Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 487–496

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

OntoRepliCov : an Ontology-Based Approach for Modeling the SARS-CoV-2 Replication Process

Wissame Laddada[a,b,*], Lina F. Soualmia[a], Cecilia Zanni-Merk[a], Ali Ayadi[b], Claudia Frydman[b], India L'Hote[c], Isabelle Imbert[c]

[a]*Normandie Universit, LITIS, 7600 Rouen, France*
[b]*Aix-Marseille Universit, LIS, 13009 Marseille, France*
[c]*Aix-Marseille Universit, AFMB, 13009 Marseille, France*

Abstract

Understanding the replication machinery of viruses contributes to suggest and try effective antiviral strategies. Exhaustive knowledge about the proteins structure, their function, or their interaction is one of the preconditions for successfully modeling it. In this context, modeling methods based on a formal representation with a high semantic expressiveness would be relevant to extract proteins and their nucleotide or amino acid sequences as an element from the replication process. Consequently, our approach relies on the use of semantic technologies to design the SARS-CoV-2 replication machinery. This provides the ability to infer new knowledge related to each step of the virus replication. More specifically, we developed an ontology-based approach enriched with reasoning process of a complete replication machinery process for SARS-CoV-2. We present in this paper a partial overview of our ontology ***OntoRepliCov*** to describe one step of this process, namely, the continuous translation or protein synthesis, through classes, properties, axioms, and SWRL (Semantic Web Rule Language) rules.

Keywords: Knowledge representation; biology domain; reasoning process; SWRL.

1. Introduction

On March 2020, the world witnesses the World Health organization declaring the COVID-19 (COrona VIrus Disease 2019) as a pandemic. The illness is caused by the coronavirus SARS-CoV-2 (Severe Acute

* *E-mail address:* wissame.laddada@univ-amu.fr

1877-0509
10.1016/j.procs.2021.08.050

Respiratory Syndrome CoronaVirus 2). This is not the first coronavirus infection. In fact, in the two past decades, two other pathogenic coronaviruses for human emerged; namely, the SARS (Severe Acute Respiratory Syndrome) in 2002 in China, and the MERS-CoV (Middle East respiratory syndrome coronavirus) in 2012 in Saudi Arabia. Detailed knowledge about the sequence of coronaviruses is a key to understand their life cycle. This life cycle concerns specifically the replication machinery of the virus. The biological process of coronavirus replication implies first, the production of several viral proteins. Then, interactions between them lead to genome replication.

The reproduction of the replication machinery through an automatic approach provides relative ease to extract elements such as protein sequences, resulting from the replication steps. Moreover, adding metadata related to genome knowledge would help biological experts to perform pattern analysis/ recognition of amino acid sequences. With this goal in mind, we expect ontologies to be a relevant way to add semantics to the genome sequence and help the design of the replication machinery. Modeling SARS-CoV-2 knowledge with an ontology should provide an exhaustive vocabulary related to the virus replication. Additionally, ontologies are highly qualified when asserting relations between entities. Each entity can be defined with a complex formalization, furthering the likelihood of inferring new knowledge with only the SARS-CoV-2 genome as an assertion.

The SARS-CoV-2 is a positive single-stranded RNA genome (30 kilobases) that encompasses several open reading frames (ORFs). Two of them are major in the replication machinery: The ORF1a and the ORF1b. The translation process handled by a ribosomal activity produces an overlapping between the two frames. A large part of the genome $(2/3)$ is included in these two frames and translated to produce two polyproteins $pp1a$ and $pp1ab$. Polyproteins gather sixteen nonstructural proteins (NSPs). The rest of the genome $(1/3)$ encodes structural proteins such as spike (S), Membrane (M), Envelope (E), Nucleocapsid (N), and other accessory proteins [18].

This hierarchical structure of the virus organization consolidates our idea of an ontology-based approach. Furthermore, the entities resulting from the replication steps are easily considered as a consequence in inferences. Accordingly, this paper describes an approach based on semantic technologies to design the SARS-CoV-2 organization and the first step of the replication process. After browsing some related work in section 2, we present the **OntoRepliCov** domain of interest in section 3 where atomic concepts of the virus organization are defined. An explicit model of our knowledge base is detailed in section 3.2 with entities involving the translation step of the replication. Also, relationships –routinely created through data properties and object properties–, are proposed to gather the interactions among concepts. Semantic enhancement through a reasoning process using axioms and rules is developed in section 3.3. Section 4 develops an example to show the feasibility of our approach on a concrete situation. Finally, we present our conclusion and some perspectives of future work.

2. Related work

Mathematical and statistical models contribute in several issues related to virology [11]. Modeling and simulating the molecular-level dynamics of SARS-CoV-2 in detail at every stage can greatly aid understanding and management of infection with the virus [1]. For example, dynamics of Ebola [17], influenza A [10], HIV-AIDS [15] and Zika [4] have been subjects for virus modeling. Differential equations are the main formalization for these methods. Even though their relevance, these methods require optimization [7]. In fact, a significant computation power is needed to obtain quantitative solutions. Furthermore, obtaining a value for the huge number of parameters through theoretical or experiment calculations is impossible [19].

Since the beginning of the current pandemic COVID-19, many challenges have emerged. Consequently, a lot of methods using artificial intelligence (machine and deep learning) are constantly proposed to suggest solutions for different issues: predict the protein-protein interactions [5], mutation studies [8], SARS-CoV-2 detection (through screening) [20], drug discovery [6], detecting symptoms [14], etc. However, to the best of our knowledge, there is no complete modeling for the SARS-CoV-2 life cycle. According to the literature, some ontologies were designed to detail knowledge related to the SARS-CoV-2. OGG-CoV is an ontological

representation (high level) of genes and genomes that encompasses several classes and properties allowing us to understand the SARS-CoV-2 structure and compare different genes [12] but the different steps of the replication process are not designed in this ontology. Gene Ontology (GO) [2] is popular when it comes to attributing genes and gene products of different species in biology. In the context of the COVID-19, this ontology is used to gather knowledge on proteins interacting with the host cells [9] but there is no representation for the SARS-CoV-2 life cycle in this ontology. The CoV2K knowledge base [13] features a taxonomy of variant impacts. It is designed with three main categories (protein stability, epidemiology, and immunology). This ontology encompasses knowledge about the protein encoded by the gene (called product), the type of variant (i.e., substitution, insertion, or deletion), the amino acid variation (composed of a reference sequence, its position on the reference genome, and an alternative sequence), etc. However, the CoV2K does not detail knowledge about the replication steps such as the transcription activity that may impact the amino acid variations. The knowledge representation through the ontology CIDO is designed for host-coronavirus interactions and their interactions with individual drugs [16]. Therefore, it includes some keywords related to the replication process. Semantic enrichment and alignment of some ontologies (VIDO, CIDO, and IDO-COVID-19) involved the integration of the replication process [3]. Nevertheless, the micro-level is not included and the replication steps are represented only through classification and not detailed. This ontologies enhancement does not handle all the replication process which is our main aim.

3. Ontology-based approach

Using ontologies in a specific field adheres to the philosophical idea about describing all existing entities, properties, and their connections. Hence, different ontologies should be created to carry out the enhancement of concepts or facts that cover the entire field. We describe in this section an ontology representing the genome structure and a part of the replication process of SARS-CoV-2 (protein synthesis). Then, we present rules and axioms that handle the translation process from the virus life cycle.

3.1. The domain of interest

For a better understanding, we start by explaining the translation process, a step of the virus life cycle through its replication. Thereafter, we highlight the most important components of the ontology.

3.1.1. Translation process

After the injection of the genome RNA (gRNA+) to the cell for the replication process, it will be considered as a messenger RNA (mRNA) to be first, translated by the ribosome. The replication process includes two sub-processes. We summarize them in the following:

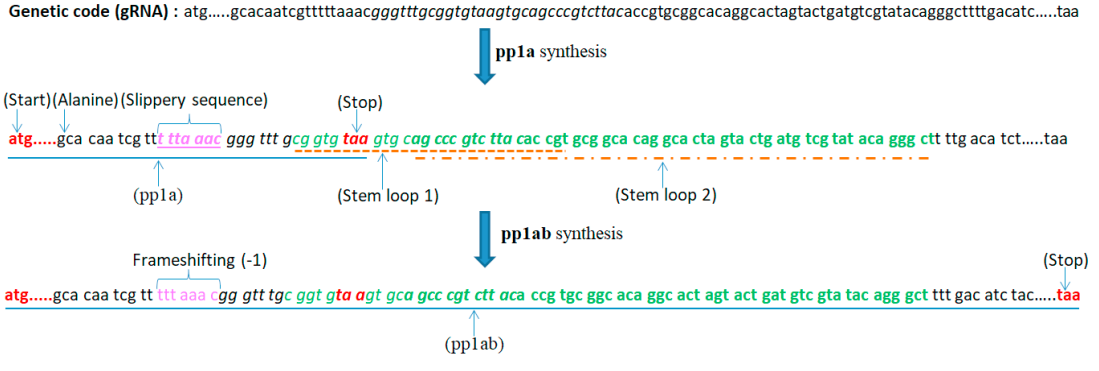


Fig. 1: Protein synthesis steps example: from a nucleobase sequences to polyproteins

- Translation of ORF1a and ORF1b yields the polyproteins $pp1a$ and $pp1a$b (continuous RNA synthesis)
- Discontinuous RNA synthesis includes translation and transcription of mRNAs that encode structural and accessory proteins

Our point of interest is the first step of the replication process (continuous RNA synthesis). The translation process also called protein synthesis is ensured by the ribosomal activity that converts only the genetic code (sequence of nucleobases) of the ORF1a and ORF1b into amino acids. The process follows the steps illustrated in Figure 1 as explained below:

i) The host ribosome reads the genetic code by triplet called codon (three nucleobases at a time). Each codon will form an amino acid starting therefore with the amino acid Methionine/Start (ATG or AUG for RNA). Binding amino acids together contributes to building polyproteins $pp1a$ or $pp1ab$.
ii) When the ribosome encounters the Stop codon, the translation process of the ORF1a yields the polyprotein $pp1a$.
iii) Sometimes the proximity of a stem-loop (secondary structure) nearby the codons sequence (XXU UUA AAC), known as a slippery sequence, causes a ribosomal frameshifting (-1). In other words, the ribosome slips back one nucleobase allowing hence, the translation of ORF1b that yields $pp1ab$. The initial sequence of nucleobases (TAA) forming the Stop codon will be shifted allowing the translation progression ($pp1ab$ synthesis in Figure 1)

3.1.2. Atomic concepts

With these notions in mind, we identify the most important elements encoding the structure of the virus and establish links among them. Moreover, we highlight other elements generated through the replication steps. At first sight, the main atomic concepts to design the ontology are summarized in Table 1. This conceptualization highlights the domain and scope of **OntoRepliCov**.

Table 1: Main concepts of SARS-CoV-2 organization and life cycle

Main Concept	**Interpretation**
Genome	Encapsulates all elements that structure the virus
Element replication	Knowledge about elements resulting from replication process
Nucleobase	Adenine (A), Cytosine (C), Guanine (G), and Uracil (U) are the fundamental units that form the genetic code of an RNA (T in the genome is considered as U for RNA)
Codon	A sequence of three nucleobases
Amino acid	Defined by a codon, the several combinations of the nucleobases (triplets) gather 22 amino acids (e.g. AUG: Methionine)
Polyprotein	Continuous sequence of amino acids delimited by a start codon (Methionine) and a Stop Codon (e.g. UAA: Stop)
Nonstructural Proteins	Formed trough the virus replication when the polyproteins are processed by the viral proteases (NSP_1 to NSP_{16})
Structural proteins	Encapsulate the proteins S, M, E, and N
Accessory proteins	Gather several ORFs such as ORF10 or ORF8

The next section represents the ontology modeling with several classes related to the main concepts presented in Table 1 and roles determining relationships among them.

3.2. Ontology modeling

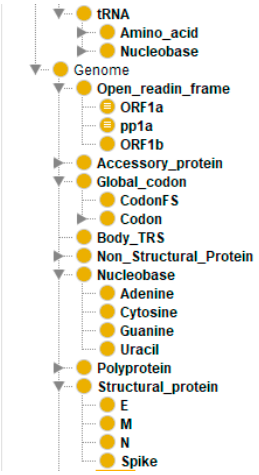
Knowledge and semantic representation are based on a formalism that allows the description of a given field and also performs a reasoning process to build a system based on a symbolic Artificial Intelligence.

Technical adjustments, encourage us to design our knowledge base with different vocabularies. **OntoRepliCov** is designed using Protégé[1] and made publicly available on BioPortal[2].

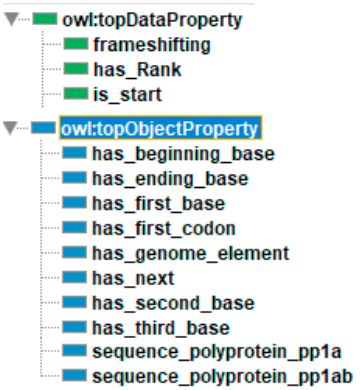
3.2.1. Classes taxonomy

Considering the atomic concepts presented in Table 1, two main classes describe the ontology; namely, **Replication_element** and **Genome**. The latter, captures a taxonomy of concepts related to the genome organization. Therefore, we add to the Genome class the sub-classes: **Accessory_protein**, **Open_reading_frame**, **Polyprotein**, **Stem_loop, Non_structural_protein**, **Structural_protein**, **Nucleobase**. Knowledge related to codons are also integrated. In Fact, as explained in section 3.1.1, the translation step generates two kinds of codons: initial codons and condons after the frameshifting. **OntoRepliCov** handles the two descriptions: the codons formed before the frameshifting are described with the class **Codon** and the class **CodonFS** represents codons produced after the frameshifting. A **Global_codon** class is included and congregates both of them. The latter class is considered as a sub class of Genome. Each NSP (from 1 to 16) is added as a concept and gathered by the class Non_structural_protein. Likewise, the class Polyproteins is extended with the concepts **pp1a** and **pp1ab**. Also, the class Nucleobase subsumes **Adenine**, **Cytosine**, **Guanine** and **Uracil** concepts.

Regarding the **Replication_element** class, several classes are introduced not only for the need of translation process but also for the different steps of the virus life cycle. We describe here the **Amino_acid** concept which is a subclasses of **tRNA** and gathers twenty amino acids plus the codons **Stop** and **Start**. Figure 2a illustrates a partial overview of the classes hierarchy. The next section describes the relationships between the classes.



(a) Partial taxonomy classes



(b) Partial properties overview

Fig. 2: Partial ontology overview

3.2.2. Properties definition

OntoRepliCov carries out other semantics through the existing roles between classes. Figure 2b shows these roles by means of properties (object properties and data properties). Each property P is presented here as a binary predicate *P(x,y)* to indicate that the class *x* is related to the class *y* through the property P. The sequence of nucleobases and codons is expressed by the *has_next (Codon/Nucleobase, Codon/Nucleobase)*

[1] https://protege.stanford.edu
[2] https://bioportal.bioontology.org/ontologies/ONTOREPLICOV

object property. Each nucleobase and each codon will have a rank starting with the integer '1'. This information is asserted through the data property *has_Rank (Codon/Nucleobase, xsd:integer)*. Formally, a codon is a sequence of three nucleobases. We take into account this fact by adding three object properties; namely, *has_first_base (Codon, Nucleobase)*, *has_second_base (Codon, Nucleobase)*, *has_third_base (Codon, Nucleobase)*. In our context, a stem-loop is delimited by two nucleobases. Consequently, we include the roles *has_beginning_base (Stem-loop, Nucleobase)* and *has_ending_base (Stem-loop, Nucleobase)*. The data property *frameshifting (Nucleobase, xsd:boolean)* marks with, a 'true' value, the nucleobase where the ribosme slips back in the translation process. Likewise, the starting codon of the translation process is labeled with a 'true' value through the property *is_start(Codon, xsd:boolean)*. Other roles are introduced for a reasoning mechanism to demarcate polyproteins: the object properties *sequence_polyprotein_pp1a (Codon, Codon)* and *sequence_polyprotein_pp1ab(Codon, CodonFS)*. This reasoning mechanism is explained in what follows.

3.3. Reasoning mechanism for the translation process

The reasoning process revolves around the translation process. It includes axioms based on Description Logic (DL) to infer an individual's type considering a class formalization. However, the high expressiveness of RDF, RDFS, and OWL does not allow us to perform some functions and infer new facts between entities. Consequently, we exploit SWRL[3] to enhance our model by adding a set of rules. We describe here both axiomatic and rule-based reasoning processes.

3.3.1. Rules reasoning

As explained in section 3.1.1, different steps occur during the translation process namely; codons definition, ribosomal frameshifting, and polyproteins production. The result of each step is related to the consequence of a rule. In fact, SWRL rules are a conjunction of predicates forming a head (consequence: the results of inferences) and a body (conditions to fulfill and get the inferences). SWRL rules are DL-safe rules which means: every argument representing an individual in the head has to be also in the body. Otherwise, the reasoning may lead to undecidability.

A. Codons definition

SWRL Rule 1 and 2 summarize the inferences that associate each codon with its first, second, and third nucleobase. As shown in Figure 3, we exploit in the first rule, the built-ins from the namespace *swrlb*, such as *mod* and *divide* to divide the genome into triplet and attribute each triplet to a codon following its rank. For example, a nucleobase with the rank 9 will be the third nucleobase of the codon having the rank 3. The codon sequences are induced through the second rule.

SWRL Rule 1

```
Nucleobase(?x) ^ Nucleobase(?y) ^ Nucleobase(?z) ^ has_next(?x, ?y) ^ has_next(?y, ?z) ^
has_Rank(?z, ?t) ^ swrlb:mod( 0 , ?t , 3 ) ^ swrlb:divide(?d, ?t , 3 ) ^ Codon(?c) ^
has_Rank(?c, ?d) => has_first_base(?c, ?x) ^ has_second_base(?c, ?y) ^ has_third_base(?c, ?z)
```

SWRL Rule 2

```
Codon(?c1) ^ has_third_base(?c1, ?x) ^ Codon(?c2) ^ has_first_base(?c2, ?y) ^ has_next(?x, ?y)
 => has_next(?c1, ?c2)
```

Fig. 3: Codons division and definition

B. Ribosomal frameshifting

In the context of SARS-CoV-2 translation, the ribosome slips back to translate the ORF1b if the pattern [XXU UUA AAC] is detected nearby a secondary structure. We introduce in Figure 4, how our model infers this knowledge. In the third rule, we identify first, the beginning of a stem-loop through the property has_beginning_base(?s, ?b). Then, we formalize, the presence of a slippery sequence [XXU UUA AAC] thanks to the conjunctions Codon(?x) $\wedge$...$\wedge$ Cytosine(?c). We employ built-ins from *swrlb* to specify that the

[3] https://www.w3.org/Submission/SWRL/

pattern is nearby the stem-loop. When these conditions are fulfilled, the inference labeled the first Uracil of the pattern with the data property frameshifting (?u1, true). In the fourth rule, the rank of this Uracil is considered as a starting rank to infer the new codons (CodonFS). Like the second rule, Rule 5 combines only codons generated after the frameshifting.

SWRL Rule 3

```
Stem_Loop(?s) ^ has_beginning_base(?s, ?b) ^ has_Rank(?b, ?r1) ^ Codon(?x) ^ Codon(?y) ^ Codon(?z) ^
has_next(?x, ?y) ^ has_next(?y, ?z) ^ has_third_base(?x, ?u1) ^ Uracil(?u1) ^
has_first_base(?y, ?u2) ^ Uracil(?u2) ^ has_second_base(?y, ?u3) ^ Uracil(?u3) ^
has_third_base(?y, ?a1) ^ Adenine(?a1) ^ has_first_base(?z, ?a2) ^ Adenine(?a2) ^
has_second_base(?z, ?a3) ^ Adenine(?a3) ^ has_third_base(?z, ?c) ^ Cytosine(?c) ^
has_Rank(?u1, ?r2) ^ swrlb:subtract(?fs, ?r1, ?r2) ^ swrlb:lessThanOrEqual(?fs, 15 )
 => frameshifting(?u1, true)
```

SWRL Rule 4

```
Nucleobase(?u) ^ frameshifting(?u, true) ^ has_Rank(?u, ?r0) ^ Nucleobase(?x) ^ Nucleobase(?y) ^
Nucleobase(?z) ^ has_next(?x, ?y) ^ has_next(?y, ?z) ^ has_Rank(?x, ?r1) ^ has_Rank(?z, ?r3) ^
swrlb:mod( 0 , ?r1, 3 ) ^ swrlb:greaterThanOrEqual(?r1, ?r0) ^ CodonFS(?c) ^ has_Rank(?c, ?r4) ^
swrlb:add(?s, ?r3, 1 ) ^ swrlb:divide(?d, ?s, 3 ) ^ swrlb:equal(?d, ?r4)
=> has_first_base(?c, ?x) ^ has_second_base(?c, ?y) ^ has_third_base(?c, ?z)
```

SWRL Rule 5

```
CodonFS(?c1) ^ has_third_base(?c1, ?x) ^ CodonFS(?c2) ^ has_first_base(?c2, ?y) ^ has_next(?x, ?y)
=> has_next(?c1, ?c2)
```

Fig. 4: Ribosomal frameshifting

C. Post frameshifting codons

The sequence of initial codons (Codon) and the sequence of the new codons (CodonFS) have to be linked which forms the overlapping between ORF1a and ORF1b. To produce this association between codons, we include the rule presented in Figure 5 in **OntoRepliCov**. After the frameshifting, the pattern [XXU UUA AAC] will become [UUU AAA CXX]. The connection will be generated between the Codon (AAC) and the CodonFS (CXX).

SWRL Rule 6

```
Codon(?c1) ^ has_third_base(?c1, ?u) ^ CodonFS(?c2) ^ has_first_base(?c2, ?u) ^
frameshifting(?u, true) ^ has_Rank(?c1, ?r1) ^ has_Rank(?c2, ?r2) ^
swrlb:add(?a1, ?r1, 2 ) ^ swrlb:add(?a2, ?r2, 2 ) ^ Codon(?x) ^ CodonFS(?y) ^ has_Rank(?x, ?a1) ^
has_Rank(?y, ?a2) => has_next(?x, ?y)
```

Fig. 5: Codons sequence after frameshifting

D. Polyprotein production

At the end of the translation process, polyproteins are produced. Formally, each polyprotein is a sequence of amino acids delimited by the amino acids Methionine and Stop. The rules presented in Figure 6 describe the inference of the polyproteins *pp1a* and *pp1ab* presented respectively by the object property sequence_polyprotein_pp1a and sequence_polyprotein_pp1a.

SWRL Rule 7

```
Codon(?x) ^ is_start(?x, true) ^ Codon(?y) ^ Stop(?y) => sequence_polyprotein_pp1a(?x, ?y)
```

SWRL Rule 8

```
Codon(?x) ^ is_start(?x, true) ^ CodonFS(?y) ^ Stop(?y) => sequence_polyprotein_pp1ab(?x, ?y)
```

Fig. 6: Polyproteins definition

3.3.2. Axiomatic reasoning

As explained before, a polyprotein is a sequence of amino acids. Hence, each codon has to be inferred as an amino acid. For this purpose, we use axiomatic reasoning by defining an axiom for each amino acid. Consequently, twenty-two axioms are defined. We present in Table 2 an example of two axioms formalized in DL to define the Methionine/Start (AUG) and the Asparagine (AAC/AAU).

Table 2: Axioms to infer amino acids from codons

Example of amino acids definition
Start≡Global_codon⊓ (∃has_first_base.Adenine)⊓(∃has_second_base.Uracil)⊓(∃has_third_base.Guanine)
Asparagine ≡ Global_codon ⊓ (∃has_first_base.Adenine) ⊓ (∃has_second_base.Adenine) ⊓ (∃has_third_base.(Cytosine⊔Uracil))

4. Use Case

Analyzing the amino acids sequences of polyproteins is one of the tasks that biologists tend to achieve. One of the **OntoRepliCov** purpose is to infer the polyproteins structure (amino acids sequence) of SARS-CoV-2. We present here examples of the reasoning results of **OntoRepliCov** after the data enrichment of a genome sequence.

4.1. Reasoning pipeline

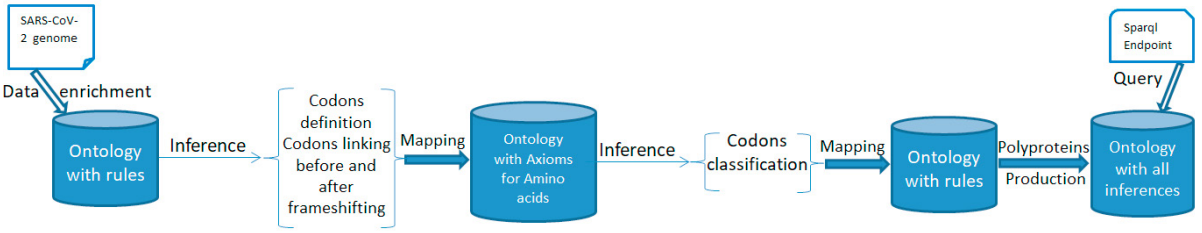


Fig. 7: Pipeline for reasoning mechanisms

The pipeline illustrated in Figure 7 shows how we managed to handle both axiomatic and rule reasoning. These reasoning mechanisms have to be separated to maintain the system's decidability. First, we populate our knowledge base with a genome sequence from the NCBI[4]. We take into account the fragment representing ORF1a and ORF1b. Then, we add the set of SWRL rules (from 1 to 6) to our ontology with the prospect to infer new knowledge about codons sequence, before and after the ribosomal frameshifting. However, in this step, we ignore all the complex axioms. Once the inferences are deduced, a mapping is established to add all individuals (asserted and inferred) to the ontology designed with axioms (the set of rules are ignored). This mapping leads the axiomatic reasoning to infer the classification of the amino acids. Another mapping is then established to populate the same model with this classification. At the end, the rules related to polyproteins definition (SWRL rules 7 and 8) are considered, and axioms are ignored. The polyproteins sequence can be finally fetched with SPARQL queries (Protocol and RDF Query Language). The pipeline was developed with Java. Technical parameters motivated us to opt for Openllet [5] which is an OWL 2 DL reasoner. We combined it with both Jena[6] and OWL-API[7] libraries.

[4] National Center for Biotechnology Information Search database: `https://www.ncbi.nlm.nih.gov/`
[5] `https://github.com/Galigator/openllet`
[6] `https://jena.apache.org/`
[7] `http://owlcs.github.io/owlapi/`

Table 3: Assertion and reasoning results

	has_next	has_first_base	has_second_base	has_third_base	Classification
a35	a36	/	/	/	Adenine
a36	c31	/	/	/	Adenine
c31	g34	/	/	/	Cytosine
g34	g35	/	/	/	Guanine
codon48	**codonFS49/codon49**	**a35**	**a36**	**c31**	**Asparagine**
codonFS49	**codonFS50**	**c31**	**g34**	**g35**	**Arginine**

4.2. Assertions and inferences

We illustrate in table 3 an example of a sequence fragment where the overlapping occurs between the ORF1 and ORF2. All inferences are presented in bold. The axiomatic inferences concern only the classification of amino acids. Otherwise, the reasoning is via SWRL rules.

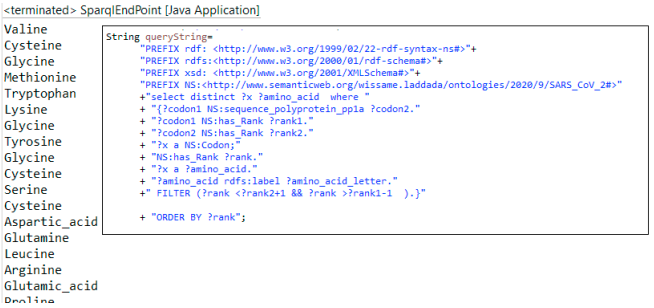
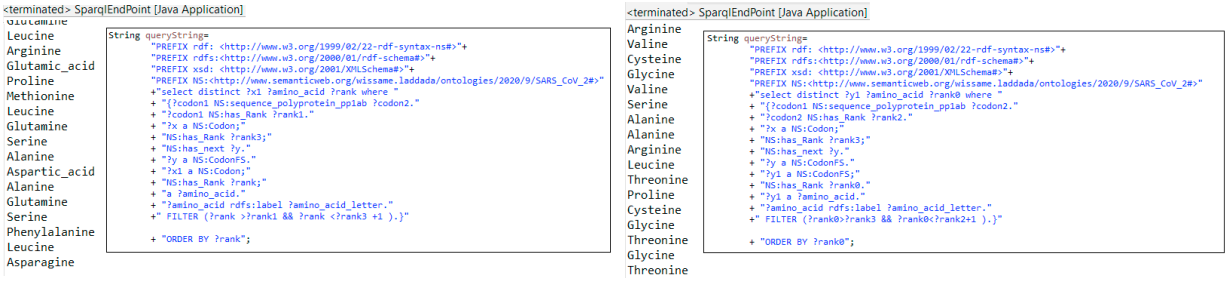


Fig. 8: Query to extract amino acids sequence of *pp1a*.



(a) With the slippery sequence

(b) After the slippery sequence

Fig. 9: Queries to extract amino acids sequence of *pp1ab*

4.3. SPARQL query for *pp1a* and *pp1ab*

At the end of the translation process through **OntoRepliCov**, biologists would want to extract the amino acids sequence of the polyproteins *pp1a* and *pp1ab* for analyzing process. We suggest SPARQL queries tested through a SPARQL endpoint (Java application). In Figure 8, the result shows a fragment of the amino acids sequence of *pp1a*. For the sequence of *pp1ab* we split the query into two subqueries. The first one (Figure 9) gets the amino acids sequence including the slippery sequence. Regarding the second

subquery, the result gathers the next amino acids sequence of $pp1ab$ after the frameshifting (including the last nucleobase of the slippery sequence).

5. Conclusion and future work

OntoRepliCov is designed to describe the SARS-CoV-2 replication machinery and to infer the virus's elements produced at every stage of the replication process. Inferring this knowledge and interactions between elements would be relevant to understand how the process can be disrupted. In addition to the classes taxonomy and the properties, the ontology is semantically enhanced by several rules and axioms allowing a high level of expensiveness and inferences. Our use case demonstrates the feasibility of the approach by deducting the polyproteins $pp1a$ and $pp1ab$, vital elements produced at the first stage by the translation process. However, because of the reasoning complexity, the quantitative parameters like time and percentage production can not be handled by **OntoRepliCov**. To tackle this issue, we exploit discrete event system specification (DEVS) modeling to design the dynamic behavior of the virus life cycle over time. This approach combination provides the integration of the polyprotein percentage production at the first stage which is 70% for the polyprotein $pp1a$ and 30% for the polyprotein $pp1ab$.

Thanks to the micro-level model, the ontology will be enhanced to integrate mutations caused by the discontinuous transcription which is an important stage of the viral replication. Hence, in addition to the eight SWRL rules and the twenty-two axioms, the reasoning process will be enriched with other rules to describe the discontinuous transcription.

References

[1] Adly, A.S., Adly, A.S., Adly, M.S., 2020. Approaches based on artificial intelligence and the internet of intelligent things to prevent the spread of covid-19: Scoping review. J Med Internet Res 22.
[2] Ashburner, M., et al., 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium 25, 25–29.
[3] Babcock, S., Beverley, J., Cowell, L., Smith, B., 2020. The infectious disease ontology in the age of covid-19.
[4] Best, K., Perelson, A.S., 2018. Mathematical modeling of within-host zika virus dynamics. Immunological reviews 285, 81–96.
[5] Dey, L., Chakraborty, S., Mukhopadhyay, A., 2020. Machine learning techniques for sequence-based prediction of viral-host interactions between sars-cov-2 and human proteins. Biomedical Journal , 438–450.
[6] Ekins, S., et al., 2020. Dj vu: Stimulating open drug discovery for sars-cov-2. Drug Discov Today 5.
[7] Fabreti, et al., 2019. Stochastic modeling and simulation of viral evolution. Bulletin of mathematical biology 81, 1031–1069.
[8] Garvin, M., et al., 2020. Potentially adaptive sars-cov-2 mutations discovered with novel spatiotemporal and explainable ai models. Genome Biology 21.
[9] Gordon, D.E., et al., 2020. A sars-cov-2-human protein-protein interaction map reveals drug targets and potential drug-repurposing. bioRxiv .
[10] Handel, A., Liao, L.E., Beauchemin, C.A., 2018. Progress and trends in mathematical modelling of influenza a virus infections. Current Opinion in Systems Biology 12, 30–36.
[11] Hattaf, K., Elaiw, A.M., Lashari, A.A., Yousfi, N., 2018. Mathematical modeling in virology by differential equations.
[12] Huffman, A., He, Y., 2020. Ogg-cov: Ontology representation and analysis of genes and genomes of coronaviruses. CEUR Workshop Proceedings 2807.
[13] Khalaf, R.A., et al., 2021. Cov2k: a knowledge base of sars-cov-2 variant impacts, in: Research Challenges in Information Science.
[14] Kouam, K.M., Mcheick, H., 2021. An ontological approach for early detection of suspected covid-19 among copd patients. Applied System Innovation 4.
[15] Li, Z., Teng, Z., Miao, H., 2017. Modeling and control for hiv/aids transmission in china based on data from 2004 to 2016. Computational and mathematical methods in medicine 1.
[16] Liu, Y., et al., 2021. Ontological modeling and analysis of experimentally or clinically verified drugs against coronavirus infection. Scientific Data 8.
[17] Madelain, V., et al., 2018. Ebola viral dynamics in nonhuman primates provides insights into virus immuno-pathogenesis and antiviral strategies. Nature communications 9, 1–11.
[18] Naqvi, A.A.T., et al., 2020. Insights into sars-cov-2 genome, structure, evolution, pathogenesis and therapies: Structural genomics approach. Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease , 165878.
[19] de Oliveira, et al., 2016. A review of kinetic modeling methodologies for complex processes. Oil & Gas Science and Technology–Revue dIFP Energies nouvelles 71, 45.
[20] Ozturk, T., et al., 2020. Automated detection of covid-19 cases using deep neural networks with x-ray images. Computers in Biology and Medicine 121.