Universiteit Leiden
The Netherlands

Opleiding Informatica

Variant Algebra based Star Allele Calling

Lucas C.A.W. van Osenbruggen

Supervisors:
Mark A. Santcroos & Jonathan K. Vis

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

04/07/2023

Abstract

Personal genetic variation has a significant impact on the metabolism of drugs. This leads to harmful side effects and reduced effectiveness for some people. Genetic screening is recommended for several drugs, but this is not yet common practice. Problems with current methods are that they often require high-quality sequencing data and that it is not always clear how a method arrives at its conclusions. Here, we propose a novel method of finding star allele classifications for individuals based on their genetic sequence. This method achieves perfect accuracy on a benchmark dataset with phased sequencing data. The method also achieves an accuracy of 94.7% on lower-quality data with incomplete phasing information. Further, we propose a method of visualisation of the call that makes the conclusion more explainable.

Contents

1 Introduction

Personalised medicine is the concept of adjusting medical treatment to the individual patient. People differ in many ways, but often receive the same medical treatment for a given condition. The idea of an "average patient" is problematic in itself as clinical research is often done on a small subset of the population. Here, we focus on *pharmacogenetics*, the study of how an individual's genetic makeup affects medication response. It is shown that taking genetic factors into account when prescribing medication can significantly reduce negative side effects [1].

One of the challenges of pharmacogenetics is the large amount of genetic variation in the human population relevant to medicine. A genetic *variant* is a small genetic difference in a gene between individuals. When a variant is present in a gene, it is called an *allele* of that gene. Individuals have a unique set of variants and this makes it difficult to generalise the findings of clinical research. Furthermore, some variants are rarer than others and have been studied less extensively [2]. In addition, technical limitations of sequencing technologies reduce the accuracy of genetic data [3].

The human genome is organised into pairs of chromosomes, one inherited from each parent. For each region, an individual thus has two copies that can have different variants. These copies are called *haplotypes* when considered separately and are called *diplotype* when considered together. Standards exist that aim to organise knowledge on genetic variation relevant to pharmacogenetics, one of which is the *star allele* nomenclature. Star alleles are a way of grouping haplotypes based on their clinical relevance. The diplotype of any individual can be described with two star alleles. For instance, a possible diplotype of the gene *CYP2D6* is *CYP2D6**2/*4. This allows researchers to draw general conclusions from clinical research on individual diplotypes.

The goal is to determine or *call* the star alleles for an individual. Various methods have been proposed, but these are often limited to high-quality sequencing data [3]. In addition, methods are often statistical in nature and population-based, meaning that they return only the most likely star alleles. Existing methods are also often not transparent in how they arrive at their conclusion. As a consequence, a human cannot easily detect whether the conclusion is correct.

In this thesis, we propose a novel deterministic method of star allele calling. This method makes use of the *variant algebra* [4]. The variant algebra is a mathematical framework that allows us to reason about genetic variants by defining relations between them. We describe how the structure of these relations is visualised and how this visualisation is used to call star alleles. In practice, *phasing* information, which describes which variants are in the same haplotype, is often missing. Therefore, we also describe how the method is adjusted for data with missing phasing information. The method is explainable through the visualisation which shows the conclusions and relevant data of the call.

In Section 2, more background on the concepts of pharmacogenetics and star allele calling is given. Additionally, the variant algebra is discussed in more detail. In Section 3, the method of star allele calling and visualisation is described. Section 4 contains the results of the calling method on a benchmark dataset as well as some auxiliary results. In Section 5, we discuss the results and limitations of the method. Finally, we end with a conclusion in Section 6.

2 Background

In this section, we look in more detail at the field of pharmacogenetics. We also consider the main difficulties of star allele calling: comparing variants and interpreting imperfect sequencing data. We further describe how we use the variant algebra for comparing variants. Finally, we discuss existing methods of star allele calling and their limitations.

2.1 Pharmacogenetics

One of the most extensively studied pharmacogenetic genes is *Cytochrome P450 2D6* or *CYP2D6* for short. *CYP2D6* is involved in the breakdown of 15–25% of medications, including antidepressants, opioids used in pain management and cancer medication [5]. The gene exists on chromosome 22 and has a length of around 4,000 base pairs. It contains nine exons and encodes a protein of 497 amino acids [6]. As of writing, 150 unique variants have been described that affect the protein structure of *CYP2D6* (see Section 4.1). Some of these variants also affect the function of the protein leading to individual differences in medication metabolism [3]. Due to the high number of alleles being discovered, online resources have been developed to organise this information.

The *Pharmacogene Variation Consortium* (PharmVar) is a group of experts that maintains the star allele nomenclature [7]. PharmVar provides a curated database of star alleles which have been observed in the population (`https://www.pharmvar.org`). Star allele definitions are being updated as new haplotypes are discovered in the population. As of writing, 164 star alleles have been defined for *CYP2D6* (see Section 4.1). Additionally, the PharmVar database includes annotations about star alleles, such as their impact on protein function. The data in the PharmVar database is aggregated from multiple sources, including the Pharmacogenomic Knowledgebase (PharmGKB) [8] and the Clinical Pharmacogenetics Implementation Consortium (CPIC). PharmGKB and CPIC provide more detailed clinical information about variants, while PharmVar focuses on standardizing information. Here, we focus on the PharmVar database, as it is the most comprehensive source of star allele definitions for *CYP2D6*.

The process of *sequencing* is determining the molecular structure of DNA. Through this process, the sequence of a haplotype is observed. This observed sequence is then compared to a *reference sequence*, a consensus of the sequence in the population, to identify variants. There are several advantages to communicating variants instead of sequences. The length of the full genome is large with approximately $3 \cdot 10^9$ nucleotides, but the number of differences between individuals is small; in the order of 10^7 [9]. Therefore, storing variants is more space efficient than storing sequences. Additionally, variants allow for human interpretable reporting. One standard for describing variants is the *HGVS nomenclature* [10]. As an example, the variant `NC_000022.11:g.42128945C>T` describes reference sequence `NC000022.11`, GRCh38 chromosome 22 [6]. The type of reference sequence, `g`, indicates that it is genomic. The position of the variant is 42,128,945 relative to the reference sequence. The *operator*, or type of variant, is a substitution changing a `C` to a `T`. The `C` in this representation also follows from the reference sequence and position.

While variant descriptions are useful, they also introduce complications. The first is that different variant descriptions of the same observed sequence are possible. One reason for this is the choice of operator. Some operators defined in the HGVS notation are insertion, deletion, inversion, duplication and substitution. Variants can be described with different operators. For instance, an insertion of a `T` next to another `T` can also be described as a duplication of the `T`.

Another way in which alternative variant descriptions are possible is through the choice of

reference sequence. Reference sequences can span different parts of the genome. Here, we use the reference sequence `NC_000022.11`, which spans the entire chromosome 22. Reference sequences also exist that span a single gene. Variant descriptions using a gene reference sequence have the advantage of being shorter and therefore more readable. Not all genes and other genomic features are described by gene reference sequences. By using a genomic reference sequence, we can describe variants on any position on the chromosome.

Often there are multiple positions on the reference sequence on which a change could have led to the same observed sequence. For instance, for the reference sequence `AAA` and observed sequence `AA`, there are three possible positions on which a deletion could have occurred: `1delA`, `2delA` and `3delA`. The variant descriptions `NC_000022.11:g.42132027_42132028insT` and `NC_000022.11:g.42132049_42132049insT` describe the same observed sequence since both insertions are in a sequence of repeating `T`'s.

Normalisation is the process of selecting a *canonical* representation of a variant out of the different possibilities. HGVS nomenclature uses the 3'-rule, which states that the representation that is the nearest to the 3'-end of the sequence should be used [10]. Additionally, the HGVS nomenclature recommends giving a DNA description of a variant and prioritising the different operators in the following order: substitution, deletion, inversion, duplication, and insertion. Using normalisation a single canonical representation of a variant is chosen. In the previous example, `NC_000022.11:g.42132027_42132028insT` is the canonical representation. A canonical representation thus allows for the comparison of variants as the same observed sequence will yield the same representation for a given reference sequence.

However, the normalisation method may differ for other representation languages. *Variant call format (VCF)* is another standard of describing variants. While HGVS nomenclature is a syntax for describing variants, VCF is a file format for storing variants. VCF uses the 5'-rule which states that the representation that is the nearest to the 5'-end of the sequence should be used [11]. This is in contrast to the 3'-rule used by the HGVS nomenclature and variant representations may have different positions in these languages. This can make it difficult to identify variants between different representation languages.

Variants are represented as individual changes, but are often present together with other variants in a haplotype. Variation in haplotypes results in variation in *phenotypes*, which in this context is the metabolism of medication. The phenotype is determined by protein function and since not all variants have an impact on protein function, different haplotypes can result in the same phenotype [12]. Due to this, many unique haplotypes exist, making clinical studies based on a single haplotype often hard to generalise. This is what the star allele nomenclature aims to solve [7]. A *core allele* is a collection of clinically relevant variants that are shared between haplotypes with functionally equivalent protein functions. The star allele nomenclature defines clinically relevant as impacting protein structure, function or expression. For instance, the *CYP2D6**10 allele is described as `NC_000022.11:g.42126611C>G;42130692G>A`. There are eight described haplotypes of the *CYP2D6**10 allele, all of which include the core allele. These clinically studied haplotypes are called *suballeles* of the *CYP2D6**10 allele. For example, *CYP2D6**10.001 includes the core allele, but also the variant `NC_000022.11:g.42129130C>G` which is believed to have no clinical relevance. Core alleles thus allow us to group haplotypes. Often, the term star allele is used synonymously with core allele.

Describing variants in a haplotype individually introduces complications. Individuals who have all variants in a core allele definition are believed to have the same phenotype. However, this is not always the case. A haplotype that includes a star allele definition may also have *personal*

variants, which are not described in PharmVar. In this case, it cannot be predicted with certainty what the effect of the variant is on the phenotype. Yet, due to the relative rarity of variants compared to the size of the gene and the fact that many positions will not result in a change in the protein, it is unlikely that they have a significant impact on the phenotype [12].

Another complication is the comparison of haplotypes with star alleles. Different descriptions of variants are possible and there are thus different descriptions of haplotypes as well. For instance, the variant `NC_000022.11:g.42128176_42128178del` defines the core allele *CYP2D6**9. If this variant and the personal variant `NC_000022.11:g.42128180C>T` are both observed in a haplotype, this haplotype is described as `NC_000022.11:g.42128176_42128180delinsCT`. While in practice most variants are more distant from each other, this example shows that in general, it is not always clear how to match a haplotype to a core allele. In Section 2.2, we look at a method of comparing haplotypes that can address this complication.

We considered two problems that variant descriptions introduce, but a third problem arises from sequencing methods. Some sequencing methods require the DNA to be *fragmented* before sequencing since they have a maximum length of DNA that they can read at once. A consequence of fragmentation is that *phasing* information is lost. Chromosomes exist in pairs and each person has two alleles for every gene. Variants that exist on the same chromosome are said to be *in cis* or *phased*. However, as these chromosomes are fragmented during sequencing it is not always possible to determine which variants are *in cis*. For some variants, it is possible to infer phasing information. For instance, if two variants are close together they are likely to be observed in a single read or in overlapping reads. Newer sequencing methods have been introduced with longer read lengths and this can help in determining phasing, but these methods are less accurate and can introduce false positives [13]. Phasing can also be determined by using the fact that one haplotype is inherited from the mother and the other from the father. By also sequencing the parents' genome, it is possible to determine phasing. However, this is not always an option. It is possible to determine if a variant occurs on both chromosomes in a pair, i.e., is *homozygous*, or occurs on only one chromosome, i.e., is *heterozygous*. This is possible since the sequencing process finds a single or two different sequences for a given position for homozygous and heterozygous variants respectively. For homozygous variants, the phasing is known as they exist on both chromosomes. Still, in general, incomplete phasing presents a problem for star allele calling.

Finally, some variants may be difficult to identify by sequencing. *Structural variations* are large-scale changes in the genome. This presents problems for variant calling [14, 15]. One type of structural variation is *copy number variation* (CNV). In the case of CNV, there is a different number of copies of a gene present on a single chromosome. *CYP2D6**5 for instance, is a deletion of the *CYP2D6* gene which means there is only one gene copy. Multiple copies of a gene can also be present on the same chromosome. CNV can present problems for variant calling as the reads from the different copies will be aligned to the same gene. This can also prevent phasing detection. There are also star alleles definitions that are hybrids of two different genes, such as *CYP2D6**13 which is a hybrid of *CYP2D6* and *CYP2D7*. These hybrid genes are typically not described on a sequence level.

2.2 Relations between genetic variants

Comparing two sequences using variants can present some difficulties because of different possible representations. The *boolean algebra for genetic variants*, or variant algebra, is a formal method of describing the relations between pairs of variants [4]. In this model, variants are defined as the minimal sets of insertions and deletions that unambiguously describe the transformation from a reference sequence to an observed sequence. The minimal number of insertions and deletions

in this set is called the *simple edit distance* between two sequences. More complex operations are expressed as combinations of insertions and deletions. Similarly, alleles are simply "larger" variants in the variant algebra. However, here we refer to all combinations of zero or more variants as alleles.

In the variant algebra, relations between alleles are defined that consider all representations together. This avoids the difficulties of comparing different representations of the same variant. Each pair of alleles has one of the following four relations:

1. *Equivalence*. Two alleles are equivalent if they describe the same observed sequence. This relation is used to identify if a haplotype exactly matches a star allele. We have seen how normalisation is used to determine equivalence in Section 2.1. The advantage of the variant algebra is that this avoids the differences in normalisation between different variant description standards. For instance, a variant is normalised to the 5'-end in VCF, which is a common file format for sequencing data, but is normalised to the 3'-end in HGVS, which is what PharmVar uses.
2. *Containment*. One allele can contain another allele. In the context of star allele calling, we use this to identify if a haplotype is a suballele of a core allele. As a suballele in PharmVar consists of the core allele and neutral variants it is expected that a suballele contains its core allele. The converse also holds, the core allele *is contained* in its suballeles. Therefore, the core alleles that are contained in a haplotype describe the observed sequence of the haplotype.
3. *Overlap*. Two alleles overlap if they share some information. This relation may indicate that a haplotype shares some information with a star allele. However, the characterisation of overlap, or determining which information is shared, is generally not trivial.
4. *Disjoint*. Two alleles are disjoint if they do not share any information. This is often the case when two alleles are distant from each other.

An efficient algorithm for determining these relations is publicly available [4]. We use these relations to reason about genetic variants and to identify star alleles in Section 3.

2.3 Related Work

Currently, the recommended method of star allele calling is through experimental validation [3]. Experimental validation selectively detects specific star alleles by several molecular techniques such as polymerase chain reaction [16]. This approach is expensive and time-consuming. It is also difficult to scale to all known star alleles. Additionally, the results are limited to the specific star alleles that were tested for and conclusions cannot be updated to reflect new findings.

Recently, tools have been developed that can call star alleles computationally based on sequencing data. Calling on sequencing data has the advantage of being less biased than experimental validation as it is easily scaled to all known star alleles. In addition, this data is reusable as new star alleles can be called from the same data. Some notable examples of calling methods are Astrolabe [17], Aldy [18], and Stargazer [19]. These methods have been tested with both real-world data and simulated data consisting of all possible star alleles [20]. No method achieved perfect accuracy in the test, but all methods performed well on real data with Aldy and Astrolabe being the most accurate. All methods achieved a lower accuracy on structural variations, such as copy number variations and hybrid alleles.

There are several limitations to the current methods. All algorithms require high-quality

sequencing data [3]. Additionally, they find the presence of specific variants in a sample and find the star allele that matches these variants most closely. As there are different possible alignments for a set of variants, matching is not always sufficient. Furthermore, these methods are statistical and make use of population genetics to determine the most likely star alleles [20]. This means that there is an inherent bias toward more common star alleles. Finally, the methods do not provide a clear explanation of why a certain star allele was called. As data may be ambiguous or limited, transparency is important to allow for manual inspection and correction [16].

3 Method

In this section, we describe a novel method of star allele calling based on a graph representation of the relations between star alleles. We first describe the visualisation of this graph, followed by the calling method for phased sequencing data. We then describe how the method is extended to sequencing data with missing phasing information. An implementation of both the visualisation and calling method is available on GitHub at `https://github.com/lukaas33/VA-star-allele-calling`.

3.1 Visual representation of relations

We represent the relations between star alleles in a *graph*. Star alleles are represented as nodes in the graph, and the relations between them are represented as edges. Since two star alleles always have a relation, the graph is fully connected or complete, consisting of n^2 relations for n variants. Every star allele has an equivalence relation to itself, creating self-loops. Edges have labels describing the type of relation and are weighted by the simple edit distance between two star alleles. Furthermore, the graph is *mixed* since it consists of both directed asymmetrical and undirected symmetrical relations [21]. In this section, we describe how this graph representation is simplified and visualised.

3.1.1 Simplification of the graph representation

The fully connected graph is useful for some applications as any relation between two star alleles is available directly. However, many of the relations in the graph are redundant since they are derivable from other relations. Redundancy can occur because of the properties of the boolean relations [4]. By pruning redundant relations the graph is simplified which makes it more interpretable.

We do not have to calculate all relations. As every star allele is equivalent to itself, i.e., equivalence is reflexive, we do not represent this in the graph. Furthermore, the overlap, equivalence and disjoint relations are symmetrical meaning that they apply in two directions. We therefore only need to calculate one direction of the relation. We also have to calculate only one direction of the containment relation since the other direction is the converse. For instance, by calculating that *CYP2D6**39 is contained in *CYP2D6**10 we know that *CYP2D6**10 contains *CYP2D6**39. We arbitrarily choose the "is contained" direction of the containment relation to represent the relation. We thus only have to calculate the upper or lower triangular matrix of allele pairs which saves computation time.

The graph representation is mixed since it contains both directed asymmetrical and undirected symmetrical relations. This is a problem for the implementation as many existing algorithms are designed for either directed or undirected graphs. For implementation purposes, it is therefore useful to consider each relation type as a separate graph. The containment relation is represented in a directed graph, and the overlap and equivalence relations are represented in undirected graphs. There is no graph of disjoint relations. If there is no path between a pair of star alleles in any of the three graphs, this is understood as a disjoint relation.

We can further simplify the graph by pruning redundant relations. Some relations are redundant because of equivalent star alleles. Equivalent star alleles must have the same relations to all other star alleles, and we can thus prune the relations of one of them. The star allele to keep is partially arbitrary, but in the context of star allele calling we choose to keep the core allele. To avoid information loss, we keep the equivalence relations. An example of equivalence contraction

simplification is shown in Figure 1. In this example, the core allele *CYP2D6**44 is equivalent to its suballele *CYP2D6**44.001, and we prune the relations of the suballele. In general, it may occur that equivalence exists between non-core alleles. For this case, we define the following ordering of which star alleles to keep: core alleles, suballeles, and finally variant alleles.

The containment graph also has redundant relations. An example is given in Figure 2, where *CYP2D6**39 is contained in *CYP2D6**10 and *CYP2D6**10 in *CYP2D6**147. Due to the transitivity of the containment relation we derive that *CYP2D6**39 is also contained in *CYP2D6**147. It is therefore redundant to represent this relation in the graph. The process of removing these relations is called *transitive reduction* [22]. For the directed containment graph, this results in an acyclic graph with a minimal number of relations. Equivalence is also transitive, and we remove redundant equivalence relations by keeping only the relations of the core allele as described in the previous paragraph.

Figure 1: The core allele *CYP2D6**44 is equivalent to its suballele *CYP2D6**44.001. They have the same relations to all other variants, like the containment relation with *CYP2D6**22. Here we keep the core allele *CYP2D6**44 and the relation between *CYP2D6**44.001 and *CYP2D6**22 is thus pruned.

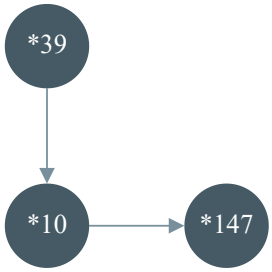


Figure 2: Both the equivalence and containment relation are transitive. A transitive reduction is applied to the graph to remove redundant relations. In this example *CYP2D6**39 is contained in *CYP2D6**10 which is contained in *CYP2D6**147. The transitive reduction removes the edge between *CYP2D6**39 and *CYP2D6**147.

Some overlap relations follow from containment relations. When two star alleles both contain a third star allele, they share the information of the third and must therefore at least overlap. An example of this is shown in Figure 3. All overlap relations are pruned between star alleles with a common *ancestor* in the containment graph. An ancestor is a star allele that is contained in the star alleles of interest. While the common ancestor means that there must be some overlap, it does not necessarily describe the entire overlap since the two star alleles may share more information.

Another simplification of overlap relations is made based on containment relations. When a star allele overlaps with a second star allele it also overlaps with any star allele that contains the second star allele. The overlap with the second star allele is more specific and implies the overlap with the other star alleles it is contained in. Therefore, this overlap relation can be pruned. An example is shown in Figure 4. Since a containment relation can be represented indirectly when the transitive reduction has been applied, a *topological ordering* of the contained star alleles is needed to find the most specific star allele.

By applying the methods in this section, the graph representation is simplified significantly. A complete mixed graph with self-loops is simplified to one directed acyclic graph and two undirected simple graphs. For the *CYP2D6* gene, 99.5% of all relations are pruned (see Section 4.1). This simplified graph representation thus is easier to interpret visually and is more space efficient.

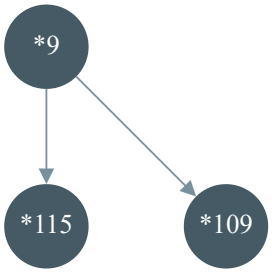


Figure 3: In this example both *CYP2D6**109 and *CYP2D6**115 contain *CYP2D6**9. It is implicit that *CYP2D6**109 and *CYP2D6**115 overlap since they share some information. Therefore, the overlap relation is pruned.

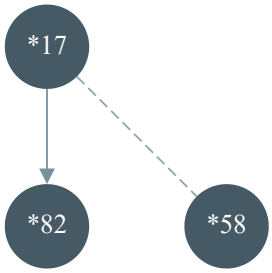


Figure 4: In the example above *CYP2D6**17 is contained in *CYP2D6**82. *CYP2D6**58 overlaps with *CYP2D6**17 and therefore *CYP2D6**58 must also overlap with *CYP2D6**82. Only the most specific overlap has to be represented.

3.1.2 Information visualisation

The goal of information visualisation is to improve human cognition of data. Information visualisation is providing a mapping of data properties to visual attributes which helps people form a mental model of abstract information. We implemented an interactive visualisation of star allele data as a graph. The implementation makes use of the Cytoscape Javascript library [23]. This visualisation is useful in data exploration. Additionally, visualisation can be used to explain the reasoning behind the call.

We differentiate visually between different types of star alleles by the use of node shape and size. Core alleles are displayed as circles and suballeles as smaller circles. Observed haplotypes are displayed using hexagons. We also display star alleles consisting of a single variant from the definition of other star alleles. These *variant alleles* are represented as rectangles when the variant is described in PharmVar and as rounded rectangles when the variant is not described in PharmVar. This serves as an annotation to help make the connection to PharmVar star allele definitions. An example is given in Figure 5. The core allele *CYP2D6**2 for instance, has the variant `NC_000022.11:g.42126611C>G` in its definition on PharmVar, and we thus also display a star allele node for this variant.

All star allele nodes are displayed with their star allele number as a label. The gene prefix is left out as this is constant within the graph. Additionally, we omit the zeroes from the suballele number for brevity. Similarly, variant alleles are displayed in the HGVS nomenclature with the reference sequence omitted.

(a) Observed allele. **(b)** Core allele. **(c)** Suballele. **(d)** PharmVar variant allele. **(e)** Personal variant allele.

Figure 5: Different node styles are used to differentiate the types of star alleles in the graph.

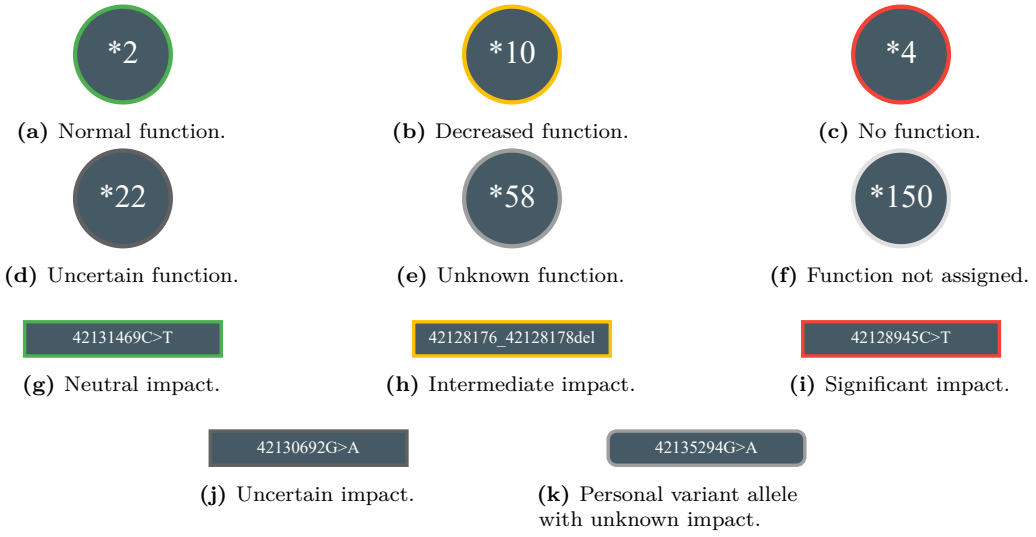
Relations are either directed or undirected. Symmetrical relations like overlap and equivalence are indicated with a simple line, while the asymmetrical relation containment is indicated with an arrow. To differentiate overlap from equivalence a second attribute of line style is used. Overlap

is displayed using a dashed line while the others are displayed using a solid line.

Additionally, we represent the strength of the relation between two star alleles using the line width. For simplicity, we use an ordinal scale of the relation types instead of the simple edit distance. As equivalence is the strongest relation between two star alleles, it is displayed with the thickest line. This is followed by containment and then overlap.

The previous attributes show the relations between star alleles. We also display functional annotations of the star alleles. The border colour communicates the impact on the function of star alleles. This is less prominent than body colour and still allows for discrimination as shown in Figure 6. An annotation of "no function" is displayed using a red border which indicates the most significant effect on the protein. Whether a specific star allele has a clinical effect on an individual depends on context such as diplotype, medication in question, etc. An annotation of "decreased function" is displayed in yellow and "normal function" in green. The other functional annotations in PharmVar, "unknown", "uncertain" and "not assigned", all indicate a lack of certainty. They are displayed as different shades of grey, which still allows for discrimination, but also the easy grouping of these annotations. Gray indicates missing information.

For variant alleles, we use the impact annotation. When the impact is neutral on a protein level, the nodes have a green border. For annotations such as "deletion", "insertion" and "duplication" the nodes are displayed with a yellow border as they indicate a potentially more significant impact on the protein. Third, annotations such as "splice defect", "frameshift" and "early stop" indicate the largest impact and these nodes are indicated with a red border. Furthermore, the impact of the annotation "missense" is not certain and is this indicated with a dark grey border. Finally, an unknown impact is indicated with a light grey border. This is often the case for variants that are not described in PharmVar.



(a) Normal function. **(b)** Decreased function. **(c)** No function.

(d) Uncertain function. **(e)** Unknown function. **(f)** Function not assigned.

(g) Neutral impact. **(h)** Intermediate impact. **(i)** Significant impact.

(j) Uncertain impact. **(k)** Personal variant allele with unknown impact.

Figure 6: The functional annotations of star alleles are communicated using border colour.

When displaying information about a specific call of a haplotype, additional properties are shown. Not the entire PharmVar database is relevant for a call. We define *context* as all star alleles with a path of increasing or equal relation strength from the haplotype. For instance, a haplotype

that contains *CYP2D6**74* has this star allele in its context. As *CYP2D6**74* is equivalent to `NC_000022.11:g.42129819G>T` the latter is also in the context. However, *CYP2D6**84* which overlaps with *CYP2D6**74* is not in the context as the path from the haplotype decreases in strength. Next, it is indicated which star alleles are homozygous in the haplotype with a thicker border. Further, a haplotype often has direct relations to many variant alleles. To display these compactly without loss of information we group these. Finally, the called alleles are highlighted in the graph. We call this visualisation a call graph. Examples of call graphs are given in Section 3.2.

Cytoscape allows for creating automatic placements of nodes, called *layouts*. Some prominent layout options are: The *cose* layout simulates nodes as a physical system with forces between them. The layout places more highly connected nodes centrally and allows for identifying components and clusters [24]. It is suitable for displaying the entire collection of star alleles of a single gene in one image. The *dagre* layout is a depth-first tree layout suitable for hierarchical data. It is therefore useful for displaying calls. The *concentric circles* layout places the most highly connected at the centre with all others in a circle around it. This is also a hierarchical layout, but will reduce overlapping edges while also making better use of the available space. However, it will not show hierarchy above one level, making dagre preferable.

Cytoscape allows for interactions with the data. It is possible to zoom in on detail. It is also possible to select nodes and display the context of the node. Additional annotations of the selection are shown in a sidebar. There is an option to filter selected nodes and display them as a new graph. Additional filtering is possible by dragging nodes. The layout is changeable which allows the user to experiment with the best way to display the selection. A search function allows for fast localising of specific nodes.

3.2 Star allele calling

In the next sections, we describe a novel method of star allele calling. We first consider the problem of calling a star allele with phased data. We then describe how the method can be applied to data with missing phasing information.

3.2.1 Star allele calling on phased data

Variant data is often provided as a VCF file, which describes the variants observed together in a diplotype (see Section 2.1). VCF files also contain phasing information describing which variants are on the same chromosome or *in cis*. For now, we assume that the phasing is perfectly known. We separate the data into two haplotypes. For instance, one of the examples in the benchmark dataset that is used in the results comes from sample `HG00111`. Only one variant is observed in this example: `NC_000022.11:g.42128242del`. This variant is known to be in both haplotypes, i.e., is homozygous. Two alleles are observed, `HG00111-A` and `HG00111-B` which both consist of a single variant in this case. The assignment of `A` and `B` is arbitrary as there is no ordering of the haplotypes.

The next step is using the variant algebra to find all relations between haplotypes and PharmVar star alleles. We also find all relations with variant alleles, alleles consisting of single variants from the PharmVar database or sequencing data. Variant alleles serve as an annotation in the visualisation and are never present in a call. Additionally, we find all relations between the PharmVar alleles themselves which is needed for the third step, where we apply the simplification method described in Section 3.1.1.

Calculating all relations is time-consuming and grows exponentially with the number of alleles as there are n^2 pairs for n alleles. Calculating relations is therefore done in parallel.

We now have the simplified graph representation with the relations of the haplotypes. In this method, we distinguish between direct and indirect relations. For instance, a haplotype may contain both *CYP2D6**10 and *CYP2D6**39, but since *CYP2D6**39 is contained in *CYP2D6**10 the relation with *CYP2D6**39 is indirect. The direct relations are *more specific* in that they cover more variants of a haplotype. We consider all direct relations as star allele matches of a haplotype. There are several scenarios possible:

Default call. A haplotype is disjoint with all star alleles except for variant alleles. An example is shown in Figure 7. In this case, the haplotype consists of personal variants that together and individually are disjoint with all star alleles. As there is no evidence for any other star allele, we call the default star allele. In the case of *CYP2D6* this is *CYP2D6**1. This star allele is equal to the reference sequence and is thus an empty allele. An empty allele is disjoint with all non-empty star alleles in the variant algebra [4]. Therefore, calling *CYP2D6**1 allele in this instance is an exception to the method presented here.

Equivalence. A haplotype is equivalent to a star allele. After simplification, a haplotype can only have one equivalence relation. The simplest example is a haplotype that is empty and thus equivalent to *CYP2D6**1. Another example is given in Figure 8. The haplotype `HG00111-A` is equivalent to *CYP2D6**3. The equivalence relation with *CYP2D6**3 is the only match as the other relations are indirect. Since `HG00111-B` is equivalent to *CYP2D6**3 as well, we call the diplotype of `HG00111` as *CYP2D6**3/*3.

In some instances, a haplotype is equivalent to a suballele. An example is shown in Figure 9. The suballele *CYP2D6**22.001 is the most specific description of `HG00337-A` and is therefore reported. However, for some purposes, it is useful to report the core allele instead. For instance, we use the core allele representations to validate the results of this method in Section 4.2. We can find the core allele by traversing the call graph from the suballele to the first core allele. For *CYP2D6**22.001 this is *CYP2D6**22. However, this approach can result in multiple core alleles, such as for *CYP2D6**4.002 which contains the core alleles *CYP2D6**4, *CYP2D6**10 and *CYP2D6**74 as shown in Figure 11. Additionally, this approach does not find the core allele when the suballele does not contain it. This is relatively rare in PharmVar, but does occur (see Appendix B). Therefore, we use the described core allele of a suballele.

One contained match. A haplotype is not equivalent to any star allele described in PharmVar, but does contain a star allele. In this case, the variant algebra proves useful since it allows us to find the closest match for a haplotype The haplotype describes all the variants in the definition of a contained star allele, but also describes additional variants. In the simplified graph, the most specific star allele is directly contained while less specific star alleles are indirectly contained. As an example, we consider the haplotype `NA07357-B` in Figure 10. The haplotype is not equivalent to any star allele, but does contain *CYP2D6**6.005. This is not an equivalence as `NA07357-B` also contains `NC_000022.11:g.42132049dup`. *CYP2D6**6.005 is therefore the closest match for a haplotype.

When a haplotype is not equivalent to any star allele it describes additional variants that are not covered by their star allele match. Some of these variants are personal while others are described in PharmVar in the definition of a star allele not in the context. If the star allele call is used as a prediction of the phenotype, these additional variants must have a neutral effect on the phenotype. For instance, a haplotype contains a star allele with a normal function, but also contains an impactful variant that may have a different phenotype. For variant alleles described in PharmVar, annotations of their impact on the protein are available and for many personal variants,

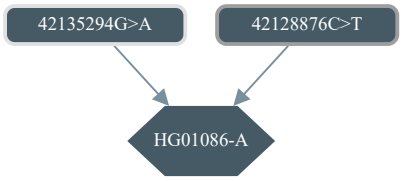


Figure 7: `HG01086-A` contains only personal variant alleles. The only match for this haplotype is therefore *CYP2D6**1.

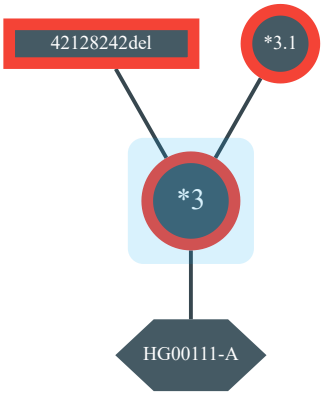


Figure 8: `HG00111-A` is equivalent to *CYP2D6**3. *CYP2D6**3 is also equivalent to *CYP2D6**3.001 and `NC_000022.11:g.42128242del`. We consider only the direct relations to a haplotype and thus report *CYP2D6**3.

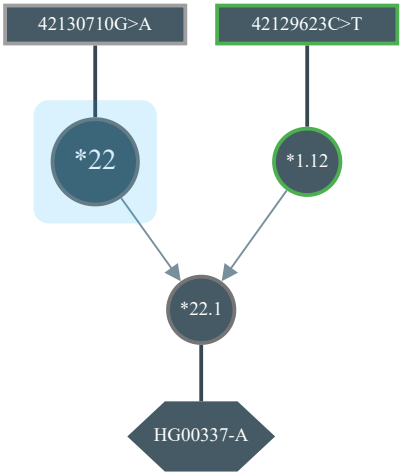


Figure 9: `HG00337-A` is equivalent to *CYP2D6**22.001. This suballele is the only direct match for this haplotype. For some purposes, it may be desired to report the highlighted core allele *CYP2D6**22 instead.

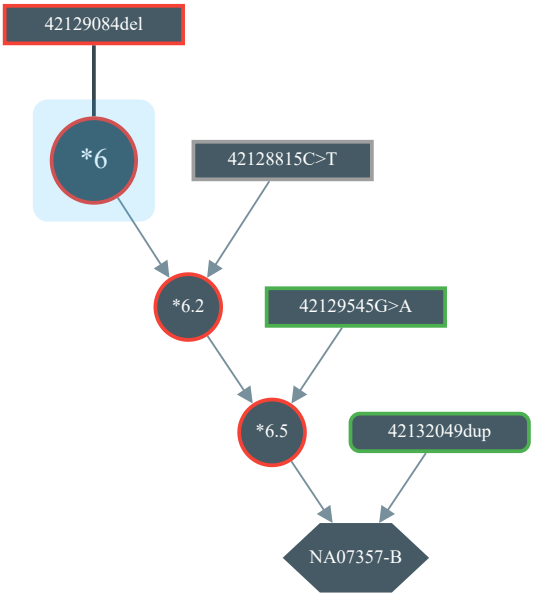


Figure 10: `NA07357-B` is not equivalent to any star allele since it contains a personal variant. It does, however, contain *CYP2D6**6.005 which is the most specific star allele match.

annotations are available in other databases. Using these annotations it is possible to filter out some variants that are known to be neutral. However, it may be difficult to predict the impact of a combination of variants on the phenotype. For these reasons, we cannot guarantee that the additional variants are neutral. Therefore, we display these as variant alleles in the call graph with their impact annotations indicated. For instance, in Figure 10 the haplotype contains the personal variant allele `NC_000022.11:g.42132049dup`. Furthermore, `NC_000022.11:g.42129084del` is annotated as causing a frameshift which is not likely to be undone. We will return to this issue in Section 5.2.

Multiple matches. A haplotype directly contains multiple star alleles. In the star allele nomenclature, there is always a single star allele description for a haplotype. When we find multiple matches for a haplotype, we must therefore choose a single star allele to represent the haplotype. An example is given in Figure 11 where `HG00276-A` directly contains *CYP2D6**1.054, *CYP2D6**4.002, *CYP2D6**4.005 and *CYP2D6**4.011. In this case, the haplotype consists of the variant in the definition of *CYP2D6**4 and additional variants. In general, we call haplotypes that match multiple suballeles of the same core allele. Furthermore, suballeles of *CYP2D6**1 also do not affect the call of a haplotype these are annotated as neutral.

Calling is not as straightforward for matches with different core alleles. An example of this scenario is given in Figure 12. `HG03703-B` directly contains *CYP2D6**10.001 and *CYP2D6**99. It is unclear whether this haplotype should be seen as a suballele of *CYP2D6**10 or of *CYP2D6**99. We define a prioritisation scheme to determine which star allele should be reported in this scenario. The rationale is that star alleles that have the most significant effect on the phenotype should be prioritised since these are more clinically relevant. Star alleles with a functional annotation of "no function" are prioritised over those with a "decreased function" and those with a "decreased function" are prioritised over those with a "normal function". If the functional annotations are the same, no choice can be made. In this case, we prioritise *CYP2D6**99 as this has a more significant effect on the phenotype. We see *CYP2D6**99 thus as the closest match of this haplotype.

However, there are star alleles without a functional annotation because of missing evidence. We do know that the annotation cannot be more significant than "no function". We can therefore prioritise a star allele with a functional annotation of "no function" over one with a functional annotation of "unknown function". Yet, we cannot prioritise a star allele with an annotation of "decreased" function over a star allele with an annotation of "unknown function" as it is potentially more significant. If no choice can be made, both star alleles are reported as the call for the haplotype.

Overlapping matches. The haplotype overlaps with one or more star alleles. Overlap is a weaker relation than containment, but still means that a haplotype and star allele are not disjoint. In the simplified graph, overlap relations are often indirect as they can be explained by containment relations. We found no examples of observed directly overlapping star alleles in a benchmark dataset 4.2.1. In other datasets, however, it may occur that a haplotype directly overlaps with a star allele. Therefore, we consider directly overlapping star alleles as matches, unless the haplotype contains another star allele since this is a stronger relation.

The described method always calls a haplotype. The supported output of the method is a visual representation of the call graph. However, a textual representation of the call is useful for validation as this is the conventional method of reporting star alleles. Any textual representation is a simplification of the information in a haplotype. We define several levels of detail. The

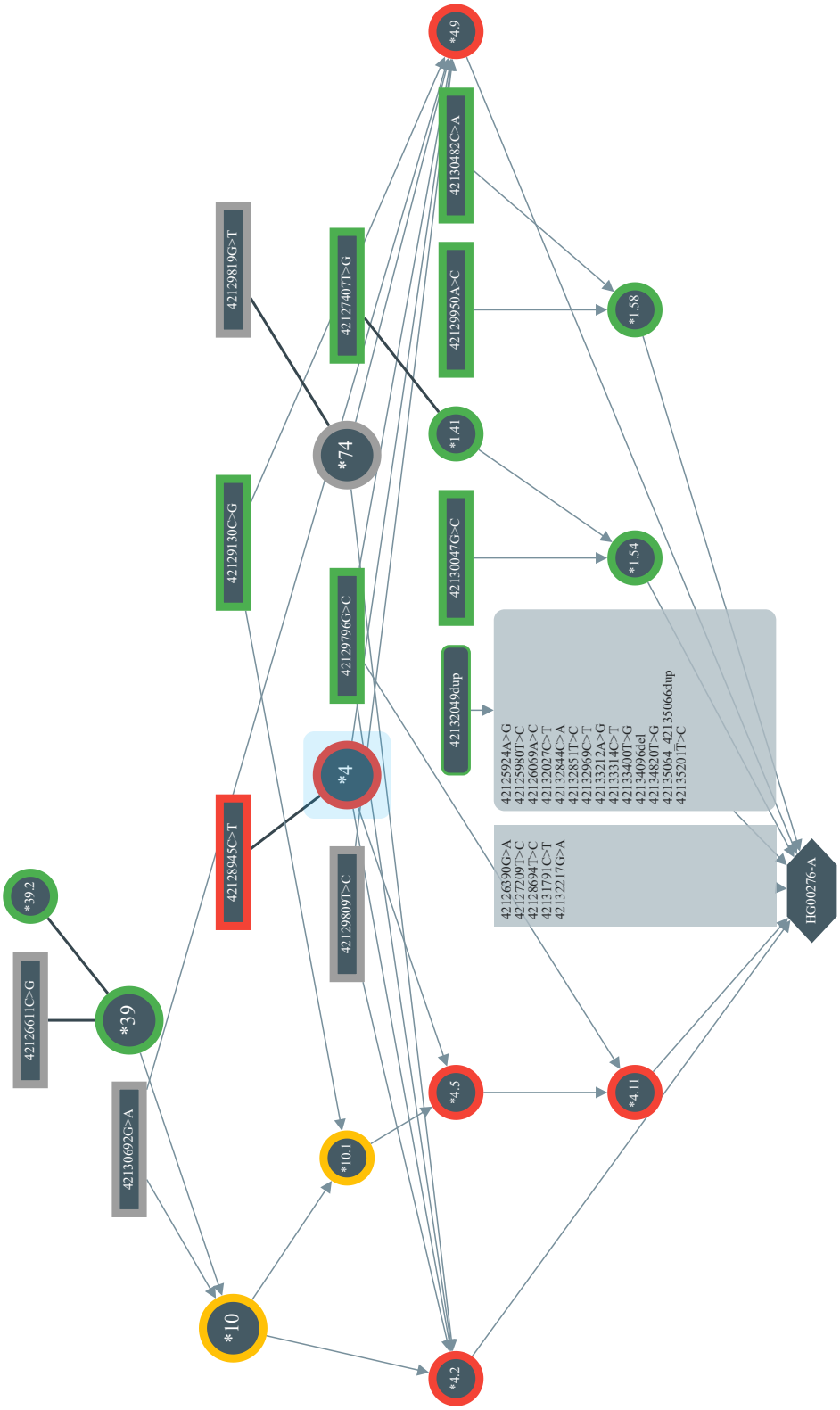


Figure 11: `HG00276-A` directly contains multiple suballeles of *CYP2D6**4 and a suballele of *CYP2D6**1. We can call this haplotype as *CYP2D6**4 as it consists of the variants in the definition of *CYP2D6**4 and additional variants.

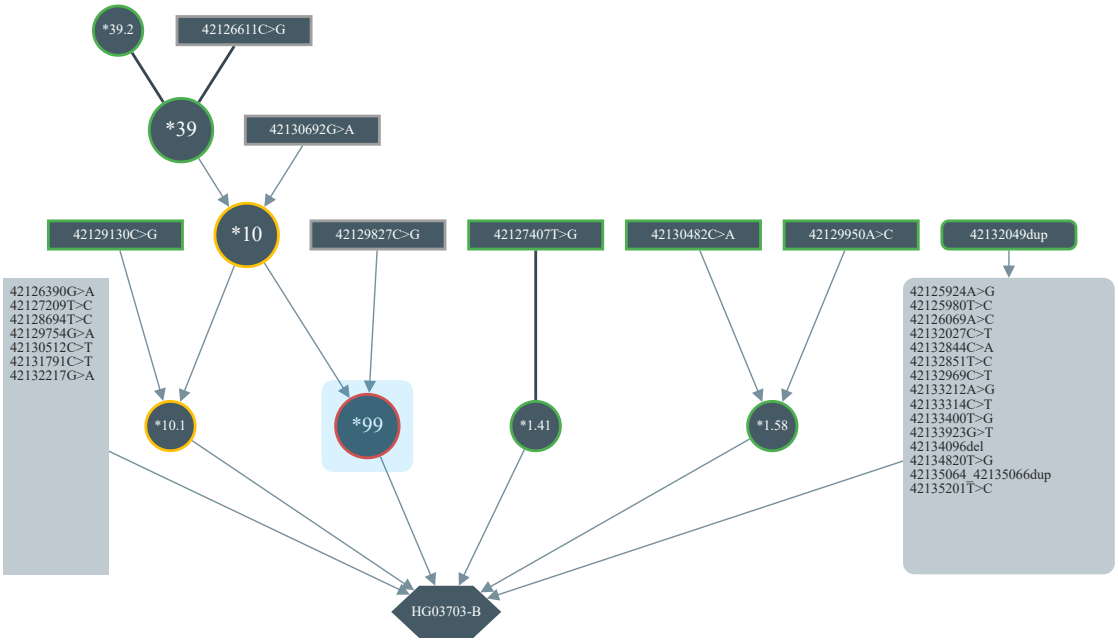


Figure 12: `HG03703-B` directly contains both *CYP2D6**10.001 and *CYP2D6**99. We call this haplotype as *CYP2D6**99 as this has an annotation of "no function" and has the most significant effect on the phenotype. We omit suballeles of *CYP2D6**1 as these are not part of a call.

most complete textual representation of this call consists of all directly related star alleles and variants. For the example in Figure 10 this is: *CYP2D6**6.005, `NC_000022.11:g.42132049dup`. An intermediate level of simplification is to show only the prioritised star allele: *CYP2D6**6.005 And finally, the simplest representation is to show only the prioritised core allele: *CYP2D6**6. We omit suballeles of *CYP2D6**1 from a call unless there are no other alleles. This last representation is the conventional method of clinically reporting star alleles and is therefore used in validation in Section 4.2.

The call graph output, such as Figure 12, does not remove any information of a haplotype. Additionally, the reasoning behind the call is shown, and the conclusion is highlighted. This visualisation is useful for human interpretable reporting.

3.2.2 Approach to calling with missing phasing

The calling method described in the previous section assumes that perfect phasing information is available. However, as phasing data is often missing in practice, we extend the method to handle this. We consider data that has no phasing information besides variants being either homozygous or heterozygous. When there are some homozygous variants this information is useful for calling. We define a hypothetical allele with the suffix "all" that consists of all variants in a diplotype.

We first apply the calling method described in the previous section on these diplotypes. This results in multiple matches that have to be separated into two haplotypes. In Figure 13, the diplotype `HG00111-all` consists of all variants in the sequencing data, in this case only `NC_000022.11:g.42128242del`. Furthermore, we know that this variant is homozygous and since the star allele *CYP2D6**3 consists of only this variant this star allele is also homozygous. The

only match of `HG00111-all` is thus *CYP2D6**3, and we call *CYP2D6**3/*3. As we only consider the gene *CYP2D6* here, we shorten this to *3/*3.

Generally, separating matches into two haplotypes is not trivial. A combination of unphased variants in the diplotype may result in a star allele match that is not actually present in either haplotype. We illustrate this in Figure 14. In this example a simplified call graph, leaving out suballeles and directly contained variant alleles, of the unphased haplotype `HG00421-all` is shown. The phased call for this diplotype is *2/*10. However, the definition of the *CYP2D6**65 consists of the variants from the definitions of *CYP2D6**2 and *CYP2D6**10. The unmodified calling method thus results in either *1/*65 or *65/*65.

We use the knowledge of homozygous variants to improve the calling method. The variant `NC_000022.11:g.42126611C>G` is homozygous in the diplotype. In the call *1/*65 however, this variant is not present in haplotype `A`. We can thus conclude that this call does not fit the data. Additionally, we know that *CYP2D6**65 is heterozygous as it consists of some heterozygous variants, and we thus rule out *65/*65. A possible explanation of the match with *CYP2D6**65 then is that it is a combination of *CYP2D6**2 and *CYP2D6**10. If we look at indirect relations in the call graph we find the call *2/*10 which is valid since the homozygous variant is present in both haplotypes.

Another valid call is *39/*65 since the homozygous variant `NC_000022.11:g.42126611C>G` is present in both haplotypes. Both alternative calls are equally valid explanations of the data. Often there are multiple alternative calls possible when phasing information is missing. We want to find all alternative calls as we cannot determine which one is correct.

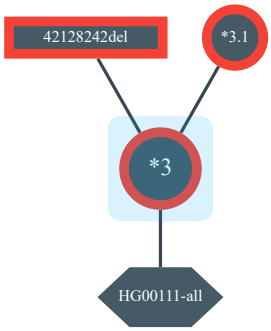


Figure 13: The call graph for `HG00111-all`. This haplotype consists of a single homozygous variant and is equivalent to *CYP2D6**3.

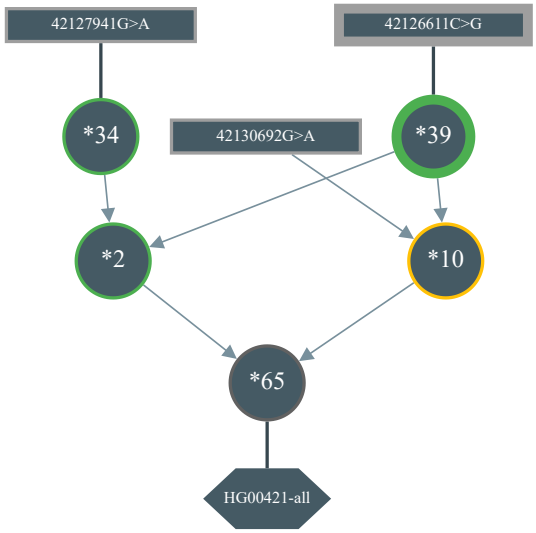


Figure 14: A simplified call graph of `HG00421-all`. The only match is the star allele *CYP2D6**65. This star allele is a combination of *CYP2D6**2 and *CYP2D6**10.

Alternative calls exist because some variants are unphased and thus placeable in either haplotype. One strategy is to try all possible distributions of heterozygous variants over the two haplotypes exhaustively. For the example in Figure 14 we have the following variants:

`NC_000022.11:g.42130692G>A` is heterozygous, `NC_000022.11:g.42127941G>A` is heterozygous and `NC_000022.11:g.42126611C>G` is homozygous. Only two haplotypes are possible and these correspond to *2/*10 and *39/*65.

This exhaustive search strategy finds all valid haplotypes for the unphased variants. However, this is not feasible for large samples as the number of diplotypes grows exponentially with the number of heterozygous variants. The number of diplotypes possible is limited by Equation 1. In this equation, n is the number of heterozygous variants. We see that in the case of `NA19143-all` with $n = 22$ there are 2,097,152 distributions possible.

$$d(n) = \frac{1}{2} \cdot \sum_{k=0}^{n} \binom{n}{k} = 2^{n-1} \tag{1}$$

Another problem with this approach is that we create haplotypes that may not be described by any star allele. In the example in Figure 14 we create haplotypes that are equivalent to star alleles, but this is often not possible because of personal variants. Finding the closest match for each haplotype involved finding the relations with all star alleles which is not feasible, and we still create many haplotypes not described by any star allele.

We want to use the relations of the diplotype to the star alleles in PharmVar to efficiently find alternative calls of existing star alleles. To achieve this, we use an approach similar to the method for phased data but with some adjustments. We use a call graph to identify star allele matches, but instead of finding one star allele that describes a haplotype, we must now find two star alleles that describe a diplotype.

In the structure of a call graph, a lower distance to the diplotype indicates a closer match that covers more variants of the diplotype. We also know which star alleles are homozygous since these consist of only variants that are homozygous in the diplotype. Therefore, the other star alleles heterozygous since they consist of at least one heterozygous variant. We check if we can find a valid call with these closest matches. This is done by trying all haplotype combinations of heterozygous star allele matches. Only haplotypes described by one core allele are considered. This avoids ambiguous calls where we do not know which star allele to report and also ensures that a call consists of alleles described in PharmVar. Furthermore, a call must not contradict the information of homozygous and heterozygous variants.

The main obstacle to calling with missing phasing is that a diplotype may match star alleles that do not describe either of the two haplotypes. We therefore need to consider star alleles with a higher distance to the diplotype. We do this by considering the hypothetical call graph without some star alleles, i.e., ignoring star alleles. By recursively ignoring all combinations of star alleles we find all possible calls.

We prefer valid calls that consist of star alleles that cover the most variants of the diplotype as these describe the diplotype the best. Therefore, we order the calls by how close the star alleles are to the diplotype. Furthermore, we do not want to report all possible calls when this is not necessary. We define an algorithm using the above approach in Section 3.2.3.

3.2.3 An algorithm for calling with missing phasing

In this section, we extend upon the ideas of Section 3.2.2 and describe an algorithm for finding alternative calls for data with missing phasing information. We do not consider copy number variation here, but discuss how this algorithm can be adjusted to handle different copy numbers in Section 5.1. The pseudocode for this algorithm is shown in Algorithm 1.

We find the call graph of a diplotype that describes the relations to PharmVar star alleles. We consider the star alleles with the lowest distance to the diplotype to be the closest matches of the diplotype. A *state* in the algorithm represents a view of the call graph that includes only the closest star alleles and does not include variant alleles. Variant alleles are not included since they are not present in a call. We represent a state as a multiset since the star allele matches are unordered and homozygous star alleles are present twice in a diplotype. In Figure 13, `HG00111-all` is equivalent to *CYP2D6**3. This star allele is homozygous in the call graph since it is defined by a variant that is known to be homozygous in the diplotype. The initial state for this example is thus $\{$*CYP2D6**$3^2\}$ which is written more compactly as $\{$*$3^2\}$. Most often, the diplotype contains one or more star alleles. For simplicity, we ignore suballeles of *CYP2D6**1 in this algorithm as these are often contained in haplotypes and do not affect the call (see Section 3.2.1). In Figure 15 for instance, the call graph of `NA12815-all` is shown which is described by the initial state {*41, *119.001}. The initial state has overlapping star alleles only when the diplotype has no stronger relations such as containment. Again, this is also how the phased calling method treats overlapping star alleles. Finally, the initial state is empty if the diplotype is disjoint with all star alleles.

The algorithm then checks the combinations of heterozygous star alleles in the state. This yields multiple diplotypes consisting of the homozygous star alleles in both haplotypes and the heterozygous star alleles in one of the haplotypes. The number of diplotypes is given by Equation 1 where n is the number of heterozygous star alleles in the state. We only consider diplotypes where each haplotype is described by a single core allele. Suballeles of the same star allele are allowed to be included in the same haplotype, as shown in Figure 11 where we call the core allele *CYP2D6**4 (see Section 3.2.1). For instance, the initial state {*41, *119.001} in Figure 15 only yields the call *41/*119.001. For states that consist of a single star allele a, the call is *1/*a since *CYP2D6**1 is the default call (see Section 3.2.1). Finally, the empty state yields the call *1/*1.

A call is only valid if it is consistent with the information of homozygous and heterozygous variants. We find the set of variants alleles in the contexts of the star alleles in each haplotype (see Section 3.1.2). We count how many variant allele sets include a heterozygous variant allele, which cannot be two. In Figure 15 for instance, the call *41/*119.001 is not valid since the variant allele `NC_000022.11:g.42127803C>T` is present in both haplotypes and is known to be heterozygous in the diplotype. No valid alternative calls can thus be found from the initial state in this example. States do not include variant alleles and as a result, a heterozygous variant allele may have a count of zero. For instance, `NC_000022.11:g.42125924A>G` is directly contained in the diplotype `NA12815-all` and is present in neither of the variant allele sets of the initial state. For the same reason, homozygous variant alleles may have a count of zero, one or two. We thus do not consider calls invalid based on homozygous variant alleles. Since states do include star alleles, calls are invalid if homozygous star alleles are not included in both haplotypes. In Figure 14, both valid calls, *2/*10 and *39/*65, include *CYP2D6**39 in both haplotypes, either by *CYP2D6**39 itself or as a star allele that contains it. We check if star alleles known to be homozygous in the diplotype are present in both haplotypes of a call.

The closest star alleles do not always yield valid calls when phasing information is missing. Therefore, we need to consider more distant star alleles in the call graph as well. We do this by ignoring the presence of some combination of matches in the call graph. Figure 16 for example, shows the full call graph of the diplotype `HG00421-all` with initial state {*2.002, *10.004, *65.001}. No valid calls are found from this state since there are three star alleles. For a state with m star alleles, there are $2^m - 1$ possible combinations of star alleles that we can ignore. In this example, we ignore three single star alleles, three pairs of star alleles and all star alleles. Ignoring a star allele means that we consider a view of the call graph where this star allele is not present.

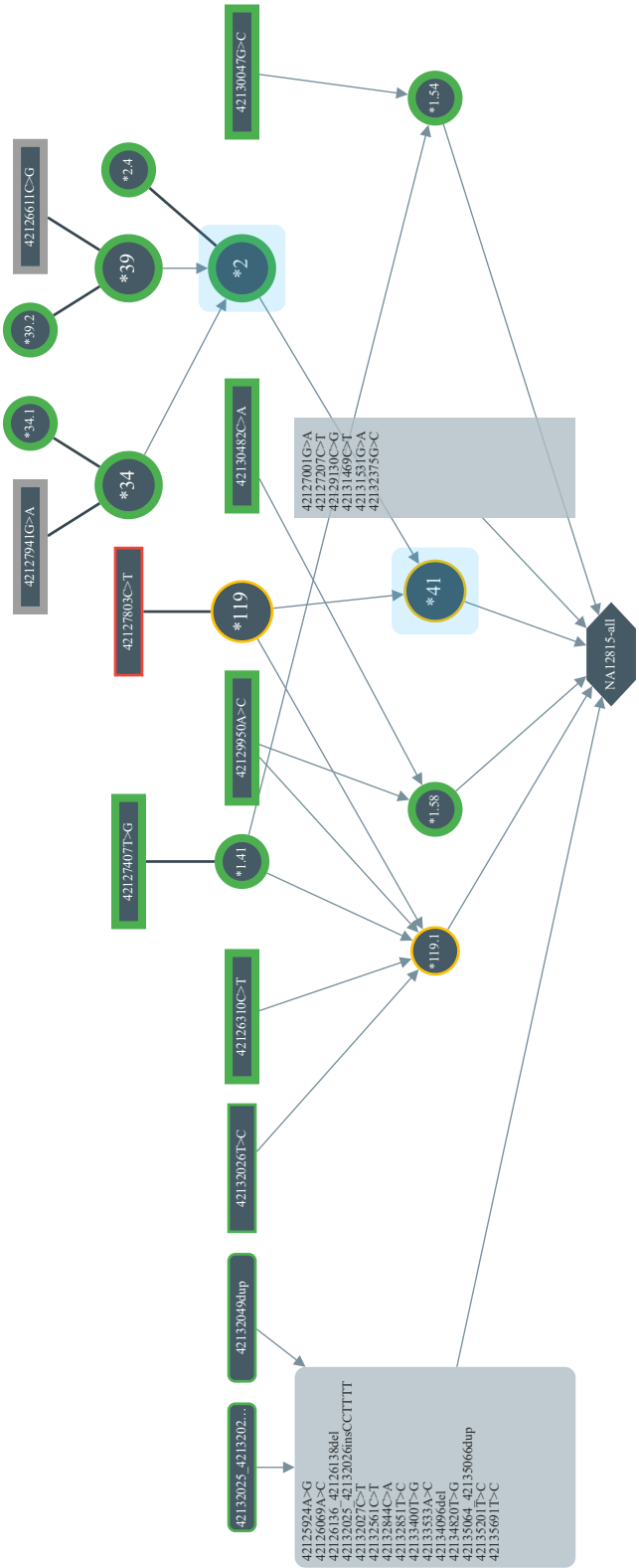


Figure 15: The call graph of `NA12815-all`.

Due to the transitive property of the containment relation, the ancestors of the ignored star allele are also contained in the diplotype but indirectly because of the transitive reduction. Ignoring a star allele thus involves replacing it with its direct ancestors in the call graph. As a result, other star alleles are directly contained in the diplotype and therefore a state with new matches is created. In the same way, we can replace a star allele with an equivalent star allele. We cannot replace star alleles with their overlapping star alleles as this relation is not transitive. As an example, we ignore *CYP2D6**65.001 by replacing it with *CYP2D6**65 which results in the new state {*2.002, *10.004, *65}. States consist only of star alleles and therefore variant allele are not included in the new state in which case the state covers fewer variants of the diplotype.

We recurse and for each state find all valid calls and create new states by ignoring star alleles. For instance, in the previous state {*2.002, *10.004, *65} we can ignore both *CYP2D6**2.002 and *CYP2D6**65. *CYP2D6**2.002 has the ancestor *CYP2D6**2, and *CYP2D6**65 has the ancestors *CYP2D6**2 and *CYP2D6**10. *CYP2D6**10 is contained in *CYP2D6**10.004 and therefore *CYP2D6**10.004 is the closer star allele and included in the state. Furthermore, we only allow the heterozygous allele *CYP2D6**2 to be present once in the state. The new state is thus {*2, *10.004} which yields the call *2/*10.004. In general, ignoring star alleles can result in states with more star alleles since a star allele may have multiple ancestors. The number of star alleles in a state does not explode in practice since many star alleles are contained in multiple star alleles. For instance, we saw that *CYP2D6**10 was not included in the state {*2, *10.004} since it is contained in *CYP2D6**10.004. Another scenario is that a star allele has no star allele ancestors, such as *CYP2D6**39. Ignoring these terminal star alleles thus results in a state with fewer star alleles. The recursive process of ignoring combinations of star alleles results in a search tree of states that yields all possible calls. The recursion stops at the empty state when all star alleles have been ignored which yields the default call *1/*1.

Of the valid alternative calls, we prefer the calls with star alleles that cover the most variants of the diplotype, i.e., are the most specific. In the structure of a call graph, these are the star alleles with the lowest distance to the diplotype. For instance, the allele *CYP2D6**10.004 has a height of 1 as it is directly connected to the diplotype in Figure 16. For *CYP2D6**10, we find the shortest path to the diplotype which is of length 2. We order the alternative calls ascending by the maximum distance of their star alleles. This often results in a tie. In this case, we prefer calls with star alleles that have a lower simple edit distance to the diplotype. We order ascending by the average simple edit distance of the star alleles in a call. The simple edit distance is a measure of the similarity between two star alleles on a sequence level and has a large range of values. In practice, this results in a fixed ordering of the alternative calls. As the number of states is high compared to the number of alternative calls, it is more efficient to generate all calls in some order followed by sorting than to generate them in order of specificity by use of a priority queue.

After finding a state that yields a valid call, ignoring star alleles can only result in less specific calls, and we may not want to include all calls. To illustrate, we consider the state {*2, *10.002} in Figure 16 that yields the valid call *2/*10.002. Replacing *CYP2D6**10.002 with *CYP2D6**10.001 yields the less specific call *2/*10.001. The difference between the contexts of these star alleles includes heterozygous variant alleles, so these may be present in the other haplotype. Indeed, phasing information shows that *CYP2D6**10.002 is not present in either haplotype which shows that less specific calls are sometimes necessary to find the correct call. However, replacing *CYP2D6**10.001 with *CYP2D6**10 leaves out the homozygous variant allele `NC_000022.11:g.42129130C>G`. This variant allele is present in both haplotypes, and we are removing certain information. Therefore, we want to filter out calls that are less specific than another call when the difference in contexts consists only of homozygous variant alleles. An

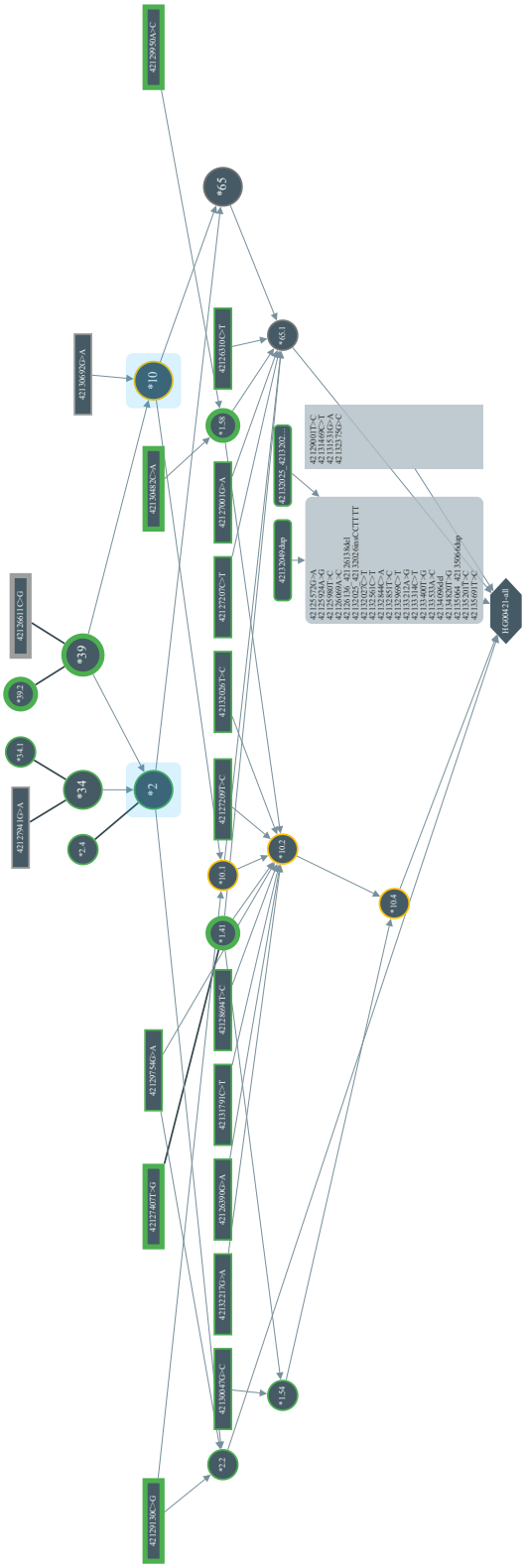


Figure 16: The call graph of `HG00421-all`.

observation is that the same state is reachable in multiple ways. In Figure 16, we can reach the state {*2, *10} from the state {*2, *10.001} and from the state {*2, *10, *65}. We found no order of generation that allows for filtering out these less specific calls in all cases. Therefore, we do not filter by a stop condition in the recursion, but rather mark these calls and filter them out after generating all calls.

In the same way, we filter out calls that are less clinically significant than a another valid call. This is illustrated in Figure 17. The state {*21} yields the valid call *1/*21. *CYP2D6**21 is annotated as "no function" and has the ancestor *CYP2D6**2 with an annotation of "normal function". Ignoring *CYP2D6**21 thus results in a less significant call that does not cover the variant allele `NC_000022.11:g.42128218dup`. As this variant allele potentially impacts the phenotype, and we have a call that includes this variant allele, we filter out the call *1/*2. In the case of star alleles without a functional annotation we do not know their clinical significance and therefore do not filter out calls based on this.

Third, the recursion always terminates in the empty state. However, when the call graph includes a star allele that consists of a single variant allele we know that this star allele is included in a haplotype. In Figure 16, the star alleles *CYP2D6**34 and *CYP2D6**39 are equivalent to single variant alleles. We can therefore filter out the default call *1/*1. Similarly, we can filter out the call *1/*34 since we know that the homozygous *CYP2D6**39 is included in both haplotypes.

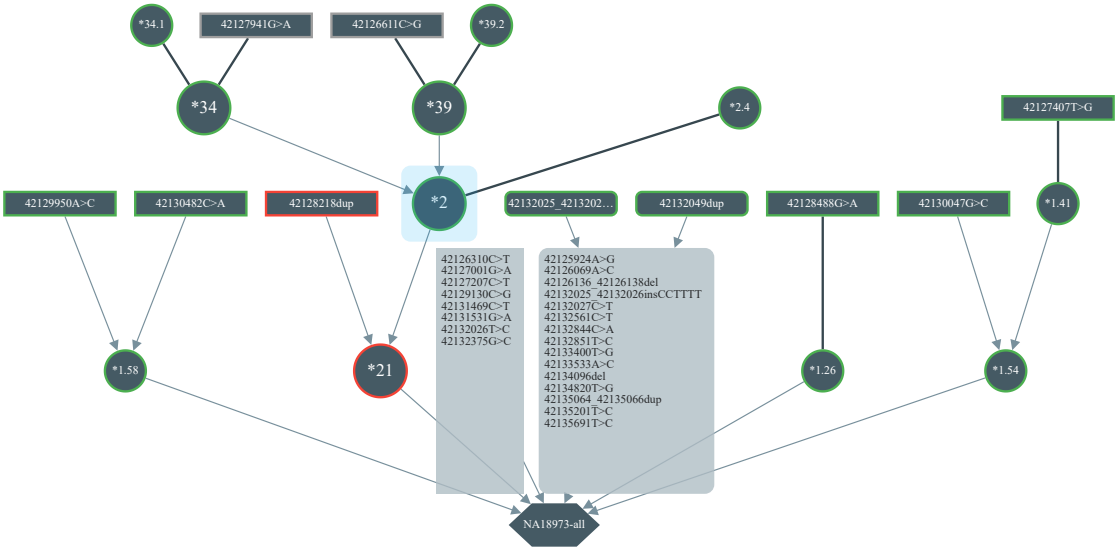


Figure 17: The call graph of `NA18973-all`.

The calling method is based on the idea that the closest star alleles in a call graph are the ones that best describe the haplotypes. We extended this method in this section for missing phasing information by finding the closest valid combination of star alleles that describe a diplotype and ignoring star alleles that may not describe either haplotype. This works well for the majority of call graphs, but not all. In Figure 15 for example, the star allele *CYP2D6**2 is contained in the star allele *CYP2D6**41. Since ignoring *CYP2D6**41 is done by replacing it with *CYP2D6**2, and *CYP2D6**119, the state {*2, *41} which yields *2/*41 is not considered. However, this is a valid call since the homozygous star allele *CYP2D6**2 is included in both haplotypes. When a star allele has a homozygous star allele ancestor in the call graph, we need to consider the

possibility that both are matches. In addition to the recursion of the initial state described above, we also try to find a state describing one haplotype `A` under the assumption that haplotype `B` is one of the star alleles in the initial state. In the previous example, we assume that haplotype `B` is described by *CYP2D6**41 and start with the initial state {*41, *119.001}. At some point in the search tree, the state is {*2} which together with the other haplotype yields the call *2/*41.

The algorithm described here finds the valid alternative calls for data with missing phasing information and only considers haplotypes that are described by known star alleles. The number of calls that the algorithm considers is significantly lower than the number of possible diplotypes possible based on the number of heterozygous variants (see Equation 1). Instead, the number of calls considered depends on the number of star alleles in the call graph, which is usually much lower. This algorithm considers on average around 100 states per diplotype in a benchmark dataset. The largest number of states considered is around three thousand for the diplotype `NA19143-all` which has over two million possible diplotypes (see Section 3.2.2).

Input: $G \leftarrow (A, R)$, a call graph with alleles A and relations R.
Output: *calls*, a list of valid alternative calls.

```
calls ← [ ];
queue ← [ ];
initialstate ← ∅;
for a ∈ A do
    if distance(diplotype, a) = 1 then
        if a is variant allele then
            continue;
        if cores(a) is default then
            continue;
        if a is homozygous then
            initialstate.add(a²);
        else
            initialstate.add(a);
end
queue.push((∅, initialstate));
for a ∈ initialstate do
    if ∃a′ ∈ ancestors(a) : a′ is homozygous then
        queue.push(({a′}, initialstate));
end
while queue is not empty do
    haplotype_B, state ← queue.pop();
    state_hom ← {a² ∈ state};
    state_het ← {a¹ ∈ state};
    for 0 ≤ k ≤ |state_het| do
        for haplotype_A ∈ combinations(state_het, k) do
            haplotype_A ← haplotype_A ∪ state_hom;
            haplotype_B ← haplotype_B ∪ (state_het \ haplotype_A) ∪ state_hom;
            if |cores(haplotype_A)| > 1 ∨ |cores(haplotype_B)| > 1 then
                continue;
            if ∃a ∈ state_hom : a is not homozygous then
                continue;
            variants_{x∈{A,B}} ← {a ∈ context(haplotype_x) : a is variant allele};
            if variants_A ∩ variants_B ∩ state_het ≠ ∅ then
                continue;
            if |haplotype_{x∈{A,B}}| = 0 then
                haplotype_x ← default;
            calls.append({haplotype_A, haplotype_B});
        end
    end
    for 1 ≤ k ≤ |state| do
        for ignore ∈ combinations(state, k) do
            replace ← {a ∈ ancestors(ignore) : a is star allele ∧ a ∉
              context(state \ ignore)};
            queue.append((haplotype_B, (state \ ignore) ∪ replace));
        end
    end
end
sort calls ascending by max(a ∈ call : distance(diplotype, a)) and average(a ∈ call :
  edit(diplotype, a));
filter calls where ∀a ∈ (context(call) \ context(call′)) : a is homozygous variant allele;
filter calls where max(a ∈ call : significance(a)) < max(a ∈ call′ : significance(a));
filter calls where call is default² ∧ ∃a ∈ A : |ancestors(a)| = 1;
filter calls where default ∈ call ∧ ∃a ∈ A : a is homozygous ∧ |ancestors(a)| = 1;
```

Algorithm 1: Finding alternative calls from data with missing phasing information.

4 Results

In this section, we test the star allele calling method. All star alleles, variant alleles and functional annotations are retrieved using the PharmVar API (`https://www.pharmvar.org/api-service`, version 6.0.2).

4.1 Simplification of *CYP2D6* star allele relations

In PharmVar version 6.0.2, there are 159 core alleles without structural variants for the *CYP2D6* gene. Four star alleles with structural variants are described as hybrid alleles. *CYP2D6**5 is a gene deletion. These 159 core alleles are defined by 150 unique variants. If we include suballeles the database has 511 star alleles and 403 unique variants. Since each pair of variants has a relation between them, there are 835,396 relations in theory The breakdown of the different relation types is shown in Table 1. The total number of relations found differs from the theoretical number of relations as one star allele in PharmVar is uninterpretable. The definition of *CYP2D6**4.030 includes variants that overlap in position, and it is thus unclear which sequence was observed (see Appendix B). This star allele and one of its variant are therefore excluded from the analysis. The most common relation is disjoint, followed by overlap, containment and equivalence.

The simplification method described in Section 3.1.1 reduces the number of relations significantly. The number of relations after simplification is also shown in Table 1. The total number of relations is reduced by 99.6%. No disjoint relations are represented in the simplified graph. Additionally, more than half of all relations are removed due to symmetry. All instances of equivalence are cases in which a core allele is equivalent to its suballele. In some of these cases, the core allele is also equivalent to a single variant. There are no other cases of equivalence. As discussed in Section 2.1, this is to be expected in a database of normalised variants. Furthermore, 75.3% of containment relations are pruned. This suggests that much hierarchy exists in the dataset. Indeed, the star allele *CYP2D6**2 is contained in 46 star alleles and *CYP2D6**10 in 19 star alleles. The star allele *CYP2D6**39 is only directly contained in 6 star alleles, but as these include *CYP2D6**2 and *CYP2D6**10 it is indirectly contained in many more. Finally, 99.6% of the overlap relations are pruned.

Table 1: The occurrence of different relations in the dataset of *CYP2D6* star alleles and associated variants before and after simplification of the graph.

Relation	**Count before**	**Count after**
Equivalence	1,260	135
Containment	13,174	1,630
Overlap	93,758	174
Disjoint	723,552	0
Total	831,744	3,569

The simplified graph representation of all PharmVar star allele relations is not very useful for data exploration. It is outside the scope of this thesis to develop a comprehensible layout of the entire dataset. In practice, we are often interested in the relations of a single star allele. For instance, we visualise only the context of haplotypes in call graphs.

4.2 Calling on a benchmark dataset

In this section, the star allele calling method is tested using a benchmark dataset. The dataset was compiled by the GeT-RM project [25]. It describes the consensus call for the gene *CYP2D6* of 179 samples. Out of the samples, we select 120 that have perfect phasing, have no structural variations and are interpretable. In the case of possible copy number variation, we select the most likely star allele assuming a diploid gene. The calls for these samples are compared to a priorly available truth set that is displayed in Appendix A.

4.2.1 Calling on phased benchmark data

Our method finds an unambiguous core allele call for all samples in the benchmark dataset. The method calls suballeles but we simply these to core alleles since the consensus calls are described as core alleles. In all 240 haplotypes, we call the correct star allele according to the consensus truth set. The accuracy of the method is thus 100% for this dataset.

A summary of the types of calls encountered is shown in Table 2. The majority of the calls are heterozygous with *CYP2D6**1, followed by heterozygous calls without *CYP2D6**1. What is perhaps unexpected, is that the number of calls with homozygous *CYP2D6**1 is the lowest. For this benchmark dataset, most diplotypes have at least one haplotype described by a star allele.

Table 2: The occurrence of different types of calls on the benchmark dataset.

Calling	**Count**
Heterozygous	44
Heterozygous *CYP2D6**1	42
Homozygous	24
Homozygous *CYP2D6**1	10
Total	120

The frequencies of the most common star alleles in the dataset are shown in Table 3. The population frequencies of the star alleles are also shown [8]. The observed frequencies of star alleles are similar to the population frequencies. This suggests that the benchmark dataset is representative of the population.

We also observe that 8 of the 158 described star alleles cover 82% of the haplotypes in this dataset. An experimental approach that detects the most common star alleles is sufficient for the majority of the population. However, a method based on sequencing can be scaled more easily to include all star alleles.

Next, we observe that the two most frequent star alleles have a “normal function” annotation while the next six have either a “decreased function” or “no function” annotation. The counts of the functional annotations of the called star alleles are shown in Table 4. Nearly half of the star alleles have a “decreased” or “no function” annotation. The majority of the samples has decreased metabolic activity for the *CYP2D6* gene. This is an argument for the importance of star allele calling in general.

The breakdown of the relations that were used to call a haplotype are shown in Table 5. For 50 calls, the haplotype is equivalent to a star allele. In 22 of these calls, this star allele is *CYP2D6**1. While calls by equivalence are the most certain, they are also the most trivial to find. The majority of calls are found by containment. This shows the advantage of using the variant algebra as it is used to find more complicated relations than equivalence. In 31 instances

Table 3: The observed and population frequencies of the most common star alleles in the benchmark dataset are shown. Additionally, the functional annotations of the star alleles are shown.

Star allele	**Annotation**	**Observed frequency**	**Population frequency**
*CYP2D6**1	Normal function	0.26	0.31
*CYP2D6**2	Normal function	0.18	0.18
*CYP2D6**4	No function	0.13	0.08
*CYP2D6**10	Decreased function	0.10	0.09
*CYP2D6**17	Decreased function	0.04	0.05
*CYP2D6**41	Decreased function	0.04	0.06
*CYP2D6**29	Decreased function	0.03	0.02
*CYP2D6**3	No function	0.03	0.00
Other		0.18	0.21

Table 4: The occurrence of different functional annotations in the star alleles of the benchmark dataset.

Functional annotation	**Count**
Normal function	113
Decreased function	59
No function	51
Other	17
Total	**240**

the only contained star allele is a suballele of *CYP2D6**1 and we call *CYP2D6**1. There are 29 instances of haplotype matching multiple suballeles, ignoring suballeles of *CYP2D6**1. All these instances were the same three suballeles of *CYP2D6**4 as shown in Figure 11. There are also five instances in which prioritisation between different core allele matches is needed. In two cases this is between *CYP2D6**10.001 and *CYP2D6**99 like in Figure 12. No overlap relations are used to call haplotypes, meaning that this relation does not help with calling for this dataset. Finally, 11 haplotypes have no matches and are therefore called as *CYP2D6**1. In all cases the variants that these haplotypes consist of are personal.

Table 5: Occurrence of different relations used to call a haplotype.

Relation	**Count**
Equivalence	50
Containment	179
Overlap	0
Disjoint (no matches)	11
Total	**240**

4.2.2 Calling on benchmark data with missing phasing

To compare the performance of our method for missing phasing information to the results in Section 4.2.1, we use the same benchmark dataset. We manually remove the phasing information of this dataset. This introduces seven additional uninterpretable samples. We ignore these samples for the remainder of this section and revisit this issue in Section 6.1. As in the previous section, our method calls suballeles we simplify these to core alleles.

Our method finds a list of alternative calls for each sample We consider the call that is first in the ordering of the calls as the *preferred call*. The preferred call is the most specific description of the diplotype. A summary of the results is shown in Table 6. For the majority of the calls, the preferred call is correct according to the truth set. Our method achieves an accuracy of 94.7% for this benchmark dataset without phasing information. As expected, the accuracy is not perfect since the data is incomplete. However, we still find the correct call in many cases which is useful as much real-world data is unphased. These results show that, in general, phasing information is required to achieve perfect accuracy in calling. In Section 6.1 we discuss how this method may be improved.

There are only three samples for which the preferred call is not correct. This can occur when another call is valid and more specific. For all three samples, the second call in the ordering is the correct call. This suggests that the defined ordering of alternative calls is useful for finding the correct call. One of the samples, `NA12815`, is shown in Figure 15. In this case, the preferred call is *CYP2D6**2/*119 instead of *CYP2D6**2/*41. This may be counterintuitive, but the first call is based on the state {*2, *119.001} which is more specific than the state {*2, *41} since it covers more variants. For both `NA18544` and `NA18572`, the preferred call is *CYP2D6**2/*10 instead of *CYP2D6**2/*41. Again, this is because the method calls suballeles, and in this case {*2.002, *10.001} is the most specific state.

Finally, for three samples all alternative calls are incorrect. This is the case for `NA18973`. The call graph for this sample is shown in Figure 17. The call *1/*21 is the most specific. The method filters out *1/*2 since the star alleles have a less significant functional annotation. However, according to phasing information, `NC_000022.11:g.42128218dup` is part of the haplotype that is described as *CYP2D6**1. It is unlikely that a variant that is annotated as "frameshift" does not have an impact on the protein so this filter step may be useful in general. We discuss this further in Section 6.1.

In both the case of `NA12006` and `HG01190` the correct call *CYP2D6**4/*41 is not found since the *CYP2D6**4/*69 is also found and *CYP2D6**69 with a functional annotation of "no function" is preferred over *CYP2D6**41 with a functional annotation of "decreased function". *CYP2D6**69 is not present in either haplotype and is a match due to the combination of the variants in *CYP2D6**10 and *CYP2D6**41. This illustrates the limitations of calling when phasing information is missing.

Table 6: The number of times that the unphased call matches the phased call is shown. The preferred call is the first call in the ordering of alternative calls.

Call	**Count**
Preferred call	107
Not preferred call	3
Incorrect	3
Total	**113**

The number of alternative calls is generally low. The average number of alternative calls is 2.39 with `NA19174` having the highest number of 19 alternative calls. For 54 samples, there is only one alternative call. Of these, 34 calls are homozygous. For most samples in this dataset, a decision thus has to be made between several alternative calls. Many of these alternative calls arise from the ancestors of *CYP2D6**2 as this is one of the most common star alleles. To illustrate, we consider the `NA12892-all` which is shown in Figure 18. In this case, the most specific call is

*CYP2D6**2/*3 which matches the benchmark call. However, since all variants are heterozygous we cannot exclude the possibility that the call is actually *CYP2D6**3/*39 or *CYP2D6**3/*34. These calls are less intuitive since they contain less common star alleles. In Section 6.1 we discuss whether the calling method should include such calls when a more specific call is also possible. The method filters out the call *CYP2D6**1/*39 in this case *CYP2D6**3 and *CYP2D6**34 must be present in one haplotype.

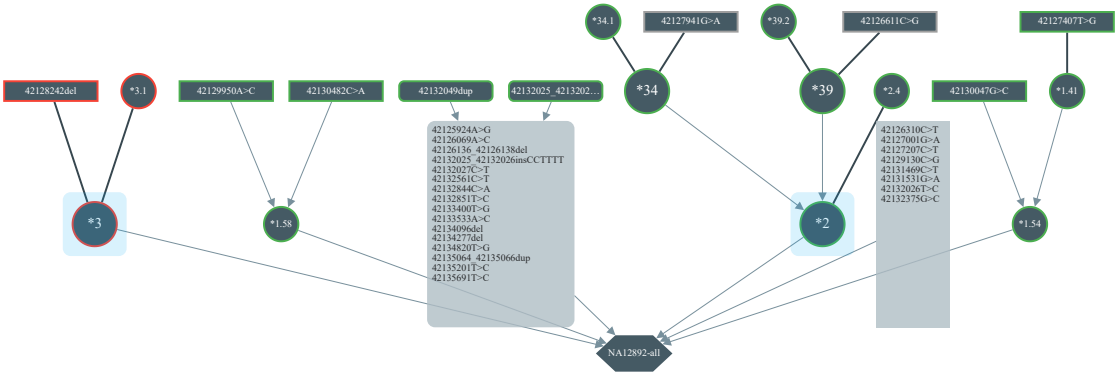


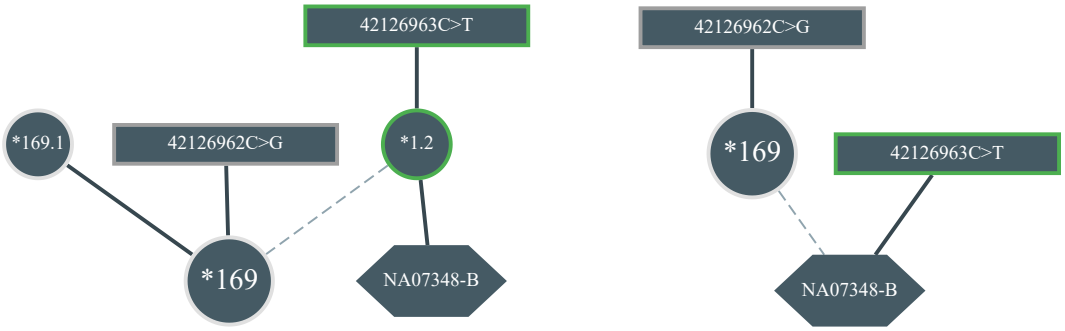
Figure 18: The call graph of `NA12892-all`.

4.2.3 Calling without suballeles

In many contexts, only core alleles are considered and suballeles are ignored. For clinical applications, core alleles are sufficient since they are functionally equivalent to their suballeles. Furthermore, the variants that the suballeles add are neutral and do not affect the phenotype. Finally, considering suballeles makes the problem of calling more complex.

Therefore, a simplified version of the calling method that only considers core alleles is also tested. This method is expected to give the same results since most core alleles are contained in their suballeles. However, this can lead to different conclusions in some cases. For the *CYP2D6* gene, there is one example of this: `NA07348-B`. As seen in Figure 19a, the haplotype matches the star allele *CYP2D6**1.002 which overlaps with *CYP2D6**169. The only match is thus *CYP2D6**1.002 which is simplified to *CYP2D6**1. If suballeles are ignored, like in Figure 19b, the only match is *CYP2D6**169 due to overlap. A second example is shown in Figure 11. If suballeles are left out in this case, the matches are *CYP2D6**4, *CYP2D6**10 and *CYP2D6**74. This is an ambiguous result, requiring the prioritisation scheme. Including suballeles thus helps with calling as all matches are suballeles of *CYP2D6**4.

Generally, suballeles are additional evidence for a call as they are more specific than core alleles. They cover more variants are thus more specific descriptions of the haplotype. Therefore, we do not ignore suballeles in our method.



(a) Considering suballeles. **(b)** Without considering suballeles.

Figure 19: The calling method for the haplotype `NA07348-B` when considering suballeles and when ignoring suballeles. In 19a the context in the call graph is extended as normally an overlap relation of an equivalent star allele is not shown. Ignoring suballeles may lead to a different conclusion as *CYP2D6**1.002 is the only match in 19a, but *CYP2D6**169 is the only one in 19b due to the containment.

5 Discussion

In this section, we discuss the limitations of the method which include the calling of structural variants and incomplete reference data.

5.1 Structural variations

Our method does not call structural variations. Structural variations occur relatively often in *CYP2D6* partially because of the highly similar pseudogenes *CYP2D7* and *CYP2D8*. In general, structural variants can present difficulties for calling star alleles because of the misalignment of sequencing data. Furthermore, structural variations that are not defined on sequence level are not interpretable by the variant algebra. For instance *CYP2D6**5 describes a full gene deletion, but the exact deleted length of this allele is not defined. This description is therefore not interpretable as a set of deletion and insertion operations and cannot be expressed in the variant algebra. Similarly, hybrid star alleles are not always defined on sequence level. One possible solution to this problem is representing these star alleles as the minimal set of variants that all haplotypes share. For instance, we could represent *CYP2D6**5 as a set of deletions that occur in all full gene deletion alleles. This has already been done in the case of *CYP2D6**13 which is a hybrid allele of *CYP2D6* and *CYP2D7*, but is also described as a single insertion of `T` [3]. The problem of hybrid alleles may thus be resolved by increased standardisation of star allele descriptions for structural variations.

It is also possible for individuals to have more than one copy of a gene on the same chromosome. These copies can differ from each other. Multiple copies on the same chromosome often cause misalignment of the reads to the reference sequence. This presents a problem for the described method since variants from multiple copies on the same chromosome can be wrongly detected as homozygous. If the number of times that a variant is observed is reliably determined, our algorithm may be extended to call structural variations (see Algorithm 1). The algorithm can be made more general by requiring haplotypes to be described by $c-1$ star alleles where c is the copy number of the gene. Furthermore, the checking of valid calls can be done more generally by checking if the number of haplotypes that include a variant does not exceed the observed copy number of the variant. In practice, the copy number of variants cannot always be determined

from the sequencing data. More work is needed to determine if this algorithm can be applied to the case of copy number variations.

5.2 Incomplete reference data

The method proposed here is limited by the star allele definitions in the PharmVar database. Only known star alleles are considered in the method, which thus cannot find novel core alleles, but returns a combination of star alleles or *CYP2D6**1. The explainability of the method is useful in this regard as it allows a human to verify the results. For instance, when a haplotype is called as *CYP2D6**1, but it contains many variants, a human may be able to decide that the haplotype is a novel star allele. Since only known star alleles are used in clinical practice this is not a major limitation.

Another issue is that haplotypes often include personal variants. The effects of personal variants are not known and may have an impact on the phenotype of an individual. More work could be done to determine the effect of these variants to ensure that they are not relevant for the calling of a star allele. For instance, a prediction of the protein structure can be made. However, this does not guarantee that the variant is not relevant and that experimental validation is still required. As more variants are discovered and described, this issue will become less prevalent.

6 Conclusions

We have developed a method of calling star alleles from sequencing data. This method makes use of the variant algebra to find relations between the observed haplotypes and known star alleles. The variant algebra is a formal model of reasoning about genetic variation independent of reference data. The relations between observed haplotypes and known star alleles are visualised as a graph. We simplify the graph in several ways to make it more understandable and this removes 99.6% of relations. The simplified graph shows structure in the relations which is used to call star alleles. This visualisation is also used to explain the reasoning behind the call. As data may be ambiguous, explainability is important to allow humans to make the final decision.

Some haplotypes are equivalent to known star alleles, but most often this is not the case as a result of individual variation. We thus want to identify the closest matching star allele. We base this on the star alleles that are contained in the haplotype. The structure of the graph is used to find the closest contained star alleles since some containment relations may be indirect due to transitivity. When there are multiple matches we prefer the star allele with the most significant functional annotation. This method is able to call suballeles which are more specific descriptions of haplotypes than core alleles. We also argue that considering suballeles is important for calling. We validated this method on a benchmark dataset with complete phasing information and no structural variations, this method achieves 100% accuracy. The samples in the benchmark dataset are diverse and *CYP2D6* is one of the most complex pharmacogenes [25]. Therefore, we expect the method to generalise to new data well.

We extend this method to call from sequencing data with missing phasing information. In general, there are multiple alternative calls possible when phasing information is missing. Instead of finding a single match to describe a haplotype, we find two matches to describe a diplotype. The main difficulty is that the diplotype may match star alleles that describe neither haplotype individually. We solve this by recursively ignoring some star alleles and checking if this yields a valid call based on the homozygous and heterozygous variants. On the same benchmark dataset with missing phasing information this extended method achieved 94.7% accuracy.

6.1 Future work

More work should be done on testing this method with different examples. We have validated the method on a benchmark dataset, but more testing is needed to ensure that the method does not overfit on this dataset. Examples of different genes should also be tested to ensure that the method does not overfit on *CYP2D6*. This is outside the scope of this thesis.

Furthermore, with more data we can explore the method further. It would also be interesting to see if haplotypes in other datasets exist that only overlap with a star allele since we found no such examples in the benchmark dataset. Overlap between two alleles means that they are not disjoint, but in general characterization of overlap relations is complex. Overlap relations are considered in the method, but we are not able to validate this part of the method. More analysis is needed to determine the full effect of some decisions made in the method. For instance, the alternative calls often include calls such as *CYP2D6**39/*65 and *CYP2D6**3/*34 which are less specific than some other callings, but cannot be excluded based on the data. Such calls are less intuitive since they contain a less common star allele. In the case of *CYP2D6**65, the call is not clinically actionable since this star allele has no functional annotation. Future work should evaluate whether these calls can be useful in practice.

The basis of the calling method is a method of simplifying the relations in a graph. The

simplification method should be validated systematically. For instance, by inferring all relations from the simplified relations and comparing the inferred relations to the original relations. This is important to ensure that there is no loss of information due to the simplification. It may also be possible to prove this formally.

There are several ways in which the accuracy of the method for missing phasing information may be improved. All instances in which the method does not find the correct call are the correct call has less significant functional annotations than another call and is filtered out. A possible improvement is to use the functional annotations of star alleles in the ordering of alternative calls instead of filtering them out. However, more research is needed to determine how functional annotation should be used in the ordering of alternative calls.

Next, the calling method considers samples that are either perfectly phased or samples for which it is only known whether variants are homozygous or heterozygous. However, in practice data is often partially phased meaning that it is known for some variants whether they are *in cis*. The method can be improved by taking this information into account. Based on partial phasing information, it may be possible to prune alternative calls. This improves the interpretability of the results and can lead to higher accuracy.

We found that not all haplotypes in the GeT-RM dataset were interpretable. For example, `NA18526-A` is not interpretable since it contains the variants `NC_000022.11:g.42132027C>T` and `NC_000022.11:g.42132027delinsCT` which cover the same position. We suspect that this is the result of incorrect variant calling, but this is not certain. Samples that are not interpretable are therefore not included in the benchmark dataset.

Finally, work could be done on a more efficient method of generating all relations that makes use of the graph simplifications. For example, if a containment relation is found between *CYP2D6**39 and *CYP2D6**2, and between *CYP2D6**2 and *CYP2D6**65, the containment relation between *CYP2D6**39 and *CYP2D6**65 can be inferred and does not have to be found using the variant algebra. This reduces the number of relations that have to be calculated which is the most time-consuming part of the method.

References

[1] Jesse J. Swen et al. “A 12-gene pharmacogenetic panel to prevent adverse drug reactions: an open-label, multicentre, controlled, cluster-randomised crossover implementation study”. In: *The Lancet* 401.10374 (2023), pp. 347–356.

[2] Magnus Ingelman-Sundberg et al. “Integrating rare genetic variants into pharmacogenetic drug response predictions”. In: *Human Genomics* 12 (2018), p. 26.

[3] Charity Nofziger et al. “PharmVar GeneReview: CYP2D6”. In: *Clinical pharmacology and therapeutics* 107.1 (2020), pp. 154–170.

[4] Jonathan K. Vis et al. “A Boolean algebra for genetic variants”. In: *Bioinformatics* 39.1 (2023).

[5] Ulrich M. Zanger et al. “Cytochrome P450 enzymes in drug metabolism: Regulation of gene expression, enzyme activities, and impact of genetic variation”. In: *Pharmacology & Therapeutics* 138.1 (2013), pp. 103–141.

[6] Nuala A. O’Leary et al. “Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation”. In: *Nucleic Acids Research* 44 (2016), pp. 733–745.

[7] Andrea Gaedigk et al. “Pharmacogene Variation Consortium: A Global Resource and Repository for Pharmacogene Variation”. In: *Clinical Pharmacology and Therapeutics* 110.3 (2021), pp. 542–545.

[8] Michelle Whirl-Carrillo et al. “An Evidence-Based Framework for Evaluating Pharmacogenomics Knowledge for Personalized Medicine”. In: *Clinical Pharmacology and Therapeutics* 110.3 (2021), pp. 563–572.

[9] Adam Auton et al. “A global reference for human genetic variation”. In: *Nature* 526.7571 (2015), pp. 68–74.

[10] Johan T. den Dunnen et al. “HGVS Recommendations for the Description of Sequence Variants: 2016 Update”. In: *Human Mutation* 37.6 (2016), pp. 564–569.

[11] Petr Danecek et al. “The variant call format and VCFtools”. In: *Bioinformatics* 27.15 (2011), pp. 2156–2158.

[12] “Mutation Rates and Gene Location: Some Like It Hot”. In: *PLoS Biology* 2.2 (2004), p. 51.

[13] Muhammad Tariq Pervez et al. “A Comprehensive Review of Performance of Next-Generation Sequencing Platforms”. In: *BioMed Research International* (2022), p. 3457806.

[14] Daniel C. Koboldt. “Best practices for variant calling in clinical sequencing”. In: *Genome Medicine* 12.1 (2020), p. 91.

[15] Medhat Mahmoud et al. “Structural variant calling: the long and the short of it”. In: *Genome Biology* 20.1 (2019), p. 246.

[16] Andrea Gaedigk. “Complexities of CYP2D6 gene analysis and interpretation”. In: *International Review of Psychiatry* 25.5 (2013), pp. 534–553.

[17] Greyson P. Twist et al. “Constellation: a tool for rapid, automated phenotype assignment of a highly polymorphic pharmacogene, CYP2D6, from whole-genome sequences”. In: *Genomic Medicine* 1.1 (2016), pp. 1–10.

[18] Ibrahim Numanagić et al. “Allelic decomposition and exact genotyping of highly polymorphic and structurally variant genes”. In: *Nature Communications* 9.1 (2018), p. 828.

[19] Seung-been Lee et al. “Stargazer: a software tool for calling star alleles from next-generation sequencing data using CYP2D6 as a model”. In: *Genetics in Medicine* 21.2 (2019), pp. 361–372.

[20] David Twesigomwe et al. “A systematic comparison of pharmacogene star allele calling bioinformatics algorithms: a focus on CYP2D6 genotyping”. In: *Genomic Medicine* 5.1 (2020), pp. 1–11.

[21] Matthias Beck et al. “On Weak Chromatic Polynomials of Mixed Graphs”. In: *Graphs and Combinatorics* 31.1 (2015), pp. 91–98.
[22] Alfred V. Aho et al. “The Transitive Reduction of a Directed Graph”. In: *SIAM Journal on Computing* 1.2 (1972), pp. 131–137.
[23] Paul Shannon et al. “Cytoscape: a software environment for integrated models of biomolecular interaction networks”. In: *Genome Research* 13.11 (2003), pp. 2498–2504.
[24] Ugur Dogrusoz et al. “A layout algorithm for undirected compound graphs”. In: *Information Sciences* 179.7 (2009), pp. 980–994.
[25] Andrea Gaedigk et al. “Characterization of Reference Materials for Genetic Testing of CYP2D6 Alleles: A GeT-RM Collaborative Project”. In: *The Journal of Molecular Diagnostics* 21.6 (2019), pp. 1034–1052.

A Benchmark calls

Table 7 displays the consensus calls for *CYP2D6* of 120 samples without structural variants from the GeT-RM project [25].

Table 7: The calls for *CYP2D6* in the benchmark dataset.

Sample	Calling	Sample	Calling	Sample	Calling
HG00111	*3/*3	NA10854	*1/*4	NA18873	*17/*17
HG00276	*4/*4	NA10855	*1/*4	NA18942	*2/*2
HG00337	*2/*22	NA10856	*1/*1	NA18945	*1/*1
HG00373	*2/*2	NA10859	*1/*2	NA18952	*2/*2
HG00421	*2/*10	NA10860	*1/*4	NA18959	*2/*10
HG00423	*10/*10	NA10865	*1/*41	NA18966	*1/*2
HG00436	*2/*71	NA11832	*1/*4	NA18973	*1/*2
HG00463	*10/*10	NA11839	*1/*2	NA18980	*2/*10
HG00589	*1/*21	NA11881	*2/*3	NA18992	*1/*1
HG01086	*1/*31	NA11993	*1/*9	NA19003	*1/*1
HG01094	*1/*31	NA12003	*4/*35	NA19007	*1/*1
HG01108	*2/*106	NA12006	*4/*41	NA19035	*2/*2
HG01190	*4/*41	NA12145	*1/*4	NA19095	*1/*29
HG01680	*28/*59	NA12154	*4/*33	NA19109	*2/*29
HG02373	*10/*14	NA12156	*4/*4	NA19122	*2/*17
HG03225	*10/*10	NA12236	*1/*4	NA19143	*10/*45
HG03246	*43/*43	NA12336	*41/*41	NA19147	*17/*29
HG03259	*106/*106	NA12717	*1/*1	NA19174	*4/*40
HG03619	*2/*113	NA12753	*2/*3	NA19176	*1/*2
HG03643	*2/*7	NA12813	*2/*4	NA19207	*2/*10
HG03703	*1/*99	NA12815	*2/*41	NA19213	*1/*1
HG03780	*1/*112	NA12873	*1/*1	NA19226	*2/*2
HG03781	*2/*99	NA12878	*3/*4	NA19238	*1/*17
HG03882	*1/*112	NA12892	*2/*3	NA19239	*15/*17
HG04206	*2/*113	NA18484	*1/*17	NA19317	*17/*17
NA06989	*9/*9	NA18518	*17/*29	NA19700	*4/*29
NA06991	*1/*4	NA18519	*29/*106	NA19777	*1/*74
NA07000	*9/*35	NA18544	*10/*41	NA19785	*1/*2
NA07019	*1/*4	NA18545	*10/*10	NA19789	*1/*1
NA07029	*1/*35	NA18552	*1/*14	NA19790	*1/*2
NA07048	*1/*4	NA18563	*1/*10	NA19819	*2/*4
NA07055	*4/*4	NA18564	*2/*10	NA19908	*1/*46
NA07056	*2/*4	NA18565	*10/*10	NA19917	*1/*40
NA07348	*1/*6	NA18572	*10/*41	NA19920	*1/*4
NA07357	*1/*6	NA18617	*10/*10	NA20289	*2/*6
NA10831	*4/*4	NA18632	*10/*52	NA20296	*1/*2
NA10838	*2/*4	NA18642	*1/*10	NA20509	*4/*35
NA10846	*1/*4	NA18855	*1/*1	NA20803	*2/*22
NA10847	*1/*41	NA18861	*29/*29	NA20875	*1/*111
NA10851	*1/*4	NA18868	*2/*2	NA21105	*3/*111

B Consistency of the PharmVar database

During the development of the method, several inconsistencies were found in the PharmVar database. These inconsistencies are described here and were also communicated to the PharmVar consortium.

Uninterpretable star alleles Some star alleles are not interpretable by the variant algebra. This occurs when a star allele definition consists of variants that cover the same region. One such instance was found for the *CYP2D6* gene: *CYP2D6**4.030. The definition of *CYP2D6**4.030 consists of, among others, the variants `NC_000022.11:g.42128764_42128772del` and `NC_000022.11:g.42128771T>C`. As a result, we do not know which change occurs at position 42,128,771 as it could be either a deletion or substitution with `C`. One possible interpretation of these overlapping variants is that they describe different placements of the same variants. However, as this is not certain, this star allele is excluded from the dataset.

Additionally, eight star allele definitions initially contained a variant multiple times, making them uninterpretable. These are listed in Table 8. However, as a result of our feedback, these star alleles were updated in PharmVar version 5.2.19.2 to remove the duplicate variants.

Function differs between subalelles The star allele nomenclature defines subalelles of a core allele as being functionally equal. However, in four instances of *CYP2D6* star alleles, this is not the case. These are listed in Table 9. For *CYP2D6**124, which has the annotation "no function", this is because of an uncertain annotation. The suballele *CYP2D6**124.001 has an annotation of "function not assigned". More problematic is *CYP2D6**46, which is assigned "normal function", while its suballele *CYP2D6**46.003 is annotated as "no function". As a result of our feedback, the functional annotation of *CYP2D6**148.001 was updated to "uncertain function" which is the same as its core allele.

Subalelles that do not contain the core allele It is expected that the variant algebra finds that all subalelles are contained in their core alleles since definitions of subalelles must include all variants of their core allele. There are, however, nine subalelles that do not strictly include their core allele. These are listed in Table 10. In these cases, there are alignments possible with the core variants that are not possible with the suballele variants.

Furthermore, star alleles do not always strictly include the variant alleles that are constructed from the individual variants in their definitions. These variants individually have different possible alignments than together in an allele. This is not incorrect, but rather an argument for the use of the variant algebra. Matching a list of variants to an allele definition requires a method that considers the different placements of variants, which the variant algebra does.

Table 8: Star allele definitions that were previously uninterpretable by the variant algebra due to variants occurring twice. These definitions have been updated to remove the double variants in version 5.2.19.2 of the PharmVar database.

Star allele	**Double variant**
*CYP2D6**4.030	`NC_000022.11:g.42128764_42128772del`
*CYP2D6**6.006	`NC_000022.11:g.42131526_42131529del`
*CYP2D6**19	`NC_000022.11:g.42128251_42128254del`
*CYP2D6**19.001	`NC_000022.11:g.42128251_42128254del`
*CYP2D6**19.002	`NC_000022.11:g.42128251_42128254del`
*CYP2D6**35.003	`NC_000022.11:g.42129467del`
*CYP2D6**120	`NC_000022.11:g.42130729del`
*CYP2D6**120.001	`NC_000022.11:g.42130729del`

Table 9: Core star alleles with a function annotation that differs from one of their suballeles. This violates our interpretation of the definitions of the star allele nomenclature. The functional annotation of *CYP2D6**148.001 has since been updated to "uncertain function".

Core allele	**Functional annotation**	**Suballele**	**Functional annotation**
*CYP2D6**46	normal function	*CYP2D6**46.003	no function
*CYP2D6**106	uncertain function	*CYP2D6**106.001	unknown function
*CYP2D6**106	uncertain function	*CYP2D6**106.002	unknown function
*CYP2D6**124	no function	*CYP2D6**124.001	not assigned function
*CYP2D6**148	uncertain function	*CYP2D6**148.001	normal function

Table 10: Suballeles that do not contain their core allele. Instead, these suballeles have the overlap relation with their core alleles.

Core allele	**Suballele**
*CYP2D6**36	*CYP2D6**36.001
*CYP2D6**36	*CYP2D6**36.002
*CYP2D6**36	*CYP2D6**36.003
*CYP2D6**36	*CYP2D6**36.004
*CYP2D6**57	*CYP2D6**57.001
*CYP2D6**83	*CYP2D6**83.001
*CYP2D6**83	*CYP2D6**83.002
*CYP2D6**83	*CYP2D6**83.003
*CYP2D6**141	*CYP2D6**141.001