BIOINFORMATICS APPLICATIONS NOTE

Vol. 28 no. 1 2012, pages 125–126
doi:10.1093/bioinformatics/btr595

Sequence analysis

Advance Access publication October 28, 2011

RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data

Yongan Zhao[1], Haixu Tang[1,2] and Yuzhen Ye[1,*]

[1]School of Informatics and Computing and [2]Center for Genomics and Bioinformatics, Indiana University, Bloomington, IN 47404, USA

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: With the wide application of next-generation sequencing (NGS) techniques, fast tools for protein similarity search that scale well to large query datasets and large databases are highly desirable. In a previous work, we developed RAPSearch, an algorithm that achieved a ~20–90-fold speedup relative to BLAST while still achieving similar levels of sensitivity for short protein fragments derived from NGS data. RAPSearch, however, requires a substantial memory footprint to identify alignment seeds, due to its use of a suffix array data structure. Here we present RAPSearch2, a new memory-efficient implementation of the RAPSearch algorithm that uses a collision-free hash table to index a similarity search database. The utilization of an optimized data structure further speeds up the similarity search—another 2–3 times. We also implemented multi-threading in RAPSearch2, and the multi-thread modes achieve significant acceleration (e.g. 3.5X for 4-thread mode). RAPSearch2 requires up to 2G memory when running in single thread mode, or up to 3.5G memory when running in 4-thread mode.

Availability and implementation: Implemented in C++, the source code is freely available for download at the RAPSearch2 website: http://omics.informatics.indiana.edu/mg/RAPSearch2/.

Contact: yye@indiana.edu

Supplementary information: Available at the RAPSearch2 website.

Received on August 10, 2011; revised on September 29, 2011; accepted on October 23, 2011

1 INTRODUCTION

The applications of next-generation sequencing (NGS) in transcriptome sequencing (Marioni *et al.*, 2008) and metagenomics projects (Riesenfeld *et al.*, 2004) have resulted in enormous amounts of sequence. A key step to analyzing these sequences is to identify the protein-coding genes and their putative functions by similarity searches, which, for example, is useful for studying the functional content (Dinsdale *et al.*, 2008), or the taxonomic composition (Brady and Salzberg, 2009; Huson *et al.*, 2007), of a microbial community. As a popular protein similarity search tool, BLAST (Altschul *et al.*, 1990), however, has become a bottleneck for the computational analysis of massive NGS datasets. On the other hand, fast algorithms such as BLAT (Kent, 2002) can work up to 100 times faster than BLAST at identifying very similar protein sequences, but will still miss a substantial fraction ($>20\%$) of weaker similarities (Ye *et al.*, 2011).

*To whom correspondence should be addressed.

To address this challenge, we developed a new protein database search tool RAPSearch, which follows the seed-extension approach as used in BLAST, but is based on flexible-length seeds on a reduced amino acid alphabet of 10 symbols, each representing a group of amino acids (Ye *et al.*, 2011). When tested on several NGS datasets, RAPSearch achieved up to a 90X acceleration when compared with BLAST, while missing <5% of potential protein hits. However, because RAPSearch uses a suffix array to index the database sequences for seed identification, it uses substantial memory for searches in large databases.

Here, we present RAPSearch2, a new implementation of the RAPSearch algorithm, which uses a collision-free hash table to index a protein database. RAPSearch2 significantly reduces the memory requirement (e.g. from ~8G to ~2G when searching against the NCBI NR database) while further accelerating the similarity search process another 2–3 times. In RAPSearch2, we also implemented a multi-threading technique that allows users to accelerate the similarity search even further on multi-core CPUs.

2 METHODS AND IMPLEMENTATION

RAPSearch2 uses a collision-free hash table to index the protein sequences in a given search database. Each key of the hash table represents a 6-mer on the reduced amino acid alphabet, and all positions of 6-mers in the database are sorted according to the hash values of the 6-mers on the regular amino acid alphabet. (Note that the reduced alphabet representation is only used for seed storage and retrieval, while the seed extension and significance evaluation are based on the original sequences.) RAPSearch2 uses 4 bytes (32 bits) to encode each 6-mer in the database: the first 20 bits are used to represent the hash keys on the reduced alphabet for all 6-mers (since $2^{20} \approx 10^6$) in the database; and the lower 12 bits are used to represent the offset of each instance of the 6-mers in the regular amino acid alphabet. Such a representation allows us to use bit shift operations to retrieve the position of each 6-mer (in the full 20 amino acid alphabet) in constant time. For each entry of the same key, the hash values are sorted according to the four amino acids following the 6-mer, allowing seeds of up to 10 residues.

In the search step, RAPSearch2 first scans a query sequence and finds the entry of each 6-mer in the hash table, then uses a binary search to find all occurrences of the seed (beyond the 6-mers) from the range of actual instances in the same entry.

We also implemented a multi-threading technique in RAPSearch2. Since both the searches for individual queries and the seed-extensions between a query and individual

Table 1. The comparison of the running time of BLAST, RAPSearch and RAPSearch2

Database	Query			Running time (min)				
	Dataset	Number of reads	Read length (nt)	BLAST[a]	RAPSearch	RAPSearch2 (# threads)		
						1	4	8
IMG (1.6G)	SRR020796[b]	1 164 805	72	95 800	1170	587	170	100
	4440037[c]	188 445	100	9240	378	120	36	22
	TS28[d]	622 554	200	67 000	3872	1341	331	242
	TS50[d]	312 665	329	39 200	4105	1512	385	281
NCBI NR (3.2G)	SRR020796			271 000	2910	1229	362	250
	4440037			25 680	889	335	110	58
	TS28			177 900	8471	3019	859	518
	TS50			103 600	9195	3545	901	644

[a]The running time was estimated using 1% of the original query dataset; the actual BLAST search of the original datasets was carried out on a computer cluster. Note that we compared RAPSearch with BLAST (blast2.2.18) and BLAST+ (blast+-2.2.23). The comparison showed that BLAST and BLAST+ have almost identical sensitivity (but BLAST+ is twice as slow), so we only show the comparison with BLAST in this article (and the speedup will be even greater if we compare RAPSearch to BLAST+).
[b]The SRR020796 dataset was downloaded from the NCBI website (from the rumen microbiota response study), and only 2% of the reads were used for testing.
[c]The dataset was from the nine biomes project (Dinsdale *et al.*, 2008).
[d]TS50 (accession number: 4440615.3) and TS28 (4440613.3) datasets were from the Twin Study (Turnbaugh *et al.*, 2009). 4440037, TS50 and TS28 datasets were downloaded from the MG-RAST server (http://metagenomics.anl.gov/).

subjects are independent, we implemented an inter-query scheme to process queries in parallel to reduce overhead on the thread switch. RAPSearch2 was implemented in C++ using Boost library 1.42 (http://www.boost.org/) and threadpool 0.2.5 (http://threadpool.sourceforge.net/), and was tested extensively on Linux platforms.

3 RESULTS

We used the same NGS datasets as used in Ye *et al.* (2011) to test RAPSearch2 (see the example datasets in the Supplementary Material). The Integrated Microbial Genome (IMG) v3.0 and the NCBI NR (as of June 2009) (98% non-redundant set) databases were used as the search databases. Table 1 compares the performance of RAPSearch2 to RAPSearch and BLAST on four query datasets on a computer with four Xeon 2.93 GHz X5570 CPUs with 48G RAM [see more detailed comparison between RAPSearch and BLAST in (Ye *et al.*, 2011)]. Since RAPSearch2 uses the same seed-extension algorithm as RAPSearch, it gives identical results as RAPSearch, except that it runs 2–3 times faster due to its optimized data structure. Therefore, RAPSearch2 retained the high sensitivity of RAPSearch relative to BLAT [see the comparison between RAPSearch and BLAT in (Ye *et al.*, 2011) and in Supplementary Materials]. RAPSearch2 also requires less memory than RAPSearch. Running in single thread mode requires up to 2G memory (one-fourth the ~8G memory required by the original RAPSearch), whereas running in 4-thread mode requires up to 3.5G memory, which is typically available on regular computer clusters. Furthermore, the 4-thread mode achieved about a 3.5X acceleration compared with single-thread mode, while the 8-thread mode achieved an almost 6X acceleration, indicating that our multi-threading implementation is efficient. Note that BLAT runs only slightly faster than RAPSearch, but ~2–3 times slower than RAPSearch2.

RAPSearch2 uses the same probabilistic model to evaluate the significance of protein sequence alignments used in BLAST (Altschul *et al.*, 1997), and reports the *E*-values similarly. The output of RAPSearch2 has the same format as BLAST, and can be directly adopted into any analytical pipeline to replace BLAST as the similarity search engine. Therefore, RAPSearch2 is readily used for routine protein similarity searches for NGS data.

Funding: National Institutes of Health (1R01HG004908).

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
Brady,A. and Salzberg,S.L. (2009) Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nat. Methods*, **6**, 673–676.
Dinsdale,E.A. *et al.* (2008) Functional metagenomic profiling of nine biomes. *Nature*, **452**, 629–632.
Huson,D.H. *et al.* (2007) MEGAN analysis of metagenomic data. *Genome Res.*, **17**, 377–386.
Kent,W.J. (2002) BLAT–the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.
Marioni,J.C. *et al.* (2008) RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res.*, **18**, 1509–1517.
Riesenfeld,C.S. *et al.* (2004) Metagenomics: genomic analysis of microbial communities. *Annu. Rev. Genet.*, **38**, 525–552.
Turnbaugh,P.J. *et al.* (2009) A core gut microbiome in obese and lean twins. *Nature*, **457**, 480–484.
Ye,Y. *et al.* (2011) RAPSearch: a fast protein similarity search tool for short reads. *BMC Bioinformatics*, **12**, 159.