Deep Symbolic Learning Architecture for Variant Calling in NSG

Ángel Canal-Alonso[1], Pedro Jiménez[1] and Noelia Egido[1], Javier Prieto[1], Juan Manuel Corchado[1]

[1] Departamento de Bioinformática y Biología Computacional, AIR Institute, Carbajosa de la Sagrada, España

E-mail: acanal@air-institute.com

Resumen

In the era of genomics, efficient and accurate analysis of genomic sequences is essential. Next-generation sequencing (NGS) technology has revolutionised the field of genomics by providing a massive volume of data on an unprecedented scale. One of the critical steps in the analysis of this data is variant calling, where genetic variations are identified from DNA sequences. In this context, we have explored the use of Deep Symbolic Learning (DSL) as an innovative computational approach that combines deep learning with symbolic representations. In this article, we discuss the principles of DSL and its applicability in genomics. We examine the advantages and challenges of its use in the context of variant calling and highlight the importance of meticulous validation. To ensure the quality of the results, it is essential to adopt appropriate validation techniques and specific software tools. We provide a detailed overview of these techniques and tools, with the aim of establishing clear standards for the implementation and validation of DSL algorithms in genomic pipelines. This research highlights the potential of the DSL to improve the accuracy of variant discovery, offering promising prospects for the genomics of the future.

Palabras Clave: Next-Generation sequencing, Validación, Deep Symbolic Learning

Introducción

The vast universe of the human genome has been a subject of fascination and study since the conception of molecular biology. With the advent of next-generation sequencing (NGS) technology, an unprecedented window into the detailed study of our genetic information has opened, allowing us to unravel mysteries related to disease, evolution and other fundamental aspects of biology. NGS has driven an exponential growth in the generation of genomic data, making the analysis of this data a complex but crucial task.

Within this analysis, one of the most relevant phases is variant calling, or variant detection. This stage consists of identifying the differences between a genomic sample and a reference, thus allowing mutations, polymorphisms and other genetic variants to be recognised. These variants are essential for understanding genetic diversity, predisposition to disease and, in many cases, response to treatment. Accurate and reliable variant identification is therefore essential to translate the vast amount of data generated by NGS into biologically relevant and applicable information.

However, the task is not simple. Genetic variations may be subtle or in regions of the genome that are difficult to sequence. In addition, intrinsic errors in sequencing techniques, as well as the inherent complexity of molecular biology, present challenges in accurately detecting variants. This is where advanced computational techniques, such as Deep Symbolic Learning (DSL), present themselves as promising tools, capable of integrating and learning from large volumes of genomic data, and offering accurate and reliable results in the task of variant calling.

Throughout this article, we will explore the potential of DSL in the context of variant calling, outlining its integration into a genomic data analysis pipeline and discussing the validation techniques needed to ensure the quality of its results.

Next-generation sequencing technology, commonly known as NGS (Next-Generation Sequencing), represents a revolution in the way genetic information is analysed and understood. Prior to the advent of NGS, DNA sequencing was mainly performed using the Sanger technique, a costly and relatively slow process that allowed small DNA fragments to be sequenced individually. While this technique was pioneering and fundamental for flagship projects such as the sequencing of the human genome, the growing need to analyse multiple samples and large genomic regions required more efficient technology.

NGS arose in response to this need. Unlike Sanger sequencing, NGS allows millions of DNA fragments to be sequenced simultaneously, in a single experiment. This "high-throughput" capability has drastically reduced sequencing costs and times, allowing genomics to become more accessible and huge volumes of data to be generated in relatively short timescales.

The general NGS process begins with the preparation of a library of DNA fragments to be sequenced. These fragments are attached to a surface and amplified to generate clonal DNA colonies, each corresponding to a single starting DNA molecule. Sequencing is then performed cycle after cycle, incorporating fluorescently labelled nucleotides. Upon incorporation, each nucleotide emits a specific light signal, which is detected and translated to determine the sequence of the fragment.

There are several platforms and methodologies within the NGS spectrum, such as Illumina, Roche's 454 and Ion Torrent, each with its own particularities, advantages and limitations. However, they all share the basic principle of sequencing multiple fragments in parallel, which gives NGS its power and versatility.

The advent of NGS has driven significant advances in areas such as personalised genomics, metagenomics, evolutionary genomics, among others. However, the flood of data it produces has posed computational and analytical challenges, making the integration of advanced tools such as Deep Symbolic Learning essential to extract the maximum value from the genetic information obtained.

Deep Symbolic Learning (DSL) has positioned itself as an innovative approach in the world of machine learning and artificial intelligence, merging two traditionally distinct paradigms: deep learning and symbolic learning.

Deep learning, mainly represented by deep neural networks, is known for its ability to handle large amounts of data, learn features automatically and perform prediction tasks with high accuracy. However, their models are often seen as "black boxes", because the interpretation of their internal processes can be complex.

On the other hand, symbolic learning, which has roots in logic and knowledge-based artificial intelligence, focuses on the manipulation and reasoning of explicit symbols and rules. This approach offers transparency and explainability, but can be limited in terms of adaptability and generalisability to large volumes of data or complex tasks.

DSL combines the best of both worlds. Through this fusion, it is possible to train models that are not only powerful and accurate, but also interpretable and based on rules and symbolic structures. This means that the DSL can leverage the ability of deep learning to automatically learn from data, while retaining a logical and symbolic structure that facilitates interpretation and decision making based on clear rules.

In the context of genomics, this combination is particularly valuable. Genomic data is inherently complex, with hierarchical structures, non-linear relationships and subtle patterns that can be crucial for tasks such as variant calling. The DSL, being able to effectively model both explicit and implicit patterns, positions itself as a promising tool for genomics.

For example, while a deep learning model could identify complex patterns in genomic sequences that indicate the presence of a variant, the symbolic component could represent known biological rules, such as the functional implications of a mutation in a particular region of the genome. This combination allows the analysis process to be both adaptive and based on previously established knowledge, thus optimising the accuracy and interpretability of the results..

Given the growing need for robust and explainable computational methods in genomic analysis, the DSL presents itself as a potentially revolutionary solution, offering a balance between analytical power and transparency in the decision-making process..

DSL Fundamentals

Deep Symbolic Learning (DSL) arose in response to a palpable need in the field of artificial intelligence: to combine the efficiency and generalisability of deep learning with the explainability and structure of symbolic learning.

Deep learning is based on neural networks with multiple layers, known as deep neural networks. These networks are capable of modelling complex, non-linear relationships between inputs and outputs, and have been noted, especially in recent years, for their performance in image processing,

natural language and other tasks. Their strength lies in their ability to learn features and data representations automatically and hierarchically, i.e. without requiring manual feature engineering. However, the interpretation of what these networks "learn" can be opaque, giving rise to the term "black box".

Symbolic learning, on the other hand, is a branch of artificial intelligence that uses symbols to represent knowledge. Unlike deep learning, which is usually inductive (learning from data), symbolic learning is mainly deductive (learning from pre-established rules and facts). In this paradigm, knowledge is represented by rules, facts and symbols, allowing for logical reasoning and easy interpretation. Symbolic learning has been fundamental in areas where explainability is essential, such as in expert systems and rule-based decision making.

DSL seeks to integrate these two approaches to take advantage of their complementary strengths. In a typical DSL system, the representations learned by deep neural networks are translated or transformed into symbolic structures. These structures can be rules, decision trees, or any other representation that allows for clear interpretation. Conversely, symbolic knowledge can also be incorporated into deep learning, guiding or constraining the neural network's learning process.

The essence of the DSL lies in its ability to learn end-to-end, i.e. from raw data to rules and symbolic representations, without losing the power of deep learning or the clarity of symbolic learning. This integration is achieved through various mechanisms, such as knowledge-based regularisation, conversion of neural networks into decision trees, among others.

In summary, the DSL represents a promising fusion of two worlds, allowing the inherent complexity of real data to be modelled, while providing clear, rule-based interpretations, facilitating decision making and reliability in a variety of applications.

Advantages and disadvantages of the DSL in the context of genomics

Genomics is a discipline that, by its nature, generates massive volumes of data. This data is intricate and has complex structures, subtle patterns and multi-level relationships. When approaching genomics through the lens of Deep Symbolic Learning (DSL), a number of notable advantages and disadvantages emerge:

Advantages:

1. Modelling Complex Relationships: Genomic data often contains non-linear, hierarchical and multi-factorial relationships. The deep learning component in the DSL allows capturing and modelling these intricate relationships with high accuracy..

2. Interpretability: One of the most common criticisms of deep learning models in genomics is their lack of explainability. With DSL, symbolic reasoning is integrated, providing clear interpretations in terms of rules or symbolic structures, which is crucial for understanding biological or genetic implications..

3. Incorporation of Prior Knowledge: Genomics has decades of research and accumulated knowledge. DSL allows the integration of this knowledge in the form of rules or facts, which can guide or improve the learning and prediction process..

4. Flexibility: DSL is adaptable. It can be tuned to prioritise accuracy (deep learning) or explainability (symbolic learning) according to the needs of the genomic analysis in question..

Disadvantages:

1. Computational complexity: Merging deep learning with symbolic learning can increase computational complexity, requiring more resources and time, especially with large genomic datasets..

2. Integration challenges: It is not always easy to combine symbolic knowledge with deep models. This can lead to situations where the model does not converge or fails to adequately represent prior knowledge..

3. Overfitting to Rules: When integrating rules and known facts, there is a risk that the model will overfit to these rules and lose the ability to generalise to new data or situations not covered by prior knowledge..

4. Dual Expertise Need: Implementing DSL in genomics requires a deep understanding of both machine learning and genetics and molecular biology. This can make the barrier to entry higher compared to using more conventional techniques..

In conclusion, while the DSL offers significant potential to address the challenges of genomic analysis, it also presents challenges inherent to its hybrid nature. As with any tool, its effectiveness will largely depend on proper implementation and adaptation to the specific problem at hand..

Application of the DSL in the variant calling phase in NGS

Variant calling is an essential process in genomic data analysis, which identifies variants (such as SNPs and indels) from sequencing data. Given the crucial nature of this task, accurate, robust and explainable tools are imperative. Deep Symbolic Learning (DSL) offers a promising approach in this area, and we detail its application in the variant calling phase below.

1. Modelado de Características Genómicas:

Genomic sequences have a variety of features that can be crucial for identifying variants. While deep learning in DSL can automatically extract features from reads and their context, the symbolic component can represent prior biological knowledge, such as conserved regions, splicing sites or mutation-prone areas..

2. Incorporation of Prior Knowledge:

There are rules and facts in genetics that are widely known. For example, certain variants in specific regions may be pathogenic or have a certain functional effect. DSL allows the integration of this knowledge into the variant calling process, prioritising or assigning reliability to variants based on previously established biological facts.

3. Probabilistic Reasoning and Variant Determination:

The variant calling process is not always binary. Sometimes, the evidence may be ambiguous or insufficient to make a clear determination. DSL, by combining symbolic reasoning with neural networks, can provide a probability-based decision framework, weighing the evidence in the data and prior knowledge..

4. Call Evaluation and Validation:

Once variant calling is done, it is vital to validate the accuracy and reliability of the calls. With DSL, this validation can be based not only on performance metrics, but also on the interpretation and consistency of the derived rules and symbolic representations..

5. Explainability and Reporting:

Finally, once variants have been identified, researchers or health professionals may require clear explanations of how and why a specific variant was determined. The symbolic component of the DSL facilitates this task by providing rule- or fact-based justifications that are easily understandable..

In summary, the application of DSL to variant calling in NGS promises to improve not only the accuracy and robustness of the process, but also its interpretability and reliability. By integrating advanced machine learning with logical and symbolic reasoning, the door is open to more informed and reliable genomic analysis..

Validations techniques

Traditional validation techniques

Validation is a crucial step in the variant calling process, as it ensures that the variants identified are authentic and not artefacts or errors of the sequencing or analysis process. In the context of variant calling, there are several traditional validation methods that have been used extensively:

1. Replication validation: This involves sequencing the same sample several times and comparing the results. If a variant is consistently identified in independent replicates, it is likely to be genuine. This technique is straightforward but can be costly in terms of time and resources..

2. Comparison with Reference Databases: The set of identified variants is compared with known variant databases, such as dbSNP or gnomAD. If a variant has been previously reported and validated, there is a high probability that it is real. However, this does not guarantee the identification of new or rare variants..

3. Homozygosity analysis: Variants occurring in regions of homozygosity (regions of the genome where an individual has two identical copies) may be easier to validate because any heterozygous variants in these regions are likely to be artefacts..

4. Sanger techniques: Sanger sequencing is considered the "gold standard" for variant validation. It consists of resequencing a specific region using the Sanger methodology and comparing the results with those obtained by NGS. If both techniques identify the same variant, it is considered validated..

5. Quality and Coverage Analysis: Assess base quality and coverage at positions where variants were detected. Variants identified in regions with poor quality or coverage should be treated with caution, as they have a higher probability of being artefacts..

6. Bioinformatics tools: There are several tools and software specifically designed for variant validation, such as GATK's VariantFiltration, which apply a series of filters and criteria to discern between real variants and artefacts..

These traditional methods have been mainstays in the validation of variant calling. However, with the incorporation of advanced techniques such as Deep Symbolic Learning, new validation strategies may be required or can be developed, combining the robustness of traditional methods with the advantages of machine learning and symbolic reasoning..

Particularities of validation when using DSL

By incorporating Deep Symbolic Learning (DSL) into the variant calling process, new dimensions and challenges in validation emerge. Some of the particularities and special considerations when validating variants identified through DSL are described below.:

1. Validation of Symbolic Rules: DSL, when generating symbolic rules or structures, requires validation of these representations. It is essential to ensure that the rules generated make biological sense and are not the product of overfitting or artefacts of the model..

2. Interaction between Deep and Symbolic Components: In DSL, deep and symbolic learning components interact. When validating, it is crucial to understand how these interactions affect variant identification and whether they introduce biases or errors..

3. Sensitivity to Training Data: Like any machine learning model, DSL-based models are sensitive to the data they are trained on. Cross-validation and testing must be performed on independent data sets to ensure model generalisability..

4. Explainability versus Accuracy: A trade-off may arise between the accuracy of the model and its explanatory power. In validation, it is essential to consider both aspects and determine whether one is being sacrificed at the expense of the other..

5. Iterative Validation: Since DSL combines machine learning techniques with symbolic reasoning, it can be beneficial to adopt an iterative approach to validation. For example, after a first round of variant calling, the symbolic rules can be refined and the model retrained for a second round, thus improving accuracy and reliability..

6. Ambiguity Assessment: Symbolic reasoning in DSL can introduce or resolve ambiguities in the variant calling process. During validation, it is crucial to identify and deal with these ambiguities in an appropriate way..

7. Use of Complementary Validation Metrics: In addition to traditional validation metrics, such as accuracy, recall and F1-score, it may be useful to introduce metrics that assess the quality and consistency of the generated symbolic representations..

In summary, while the DSL presents promising potential for variant calling, it also introduces validation peculiarities that must be approached with care. These considerations ensure that the results are not only accurate but also biologically meaningful and explainable..

Considerations on the validation dataset

By introducing advanced techniques such as Deep Symbolic Learning (DSL) into the variant calling process, the validation dataset becomes even more important. Here are some key considerations when selecting and working with validation datasets in this context:

1. Genomic diversity: It is essential that the validation dataset reflects the genomic diversity of the population under study. This ensures that the model can identify variants in a wide variety of genomic contexts and not just those present in the training set..

2. Data Set Size: While it is tempting to opt for larger datasets due to the perception that they provide more robust results, it is crucial that these data are of high quality. Sometimes, a smaller but meticulously curated dataset is preferable..

3. Independent Data Sets: To avoid overfitting and ensure model generalisation, it is vital to use validation datasets that are completely independent of the training set..

4. Known versus Unknown Variants: The validation set should contain both previously identified and validated variants (to assess the accuracy of the model) and new unknown variants (to assess the model's ability to identify previously unseen variants).

5. Biological Representativeness: In addition to variants, it is important that the validation dataset reflects other biological aspects of the genome, such as highly conserved regions, mutation-prone areas or regulatory regions..

6. Associated metadata: Whenever possible, it is useful to have access to metadata associated with the samples in the validation set. This may include information about the individual's clinical condition, phenotype, or any treatment received. This metadata can help interpret the results and provide additional context..

7. Known Errors and Artefacts: It is beneficial to be aware of errors and artefacts associated with the sequencing technologies used in the validation dataset. These may influence the interpretation of the results and should be considered when validating the identified variants..

8. Flexibility in Evaluation: Due to the particularities of the DSL, it may be necessary to adapt or develop new validation protocols. The validation dataset must be flexible enough to allow for these adaptive approaches..

Ultimately, the validation dataset is not only a tool for assessing the effectiveness of a model, but a key component that, if properly selected and used, can significantly improve the accuracy and reliability of the variant calling process in a DSL context..

Metrics and evaluation tools

Quantitative validation of variant calling is crucial to ensure the quality and reliability of the results. Evaluation metrics provide a way to measure the effectiveness of the variant calling process. Here, we describe some of the key metrics used in this context:

1. Accuracy: This metric assesses the proportion of correct predictions (both true and non-true variants) relative to all predictions made. It is a global metric that considers both true positives and true negatives..

$$\mathrm{P} = \frac{\text{Positivos Verdaderos} + \text{Negativos Verdaderos}}{\text{Total de Predicciones}}$$

2. Sensitivity (Recall or True Positive Rate): Measures the proportion of true variants that were correctly identified by the model relative to all true variants present in the data..

$$\text{Sensibilidad} = \frac{\text{Positivos Verdaderos}}{\text{Positivos Verdaderos} + \text{Falsos Negativos}}$$

3. Specificity (True Negative Rate): Evaluates the proportion of non-variants that were correctly identified in relation to all non-variants present..

$$\text{Especificidad} = \frac{\text{Negativos Verdaderos}}{\text{Negativos Verdaderos} + \text{Falsos Positivos}}$$

4. Positive Predictive Value (PPV): This metric indicates the proportion of identified variants that are true variants relative to all identified variants..

$$\text{VPP} = \frac{\text{Positivos Verdaderos}}{\text{Positivos Verdaderos} + \text{Falsos Positivos}}$$

5. Negative Predictive Value (NPV): Indicates the proportion of identified non-variants that are true non-variants relative to all identified non-variants..

$$\text{VPN} = \frac{\text{Negativos Verdaderos}}{\text{Negativos Verdaderos} + \text{Falsos Negativos}}$$

6. F1-Score: This is a metric that combines accuracy and sensitivity into a single figure. It is especially useful when classes are unbalanced..

$$\text{F1-Score} = 2 \times \frac{\text{Precisión} \times \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

These metrics provide a quantitative view of the quality of variant calling. However, it is important to note that there is no single "perfect" metric and often they must be considered together to get a complete picture of the effectiveness of the variant calling process, especially in the context of Deep Symbolic Learning..

Recommended software tools

As genomic analysis has advanced, various software tools have emerged designed to facilitate the assessment and validation of variant calls. These tools vary in their capabilities and applications, but many of them are widely recognized in the genomics community. Here, we present some of the most recommended ones for variant validation, particularly in the context of Deep Symbolic Learning-based models.:

1. GATK (Genome Analysis Toolkit): Developed by the Broad Institute, GATK is perhaps one of the most popular software packages for genome analysis. While it is widely known for its variant calling tools, GATK also provides tools like `VariantEval` for variant call assessment and validation..

2. VCFtools: This is a suite of programs for working with VCF (Variant Call Format) and BCF (Binary Call Format) files. It provides utilities for comparing variants, analyzing variant sites and populations, and evaluating call quality..

3. bcftools: Similar to VCFtools, bcftools is a collection of utilities for the analysis of genetic variants in the VCF/BCF format. It is especially useful for filtering and statistics..

4. BEDTools: Although not strictly limited to variant validation, BEDTools is a powerful tool for working with genomic data in various formats, including BED. It can be useful for operations such as intersecting and comparing variant sets..

5. RTG Tools: Real Time Genomics (RTG) provides tools that enable the rapid and accurate evaluation of variants against a reference set. Its `vcfeval` function is particularly useful for variant call assessment..

6. hap.py: It is a tool that compares genomic variant sets to find differences in calls and genotypes. It is based on the re-genotyping of a truth set and produces detailed concordance metrics..

7. DeepVariant: While it is a variant calling tool itself based on deep learning, DeepVariant offers integrated capabilities to assess call quality. Its image-based approach may be of interest when considering alternative evaluation methods in the context of DSL (Deep Symbolic Learning)..

When selecting tools for validation, it is crucial to consider the nature of the data, the specific project requirements, and the research team's expertise. The right combination of software and metrics can significantly improve the accuracy and reliability of variant calls in genomics analysis based on Deep Symbolic Learning.

References

Garcia-Retuerta D, Canal-Alonso A, Casado-Vara R, Rey AM, Panuccio G, Corchado JM. Bidirectional-Pass Algorithm for Interictal Event Detection. In Practical Applications of Computational Biology & Bioinformatics, 14th International Conference (PACBB 2020). PACBB 2020. Advances in Intelligent Systems and Computing, vol 1240. Springer, Cham. https://doi.org/10.1007/978-3-030-54568-0_20

Castillo Ossa LF, Chamoso P, Arango-López J, Pinto-Santos F, Isaza GA, Santa-Cruz-González C, Ceballos-Marquez A, Hernández G, Corchado JM. A Hybrid Model for COVID-19 Monitoring and Prediction. Electronics. 2021; 10(7):799.
https://doi.org/10.3390/electronics10070799

Intelligent Platform Based on Smart PPE for Safety in Workplaces. Márquez-Sánchez S, Campero-Jurado I, Herrera-Santos J, Rodríguez S, Corchado JM. Sensors (Basel). 2021 Jul 7;21(14):4652
https://doi.org/10.3390/s21144652

A. Canal-Alonso, R. Casado-Vara and J. Manuel Corchado, "An affordable implantable VNS for use in animal research," 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2020, pp. 1-4,
doi: 10.1109/ICECS49266.2020.9294958

An Agent-Based Clustering Approach for Gene Selection in Gene Expression Microarray. Ramos J, Castellanos-Garzón JA, González-Briones A, de Paz JF, Corchado JM. Interdiscip Sci. 2017 Mar;9(1):1-13
DOI 10.1007/s12539-017-0219-6

Acknowlegments

The present study has been funded by the AIR Genomics project (file number CCTT3/20/SA/0003) through the 2020 call for R&D Projects Oriented towards Excellence and Competitive Improvement of CCTT by the Institute of Business Competitiveness of Castilla y León and FEDER funds.