Reasoning Challenges on Gene Variants Data

Asha Subramanian[a], Gopi Sumanth Bhaskar Boddeda[a], Manikanta Vikkurthi[a] and Ikrormi Rungsung[a]

[a]*Semantic Web India, Bangalore, Karnataka, India. Home Page: http://www.semanticwebindia.com/*

Abstract

The ability to perform logical reasoning is a human trait which when simulated using knowledge representation, reasoning using ontologies and semantic AI equip applications to generate useful insights. In the life sciences domain, ontologies are widely used to characterize knowledge related to complex definitions and relations between genes, diseases, proteins and other biomedical information to facilitate search, reuse and computations. In this paper, we present our efforts towards using one such homegrown knowledge framework - Sandhi GVA (Gene Variant Analysis) and share the analysis of the reasoning performance. This paper is submitted towards "Task-1" of the Challenge Description.

Keywords

Reasoning, Knowledge Frameworks, Gene Variant Analysis, Ontology

1. Introduction

Our genes encode a unique genome sequence characterizing our genetic disposition, appearance and functions in our body. We inherit this genome from our parents. All individuals carry some amount of genetic variations, however some of these variations may affect the critical functions of our body causing a disease or a genetic disorder. The sample from the patient such as blood, hair, tissue etc. is processed for DNA[1] extraction and converted into digitized data (Genome Sequencing Data) by Next Generation Sequencing technologies (NGS).[2] A series of industry standard software pipelines converts the "Sequenced data" into "Variants data", ie. list of variations in specific genes. These resulting variations (in millions) have to be prioritized to arrive at the top variants that are the potential causes for certain genetic disorders. We use a knowledge framework based solution to prioritize the gene variants using annotated semantic entities from multiple vocabularies in the bio-medical domain. Our core semantic framework – Sandhi Gene Variant Analysis (Sandhi GVA) – creates the foundation for capturing the gene variants and establishing reasoning rules to prioritize the variants linked to potential pathogenic conditions given a disease manifestation. For this paper, we focus on the reasoning challenge and the sequence of tasks to evaluate the framework for the specific classification task. The reasoner uses inferencing rules to classify the variants into a semantic class denoting

SemREC'21: Semantic Reasoning Evaluation Challenge, 20th International Semantic Web Conference (ISWC 2021), October 24–28, 2021

asha@semanticwebindia.com (A. Subramanian); gopi@semanticwebindia.com (G. S. B. Boddeda); manikanta@semanticwebindia.com (M. Vikkurthi); ikrormi@semanticwebindia.com (I. Rungsung)

[1]DNA: https://www.genome.gov/genetics-glossary/Deoxyribonucleic-Acid

[2]NGS: https://ghr.nlm.nih.gov/primer/testing/sequencing

Figure 1: Components for the Reasoning Task

its pathogenicity (illustrated in Figure 3). The reasoning challenge is the inability to prioritize large sized (> 2000 variant rows) variants data files.

Figure 1 depicts the datasets involved in this paper and the overall workflow to accomplish the reasoning task.

The core Sandhi GVA ontology or the T-Box is contained in *GVA.owl*. The "variants data" received from the NGS pipelines is processed for a series of filtering and prioritization routines and converted into a knowledge graph (A-Box), aligned with the Sandhi GVA ontology (T-Box) before subjecting it to the reasoner. This variants graph or the A-Box is contained in *variants.nt*. The python module *reasoningapi.py* uses Owlready2[3] libraries and the Pellet reasoner[4] [1] to derive new facts based on reasoning rules to assign a "Pathogenic" classification to each variant tuple in the *variants.nt*.

Section 2 explains the Sandhi GVA Ontology, the A-Box generation and the details pertaining to the reasoning code execution. Section 3 summarises the results of the reasoner performance for the classification task, the challenges faced and the workarounds implemented to circumvent the challenges. Importance of this work in practical applications is discussed in Section 4. We conclude this paper with Section 5 by discussing the future interventions planned to enrich the reasoner and tackle the implementation challenges.

2. Dataset Description

The datasets and the programming modules used in describing the reasoning challenge is explained in this section. The components include

1. The Sandhi GVA Ontology that forms the core framework or the T-BOX (*GVA.owl*)
2. The knowledge graph generated from the raw gene variants files (*variants.nt*). The raw gene variant files received from the NGS pipelines go through a series of filtering and prioritizing routines including semantic annotations using bio-medical vocabularies to generate the variant N-Triples[5] file (*variants.nt*). This N-Triples file forms the A-Box and is in line with the T-Box definitions in *GVA.owl*.

[3]Owlready2: https://owlready2.readthedocs.io/en/latest/
[4]Pellet: https://github.com/stardog-union/pellet
[5]N-Triples: https://www.w3.org/TR/rdf-testcases/#ntriples

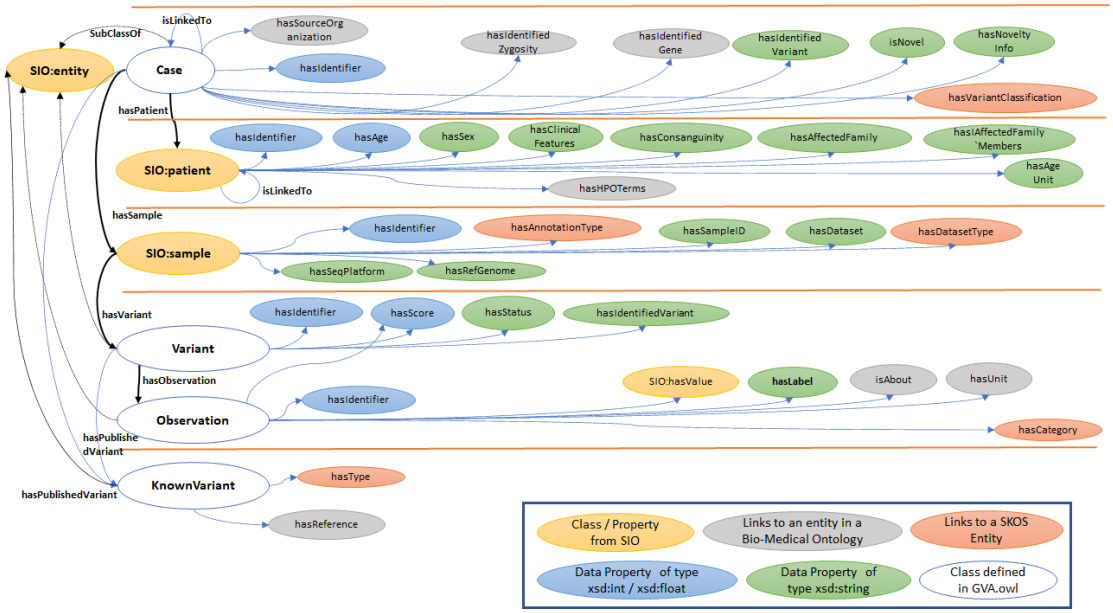

Figure 2: The Ontology Model for Sandhi GVA

3. The python programming module (reasoningapi.py) that executes the reasoning rules on the variants N-Triples file.

Each of these components is explained in detail in the following sections.

2.1. Sandhi GVA Ontology

Figure 2 illustrates the ontology model for Sandhi GVA. The ontology consists of 6 main classes namely "Case", "Patient", "Sample", "Variant", "Observation" and "KnownVariant". The "Case" class holds the summarised information regarding the prioritized variants detected for the DNA sample. This class also holds links to the respective patient metadata and the information regarding physical sample information processed for the case. The "Patient" and "Sample" classes have been extended from the universal Semanticscience integrated ontology (SIO)[6] to maintain consistency and standardization in knowledge representation using bio-medical terms. The "Patient" and "Sample" classes hold the metadata information related to patient clinical information and sample details such as type of sequencing, reference genome used etc. The "Variant" class is a place holder for each variant detected for the patient's DNA sample. A variant contains many observations; a subset of these observations participate in the decision to qualify the variant into a specific pathogenic classification. Each observation belongs to a particular semantic class (represented using the *isAbout* property) determined by the prioritization routine that generated the variant graph (*variants.nt*). These semantic classes have been defined using the SKOS[7] vocabulary. The SKOS vocabulary has been used to

[6]SIO: https://bioportal.bioontology.org/ontologies/SIO
[7]SKOS: https://www.w3.org/TR/swbp-skos-core-spec/

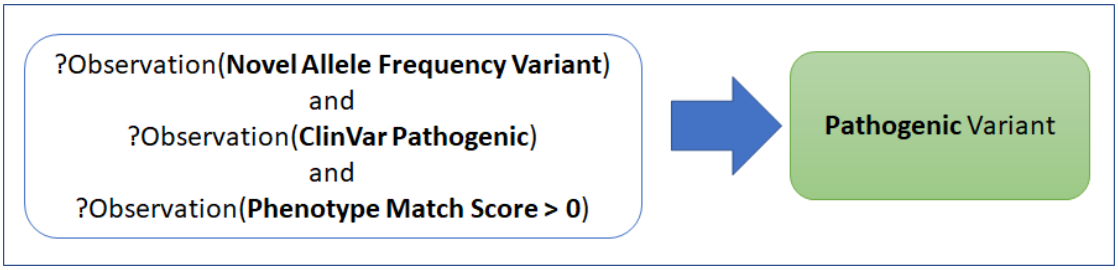

Figure 3: Sample Reasoning Rule modelled in the Reasoner

represent all hierarchical information and taxonomy pertaining to "Observation" categories, classifications and "Sample" types. The "Observation" class holds properties to link each variant observation to its semantic entity using the *isAbout* property. The *isAbout* property of the SIO ontology is used to link entities from other biomedical ontologies such as OMIM,[8] HGNC[9] and entities defined using the SKOS vocabulary. Finally the "KnownVariant" class links the identified variants to well known variant databases such as ClinVar[10] and HGMD.[11]

2.2. A-Box Generation

The A-Box is contained in two N-Triples files namely - *GVA.nt* and *variants.nt*. *GVA.nt* contains all the taxonomy definitions for the variant classification definitions using the SKOS vocabulary along with the base *GVA.owl* that represents the T-BOX for the Sandhi GVA ontology.

The variants graph is generated using an extensive filtering and prioritization routine that converts the raw gene variants files received from the NGS pipelines into a knowledge graph using the T-BOX definitions in *GVA.owl*. A collection of pre-generated *variants.nt* files have been included with this paper for simulating the reasoning challenge.

2.3. Reasoning Code Execution

This is a python module (*reasoningapi.py*) which uses the *Owlready2* libraries and *Pellet* reasoner to execute the reasoning statements that finally classify each variant into one of the 5 "Pathogenic" classifications namely - 'Pathogenic', 'Likely Pathogenic', 'Benign', 'Likely Benign', 'Uncertain Significance'.

Figure 3 shows a typical reasoning rule and the subsequent classification of the variant.

3. Reasoner Analysis

This section explains the content of the Git repository to simulate the reasoner challenges using the ontology model and the A-Box. The Git repository can be accessed at https://github.com/

[8]OMIM: https://bioportal.bioontology.org/ontologies/OMIM
[9]HGNC: https://download.bio2rdf.org/files/release/3/hgnc/hgnc.html
[10]ClinVar: https://www.ncbi.nlm.nih.gov/clinvar/intro/
[11]HGMD: http://www.hgmd.cf.ac.uk/ac/index.php

Table 1
Summarizing the Reasoner Performance

File Name	Number of Variants	Reasoning Time	Status of Reasoner Execution
Variant1.nt	3902 (100 MB)	NA	Failure, Out of Memory Error
Variant2.nt	1265 (35 MB)	135.7 secs	Success, Lengthy Execution Time
Variant3.nt	1931 (50 MB)	885.6 secs	Success, Lengthy Execution Time
Variant4.nt	4114 (120 MB)	NA	Failure, Out of Memory Error
Variant5.nt	3891 (100 MB)	NA	Failure, Out of Memory Error

SWIUser1/SemanticReasoner and contains the components explained in Section 2 namely *GVA.owl*, *GVA.nt*, *variants.nt* and *reasoningapi.py*.

3.1. Reasoning Performance

Table 1 summarises the performance of the reasoner on the various samples. For fair comparative analysis, the heap size was set to 8000 MB (owlready2.reasoning.JAVA_MEMORY = 8000) for all the variant files.

3.2. Reasoning Challenge

There are two challenges faced with the reasoner - 1) Out of Memory error and 2) Time taken for reasoner execution.

- Out of Memory error:
  The reasoner fails to execute for variant graphs constructed from samples containing more than 2000 variants approximately, with the error message "Exception in thread 'main' java.lang.OutOfMemoryError: GC overhead limit exceeded". While one direct solution is to increase the heap size for the reasoner, this cannot be a permanent solution as the configuration of heapsize is limited to the compute's memory resources.
- Time taken for reasoner execution:
  The reasoner takes inordinate amount of time to complete the execution and generate the derived facts, in some cases the reasoning time is in excess of 15 minutes.

3.3. Workarounds Implemented for the Reasoner

The following interventions were implemented to circumvent the failure of the reasoner to process the entire set of tuples from the variants graph and/or substantially reduce the time taken by the reasoner to complete execution:

- Chunking the Variants Graphs in Smaller Blocks:
  In this intervention, the variants graph was generated in small batches of 100 variants in an iteration loop and passed to the pellet reasoner. However, with each iteration, the reasoning time increased exponentially. This was due to the accumulation of instances from the previous iterations. Alternate tactics to reset the variants graph was attempted

at the end of each iteration as suggested by the Owlready2 support team, however this approach could not address the challenges with the reasoner. This technique however, resolved the Out of Memory error for the reasoner execution.

- Evaluation of the Map Reduce Technique:
  In this approach, Apache Spark was used in standalone mode configured with a Master and two Workers with 2 GB RAM each. The variants data was divided into two batches and passed to each worker. While this approach seemed to work successfully for variant files of smaller sizes, the worker nodes need to be assigned with more memory resources as the size of the variants graph increase. The reasoner execution time is directly proportional to the memory resources allocated to each worker.
- Limiting the size of the Variants Graph:
  In this workaround, the number of variants generated by the filtering and prioritizing routines were restricted by limiting the search for potential gene variants in certain sections of the genome sequence that are highly probable to host pathogenic variants. This entailed repeated analysis of the same sample analysing each section of the genome separately, however this intervention seems the best solution given the existing limitations of the reasoner. This approach currently, successfully addresses both the "Out of Memory" and "Reasoning Execution Time" challenges with the reasoner.

4. Potential Applications

An estimated 10,00,000+ infants are born in India every year with common genetic disorders such as congenital malformations, inherited blood disorders and other growth disorders every year with a prevalence rate of 1 out of every 20 newborns. If both the partners are carriers of the genetic disorder, the risk of an affected child is as high as 25% which is further aggravated in case of consanguineous (marriage between individuals who are closely related) couples [2] [3] [4].

With the emergence of NGS technology and its cost dropping rapidly, medical genetic testing in India is accelerating towards Whole Exome and Whole Genome sequencing [5] [6] [7]. In this scenario, technology intervention for comprehensive genomic data analysis, knowledge-based interventions in variant analysis and regulation for standard genetic variants test reports are crucial need of the hour. India being a very heterogeneous population and second most populated country with endogamous practices within communities or consanguineous union, harbors many genetic diseases. Some genetic diseases common to India are Beta Thalassemia, Spinal Muscular Atrophy, Duchenne Muscular Dystrophy, Hemophilia, Achondroplasia, Huntington disease, Lysosomal storage disorders are reported from all parts of the country [8]. More recently, drug metabolism defect in people of some communities practicing endogamy has been linked to the harmful variant of specific CYP2C9 gene [9]. Such studies play an important role in transitioning towards personalized medicine.

However, nationwide prevalence of such genetic disorders can only be established with concrete evidence in the form of actionable data. While many open-source and commercial tools have been developed to address the gene variant filtering, prioritization, analysis and reporting, there are limitations. Open-source tools suffer from flexibility in ease of use and

commercial tools, especially the ones developed outside India heavily rely on large cohorts of Caucasian population data to determine potential variants. Commercial tools in India hold the variant analysis information in private in-house databases.

Our work [10] has the potential of providing a cost effective, comprehensive and scalable platform for addressing the bio-informatics component of the gene variant prioritization and analysis process.

To the best of our knowledge, there have been limited efforts to build a semantic knowledge framework enabled platform to link identified potential variants for multiple known genetic conditions for the Indian population that can be intelligently applied for accurate identification and prioritization of gene variants to analyze rare mendelian genetic disorders in humans.

5. Conclusions and Future Work

The current workarounds for the implementation of the reasoner include modifications in our filtering and prioritization routines to limit the number of variants generated in the variants graph before sending it to reasoner. This is managed by limiting the variants to certain sections of the genome such as “Exonic” and “Intronic” to limit the search for potential variants and complete the analysis in multiple runs. Future interventions will explore Spark based OWL2 Reasoning [11] and incorporating algorithms for Distributed Reasoning of RDF data [12].

Acknowledgments

This project has been selected by the Atal Incubation Centre, Centre for Cellular and Molecular Biology (AIC-CCMB) as part of the TIDE 2.0 program to develop a prototype from idea under the MEITY Startup Hub Scheme. We sincerely thank CCMB for their support in providing us the necessary support to test our proposed knowledge framework and its applications.

References

[1] E. Sirin, B. Parsia, Pellet: An owl dl reasoner, in: Proc. of the 2004 Description Logic Workshop (DL 2004), 2004, pp. 212–213.

[2] S. K. Pemmasani, R. Raman, R. Mohapatra, M. Vidyasagar, A. Acharya, A review on the challenges in indian genomics research for variant identification and interpretation, Frontiers in Genetics 11 (2020).

[3] B. I. Article, Inherited diseases spreading its web in india, https://www.biospectrumindia.com/features/69/8967/inherited-diseases-spreading-its-web-in-india.html, 2017.

[4] K. Singh, S. Bijarnia-Mahay, V. L. Ramprasad, R. D. Puri, S. Nair, S. Sharda, R. Saxena, S. Kohli, S. Kulshreshtha, I. Ganguli, et al., Ngs-based expanded carrier screening for genetic disorders in north indian population reveals unexpected results–a pilot study, BMC medical genetics 21 (2020) 1–15.

[5] A. Uttarilli, H. Shah, G. S. Bhavani, P. Upadhyai, A. Shukla, K. M. Girisha, Phenotyping and genotyping of skeletal dysplasias: evolution of a center and a decade of experience in india, Bone 120 (2019) 204–211.

[6] R. D. Puri, M. Tuteja, I. Verma, Genetic approach to diagnosis of intellectual disability, The Indian Journal of Pediatrics 83 (2016) 1141–1149.

[7] B. Singh, K. Mandal, M. Lallar, D. L. Narayanan, S. Mishra, P. S. Gambhir, S. R. Phadke, Next generation sequencing in diagnosis of mlpa negative cases presenting as duchenne/becker muscular dystrophies, Indian J. Pediatr 85 (2018) 309–310.

[8] S. Aggarwal, S. R. Phadke, Medical genetics and genomic medicine in india: current status and opportunities ahead, Molecular genetics & genomic medicine 3 (2015) 160.

[9] H. Prasad Nichenametla, DHNS, Ccmb finds drug metabolism defect in people of some communities practising endogamy, https://www.deccanherald.com/science-and-environment/ccmb-finds-drug-metabolism-defect-in-people-of-some-communities-practising-endogamy-945777.html, 2021.

[10] A. Subramanian, A. D. Bhowmik, G. Pattnayak, M. Vikkurthi, A. K S, Analysis and prioritisation of potential gene variants powered by semantic intelligence, in: 8th ACM IKDD CODS and 26th COMAD, CODS COMAD 2021, Association for Computing Machinery, New York, NY, USA, 2021, p. 390–394. URL: https://doi.org/10.1145/3430984.3430990. doi:10.1145/3430984.3430990.

[11] Y. Liu, P. McBrien, Spowl: spark-based owl 2 reasoning materialisation, in: Proceedings of the 4th ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond, 2017, pp. 1–10.

[12] S. Sakr, M. Wylot, R. Mutharaju, D. Le Phuoc, I. Fundulaki, Distributed Reasoning of RDF Data, Springer International Publishing, Cham, 2018, pp. 109–126. URL: https://doi.org/10.1007/978-3-319-73515-3_6. doi:10.1007/978-3-319-73515-3_6.