Lossless Compression Methods for Archiving Nanopore DNA Signal Data

Sasha Jenner
John Stavrakakis
Ira Deveson
Hasindu Gamaarachchi

B Science and B Adv Studies

Motivation

Human DNA

500 000 000 000 000
data points

Motivation

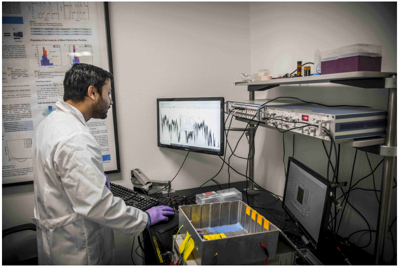
Walk around Earth
7700 times

500 000 000 000
steps

Motivation

1. Record

2. Analyse



3. Archive

Motivation

1 PB / year

$\Downarrow$

Compression

State-of-the-Art

Space saving: 65.9%
Downside: Too generic

Objective

Design compression method

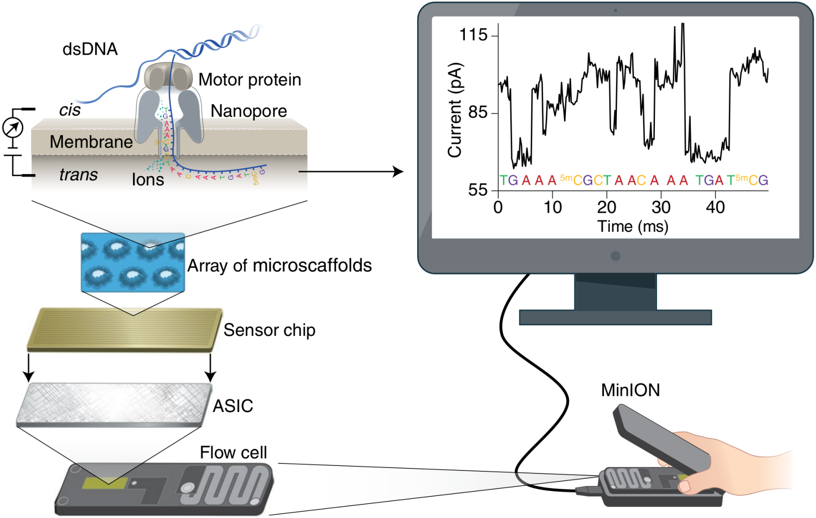
1. More space saving
2. Suitable for nanopore

Objective

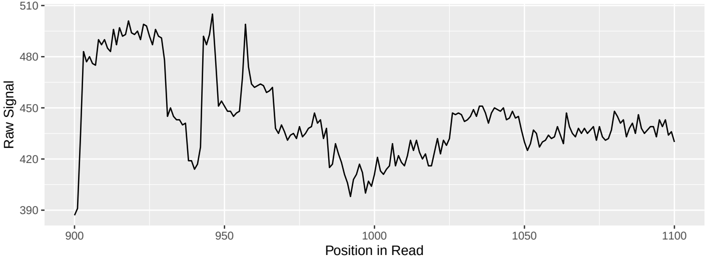
Suitable?

- Lossless
- Better than naive entropy ($>$52%)
- Random access

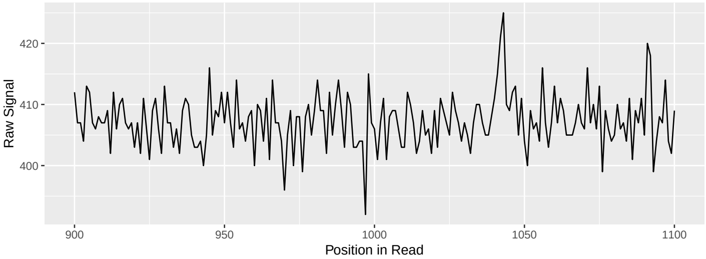
Background


dsDNA
Motor protein
cis
Nanopore
Membrane
trans
Ions
Array of microscaffolds
Sensor chip
ASIC
Flow cell
115
85
55
Current (pA)
TG A A A 5mCGCT A AC A A A TGA T5mCG
0
10
20
30
40
Time (ms)
MinION

Background


Read 1

…


Read 500 000

Background

Read 1

…,462,455,463,464,466,467,460,464,465,463,…

…

Read 500 000

…,407,411,412,400,408,402,402,407,409,406,…

Background

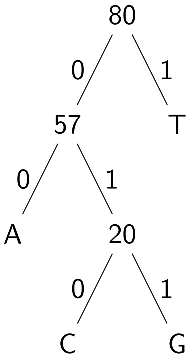
Entropy $H(X)$: measure of information

Coin toss $=$ 1 bit
Dice throw $=$ 2.58 bits
Nanopore data $=$ 7.70 bits

Huffman coding

AACATTAAAC AATTCAAATG
TGTGTGCGTC TGTCTGAATT
CATTTAATTA TTCGTTAATT
GATTTTCTAC ACAATTAATA

Symbol	Frequency	Code
A	27	00
C	11	010
G	9	011
T	33	1



AAC: 0000010

Range coding

AACATTAAAC AATTCAAATG
TGTGTGCGTC TGTCTGAATT
CATTTAATTA TTCGTTAATT
GATTTTCTAC ACAATTAATA

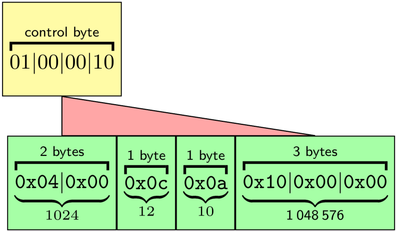
Symbol	Frequency	Range
A	27	$[0, 2700)$
C	11	$[2700, 3800)$
G	9	$[3800, 4700)$
T	33	$[4700, 8000)$

$\emptyset$: $[0, 8000)$
A: $[0, 2700)$
AA: $[0, 911)$
AAC: $[307, 433)$
...

Background

Stream VByte (svb)

Num Bytes	Control Code
1	00
2	01
3	10
4	11



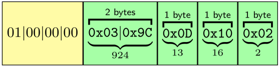
State-of-the-Art (zstd-svb-zd)

1. Nanopore data
2. Differences (delta)
3. Map to unsigned (zig-zag)
4. Stream VByte
5. Zstandard

462,455,463,464

462,-7,8,1

924,13,16,2



. . .

Test Data (5%)

Description	Human DNA
No. of reads	500 000
No. of data points	57 000 000 000
Avg. read length	113 471
Size	106 GiB

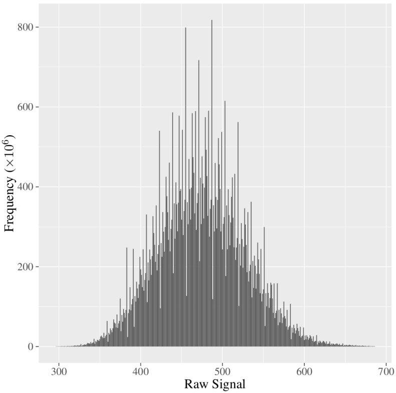
Method

Differential Coding

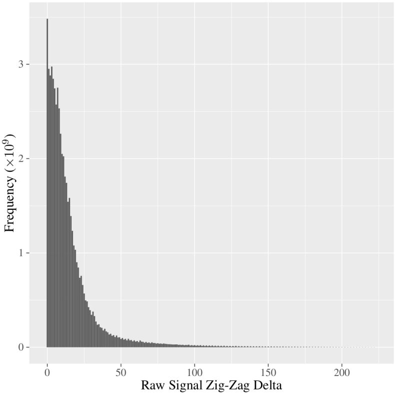
Transformation	None	Delta	Zig-Zag Delta
Min	158	-1159	0
Q1	439	-5	4
Q2	474	0	10
Q3	511	5	18
Max	1748	913	2317
Mean	475.2245	~ 0	15.5679
Mode	487	0	0
SD	35.0675	13.0625	20.6060
Entropy	7.70	5.39	5.39

Method

None

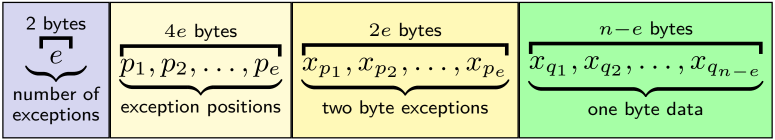


Zig-Zag Delta



Method

Remove redundancy (vbe21)

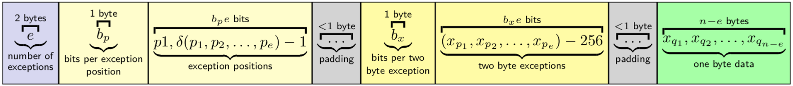


$$\underbrace{\overbrace{e}^{\text{2 bytes}}}_{\text{number of exceptions}} \quad \underbrace{\overbrace{p_1, p_2, \ldots, p_e}^{4e \text{ bytes}}}_{\text{exception positions}} \quad \underbrace{\overbrace{x_{p_1}, x_{p_2}, \ldots, x_{p_e}}^{2e \text{ bytes}}}_{\text{two byte exceptions}} \quad \underbrace{\overbrace{x_{q_1}, x_{q_2}, \ldots, x_{q_{n-e}}}^{n-e \text{ bytes}}}_{\text{one byte data}}$$

Space saving improvement: 6.24%

Method

Even smaller (vbbe21)



2 bytes $\underbrace{e}$ number of exceptions

1 byte $\underbrace{b_p}$ bits per exception position

$b_p e$ bits $\underbrace{p1, \delta(p_1, p_2, \ldots, p_e) - 1}$ exception positions

<1 byte $\underbrace{\ldots}$ padding

1 byte $\underbrace{b_x}$ bits per two byte exception

$b_x e$ bits $\underbrace{(x_{p_1}, x_{p_2}, \ldots, x_{p_e}) - 256}$ two byte exceptions

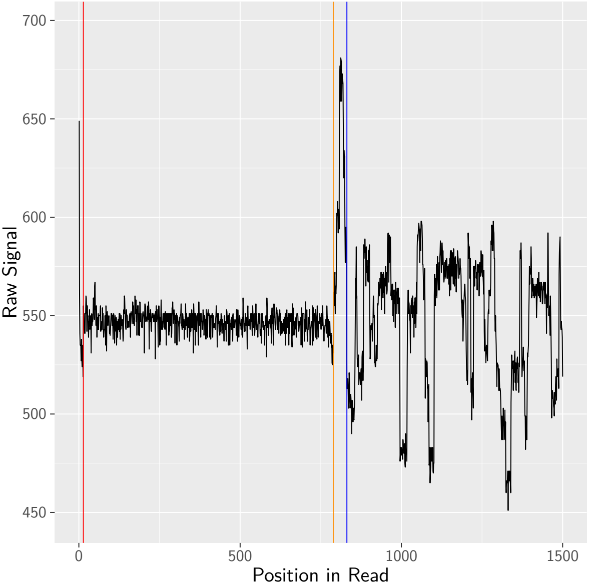
<1 byte $\underbrace{\ldots}$ padding

$n - e$ bytes $\underbrace{x_{q_1}, x_{q_2}, \ldots, x_{q_{n-e}}}$ one byte data

Space saving improvement: 6.25%

Method

The Stall



Method

Encoding the Stall

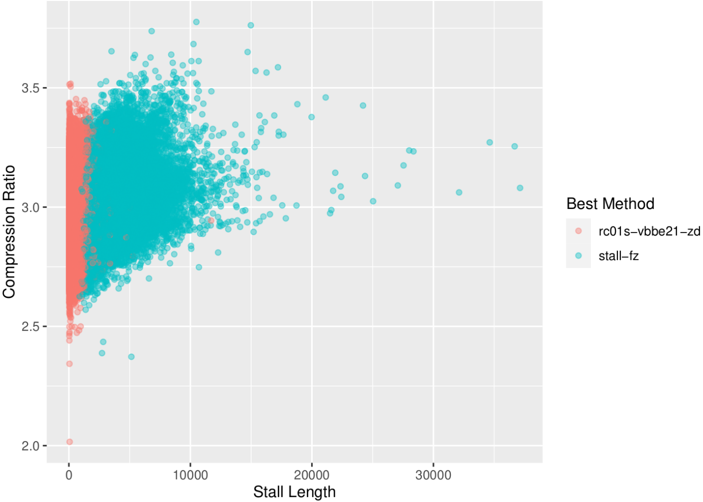
$\underbrace{\overbracket{p}^{\text{2 bytes}}}_{\text{stall start position}}$	$\underbrace{\overbracket{\lvert r_s\rvert}^{\text{2 bytes}}}_{\text{stall length}}$	$\underbrace{\overbracket{m_s}^{\text{2 bytes}}}_{\text{stall compressed size}}$	$\underbrace{\overbracket{C_{specific}(r_s)}^{m_s\text{ bytes}}}_{\text{stall compressed data}}$	$\underbrace{\overbracket{m}^{\text{4 bytes}}}_{\text{non-stall compressed size}}$	$\underbrace{\overbracket{C_{generic}(r \setminus r_s)}^{m\text{ bytes}}}_{\text{non-stall compressed data}}$

Specific: Frame of reference +

Generic: Zig-zag delta

= stall-fz

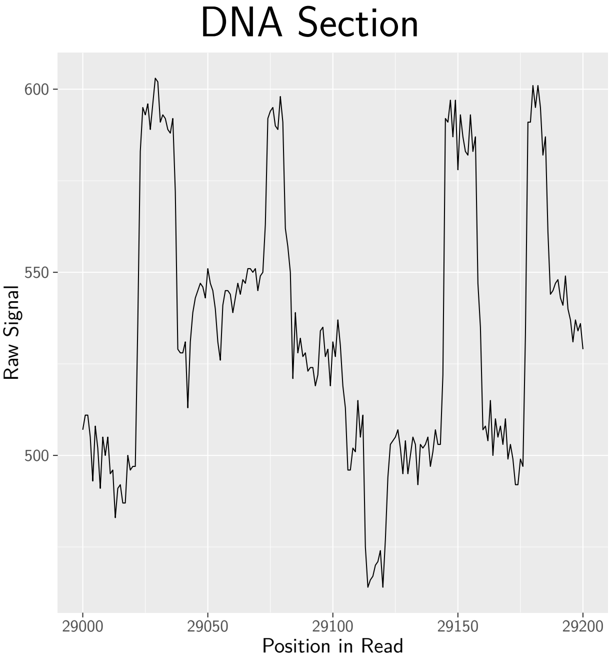
Dynamic Stall



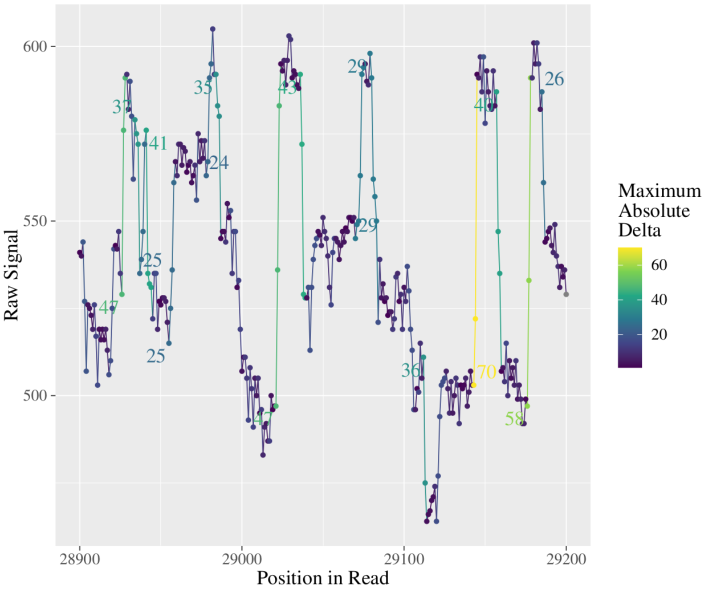
dstall: Choose best

dstall-1500: If length $\geq$ 1500, encode stall

Method

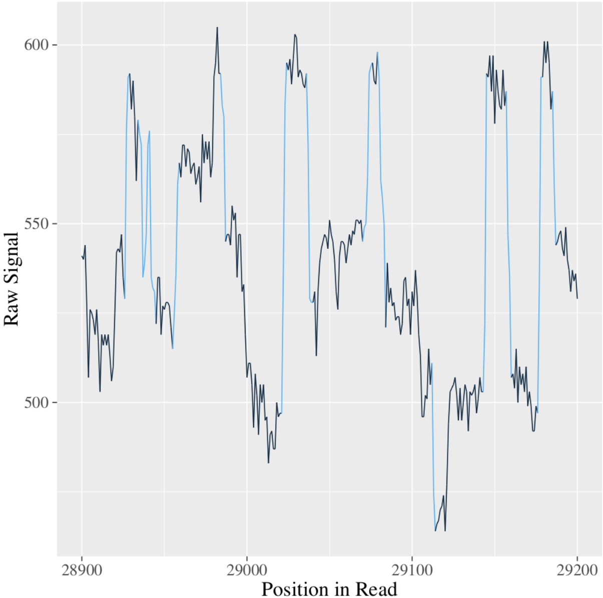

DNA Section
Raw Signal
600
550
500
Position in Read
29000
29050
29100
29150
29200

Jumps and Falls



Method

Minimum Absolute Delta = 25



Results

First Benchmark

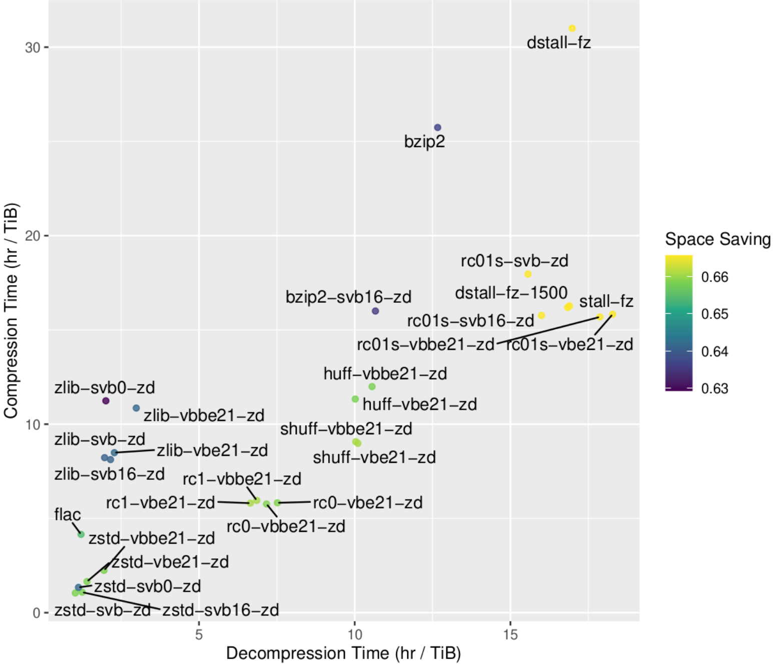
- Sequential (de)compression
- Lossless
- Size and time

`https://github.com/sashajenner/honours`

Results

Method	Compression Ratio	Space Saving	Bits Per Symbol	Compressed Size (GiB)
none	1.000000	0.00000000	16.000000	105.67848
svb-zd	1.599930	0.37497255	10.000527	66.05195
svb0-zd	1.682548	0.40566348	9.509468	62.80858
svb16-zd	1.777690	0.43747228	9.000523	59.44707
zstd	1.790916	0.44162666	8.934052	59.00804
vbe21-zd	1.999519	0.49987982	8.001993	52.85194
vbbe21-zd	1.999714	0.49992849	8.001215	52.84680
zlib	2.001465	0.50036604	7.994214	52.80056
zlib-svb0-zd	2.697205	0.62924589	5.932118	39.18073
bzip2-svb16-zd	2.742621	0.63538529	5.833887	38.53193
bzip2	2.750089	0.63637539	5.818045	38.42729
zlib-svb-zd	2.783474	0.64073678	5.748262	37.96639
zlib-svb16-zd	2.786146	0.64108121	5.742751	37.92999
zstd-svb0-zd	2.789808	0.64155240	5.735212	37.88020
zlib-vbe21-zd	2.790276	0.64161254	5.734250	37.87384
zlib-vbbe21-zd	2.790488	0.64163978	5.733814	37.87096
flac	2.893409	0.65438689	5.529859	36.52387
huff-vbe21-zd	2.927298	0.65838802	5.465840	36.10103
huff-vbbe21-zd	2.927709	0.65843599	5.465072	36.09596
zstd-vbe21-zd	2.928103	0.65848199	5.464336	36.09110
zstd-svb16-zd	2.928344	0.65851007	5.463887	36.08814
zstd-vbbe21-zd	2.928413	0.65851816	5.463758	36.08728
zstd-svb-zd	2.928430	0.65852009	5.463727	36.08708
rc0-vbe21-zd	2.930661	0.65878001	5.459568	36.05961
rc0-vbbe21-zd	2.931079	0.65882867	5.458789	36.05447
rc1-vbe21-zd	2.947403	0.66071828	5.428555	35.85477
shuff-vbe21-zd	2.947726	0.66075550	5.427960	35.85084
rc1-vbbe21-zd	2.947826	0.66076694	5.427777	35.84963
shuff-vbbe21-zd	2.948147	0.66080385	5.427186	35.84573
rc01s-svb-zd	2.990472	0.66560461	5.350373	35.33840
rc01s-svb16-zd	2.990579	0.66561660	5.350182	35.33713
rc01s-vbe21-zd	2.990877	0.66564996	5.349648	35.33360
stall-fz	2.991124	0.66567752	5.349207	35.33069
rc01s-vbbe21-zd	2.991313	0.66569862	5.348869	35.32846
dstall-fz-1500	2.991704	0.66574236	5.348169	35.32384
dstall-fz	2.991729	0.66574516	5.348124	35.32354

Results



dstall–fz
bzip2
rc01s–svb–zd
bzip2–svb16–zd
dstall–fz–1500
stall–fz
rc01s–svb16–zd
rc01s–vbbe21–zd
rc01s–vbe21–zd
huff–vbbe21–zd
zlib–svb0–zd
huff–vbe21–zd
zlib–vbbe21–zd
shuff–vbbe21–zd
zlib–svb–zd
zlib–vbe21–zd
shuff–vbe21–zd
zlib–svb16–zd
rc1–vbbe21–zd
rc1–vbe21–zd
rc0–vbe21–zd
flac
rc0–vbbe21–zd
zstd–vbbe21–zd
zstd–vbe21–zd
zstd–svb0–zd
zstd–svb–zd
zstd–svb16–zd
Compression Time (hr / TiB)
0
10
20
30
Decompression Time (hr / TiB)
5
10
15
Space Saving
0.66
0.65
0.64
0.63

Discussion

Objective Complete

- vbbe21 $>$ svb
- Range coding exceeds entropy
- Space saving improvement: 0.72%

Contributions

1. First systematic analysis
2. New state-of-the-art
3. First benchmark

Future Work

Even better?

- Multithreading
- Multi-read compression
- Other data: RNA, non-human

Conclusion

Recommend dstall-fz-1500
for archiving

Superior space saving
66.6% vs 65.9%

Questions?