MINIA on a Raspberry Pi, Assembling a 100 Mbp Genome on a Credit Card Sized Computer

Guillaume Collet, Guillaume Rizk, Rayan Chikhi, Dominique Lavenier

Minia on Raspberry Pi

Assembling a 100 Mbp genome on a Credit Card Sized Computer

Guillaume Collet, Guillaume Rizk, Rayan Chikhi, Dominique Lavenier

MINIA: contig de novo assembler

This work shows that the genome assembly program MINIA is able to assemble a 100 Mbp genome on a Raspberry Pi. The MINIA software was developed to drastically reduce the memory footprint needed for genome assembly, enabling human genomes to be assembled on a desktop computer. The efficiency of MINIA is based on the DSK k-mer counting [1] and a compact de Bruijn graph data structure [2]. Here we show that it is also able to successfully assemble a genome on a very low-end, low-power system with 512 MB RAM and a 32 GB flash drive.

K-mer counting with DSK

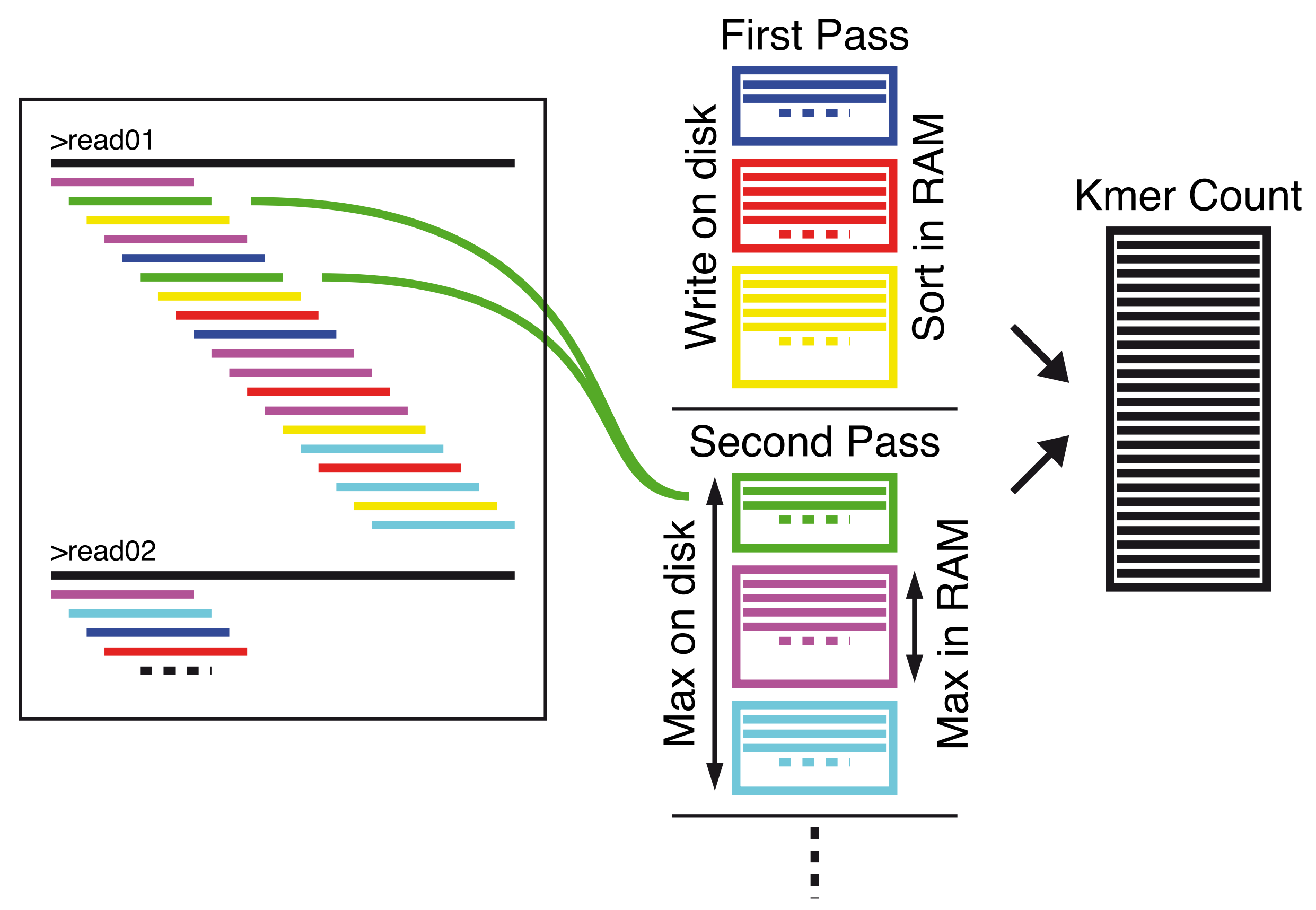


Figure 1: K-mer counting is performed by the fixed-memory and fixed-disk space algorithm DSK (Disk Streaming of K-mers). The set of k-mers is divided in partitions (colored boxes). Each k-mer is written only once on disk. Then, each partition is sorted and k-mers are counted. The trade-off between memory, disk-space, and computation allows to use DSK on a very small system.

Compact de Bruijn graph data structure

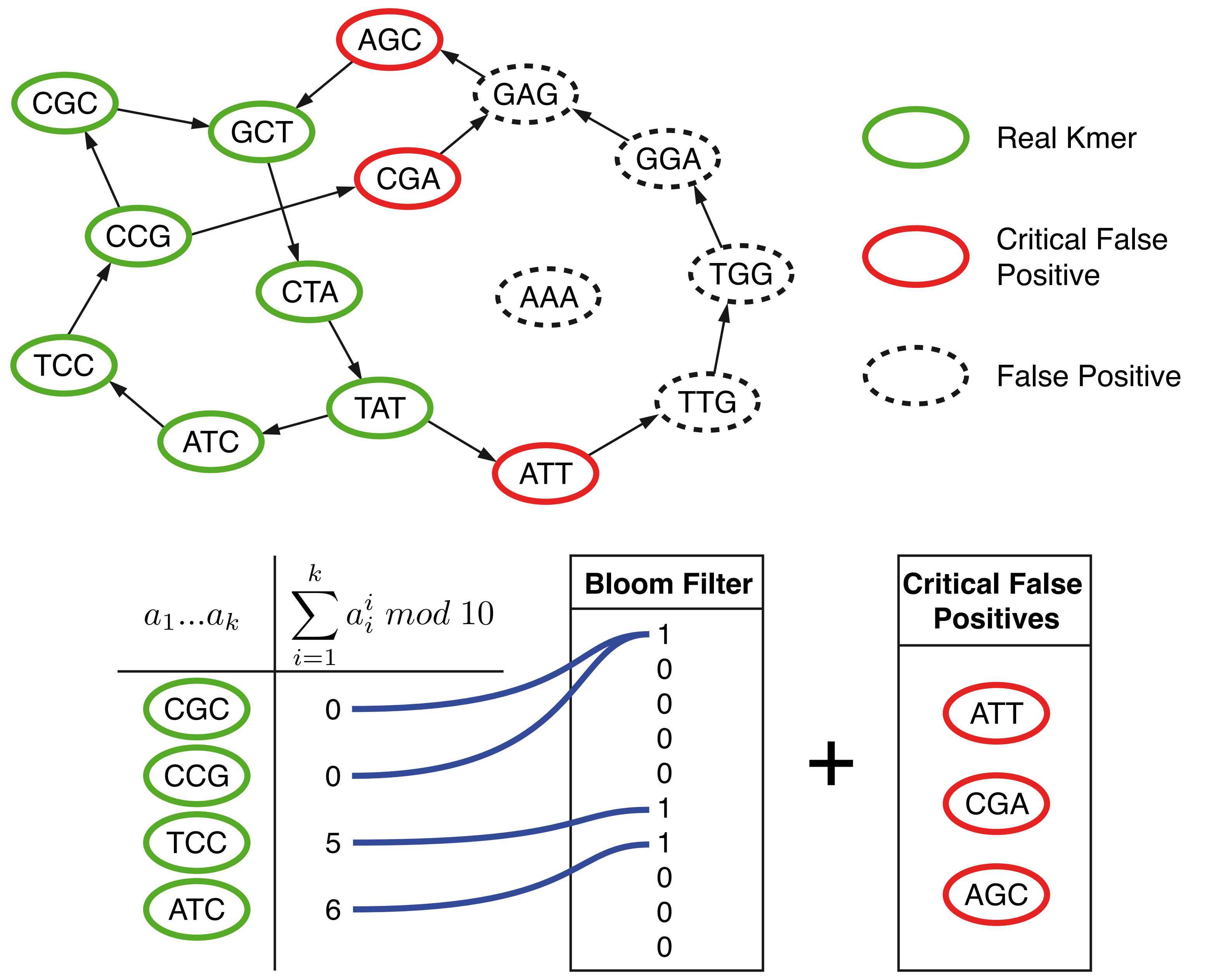


Figure 2: The probabilistic de Bruijn graph representation is obtained by inserting all the k-mers in a Bloom filter. Querying the Bloom filter for the membership of a k-mer may return a false positive answer. To avoid false positives, and consequently false branching, we propose to store the critical false positives only, in a separate structure. Thus false positives are not reachable.

Picontigotron

Raspberry Pi®

35€
ARM11 700MHz
512 MB RAM
16 GB sdcard
32 GB flash drive

Live demo
during
the poster sessions

C. elegans assembly on Raspberry Pi

Our experiment consists in assembling the nematode *C. elegans*. We used 33 million unfiltered paired-end reads of length 100 bp (SRR065390), covering the genome at about 64x. Paired-end information was not used.

Method	Minia	SOAPdenovo	Velvet
System	Raspberry Pi	64GB/Xeon E5462	64GB/Xeon E5462
CPU Time (h)	18.9	6.25	13.5
Peak memory (GB)	**0.2**	29.6	30.6
Number of contigs (K)	29.5	29.5	28.2
Longest contig (Kbp)	75.2	90.9	62.6
Contig N50 (bp)	5741	5975	6031
Sum (Mbp)	86.4	88.3	90.4
Misassemblies	12	7	419
Genome fraction (%)	80.9	82.8	85.0
mismatches (per 100 kbp)	3.2	0.75	25.6

Table 1: De novo *C elegant* contigs assembled by Minia [2], SOAPdenovo2 [4], and Velvet [3]. Assembly quality was computed using the QUAST software [5]. MINIA and Velvet were single-threaded. For SOAPdenovo2, the CPU time is the sum for each thread.

MINIA applications

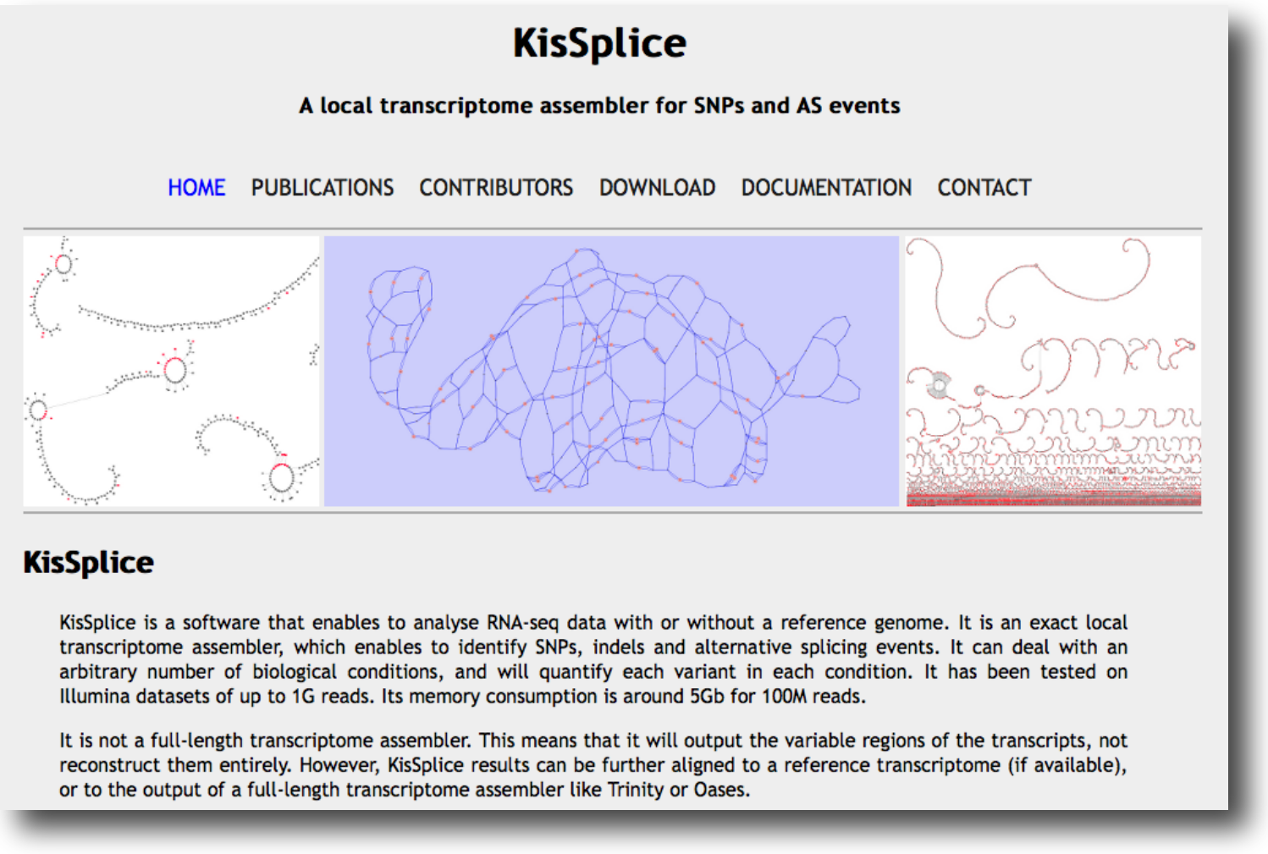
Human genome assembly with less than 6 GB RAM

Colib'read

https://colibread.inria.fr/mapsembler2/
https://colibread.inria.fr/read2snps/

http://minia.genouest.org/

http://kissplice.prabi.fr/



KisSplice

A local transcriptome assembler for SNPs and AS events

HOME PUBLICATIONS CONTRIBUTORS DOWNLOAD DOCUMENTATION CONTACT

KisSplice

KisSplice is a software that enables to analyse RNA-seq data with or without a reference genome. It is an exact local transcriptome assembler, which enables to identify SNPs, indels and alternative splicing events. It can deal with an arbitrary number of biological conditions, and will quantify each variant in each condition. It has been tested on Illumina datasets of up to 1G reads. Its memory consumption is around 5Gb for 100M reads.

It is not a full-length transcriptome assembler. This means that it will output the variable regions of the transcripts, not reconstruct them entirely. However, KisSplice results can be further aligned to a reference transcriptome (if available), or to the output of a full-length transcriptome assembler like Trinity or Oases.

References

[1] G. Rizk, D. Lavenier, and R. Chikhi (2013). DSK: k-mer counting with very low memory usage. Bioinformatics, 29(5), 652-653.

[2] R. Chikhi and G. Rizk (2012). Space-efficient and exact de Bruijn graph representation based on a Bloom filter. In Lecture Notes in Computer Science (Ed.), wabi (Vol. 7534, pp. 236–248).

[3] D. Zerbino and E. Birney (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res., 18, 821-829.

[4] R. Li et al. (2012). SOAPdenovo2: an empirical improved memory-efficient short-read de novo assembler. GigaScience, 1(1), 1-6.

[5] G. Alexey et al. (2013). QUAST: quality assessment tool for genome assemblies. Bioinformatics, 29(8), 1072-1075.