Implementation of the Graphlet Screening Method in Genomic Analysis Using Hail

Grant Proposal

Rishit Avadhuta

Massachusetts Academy of Math and Science at Worcester Polytechnic Institute

Worcester, MA

Table of Contents

Executive Summary

This paper describes a project that implements an effective linear regression technique (Graphlet Screening) into genomic analysis using the Hail Python package, an efficient genomic data parser and storage method. Improving linear regression in genomic analysis is crucial as it defines the correlation between genomic variations (SNPs, as in Single-Nucleotide Polymorphism) and phenotypic traits. Identification of these correlations is the main objective of Genome-Wide-Association Studies (GWAS). GWAS helps to identify and treat genetic disorders, among other utilities. Existing methods of regression, such as L_0-regularization or LASSO, typically struggle with genomic data for three reasons, despite their popularity: lots of different SNPs contribute to one single trait, only a small subset of the genome corresponds to a trait (rare), and each SNP that contributes might only contribute to a very trivial part of the key phenotype (weak), creating datasets that are weak and rare. Graphlet Screening is better at handling this type of data thanks to its screen-and-clean procedure of penalizing error; however, it has never been implemented in genomic analysis before. This project aims to implement the algorithm by first running an independent test against other regression algorithms, testing the algorithm with real genomic data, transferring the algorithm to Python and integrating it with a high-performance genomic parser (Hail) to form a Python package, and finally testing the final Python package with other genomic data present in the NIH "All of Us" genomic database. Existing preliminary data suggests that the Graphlet Screening method would outperform other regression methods if implemented into a genomic analysis environment, and more experimentation would further quantify the algorithm's benefit and ultimately form a genomic analysis solution that is more accurate and runs faster than traditional methods.

Keywords: graphlet screening, regression, genomics, genomic analysis, python, R, variable selection, correlation, SNPs

Implementation of the Graphlet Screening Method in Genomic Analysis Using Hail

Genomic analysis involves the study of the genome, which contains the entire genetic code for humans. Through various methods of analyzing the genome, significant links between SNPs (single-nucleotide polymorphisms) and phenotypic traits can be made to understand how the DNA within human cells interact with bodily processes, particularly using GWAS (Genome Wide Association Studies). GWAS focuses on the correlations between SNPs and their phenotypes, helping to find genetic variations associated with a disease that could lead to better strategies to treat and diagnose the condition (National Human Genome Research Institute, 2020). A unique challenge within the human genome datasets comes during analysis when correlations must be found between many different SNPs and a given phenotype. In other words, high dimensional variable selection must occur to find SNPs that correlate to phenotypes (Hong et al., 2015). Genomic vector signals within these high-dimensional matrices tend to be rare and weak, which means that there are a few SNPs out of the entire genome that might correspond to a given phenotype (there is a sparse minority of non-zero values). Furthermore, the strength of their correlation is rather weak, making detection hard using traditional models (Jin et al., 2014). These weak signals exist because many SNPs can contribute to a single trait, and there are between 4 to 5 million SNPs (MedlinePlus, n.d.). Typically, a regression model is used to analyze the correlation between two factors by determining non-zero signals in correlated factors; however, many existing regression models are not optimal for data sets with weak and rare data, making them generally unfit for usage in the field of genomic analysis. Additionally, they also tend to be computationally challenging. For example, L_0 penalization is a well-known method for regression and high-dimensional variable selection. However, it is both NP-hard and notoriously challenging computationally. Although there are ways to optimally approximate the goal of L_0, they sacrifice accuracy for performance (Liu, 2016). LASSO is also an often-used method of regression and high-

dimensional variable selection. However, this method is also ineffective as it struggles to penalize smaller coefficients (weaker signals) effectively (Horowitz, 2015).

Graphlet Screening is a relatively novel method of statistically analyzing rare and weak data sets for their correlations to a general response vector (Jin et al., 2014). However, it has never been used in genomic analysis.

	τ_p	6	7	8	9	10
$\vartheta = 0.25$	Graphic Screening	24.7750	8.6750	2.8250	0.5250	0.1250
	UPS	48.5500	34.6250	36.3500	30.8750	33.4000
	lasso	66.4750	47.7000	43.5250	35.2500	35.0500
$\vartheta = 0.40$	Graphic Screening	6.9500	2.1500	0.4000	0.0750	0.0500
	UPS	7.7500	4.0000	2.2000	2.7750	2.4250
	lasso	12.8750	6.8000	4.3250	3.7500	2.6750
$\vartheta = 0.55$	Graphic Screening	1.8750	0.8000	0.3250	0.2250	0.1250
	UPS	1.8750	0.8000	0.3250	0.2250	0.1250
	lasso	2.5000	1.1000	0.7750	0.2750	0.1250

Figure 1: Comparison of average Hamming errors (Experiment 1) (Jin et al., 2014).

The table above, as described in Jin et al., (2014) compares Graphlet Screening to UPS (Univariate Screening) and LASSO. During this experiment, the sparsity (ϑ) and the minimum signal strength (τ_p) are compared and tested across the three different algorithms by taking the average of the error in hamming distance across 40 different simulations per dataset, given a fixed ϑ and τ_p value. Lower values of ϑ indicate less sparsity, which means more significant signals within a generated dataset. An increase in τ_p indicates that the generated signals within the dataset are stronger. This table shows that as sparsity decreases and there are many signals to detect, Graphlet Screening's average error becomes significantly lower than other regression methods, indicating that Graphlet Screening operates comparably on sparse datasets and even better with datasets with many signals. Additionally,

this also indicates that as the signal strength decreases, Graphlet Screening can perform better than its peers. Graphlet Screening performs exceptionally well when tackling rare and weak datasets.

Additionally, newer methods of managing genomic data have been introduced, which serve to store and parse genomic data more efficiently using custom data structures. One example of these custom data structures is the Hail Python package, which uses “Matrix Tables” to implement genomic analysis in a scalable, performant environment (Hail Team, 2025). This project proposes a Python package for the integration of Graphlet Screening, a general regression model, into genomic analysis by using the Hail Python package to efficiently manage data to find correlations that were previously undetectable using industry standard methods.

First, the effectiveness of the Graphlet Screening model will be evaluated in identifying correlations in data sets ranging in their sparsity and signal strength. Pre-generated matrices will be passed into Graphlet Screening and other popular regression models (such as L_0-Penalized Regression and LASSO) to test the model. These data sets will be varied by signal strength and sparsity to determine their performance specifically amongst “rare and weak” data sets. Errors in performing the variable selection, ability to identify correlations, and time taken to run will be compared to each other to determine which data sets work best with which model. Then, the Graphlet Screening model will be tested using genomic data, utilizing manual integration with Hail that involves importing the data between R and Python. The data in Hail’s documentation will be used to observe consistency and comparisons between Hail’s built-in regression tool and Graphlet Screening (Hail Team, 2025). Results between Hail’s built-in regression tool and the Graphlet Screening tool will also be compared within the genetic context. Finally, the Graphlet Screening method will be coded in Python, integrated, and tested with the Hail package, noting the difference of the time taken and the ability to scale between manual and packaged interactions. The package will then be plugged into genomic databases (such as the NIH

"All of Us" research program) and demonstrated as a tool to complete GWAS studies and find more accurate correlations between SNPs and their phenotypes.

Section II: Project Plan

This project's goal is to improve the identification of correlations between genetic SNPs and phenotypes, therefore improving the analysis process of the typical GWAS and RNA-Sequencing study. This study will achieve this in a 4-step process. First, the existing methods of linear regression will be investigated and evaluated based on their runtime performance and their accuracy with a focus on three different algorithms – L_0 regularization, LASSO, and Graphlet Screening. L_0 and LASSO are commonplace in the genomic analysis process, while Graphlet Screening has not yet been implemented for use with genomic data. L_0 regularization is a regression model that is commonly used in high dimensional variable selection scenarios. Although this algorithm is computationally inefficient and fits under a category of problems in computer science known as "NP-hard" (Liu & Li, 2016). NP-hard problems are difficult to find solutions for, and they cannot be easily checked for accuracy (GeeksforGeeks, 2023). As such, it suffers from slow runtime performance for large-scale genomic analysis. LASSO (Least Absolute Shrinkage and Selection Operator), also used commonly in high dimensional variable selection, builds on L_0-Regularization by shrinking irrelevant coefficients to achieve a similar result to L_0 while achieving better runtime performance (Horowitz, 2015). However, it can fall behind in accuracy, specifically in higher sparsity and weaker signal datasets (Horowitz, 2015).

Graphlet Screening fixes these issues by using a two-step screen-and-clean procedure. During the screening process, connected subgraphs of correlated variables are compared with χ^2 (chi-squared) tests to determine true correlation from the connected variables to assemble a graph of strong dependency (GOSD). Then, the GOSD is cleaned using a penalized maximum likelihood estimation model which uses a probability distribution to eliminate unlikely dependency factors that may affect the regression. Graphlet Screening is uniquely advantageous within genomic analysis because of its ability to

work with the rare and sparse data commonly observed within the genome in a more efficient matter than L_0-regularization and LASSO (Jin et al., 2014). During the first step, the algorithms will be compared across their error (in terms of hamming distance) and their runtime performance (in seconds). These attributes will be compared by exposing the three different algorithms to simulated data sets varied based on their sparsity and general signal strength. Hamming distance is a simple technique of measuring error by comparing the number of positions within the intended beta vector and the beta hat (regression estimate) vector that are different. Hamming distance will be used as the ideal loss function as it is more appropriate for weaker signals due to its inherent low error tolerance (Jin et al., 2014). In the second step, the Graphlet Screening algorithm will be used with the 1000 public genomes dataset to test the algorithm's success with real genomic data. The results of this algorithm will be compared to the results of the Hail Python package as per their documentation. The Hail Python package efficiently stores and parses through genomic data by storing them in proprietary data structures called MatrixTables (Hail Team, 2025). Then, during the third step, the Graphlet Screening algorithm will be converted from the R programming language to the Python programming language and integrated with the Hail Python package so that it can take advantage of Hail's existing optimizations and bundled as a Python package. Finally, during the fourth step, the Graphlet Screening method will be tested with the genomic data in the NIH "All of Us" dataset, which contains genomic data for analysis.

Section III: Methodology

This project will require a computer capable of running R and Python. Additionally, it will use the L0Learn (L_0), glmnet (LASSO), GS (Graphlet Screening), and Hail programming packages. To retrieve and test with real genomic data, the 1000 public genomes dataset and the NIH "All of Us" researcher platform will be used (National Institutes of Health, *All of Us* Research Program, 2025).

Step one of the process aims to identify which algorithm is better in data sets as they vary in their sparsity and signal strength. The simulated data sets will be generated based on a random distribution that adheres to a given sparsity and signal strength factor, such that both the sparsity and the signal strength may be tightly controlled. Then, three algorithms (L_0 using the L0Learn package, LASSO using the glmnet package, and the Graphlet Screening method) will go through each type of data set 100 times per sparsity and signal strength value. Then, the runtime performance and the hamming distance error will be observed for each attempt. Hamming distance is a form of measuring error that, for binary strings of equal length, corresponds to the number of differing characters between two binary strings (Orlitsky, 2003). In this case, the same tactic can be applied to the vectors received in the experiment, differentiating the difference by the differences in vectors at a certain position rather than for binary strings. Hamming Distance is useful for comparing regression models as it punishes error in an easily comparable and more accurate way and puts error in more context by relating it to the number of signals present (Jin et al., 2014). Then, the error rates and runtime performance for each unique data scenario can be determined significant using ANOVA (to compare all three regression techniques) and t-tests (for comparisons within 2 different models).

Step two of the process aims to compare the Graphlet Screening model to Hail's existing regression model and determine if Graphlet Screening is effective in genomic analysis scenarios outside of pre-generated data. The procedure is fairly similar to procedure one, except that the regression models now involve a new regression model currently used in the Hail Python package (Poisson Regression). Both runtime performance and hamming distance will be compared in pre-made data scenarios. However, will also use the same data as used in the Hail documentation stemming from the 1000 public genomes dataset to compare with Hail's current findings for a certain subset of existing data. Traditional error measuring methods cannot be used as the data is not pre-generated. As such, it

does not have a true correlation factor that can be used for comparison. Rather, this step of the process works as proof of concept to show that Graphlet Screening can obtain comparable results to existing methods of analysis, such as the Hail Python package.

Step three involves the main bulk of the project in which the Graphlet Screening method, which is currently packaged as a package in the MATLAB/R programming languages, will be converted to Python and integrated with the Hail Python package to form a single Python package capable of higher-accuracy genome analysis. Currently, the Hail package utilizes a custom data structure known as the MatrixTable to efficiently parse and store genomic data from .vcf (Variant Call Format) files, which are used commonly within the genomic analysis field (Hail Team, 2025). This data structure would need to be converted into matrices with SNPs among different genomes being placed in the design matrix X. At the same time, the response vector Y contains a phenotype with values associated with their sequence. For example, if the experiment desires the genomic link for height, then the response vector would include many different heights across multiple patients, comparing their SNPs to see which of them contribute to the expression of the phenotype. After the data structure is converted for use and the Python package is assembled, procedure two can be reutilized to check for major differences in the runtime performance and accuracy.

Finally, step four would involve utilizing new genetic data that has not been analyzed before to see how the Graphlet Screening package functions in a typical analysis environment. For this study, the NIH “All of Us” database will be used. The package would be imported into the workbench environment and then utilized in existing research done on the platform to see if there are any major differences, similar to procedure 3.

Section IV: Comparison with Graphlet Screening

In this section, preliminary data and the findings from procedure one will be described. The Graphlet Screening method is compared to L_0-Regularization and LASSO.

Regression Model	Simulation Count	Sum of Hamming Distance	Average of Hamming Distance	Variance (Standard Deviation)
Graphlet Screening	100	6041	60.41	572.81
LASSO	100	21389	213.89	846.3211
L_0-Regularization	100	8928	89.28	93.41576

*P-Value: 1.7*10^-148 | v (sparsity) = 0.35 | r (signal strength) = 3.5*

Figure 2: ANOVA Test Results Between Different Regression Models for Hamming Distance

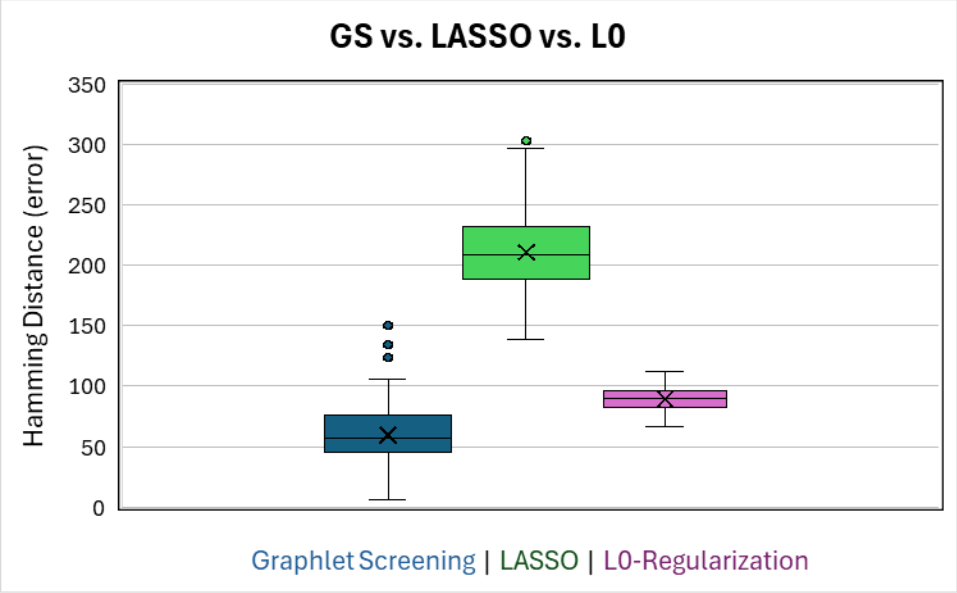

Figure 3: Hamming Distance for all 3 algorithms represented through a box and whisker plot.

Figure 2 and Figure 3 show the results for the ANOVA test between the three different algorithms on the hamming distance (loss function). Higher values of hamming distance indicate that there were more differences in the predicted correlation matrices than the actual correlation matrix; therefore, lower hamming distance values are better. The v value represents the sparsity. The higher the v value, the lower the number of signals in the simulated data sets. Sparsity can range from 0 – 1, with 0 having every value in the matrix be non-zero and 1 having only one value in the matrix be non-zero. In this case, the sparsity is low which means that this data set has more non-zero values than typically observed in genomic analysis scenarios. The r value represents the minimum signal strength to be simulated within the generated datasets. The r value can have quite a broad range, although within the genetic context, it makes more sense for a genomic context to be anywhere from 3.5 to 5. These values indicate that our data set is weak and somewhat rare, although not exactly weak and rare. Future simulations will bridge this gap and observe the changes in accuracy and runtime performance across different sparsity values and signal strengths. The average hamming distance for Graphlet Screening is far lower than that of LASSO and slightly better than L_0-regularization, indicating better performance generally. The variance, however, is the lowest among L_0-regularization. The increased variance present in Graphlet Screening tends to involve specific instances where Graphlet Screening would obtain an astronomically low error, introducing rare outliers in which Graphlet Screening far performed its peers. Only two significant outliers (with hamming distances of 7 and 8, respectively) were collected out of 100 simulations, all of which benefited the average outcome of the algorithm. Therefore, it can be concluded that the variance in the Graphlet Screening method does not necessarily detract from its stability rather than from random chance due to the generation of the correlated matrices. Recall that the generated datasets remain affected by random chance as they involve normal distributions and can naturally lead to outliers within the data that benefit Graphlet Screening. Excluding the two major outliers, Graphlet Screening receives an average of approximately 61.49, a +1.08 change to the average, still

outperforming L_0-regularization. Therefore, there is significant statistical evidence to indicate that Graphlet Screening outperforms commonly used regression algorithms when used with weak and somewhat rare data. More data remains to be obtained on higher sparsity values to observe how comparative performance between the three different algorithms changes.

Section V: Ethical Considerations

A part of the genomic data used in this project is public and fully available to the general population (such as the 1000 public genomes dataset). However, the data from the NIH "All of Us" database, particularly genetic data, is considered part of the Controlled Tier. This means that the data contains some level of Personally Identifiable Information (PII) requires ID verification and 2 courses on omitting and avoiding population bias while using the researcher workbench.

This project will always be compliant with the following policies as outlined by the NIH "All of Us" Research Program:

- The Terms of Use (https://workbench.researchallofus.org/aou-tos.html) (*All of Us* Research Program, 2025)
- The Data User Code of Conduct (https://support.researchallofus.org/hc/en-us/article_attachments/22130615077908) (*All of Us* Research Program, 2023)
- The Data and Statistics Dissemination Policy (https://support.researchallofus.org/hc/en-us/article_attachments/22307203778836) (*All of Us* Research Program, 2020)

Section VI: Timeline

This project is split into four steps. Broadly, they are:

1. Compare the Graphlet Screening regression algorithm to 2 other regression models, specifically LASSO and L_0-Regularization.
2. Try it with the 1000 public genomes dataset, inputting it manually and comparing it to the Hail Python package.

3. Create a Python package which connects the Graphlet Screening method to the Hail Python package.
4. Test the Python package with genomic data using the NIH “All of Us” database.

Step one has already been completed, while step 2 is underway and will be completed by late December. Step three is planned for completion in January, while step four will be completed in February.

Section VII: References

All of Us Research Program. (2023, October 31). *Data User Code of Conduct*. https://support.researchallofus.org/hc/en-us/articles/22346176432532-Data-User-Code-of-Conduct

All of Us Research Program. (2020, May 12). *Data and Statistics Dissemination Policy*. https://support.researchallofus.org/hc/en-us/articles/22346276580372-Data-and-Statistics-Dissemination-Policy

All of Us Research Program. (2025). *Terms of use*. https://workbench.researchallofus.org/aou-tos.html

GeeksforGeeks. (2023, October 3). *P, NP, CoNP, NP hard and NP complete*. GeeksforGeeks. https://www.geeksforgeeks.org/types-of-complexity-classes-p-np-conp-np-hard-and-np-complete/

Hail Team. (2025). *Hail* (Version 0.2.133). [Computer software]. GitHub. https://github.com/hail-is/hail

Hong, S., Kim, Y., & Park, T. (2015). Practical issues in screening and variable selection in genome-wide association analysis. *Cancer informatics*, *13*(Suppl 7), 55–65. https://doi.org/10.4137/CIN.S16350

Horowitz, J.L. (2015), Variable selection and estimation in high-dimensional models. *Canadian Journal of Economics/Revue canadienne d'économique*, *48*: 389-407. https://doi.org/10.1111/caje.12130

Jin, J., Zhang, C.-H., & Zhang, Q. (2014). Optimality of graphlet screening in high dimensional variable selection. *Journal of Machine Learning Research: JMLR*, *15*(79), 2723–2772. https://www.jmlr.org/papers/v15/jin14a.html

Liu, Z., & Li, G. (2016). Efficient regularized regression with L0 Penalty for variable selection and network construction. *Computational and Mathematical Methods in Medicine*, *2016*, 1–11.

https://doi.org/10.1155/2016/3456153

National Human Genome Research Institute. (2020, August 17). *Genome-wide association studies fact sheet*. National Institutes of Health. https://www.genome.gov/about-genomics/fact-sheets/Genome-Wide-Association-Studies-Fact-Sheet

National Institutes of Health, *All of Us* Research Program. (2025). *All of Us* Research Program dataset. https://www.researchallofus.org/

Orlitsky, A. (2003). Information Theory. In R. A. Meyers (Ed.), *Encyclopedia of Physical Science and Technology (Third Edition)* (Third Edition, pp. 751–769). Academic Press. https://doi.org/10.1016/B0-12-227410-5/00337-9

What are single nucleotide polymorphisms (SNPs)? (n.d.). Medlineplus.gov. Retrieved November 8, 2024, from https://medlineplus.gov/genetics/understanding/genomicresearch/snp/