sst +Sst +ilxtr:Zip1 +ilxtr:Znt3 +Inhbb +Nppc +Pde2a +Adcy2 +Rgs7 +Grin3a +Rgs6) +Calb2 +Sst intrinsic neuron (Huang2017)
Rattus norvegicus S1 Martinotti +GABA intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf AC b) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf AC c) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf IR b) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf NAC c) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf NAC d) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf STUT b) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neuron (Markram2015)
Rattus norvegicus S1 Martinotti (intersectionOf STUT c) Regular spiking non pyramidal +GABA +Calb +Npy +Sst -Calb2 -PV -VIP intrinsic neuron (Markram2015)
Mammalia neocortex Chandelier +GABA neuron
Mammalia neocortex Double bouquet +Calb +GABA neuron
Mammalia neocortex L1 +GABA intrinsic neuron

Class Annotations | Class Usage

Annotations: Mammalia neocortex (unionOf L2 L3 L5) Martinotti +Sst –PV –VIP +GABA in

enLabel
ammalia neocortex (unionOf L2 L3 L5) Martinotti +Sst –PV –VIP +GABA intrinsic neuron

origLabel
Neocortex Martinotti cell

simpleLabel
Neocortex Martinotti cell

Description: Mammalia neocortex (unionOf L2 L3 L5) Martinotti +Sst –PV –VIP +GABA int

Equivalent To

Neuron and (not (hasMolecularPhenotype some 'parvalbumin (indicator)')) and (not (hasMolecularPhenotype some 'VIP (indicator)')) and (hasCircuitRolePhenotype some 'Intrinsic phenotype') and (hasInstanceInTaxon some Mammalia) and (hasMolecularPhenotype some 'somatostatin (indicator)') and (hasMorphologicalPhenotype some 'Martinotti phenotype') and (hasNeurotransmitterPhenotype some 'GABA (indicator)') and (hasSomaLocatedIn some ('part of' some neocortex))

SubClass Of

(hasSomaLocatedInLayer some ('part of' some 'cortical layer II')) or (hasSomaLocatedInLayer some ('part of' some 'cortical layer III')) or (hasSomaLocatedInLayer some ('part of' some 'cortical layer V'))
hasAxonLocatedIn some ('part of' some 'cortical layer I')
hasDendriteMorphologicalPhenotype some 'Tufted phenotype'
hasMolecularPhenotype some 'GABA receptor'
hasMolecularPhenotype some 'Glutamate receptor'

Fig. 5 Inferred hierarchy after reasoning over the ontology for the Martinotti cell. Panel **A** shows the hierarchy generated under the NeuronCUT class. The position of the Marinotti CUT is indicated by the lower red arrow. An enlargement of the Martinotti classification is shown in panel **B**. Panel **C** shows the OWL representation of the Martinotti CUT

and nest basket phenotype. These morphologies are modeled as subtypes of BasketPhenotype.

For these types of comparisons, the NPO facilitates comparison across diverse experimental techniques and anatomical nomenclatures and can help to generate testable hypotheses regarding phenotypes. In this example, it is difficult to tell from the information provided whether there is a 1:1 correspondence between any of the Huang and Markram cells. The only molecules mentioned by all 5 cells are GABA, PV and VIP. The Huang PVBC neuron is PV+while the CCKC neuron is VIP+. Two Markram neurons are positive for both PV and VIP, while the small basket cell is asserted to be PV+and VIP-. No negative phenotypes were recorded for the Huang neurons, as we based the equivalence classes on the information available in Table S7 which only included positive phenotypes. In the NPO, we operate under an open world assumption, that is, unless there is an explicit statement that a molecule is lacking, we do not assume that it is absent. We do provide additional information in the form of disjointness axioms based on Fig. 1b of Paul et al. (2017) that the PV-containing and the VIP-containing cells are non-overlapping. This approach dovetails with EBTs making assertions about disjointness of cell types within a species which can be true even if there is not a universal axiom about molecular constituents. Disjointness therefore doesn't mean that there is no expression, but an inspection of the data provided in Fig. 1e indicates that expression of PV in the CCKC neuron is very low. Inspecting the data therefore suggests that the CCKC neuron is VIP+and PV-, consistent with the small basket cell of Markram.

This example illustrates some of the difficulties involved in comparing across phenotypes, particularly when the different phenotypes are measured across experiments. It also illustrates the importance of tying EBTs to experimental data, so that predictions generated from these comparisons can be explored. In this case, Paul et al. (2017) provided expression data for several key molecules in Fig. 1e. This figure shows that while the CCKC neuron expresses little to no PV, consistent with the small basket cell, it also expresses little to no Sst and detectable Calb2, in contrast to the small basket cell. However, as is easily seen in the labels, the Huang and Markram cells come from mouse and rat respectively and how complex molecular phenotypes compare across species is unknown (Yuste et al., 2020).

CQ4: What EBTs are Related to the Martinotti Cell?

To address this competency query, we reasoned over the ontology to determine which neurons would classify under the Neocortex Martinotti neuron CUT. For a neuron to be classified as a type of Martinotti cell, it has to share necessary and sufficient conditions of that class as coded in

Table 5 This rubric (Hodson et al., 2018) organizes the 15 FAIR principles (Applicable principles) into a hierarchical table according to how easy they are to achieve, starting from a basic core (Summary) and rates data according to level of compliance, from 1 to 4 * (Rating). We provide an evaluation of the NPO/NPOKB against these principles in column 4

Rating	Summary	Applicable principles	NPO/NPOKB
*	The basic core: metadata, PID & access	F2. data are described with rich metadata F1. (meta)data are assigned a globally unique and persistent identifier A1. (meta)data are retrievable by their identifier using a standardized communications protocol	• F2: Full descriptive metadata for the ontology are included in the.ttl file. Metadata for the datasets and code are included in Pypi from setup.py, Zenodo, MIRO; The NPOKB includes complete authoring metadata • F1: All datasets referenced in this paper have been assigned DOIs • F1. The NPOKB is assigned a unique identifier (RRID) RRID:SCR_017403 • A1. RRIDs are resolvable through identifiers.org: https://identifiers.org/RRID:SCR_017403 and through the SciCrunch Registry resolver service: https://scicrunch.org/resolver/RRID:SCR_017403 by the Neuroscience Information Framework and dkNET
**	Enhanced access: catalogues for discovery, standard (controlled) access & licences	F4:. (meta)data are registered or indexed in a searchable resource A1.1. the protocol is free, open and universally implementable A1.2. the protocol allows for an authentication and authorization procedure, where necessary R1.1. (meta)data are released with a clear and accessible data usage license	• F4: All python code is available via pypi. ebuilds for Gentoo are available from tgbugs-overlay • F4. The NPO is registered in BioPortal and in the SciCrunch Registry (RRID:SCR_017403) • A1.2 API access is provided via Bioportal and also via SciGraph maintained by the Neuroscience Information Framework and dkNET • R1.1 The NPO is covered under a CC-BY 4.0 license
***	Use of standards: for metadata and data	I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation R1.3. (meta)data meet domain relevant community standards F3: metadata clearly and explicitly include the identifier of the data it describes	• I1: The ontology is built in OWL2, a recognized standard for ontologies • R1.3: The phenotype bags are built out of terms from community standard ontologies • F3: All terms are defined by a URI as well as a compact identifier
****	Rich, FAIR metadata	R1. (meta)data are richly described with a plurality of accurate and relevant attributes I2. (meta)data uses vocabularies that follow FAIR principles	• R1: The ontology has complete metadata associated with it • I2: The NPO has been designed in accordance with the FAIR principles. Documentation • I2: The NPO/NPOKB imports relevant community vocabularies (see Table 1) that adhere to the FAIR principles
*****	Provenance and additional context	R1.2 (meta)data are associated with data provenance I3. (meta)data include qualified references to other (meta)data A2. metadata are accessible, even when the data are no longer available	• R1.2: References that support assertions are included in the annotations although unfortunately OWL does not provide an easy way to annotate specific triples •!3: I2: The NPO/NPO-KB imports relevant community vocabularies (see Table 1) that adhere to the FAIR principles • A2: The NPO and associated tools have been registered with the SciCrunch Registry, which maintains metadata pages for similar resources. They ensure that their metadata is accessible even if the resource is no longer available

the equivalence statements. As discussed in the methods, we deliberately chose to model a minimum of properties as necessary and sufficient due to the large variability in the number of phenotypes recorded for the EBTs. Additional properties are included (Fig. 5C) but not in the form of OWL restrictions, so they do not factor into the reasoning. We also only represent the major classes of CUTs and do not include subtypes, as these are less well agreed upon. In OWL, if we were to require that a Martinotti neuron must have calretinin,then if a given EBT did not state that calretinin was a defining characteristic, the neurons would not classify. In fact, according to Rudy et al. (2011), Martinotti cells contain two subclasses, one that contains calretinin and one that does not. In the NPO, the NeuronHuang2017 EBT notes the presence of calretinin (+Calb2), while the NeuronMarkram2015 EBT says it is absent (-Calb2), perhaps representing these two subclasses.

As Fig. 5 shows, the Allen EBTs do not classify under the Martinotti CUT. In v1.0 of the NPO, we only model morphological phenotypes at a coarse level, e.g., Martinotti phenotype, which is assigned to the level of the entire cell. In contrast, NeuronACT provided morphological information only for the dendrites of each cell. For the cortical somatostatin containing cells, it was noted that they lack an apical dendrite and dendritic spines, but no assertion was made about a Martinotti phenotype, unlike in the other two classifications. In the future, the NPO will include additional defining features of a Martinotti phenotype.

FAIR Properties of the NPO

The NPO was designed to be consistent with the FAIR principles. In Table 5, we show how the NPO achieves FAIR using the rubric in Hodson et al. (2018). The key features are machine readability, the use of identifiers (FAIR vocabularies), common knowledge representation languages and community standards. We provide a comparison with other cellular ontologies in Table S1.

Discussion and Conclusion

The NPO provides a semantically-enriched, FAIR data model for representing the complex cellular phenotypes being generated by neuroscientists involved in individual and large scale brain initiatives. It allows the creation of machine generated taxonomies, and provides a consistent naming convention that is machine configurable. Using the NPO, we showed that we could take cellular data arising from high throughput activities, e.g., the Allen Cell Atlas, large projects like the Blue Brain Project, and from individual investigators to cross between different techniques to show areas of agreement and non-alignment. This exercise is not trivial, as the multiplicity of techniques, the incomplete sampling, and the complex nomenclature present challenges. However, the NPO helps to mitigate these by allowing translation of custom lab nomenclature and experimental results into a common, semantic, and computable representation using community ontologies. The names themselves can be customized to conform to any nomenclature standard that might emerge for human consumption (e.g., Shepherd et al., 2019), but this process is managed as a formal specification rather than through agreed upon naming conventions.

We have focused our efforts on addressing the problem of cell classification vs the issue of determining neuronal types by providing a means to compare our current knowledge about cell types (our common usage types) with the many different classifications being generated by data driven methods and other experimental techniques. The distinction between a neuron type vs a neuron class is not entirely clear, and the terms are often used interchangeably. We use class here to refer to a set of neurons that satisfy a set of criteria, e.g., GABAergic neurons = all neurons that use GABA as a neurotransmitter. The number of potential classes given the number of phenotypic dimensions measured is therefore very large. Types, however, refer to neurons that are sufficiently distinct that the presence of a given set of features will reliably predict the presence of additional features that have not been measured. For example, when a cerebellar Purkinje cell is identified by a Nissl stain based on its size, shape, and location, we can reliably infer that it contains parvalbumin and calbindin, has dendrites densely covered in dendritic spines, and uses GABA as a neurotransmitter whether or not we explicitly measure them. This definition is similar to that proposed by Zeng and Sanes (2017) who propose that types represent discrete groups which notionally serve a specific function while classes represent aggregates of types that share common features. Types are also the categories of cells that must be accounted for when building circuit diagrams of the nervous system (Luo et al., 2008).

The NPO allows us to communicate about and compare measured neuronal phenotypes in a way that reflects human understanding but that can also be fully managed using modern computational methods. Genomics benefitted enormously from a community ontology for annotation of experimental results that allowed them to be communicated in a consistent and machine-processable manner. The issue of neuron typology will also benefit from a consistent annotation framework. Although there are challenges, phenotypes lend themselves to a consistent annotation framework, e.g. genes and morphological features. However, the issue of cell type itself is more fluid. Thus the NPO implements a model that distinguishes between observations in single cells (instances), proposals about cell types derived from computational analyses (EBTs), and cell types that have been recognized by one or more criteria across multiple labs and techniques (CUTs). None of these categorizations represent ground truth. Nevertheless, transcriptomics combined with data driven approaches have shown promise as a unifying technique that may allow stable cell populations to be described within a probabilistic framework (Yuste et al., 2020). Such abstractions will still likely reference entities such as brain regions, marker genes, morphology, and connections. Likewise many of these abstractions will map onto well-known cell types (Yuste et al., 2020). Disagreements are still likely to arise about the nature of these populations, particularly at finer levels of granularity. The NPO and the associated knowledge environment provide a bridge between classifications generated using high throughput and integrative techniques and our accumulated knowledge over the past 100 years on cell types in the nervous system.

Looking to the future, extension of the NPO beyond the contents described in this paper is already underway. We have started to create new evidence based types for the peripheral nervous system as part of the NIH SPARC consortium (Osanlouy et al., 2021). Application to the peripheral nervous system is an extension along the location dimension. Extensions along other dimensions are also possible. The taxonomic dimension is an obvious candidate. The inclusion of invertebrate and avian neuron types would significantly broaden the generality of the content of the NPO and further test the flexibility of the approach. To truly understand the nervous system we will likely need to study it in all its variation across a menagerie of clades and dimensions. We designed the NPO to have a flexible data model so that it could not only accommodate such diversity, but also be enhanced by it. The ongoing initiatives to exhaustively catalog neuron types for Drosophila melanogaster seem like they could provide a tractable testing ground for applying the NPO at scale and for the infrastructure that will be needed to manage the flood of vertebrate data that will be collected over the coming years.

The work reported here should be considered a proof-of-concept; in order for the NPO to be used at the scale we envision significant additional tooling would be required. Currently, the Python code can be used by a researcher to translate their phenotypes into NPO and they can compare their neurons locally to the NPOKB using Protégé. To gain traction, increase ease of use, and populate the knowledge base, we envision a set of on-line tools that would assist researchers in translating their phenotypes into the NPO, along with a web-accessible growing knowledge base with visualization and analysis tools for researchers to compare their neurons to what is known. Yuste and colleagues (2020) also envision an online community knowledge base where information on cell types is accumulated and linked. In addition, the NPO currently only provides the skeleton of discrete types on top of which the continuous nature of measurements needs to be integrated. Nonetheless, the goals of the BRAIN initiative and other large scale data projects are to transform our understanding of the brain using new technologies and data science and understanding the "parts list" of the nervous system is a key objective (Zeng & Sanes, 2017). If we accept the premise that no single project or group can do it alone, then neuroscientists must produce data and knowledge artifacts like atlases and taxonomies in a way that is amenable to computation. The FAIR data principles outline some of the basic ways to do that (Table 5). Integral to FAIR is the use of community standards that make the process of searching, aggregating, and reusing data more tractable. The proposed methods do not require that we all think alike, rather, they ensure that we can employ computational methods to compare and contrast across different classification schemes. Although the proposed approaches would require a significant investment by funders and researchers alike to develop and adopt these methods, we have to measure this against the time we currently spend trying to reconcile computationally opaque and un-FAIR neuroscience data. In an ideal world, we would focus our resources on grappling with the innate complexity of the issue of cell types in the brain, rather than having to focus on reconciling the myriad number of ways we can refer to common entities in neuroscience.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12021-022-09566-7.

Acknowledgements This work was supported by NIH Brain Initiative award 5U24MH114827-04, a Canadian Institute for Health Research post-doctoral fellowship and National Institutes of Health grant MH111099, the Krembil Foundation, and by funding to the Blue Brain Project, a research center of the École polytechnique fédérale de Lausanne (EPFL), from the Swiss government's ETH Board of the Swiss Federal Institutes of Technology. Meetings supporting this work were facilitated by the International Neuroinformatics Coordinating Facility (INCF) through the Neuroinformatics for Cell Types Special Interest Group. The authors thank Felix Schürmann and Karin Holm for helpful comments.

Funding Open access funding provided by EPFL Lausanne.

References

Bug, W. J., Ascoli, G. A., Grethe, J. S., Gupta, A., Fennema-Notestine, C., Laird, A. R., Larson, S. D., et al. (2008). The NIFSTD and BIRNLex Vocabularies: Building Comprehensive Ontologies for Neuroscience. *Neuroinformatics, 6*(3), 175–194.

Cauli, B., Audinat, E., Lambolez, B., Angulo, M. C., Ropert, N., Tsuzuki, K., Hestrin, S., & Rossier, J. (1997). Molecular and Physiological Diversity of Cortical Nonpyramidal Cells. *The Journal of Neuroscience: THe Official Journal of the Society for Neuroscience, 17*(10), 3894–3906.

DeFelipe, J., López-Cruz, P. L., Benavides-Piccione, R., Bielza, C., Larrañaga, P., Anderson, S., Burkhalter, A., et al. (2013). New Insights into the Classification and Nomenclature of Cortical GABAergic Interneurons. *Nature Reviews Neuroscience, 14*(3), 202–216.

Diehl, A. D., Meehan, T. F., Bradford, Y. M., Brush, M. H., Dahdul, W. M., Dougall, D. S., He, Y., et al. (2016). The Cell Ontology 2016: Enhanced Content Modularization and Ontology Interoperability. *Journal of Biomedical Semantics, 7*(1), 44.

Ecker, J. R., Geschwind, D. H., Kriegstein, A. R., Ngai, J., Osten, P., Polioudakis, D., Regev, A., Sestan, N., Wickersham, I. R., & Zeng, H. (2017). The BRAIN Initiative Cell Census Consortium: Lessons Learned toward Generating a Comprehensive Brain Cell Atlas. *Neuron, 96*(3), 542–557.

Franklin, K. B. J., & Paxinos, G. (2008). *The Mouse Brain in Stereotaxic Coordinates* (3rd ed.). Academic Press.

Gillespie, T. H., Martone, M. E., & Hill, S. L. (2020). Results of Neuron Phenotype Ontology Competency Queries. https://doi.org/10.5281/zenodo.4007065

Hamilton, D. J., Shepherd, G. M., Martone, M. E., & Ascoli, G. A. (2012). An Ontological Approach to Describing Neurons and Their Relationships. *Frontiers in Neuroinformatics, 6*, 15.

Hodge, R. D., Bakken, T. E., Miller, J. A., Smith, K. A., Barkan, E. R., Graybuck, L. T., Close, J. L., et al. (2019). Conserved Cell Types with Divergent Features in Human versus Mouse Cortex. *Nature, 573*(7772), 61–68.

Hodson, S., Jones, S., Collins, S., Genova, F., Harrower, N., Laaksonen, L., Mietchen, D., Petrauskaité, R., & Wittenburg, P. (2018). Turning FAIR Data into Reality: Interim Report from the European Commission Expert Group on FAIR Data. https://doi.org/10.5281/zenodo.1285272

Larson, S. D., Fong, L. L., Gupta, A., Condit, C., Bug, W. J., & Martone, M. E. (2007). A Formal Ontology of Subcellular Neuroanatomy. *Frontiers in Neuroinformatics, 1*, 3.

Larson, S. D., & Martone, M. E. (2013). NeuroLex.org: An Online Framework for Neuroscience Knowledge. *Frontiers in Neuroinformatics*, *7*, 18

Luo, L., Callaway, E. M., & Svoboda, K. (2008). Genetic Dissection of Neural Circuits. *Neuron, 57*(5), 634–660.

Markram, H. (2006). The Blue Brain Project. *Nature Reviews Neuroscience, 7*(2), 153–160.

Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., Ailamaki, A., et al. (2015). Reconstruction and Simulation of Neocortical Microcircuitry. *Cell, 163*(2), 456–492.

Mungall, C. J., Torniai, C., Gkoutos, G. V., Lewis, S. E., & Haendel, M. A. (2012). Uberon an Integrative Multi-Species Anatomy Ontology. *Genome Biology, 13*(1), R5.

Osanlouy, M., Bandrowski, A., de Bono, B., Brooks, D., Cassarà, A. M., Christie, R., Ebrahimi, N., Gillespie, T., Grethe, J. S., Guercio, L. A., Heal, M., Lin, M., Kuster, N., Martone, M. E., Neufeld, E., Nickerson, D. P., Soltani, E. G., Tappan, S., Wagenaar, J. B., … Hunter, P. J. (2021). The SPARC DRC: Building a Resource for the Autonomic Nervous System Community. *Frontiers in Physiology, 12*(929), 693735. https://doi.org/10.3389/fphys.2021.693735

Osumi-Sutherland, D. (2017). Cell Ontology in an Age of Data-Driven Cell Classification. *BMC Bioinformatics, 18*(Suppl 17), 558.

Paul, A., Crow, M., Raudales, R., He, M., Gillis, J., & Huang, Z. J. (2017). Transcriptional Architecture of Synaptic Communication Delineates GABAergic Neuron Identity. *Cell, 171*(3), 522–539.

Petilla Interneuron Nomenclature Group, Ascoli, G. A., Alonso-Nanclares, L., Anderson, S. A., Barrionuevo, G., Benavides-Piccione, R., Burkhalter, A., et al. (2008). Petilla Terminology: Nomenclature of Features of GABAergic Interneurons of the Cerebral Cortex. *Nature Reviews Neuroscience, 9*(7), 557–568.

Regev, A., Teichmann, S. A., Lander, E. S., Amit, I., Benoist, C., Birney, E., Bodenmiller, B., et al. (2017). The Human Cell Atlas. *eLife, 6*, e27041. https://doi.org/10.7554/eLife.27041

Richardet, R., Chappelier, J. C., Tripathy, S., & Hill, S. (2015, November). Agile text mining with Sherlok. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 1479–1484). IEEE. https://doi.org/10.1109/BigData.2015.7363910

Rudy, B., Fishell, G., Lee, S., & Hjerling-Leffler, J. (2011). Three Groups of Interneurons Account for Nearly 100% of Neocortical GABAergic Neurons. *Developmental Neurobiology, 71*(1), 45–61.

Shekhar, K., Lapan, S. W., Whitney, I. E., Tran, N. M., Macosko, E. Z., Kowalczyk, M., Adiconis, X., et al. (2016). Comprehensive Classification of Retinal Bipolar Neurons by Single-Cell Transcriptomics. *Cell, 166*(5), 1308–1323.

Shepherd, G. M., Marenco, L., Hines, M. L., Migliore, M., McDougal, R. A., Carnevale, N. T., Newton, A. J. H., Surles-Zeigler, M., & Ascoli, G. A. (2019). Neuron Names: A Gene- and Property-Based Name Format, With Special Reference to Cortical Neurons. *Frontiers in Neuroanatomy, 13*, 25.

Stevens, R., & Sattler, U. (2012). Disjointness Between Classes in an Ontology. *Ontogenesis*

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., et al. (2016). The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Scientific Data*, 3, 160018

Yuste, R., Hawrylycz, M., Aalling, N., Aguilar-Valles, A., Arendt, D., Armañanzas, R., ... & Lein, E. (2020). A community-based transcriptomics classification and nomenclature of neocortical cell types. *Nature Neuroscience*, *23*(12), 1456–1468. https://doi.org/10.1038/s41593-020-0685-8

Zeng, H., & Sanes, J. R. (2017). Neuronal Cell-Type Classification: Challenges Opportunities and the Path Forward. *Nature Reviews Neuroscience, 18*(9), 530–546.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgments

Chapter 1 of this dissertation is a reprint that appeared in full as ”The Neuron Phenotype Ontology: A FAIR Approach to Proposing and Classifying Neuronal Types,” Thomas H. Gillespie, Shreejoy J. Tripathy, Mohameth François Sy, Maryann E. Martone, and Sean L. Hill, Neuroinformatics 20:793-809 2022 [29]. The dissertation author was the primary investigator and was the first author of this paper.

Chapter 2

AtOM, an ontology model to standardize use of brain atlases in tools, workflows, and data infrastructures

scientific **data**

AtOM, an ontology model to standardize use of brain atlases in tools, workflows, and data infrastructures

Heidi Kleven[1,5], Thomas H. Gillespie[2,5], Lyuba Zehl[3], Timo Dickscheid[3,4], Jan G. Bjaalie[1], Maryann E. Martone[2] & Trygve B. Leergaard[1]✉

Brain atlases are important reference resources for accurate anatomical description of neuroscience data. Open access, three-dimensional atlases serve as spatial frameworks for integrating experimental data and defining regions-of-interest in analytic workflows. However, naming conventions, parcellation criteria, area definitions, and underlying mapping methodologies differ considerably between atlases and across atlas versions. This lack of standardized description impedes use of atlases in analytic tools and registration of data to different atlases. To establish a machine-readable standard for representing brain atlases, we identified four fundamental atlas elements, defined their relations, and created an ontology model. Here we present our Atlas Ontology Model (AtOM) and exemplify its use by applying it to mouse, rat, and human brain atlases. We discuss how AtOM can facilitate atlas interoperability and data integration, thereby increasing compliance with the FAIR guiding principles. AtOM provides a standardized framework for communication and use of brain atlases to create, use, and refer to specific atlas elements and versions. We argue that AtOM will accelerate analysis, sharing, and reuse of neuroscience data.

Introduction

Brain atlases are essential anatomical reference resources that are widely used for planning experimental work, interpreting and analyzing neuroscience data[1–12]. Three-dimensional (3D) digital brain atlases[3,13–17] are increasingly employed as frameworks for integrating, comparing, and analyzing data based on atlas-defined anatomical locations (e.g. Allen brain map (https://portal.brain-map.org); the BRAIN Initiative Cell Census Network (https://www.biccn.org); the EBRAINS research infrastructure (https://ebrains.eu)). These resources provide anatomical context suitable for brain-wide or region specific analysis using automated tools and workflows[18–26] and facilitate sharing and using data in accordance with the FAIR principles[27], stating that data should be findable, accessible, interoperable, and reusable. However, the use and incorporation of different atlas resources in such workflows and infrastructures requires that atlases, tools, and data are interoperable, with relatively seamless exchange of standardized machine-readable information.

Most brain atlases share a set of common properties, but the specifications and documentation of their parts differ considerably. Detailed versioning is not yet common practice for all atlases and lack of specific information about changes in the terminology or anatomical parcellation make it difficult to compare atlas versions. While some gold standards have been established[28], lack of consensus regarding the presentation, specification, and documentation of atlas contents hampers reproducible communication of locations[11] and comparison of data that have been anatomically specified using different atlases[10,24]. Atlases and their versions need to be uniquely identifiable and interoperable to enable researchers to communicate specific and reproducible location data and integrate data across specialized neuroscience fields and modalities.

[1]Department of Molecular Medicine, Institute of Basic Medical Sciences, University of Oslo, Oslo, Norway. [2]Department of Neurosciences, University of California, San Diego, USA. [3]Institute of Neuroscience and Medicine (INM-1), Research Centre Jülich, Jülich, Germany. [4]Institute of Computer Science, Heinrich Heine University Düsseldorf, Düsseldorf, Germany. [5]These authors contributed equally: Heidi Kleven, Thomas H. Gillespie. ✉e-mail: t.b.leergaard@medisin.uio.no

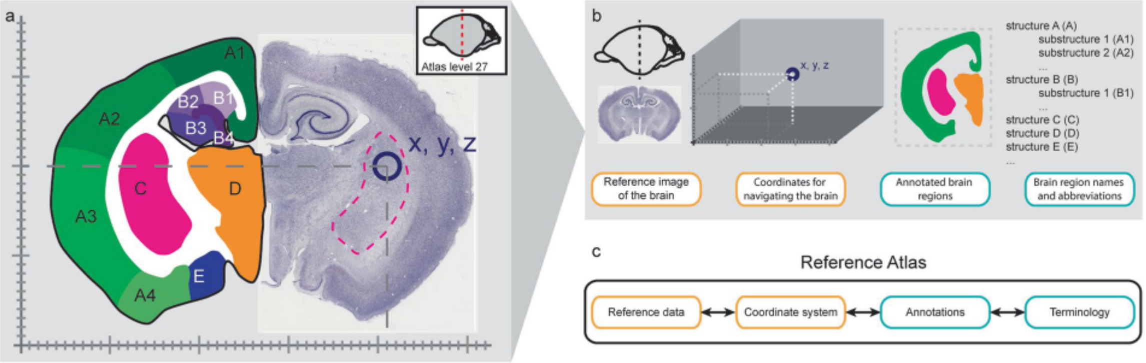


Fig. 1 Atlas Ontology Model elements. (**a**-**b**) The elements of a fictional two-dimensional brain atlas illustrated using a coronal Nissl-stained section and a drawing of the Platypus (*ornithorhynchus anatinus*) brain[79]. (**c**) The Atlas Ontology Model, formalizing the elements of a reference atlas.

To address the lack of standardization of atlas metadata, we identified four common atlas elements, defined their relations, and created the Atlas Ontology Model (AtOM). By specifying the relations and hierarchies of objects and processes in an ontology model[29], we created systematic and coherent links among data files, metadata, and process descriptions enabling automated retrieval of information in using computational tools[30].

Here we characterize the properties and relations of the elements of brain atlases and explain their organization in the model. Using the relations defined by AtOM we show that any specific set of atlas elements and their associated metadata makes up a unique version of an atlas. Furthermore, we suggest a set of minimum requirements for atlases inspired by the FAIR principles and discuss how atlases adhering to AtOM could accelerate neuroscience data integration.

Results

We investigated a broad selection of mammalian brain atlases[3,13,14,17,31–39] and identified four common elements: (1) a set of reference data, (2) a coordinate system, (3) a set of annotations and (4) a terminology. Below, we describe these atlas elements and their relations, and exemplify how they can be identified in different atlases. We go on to show how the ontology model allows specification of unique atlas versions. Lastly, we employ AtOM to suggest minimum requirements for FAIR brain atlases and briefly describe how these requirements facilitate the incorporation of brain atlases into research workflows and software tools.

The atlas elements. The atlas elements in AtOM are the reference data, coordinate system, annotation set, and terminology (Figs. 1, 2). Each of the four elements have properties, such as identifier, species, sex, and age, specified with detailed metadata (Fig. 2b,c).

The *reference data* of a brain atlas are graphical representations of one or several brains, or parts of brains, chosen as the biological reference for that atlas. The reference data typically consist of histological or tomographic images. These images may be derived from a selected specimen, with the assumption that it represents generalizable biological features within its age category and biological sex. This is the case for the BigBrain human brain atlas (with reference data showing cytoarchitecture of one adult male[16]), and for many rodent atlases (which typically use reference data from a single adult male of a certain strain, e.g. Sprague Dawley[14,35] or Wistar[34], Fig. 2b). Alternatively, some atlases use reference data compiled from several subjects representing different features or image orientations, e.g. several rat brains cut in one or all three standard orthogonal planes[37,40]. Reference data may also be acquired by averaging data across many subjects, i.e. by creating a population average constructed from spatially co-registered images[17]. An example of this is the Allen Mouse Brain Atlas Common Coordinate Framework (AMBA CCF)[3,13], generated by averaging 1675 mouse brains acquired by serial two-photon microscopy. The spatial resolution of the reference data determines the level of detail that can be identified. For example, the widely adopted human reference datasets of the Montreal Neurological Institute (MNI)[41,42] are based on averaged magnetic resonance imaging (MRI) scans and represent suitable reference data for macroscopic anatomy, while the single-subject *BigBrain* model[33] provides a reference dataset for identification of cortical layers and more fine-grained cortical and subcortical structures[16].

The *coordinate system* of an atlas provides a framework for specifying locations with origin, units and direction of the axes[43] (Fig. 2c). In brain atlases, the coordinate system origin is often defined using a characteristic feature of the skull, e.g. the *bregma* in a stereotaxic coordinate system[34,35], or a specific anatomical landmark identified within the brain, e.g. the decussation of the anterior commissure in the Talaraich-Tournoux space[44] and Waxholm Space coordinate system[14,45]. The orientation is given by the direction of the axes. For example, the axes of AMBA CCF are directed towards posterior (P), inferior (I) and right (R), giving the orientation PIR (http://help.brain-map.org/display/mousebrain/API). The coordinate system is usually, but not always, a 3D Cartesian coordinate system. Examples of coordinate systems which go beyond a 3D Cartesian system include spatio-temporal systems, with additional time or surface dimensions[46].

The *annotation set* of an atlas consist of graphical marks or labels referring to spatial locations determined by features observed in, inferred from, or mapped onto the reference data, specifying structures or boundaries.

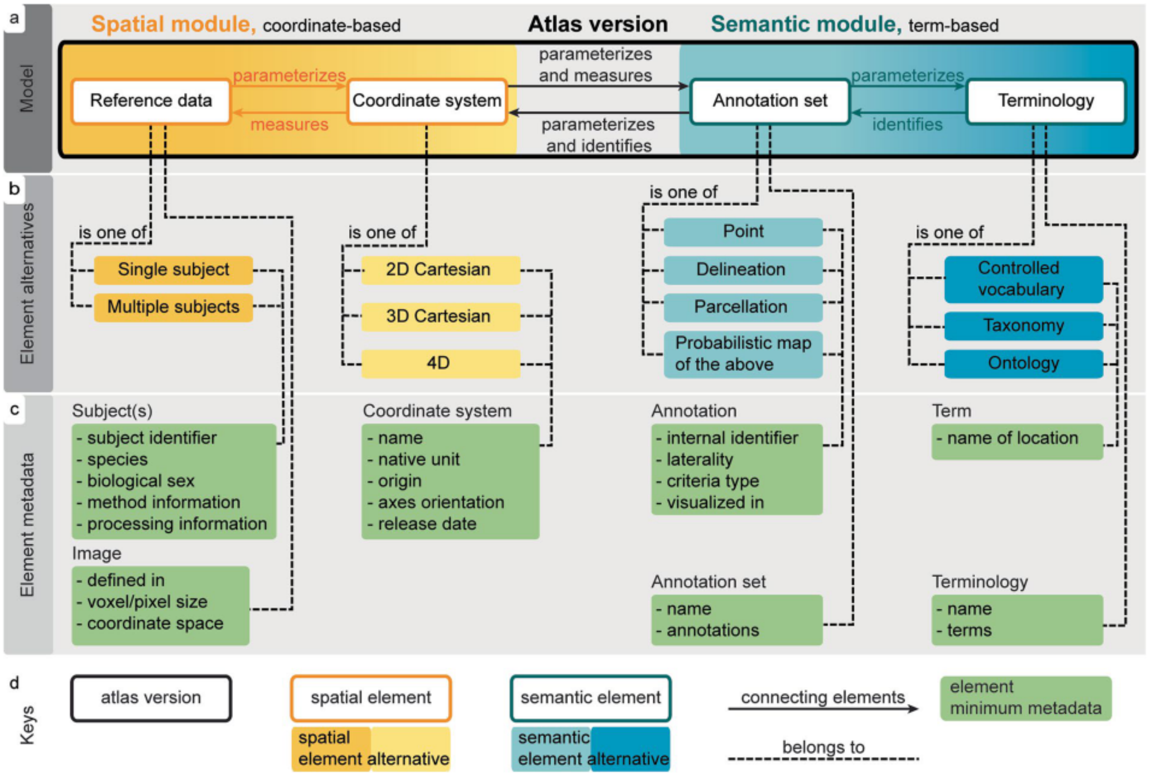


Fig. 2 The relations and metadata of the AtOM elements. (**a**) Diagram illustrating the relations between the AtOM elements: *measures* (to provide a metric to), *parameterizes* (to set the conditions of its operation) and *identifies* (to recognize, establish or verify the identity of something). Thus, the coordinate system measures the reference data and annotation set with coordinates as units. The terminology identifies the annotation set and coordinate system with terms as units. The reference data and the annotation set provide physical dimensions embodying the coordinate system and the terminology. The model consists of two reference modules: *spatial* (containing the coordinate system and reference data, yellow) and *semantic* (containing annotations and terminology, blue). Each element can be one of a set of alternatives (**b**), which have a set of metadata (**c**). (**d**) Key for reading the figure.

An annotation set may identify features-of-interest as points, for example by placing a name or abbreviation on the area of a brain region. Although such annotations can give the user an overview of prominent landmarks and regions in the brain, they are limited in that they do not define the borders of the regions. Thus, most book atlases[34,35] demarcate anatomical boundaries or regions with lines, while 3D brain atlases such as the AMBA CCF v3[3] or WHS rat brain atlas v4[14,47] fully delineate regions with closed curves. In the case of probabilistic maps, coordinates are labeled with the probabilities of a certain region or feature being present at a given location[17,48–50]. Probabilistic maps are typically aggregated from annotations identified in different individuals, encoding variation across subjects[17]. To summarize, an annotation set can consist of points, lines or closed curves, or probabilistic representations of any of these (Fig. 2b).

The *terminology* of an atlas is a set of terms that identifies the annotations, providing human readability and context, and allowing communication about brain locations and structural properties. In its simplest form, a terminology can be a list of unique identifiers, but is typically a set of descriptive anatomical terms following specific conventions. Atlases employ different terms, conventions, and approaches to organize brain structures into systems based on the methodology used to create them as well as their intended use cases. For example, some use developmental organization[51,52], while others use brain systems[39], microstructural organization[16], multimodal features[53], or are specialized for particular brain regions[54,55]. An atlas terminology may be a controlled vocabulary (flat list, e.g. the label file of the Waxholm Space atlas of the Sprague Dawley rat brain), a taxonomy and partonomy (hierarchical list, e.g. the Allen Mouse Reference Atlas Ontology (RRID:SCR_021000)), or an ontology (hierarchy and additional axioms, e.g. that two structures are adjacent).

Relations among the elements. The four elements of AtOM have specific relations (specified in Fig. 2a), sorted into a *spatial module*, consisting of the reference data and the coordinate system (Fig. 2a, yellow), and a *semantic module*, consisting of the annotation set and the terminology (Fig. 2a, blue).

The elements of the *spatial module* provide the physical and measurable dimensions of the atlas. The biological dimensions of the reference data give the conditions of operation for (i.e., *parameterize*) the coordinate system. The coordinate system provides a metric for (i.e., *measures*) the reference data, specifying the origin,

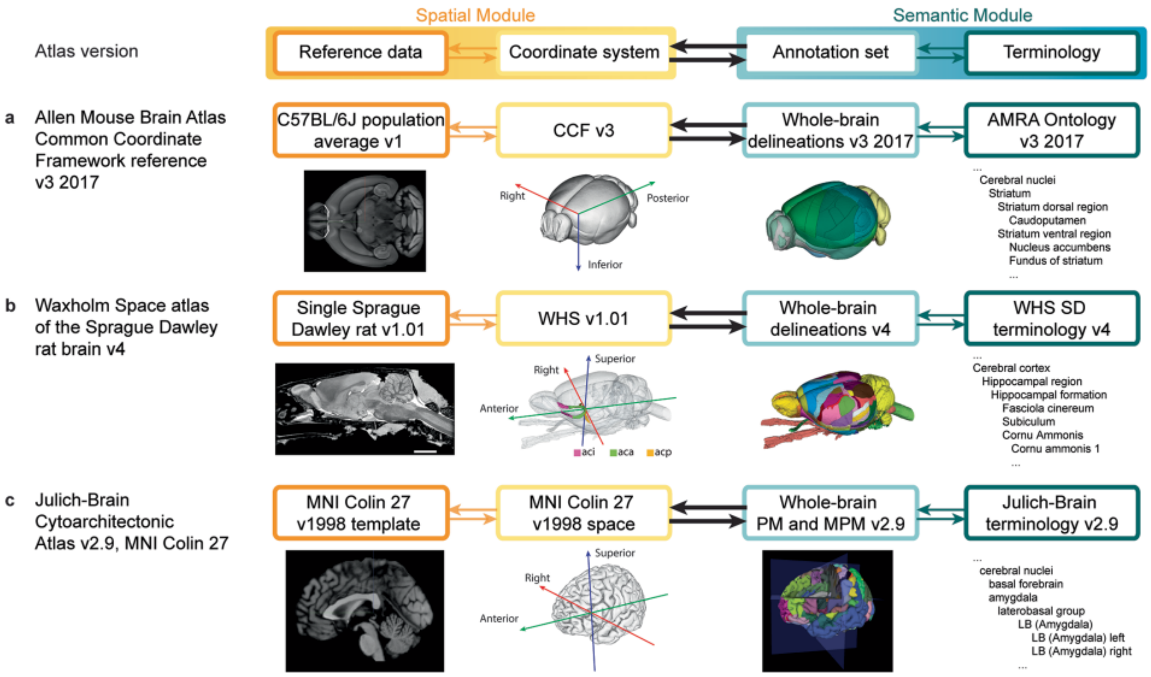


Fig. 3 AtOM elements illustrated for three brain atlas versions. (**a**–**c**) Tabular illustration of the most recent versions of (**a**) the Allen Mouse Brain Atlas Common Coordinate Framework[3], (**b**) the Waxholm Space atlas of the Sprague Dawley rat brain[14] and, (**c**) one alternative representation of the Julich-Brain cytoarchitectonic atlas[17] organized in accordance with the AtOM diagram (top row). All atlases are accessible in the EBRAINS research infrastructure. Specification of the metadata, licenses and versions of these atlases are given in Tables 1, 2. CCF, Common Coordinate Framework; AMRA, Allen Mouse Reference Atlas; WHS, Waxholm Space; MNI, Montreal Neurological Institute; PM, probabilistic maps; MPM, maximum probability maps.

orientation, and units (Fig. 2a). Coordinates are the means to derive measurements, indicate directions, and spatially locate features in the reference data. The coordinate system also *measures* the annotation set, and thus connects the annotations to the features of the reference data. The two spatial elements can be intricately linked, for example through the process of generating the reference data based on multiple subjects. Knowing the detailed information about these links or processing dependencies is not necessarily needed for using an atlas version. However, it is often very useful to have as much metadata and documentation as possible to understand how the two elements are related to each other, especially if one of the elements is changed or when comparing two different atlases to translate information between them.

The elements of the *semantic module* provide semantic identities for the atlas. The annotation set *parameterizes* the terminology in the spatial domain according to or inspired by the reference data. The terminology provides terms to establish the identity of (i.e., *identifies*) each annotation (Fig. 2a). While anatomical terms are not unique identifiers (see Atlas versioning below), they provide a means to semantically address annotations, conveying neuroanatomical knowledge and context. In this way, the terms are semantic units suitable for navigating the atlas annotations, while annotations capture the scholarly interpretations and knowledge underlying the experimental and anatomical criteria used to make them (parcellation criteria). Further, the annotation set propagates the semantic identities from the terminology, and thus semantically *identifies* locations in the coordinate system. The semantic elements may also be linked through the criteria for defining the extent of an annotation, which is often summarized in the name and thus in the terminology. Again, this information is not essential for using an atlas version, but critical for translating information across elements.

The relations of the atlas elements are pathways for translating information between the spatial and semantic modules. A researcher may consult an atlas to observe the physical shape and location associated with a given anatomical term, or to identify the anatomical term assigned to specific coordinates, or biological features observed in the reference data. Thus, the model is a continuous, bidirectional loop providing several starting points for researchers to translate and compare information across atlas elements.

Using AtOM to identify elements in brain atlases and communicate location. AtOM is also readily applied to traditional stereotaxic book atlases[34,35,56–58] as illustrated in the fictive brain atlas in Fig. 1. In principle, a brain atlas can be a set of images with names indicating areas, coordinates for each histological image, and orientation indicators. While the precision of such an atlas might be limited, it can still be versioned and used to communicate reproducible information about brain location.

Figure 3 illustrates how AtOM can be used to identify elements and modules in 3D brain atlases. The reference data for the AMBA CCF v3 2017[3] (Fig. 3a) consists of a population averaged serial two-photon tomography

Full name	**Allen Mouse Brain Atlas Common Coordinate Framework v3 2017**	**Waxholm Space atlas of the Sprague Dawley rat brain v4**	**Julich-Brain Cytoarchitectonic Atlas v2.9, MNI Colin 27**
Short name	AMBA CCF v3 2017	WHS rat brain atlas v4; WHSSDv4	Julich-Brain v2.9, Colin 27
Version identifier	3, 2017	4	2.9, Colin 27
Version innovation	Publication[3]; White paper AMBA CCF v3 2017 (http://help.brain-map.org/display/mouseconnectivity/Documentation)	Publication[14,47]; Webpage (https://www.nitrc.org/projects/whs-sd-atlas)	Publication[17]; EBRAINS datasets[59,60]
Alternative version of	NA	NA	Julich-Brain v2.9, MNI 152; Julich-Brain v2.9, BigBrain; Julich-Brain v2.9, fsaverage
New version of	AMBA CCF v3 2016	WHS rat brain atlas v3.01	Julich-Brain v2.5, Colin 27
Release date	NA	01.10.2021	31.07.2021
Reference data	C57BL/6 J population average v1	Sprague Dawley rat v1.01	MNI Colin27 v1998 template
Coordinate system	CCF v3	WHS v1.01	MNI Colin27 v1998 space
Annotation set	Whole-brain parcellation, v3 2017	Whole-brain parcellation, v4	Whole-brain probabilistic maps and maximum probability maps
Terminology	Allen Mouse Reference Atlas Ontology	WHS SD terminology, v4	Julich-Brain terminology, v2.9
License	Not available, but see legal note (https://alleninstitute.org/legal/citation-policy)	Creative Commons Attribution (CC BY) 4.0	Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) 4.0

Table 1. Overview of metadata and licenses provided with mouse, rat and human brain atlas versions used in the EBRAINS research infrastructure. AMBA, Allen Mouse Brain Atlas; CCF, Common Coordinate Framework; MNI, Montreal Neurological Institute; SD, Sprague Dawley; WHS, Waxholm Space.

(STPT) volume created from 1,675 mice. The coordinate system is the CCF v3, which was created specifically for the Allen Institute mouse brain atlases. The annotation set is the whole-brain delineations from 2017, described in the accompanying white paper (http://help.brain-map.org/display/mouseconnectivity/Documentation), and the terminology is Allen Mouse Reference Atlas Ontology (RRID:SCR_021000). All the version specific metadata for the three atlas versions are listed in Table 1.

Atlas versioning. With an overview of the elements and relations of AtOM at hand, we are now in position to examine how they facilitate clear versioning of an atlas. In AtOM, an atlas version is a concrete instance of an atlas, and consists of specific elements, relations, and metadata (Fig. 2). Figure 3 and Table 1 show the metadata available for the most recent versions of the EBRAINS research infrastructure supported mouse[3], rat[14], and human[17] brain atlases modeled using AtOM. An important consequence of AtOM is that the atlas version changes if there are alterations to any element. Examples of alterations include revising annotations or terms, modifying the reference data or coordinate system, or replacing an element. Such changes have consequences for the specific properties and use of an atlas and should be specified as a new atlas version. The changes made from one version to another can be described in atlas version documentation, and new versions of an atlas are usually distinguished by a new version name. The simplest way to do this is by iterative version numbering. Table 2 shows a complete overview of all versions of the AMBA CCF[3,13], the Waxholm Space atlas of the Sprague Dawley rat brain (WHS rat brain atlas)[14,38,39,47], and selected alternative versions of the Julich-Brain Cytoarchitectonic Atlas (Julich-Brain Atlas)[17]. In the last versions of the AMBA CCF (v3 2015–2017; http://help.brain-map.org/display/mousebrain/Documentation)[3,13,32] and the WHS rat brain atlas (v1.01-v4)[14,31,38,39,47] the semantic elements (annotation set and terminology) have been changed across versions, while the spatial elements (reference data and coordinate system, Table 2) have been kept constant. This continuation across versions allows translation of information and experimental data registered to the reference data are compatible with all versions of the mouse and rat atlas versions.

To clearly reference a specific atlas version or AtOM element, it needs a unique identifier (ID). This is particularly important when combining different versions of elements into alternative atlas versions. The major release v2.9 of the Julich-Brain Atlas (Table 2) has four alternative versions due to its use of four complementary spatial modules: the “MNI Colin 27” (individual specimen, 1 mm resolution), “MNI 152” (population average, 1 mm resolution), “BigBrain” (individual specimen, 20 µm resolution) and “fsaverage” (cortical surface representation)[16,33,59–61]. These alternative versions are identified by combining the major release identifier (v2.9) with the abbreviated name of the respective reference data and coordinate systems. Unique identifiers are also important to differentiate between identical terms, which are often similar, but not identical, anatomical areas within and across species and atlases. Ambiguity can be avoided by indexing atlas version specific terms and providing unique ontology IDs defining their properties and relations. Following AtOM, an atlas version should have unique IDs for each element and their instances, which together with version documentation facilitate clear referencing of atlas versions and specific atlas elements.

Minimum requirements for FAIR brain atlases. Atlases are a type of research data and thus can be evaluated using the foundational principles of the FAIR guidelines[27]. These principles state that data should be findable, accessible, interoperable, and reusable through both human and machine-driven activities. Like experimental data, atlases can support these principles through use of unique identifiers, specific metadata, open protocols, and clear usage licenses. Furthermore, interoperability and reuse of data also

Species	Version number	Atlas version name (semantic ID)	Reference data	Coordinate system	Annotation set	Terminology	Reference(s)

Mouse	1	Allen Mouse Brain Common Coordinate Framework reference atlas v1	C57BL/6 J population average v1	CCF v1	Whole-brain delineations v1	OWL AMBA terminology v1	RRID:SCR_020999;
							http://help.brain-map.org/display/mousebrain/Documentation[32];

	2	Allen Mouse Brain Common Coordinate Framework reference atlas v2		CCF v2	Whole-brain delineations v2	Allen Mouse Reference Atlas Ontology	RRID:SCR_020999;
							RRID:SCR_021000;
							http://help.brain-map.org/display/mousebrain/Documentation[13];

	3	Allen Mouse Brain Common Coordinate Framework reference atlas v3 2015		CCF v3	Whole-brain delineations v3 2015	Allen Mouse Reference Atlas Ontology	RRID:SCR_020999;
							RRID:SCR_021000;
							http://help.brain-map.org/display/mousebrain/Documentation[3];

		Allen Mouse Brain Common Coordinate Framework reference atlas v3 2016			Whole-brain delineations v3 2016	Allen Mouse Reference Atlas Ontology	RRID:SCR_020999;
							RRID:SCR_021000[3];

		Allen Mouse Brain Common Coordinate Framework reference atlas v3 2017			Whole-brain delineations v3 2017	Allen Mouse Reference Atlas Ontology	RRID:SCR_020999;
							RRID:SCR_021000;
							http://help.brain-map.org/display/mouseconnectivity/Documentation[3];

Rat	1	Waxholm Space atlas of the Sprague Dawley rat brain v1	Single Sprague Dawley rat v1	WHS v1	Whole-brain delineations v1	WHS SD terminologyv1	RRID: SCR_017124;
							https://www.nitrc.org/projects/whs-sd-atlas[14];

	1.01	Waxholm Space atlas of the Sprague Dawley rat brain v1.01	Single Sprague Dawley rat v1.01	WHS v1.01	Whole-brain delineations v1.01	WHS SD terminology v1.01	RRID: SCR_017124;
							https://www.nitrc.org/projects/whs-sd-atlas[31];

	2	Waxholm Space atlas of the Sprague Dawley rat brain v2			Whole-brain delineations v2	WHS SD terminology v2	RRID: SCR_017124;
							https://www.nitrc.org/projects/whs-sd-atlas[38];

	3	Waxholm Space atlas of the Sprague Dawley rat brain v3			Whole-brain delineations v3	WHS SD terminology v3	RRID: SCR_017124;
							https://www.nitrc.org/projects/whs-sd-atlas[39];

	3.01	Waxholm Space atlas of the Sprague Dawley rat brain v3.01			Whole-brain delineations v3.01	WHS SD terminology v3.01	RRID: SCR_017124; https://www.nitrc.org/projects/whs-sd-atlas

	4	Waxholm Space atlas of the Sprague Dawley rat brain v4			Whole-brain delineations v4	WHS SD terminology v4	RRID: SCR_017124;
							https://www.nitrc.org/projects/whs-sd-atlas[14,47];

Human*	1.18	Julich-Brain Cytoarchitectonic Atlas v1.18, MNI Colin 27	MNI Colin 27 v1998 template	MNI Colin 27 v1998 space	Whole-brain PM and MPM v1.18	Julich-Brain terminology v1.18	RRID:SCR_023277[78];
		Julich-Brain Cytoarchitectonic Atlas v1.18, MNI 152	MNI ICBM 152 (2009c nonlin asym) template	MNI ICBM 152 (2009c nonlin asym) space			RRID:SCR_023277[78];
		Julich-Brain Cytoarchitectonic Atlas v1.18, BigBrain	BigBrain (v2015) template	BigBrain (v2015) space	High-resolution maps v1.18		RRID:SCR_023277[33];
	2.9	Julich-Brain Cytoarchitectonic Atlas v2.9, MNI Colin 27	MNI Colin 27 v1998 template	MNI Colin 27 v1998 space	Whole-brain PM and MPM v2.9	Julich-Brain terminology v2.9	RRID:SCR_023277[17,59,60];
		Julich-Brain Cytoarchitectonic Atlas v2.9, MNI 152	MNI ICBM 152 (2009c nonlin asym) template	MNI ICBM 152 (2009c nonlin asym) space			RRID:SCR_023277[17,59,60];
		Julich-Brain Cytoarchitectonic Atlas v2.9, BigBrain	BigBrain (v2015) template	BigBrain (v2015) space	High-resolution maps v2.9		RRID:SCR_023277[16,33];
		Julich-Brain Cytoarchitectonic Atlas v2.9, fsaverage	fsaverage surface v1	fsaverage space v1	Surface projections v2.9		RRID:SCR_023277[17,61];

Table 2. Overview of the AtOM elements constituting the mouse, rat and human brain atlas versions currently supported by the EBRAINS research infrastructure. *Only two major releases, each with their alternative versions (representations of the annotation set in different coordinate systems and respective reference data) of the human brain atlas are shown here.

requires use of "formal, accessible, shared, and broadly applicable language for knowledge representation", as well as metadata providing detailed descriptions. Based on our proposed ontology model, we suggest the following set of four minimum requirements for FAIR brain atlases: 1) machine readable digital components, 2) defined spatial and semantic modules with element metadata, 3) specification of element versions with detailed documentation, and 4) defined element relations and metadata (Fig. 1d,e). We elaborate on these requirements below.

First, *machine-readable digital atlas components* imply that all files and metadata are available in open and non-proprietary file formats suitable for direct processing by a machine. This enables programmatic access to critical information about brain atlases without the need to retrieve entire, potentially distributed, datasets. It also makes it possible to incorporate the information into research workflows and software tools, e.g. the siibra tools suite[62,63] for exploring high-resolution atlases such as the multilevel framework established for the human brain (https://ebrains.eu/service/human-brain-atlas) and connecting them to computational workflows. The files and metadata for all the atlas versions shown in Fig. 3 are available online, either on public websites, domain repositories, or at the atlases' respective homepages. Table 1 shows brain atlas version metadata for the four brain atlas versions shown in Fig. 3.

Second, *defined spatial and semantic modules* in an atlas mean that all elements are identifiable and accessible with clear metadata. This makes common elements between atlases or atlas version more comprehensible and facilitates the maintenance of atlases and their versions. At a minimum, this can be clear naming of the essential files or documentation about the location of all necessary information (Table 1). For example, all the files needed for using the WHS rat brain atlas are available via a domain repository (Table 2).

Third, *clear versioning with granular documentation* that state all changes differentiating two version of an atlas are needed to adhere to open science and FAIR principles. Currently this is partially achieved through use of persistent identifiers for atlas releases using either International Standard Book Numbers (ISBN), and Digital Object Identifiers (DOI) or Research Resource Identifiers (RRID)[64]. In addition, atlas reference data are made available as associated files[40], as downloadable internet resources[3,16,17,39], or by providing selected methodological descriptions in publications[14,17]. Some atlases also provide documentation as a list, or as text describing new features or a high-level inventory of changes. Ideally, clear versioning of all atlas elements would enable novice users to quickly identify the differences between two versions (Table 2).

Fourth, the *explicit relations between atlas elements*, such as parcellation criteria and coordinate system definitions, provide an empirical foundation for translating information across the elements. This allows users to connect data to different atlas elements (semantic or spatial), and enables automated search or comparison of data based on atlas elements. Traditionally, methodological information is mainly presented in a human-readable format through publications[14,17], white papers or via a webpage, but it is now possible to document information in machine-readable, structured formats following standards, e.g. as single or distributed data publications[60] (Table 2).

Brain atlases that fulfill these four requirements are thus expected to be sufficiently well defined to be incorporated into research workflows and enable automated transfer of information across atlases. The advantage of AtOM can be demonstrated with a concrete scenario where a researcher wants to create a modified version of an atlas to adapt the granularity of the brain annotations to their data. For example, in the following publication[65] they used the hierarchical terminology to group selected brain regions of the AMBA CCF v3 2017 into larger custom regions and thus create a custom brain atlas version for their analysis. This was possible as the annotation set, terminology and metadata were readily available and identified (according to AtOM) and allowed the researchers to create and cite the changes in their custom atlas version. Another potential advantage of having individual atlas elements provided as separate files is that they may be used as exchangeable components in viewers or analysis tools such as siibra-explorer[63] or siibra-python[62]. This allows for comparative analysis or re-analysis using different atlas versions[47].

Discussion

We have identified spatial and semantic elements of brain atlases, defined their relations, and created an Atlas Ontology Model (AtOM), specifying human and machine-readable metadata. Even though the AtOM elements are readily recognized in different atlases, they are often named according to traditions or common practice. For example, the reference data and the coordinate system are often considered as one entity, and referred to as the common coordinate space, reference template, reference space, brain model or atlas[9,42]. The term atlas is variably used to address reference data, an atlas version, any of a series of atlas versions or the annotation set. The annotation set, often in combination with the terminology, has also been called parcellations, segmentations or delineations[16,17,39,48].

Some of the AtOM elements have been suggested earlier[9], as well as similar approaches to versioning and atlas organization[17]. However, AtOM is the first model for standardizing the common elements of any brain reference atlas, their definitions, and metadata, creating a standard to organize and share information about atlases or as a template to create an atlas.

When implemented, AtOM will facilitate precise and unique referencing of parts of an atlas, as well as the incorporation of atlases in digital tools or workflows. AtOM further provides a basis for specifying minimum requirements for brain atlases to comply with the FAIR principles. Below, we discuss how AtOM may contribute to increase interoperability among atlases, enable more standardized use of brain atlases in computational tools, and advance FAIR data sharing in neuroscience. Interoperable atlases allow for exchange and translation of information across atlases, tools and data. Experimental data generated by different researchers typically relate to an atlas via spatial coordinates or anatomical terms, often defined by visual comparison of images or use of other observations such as measurements of functional properties. Researchers translate between the semantic and spatial location information using human readable metadata. At the same time, automated translation can be enabled via standardized, machine-readable files specifying properties and relations among atlas elements. The translation of information is dependent on interoperability across atlas elements, which can be specified at three levels: practical, technical, and scholarly.

At the *practical level*, translation of information across atlas elements is essential for interpretation and communication of anatomical locations, such as relating machine-readable coordinates to human-readable brain structure names. The relations specified between atlas elements and the defining metadata allow comparisons of annotations and terminologies across atlases representing different species or strains, developmental stages, or disease states. By aligning reference data or coordinate systems of two different atlases, information can be directly compared or translated. For example, aligned brain region annotations can be inspected and their respective terminology aligned, establishing a semantic translation across two atlas terminologies[66]. Alternatively, terminology and annotations from different atlases may be combined, as was demonstrated when creating a new unified mouse brain atlas by adopting the semantic elements from the Franklin and Paxinos mouse brain atlas into the AMBA CCF[67]. However, it is important to keep in mind that reproducible use of atlas resources depends on unambiguous citation of atlases and their versions. When the atlas version reference is ambiguous, or if anatomical names are given without specification of the employed atlas version terminology, it is difficult to compare location between datasets[11]. Versioning, documentation, and clear references are therefore essential for atlases that change over time.

At a *technical level*, atlas information can be accessed using computational tools, requiring specification of essential parameters and versions, such as file formats and other technical metadata. Atlases that have closed proprietary file formats may technically be digital, but without being fully machine accessible and interoperable, they are difficult to utilize in analytic tools and infrastructures.

At a *scholarly level*, anatomical parcellation and terminology should be comparable across atlases. The lack of consensus about terminologies, parcellation schemata, and boundary criteria among neuroanatomists is a major challenge for the development, use, and comparison of brain atlases[68–75]. Following different traditions, knowledge, and criteria, both domain experts and non-expert researchers may inevitably convey subjective and sometimes irreproducible information that is difficult to document. AtOM provides a foundation for organizing and communicating specific information about brain atlases in a standardized way that allows researchers to describe their interpretations more precisely, and thus contribute to increased reproducibility of results.

The value of interoperable atlases is substantial, allowing data integration, analysis and communication based on anatomical location. Brain atlases incorporated in various analytical tools open the possibility for efficient approaches to analyzing, sharing, and discovering data. For example, by analyzing images mapped to an atlas, the atlas information can be used to assign coordinates and terms to objects-of-interest[10,76]. Data from different publications analyzed with the same atlas are comparable, and data registered to the spatial module (reference data and coordinate system) of an atlas may also be re-analyzed with new or alternative annotation sets. Perhaps more importantly, by specifying the AtOM elements as standardized machine readable files, it becomes possible to incorporate different atlases as exchangeable modules in analytic tools and infrastructure systems[20–22,25,26]. Tools and systems using interoperable atlases can exploit the defined relations among the elements for automated operations, like data queries, calculations, or assignment of location identity to experimental data that have been associated with an atlas by spatial registration or semantic identification.

AtOM is used by multiple research and infrastructure groups, and is part of the Neuroscience Information Framework (RRID:SCR_002894), see Methods, and the openMINDS metadata framework for neuroscience graph databases (RRID:SCR_023173; https://github.com/HumanBrainProject/openMINDS). In particular, AtOM has served as base for the openMINDS SANDS extension (RRID:SCR_023498) which is focusing on the spatial anchoring of neuroscience data structures and includes the provision of controlled graph database descriptions for brain atlas and common coordinate spaces. openMINDS defines the semantic architecture of the EBRAINS Knowledge Graph. Other EBRAINS services, such as the EBRAINS Atlases (https://ebrains.hbp.eu/services/atlases) rely on openMINDS to robustly query for relevant data and correctly represent brain atlases and common coordinate spaces. The multilevel human brain atlas, an atlas framework that spans across multiple spatial scales and modalities hosted on the EBRAINS research infrastructure, exemplifies how several reference data, coordinate systems, and annotation set, developed over time, can be seamlessly incorporated, and presented to users through a single viewer tool. A growing repertoire of tools, services, and workflows within and outside of the EBRAINS research infrastructure rely on formal descriptions for automated incorporation of research products, including brain atlases and common coordinate spaces. AtOM provides a framework for keeping track of the complex relations among these resources and research products.

In conclusion, the primary value of AtOM is that it establishes a standardized framework for developers and researchers using brain atlases to create, use, and refer to specific atlas elements and versions. Atlas developers can use the model to create clearly citable and interoperable atlases. For developers incorporating atlases in tools, AtOM defines atlas elements as modules that can be seamlessly exchanged to accommodate atlases for other species or developmental stages, or to switch between versions, coordinate systems, or terminologies. By standardizing the communication and use of fundamental reference resources, we are convinced that AtOM will accelerate efficient analysis, sharing and reuse of neuroscience data.

Methods

The first draft of AtOM (at the time called parcellation.ttl[77]) was developed by eliciting requirements and use cases from the Blue Brain Project (https://github.com/SciCrunch/NIF-Ontology/issues/49). To ingest atlas terminologies into the NIF standard ontology (RRID:SCR_005414) following AtOM, a python module (https://github.com/tgbugs/pyontutils/tree/master/nifstd/nifstd_tools/parcellation) was written to convert from a variety of formats into Web Ontology Language (OWL). An initial version of the core ontology and 24 atlas terminologies were created. These ontologies were loaded into SciGraph (RRID:SCR_017576; https://github.com/SciGraph/SciGraph) and queries (https://github.com/SciCrunch/sparc-curation/blob/67b534a939e2a271050c6edad97c707d8ec075d3/resources/scigraph/cypher-resources.yaml#L51-L267) were then written against the original data model using the Cypher query language to find atlases, terminologies, and individual terms for specific atlases, species, and developmental stages. These queries have been used in production systems for over 4 years. During this time additional atlases were ingested using the python module (now totaling 40) and an initial draft of the conceptual model for AtOM was developed (https://github.com/SciCrunch/NIF-Ontology/blob/master/docs/brain-regions.org). For a full record of the iterative development of the model to fully distinguish the major elements found in the current version (though not under their current names) see https://github.com/SciCrunch/NIF-Ontology/issues/49.

A second round of development involved further requirements collection in the context of atlas creation and the conceptual model was heavily revised, regularized, and extended in the context of the needs of the Human Brain Project (HBP) (https://github.com/SciCrunch/NIF-Ontology/commits/64c32abed9963073fab90dd5901d806fd8503da2 commit history from work during the HBP meeting in Oslo in November 21-22 2019) and the Allen Institute for Brain Sciences (https://github.com/SciCrunch/NIF-Ontology/commit/a40a8c786529f5b2e2a3a8007776d057c5830d2d, other interactions occurred, but do not have public records of their occurrence). Various iterations of the model were applied to a wide variety of atlases and atlas-like things, such as paper and digital atlases, ontologies, figures from publications, crudely drawn diagrams on table cloths, globes, geographic information systems, traditional cartographic maps, topological maps of the peripheral nervous system, and more. This was followed by collection of requirements and live ontology development carried out in the context of the HBP, which included alignment with the schemas of the openMINDS SANDS (RRID:SCR_023498) metadata model for reporting spatial metadata. The resulting ontological model was applied to a number of existing atlases, specifically the WHS rat brain atlas (RRID:SCR_017124)[14,38,39], the AMBA CCF (RRID:SCR_020999) v3[3,13], and the human Julich-Brain atlas (RRID:SCR_023277)[17,61].

Data availability

AtOM (RRID:SCR_023499) is publicly available via GitHub: https://github.com/SciCrunch/NIF-Ontology/blob/atlas/ttl/atom.ttl. The ontology is available via BioPortal: http://purl.bioontology.org/ontology/ATOM. The 1.0 release of AtOM that corresponds to this paper is available via GitHub at https://github.com/SciCrunch/NIF-Ontology/releases/tag/atom-1.0.

Code availability

Python code for generating parcellations for the NIF-Ontology is publicly available via GitHub: https://github.com/tgbugs/pyontutils/tree/master/nifstd/nifstd_tools/parcellation. Archives of release are available via Zenodo[77].

Received: 16 November 2022; Accepted: 14 July 2023;
Published online: 26 July 2023

References

1. Bjaalie, J. Localization in the brain: new solutions emerging. *Nat. Rev. Neurosci.* **3**, 322–325 (2002).
2. Sunkin, S. & Hohmann, J. Insights from spatially mapped gene expression in the mouse brain. *Hum. Mol. Genet.* **16**, R209–R219 (2007).
3. Wang, Q. *et al.* The Allen Mouse Brain Common Coordinate Framework: A 3D Reference Atlas. *Cell* **181**, 1–18 (2020).
4. Börner, K. *et al.* Anatomical structures, cell types and biomarkers of the Human Reference Atlas. *Nat. Cell Biol.* **23**, 1117–1128 (2021).
5. Nowinski, W. Evolution of Human Brain Atlases in Terms of Content, Applications, Functionality, and Availability. *Neuroinformatics* **19**, 1–22 (2021).
6. Osumi-Sutherland, D. *et al.* Cell type ontologies of the Human Cell Atlas. *Nat. Cell Biol.* **23**, 1129–1135 (2021).
7. Tyson, A. & Margrie, T. Mesoscale microscopy and image analysis tools for understanding the brain. *Prog. Biophys. Mol. Biol.* **168**, 81–93 (2022).
8. Newmaster, K., Kronman, F., Wu, Y. & Kim, Y. Seeing the Forest and Its Trees Together: Implementing 3D Light Microscopy Pipelines for Cell Type Mapping in the Mouse Brain. *Front. Neuroanat.* **15**, 1–19 (2022).
9. Amunts, K. *et al.* Interoperable atlases of the human brain. *Neuroimage* **99**, 525–532 (2014).
10. Bjerke, I. *et al.* Data integration through brain atlasing: Human Brain Project tools and strategies. *Eur. Psychiatry* **50**, 70–76 (2018).
11. Bjerke, I. *et al.* Navigating the Murine Brain: Toward Best Practices for Determining and Documenting Neuroanatomical Locations in Experimental Studies. *Front. Neuroanat.* **12**, 1–15 (2018).
12. Feo, R. & Giove, F. Towards an efficient segmentation of small rodents brain: A short critical review. *J. Neurosci. Methods* **323**, 82–89 (2019).
13. Oh, S. *et al.* A mesoscale connectome of the mouse brain. *Nature* **508**, 207–214 (2014).
14. Papp, E., Leergaard, T., Calabrese, E., Johnson, G. & Bjaalie, J. Waxholm Space atlas of the Sprague Dawley rat brain. *Neuroimage* **97**, 374–386 (2014).
15. Woodward, A. *et al.* The Brain/MINDS 3D digital marmoset brain atlas. *Sci. Data* **5**, 180009 (2018).
16. Wagstyl, K. *et al.* BigBrain 3D atlas of cortical layers: Cortical and laminar thickness gradients diverge in sensory and motor cortices. *PLOS Biol.* **18**, e3000678 (2020).
17. Amunts, K., Mohlberg, H., Bludau, S. & Zilles, K. Julich-Brain: A 3D probabilistic atlas of the human brain's cytoarchitecture. *Science* **369**, 988–992 (2020).
18. Vandenberghe, M. *et al.* High-throughput 3D whole-brain quantitative histopathology in rodents. *Sci. Rep.* **6**, 20958 (2016).
19. Fürth, D. *et al.* An interactive framework for whole-brain maps at cellular resolution. *Nat. Neurosci.* **21**, 139–149 (2018).
20. Puchades, M., Csucs, G., Ledergerber, D., Leergaard, T. & Bjaalie, J. Spatial registration of serial microscopic brain images to three-dimensional reference atlases with the QuickNII tool. *PLoS One* **14**, e0216796 (2019).
21. Yates, S. *et al.* QUINT: Workflow for Quantification and Spatial Analysis of Features in Histological Images From Rodent Brain. *Front. Neuroinform.* **13**, 1–14 (2019).
22. Groeneboom, N., Yates, S., Puchades, M. & Bjaalie, J. Nutil: A Pre- and Post-processing Toolbox for Histological Rodent Brain Section Images. *Front. Neuroinform.* **14**, 37 (2020).
23. Pallast, N., Wieters, F., Fink, G. & Aswendt, M. Atlas-based imaging data analysis tool for quantitative mouse brain histology (AIDAhisto). *J. Neurosci. Methods* **326**, 108394 (2019).

24. Bjerke, I. *et al*. Densities and numbers of calbindin and parvalbumin positive neurons across the rat and mouse brain. *iScience* **24**, 1–20 (2021).
25. Newmaster, K. *et al*. Quantitative cellular-resolution map of the oxytocin receptor in postnatally developing mouse brains. *Nat. Commun.* **11**, 1–12 (2020).
26. Attili, S., Silva, M., Nguyen, T. & Ascoli, G. Cell numbers, distribution, shape, and regional variation throughout the murine hippocampal formation from the adult brain Allen Reference Atlas. *Brain Struct. Funct.* **224**, 2883–2897 (2019).
27. Wilkinson, M. *et al*. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016).
28. Amunts, K. & Zilles, K. Architectonic Mapping of the Human Brain beyond Brodmann. *Neuron* **88**, 1086–1107 (2015).
29. Guarino, N. Formal ontology, conceptual analysis and knowledge representation. *Int. J. Hum. Comput. Stud.* **43**, 625–640 (1995).
30. Chandrasekaran, B., Josephson, J. & Benjamins, V. What are ontologies, and why do we need them? *IEEE Intell. Syst.* **14**, 20–26 (1999).
31. Papp, E., Leergaard, T., Calabrese, E., Johnson, G. & Bjaalie, J. Addendum to "Waxholm Space atlas of the Sprague Dawley rat brain" [NeuroImage 97 (2014) 374–386]. *Neuroimage* **105**, 561–562 (2015).
32. Lein, E. *et al*. Genome-wide atlas of gene expression in the adult mouse brain. *Nature* **445**, 168–176 (2007).
33. Amunts, K. *et al*. BigBrain: An Ultrahigh-Resolution 3D Human Brain Model. *Science* **340**, 1472–1475 (2013).
34. Paxinos, G. & Watson, C. *The rat brain in stereotaxic coordinates*. (Academic Press, 1982).
35. Swanson, L. *Brain Maps: Structure of the rat brain*. (Elsevier, 1992).
36. Paxinos, G., Watson, C., Calabrese, E., Badea, A. & Johnson, G. *MRI/DTI Atlas of the Rat Brain*. (Academic Press, 2015).
37. Swanson, L. Brain maps 4.0-Structure of the rat brain: An open access atlas with global nervous system nomenclature ontology and flatmaps. *J. Comp. Neurol.* **526**, 935–943 (2018).
38. Kjonigsen, L., Lillehaug, S., Bjaalie, J., Witter, M. & Leergaard, T. Waxholm Space atlas of the rat brain hippocampal region: Three-dimensional delineations based on magnetic resonance and diffusion tensor imaging. *Neuroimage* **108**, 441–449 (2015).
39. Osen, K., Imad, J., Wennberg, A., Papp, E. & Leergaard, T. Waxholm Space atlas of the rat brain auditory system: Three-dimensional delineations based on structural and diffusion tensor magnetic resonance imaging. *Neuroimage* **199**, 38–56 (2019).
40. Paxinos, G. & Watson, C. *The Rat Brain in Stereotaxic Coordinates*. (Academic Press, 2018).
41. Fonov, V. *et al*. Unbiased average age-appropriate atlases for pediatric studies. *Neuroimage* **54**, 313–327 (2011).
42. Evans, A., Janke, A., Collins, D. & Baillet, S. Brain templates and atlases. *Neuroimage* **62**, 911–922 (2012).
43. Kleven, H. *et al*. A neuroscientist's guide to using murine brain atlases for efficient analysis and transparent reporting. *Front. Neuroinform.* **17**, 1–8 (2023).
44. Talairach, J. & Tournoux, P. *Co-planar stereotaxic atlas of the human brain: 3-Dimensional proportional system: An approach to cerebral imaging*. (Thieme Medical Publishers, Inc., 1988).
45. Johnson, G. *et al*. Waxholm Space: An image-based reference for coordinating mouse brain research. *Neuroimage* **53**, 365–372 (2010).
46. Dale, A., Fischl, B. & Sereno, M. Cortical Surface-Based Analysis. *Neuroimage* **9**, 179–194 (1999).
47. Kleven, H. *et al*. Waxholm Space atlas of the rat brain: A 3D atlas supporting data analysis and integration. *Res. Sq.* 1–25, https://doi.org/10.21203/rs.3.rs-2466303/v1 (2023).
48. Dadi, K. *et al*. Fine-grain atlases of functional modes for fMRI analysis. *Neuroimage* **221**, 117126 (2020).
49. López-López, N. *et al*. From Coarse to Fine-Grained Parcellation of the Cortical Surface Using a Fiber-Bundle Atlas. *Front. Neuroinform.* **14**, 1–22 (2020).
50. Fan, L. *et al*. The Human Brainnetome Atlas: A New Brain Atlas Based on Connectional Architecture. *Cereb. Cortex* **26**, 3508–3526 (2016).
51. Valverde, F. *Golgi atlas of the postnatal mouse brain*. (Springer, 1998).
52. Altman, J. & Bayer, S. *Atlas of prenatal rat brain development*. (CRC Press, 1995).
53. Glasser, M. *et al*. A multi-modal parcellation of human cerebral cortex. *Nature* **536**, 171–178 (2016).
54. Boccara, C. *et al*. A three-plane architectonic atlas of the rat hippocampal region. *Hippocampus* **00**, 1–20 (2015).
55. Olsen, G. & Witter, M. Posterior parietal cortex of the rat: Architectural delineation and thalamic differentiation. *J. Comp. Neurol.* **524**, 3774–3809 (2016).
56. Paxinos, G. & Franklin, K. B. J. *The Mouse Brain in Stereotaxic Coordinates*. (Academic Press, 2012).
57. Morel, A. *Stereotactic Atlas of the Human Thalamus and Basal Ganglia*. (CRC Press, 2007).
58. Mai, J., Majtanik, M. & Paxinos, G. *Atlas of the Human Brain*. (Academic Press, 2016).
59. Amunts, K. *et al*. Julich-Brain Atlas - whole-brain collections of cytoarchitectonic probabilistic maps (v2.9). *EBRAINS* https://doi.org/10.25493/46HK-XMM (2021).
60. Amunts, K. *et al*. Whole-brain parcellation of the Julich-Brain Cytoarchitectonic Atlas (v2.9). *EBRAINS* https://doi.org/10.25493/VSMK-H94 (2021).
61. Mangin, J., Rivière, D. & Amunts, K. Surface projections of Julich-Brain cytoarchitectonic maps (v2.9). *EBRAINS* https://doi.org/10.25493/NZGY-6AS (2021).
62. Dickscheid, T. *et al*. siibra-python - Software interface for interacting with brain atlases. *ZENODO* https://doi.org/10.5281/ZENODO.7885728 (2023).
63. Gui, X., Gogshelidze, D., Chervakov, P., Amunts, K. & Dickscheid, T. siibra-explorer - Interactive web viewer for multilevel brain atlases. *ZENODO* https://doi.org/10.5281/zenodo.7885733 (2023).
64. Bandrowski, A. *et al*. The Resource Identification Initiative: a cultural shift in publishing. *Brain Behav.* **6**, e00417 (2016).
65. Gurdon, B. *et al*. Detecting the effect of genetic diversity on brain composition in an Alzheimer's disease mouse model. *bioRxiv* https://doi.org/10.1101/2023.02.27.530226 (2023).
66. Bjerke, I., Puchades, M., Bjaalie, J. & Leergaard, T. Database of literature derived cellular measurements from the murine basal ganglia. *Sci. Data* **7**, 1–14 (2020).
67. Chon, U., Vanselow, D., Cheng, K. & Kim, Y. Enhanced and unified anatomical labeling for a common mouse brain atlas. *Nat. Commun.* **10**, 5067 (2019).
68. Bota, M. & Swanson, L. 1st INCF Workshop on Neuroanatomical Nomenclature and Taxonomy. *Nat. Preced.* 12–17, https://doi.org/10.1038/npre.2008.1780.1 (2008).
69. Hawrylycz, M. *et al*. The INCF Digital Atlasing Program: Report on Digital Atlasing Standards in the Rodent Brain. *Nat. Preced.* https://doi.org/10.1038/npre.2009.4000 (2009).
70. Bohland, J., Bokil, H., Allen, C. & Mitra, P. The Brain Atlas Concordance Problem: Quantitative Comparison of Anatomical Parcellations. *PLoS One* **4**, e7200 (2009).
71. Azimi, N., Yadollahikhales, G., Argenti, J. & Cunningham, M. Discrepancies in stereotaxic coordinate publications and improving precision using an animal-specific atlas. *J. Neurosci. Methods* **284**, 15–20 (2017).
72. Khan, A., Perez, J., Wells, C. & Fuentes, O. Computer Vision Evidence Supporting Craniometric Alignment of Rat Brain Atlases to Streamline Expert-Guided, First-Order Migration of Hypothalamic Spatial Datasets Related to Behavioral Control. *Front. Syst. Neurosci.* **12**, 1–29 (2018).
73. Van De Werd, H. & Uylings, H. Comparison of (stereotactic) parcellations in mouse prefrontal cortex. *Brain Struct. Funct.* **219**, 433–459 (2014).

74. Laubach, M., Amarante, L., Swanson, K. & White, S. What, If Anything, Is Rodent Prefrontal Cortex? *eNeuro* **5**, ENEURO.0315-18.2018 (2018).
75. Mai, J. & Majtanik, M. Toward a Common Terminology for the Thalamus. *Front. Neuroanat.* **12**, 1–23 (2019).
76. Bjerke, I., Yates, S., Puchades, M., Bjaalie, J. & Leergaard, T. Brain-wide quantitative data on parvalbumin positive neurons in the rat. *EBRAINS* https://doi.org/10.25493/KR92-C33 (2020).
77. Gillespie, T. *et al*. tgbugs/pyontutils: neurondm-0.1.5 (neurondm-0.1.5). *ZENODO*. https://doi.org/10.5281/zenodo.7946734 (2023).
78. Amunts, K., Eickhoff, S., Caspers, S., Bludau, S. & Mohlberg, H. Whole-brain parcellation of the Julich-Brain Cytoarchitectonic Atlas (v1.18). *EBRAINS* https://doi.org/10.25493/8EGG-ZAR (2019).
79. Mikula, S., Trotts, I., Stone, J. & Jones, E. Internet-enabled high-resolution brain mapping and virtual microscopy. *Neuroimage* **35**, 9–15 (2007).

Acknowledgements

The present work builds on our earlier contributions to development of brain atlases, neuroinformatics and ontologies in neuroscience with contributions from many researchers. The inspiration for developing the Atlas Ontology Model came through fruitful and valuable discussions at the INCF Workshop on Digital Brain Atlasing in Warzaw, 2019. We particularly thank Michael Hawrylycz, Alexander Woodward, Rembrandt Bakker, Ingvild E. Bjerke, Martin Øvsthus, Ulrike Schlegel, Louisa Köhnen, Xiao Gui, and Camilla Blixhavn for valuable discussions during the different stages of developing the Atlas Ontology Model. This work was funded by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2) and Specific Grant Agreement No. 945539 (Human Brain Project SGA3), The Research Council of Norway under Grant Agreement No. 269774 (INCF Norwegian Node, to J.G.B. and T.B.L.), and the Helmholtz Association's Initiative and Networking Fund through the Helmholtz International BigBrain Analytics and Learning Laboratory (HIBALL) under the Helmholtz International Lab grant agreement InterLabs-0015 (to T.D.).

Author contributions

H.K. and T.H.G. contributed equally. H.K. contributed to conceiving the study, establishing and validating the model, writing the paper, and creating figures. T.H.G. contributed to conceiving the study, establishing and validating the model, creating and maintaining the ontology, writing the paper, and creating figures. L.Z. contributed to establishing and validating the model, and writing the paper. T.D. contributed to establishing and validating the model, and writing the paper. J.G.B. contributed to establishing and validating the model, and writing the paper. M.E.M. contributed to conceiving the study, establishing and validating the model, writing the paper, and supervising the study. T.B.L. contributed to conceiving the study, establishing and validating the model, writing the paper, and supervising the study.

Competing interests

M.E.M. is the founder and has equity interest in SciCrunch Inc, a tech start-up out of UCSD that provides tools and services in support of reproducible science and Research Resource Identifiers. J.G.B. is a member of the Management Board of the EBRAINS AISBL, Brussels, Belgium. The other authors declare that no competing interests or conflicts of interest exist for any of the authors.

Additional information

Correspondence and requests for materials should be addressed to T.B.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgments

Chapter 2 of this dissertation is a reprint that appeared in full as ”AtOM, an ontology model to standardize use of brain atlases in tools, workflows, and data infrastructures,” Heidi Kleven, Thomas H. Gillespie, Lyuba Zehl, Tim Dickscheid, Jan G. Bjaalie, Maryann E. Martone, and Trygve B. Leergaard, Scientific Data 10:486 2023 [55]. The dissertation author was the primary investigator and co-first author of this paper along with Heidi Kleven.

Chapter 3

`protc/ur`, a formal language for experimental protocols

3.1 Introduction

Scientific protocols are key information artifacts produced and used by working experimental scientists [28]. In the simplest sense they are documents that are used as mnemonics to guide and constrain scientific processes. However, until quite recently [96] they were rarely published as first class scholarly artifacts[1]. From some perspectives protocols are the single most important document for understanding scientific results, more so even than data or papers, because if a protocol is flawed everything after it is at risk of being uninterpretable [58, 78].

Before delving further into protocols we must briefly discuss methods sections. Protocols and methods sections are not the same thing. Protocols must exist prior in time to the collection of data so that they can have been followed to perform an experiment. On the other hand, methods sections tend to be written retrospectively after an experiment is completed. They are distinct but

[1] Nature Protocols (`https://www.nature.com/nprot`) has existed since 2006, however it publishes about 190 protocols a year which is orders of magnitude lower than the number of papers published, even assuming protocols could be reused between papers. Given the number of labs conducting research and the number of protocols they are actively using, the bandwidth needed would be far beyond traditional publishing.

complementary. Much could be said about the insufficiency of methods sections for capturing and communicating the practical knowledge of how to perform a protocol, but that is not the role that methods sections are intended to play. Historically, such practical knowledge has been transmitted by apprenticeship in a lab [60, 59, 22]. Reconstructing a protocol that can be used to perform an experiment from a methods section can be a substantial undertaking. Witness the repeated expansion of single sentences from a methods section into whole paragraphs from another in Figure 3.1. Before the advent of modern information systems (and likely even after) it would have been faster and easier to travel to the lab that knew the technique and learn it from them.

Methods sections are critical for evaluating the validity of scientific results, and issues with reporting of methods are well known [34, 1, 79]. While there are efforts under way to improve the quality of methods reporting [66, 5, 75, 89], such effort would likely benefit from more widely published and detailed protocols. Much of the information missing in methods reporting must unavoidably be collected in order to perform experiments, if it were documented in a protocol then authors could simply reference the protocol. However, this is not always the case, and there are variables that cannot be known ahead of time and must be recorded after the creation of a protocol. These are often the key information in a methods section and are why we say that protocols and methods sections are complementary.

Given the critical role that protocols play in scientific research, they present an ideal target for formalization. One approach to this has been the creation of electronic lab notebooks (ELNs). However, despite numerous efforts to create ELNs most protocols are still hand-written in paper lab notebooks [51]. At the other end of the spectrum are Laboratory Information Management Systems (LIMS) [97]. LIMS are complex software systems that employ the full power of modern databases and software tools to manage the entire research life cycle. However, the complexity and scale of most LIMS make them difficult for individual academic research labs to operate and maintain, and the cost of subscribing to commercial offerings can be prohibitive [80, 10].

Somewhere between these two ends of the spectrum are tools like protocols.io that focus on providing a way to write and share structured natural language protocols [96].

One approach that could be employed to formalize scientific protocols, that falls between the two extremes of natural language with all its complexities and full fledged modern software systems with their own set of complexities, is to create a formal language [83]. Formal languages have enough flexibility to express scientific protocols and could be designed to avoid the kind of friction that can occur when trying to use a more general formal language such as Structured Query Language (SQL) directly [90, 68]. At the same time, a formal language could provide sufficient structure to make the contents of protocols machine-readable and precise in ways that natural language cannot guarantee [14, 82].

For these, and many other reasons, we set out to create a formal language for protocols. Our objectives in the creation of a formal language for scientific protocols are that a maintained implementation of such a language exists, that it be practically useful, that it might even be able to make life a little bit easier for experimental scientists, and that formal scientific protocols might one day be able to provide a foundation for automatically performing experiments and for improving our ability to assess the methodology used to produce scientific results by automatically validating parts of protocols. While the last two goals remain in the future, we think we have made substantial progress toward the first two and report on that progress here.

The formalization of scientific protocols is not a new idea. In the more general domain of process specification, attempts to formalize natural language specifications have a history going back at least to Specification Acquisition from Experts (SAFE) [35]. One of the most fully developed approaches to modeling processes both as a language and as an ontology [39] is the Process Specification Language (PSL) [86, 85, 38, 9]. However, despite becoming an ISO standard (ISO 18629)[2] there does not appear to be a practical implementation of PSL.

Remaining in the realm of slightly more general approaches, there are a number of for-

[2] https://www.iso.org/standard/35431.html

mal ontologies that have been created to model scientific experiments. In this context the Ontology for Biomedical Investigations (OBI) and the Experimental Factors Ontology (EFO) are well known and widely used inside knowledge management systems [12, 4, 65]. Another example is the PROV Ontology, which was not originally designed to model scientific protocols but has been used to capture metadata about scientific processes [74, 71, 43]. Focusing on the protocol document, there is an existing body of work on the semantic representation of protocols (SMART protocols) [32, 33, 31]. Most of these ontologies are implemented in the Web Ontology Language version 2 (OWL) [44] and are excellent for use in knowledge management systems but can be difficult for users who are not already knowledge engineers to apply directly to, e.g., write a protocol [68]. Thus, while ontologies are certainly a useful or even critical part of the larger system to support formal protocols, they themselves are not a particularly effective interface for trying to help experimental scientists express protocols.

Moving to focus on protocols themselves, the work of Kwasnikowska, et al. presents theoretical foundations for formal scientific protocols and their relation to information systems [57]. Although we did not implement their exact formalism, the insights they have provided are valuable to anyone interested in a more mathematical treatment of modeling protocols. Unfortunately, the practical implementation of the system named ProtocolDB described in that and later work (e.g., [54]) appears to have been lost to time [3].

There has also been work to use formal protocols as a way to automatically capture and structure experimental data and metadata. For example, the work on Knowledge Engineering from Experimental Design (KEfED) has shown that structured protocols can be used to automatically capture data from experiments [84, 94]. The underlying model of protocols proposed in KEfED is similar to the one that we ultimately converged on in our work, and the original implementations of the KEfED system are publicly available or open source[4]. However, they

[3] https://web.archive.org/web/20100824114553/http://bioinformatics.eas.asu.edu/siteProtocolDB/indexProtocolDB.htm

[4] https://github.com/BMKEG/bioscholar https://github.com/BMKEG/kefedEditor

are complex web applications and the underlying model for protocols is hard to decouple and use in other contexts.

In summary, the existing work on protocols and formalization of scientific processes is generally internally consistent as a whole and, if you rename a term here and there, the underlying models are mostly compatible. Despite this, there are significant issues with applying such formal models in the lab even when effort has been made to create electronic lab notebooks specifically targeting the laboratory use case [51]. One reason for this is that while ontology languages are extremely powerful, they ,by design, present a domain *general* interface for encoding and extracting knowledge and are therefore hard for users to apply directly to a specific use case. In addition, ontologies usually lack an obvious entry point or clear user interface that can hide the complexity of the domain they are modeling [68]. The need for simplified interfaces to these systems is also suggested by the finding that working scientists tend to prefer a small number of tags when asked to apply semantic tags to their work due to the overwhelming number of classes and relationships present in ontologies [50].

Our contribution to formal protocols is the development of a domain-specific language (DSL) for scientific protocols called `protc/ur` that has an open source implementation[5], and tooling and pipelines that can interoperate with knowledge management infrastructures[6]. Developing domain-specific languages to provide tractable interfaces to complex problems has a long history [91, 76, 99, 98, 24]. Our objective was to create a language with minimal surface syntax, understandable composition rules, and as few language elements as possible that could act as an interface to the complex underlying model for protocols and scientific processes that is reflected in the existing literature.

[5] `https://github.com/tgbugs/protc/blob/master/AGPL-3`

[6] See for example `https://github.com/SciCrunch/sparc-curation/blob/f2b1c74df3270cb00bd48801e076eca368a53033/docs/developer-guide.org?plain=1#L1433-L1434`.

3.2 Methods

Language ecosystem

As it exists now, the `protc/ur` ecosystem has two complementary implementations that leverage different aspects of the structure of the language, enforce different rules, and serve different use cases. One implementation is in Racket [23, 24, 27] and is referred to as `protc/ur`. The other is in Python and is part of the `protcur` library [7]. Figure 3.2 provides and overview of the system as a whole. In essence the relationship between the two implementations is that `protc/ur` is the target output language of the `protcur` pipelines. On the Racket side `protc/ur` is implemented as a Racket `#lang` [98]. `#lang protc/ur` checks and enforces the nesting structure of the `protc/ur` abstract syntax tree and expands to a Racket module that can be imported and used by the sibling `protc` language (discussed below)[8]. On the Python side `protcur` is responsible for converting between a variety of alternate representations for `protc/ur` that are discussed in detail below.

The `protc/ur` language currently contains five core elements (input, output, aspect, parameter, and invariant) and five extended elements (executor-verb, objective, black-box-component, symbolic-input, and measure). Table 3.1 provides an overview of these elements and their definitions. The five core elements can be grouped into three major categories by collapsing inputs and outputs to participants, and parameters and invariants to constraints. These three categories, participants, aspects, and constraints are helpful for understanding the curation workflow described below. Figure 3.13 shows an example of how the core elements can be composed into a minimal specification for a protocol. These elements and their composition rules make up what we call the *`protc` domain model*.

The system as it exists now is the product of a long process of iterative development. The

[7] `https://github.com/tgbugs/protc/tree/master/protcur`

[8] At the moment the import functionality is clunky and uses the annotation ids as identifiers for variables in Racket.

initial development of protc/ur is covered next. After that we cover validation of the domain model. Finally, we cover evaluation of the resulting language.

Language design and conceptualization

protc/ur is a sibling of a larger language named protc. While protc is not the focus of this paper, the development of protc/ur is best understood as an extension and refinement of protc for the specific use case of curating natural language protocols. Therefore we start with an overview of the development of protc and how that led to the development of protc/ur.

Work on protc started with early experiments on syntax and a number of written thought experiments exploring the design space. The whole history of development can be viewed in the git log of the protc repository[9].

Early experiments were conducted using Common LISP [93] and modern variants of the Lex lexical analyzer [62] and the Yacc parser generator [47] for eBNF grammars. These were ultimately abandoned due to requirements collection for an implementation language which suggested that Racket would be more suitable for our use case. This was primarily due to Racket's focus on language oriented programming and its origins in research on programming pedagogy. Racket's origins in pedagogy have resulted in an ecosystem that works across operating systems and is accessible to novices [23, 26], both key requirements for supporting experimental scientists and curators [10].

While experimenting with syntax, the underlying domain model for protc was also under development. Numerous iterations with varying levels of complexity were considered, but eventually we arrived at an initial set of core elements that were the starting point for validating

[9] To view the entire history you can run git clone https://github.com/tgbugs/protc and then git -C protc log -p --reverse or see https://github.com/tgbugs/protc/commits/master?after=178795f5161a2289efbbeacbdbb5a1463c07e4b8+567 and click the commit hash on each entry to view the diff. A summary of the early thoughts is documented in https://github.com/tgbugs/protc/blob/master/thoughts/README.md.

[10] The first author can report in retrospect that choosing to use Racket was a fantastic decision.

the model by using it to annotate methods sections[11].

Early work on the domain model was assisted by related work on an ontology for methods used in neuroscience [12]. Due to its broad scope, neuroscience makes use of nearly every technique employed in life sciences and even some beyond. Work on the methods ontology provided a survey of the major techniques and tools of the domain, but critically lacked a core model for the fundamental nature of techniques and how they interact with their various component parts.

Language domain model validation

We employed two approaches to validate, extend, and refine the `protc` domain model. One was to use the domain model as the core for the methods ontology and then to apply the ontology to model experimental techniques. The other was to apply the domain model to annotate methods sections of papers from the literature.

For the methods ontology, there was a point where it became clear that the `protc` domain model could be used as the foundation for the missing core of the methods ontology, and we implemented the `protc` domain model in OWL and have deeply modeled 227 techniques[13].

As such, the `protc` domain model has logical foundations that can be expressed in the OWL. The two implementations are separate; however, there is a mapping between the tags used in curation and owl classes [14]. Figure 3.4 provides an overview of the OWL model.

[11] Iteration on the domain model did not start from nothing. Prior work to create a data acquisition and metadata management library in Python, while the first author was still in an experimental electrophysiology lab, provided a starting point for iteration (`https://github.com/tgbugs/mlab`). Many of the ideas and early versions of the core elements of `protc` are present in the code and comments of the mlab codebase. The structure of the code, and the entities in the model were preliminary, but they were created based on the direct practical need to acquire and store data in the context of active experiments.

[12] `https://github.com/tgbugs/pyontutils/tree/master/nifstd/development/methods`

[13] `https://github.com/SciCrunch/NIF-Ontology/blob/methods/ttl/methods.ttl`

[14] `https://github.com/tgbugs/protc/blob/master/protc-tags.rkt`

Web annotation

To annotate methods sections we used the Hypothes.is web annotation client [15]. Hypothes.is makes it possible to annotate web pages and pdfs via a browser extension or javascript bookmarklet. Users can create a page note that is attached to the web page as a whole, highlight text to create an annotation anchored to that text, or reply to page notes, annotations, or replies. Annotations, page notes, and replies are always made in a specific Hypothes.is group or in the public group. The two groups discussed in this paper are referred to as the `protocol-annotation` group that was used to annotate methods sections and the `sparc-curation` group that was used to annotate SPARC protocols.

Hypothes.is has a web API[16] that provides programmatic access to the underlying data for each annotation structured as JSON [16, 11]. We maintain the `hyputils` python library for working with the Hypothes.is API [17], that was developed as part of `SciBot`, the machine part of a hybrid human/machine curation pipeline [2]. `hyputils` is used by the `protcur` Python library for retrieving and working with annotations.

Methods section corpus

The initial set of methods sections selected for annotation was drawn from the Markram 2015 citation tree [67]. The papers were downloaded as pdfs and hosted on a local web server. The hosts file (`/etc/hosts`) was used to ensure that the url that the Hypothes.is client saw was consistent (and not localhost). This simplified anchoring of annotations as well as control of the source documents for the annotated corpus and avoided issues with changing urls and inconsistent document identifiers. In total 5051 annotations were made on 56 papers as part of this initial

[15] `https://www.hypothes.is`

[16] See the documentation at `https://h.readthedocs.io/en/latest/api-reference/`. The Hypothes.is backend is open source (`https://github.com/hypothesis/h`) and it is possible to run it independent of Hypothes.is which is important for long term sustainability of these workflows even if we implement a new annotation client tailored to protocol curation in the future.

[17] `https://github.com/tgbugs/hyputils`

exploratory curation process.

Validation by methods section annotation

In order to apply protc to methods sections, we needed to capture the text referring to an entity, the type of that entity, and the relation between that entity and other entities referred to by other parts of the text.

To accomplish this we annotated methods sections by highlighting text referring to individual entities and adding the tag for that type of entity. Each tag is the name of a syntactic element from protc prefixed with the string protc: (e.g., protc:input). In order to capture the relationships between entities according to the protc domain model, we copy and paste the share link to an annotation into the text box of the annotation that was anchored to text for the parent entity in the domain model. For example, the link to an annotation about an aspect is pasted into the text box of an annotation about its corresponding input.

This protcur curation process, involving annotation, tagging, and linking, is mechanically complex. This was not a concern during this early exploratory phase because it involved a significant amount of thought while trying to apply the model and did not require extreme speed. However, as the annotation tag set was refined, this was no longer the case because the number of tags was reduced and their scope was better understood. Given this mechanical complexity, a lingering question was whether it was possible to employ this process at scale. We address this in the section on curation efficiency.

As we annotated we added additional tags when we encountered text in methods sections that did not seem to correspond to any existing entity type. Repeated use of a new tag indicated that the protc domain model was missing a key element, and the domain model was updated to include that new element. As the protcur curation process and tag set evolved, the Racket implementation was updated to match and enforce new rules as needed.

The most important element missing from the initial domain model was the concept of aspect. As defined in Table 3.1 as aspect is a generalization of the dimension of a unit, such as length, duration, mass, temperature, pressure, etc. which also includes concepts such as allocation, and count.

The initial domain model was also missing protc:implied-* tags. This is not surprising as implied-* tags are only relevant in the context of natural language annotation and are equivalent to their explicit counterparts in the domain model, such as aspect or input. However, unlike their explicit counterparts, they are used when there is not actually any text that can be highlighted that refers to the entity in question, that is, the entity is implied by the text. For example, in the following sentence from [49] ”recording chamber” is an explicit input but there is no corresponding text for the thing ”slices were incubated” in, i.e., a water bath[18].

> Slices were incubated for 30 min at 35°C and then at room temperature (~20-22°C) until transferred to the recording chamber (35 ± 0.5°C).

While developing the protc: tags and the protcur curation process we also created a separate set of tags to enable non-destructive editing so that it is possible to modify annotations without destroying their previous contents[19]. This was also critical because it is currently not possible to edit Hypothes.is annotations made by other users, so it is impossible, e.g., for a head curator to correct mistakes without such a system[20]. Non-destructive editing is also important for documenting the development of the thought process that resulted in current practice and is critical for tracking provenance.

protcur pipelines and transformations

In order to analyze the results of the methods section annotation workflow we needed to convert those annotations into a representation that was closer to protc than the raw annotations.

[18] It is also almost certainly not clear to anyone who has not worked in an electrophysiology lab that the brain slices do not go directly in the water, but in fact go in a slice chamber which is what is placed in the water bath.

[19] https://github.com/tgbugs/protc/blob/master/anno-tags.rkt

[20] Sharing logins is not a solution as it makes it hard to determine provenance and is a bad security practice.

Using the initial pool of annotations, we started work on Python code to structure and analyze the results of annotation, tagging, and linking. This code grew to become the protcur Python library that was mentioned above, and is responsible for transforming raw Hypothes.is annotations into structured protc/ur as an output.

Figure 3.3 shows an example of the transformations carried out by the protcur pipelines that are diagrammed in Figure 3.2. Specifically, it shows an HTML page with highlighted annotations followed by the Hypothes.is UI rendering of the JSON representation, the Racket protc/ur representation, and an RDF Turtle representation for the same annotations [102, 81]. In addition to these three representations of protc/ur, there are two more core representations not shown in the figure, one is a Python class representation, and the other is a JSON-LD representation [52].

Parsing units

Given the vital role that units of measure play in science we needed to be able to extract them from methods sections and thus wrote a parser combinator library and units parser in Python to extract units from free text into a structured representation. The units parser runs on the exact text or a curator provided correction of any annotation that is tagged with protc:parameter* or protc:invariant. The results of unit parsing and extraction can be seen in the transformations from Figure 3.3.1 to 3.3.2.

The parser combinator library grew into pysercomb [21] and is also the basis for a parser for Racket language syntax that is used by the protcur pipelines. In fact, the output of the first step of the protcur pipelines is protc/ur that is parsed back into a Python representation by the pysercomb Racket parser before subsequent use.

[21] https://github.com/tgbugs/parsercomb

Language evaluation by protocol curation

In order to evaluate the `protc/ur` language and the curation process we had a single curator apply a streamlined version of the `protcur` curation workflow to protocols. The streamlined workflow focused on a subset of the elements in the `protc` domain model, inputs, outputs, aspects, parameters, and invariants with one new addition, executor verbs (see Table 3.1). Other tags were used sporadically but were not part of the core workflow.

The streamlined flow was designed to extract quantitative constraints from protocols, attach them to aspects, and then attach aspects to participants. By focusing on these three major types of entities and starting curation at quantitative values in the text of a protocol, we were able to create a curation workflow that a curator could learn to apply and that was mostly agnostic to the domain specific contents of a protocol. As demonstrated in the results, this workflow can be applied efficiently to a variety of protocols to achieve high coverage and depth of curation[22].

Protocol curation was carried out as part of our work to develop metadata and curation standards for the NIH Common Fund's Stimulating Peripheral Activity to Relieve Conditions (SPARC) program [77] with the objective of extracting metadata from protocols that had been used to generate datasets or were reconstructed from processes that generated datasets for SPARC. Nearly all of the SPARC protocols that were annotated are contained in protocols.io [96].

[22] One thing to note about this workflow is that the curator encountered significant issues with the anchoring of annotations to pages because at the time the only source available to annotate was an interactive webapp where the url for the page changed as the user scrolled. This is no longer an issue as protocols.io now provides a static HTML version of all protocols. However, we had to align the original annotations made prior to this and create copies that target the HTML pages. As a result of these issues during curation, there are cases where entities were annotated multiple times because the curator could not determine whether a given stretch of text had already been annotated since the Hypothes.is client was unable to associate annotations made on webapp urls that for different steps with the protocol as a whole or with urls for other steps.

Alternate curation workflows

In order to assess the efficiency of the `protcur` curation workflow we compare it to two other workflows applied by the same curator to the same set of protocols. In addition to the `protcur` workflow[23] there is a page note workflow[24] and a SPARC Minimal Information Standard (SPARC MIS) workflow[25]. These workflows are described here for reference and Figure 3.6 shows an overview of the timeline for all three workflows (see results for an explication of the figure).

The `protcur` workflow was active for roughly a year from 2019-01-22 to 2020-01-27. The workflow itself is described above.

The `sparc` workflow was active for roughly 2 months from 2020-03-26 to 2020-05-29. The workflow required the curator to apply tags from a set of roughly 200 tags drawn from the SPARC MIS ontology file[26] to entities in the text using the Hypothes.is client.

The `notes` workflow was active for roughly 1 year from 2020-07-20 to 2021-06-18. The workflow had three parts. The first was to find all datasets that a protocol was associated with and add the identifiers for those datasets as tags in a page note on the protocol. The second was to read the paper and add semi-controlled tags for experimental modality[27] prefixed with `mod:`, taxon (organism) prefixed with `org:`, anatomy prefixed with `anat:`. The third was to tag resources with their corresponding Research Resource Identifier (RRID) if it was not provided elsewhere in the protocol.

As part of the `protcur` and `sparc` curation workflows the curator applied the `ilxtr:technique` tag to text that referred to experimental techniques used in the protocol. This was not done as part of the `notes` workflow because it was assumed that tagging the whole paper with `mod:` for experimental modalities would be sufficient to capture the techniques without the need to annotate

[23] sometimes referred to as the ”p” workflow

[24] sometimes referred to as the notes or ”n” workflow

[25] sometimes referred to as the sparc workflow, the mis workflow, or the ”s” workflow

[26] `https://github.com/SciCrunch/NIF-Ontology/blob/sparc/ttl/sparc-methods.ttl`

[27] aka experimental approach

techniques in the text.

Language use cases

In order to evaluate the utility of `protc/ur` against the specific use cases that it was developed to address, we had therefore to develop those use cases. A number of these use cases were taken from the original development of `protc`.

Some of these original use cases were as follows: To be able to use the language to formalize experimental protocols. To reduce the need for repeated entry of metadata shared across multiple processes by encoding it in the protocol. To be able to find data produced by protocols matching certain criteria. To be able to assist the user in validating a protocol.

Additional practical use cases were also developed as we worked on the `protcur` Python libraries and `protc/ur` Racket language. Most of these are not evaluated below because the tight development loop for `protcur` means that they were usually satisfied by the implementation within a matter of days. Examples of these follow: To extract structured data from natural language protocols. To resolve annotations with replies containing non-destructive editing tags into the correct version of the annotations. To convert the tags and linking structure of the annotations into nested s-expressions [73] (See Figure 3.3). To make it easy to get to the original source annotation (e.g. by clicking a link). To detect and alert when links have been pasted in the wrong order, i.e., that the nesting of `protc/ur` is incorrect. To detect and alert when the source text for a parameter or invariant seems to be malformed.

One use case that we have explicitly excluded from `protc/ur` is modeling of protocols describing analysis or the manual operation of software (e.g., how to operate a microscope for image acquisition). Analysis and software were excluded for a number of reasons including the following. Analysis and software systems already have formal languages that can be used to describe them (e.g. Python, MATLAB, etc.). Instructions for operating graphical software can be formalized and automated using languages such as AutoHotKey [61, 100, 70, 37, 95]. Code and software can be of arbitrary complexity and thus are at risk for consuming large amounts of time and attention if an attempt is made to model them[28]. Finally, analysis and software steps are usually separate from wet lab experimental processes and data acquisition processes which are the primary domain for `protc/ur`[29].

Evaluation of use cases by query

One approach to evaluate the use cases for data extraction from natural language, formalization, metadata entry, and data search was to query a knowledge base containing the outputs of the `protcur` pipelines. Minimally, this system needs to include information about both protocols and datasets. To this end we used the SPARC Knowledge Graph (SKG).

The SPARC Knowledge Graph has a number of parts, but in the context of evaluating `protc/ur` the relevant information is the outputs of the `protcur` pipelines, the output of the SPARC dataset curation pipelines[30], and a subset of the NIF-Ontology, all in RDF. The full conversion and release pipelines are implemented as executable documentation [31].

Queries are issued to the SKG using the SPARQL query language [88, 42] on an endpoint running Blazegraph version `2.1.6 RC` [32]. Releases of the SKG are available on GitHub [33] and are also archived on zenodo [34]. The results in this paper are derived from the 2023-08-04 release. Additional example queries are maintained in the sparc-curation documentation[35].

[28] Consider for example the difficulty of trying to develop a domain model that was capable of formalizing the contents of the methods section for this paper.

[29] This is not always true. See discussion.

[30] `https://github.com/SciCrunch/sparc-curation`

[31] sparc-curation/docs/release.org sparc-curation/docs/developer-guide.org#sckan See also the command line example `developer-guild.org sckan-release` in dockerfiles/source.org

[32] `https://github.com/blazegraph/database`

[33] `https://github.com/SciCrunch/NIF-Ontology/releases/tag/sckan-2023-08-04`

[34] `https://doi.org/10.5281/zenodo.5337441`

[35] `https://github.com/SciCrunch/sparc-curation/blob/master/docs/queries.org`

Protocol curation corpus

In order to evaluate the protc domain model and the practicality of the protcur curation workflow, we analyzed curation coverage, curation depth, and curation efficiency. In order to measure coverage, depth, and efficiency we need to define the set of protocols that will be included for analysis. Figure 3.5 shows an overview of the filtering process and the inclusion criteria and exclusion criteria are described below.

The only protocols considered for analysis were those that had annotations in the Hypothes.is sparc-curation group. Those are a subset of a larger set of protocols that also includes protocols referenced by SPARC datasets some of which have not yet been annotated. In order to ensure that we were considering only first class protocols (i.e., excluding methods sections), we included only protocols that were in protocols.io from the set that had annotations in the Hypothes.is group. This was also done to simplify measuring curation coverage because other types of documents have different text representations and included other types of sections that would never be annotated. As of our 2023-08-04 release of the SKG, applying these criteria reduce 308 total protocols down to 115.

From these 115 protocols we then applied the following criteria to arrive at our final count of 97 protocols for analysis. Protocols must have been annotated as part of the protcur curation workflow as indicated by having at least one annotation with a protc: tag[36]. Protocols must have a static HTML page on protocols.io (some protocols have been deleted since original annotation). Protocols must have content and must not be empty (some protocol contain no text and only cite other protocols). Protocols must have at least one annotation, and are removed if they have only a page note. Protocols must have a ratio of protc: to sparc: tags that is greater than 1. Finally a single extremely long protocol[37] that was five times longer than the next longest protocol was excluded as it contained extensive sections describing analysis procedures

[36] This is slightly complicated by the fact that some sparc: tags are automatically converted to protc: tags

[37] https://www.protocols.io/view/31399.html

and analysis is explicitly excluded from the current design of `protc/ur`.

As part of this process we made extensive use of the protocols.io web API[38], which provides access to the underlying data for each protocol structured as JSON. This API is used to retrieve protocol contents and metadata and is critical for resolving protocols to stable identifiers so that they can be anchored to the static HTML pages as discussed above. We have implemented a module for working with protocols from protocols.io as part of the `idlib` python library [39].

Conversion of protocol HTML to plain text

Similar to the case with the five representations of `protc/ur` discussed above, there are five representations of the protocols from protocols.io that are relevant for understanding and interpreting the results. These are protocols.io app view, protocols.io HTML, protocols.io HTML converted to text, protocols.io api JSON (3 different versions), and protocols.io JSON converted to org-mode [87, 92].

In order to analyze the results of curation we needed to convert protocols to plain text. We had previously implemented the ability to render protocols from protocols.io to lightly structured text in Org syntax[40], but for analysis of curation we needed to ensure that the plain text of protocols was as similar as possible to the rendered HTML that was originally annotated. As a result we rendered the protocols.io static HTML pages for each of the 97 protocols directly to text.

To do this we retrieved and saved each url with `curl`[41] and used `firefox`'s developer tools to obtain the http auth headers needed to access protocols shared privately. Next we used `sed` [53] to insert clear boundaries between document sections that are ambiguous when rendering to text to simplify later steps. Finally, we rendered and saved the HTML pages to text using

[38] The main documentation is available at `https://apidoc.protocols.io/` which covers v3 and v4 of the API. We also use v1 of the API.

[39] `https://github.com/tgbugs/idlib/blob/master/idlib/from_oq.py`

[40] `https://orgmode.org/worg/org-syntax.html`

[41] `https://curl.se/`

the `links` web browser launched with the `-dump` command line option [42]. All further processing of the saved protocol text is implemented in Python (See Data and code availability).

Curation coverage

In order to evaluate the completeness of the `protc` domain model we measured curation coverage. Curation coverage can be used as a proxy to assess the completeness of a domain model. If it is possible to annotate every piece of text in a natural language document using a small number of tags, that suggests that the domain model underlying those tags is complete. On the other hand, if few pieces of text can be annotated it suggests that the domain model is incomplete[43].

Curation coverage is calculated by removing strings of characters from curated protocols. First we remove all spans of annotated text from the whole text leaving only the remnant text that was not annotated (like an apple core). Then for both the remnant text and the original whole text we split the text on white space to produce a list of word tokens and then remove stop words and other low information content words from that list. After this we count the number of normalized tokens in the remnant and the whole.

To calculate curation coverage we use the formula `(whole - remnant) / whole`. We do this instead of trying to use the number of consumed tokens directly because the types of systematic error that we see (e.g., due to partial annotation) mean that `(whole - remnant) <= consumed`. The bias in the process of removing annotation text, tokenizing, and normalizing means that total number of actual tokens in the remnant will always be greater than or equal to the actual number of word tokens contained in the unannotated portion of the text. Therefore, the calculated value for `(whole - remnant)` is always less than or equal to the value that we would obtain if we tried to determine `consumed` directly by adding up the potentially overlapping

[42] `http://links.twibright.com/` `https://man.archlinux.org/man/links.1.en`

[43] If the text can be annotated multiple times using different tags it suggests that to domain model is over specified and at risk for being applied inconsistently.

tokens of the annotated text even if we could successfully detect and correct for the overlap. As a result calculated coverage is conservative because (whole - remnant) will be smaller and our calculated coverage will be lower than if we tried to calculate consumed / whole directly. Said another way, (whole - remnant) / whole is a worst case measure for coverage while consumed / whole is a best case measure for coverage. We prefer the worst case because it prevents us from thinking protc/ur is performing better than it actually is, and because it is also easier to produce the remnant than to deduplicate the contents of the annotated text.

Curation depth

Curation depth is calculated by dividing the number of annotations with at least one child annotation by the total number of annotations made on a protocol where children would be expected. Annotations where children are not expected are those tagged with protc:parameter*, protc:invariant, protc:objective*, and protc:black-box-component. Another way to think of this measure is as one minus the ratio of the number of annotations with zero children where children are expected to the total number of annotations where children are expected.

The depth threshold used in figures 3.8 and 3.9 was selected to remove protocols that had zero or almost zero depth of coverage. This was done because the full protcur curation workflow was not applied to those protocols.

The correlation coefficient R in Figure 3.9 was calculated using the NumPy implementation for computing the matrix of Pearson product-moment correlation coefficients.

Curation efficiency

Curation efficiency is calculated by building a model of duration (contiguous blocks of time) from discrete events (moments in time). The model takes a single parameter, which we refer to as maxdt, that is the maximum duration (delta time) between two moments where two consecutive events are still considered to be part of the same contiguous duration. Event duration is then calculated as the time delta between consecutive events. There are two types of events that we have access to from Hypothes.is for page notes, annotations, and replies, the time at which they were created, and the latest time at which they were updated.

One issue with this approach to determining durations from events is that the page notes workflow is particularly susceptible to censorship. That is, when new updates erase the record of previous updates. What this means is that if a curator repeatedly edited the same page note over a period of time we no longer have a record of the intervening updates. Evidence that this has occurred can be seen by comparing the sharp divergence between the curves for `page note base` and `page note other` in Figure 3.11. When other events beyond page note creation and update are included for determining durations, the amount of time that we can account for where the curator is actually working increases dramatically.

After converting events into curation bouts, we sum contiguous curation time and divide it by total number of events to get an absolute average for curation events per unit time. This is done for a variety of values for maxdt.

In order to determine a reasonable value for maxdt we compare mean event duration to the standard deviation as a function of the event duration index. The event duration index is ordering of event duration from shortest to longest. There are two points that we use, one where the standard deviation is 3 times the mean, and another where the standard deviation is 6 times the mean. These happen at roughly 86 seconds (~1.4 mins) with 399 durations longer, and 122 seconds (~2 mins) with 256 durations longer respectively.

Another way we can estimate maxdt (which might not be stable over time) is by looking at the distribution of durations between curation events that occur on different protocols. In all three workflows the curator tends to focus on a single protocol at a time and rarely switches rapidly between protocols during the same contiguous curation bout (sometimes this has been observed to happen in the page notes protocol). Therefore the duration measured for switching between protocols can be used as a proxy to estimate maxdt since we know that in most cases annotation activity would be broken up by having to switch to another protocol. The values we obtained for median maxdt were between 156 seconds (2.6 minutes) and 330 seconds (5.5 minutes) depending on whether we set a cutoff for durations greater than 4 hours. We do not use the mean in this case because even duration has an extremely long tail.

The combination of these two approaches for estimating maxdt suggests that the real value for maxdt is somewhere between 1.4 and 6 minutes, and probably somewhere closer to 2 minutes.

Data and code availability

All of the input data for the SPARC Knowledge Graph including an export of the Hypothes.is annotations from the `sparc-curation` group and conversion of those annotations to the `protc/ur` Racket, JSON-LD, and RDF Turtle representations is available on GitHub `https://github.com/SciCrunch/NIF-Ontology/releases/tag/sckan-2023-08-04` and archived on Zenodo `https://doi.org/10.5281/zenodo.5337441`.

Instructions for how to query the SKG are available at `https://github.com/SciCrunch/sparc-curation/blob/master/docs/sckan/README.org`. And the full output of running the `protcur` pipelines to produce the equivalent `protc/ur` for each protocol is available in the zenodo release under `release-*-sckan/data/protcur-sparc.rkt`. `protcur-sparc.rkt` can be run most easily in the `tgbugs/musl:kg-dev-user` docker image (see below) by retrieving the zenodo release in the docker image, unzipping it, and running `racket -it release-*-sparc/data/protcur-sparc.rkt`.

All public protocols used for analysis are available on protocols.io. Links to all 97 protocols are listed in Table 3.6 and 3.7 (26 are currently not shared publicly).

The reports on the remnants for calculating curation coverage are available at `https://github.com/tgbugs/protc/releases/remnant-review-2023-09-08`.

Code for the python libraries is available on GitHub at https://github.com/tgbugs/hyputils, https://github.com/tgbugs/idlib, https://github.com/tgbugs/parsercomb, https://github.com/tgbugs/protc, and https://github.com/SciCrunch/sparc-cruation. All python libraries are packaged and published to PyPI (see https://pypi.org/user/tgbugs/) and have Gentoo ebuilds available at https://github.com/tgbugs/tgbugs-overlay.

The Racket code for protc/ur is available on GitHub at https://github.com/tgbugs/protc.

A Docker image containing executable versions of all the code related to this paper is available on docker hub in the tgbugs/musl:kg-dev-user image. The source used to generate the images is available on GitHub at https://github.com/tgbugs/dockerfiles. Example protc/ur code can be evaluated in Racket after executing the command

```
docker run -u 1000 --entrypoint /usr/bin/racket --rm -it \
tgbugs/musl:kg-dev-user -i -e \
"(require protc/ur protc/private/curation protc/private/curation-unprefixed)"
```

Analysis code for this paper is available on GitHub as part of talk.org in https://github.com/tgbugs/dissertation. A complete description of how to execute the code for this paper and produce the paper itself by running talk.org is described in talk.org.

All analysis code for this paper is implemented in Python and was run on PyPy3 7.3.12. The examples of protc/ur shown in this paper have been tested on Racket 8.10. Setting up an environment from scratch to be able to run protc/ur is currently complicated so we suggest using the docker image and docker run command described above.

3.3 Results

Language

The primary result of this work is the protc/ur language. The most basic requirement of a practical language is that its written form can be executed. All examples in this paper can be run as described in the methods.

Figure 3.13 shows a small example protocol written directly in protc/ur. It is a minimal protocol for making a diet induced obesity mouse formalized from [21]. The example shows how output is used to bind the aspect mass to the DIO mouse at the end of the protocol and how the use of input distinguishes that binding from the binding of the aspect age to the mouse at the start of the protocol. The protocol indicates that the input mouse and input diet are to be combined in some way by the fact that both are nested within the same output and are thus inputs to the same process that produces the DIO mouse. The availability of the mouse diet is represented as an opaque (aspect "ad libitum" (parameter* (bool #t))), and though it might be more correct to say (aspect "availability" (invariant (fuzzy-quantity "ad libitum"))), such enhancement can be made when more detail is required.

Figure 3.3 shows the transformation of a protocol from annotated HTML to protc/ur and RDF Turtle. The top of Figure 3.3.1 shows the annotated HTML of first five steps of a protocol, the bottom half shows the annotation content for only the first step[44]. Figure 3.3.2 shows a slightly simplified protc/ur representation of all five steps from the HTML. The separation of the protocol into discrete steps is represented using executor-verb. We can also see one way that protc/ur provides feedback to assist the user by informing them that that the tag on the annotation might not be correct with the parse-failure construct. We can also see the output of the parsing and transformation of plain text units into a structured representation. Figure 3.12 contains a version of the protc/ur code from Figure 3.3.2 that can be selected so that it can be

[44] The full contents of the JSON representation were too verbose to include and generally not helpful.

run. Figure 3.3.3 shows the further translation of the first two steps in the original protocol from protc/ur into RDF Turtle. The top section is a helper representation to make it easier to see the equivalence to the first two steps of the protc/ur representation, and the bottom section is the actual representation of the same two steps as it is would be loaded into the SPARC Knowledge Graph and contains the original annotation ids from Hypothes.is to make it easier to track provenance and debug any issues with the transformation.

Domain specific languages can assist the user in producing correctly formed statements by providing immediate feedback if something goes wrong. Figure 3.14 shows how protc/ur is able to provide detailed error messages that can direct a user to the exact location of the problem when its syntax is violated[45]. As indicated in Figure 3.2, these error messages have been used in the curation workflow to identify and correct malformed annotation structure that would otherwise be difficult if not impossible to catch. These corrections can be seen in the Hypothes.is annotations from the sparc-curation group exported to release*/data/annotations.json contained in the zenodo release of the SKG [30].

Language ecosystem

The rest of the results presented here are the product of a combination of protc/ur and the larger system for curation and knowledge management surrounding it. Figure 3.2 provides an overview of the larger system around protc/ur and how the system interacts with human curation workflows. While protc/ur can be written by hand, the overwhelming majority of all protc/ur is automatically generated as and output of the protcur pipelines. The flow starts with a protocol. That protocol is curated by a human who applies the protcur curation workflow to create Hypothes.is annotations. Those annotations are then transformed by the protcur pipelines into two representations, protcur.rkt and protcur.ttl[46]. In the left branch in Figure 3.2,

[45] This functionality is implemented using Racket's syntax/parse library [17, 25, 98].

[46] This figure has been simplified multiple times to make it easier to understand. Therefore it omits certain implementation details such as that protcur.ttl is not produced by the protcur python library alone, but is produced

protcur.rkt is a file that contains expressions in protc/ur that are evaluated by Racket to validate their structure and contents. If issues are detected when running protcur.rkt the flow loops back to curation and changes are made to the source annotations using the set of tags for non-destructive editing discussed in Methods. In the right branch protcur.ttl is loaded as part of the SPARC Knowledge Graph into an RDF triple store so that it can be queried (See Methods).

Queries and query results

Beyond the basic structure of the language and its ability to assist users when writing protocols, protc/ur is also able to capture information from natural language protocols so that that information can be incorporated into larger integrated data stores and used, among other things, for search. Practically, we expect that search queries will be written by knowledge engineers for the foreseeable future, and as such do not expect curators or experimentalists to interact with queries directly. We use competency queries to assess the capabilities of both protc/ur and the system as a whole. We explicate and show the results of two sample queries here. More examples can be found in the sparc-curation repository queries documentation[35].

Table 3.2 shows the results of the query defined in Figure 3.15 for searching protocols by objective magnification. The caption of the table contains an articulation of the query in plain English.

The query in Figure 3.15 shows how we use SPARQL to traverse the nested structure of protc/ur as it is represented in RDF via the ?ast_ prefixed variables that are equivalent to the parentheses in the protc/ur structure and to the hyp-protcur: identifiers seen in Figure 3.3.3. Going from top to bottom the query shows how we leverage the the protc/ur structure to connect to and filter on the aspects, species, and anatomical regions in the protocol, and finally how we connect any matching protocols to datasets.

by running code in the sparcur Python library (https://github.com/SciCrunch/sparcur) that calls protcur code internally.

The results in Table 3.2 show that the query returns four distinct datasets associated with three protocols, the species the protocols were performed on include rats and mice, the regions include the inferior vagus X ganglion (aka the nodose ganglion) and the vagus nerve, and the values for magnification include 20, 40, and 63 fold objectives.

Tables 3.3 and 3.4 show results of a query defined in Figure 3.16 for searching protocols by species, drug, and dose. Table 3.3 is a subset of Table 3.4 and the caption of Table 3.3 contains an articulation of the query in plain English.

The basic structure of the query in Figure 3.16 is similar to the magnification query and connects to datasets and uses `?ast_` variables in the same way. Going from top to bottom the query first connects and filters on the units for the value of an invariant (specifying `mg/kg`), next it connects but does not yet filter the value for the upper limit of `mg/kg` so that it can be filled in later, after that it ensures that the `mg/kg` in question is linked to the specific drug specified in the query and not some other drug (since there are often multiple drugs and multiple doses listed in a protocol), and finally it connects to but does not yet filter the values for the drug and the species so that they can be filled in later. As such, the query can be called as a function with three arguments named `species-mg/kg`. The two arguments for species and drug are the same for both results tables, but the third for the upper limit of `mg/kg` is 100 for Table 3.3 and 1000 for Table 3.4.

The results in Table 3.4 also include the results in Table 3.3 and show that the query returns eight distinct datasets associated with six protocols and three distinct values for `mg/kg` with 2 less than 100, and 1 less than 1000. The drug is shown for reference as it was provided as an argument to the query (as was species though it is not included in the results for brevity). The results for both queries behave as expected in respecting the limit on `mg/kg` as demonstrated by the absence of the results for values greater than 100 in the first case despite their presence in the second.

Protocol curation corpus

In addition to the numbers provided for the protocol curation corpus shown in Figure 3.5 (See Methods), there are additional counts for various subsets of protocols found by querying the SPARC Knowledge Graph. As of the `2023-08-04` data release, from a total of `308` protocols `263` are from protocols.io. Of those protocols from protocols.io `157` have at least one `protc:` annotation. Of those protocols with at least one `protc:` annotation we have deeply annotated `100`, of which `77` are published and have a resolvable persistent digital object identifier (DOI).

Curation coverage

One way to assess the expressiveness and completeness of the `protc/ur` language, and to assess the effectiveness of our curation workflows is to measure the amount of text that is covered by annotations that are then converted to `protc/ur`. In short, coverage is effectively calculated as the ratio of the number of normalized text tokens in annotated text to the number of normalized text tokens in the whole text (See Methods for a details).

Figure 3.7 is a histogram of curation coverage for the 97 protocols in the corpus. The x axis shows curation coverage and the y axis shows the number protocols with coverage falling in the domain of that bin. For the 97 protocols, curation coverage summary statistics are mean 0.6182, median 0.6753, minimum 0.03, and maximum 0.93. The distribution is skewed to the right. The histogram shows that for 27 of 97 protocols, `protc/ur` annotations cover more than 75% of tokens, and that there are 25 protocols that fall below 50% coverage.

A direct examination of the 25 protocols with coverage less than 0.5 shows that such cases are nearly always due to the protocol containing a section that we explicitly excluded from the curation workflows, such as the description of analysis or image processing, or include verbose HTML elements such as equipment metadata. In a handful of cases it is due to a short protocol with incomplete annotation. Further, direct examination of the remnants for protocols with coverage greater than 0.5 shows that there are few if any major sections of the experimental process that are not captured. Links to the remnant reports are listed the Methods in Data and code availability.

Curation depth

Figure 3.8 shows a coverage histogram similar to Figure 3.7 except that it excludes protocols with zero or near zero curation depth, where near zero is defined as curation depth less than 0.1 (see Figure 3.9). Thresholding on depth removes 10 protocols from the original corpus[47]. For the 87 protocols remaining, curation coverage summary statistics are mean 0.6477, median 0.6895, minimum 0.03, and maximum 0.93. The distribution is skewed to the right with a mean and median curation coverage being slightly higher that the full corpus, but with the minimum and maximum remaining unchanged.

Curation coverage vs depth

Figure 3.9 shows a scatter plot of curation coverage on the x axis vs curation depth on the y axis. Blue dots are protocols above the depth threshold of 0.1 for zero or near zero and orange dots are below threshold. An examination of the near zero depth cases shows that the majority are the result of the curator only tagging a protocol without linking them by pasting share links[47]. The correlation between curation coverage and curation depth for protocols above the depth threshold is 0.58.

[47] An examination of these protocols shows that the majority are cases where tagging and linking were split into two phases and the linking phase was never started. `(defvar orange-zero-depth '(19255 26687 26737 27863 30948 31001 31076 31077 31143 31158))`

Curation efficiency

Figure 3.6 provides an overview of the timeline of protocol curation events for the different workflows. The x axis is time. The bottom of the figure shows the time span over which the workflows ran from 2019-01-22 until 2021-06-18. The top of the figure indicates the general period for each of the three workflows `protc:`, `sparc:`, and `page notes` (See Methods for a detailed description.). The y axis is divided into 7 classes based the intersection between the workflow and page notes, annotations, and replies. From top to bottom the categories are `page notes`, `sparc: replies`, `sparc: annotations`, `protc: replies`, `protc: annotations`, `other replies` and `other annotations`. For each category the color on the left indicates creation events for that type of annotation in that workflow, and the color on the right indicates update events to existing annotations of that type for that workflow.

Figure 3.10 is a histogram of all event durations for curation events with duration under 300 seconds (See Methods for a description of how event duration is calculated.). The x axis is event duration in seconds and the y axis is the number of events falling in the domain of that bin. The first bin on the left is mostly events that took well under 1 second and are discarded because they are updates that were made by machine and not the human curator. The final bin on the right includes all events with duration greater than or equal to 300 second, this was done because the tail of the distribution stretches over many orders of magnitude. Events from this histogram were used to estimate maxdt as described in the methods.

Figure 3.11 shows curation efficiency calculated as seconds per tag or text for a variety of assumptions about maxdt for all three curation workflows. The x axis of the figure shows maxdt (event duration cutoff) in minutes. The y axis shows the average number of seconds that it takes for a curator to create a tag or add text to an annotation. Each of the three workflows is show under two conditions, one with other annotations included an one with them excluded (See Methods). For both `protcur` and `sparc` workflows the inclusion of other annotations did not have a large impact on efficiency; however, for the page notes workflow it does. For an explanation for why this might be the case see the discussion in methods about censorship. The time scale for the figure was select to show the values for maxdt where protcur becomes more efficient than the other workflows. These crossing points happen a around 1.1 minutes against sparc:, 3.8 minutes for the page notes other, and 8.8 minutes for page notes base. These correspond to approximately 16, 31, or 40 seconds per tag or text respectively. Based on our estimates for a realistic values for maxdt (See Methods) the actual maxdt likely falls somewhere between the first and second of these time points.

3.4 Discussion

In summary, we have presented here protc/ur, a domain-specific language for scientific protocols and the ecosystem surrounding it. It is a production ready software system that can convert the output of human curation into structured data that can be checked for structural correctness and queried to find protocols and associated datasets.

Language structure

The ability to combine a small number of core elements to express the essence of a complex process was a key design objective of protc/ur. Our results in figures 3.13 and 3.3 show it achieves this use case.

Language assistance

The ability to assist users in writing and validating a protocol was one of our original use cases. The example of the parse-failure construct in figures 3.3.2 and 3.12 and the example error message in Figure 3.14 show two ways that we meet this use case.

Query results

Taken together, the query results in tables 3.2, 3.3, and 3.4 demonstrate that the system outlined in Figure 3.2 is fully functional. With regard to `protc/ur`, they show that it is able to connect precise quantitative information from protocols to datasets. The queries also include simpler sub-queries on species, chemicals, and anatomical regions, and aspects. This shows that the larger system surrounding `protc/ur` is able to meet the use case for finding data produced by protocols that match specific criteria. Further, the results of the `species-mg/kg` queries show that `protc/ur` can enable queries on ranges of quantitative values across multiple protocols, and that `protc/ur` provides sufficient structure to distinguish between bindings of the same aspect to different inputs. Quantitative range queries and binding disambiguation cannot be easily accomplished by simple tagging of protocols nor by traditional full text search over natural language protocols.

There is also an open question as to whether formal protocols can actually be used to automatically generate metadata for datasets. These query results show that this is possible, and that such metadata could be used to find relevant protocols and data across a large corpus. In this sense, despite the seeming mundanity of the queries we present, they demonstrate the potential that formal protocols offer with respect to reducing the need to record parameters manually. Further, if the majority of the parameters are already in a versioned protocol, then only the subset that are not known ahead of time need be recorded, potentially freeing researchers to focus on other aspects of an experiment. Further, this is not to say that these queries were at all mundane! The ability to quickly get a quantitative answer to the question ”what doses of this anaesthetic are given to rats” with supporting evidence and context in the form of protocols and datasets would be quite valuable, and currently is not easy to accomplish. While `protc/ur` can’t do that at scale for us right now, these results do show a way forward for building a system that could.

Domain model

By combining our analysis of curation coverage and curation depth we can draw strong conclusions about the completeness of our domain model. The key question we need to answer is whether incomplete coverage is due to the domain model being incomplete or is due to some other reason.

While high curation depth cannot imply that curation is complete because curation depth is only defined over the annotated text and as such is independent of curation coverage[48], low curation depth is by definition incomplete curation. This means that seeing both low curation coverage and low curation depth together for a protocol is a strong sign that curation is incomplete[49].

Since observing both low curation coverage and low curation depth on the same protocol indicates that curation is incomplete, the correlation we observe in Figure 3.9 is highly suggestive that the low coverage we see is not due to a failure of the domain model to capture elements of protocols, but rather due to incomplete curation. This, taken together with the fact that our examination of the remnants failed to find any obvious categories that were missing in the domain model, suggests that our domain model for `protc/ur` is effective at capturing the domain.

Even so, it might still be the case that the reason why certain protocols have low coverage is that they contain stretches of text that are not currently captured by the domain model. If this were the case we would expect to see evidence of this as protocols with low coverage and high depth. Noting that there are alternate explanations for why this might be the case[49], there are a number of candidates with high depth to coverage ratio that are worth investigating. On

[48] i.e., that it is possible to have curation depth of 1 and coverage of 0.01.

[49] The assumption here is that in current curation processes, annotating and tagging text tends to lead copying and pasting of shared links in time. Although there are some variants of the current curation workflow that could legitimately decouple depth and coverage, such as a curator always completing an entire loop of constraint, aspect, participant before moving on to the next, or a curator splitting the workflow into two separate phases of tagging and linking. While most of the time it seems that these do not happen and that the process falls somewhere between the two extremes, the majority of the protocols that were filtered due to having low or zero depth were the result of the second case where the curator split curation into two phases but never started the second phase[47].

examination if all cases where the ratio of depth to coverage was greater than 1.3 we found that they were either clearly incompletely curated, or contained sections that have been explicitly excluded from the domain model [50].

With more confidence that low coverage is likely due to incomplete curation, a key conclusion from the coverage results is that we have successfully applied `protc/ur` to cover up to 93% of all normalized text tokens in a protocol and more than two thirds of all tokens in half of the papers we have curated. These findings support `protc/ur` as sufficiently expressive to represent experimental protocols[51]. In addition, the tools that have been developed to measure coverage and depth can help target the existing curation process to finish incomplete protocols. The correlation between curation and depth will also be discussed below in the context of the practicality of the `protcur` curation pipelines.

Curation efficiency

A key conclusion for curation efficiency from Figure 3.11 is that, based on our estimates for maxdt, realistic values fall somewhere between 1.1 and 3.8 minutes, the values for maxdt at which the `protcur` curation workflow becomes more efficient than the `sparc` workflow and the `page notes other` workflow. Over this range, the `protcur` workflow is somewhere between 1 and 1.5 times as efficient at annotating and tagging as the `sparc` workflow, and somewhere between 0.5 and 1 times as efficient as the page notes workflow. If we take a value for maxdt around half way between the two of 2 or 2.5 minutes the relative efficiency is about .75 that of the `page notes` and 1.25 that of the `sparc`.

While the page notes workflow is more efficient in principle, the problem with the pages notes workflow is that it achieves this by giving up the ability to validate coverage and com-

[50] `(defvar high-dc-rat '(18595 18994 19139 19174 19206 19401 19798 34589))` This suggests that there might be a need for a 3rd metric in addition to coverage and depth that is capable of identifying low coverage cases that have not annotated known elements in the domain model. The obvious starting point would be numbers.

[51] With the previously noted exclusion of analysis and related sections.

pleteness in ways that can be automated. As a result the page notes workflow and the `protcur` workflow could be complementary with page notes providing speed for high level information and `protcur` filling in the detailed information in a way that can be automatically validated and that ultimately makes it possible to validate the results of the page notes workflow.

An important conclusion from the curation efficiency results is that our concerns about the mechanical complexity of the `protcur` workflow are largely unfounded. In fact, looking at the results in Figure 3.10 there are over 6000 curation events with duration less than 12 seconds. While there are certainly opportunities to improve the UI for the workflow to improve the overall curation experience and reduce errors, it is not necessary to do so purely to resolve the issues with the mechanical complexity of the process.

Curation pipelines

There has been an outstanding question about whether applying the `protcur` curation workflow to protocols is practical. Looking at results for coverage, depth, and efficiency the answer is clearly yes. From the results we see that it is possible for a curator to learn to apply `protc/ur` to protocols and to use it to achieve almost complete coverage of the scientific content of a protocol. The efficiency of the `protcur` workflow falls somewhere between the two alternate workflows while providing substantial additional benefits with regard to verifiability. Furthermore, the new tools developed to assess `protc/ur` as a language and `protcur` as a curation workflow have turned out to be quite useful for improving our ability to introspect the workflow and identify protocols where curation is incomplete. In a sense this might be a kind of meta-assistance provided to curators by the system as a whole.

Future directions

While protc/ur has met its original use cases and design goals (though not the larger goals of protc), as a result of narrowing the use case to focus on curating protocols there are a number of opportunities for protc/ur. Thus there are new features that we wish to implement and there are known issues with the current design.

The most important among the additional features are logical and practical sequencing, aka a dependency graph, and linear scheduling for a single executor (or any number of executors). These were not critical elements for the protocol curation, metadata, and search and discovery use cases. They are however critical for the acquisition automation and followable protocol use cases. Sequencing of steps and actions was also not a critical part of the domain model, in part because most natural language protocols are written in an imperative linear style and any higher order temporal structure is latent and must be inferred from the linear steps. There is also the matter of temporal constraints on scheduling that was also a late development for PSL [85], but is nonetheless of critical importance for many protocols in the life sciences where procedures must be carried out within certain time constraints due to the nature of living systems and the difficulty of keeping them in a consistent state on which to make measurements.

On the more technical side, one issue with the current version of protc/ur is that the position of inputs and aspects in the abstract syntax tree is inverted. protc/ur was designed to assist in human curation, and one compromise in the design was to keep the mechanical annotation workflow as simple as possible. As a result, the order in which annotations are made is from child to parent. The deepest nodes (invpar) are created first, their share link copied, and then the next level is annotated and the share link pasted. Thus the parent holds the pointer to the child (reversing the creation order). As a result of trying to maintain a single parent hierarchy to simplify curation, aspects in protc/ur are effectively forced to accept only a single argument since they can only be associated with a single parent. This is a problem because measurements as simple as distance require a reference to two things, either two inputs, or two black box components on the same input.

The current system thus needs to be updated to account for this, and ideally to allow for aspects that are multi-arity or variadic, or rather that the aspect function itself needs to be able to account for the arity of the particular aspect, ideally without having to have it specified ahead of time. `protc/ur` is not intended to contain exhaustive definitions for all aspects that one might ever encounter. In the core language it would be possible to define a new aspect before it is used, but in protc/ur, the definition must be generated from the contents of the annotations, while still ensuring that we can detect incorrect cases where multiple values have been pasted in. Such improvements would require a significantly improved and streamlined annotation client, or perhaps the addition of `protc:aspect-1` `protc:aspect-2` tags to make the arity explicit. This approach would allow us to continue to have `protc:aspect` nested under `protc:input` etc. as it is now, while making it possible to check and enforce that `protc:aspect-2` to `protc:aspect-n` should in fact be the parent of `protc:input`.

One of the things that we plan to implement in the next iteration of `protc/ur` is an extension to differentiate types of inputs. This is because we have found that we do not have quite enough information from input alone to accurately dispatch to various auto-completion services. For example, we would like to know that a `suture` is a tool or reagent, and not an anatomical entity where two bones fuse. The proposed additional categories fall roughly along the types differentiated in Figure 3.4, tools, reagents, and primary participants. That said, as seen in the `suture` example, the line between tool and reagent is not entirely clear. We might leverage our other work modeling the types of transitions that participants/inputs/outputs can undergo when they participate in a process[52], which would suggest names like `input-consume` and `input-wear` to indicate things like reagents that are consumed entirely or modified beyond recognition or usefulness, and inputs that survive the process with only normal wear and tear and can be used repeatedly until at some point the wear and tear accumulates and they must be maintained,

[52] See the test for `#lang protio` `https://github.com/tgbugs/protc/blob/master/protc-tools-lib/protio/test.rkt`.

repaired, or replaced.

The complexity in the account above is one of the reasons we have deferred making this distinction and might add only undifferentiated `tool` and an orthogonal `consumed` tag that could be used if and when it is needed, since differentiating `tool` aka `secondary-participant` from `primary-participant` is often sufficient to resolve the dispatch issues mentioned. That said, a pilot of this has already been conducted since `sparc:Reagent` and `sparc:Tool` tags were both used extensively as part of the `sparc` curation workflow.

Another challenge we see in leveraging protocols to supply experimental metadata is the need for affordances to help surface high level experimental design that can otherwise be hard to spot a midst the equally critical but often copiously detailed information in a protocol. There might be 20 different parameters that have to be done just-so, and only one or two that are intentionally varied in order to answer the specific question the experiment is trying to address. Potential ways to address this challenge include adding an element to the domain model that could be used to indicate aspects of the experimental design, or by making it possible to import core protocols parameterized for the current context into a protocols dedicated to the description of the high level experimental design.

As mentioned in methods, we currently exclude analysis and software steps, but there are cases where the description of how to operate a piece of software was followed by a scientist to acquire data. In those case we definitely want to capture that information. One way we could handle that is by treating those as explicit operator instructions. In fact, one part of the `protc` domain model which is not currently included in `protc/ur` is the that the meaning of a step of a protocol can be explicitly delegated to the executor performing the protocol. In such cases text could be tagged with something along the lines of `delegated-instructions`. The risk of this approach is that, technically, everything in a protocol could be tagged as such and critical information would not be extracted as a result.

Beyond technical issues and desires for future implementation there are additional reflections from developing protc/ur. One thing that is striking when reviewing the literature and reflecting on the development of protc/ur is that a number of projects (including the first iteration of our own) missed the importance of aspects in their domain models. This oversight is all the more unfortunate as the notion of aspect as it is used here has the same origin as dimensional analysis[53] and is fundamental to the practice of modern science [72, 18]. Missing aspects from the domain model can also have the unfortunate side effect of making the domain model significantly more complex. For example aspects are critical for capturing elements of a protocol such as ”anaesthesia” or ”anaesthetic administration” that are extremely complex processes that seem to require deep modeling, but are in fact trivially handled as arbitrary boolean measures of the behavior of a system. How do you know whether your mouse is anaesthetized? Well it either is or it isn’t but how do we tell? We pinch its toe to see if its reflex arcs are still active (i.e., that it doesn’t jerk in response). If it jerks it is not anaesthetized, and if it doesn’t jerk, it is anaesthetized. There are many other arbitrary aspects of complex systems that can be abstracted away as a boolean measure[54] for the purposes of the protocol.

The future of formal protocols in the life sciences is of considerable interest to those in the design domain, and the vision of a fully instrumented lab is as enticing now as the idea of a fully instrumented corporate office was in the 70s [69, 19, 56][55].

It is also conceivable that a more formal and rigorous approach to documenting methodology might also help damp the long smoldering reproducibility crisis affecting the life sciences [46, 48, 15, 3]. Regardless of the actual underlying causes for issues with reproducibility, better tools that make scientists lives easier could go a long way toward making it easier to do the right thing when it comes to data management and data sharing.

[53] James Clerk Maxwell

[54] and a sub-protocol if we are being complete

[55] Though perhaps the funding to make it so for labs is not quite as substantial as for the office.

3.5 Tables, Figures, and Listings

Table 3.1: **Core elements** `input` and `output`: physical inputs an outputs of processes e.g. mouse, forceps, etc. `aspect`: abstract dimensions independent of units, e.g. count, density, distance, etc. `parameter*`: fully factored concrete values, can be actualized directly e.g. grams. `invariant`: partially factored values, need additional info to actualize e.g. mol/l. **Extended elements** `executor-verb`: verbs where the subject is the person performing the protocol. `objective*`: applied inconsistently, generally covers high level invariants lacking concrete values or `telos` the end for which a thing is done. `black-box-component`: parts of `inputs` or `outputs` that are not physically separate e.g. named anatomical locations, injection sites, landmarks, etc. `symbolic-input`: digital equivalent of `input` when a process symbolic, computational, or digital, e.g. the symbolic-input to analysis. `*measure`: explicit statements that one or more values are measured.

core elements	count
`input`	6180
`output`	98
`aspect`	5123
`parameter*`	3291
`invariant`	2775

extended elements	count
`executor-verb`	4567
`objective*`	726
`black-box-component`	876
`symbolic-input`	379
`*measure`	75

Table 3.2: The results of the SPARQL query from listing 3.15 expressed in english as **Show me `{datasets}` generated by a `{protocol}` that involved `{magnification}` and `{nerves}` or `{ganglia}`. Include `{species}`, `{region}`, `{value}`, and `{units}`.** This is an example of the kind of query that `protc/ur` can answer in combination with the larger SPARC Knowledge Graph. First is a free text version of the query that we want to perform before it is translated (by a human) into SPARQL (the full text of the SPARQL query is in the appendix). Entities that correspond to variables in the query are surrounded by `{}`. In this version of the query `magnification`, `nerves`, and `ganglia` directly and must be translated to their respective ontology identifiers. The query is run by calling `magnification-tax-reg-val` and all results are shown. The results show that `protc/ur` makes it possible to find protocols and datasets based on aspects that were measured as part of the protocol. As we will see in the next query example, the additional columns hint at the ability to further refine query results based on additional criteria, such as species, region, or even by the quantitative value of a parameter. Datasets and protocols are links also in Table 3.5.

dataset	protocol	species	region	val	unit
d484110a-...	22831	Rattus norvegicus	inferior vagus X ganglion	40	fold
d484110a-...	22831	Rattus norvegicus	inferior vagus X ganglion	63	fold
e4bfb720-...	22831	Rattus norvegicus	inferior vagus X ganglion	40	fold
e4bfb720-...	22831	Rattus norvegicus	inferior vagus X ganglion	63	fold
6fa2666c-...	19131	Rattus norvegicus	vagus nerve	20	fold
ff6eb067-...	19143	Mus musculus	vagus nerve	20	fold

Table 3.3: The results of the SPARQL query from Figure 3.16 expressed in english as **Show me `{datasets}` generated by a `{protocol}` where a `{rat}` was given less than** `{100}` `mg/kg` **of `{ketamine}`.** This is an example of the kind of query that `protc/ur` can answer. The organization and notation are the same as in the previous query example. In this version of the query `rat` and `ketamine` cannot be used directly and must be translated to their respective ontology identifiers `NCBITaxon:10116` and `CHEBI:6121`. The query is run by calling `species-mg/kg` and the results are shown for 100 mg/kg, and 1000 mg/kg. The results show that `protc/ur` makes it possible to find protocols and datasets that match both simple queries such as ”show me protocols that involve a rat” or ”show me protocols that involve ketamine” as well as more complex queries such as ”show me protocols where the value of a parameter about a particular type of subject satisfies certain restrictions.” Datasets and protocols are links also in Table 3.5.

dataset	protocol	drug	value_lt
dataset:378d07cf-4b8b-49e8-a069-e96810e68b57	pio.api:19640	ketamine	55
dataset:fd07322c-ac30-488f-a6db-f5ff52c69e1b	pio.api:20256	ketamine	55
dataset:3a7ccb46-4320-4409-b359-7f4a7027bb9c	pio.api:23160	ketamine	80
dataset:abd1da38-dbb7-46a7-96ef-58ce33c0ebd9	pio.api:23160	ketamine	80

Table 3.4: Results for the same query (fig 3.16) that generated Table 3.3 but with `{1000}` `mg/kg`. Datasets and protocols are links also in Table 3.5.

dataset	protocol	drug	value_lt
dataset:378d07cf-4b8b-49e8-a069-e96810e68b57	pio.api:19640	ketamine	55
dataset:fd07322c-ac30-488f-a6db-f5ff52c69e1b	pio.api:20256	ketamine	55
dataset:3a7ccb46-4320-4409-b359-7f4a7027bb9c	pio.api:23160	ketamine	80
dataset:abd1da38-dbb7-46a7-96ef-58ce33c0ebd9	pio.api:23160	ketamine	80
dataset:e4bfb720-a367-42ab-92dd-31fd7eefb82e	pio.api:22831	ketamine	275
dataset:f58c75a2-7d86-439a-8883-e9a4ee33d7fa	pio.api:22832	ketamine	275
dataset:bc4071fd-aba1-4fe5-a59e-3da5affbc5fb	pio.api:22833	ketamine	275
dataset:d484110a-e6e3-4574-aab2-418703c978e2	pio.api:22831	ketamine	275

Table 3.5: Links to published SPARC datasets and protocols that appear in query results.

published dataset	dsid	protocols.io view html
`http://sparc.science/datasets/10`	d484	`http://protocols.io/view/22831`
`http://sparc.science/datasets/11`	e4bf	`http://protocols.io/view/22831`
`http://sparc.science/datasets/12`	f58c	`http://protocols.io/view/22832`
`http://sparc.science/datasets/123`	bc40	`http://protocols.io/view/22833`
`http://sparc.science/datasets/16`	6fa2	`http://protocols.io/view/19131`
`http://sparc.science/datasets/20`	378d	`http://protocols.io/view/19640`
`http://sparc.science/datasets/21`	fd07	`http://protocols.io/view/20256`
Not yet published, private protocol	3a7c	`http://protocols.io/view/23160`
Not yet published, private protocol	abd1	`http://protocols.io/view/23160`
Not yet published, private protocol	ff6e	`http://protocols.io/view/19143`

Table 3.6: Links to protocols.io HTML view of shared protocols curated and analyzed as part of this study.

http://protocols.io/view/19135.html
http://protocols.io/view/19131.html
http://protocols.io/view/19295.html
http://protocols.io/view/19269.html
http://protocols.io/view/19271.html
http://protocols.io/view/19270.html
http://protocols.io/view/19153.html
http://protocols.io/view/19134.html
http://protocols.io/view/19354.html
http://protocols.io/view/19355.html
http://protocols.io/view/18539.html
http://protocols.io/view/18595.html
http://protocols.io/view/18578.html
http://protocols.io/view/19283.html
http://protocols.io/view/19401.html
http://protocols.io/view/18769.html
http://protocols.io/view/18925.html
http://protocols.io/view/18994.html
http://protocols.io/view/19220.html
http://protocols.io/view/19364.html
http://protocols.io/view/19346.html
http://protocols.io/view/18394.html
http://protocols.io/view/19174.html
http://protocols.io/view/19262.html
http://protocols.io/view/19266.html
http://protocols.io/view/19253.html
http://protocols.io/view/19342.html
http://protocols.io/view/19576.html
http://protocols.io/view/20025.html
http://protocols.io/view/20306.html
http://protocols.io/view/19927.html
http://protocols.io/view/19640.html
http://protocols.io/view/19139.html
http://protocols.io/view/19798.html
http://protocols.io/view/19095.html
http://protocols.io/view/19206.html
http://protocols.io/view/21193.html
http://protocols.io/view/21417.html
http://protocols.io/view/22833.html
http://protocols.io/view/22831.html
http://protocols.io/view/22832.html
http://protocols.io/view/22900.html
http://protocols.io/view/22977.html
http://protocols.io/view/22953.html
http://protocols.io/view/22844.html
http://protocols.io/view/22868.html
http://protocols.io/view/22948.html
http://protocols.io/view/24077.html
http://protocols.io/view/19227.html
http://protocols.io/view/20254.html
http://protocols.io/view/25122.html
http://protocols.io/view/25121.html
http://protocols.io/view/20256.html
http://protocols.io/view/25230.html
http://protocols.io/view/19107.html
http://protocols.io/view/22894.html
http://protocols.io/view/22875.html
http://protocols.io/view/22863.html
http://protocols.io/view/22895.html
http://protocols.io/view/22888.html
http://protocols.io/view/22890.html
http://protocols.io/view/22889.html
http://protocols.io/view/22891.html
http://protocols.io/view/22887.html
http://protocols.io/view/26887.html
http://protocols.io/view/26886.html
http://protocols.io/view/26841.html
http://protocols.io/view/26704.html
http://protocols.io/view/26709.html
http://protocols.io/view/26687.html
http://protocols.io/view/26737.html
http://protocols.io/view/26688.html
http://protocols.io/view/29396.html
http://protocols.io/view/19255.html
http://protocols.io/view/31001.html
http://protocols.io/view/31076.html
http://protocols.io/view/31077.html
http://protocols.io/view/31143.html
http://protocols.io/view/30948.html
http://protocols.io/view/31158.html
http://protocols.io/view/34589.html

Table 3.7: Links to protocols curated and analyzed as part of this study that have not been shared publicly.

http://protocols.io/view/19341.html
http://protocols.io/view/19088.html
http://protocols.io/view/19127.html
http://protocols.io/view/18947.html
http://protocols.io/view/19143.html
http://protocols.io/view/18417.html
http://protocols.io/view/23160.html
http://protocols.io/view/24481.html
http://protocols.io/view/25090.html
http://protocols.io/view/25880.html
http://protocols.io/view/25917.html
http://protocols.io/view/26301.html
http://protocols.io/view/25923.html
http://protocols.io/view/25817.html
http://protocols.io/view/28180.html
http://protocols.io/view/27863.html

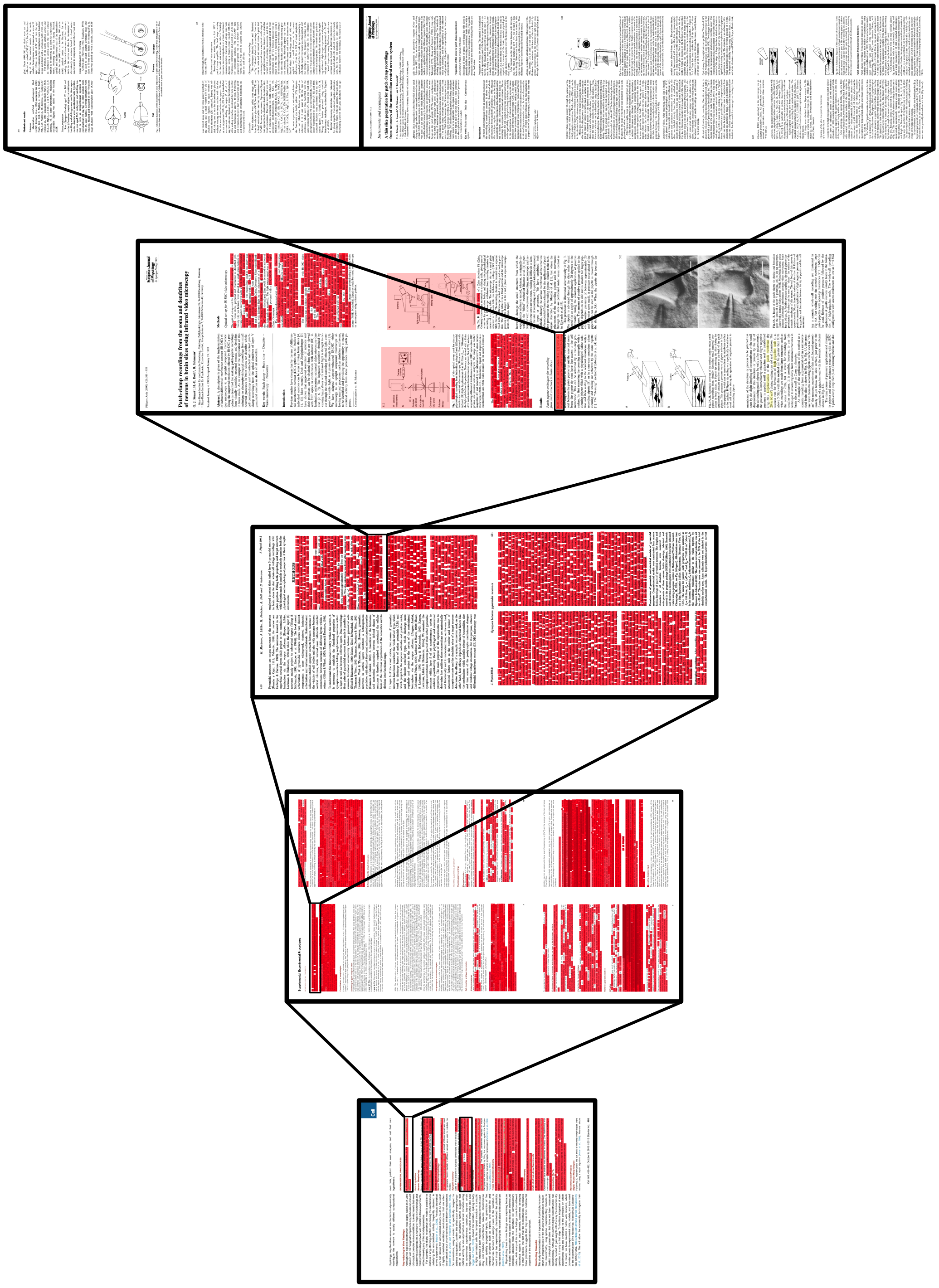


Figure 3.1: One branch of the citation tree for a single sentence in the methods section of [67] traced back to the very first papers on patch clamp electrophysiology [20, 8]. Nearly every sentence in every methods section expands to an entire methods section of another paper, until finally ending in a paper dedicated entirely to describing the original methodology.

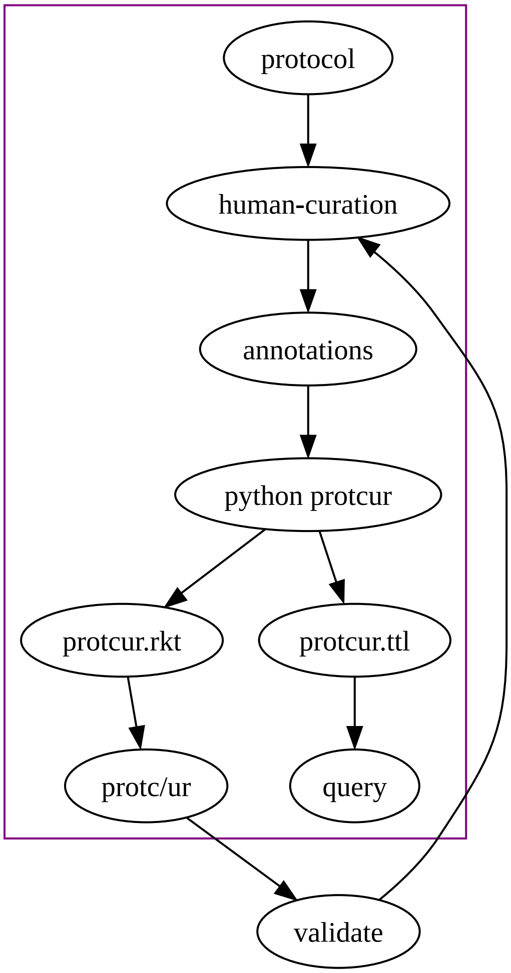
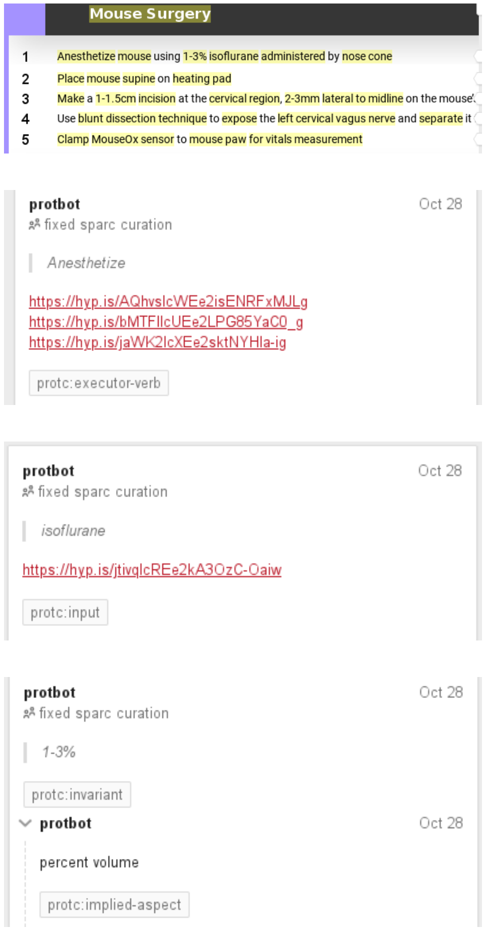


Figure 3.2: An overview of the parts of the protcur system showing the flow from a raw protocol (e.g. on protocols.io) to checkable and queryable outputs. Starting from an HTML or pdf protocol, human-curation produces Hypothes.is web annotations that are anchored to the original document. In `python protcur`, `hyputils` retrieves annotations from the Hypothes.is API as JSON, `pysercomb` and `protcur` create `protcur.rkt` and along with `sparcur` create `protcur.ttl` (`protcur.ttl` includes additional alignment and normalization). On the left `protcur.rkt` is checked for correctness by the Racket implementation of `protc/ur`. Errors are corrected by updating the human curation using a set of tags for non-destructive editing. On the right `protcur.ttl` is loaded along with SPARC datasets and ontologies into the RDF representation of the SPARC Knowledge Graph and interrogated using the SPARQL query language. It is possible to parse `protc/ur` back into python and export to rdf. In fact the python `protc/ur` parser runs internally on the same representation that is exported to `protcur.rkt` but in practice our pipelines don't parse `protcur.rkt` into `protcur.ttl`, we go straight from the annotations to both `.rkt` and `.ttl`.

1



Mouse Surgery

1 Anesthetize mouse using 1-3% isoflurane administered by nose cone
2 Place mouse supine on heating pad
3 Make a 1-1.5cm incision at the cervical region, 2-3mm lateral to midline on the mouse'
4 Use blunt dissection technique to expose the left cervical vagus nerve and separate it
5 Clamp MouseOx sensor to mouse paw for vitals measurement

protbot Oct 28
fixed sparc curation

Anesthetize

https://hyp.is/AQhvslcWEe2isENRFxMJLg
https://hyp.is/bMTFllcUEe2LPG85YaC0_g
https://hyp.is/jaWK2lcXEe2sktNYHla-ig

protc:executor-verb

protbot Oct 28
fixed sparc curation

isoflurane

https://hyp.is/jtivqlcREe2kA3OzC-Oaiw

protc:input

protbot Oct 28
fixed sparc curation

1-3%

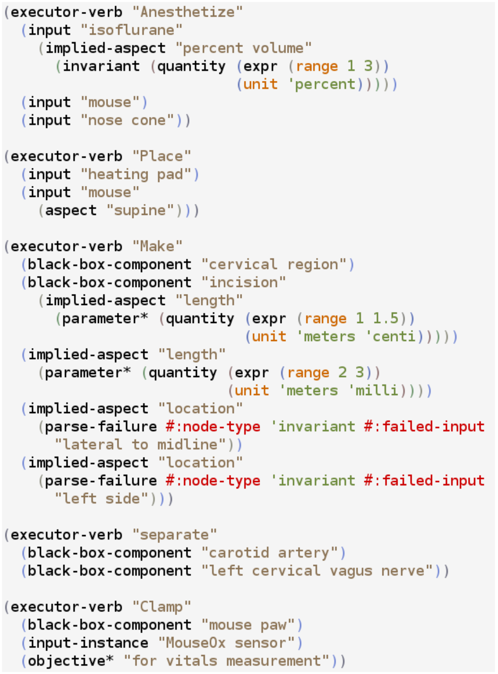
protc:invariant

protbot Oct 28

percent volume

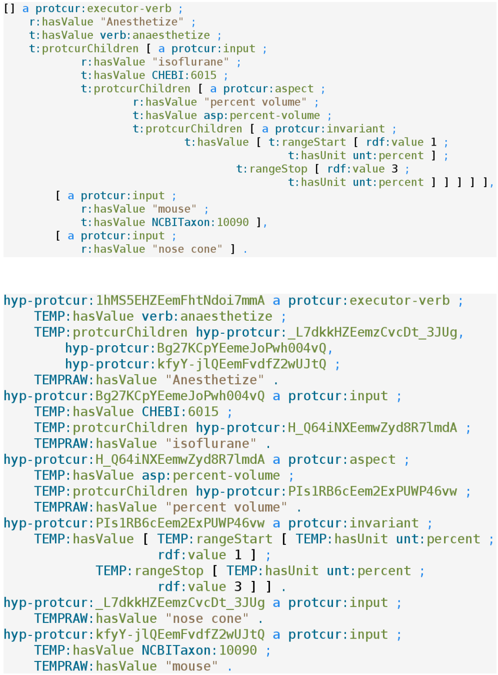
protc:implied-aspect

2



```
(executor-verb "Anesthetize"
  (input "isoflurane"
    (implied-aspect "percent volume"
      (invariant (quantity (expr (range 1 3))
                           (unit 'percent)))))
  (input "mouse")
  (input "nose cone"))

(executor-verb "Place"
  (input "heating pad")
  (input "mouse"
    (aspect "supine")))

(executor-verb "Make"
  (black-box-component "cervical region")
  (black-box-component "incision"
    (implied-aspect "length"
      (parameter* (quantity (expr (range 1 1.5))
                            (unit 'meters 'centi)))))
  (implied-aspect "length"
    (parameter* (quantity (expr (range 2 3))
                          (unit 'meters 'milli))))
  (implied-aspect "location"
    (parse-failure #:node-type 'invariant #:failed-input
      "lateral to midline"))
  (implied-aspect "location"
    (parse-failure #:node-type 'invariant #:failed-input
      "left side")))

(executor-verb "separate"
  (black-box-component "carotid artery")
  (black-box-component "left cervical vagus nerve"))

(executor-verb "Clamp"
  (black-box-component "mouse paw")
  (input-instance "MouseOx sensor")
  (objective* "for vitals measurement"))
```

3



```
[] a protcur:executor-verb ;
   r:hasValue "Anesthetize" ;
   t:hasValue verb:anaesthetize ;
   t:protcurChildren [ a protcur:input ;
           r:hasValue "isoflurane" ;
           t:hasValue CHEBI:6015 ;
           t:protcurChildren [ a protcur:aspect ;
                   r:hasValue "percent volume" ;
                   t:hasValue asp:percent-volume ;
                   t:protcurChildren [ a protcur:invariant ;
                           t:hasValue [ t:rangeStart [ rdf:value 1 ;
                                           t:hasUnit unt:percent ] ;
                                   t:rangeStop [ rdf:value 3 ;
                                           t:hasUnit unt:percent ] ] ] ] ],
       [ a protcur:input ;
           r:hasValue "mouse" ;
           t:hasValue NCBITaxon:10090 ],
       [ a protcur:input ;
           r:hasValue "nose cone" ] .
```

```
hyp-protcur:1hMS5EHZEemFhtNdoi7mmA a protcur:executor-verb ;
    TEMP:hasValue verb:anaesthetize ;
    TEMP:protcurChildren hyp-protcur:_L7dkkHZEemzCvcDt_3JUg,
        hyp-protcur:Bg27KCpYEemeJoPwh004vQ,
        hyp-protcur:kfyY-jlQEemFvdfZ2wUJtQ ;
    TEMPRAW:hasValue "Anesthetize" .
hyp-protcur:Bg27KCpYEemeJoPwh004vQ a protcur:input ;
    TEMP:hasValue CHEBI:6015 ;
    TEMP:protcurChildren hyp-protcur:H_Q64iNXEemwZyd8R7lmdA ;
    TEMPRAW:hasValue "isoflurane" .
hyp-protcur:H_Q64iNXEemwZyd8R7lmdA a protcur:aspect ;
    TEMP:hasValue asp:percent-volume ;
    TEMP:protcurChildren hyp-protcur:PIs1RB6cEem2ExPUWP46vw ;
    TEMPRAW:hasValue "percent volume" .
hyp-protcur:PIs1RB6cEem2ExPUWP46vw a protcur:invariant ;
    TEMP:hasValue [ TEMP:rangeStart [ TEMP:hasUnit unt:percent ;
                rdf:value 1 ] ;
            TEMP:rangeStop [ TEMP:hasUnit unt:percent ;
                rdf:value 3 ] ] .
hyp-protcur:_L7dkkHZEemzCvcDt_3JUg a protcur:input ;
    TEMPRAW:hasValue "nose cone" .
hyp-protcur:kfyY-jlQEemFvdfZ2wUJtQ a protcur:input ;
    TEMP:hasValue NCBITaxon:10090 ;
    TEMPRAW:hasValue "mouse" .
```

Figure 3.3: An in-depth example of protocol content as it is transformed from one representation to the next by human and machine pipelines. 1. `html -> json` shows the annotated text of an HTML protocol. Below that is the rendered JSON for a subset of the annotations. 2. `rkt` shows simplified `protc/ur` s-expressions. 3. `ttl` shows simplified ttl where annotation ids are treated as RDF blank nodes so that it is easier to see the equivalence to the `protc/ur` on the left. Below that is the actual ttl serialization where annotation ids (`hyp-protcur:`) are retained.

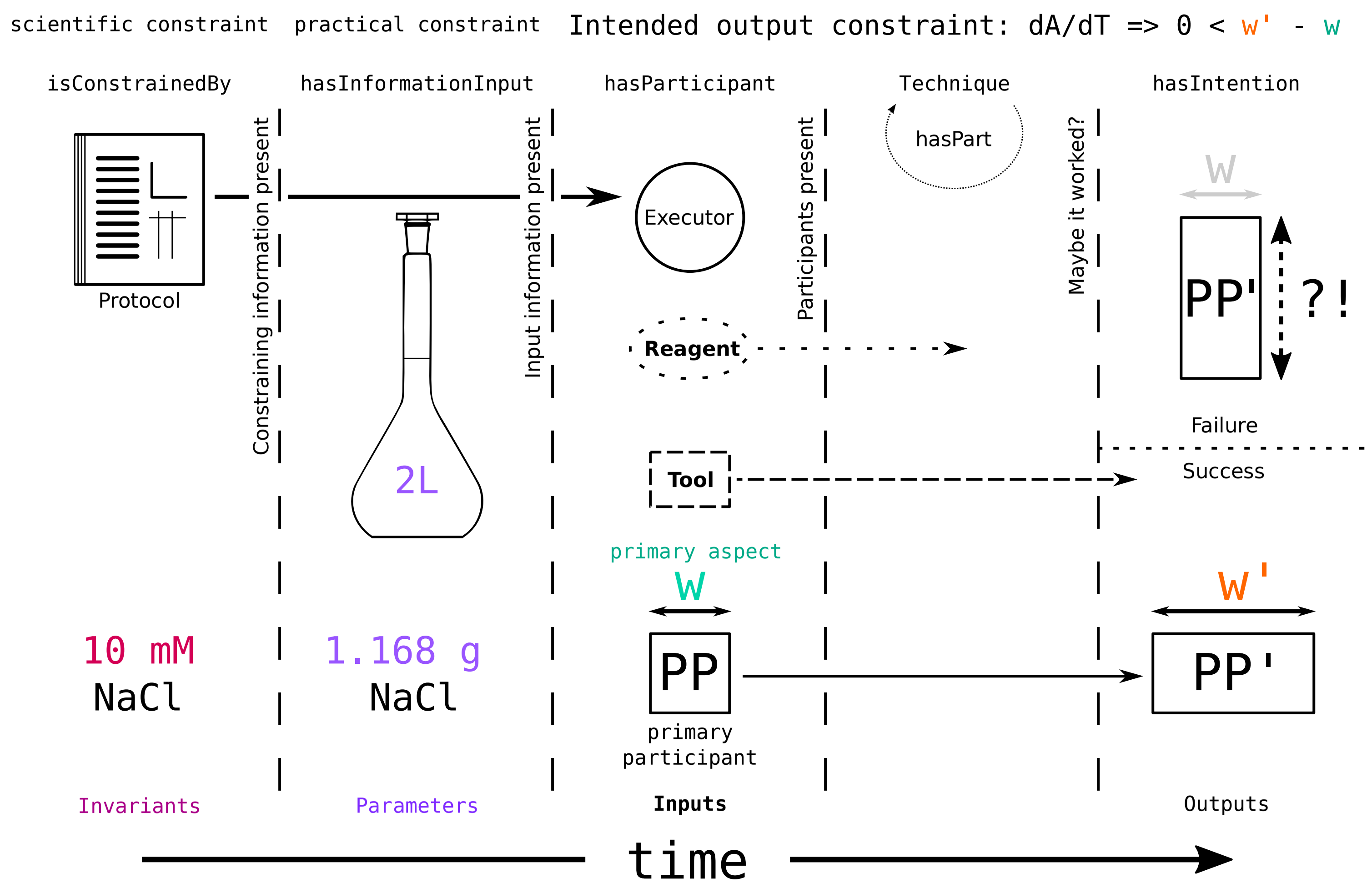


Figure 3.4: A diagram of the `protoc` domain model expressed in OWL. Each successive phase in the performance of a protocol adds information inputs to realize invariants into parameters that can be actualized directly. Arrows associated with participants show whether the particular participant survives the performance of the protocol or is consumed or transformed into something with different name (e.g. reagents). The relationship of aspects to participants is shown, as is the notion that the definition of a protocol needs to be independent of its success or failure, which is achieved by defining techniques based on their intended outcomes. This example would be a ”widening technique.”

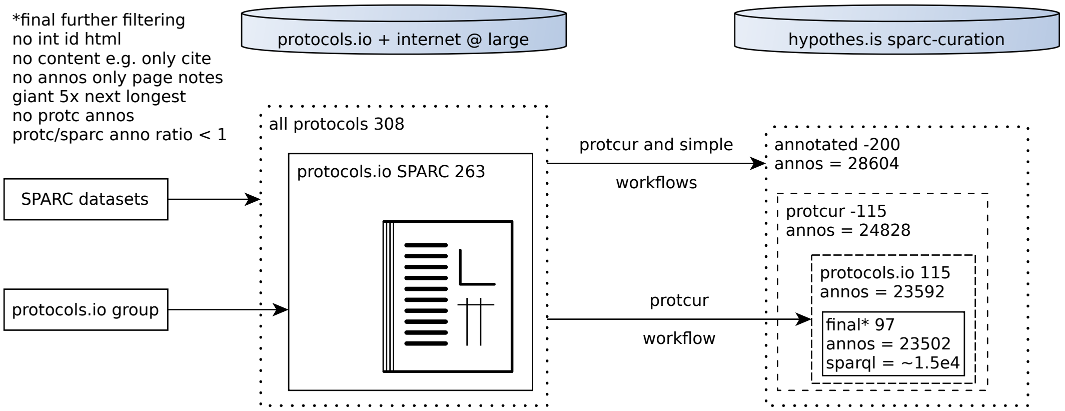


Figure 3.5: A diagram of the filtering process to select the protocols and annotations that are included for analysis.

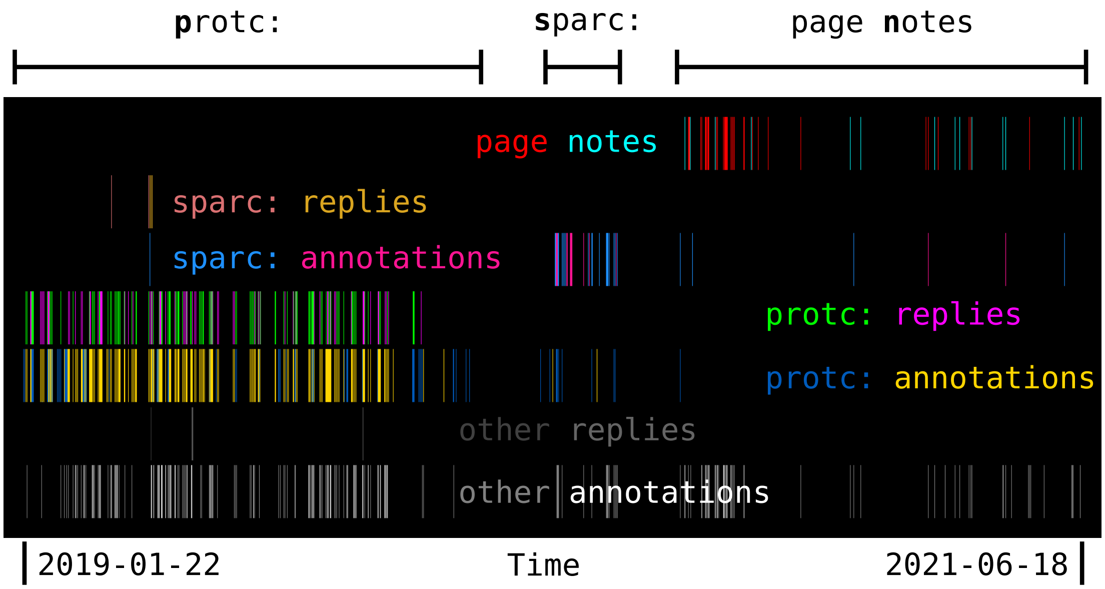


Figure 3.6: Timeline of protocol curation events for the `protc:` `sparc:` and `page notes` workflows.

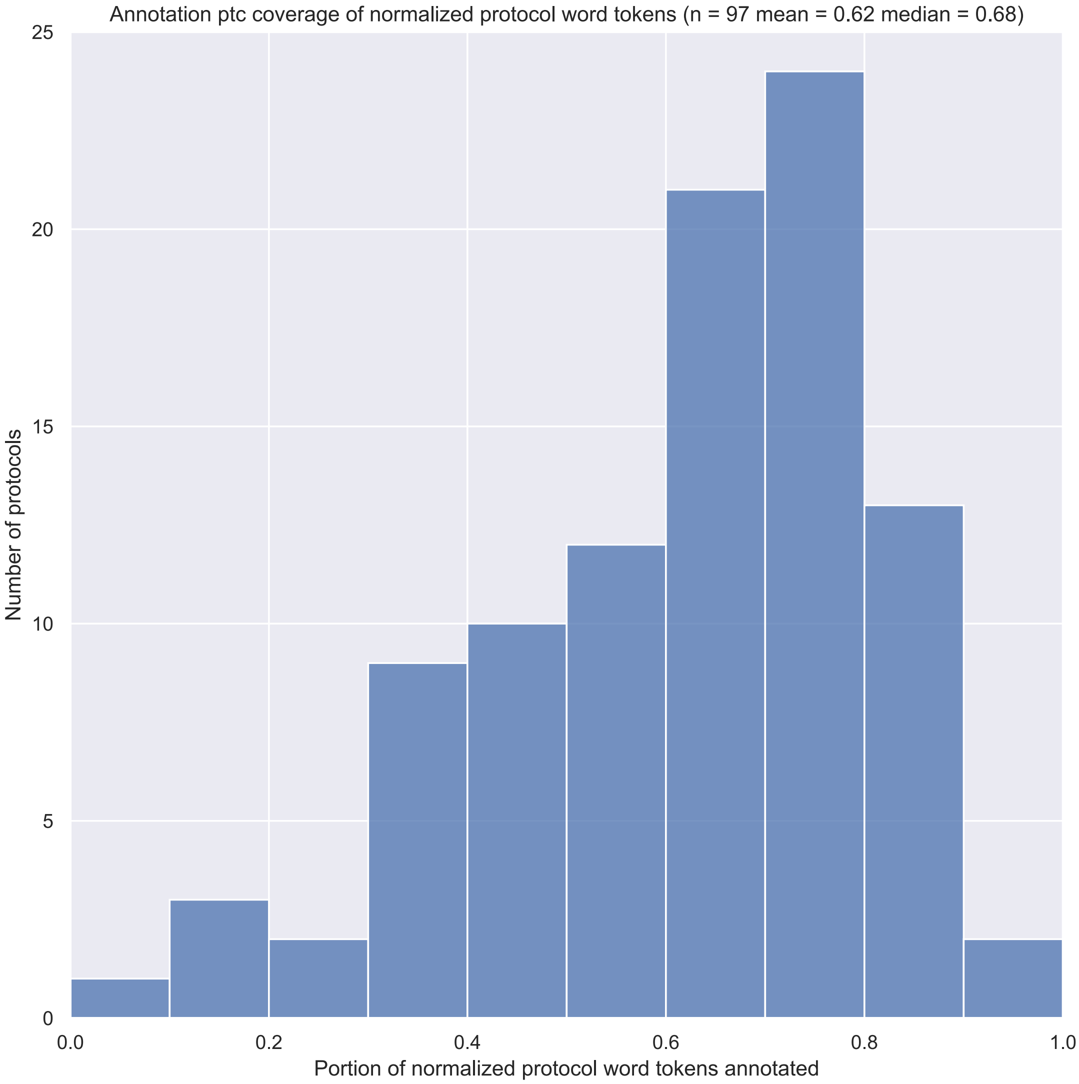


Figure 3.7: A histogram of curation coverage for the 97 protocols in the corpus with a mean curation coverage of 0.6182 and a median of 0.6753.

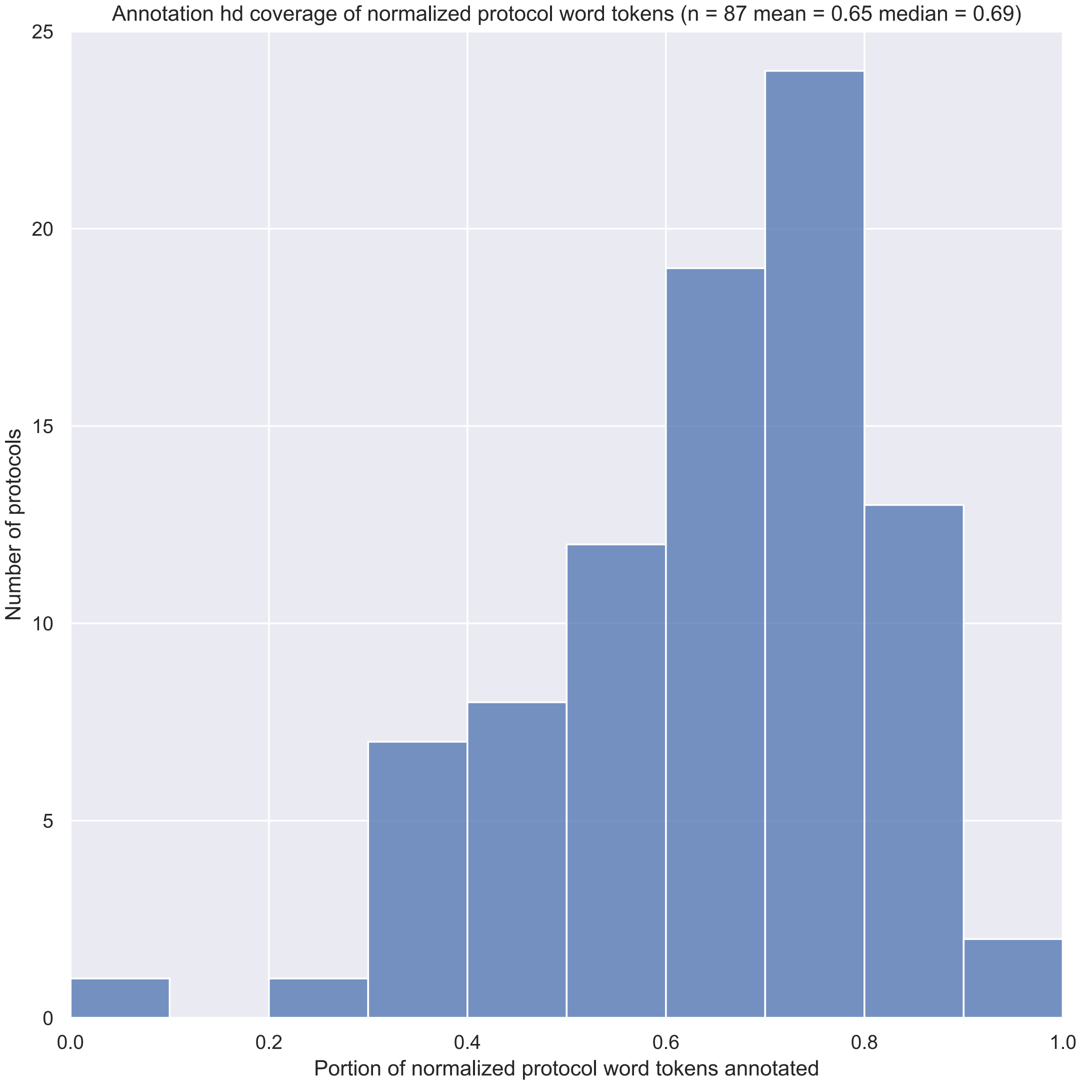


Figure 3.8: A histogram of curation coverage as in Figure 3.7 except that the protocols have been filtered to exclude protocol with curation depth < 0.1. As a result of filtering there are 87 protocols with a mean curation coverage of 0.6477 and a median of 0.6895.

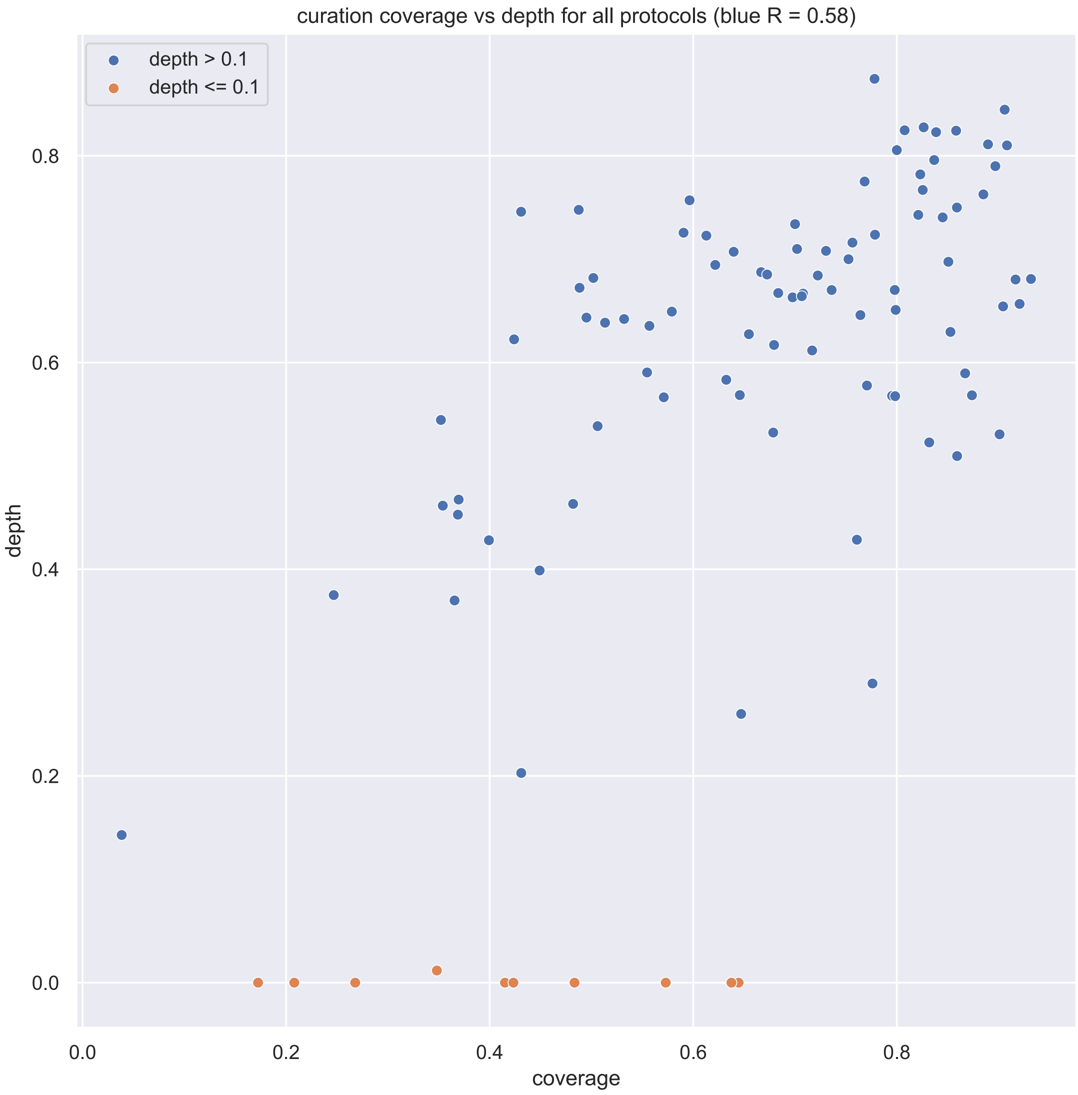


Figure 3.9: A scatter plot of curation depth vs curation coverage. The correlation coefficient for the blue (depth > 0.1) subset is approximately 0.58.

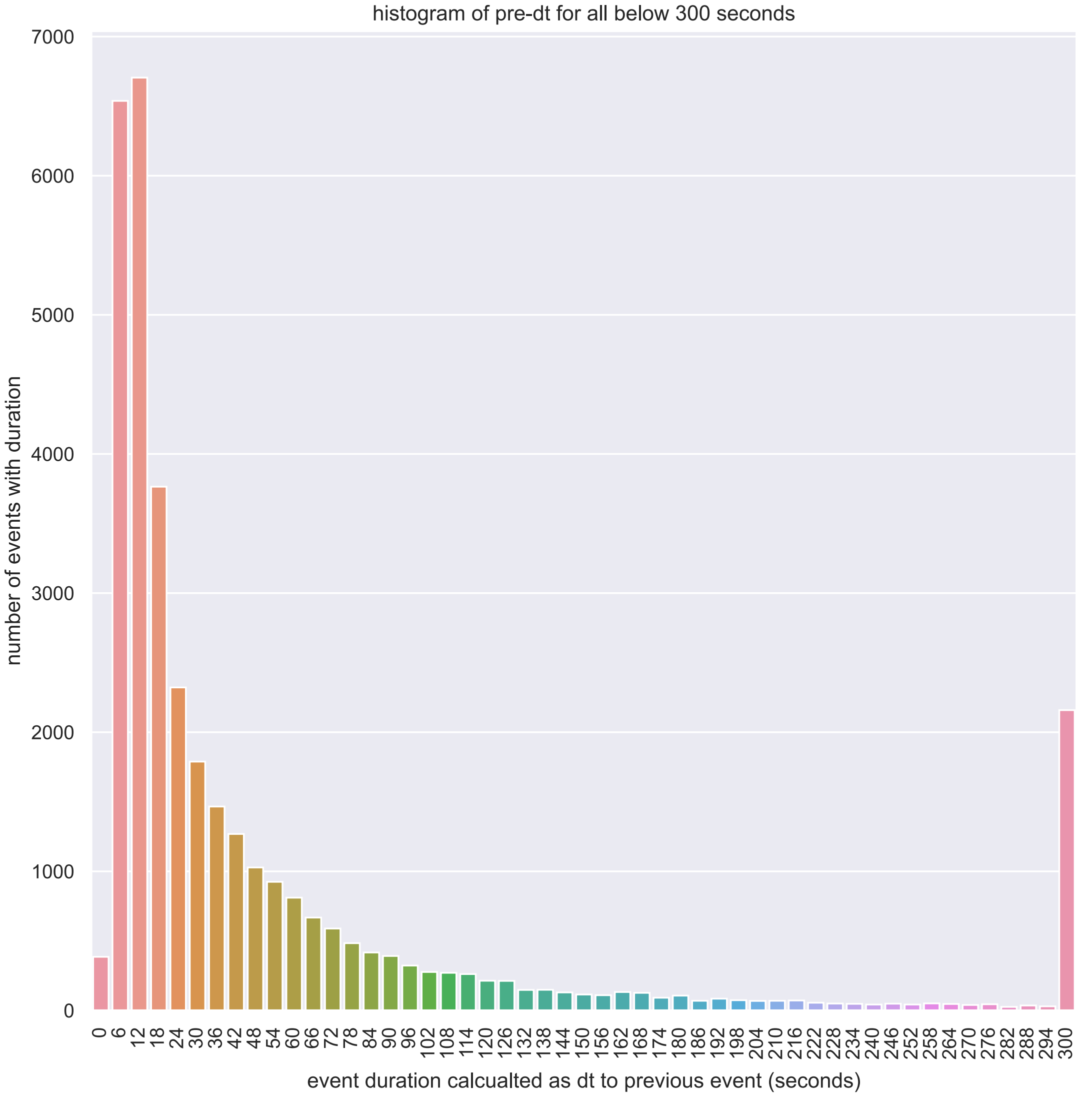


Figure 3.10: Histogram of event durations for all curation events below 300 seconds. Durations over 300 seconds are grouped into the last bin.

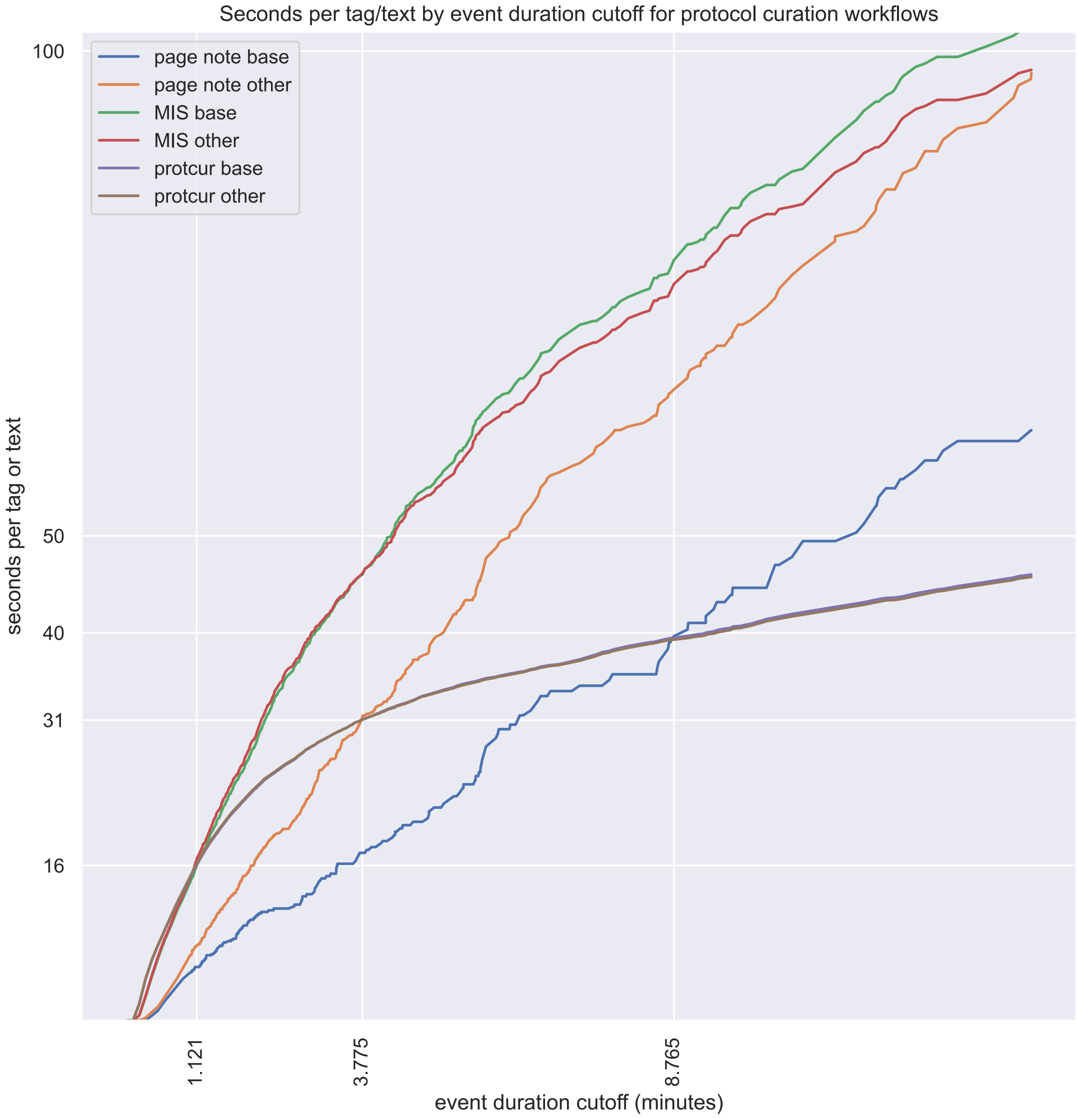


Figure 3.11: Efficiency for three different curation workflows under different assumptions about the maximum time working on a protocol without making an annotation. The x axis is the maximum minutes between consecutive curation events before the gap is considered to be a period where no curation occurred (see appendix). The y axis is the average seconds per annotation tag or text (there can be multiple tags/text per annotation, especially on page notes). In the legend base vs other indicates the events used to calculate time per event. Base includes only events where tags match the curation process. Other includes all events.

```
(executor-verb "Anesthetize"
  (input "isoflurane"
    (implied-aspect "percent volume"
      (invariant (quantity (expr (range 1 3)) (unit 'percent)))))
  (input "mouse")
  (input "nose cone"))
(executor-verb "Place"
  (input "heating pad")
  (input "mouse" (aspect "supine")))
(executor-verb "Make"
  (black-box-component "cervical region")
  (black-box-component "incision"
    (implied-aspect "length"
      (parameter* (quantity (expr (range 1 1.5))
                            (unit 'meters 'centi)))))
  (implied-aspect "length"
    (parameter* (quantity (expr (range 2 3)) (unit 'meters 'milli))))
  (implied-aspect "location"
    (parse-failure #:node-type 'invariant
                   #:failed-input "lateral to midline"))
  (implied-aspect "location"
    (parse-failure #:node-type 'invariant #:failed-input "left side")))
(executor-verb "separate"
  (black-box-component "carotid artery")
  (black-box-component "left cervical vagus nerve"))
(executor-verb "Clamp"
  (black-box-component "mouse paw")
  (input-instance "MouseOx sensor")
  (objective* "for vitals measurement"))
(executor-verb "Affix"
  (input "GRIN lens"
    (implied-aspect "dimensions"
      (parameter*
       (dimensions (quantity 1 (unit 'meters 'milli))
                   (quantity 9 (unit 'meters 'milli))))))
  (input "cuff")
  (input "super glue"))
```

Figure 3.12: A simplified `protc/ur` representation of one section of a SPARC protocol [a]

[a] `https://www.protocols.io/view/19143.html`

```
(output "DIO mouse"
  (aspect "type identifier" (invariant "JAX:380050"))
  (aspect "mass" (parameter* (quantity 52 (unit 'grams))))
  (input "mouse"
    (aspect "age" (parameter* (quantity 6 (unit 'weeks)))))
  (input "mouse diet"
    (aspect "ad libitum" (parameter* (bool #t)))
    (aspect "type identifier" (invariant "D12492"))))
```

Figure 3.13: A protocol specifying the basic inputs, aspects, and parameters/invariants that define a diet induced obesity mouse.

```
(protc:aspect "count"
  (protc:input "mouse"
    (protc:parameter* (param:quantity 10))))

;/tmp/protocol:3:0: protc:aspect: expected one of these identifiers:
 ↪  `aspect', `parameter*', `invariant', `vary', `TODO',
 ↪  `circular-link', `*measure', or `param:parse-failure'
;  at: protc:input
;  in: (protc:aspect "count" (protc:input "mouse" (protc:parameter*
 ↪  (param:quantity 10))))
;  location...:
;   /tmp/protocol:4:3
```

```
;  context...: ...

(protc:input "mouse"
  (protc:aspect "count"
   (protc:parameter* (param:quantity 10))))
```

Figure 3.14: This is an example of how `protc/ur` can assist in the creation and curation of protocols. The first code example shows an incorrect statement where `protc:aspect` and `protc:input` are inverted. This happens in the current curation process e.g. because a curator pastes a link to the wrong annotation. The second box shows the error produced when running the first code example. When processed by `protc/ur` it produces an error because it violates the syntactic constraints of the `protc/ur` language. The location of the error, the set of expected values, and the incorrect value that was provided are indicated in the error. This information is usually sufficient for a trained user to identify and correct the underlying issue in the human curation step, or in the source when writing `protc/ur` directly. The third box (second code example) shows the corrected statement where `protc:aspect` is nested inside `protc:input`. There are numerous similar syntactic rules implemented in `protc/ur` using Racket's `syntax/parse` library.

```
SELECT DISTINCT
?dataset ?protocol
?id_species ?species
?id_region ?region
?value ?unit
WHERE {
  ?protocol a sparc:Protocol ;
    TEMP:protocolInvolvesAspect ?ast_asp ;
    TEMP:protocolInvolvesInput  ?ast_species ;
    ?_                          ?ast_region .
  ?ast_asp a protcur:aspect ;
    TEMP:hasValue asp:magnification ;
    TEMP:protcurChildren [ TEMP:hasValue [ rdf:value ?value ;
                                           TEMP:hasUnit ?unit ] ] .
  ?ast_species rdf:type protcur:input ;
    TEMP:hasValue ?id_species .
                  ?id_species rdfs:subClassOf+ NCBITaxon:33208 ; #
                   ↪  metazoa
                              rdfs:label ?sl .
  BIND (str(?sl) AS ?species)
  ?ast_region TEMP:protcurChildren* ?ast_reg ;
    TEMP:hasValue ?id_region .
                  ?id_region rdfs:subClassOf+ ?nerves_and_ganglia ;
                             rdfs:label ?rl .
  BIND (str(?rl) AS ?region)
  VALUES ?nerves_and_ganglia {
    UBERON:0000122  # neuron projection bundle
    UBERON:0000045  # ganglion
  }
  OPTIONAL {
    { ?dataset TEMP:hasProtocol ?protocol } UNION
    { ?protocol
      TEMP:priorInformationalConstraintOnProcessThatGenerated
      ?dataset }}}
ORDER BY ?region ?dataset ?protocol
```

Figure 3.15: A SPARQL query to find protocols by objective magnification returning associated datasets, species, anatomical regions, and values for objective magnification.

```
SELECT DISTINCT
?dataset
?protocol
(str(?label_drug) AS ?drug)
?value_lt
WHERE {
  VALUES ?t {protcur:invariant protcur:parameter} .
  ?ast_inv a ?t .
  ?ast_inv TEMP:hasValue ?quantity .
  ?quantity TEMP:hasUnit unit:milligram%20%2F%20kilogram .
  ?quantity rdf:value ?value_lt .
  FILTER (?value_lt < ?limit)

  ?ast_drug a protcur:input .
  ?ast_drug TEMP:protcurChildren+ ?ast_child .
                                  ?ast_child TEMP:hasValue ?quantity .
  ?ast_drug TEMP:hasValue ?id_drug .
                          ?id_drug rdfs:label ?label_drug .

  ?protocol a sparc:Protocol .
  ?protocol TEMP:protocolInvolvesInput ?ast_drug .

  ?protocol TEMP:protocolInvolvesInput ?ast_in_sp .
  ?ast_in_sp rdf:type protcur:input .
  ?ast_in_sp TEMP:hasValue ?species .

  OPTIONAL {
    { ?dataset TEMP:hasProtocol ?protocol } UNION
    { ?protocol
      TEMP:priorInformationalConstraintOnProcessThatGenerated
      ?dataset }}
}
ORDER BY ?label_input ?value_lt
```

Figure 3.16: A SPARQL query to find protocols by species, drug, and dose returning associated datasets and values for dose in mg/kg.

Conclusion

Though formalization has provided the regular drumbeat for the work underlying this dissertation, the repeated motif, with variation, that forms the overarching theme for this work is that of making experimental scientific methodology explicit in information systems as a means to organize scientific data and knowledge.

In the Neuron Phenotype Ontology the central organizing principle for its core formal relationships is based on data modality and the methodology used to collect that data. The leaves of the tree of phenotype predicates presented in Fig. 3 of Chapter One all specify how that phenotype was `DeterminedBy` a particular methodology. Although it is not explicitly shown, that hierarchy can be extended all the way down to individual protocols. A future objective is thus to build an orthogonal model that can be used to generate the hierarchy of phenotype predicates from a combination of data modality, techniques, and the exact protocols applied to determine the value of a phenotype[56]. Some might say that this is more granularity than is needed, but it reflects the fact that we cannot assume *a priori* that the value of a phenotype measured by one protocol actually implies that the same value will always be obtained by the application of another similar but not exactly identical protocol (and, as noted, even this level might not be sufficient to account for important sources of variability[56]). The issue is well known to the experimental community and is expressed by the phrase ”in our hands” to indicate such uncertainty. While

[56] In principle the hierarchy could be extended even further to the individual performances of a protocol, including the executor, runtime parameters, etc. However, we are still far from having sufficiently instrumented laboratories to collect the information needed to make representing that level of granularity a reality.

thus far we have been able to make progress without going to such extreme levels of provenance tracking, our information systems for managing neuron types and cell types more generally need to be engineered to handle that level of detail should it prove necessary.

For the AtOM ontology, one of the original frustrations that prompted its creation was the fact that the versions of atlases are not routinely reported in methods. This means that it is often impossible to compare data between studies because atlases vary between versions and thus the name associated with, e.g., a set of stereotaxic coordinates can and does change. A very basic element of methodology was not being reported with sufficient granularity so we developed a model capable of illustrating the issue and used it to provide concrete recommendations to improve current practice, not only for users of atlases but for authors of atlases as well.

One area that is mentioned in Chapter Two that ties directly to using methodology to manage information is that of delineation or parcellation criteria. Like the `DeterminedBy` phenotype predicates in NPO, these delineation criteria are the ultimate definition of an anatomical region. Trying to apply coordinates from an atlas created for a particular strain to those of even a closely related strain inevitably fails for some regions due to the intrinsic variability of biological systems, and scientists have to expend effort modifying their protocols to converge on the equivalent region. This is currently accepted as a simple fact of life, but it does not have to be this way. Bottom-up processive criteria for defining anatomical regions, rather than top down assertions drawn as lines on a map, are critical for creating systems that can enable more efficient approaches to anatomy[57]. Similarly, not all criteria can be applied to the same sample[58]. It is vital that the variety of methodological criteria that have been used to define anatomical regions be accounted for so that we can compare the resulting atlas annotations.

Though not as straight forward to implement as in the case of phenotypes in NPO, we

[57] The domain of human neuroimaging shows the clearest example of how this can be achieved. They had to develop a bottom-up solution because, among other reasons, the intrinsic variability in the anatomy of human cortical regions makes it practically impossible to meaningfully apply a single static set of atlas annotations to more than one individual.

[58] For example, we usually cannot apply histological techniques to living human subjects.

can imagine creating a hierarchy of anatomical regions associated with a single anatomical term based on the general techniques and exact protocols used to define them. Thus, using AtOM as a starting point, we can imagine a future information system for anatomical atlases with regions defined by bottom-up methodological criteria that can be applied consistently across species and individuals.

Finally, `protc/ur` provides a practical implementation for capturing the fine details of experimental methodology so that it can be used in information systems. While in the examples from NPO and AtOM we draw the line at protocols, `protc/ur` shows that our ability to account for methodological variability does end at the level of a whole protocol, and can be extended down to the particulars of the individual participants, aspects, and constraints for a given protocol. In the vision of a fully instrumented laboratory, we can imagine going even further, to the performances of individual protocols.

Thus, although we are as yet only at the beginning of the path, we can see a way forward to create practical systems that can account for and track methodological variability throughout the scientific process. In this way, if we cannot banish the specter of ”human error” from science forever, we will come at least in the end to some place where it will be possible to identify sources of systematic error on the scale of individual labs and make it practical for them to take action to reduce it.

Bibliography

[1] Marc T. Avey, David Moher, Katrina J. Sullivan, Dean Fergusson, Gilly Griffin, Jeremy M. Grimshaw, Brian Hutton, Manoj M. Lalu, Malcolm Macleod, John Marshall, Shirley H. J. Mei, Michael Rudnicki, Duncan J. Stewart, Alexis F. Turgeon, Lauralyn McIntyre, and Canadian Critical Care Translational Biology Group. “The Devil Is in the Details: Incomplete Reporting in Preclinical Animal Research”. In: *PLOS ONE* 11.11 (2016), e0166733. DOI: 10.1371/journal.pone.0166733. URL: https://doi.org/10.1371/journal.pone.0166733.

[2] Zeljana Babic, Amanda Capes-Davis, Maryann E Martone, Amos Bairoch, I Burak Ozyurt, Thomas H Gillespie, and Anita E Bandrowski. “Incidences of problematic cell lines are lower in papers that use RRIDs to identify cell lines”. In: *eLife* 8 (2019). DOI: 10.7554/elife.41676. URL: https://doi.org/10.7554/eLife.41676.

[3] Monya Baker. “1,500 Scientists Lift the Lid on Reproducibility”. In: *Nature* 533.7604 (2016), pp. 452–454. DOI: 10.1038/533452a. URL: https://doi.org/10.1038/533452a.

[4] Anita Bandrowski, Ryan Brinkman, Mathias Brochhausen, Matthew H. Brush, Bill Bug, Marcus C. Chibucos, Kevin Clancy, Mélanie Courtot, Dirk Derom, Michel Dumontier, Liju Fan, Jennifer Fostel, Gilberto Fragoso, Frank Gibson, Alejandra Gonzalez-Beltran, Melissa A. Haendel, Yongqun He, Mervi Heiskanen, Tina Hernandez-Boussard, Mark Jensen, Yu Lin, Allyson L. Lister, Phillip Lord, James Malone, Elisabetta Manduchi, Monnie McGee, Norman Morrison, James A. Overton, Helen Parkinson, Bjoern Peters, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H. Scheuermann, Daniel Schober, Barry Smith, Larisa N. Soldatova, Christian J. Stoeckert, Chris F. Taylor, Carlo Torniai, Jessica A. Turner, Randi Vita, Patricia L. Whetzel, and Jie Zheng. “The Ontology for Biomedical Investigations”. In: *PLOS ONE* 11.4 (2016), e0154556. DOI: 10.1371/journal.pone.0154556. URL: https://doi.org/10.1371/journal.pone.0154556.

[5] Anita Bandrowski, Matthew Brush, Jeffery S. Grethe, Melissa A. Haendel, David N. Kennedy, Sean Hill, Patrick R. Hof, Maryann E. Martone, Maaike Pols, Serena C. Tan, Nicole Washington, Elena Zudilova-Seinstra, and Nicole Vasilevsky. “The Resource Identification Initiative: a Cultural Shift in Publishing”. In: *Brain and Behavior* 6.1 (2015). DOI: 10.1002/brb3.417. URL: https://doi.org/10.1002/brb3.417.

[6] Jon Bentley. “Programming Pearls: Little Languages”. In: *Communications of the ACM* 29.8 (1986), pp. 711–721. DOI: 10.1145/6424.315691. URL: https://doi.org/10.1145/6424.315691.

[7] Mehul Bhattacharyya, Valerie M Miller, Debjani Bhattacharyya, and Larry E Miller. “High Rates of Fabricated and Inaccurate References in Chatgpt-Generated Medical Content”. In: *Cureus* (2023). DOI: 10.7759/cureus.39238. URL: https://doi.org/10.7759/cureus.39238.

[8] Mark G. Blanton, Joseph J. Lo Turco, and Arnold R. Kriegstein. “Whole Cell Recording From Neurons in Slices of Reptilian and Mammalian Cerebral Cortex”. In: *Journal of Neuroscience Methods* 30.3 (1989), pp. 203–210. DOI: 10.1016/0165-0270(89)90131-3. URL: https://doi.org/10.1016/0165-0270(89)90131-3.

[9] Conrad Bock and Michael Gruninger. “PSL: A semantic domain for flow models”. In: *Software & Systems Modeling* 4.2 (2005), pp. 209–231. DOI: 10.1007/s10270-004-0066-x. URL: https://doi.org/10.1007/s10270-004-0066-x.

[10] Kyle Boyar, Andrew Pham, Shannon Swantek, Gary Ward, and Gary Herman. “Laboratory Information Management Systems (LIMS)”. In: *Cannabis Laboratory Fundamentals*. Cannabis Laboratory Fundamentals. Springer International Publishing, 2021, pp. 131–151. DOI: 10.1007/978-3-030-62716-4_7. URL: https://doi.org/10.1007/978-3-030-62716-4_7.

[11] Tim Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 7159. Mar. 2014. DOI: 10.17487/RFC7159. URL: https://www.rfc-editor.org/info/rfc7159.

[12] Ryan R Brinkman, the OBI consortium, Mélanie Courtot, Dirk Derom, Jennifer M Fostel, Yongqun He, Phillip Lord, James Malone, Helen Parkinson, Bjoern Peters, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Larisa N Soldatova, Christian J Stoeckert, Jessica A Turner, and Jie Zheng. “Modeling Biomedical Experimental Processes With Obi”. In: *Journal of Biomedical Semantics* 1.S1 (2010), S7. DOI: 10.1186/2041-1480-1-s1-s7. URL: https://doi.org/10.1186/2041-1480-1-S1-S7.

[13] Rhitu Chatterjee and Lisa Mullins. *New Clues Emerge in Centuries-Old Swedish Shipwreck*. URL: https://theworld.org/stories/2012-02-23/new-clues-emerge-centuries-old-swedish-shipwreck.

[14] Kenneth Church and Ramesh Patil. “Coping with syntactic ambiguity or how to put the block in the box on the table”. In: (1982).

[15] Open Science Collaboration. “Estimating the Reproducibility of Psychological Science”. In: *Science* 349.6251 (2015), aac4716–aac4716. DOI: 10.1126/science.aac4716. URL: https://doi.org/10.1126/science.aac4716.

[16] Douglas Crockford. *The application/json Media Type for JavaScript Object Notation (JSON)*. RFC 4627. July 2006. DOI: 10.17487/RFC4627. URL: https://www.rfc-editor.org/info/rfc4627.

[17] Ryan Culpepper and Matthias Felleisen. "Fortifying Macros". In: *ACM SIGPLAN Notices* 45.9 (2010), pp. 235–246. DOI: 10.1145/1932681.1863577. URL: https://doi.org/10.1145/1932681.1863577.

[18] John Dewey and Arthur Fisher Bentley. *Knowing and the known*. Boston: Beacon Press, 1949. URL: https://www.aier.org/sites/default/files/Files/WYSIWYG/page/31/KnowingKnownFullText.pdf.

[19] Shawn Douglas and Bret Victor. *Shawn Douglas - Nanoscale Instruments for Visualizing Small Proteins & Bret Victor - Dynamicland*. YouTube, 2022. URL: https://www.youtube.com/watch?v=_gXiVOmaVSo&t=865s.

[20] F. A. Edwards, A. Konnerth, B. Sakmann, and T. Takahashi. "A Thin Slice Preparation for Patch Clamp Recordings From Neurones of the Mammalian Central Nervous System". In: *Pflügers Archiv European Journal of Physiology* 414.5 (1989), pp. 600–612. DOI: 10.1007/bf00580998. URL: https://doi.org/10.1007/BF00580998.

[21] Kristin Evans, ed. *High-fat diet feeding*. 2015. URL: https://www.mmpc.org/shared/document.aspx?id=266&docType=Protocol.

[22] Allan Feldman, Kent A. Divoll, and Allyson Rogan-Klyve. "Becoming Researchers: the Participation of Undergraduate and Graduate Students in Scientific Research Groups". In: *Science Education* 97.2 (2013), pp. 218–243. DOI: 10.1002/sce.21051. URL: https://doi.org/10.1002/sce.21051.

[23] Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi, Eli Barzilay, Jay McCarthy, and Sam Tobin-Hochstadt. "A Programmable Programming Language". In: *Communications of the ACM* 61.3 (2018), pp. 62–71. DOI: 10.1145/3127323. URL: https://doi.org/10.1145/3127323.

[24] Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi, Eli Barzilay, Jay McCarthy, and Sam Tobin-Hochstadt. "The Racket Manifesto". In: *Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany*. Leibniz International Proceedings in Informatics (LIPIcs) 32 (2015). Ed. by Thomas Ball, Rastislav Bodik, Shriram Krishnamurthi, Benjamin S. Lerner, and Greg Morrisett, pp. 113–128. ISSN: 1868-8969. DOI: 10.4230/LIPICS.SNAPL.2015.113. URL: http://drops.dagstuhl.de/opus/volltexte/2015/5021/.

[25] Daniel Feltey, Spencer P Florence, Tim Knutson, Vincent St-Amour, Ryan Culpepper, Matthew Flatt, Robert Bruce Findler, and Matthias Felleisen. *Languages the Racket way*. URL: https://summer-school.racket-lang.org/2017/notes/lwc-languages-the-racket-way.pdf.

[26] Robert Bruce Findler, John Clements, Cormac Flanagan, Matthew Flatt, Shriram Krishnamurthi, Paul Steckler, and Matthias Felleisen. “Drscheme: a Programming Environment for Scheme”. In: *Journal of Functional Programming* 12.02 (2002). DOI: 10.1017/s0956796801004208. URL: https://doi.org/10.1017/S0956796801004208.

[27] Matthew Flatt. “Creating Languages in Racket”. In: *Communications of the ACM* 55.1 (2012), pp. 48–56. DOI: 10.1145/2063176.2063195. URL: https://doi.org/10.1145/2063176.2063195.

[28] Leonard P. Freedman, Gautham Venugopalan, and Rosann Wisman. “Reproducibility2020: Progress and Priorities”. In: *F1000Research* 6.nil (2017), p. 604. DOI: 10.12688/f1000research.11334.1. URL: https://doi.org/10.12688/f1000research.11334.1.

[29] Thomas H. Gillespie, Shreejoy J. Tripathy, Mohameth François Sy, Maryann E. Martone, and Sean L. Hill. “The Neuron Phenotype Ontology: A Fair Approach To Proposing and Classifying Neuronal Types”. In: *Neuroinformatics* 20.3 (2022), pp. 793–809. DOI: 10.1007/s12021-022-09566-7. URL: https://doi.org/10.1007/s12021-022-09566-7.

[30] Tom Gillespie, Bernard De Bono, Monique Surles-Zeigler, Natallia Kokash, Fahim Imam, Susan Tappan, Jyl Boline, Jeffrey Grethe, and Maryann Martone. *SPARC Connectivity Knowledge Base of the Autonomic Nervous System*. 2022. DOI: 10.5281/ZENODO.5337441. URL: https://zenodo.org/record/5337441.

[31] Olga Giraldo, Alexander Garcia, and Oscar Corcho. “A Guideline for Reporting Experimental Protocols in Life Sciences”. In: *PeerJ* 6 (2018), e4795. DOI: 10.7717/peerj.4795. URL: https://doi.org/10.7717/peerj.4795.

[32] Olga Giraldo, Alexander García, and Oscar Corcho. “SMART protocols: semantic representation for experimental protocols”. In: *Proceedings of the 4th International Conference on Linked Science-Volume 1282*. CEUR-WS. org. 2014, pp. 36–47.

[33] Olga Giraldo, Alexander García, Federico López, and Oscar Corcho. “Using Semantics for Representing Experimental Protocols”. In: *Journal of Biomedical Semantics* 8.1 (2017), p. 52. DOI: 10.1186/s13326-017-0160-y. URL: https://doi.org/10.1186/s13326-017-0160-y.

[34] Paul Glasziou, Douglas G Altman, Patrick Bossuyt, Isabelle Boutron, Mike Clarke, Steven Julious, Susan Michie, David Moher, and Elizabeth Wager. “Reducing Waste From Incomplete Or Unusable Reports of Biomedical Research”. In: *The Lancet* 383.9913 (2014), pp. 267–276. DOI: 10.1016/s0140-6736(13)62228-x. URL: https://doi.org/10.1016/S0140-6736(13)62228-X.

[35] Neil Goldman, Robert Balzer, and David Wile. “The Inference of Domain Structure From Informal Process Descriptions”. In: *ACM SIGART Bulletin* nil.63 (1977), pp. 75–

76. DOI: 10.1145/1045343.1045388. URL: https://doi.org/10.1145/1045343.1045388.

[36] Jocelyn Gravel, Madeleine D'Amours-Gravel, and Esli Osmanlliu. "Learning To Fake It: Limited Responses and Fabricated References Provided By Chatgpt for Medical Questions". In: *Mayo Clinic Proceedings: Digital Health* 1.3 (2023), pp. 226–234. DOI: 10.1016/j.mcpdig.2023.05.004. URL: https://doi.org/10.1016/j.mcpdig.2023.05.004.

[37] Arcadi Grigorian, Paul Fang, Tate Kirk, Aslan Efendizade, Jami Jadidi, Maziar Sighary, and Dan I. Cohen-Addad. "Learning From Gamers: Integrating Alternative Input Devices and Autohotkey Scripts To Simplify Repetitive Tasks and Improve Workflow". In: *RadioGraphics* 40.1 (2020), pp. 141–150. DOI: 10.1148/rg.2020190077. URL: https://doi.org/10.1148/rg.2020190077.

[38] Michael Gruninger and Christopher Menzel. "The process specification language (PSL) theory and applications". In: *AI magazine* 24.3 (2003), p. 63.

[39] Nicola Guarino. "Formal Ontology, Conceptual Analysis and Knowledge Representation". In: *International Journal of Human-Computer Studies* 43.5-6 (1995), pp. 625–640. DOI: 10.1006/ijhc.1995.1066. URL: https://doi.org/10.1006/ijhc.1995.1066.

[40] Laurel L. Haak. personal communication. 2016.

[41] Laurel L. Haak, Martin Fenner, Laura Paglione, Ed Pentz, and Howard Ratner. "Orcid: a System To Uniquely Identify Researchers". In: *Learned Publishing* 25.4 (2012), pp. 259–264. DOI: 10.1087/20120404. URL: http://dx.doi.org/10.1087/20120404.

[42] Steven Harris and Andy Seaborne. *SPARQL 1.1 Query Language*. W3C Recommendation. W3C, Mar. 2013. URL: https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[43] Karl Helmer, David Keator, Tibor Auer, Satrajit Ghosh, Camille Maumet, Thomas Nichols, and Jean-Baptiste Poline. "Constructing an Ontology of Neuroscience Experiments for the Neuroimaging Data Model (NIDM) Authors: Introduction". In: *OHBM 2019-25th Annual Meeting of the Organization for Human Brain Mapping, Jun 2019*. 2019, pp. 1–4. URL: https://www.hal.inserm.fr/inserm-02379281.

[44] Pascal Hitzler, Sebastian Rudolph, Markus Krötzsch, Bijan Parsia, and Peter Patel-Schneider. *OWL 2 Web Ontology Language Primer (Second Edition)*. W3C Recommendation. W3C, Dec. 2012. URL: https://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

[45] Frederick M Hocker. *Vasa: a Swedish warship*. Medstroms, 2011.

[46] John P. A. Ioannidis. “Why Most Published Research Findings Are False”. In: *PLoS Medicine* 2.8 (2005), e124. DOI: 10.1371/journal.pmed.0020124. URL: https://doi.org/10.1371/journal.pmed.0020124.

[47] Stephen C Johnson et al. *Yacc: Yet another compiler-compiler*. Vol. 32. Bell Laboratories Murray Hill, NJ, 1975.

[48] Daniel Kahneman. “A new etiquette for replication.” In: *Social Psychology* 45.4 (2014), p. 310.

[49] Nir Kalisman, Gilad Silberberg, and Henry Markram. “The Neocortical Microcircuit As a *tabula Rasa*”. In: *Proceedings of the National Academy of Sciences* 102.3 (2005), pp. 880–885. DOI: 10.1073/pnas.0407088102. URL: https://doi.org/10.1073/pnas.0407088102.

[50] Samantha Kanza, Nicholas Gibbins, and Jeremy G. Frey. “Too Many Tags Spoil the Metadata: Investigating the Knowledge Management of Scientific Research With Semantic Web Technologies”. In: *Journal of Cheminformatics* 11.1 (2019). DOI: 10.1186/s13321-019-0345-8. URL: https://doi.org/10.1186/s13321-019-0345-8.

[51] Samantha Kanza, Cerys Willoughby, Nicholas Gibbins, Richard Whitby, Jeremy Graham Frey, Jana Erjavec, Klemen Zupančič, Matjaž Hren, and Katarina Kovač. “Electronic Lab Notebooks: Can They Replace Paper?” In: *Journal of Cheminformatics* 9.1 (2017), p. 31. DOI: 10.1186/s13321-017-0221-3. URL: https://doi.org/10.1186/s13321-017-0221-3.

[52] Gregg Kellogg, Dave Longley, and Pierre-Antoine Champin. *JSON-LD 1.1*. W3C Recommendation. W3C, July 2020. URL: https://www.w3.org/TR/2020/REC-json-ld11-20200716/.

[53] Brian W Kernighan. “Advanced editing on UNIX”. In: *UNIX Programmer’s Manual* 2 (1978).

[54] Michel Kinsy, Zoé Lacroix, Christophe Legendre, Piotr Wlodarczyk, and Nadia Yacoubi. “ProtocolDB: Storing Scientific Protocols with a Domain Ontology”. In: *Web Information Systems Engineering - WISE 2007 Workshops*. Web Information Systems Engineering - WISE 2007 Workshops. Springer Berlin Heidelberg, 2007, pp. 17–28. DOI: 10.1007/978-3-540-77010-7_3. URL: https://doi.org/10.1007/978-3-540-77010-7_3.

[55] Heidi Kleven, Thomas H. Gillespie, Lyuba Zehl, Timo Dickscheid, Jan G. Bjaalie, Maryann E. Martone, and Trygve B. Leergaard. “Atom, an Ontology Model To Standardize Use of Brain Atlases in Tools, Workflows, and Data Infrastructures”. In: *Scientific Data* 10.1 (2023), p. 486. DOI: 10.1038/s41597-023-02389-4. URL: https://doi.org/10.1038/s41597-023-02389-4.

[56] Nicola J. Knight, Samantha Kanza, Don Cruickshank, William S. Brocklesby, and Jeremy G. Frey. "Talk2lab: the Smart Lab of the Future". In: *IEEE Internet of Things Journal* 7.9 (2020), pp. 8631–8640. DOI: 10.1109/jiot.2020.2995323. URL: https://doi.org/10.1109/JIOT.2020.2995323.

[57] Natalia Kwasnikowska, Yi Chen, and Zoé Lacroix. "Modeling and storing scientific protocols". In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer. 2006, pp. 730–739. DOI: 10.1007/11915034_97. URL: https://doi.org/10.1007/11915034_97.

[58] Daniël Lakens. "Is My Study Useless? Why Researchers Need Methodological Review Boards". In: *Nature* 613.7942 (2023), pp. 9–9. DOI: 10.1038/d41586-022-04504-8. URL: https://doi.org/10.1038/d41586-022-04504-8.

[59] Bruno Latour. "Scientific objects and legal objectivity". In: *Law, Anthropology, and the Constitution of the Social*. Law, Anthropology, and the Constitution of the Social. Cambridge University Press, 2004, pp. 73–114. DOI: 10.1017/cbo9780511493751.003. URL: https://doi.org/10.1017/CBO9780511493751.003.

[60] Bruno Latour and Steve Woolgar. *Laboratory life: The construction of scientific facts*. Princeton university press, 1979.

[61] Young Han Lee. "Efficient Radiologic Reading Environment By Using an Open-Source Macro Program As Connection Software". In: *European Journal of Radiology* 81.1 (2012), pp. 100–103. DOI: 10.1016/j.ejrad.2010.11.019. URL: https://doi.org/10.1016/j.ejrad.2010.11.019.

[62] Michael E Lesk and Eric Schmidt. *Lex: A lexical analyzer generator*. 1975.

[63] Thomas Levenson. "The truth about Isaac Newton's productive plague". In: *New Yorker* 6 (2020). URL: https://www.newyorker.com/culture/cultural-comment/the-truth-about-isaac-newtons-productive-plague.

[64] Andrew MacEwan, Anila Angjeli, and Janifer Gatenby. "The International Standard Name Identifier (ISNI): the Evolving Future of Name Authority Control". In: *Cataloging & Classification Quarterly* 51.1-3 (2013), pp. 55–71. DOI: 10.1080/01639374.2012.730601. URL: http://dx.doi.org/10.1080/01639374.2012.730601.

[65] James Malone, Ele Holloway, Tomasz Adamusiak, Misha Kapushesky, Jie Zheng, Nikolay Kolesnikov, Anna Zhukova, Alvis Brazma, and Helen Parkinson. "Modeling sample variables with an Experimental Factor Ontology". In: *Bioinformatics* 26.8 (2010), pp. 1112–1118. DOI: 10.1093/bioinformatics/btq099. URL: https://doi.org/10.1093/bioinformatics/btq099.

[66] Emilie Marcus. "A Star Is Born". In: *Cell* 166.5 (2016), pp. 1059–1060. DOI: 10.1016/j.cell.2016.08.021. URL: https://doi.org/10.1016/j.cell.2016.08.021.

[67] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W. Reimann, Marwan Abdellah, Carlos Aguado Sanchez, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, Guy Antoine Atenekeng Kahou, Thomas K. Berger, Ahmet Bilgili, Nenad Buncic, Athanassia Chalimourda, Giuseppe Chindemi, Jean-Denis Courcol, Fabien Delalondre, Vincent Delattre, Shaul Druckmann, Raphael Dumusc, James Dynes, Stefan Eilemann, Eyal Gal, Michael Emiel Gevaert, Jean-Pierre Ghobril, Albert Gidon, Joe W. Graham, Anirudh Gupta, Valentin Haenel, Etay Hay, Thomas Heinis, Juan B. Hernando, Michael Hines, Lida Kanari, Daniel Keller, John Kenyon, Georges Khazen, Yihwa Kim, James G. King, Zoltan Kisvarday, Pramod Kumbhar, Sébastien Lasserre, Jean-Vincent Le Bé, Bruno R.C. Magalhães, Angel Merchán-Pérez, Julie Meystre, Benjamin Roy Morrice, Jeffrey Muller, Alberto Muñoz-Céspedes, Shruti Muralidhar, Keerthan Muthurasa, Daniel Nachbaur, Taylor H. Newton, Max Nolte, Aleksandr Ovcharenko, Juan Palacios, Luis Pastor, Rodrigo Perin, Rajnish Ranjan, Imad Riachi, José-Rodrigo Rodríguez, Juan Luis Riquelme, Christian Rössert, Konstantinos Sfyrakis, Ying Shi, Julian C. Shillcock, Gilad Silberberg, Ricardo Silva, Farhan Tauheed, Martin Telefont, Maria Toledo-Rodriguez, Thomas Tränkler, Werner Van Geit, Jafet Villafranca Díaz, Richard Walker, Yun Wang, Stefano M. Zaninetta, Javier DeFelipe, Sean L. Hill, Idan Segev, and Felix Schürmann. "Reconstruction and Simulation of Neocortical Microcircuitry". In: *Cell* 163.2 (2015), pp. 456–492. DOI: 10.1016/j.cell.2015.09.029. URL: https://doi.org/10.1016/j.cell.2015.09.029.

[68] Catherine C. Marshall and Frank M. Shipman. "Which semantic web?" In: *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*. Aug. 2003, nil. DOI: 10.1145/900051.900063. URL: https://doi.org/10.1145/900051.900063.

[69] Wallace Marshall. *ANNUAL REPORT Center for Cellular Construction*. 2018. URL: https://ccc.ucsf.edu/sites/ccc.ucsf.edu/files/Marshall_W_CCC_Annual_Report_Extracts_fy02_2018.pdf.

[70] Gagan Mathur, Thomas H. Haugen, Scott L. Davis, and Matthew D. Krasowski. "Streamlined Sign-Out of Capillary Protein Electrophoresis Using Middleware and an Open-Source Macro Application". In: *Journal of Pathology Informatics* 5.1 (2014), p. 36. DOI: 10.4103/2153-3539.141990. URL: https://doi.org/10.4103/2153-3539.141990.

[71] Camille Maumet, Tibor Auer, Alexander Bowring, Gang Chen, Samir Das, Guillaume Flandin, Satrajit Ghosh, Tristan Glatard, Krzysztof J. Gorgolewski, Karl G. Helmer, Mark Jenkinson, David B. Keator, B. Nolan Nichols, Jean-Baptiste Poline, Richard Reynolds, Vanessa Sochat, Jessica Turner, and Thomas E. Nichols. "Sharing Brain Mapping Statistical Results With the Neuroimaging Data Model". In: *Scientific Data* 3.1 (2016), p. 160102. DOI: 10.1038/sdata.2016.102. URL: https://doi.org/10.1038/sdata.2016.102.

[72] James Clerk Maxwell. “ON FORCE”. In: *Matter and Motion*. Matter and Motion. Cambridge University Press, 1876. Chap. 3, pp. 33–49. DOI: 10.1017/cbo9780511709326.004. URL: https://doi.org/10.1017/CBO9780511709326.004.

[73] John McCarthy. “Recursive functions of symbolic expressions and their computation by machine, Part I”. In: *Communications of the ACM* 3.4 (1960), pp. 184–195. DOI: 10.1145/367177.367199. URL: https://doi.org/10.1145/367177.367199.

[74] Deborah McGuinness, Satya Sahoo, and Timothy Lebo. *PROV-O: The PROV Ontology*. W3C Recommendation. W3C, Apr. 2013. URL: https://www.w3.org/TR/2013/REC-prov-o-20130430/.

[75] Joe Menke, Martijn Roelandse, Burak Ozyurt, Maryann Martone, and Anita Bandrowski. “The Rigor and Transparency Index Quality Metric for Assessing Biological and Medical Science Methods”. In: *iScience* 23.11 (2020), p. 101698. DOI: 10.1016/j.isci.2020.101698. URL: https://doi.org/10.1016/j.isci.2020.101698.

[76] Marjan Mernik, Jan Heering, and Anthony M Sloane. “When and how to develop domain-specific languages”. In: *ACM computing surveys (CSUR)* 37.4 (2005), pp. 316–344. DOI: 10.1145/1118890.1118892. URL: https://doi.org/10.1145/1118890.1118892.

[77] Mahyar Osanlouy, Anita Bandrowski, Bernard de Bono, David Brooks, Antonino M. Cassarà, Richard Christie, Nazanin Ebrahimi, Tom Gillespie, Jeffrey S. Grethe, Leonardo A. Guercio, Maci Heal, Mabelle Lin, Niels Kuster, Maryann E. Martone, Esra Neufeld, David P. Nickerson, Elias G. Soltani, Susan Tappan, Joost B. Wagenaar, Katie Zhuang, and Peter J. Hunter. “The SPARC DRC: Building a Resource for the Autonomic Nervous System Community”. In: *Frontiers in Physiology* 12 (2021). DOI: 10.3389/fphys.2021.693735. URL: https://doi.org/10.3389/fphys.2021.693735.

[78] John H. Powers, Jie Min, and David Tribble. “Strengthen Scientific Review of Research Protocols”. In: *Nature* 617.7959 (2023), pp. 35–35. DOI: 10.1038/d41586-023-01480-5. URL: https://doi.org/10.1038/d41586-023-01480-5.

[79] Eric M. Prager, Karen E. Chambers, Joshua L. Plotkin, David L. McArthur, Anita E. Bandrowski, Nidhi Bansal, Maryann E. Martone, Hadley C. Bergstrom, Anton Bespalov, and Chris Graf. “Improving Transparency and Scientific Rigor in Academic Publishing”. In: *Brain and Behavior* 9.1 (2018), e01141. DOI: 10.1002/brb3.1141. URL: https://doi.org/10.1002/brb3.1141.

[80] Poonam J. Prasad and G.L. Bodhe. “Trends in Laboratory Information Management System”. In: *Chemometrics and Intelligent Laboratory Systems* 118.nil (2012), pp. 187–192. DOI: 10.1016/j.chemolab.2012.07.001. URL: https://doi.org/10.1016/j.chemolab.2012.07.001.

[81] Eric Prud’hommeaux and Gavin Carothers. *RDF 1.1 Turtle*. W3C Recommendation. W3C, Feb. 2014. URL: https://www.w3.org/TR/2014/REC-turtle-20140225/.

[82] P. Resnik. “Semantic Similarity in a Taxonomy: an Information-Based Measure and Its Application To Problems of Ambiguity in Natural Language”. In: *Journal of Artificial Intelligence Research* 11.nil (1999), pp. 95–130. DOI: 10.1613/jair.514. URL: https://doi.org/10.1613/jair.514.

[83] Grzegorz Rozenberg and Arto Salomaa, eds. *Handbook of Formal Languages*. Berlin and Heidelberg: Springer-Verlag, 1997.

[84] Thomas A Russ, Cartic Ramakrishnan, Eduard H Hovy, Mihail Bota, and Gully APC Burns. “Knowledge Engineering Tools for Reasoning With Scientific Observations and Interpretations: a Neural Connectivity Use Case”. In: *BMC Bioinformatics* 12.1 (2011), p. 1. DOI: 10.1186/1471-2105-12-351. URL: https://doi.org/10.1186/1471-2105-12-351.

[85] Craig Schlenoff, Michael Gruninger, Florence Tissot, John Valois, Joshua Lubell, and Jintae Lee. *The process specification language (psl): Overview and version 1.0 specification*. 2000. DOI: 10.6028/nist.ir.6459. URL: https://doi.org/10.6028/nist.ir.6459.

[86] Craig Schlenoff, Amy Knutilla, and Steven Ray. “Unified process specification language: Requirements for modeling process”. In: *Interagency Report* 5910 (1996). DOI: 10.6028/nist.ir.5910. URL: https://doi.org/10.6028/nist.ir.5910.

[87] Eric Schulte and Dan Davison. “Active Documents With Org-Mode”. In: *Computing in Science & Engineering* 13.3 (2011), pp. 66–73. DOI: 10.1109/mcse.2011.41. URL: https://doi.org/10.1109/MCSE.2011.41.

[88] Andy Seaborne and Eric Prud’hommeaux. *SPARQL Query Language for RDF*. W3C Recommendation. W3C, Jan. 2008. URL: https://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

[89] Nathalie Percie du Sert, Viki Hurst, Amrita Ahluwalia, Sabina Alam, Marc T. Avey, Monya Baker, William J. Browne, Alejandra Clark, Innes C. Cuthill, Ulrich Dirnagl, Michael Emerson, Paul Garner, Stephen T. Holgate, David W. Howells, Natasha A. Karp, Stanley E. Lazic, Katie Lidster, Catriona J. MacCallum, Malcolm Macleod, Esther J. Pearl, Ole H. Petersen, Frances Rawle, Penny Reynolds, Kieron Rooney, Emily S. Sena, Shai D. Silberberg, Thomas Steckler, and Hanno Würbel. “The Arrive Guidelines 2.0: Updated Guidelines for Reporting Animal Research*”. In: *Journal of Cerebral Blood Flow & Metabolism* 40.9 (2020), pp. 1769–1777. DOI: 10.1177/0271678x20943823. URL: https://doi.org/10.1177/0271678X20943823.

[90] Frank M. Shipman and Catherine C. Marshall. “Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems”. In: *Computer Supported Cooperative Work (CSCW)* 8.4 (1999),

pp. 333–352. DOI: 10.1023/a:1008716330212. URL: https://doi.org/10.1023/a:1008716330212.

[91] Diomidis Spinellis. “Notable Design Patterns for Domain-Specific Languages”. In: *Journal of Systems and Software* 56.1 (2001), pp. 91–99. DOI: 10.1016/s0164-1212(00)00089-3. URL: https://doi.org/10.1016/S0164-1212(00)00089-3.

[92] Luka Stanisic, Arnaud Legrand, and Vincent Danjean. “An Effective Git and Org-Mode Based Workflow for Reproducible Research”. In: *ACM SIGOPS Operating Systems Review* 49.1 (2015), pp. 61–70. DOI: 10.1145/2723872.2723881. URL: https://doi.org/10.1145/2723872.2723881.

[93] Guy L. Steele. *Common LISP: the language*. Second. Elsevier, 1990, pp. 0–1029. ISBN: 1555580416.

[94] Marcelo Tallis, Richard Thompson, Thomas A. Russ, and Gully A. P. C. Burns. “Knowledge Synthesis With Maps of Neural Connectivity”. In: *Frontiers in Neuroinformatics* 5.nil (2011), nil. DOI: 10.3389/fninf.2011.00024. URL: https://doi.org/10.3389/fninf.2011.00024.

[95] Sharon Tentarelli, Rómulo Romero, and Michelle L. Lamb. “Script-Based Automation of Analytical Instrument Software Tasks”. In: *SLAS Technology* 27.3 (2022), pp. 209–213. DOI: 10.1016/j.slast.2021.10.019. URL: https://doi.org/10.1016/j.slast.2021.10.019.

[96] Leonid Teytelman, Alexei Stoliartchouk, Lori Kindler, and Bonnie L. Hurwitz. “Protocols.io: Virtual Communities for Protocol Development and Discussion”. In: *PLOS Biology* 14.8 (2016), e1002538. DOI: 10.1371/journal.pbio.1002538. URL: https://doi.org/10.1371/journal.pbio.1002538.

[97] Kerstin Thurow, Bernd Göde, Uwe Dingerdissen, and Norbert Stoll. “Laboratory Information Management Systems for Life Science Applications”. In: *Organic Process Research & Development* 8.6 (2004), pp. 970–982. DOI: 10.1021/op040017s. URL: https://doi.org/10.1021/op040017s.

[98] Sam Tobin-Hochstadt, Vincent St-Amour, Ryan Culpepper, Matthew Flatt, and Matthias Felleisen. “Languages as libraries”. In: *ACM SIGPLAN Notices*. Vol. 46. 6. ACM. 2011, pp. 132–141. DOI: 10.1145/1993316.1993514. URL: https://doi.org/10.1145/1993316.1993514.

[99] Markus Voelter, Sebastian Benz, Christian Dietrich, Birgit Engelmann, Mats Helander, Lennart CL Kats, Eelco Visser, and Guido Wachsmuth. *DSL engineering: Designing, implementing and using domain-specific languages*. dslbook. org, 2013.

[100] Andrew Weidner, Robert John Wilson, and Daniel Gelaw Alemneh. *Digital Curation Micro-Applications: Digital Lifecycle Management with AutoHotkey*. 2013. URL: `https://digital.library.unt.edu/ark:/67531/metadc159530/`.

[101] Tanya Widen. “Formal language design in the context of domain engineering”. MA thesis. Oregon Graduate Institute of Science and Technology, 1998.

[102] David Wood, Markus Lanthaler, and Richard Cyganiak. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C, Feb. 2014. URL: `https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/`.