

A Model-driven Approach to Enhance Data Visualization through Domain Knowledge Integration

Andreia Almeida¹, Alberto Alves², Maribel Yasmina Santos^{1,*}, Ana León³ and João Moura Pires²

¹ALGORITMI Research Centre, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal

²NOVA University of Lisbon, School of Science and Technology, Lisboa, Portugal

³Research Center on Software Production Methods (PROS), Universitat Politècnica de València, Valencia, Spain

Abstract

Big Data is challenging analytical contexts, namely when aligning data and analytical requirements. While the capacity to collect and store new data is expanding rapidly, the pace at which it can be analyzed is developing more slowly. Defining these analytical requirements and selecting the most appropriate visualizations often depends on an in-depth understanding of what users need from the data. To address this problem, this paper proposes an assisted model-driven analytics approach to support visualization, taking domain knowledge and data as input. It allows the user to be guided in the mapping between domain concepts and available data, as well as in the translation of domain questions into analytical tasks that can be supported by useful visualizations for decision support. The approach is supported by a Meta-model that formalizes concepts needed to answer three fundamental questions, what, why, and how. This Meta-model contextualizes the data, the analytical tasks, and the supporting visualizations. The applicability of the proposal is shown through a demonstration case focused on the genome domain. The results highlight how useful visualizations are derived from the specified domain questions.

Keywords

Analytical requirements, Analytical visualizations, Model-driven analytics, Conceptual meta-model

1. Introduction

The amount of data that needs to be analyzed is continually increasing. This presents constant challenges when it comes to selecting and running the most appropriate visualizations for each dataset, especially when working in contexts of large volumes of data, as Big Data is challenging analytical contexts, namely when aligning data and analytical requirements and using the most appropriate visualizations for supporting users to make more informed decisions.

In this context, the multiplicity of choices and the lack of clarity regarding analytical objectives make it difficult for users to establish effective connections between the two for data visualization [1]. Each dataset has unique characteristics and not all types of visualizations

ER2024: Companion Proceedings of the 43rd International Conference on Conceptual Modeling: ER Forum, Special Topics, Posters and Demos, October 28-31, 2024, Pittsburgh, Pennsylvania, USA

*Corresponding author.

✉ a89250@alunos.uminho.pt (A. Almeida); amp.alves@campus.fct.unl.pt (A. Alves); maribel@dsi.uminho.pt (M. Y. Santos); jmp@fct.unl.pt (A. León); jmp@fct.unl.pt (J. M. Pires)

🆔 0009-0009-7965-0658 (A. Almeida); 0009-0004-5077-8571 (A. Alves); 0000-0002-3249-6229 (M. Y. Santos); 0000-0003-3516-8893 (A. León); 0000-0001-9933-936X (J. M. Pires)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

appropriately represent them [2]. Although some studies have proposed approaches to optimize data visualization ([3], [2], [1], [4], [5]), several challenges exist in aligning analytical requirements with the data, as well as translating domain questions, expressed in natural language, into analytical tasks. These tasks are then used to design analytical visualizations.

This paper proposes an assisted model-driven analytics approach to support analytical visualization, using domain knowledge and data as input. After mapping the data, the approach helps the user translate domain questions into analytical tasks that are supported by analytical visualizations. The proposed iterative process, from the identification of the most appropriate analytical tasks for each question to the analytical visualizations, is supported by the model-driven analytics component of the approach. This component includes a Meta-model that contextualizes the data, the analytical tasks applied, and the analytical visualizations that can be used to analyze the results obtained from performing these tasks. This Meta-model formalizes the concepts needed to answer three fundamental questions: *what*, the type of data the user is dealing with; *why*, the reason why the user wants to analyze that data; and *how*, how the visualization is implemented in terms of design choice.

This paper is structured as follows. Section 2 presents related work. Section 3 presents the proposed model-driven assisted analysis approach. Section 4 presents the proposed Meta-model. Section 5 presents and discusses a demonstration case applied in the genomics domain. Finally, section 6 summarizes the conclusions and future work.

2. Related Work

In the context of research approaches to data analytics, model-driven approaches are presented through the concept of modeling real-world domains as a knowledge base to ease the analysis of the modeled domains. This type of approach generally focuses on facilitating visualization design choices but is not capable of bridging the mapping of domain data into visual channels [5]. We believe that within this space, it is possible to contribute towards the inclusion of conceptual models as domain knowledge that will be used to relate domain concepts with domain data and help translate user requirements. Some works of data analytic approaches, with a focus on modeling, such as [3], propose a model-driven architecture that allows automation for the creation of visualization through the translation of the user-specific objectives/goals. Other works, e.g. [2], intend to facilitate the design choices regarding visualizations to users who lack data analysis expertise through the use of a model-driven approach in which user requirements, data profiling, and visualization design are considered. In addition, other works use iterative goal-oriented models that specify visualizations to create dashboards [6] or propose visualization frameworks that map user requirements to data visualizations [1]. The work of [7] explores how joint interactive visualization can improve the communication of knowledge between different users, promoting mutual understanding through the visual representation of data.

These studies, although they share similarities with the work presented in this paper, do not provide sound mapping strategies between the required domain knowledge and data. There is a lack of an approach that guides the user in aligning analytical requirements with the data, as well as translating domain questions into analytical tasks to ensure that analytical visualizations adequately address the identified analytical tasks. This approach seeks to bridge this gap by

aligning domain knowledge, domain data, and analytical requirements with suitable visualizations tailored to designed analytical tasks. Another defining mark of our approach is the support for identifying analytical tasks from analytical requirements using a taxonomy that maps user requirements into analytical tasks. For proposing the taxonomy, related works considered the works of [8, 5], which describe visualization tasks at varying levels of abstraction and consider that analytical tasks are driven by the need to perform complex actions based on thorough data analysis ([9, 10]), and other low-level taxonomies ([11, 12]) that typically encompass simpler actions that do not require an in-depth analysis of the overall analytical context. This taxonomy is integrated into a Meta-model that contextualizes the domain questions, the analytical tasks and the analytical visualizations useful for decision-making.

3. Model-driven Analytics Approach

Data-oriented analytics guides the identification of valuable insights from vast amounts of data. This is particularly relevant in a context where Big Data imposes complex challenges in the alignment between analytical requirements and data. Usually, the application domain is described using a conceptual model, but the definition of the analytical requirements and the identification of the corresponding visualizations are often done by looking into the users' needs. This work aims to advance model-driven analytics with an approach that considers the domain knowledge and data as input and assists the user in translating domain questions expressed in natural language into specific analytical tasks that are supported by useful visualizations. The approach here proposed follows the *human in the loop* principle proposed by [5], being supported by an iterative analytical process to augment human capabilities (and not to replace them). As [5] highlights, this iterative process i) requires interactions between the user and the several analytical visualizations supporting many possible queries and, as such, handling complexity with data analysis at different levels of detail; ii) can be framed by three essential questions: *what* data the user is dealing with, *why* the user intends to use a visualization tool, and *how* the visual encoding and interaction are constructed in terms of design choices.

Despite the relevance of the proposal presented in [5], this does not describe the concepts of the domain or map the domain to data abstractions, focusing more directly on understanding the analytical tasks required to answer domain-specific questions and how visualizations can support these tasks. Our approach considers these three fundamental pillars presented by [5], *what*, *why*, and *how*. However, it also proposes an analytical approach that, besides dealing with the concepts of the domain and the data, guides the user in mapping the concepts of the domain with the data. This is essential to ensure alignment between the domain concepts and the available data, promoting a consistent and targeted analysis of the domain's analytical requirements. The approach here proposed (Figure 1) considers three main components:

- **Domain Knowledge and Data**, where a conceptual model of the domain is available describing the main concepts and relationships, as well as the available data and the domain questions for the data;
- **Model-driven Analytics**, including the proposed Meta-model that contextualizes the data, the analytical tasks, and the analytical visualizations that can be used to analyze the results of the analytical tasks;

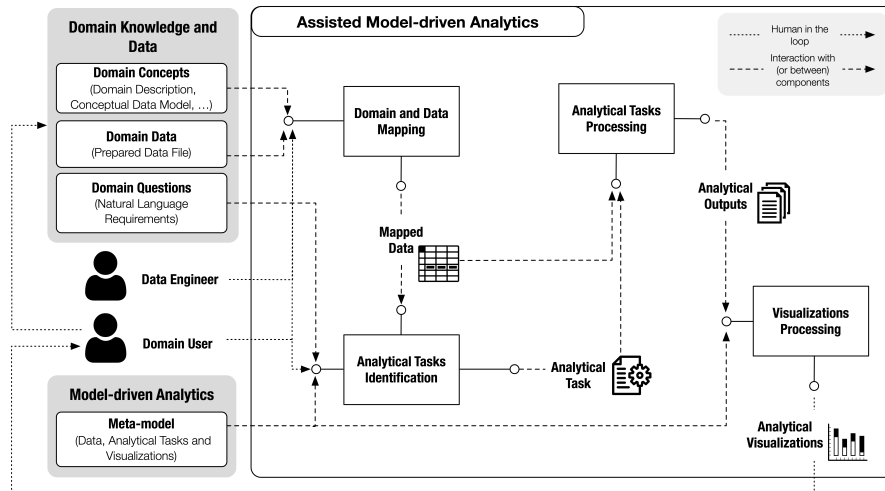


Figure 1: Assisted Model-driven Analytics Approach

- **Assisted Model-driven Analytics**, with four core steps guiding the proposed approach from the domain concepts, data, and questions to the visualizations. This encompasses the mapping of the domain concepts and data, the identification of the analytical tasks for the defined domain question(s), the processing of these tasks and, finally, the processing of visualizations that map the tasks' output into useful instruments for decision support.

Considering the Domain Concepts formalized in a data model (such as a Class Diagram or an Entity-Relationship Diagram) and a specific dataset for analysis (Domain Data), already with the prepared data, the first step of the assisted model-driven analytics components maps these two relevant pieces of information to check the alignment between them (**Domain and Data Mapping**). This step involves mapping the attributes defined in the classes of the data model with the attributes of the domain data available for analysis, ensuring a common understanding of the concepts and supporting data. A list, table, or another similar artefact must be made available as a result of this mapping step (Mapped Data). This information is useful for the identification of the analytical tasks (**Analytical Tasks Identification**), translating the Domain Questions (questions set by the domain user to be answered) into the Analytical Tasks that will be detailed with the help of the proposed Meta-model. This **Meta-model** includes a set of Analytical Tasks that define an iterative sequence of analysis processes. The Data Engineer plays a key role in supporting the identification of the analytical tasks needed to answer the domain's questions. These tasks are associated with output targets, which represent the analytical results (Analytical Outputs) obtained after the data analysis process (**Analytical Tasks Processing**). These are the inputs for the visualizations (**Visualizations Processing**). The Meta-model supports the identification of the appropriate visualizations according to the obtained results. This approach adopts a *human in the loop* philosophy, with the interaction of the **Domain User** and the **Data Engineer** and the processing components, and also interactions with or between components.

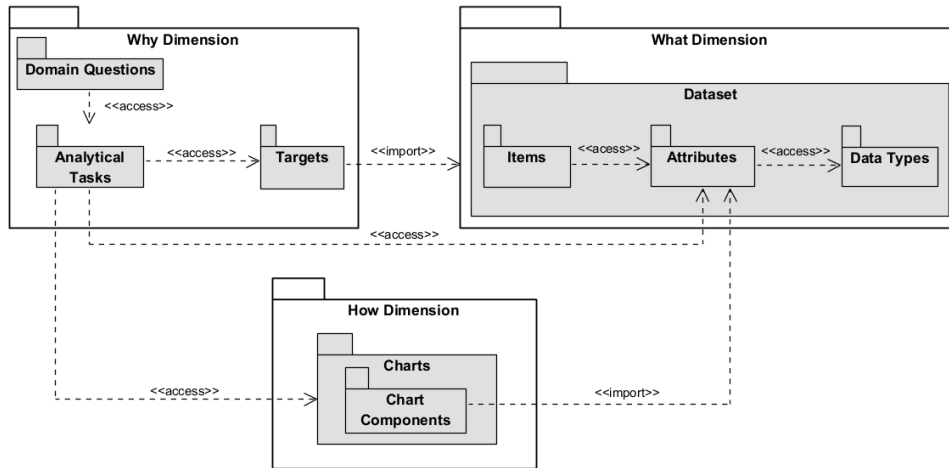


Figure 2: Proposed Conceptual Meta-model Organized into Packages

4. Model-driven Analytics Meta-model

The proposed approach is supported by a Meta-model that contextualizes the domain questions, the analytical tasks and the analytical visualizations useful for decision-making. This section first presents the proposed Meta-model and describes its main packages and concepts. The Unified Modeling Language (UML) Package Diagram presented in Figure 2 includes three main packages, What Dimension, Why Dimension and How Dimension, formalizing the concepts needed to answer three fundamental questions: *what is the focus of the analysis?*, *why are we analysing these data?*, and *how can we analyse these data?*. Each dimension includes its sub-packages and the dependencies associated with them. These dependencies can be classified into two types: *import*, where one package imports the functionality of another package, and *access*, where one package requires concepts or functionalities present in another package.

The What Dimension (Figure 3) package corresponds to the "what" component of the Meta-model and includes the Dataset sub-package with three detailed sub-packages: Items, Attributes and Data Types. Between these three sub-packages, there is an association between items and their respective attributes, and each attribute is associated with a specific data type.

The Why Dimension (Figure 4) includes three sub-packages: Domain Questions, Analytical Tasks and Targets. At the sub-package level, domain questions include the user's questions that are translated into analytical tasks and therefore require access to the functionalities present in that package. Furthermore, the analytical tasks sub-package requires access to the targets sub-package to filter, select or have as expected output one of the three possible targets available in the Meta-model, namely Attribute Target, Item Target and Dataset Target. The Analytical Tasks also have a connection with the Attributes sub-package as the Meta-model includes a relationship with a specific analytical task (Compute Attribute) which allows attributes to be derived. The other dependency occurs since certain analytical tasks allow for the creation of analytical visualizations that can be used to analyze the results of the analysis.

The How Dimension (Figure 5) package includes the Charts sub-package, with the set of

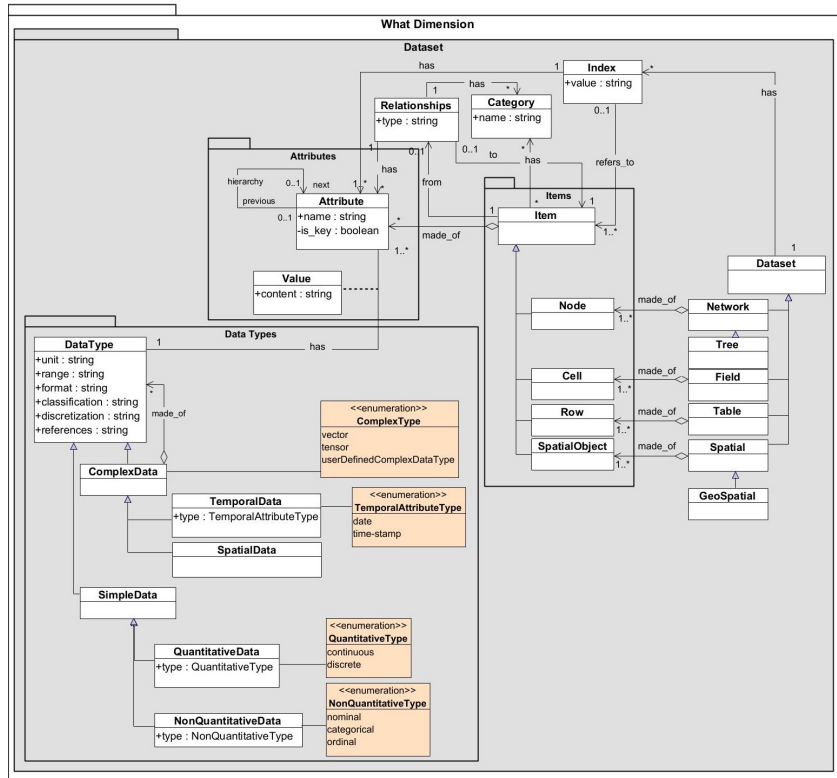


Figure 3: What Component of the Proposed Conceptual Meta-model

analytical visualizations and their components. At a lower level, the Chart Components sub-package depends on the Attributes package to identify the possible charts to use.

Detailing the Meta-model, the "what" component, presented in Figure 3, is formalized by specifying the different types of datasets (Network and the particular Tree type, Field, Table, and Spatial with the particular case of Geospatial data). The items included in these datasets aggregate different attributes that address simple data (such as a quantitative or qualitative value) or complex data (such as temporal or spatial data) and their corresponding values. Each attribute is associated with a specific data type. Datasets can include indexes for their items or attributes. Items can be classified and may establish relationships between them.

The "why" component, depicted in Figure 4, addresses the Domain Questions, which represent the questions the user wants answered and which can be translated into Analytical Tasks. The Analytical Tasks, which determine the actions that will be applied to the data and that can be formalized for addressing the analytical requirements of a domain, include tasks that express actions that can be used to find insights (tasks such as relationship, pattern, find extreme, find anomalies and find clusters), compare, determine distribution, organize, or to derive new data (that can be the expected output or be the input of another task). An analytical task usually selects data from a target (attribute, item or dataset), filters data from a target (attribute, item or dataset), and has as expected output a target (attribute, item or dataset). Depending on the

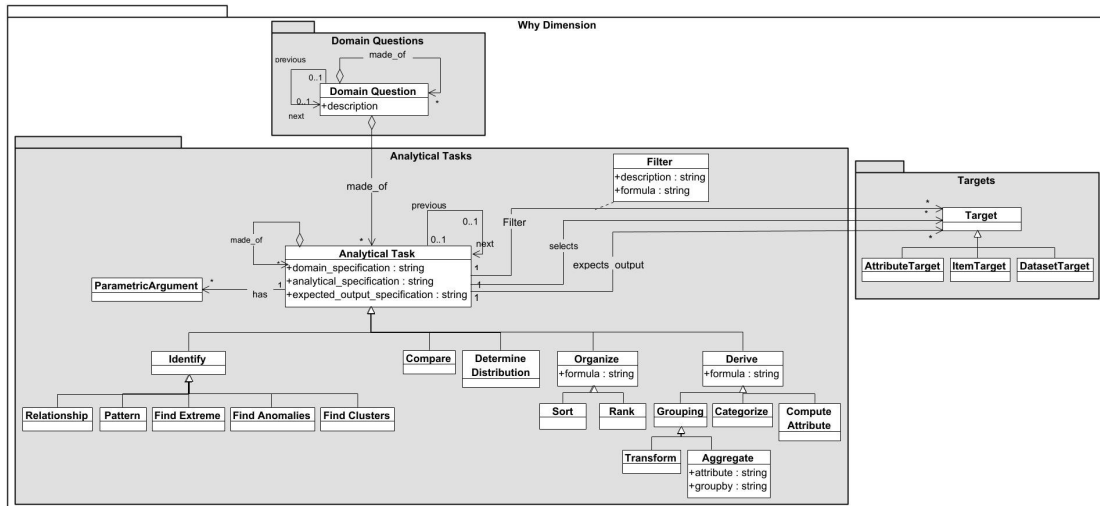


Figure 4: Why Component of the Proposed Conceptual Meta-model

analytical tasks used, some can derive visualizations to answer formulated domain questions (identify, compare and determine distribution), while others serve as intermediate steps during the analysis process (organize and derive).

The Meta-model establishes constraints on the types of charts that can be used for each analytical task since the decision on the chart will depend on the specific tasks that will be supported by analytical visualizations. The constraint next presented states that for the Relationship analytical task, whose objective is to identify and analyse the relationships and interactions between attributes, one of the possible visualization charts is a ScatterPlot (ChartType).

```

Context t:AnalyticalTask :: ChartType
If t.Identify.Relationship->notEmpty() and t.Chart-> notEmpty()
then t.Chart.type = ScatterPlot or t.Chart.type = LineChart or
t.Chart.type = HeatMap or t.Chart.type = HighlightTable or
t.Chart.type = Map or t.Chart.type = SymbolMap endif
  
```

The "how" component (Figure 5) addresses a set of analytical visualizations and their components, taking into account the analytical task(s) and the type of data used to meet users' analytical needs. Each visualization, represented by the Chart class, has derived attributes (nMarks, nAxis and nHeader) which are obtained from the number of associations between Chart and the corresponding components, ChartComponents. Each Chart can contain several chart components, depending on the type of chart (ChartType). In this way, the Chart class has different chart types (ChartType), characterized by the number of marks (nMarks), axes (nAxis) and headers (nHeader). The headers mentioned use the data from the corresponding attribute(s) to form a header with one or more entries and can be of type column or row; the axes use data that correlates with a range of values and can be of type x or y; and the marks control the type of MarkType, which can have different types of marks, such as color, size, text,

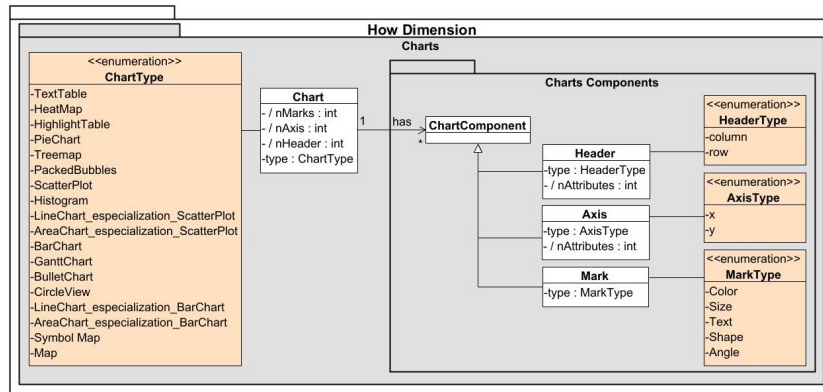


Figure 5: How Component of the Proposed Conceptual Meta-model

shape, position and angle. These can be associated with one or more data attributes (Attribute) resulting from the data analysis and have specific restrictions depending on the type of chart.

This component presents constraints related to whether or not the ChartComponents can be included, as well as the number and use of each one, impacting the presentation of the final visualization. Each group of constraints has been formulated according to the analytical requirements needed to create each specific chart.

In terms of the relationships between the classes and components of the Meta-model, each target in the why component is linked to its respective class in the what component. In addition, each target can be associated with an attribute that contains a specific order in which it will be displayed in the visualization, represented by the AttributeOrder class. The Analytical Tasks class of the why component allows the creation of zero or more visualizations depending on the type of task and, therefore, a relationship is established between it and the Chart class of the how component. Finally, for the visualization to be derived, each component (ChartComponents) is assigned to an attribute resulting from the expected output of the analytical task, thus linking the ChartComponent class to the Attribute class of the what component.

Due to the size of the images and space limitations in the paper, the global Meta-model can be found in ¹, while the full list of constraints can be found in ².

5. Demonstration Case: Genomics Domain

This section presents the application of the proposals to a demonstration case in the Genomics Domain. The domain concepts are formalized in the Conceptual Schema of Genome (CSG) [13], a data model expressed in a UML Class Diagram. Given the extension of this model, Figure 6 highlights the classes and relationships that are considered in this demonstration case. This model includes the Gene that is part of the ChromosomeElement, located in the Chromosome and can be transcribed as part of the TranscriptableElement. Additionally, a Chromosome can be located in several variations (Variation). This Variation class, Precise or Imprecise, can occur

¹<https://bit.ly/3UVCNSI>

²<https://bit.ly/3OYcU0D>

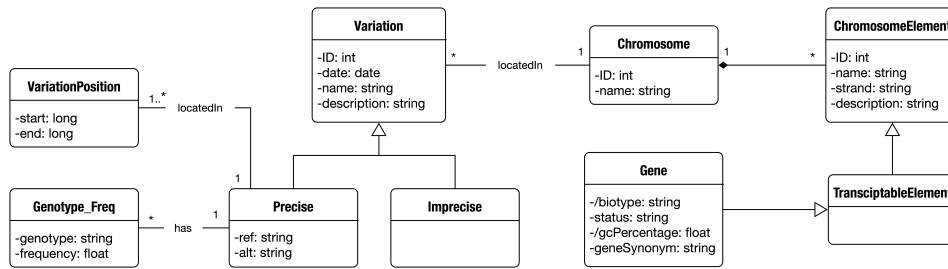


Figure 6: Excerpt of the Conceptual Schema of Genome

at specific positions in the genome (VariationPosition). In addition, the Precise class includes the reference allele (ref) and the possible alternative alleles (alt) and is associated with the Genotype_Freq class which records the frequency of different allele combinations.

The domain data with a prepared dataset includes positions in the DNA (variants) where a variation may occur, the gene affected by each variant, and the genotype of the patient (one or two copies of the alternative allele). The dataset contains six columns representing this information: Chrom, POS, REF, ALT, Genotype, and Gene. Each variant is represented by its position in the DNA. This position is represented by the chromosome (Chrom), the sequence position where the variant occurs in the chromosome (POS), the reference allele (REF), and all possible alternative alleles that could be observed in that position (ALT). The Gene column defines the gene or genes affected by the studied variant. The genotype (Genotype) determines which alleles the patient has at the studied position. The reference allele (REF) is represented by a 0, and each of the alternative alleles is represented by 1, 2, 3, Since humans have two copies of each chromosome, the Genotype value also represents if the patient presents the variant in one copy (heterozygote) or both copies (homozygote).

The result of the first step of the proposed approach (Domain and Data Mapping) is shown in Table 1, mapping the domain concepts and the available data. In the example provided, there is a direct correspondence between the attributes defined in the classes of the conceptual data model and the attributes of the data available for analysis (Domain Data). For instance, the domain concept Chromosome.name, which is an attribute of type String in the conceptual data model, corresponds to the attribute Chrom in the dataset, which is also of type String and contains the actual values of the chromosome names. Similarly, the domain concept VariationPosition.start, which is an attribute of type Long Integer and indicates the starting position of a genetic variation, corresponds to the POS attribute in the available data, which is also a Long Integer and stores the positions of genetic variations. This mapping ensures that data types and attributes are compatible between the conceptual data model and the available data, and allows for data validation, checking that all defined concepts are represented in the available data.

Next, the object diagrams with the instantiation of the Meta-model are presented. Figure 7 presents the "Why Dimension", highlighting the domain question to be discussed, the analytical tasks needed to answer the question and the targets used by the analytical tasks.

For this dataset in VCF, Variant Call Format, the following domain question was formalized "What is the distribution of variants along the genome in a sample". In the second step of the

Table 1

Domain and Data Mapping.

| Domain Concept | Domain Data |
|---------------------------------------|----------------------|
| Class.Attribute, Data Type | Attribute, Data Type |
| Chromosome.name, String | Chrom, String |
| VariationPosition.start, Long Integer | POS, Long Integer |
| Precise.ref, String | REF, String |
| Precise.alt, String | ALT, String |
| Genotype_Freq.genotype, String | Genotype, String |
| Gene.geneSynonym, String | Gene, String |

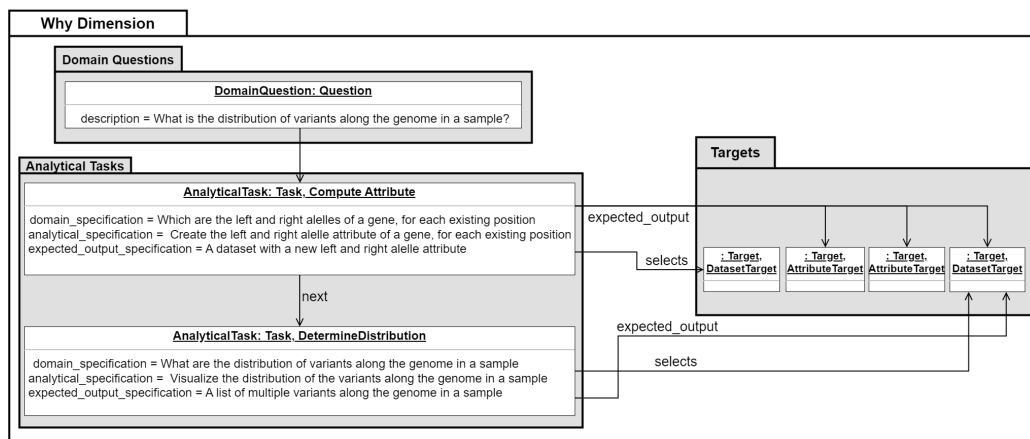


Figure 7: Why Dimension Component of the Objects Diagram

approach, Analytical Task Identification, this domain question is transformed into two analytical tasks, Compute Attribute and Determine Distribution, to analyze the data and produce the expected result. This identification of analytical tasks was supported by the Data Engineer, interacting with the Domain User. The first analytical task identified is considered fundamental, *“Create the left and right allele attribute of a gene, for each existing position”*, since this task allows the derivation of two attributes, Left Allele and Right Allele. The expected result is a dataset with the addition of the two derived attributes. The second analytical task, *“Visualize the distribution of variants along the genome in a sample”*, requires the visualization of the distribution of variants in the genomic dataset under analysis. This task targets the dataset from the first analytical task (Compute Attribute). Its expected output is a list of multiple variants.

For the “What Dimension”, Figure 8, the first analytical task selects data from a set of data in a table. These table items integrate a set of attributes with a specific data type and the corresponding attribute values. The dataset includes indexes for items and attributes. The expected output target is a dataset with two new attributes (Left Allele and Right Allele) added to the initial dataset. These attributes have the condition that, depending on the position, whenever the allele is represented by 0, the derived attribute corresponds to the REF attribute, but if one of the alleles is represented by 1, 2, 3, among others, then the derived attribute

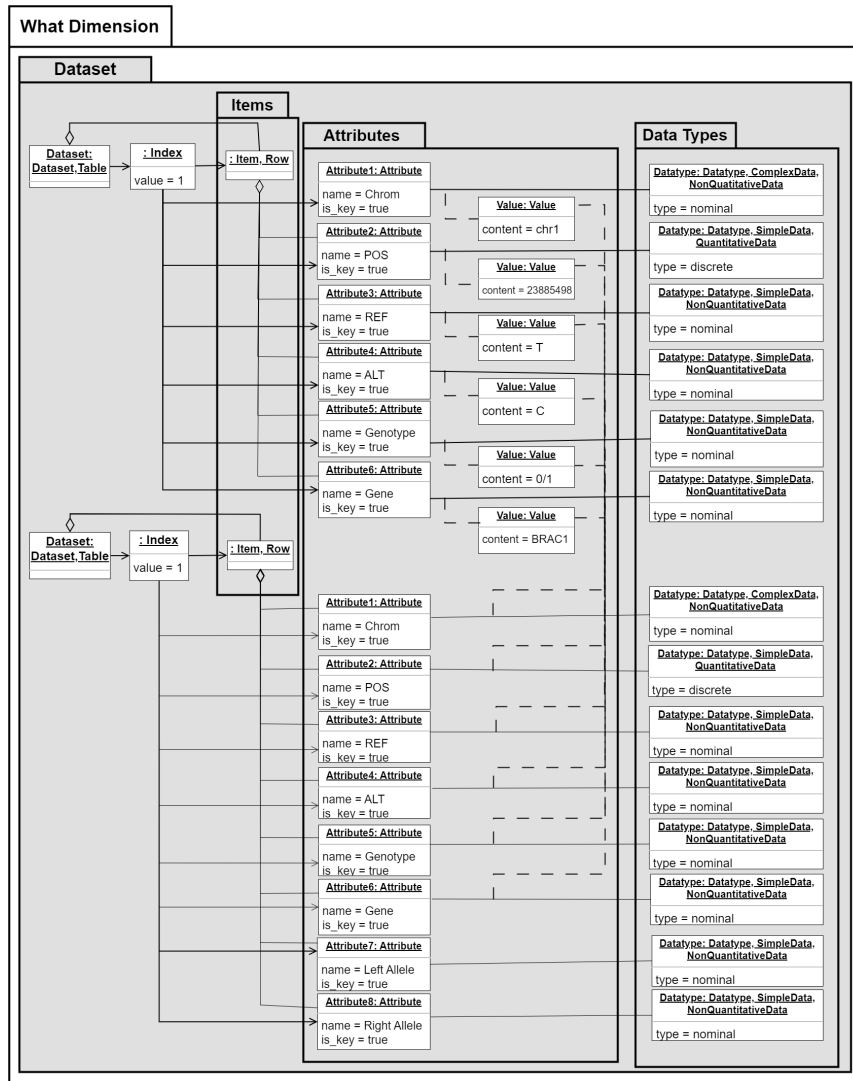


Figure 8: What Dimension Component of the Objects Diagram

corresponds to ALT, which represents all possible alternative alleles. In addition, the following analytical task involves selecting data from a table dataset, namely the dataset resulting from the previous task. The expected output target is a dataset with the six attributes belonging to the initial dataset (Chrom, REF, ALT, Genotype and Gene) and the two newly derived attributes (Left Allele and Right Allele). These analytical results derive from the processing of these analytical tasks belonging to the third stage of the approach, Analytical Tasks Processing.

After obtaining the analytical outputs, the Visualizations Processing step supports the development of a chart from the Determine Distribution task. Figure 9 shows the instantiation of the "How Dimension" component and the types of components required in terms of headers, axes and marks, as well as the type of ChartType used to form the analytical visualizations. The

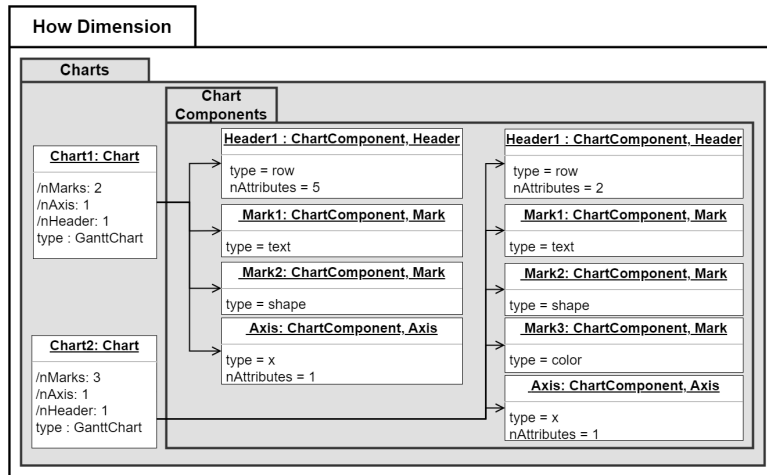


Figure 9: How Dimension Component of the Objects Diagram

complete objects' diagram joining the three dimensions can be found in ³.

The visualization generated through the Tableau tool is of the GanttChart type (Figure 10), one of the possible charts according to the restrictions of the Meta-model and the type of data. This includes a header with the Chrom and Gene attributes, an axis corresponding to the POS attribute and three marks: the first text mark includes the Chrom, Gene, Left Allele and Right Allele attributes, the second shape mark includes to the POS attribute and, finally, the color mark highlights the Gene. The order of the attributes in the visual representation is determined by the hierarchy established in the data model (Domain Concepts). In this case, the Chrom belonging to the Chromosome class according to the data mapping is presented as the first attribute, followed by the Gene and then the numerical POS attribute. There is a hierarchy between the Chromosome class and the Gene class, and genes are constituent parts of these chromosomes. The POS attribute, being a numerical attribute associated with the horizontal axis of the chart, is found after the Gene attribute in the analytical visualization. Now, it is possible to analyze, for a given chromosome and gene, its position in the genome and the corresponding alleles. For example, the chr1 chromosome with the BRAC1 gene at a position between 20 and 30 million corresponds to the T/C alleles, with the Left Allele represented by T and the Right Allele represented by C.

The obtained visualization allows analytics for the defined domain question "What is the distribution of variants along the genome in a sample?". Although it is possible to create other types of charts, they would not be as effective in demonstrating what the Domain User requires. It is relevant to note that the visualization presented as a result of this demonstration case has been validated by a domain expert, thus ensuring that the selected visualization efficiently meets the analytical objectives. Furthermore, it is important to mention that it is up to the user to make the final decision regarding the choice of the most suitable visualization within the range of possible visualizations suggested in the Meta-model.

Based on this demonstration case, we found that the analytical approach proposed facilitates

³<https://bit.ly/3SSTvQ5>

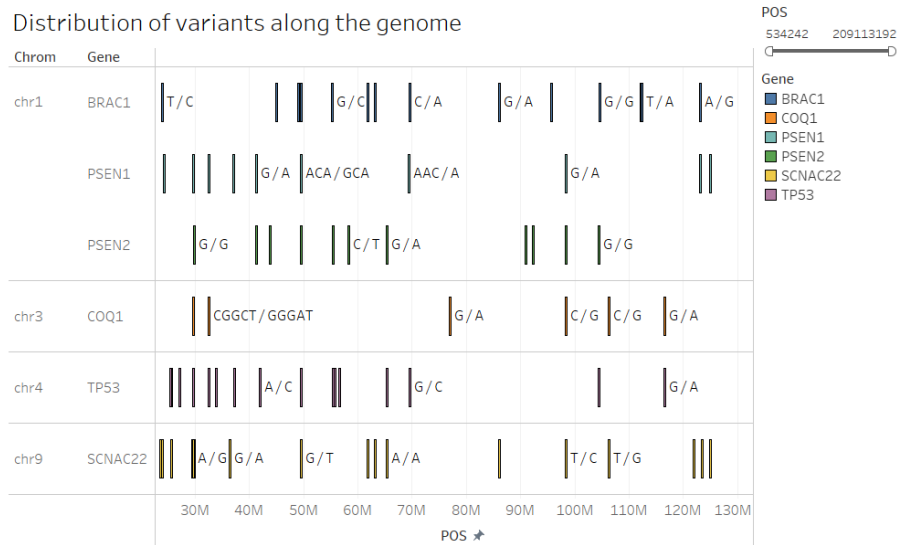


Figure 10: Gantt Chart Visualization

the development of visualizations that effectively address domain questions. By integrating domain knowledge and data as input, this approach aligns analytical requirements with data and assists users in translating domain questions into actionable analytical tasks supported by useful visualizations. The Meta-model plays a crucial role in this iterative process by contextualizing data, identifying applicable analytical tasks, and guiding the creation of visualizations. This approach is applicable across various domains and datasets.

6. Conclusions

In this paper, we have presented an approach to supporting analytical visualization that provides guidance, from mapping data and identifying analytical tasks to creating analytical visualizations capable of responding to users' analytical needs. This approach is supported by a Meta-model, which contextualizes the data, the analytical tasks and the analytical visualizations that make it possible to analyze the results of these tasks. To verify the validity of the approach, we applied it to a demonstration case in the genomics domain, presenting an example of a useful analytical visualization.

In future work, further evaluation and the extension of the Meta-model, to add interactivity between the user and the proposed visualizations, are considered.

Acknowledgements

This work has been supported by *FCT – Fundação para a Ciência e Tecnologia* within the R&D Units Project Scope UIDB/00319/2020 (ALGORITMI) and UIDB/04516/2020 (NOVA LINCS), and by the Spanish Ministry of Universities and the Universitat Politècnica de València under

the Margarita Salas Next Generation EU grant. This paper uses icons made available by www.flaticon.com.

References

- [1] T. Li, X. Wei, Y. Wang, A requirements-driven framework for automatic data visualization, in: *Enterprise, Business-Process and Information Systems Modeling*, Springer Nature Switzerland, 2023.
- [2] A. Lavalle, A. Maté, J. Trujillo, Requirements-driven visualizations for big data analytics: A model-driven approach, in: *Conceptual Modeling*, Springer International Publishing, 2019.
- [3] M. Golfarelli, S. Rizzi, A model-driven approach to automate data visualization in big data analytics, *Information Visualization* (2019). doi:10.1177/1473871619858933.
- [4] S. J. Mellor, A. N. Clark, T. Futagami, Model-driven development - guest editor introduction (2003). doi:10.1109/MS.2003.1231145.
- [5] T. Munzner, *Visualization Analysis and Design*, A K Peters/CRC Press, 2014. doi:10.1201/b17511.
- [6] A. Lavalle, A. Maté, J. Trujillo, S. Rizzi, Visualization requirements for business intelligence analytics: A goal-based, iterative framework, in: *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 2019, pp. 109–119. doi:10.1109/RE.2019.00022.
- [7] M. J. Eppler, Facilitating knowledge communication through joint interactive visualization, *JUCS - Journal of Universal Computer Science* 10 (2004) 683–690. doi:10.3217/jucs-010-06-0683.
- [8] M. Brehmer, T. Munzner, A multi-level typology of abstract visualization tasks, *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis)* (2013) 2376–2385.
- [9] J. Heer, B. Shneiderman, Interactive dynamics for visual analysis, *Communications of the ACM* 55 (2012) 45–54. doi:10.1145/2133806.2133821.
- [10] E. R. A. Valiati, M. S. Pimenta, C. M. D. S. Freitas, A taxonomy of tasks for guiding the evaluation of multidimensional visualizations, in: *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 1–6. URL: <https://doi.org/10.1145/1168149.1168169>.
- [11] M. X. Zhou, S. K. Feiner, Visual task characterization for automated visual discourse synthesis, in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, ACM Press/Addison-Wesley Publishing Co., USA, 1998, p. 392–399. URL: <https://doi.org/10.1145/274644.274698>. doi:10.1145/274644.274698.
- [12] S. Wehrend, C. H. Lewis, A problem-oriented classification of visualization techniques, *Proceedings of the First IEEE Conference on Visualization: Visualization '90* (1990) 139–143.
- [13] A. García, J. C. Casamayor, On how to generalize species-specific conceptual schemes to generate a species-independent conceptual schema of the genome, *BMC Bioinformatics* (2021). doi:10.1186/s12859-021-04237-x.