A Fast CUDA Compatible Short Read Aligner to Large Genomes

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Yongchao Liu, Bertil Schmidt and Douglas L. Maskell

NANYANG TECHNOLOGICAL UNIVERSITY

Pattern Search using BWT

Index	Suffixes
0	cattattagga$
1	attattagga$
2	ttattagga$
3	tattagga$
4	attagga$
5	ttagga$
6	tagga$
7	agga$
8	gga$
9	ga$
10	a$
11	$

→

Sorted Suffixes	Suffix Array
$	11
a$	10
agga$	7
attagga$	4
attattagga$	1
cattattagga$	0
ga$	9
gga$	8
tagga$	6
tattagga$	3
ttagga$	5
ttattagga$	2

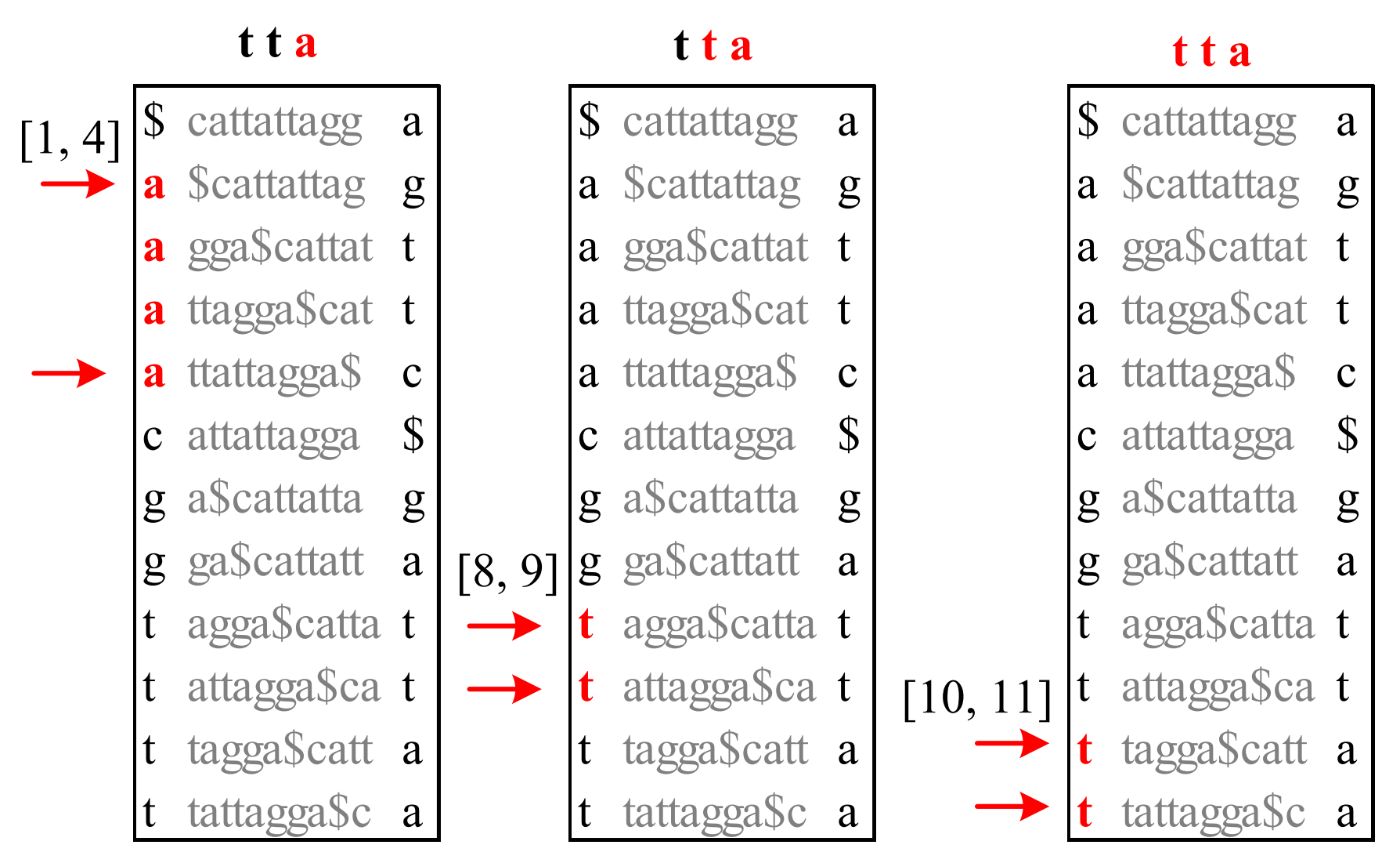
- A suffix array SA of sequence G stores the starting positions of all suffixes of G in lexicographical order;
- SA[i] = j means that the i^{th} lexicographically smallest suffix starts at position j in G.

The BWT of sequence G can be constructed in three steps:

- append a special character $, which is lexicographically smaller than any character in Σ, to the end of G to form a new sequence G$.
- construct a conceptual matrix M_G whose rows are all cyclic rotations of G$ (equivalent to all suffixes of G) sorted in lexicographical order.
- take the last column of M_G to form the BWT of G.

Cyclic Rotations	M_G		BWT
cattattagga$	$cattattagg	**a**	a
attattagga$c	a$cattattag	**g**	g
ttattagga$ca	agga$cattat	**t**	t
tattagga$cat	attagga$cat	**t**	t
attagga$catt	attattagga$	**c**	c
ttagga$catta	cattattagga	**$**	$
tagga$cattat	ga$cattatta	**g**	g
agga$cattatt	gga$cattatt	**a**	a
gga$cattatta	tagga$catta	**t**	t
ga$cattattag	tattagga$ca	**t**	t
a$cattattagg	ttagga$catt	**a**	a
$cattattagga	ttattagga$c	**a**	a

The i^{th} entry in SA has a one-to-one correspondence relationship with the ith row of M_G. M_G has a property called "last-to-first column mapping", which means that the i^{th} occurrence of a character in the last column corresponds to the i^{th} occurrence of the same character in the first column.



	t t a	
$	cattattagg	a
a	$cattattag	g
a	gga$cattat	t
a	ttagga$cat	t
a	ttattagga$	c
c	attattagga	$
g	a$cattatta	g
g	ga$cattatt	a
t	agga$catta	t
t	attagga$ca	t
t	tagga$catt	a
t	tattagga$c	a

Define C(•) to denote an array of length |Σ|, where C(x) represents the number of characters in G that are lexicographically smaller than x∈Σ, and Occ(•) to denote the occurrence array, where Occ(x, i) represents the number of occurrences of x in B[0,i].

Given a substring S of G, we can find all the occurrences of S using a backward search procedure based on the FM-index, which employs the arrays C(•) and Occ(•) to compute the SA interval of S. Thus, using the forward BWT, the SA interval can be recursively calculated, from the rightmost to the leftmost suffixes of S, as

$$\begin{cases} I_a(i) = C(S[i]) + Occ(S[i], I_a(i+1)-1) + 1, & 0 \le i < |S| \\ I_b(i) = C(S[i]) + Occ(S[i], I_b(i+1)), & 0 \le i < |S| \end{cases}$$

where $I_a(i)$ and $I_b(i)$ represent the starting and end indices of the SA interval for the suffix of S starting at position i, and $I_a(|S|)$ and $I_b(|S|)$ are initialized as 0 and |G| respectively. The calculation stops if it encounters $I_a(i+1)> I_b(i+1)$, and the condition $I_a(i)\le I_b(i)$ stands if and only if the suffix of S starting at position i is a substring of G. The total number of the occurrences is calculated as $I_a(0)- I_b(0)+1$ if $I_a(0)\le I_b(0)$, and 0, otherwise.

Abstract

New high-throughput sequencing technologies have promoted the production of short reads with dramatically low unit cost. The explosive growth of short read datasets poses a challenge to the mapping of short reads to reference genomes, such as the human genome, in terms of alignment quality and execution speed.

We present CUSHAW, a parallelized short read aligner based on the compute unified device architecture (CUDA) parallel programming model. We exploit CUDA-compatible graphics hardware as accelerators to achieve fast speed. Our algorithm employs a quality-aware bounded search approach based on the Burrows-Wheeler transform (BWT) and the Ferragina Manzini (FM)-index to reduce the search space and achieve high alignment quality. Performance evaluation, using simulated as well as real short read datasets, reveals that our algorithm running on one or two graphics processing units (GPUs) achieves significant speedups in terms of execution time, while yielding comparable or even better alignment quality for paired-end alignments compared to three popular BWT-based aligners: Bowtie, BWA and SOAP2 (availability: http://cushaw.sourceforge.net)

References

1. Yongchao Liu, Bertil Schmidt, and Douglas L. Maskell: "CUSHAW: a CUDA compatible short read aligner to large genomes based on the Burrows-Wheeler transform", Bioinformtics, 2012, doi: 10.1093/bioinformatics/bts276.
2. Yongchao Liu and Bertil Schmidt: Evaluation of GPU-based seed generation for computational genomics using Burrows-Wheeler transform. 11th IEEE International Workshop on High Performance Computational Biology (HiCOMB 2012).
3. Paolo Ferragina and Giovanni Manzini: Indexing compressed text. Journal of the ACM 2005, 52: 4.

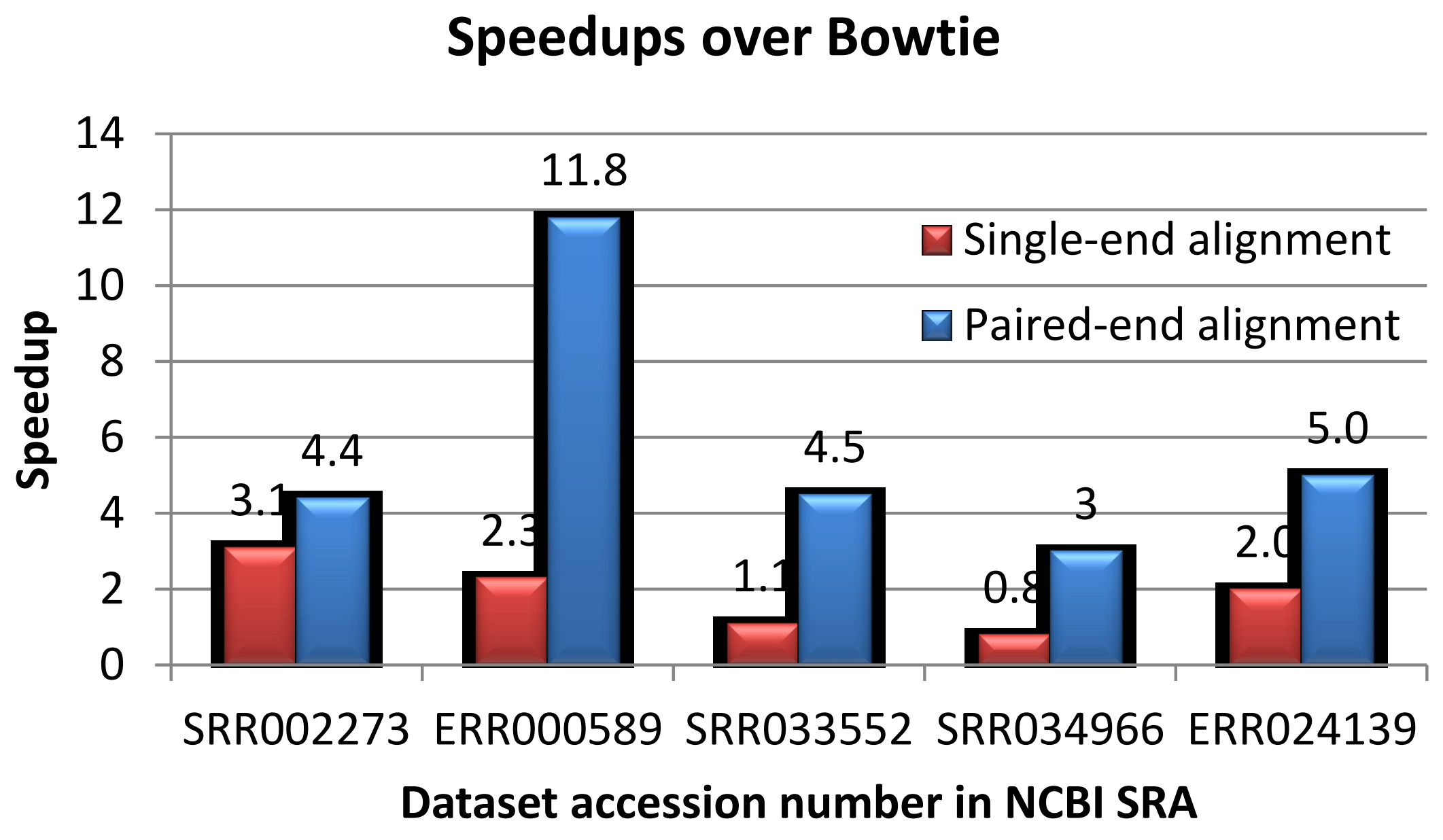
Contact

Yongchao Liu: liuy@uni-mainz.de
Bertil Schmidt: bertil.schmidt@uni-mainz.de
Douglas L. Maskell: asdouglas@ntu.edu.sg

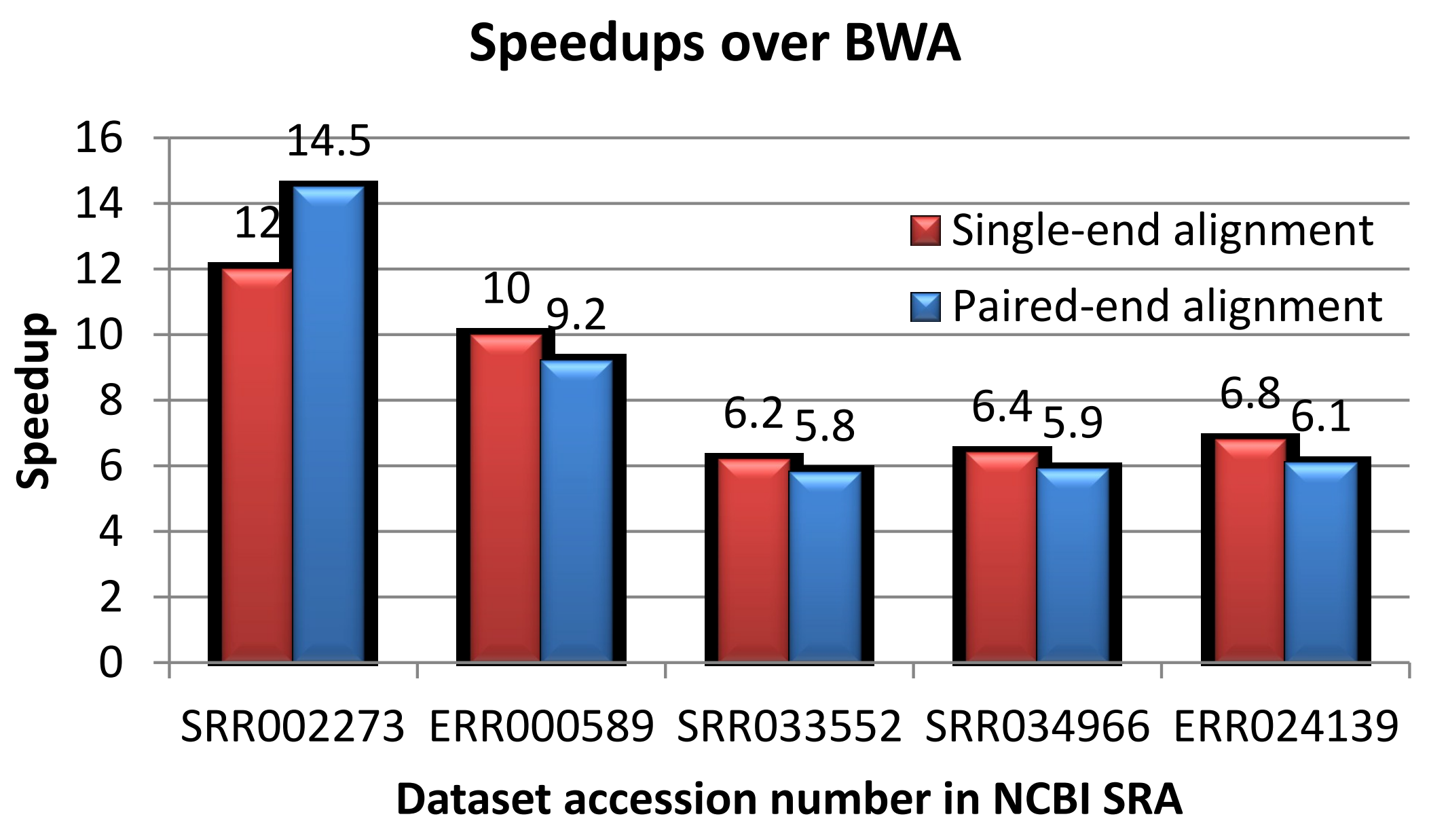
Performance Evaluation

Datasets	Type	CUSHAW	Bowtie	BWA	SOAP2
SRR002273	SE	92.58	94.69	90.26	94.85
	PE	95.77	87.56	90.85	87.89
ERR000589	SE	94.76	96.51	94.06	96.87
	PE	97.72	92.42	94.60	91.09
SRR033552	SE	88.71	91.70	89.12	92.03
	PE	94.46	86.86	92.35	82.17
SRR034966	SE	78.89	91.25	85.10	85.10
	PE	90.56	90.55	90.51	73.11
ERR024139	SE	89.69	92.09	93.28	92.68
	PE	95.11	87.58	94.46	86.25

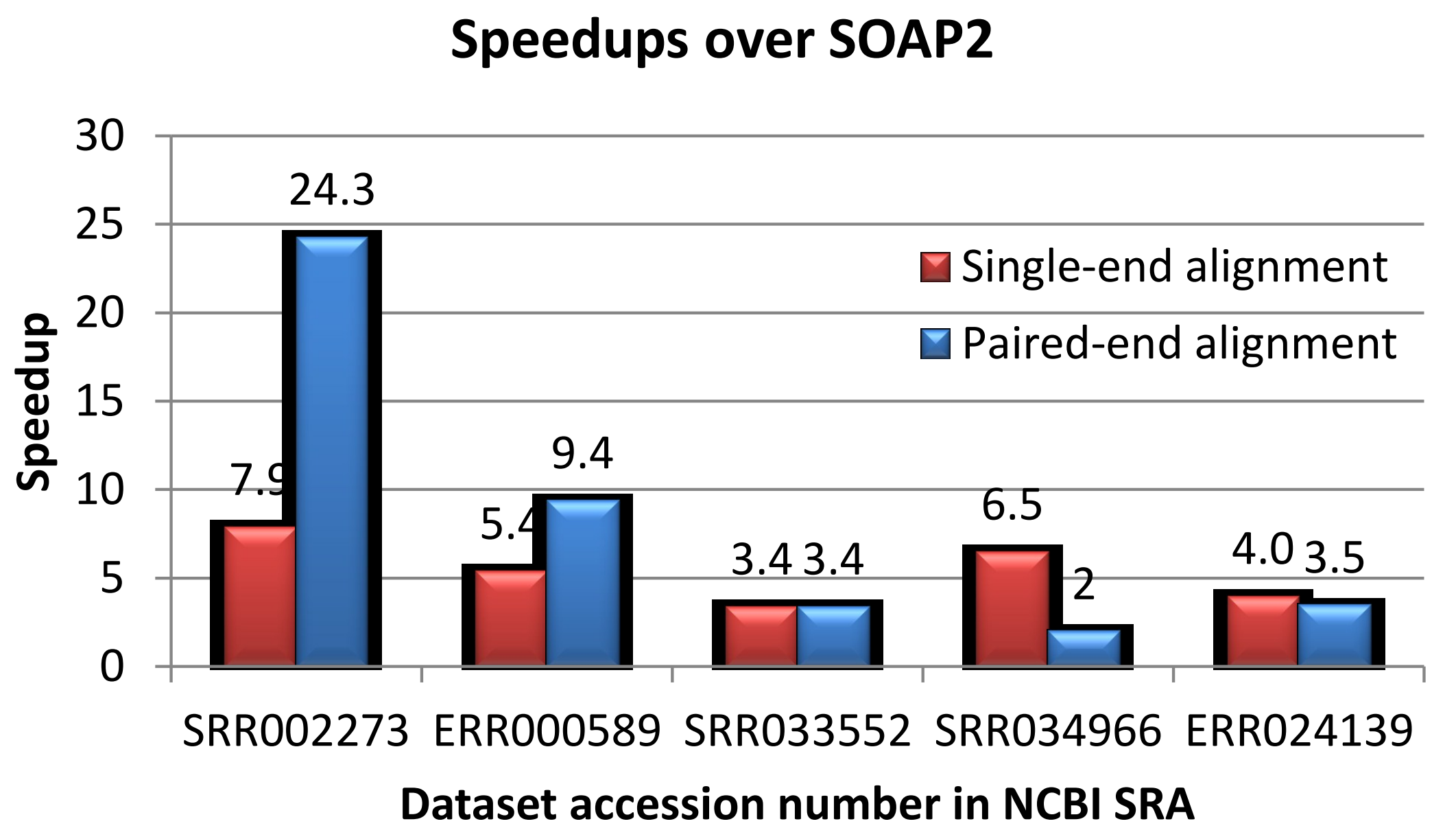
Percentages of aligned (paired) real reads for single-end and paired-end alignments.



A highest speedup of 3.1 (11.8) is achieved by CUSHAW on a single Tesla C2050 GPU over Bowtie on a single AMD 2.4 GHz CPU core for single-end (paired-end) alignments.



A highest speedup of 12 (14.5) is achieved by CUSHAW on a single Tesla C2050 GPU over BWA on a single AMD 2.4 GHz CPU core for single-end (paired-end) alignments.



A highest speedup of 7.9 (24.3) is achieved by CUSHAW on a single Tesla C2050 GPU over SOAP2 on a single AMD 2.4 GHz CPU core for single-end (paired-end) alignments.