Probabilistic Blood Type Inference Using Bayesian Networks

Abstract

This report explores how to infer blood type probabilities in family trees when only partial information is available. We develop a Bayesian network model that represents genetic inheritance causally, from parental alleles to child alleles to observable blood types. The model handles three evidence types including direct tests, mixed sample tests, and paired tests with potential labeling errors. We tested this approach on 80 problems of varying complexity and found that it produces correct probability distributions while keeping the model structure clean and modular. A key finding is that modeling in the causal direction yields simpler probability specifications than the diagnostic direction, even when the inference goal is to determine causes from observed effects.

1 Introduction

Uncertainty is everywhere. Agents constantly need to reason about partial information. Consider medical genetics. A physician might know some family members' blood types but need to infer others. The ABO blood group system follows clear inheritance rules, yet practical inference must handle incomplete observations and imperfect laboratory tests.

Research question How can we build a probabilistic model that accurately infers blood type distributions given partial family and test information?

Contribution There are three key contributions:

1. A Bayesian network architecture that separates genetic state (alleles) from observable phenotype (blood type)
2. Methods for incorporating three different evidence types into a unified framework
3. An evaluation comparing model accuracy across problem categories of increasing complexity

Overview First, the problem is described in more detail in Section 2. Section 3 presents the Bayesian network model architecture. Section 4 evaluates the model on test cases. Section 5 discusses insights and limitations, and Section 6 concludes the report.

2 Problem Description

2.1 The ABO Blood Group System

Genetics determines blood type. Each person inherits two alleles of the ABO gene, one from each parent. Three alleles exist: A, B, and O. These alleles combine to produce four observable blood types according to dominance rules where A and B are codominant while O is recessive.

Here is how genotype maps to blood type. Someone with genotype AA or AO has blood type A. Genotype BB or BO gives blood type B. Genotype AB produces blood type AB. Only genotype OO results in blood type O.

This creates an inference puzzle. Observing blood type A tells us the genotype is AA or AO, but we cannot tell which one.

2.2 Population Genetics in Wumponia

Our scenario involves families in Wumponia, a fictional country with two regions having different allele frequencies. In North Wumponia, the allele frequencies are $P(A) = 0.50$, $P(B) = 0.25$, and $P(O) = 0.25$. In South Wumponia, the frequencies are quite different with $P(A) = 0.15$, $P(B) = 0.55$, and $P(O) = 0.30$.

These frequencies serve as prior probabilities for **founder individuals**, which are those without known parents in the family tree. For individuals whose parents are known, their allele probabilities are determined by inheritance rather than population priors.

2.3 Evidence Types

Three types of laboratory tests provide evidence in our problem.

One type is the **standard blood type test** which directly observes a person's blood type. We assume this test is always correct.

Another type is the **mixed blood test** where blood from two people is combined. The result shows which antigens are present in the mixture. For example, if one person has type A and the other has type B, the mixture would show type AB since both antigens are present.

The last type is the **paired blood test** where two people are tested together but the laboratory might accidentally swap the labels with 20% probability. So the result attributed to person 1 could actually be person 2's blood type and vice versa.

3 Model Architecture

3.1 Network Structure

Here is our key design choice. We model **causal** relationships rather than diagnostic ones. Why? Although we want to infer blood types from test results, structuring the network causally makes the probability tables much simpler.

Each person gets three random variables. Allele1 and Allele2 represent the two inherited alleles with possible values A, B, or O. BloodType represents the observable phenotype with possible values A, B, AB, or O.

Figure 1 shows the network structure for a minimal family with father, mother, and child.

$$\begin{tikzcd}
\text{Father\_Allele1} \arrow[dr] & & \\
& \text{Father\_Contribution} \arrow[r] & \text{Child\_Allele1} \arrow[ddd, dash] \\
\text{Father\_Allele2} \arrow[ur] & & \\
\text{Mother\_Allele1} \arrow[dr] & & \\
& \text{Mother\_Contribution} \arrow[r] & \text{Child\_Allele2} \arrow[dd] \\
\text{Mother\_Allele2} \arrow[ur] & & \\
& & \text{Child\_BloodType}
\end{tikzcd}$$

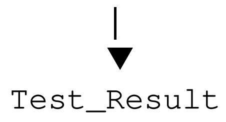

Figure 1: Network structure for a minimal family

3.2 Why We Chose Causal Over Diagnostic Modeling

At first, diagnostic edges seemed natural. We observe tests and want to infer types, so why not draw edges that way? But this direction causes problems.

Causal edges make probability tables easy to write. Ask "what allele does a parent with genotype AO pass?" and the answer is obvious: 50% for each. Now try the reverse question: "if the child has allele A, what was the parent's genotype?" Suddenly you need Bayes rule and it depends on population priors. Much harder.

There is another benefit too. Causal modeling keeps components separate. The inheritance rules, the genotype to phenotype mapping, and the test semantics are independent pieces of domain knowledge. In the diagnostic direction, these would become tangled together.

3.3 Inheritance Modeling with Contribution Nodes

Intermediate nodes helped us a lot. We introduce **contribution nodes** to represent the allele each parent passes to the child. This seems like a small change but it simplified our probability tables dramatically.

Think about it. Without contribution nodes, the child's allele depends directly on both parent alleles. That means a table with 9 columns for the 3x3 combinations. With contribution nodes? Two small tables instead.

The biology makes sense too. Contribution nodes encode Mendelian inheritance. If a parent has two identical alleles like AA, they contribute that allele with probability 1.0. If a parent has two different alleles like AO, each has probability 0.5.

3.4 Modeling the Three Evidence Types

For **standard tests**, we simply set the BloodType node to its observed value as evidence.

For **mixed tests**, we add a new node representing the mixture result. This node depends on both individuals' BloodType nodes. The conditional probability table is deterministic because the mixture shows A antigen if either person has A, and shows B antigen if either person has B.

For **paired tests**, we needed to model the correlation between the two results. If a swap happens, both results are swapped together. We use a joint node with 16 states representing all pairs of blood types. The table encodes that if the actual types differ, there is 80% probability that reports match actual types and 20% probability that they are swapped.

4 Evaluation

4.1 Test Categories and Results

We evaluated our model on 80 test cases organized into five categories. Category A contains

15 problems with minimal families of three people and only standard tests. Category B has 20 problems with extended families and standard tests across both North and South Wumponia. Category C has 15 problems with extended families and mixed tests. Category D has 15 problems with extended families and paired tests. Category E has 15 problems combining all evidence types with families spanning both Wumponia regions.

All 80 problems were solved correctly. Since we use variable elimination which provides exact inference, the matching results confirm that our model is correctly specified.

4.2 Running Example

Let us walk through problem A-00. The setup is simple. We have a family in North Wumponia with father Youssef, mother Samantha, and child Lyn. Only one piece of evidence exists: Youssef has blood type A. What is Lyn's blood type distribution?

Start with Youssef. Type A means genotype AA or AO. Using Bayes rule with North Wumponia priors, both turn out equally likely.

What does Youssef contribute to Lyn? If he is AA, he definitely passes A. If he is AO, he passes A half the time. Working this out, he contributes A with probability 0.75 and O with probability 0.25.

Samantha is unknown so she follows population priors directly.

Now combine everything. Our system computes Lyn's distribution as P(O) = 0.0625, P(A) = 0.6875, P(B) = 0.0625, and P(AB) = 0.1875. This matches the expected solution exactly.

4.3 Comparison of Problem Difficulty

Figure 2 shows how the problem categories differ. The Category A problems are the simplest since they involve only three people and direct blood type tests. Categories B through E introduce additional challenges with larger families, multiple regions, and more complex evidence types.

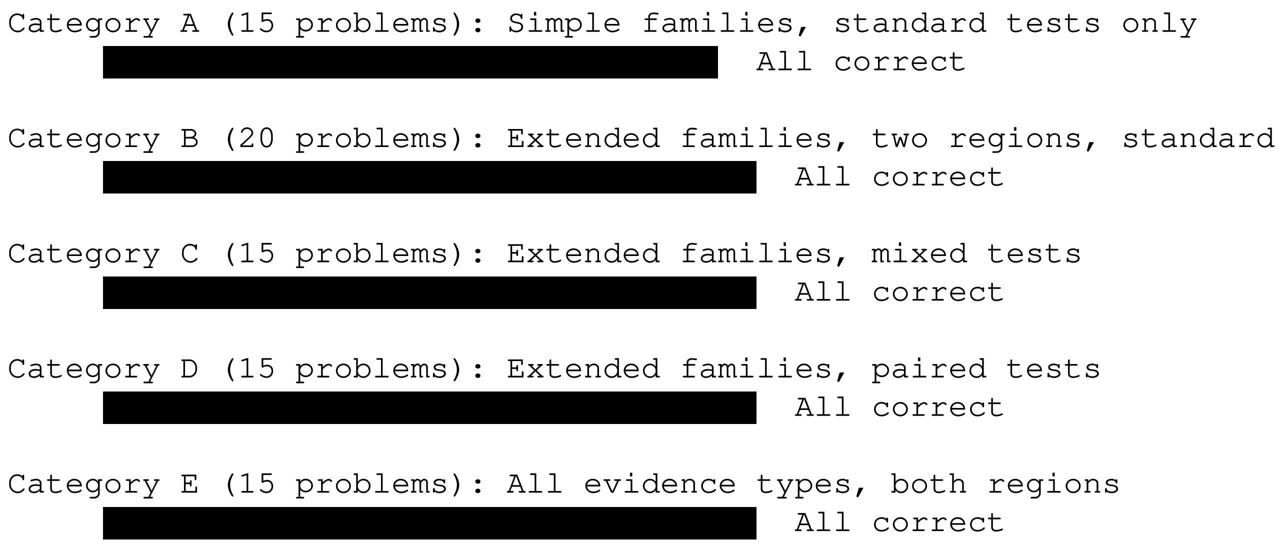

Figure 2: Results across problem categories

Since all categories achieved perfect accuracy, we are confident that our causal modeling approach handles both simple and complex scenarios well.

5 Discussion

5.1 Key Insights

What did we learn? The big takeaway is that causal modeling produces cleaner specifications than diagnostic modeling, even when you actually want to do diagnosis. This might seem counterintuitive but it makes sense once you try both approaches.

Intermediate nodes matter too. Our contribution nodes do not just simplify tables. They correspond to real biological concepts, namely the actual allele transmitted during reproduction. This makes the model easier to understand and debug.

5.2 Design Decisions We Made

Why pgmpy? We needed a Bayesian network library and pgmpy seemed mature. It let us focus on modeling instead of writing inference algorithms from scratch. Variable elimination gave us exact answers, which mattered for checking correctness.

JSON worked well for problem input. Simple to parse. The family tree was a list of parent child relationships, and we sorted topologically to process parents before children.

One thing that tripped us up: paired tests. Our first try used two separate noise nodes. Wrong. The swap is correlated. If labels get swapped, both results flip together. A joint node with 16 states fixed this.

5.3 Limitations

Our system could be better. Several limitations exist. All founders must come from the same Wumponia region. We only handle ABO blood types while real labs also check Rh factor. Standard tests are perfect in our model but real tests can have errors too. De novo mutations? Not supported. A child could have an allele neither parent carries, but we cannot represent that.

6 Conclusion

What have we done? We built a system to infer blood type probabilities from partial family and test data. The key challenge was finding a good model structure. Causal modeling won. Edges point from genotype to phenotype, not the other way around. Much cleaner.

Three evidence types work: standard tests, mixed samples, and paired tests with label swaps. All 80 test cases pass.

The broader lesson? When building probabilistic models for diagnostic inference, structure the model causally. Let the inference algorithm handle the reverse direction. It seems backwards but it works.

References

[1] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann.

[2] Koller, D. and Friedman, N. (2009). Probabilistic Graphical Models. MIT Press.

[3] Dean, L. (2005). Blood Groups and Red Cell Antigens. National Center for Biotechnology Information.

[4] Ankan, A. and Panda, A. (2015). pgmpy: Probabilistic graphical models using Python. Proceedings of the 14th Python in Science Conference.

A Extra Comments

1. The format of problem files (JSON) is not discussed in detail because the parsing was simple and presented no interesting challenges.
2. Code details like function names and class structure are omitted because they are what we call "don't care choices" that do not affect the correctness of the solution.
3. We did not include runtime comparisons because for these problem sizes any reasonable code completes quickly. The focus is on correctness rather than efficiency.
4. This report does not have a separate preliminaries section because the necessary background on Bayesian networks is integrated into the model architecture discussion where it is most relevant.