Expressing Biological Problems with Logical Reasoning Languages

Tommaso Alfonsi[1], Luigi Bellomarini[2], Anna Bernasconi[1] and Stefano Ceri[1]

[1]Dip. di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133, Milano, Italy
[2]Banca d'Italia, 00184, Roma, Italy

Abstract

Biology represents a very challenging domain that is typically tackled by experts in the field, with few or no interactions with the Web knowledge and rules interoperation community. However, there has been a considerable growth of data regarding biological aspects in the last decades. Moreover, the COVID-19 pandemic has traced an unprecedented point in history, where tons of information have been collected in laboratories worldwide and deposited into open data banks. Inspired by the current needs and backed by a solid knowledge base (our *extensional knowledge* source) called CoV2K, we propose to express and resolve a series of problems related to the SARS-CoV-2 virus and its interpretation. We formulate our queries as rules in Vadalog (our *knowledge representation and reasoning language*) and input them to its related *logic-based reasoning system*. Four cases are presented that allow to explore 1) variants effects and how they are explained in scientific literature; 2) the most typical mutations of a variant; 3) the most likely acquisition of a new mutation by a given variant and the associated reported effects; 4) the most relevant mutations of the virus according to the community. Expressing biological problems using a logic formalism is a major challenge, due to the intrinsic complexity of the domain. The four use cases show that a logical formalism is effective in expressing relevant problems for understanding the current evolution of SARS-CoV-2 variants, an essential aspect of the COVID-19 pandemic.

Keywords

Logical reasoning, Vadalog, Biology, Virology, SARS-CoV-2, COVID-19

1. Introduction

Cancer treatment, personalized medicine, and vaccine development are only few of the application fields where computer science and artificial intelligence are being widely adopted to achieve important discoveries. One apparent challenge is that of expressing complex domain-specific problems – typically conceived by biologists or clinicians – into a clear and actionable formalization, which then enables sophisticated query answering and data analysis.

The use of declarative languages for expressing and resolving biology-related problems has thus far been sporadic. Attempts date back to 1996, when a declarative programming language was used to describe nucleic acids' secondary structures [1]; along a thirty-year range, we can report a web-based declarative workflow language for Life Sciences [2], a layer-oriented

RuleML+RR'22: 16th International Rule Challenge and 6th Doctoral Consortium, September 26–28, 2022, Virtual
tommaso.alfonsi@polimi.it (T. Alfonsi); luigi.bellomarini@bancaditalia.it (L. Bellomarini); anna.bernasconi@polimi.it (A. Bernasconi); stefano.ceri@polimi.it (S. Ceri)
0000-0002-4707-8425 (T. Alfonsi); 0000-0001-6863-0162 (L. Bellomarini); 0000-0001-8016-5750 (A. Bernasconi); 0000-0003-0671-2415 (S. Ceri)

approach for biological modeling [3], and, more recently, Datalog extensions for bioinformatic data analysis [4]. Declarative languages eliminate the need to program complex control flows or to design algorithms. Instead, problems are expressed at a high level, in a readable, modular syntax and reduced code footprint. These aspects are extremely convenient in the life sciences, where high-level problem formalization enables immediate and simpler communication with domain experts.

Contributions are missing that fully exploit the power of declarative languages for supporting the expression of interesting queries over biologically-relevant data, paving the way to their resolution. To demonstrate our claim and attract interest towards this novel challenge, we draw on a particularly current domain, that is, the data and knowledge related to SARS-CoV-2, the virus responsible for the COVID-19 pandemic. In the past two years there has been a wealth of available data and knowledge collected on open data repositories and literature archives; moreover a multitude of open questions of broad research interest have been and still can be formulated on this topic.

As a starting point to this challenge, we chose a basic methodological framework that supports our proposal with i) a *Knowledge Representation and Reasoning* (KRR) language (Vadalog [5]); ii) a source of *extensional knowledge* (CoV2K [6]); and iii) a *logic-based reasoning system* (the Vadalog System [7]). Vadalog is a KRR language of the Datalog$^{\pm}$ family [8]; it supports PTIME ontological reasoning and high expressive power, capturing Datalog, and thus featuring full recursion, while covering all SPARQL queries under the entailment regime of OWL2QL. The extensional knowledege is extracted from CoV2K, a knowledge base that integrates data about viral sequences and established information about the virus (e.g., variants, mutation, their effects or protein structures). The Vadalog System then interprets the logic-based rules and derives new knowledge in the form of new facts, via the reasoning process. The integration of these three components (Vadalog, CoV2K, and the Vadalog System) allows to express and reason on a wide range of problems that could explain the complex mechanisms of the virus that has spread and mutated worldwide in the last 2.5 years. We exemplify the application of the proposed framework through four use cases. The first two introduce the reader to the use of the knowledge base CoV2K through simple queries and aggregations. The third use case explores a graph of frequently co-occurring genomic mutations in viral samples and then draws conclusions on the variants that are prone to acquire novel biologically relevant features. The last example provides two recursion examples while searching for the most relevant mutations of the Spike protein in the scientific literature.

The remainder of this paper is structured as follows. Section 2 provides a background on declarative languages and presents the Vadalog language. Section 3 gives an overview of the CoV2K knowledge model and how this resource is used within Vadalog. Section 4 presents a set of interesting applications concerning the SARS-CoV-2 virus and the possibility to derive knowledge in this domain. Finally, Section 5 discusses the implications of our approach and draws the conclusions.

2. Using Declarative Languages and Their Applications

At the core of our methodological approach, there is a Knowledge Graph Management System (KGMS), a tool that enables reasoning on knowledge graphs, a data model for KRR. A knowledge graph is essentially composed of three parts: (i) a ground extensional component, which includes ground knowledge about a domain of interest; in logical approaches, this component is represented as facts, i.e., ground atoms, defined on domain constants; (ii) an intensional component, that encodes domain knowledge, for instance if a logical approach is adopted, as a set of rules that define domain objects on the basis of others; (iii) a derived extensional knowledge, which is the set of facts resulting from the application of the intensional logic rules to the extensional knowledge, in the reasoning process.

Formalizing complex intensional knowledge poses a number of requirements to the adopted logical language, which must strike a favourable balance between expressive power and computational complexity, so as to capture all the features of real-world scenarios while keeping decidability and tractability.

Specifically, in order to address a broad class of biological problems, we resort to Vadalog, a language for ontological reasoning with support for recursion and existential quantification. Vadalog is a language—technically, a fragment—of the broad Datalog$^{\pm}$ family of database languages [8], whose core revolves around Warded Datalog$^{\pm}$ [9]. Warded captures Datalog (thus incorporating full recursion) and allows the use of existential quantification; at the price of very mild syntactic restrictions, it guarantees PTIME query answering. Moreover, the Piecewise [10] restriction of Warded, applicable to a wide variety of settings, addresses extreme-scale settings, thanks to sub-polynomial complexity of the query answering task. Vadalog extends Warded with features of practical utilities such as monotonic aggregations [11], conditions, and algebraic operations.

Like in plain Datalog, Vadalog rules are first-order dependencies of the form $\forall \overline{x} \forall \overline{y}\,(\phi(\overline{x}, \overline{y}) \rightarrow \exists \overline{z}\ \psi(\overline{z}, \overline{x}))$, where ϕ (the body) and ψ (the head) represent a conjunction of atoms, $\overline{x}, \overline{y}$ and $\overline{z}$ are tuples of variables. The reasoning process incrementally satisfies the rules, if the case introducing new values (namely, *labelled nulls*) for the existentially quantified variables of $\overline{z}$.

The Vadalog language is implemented by a fully engineered system, the Vadalog Engine [7], currently used to compute the derived extensional component in multiple production scenarios, so far in the financial setting.

The well-known advantages of high-level declarative approaches that we have recalled, combined with the high expressive power and scalability of Warded Datalog$^{\pm}$, along with the availability of a state-of-the-art execution runtime motivated our choice for Vadalog as the reference KRR language to encode our knowledge about COVID-19, namely CoV2K [6].

3. A Knowledge Model for SARS-CoV-2 Viral Sequences

In [12, 6] we presented CoV2K, a model of concepts related to the SARS-CoV-2 virus data and knowledge; this model drove the integration process to build a novel source of curated and interlinked instances.
Figure 1 illustrates a portion of the original CoV2K model, including the relevant concepts for

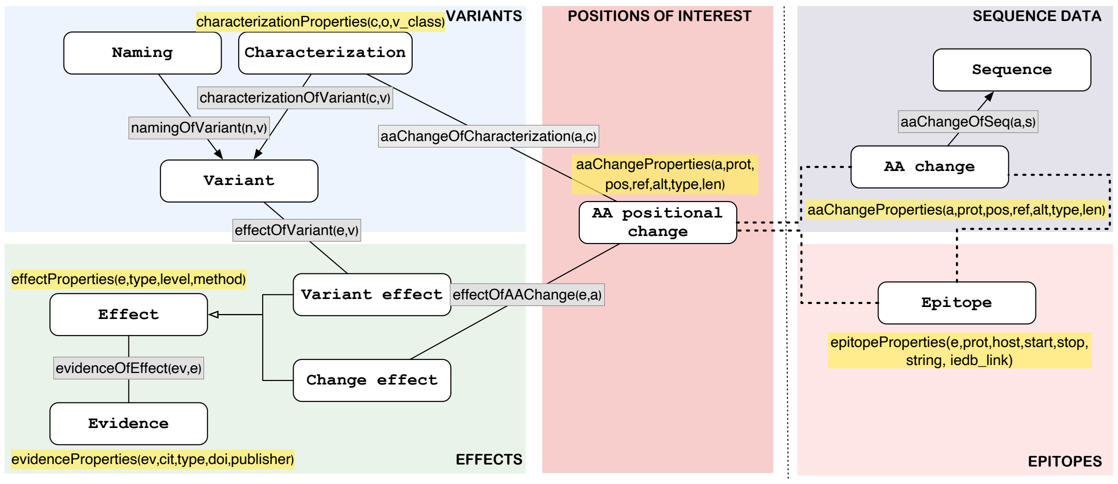
VARIANTS
characterizationProperties(c,o,v_class)
Naming
Characterization
characterizationOfVariant(c,v)
namingOfVariant(n,v)
aaChangeOfCharacterization(a,c)
Variant
effectOfVariant(e,v)
POSITIONS OF INTEREST
aaChangeProperties(a,prot, pos,ref,alt,type,len)
AA positional change
SEQUENCE DATA
Sequence
aaChangeOfSeq(a,s)
AA change
aaChangeProperties(a,prot,pos,ref,alt,type,len)
effectProperties(e,type,level,method)
Effect
Variant effect
effectOfAAChange(e,a)
evidenceOfEffect(ev,e)
Change effect
Evidence
evidenceProperties(ev,cit,type,doi,publisher)
EFFECTS
Epitope
epitopeProperties(e,prot,host,start,stop, string, iedb_link)
EPITOPES

Figure 1: Excerpt of the CoV2K model used in the four presented use cases.

the use cases demonstrated in this paper. Information inside the knowledge model is represented as a collection of entities and edges of a graph. Entities populate the areas of the knowledge model and are named as the category of information they contain. Three distinct areas on the left of the figure regard established knowledge. The VARIANTS area denotes the variants of SARS-CoV-2 (i.e., viruses which have progressively evolved from the original Wuhan virus by accumulating mutations). The same variant can be named differently and can be described by a different set of characterizing mutations (i.e., most relevant or prevalent mutations); these are described by the `Naming` and `Characterization` entities, which can both be assigned by multiple organizations. Mutational processes concern specific positions of a variant sequence; in this work, we consider the mutations that affect viral proteins, also called amino acid changes (`AA positional changes`). The EFFECTS area describes effects that are relevant from the points of view of epidemiology (e.g., viral transmission and disease severity) or immunology (e.g., host receptor binding affinity and sensitivity to monoclonal antibodies). An `Effect` is described as the consequence of a single amino acid change or of a variant. Each `Effect` is associated with the source of information that reports it, i.e., `Evidence` from scientific literature. The POSITIONS OF INTEREST area contains all the possible mutations (here called `AA Positional Changes`), which fall within annotated regions of the genome, i.e., proteins, and are present in some variant characterization.

The two areas on the right part of the model contain instead data collected and deposited within large databases. The SEQUENCE DATA area concerns viral samples (in the `Sequence` entity) provided by laboratories with their describing metadata. Out of the many aspects that describe each sequence, we hereby consider only the protein mutations (i.e., `AA changes`). Depending on the characteristics of the viral host organism, given protein regions of the virus may be antigenic and elicit an immune response; the EPITOPES area describes these regions by means of the `Epitope` entity, whose attributes include the starting and ending position of the region within a protein. Note that epitopes can be connected to the `AA positional changes` in the POSITIONS OF INTEREST area (and the `AA changes` in the SEQUENCE DATA area) when the position attribute *pos* of a change is included in the *start-stop* range of the epitope.

Within and across the areas, entities are connected by relationships of various types, defining the connection between their concepts: indirect edges for many-to-many cardinality, direct edges for one-to-many cardinality, empty-point-arrow edges for generalization hierarchy. Dashed lines represent connections between parts of the model that are based on positional information.

In building the CoV2K content, we employed a classical data integration process driven by the model, with pipelines for the integration and harmonization of different data silos. Knowledge areas are filled via the most updated sources in the landscape of SARS-CoV-2-related information, including CoVariants.org [13], Public Health England (PHE, [14]), the COG-UK Mutation Explorer [15], ECDC [16], and several preprints or published papers deposited on bioRxiv, medRxiv, or PubMed. For what concerns sequence data, CoV2K includes two large databases, GenBank and COG-UK as extracted via our ViruSurf pipeline [17, 18]. We include epitopes experimented for SARS-CoV-2 from the Immune Epitope Database (IEDB [19]).

According to our methodological framework, we selected CoV2K as the principal source of extensive knowledge. Several key features justify such a choice. First, the graph-like organization of information into entities and relations makes the translation into predicates straightforward. Second, the availability of already cleaned/integrated data and knowledge-related information is a unique feature of CoV2K that greatly simplifies the process of data ingestion. Third, CoV2K is provided online through an application programming interface (API) at http://www.gmql.eu/cov2k/api; as APIs are meant to connect systems, a multitude of data processing systems supports them, including Vadalog. Thus, we are able to query CoV2K and integrate the request's output for further analysis directly inside the reasoner.

In order to use CoV2K within a declarative language, we defined Vadalog rules' head-predicates that correspond to:

- Entities, using the notation $\langle entity\rangle Properties(\{attributes\, of\ entity\})$; for example, *effectProperties(e,type,level,method)* is an atom whose variables evaluate to any instance of the `Effect` entity.

- Relationships between two entities, using the notation $\langle entity1\rangle Of\langle Entity2\rangle(e1, e2)$; for example, *effectOfVariant(e,v)* evaluates to all the pairs of effects *e* and variants *v*, where *e* is an instance of the entity `Variant effect`, *v* is an instance of the entity `Variant`, and there exists a path connecting *v* and *e* within CoV2K.

The binding of CoV2K instances to the variables inside predicates is made by replacing the variables to valued constants. Namely, an atom such as $effectProperties($1143, "infectivity"", "higher", "epidemiological" $)$ is called a fact and denotes an instance of the `Effect` entity having the ID 1143 and describing higher infectivity ascertained through an epidemiological study. Rules' body-predicates are freely named according to the semantics of the use cases.

Furthermore, we present a recurrent predicate form that allows to group several instances of the CoV2K model into a set-like variable. To this aim, we adopt operators denoting monotonic aggregation, such as "munion", that is used to recursively form a set as the union of its components. In the example rule (1), we iterate over the variables of the atom *aaChangeOfCharacterization* and, for every characterization c, we add the amino acid changes a of c to the set

A.

$$\begin{aligned} aaChangeOfCharacterization(a, c), A = munion(a), \\ \rightarrow aaChangeSetOfCharacterization(A, c) \end{aligned} \tag{1}$$

4. Applying Vadalog to Viral Genomics Problems

In the following, we show a number of formalized examples of biological problems in the domain of viral sequences. Following a pedagogical approach, the first two examples are meant for readers to get acquainted with the Vadalog syntax and the predicates representing the knowledge graph of CoV2K. The third use case presents a more advanced application that uses several key features of Vadalog: monotonic aggregates, graph exploration and Set objects. The example proposes possible new SARS-CoV-2 variant characterizations, suggested by the co-occurrence of new amino acid changes with existing current variant characterizing changes. The immunogenic consequence of each acquired change is then evaluated through CoV2K by exploring the associated `Change effect` and the newly modified `Epitope` regions. The last example finds the most discussed amino acid changes in the scientific literature by traversing the graph generated by references of scientific articles. In doing so, we provide an example of a recursive procedure and show how it is implemented in Vadalog.

4.1. Most Represented Effects of Variants in Published Scientific Articles

Variants differ in their genomic characteristics and hence in their effects. Here, we are interested in finding which are the variant effects that are most discussed in the scientific community. This can be formalized as in rule (2).

$$\begin{aligned} effectOfVariant(e, v), \\ effectProperties(e, type, level, method), \\ evidenceOfEffect(ev, e), \\ evidenceProperties(ev, cit, \text{“published"}, doi, publisher) \\ \rightarrow publishedVariantEffects(type, level, method, count()), \end{aligned} \tag{2}$$

In the rule body, we join different predicates from CoV2K EFFECTS area through the effect *e*: *effectOfVariant*, *effectProperties*, *evidenceOfEffect*, and *evidenceProperties*. We select only the effects from peer-reviewed scientific articles by setting the value “published” as the evidence *type*. Finally, in the rule head, we count the occurrences of every combination of the effect’s *type*, *level* and *method* attributes. The count for the tuple ⟨*type, level, method*⟩ is obtained through grouping by the number of variants *v* and evidence *ev* sharing the effect. By sorting the values in *publishedVariantEffects* by descending count, we obtain the most cited variant effects. As shown in Table 1, the most relevant results are those related to lower effectiveness of vaccines / sensitivity to convalescent sera and to higher risk of hospitalization / disease severity.

Table 1
Excerpt of the results computed by *publishedVariantEffects*, showing the effects that are most represented in articles of CoV2K.

type	level	method	count
effectiveness of vaccines	lower	epidemiological	18
sensitivity to convalescent sera	lower	experimental	17
risk of hospitalization	higher	epidemiological	13
disease severity	higher	epidemiological	10

4.2. The Delta Variant Characterizations

Since the beginning of the pandemic, several organizations provided their own nomenclature and description for variants. The CoV2K knowledge model clusters different namings for each variant; for example, the well known Alpha variant (as named by the World Health Organization, WHO [20]) was also called B.1.1.7 and VOC-20DEC-01, among others. A user may want to request the variant's characterization based on one of the many available names. Suppose we are interested in the Delta variant; we traverse the edge predicate *namingOfVariant*, by setting "Delta" as the naming attribute. From the variant *v*, we can find the characterization *c* of a variant according to the organization *org*. We then obtain the amino acid changes that are part of the characterization through the relation *aaChangeOfCharacterization.* As multiple organizations may provide the characterization of a variant, the inclusion of *org* in the rule's head allows to indicate the source.

$$\begin{aligned}
namingOfVariant(\text{“Delta"}, v),\\
characterizationOfVariant(c, v),\\
characterizationProperties(c, org, v\_class),\\
aaChangeOfCharacterization(a, c)\\
\rightarrow deltaCharacterization(v, org, a)
\end{aligned} \tag{3}$$

The rule (3) produces triplets $\langle v, org, a\rangle$, where each amino acid change a is associated to the kind of variant v and organization org who provided the characterization. With an additional rule (4) we build one set of mutations for each pair $\langle v, org\rangle$. The result of rule (4) clarifies that the WHO-named Delta variant maps to two distinct but similar variants v commonly referenced in the scientific community as "B.1.617.1" and "B.1.617.2" using the Pango [21] nomenclature. CoV2K provides two complete characterizations for each of them, according to the variant description published by the organizations (org) Public Health England (PHE) and Covariants.org. As a result, according to PHE, the variants "B.1.617.1" and "B.1.617.2" are described with respectively 10 and 8 amino acids, while Covariants includes many more, reaching 43 and 22 characteristic amino acids respectively. In Table 2 we show some of the amino acid changes corresponding to rule (4).

Table 2
Excerpt of the results computed by the rule *deltaCharacterizationSet*.

v	org	A
B.1.617.1	PHE	{NSP6:T77A, S:L452R, S:E484Q, ...}
B.1.617.2	PHE	{M:I82T, S:L452R, S:D950N, ...}
B.1.617.1	Covariants.org	{M:I82T, N:D377Y, S:E154K, ...}
B.1.617.2	Covariants.org	{M:I82T, N:R203M, S:E156-, ...}

$$\begin{aligned} &deltaCharacterization(v, org, a), A = munion(a) \\ &\quad \rightarrow deltaCharacterizationSet(v, org, A) \end{aligned} \tag{4}$$

4.3. Severity of Putative New SARS-CoV-2 Variants

Since the beginning of the pandemic, hundreds of SARS-CoV-2 variants spread across the world, and some of them completely replaced the other circulating variants in a few weeks; this is the case of the Delta variant, for example. Variants are essentially a collection of amino acid changes that are found to consistently co-occur in sequences. Such amino acid changes can bring substantial advantages or disadvantages to the virus in terms of infectivity, infection severity, among other aspects. Since a new variant could spread quickly and bring unexpected threats to living beings, it is important to monitor circulating viral sequences. Putting in place methods that anticipate variants contributes to develop adequate countermeasures against the evolving pandemic. In this use case, we focus on the characterization of previously unobserved and possible emerging SARS-CoV-2 variants. Then, using CoV2K, we discover the effects possibly inherited by the novel characterization as well as their consequences from the point of view of immunogenicity.

Using CoV2K, we can browse millions of SARS-CoV-2 sequences collected worldwide and look at their genomic features. Changes inside a sequence are not statistically independent events (for biological reasons pertaining to the chemical stability and protein folding mechanisms); indeed, some changes are found to co-occur in viral samples with a high frequency. The pairs of amino acid changes that are frequently co-occurring are called *coPairs* in rule (5). We assign to each such pair a weight w corresponding to the number of sequences that exhibit the co-occurrence relation divided by the total number of sequences in the database (totalSequences).

$$\begin{aligned} aaChangeOfSeq(a1, s), aaChangeOfSeq(a2, s), a1 != a2, \\ w = count(s)/\text{totalSequences} \\ \rightarrow coPairs(a1, a2, w) \end{aligned} \tag{5}$$

Considering a simplified problem setting, we assume that the frequent co-occurrence of two amino acid changes is connected to an advantageous characteristic for the spread of the virus. Therefore, given a frequently co-occurring pair of amino acid changes, there may be a high chance of observing the same pair of changes also in future sequences. In fact, this assumption is only valid in specific circumstances, not considered here for simplification.

We can model a putative new SARS-CoV-2 variant (named *novelVariantCharacterization* in the following) as an existing variant that acquires a change appearing in a frequently co-occurring pair of amino acid changes. First, we consider the set A of characterizing changes of variant *v*, computed as in rule 6.

$$\begin{aligned} aaChangeSetOfCharacterization(A, c), characterizationOfVariant(c, v) \\ \rightarrow aaChangeSetOfVariant(A, v) \end{aligned} \tag{6}$$

Then, for each pair $\langle a1,a2 \rangle$ in a *coPairs* relation (where *a1* is part of A and *a2* is not) our rule (7) builds a *novelChangeForVariant(v,a2)*. The result is equipped with a weight w_{a2}, representing the maximum weight $w_{a1,a2}$ among *coPairs* built between *a2* and any change of *A*.

$$\begin{aligned} aaChangeSetOfVariant(A, v), \\ a1 \in A, a2 \notin A, coPairs(a1, a2, w_{a1,a2}), \\ w_{a2} = max(w_{a1,a2}) \\ \rightarrow novelChangeForVariant(v, a2, w_{a2}) \end{aligned} \tag{7}$$

With rule (8), we generate a *novelVariantCharacterization* by joining a novel change with an existing *aaChangeSetOfVariant*. The third line in the rule's body references the definition given in rule (9). The line serves the purpose of excluding any potential characterization coinciding with an already existing variant characterization. To this aim, in the body of rule (8), we generate a new variant characterization *B*, made by the characterization of *v* plus *a*, and check that it is not present in *anyExistingCharacterization*.

$$\begin{aligned} novelChangeForVariant(v, a, w), \\ aaChangeSetOfVariant(A, v), \\ \neg anyExistingCharacterization(B), B = A \cup \{a\} \\ \rightarrow novelVariantCharacterization(v, A, a, w) \end{aligned} \tag{8}$$

$$aaChangeSetOfVariant(B, v) \rightarrow anyExistingCharacterization(B) \tag{9}$$

We can then find the effects of a *novelVariantCharacterization* through rule (10), where we join the novel amino acid change with *effectOfAAChange* and *effectProperties*; a small subset of the extracted *novelVariantEffects* is presented in Table 3.

$$\begin{aligned} novelVariantCharacterization(v, A, a, w), \\ effectOfAAChange(e, a), effectProperties(e, type, level, method) \\ \rightarrow novelVariantEffect(v, a, w, type, level, method) \end{aligned} \tag{10}$$

Table 3
Excerpt of the results computed by *novelVariantEffect*, showing the effects connected to amino acid changes possibly acquired by variants.

v	a	w	effect type	level	method
B.1.1.529	S:L452R	0.635	binding to host receptor	higher	computational
B.1.1.529	S:L452R	0.635	sensitivity to neutralizing monoclonal antibodies	lower	experimental
B.1.351	S:P681H	0.169	protein flexibility	higher	computational
B.1.351	S:P681H	0.169	protein stability	higher	computational
B.1.1.318	S:S477N	0.157	infectivity	higher	epidemiological
B.1.1.318	S:S477N	0.157	viral transmission	higher	inferred

Some amino acid changes can be more relevant than others depending on their position along the viral sequence; special regions along the viral proteins, called epitopes, are short amino acids sequences recognised by the host organism as external organisms, thus capable of eliciting an immunogenic response. The host organism may fail to recognize a mutated epitope, so these regions are typically of large interest to scientists involved in immunological studies and vaccine development – the typical purpose of vaccines is that of helping the organism to recognize the virus epitopes. Amino acid changes can be linked to their enclosing epitopes using positional information. In rule (11), we extract pairs of amino acid changes *a* and epitopes *e* that are *aaChangeModifyingEpitope*.

$$\begin{aligned}aaChangeProperties(a, protein, pos, ref, alt, type, length)\\ epitopeProperties(e, protein, \text{“human"}, start, stop, string, iedb\_link),\\ start \leq pos \leq stop\\ \rightarrow aaChangeModifyingEpitope(a, e)\end{aligned} \tag{11}$$

As we are interested in the new epitopes modified by the acquisition of a novel change into a variant characterization, we ignore the epitopes that are already modified. Rule (12) joins *aaChangeOfCharacterization* with *aaChangeModifyingEpitope* and recursively forms the set *E* as the union of all the epitopes modified by a characterization. Then, we join the characterization to its variant.

$$\begin{aligned}aaChangeModifyingEpitope(a, e), E = munion(e),\\ aaChangeOfCharacterization(a, c),\\ \rightarrow modifiedEpitopesOfCharacterization(E, c)\\ \\ modifiedEpitopesOfCharacterization(E, c),\\ characterizationOfVariant(c, v)\\ \rightarrow modifiedEpitopesOfVariant(E, v)\end{aligned} \tag{12}$$

In rule (13), by joining the amino change *a* of a *novelVariantCharacterization* with *aaChangeModifyingEpitope*, we can find epitopes that could potentially be mutated by the new characterization

Table 4
Excerpt of the results computed by *novelModifiedEpitope*, showing few epitopes impacted by new changes of the variants.

v	a	start	stop	string	w	iedb_link
AV.1	S:G446S	444	446	KVG	0.146	http://www.iedb.org/epitope/1310239
B.1.1.519	S:L452R	452	452	L	0.635	http://www.iedb.org/epitope/1075135
P.1	S:S477N	473	477	YQAGS	0.157	http://www.iedb.org/epitope/1083498

of the variant. While doing this, we first exclude those epitopes that are already modified by the same variant. Therefore, the epitope modified by the novel change must not appear in the *modifiedEpitopesOfVariant* of *v*. In addition to the variant, the change and the weight, our result (e.g., shown in Table 4) includes the epitope *start* and *stop* coordinates, the sequence of amino acids (*string*) forming the epitope and the corresponding link (*iedb_link*) to the used epitope's data source, i.e., IEDB.

$$\begin{aligned} novelVariantCharacterization(v, A, a, w), \\ aaChangeModifyingEpitope(a, e), \\ modifiedEpitopesOfVariant(E, v), e \notin E \\ epitopeProperties(e, protein, \text{“human"}, start, stop, string, iedb\_link) \\ \rightarrow novelModifiedEpitope(v, a, start, stop, string, w, iedb\_link) \end{aligned} \tag{13}$$

4.4. Amino Acid Changes Relevant for the Scientific Community

Among the methods employed into COVID-19 vaccines to defend against the infection, one is to teach the immune system to recognize and neutralise the Spike protein. This particular protein surrounds the external surface of the viral capsid and starts the infection process by hooking into the host cells. However, while the virus can quickly evolve and adapt to the environment, vaccines take a long time to develop and be updated. Hence, a highly mutated variant of the Spike protein could be treated as an unknown sequence by the host, evading the protection given by the current vaccines. As such, biologists typically look to specific changes in the amino acid sequence of the Spike protein with much interest.

Based on such premise, the goal of this example is to find which Spike amino acid changes are the most studied by scientists. To measure the interest in the scientific community, we resort the the list of published articles discussing a change. In rule (14), we look for the amino acid changes falling inside the Spike protein through the predicate *aaChangeProperties* where we set the protein value to “Spike". We join the changes with their effects and the articles discussing them. The predicate *citedBy* binds any Spike amino acid change *a* to the Document Object Identifier (DOI) of the associated evidence, neglecting the information about the effect as it is not relevant for our goal. Furthermore, we annotate *citedBy* with the distance (*hops*) between the change and the evidence discussing it. As rule (14) explores the pieces of evidence that are directly reported in CoV2K, we assume the distance to be the minimum value, equal to 1.

$$\begin{aligned} aaChangeProperties(a, \text{“Spike"}, pos, ref, alt, type, len),\\ effectOfAAChange(e, a), evidenceOfEffect(ev, e),\\ evidenceProperties(ev, cit, \text{“published"}, doi, publisher), hops = 1,\\ \rightarrow citedBy(a, doi, hops) \end{aligned} \tag{14}$$

In addition, we can enrich the previous analysis by adding the resonance of a piece of information in the scientific literature beyond its first appearance. To do so, we query the OpenCitations [22] database through the function *findCitations*[1] which returns a list of DOIs referencing the input argument. This feature is used inside rule (15) to find the scientific papers citing the articles found in the previous rule. From an *article* in *citedBy*, we call the function *findCitations*, obtaining a list of scientific papers citing the original one. The *citationList* is stored in a predicate *citedByList* and connected to the referenced amino acid change *a*, the size of the list (*lastIndex*), and the updated distance value (*hops*).

$$\begin{aligned} citedBy(a, article, hops),\\ citationList = findCitations(article),\\ lastIndex = size(citationList)\\ hops \leq \text{threshold},\\ \rightarrow citedByList(a, citationList, lastIndex, hops + 1) \end{aligned} \tag{15}$$

The procedure is repeated for as many articles as those in *citedBy* and until an arbitrary threshold value for *hops* is reached; this, eventually, happens when new atoms of *citedBy* are created as an effect of rule (16).

$$\begin{aligned} citedByList(a, citationList, lastIndex, hops),\\ article = citationList[lastIndex], lastIndex > 0\\ \rightarrow citedByList(a, citationList, lastIndex - 1, hops),\\ citedBy(a, article, hops) \end{aligned} \tag{16}$$

The purpose of rule (16) is to unpack a *citationList* into individual atoms of *citedBy*. The generated instances of *citedBy* connect each item of *citationList* to the amino acid change from *citedByList*. As new *citedBy* atoms are derived from rule (16), rule (15) runs again in a recursive fashion to find additional citations of a change, up to the *threshold* distance.

Rule (16) is also recursive: it considers a *citedByList* atom and uses the value of *lastIndex* to access the list of items. At every iteration, the rule selects a new *article* from *citationList* using the current value of *lastIndex*; then the value is decreased by one and saved into a new *citedByList*. The procedure continues until *lastIndex* becomes zero, i.e., when all the list items have been visited.

[1]As Vadalog supports the invocation of external Java methods, we packed the instructions for making API requests inside a Java executable file (JAR) that exposes the method *findCitations*.

Table 5
Excerpt of the result computed by *aaChangeRelevance*.

a	citations	hops	a	citations	hops	a	citations	hops
S:R408K	1	1	S:C15S	2	1	S:F490S	6	1
S:R408K	402	2	S:C15S	417	2	S:F490S	1448	2
S:R408K	4821	3	S:C15S	5023	3	S:F490S	14303	3

Finally, rule (17) counts the number of articles that reference Spike amino acid changes, by aggregating the atoms of *citedBy* by change and *hops*. According to our initial assumption, the value of *citations* reflects the interest of the scientific community. In the results (e.g., see Table 5) we just list, for each amino acid change, the counts of *citations* for each *hop*. Note that the first and third reported changes are located in a specific area of the Spike protein – the Receptor Binding Domain – that is of particular interest as it corresponds to the first access point of the virus into hosts' cells [23].

$$\begin{aligned} &citedBy(a, article, hops), citations = count(article) \\ &\quad \rightarrow aaChangeRelevance(a, citations, hops) \end{aligned} \tag{17}$$

5. Discussion and Conclusions

In this paper, we used a knowledge base and logical reasoning language for exploring a number of crucial aspects of the evolution of SARS-CoV-2 variants, which are important, in turn, for understanding the progress of the COVID-19 pandemic. We resorted to CoV2K, which collects information about SARS-CoV-2 in a knowledge graph, as it is the result of a significant data abstraction and integration process, which exploits large databases as well as ontological resources. In our declarative problem descriptions, we used several key features of Vadalog, including recursion, grouping with counting, and monotonic aggregation, whose expressive power was sufficient for our needs.

The methodological framework hereby discussed is a showcase of how declarative languages such as Vadalog could be applied to biological problems. Indeed, declarative languages allow answering complex queries using clearly stated logical formulae, favoring the progressive modeling and understanding of the essence of problems – during the rule production phase and then for sharing and maintaining them. In contrast, biologists typically face these problems through low-level abstractions, often based upon custom scripts, hard to explain and to maintain. Thus, this paper introduces a major challenge – can a logical language effectively be used in biology? The challenge is quite hard, as biology is intrinsically a very complex application domain. Our experience shows some very concrete use cases as proof of concept, and provides a first step along this direction.

Acknowledgments

This research is funded by the ERC Advanced Grant 693174 GeCo (Data-Driven Genomic

Computing).

References

[1] B. Billoud, M. Kontic, A. Viari, Palingol: a declarative programming language to describe nucleic acids' secondary structures and to scan sequence databases, Nucleic Acids Research 24 (1996) 1395–1403.

[2] H. Jamil, B. El-Hajj-Diab, Bioflow: A web-based declarative workflow language for life sciences, in: 2008 IEEE Congress on Services-Part I, IEEE, 2008, pp. 453–460.

[3] I. Raikov, E. De Schutter, The layer-oriented approach to declarative languages for biological modeling, PLoS Computational Biology 8 (2012) e1002521.

[4] J. Seo, Datalog extensions for bioinformatic data analysis, in: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2018, pp. 1303–1306.

[5] L. Bellomarini, G. Gottlob, A. Pieris, E. Sallinger, Swift logic for big data and knowledge graphs, in: IJCAI, ijcai.org, 2017, pp. 2–10.

[6] T. Alfonsi, R. Al Khalaf, S. Ceri, A. Bernasconi, CoV2K model, a comprehensive representation of SARS-CoV-2 knowledge and data interplay, Scientific Data 9 (2022) 1–12.

[7] L. Bellomarini, E. Sallinger, G. Gottlob, The vadalog system: Datalog-based reasoning for knowledge graphs, Proc. VLDB Endow. 11 (2018) 975–987.

[8] A. Calì, G. Gottlob, T. Lukasiewicz, A general datalog-based framework for tractable query answering over ontologies, J. Web Semant. 14 (2012) 57–83.

[9] G. Gottlob, A. Pieris, Beyond SPARQL under OWL 2 QL entailment regime: Rules to the rescue, in: IJCAI, AAAI Press, 2015, pp. 2999–3007.

[10] G. Berger, G. Gottlob, A. Pieris, E. Sallinger, The space-efficient core of vadalog, ACM Trans. Database Syst. 47 (2022) 1:1–1:46.

[11] C. Zaniolo, M. Yang, M. Interlandi, A. Das, A. Shkapsky, T. Condie, Fixpoint semantics and optimization of recursive datalog programs with aggregates, CoRR abs/1707.05681 (2017).

[12] R. Al Khalaf, T. Alfonsi, S. Ceri, A. Bernasconi, CoV2K: A Knowledge Base of SARS-CoV-2 Variant Impacts, in: S. Cherfi, A. Perini, S. Nurcan (Eds.), Research Challenges in Information Science, Springer International Publishing, Cham, 2021, pp. 274–282.

[13] E. B. Hodcroft, CoVariants: SARS-CoV-2 Mutations and Variants of Interest, 2022. URL: https://covariants.org/, last accessed: June 30th, 2022.

[14] Public Health England, COVID-19 variants: genomically confirmed case numbers, 2022. URL: https://www.gov.uk/government/publications/covid-19-variants-genomically-confirmed-case-numbers, last accessed: June 30th, 2022.

[15] The COVID-19 Genomics UK (COG-UK) consortium, An integrated national scale SARS-CoV-2 genomic surveillance network, The Lancet. Microbe 1 (2020) e99.

[16] European Centre for Disease Prevention and Control, SARS-CoV-2 variants of concern, 2022. URL: https://www.ecdc.europa.eu/en/covid-19/variants-concern, last accessed: June 30th, 2022.

[17] T. Alfonsi, P. Pinoli, A. Canakoglu, High performance integration pipeline for viral and epitope sequences, BioTech 11 (2022) 7.

[18] A. Canakoglu, P. Pinoli, A. Bernasconi, T. Alfonsi, D. P. Melidis, S. Ceri, ViruSurf: an integrated database to investigate viral sequences, Nucleic Acids Research 49 (2021) D817–D824.

[19] R. Vita, S. Mahajan, J. A. Overton, S. K. Dhanda, S. Martini, J. R. Cantrell, D. K. Wheeler, A. Sette, B. Peters, The immune epitope database (IEDB): 2018 update, Nucleic Acids Research 47 (2019) D339–D343.

[20] World Health Organization, Tracking SARS-CoV-2 variants, 2022. URL: https://www.who.int/en/activities/tracking-SARS-CoV-2-variants/, last accessed: June 30th, 2022.

[21] A. Rambaut, E. C. Holmes, Á. O'Toole, V. Hill, J. T. McCrone, C. Ruis, L. du Plessis, O. G. Pybus, A dynamic nomenclature proposal for SARS-CoV-2 lineages to assist genomic epidemiology, Nature Microbiology 5 (2020) 1403–1407.

[22] S. Peroni, D. Shotton, Opencitations, an infrastructure organization for open scholarship, Quantitative Science Studies 1 (2020) 428–444.

[23] J. Lan, J. Ge, J. Yu, S. Shan, H. Zhou, S. Fan, Q. Zhang, X. Shi, Q. Wang, L. Zhang, et al., Structure of the sars-cov-2 spike receptor-binding domain bound to the ace2 receptor, Nature 581 (2020) 215–220.