CATTAGAT – Web Server for Primer Specificity Scan

Magnús M. Halldórsson[12] mmh@hi.is
Haukur Thorgeirsson[1] haukurth@hi.is
Ýmir Vigfússon[1]
Hans Thormar[34] hans@hi.is
Jón Jóhannes Jónsson[35] jonjj@hi.is

[1] Department of Computer Science, Faculty of Engineering, University of Iceland, IS-107 Reykjavik, Iceland.
[2] Iceland Genomics Corp., Snorrabraut 60, IS-101 Reykjavik, Iceland.
[3] Department of Biochemistry and Molecular Biology, Faculty of Medicine, University of Iceland, IS-101 Reykjavik, Iceland.
[4] BioCule, IS-101 Reykjavik, Iceland.
[5] Department of Genetics and Molecular Medicine, Landspitali-University Hospital, IS-101 Reykjavik, Iceland.

Keywords: primer search, full alignment, genome scan

1 Introduction

We have implemented a program, CATTAGAT, for quickly finding likely binding sites of PCR primers in the human genome. It allows searching for individual primers as well as primer pairs, with a special batch processing of a large number of primers. A typical query takes about 10 seconds, with results available both visually and in FASTA format.

Primer search Searching for a string of bases in the human genome may be done in a variety of applications. Our service is tailored to the needs of someone about to perform a Polymerase Chain Reaction (PCR) hybridization experiment. We are thus interested in finding the locations in the genome where a given primer, or a primer pair, is likely to hybridize. Based on empirical experiments and the existing literature we have developed three criteria for determining this.

1) The last *suff* bases of the primer must match exactly with the candidate area of the genome. Hybridization at the 3′-end of the primer seems critical for primer extension. Typically, *suff*=4.

2) At most *dist* edits can separate the primer and the matched area. Here edit is defined in the sense of edit distance and corresponds to the insertion, deletion or replacement of a single base. Typically, *dist*is about 4, or 20% of primer length.

3) At most *adj* consecutive differences are allowed. Our empirical evidence suggests that consecutive mismatches are significantly more deleterious to primer hybridization than disjoint mismatches. Typically, *adj*=2.

Note that this last criterion differs from the usual BLAST-implementation which has a larger penalty for opening a gap than for extending it. Another important difference from BLAST is that our method is designed for shorter query strings. Unlike BLAST we cannot assume the existence of a contiguous area of 11 bases identical in the search primer and the match.

Our program accepts *suff*, *dist* and *adj*, within reasonable limits, as parameters from the user. To perform an exact search *dist* is simply set to 0.

The classical form of PCR uses two primers to amplify a sequence of length 100-1000 base pairs. Our service supports doing this *in silico*. The two primers can have different values of *suff*, *dist* and *adj* as described above. In addition the user must specify the maximum number of bases between the two primers.

2 Implementation

Algorithms for computing the edit distance between two strings are well known. Our additional criterion of a maximum number of consecutive mismatches requires an extension to the usual dynamic programming approach, where a a dimension representing the constraint is added to the edit distance matrix. This, however, slows down an already expensive algorithm.

To fit the whole genome in the memory of a standard workstation, we encode each base using 2 bits. For the purpose of our application, we can simply represent unknown bases as *A*s. This allows us to speed up the search using parallel processing, proceeding as follows.

We first make use of the (biochemically motivated) criterion that the last few bases must match exactly, performing an exact search. Next we introduce a fast conservatively approximate matching filter to eliminate those sites that are far from matching the primer. By using 16 bits of the input string at a time as an index into a table storing the editing distance to the primer we can quickly eliminate a large fraction of non-matches. This is similar to an approach suggested by [1]. The next sieve consists of a bit-parallel algorithm due to [3]. This efficient method is based on the observation that the dynamic programming involved in aligning two strings can be computed whole columns at a time, with entries differing at most by a unit between columns.

For most reasonable searches the Myers algorithm will quickly eliminate all but a handful of possible matches. The few remaining candidates can then be resolved by the full dynamic programming algorithm without significantly affecting the time complexity.

In many practical applications the scientist would like to review a batch of primers at a time, for example to weed out primers with very many matches. We have added a variation for this processing, which allows for a more cache-sensitive approach than searching for each primer at a time. This reduces the amortized time for each primer to about one second.

Access A web server running CATTAGAT is available for public access at `http://genome.cs.hi.is`. We hope to maintain a high up-time but can presently make no guarantees. Normal queries should take no longer than a minute to be processed and usually much less. The user query and its results are stored in our database for a short period of time. This allows the user to review the results of old queries, thus saving time. The code (written in C and PHP) is available at the website under the terms of the GNU General Public License ([2]). It would be straightforward to install the service with a different genome or genomes.

Acknowledgments The authors thank the Iceland Research Council for funding the project and Iceland Genomics Corporation for use of their facilities.

References

[1] Chang, W. I. and Marr, T. G. (1994) Approximate string matching and local similarity. In *Proceedings of the 5th Annual Symposium on Combinatorial Pattern Matching*, pp. 259–273. Springer-Verlag.

[2] Free Software Foundation (1991) GNU general public license version 2. URL `http://www.gnu.org`.

[3] Myers, G. (1999) A fast bit-vector algorithm for approximate string matching based on dynamic programming. *Journal of the ACM*, **46**, 395–415.