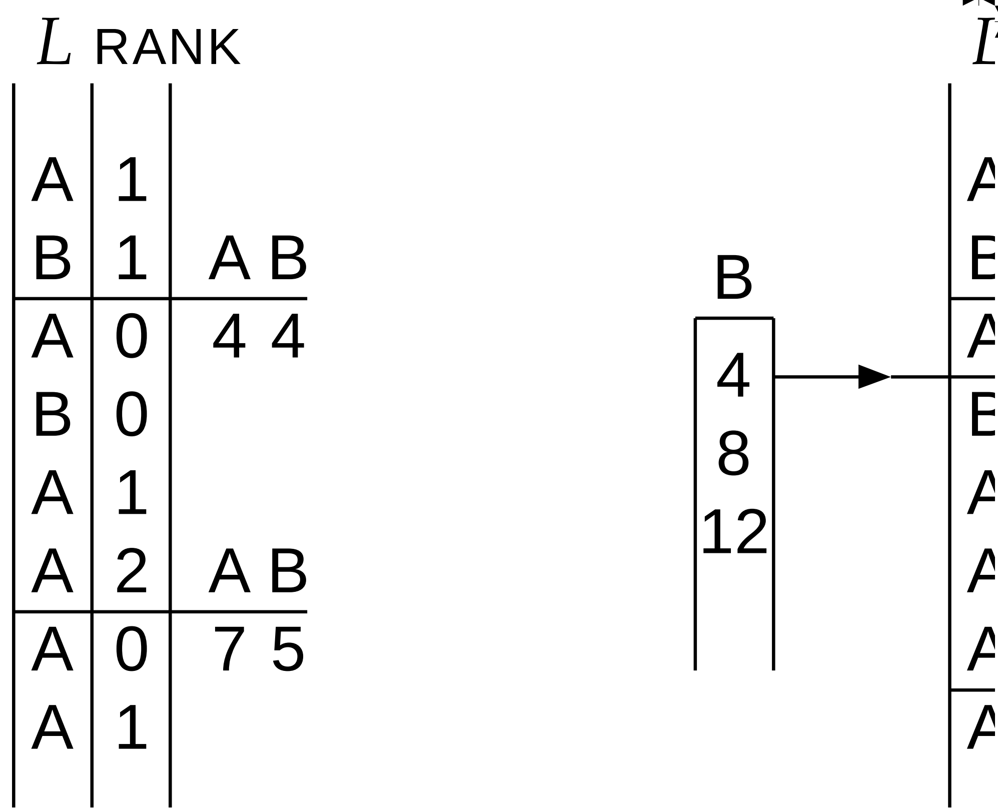
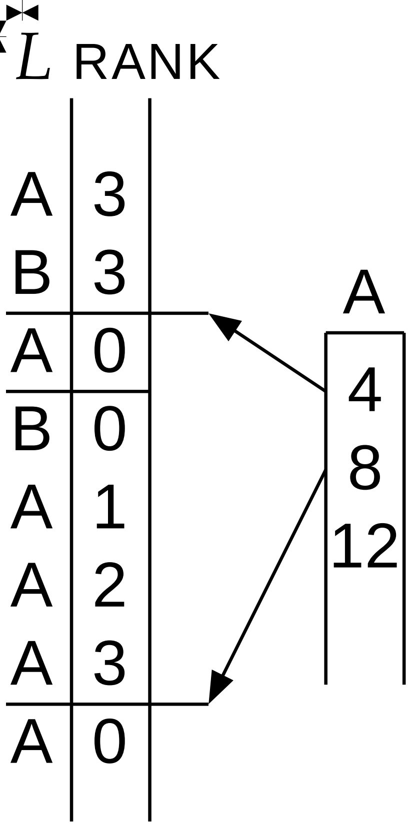
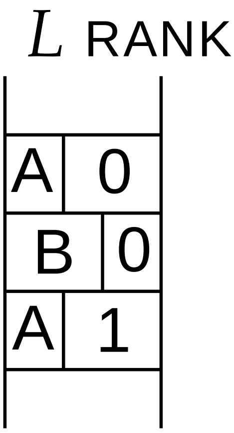


A new improvement is to use variable length encoding, where a frequent symbol uses less bits for symbol and more bits for rank.

Finally, we can trade time for space RANK with scanning from the nea point [1].





…e by replacing
…arest reference

WAVELET TREES

Wavelet tree is a text rep…
both compressed and pre…
ries with little additiona…
with compressed text in…
eral rank queries:

$$\mathrm{RANK}_c(j) = \left|\{i \mid i \cdots$$

We need wavelet trees fo…

$$\mathrm{RANK}(j) = \mathrm{I}\cdots$$

We use our own wavele…
optimized for special ran…

We combine wavelet tre…
ranks, obtaining the *most*…

presentation that can be
reprocessed for rank que-
al space. They are used
ndexes [2] to answer *gen-*

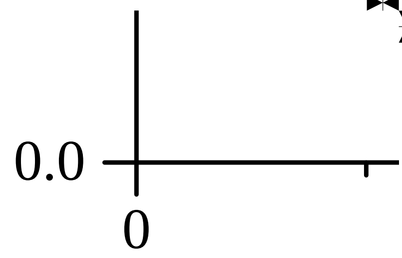
$< j$ and $L[i] = c\}\big|.$

or *special rank queries*:

$\textsc{rank}_{L[j]}(j).$

let tree implementations
nk queries.

ees with reference point
t space-efficient algorithm.

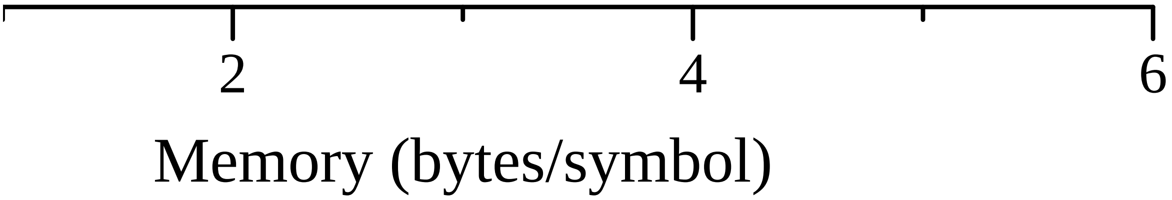


REFERENC

[1] U. Lauther a
rithms for th
In *Proc. 13th*
gorithms, vo
Springer, 200

[2] G. Navarro a
indexes. *AC*
2007.

[3] J. Seward. S
transform. In
ence, pages



CES

and T. Lukovszki. Space efficient algo-
ne Burrows-Wheeler backtransformation.
th Annual European Symposium on Al-
olume 3669 of *LNCS*, pages 293–304.
05.

and V. Mäkinen. Compressed full-text
CM Computing Surveys, 39(1):Article 2,

Space-time tradeoffs in the inverse B-W
n *Proc. IEEE Data Compression Confer-*
s 439–448. IEEE, 2001.

The basic inversion algorithm des
has linear time and space complex
dominates the time and space requ
ing decompression in programs lik

It is slow because each memory a
the permutation traversal is essen
causing many cache misses.

It needs a lot of space for the RANK

$$|\mathrm{RANK}| = n \log n \text{ bits} = 4n$$
$$|\text{text}| = n \log \sigma \text{ bits} = n\ \text{b}$$

where n = text length and σ = alp

REFERENCE POINT RANKS

We reduce space by storing ranks r
erence points, which can be placed

Every kth position [1] Every kth oc

escribed above
xity, but it still
uirements dur-
ke bzip2.

access during
ntially random

K array:

n bytes
bytes

phabet size.

S

relative to ref-
d in two ways:

ccurrence [new]

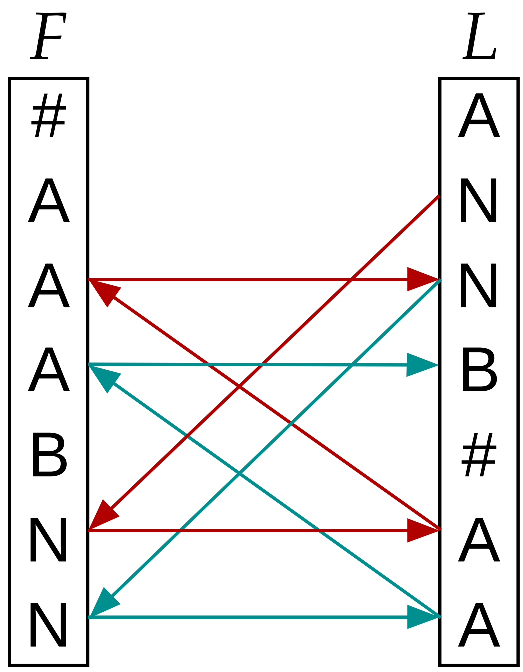
N RANK

Grid (2 , 2)

REPETITION SHOR

Repetitions in the text ma
paths (PPP) in the inverse
use this as follows.

1. On the first pass,
observe the PPP
(due to repeated ANA)



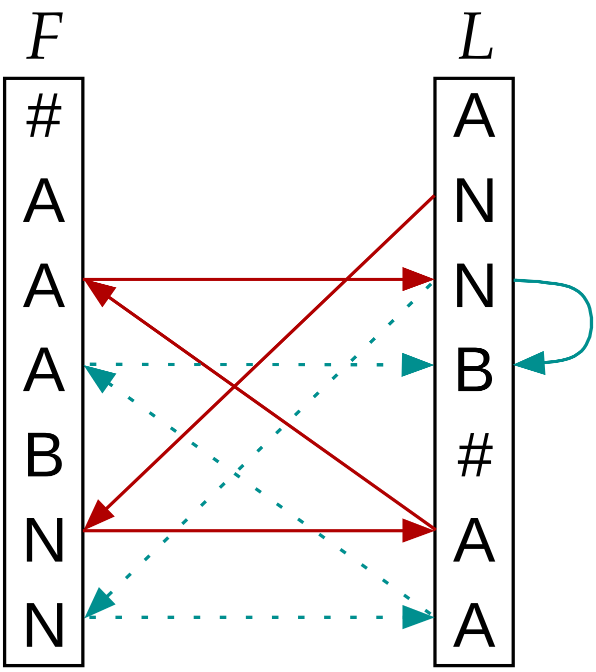
The shortcuts reduce the
This is the *fastest* known

WAVELET TREES

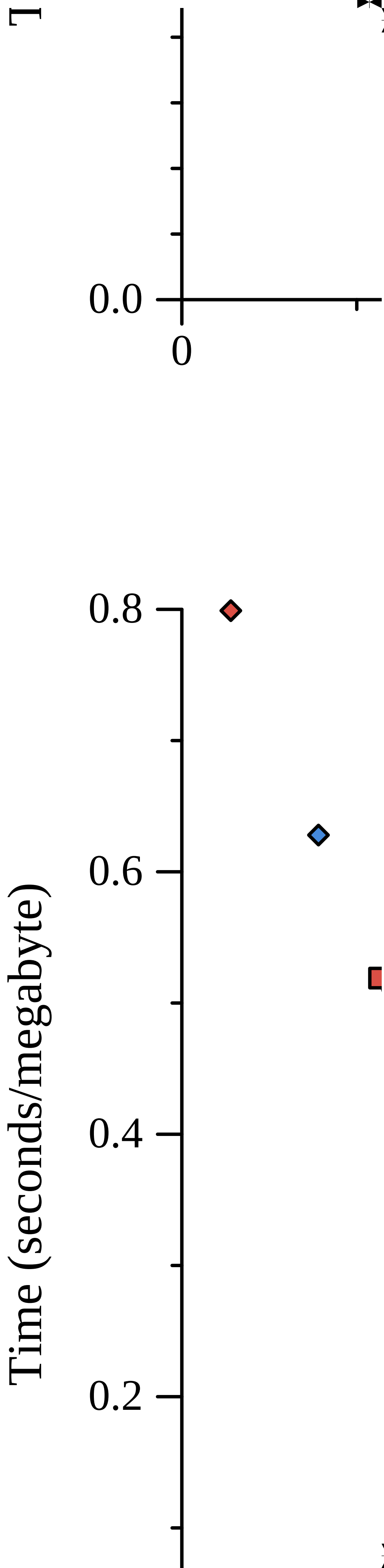
RTCUTS

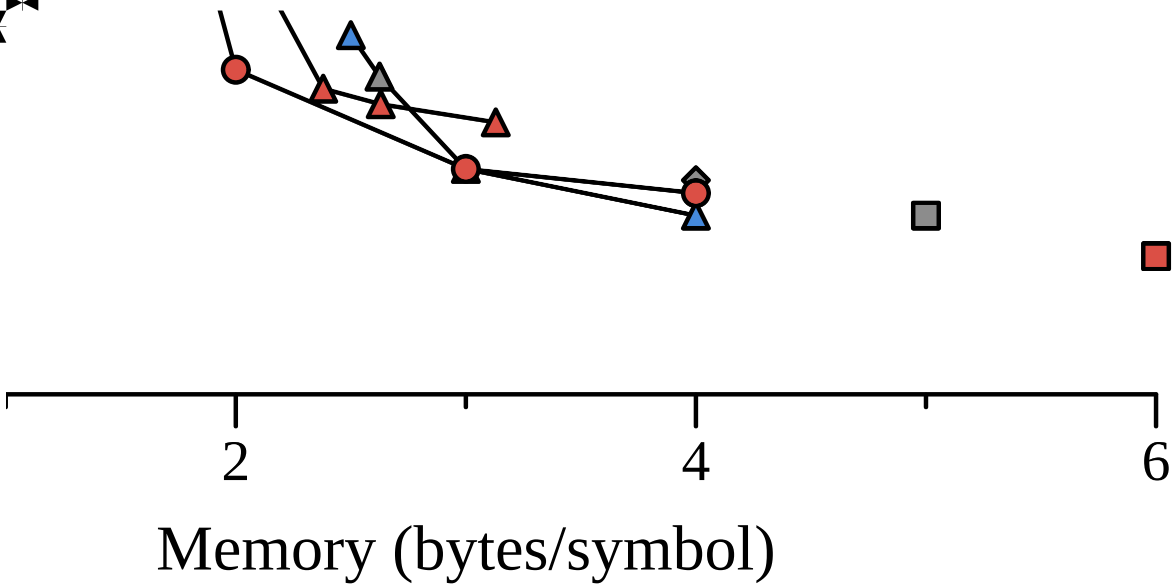
anifest as *pairs of parallel*
se BWT permutation. We

2. Replace the other path
by shortcut and follow it
on the second pass

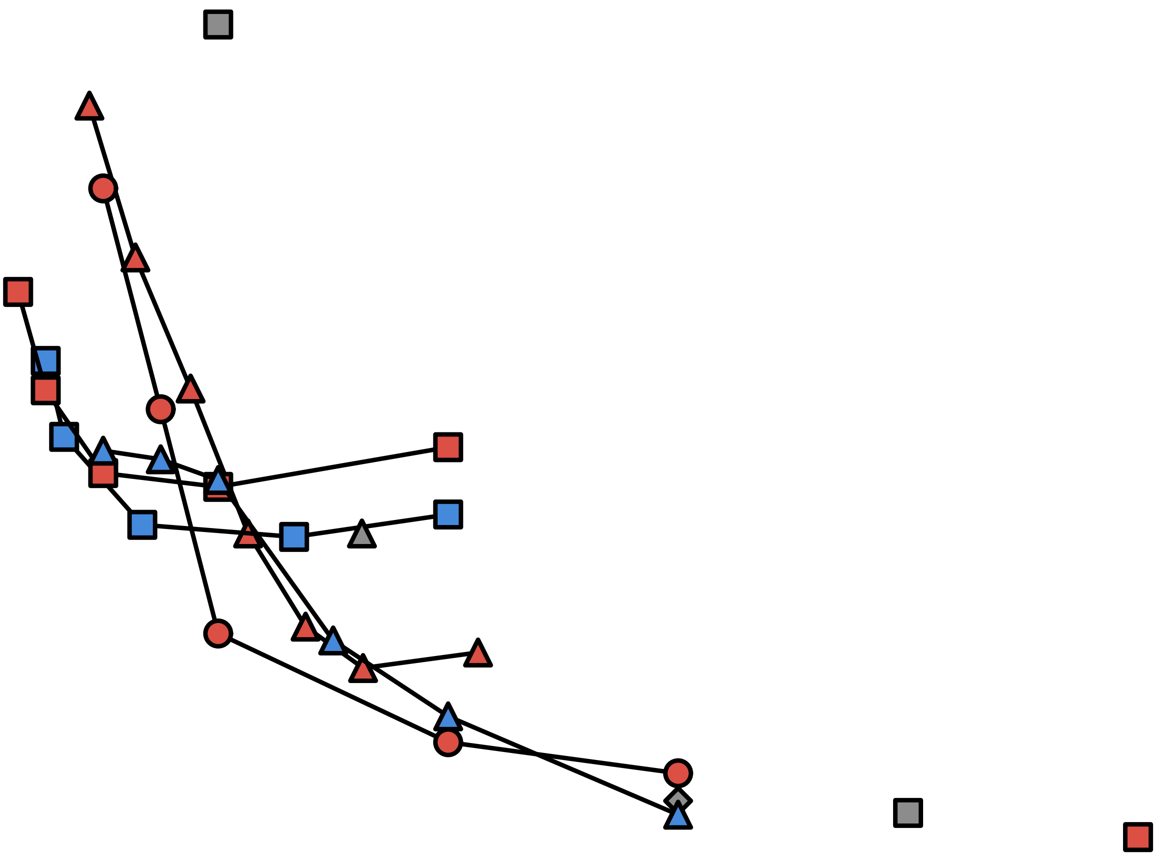


e number of cache misses.
algorithm.





DNA 100MB



The Burrows–Wheeler transform (B
vertible text transform defined as f

Input: text $T =$ BANANA#

1. Build a matrix with the text *rotations* as rows

B	A	N	A	N	A	#
A	N	A	N	A	#	B
N	A	N	A	#	B	A
A	N	A	#	B	A	N
N	A	#	B	A	N	A
A	#	B	A	N	A	N
#	B	A	N	A	N	A

2. Sor

F		
#	B	A
A	#	B
A	N	A
A	N	A
B	A	N
N	A	#
N	A	N

Output: BWT $L =$ ANNB#AA (the last

The properties of the BWT make it
press than the original text. It is us
stage in many compression progra
the widely used bzip2 (thus the b).

NEW ALGORITHMS FOR IN

(BWT) is an in-
follows.

ort the rows

				L
A	N	A	N	A
B	A	N	A	N
A	#	B	A	N
A	N	A	#	B
N	A	N	A	#
#	B	A	N	A
N	A	#	B	A

st column)

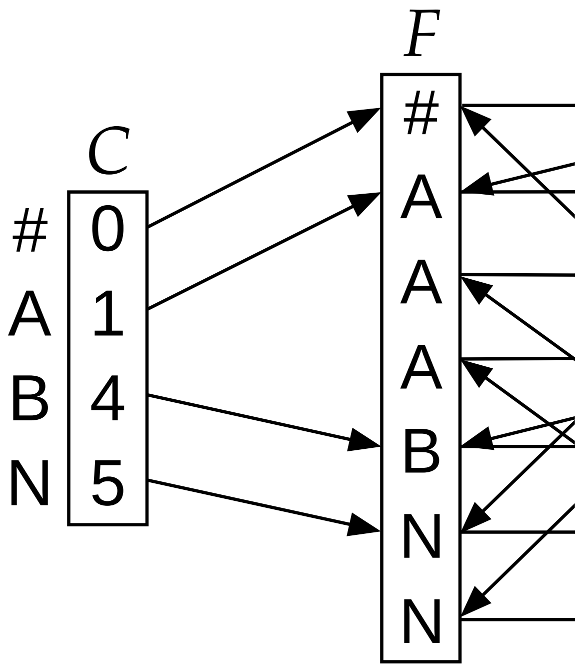
easier to com-
sed as the first
rams including
).

NVERSE BWT

Define $\textsc{rank}(j) = |\{i \mid i$
The BWT can be invertec

Input: BWT $L =$ ANNB#A

1. Compute C and $\textsc{rank}$ ar



2. Starting at $L[i] = $ #, follo

$$i \mapsto C[L[i]]$$

Output $L[i]$ at each step

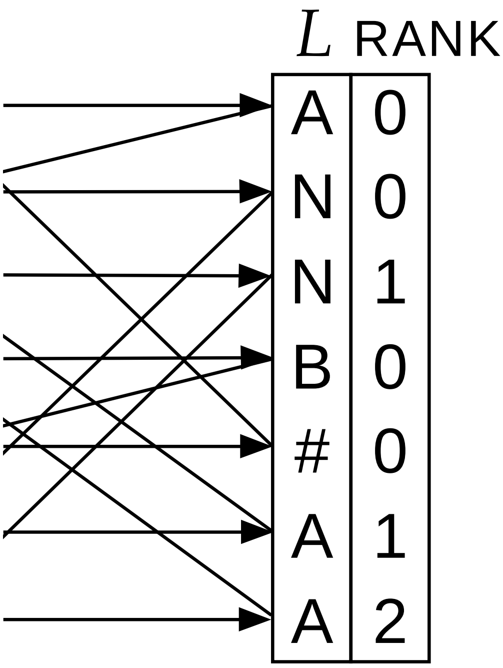
Output: reverse text $T^R =$

$i < j$ and $L[i] = L[j]\}|$.

d as follows.

AA

rrays by scanning L



ow the permutation:

$] + \mathrm{RANK}(i)$

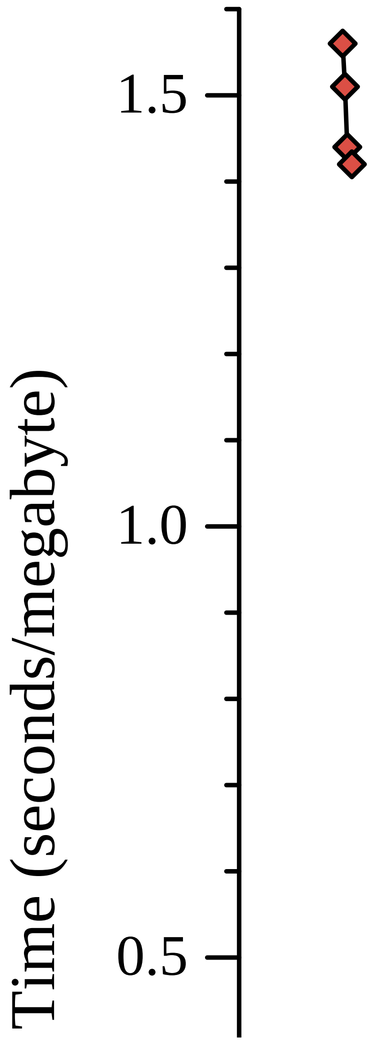
$=$ #ANANAB

The graphs be
quirements of
The algorithms

New algorith
repetiti

Improved im
algorith

Prior algorith



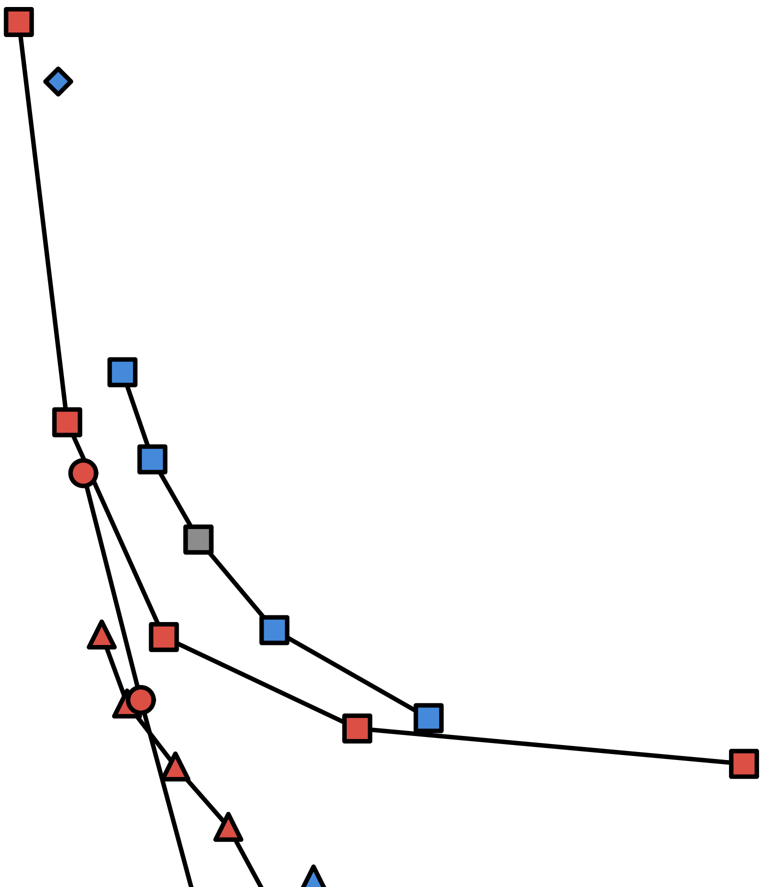
elow show the time and space re-
f several algorithms on two texts.
ıs are divided into three groups:

ıms based on reference point ranks,
ion shortcuts and wavelet trees

ıplementations of wavelet trees and
hms from [1]

ıms from [3, 1]

ENGLISH 100MB



Grid (3 , 4)

IMPROVING INV
THROUGHOUT

The Burrows-Wheeler transform
pression used for example in the
The *inverse* BWT is usually the b
with respect to both space and ti

BURROWS–WHEELER TRA

The Burrows–Wheeler transform (B

/ERSE BURROWS–\
THE SPACE–TIME S

ı (BWT) is a powerful tool for data com-
e popular compression program bzip2.
oottleneck in the decompression phase
time.

ANSFORM

(BWT) is an in-

INVERSE BWT

Define $\mathrm{RANK}(i) = |\{j \mid j$

MATEMAATTIS-LUON
MATEMATISK-NATU

WHEELER TRANSFO
SPECTRUM

Juha K

Our new algorithms improve the perfo range from the fastest known algorithm and cover the whole space-time tradeo

EXPERIMEN

$i \le i$ and $L[i] = L[j]\}|$

The graphs be

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
NNONTIETEELLINEN TIEDEKUNTA
URVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

ORM

Kärkkäinen and Simon Puglisi

---

formance of inverse BWT. They n to the most space-efficient one, eoff spectrum in between.

NTAL RESULTS

elow show the time and space re-

Grid (4 , 4)