Correctness and Algorithmic Efficiency of a Method for Systems Genetics

Jan Michael Yap*
Department of Computer Science
University of the Philippines Diliman
jcyap@dcs.upd.edu.ph

Henry Adorna
Department of Computer Science
University of the Philippines Diliman
ha@dcs.upd.edu.ph

Ramil Mauleon
T.T. Chang Genetic Resource Center
International Rice Research Institute
r.mauleon@irri.org

Eduardo Mendoza
Department of Membrane Biochemistry
Max Planck Institute of Biochemistry
mendoza@biochem.mpg.de

1. ABSTRACT

In systems biology and genetics, data integration is a key task towards providing a multi-perspective analysis of biological phenomenon. To validate algorithms developed for such purposes, empirical testing is usually done. Theoretical analysis of the algorithm would further highlight properties of a method. In this paper, a method for integrating genetic marker, quantitative trait loci (QTL), phenotypic, and gene sequence data to identify candidate genes causal to a trait is analyzed for algorithmic effectivity and efficiency. Results of the analysis prove the correctness of the method. Additionally, the results show that the method has time complexity $O(ng(m+g))$, while the space complexity is $O(n(m+g+q))$, where n is the number of individuals, g is the number of genes, m is the number of markers used as input, and q is the number of quantitative trait loci associated with the trait.

2. INTRODUCTION

There is no denying how systems biology has become a thriving discipline in recent years. Endeavors including, but not limited to, modeling [1, 2, 3, 4], predictive analysis [2, 3], and exploratory analysis[5, 6, 7, 8] now has an infusion of systemic approaches to carry out research activities and analysis. Genetics has also benefited from the systemic approach of analyzing genetic and genomic data. Aptly referred to as systems genetics [2, 4], the discipline's approach to analysis involve the integration of several types of data sets. One interesting application of systems genetics is inferring candidate genes causal to a trait [3, 6, 7]. Here, several data sets are used in conjunction with gene expression data to identify which genes are potentially causal towards the expression of a particular trait. Such analyses thus gives more resolution to the inference made by the endeavor by introducing supporting factors that are not "visible" when using gene expression data alone.

With the growth of systems biology and systems genetics, the need to develop computational methods for carrying out tasks relevant to the field has also become evident. The methods come in different varieties, but most prominent of which are graph-theoretic approaches [2, 4] and statistics-based techniques [3, 5, 6, 7]. To validate the "usefulness" of such methods, empirical testing is employed. This is done by using existing data sets or generating simulated ones as input for the method, and the output is checked against a reference to check if they are similar. For practical purposes, analyzing the capability of computational methods this way definitely is an intuitive manner of checking if the method is useful and, to a certain point, accurate in processing the input into output. On the one hand, theoretical analysis of the method, particularly on its correctness and algorithmic efficiency, would further enhance the underlying properties of the method.

Proving the algorithm's correctness in a nutshell is showing that the method comes up with the proper output given a (proper) input [9]. This will effectively show that the processing done by the method, under certain assumptions, will work as intended and indeed adheres to specifications. Analyzing algorithmic efficiency is tantamount to obtaining the theoretical running time and memory requirement of the method, i.e. time and space complexities. Such analysis will provide the behavior of the algorithm with regard to processing speed and memory space needed as the nature of the input, particularly the input size, changes.

To emphasize the merits of performing the theoretical analyses, a systems genetics method will be examined and the implications of the results will be discussed. The succeeding sections of the paper are as follows: First, the problem of inferring candidate causal genes to a trait will be described and formally defined. Next, the systems genetics method for inferring candidate causal genes and its features will be presented. Afterwards, the proof of correctness for the method will be given. The method's time and space complexity is shown next. Finally, some of the implications of the results of the analysis is discussed.

3. INFERRING CANDIDATE GENES CAUSAL TO A TRAIT

The task of inferring candidate genes causal to a trait involves the selection from a set of genes a smaller subset which are potentially causal to the expression of a trait. A systems genetics approach to the endeavor is done via integrating different kinds of genetic and genomic data sets [3, 4]. The typical strategy for systems genetics-based inference is integration of sequence data, gene expression data, and phenotypic (i.e. trait measurement) data [3, 6, 7].

3.1 Sequence and Gene Expression Data

Sequence data in the candidate causal gene inference task typically involves genetic markers, areas in a species' genome whose locations are known and whose states can be observed and recorded. Location of each marker is typically "stored" in a marker map. In some cases [3, 8], certain areas of the genome called quantitative trait loci or QTLs are also included in the inference. QTLs are areas of the genome that have been identified to be causing the expression of a trait depending on their state. Simplistically, QTLs are identified (in a technique called QTL mapping) using statistical analysis of genetic markers and phenotypic data [10].

Gene expression data consists of measurements of how expressed a gene is, usually by measuring the amount of messenger RNAs (mRNAs) or proteins associated with the gene. As with markers, genes are also known locations in the genome. A main characteristic of a gene though, and thus differentiating them from markers, is that they are regions that are coded and transcribed into amino acids which serve as building blocks for proteins.

3.2 Formal Definition of Systems Genetics-Based Candidate Causal Gene Inference

Before formally defining candidate causal gene inference, some preliminary notions will be introduced. First, individuals that make up the population to be used for the inference are defined:

Definition 3.1 (Individual). *Let $M = \{1, 2, 3, \ldots, m\}$ be a finite set of indices each uniquely representing a marker. Define a 2-tuple $Ind = (S, trait\_val)$, representing an individual to be used in the inference where $S = \{S_i | i \in M\}$ is a set of discrete random variables where $S_i \in \{0, \ldots, k\}$ for $i \in M$ and some $k \in \mathbb{Z}^+$. The set S represents a set of markers states for the individual. $trait\_val \in \mathbb{R}$ is the measured trait value for an individual.*

Next, QTLs are defined based on the manner of their identification, as mentioned in the previous subsection:

Definition 3.2 (QTL). *Let $P = \{Ind_i | 1 \leq i \leq n\}$. We thus refer to components of Ind_i as $(S^i, trait\_val_i)$ to mean the marker state set and trait value for the i^{th} individual in P, respectively. Define $Q \subset M$ be a set of QTLs such that $\forall q \in Q$ and a function $c : \mathbb{Z} \times \mathbb{R} \to \mathbb{R}$, the cost function*

$$\sum_{1 \leq i \leq n} c(S^i_q, trait\_val_i) < \varepsilon$$

for some threshold value $\varepsilon \in \mathbb{R}, \varepsilon \geq 0$.

Finally, here is the formal definition of the systems genetics-based approach to inferring candidate genes causal to a trait:

Definition 3.3 (Systems Genetics-based Approach to Inferring Candidate Genes Causal to a Trait). *Let $G = \{1, 2, 3, \ldots, g\}$ be a finite set of indices each uniquely representing a gene. Define the Systems Genetics-based Approach to Inferring Candidate Genes Causal to a Trait as looking for a set of candidate genes causal to a trait $G_{CCT} \subset G$ such that given a set of individuals P, a set of QTLs Q, and a function $f : S \times G \times \mathbb{R} \to \mathbb{R}$, the cost function*

$$\sum_{1 \leq i \leq n} f\left(\left\{S^i_q | q \in Q\right\}, G_{CCT}, trait\_val_i\right) < \delta$$

for some threshold value $\delta \in \mathbb{R}, \delta \geq 0$.

3.3 Using Gene Sequence Data In Lieu of Gene Expression Values

The strategy of integrating sequence, gene expression, and phenotypic data assumes that sufficient number of samples for each type of data exists. This may not be the case however, especially with gene expression data. For example, if one were to apply the framework in [3] to rice salt stress response and only using publicly available data, one would find that available and appropriate rice gene expression data would be insufficient (around 11 samples are only available as per [11]).The workaround to this is to instead use gene sequence information instead of relying on their expression values. The basis for this alternative procedure is similar to the idea of QTL mapping: identifying regions of the genome that affect expression of a trait based on their state. This modified approach thus serves as the basis for the development of the proposed method.

4. A SYSTEMS GENETICS METHOD FOR INFERRING CANDIDATE CAUSAL GENES

The method being presented integrates several types of data to obtain a set of candidate genes which are inferred to be causal to the expression of a trait. More particularly, the data integration method is designed for use on data obtained from recombinant inbred lines or RILs. The preference for RILs largely stems on reducing the number of states a particular location on a genome could have [10] and thus simplifies designation of gene states and even the calculation of similarity of the states of adjacent loci. The preceding premise is a crucial factor in the execution of the algorithm to be shown in a later section. Figure 1 shows the framework of the algorithm.

4.1 Input and Output

The method takes in as input three kinds of data from a set of individuals: marker map data, marker state and quantitative trait loci (QTL) data, phenotypic data, and gene map data. Markers are known regions within the genome whose state can be observed and documented. QTL are regions in the genome that were identified to affect the expression of a (quantitative) trait based on their state. The process of identifying QTLs is known as QTL mapping. Genes, apart

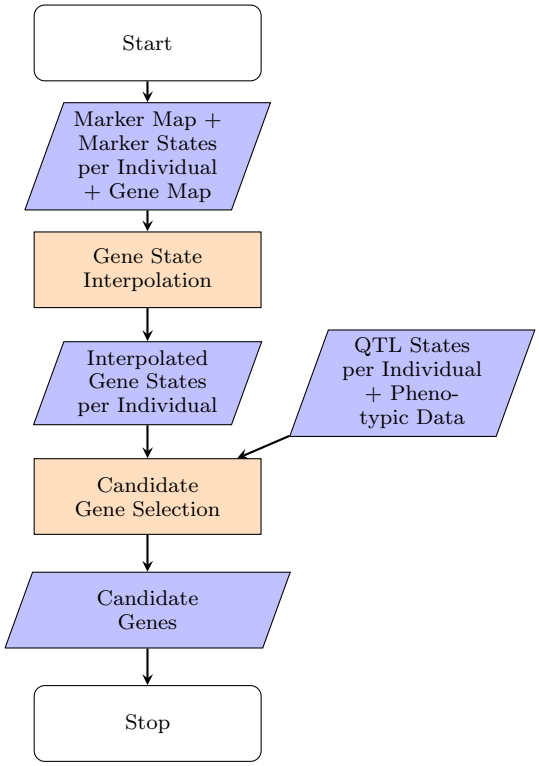

Figure 1: Flowchart of the Data Integration Algorithm

from the common notion that they are units of heredity, are regions within the genome that can be transcribed and coded to form amino acids, and subsequently proteins. As markers, QTLs, and genes are regions within a genome, their location particularly in what chromosome they are located as well as their distance within the chromosome is known.

The marker map data contains the location of the markers to be used for the analysis. The location includes containing the chromosome where the marker is located, and the displacement of the marker from the physical beginning of the chromosome. The location may be represented as an integer if it refers to a physical location in the chromosome, the unit of which is in megabase pairs (Mb) or as a real number if it refers to a measure of genetic linkage, the unit of which is in centimorgans (cM). Either way, the marker map can be represented programmatically as a two-dimensional array (later referred to as *mmap*) with 2 rows and m columns, wherein the columns represent the markers, the first row containing the chromosome where the markers are located, and the second row containing the location of the marker within the chromosome.

The marker state data on one hand contains the state of each marker of each individual in the data set. Marker states can be represented as nonnegative integers, and that the range is limited, say from 0 to k - 1, where k is the of alleles used in the analysis. This can be represented in programs as two-dimensional arrays with n rows and m columns. n represents the number of individuals from which the data was obtained, while m represents the number of markers. In the later parts of the paper, the reference to this data is *mstate*. The QTL data can be fashioned in a similar manner, only this time the number of rows is $q \leq m$, where q is the number of QTLs in a trait.

Phenotypic data contains the measurements of the trait obtained from the individuals. The values are real numbers. In the case of qualitative or nominal data, each value is assigned a numerical equivalent. The structure used to represent this is a linear array of length n.

The gene map data is somewhat similar to the marker map data in that it contains the location of each gene to be used in the analysis. The difference is that the location/offset of the "start" of the gene sequence (which we will refer to subsequently as the *5' end*) as well as the "end" of the sequence (the *3' end*) within a chromosome is noted. As such, a two-dimensional array (later referred as *gmap*) with 3 rows and g columns (g being the number of genes) can represent the gene map data. In this structure, the columns represent the genes, the first row contains the chromosome where the gene is located, the second row contains the location of the 5' end within the chromosome, and the third row contains the location of the 3' end.

The output of the algorithm is a set of candidate genes inferred to be causal to the expression of the trait. This can be represented as a linear array of length g, where an index i in the array correspond to a gene from the gene map data. The values of the elements in the array are of boolean data type. An element has a value of "true" if the gene corresponding to index i is a candidate gene (i.e. selected by the method as a potentially causal to the expression of the trait), or "false" otherwise.

4.2 The Algorithm

The algorithm is divided into two phases: the gene state interpolation phase and the candidate gene selection phase. In the gene state interpolation phase, the (allelic) state of the gene is interpolated based on the (allelic) state of the markers nearest to the ends of its sequences. The output of this phase is the interpolated state of each gene for each of the individuals. Algorithm 1 gives the pseudocode for the gene state interpolation phase. To determine the value of the variable representing the allelic state of a particular gene, genetic markers used in QTL mapping are employed. Markers near the 5' and 3' ends of a gene's sequence are identified (lines 3-20). Without loss of generality, assume that there are only two possible marker states. A simple two-point probability formula for 2-way selfed RILs [12] can be employed to infer the allelic state of one end of the gene sequence based on the state of the nearest marker. Given two adjacent points/loci in the genome of a 2-way selfed RIL, x_i, x_j, the probability that the allelic state of the two points/loci are different is given by :

$$p(g_{x_i} \neq g_{x_j}) = \frac{2r}{1+2r}$$

where g_{x_i}, g_{x_j} is the allelic (or marker) state of point/locus x_i, x_j and r is the recombination rate of x_i and x_j. The

Algorithm 1 Gene State Interpolation

Input: $mstate[n][m], mmap[2][m], gmap[3][g]$
Output: $gstate[n][g]$

```
 1: for i ← 1 to n do
 2:     for j ← 1 to g do
 3:         min_dist_mkr_5 ← ∞
 4:         closest_mkr_5 ← ∞
 5:         min_dist_mkr_3 ← ∞
 6:         closest_mkr_3 ← ∞
 7:         for k ← 1 to m do
 8:             if mmap[1][k] = gmap[1][k] then
 9:                 dist_mkr_5 ← |mmap[2][k] − gmap[2][j]|
10:                 if dist_mkr_5 < min_dist_mkr_5 then
11:                     min_dist_mkr_5 ← dist_mkr_5
12:                     closest_mkr_5 ← k
13:                 end if
14:                 dist_mkr_3 ← |mmap[2][k] − gmap[3][j]|
15:                 if dist_mkr_3 < min_dist_mkr_3 then
16:                     min_dist_mkr_3 ← dist_mkr_3
17:                     closest_mkr_3 ← k
18:                 end if
19:             end if
20:         end for
21:         rrate_mkr_5 ← recombn_fn(min_dist_mkr_5)
22:         prob_mkr_5_state_diff ← (2×rrate_mkr_5)/(1+2×rrate_mkr_5)
23:         rrate_mkr_3 ← recombn_fn(min_dist_mkr_3)
24:         prob_mkr_3_state_diff ← (2×rrate_mkr_3)/(1+2×rrate_mkr_3)
25:         if prob_mkr_5_state_diff ≤ 0.5 then
26:             5_state ← mstate[i][closest_mkr_5]
27:         else
28:             5_state ← alt_state(mstate[i][closest_mkr_5])
29:         end if
30:         if prob_mkr_3_state_diff ≤ 0.5 then
31:             3_state ← mstate[i][closest_mkr_3]
32:         else
33:             3_state ← alt_state(mstate[i][closest_mkr_3])
34:         end if
35:         if (5_state = 3_state) then
36:             gstate[i][j] ← 5_state
37:         else
38:             gstate[i][j] ← 0
39:         end if
40:     end for
41: end for
42: return gstate
```

computation of the two-point probability is done in lines 21-24. In the pseudocode, the auxiliary function $recombn\_fn$ in lines 21 and 23 converts the (physical) distance of a marker and a gene to the corresponding recombination rate, and is done so in a constant number of steps.

If $p(g_{x_i} \neq g_{x_j})$ is less than or equal to 0.5, then the allelic state of the 5' or 3' end of the gene sequence should the same as the state of the nearest marker (lines 25-26 and lines 30-31). Otherwise, the 5' or 3' end's allelic state is set to the "alternative" state, i.e. different from the nearest marker's state (lines 27-28 and 32-33). The *alt_state* auxiliary function in lines 28 and 33 just returns the "alternative" state that is different from the input marker state, and is done so in a constant number of steps. To finally infer the allelic state of the gene sequence, the inferred allelic states of both 5' and 3' ends are checked (line 35). If the allelic states of both ends are the same, then the probe set assumes that allelic state (line 36). If the allelic states of both ends are different, then the gene can be thought of as having a heterozygous allelic state (line 38). Such genes are not included in the analysis.

The output of the gene state interpolation (referred in the pseudocode as *gstate*) in conjunction with QTL data and phenotypic data would be used as input to the candidate gene selection phase. In the candidate gene selection phase, genes inferred to be causal to the expression of a trait are selected. This is done by performing partial regression coefficient analysis on the genes using the interpolated state as variables. For the analysis, a multiple linear regression model is used:

$$\hat{Y}_{ik} = \mu_i + \sum_{j=1}^{q} \alpha_j x_{jk} + \sum_{l=1}^{g} \beta_l z_{lk}$$

where $\hat{Y}_{ik}$ is the estimated value of i^{th} phenotypic trait for individual k, μ_i is the intercept for i^{th} phenotypic trait, α_j is the additive effect of j^{th} QTL, x_{jk} is the coded variable representing allelic state of j^{th} QTL in individual k, β_l is the contributory effect of l^{th} gene, and y_{lk} is the coded variable representing allelic state of l^{th} gene in individual k. In the regression model, the intercept and the QTL-related variables are held constant, while the gene-related variables are the ones being subjected to selection. The motivation behind this is that since QTLs are already established causal factors towards trait expression, we check if there are certain genes that significantly affect trait expression *given that the QTLs are also affecting it.*

To perform selection of candidate causal genes, forward stepwise regression is done. This is performed by adding variable to the regression model, one by one, and checking which variable would significantly lessen the error of the model. The variable that gives the model the most significant drop in error is selected to be part of the regression equation. The method then iterates and checks which of the remaining variables may be selected for inclusion in the model. The selection process terminates if no other variable can provide significant drop in the error, or if all variables have been included. Forward stepwise regression was chosen as the endeavor would entail exploratory analysis on a large number of variables to be considered for inclusion, and as such the aforementioned method is suitable for the stated purpose [13].

5. PROOF OF CORRECTNESS

Proving the correctness of a algorithm would involve establishing the adherence of the algorithm to postconditions given certain preconditions [9]. With the data integration algorithm divided into two phases, proving its correctness may be done by proving that each of the two phases are correct.

In the gene state interpolation algorithm, the precondition is that the input structures, i.e. $mstate, mmap, gmap$, are of the proper form. The postcondition would be that the $gstate$ contains the interpolated states of each gene for each individual based on the state of the closest marker to each gene. The first and second for-loops (lines 1-2) account for iterating through each individual and gene. So to prove the correctness of the aforementioned loops, and thus the whole algorithm, the focus of the analysis will be mostly on the statements contained inside the second for-loop, i.e. lines 3-39 as those statements are what really assigns the state of a particular gene of a particular individual.

The first 3 statements (lines 3-6) are trivially correct as they assign the "infinity" value to the variables corresponding to the distance of the closest markers to the 5' and 3' ends of gene j (line 3 and 5) which the process intends to do. Additionally, the index of the closest marker in the marker map structure (line 4 and 6) to the 5' and 3' ends are assigned "infinity" values . These statements though are actually important in establishing the correctness of the third for-loop (lines 7-20).

To prove the correctness of the third for-loop , a loop invariant must be established [9]. The loop invariant is a condition relevant to the algorithm that holds true at any point during the execution of the loop. In this case, the loop invariant is the condition that the distance of the closest markers to the 5' and 3' ends of gene j (represented in the pseudocode as $min\_dist\_mkr\_5$ and $min\_dist\_mkr\_3$ respectively) found at any point of the loop is indeed the minimum among all others that have markers checked thus far. The base case is that before the loop begins, no markers have yet been checked, so the distance of the closest markers to the 5' and 3' end may be assigned any value as it would still be minimum. The value of infinity is chosen as the initial value as it would guarantee that a smaller distance value would be found at each iteration of the loop. The inductive step would have us assume that for some k, $1 \leq k \leq m$, we have a current value for $min\_dist\_mkr\_5$ and $min\_dist\_mkr\_3$. Now at the next iteration step k + 1, we check the (k + 1)th marker. If the distance between the (k + 1)th marker and the 5' and/or 3' end of gene j is lesser than the current values of $min\_dist\_mkr\_5$ and/or $min\_dist\_mkr\_3$ (lines 10 and 15), then (k + 1)th marker is thus closer to the 5' and/or 3' end of gene j, and the value/s of the minimum distance/s are updated (lines 11 and 16) and the index of the (k + 1)th marker is stores (lines 12 and 17). If the distance between the (k + 1)th marker and the 5' and/or 3' end of gene j is greater than the current values of $min\_dist\_mkr\_5$ and/or $min\_dist\_mkr\_3$, then the status quo with respect to the pertinent variables is retained. Either way, the loop invariant condition is preserved, and thus proves the correctness of the for-loop.

The succeeding statements get the recombination rate of the closest marker to the 5' and itself based on their distance from each other (line 21), and getting the probability that their states would be different (line 22). The same process of the getting the recombination rate and probability is applied to the 3' end of gene j and the closest marker to it (lines 23-24). Lines 25-34 assigns the interpolated state of the 5' and 3' ends based on the probabilities computed in lines 22 and 24. The idea is if the probability that the 5' end's state and that of the closest marker to it is at most 50% (line 25), then we just set the interpolated state of the 5' end to be the same as that of its closest marker (line 26). Otherwise, we set the state of the 5' end to be the "alternative" state different from the closest markers (lines 27-28). The same test and process for the 3' end and its closest marker is performed in lines 30-33.

Finally, the interpolated state for gene j will be assigned. The condition is if the interpolated state of the 5' end and the 3' end are equal (line 35), then the state of the whole gene j can be putatively assigned the state of both ends (or in the case of the pseudocode, the state of the 5' end as per line 36. If they are not equal, then the interpolated state of gene j can be interpreted as "heterozygous", i.e. "halfway" between the two alternative states. In this scenario, gene j for individual i can be "discarded" from the succeeding regression analysis, and as such is assigned a value of zero (line 38).

Aggregating the analysis of the statements in the second for-loop, each iteration does give the interpolated state of a particular gene for a particular individual based on the closest markers to its 5' and 3' ends. The second for-loop should therefore give the interpolated states of all genes in an individual. Going further, the first for-loop would thus give the interpolated states of all genes for all individuals, and hence would prove the correctness of the gene interpolation phase.

As to the correctness of the candidate gene selection phase, the reader is referred to [13] and [14]. To summarize the pertinent parts, the forward selection algorithm will select the (sub)set of variables that will lessen the error of a multiple linear regression model. In the context of the problem the algorithm is trying to solve, the method will select the (sub)set of genes that will best "predict" the value of some trait. This is tantamount, albeit simplistically, that the (sub)set of genes chosen in the candidate gene selection phase are indeed causal towards the expression of a trait, and thus proves the correctness of the phase, and that of the whole data integration method.

6. TIME AND SPACE COMPLEXITY

As with proving correctness, we obtain the time and space complexity of the whole method by getting the corresponding complexities of each phase and then aggregating them. First, we acquire the time and space complexities of the gene state interpolation phase. For time complexity, the algorithm only contains nested for-loops with simple statements, each loop running in linear time with respect to the variable being iterated. As mentioned prior the $recombn\_fn$ and alt_state auxiliary functions are just implementations of simple functions that run in constant time. Hence, the execution of the for-loops dominate the computational running time of the algorithm. The time complexity is thus $O(ngm)$, where n is the number of individuals (iterated through in the for loop in line 1), g is the number of genes (iterated through in the for loop in line 2), and m is the number of markers (iterated through in the for loop in line 7).

Next, we obtain the space complexity of the gene state interpolation phase. For the input, which subsequently are the structures used and manipulated in the course of the interpolation phase, the marker states per individual ($mstate$) has space complexity $O(nm)$, the marker map has $O(2m)$, and for the gene map, $O(3g)$. The output which is a structure containing the interpolated gene states per individual has space complexity $O(ng)$. The space complexity for the gene state interpolation phase is thus $O((n+2)m+(n+3)g)$ or simply $O(n(m+g))$.

The time and space complexity for the candidate gene selection phase is tantamount to getting the complexity of forward stepwise regression. In [15], it was noted that the time complexity of the aforementioned method is $O(nv^2)$, where n is the number of individuals and v is the number of variables for inclusion in the regression model. In the case of the candidate gene selection, the variables to be included in the model are the genes, hence $v = g$, and thus the time complexity of the candidate gene selection is $O(ng^2)$.

As for space complexity, the structures significantly contributing to it are the following: the structure containing the interpolated gene states per individual with space requirement $O(ng)$; the structure for the phenotypic data wherein measurement of the trait for n individuals are stored, hence $O(n)$; and the structure for the QTL state where the state of q QTLs per individual are stored amounting to $O(nq)$ space. The space complexity of the candidate gene selection phase is thus $O(n(g+q))$.

All in all, the time complexity of the whole method is $O(ngm)+O(ng^2) = O(ng(m+g))$. The space complexity on the one hand is $O(n(m+g)) + O(n(g+q)) = O(n(m+g+q))$.

7. DISCUSSION

7.1 Correctness of the algorithm and statistical "validity"

Through the analysis, the data integration method is algorithmically correct in that the expected proper output is obtained upon running the algorithm with the proper input. It should be noted though that there are simplifying assumptions, particularly on those segments where statistical methods are implemented. In lines 25-28, the condition is discretized by using a simple cut-off/threshold (0.5) to determine the interpolated state of the 5' and 3' end of a gene. Statistical theory would dictate that a (proper) test of hypothesis be done to further strengthen the claim on what state should be assigned. Additionally, in the candidate gene selection phase, a linear model is assumed and more so, least squares regression is employed in stepwise regression without regard on the distribution of the variables in the data set, e.g. if normality assumption hold, samples are truly identically, independently distributed. For certain data sets though, the simplified assumptions do not hinder the utility of the method. In an accompanying paper [16], the method was applied on rice salinity stress response data, and still yielded meaningful results. Furthermore, the method is designed to be more of a framework rather than a rigid implementation. Hence, other statistical techniques may be added to the data integration technique as needed.

7.2 Number of genes and time and space complexity

For time and space complexity, it can be seen that the running time and memory requirement both depend on the number of individuals, the number of markers, and the number of genes included as input for the framework. In terms of cardinality, genes typically greatly outnumber the number of individuals and number of markers, even when combined. While the number of individuals used in experiments are typically in the range of hundreds and the number of markers from hundreds to a few thousands, the number of genes are usually tens of thousands. Taking this into account, we can simplify the time complexity of the method to $O(g^2)$, while the space complexity is $O(g)$. Hence, it can be simply stated that the running time of the data integration method to be quadratic, while the memory requirement is linear, both with respect to the number of genes.

It is recommended though that a filtering step be done to significantly cut down the number of genes to be included in the analysis. An example of filtering would be to include only those genes that are significantly differentially expressed when subjected to a particular condition. For consistency, it should be made sure that the rest of the data sets, particularly the QTL and phenotypic data, should also have been obtained from samples that were subjected to similar conditions as those samples from which significant differential expression tests were done.

7.3 Comparing time and space complexity with other inference methods

In Section 3 of this paper, some works were cited that detail other causal gene inference methods [3, 6, 7]. The method in [3] employs a Bayesian framework for causality model building and analysis between genes and a trait. Using the same notion and variables g, m, n, and q in the previous section, the time and space complexity of the aforementioned method as described in the Supplementary Information of the literature is $O(ngq)$ and $O(n(m+g+q))$ respectively. The technique in [6] employs the construction and analysis of gene coexpression (sub)networks and then mapping them back to associated loci in the genome. Time complexity of the aforementioned method is $O(ng(g+q))$ while space complexity is $O(n(g^2+q))$. A Bayesian network was constructed and analyzed for candidate causal gene inference in [7]. The method's time complexity is $O(ngq)$ and the space complexity is $O(n(g+q))$.

While the space complexity of the proposed method is comparable relative to the aforementioned methods, it is relatively "slower". This is due to the need for an additional preprocessing phase, i.e. the gene state interpolation phase, in the analysis. It should be pointed out though that the proposed method assumes that insufficient gene expression data is available - something that is contrary to the assumption of the three methods used for comparison. Hence, the trade-off here is that candidate causal gene inference can still performed without relying on gene expression at the cost of additional steps to be done in carrying out the task.

8. ACKNOWLEDGMENT

The corresponding author would like to thank the Engineering Research and Development for Technology (ERDT) program for funding this research.

9. REFERENCES

[1] Chu J, Weiss ST, Carey VJ, Raby BA. *A graphical model approach for inferring large-scale networks integrating gene expression and genetic polymorphism*. BMC Syst Biol 2009; 3(55).

[2] Druka A, Druka I, Centeno AG, Li H, Sun Z, Thomas WTB, Bonar N, Steffenson BJ, Ullrich SE, Kleinhofs A, Wise, RP, Close TJ, Potokina E, Luo Z, Wagner C, Schweizer GF, Marshall DF, Kearsey MJ, Williams RW, Waugh R. *Towards system genetic analyses in barley: Integration of phenotypic, expression, and genotype data into GeneNetwork*. BMC Genet 2008; 9(73).

[3] Schadt E, Lamb J, Yang X, Zhu J, Edwards S, GuhaThakurta D, Sieberts SK, Monks S, Reitman M, Zhang C, Lum PY, Leonardson A, Thieringer R, Metzger JM, Yang L, Castle J, Zhu H, Kash SF, Drake TA, Sachs A, Lusis AJ. *An integrative genomics approach to infer causal associations between gene expression and disease*. Nat Genet 2005; 37(7): 710-717.

[4] Lusis AJ, Attie AD, Reue K. *Metabolic syndrome: from epidemiology to systems biology*. Nat Rev Genet 2008; 9: 819-830.

[5] Al-Shahrour F, Diaz-Uriarte R, Dopazo J. *Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information*. Bioinformatics 2005; 21(13): 2988-2993.

[6] Kang HP, Yang X, Chen R, Zhang B, Corona E, Schadt EE, Butte AJ. *Integration of disease-specific single nucleotide polymorphisms, expression quantitative trait loci and coexpression networks reveal novel candidate genes for type 2 diabetes*. Diabetologia 2012; 55: 2205-2213.

[7] Lionikas A, Meharg C, Derry JMJ, Ratkevicius A, Carroll AM, Vandenbergh DJ, Blizard DA. *Resolving candidate genes of mouse skeletal muscle QTL via RNA-Seq and expression network analyses*. BMC Genomics 2012; 13(592).

[8] Degnan JH, Lasky-Su J, Raby BA, Xu M, Molony C, Schadt EE, Lange C. *Genomics and genome-wide association studies: An integrative approach for expression QTL mapping*. Genomics 2008; 92(3): 129-133.

[9] Cormen TH, Leiserson CE, Rivest RL, Stein C. *Chapter 2: Getting Started*. In: Introduction to Algorithms. Second Edition. London: The MIT Press, 2001: 15-36.

[10] Collard BCY, Jahufer MZZ, Brouwer JB, Pang ECK. *An introduction to markers, quantitative trait loci (QTL) mapping and marker-assisted selection for crop improvement: The basic concepts*. Euphytica 2005; 142(1-2) 169-196.

[11] Walia H, Wilson C, Condamine P, Liu X, Ismail AM, Zeng L, Wanamaker SI, Mandal J, Xu J, Cui X, Close TJ. *Comparative Transcriptional Profiling of Two Contrasting Rice Genotypes under Salinity Stress during the Vegetative Growth Stage*. Plant Physiol 2005; 139(2): 822-835.

[12] Broman KW. *The Genomes of Recombinant Inbred Lines*. Genetics 2005; 169: 1133-1146.

[13] Miller AJ. *Finding subsets which fit well*. In: Isham V, Keiding N, Louis T, Reid N, Tibshirani R, Tong H, eds. Subset Selection in Regression. Second Edition. Boca Raton: Chapman and Hall/CRC, 2002: 37-85.

[14] Miller AJ. *Selection of Subsets in Regression Variables*. JR Statist Soc A 1984; 147(3): 389-425.

[15] Lin D, Foster DP, Ungar LH. *VIF Regression: A Fast Regression Algorithm for Large Data*. J Am Statist Assoc 2011; 106(493): 232-247.

[16] Yap JM, Mauleon R, Mendoza E, Adorna H. *A partial regression coefficient analysis framework to infer candidate genes causal to traits in recombinant inbred lines*. Manuscript submitted for publication. 2013.