Stochastic Approach to Binary Matrix Partitioning for Phylogenetic Networks

© Victoria Kusherbaeva Nikolay Vyahhi

St.Petersburg State University
{kusher.v,vyahhi}@gmail.com

Abstract

In this research we introduce the problem of the binary matrix partitioning in a biological context. Our idea is to use SNP matrix to construct a set of phylogenetic networks to retrieve underlying biological meanings and dependencies. We emphasize stochastic methods for matrix clustering and briefly describe the search algorithm. It will allow us to perform fast distributed search on huge biological data for calculating person's similarity.

1 Introduction and Motivation

A lot of recent publications in bioinformatics emphasize *SNPs (Single Nucleotide Polymorphism)* which describe one bit of difference in DNA between members of a species [17].

AAGC**C**TA
AAGC**T**TA

Almost all common SNPs have only two variations (alleles) so they can be described as only one binary bit each. More than 6.2 million human SNPs are presented now in public databases (e.g. in [2]) describing different aspects of genome. They can be used to identify a single person or to find similarities between two or more persons [12].

Based on SNPs of a group of people one can construct a *phylogenetic network* [7]. It represents evolution relations among biological entities and is a generalization of the well known phylogenetic tree. There are different variations of phylogenetic networks, e.g. perfect and galled tree [20, 5]. Based on type of a network there are different constraints on underlying bit sequences [15]. Besides biology phylogenetics networks can be used in other fields, for example, in natural language analysis [16].

Recently introduced the personal web service by [1] allows to find more than 500.000 SNPs in an individual genome (obtained with special device) and analyzes them independently. Also this service supports Family Inheritance tool to analyze meaning of SNPs in aspect of a person's family, but does not provide operations on whole database (e.g. to find unknown relatives).

Our idea is *to provide framework* to store and analyze a lot of SNPs for the large number of individuals. So everyone will be able to:

1. share his own genome,
2. find relatives,
3. find similar persons.

As long as number of individuals and number of SNPs can be huge (each more than 10^6), familiar methods do not work in acceptable time.

In this research we focus on *clustering this binary data to ensure fast search*. We use a *combination of stochastic approximate methods of cluster optimization with other combinatorial methods* [10]. Such kinds of algorithms are characterized by easy understanding, not requiring gradient information, dealing with a large number of parameters and insignificance of possible discontinuities in the fitness function [4].

We *construct phylogenetic networks for each cluster* and then use it for understanding underlying biological dependencies. Different types of networks and different constraints on data give us wide tradeoff between complexity and accuracy.

We do not consider legal aspects (such as discrimination and human rights) because it is not a subject of this research. From the practice of existent social networks sharing personal data and searching among them are possible and do not conflict with a law.

2 Problem statement

2.1 Binary matrix

Each person has *a bit sequence* where every bit represents one SNP (almost all common SNPs have only two alleles). Some SNPs (or group of them) can be additionally described (e.g. "hair color", "eye color", "height" etc.) for convenient usage.

	hair				eyes		×
John	0	1	0	1	1	1	0
Mary	1	0	0	1	1	0	1
Bill	0	1	1	0	0	0	1
Ann	0	1	0	1	1	1	1

If we have N individuals and M different SNPs, these data form *matrix* D with N rows and M columns.

Proceedings of the Spring Young Researcher's Colloquium on Database and Information Systems, St. Petersburg, Russia, 2008
This work was partially supported by RFBR (grant 07-07-00268a)

$$D = \overbrace{\left.\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}\right\}N}^{M}$$

Besides SNPs such type of matrix can present a lot of other data, e.g. market baskets, graph connections etc.

2.2 Search

Search should answer to query:

"Find all individuals that are equal to the given one in the given SNP positions with the given error"

It is reasonable to get a list of these individuals ordered by an error. More formal, *query input* consist of:

- individual number $p : 1 \leq p \leq N$,
- set of SNP indices $a = [a_1, ..., a_T]$, $1 \leq a_i \leq M$,
- integer error $k \geq 0$.

The difference between two persons p and r on SNPs a can be defined as Hamming distance between corresponding bit subsequences:

$$diff_a(p, r) = H(D_{p,a}, D_{r,a}) = \sum_{i=1}^{T} (D_{p,a_i} \oplus D_{r,a_i}),$$

For instance, the difference between Mary and Ann on SNPs $(1, 3, 5, 6, 7)$ will be the following:

$$diff_{[1,3,5,6,7]}(2, 4) = H(10101, 00111) = 2.$$

$$D = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Finally, *query answer* is an ordered by $diff$ list:

$$A(p, a, k) = \{r \in [1..N] : diff_a(p, r) \leq k\}.$$

A native implementation of this search will calculate $diff_a(p, r)$ for every r. It works in $O(NT)$ time and is not acceptable for large N and T. Also, if $T << M$ then we do not need to keep all $N \times M$ matrix in memory and can request only small necessary parts of it. To address these issues we introduce matrix partitioning.

2.3 Matrix partitioning

The main objective of this research is **to divide a matrix D into a number of submatrices $(C_1, ..., C_s)$ in such a way to guarantee a fast execution of search query**.

Each submatrix (let's name it "cluster") C_i can be described with two vectors: indices of rows and indices of columns in the original matrix D. For instance:

$$D = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$C_1 : rows = (1, 3), columns = (1, 3, 4),$
$C_2 : rows = (2, 4), columns = (1, 3, 4, 7),$
$C_3 : rows = (1, 2, 3, 4), columns = (2, 5, 6),$
$C_4 : rows = (1, 3), columns = (7).$

How to use this clustering to speed up search a query execution we will discuss in section 4.

2.4 Terminology

We often use synonyms "part", "cluster" or "class" for "submatrix". And the process of dividing matrix into parts can be named as "partitioning", "clustering" or even "classification". A "person" sometimes can be an "individual" or even just a "biological entity".

3 Background

3.1 Clustering and random search

The problem of *clustering* is partitioning of a data space X into some classes. It means that points in the same class share some common trait and this trait forms this class. Such property, for example, can be a closeness of the point to some "center". Formally, the problem is to find an assignment function: $x \in X \rightarrow c \in C$ where C is a set of clusters [4]. This assignment function can be deterministic (certain class for each point) or non-deterministic (set of probabilities of being in number of classes).

Also there are *penalty functions* which increase if x moves far from the center of a corresponding class.

Thus, clustering problem can be described as estimation such parameters that minimize an average *risk functional* or finding the class centers in which a total dispersion is minimal.

The adaptive optimization algorithm is characterized by accumulating the information during the search of an extremum and using it to increase probability of convergence to an optimum [10].

3.2 Phylogenetic networks

The powerful structure to investigate phylogenetic relations (evolutionary relatedness) are phylogenetic networks based on bit sequences. Unfortunately, as shown in [7], there is confusion in different meanings of this term.

Classically *phylogenetic network* is a directed acyclic graph containing one root node, a set of internal nodes and exact n leaves. Every node has associated bit sequence of equal lengths. Every internal node can have either one or two incoming edges. One edge biologically means "single polymorphism" and can change single bit of sequence from 0 to 1 and each position changes only once in whole network. Two incoming edges mean

"single-crossover recombination" and form a new sequence as a composition of two ancestors, such a node calls recombination node. A phylogenetic network without recombination nodes turns out perfect phylogenetic tree.

A complete definition can be found in [6].

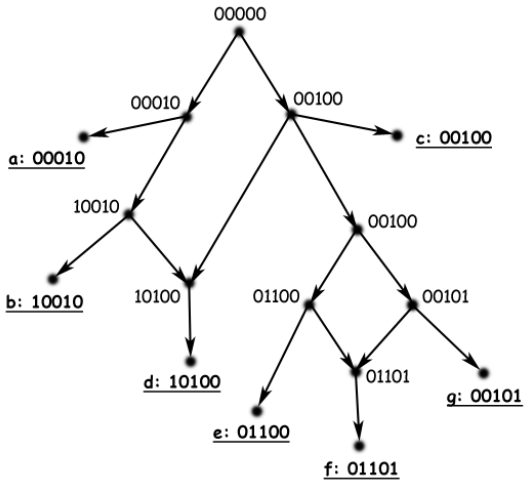

Figure 1: Example of phylogenetic network

Fig. 1 shows phylogenetic network of the following matrix (bit sequences):

a	0	0	0	1	0
b	1	0	0	1	0
c	0	0	1	0	0
d	1	0	1	0	0
e	0	1	1	0	0
f	0	1	1	0	1
g	0	0	1	0	1

where the root node has sequence 00000 and there are two recombination nodes with sequences 10100 and 01101.

Also, there are *different concepts* of phylogenetic networks and this provides a wide variety of data structures and algorithms. In [7] Huson presents the following hierarchy:

- splits networks: median networks, split decomposition neighbor-net, consensus (super) networks;
- phylogenetic trees;
- reticulate networks: hybridization networks (special case: "galled trees"), recombination networks, ancestor recombination graphs;
- other types of phylogenetic networks: augmented trees, any graph representing evolutionary data.

Network construction algorithms are also widely described in literature and vary in supported types of the network, complexity, proved bounds (e.g. of number of recombinations) etc. [14].

4 Feasible solution

The initial problem is proposed to be divided into several subproblems.

4.1 Partitioning

For this purpose we plan to use clustering by an adaptive algorithm of random search [10]. As a result of it we should receive a set of classes for local optimizing and searching inside. Inside of each cluster we will construct a phylogenetic network and maintain it for using in search operations.

In this case we have following *key choices*:

1. *type of risk function* for clusterization,
2. *type of phylogenetic networks*.

A risk function will also strongly depend on constraints of sequences to fit phylogenetic networks. Also still network constructing complexity can be more than $O(nm)$ and we need high parallelism for fast search, we cannot allow huge clusters.

4.2 Search algorithms

It should be divided into two parts:

1. *search of appropriate set of clusters*,
2. *local search* inside of this set.

It has become complicated because partitioning makes parts (clusters) not only with different individuals (rows), but also with different SNPs too (columns). So the whole search query for one person can occur in more than one cluster and *highly distributed search* should be used.

As interesting achievement of this approach we can notice a secure query execution because of data distribution between a lot of clusters. So this system can form something like anonymous peer-to-peer network where nobody knows significant information about others, but still can execute search queries.

5 Related work

Binary matrix clustering is a widely studied field, but most papers emphasize grouping only $1s$ by some rule, considering binary matrix as a plain where 1 means the point and 0 means the gap [8]. There is another approach to approximate the initial matrix with smaller ones, for example, Tao Li in [11] describes next following model:

$$D = AXB^T + E,$$

where X is a centroid, A and B are somewhat smaller than D matrices and E is an error matrix.

Our claim is to use all initial data and partition whole matrix into clusters. We will try to absorb existed ideas and modify them for proposed problem.

Data clustering in general case is reviewed by Anil K. Jain in [8]. Interesting special case with stochastic local clustering are described in [18].

In [10] Kusherbaeva and Sushkov describe a modifacation of an optimization method which we will use for partitioning. Nowadays it is usual to support and optimize systems with uncertainties of measurements, especially with self-training algorithms. These problems are solved by stochastic methods quite good, e.g. in [19, 9, 13].

Phylogenetic networks based on bit sequences have rapidly growing research, e.g. in [15, 7, 14, 20, 6, 16].

In 2007 year there were a lot of publications in this field [3]. We will use proposed algorithms for constructing networks from each part of the matrix.

6 Conclusions

Nowadays there are a lot of individual genome information available for each person, and this information processing will make a significant role in future.

We stated the problem of the binary matrix partitioning in a key of biological search and have a plan to investigate the usage of adaptive algorithms of random search for clustering binary data. Using the set of phylogenetic networks we want to support a fast search operation for personal similarity evaluation. There are a lot of different types of networks, so we need to choose appropriate one and analyze its search abilities and possibilities to divide the initial matrix in such a way to satisfy constraints on sequences.

7 Further work

In this research we focus more on clustering, while distributed search algorithm will be described shortly and only in significant parts. More details and optimization are remained for future. Also, we can try to support more complex queries than presented in this paper.

Measurement errors that are frequent in biology were not mentioned either. Security and anonymization of this system can also be investigated in future.

References

[1] 23andMe. https://www.23andme.com.

[2] dbSNP. http://ncbi.nlm.nih.gov/projects/SNP.

[3] The Who's Who of Phylogenetic Networks. http://lirmm.fr/ gambette/PhylogeneticNetworks.

[4] V. N. Fomin. *Recurring Estimation and Adaptive Filtering*. Nauka, 1984.

[5] Dan Gusfield, Satish Eddhu, and Charles H. Langley. The Fine Structure of Galls in Phylogenetic Networks. *INFORMS Journal on Computing*, 16(4):459–469, 2004.

[6] Dan Gusfield, Dean Hickerson, and Satish Eddhu. An Efficiently Computed Lower Bound on the Number of Recombinations in Phylogenetic Networks: Theory and Empirical Study. *DAM*, 155(6-7):806–830, 2007.

[7] Daniel H. Huson. ISMB Tutorial: Introduction to Phylogenetic Networks., July 2007.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: a Review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 1983.

[10] V. T. Kusherbaeva and Yu. A. Sushkov. Statistical Investigation of Random Search, 2007.

[11] Tao Li. A General Model for Clustering Binary Data. In *KDD*, pages 188–197, 2005.

[12] A. V. Lobashev, M. Yu. Skoblov, M. Stepanova, I. Moiseev, L. Dickerson, A. V. Baranova, A. A. Mironov, and N. K. Yankovsky. Single Nucleotide Polymorphism Density Distribution in the Human Genome as Seen in Public Databases: Possibility of Gene Set Comparisons but not Comparisons of Genomic Regions. *Biophysics*, 48:S81–S84, December 2003.

[13] A. S. Lopatin. Simulated Annealing Algorithm. *Stochastic Optimization in Information Science*, 1, 2005.

[14] Vladimir Makarenkov, Dmytro Kevorkov, and Pierre Legendre. Phylogenetic Network Construction Approaches. *Applied Mycology and Biotechnology*, 6, 2006.

[15] Bernard M. E. Moret, Luay Nakhleh, Tandy Warnow, C. Randal Linder, Anna Tholse, Anneke Padolina, Jerry Sun, and Ruth E. Timme. Phylogenetic Networks: Modeling, Reconstructibility, and Accuracy. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 1(1):13–23, 2004.

[16] L. Nakhleh, D. Ringe, and T. Warnow. Perfect Phylogenetic Networks: A New Methodology for Reconstructing the Evolutionary History of Natural Languages. Under review for LANGUAGE, Journal of the Linguistic Society of America, 2005.

[17] Kwok Pui-Yan. *Single Nucleotide Polymorphisms / Methods and Protocols*. Methods in Molecular Biology. Humana Press, 2003.

[18] Satu Elisa Schaeffer. Stochastic Local Clustering for Massive Graphs. In *Proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-05)*, volume 3518 of *Lecture Notes in Computer Science*, pages 354–360. Springer-Verlag GmbH, 2005.

[19] James C. Spall. *Introduction to Stochastic Search and Optimization*. Estimation, Simulation, and Control, 2003.

[20] Lusheng Wang, Kaizhong Zhang, and Louxin Zhang. Perfect Phylogenetic Networks with Recombination. In *SAC01*, pages 46–50, 2001.