A New Module in RAPSearch2 for Fast Protein Similarity Search of Paired-end Sequences

Xiaoqian Zhang

Advisor: Haixu Tang

SoIC, Indiana University, Bloomington

July 13th, 2013

Abstract:

Protein similarity search is a fundamental step for taxonomic classification and function annotation of sequencing data from metagenomic and metatranscriptomic projects. Currently, the most popular tool for similarity search is BLAST (or specifically, the blastx), which have proved very efficient in aligning conventional sequencing data such as Sanger reads. The application and extension of Next Generation Sequencing (NGS) technology, which generates massive sequencing data, poses new challenge for classical algorithms of sequence comparison and similarity search. If we use BLAST to precede NGS sequences, the speed will be too slow. To address this challenge, RAPSearch [12,13] has been developed. It is a fast protein similarity search tool, which utilizes reduced amino acid alphabet to speed up the similarity search a few magnitudes and meet the demand of NGS sequence analysis.

Paired end sequencing is a common technique used in NGS. It produces two reads from proximal locations of a target DNA or RNA molecule in both forward and reverse direction, which could be potentially utilized to enhance the alignment precise and coverage. RAPsearch has two versions (RAPsearch and RAPsearch2), both can only treat single-end sequences. Here, I will present a method applying to RAPSearch2 that combine paired-end reads as one hit and evaluate the significance in the similarity search to improve sensitivity of alignment.

Based on the RAPSearch2 algorithm, I built a new module that could process the paired-end reads simultaneously. By using the paired end sequences aligned on the proximal locations on the same subject sequences, the method could increase the searching sensitivity by about 0.5%~0.6%, comparing to the similarity search by using each of the paired end sequences individually.

Background

1, Sequence comparisons and the challenge of Next Generation sequencing

Protein sequence comparison and similarity search is a fundamental computational problem that has been extensively studied in bioinformatics. It is the first and an effective step to annotate newly acquired protein or DNA sequences. Great efforts have been invested in improving the searching algorithms in bioinformatics since the beginning of this field. Some algorithms had been used broadly during a long period of time. Smith-Waterman [1] is one of the classical methods that could compute the optimal local alignment between two protein sequences. It uses dynamic programming and only considers score above zero (or an equivalent threshold) to ensure the results are locally optimized. However, this method becomes too time-consuming when applied to the comparison of many pairs of protein sequences, e.g., in the case of similarity search of proteins against a large protein database.

Now, the most broadly applied method for homology database search is the Basic Local Alignment Search Tool (BLAST) [2][3]. Comparing to the Smith-Waterman algorithm, this method could save much time with a slightly downside of optimal results. BLAST is a heuristic method that find exact maximum match as seeds, and then extends them on both directions until the alignment score reaches a threshold or drop them. Extended segments that have been kept were referred as High-Scoring Segment Pairs (HSP). Several HSPs could be connected into one alignment, if applicable. A rigorous statistical model has been developed to compute the significance of each alignment [4][5]. This method is reported to be up to 40 times faster than Smith-Waterman algorithm, but with only a tiny loss of accuracy [4]. Nowadays, many software tools have been developed on the basic algorithm of BLAST for specific comparison purposes including protein similarity search tool (BLASTp and BLASTx) [6].

The next-generation sequencing (NGS) techniques are high or ultra-high throughput technologies developed in recent decades. It produces millions to billions of sequencing reads in a single experiment that can be completed in one day to a few days. In many sequencing projects,

particularly those for metagenomic and metatranscriptomic studies, function annotation of these reads start from the similarity search against a large protein databases that typically comprise of millions of protein sequences. The similarity search using BLAST may take up to thousands of hours even when a computer cluster with hundreds of CPUs were used, which becomes a severe analysis bottleneck with the explosion of NGS data. Therefore, there is a demand for more efficient ways for protein similarity search of NGS data.

So far, many algorithms and tools have been developed to improve BLAST, such as BLAT[7], Mummer[8], PatternHunter [9][10] and BLASTZ[11], most of which addresses the challenges of genome comparison. In case of protein similarity search, RAPsearch[12] (Reduced Alphabet based Protein similarity Search), which uses a reduced (compressed) amino acid alphabet [12][15] as basic elements of comparison while BLAST uses individual amino acids, is one of outstanding methods to speed up the protein similarity search. It is based on the assumption that a group of amino acids with similar chemical attributes can be considered as similar in the sequence comparison. By using reduced amino acids alphabet, many mutations occurred between two chemically similar amino acids could be tolerated. Significant matched segments on the reduced alphabet could be extended to longer alignment on the amino acid level. As a result, we can use longer threshold for seed length and ignore less significant seed at the same time [12]. As an ultra-fast protein similarity search tool for NGS data, RAPsearch can achieve a speed acceleration between 20~90 times over BLAST with similar levels of sensitivity in short reads [12]. To improve the performance of RAPSearch, RAPsearch2 [13] has been released. In RAPSearch2, the target protein sequence database will be indexed using hash table, and a multi-threading parameter has been added in the program. These implementation has achieved an additional 2~3 times over RAPSearch [13], while reducing the memory usage.

2, Paired-end sequences

Paired-end sequencing represents a specific setting of Next Generation Sequencing technology, which is also known as “double-barrel shotgun sequencing”. This technology takes two tags on both ends of one DNA fragment and extends in opposite direction to sequence the DNA. As a result, the paired-end reads obtained from the opposite DNA strands with a small (typically 300-500 bps) distance are output in two companion files. Each record in one file has a corresponding opposite direction record at the same line of the other file. These two reads

sequenced one DNA fragment from each end following opposite directions into the center. Each one of the records is about 100bp. It is shown that more information could be contained in paired-end sequencing result than single end one [14]. To get more information buried in paired-end sequences in similarity search, one assumption has been raised that if we take two paired sequences files as queries at the same time and combine the paired results in a proper way, the result of the alignment would be improved. So this project is to apply the RAPsearch2 algorithm to processing paired-end reads, and to assess how better sensitivity we can achieve.

Method

The project was done on Linux platform, the source code was written in C++, and the test and data processing scripts were written in python. The alignment algorithm is based on the RAPSearch 2.10 and named as RAPsearch2.12. You can download the source code of RAPsearch2.12 from http://omics.informatics.indiana.edu/mg/get.php?software=rapsearch2.12_pair_64bits.tar.gz. All the source code about RAPsearch 2.10 and other versions could be found and downloaded on the website: http://omics.informatics.indiana.edu/mg/RAPSearch2/ .

1, Query Files

By adding a new option "-c" in the main function, the RAPSearch2 program could take paired files as input. For the first query file, we call the single file processing function that separate the file into several temporary blocks by a pre-setting size [13]. The second query file is been separated into the same number of temporary blocks as the first query file and each block contains the same number of sequences as the first one.

2, High-scoring segment pairs (HSPs) from paired-end reads

High-scoring segment pairs **(**HSPs) [2][3] is a fundamental concept in RAPsearch2 adopted from BLAST. It is referred to as an alignment of two equal length segments, one from a query sequence, and the other from a subject sequence in the database, with the maximum local alignment score. Any pair of aligned segments are considered as an HSP if their score is above a

threshold under an alignment scoring scheme, such as the BLOSUM62 scoring matrix for amino acid residues in protein similarity search. Based on its score, one can compute an E-value for each HSP, by using the Altschul-Karlin model [4][5]. When there are more than one HSP derived between the same pair of query and subject sequences, two methods can be used to compute the significance of the long gapped alignment resulted from the merging of these HSPs: 1) the lowest score method that takes the lowest score among the merged HSPs as the score of the long alignment; or 2) the sum-of-score method that takes the sum of scores of all HSPs as the score of the long alignment. The significance of the long gapped alignment can then be computed using the same statistical model [4][5] based on the resulting scores. Similar as BLAST, RAPSearch2 uses the sum-of-score method for computing the significance of long alignment comprising of multiple HSPs. Here, I applied the same method to compute the sum-of-score for two or more HSPs resulted between each read pair against the same subject protein sequence. This means, I treat two paired-end reads as a single query sequence, and merge all HSPs from both of them into a single long gapped alignment, in which an HSP of each read is considered as the HSP of one region in the query sequence.

$$HSP_{(pair)} = HSP_1 + HSP_2 \quad \text{.....(1)}$$

3, E-value Calculation

E-value indicates the significant of an HSP, i.e., the probability of obtaining an HSP with the score or higher when such a query sequence is searched against a database of certain size consisting of proteins that are not similar with the query. Smaller E-value means higher significance of the alignment. In both BLAST and RAPSearch, E-value is be calculated as [4][5]:

$$E = Kmne^{-\lambda S} \quad \text{.....(2)}$$

where S is the alignment score of the HSP, n and m are the length of query sequence and the size of the subject database, respectively, and the parameters λ and K depend on the substitution matrix and the gap penalties [17]. Since in RAPSearch2 we use BLOSUM62, the gapped λ = 0.267, K= 0.041. In the case of paired-end reads, the query length n is the sum of the lengths of two paired-end query reads and S is the sum-of-score of HSPs for both reads calculated by (1).

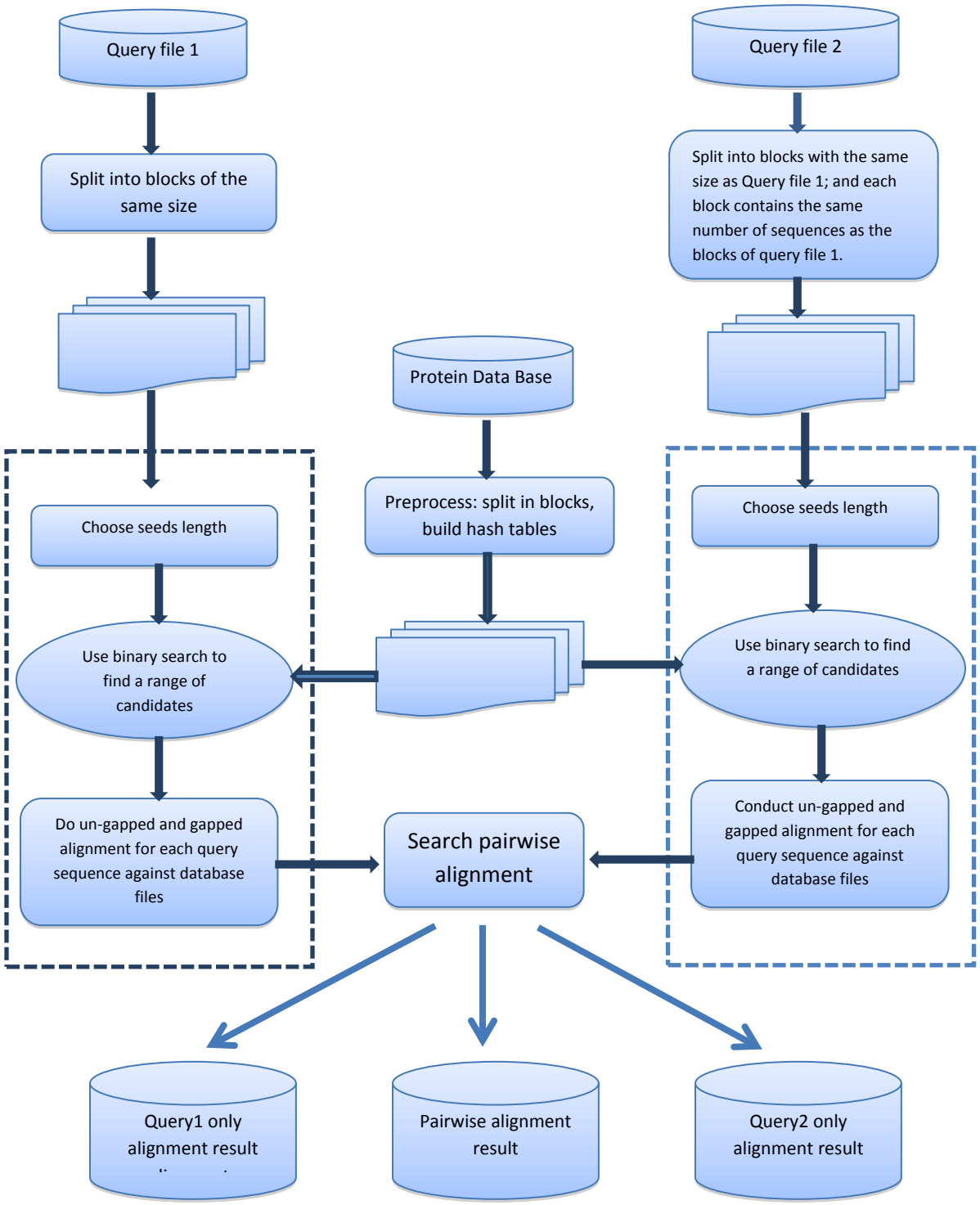
4, Results and Output

In paired-end reads processing, each paired-end query sequence was searched against the database separately at the same time. Then we parse through the alignment results, get HSPs in which both query sequences match the same protein sequence in database. In some cases, there are more than one HSP between the paired-end reads and the subject protein sequence. We screen each possible combination of the pairwise alignments, using the following criteria: 1) two reads are aligned to the subject protein in opposite directions; 2) the alignments may overlap; 3) at least one of the alignments has a score above a threshold. Then we calculate the E-value of their sum-of-score following formula (1) and (2). When the E-value is below threshold, we merge the paired-end alignments into one alignment record and put it into a "paired" result file while deleting the original alignments from the alignment result file for each query file of the pair-end reads. The remaining alignments were kept as the single alignments and were put into separate result files by different query files. In summary, there will be three output files for each paired-end alignment: two files storing alignment results that are not paired, and one file storing the alignment results of the paired-end reads.

5 Workflow:

The general workflow of pair-end files processing is described as below. The protein database was pre-processed and split into several blocks, and a hash-table was built for each block. The first of the two query files (one represents the forward tag, and the other represents the reverse tag of the paired-end reads) may also be split into several blocks according to certain file size and indexed accordingly. While paired-end reads are one-to-one correspondingly, to screen the pairwise alignments of one pair of query sequences a time, we split the second query file into the blocks of the same size as the first query file. All query files are processed in RAPSearch2. Each paired-end query sequence was searched against the same database at the same time. Then a screening process is applied to compute an E-value from the sum-of-score from the two alignments of paired-end query sequences against the same subject protein

sequence in the database and store the significant alignments in the pairwise alignment result. Those significant alignments of only one-end query read will be stored into two files separately



Results and analysis:

The Pairwise module of RAPSearch2 was tested by a query dataset of paired-end reads files StoolA_gDNA-trimmed.1.fa and StoolA_gDNA-trimmed.2.fa taken from a metagenomic study of human gut microbiome [19]. Each file contains 17,427,925 reads acquired by using Illumina sequencers (with 101 bases). This query dataset can be downloaded from following links:
StoolA_gDNA-trimmed.1.fa(2.3G):
http://omics.informatics.indiana.edu/mg/get_data.php?file=StoolA_gDNA-trimmed.1.fa
StoolA_gDNA-trimmed.2.fa(2.3G):
http://omics.informatics.indiana.edu/mg/get_data.php?file=StoolA_gDNA-trimmed.2.fa.
The subject database is eggNOG (automated construction and annotation of orthologous groups of genes) [18] and can be downloaded from:
http://omics.informatics.indiana.edu/mg/get_data.php?file=nogCOGdomN95.faa .
Both query and subject dataset could be downloaded from RAPsearch2 web site: http://omics.informatics.indiana.edu/mg/RAPSearch2/ (Query datasets SRA ID: SRA051242, and subject database name: nogCOGdomN95.seq).

Query data were searched against eggNOG annotated database in paired-end model and single model separately with the E-Value threshold of 1e-3. For comparing the time and efficiency of the RAPsearch2, 1/1000 of the query reads were tested by using BLASTx. As a result, the running time of the complete query data on BLASTx was estimated based on the 1000 times of the actual running time with the small query file.

The command I used to generate alignments:

Pair-RAPsearch2:

rapsearch –c [query1 file] [query2 file] –d [database file] –e [-3] –o [result file]

RAPsearch2:

rapsearch –q [query file] –d [database file] –e [-3] –o [result file]

BLASTx:

blastx –query [query file] –db [database file] –e [-3] –o [result file]

Test was performed on a 16-core 2.93 GHz Intel Xeon CPU and 8 threads are used.

Table 1. Alignment Hits result of pair-RAPsearch2

	singleton alignment result	Paired-alignment result
Hits #(1e-3)	248,249,353	249,901,425

Table 2. Alignment Time of RAPsearch2, pair-RAPsearch2 And BLASTx

	RAPsearch2	Paired-RAPsearch2	BLASTx
Time	15h50m	16h10m	4308h20m

Table 1 shows the alignment hits results of Paired_RAPsearch2 under threshold of 1e-3. One hit means exclusively a pair of one query sequence and one target database read. Singleton alignment results report the sum of the number of hits in two single results file, while Hits of Paired-RAPsearch2 report the number of hits in the paired-end results. Table 2 shows the running time of RAPsearch2, pair-RAPsearch2 and BLASTx for processing the testing query files.

As Table 1 shows, the paired alignment can give us 0.552% more exclusive hits when we set the E-value threshold as 1e-3. These paired alignments also including extra hits from 7925 query read pairs that do not have any hit when these reads were searched individually. The time for running Pair-RAPsearch2 is about the same as RAPsearch2, with about 2% overhead for processing the read pairs. Comparing with BLASTx (Table 2), the running time is 2-3 magnitudes shorter, whereas the result of BLASTx and RAPsearch are similar [12][13].

The reason that more significant hits can be obtained when paired-end reads were searched together than those from the search of individual reads is that, when two paired end reads both aligned with same subject sequences, the e-value of pairwise alignment may be much smaller than any single of them. As a result, for some sequences that have HSPs above an e-value threshold, when their paired-end read has HSPs with the same subject sequence, the e-value of HSPs from both read may well below the threshold. Therefore, they are reported in the paired alignment results as significant alignments.

To examine the effectiveness of Paired-RAPsearch2, I compared the alignment result of Paired-RAPsearch2 with BLASTx in accuracy. A query file containing 2000 paired-end reads

were simulated from 10 E. coli protein coding genes in different COGs groups [20,21] that are conserved across a diverse range of bacterial genomes, by using Meta-Sim with 100bp Illumina sequencing error model (errormodel-100bp.mconf). The 100 Illumina sequencing error model was downloaded from http://www.plantagora.org/tools_downloads/files/errormodel-100bp.php. I collected all orthologous genes in 66 genomes [21] from COG groups [20,21] as subject database. Note that these each genome has at least one orthologous gene for each of these 10 *E. coli* genes in COG database. I searched the simulated *E. coli* query reads against the protein database containing all the ortholog genes by using Paired-RAPsearch2 and BLASTx, respectively, with the E-value threshold of 1e-3 and 1e-5, and compared the results. I consider an alignment hit as *true positive*, if the query and subject reads belong to the same COG group, which means the software has correctly assigned the query sequence to the corresponding protein family. Otherwise, the hit was considered as *false positive*, as the subject read and the matched query read belong to different COG groups. **Table 4** shows the statistics of alignment results. From it, we can see the alignment accuracy of paired-RAPsearch is similar as BLASTx: when E-value threshold is set as 1e-3. 87.8% of significant alignments are true positives for each software tool; when E-value threshold is 1e-5, the true positive rate of paired-RAPsearch is 0.3% higher than BLASTx.

Table 3 10 conservative protein-coding genes in Ecoli used for the accuracy tests.

Gene Name	COG group number
Glutamyl- and glutaminyl-tRNA synthetases (glnS)	COG0008
Prolyl-tRNA synthetase (proS)	COG0442
Translation elongation factor P (efp)	COG0231
Topoisomerase IA (topA 1)	COG0550
EMAP domain (metG 2)	COG0073
Predicted GTPase, probable translation factor (ychF)	COG0012
Alanyl-tRNA synthetase (alaS)	COG0013
Arginyl-tRNA synthetase (argS)	COG0018

Ribosomal protein S12 (rpsL)	COG0048
Pseudouridylate synthase (truA)	COG0101

Table 4 The accuracy comparison between Paired-RAPsearch2 and BLASTx

Log E-value	Alignment method	Total exclusive alignments[a]	True Positive alignments	False Positive alignments	True Positive Rate
-3	BLASTx	28035	24615	3420	0.878
	PED_Rap[b]	24805	21776	3029	0.878
-5	BLASTx	18239	15688	2551	0.86
	PED_Rap	18467	15928	2539	0.863

Table 4 [a]: exclusive alignment means a pair of one query and one database reads with specific alignment regions, which is non-redundant in result file. [b]: means paired-Rapsearch.

Conclusion:

Through implementing and testing the "paired-end" module in RAPSearch2, we confirmed that, we can get more hits by taking paired-end reads and evaluate the significance of their HSPs together. Paired-end reads are sequenced from one subject sequence. When they are considered together, their similarity against a single subject sequence should be more significant than the one from each read alone. Thus, by evaluating the hits from paired-end reads against the same subject sequence, we improved the sensitivity of protein similarity search. Since NGS reads are relatively short, in sequence comparison and similarity search, considering paired-end reads together could improve the annotation of NGS reads in query datasets. Finally, our analysis also shows that the alignment results of paired-RAPsearch are as accurate as BLASTx when handling short reads.

Acknowledgement

I'd like to thank Professor Haixu Tang for the idea and helpful advises. I also want to Professor Volker Brendel for advises on my presentation and pushing forward my project progress. As well, thanks to Yongan Zhao for helpful discussion and explanation of his work on RAPSearch2.

References

[1].Smith, T.F. and M.S. Waterman, Identification of common molecular subsequences. J Mol Biol, 1981. 147(1): p. 195-7.

[2].Altschul, S.F., et al., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res, 1997. 25(17): p. 3389-402.

[3].Altschul, S.F., et al., Basic local alignment search tool. J Mol Biol, 1990. 215(3): p. 403-10.

[4].Karlin, S. and S.F. Altschul, Applications and statistics for multiple high-scoring segments in molecular sequences. Proc Natl Acad Sci U S A, 1993. 90(12): p. 5873-7.

[5].Karlin, S. and S.F. Altschul, Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc Natl Acad Sci U S A, 1990. 87(6):p. 2264-8.

[6]. http://blast.ncbi.nlm.nih.gov/Blast.cgi

[7].Kent, W.J., BLAT--the BLAST-like alignment tool. Genome Res, 2002. 12(4): p. 656-64.

[8].Delcher, A.L., et al., Alignment of whole genomes. Nucleic Acids Res, 1999. 27(11): p.2369-76.

[9].Ma, B., J. Tromp, and M. Li, PatternHunter: faster and more sensitive homology search.Bioinformatics, 2002. 18(3): p. 440-5.

[10]. Li, M., et al., Patternhunter II: highly sensitive and fast homology search. J Bioinform Comput Biol, 2004. 2(3): p. 417-39.

[11]. Schwartz, S., et al., Human-mouse alignments with BLASTZ. Genome Res, 2003. 13(1): p.103-7.

[12]. Ye, Y., J.H. Choi, and H. Tang, RAPSearch: a Fast Protein Similarity Search Tool for ShortReads. BMC Bioinformatics, 2011. 12(1): p. 159.

[13]. Zhao Y, Tang H, Ye Y. RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. Bioinformatics. 2012 Jan 1;28(1):125-6.

[14]. Edwards, A; Voss, H.; Rice, P.; Civitello, A.; Stegemann, J.; Schwager, C.; Zimmerman, J.; Erfle, H.; Caskey, T.; Ansorge, W. (1990). "Automated DNA sequencing of the human HPRT locus". Genomics 6 (4): 593-608.

[15]. Edgar RC. Local homology recognition and distance measures in linear time using compressed amino acid alphabets. Nucleic Acids Res. 2004 Jan 16;32(1):380-5. Print 2004

[16]. Roach JC, Boysen C, Wang K, Hood L. Pairwise end sequencing: a unified approach to genomic mapping and sequencing. Genomics.

[17]. Altschul, S.F., et al., Basic local alignment search tool. J Mol Biol, 1990. 215(3): p. 403-10.

[18]. Jensen LJ, Julien P, Kuhn M, von Mering C, Muller J, Doerks T, Bork P. eggNOG: automated construction and annotation of orthologous groups of genes. Nucleic Acids Res. 2008 Jan;36(Database issue):D250-4. Epub 2007 Oct 16

[19]. Giannoukos, et. al., Efficient and robust RNA-seq process for cultured bacteria and complex community transcriptomes, Genome Biology 2012, 13:r23.

[20]. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, Koonin EV, Krylov DM, Mazumder R, Mekhedov SL, Nikolskaya AN, Rao BS, Smirnov S, Sverdlov AV,Vasudevan S, Wolf YI, Yin JJ, Natale DA. The COG database: an updated version includes eukaryotes. BMC Bioinformatics. 2003 Sep 11;4:41.

[21]. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, Koonin EV, Krylov DM, Mazumder R, Mekhedov SL, Nikolskaya AN, Rao BS, Smirnov S, Sverdlov AV, Vasudevan S, Wolf YI, Yin JJ, Natale DA. The COG database: an updated version includes eukaryotes. BMC Bioinformatics.