

Efficient Computation of Close Lower and Upper Bounds on the Minimum Number of Recombinations in DNA Sequence Evolution

Yun S. Song,¹ Yufeng Wu,¹ Dan Gusfield¹

Keywords: combinatorial optimization, computational genetics, molecular evolution, quantitative or population genetics

1 Introduction.

We are interested in studying the evolution of DNA SNP sequences which have undergone (meiotic) recombination. For a given set M of sequences, computing the *minimum* number $R_{\min}(M)$ of recombinations needed to explain the sequences (with one mutation per site) is a standard question of interest, but has been claimed to be NP-hard[4], and the only algorithms that compute it exactly either work only on very small datasets[3], or on problems with special structure[1]. In our work, we construct efficient, practical methods for computing both upper and lower bounds on the minimum number of needed recombinations, and for constructing evolutionary histories that explain the input sequences. When the computed upper and lower bounds match, which happens in a surprisingly wide range of data, the evolutionary histories found by our algorithm correspond to the most parsimonious histories with exactly $R_{\min}(M)$ recombinations.

There currently exist several methods to compute *lower bounds* on $R_{\min}(M)$. Some of these methods can compute a lower bound only for very small datasets. Among the methods that can compute a lower bound for large datasets, the program *RecMin*[2] using the haplotype lower bound, is far superior (in time and accuracy) to all other known practical lower bound methods. However, *RecMin* requires the setting of two parameters which affect both the time and the quality of the bounds produced, and there is no theory that specifies a good time/accuracy tradeoff for setting the parameters. Increasing the parameters never reduces the bound produced and often increases it, while typically increasing the running time. Since even the default setting of *RecMin* produces very impressive bounds compared to other practical methods, but higher bounds are typically observed when the parameters are increased, and one does not know where to stop, we define **The Optimal RecMin Problem:** Compute the lower bound that *RecMin* would produce (if allowed enough time) when both parameters are set to their maximum possible value. The bound produced is called the *Optimal RecMin Bound*.

On the *upper bound* side, there are no known efficient methods (in theory or practice) that compute for general data, non-trivial upper bounds on the number of needed recombinations, or methods that construct a network explaining the sequences using a small number of recombinations.

2 Main Results.

In our work, we do several things. **First**, we introduce an algorithm that uses Integer Linear Programming (ILP) to compute the Optimal *RecMin* Bound. **Second**, with additional

¹Department of Computer Science, University of California, Davis, CA 95616, USA. E-mail: yssong, wuyu, gusfield@cs.ucdavis.edu

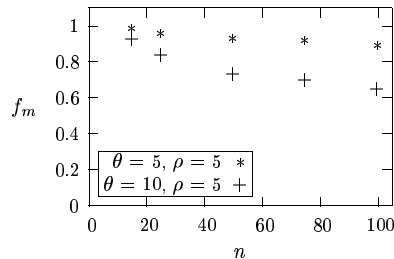


Figure 1: The frequency of the time that our lower and upper bounds match. Here, θ and ρ are scaled mutation and recombination rates, respectively, whereas n denotes the number of sequences.

ideas that dramatically speed up the ILP, we show through extensive experimentation using simulated and real datasets, that this approach computes the Optimal *RecMin* Bound faster than *RecMin* (when *RecMin* can compute it) and that it can efficiently compute the Optimal *RecMin* Bound for problem sizes considered large in current applications (where *RecMin* fails). **Third**, we introduce additional ideas that allow the algorithm to find lower bounds even *better* than the Optimal *RecMin* Bound, and show through extensive experiments that this approach remains practical on problem sizes considered large today. Thus, we provide a practical method that is superior to all other known practical lower bound methods. **Fourth**, on the *Upper Bound* side, we present an efficient algorithm that, when given an input set M of sequences, *constructs* a network that generates M using recombinations and one mutation per site. (For a formal definition of a network—or an Ancestral Recombination Graph, ARG, in the population genetics literature—see [1].) The number of recombinations used in the network produced by the algorithm provides an *upper bound* on $R_{\min}(M)$, but the network itself is of independent interest. **Fifth**, and most importantly, through extensive experimentation with simulated and real data, we show that the computed upper and lower bounds are frequently very close, and are **equal** with high frequency for a surprisingly large range of data. (See Figure 1 for results on simulated data.) Thus, with the use of a very effective lower bound and an efficient algorithm for computing upper bounds, this approach allows the efficient, **exact** computation of $R_{\min}(M)$ with high frequency in a large range of data, much larger than with the use of the algorithm in [3]. This is an important empirical result that is expected to have a very significant impact.

Programs implementing our new algorithms mentioned in this poster are available at <http://wwwcsif.cs.ucdavis.edu/~gusfield/lu.html>. The lower bounds are computed by the program *HapBound*, and the upper bounds and networks are computed by the program *SHRUB*. *SHRUB* also produces code that can be input to an open source program to display nicely the constructed network.

References

- [1] Gusfield, D., Eddhu, S., and Langley, C.H. 2004. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinf. and Comput. Biol.* 2:173–213.
- [2] Myers, S.R. and Griffiths, R.C. 2003. Bounds on the minimum number of recombination events in a sample history. *Genetics* 163:375–394.
- [3] Song, Y.S. and Hein, J. 2003. Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minimum number of recombination events. In: *Proceedings of 2003 Workshop on Algorithms in Bioinformatics*, Berlin: Springer-Verlag. pp. 287–302.
- [4] Wang, L., Zhang, K. and Zhang, L. 2001. Perfect phylogenetic networks with recombination. *J. Comput. Biol.* 8:69–78.