

Topology and computational hardness of contig variation graphs in Minimum Flow Decomposition

What makes viral genome graphs 'easy' or 'hard' to solve?

Matej Kliment [1]

Supervisor: Jasmijn A. Baaijens

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 25, 2026

Name of the student: Matej Kliment
Final project course: CSE3000 Research Project
Thesis committee: Jasmijn A. Baaijens, Petr Kellnhofer

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Reconstructing viral haplotypes from sequencing reads is essential for understanding viral diversity, evolution, and resistance to treatment. Minimum Flow Decomposition (MFD) provides an effective framework for modeling haplotypes as flow paths in a network flow graph; however, solving MFD remains computationally challenging. In practice, solver runtime varies widely: while many instances are solved within seconds, the variance in the runtime is high. This work investigates which structural properties of contig variation graphs determine their computational hardness in the context of viral genome analysis.

Using benchmark viral assembly graphs, we compute a broad set of topological features, including path-based complexity measures, flow-related attributes, and local structural metrics. We analyze the relationship between these features and MFD solver runtime using both correlation analysis and classification of completed versus timed-out instances. To distinguish correlated factors, we further perform targeted synthetic experiments that isolate individual features, such as maximum flow, while keeping graph topology fixed.

Our results show that runtime is primarily driven by global path-related complexity, particularly the minimum path cover (MPC), a measure closely related to the true number of haplotypes in the sample. Many intuitive structural features, including local node degree and flow magnitude, exhibit limited independent predictive power once path complexity is accounted for. These findings refine previous observations on the exponential dependence of MFD runtime on haplotype count [11] and demonstrate that computational hardness can often be anticipated from graph structure alone. Overall, this work provides practical insight into predicting difficult MFD instances and informs preprocessing and solver strategies for scalable viral haplotype reconstruction.

1 Introduction

1.1 Background

Viral genomes such as HIV and hepatitis C virus do not exist as single uniform genomes within a host but as sets of closely related sequence variants, called haplotypes. These variants arise because viruses mutate quickly and replicate in large numbers. Identifying which haplotypes are present, and in what proportions, is crucial for studying viral evolution, drug resistance, and treatment effectiveness. Modern sequencing technologies produce large numbers of short DNA reads from such genome pools, but these reads are mixed and do not directly indicate which haplotype they originate from [2]. Reconstructing the original haplotypes from this mixture is therefore a challenging computational problem.

A widely used approach models haplotype reconstruction as a network flow problem [1; 7]. In this framework, the graph represents a contig variation graph, where nodes correspond to contigs, assembled sequences derived from overlapping sequencing reads, and directed edges represent feasible adjacencies supported by read alignments or paired-end information [10]. Each edge is assigned an integer flow value reflecting the estimated abundance of that adjacency in the sequencing data. A complete viral haplotype corresponds to a directed source–sink path traversing contigs in genomic order. The reconstruction task then becomes to explain the observed flows using a small set of such paths, leading naturally to the Minimum Flow Decomposition (MFD) problem: decomposing the flow into the minimum number of valid source–sink paths consistent with the data.

To further constrain this problem, additional biological or preprocessing information is often incorporated in the form of subpath constraints, which enforce that certain short paths must appear in at least one reconstructed haplotype [9]. These constraints can reduce ambiguity and improve reconstruction accuracy, but they may also affect solver performance. Understanding how such constraints interact with graph structure is therefore an important part of the overall problem.

While biologically meaningful, the MFD problem is computationally challenging. It is NP-hard, and in practice solver performance varies widely [11]. Integer Linear Programming (ILP) approaches have been shown to improve reconstruction accuracy, but they can suffer from severe runtime issues on more complex graphs. In real datasets, some instances are solved almost instantly, while others fail to finish within reasonable time limits [7].

This inconsistency suggests that graph size alone does not explain computational difficulty. Instead, the structure of the graph plays a crucial role. Graphs that admit many valid paths tend to be harder to decompose, as the solver must consider a larger solution space [4]. Assembly errors, cycles, and the absence of forced segments can further increase complexity. Understanding which structural properties most strongly influence runtime is essential for predicting difficulty and designing more efficient algorithms.

Overall, while Minimum Flow Decomposition provides a strong theoretical framework for viral genome analysis, its practical applicability depends heavily on the properties of the underlying graphs. Understanding why some variation graph instances admit efficient optimal MFD solutions while others are computationally hard is essential for improving the reliability and scalability of these methods in practice, and this is the question addressed in this paper.

1.2 Research Question

How can we characterize the differences between graphs that are computationally simple or difficult to solve in the context of Minimum Flow Decomposition for viral genome analysis? To answer this question, our research focuses on two main problems:

- **Classification:** How can we predict for which graphs will the MFD solver finish and for which it will time-out?

- **Correlation:** Which structural features correlate the most with the MFD solution runtime?

By addressing these questions, this work seeks to improve our understanding of when ILP-based MFD methods are practical and how graph structure influences their performance.

2 Methodology

The goal of this experimental paper is to understand which properties of network flow graphs make Minimum Flow Decomposition (MFD) instances easy or hard to solve in practice. Rather than proposing a new solver, we focus on a systematic empirical analysis of graph structure and solver behavior. Our methodology is designed to measure a set of graph features that are plausibly related to computational difficulty, quantify how strongly these features correlate with solver runtime, and validate the observed relationships using controlled synthetic experiments.

We base our analysis on contig variation graphs derived from viral genome assemblies. These graphs were generated to simulate data occurring naturally in haplotypes of viral genomes and their real reconstruction pipelines and reflect realistic sequencing artifacts such as branching, contig length variations, and ambiguous connections. Studying such graphs allows us to draw conclusions that are directly relevant to viral genome analysis, rather than relying only on abstract or randomly generated networks.

2.1 Structural features analysis

For each graph instance, we compute a collection of structural features that capture different aspects of graph complexity. The selected features are motivated by prior theoretical results on flow decomposition as well as previous practical observations from ILP-based solvers [3]. Specifically, we measure properties related to branching (e.g. node degree, s-t cut size), flow distribution (e.g. maximum flow and abundance values), constraint density (e.g. number of subpath constraints), and solution size (e.g. number of paths in the decomposition). Together, these features provide a versatile description of each graph instance, that analyzes its structure beyond simple graph size metrics.

After these features are computed, we perform a correlation analysis to evaluate how strongly each feature is associated with solver runtime. Because runtime behavior is often fluctuating and can include outliers, we use both linear correlation measures and rank-based methods. This allows us to capture monotone relationships even when exact proportionality does not hold. Instances that fail to complete within the specified time limit of 8 hours are explicitly flagged. These timed-out graphs are then analyzed separately in a second, classification-oriented analysis to investigate whether there exist structural characteristics that clearly distinguish graph instances solved within the time limit from those that exceed it. All analyses are performed independently for each dataset and subsequently compared across datasets to identify features that are consistently indicative of computational difficulty.

2.2 Synthetic graph generation

Correlation alone does not establish causation. To strengthen the found relationships from the feature analysis, we complement the observational dataset analysis with synthetic experiments. Based on the strongest correlating features, we generate artificial flow graphs, starting with a graph instance known to be 'easy', meaning graphs for which the MFD solution is consistently computed within a short time limit, in which we vary only a small number of targeted properties while keeping all others as stable as possible. Starting from a graph that is known to be easy to solve, we gradually increase the value of a selected feature and observe how solver runtime changes. This controlled setup allows us to test whether the identified features directly influence computational difficulty or simply correlate with other factors that increase the graph's complexity in the real data.

2.3 Datasets

We conduct experiments on three datasets of contig variation graphs representing different viral genome assembly types:

1. HIV graphs generated from reads assemblies,
2. HCV graphs generated from reads assemblies,
3. Perfect viral assemblies of generated haplotypes, which serve as reference cases with no sequencing errors

These datasets were chosen to cover a range of graph structures, from relatively simple to highly complex, while remaining biologically realistic. All graphs are directed flow networks with a single source and sink, and edge weights represent estimated abundances. All three datasets include a fair range of graph complexity, starting from 2 haplotype strains up to 7-9 strains on the larger instances. Data points above this limit were largely infeasible for the computational scale of this research. These datasets were used from previous work by Huizenga [7].

2.4 Feature extraction

For each of the graph instances, we compute a diverse set of structural features that capture key aspects of the graph's structure and provide a basis for analyzing which properties influence MFD runtime.

- **Maximal node degree**, the highest number of edges leaving from a single node,
- **Maximal flow value**, the highest flow value through a single node, limited by the node's abundance,
- **Highest number of branches in a possible s–t cut**,
- **Average subpath degree**, average number of all in and out degrees for every subpath,
- **Number of subpath constraints**, number of subpaths assigned to the graph,
- **Paths count**, actual number of paths found by the MFD solver,
- **MPC size**, the minimal number of simple paths from s to t, to cover all edges of the graph, including subpath constraints,

- **MPC size without subpaths**, the minimal path cover, without subpath constraints,
- **Non-overlapping MPC size**, number of non-overlapping paths that cover every node of the graphs

Feature computation is performed using custom Python scripts operating directly on the graph representations. To ensure correctness, these implementations were validated on small graphs with known properties and expected outputs. Scripts for these operations are available in the GitLab repository [6].

2.5 Solver and Implementation

All minimum flow decomposition (MFD) instances are solved using the Flowpaths implementation of an ILP-based MFD solver, with Gurobi employed as the external optimization engine and a standard formulation of the minimum flow decomposition problem. To ensure comparability across experiments, the solver is executed with fixed parameter settings throughout. Subpath constraints are enabled in all runs.

All computations were performed on the Delft AI Cluster. Each experiment was run using 8 threads, with a time limit of 8 hours and a maximum memory limit of 64 GB.

The experimental pipeline, including structural feature extraction, solver invocation, and results collection, is implemented in Python. Core functionality relies on the publicly available Flowpaths Python package, which provides essential MFD-solving routines, together with standard scientific computing libraries. For synthetic instances, graph generation is fully deterministic given a fixed random seed or by a definitive algorithm.

The complete codebase is publicly available in the GitLab repository [6] to support full reproducibility.

2.6 Experimental Procedure and Statistical Methods

For each graph instance, we recorded the runtime of the MFD solver. If the solver failed to terminate within the predefined time limit, the instance was assigned the corresponding cutoff runtime and flagged as timed out. These flagged instances were excluded from correlation-based analyses and instead used in a separate classification study comparing finished and unfinished graphs. Each structural feature was then paired with the recorded solver runtime to evaluate its relationship with computational performance.

To quantify these relationships, we computed both Pearson correlation coefficients (r) and Spearman rank correlation coefficients (ρ) between solver runtime and each structural feature. Pearson correlation measures the strength of linear dependencies between feature values and runtime, while Spearman correlation captures monotonic relationships and is therefore more robust to non-linear trends and the presence of outliers, both of which are common in solver runtime data. Reporting both measures provides a more complete and reliable assessment of how individual graph characteristics relate to MFD solver runtime.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{2}$$

n - number of datapoints, x,y - two datapoints, d - difference between the ranks of corresponding variables ($rank(X_i) - rank(Y_i)$).

All correlation analyses were performed separately for each dataset to account for structural differences between viral assemblies of different samples. The resulting correlations were then compared across datasets to identify features that consistently exhibit predictive power with respect to solver runtime.

3 Results

3.1 Correlation Analysis

In this section, we present the results of the analysis for the relationship between structural graph features and MFD solver runtime using both Spearman rank correlation (Table 1) and Pearson linear correlation (Table 3).

Table 1: Non-linear correlation values for structural attributes of the finished graph instances of HIV, HCV and the Perfect graphs compared with runtime.

Attribute	HIV	HCV	PG
Number of vertices	0.6127	0.3082	0.9164
Number of edges	0.6108	0.3007	0.9168
Num of subpath constraints	0.6200	0.5044	0.9323
Paths count	0.8687	0.0909	0.9046
Non-overlapping MPC size	0.5855	0.0909	0.9184
MPC size	0.7509	0.6478	0.8963
Max node degree	0.6058	0.3548	0.4287
Maximum flow	0.7635	0.6084	0.9059
Max parallel paths	0.6849	0.6478	0.5195
Avg subpath degree	0.4585	0.0560	0.5454
Long-short path difference	0.5703	0.3012	-0.1522
Number of diverging paths	0.5746	0.1961	0.9176

Across all datasets, the perfect graphs (PG) exhibit noticeably higher correlation values for most features, particularly under Spearman correlation. Many attributes show correlations above 0.9 (Table 1), including the number of subpath constraints, MPC size, and number of diverging paths. This indicates a highly systematic relationship between graph structure and runtime in these instances. While this makes the trends very clear, it also suggests that perfect graphs are less representative of real-world data, as their construction lacks the irregularities and noise present in biological samples. Consequently, correlations observed in the perfect graphs likely overestimate how predictable the runtime behavior is in real viral genome analysis.

In contrast, the HIV dataset shows moderate but consistent correlations. Features directly related to paths, most notably the number of paths in the solution and the minimum path cover (MPC) size, exhibit the strongest correlations with runtime. The number of paths has a particularly high Spearman correlation $(0.8687)(1)$ and Pearson correlation $(0.8771)(3)$, making it the single most predictive feature for this dataset. (Keep in mind that this feature is not known prior to the MFD solution and therefore serves mainly as a validation benchmark.) The MPC size shows a strong monotonic relationship with runtime, supporting the view that solver difficulty is primarily driven by the effective number of haplotypes in the sample represented by the graph. These findings align well with previous work suggesting that the number of haplotypes is a key determinant of MFD complexity.

The HCV dataset shows substantially weaker and less stable correlations under Spearman analysis. This is largely explained by the fact that only 12 out of 40 HCV graph instances completed within the 8-hour time limit. The resulting small sample size significantly limits the reliability of rank-based correlations. Pearson correlations for HCV appear artificially high for several features, but this is likely due to the restricted set of completed instances rather than a genuine linear relationship. As such, conclusions drawn from the HCV dataset must be treated with caution.

The maximum flow value displays moderately high correlations with runtime across all datasets, particularly for HIV and perfect graphs. This initially suggests that flow magnitude might be an important predictor of difficulty. However, maximum flow is strongly linked to the number of haplotypes (Figure 1), graphs with more haplotypes require larger total flow to allow integer decomposition across more branching paths. To investigate this dependency, we performed a secondary analysis controlling for the number of haplotypes. This analysis did not reveal a significant independent effect of maximum flow on runtime, though the small number of available data points (ten instances per haplotype count per dataset) limits the statistical power of this result (Table 2). Overall, these findings suggest that maximum flow correlates with runtime mainly as a proxy for path complexity rather than as an independent driver of hardness.

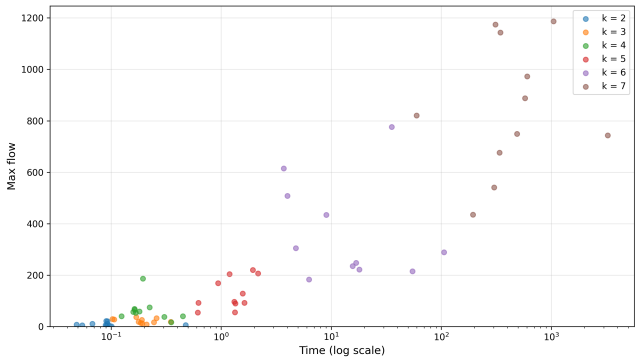


Figure 1: Maximum flow compared to MFD runtime on perfect graphs, with distinct colour for each k number of haplotypes.

Table 2: Non-linear correlation values for Maximum flow of HIV and Perfect graphs compared with runtime separated by number of haplotypes (k).

k	2	3	4	5
Max flow (HIV)	0.5030	0.3571	0.6833	0.356
Max flow (PG)	-0.2838	-0.3690	-0.2825	0.4875

Several other features turn out to be poor predictors of runtime in practice. In particular, the maximum node degree shows relatively low correlation across all datasets. This is explained by the low variance of this feature, in nearly all instances, the maximum node degree only varies between the values 2, 3, or 4, making it insufficiently expressive to distinguish different instances in our datasets. Similar limitations apply to features such as average subpath degree and long–short path difference, which exhibit weak or inconsistent correlations. Overall, the correlation analysis highlights that global, path-based features, especially path count and MPC size, are the most reliable indicators of MFD runtime in realistic datasets. In contrast, many intuitive local or size-based metrics either correlate only indirectly with runtime or fail to provide meaningful predictive power due to low variance or dataset-specific artifacts.

3.2 Classification study on timed-out instances

In addition to correlation analysis on completed runs, we perform a classification-oriented study that contrasts graph instances solved within the 8-hour time limit with those that timed out. The goal of this analysis is to determine whether there exist structural characteristics that clearly distinguish computationally hard instances from tractable ones, beyond what can be explained by continuous runtime variation.

Dataset observations

At a high level, the three datasets exhibit markedly different timeout behavior. In the perfect graph dataset (Figure 2), the majority of graphs with 8 or 9 haplotypes fail to complete within the time limit, while instances with fewer haplotypes are almost always solved. This again highlights the strong dependence of solver difficulty on the number of haplotypes and reinforces the highly systematic nature of this dataset.

The HCV dataset shows the most extreme behavior with only 12 out of 40 graph instances complete within 8 hours, and these are exclusively graphs with 2 or 3 haplotypes. As a result, most higher-haplotype instances are timed out, leaving limited data for meaningful separation of finished and unfinished classes based on finer structural properties.

In contrast, the HIV dataset exhibits a more heterogeneous pattern. Although most timeouts occur in graphs with 5 haplotypes, there are also completed instances at the same haplotype count, as well as occasional timeouts for graphs with 3 or 4 haplotypes. This makes the HIV dataset particularly informative for classification, as it suggests that factors beyond haplotype count contribute to computational hardness.

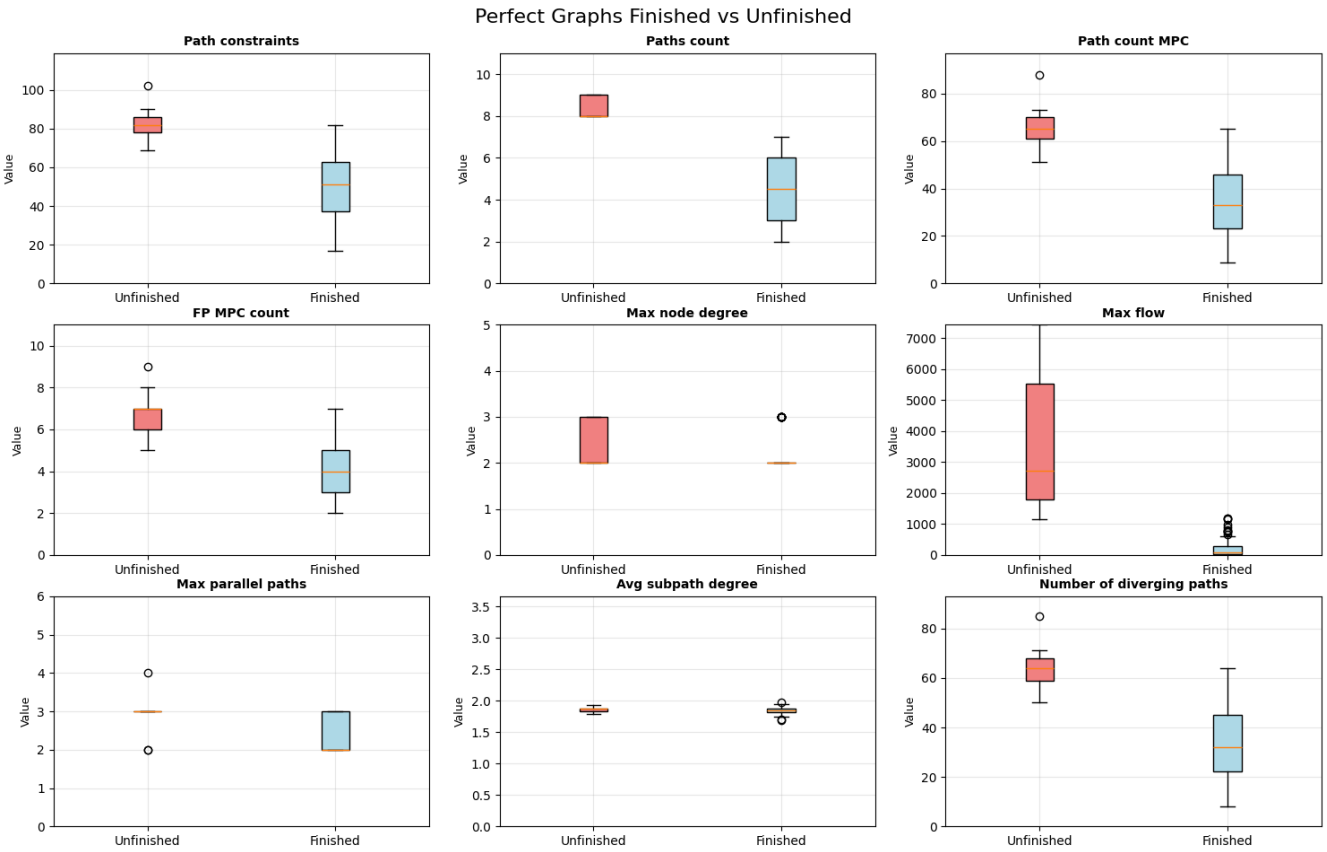


Figure 2: Individual feature values of Perfect graphs split into solved ”finished” and timed-out ”unfinished” instances.

Feature Distributions for Finished vs. Unfinished Instances

Boxplot comparisons of finished and unfinished instances (Figure 2) provide further insight. In the PG dataset, several features show clear shifts in their distributions for timed-out graphs. Path-related attributes, including the number of paths in the solution, MPC size, and non-overlapping MPC size, have substantially higher values for unfinished instances. This separation largely mirrors haplotype count (with unfinished instances corresponding to 8 or 9 paths and finished ones ranging from 2 to 7) and therefore does not constitute an independent distinguishing factor.

Other features show weaker separation. The number of subpath constraints has higher median values for unfinished instances, but the ranges overlap significantly. Maximum flow values exhibit a clearer difference, with unfinished graphs typically having much higher flow values than finished ones. Nevertheless, as discussed earlier, maximum flow is strongly coupled with haplotype count and is unlikely to be an independent driver of timeouts. Features such as maximum node degree, maximum parallel paths, and average subpath degree show substantial overlap between finished and unfinished instances and offer little discriminatory power.

For the HIV dataset, the small number of unfinished graphs limits the strength of conclusions. Most feature values for timed-out instances fall within the range of finished ones, differing mainly in having slightly higher medians and reduced variance. One notable exception is the number of subpath constraints, where unfinished instances occupy a narrower but higher range compared to the much broader distribution observed for finished graphs. While suggestive, this pattern is not sufficient on its own to reliably separate the two classes.

The HCV dataset exhibits similar behavior but with an even stronger imbalance between finished and unfinished instances. Due to the dominance of timeouts, most features show wide ranges for unfinished graphs, making it difficult to identify clear separation without controlling for haplotype count.

Classification Conditional on Haplotype Count

To assess whether any features can distinguish hard instances independently of haplotype count, we repeat the analysis while fixing the number of haplotypes. Due to dataset limitations, this analysis is feasible only for a single value of k per dataset: $k = 7$ for PG, $k = 5$ for HIV, and $k = 2$ for HCV.

Within these fixed-k subsets, feature distributions between finished and unfinished graphs largely overlap. While some attributes, such as maximum flow or number of diverging paths, show slightly higher median values for unfinished instances, no single feature exhibits a clear threshold separating the two classes. This suggests that once haplotype count is controlled for, remaining structural features have limited

ability to independently predict whether an instance will time out.

Overall, the classification study reinforces the central finding of this work: solver timeouts are primarily driven by haplotype-related complexity rather than by secondary structural features. While some attributes exhibit weak shifts between finished and unfinished instances, these effects are generally small, dataset-dependent, and strongly confounded by haplotype count. As a result, classification of hard instances without considering path complexity remains challenging, particularly in realistic biological datasets.

3.3 Synthetic Evaluation of Maximum Flow

The correlation analysis suggested that the maximum flow value might be related to MFD runtime. As maximum flow is strongly coupled with the number of haplotypes and overall path complexity, it remained unclear whether it has an independent impact on solver performance. To investigate this, we conducted a small-scale synthetic experiment designed to isolate the effect of maximum flow while keeping all other graph properties unchanged.

We selected a graph instance from the HIV dataset with two haplotypes that is known to be solved quickly by the MFD solver. Starting from this base instance, we generated a set of ten synthetic graphs by increasing the abundance values along a single simple path by a constant factor $c = 500$. This modification increases the total and maximum flow of the graph without altering its topology. As a result, maximum flow is the only feature that varies across the generated instances.

The solver runtimes for these synthetic graphs are shown in the scatter plot in Figure 3, where runtime is plotted against maximum flow. For the first few instances, we observe an increase in runtime as the maximum flow increases. However, this trend does not persist. After the third instance, all remaining graphs are solved within approximately 30 seconds, despite continuing increases in maximum flow. Notably, these runtimes are substantially lower than that of the original base graph, which required approximately 100 seconds to solve.

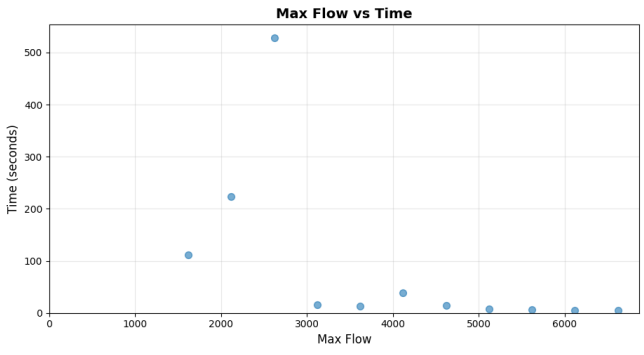


Figure 3: Time against Maximum flow on synthetically generated graph instances.

A plausible explanation for this behavior is that, once the flow values exceed a certain threshold, the solver's internal branching and constraint propagation are no longer sensitive to the absolute magnitude of the flow. Since the decomposition structure remains unchanged and the number of feasible paths is fixed, the solver is able to quickly identify a valid decomposition regardless of how large the flow values become. The initial increase in runtime may reflect solver overhead related to handling larger integer values, but this effect appears limited and does not scale with further increases in flow.

Overall, this synthetic experiment provides evidence that maximum flow alone is not a primary driver of MFD runtime. Instead, its correlation with runtime in real datasets is likely indirect, arising from its association with haplotype count and path complexity. These findings support the conclusion that path-based structural features, rather than flow magnitude, are the dominant factors influencing computational difficulty in MFD.

4 Discussion

4.1 Interpretation of the results

The results of the correlation and classification analyses consistently indicate that path-related complexity is the dominant factor influencing the runtime of Minimum Flow Decomposition in viral variation graphs. Features closely tied to the number of haplotypes, most notably minimum path cover (MPC) exhibits the strongest and most stable correlation with runtime across datasets. In contrast, many local or size-based features, such as maximum node degree, show limited predictive power due to low variance or weak association with global solution structure.

The classification study further emphasizes this conclusion. Instances that timed out within the 8-hour limit are overwhelmingly concentrated at higher haplotype counts, particularly in the perfect graph and HCV datasets. Although some secondary features show shifts between finished and unfinished instances, these effects largely disappear when the number of haplotypes is controlled. This suggests that, beyond path multiplicity, no single structural attribute provides a reliable, independent criterion for separating easy and hard instances in the studied datasets.

The synthetic experiments isolating maximum flow provide additional support for this interpretation. Although maximum flow correlates with runtime in real data, controlled manipulation of this feature alone does not lead to sustained increases in solver time. This suggests that flow magnitude itself is not a primary driver of hardness, but instead closely reflects the underlying path complexity.

4.2 Conclusion

This study shows that the computational hardness of Minimum Flow Decomposition in viral variation graphs is dominated by global path-related complexity rather than local structural or flow-based properties. In particular, the minimum path cover (MPC) emerges as a strong and computable proxy for the effective number of haplotypes and provides a reliable indicator of solver runtime. While features such as maximum flow or subpath constraints may correlate with runtime, their effects are largely indirect and strongly confounded by path complexity. These findings suggest that difficult MFD instances can often be anticipated from graph

structure alone, enabling better preprocessing, solver selection, and scalability in viral haplotype reconstruction.

4.3 Relation to previous work

These findings are consistent with prior observations in the literature. Huizenga suggested that MFD runtime grows exponentially with the number of haplotypes, and our results empirically confirm this behavior across both real and synthetic datasets [7]. Importantly, this work refines that observation by demonstrating that graph-theoretic aspects such as MPC can serve as effective predictors of runtime without requiring a full MFD solution.

More broadly, the results align with studies on graph theory showing that network optimization problems become harder as the number of feasible decompositions increases, rather than as local connectivity grows [11]. Features reflecting global ambiguity, such as cut capacity or branching, appear more relevant than local degree-based metrics. Our analysis supports this perspective in the specific context of viral genome analysis, where biological constraints impose structured but highly variable graph topologies.

Further, confirming this exponential behavior, our study continues by identifying structural proxies for solution size. In particular, the non-overlapping definition of minimum path cover (MPC) consistently correlates with runtime almost as strongly as the actual number of paths produced by the solver. Since MPC is a graph-theoretic property that can be computed without solving the MFD instance, this finding suggests that computational difficulty can often be anticipated in advance. This helps explain why graphs of similar size may behave very differently in practice, a phenomenon that has been observed but not previously quantified [7].

4.4 Connection to Graph Theory and Complexity

From a graph-theoretic perspective, several studies have shown that network optimization problems become harder when graphs have many alternative feasible solutions or lack forced structure [8]. Our findings align with this concept. Features such as the maximum s–t cut, which reflect the capacity for abundance (flow) division, show a moderate but consistent relationship with runtime, indicating that global ambiguity in routing contributes to solver difficulty.

At the same time, local measures such as maximal node degree prove to be weak predictors of runtime on their own. This highlights an important distinction, while branching is a necessary condition for multiple haplotypes, it is the global arrangement of edges into full source–sink paths that determines difficulty. Metrics like MPC capture this global structure more effectively than local degree-based features, explaining their stronger predictive power.

Subpath-related features further illustrate this point. Graph theory suggests that forced or "safe" substructures reduce solution space by decreasing the number of feasible decompositions [5]. Consistent with this, we observe that the presence of subpath constraints does not increase runtime and may even reduce it slightly by eliminating symmetric or redundant solutions. This supports the view that certain structural constraints simplify, rather than complicate, MFD instances.

4.5 Future Work

With additional time and data, several extensions of this work would be valuable. First, the set of studied features could be revisited and expanded to include more expressive measures of global structure, such as attributes for quantifying path diversity or solver-specific relaxation bounds. Further, a larger dataset would enable a more rigorous evaluation framework, including a clear separation between training and validation sets, allowing predictive models of runtime to be quantitatively assessed rather than explored descriptively.

Future work could also investigate hybrid approaches that combine structural feature analysis with solver feedback, such as early branching statistics or relaxation gaps, to improve runtime prediction and adaptive solver strategies.

4.6 Limitations

This study has several limitations that should be considered when interpreting the results. The datasets are relatively small, particularly after separating instances that exceed the time limit, which restricts the statistical power of correlation and classification analyses. The 8-hour cutoff, while practical, prevents observation of runtime behavior for the hardest instances and may bias results toward moderately difficult graphs.

Additionally, the perfect graph dataset, while useful for controlled analysis, is less representative of real-world viral assemblies and likely overstates the predictability of solver behavior. Finally, the real datasets themselves are generated through simulation and preprocessing pipelines that may not capture the full variability of biological sequencing data. As such, while the observed trends are robust within the scope of this study, caution is assumed when generalizing these findings to all real-world MFD applications.

4.7 Broader impact

More broadly, this paper contributes to closing the gap between theoretical hardness results for MFD and its observed behavior in real biological data. By linking solver runtime to concrete, measurable graph features, we move toward a more predictive and explainable understanding of MFD performance. This does not only benefit viral haplotype reconstruction process but may also inform the use of flow decomposition techniques in other domains where similar graph structures can be used.

5 Responsible Research

5.1 Ethics

This research focuses on algorithmic and structural analysis of graph-based models for viral genome reconstruction and does not involve human subjects, patient data, or sensitive personal information, therefore posing minimal ethical risks. Insights into viral haplotype reconstruction may indirectly support biomedical research, emphasizing the importance of accurate, unbiased, and transparent computational methods. To ensure reproducibility, all graph metrics, experimental scripts, and synthetic graph generators were implemented using deterministic procedures, and experiments were

run on clearly documented software scripts and solver configurations, available in the linked GitHub repositories [6]. The benchmark datasets used in this study are publicly available, and all pre-processing steps are explicitly described to enable independent replication.

5.2 Use of generative AI and LLMs

Large language models were used exclusively to help with the language refinement, clarity, and wording for editing of the final paper and did not influence the experimental design, data analysis, or interpretation of results.

References

[1] J. A. Baaijens, L. Stougie, and A. Schonhuth. Strain-aware assembly of genomes from mixed samples using flow variation graphs. in Research in Computational Molecular Biology, 2020.

[2] F. H. C. Dias, L. Williams, B. Mumey, and A. I. Tomescu. Fast, flexible, and exact minimum flow decompositions via ilp. Springer International Publishing, Research in Computational Molecular Biology, 2022.

[3] Jorge-Hernandez F, Garcia Chimeno Y, Garcia-Zapirain B, Cabrera Zubizarreta A, Gomez Beldarrain MA, and Fernandez-Ruanova B. Graph theory for feature extraction and classification: A migraine pathology case study. Bio-Medical Materials and Engineering. 2014;24(6):2979-2986. doi:10.3233/BME-141118.

[4] A. I. Tomescu F. H. C. Dias. Accurate flow decomposition via robust integer linear programming. Institute of Electrical and Electronics Engineers (IEEE), 2024.

[5] Daniel Gibney, Sharma V. Thankachan, and Srinivas Aluru. Feasibility of flow decomposition with subpath constraints in linear time. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik, 2022.

[6] https://gitlab.ewi.tudelft.nl/jbaaijens/cse3000-viralmfd.

[7] R. R. Huizenga. Minimum flow decomposition for haplotype-aware genome assembly. Master's thesis, TU Delft, 2025.

[8] K. Kloster, P. Kuinke, M. P. O'Brien, F. Reidl, F. S. Villaamil, B. D. Sullivan, and A. Poel. A practical fpt algorithm for flow decomposition and transcript assembly. Society for Industrial and Applied Mathematics, 2018.

[9] B. Mumey L. Williams, A. I. Tomescu. Flow decomposition with subpath constraints. Institute of Electrical and Electronics Engineers (IEEE), 2023.

[10] B. Paten, A. M. Novak, J. M. Eizenga, and E. Garrison. Genome graphs and the evolution of genome inference. (2017), Genome Research.

[11] B. Vatinlen, F. Chauvet, P. Chretienne, and P. Mahey. Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths. (2008), The European Journal of Operational Research.

A Appendix

A.1 Pearson correlation table

Attribute	HIV	HCV	PG
Number of vertices	0.4490	0.9158	0.3204
Number of edges	0.4453	0.9185	0.3209
Num of subpath constraints	0.5286	0.1094	0.3166
Paths count	0.8771	0.9180	0.3466
Non-overlapping MPC size	0.4078	0.9180	0.3195
MPC size	0.4762	0.9555	0.2632
Max node degree	0.4349	0.6709	0.2396
Maximum flow	0.6323	0.9410	0.5155
Max parallel paths	0.5629	0.9555	0.2494
Avg subpath degree	0.4605	0.1641	0.2452
Long-short path difference	0.4633	0.6018	-0.0308
Number of diverging paths	0.4058	0.9608	0.3165

Table 3: Pearson correlation values for structural attributes of the finished graph instances of HIV, HCV and the Perfect graphs.

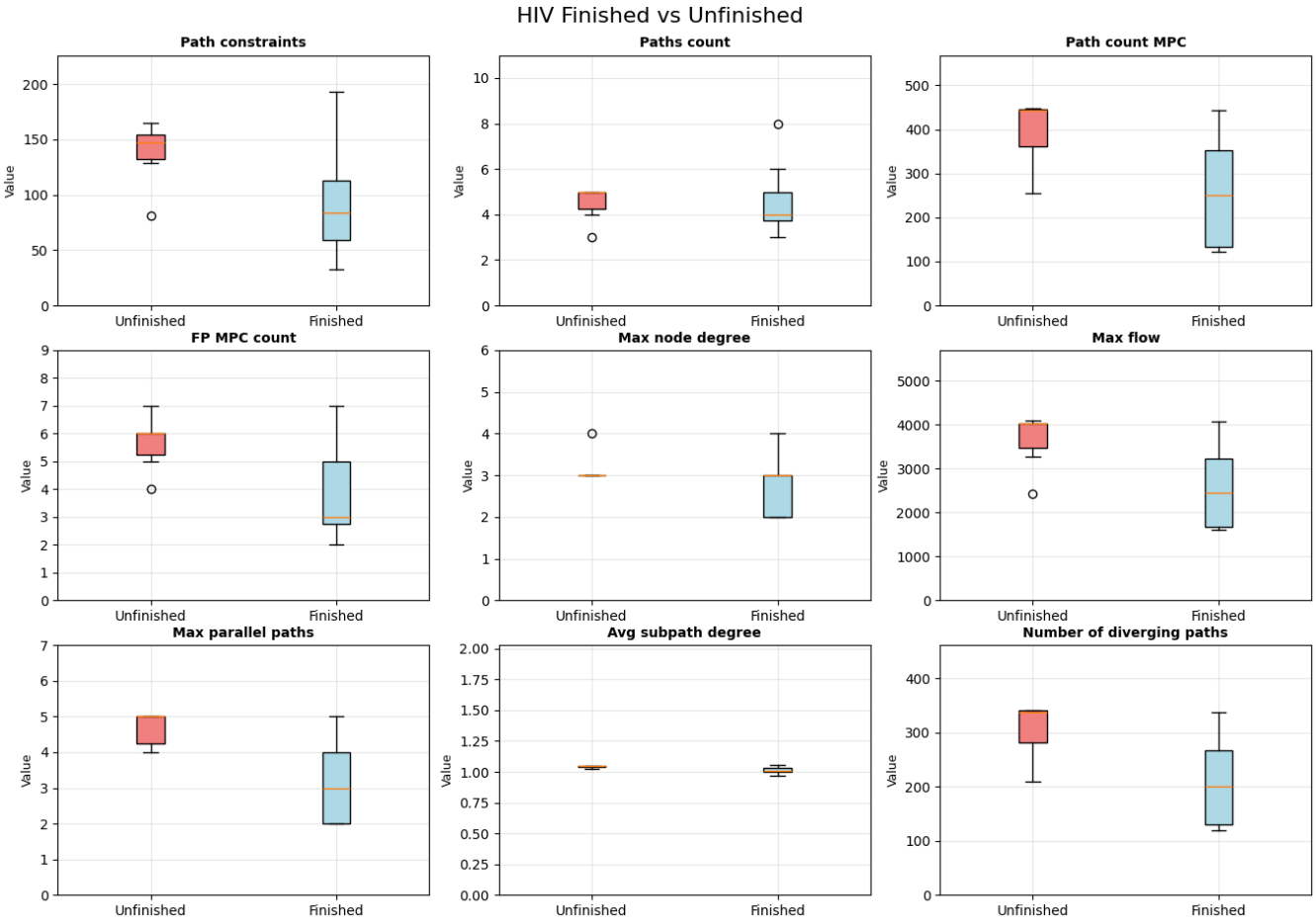


Figure 4: Individual feature values of HIV graphs split into solved ”finished” and timed-out ”unfinished” instances.

A.2 Boxplot for HIV classification