Superlink: A New Program for Exact Genetic Linkage Analysis of General Pedigrees

Fishelson, M. and Geiger, D.
Department of Computer Science, Technion

Genetic linkage analysis is a useful tool for mapping disease genes. It allows one to use statistical tools to associate functionality of genes to their location on the chromosome. Generally speaking, this analysis uses a probabilistic model of inheritance of genetic materials and applies it to data in the form of pedigrees, where some of the individuals are annotated with information on the trait of interest and information on their genetic makeup. As highly-informative genetic marker maps have been developed, multipoint linkage analysis has become a crucial part in linkage analysis studies due to its supremacy on pairwise linkage analysis for locating genes and detecting linkage. However, the computational complexity required to perform such calculations increases exponentially due to the large number of markers that participate in the analysis, the high polymorphism of the markers under study, the size of the pedigree, and the number of untyped people in the pedigree. These factors highly constrain the space and time requirements of existing programs. Some programs fail to run as the number of markers, the degree of polymorphism of the markers, or the size of the pedigree increase. Other programs can handle a large number of markers but can only analyze small pedigrees. We have addressed the increasing need for a program that performs multipoint likelihood calculations on general pedigrees with a higher number of polymorphic markers. We implemented our algorithms in a computer program, called Superlink, that computes pedigree likelihood for complex diseases in the presence of multiple polymorphic markers in fully general pedigrees, taking into account a variety of disease models. Superlink compares favorably with current linkage software with regards to the following criteria: functionality, speed, memory requirements and extensibility. This can be seen from the experimental results described below.

Currently, there are two main approaches to computing pedigree likelihood exactly: Elston-Stewart [3] and Lander-Green [5,6,7]. Both of these algorithms are variants of variable elimination methods [2,16] that depend on different strategies to finding an elimination order. The complexity of the Elston-Stewart algorithm is linear in the pedigree size (for pedigrees with a simple structure) but exponential in the number of markers. On the other hand, the Lander-Green method is linear in the number of markers but exponential in the number of individuals. In Superlink, we used the framework of Bayesian networks as the internal representation of linkage analysis problems [4]. Using this representation allows us to give a unified treatment to both approaches and to handle a wide variety of linkage analysis problems. Whenever feasible, we use variable elimination alone to calculate the likelihood of the pedigree. Otherwise, our algorithm combines variable elimination with conditioning (a divide and conquer approach) to achieve the best time-space tradeoff given the memory available for the linkage analysis problem. The crucial point of the algorithm is that conditioning is performed only after some steps of variable elimination, when the memory requirements are about to exceed the limitations. Such conditioning often applies only to parts of the Bayesian network and thus, computations in other, unrelated, parts of the network are not repeated unnecessarily. The elimination order is chosen automatically according to the parameters of the specific linkage problem. For

small pedigrees with a large number of markers, the algorithm chooses a peeling order, based on the Lander-Green approach, that proceeds locus after locus. For large pedigrees with a few markers, the algorithm chooses an Elston-Stewart style elimination order which "peels" one nuclear family at a time. Other linkage problems are handled by finding a good elimination order. Often the program chooses an elimination order that is a combination of these two extreme known choices of ordering.

Another crucial feature of our program is the preprocessing step performed on the Bayesian network that reduces the range of values that are feasible for each variable given the data. This step often has a large impact on the memory and time requirements of the calculations. Superlink allows for analysis of sex-linked traits and also allows for a disease phenotype to be under the control of two loci [11, 14, 15].

We have run several experiments to compare our program to some of the leading linkage programs currently, Fastlink [1, 8, 12, 13], Genehunter [5, 6, 7] and Vitesse v1.0 [10]. We have not been able, so far, to try Vitesse v2.0 [9] but we have indications that our program outperforms it on all inputs. The running environment on which all experiments were performed was a Sun OS version 5.7 (sun4u) with 2624 MB RAM. In one of the experiments, we used 12 datasets with a medium sized topology taken from a coronary heart disease study and increasing complexity in terms of the number of loci. The pedigree size exceeds the size that can be handled by Genehunter and only the first few files can be run by Fastlink and Vitesse before the memory requirements become too large. Superlink can run on all the files except for the last one on which the computation will require over a 100 hours in order to complete. It is also important to note that, for the files that could be run by Fastlink and Vitesse, the running times are shorter for Superlink. For example, datasetEA2, required 0.39 seconds by Vitesse and 79.32 seconds by Fastlink and only 0.14 seconds by our program. DatasetEA5 required 84.66 seconds by Vitesse and only 1.19 seconds by Superlink. This dataset cannot run on Fastlink. In another experiment we used a medium-sized looped topology. Vitesse doesn't handle looped pedigrees and therefore failed to run on these files. Fastlink can only run on the first file and its running time is 3933.7 seconds, whereas Superlink takes only 2.56 seconds to run on this file. More experimental results, the full paper, data sets, and executables, are available at bioinfo.cs.technion.ac.il/superlink

References

[1] Cottingham, R.W., Idury, R.M. and Schäffer, A.A. 1993. *Am. J. of Hum. Genet.*, 53:252-263.

[2] Dechter, R. 1998. In J.M.I. (Ed.) *Learning in Graphical Models (pp.*75-104). Kluwer Academic Press.

[3] Elston, R.C. and Stewart, J. 1971. *Hum. Hered.*, 21:523-542.

[4] Friedman, N., Geiger, D. and Lotner, N. 2000. *Proc. Sixteenth Conf. Of UAI.*

[5] Kruglyak, L., Daly, M.J. and Lander, E.S. 1995. *Am. J. of Hum. Genet.*, 56:519-527.

[6] Kruglyak, L., Daly, M.J., Reeve-Daly, M.P. and Lander, E.S. 1996. *Am. J. of Hum. Genet.*, 58:1346-1363.

[7] Lander, E.S. and Green, P. 1987. *Proc. Natl. Acad. Sci.*, 84: 2363-2367.

[8] Lathrop, G.M. and Ott J. 1990. *Am. J. of Hum. Genet.*, 47(A188).

[9] O'Connell JR. 2001. *Hum. Hered.*, 51(4):226-40.

[10] O'connell, J.R. and Weeks, D.E. 1995. *Nat. Genet.*, 11: 402-408.

[11] Risch, N. 1990. *Am. J. of Hum. Genet.*, 46:222-228.

[12] Schäffer, A.A. 1996. *Hum. Hered.*, 46:226-235.

[13] Schäffer, A. A., Gupta, S.K., Shriram, K. and Cottingham R.W. 1994. *Hum. Hered.*, 44:225-237.

[14] Schork, N.J., Boehnke, M., Terwilliger, J.D. and Ott, J. 1993. *Am. J. of Hum. Genet.*, 53:1127-1136.

[15] Strauch, K., Fimmers, R., Kurz, T., Deichmann, K.A., Wienker, T.F., and Baur M.P. 2000. *Am. J. of Hum. Genet.*, 66: 1945-1957.

[16] Zhang, N.L. and Poole, D. 1994. In *Proc. of the Tenth Canadian Conference on Artificial Intelligence,* 171-178.