Feature Selection via Sequential Parallel LASSO for eQTL Analysis

Anhong He
Syracuse University
900 South Crouse Ave
Syracuse, NY 13244, U.S.
anhonghe@gmail.com

Benika Hall
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC 28223
bjohn157@uncc.edu

Jia Wen
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC 28223
jwen6@uncc.edu

Yingbin Liang
Syracuse University
900 South Crouse Ave
Syracuse, NY 13244, U.S.
yliang06@syr.edu

Xinghua Shi
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC 28223
x.shi@uncc.edu

ABSTRACT

Human gene expression is subjected to multiple layers of controls including the impact of genetic variants. Expression quantitative trait loci (eQTL) analysis has emerged as a powerful tool for investigating this genetic impact. In eQTL analysis, the data is typically high-dimensional whereas the number of observations exceed the number of samples. Therefore, sparse learning models are suitable for eQTL analysis. Sparse learning models have the ability to reflect the underlying biology in cases where only a small subset of genetic variants significantly affect gene expression. Therefore, sparse learning models such as LASSO, have shown their strengths to select associated features in high dimensional data. However, classical LASSO performs poorly when dealing with extremely high dimensional datasets in human genomic data. In this study, we introduce two novel methods named sequential LASSO and parallel LASSO. These methods allow efficient learning for datasets of ultra-high dimension. We theoretically prove the consistency of these methods when exact recovery is achieved in each sub-procedure. We further provided a multi-round process to address the sample size limitation in real applications. In addition to our theoretical analysis, we also perform extensive simulations on synthetic data to validate our methods. When applying to a real human genomics data set, we identify a set of SNP eQTLs that affect genes previously reported to be associated with human traits and diseases. Our methods are not limited to classical LASSO models and can be extended to variations of LASSO and many other machine learning models.

ACM BCB '2015 Atlanta, Georgia USA

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Bioinformatics**]: Genetic data processing—*high dimensional data processing, feature selection*

General Terms

Graphical Models, Feature Selection

Keywords

Sequential/Parallel LASSO, eQTL analysis

1. INTRODUCTION

Human gene expression is often mediated by genetic, epigenetic, and environmental factors. The effects of genetic variation plays an important role in modulating gene expression. Genetic variation is reflected by genetic variants of genomes between members of species, or between groups of species thriving in different parts of the world. Although humans share the majority of their genetic sequence, the genetic variation among individuals contributes to considerable phenotypic differences in human populations. Genetic variation is composed of genetic differences that range from single nucleotide polymorphisms (SNPs - one base pair substitutions) to structural variants including large deletions, duplications, insertions, translocations, and inversions which can be as large as whole chromosomes. Comprehensive identification, annotation, and assessment of these genetic variants will help us understand the biology of their functional effects on human health and disease.

The biological community has generated detailed catalogs and genotypes of genetic variation in populations of different organisms including humans. Two genetic variation databases, namely dbSNP[2] and dbVar[3], have respectively stored short genetic variation (including SNPs), and large structural variation of various organisms. The largest set of genetic variation is from human HapMap individuals across diverse populations. The recent release of human build in dbSNP includes 260,570,204 SNPs with 73,909,256

genotyped. Various studies have reported hundreds of thousands of structural variants curated in dbVar [3] as well.

Given the accessibility of such massive data collections, dimensionality reduction is critical for us to improve the algorithm performance allowing us to interpret the relationships accurately. Many methods in dimensionality reduction have been proposed. For example, one particular sparse model, least absolute shrinkage and selection operator (LASSO)[32] based methods have been recently used in human genetic studies including eQTL analysis[37, 38, 36]. In a LASSO model for eQTL analysis, based on the L_1 regularization, the parameters of the majority of the genetic variants are shrunken to zeros and those variants corresponding non-zero terms are selected as the identified eQTL associations. Different variations of LASSO have been suggested [18, 19, 24, 23, 11, 12] to take into consideration the structures embedded in the data for eQTL analysis.

However, there are several essential limitations of LASSO when it is directly applied to biological datasets. First, LASSO is dependent on the input size therefore the input size of features should not be too large. For modern datasets especially in human eQTL mapping, the data high-dimensional and intractable. Many presented approaches begin by choosing some "important" features and then perform the selection based on the that subset. In general, there are two kinds of pre-selection methods: random selection or covariance-based selection. Random selection is obviously error-prone if an important feature was dropped at the beginning it can never be retrieved. For the covariance-based pre-selection, the covariance matrix only represents pairwise dependencies rather than conditional dependencies. While the essential goal for feature-selection is to retrieve features that are conditionally independent, thus the covariance based pre-selection is not a proper method for trimming as well. Moreover, classical LASSO cannot update data systematically, meaning that each time when new data becomes available LASSO has to reprocess the whole dataset again.

In this paper we propose two novel methods, termed as sequential and parallel LASSO, to scale up classical LASSO models for extremely high dimensional data and to overcome the drawbacks of existing pre-selection methods. Our theoretic analysis demonstrates the consistency, stability and performance of our new approaches. We perform numerical simulations to further demonstrate the performance of our methods.

Another limitation of LASSO lies in parameter tuning. In classical LASSO, a parameter λ should be chosen properly to meet the theoretical requirements for exactly recover with a high probability. Generally, λ is determined by many factors such as singular-values of the design matrix, dimensionality of the data and ground-truth sparsity level[34]. Meanwhile, there isn't a straightforward way to set λ before applying LASSO. In practice if we have prior information for the sparse level that allow us to set λ by trial and error until we obtain a proper solution with the desired sparse level. But for biological applications, we often do not have any prior knowledge for the ground-truth and thus it is extremely hard to set a proper λ. For real datasets, the sample size can never meet the required theoretical condition, then the "exact recovery" is out of reach and the performance must be optimized based on limited samples.

One popular solution for tuning λ is to apply cross-validation [14]. In this paper we follow this idea. Specifically, we properly pick an initial set of λ's and minimize the test error over all these candidates to obtain an optimal λ. One small change in our methods is that at every minimization step of λ_n, we use the solution w.r.t λ_{n-1} as initial point. We apply this warm-start strategy to reduce time since for small change of λ's, we would expect close solutions. Faced with the curse of high-dimensionality and the insufficient sample sizes in human genetics studies, we derive a "multi-round" method to improve the model performance measured by F_1 scores.

Active set methods[22, 33, 17] are another way for solving LASSO problem. In [22] the authors proposed a gradient-descent algorithm with an active set associated method. For these active set methods, the optimization problem is the original problem, but for our method, the problem becomes different when we derive a sub-problem using partitions over the dataset. For each sub-problem, we apply FISTA[9] which guarantees $O(n^2)$ convergence rate. In active set methods, the convergence rate is not clearly defined. Additionally, the method in [22] requires that the memory should fit the design matrix X at least since each iteration it need to compute $X^T X\beta$. An intuitive idea for optimizing the time consuming is based on the prior knowledge that β is sparse, then the algorithm every time add at most one support into the selected feature set, for larger λ or equivalently, sparser groundtruth it's a cleaver idea. However for real data we need to apply cross validation to fit proper λ^*, there would be some candidates of λ's which are very small, or equivalently, the solution would be dense, this cause huge problem for algorithm proposed in [22]. On opposite our methods do not require that because we can design our subproblem of a proper size which is deemed fit for the hardware limitation, and our optimization tool for each sub-procedure is of FISTA which do not require a prior knowledge of sparse solution, and instead we use warm-start strategy to force our algorithm start from a sparse initial point when λ_n get larger. Moreover, our methods are flexible to streaming data which further improves their scalability.

There is another paper[25] which is also called Sequential LASSO, but the idea in that paper and ours are totally different. In [25] the author state a new approach for proper penalty for sparse solution, they gives a systematic way to penalty features in the design matrix X in a sequential manner. The also give theoretical guarantee for the performance. For our paper we use cross validation for proper penalty parameter λ.

2. METHODS

We employ a multi-variate linear regression model to capture the associations between genotypes of genetic variants and gene expression in eQTL analysis. In this paper, we consider SNPs as explanatory variables or features, but the model can be applied to any type of genetic variants. We use the $N \times P$ matrix X to denote the design matrix encoded by SNP genotypes, with each entry of X taking a value 1, 2, 3 or 4. Here, we consider the phased genotypes as in the 1000 Genomes Project[1, 27]. In applications with un-phased genotypes, this model works as X would encode three values from the set of {1,2,3}. Each column of X denotes N realizations for one SNP in N individuals. We assume that we have P SNPs in total. Then X was normalized column-wisely to have zero mean and variance of 1. We then represent gene expressions as the response vector $y_{N\times 1}$

which takes continuous values of gene expressions. Here, N denotes there are N realizations for each gene in the N individuals. We normalize y to be of zero-mean and unitary variance.

To retrieve the associations between $X \to Y$, we assume that X and Y Gaussian distributed and are subject to the following linear model. If we take y as the response variable and $\vec{x}$ as explanatory variable, here we use $\vec{x}$ to denote a p-dimensional feature vector, then

$$y = \vec{x}\vec{\beta} + z \tag{1}$$

where $\vec{x} \sim \mathcal{N}(0, \Sigma_x)$, $z \sim \mathcal{N}(0, \sigma^2)$ and z is independent from $\vec{x}$. By the properties of Gaussian graphical model[35, 20], when conditioning on all x_i w.r.t. $\beta_i \neq 0$, y should be independent from all other x_j's of which $\beta_j = 0$. Here we define that x_i and y has an eQTL association if $\beta_i \neq 0$.

Then, performing eQTL analysis can be modeled as a linear regression problem between genotypes and gene expressions. Linear regression is a simple yet useful method to retrieve associations between explanatory variables and response variables. By the property of MMSE [16] we know β in Equation (1) is the solution of the following problem

$$\arg\min_{\beta} E\,|y - \vec{x}\vec{\beta}|^2 \tag{2}$$

and in practice we minimize the empirical mean with n samples of $\vec{x}$ and y. Further, let $X_{n\times p}$ denote the design matrix as n samples of $\vec{x}$ stacked column-wisely, and $\vec{y}_{n\times 1}$ denote n samples of y. Then Equation (2) becomes

$$\vec{\beta}^* = \arg\min_{\vec{\beta}} \frac{1}{n}\|\vec{y} - X\vec{\beta}\|_2^2 \tag{3}$$

In general, statisticians would add some penalty term to Equation (3) to achieve a special structure of solutions. For feature selection problems, sparsity is another assumption that reflects the underlying biology. In eQTL analysis, only a few genetic variants are believed to affect the response variable directly, which means we only need to select a small portion of those factors from the whole dataset. LASSO[32] is one of most powerful tools for achieving sparsity, which adds $\|\vec{\beta}\|_1$ as a penalty term and a parameter λ to fine tune the trade-off between regression precision and solution sparsity. A classical LASSO model is formalized in Equation (4). All features that are associated with y in the graphical model can be retrieved by taking the support set of β^* after minimization. Previous work has shown that the support set of $\vec{\beta}$ can be exactly recovered with proper chosen λ under some conditions[28, 34].

$$\vec{\beta}^* = \arg\min_{\vec{\beta}} \frac{1}{n}\|\vec{y} - X\vec{\beta}\|_2^2 + \lambda\|\vec{\beta}\|_1 \tag{4}$$

Equation (4) is the general form of classical LASSO that plays a pivot role in our methodology. When the dimension of data is moderate, classical LASSO works pretty well. However, in classical LASSO the computation complexity increases as the dimension of features increase. Moreover, classical LASSO requires to take the input of the entire design matrix of x before computing. In human genomic datasets, the dimension of X can be tens of millions or even higher. With a high demand on memory and time usage, classical LASSO might not be directly applicable in many applications. A classical LASSO is also an off-line method, meaning that we need to collect all features before solving the problem, and for any new data coming we need to restart this time consuming procedure.

In contrast to solving the classical LASSO problem in a distributed manner [26], a straightforward idea is to divide the LASSO problem into sub-problems. In each sub-problem we only need partial information of X and we expect to achieve the consistency of support set. Here we propose two novel methods called sequential/parallel LASSO to address the high-dimensionality problem. Theoretical guarantee of consistency of these new models are given in Theorem (1, 2).

2.1 Sequential LASSO

The sequential LASSO and parallel LASSO we propose to handle the high dimensional problem in two different manners. Nonetheless, for both methods we should first partition the large feature set into smaller partitions. Let first give some notations here.

For each feature X_j we use index j to indicate it. A *partition* $\{G_1, \cdots, G_K\}$ over the feature set $\{1, \cdots, P\}$ is defined as follows:

1. $G_k \neq \emptyset$, $\forall\ k = 1, \cdots, K$.
2. $\bigcup_{i=1}^{K} G_i = \{1, \cdots, P\}$
3. $G_i \cap G_j = \emptyset$, $\forall i \neq j$

and a feature set X_G indicated by index set G means

$$X_G = [X_{i_1}, \cdots, X_{i_k}, \cdots, X_{i_{|G|}}], \text{ where } i_k \in G \tag{5}$$

The processing structure of sequential LASSO is illustrated as in Figure (1). We can see that sequential LASSO is composed of a cascading structure with K LASSO selectors. For the k^{th} selector, the input feature set is the union of G_k and S_{k-1}, where S_{k-1} is the selected feature set by the $(k-1)^{th}$ selector. For each selector, we take $\vec{y}$ as the response vector and apply the basic LASSO procedure. Algorithm (1) describes the pseudo code for Sequential LASSO.

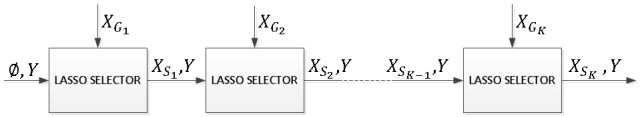

Figure 1: Structure of Sequential LASSO

Algorithm 1 Sequential LASSO

Require: Valid partition $\{G_1, \cdots, G_K\}$ over feature set $\{1, \cdots, P\}$
  $S_0 \Leftarrow \emptyset$, $k \Leftarrow 1$
  while $k \leq K$ **do**
    $X \Leftarrow X_{G_k \cup S_{k-1}}$
    $\vec{\beta}^* \Leftarrow \arg\min_{\vec{\beta}} \|\vec{y} - X\vec{\beta}\|_2^2 + \lambda\|\vec{\beta}\|_1$
    $S_k \Leftarrow (G_k \cup S_{k-1})_{supp(\vec{\beta}^*)}$
    $k \Leftarrow k + 1$
  end while
  return $S \Leftarrow S_K$

For the original LASSO process, all features will be fed at one time to find the feature subset of which directly links to y. But for sequential processing, in every selection step we remove some features which are independent from y conditioning on those selected features, while some new features will be included. In view of a Gaussian graphical model, the k^{th} LASSO selector is working on a marginalized subgraph [20] with vertices $\{X_{S_{k-1}\cup G_k}, y\}$ and edges from the original graph $\{X_{\{1,2,\dots,P\}}, y\}$ or induced by marginalization. We repeat this process over the whole feature set and hopefully all features associated with y could be retrieved. The idea of this process looks intuitive but works well, and its performance is guaranteed by Theorem (1).

Theorem 1. *The sequential LASSO algorithm exactly recovers all X_i's associated with y if each LASSO selector can exactly recover all X_i's associated with y in its input Gaussian subgraph.*

Theorem (1) states that we need exact recovery for each step to achieve consistency, and different partitioning won't affect the result. Given sufficient number of samples and with proper λ, the neighborhood set to y can be exactly recovered with high probability by the deeply studied properties of LASSO ([28, 34]). In many bioinformatics applications as in our study, we assume the exact recovery of LASSO. We will later show that the exact recovery assumption can be relaxed in our algorithms. Theorem (1) guarantees the performance of sequential LASSO, which provides us a way to deal with ultra-high dimensional data in a "divide and conquer" fashion. Additionally, the sequential LASSO is an online method. Meaning, when new features become available, we can union the new features with the selected features and perform LASSO again, thus obtaining final results without re-calculation over the whole dataset.

2.2 Parallel LASSO

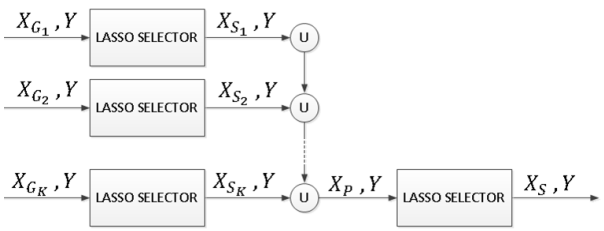

Figure 2: Structure of Parallel LASSO

Parallel LASSO is another method proposed in this paper. Details of the processing structure of parallel LASSO is shown in Figure (2). Similar to the partitioning step in sequential LASSO, the entire feature set should first be partitioned into K subsets in parallel LASSO. However, parallel LASSO feeds module k^{th} with feature set X_{G_k} only. Here, we denote the output set as S_k. After finishing all these K operations, we union all the selected feature sets as input set and do selection again via the last LASSO selector to get the final output feature set S

$$\bigcup_{i=1}^{K} S_k \xrightarrow{LASSO} S \tag{6}$$

Algorithm (2) shows the pseudo code of parallel LASSO. The exact recovery for parallel LASSO is guaranteed by Theorem (2).

Algorithm 2 Parallel LASSO

Require: Valid partition $\{G_1, \cdots, G_K\}$ over feature set $\{1, \cdots, P\}$
  for $k \in \{1, \cdots, P\}$ **do**
    $X \Leftarrow X_{G_k}$
    $\vec{\beta_k}^* \Leftarrow \arg\min_{\vec{\beta}} \|\vec{y} - X\vec{\beta}\|_2^2 + \lambda\|\vec{\beta}\|_1$
    $S_k \Leftarrow (G_k)_{supp(\vec{\beta_k}^*)}$
  end for
  $X \Leftarrow X_{S_1\cup\cdots S_K}$
  $\vec{\beta}^* \Leftarrow \arg\min_{\vec{\beta}} \|\vec{y} - X\vec{\beta}\|_2^2 + \lambda\|\vec{\beta}\|_1$
  return $S \Leftarrow (S_1 \cup \cdots S_K)_{supp(\vec{\beta}^*)}$

Theorem 2. *The parallel LASSO algorithm exactly recovers all X_i's associated with y if each LASSO selector can exactly recover all X_i's associated with y in its input Gaussian subgraph.*

2.3 Theorem proofs

In this section, we present the proofs to Theorems (1) and (2) in order to explain why our novel method works. Here, we use T to denote the set of all X_i's associated with y. In other words, T is the ground-truth. S_k is the selected feature set by k^{th} LASSO selector. S is the final set of selected features when applying the sequential/parallel algorithm. With all these notations, we prove that with exact recovery of each LASSO selector, we should have $S = T$.

Proof Proof for Sequential LASSO. For $T \subset S$ where $S = S_K$ for sequential LASSO: If $\exists X_r$ s.t. $X_r \in G_k$ that $\forall k = 1, \dots K$, then by the exact recovery of LASSO, we should have $X_r \in S_k$. Since any marginalization won't cancel the association of $X_r - Y$ if X_r is present in the marginalized graph [20]. Meanwhile, by the exact recovery of LASSO, we have $X_r \in S_i$ that $\forall i \geq k$. Hence, we have $X_r \in S$. This proves $T \subset S$.

For $S \subset T$: We assume $\exists X_r \notin T$ but $X_r \in S$, by the exact recovery property of LASSO selector, there should $\exists X_r - Y$ in the marginalized graph $\{X_{S_{K-1}\cup G_K}, Y\}$, or equivalently, $\exists X_Q$ s.t. $X_r - X_Q - Y$ where $X_Q \subset (S_{K-1} \cup G_K)^C$. However, since $T \subset S$, $S = S_K$ and $S_K \subset S_{K-1} \cup G_K$, we have $T \subset X_{S_{K-1}\cup G_K}$, meaning for those $X_q \in (S_{K-1} \cup G_K)^C$, X_q should either be isolated or indirectly associated with Y. This gives us the contradiction. □

Proof Proof for Parallel LASSO. For $T \subset S$: If $\exists X_r$ s.t. $X_r \in G_k$ that $\forall k = 1, \dots K$, then by the exact recovery of LASSO, we should have $X_r \in S_k$. Since in the second step of parallel LASSO, the input feature set is the union of S_k that $k = 1, \dots K$, again by the exact recovery of LASSO we should have $X_r \in S$. This proves $T \subset S$.

For $S \subset T$: Now we assume $\exists X_r \notin T$ but $X_r \in S$, by the exact recovery property of LASSO selector, there should $\exists X_r - Y$ in the marginalized graph $\{X_{S_1\cup S_2\cup\cdots\cup S_K}, Y\}$, or equivalently, $\exists X_Q$ s.t. $X_r - X_Q - Y$ where $X_Q \subset (S_1 \cup S_2 \cup \cdots \cup S_K)^C$. However, since $T \subset S_1 \cup S_2 \cdots \cup S_K$, meaning for those $X_q \in (S_1 \cup S_2 \cup \cdots \cup S_K)^C$, X_q should either be isolated or indirectly linked to Y. This gives us the contradiction. □

In the proofs above, we used the marginalization properties of Gaussian graphical model [20]. Actually, the requirement for exact recovery of each LASSO selector is essential, but not necessary. Algorithms (1) and (2) will recover the ground-truth T as long as each LASSO selector can recover all features in T presented in its marginalized input subgraph. For those associations induced by the marginalization, no errors would be induced even if the LASSO selector missed them. In practice, these associations are easy to miss since their amplitudes are much weaker than that of the ground-truth. This gives us a possibility that even if the LASSO selectors made "small mistakes", we still have the chance to recover T.

With theoretical analysis described in previous sections, we derive approaches to explore all of the associations using the proposed sequential and parallel LASSO. In practice, we can use a mixed procedure "sequential/parallel" LASSO which draws the benefits of sequential LASSO and parallel LASSO. Sequential LASSO can represent the response variable with the feature set which is "already accessible", and parallel LASSO can significantly reduce the processing time.

2.4 The multi-round method

In Theorem (1, 2) the consistency of sequential/parallel LASSO is based on the exact recovery of each LASSO selector. For all scenarios ([28, 34]) about exact recoverability of LASSO, the sufficient sample size is always required. In other words, the probability of exact recovery approaches 1 as the number of samples approaches infinity. However in practice, the sufficient sample condition may be out of reach, especially in genomic studies.

Cross validation (CV) also leads to inconsistent results. In simulation studies, we found that cross validation tends to weaken the sparse selection characteristics of LASSO, meaning that the optimal λ^* always gives us denser solution than the ground-truth we assumed. The final output feature set will contain false positives with coefficients(β_i's) that are non-zero but with very small amplitudes. Setting a hard threshold to filter out all these coefficients is a straightforward idea, however, this also drops those true positive features with small amplitudes. Since we do not have any prior of the amplitude of the support coefficients, it's hard to set a proper threshold on the support coefficients. To overcome these drawbacks, some varied versions of CV, such as "One standard error rule" [31], have been proposed.

Here we propose a "multi-round" method which runs the sequential/parallel LASSO in multiple rounds with different partitions, and find those overlapped support sets. In practice this method significantly suppresses false positives while remaining robust for detecting true positives. Specifically, we run sequential/parallel LASSO M times with different partitions over the entire dataset, and find those support sets which overlapped at least γ_M time in all M rounds. When the sample is sufficient we expect that the same support set would repeat M times. Since sample size is often limited in real applications, the support sets would differ. But those in "true support sets" would be selected continuously with higher probability, and those support sets that vary between rounds may be false. A larger M will strengthen our confidence, but with higher time cost. Thus, we find a reasonable (M, γ_M) pair empirically by learning from the data in practice.

3. RESULTS

To evaluate the performance of sequential/parallel LASSO, we perform comprehensive simulations on synthetic data. We then demonstrate our method in real scenarios by applying it to a real human genomic data set.

3.1 Validation for sequential/parallel LASSO

Theorems (1, 2) show that for sufficient sample size, the selected support set would approach to the ground-truth with high probability, allowing different partitions of feature set. Here, we perform two simulations to illustrate the exact recoverability of sequential and parallel LASSO. In simulations, synthetic data is generated based on the following linear model

$$y = X\beta + z \tag{7}$$

where $X \sim \mathcal{N}(0, \Sigma_x)$ ($\Sigma_x \neq I$) and β is the coefficient vector we try to recover. z is a Gaussian noise independent from X. In simulations we set

1. Dimension of X is 1×10000.
2. $Var(X_i) = 1$ and $z \sim \mathcal{N}(0, 1)$.
3. Sparsity level of β is 0.002, $\beta_i = 0$ or ± 1.

To recover β we apply sequential LASSO with fixed λ and fixed feature capacity C. Here, "feature capacity C" means that for one LASSO selection step, if L features are dropped, then we add L new features and union with the remaining $C - L$ features to do the next LASSO selection. For each LASSO selector we set:

1. $C = 1000$.
2. $\lambda = 0.90$.

Figure (3) shows that the probability of exact recovery approaches 1 as sample size becomes sufficient in sequential LASSO. Three curves of probability are drawn to illustrate the consistency for different partitions of feature set.

Similarly, Figure (4) shows the stability of parallel LASSO for different feature set partitions.

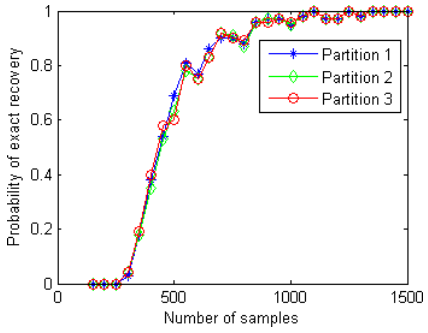

Figure 3: Recoverability for sequential LASSO with different partitions of feature set.

3.2 Setting parameters for the multi-round method

For our multi-round method of running sequential/parallel LASSO, we need to set the parameter pair(M, γ_M). In general $\gamma_M \leq M$. Here, we simply set $\gamma_M = M$, which means that we'll select those support features selected during all

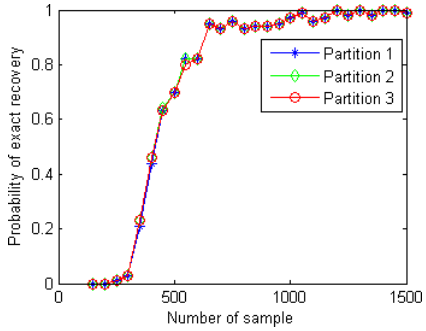

Figure 4: Recoverability for parallel LASSO with different partitions of feature set.

rounds. Taking $\gamma_M = M$ is based on Figure (5, 6), which shows when applying sequential/parallel LASSO, F_1 score stays increasing when γ_{50} increases and when $\gamma_{50} = 50$, F_1 score achieves its maximum. Different settings of M show that F_1 score increases as γ_M increases.

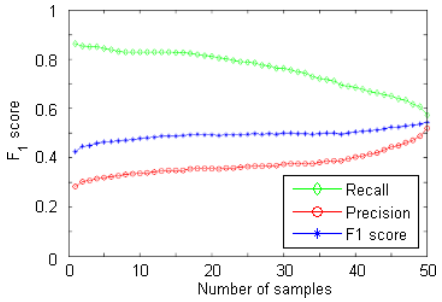

Figure 5: F_1 scores with varying γ_{50} for multi-round method in sequential LASSO

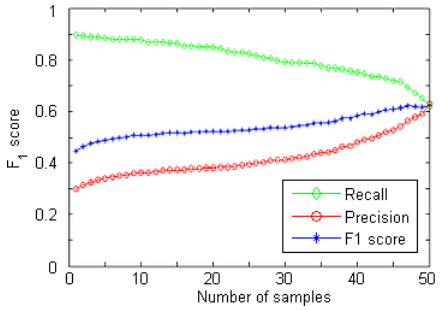

Figure 6: F_1 scores with varying γ_{50} for multi-round method in parallel LASSO

We next set a proper M for the multi-round method. A larger M means higher confidence but with more time consuming. Figure (7) shows the behavior of multi-round method when the sample size is insufficient with different M. Here, sequential LASSO was applied based on the linear model (7). We calculate F_1 score as a criteria for $M = 1, 3, 5, 7, 9$.

Figure (3) shows that for exact recovery, the sample size should be around 1,000. In Figure (7), sample size varies between 100 to 400 and thus we are in the condition of insufficient sample size. For a different M, we calculate the

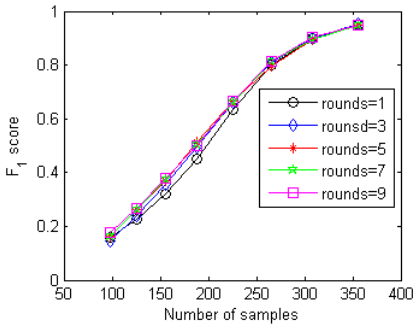

Figure 7: Curves of F_1 score based on multi-round method for different M in sequential LASSO.

F_1 score to illustrate the recovery of sequential LASSO with multi-round method. In Figure (7), the curves with larger M always dominate the curves with smaller M, which makes sense since larger M means higher confidence. However, when M is large enough, increasing M doesn't significantly improve the performance of our method, as F_1 score curves with $M = 5, 7, 9$ are pretty close. Note that a larger M costs more time. In this study, we set $M = 5$ for applications on real data.

3.3 eQTL analysis on real genomic data

To demonstrate the application of sequential/parallel LASSO in practical scenarios, we apply our methods to genomic datasets available on HapMap Yoruba individuals. We use the SNP genotypes recently published form the 1000 Genomes Project[1, 27] and gene expression [30]. We pick 177 gene expressions on chromosome 22 as examples, although our methods are capable of analyzing all human genes. We collect the genotypes of 184,158 SNPs and expression profiles of 385 genes in 38 samples. Before analysis, we use Plink [5] to generate the tagSNPs by considering linkage disequilibrium (LD) among SNPs with a correlation efficient cutoff at 0.9. Then we perform regression of gene expression on tagSNPs to obtain eQTLs using sequential LASSO.

After the preprocessing step described above, we have 94,588 SNP's that would be fed into our model. By applying the sequential LASSO method, we selected 6,690 tagSNP's to be associated with gene expressions. A supplementary document for all these gene expression and selected SNP's respectively is available online[7]. Note that we don't have distance cutoff when investigating the association between SNPs and genes. Thus these SNPs affect gene expression either locally (i.e. *cis* eQTLs) or globally (i.e. *trans* eQTLs). We compared our results with previously reported SNP eQTLs in the GEUVADIS study [21], although the later study only assessed local eQTLs where the SNPs and genes are within 1MB window. We found that our results replicated some of the eQTLs reported in the GEUVADIS study. For example, for the expression of C22ORF34, we found 24 common SNP-gene pairs that are also found in GEUVADIS[21], as summarized in Table-(1).

Figure-(9, 8) shows the Venn diagrams for overlapping SNP eQTLs and associated genes respectively between our study and that of GEUVADIS[21] using Venny[4]. We clearly see that the two studies replicate each other to some extent, while are capable of finding novel eQTLs that cannot be

Index	SNP ID
1	rs739239
2	rs739241
3	rs739248
4	rs763127
5	rs1018812
6	rs2008439
7	rs2051626
8	rs2071904
9	rs4524218
10	rs5769698
11	rs5769707
12	rs5769712
13	rs5770585
14	rs5770594
15	rs5770610
16	rs7288869
17	rs8141807
18	rs9616327
19	rs9616332
20	rs9616333
21	rs9616701
22	rs9616713
23	rs9616714
24	rs12485195

Table 1: Replicated SNP eQTLs for the C22ORF34 gene.

identified from the other method.

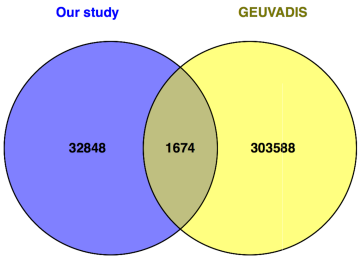

Figure 8: Venn diagram of the SNP eQTLs found in our study and the GEUVADIS study [21]

As our study and the GEUVADIS study [21] use very different methods, where [21] performed pairwise linear regression among each SNP and each gene and then conducted multi-test correction. Although common eQTLs reported between these two studies provided replication, we argue that different SNP eQTLs selected between these two studies should be investigated more carefully. One possibility is that as these two studies using two different models, their results may reflect different angles of dissecting the underlying biology of how genetic variants affect gene expression. These two methods thus are complementary to each other.

To visualize the eQTL associations for the chosen gene, C22ORF34, we build a network using Cytoscape [6]. In Figure (10), we show the observed tagSNPs and associated genes. The eQTL associations are represented by the solid edges between the SNP eQTLs (nodes in blue rectangles) and the target gene (the green node in diamond).

Additionally, we overlap the selected SNPs that are associated with gene expression of LOC391322, with SNPs reported to be associated with human traits or diseases in the GWAS catalog [13]. As summarized in Tab-(2), we found that some of our SNPs observed were reported in recent studies to increase susceptibility in particular diseases. For example, $rs713875$ was reported as a susceptibility locus in Crohn's Disease [15]. Another example is $rs11089637$ that was reported as a risk locus for rheumatoid arthritis [29].

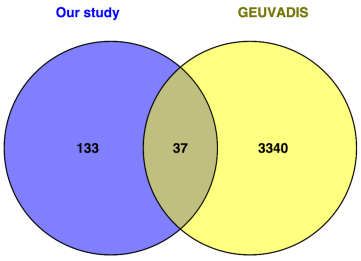

Figure 9: Venn diagram of the genes associated with SNP eQTLs found in our study and the GEUVADIS study [21]

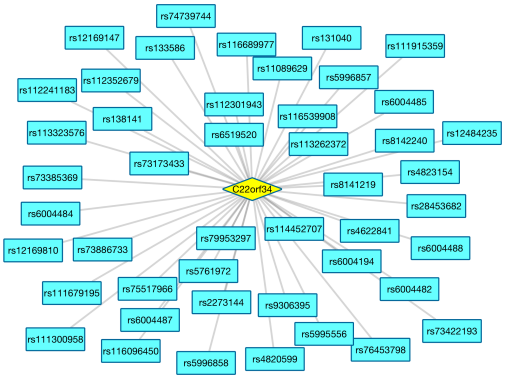

Figure 10: A network visualization of the observed SNP eQTLs associated with the expression of C22ORF34.

4. CONCLUSION

In this paper, we considered the problem of retrieving eQTL associations in a high-dimensional Gaussian graphic model. We formulated our problem with the linear model and in the context of sparse regression. We proposed two novel methods called sequential/parallel LASSO which can be combined to handle high-dimensional problems in a "divide and conquer" manner. The parallel LASSO can be parallelized to alleviate computing time, and the sequential LASSO is naturally extendable to a systematic data stream.

Theoretical analysis showed that our methods are consistent if exact recovery is achieved in LASSO with sufficient sample sizes. In addition to theoretical analysis, we performed extensive simulations on synthetic data to validate our methods. When applying to the human genomics data set, we identified SNP eQTLs that have been previously reported to be associated with human traits including diseases.

When applying sequential/parallel LASSO, there would be several reasons for having false positives and false negatives such as insufficient sample size, inappropriate λ, ill-

Table 2: A table showing all selected SNPs that are associated with human traits/diseases in the GWAS catalog.

Information of the selected SNPs overlapping with GWAS catalog						
SNPs	**Region**	**Chr_pos**	**Reported Gene(s)**	**Mapped_gene**	**Context**	**Associated Disease/Trait**
rs713875	22q12.2	30196498	MTMR3	RPS3AP51 - LIF	Intergenic	Crohn's disease
rs5754217	22q11.21	21585386	UBE2L3	UBE2L3	intron	Systemic lupus erythematosus
rs5754217	22q11.21	21585386	UBE2L3, YDJC	UBE2L3	intron	Red blood cell traits
rs5754217	22q11.21	21585386	UBE2L3	UBE2L3	intron	Systemic lupus erythematosus
rs181362	22q11.21	21577779	UBE2L3	UBE2L3	intron	HDL cholesterol
rs11089937	22q11.22	22152176	VPREB1	IGL	NA	Periodontitis (PAL4Q3)
rs11089637	22q11.21	21624807	UBE2L3, YDJC	UBE2L3 - YDJC	Intergenic	Rheumatoid arthritis
rs11089637	22q11.21	21624807	UBE2L3, YDJC	UBE2L3 - YDJC	Intergenic	Rheumatoid arthritis

behaved dataset or failure to meet the ideal conditions of LASSO. For the problem of insufficient sample size, we proposed to use a multi-round method to suppress false positives with the trade-off of additional time consumption. That is, we used multi-round method to improve precisions with the cost of decreasing recall rates. Using a balanced criteria F_1 score combining precision and recall rates, we concluded that more rounds gave us better performance with the cost of computing time. Time cost can be reduced by applying parallel LASSO and also we use the warm-start strategy when scanning over all candidates of λ's for cross validations.

Sequential/Parallel LASSO involves different combinations of several classical LASSO selectors. Hence, for each LASSO selector we need to set proper λ to ensure the sequential/parallel LASSO method works. Here, we used cross validation to set λ in each LASSO selection step. Although effective, cross validation tends to weaken the regression effect and provides denser results comparing with the ground-truth. In the future, we will explore other parameter learning methods and potentially encode priors from biological domain knowledge.

Ill-behaved dataset is another issue because the exact recovery property of sequential/parallel LASSO is based on a Gaussian graphical model. We assumed Gaussian explanatory variables, a linear model and Gaussian noises for our model. In practice, this Gaussian assumptions might not be satisfied although we can transfer data to be Gaussian distributed. We preformed simulation for uniformly discrete distributed explanatory variables with linear model and it worked well. However, the theoretical performance guarantee was based on Gaussian distribution of the data.

In this study, our methods address the problem of learning on a single task, where each gene is treated as an independent response variable. One extension to this method is to apply multi-task learning methods that can handle all the tasks, and thus capture the correlations or structures among the tasks. For example, we can apply multi-task LASSO models previously used in eQTL analysis[19, 24, 11]. In addition to LASSO based methods, we can further extend the framework to use other sparse models such as sparse graphical models [10, 39, 8].

The essence of our methods is that we provide an extendable framework encoding classical LASSO models to find "direct" links between features and labels, under the assumption of Gaussian distributions. Such a framework can be adapted for various applications, where we can encode many different machine learning models, not necessarily a LASSO variation or sparse learning models. Since the sequential/parallel LASSO requires Gaussian graphical model and standard condition for classical LASSO only, our methods can be easily extended to more general scenarios. For a different model, we may need to theoretically and experimental assess their performance when applying this sequential and parallel procedure.

5. REFERENCES

[1] The 1000 genomes project. http://www.1000genomes.org.

[2] dbsnp: Database for short genetic variations. www.ncbi.nlm.nih.gov/snp/.

[3] dbvar: Database of genomic structural variation. www.ncbi.nlm.nih.gov/dbvar.

[4] Oliveros, j.c. (2007-2015) venny. an interactive tool for comparing lists with venn's diagrams. http://bioinfogp.cnb.csic.es/tools/venny/index.html.

[5] Package: Plink (including version number) author: Shaun purcell url: http://pngu.mgh.harvard.edu/purcell/plink/ purcell s, et al (2007). plink: a toolset for whole-genome association and population-based linkage analysis. american journal of human genetics, 81.

[6] Shannon p, et al. (2003). cytoscape: a software environment for integrated models of biomolecular interaction networks. genome res. nov;13(11):2498-504.

[7] Shi lab webpage. http://shilab.uncc.edu/.

[8] E. B. and A. RP. Bayesian Structured Sparsity from Gaussian Fields. *arXiv preprint*, page arXiv:1407.2235v1, 2014.

[9] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[10] X. Chen, S. Kim, Q. Lin, J. G. Carbonell, and E. P. Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *Preprints, arXiv*, 2010.

[11] X. Chen, X. Shi, X. Xu, Z. Wang, R. Mills, C. Lee, and J. Xu. A two-graph guided multi-task lasso approach for eqtl mapping. In *International Conference on Artificial Intelligence and Statistics*, pages 208–217, 2012.

[12] W. Cheng, X. Zhang, Z. Guo, Y. Shi, and W. Wang. Graph regularized dual lasso for robust eqtl mapping.

Bioinformatics, Special Issue of the Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB)., 30(12):i139–i148, 2014.

[13] W. D, M. J, M. J, B. T, H. P, J. H, K. A, F. P, M. T, H. L, and P. H. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research*, 42(7):D1001–D1006, July 2014.

[14] P. Devyver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.

[15] A. Franke, D. P. McGovern, J. C. Barrett, K. Wang, G. L. Radford-Smith, T. Ahmad, C. W. Lees, T. Balschun, J. Lee, R. Roberts, et al. Genome-wide meta-analysis increases to 71 the number of confirmed crohn's disease susceptibility loci. *Nature genetics*, 42(12):1118–1125, 2010.

[16] B. Hajek. Notes for ece 534: An exploration of random processes for engineers. 2014.

[17] J. Kim and H. Park. Fast active-set-type algorithms for l1-regularized linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 397–404, 2010.

[18] S. Kim and E. P. Xing. Statistical Estimation of Correlated Genome Associations to a Quantitative Trait Network. *PLoS Genetics*, 5(8):e1000587, 2009.

[19] S. Kim and E. P. Xing. Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity. *ICML*, 2010.

[20] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[21] T. Lappalainen, M. Sammeth, M. R. Friedländer, P. ACąőt Hoen, J. Monlong, M. A. Rivas, M. Gonzàlez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira, et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511, 2013.

[22] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[23] S. Lee and E. P. Xing. Leveraging input and output structures for joint mapping of epistatic and marginal eqtls. *Bioinformatics*, 28(12):i137–i146, 2012.

[24] S. Lee, J. Zhu, and E. P. Xing. Adaptive Multi-Task Lasso: with Application to eQTL Detection. *NIPS*, 2010.

[25] S. Luo and Z. Chen. Sequential lasso cum ebic for feature selection with ultra-high dimensional feature space. *Journal of the American Statistical Association*, 109(507):1229–1240, 2014.

[26] G. Mateos, J. A. Bazerque, and G. B. Giannakis. Distributed sparse linear regression. *Signal Processing, IEEE Transactions on*, 58(10):5262–5276, 2010.

[27] G. Mcvean and The 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

[28] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.

[29] Y. Okada, D. Wu, G. Trynka, T. Raj, C. Terao, K. Ikari, Y. Kochi, K. Ohmura, A. Suzuki, S. Yoshida, et al. Genetics of rheumatoid arthritis contributes to biology and drug discovery. *Nature*, 506(7488):376–381, 2014.

[30] J. Pickrell and et al. Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Science*, 464(7289):768–72, 2010.

[31] Tibshirani. Model selection and validation 2: Model assessment, more cross-validation. 2013.

[32] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[33] R. J. Tibshirani et al. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7:1456–1490, 2013.

[34] M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on*, 55(5):2183–2202, 2009.

[35] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[36] W. Z, H. B, X. J, and S. X. A sparse learning framework for joint effect analysis of copy number variants. *IEEE Transactions on Computational Biology and Bioinformatics (TCBB)*, page to appear, 2015.

[37] W. Z, X. J, and S. X. Cnvnet - combining statistical learning and biological networks to characterize joint effect of copy number variants. *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM-BCB).*, pages 466–472, 2014.

[38] W. Z, X. J, and S. X. Finding alternative eqtls by exploring sparse model space. *Journal of Computational Biology*, 21(5):385–393, 2014.

[39] L. Zhang and S. Kim. Learning gene networks under snp perturbations using eqtl datasets. *PLoS Computational Biology*, pages 466–472, 2014.