IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 4, Ver. III (Jul - Aug 2018), PP 104-110
www.iosrjournals.org

An Analogy of Algorithms for Tagging Of Single Nucleotide Polymorphism and Evaluation through Linkage Disequilibrium

Sarala prasanna pattanaik[1], Tapan Kumar Panda[2]
1(Department of Basic Science &Humanities ,Gandhi Engineering College, India)
2(Department of Basic Science &Humanities ,Gandhi Institute For Technology, India)

Abstract: *Recent years have seen an explosive growth in biological data. It should be managed and stored for various purposes. Demand has never been greater for revolutionary technologies that deliver fast, inexpensive and accurate and easy to comprehend genome information. Here comes the relevance of tagging data. From huge large DNA sequence information scientists needed some small efficient dataset that they can do their research on and that is exactly why some optimization needed to be carried upon these big data. A subset of SNPs that are selected to represent the original information embedded in the full set of SNPs is referred to as the set of Tag SNPs. Large sequencing projects are producing increasing quantities of nucleotide sequences. The contents of nucleotide databases are doubling in size approximately every fourteen months. So to track and analyze this amount of data scientists need some small set of data that can represent the whole database characteristically. So computer scientists came up with some innovative algorithms to find tag SNPs. We have done a comparative study by implementing the popular algorithms and evaluating them by scoring LinkageDisequilibrium*

I. Introduction

Here we will be discussing about current methods for selection of informative single nucleotide polymorphisms (SNPs) using data from a dense network of SNPs that have been genotyped in a relatively small panel of subjects. We discuss the following issues: (1) Optimal selection of SNPs based upon maximizing either the predictability of unmeasured SNPs or the predictability of SNP haplotypes as selection criteria. (2) The dependence of the performance of tag SNP selection methods upon the density of SNP markers genotyped for the purpose of haplotype discovery and tag SNP selection. (3) The likely power of case-control studies to detect the influence upon disease risk of common disease-causing variants in candidate genes in a haplotype-based analysis.

To choose a subset of SNPs, [tag SNPs] that can predict other SNPs in the region with small probability of error and remove redundant information the following methods we have sincerely workedon:

- GaussAlgorithm
- Gauss-JordonAlgorithm
- GreedyAlgorithm
- Binary Optimization Algorithm

We also implemented a SNP scoring procedure to do a comparison study among all these procedures and to give an overall view of the problem structure.

II. Overview

SNP

A single-nucleotide polymorphism (SNP, pronounced snip) is a DNA sequence variation occurring when a single nucleotide A, T, C or G in the genome (or other shared sequence) differs between paired chromosomes in an individual. The genetic code is specified by the four nucleotide "letters" A (adenine), C (cytosine), T (thymine), and G (guanine). SNP variation occurs when a single nucleotide, such as an A, replaces one of the other three nucleotide letters C, G, or T. An example of a SNP is the alteration of the DNA segment AAGGTTA to ATGGTTA, where the second "A" in the first snippet is replaced with a "T". On average, SNPs occur in the human population more than one percent of the time. Because only about three to five percent of a person's DNA sequence codes for the production of proteins, most SNPs are found outside of "coding sequences". SNPs found within a coding sequence are of particular interest to researchers because they are more likely to alter the biological function of a protein. Because of the recent advances in technology, coupled with the unique ability of these genetic variations to facilitate gene identification, there has been a recent flurry of SNP discovery and detection. The most important application of SNP array is in determining disease susceptibility and consequently, in pharmaco- genomics by measuring the efficacy of drug therapies specifically for the individual. SNP-based genetic linkage analysis could be performed to map disease loci, and hence determine disease susceptibility genes for an individual. The results of the different sectors on SNP studies may help to gain insights into mechanisms of these diseases and to create targeteddrugs.

Tag SNP

Tag SNP is a representative single nucleotide polymorphism (SNP) in a region of the genome with high linkage disequilibrium (the non-random association of alleles at two or more loci). It is possible to identify genetic variation without genotyping every SNP in a chromosomal region. Tag SNPs are useful in whole-genome SNP association studies in which hundreds of thousands of SNPs across the entire genome are genotyped. Tag SNP mainly helps in analyzing the huge (bulk) snp data in very short time. With the help of tag SNP, a very large SNP data set can be represented with merely a small chunk of tag SNP data. So, it computationally easier to work with tag SNP than whole SNP haplotype. It mainly saves the computational time and memory space. Without tag SNP it would have been impossible to analyze all the SNPdata.

III. Problem SolvingMethodologies

Gauss & Gauss-Jordon Algorithm for Finding TagSNPs

In linear algebra, Gaussian elimination is an algorithmfor solving systems of linear equations.Itcanalsobeusedtofindtherankofamatrix,tocalculatethedeterminantofamatrix, and to calculate the inverse of an invertible square matrix. The method is named after Carl Friedrich Gauss. We implemented this algorithm in order to find Tag SNPs.

Gauss JordanElimination:

In linear algebra, Gauss–Jordan elimination is an algorithm for getting matrices in reduced row echelon form using elementary row operations. It is a variation of Gaussian elimination. Gaussian elimination places zeros below each pivot in the matrix, starting with the top row and working downwards. Matrices containing zeros below each pivot are said to be in row echelon form. Gauss–Jordan elimination goes a step further by placing zeros above and below each pivot, such matrices are said to be in reduced row echelon form. Every matrix has a reduced row echelon form, and Gauss–Jordan elimination is guaranteed to findit.

Computer science's complexity theory shows Gauss–Jordan elimination to have a time complexity of O(n3) for an n by n matrix (using Big O Notation). This result means it is efficiently solvable for most practical purposes. As a result, it is often used in computer software for a diverse set of applications. However, it is often an unnecessary step past Gaussian elimination. Gaussian elimination shares Gauss-Jordan's time complexity of O(n3) but is generally faster. Therefore, in cases in which achieving reduced row echelon form over row echelon form is unnecessary, Gaussian elimination is typically preferred.

The Algorithm (GaussElimination)

Step1: The first part (Forward Elimination) reduces a given system to either triangular or echelon form, or results in a degenerate equation, indicating the system has no unique solution but may have multiplesolutions(rank<order).Thisisaccomplishedthroughtheuseofelementaryrowoperations.

Step2: The second step uses back substitution to find the solution of the system above. Stated equivalently for matrices, the first part reduces a matrix to row echelon form using elementary row operations while the second reduces it to reduced row echelon form, or row canonical form.

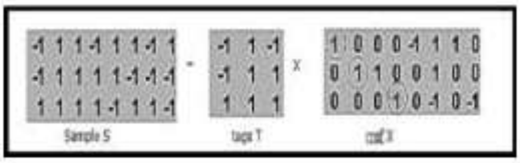
Step3: From this canonical form the tag SNP are chosen for a particular set of haplotype.

The Algorithm for (Gauss JordanElimination):

Step1: Using Gauss-Jordan Elimination, find Row Reduced Echelon Form (RREF) X of sample matrixS

Step2: Extract the basis T of sample S **Step3**: Factorize sample S = T x reff [X] **Step4**: Output set of tags T

Fact: In sample, each SNP is a linear combination of tag SNPs Conjecture: In entire population, each SNP is same linear combination of tags as in sample follows:



$$\begin{matrix} -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 \\ -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 \end{matrix} = \begin{matrix} -1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \times \begin{matrix} 1 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 \end{matrix}$$

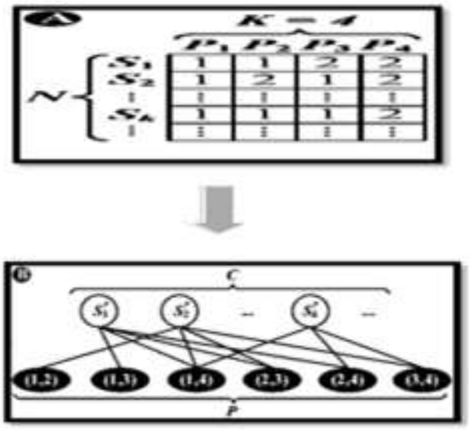
Figure 1: Example of Gauss Jordan Elimination Procedure

GreedyAlgorithm

A greedy algorithm is any algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding the global optimum. Recent studies have shown that a small subset of SNPs (called tag SNPs) is sufficient to distinguish each pair of haplotype patterns in the block.

In reality, some tag SNPs may be missing, and we may fail to distinguish two distinct haplotypes due to the ambiguity caused by missingdata.

In our project we applied a greedy algorithm which can find a subset of SNPs which can still distinguish all distinct haplotypes even when some tag SNPs are missing. Assume we are given a haplotype block containing N SNPs and K haplotype patterns. This block is denoted by an N × K binary matrix Mh (see Figure 2). Define Mh[i,j] ∊ {1,2} for each i∊ [1, N] and j ∊ [1, K], where 1 and 2 represent the major and minor alleles, respectively. In reality, the haplotype block may also contain missing data. This formulation can be easily extended to handle missing data by treating them as wild card symbols. To simplify, we will assume no missing data in theblock.



	P1	P2	P3	P4
S1	1	1	2	2
S2	1	2	1	2
⋮	⋮	⋮	⋮	⋮
Sk	1	1	1	2
⋮	⋮	⋮	⋮	⋮

Figure 2: Haplotype block with Snp pattern

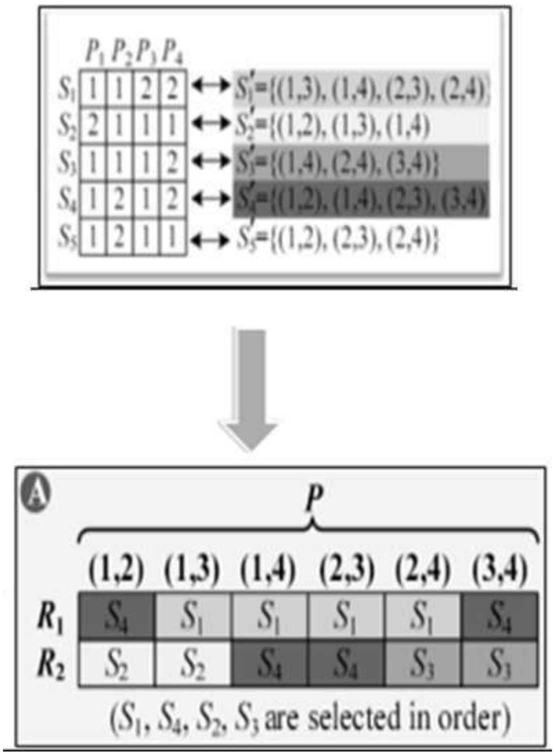
Let C be the set of all SNPs in Mh. The tag SNPs C' □ C are a subset of SNPs which is able to distinguish each pair of haplotype patterns unambiguously. Note that the missing data may occur at any SNP locus and thus create different missingpatterns.

Problem: Minimum Tag SNPs(MTS)

To solve MTS efficiently, we applied a greedy algorithm which returns a solution not too largerthantheoptimalsolution.AssumethattheSNPsselectedbythisalgorithmarestoredina(m+ 1) × |P| table. Here we have considered no missing SNP, so m=0. Initially, each grid in the table is empty.OnceaSNPSk,(thatcandistinguishpatternsPiandPj)isselected,onegridofthecolumn(i, j) is filled in with Sk, and we say that this grid is covered by Sk.

This greedy algorithm works by covering the grids from the first row to the (m + 1)-th row, and greedily selects a SNP which covers most uncovered grids in the i-th row at each iteration. In other words, while working on the i-th row, a SNP is selected if its reformulated set S' maximizes|S' ∩ Ri |, where Ri is the set of uncovered grids at the i-th row.

Figure below illustrates an example for this algorithm to tolerate one missing tag SNP (i.e., m = 1). The SNPs S1, S4, S2, and S3 are selected in order. When all grids in this table are covered, each pair of patterns is distinguished by (m + 1) SNPs in the corresponding column. Thus, the SNPs in this table are the robust tag SNPs which can tolerate up to m missingSNPs.



	P_1	P_2	P_3	P_4		
S_1	1	1	2	2	↔	S_1'={(1,3), (1,4), (2,3), (2,4)}
S_2	2	1	1	1	↔	S_2'={(1,2), (1,3), (1,4)
S_3	1	1	1	2	↔	S_3'={(1,4), (2,4), (3,4)}
S_4	1	2	1	2	↔	S_4'={(1,2), (1,4), (2,3), (3,4)
S_5	1	2	1	1	↔	S_5'={(1,2), (2,3), (2,4)}

	(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)
R_1	S_4	S_1	S_1	S_1	S_1	S_4
R_2	S_2	S_2	S_4	S_4	S_3	S_3

(S_1, S_4, S_2, S_3 are selected in order)

Figure 3: Pictorial example of Greedy Algorithm The pseudo code of this greedy algorithm is given below:

```
GreedyAlgorithm(C,P,m) Step1: Ri ← P, □i∈ [1, m + 1] Step2: C' ←φ
Step3: for i = 1 to m + 1 do Step4: while Ri ≠ φ do
Step5: select and remove a SNP S from C that maximizes |S' ∩ Ri|
Step6: C' ← C' ∪ S Step7: j ← i
Step8: while S' ≠ φ and j ≤ m + 1 do
Step9: Stmp ← S'∩Rj      //Stmp is a temporary variable for holding the result of S' ∩Ri Step10: Rj ← Rj–Stmp
Step11: S' ← S' - Stmp
Step12: j ← j + 1
Step13: endwhile
Step14: endwhile
Step15: endfor
Step16: return C'
```

The time complexity of this algorithm is analyzed as follows. At Line 4, the number of iterations of the intermediate loop is bounded by |Ri| ≤ |P|. Within the loop body (Lines 5–13), Line 5 takes O (|C||P|) because we need to check all SNPs in C and examine the uncovered grids of Ri. The inner loop (Lines 8–13) takes only O (|S'|). Thus, the entire programruns in O (m|C||P|2).

Tag Snp Selection using Binary OptimizationFunctions

Single nucleotide polymorphisms (SNPs) hold much promise as a basis for disease-gene association. However, they are limited by the cost of genotyping the tremendous number of SNPs. It is therefore essential to select only informative subsets (tag SNPs) out of all SNPs. Several promising methods for tag SNP selection have been proposed, such as the haplotype block-based and block-free approaches. The block-free methods are preferred by some researchers because most of the block- based methods rely on strong assumptions, such as prior block-partitioning, bi-allelic SNPs, or a fixed number or locations for tagging SNPs. We employed the feature selection idea of binary optimization function (BOF) to find informative tag SNPs. This method is very efficient, as it does not rely on block partitioning of the genomicregion.

The BOFAlgorithm

Step1: Take one SNP each at atime.
Step2: Using the sliding window method, cut the SNP in suitable size (according to the function used).

Step3: Find the fitness value of each window at a time by the fitness function (objective function). Step4: Add the fitness value of all the windows of a SNP.
Step5: Repeat all the above steps for all the SNPs present in the database Step6: Find the SNPs with same fitness value and make groups.
Step7: Take a random SNP from each fitness group and these are the tag SNPs.
In binary optimization, it is very easy to design some algorithms that are extremely good on some benchmarks (and extremely bad on some others). It means we have to be very careful when we choose a test function set.

Goldberg'sorder-3
The fitness f of a bit-string is the sum of the result of separately applying the following function to consecutive groups of three componentseach:

$$f_1(x) = \begin{cases} 0.9 \text{ if } |y| = 0 \\ 0.6 \text{ if } |y| = 1 \\ 0.3 \text{ if } |y| = 2 \\ 1.0 \text{ if } |y| = 3 \end{cases}$$

Figure 4: Goldberg Function

IfthestringsizeisD,themaximumvalueisobviouslyD/3,forthestring1111...111.In practice, we will then use as fitness the value D/3 ——— f so that the problem is now to find the minimum0.

Bipolarorder-6
The fitness is the sum of the result of applying the following function to consecutive groups of six componentseach:

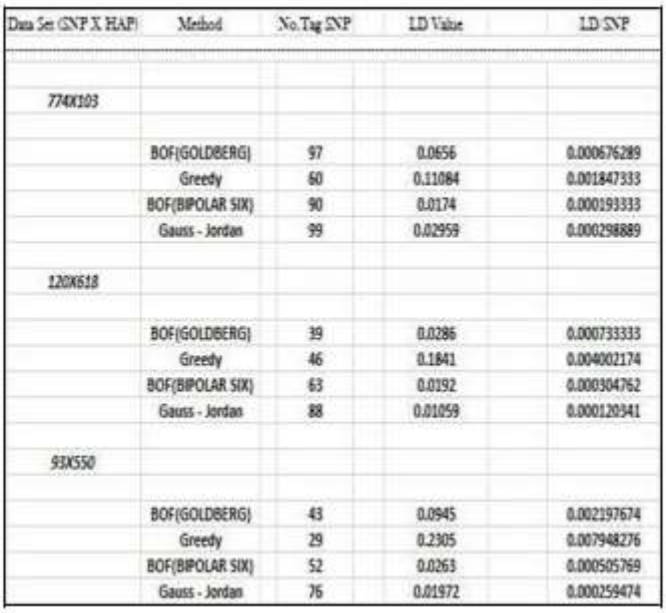
$$f_2(y) = \begin{cases} 1.0 \text{ if } |y| & = & 0 \text{ or } 6 \\ 0.0 \text{ if } |y| & = & 1 \text{ or } 5 \\ 0.4 \text{ if } |y| & = & 2 \text{ or } 4 \\ 0.8 \text{ if } |y| & = & 3 \end{cases}$$

Figure 5: Bipolar Six Function

So the solutions are all combinations of sequences 6x1 and 6x0. In particular, 1111... 111 and 0000...000 are solutions. The maximum value is D/6.

IV. Experimental Results

For our paper, we have selected 3 data sets as (774 X 103), (120 X 618) & (93 X 550), where the first value in each set represents the row (no of SNP) and second value represents the no of column(no of haplotype). The LD Value [7], [8] for each data set is very close to the range ($0 \leq LD \leq 0.3$). We have compared (LD/No. of Tag SNP) ratio for each method in each 3 database ie higher ratio means the tag SNPs are highly correlated with the rest of the SNPs in the data. Also it is notifiable that the number of tag SNPs for the data sets is fairly low which is good because then only that small amount of data can represent the wholedataset.



Data Set (SNP X HAP)	Method	No.Tag SNP	LD Value	LD SNP
774X103				
	BOF(GOLDBERG)	97	0.0656	0.000676289
	Greedy	60	0.11084	0.001847333
	BOF(BIPOLAR SIX)	90	0.0174	0.000193333
	Gauss - Jordan	99	0.02959	0.000298889
120X618				
	BOF(GOLDBERG)	39	0.0286	0.000733333
	Greedy	46	0.1841	0.004002174
	BOF(BIPOLAR SIX)	63	0.0192	0.000304762
	Gauss - Jordan	88	0.01059	0.000120341
93X550				
	BOF(GOLDBERG)	43	0.0945	0.002197674
	Greedy	29	0.2305	0.007948276
	BOF(BIPOLAR SIX)	52	0.0263	0.000505769
	Gauss - Jordan	76	0.01972	0.000259474

Figure 7: No. of Tag Snp values for different methods in each dataset along with their LD values

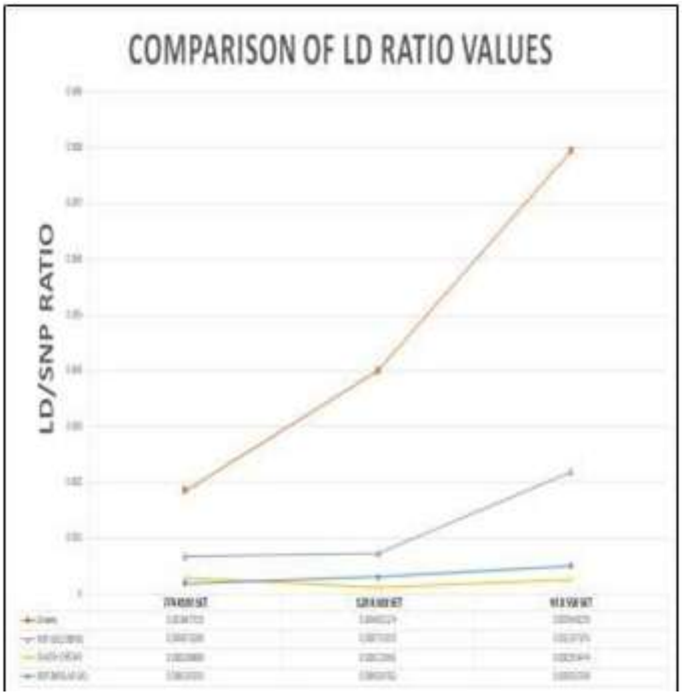


Figure 8: LD/TAG SNP ratio comparison value

From the graph it clear, among the methods we used greedy gives the best result whereas Binary Optimization function (Using Goldberg Function) gives a quite moderate result but Bipolar- Six and Gauss-Jordan Algorithm gives a less convincing result as in those cases no of TAG SNP is quite high as well as co-relation value islow.

V. Conclusion

We defined a new natural measure for evaluating the prediction accuracy of a set of tag SNPs, and use it to develop a new method for tag SNPs selection. This method is based on some novel algorithm that predicts the values of the rest of the SNPs given the tag SNPs. This methods are very efficient. We compared different popular methods of tag SNP selection algorithms on some different genotype datasets from different sources. We also have done a comparative study on the result of different algorithms. In BOF we had to choose from multiple sequences having same score in random. So different sequences could be the result at different instances. So to obtain fixed sequence at every run some mechanism could be introducedhere.

References

[1]. Halldórsson BV, Bafna V, Lippert R, Schwartz R, Vega FM, ClarkAG,Istrail S: Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies. GenomeResearch2004:1633-1640.

[2]. Carlson CS, Eberle MA, Rieder MJ, Yi Q, Kruglyak L, Nickerson DA:Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium.

[3]. Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES: High-resolution haplotype structure in the human genome. Nat Genet2001,29(2):229-232.

[4]. Cormen TH, Leiserson CE, Rivest RL, Stein C: Introduction to algorithms The MIT Press; 2001.

[5]. McCluskey E. Minimization of Boolean Functions. Bell System Technical Journal. 1956; 35:1417–1444.

[6]. Tomaszewski, S. P., Celik, I. U., Antoniou, G. E., "WWW-based Boolean function minimization" INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS AND COMPUTER SCIENCE, VOL 13; PART 4, pages 577-584,2003.

[7]. Zhang K, Qin ZS, Liu JS, Chen T, Waterman MS, Sun F: Haplotype block partition and tag SNP selection using genotype data and their applications to association studies. Genome Research 2004,14:908-916.

[8]. Zhao JH, Lissarrague S, Essioux L, Sham PC: GENECOUNTING: haplotype analysis with missing genotypes. Bioinformatics 2002,18:1694-1695.

[9]. B. Devlin, and N. Risch. A comparison of linkage disequilibrium measures for fine-scale mapping. Genomics.29, 311–322,1995.

[10]. B. Halldórsson, V. Bafna, R. Lippert, R. Schwartz, F. de la Vega, A. Clark, and S. Istrail. Optimal haplotype block-free selection of tagging snps for genome-wide association studies. Genome research.14, 1633-1640,2004.

[11]. Z. Meng, D. Zaykin, C. Xu, M. Wagner, and M. Ehm. Selection of genetic markers for association analyses, using linkage disequilibrium and haplotypes. Am. J. Hum. Genet. 73: 115–130,2003.

[12]. He, J. and Zelikovsky, A. (2004) 'Linear Reduction Methods for Tag SNP Selection', Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology (EMBC'04), pp.2840–2843.

[13]. Zhang, K., Qin, Z., Liu, J., Chen, T., Waterman, M., and Sun, F. (2004) 'Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies', Genome Research, Vol. 14, pp.908–916.

[14]. Jingwu He, Kelly Westbrooks and Alexander Zelikovsky 'Linear Reduction Method for Predictive and Informative Tag SNP Selection' Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology(EMBC'09)

[15]. Tu Minh Phuong , Zhen Lin Russ B. Altman 'Choosing SNPs Using Feature Selection' 2005 IEEE Computational Systems Bioinformatics Conference(CSB'05)

[16]. H. J. Greenberg, W. E. Hart, and G. Lancia, "Opportunities for combinatorial optimization in computational biology.," IN- FORMS Journal on Computing, Vol. 16, No. 3, pp. 211–231, 2004.

[17]. Shambhavi.B.R and Dr.RamakanthKumar.P, "Current State of the Art Pos Tagging for Indian Languages – A Study", International Journal of Computer Engineering & Technology (IJCET), Volume 1, Issue 1, 2010, pp. 250 - 260, ISSN Print: 0976 – 6367, ISSN Online: 0976 –6375.

[18]. MoitreeBasu and Pradipta Deb, "Tag SNP Selection using Quine-Mccluskey Optimization Method", International Journal of Advanced Research in Engineering & Technology (IJARET), Volume 4, Issue 5, 2013, pp. 74 - 81, ISSN Print: 0976-6480, ISSN Online: 0976- 6499.