DUK – A Fast and Efficient Kmer Matching Tool

Mingkun Li[1], Alex Copeland[1], James Han[2]

[1] Lawrence Berkeley National Laboratory
[2]Lawrence Livermore National Laboratory

March 2011

The work conducted by the U.S. Department of Energy Joint Genome Institute is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231

DUK – A Fast and Efficient Kmer Based Sequence Matching Tool

Mingkun Li[1,2], Alex Copeland[1,2] and James Han[1,3]
[1]DOE Joint Genome Institute, 2800 Mitchell Drive, Walnut Creek, CA 94598 USA
[2]Genomics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA
[3]Genomics Division, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA

Abstract

Efficient DNA sequence classification tools are needed to analyze large volumes of data produced by 2nd generation DNA sequencing technologies. Sequence matching is a method that can be used to determine whether a query sequence partially or completely matches a reference sequence. A fast, accurate, versatile, and memory efficient matching program called DUK is developed. DUK relies on a kmer hashing to identify matching DNA sequences. A p-value is calculated from a Poisson distribution to estimate the reliability of matches. Simulated and real sequence data demonstrate its speed, accuracy, and versatility.

Sequence Matching

- Sequence matching is to find whether a query sequence partially or completely matches given reference sequences or not.
- Sequence matching is similar to alignment. But in sequence matching, the position information is not needed.

Applications

- Contamination removal
- Organelle read separation
- Assembly refinement

Advantage

- Fast
  In our experiment, DUK runs at least one order of magnitude faster than aligner such as Megablast and BWA; it also runs faster than Tagdust.
- Memory efficient
  Each DNA base is encoded as two bits. Linear probing is used to remove address pointer's memory usage in hashing table implementation. The memory usage is at most 16 times the number of unique kmers bytes in reference sequences.
- Accurate
  DUK is much more accurate and sensitive than tools used in the benchmark which include Megagblast, bwa, and Tagdust.

Limitation

- Maximum Kmer size is 32
  Currently each kmer is stored as a long integer which is 64 bits. 32 kmer size is sufficient for most applications.

Availability

The c++ source code and binary executable are available under FreeBSD license at http://duk.sourceforge.net/.

Usage

duk [options] ref.fa query

Options:

```
-o, -output  <file>     print log information to file, default is stdout.
-n, -nomatch <file>     output the not matched reads to file,
                        the opton value - stands for standard output
-m, -match <file>       output matched reads to file.
-k, -kmer               the k mer size,  default is 16.
-s, -step               the step size, default is 4.
-c, -cutoff             the cut off threshold for matched reads, default is 1.
-h, -help               print out the help information
```

Identify the reads in query file whether they match to ref.fa reads. The ref.fa must be in fasta format and the query can be in fastq or fasta format. If there is no query file, the tool gets reads from standard input

Examples:

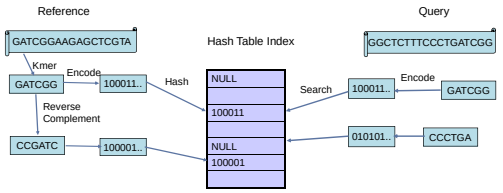
- duk –m matchrds.fq –n nomatchrds.fq ref.fa query.fq
- cat query.fq | duk –k 20 –c 2 –n – ref.fa > nomatchedRds.fq

Parameter Recommendations

- k=16 and c=1 for adapter removal.
- k>=20 and c=2 for organelle read separation or when a reference sequence has high diversity (ex. Metagenomic sequence)

Methods

- **Kmer indexing and searching using hash table**



- **P-value calculation**

Assume kmers are random distributed in the DNA sequence.

Assumes that there are *M* distinct kmers in reference sequences, *N* distinct kmers in the query sequence, kmer size is *k*.

The probability of having *c* or more common kmers between reference and query sequences is:

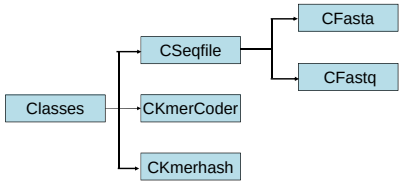
$$1 - pois(c, \mu)$$

where *pois* is the Poisson distribution function and $\mu = M*N/4^k$

A large p-value indicates that the classified match can be attributed to random sequence variation rather than a high probability match event arising from true sequence homology between the query and reference sequences.

- **Object oriented design**

DUK was developed using C++ for its efficiency and object oriented design. Several classes were carefully designed and they can be used for quickly developing other tools.



➢**CSeqfile, CFastq, CFastq**

CSeqfile is the base and virtual class for manipulating DNA sequence files. Currently CFastq and CFasta were implemented to read and write fastq and fasta files, respectively.

➢**CKmerCoder**

CKmerCoder can efficiently extract and convert a kmer from nucleotide bases to two-bit binary coding long integer.

➢**CKmerhash**

CKmerhash efficiently inserts an integer kmer into hash table and find a key from hash table.

Experimental Results

DUK has been widely used at the DOE Joint Genome Institute (JGI) for a range of sequence QC and classification applications. It has been incorporated in JGI sequence QC pipeline and is used to screen adapter contamination in projects. It is also incorporated in JGI in-house galaxy system. Several experiments were performed around these use cases to validate the performance of DUK on simulated and experimentally produced Illumina sequencing reads

1. Adapter Sequence Detection with Simulated Reads

One million 76bp synthetic reads were created by randomly sampling the finished genome sequence of Actinobacillus Succinogenes 130Z. An adapter sequence of at least 20 bases was randomly chosen from a set of known adapters, and attached to the beginning or end of 50% of the artificially created A.Succinogenes sequence reads. Errors were introduced in the synthetic reads at a rate of 0.5% over the entire read to simulate sequencing errors. Specificity and sensitivity were used to measure the accuracy of DUK. The UNIX time command was used to measure the execution speed of the programs in the experiment. Sensitivity was defined as the percentage of predicted contaminated reads among all known contaminated reads. Specificity was defined as the percentage of predicted clean reads among all the clean reads.

	Sensitivity	Specificity	Adapter Detection Rate	Execution Time(s)
DUK	97.26%	99.99%	48.69%	12
Tagdust	53.29%	99.71%	26.82%	28
Megablast	41.16%	100%	21.00%	456

2. Organelle Read Detection and Separation

In this experiment, chloroplast associated reads were separated from Sorghum shotgun genome reads using DUK, Megablast, and BWA with Sorghum bicolor chloroplast as the reference sequence. A 20% sampled dataset from Illumina Hiseq lane yielded 6.9 gigabytes of data with 30 million reads. DUK ran at least an order of magnitude faster than both Megablast and BWA with a minimal difference in the number of detected chloroplast reads. Further analysis showed that more than 99.95% reads detected by Megablast and BWA were also detected by DUK.

	Detection Rate	Time(s)
DUK	6.38%	151
Megablast	6.18%	9962
BWA	5.90%	1557

REFERENCES

Flicek, P., Birney, E., (2009) Sense from sequence reads: methods for alignment and assembly. *Nature Methods*, 6 - 12.

Li, H., and Durbin, R. (2009) Fast and Accurate Short Read Alignment with Burrows-Wheeler Transform. *Bioinformatics*, 25(14), 1754–60.

Lassmann T., et al. (2009) TagDust - A program to eliminate artifacts from next generation sequencing data. *Bioinformatics*.

Zhang Z., et al (2000), A greedy algorithm for aligning DNA sequences. *J Comput Biol 2000*, 7(1-2):203-14

Zerbino, D., Birney E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18 (5): 821–9.

Margulies M, Egholm M, Altman WE, et al (2005). Genome sequencing in microfabricated high-density picolitre reactor. *Nature* 437 (7057): 376–80.

Schuster, Stephan C. (2008). Next-generation sequencing transforms today's biology. *Nature methods* **5** (1): 16–18

Goecks, J, Nekrutenko, A, Taylor, J and The Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* 2010 Aug 25;11(8):R86

Bentley D., et al (2008), Accurate whole human genome sequencing using reversible terminator chemistry, *Nature* 456, 53-59

LLNL-POST-475248

The work conducted by the U.S. Department of Energy Joint Genome Institute is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.