

APPLICATION NOTE

PASS: a Program to Align Short Sequences

Davide Campagna, Alessandro Albiero, Alessandra Bilardi, Elisa Caniato, Claudio Forcato, Svetlin Manavski, Nicola Vitulo and Giorgio Valle*

CRIBI Biotechnology Centre, University of Padua, via Ugo Bassi 58/B, 35121 Padova, Italy

Associate Editor: Dr. Alex Bateman

ABSTRACT

Summary: Standard DNA alignment programs are inadequate to manage the data produced by new generation DNA sequencers. To answer this problem we developed PASS with the objective of improving execution time and sensitivity when compared with other available programs. PASS performs fast gapped and ungapped alignments of short DNA sequences onto a reference DNA, typically a genomic sequence. It is designed to handle a huge amount of reads such as those generated by Solexa, SOLiD or 454 technologies. The algorithm is based on a data structure that holds in RAM the index of the genomic positions of “seed” words (typically 11-12 bases) as well as an index of the precomputed scores of short words (typically 7-8 bases) aligned against each other. After building the genomic index, the program scans every query sequence performing 3 steps: 1) finds matching seed words in the genome; 2) for every match checks the precomputed alignment of the short flanking regions; 3) if it passes step 2, then it performs an exact dynamic alignment of a narrow region around the match. The performance of the program is very striking both for sensitivity and speed. For instance, gap alignment is achieved hundreds of times faster than BLAST and several times faster than SOAP, especially when gaps are allowed. Furthermore, PASS has a higher sensitivity when compared with the other available programs.

Availability and implementation: Source code and binaries are freely available for download at <http://pass.cribi.unipd.it>, implemented in C++ and supported on Linux and Windows.

Contact: pass@cribi.unipd.it

1 INTRODUCTION

Over the past few years there has been a considerable advance in DNA sequencing technologies and there is a general expectation that the trend of producing more data at a progressively decreasing cost will continue for some time (Church, 2005). Today, the ABI-SOLiD and the Illumina-Solexa are able to generate several GigaBases of mappable data per run, as millions of short reads of about 35 bases. A considerable bottleneck is their alignment (mapping) on a reference genome, allowing some gaps and mismatches.

Exact algorithms based on dynamic programming are far too slow to manage such a huge amount of data. We have recently implemented a version of the Smith-Waterman algorithm running on graphic processors, achieving much faster execution times (Manavski and Valle, 2008), but even so the time required to align hundreds of millions of reads would be too long.

Heuristic solutions, such as those implemented in FASTA (Pearson and Lipman 1988) and BLAST (Altschul et al. 1997) were designed to search single sequences in large databases and are not suitable for millions of reads. More recent approaches, such as SSAHA (Ning et al., 2001) and BLAT (Kent, 2002), make an index of the words occurring in the genomic sequence, but are optimized for longer alignments, not for short reads.

The need for algorithms specifically optimized for massive alignment of short sequences has recently led to the development of new programs such as SX Oligo Search, SlimSearch and ELAND, which are all proprietary software, the latter coming as a part of the Solexa suite. Amongst open source software there are also some noteworthy developments such as PatMaN (Prüfer et al., 2008), RMAP (Smith et al., 2008), MAQ (Li et al., 2008a), SeqMap (Jiang and Wong, 2008) and SOAP (Li et al., 2008b).

Here we propose PASS, a new algorithm to align short DNA sequences allowing gaps and mismatches, with high sensitivity and speed.

2 METHODS

In general, we will refer to the reference sequence as genomic sequence and to the query sequences as reads. As a first step, PASS makes an index of seed words (default 12 bases) occurring in the genomic sequence. This step is fast, requiring $O(L)$ time and $O(L)$ space to be completed, where L is the length of the genomic sequence. After producing the genomic index, PASS scans every read executing three steps: firstly, it identifies seed words on the genomic index; secondly, it checks if it is possible to extend the alignment; thirdly, it refines the alignment and/or score and prints it.

To verify the possibility of extending the seeds, PASS uses a very simple and yet effective approach that is accomplished extremely rapidly, with perfect specificity and sensitivity. It uses a precomputed table of all the possible short words aligned against each other. This

* To whom correspondence should be addressed.

table is loaded in RAM thus allowing an immediate analysis of the two flanking regions adjacent to seed words.

Several precomputed score tables (PST) have been calculated with the Needleman and Wunsch (1970) algorithm, using different values for matches, mismatches and gaps. They are supplied together with PASS, ready to be used. A PST suitable for most purposes is W7M1m0G0X0 (word length 7, match 1, mismatch 0, gap start 0, gap extension 0) that essentially gives a score of one for every aligned base; while W7M3m0G-1X-1 produces different scores depending whether gaps or mismatches are included.

The size of PST requires $O(4^{2w})$, where w is the length of the short words. For $w=6$ there are 4,096 possible words that are aligned against each other producing more than 16 million possible alignments. For $w=7$ there are about 268 million, while for $w=8$ there are more than 4 billion. Since each score is easily contained in a byte, it should not be a problem to load onto the computer memory of any personal computer a 7 bases PST, while 8 bases PSTs require 4Gbytes.

The PST step, more than a proper extension step, should be considered as a filter to discard seed words that will not produce a valid alignment. To complete the discrimination of useless positions, two more filters can be optionally applied: one for discarding low complexity regions and another to discard AT rich regions. The regions that pass the above selections are finally aligned by a dynamic algorithm.

A further optimization came after comparing the sensitivity obtained with gapped patterns rather than uninterrupted seed words. The best results were achieved including two gaps, producing three substrings of similar length. Therefore, the genomic indexes were built using gapped words rather than contiguous bases.

The test set used in this work consisted of one million human Solexa reads from a cDNA library (obtained from the NCBI Short Reads Trace Archive, SRA000299), trimmed to 32 bases by the program ELAND and aligned on the whole human genome (Build 36.3). The computer used for the analysis was a 2 CPUs Dual core AMD with 32 Gb RAM, running at 2.4 Ghz.

3 RESULTS

To evaluate the performance of PASS we compared it with SOAP (Li et al., 2008b), that is probably one of the best programs available in the public domain. The results are shown in Fig. 1.

The importance of the length of the seed words is easily understandable. Seed words of 12-13 bases guarantee the identification of any alignment longer than 26 bases with less than two mismatches (Baeza-Yates and Navarro, 1996), but more degenerated alignments may be missed. The risk of missing alignments increases if longer words are used for seeding. On the other hand, shorter words result in more seeds to extend, hence longer execution times.

The extension process is generally the most critical step of alignment algorithms based on hashing tables, both for execution time and sensitivity. PASS uses pre-computed score tables (PST) of every short word aligned against each other (see Methods). To evaluate the selectiveness of the PST step we considered all of the possible alignments of 7 bases and we found that only in 0.7% of the cases there were less than two discrepancies (gaps or mismatches), while in 11.5% of the cases there were less than three. As a result we calculated that the combined probability that the two sides of a seed word have a total of 3 or less discrepancies is about 0.08%, indicating that more than 99.9% of the random alignments would not pass this selection. The same analysis performed on 8 bases words

showed that 99.997% of the random alignments would be blocked. We think that this level of selectiveness is very effective in reducing the execution time, without affecting at all the sensitivity. As a result the program performs very well both for speed and sensitivity, as shown in Figure 1.

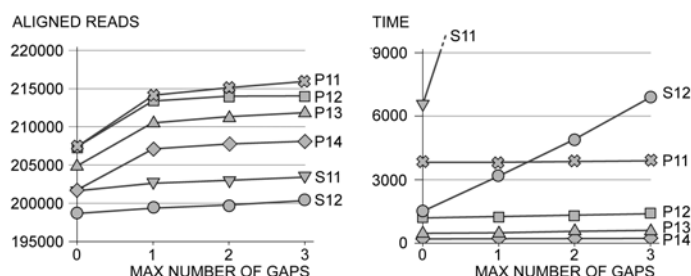


Figure 1. Comparison of PASS with seed words of 11, 12, 13 and 14 bases (P11, P12, P13, P14), and SOAP with words of 11 and 12 bases (S11, S12). The extension step was done using W7M1m0G0X0 (see Methods). In all cases, up to three discrepancies (gaps plus mismatches) were allowed in the final alignment; however, the maximal number of gaps was limited to 0, 1, 2 or 3, as indicated. The left panel shows the number of aligned reads. The panel on the right shows the execution time expressed in seconds.

It can be easily appreciated from the figure that the sensitivity of both programs increases with the shortening of the seed words. However, even with words of 14, PASS has a better sensitivity than SOAP with words of 11, and it runs at least 10 times faster. Our recommendation is to use PASS with words of 12 or 13 bases, thus making the alignment process much faster without significantly compromising the sensitivity.

ACKNOWLEDGMENTS

Funding: Italian Telethon Foundation (grant GSP04289/1C), MiPAF VIGNA Project and FIRB (grant RBLA0345SF).

REFERENCES

- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J. (1990) Basic local alignment search tool. *J Mol Biol.* 215, 403-410.
- Baeza-Yates, R., Navarro, G. (1996). A faster algorithm for approximate string matching. In *Combinatorial Pattern Matching*, Irvine, CA, LNCS1075, Jun 96, 1-23.
- Church, G. (2005). The Personal Genome Project. *Mol. Syst. Biol.* 1, 0030.
- Jiang, H., Wong, W.H. (2008). SeqMap: mapping massive amount of oligonucleotides to the genome. *Bioinformatics.* 24, 2395-2396.
- Kent, W.J. (2002) BLAT-the BLAST-like alignment tool. *Genome Res.* 12, 656-664.
- Li, H., Ruan, J., Durbin, R. (2008a). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* 18, 1851-1858.
- Li, R., Li, Y., Kristiansen, K., Wang, J. (2008b) SOAP: short oligonucleotide alignment program. *Bioinformatics.* 24, 713-714.
- Manavski, S.A., Valle, G. (2008). CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. *BMC Bioinformatics.* 9, Suppl 2:S10.
- Needleman, S.B., Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol.* 48, 443-453.
- Ning, Z., Cox, A.J., Mullikin, J.C. (2001) SSAHA: a fast search method for large DNA databases. *Genome Res.* 11, 1725-1729.

- Pearson, W.R., Lipman, D.J. (1988) Improved tools for biological sequence comparison. *PNAS* 85, 2444-2448.
- Prüfer, K., Stenzel, U., Dannemann, M., Green, R.E., Lachmann, M. and Kelso, J. (2008). PatMaN: rapid alignment of short sequences to large databases. *Bioinformatics* 24, 1530-1532.
- Smith, A.D., Xuan, Z. and Zhang, M.Q. (2008). Using quality scores and longer reads improves accuracy of Solexa read mapping. *BMC Bioinformatics*, 9:128.
- Lin, H., Zhang, Z., Zhang, M.Q., Ma, B., Li, M. (2008). ZOOM: Zillion of Oligos Mapped. *Bioinformatics*, 24, 2431-2437.