UNIVERSITY OF PADUA

DEPARTMENT OF INFORMATION ENGINEERING

Fast and Succinct Compression of k-mer Sets with Plain Text Representation of Colored de Bruijn Graphs

Enrico Rossignolo Matteo Comin

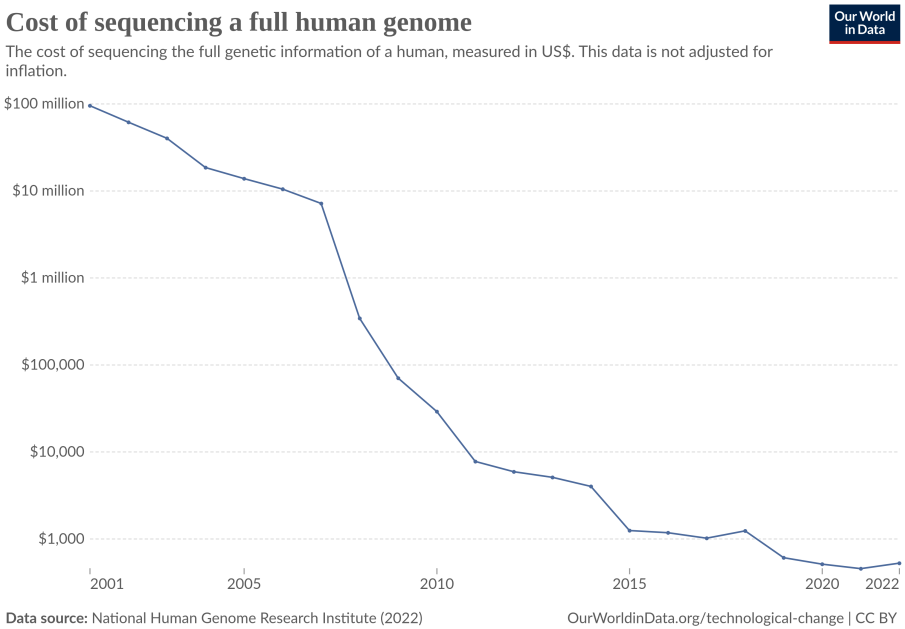
ICCABS 2025

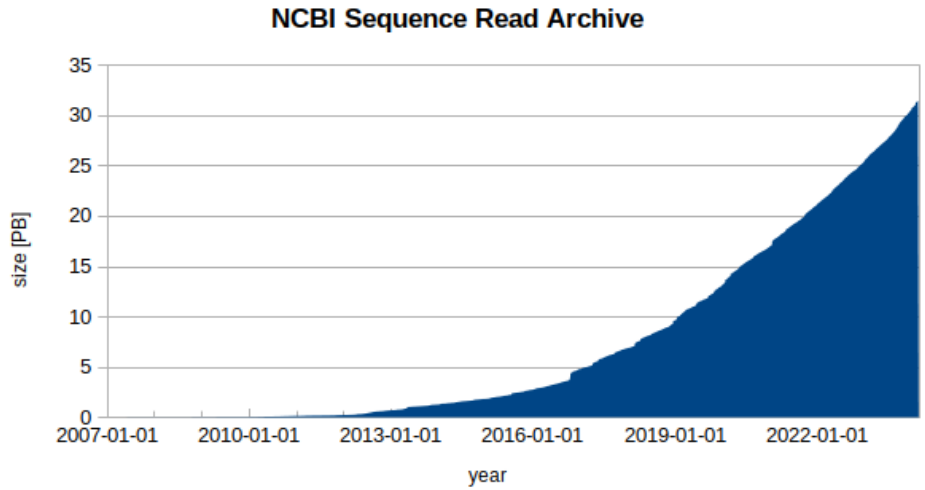
Outline

- Introduction
  - k-mers and colors
  - Applications
  - Problems
- Methods
  - Colored maximal unitigs
  - Colored greedy matchtigs
  - Our method
- Experimental setup
- Results
  - Compression
  - Performance
- Conclusions

Motivation

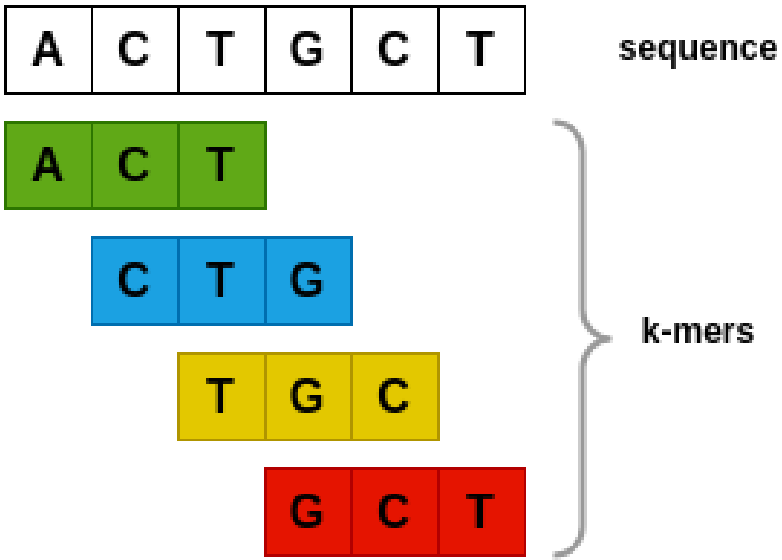
- Cost per genome drops $\Longrightarrow$ **a lot of data** (SRA: >30 PB)
- Huge datasets need medium to long term storage for analysis
- We need efficient disk compression that allows fast loading of data



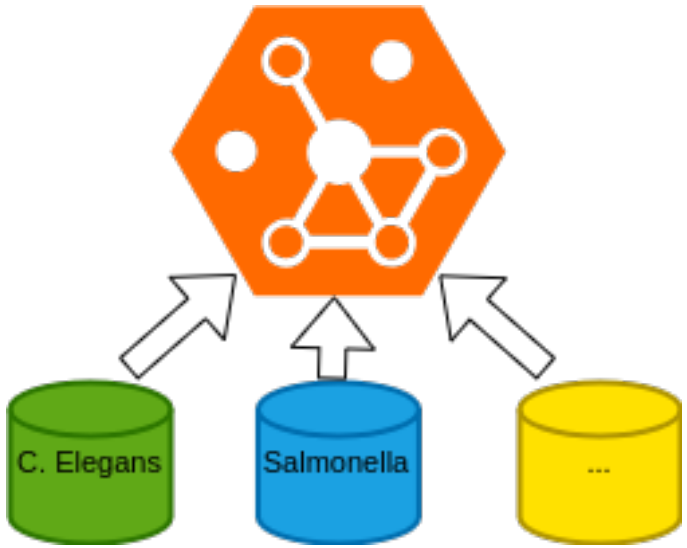


k-mers and colors

- DNA sequence: string with alphabet A,C,T,G
- k-mers: DNA **substrings of length *k***

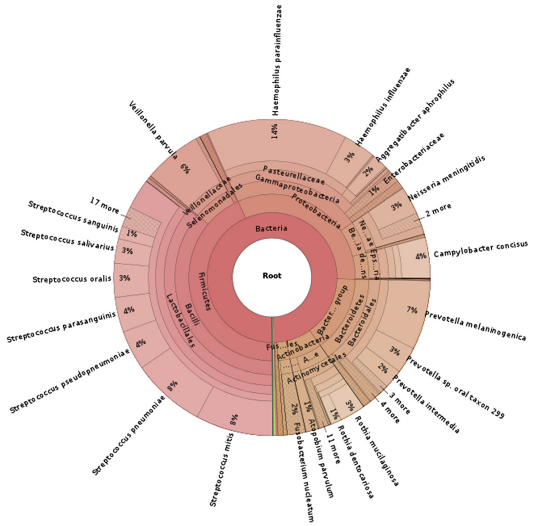


- colors: k-mers tags
  - express its origin
- set of colored *k*-mers aka set of *k*-mer sets



Applications based on k-mers

- Assembly: Eulerian walks in graph of k-mers that can be found **efficiently** (Spades)
- Phylogenetics: Mash[1] creates trees using k-mers
- Database searching (BIGSI)
- Metagenomics: Kraken[2]
  - Kraken need to store large k-mers tables ($k = 31$)
  - 900 times faster than tools based on alignment



taxonomic unit XYZ
ATCGATCGATCGATCGATCGATCGATCGA
TCGATCGATCGATCGATCGATCGATCGAT
CGATCGATCGATCGATCGATCGATCGATC
GATCGATCGATCGATCGATCGATCGATCG
GATCGATCGATCGATCGATCGATCGATCG
ATCGATCGATCGATCGATCGATCGATCGA
TCGATCGATCGATCGATCGATCGATCGAT
...
CGATCGATCGATCGATCGATCGATCGATC

[1] *Ondov, B.D., Treangen, T.J., Melsted, P. et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol 17, 132 (2016)*
[2] *Wood, D.E., Salzberg, S.L. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol 15, R46 (2014)*

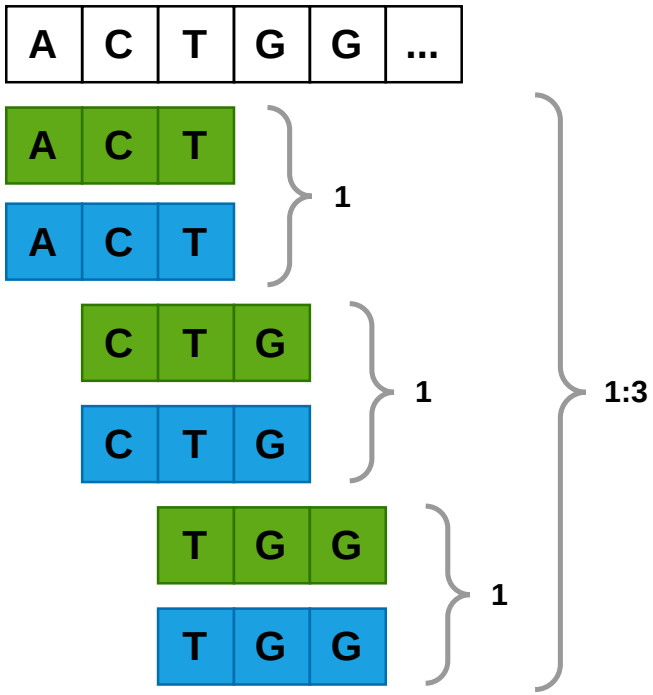
Colored k-mers sets compression

- Sets can be individually compressed using existing k-mer compression tools
  - Matchtigs[1]: compute minimum size representation of a k-mer set
  - USTAR2[2]: efficiently compress k-mers
- If different colors share k-mers then individual compression is not optimal
- Colored k-mer compression tools already exist
  - Bifrost[3]: clever use of Bloom filters to build the graph and bitmaps to store set of colors
  - GGCAT[4]: uses colored de Bruijn Graph, improves colors compression; state-of-the-art tool

[1] *Schmidt, et al. "Matchtigs: minimum plain text representation of k-mer sets." Genome Biology 24.1 (2023)*
[2] *Rossignolo, and Comin. "USTAR2: Fast and Succinct Representation of k-mer Sets Using De Bruijn Graphs." BIOSTEC (1). 2024.*
[3] *Holley, and Melsted. "Bifrost: highly parallel construction and indexing of colored and compacted de Bruijn graphs." Genome biology 21 (2020)*
[4] *Cracco, and Tomescu. "Extremely fast construction and querying of compacted and colored de Bruijn graphs with GGCAT." Genome Research 33.7 (2023)*

GGCAT (maximal unitigs)

- GGCAT associates each k-mer to a set of colors represented by an index
- For each sequence, the index list is compressed with Run Length Encoding
- GGCAT computes maximal unitigs from wich we can extract k-mers



GGCAT (greedy matchtigs)

- GGCAT can optionally compute greedy matchtigs with colors
  - unitigs are merged saving $k-1$ characters per connection

- Matchtigs are a special k-mers set representation
  - May contain repeated k-mers
  - Minimum size
  - CPU and memory intensive

    $\Longrightarrow$ computed with a greedy algorithm

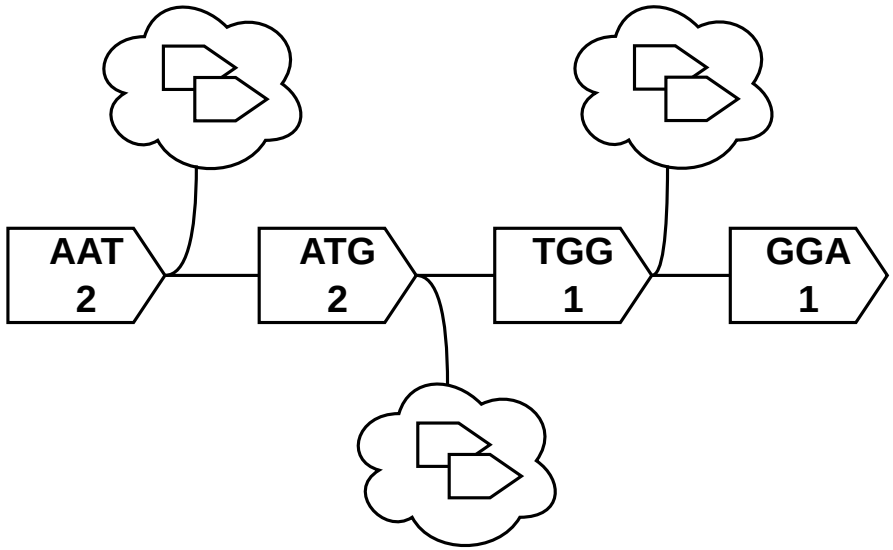
Our methods

- USTAR-C (USTAR Colors)
  - USTAR2 with colors
  - It uses the colored de Bruijn graph built by GGCAT
  - It exploit the graph connectivity to build longer contig

- USTAR-CR (USTAR Colors Reordering)
  - Rearrange colors to improve compression

USTAR Colors

- USTAR-C uses the USTAR2's algorithm for choosing path in a de Bruijn graph
  - Unitigs are merged based on the degree of the nodes, with preference given to less connected ones



- Colors are accumulated and compressed using Run Length Encoding

USTAR Colors

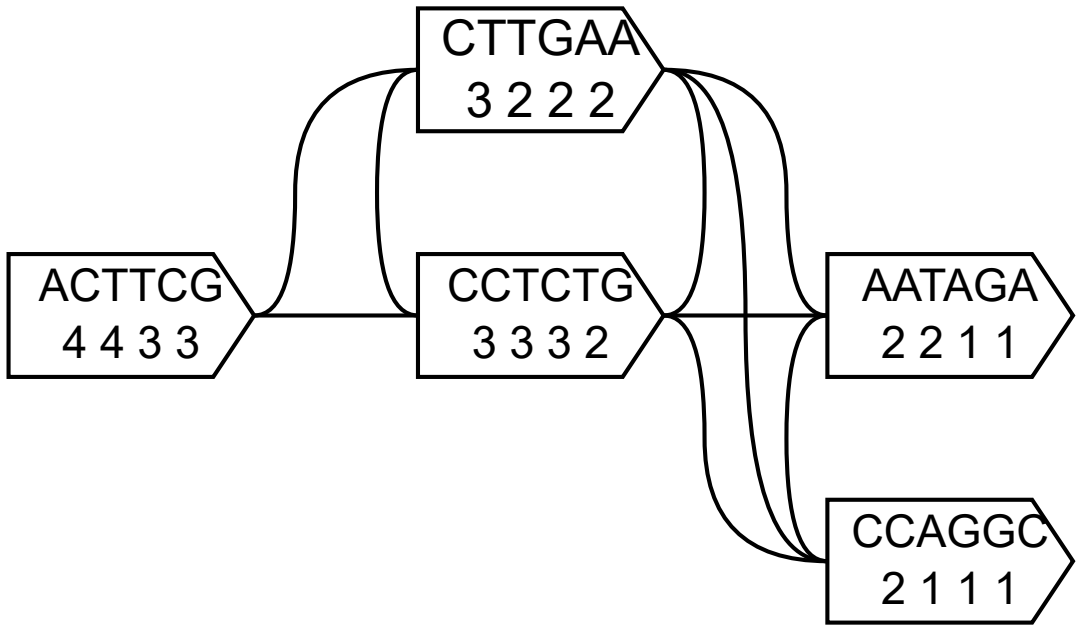
sequences	AATGGA	ACTTCG	CCAGGC	CCTCTG	CTTGAA
colors set indices (plain)	2 2 1 1	4 4 3 3	2 1 1 1	3 3 3 2	3 2 2 2
colors set indices (RLE)	2-2 1-2	4-2 3-2	2-1 1-3	3-3 2-1	3-1 2-3

- Note that some sequences share the same colors
- We want longer runs to exploit RLE
- Need to rearrange colors (and associated sequences) so that end-points can be connected
- An optimal algorithm already exists[1]
- Our implementation is fast and works well in practice

[1] *Pibiri, On weighted k-mer dictionaries. Algorithms for Molecular Biology 18(1), 3 (2023)*

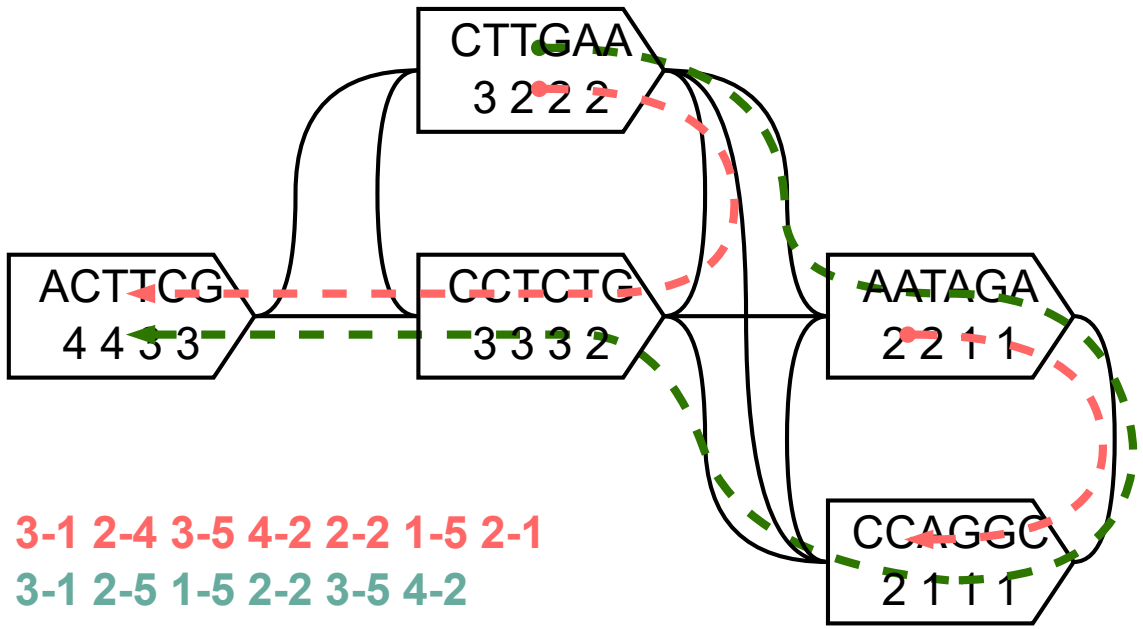
USTAR Color Reordering

- End-point Weight Graph (ewG)
  - nodes are oriented and labelled with sequences and colors
  - arcs exist between nodes that shares and end-point color



USTAR Color Reordering

- Objective: make long path minimizing isolated nodes
- $\Longrightarrow$ use node connectivity to make choices



Experimental setup

- 20 dataset taken from major papers on k-mer compression ranging from 10 to 500 millions of k-mers
- each dataset represents one different color

- Given the k-mer set S, we used the following metrics:
  - Cumulative Length, $CL = \sum_{s \in S} |s|$
  - Sequence Count, SC: total number of sequences
  - $\#$*color runs*: total number of color runs
  - *total compression*: size of compressed colored $k - mers$

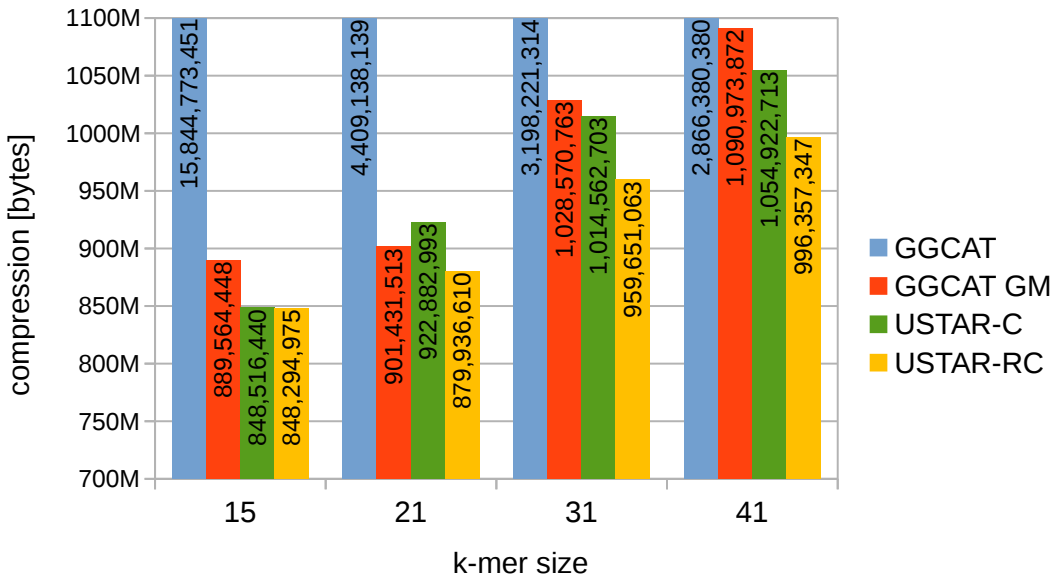
- Tools to compare:
  - GGCAT (maximal unitigs)
  - GGCAT GM (greedy matchtigs)
  - USTAR-C
  - USTAR-CR

Results for k=31

k=31	GGCAT	GGCAT GM	USTAR-C	USTAR-CR
CL	6,266,509,634	**3,290,519,704**	3,681,600,490	3,681,600,490
SC	135,191,765	**41,341,022**	43,520,096	43,520,096
#color runs	468,952,986	78,730,727	92,771,770	**54,026,538**
total compression	3,198,221,314	1,028,570,763	1,014,562,703	**959,651,063**

- As expected, plain GGCAT does not compress
- GGCAT GM produces the best CL and SC
- USTAR-CR improves #color runs of USTAR-C resulting the best value
- Both version of GGCAT compress sequences and colors together
- USTAR-CR produces the smallest representation

Results: testing different k



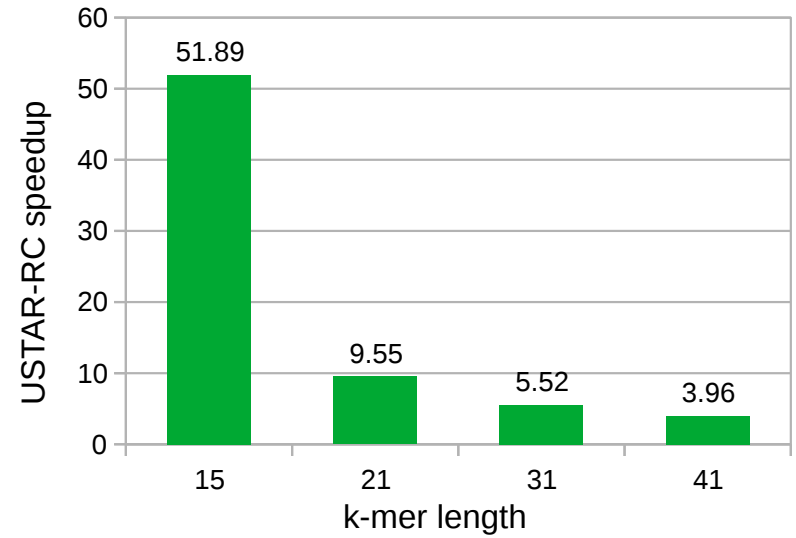
- We vary the k-mer length and observe the total compression
- For all the k-mer sizes, USTAR-CR obtains the smaller compression
- A special case is $k = 15$, where color reordering seems to have very little effects on the final compression

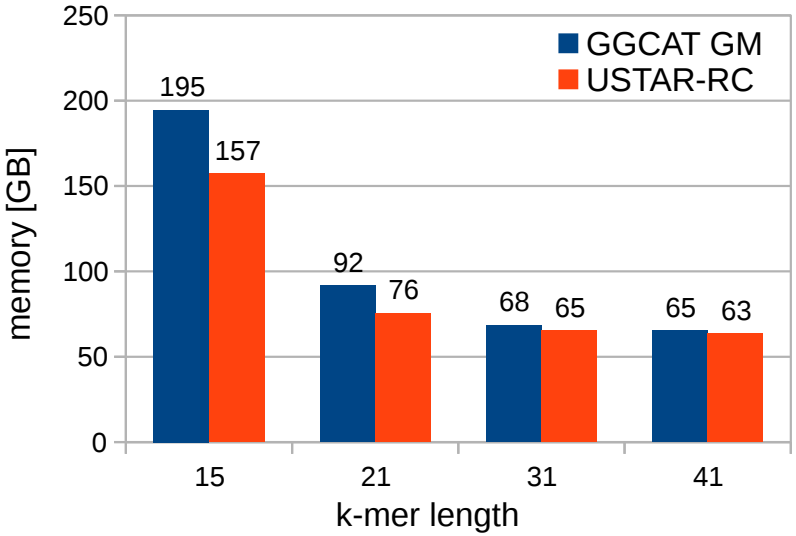
Results: different number of colors

k=31	c=10		c=15		c=20	
	GGCAT GM	USTAR-CR	GGCAT GM	USTAR-CR	GGCAT GM	USTAR-CR
#color runs	36,587,632	26,418,625	40,832,766	29,116,036	78,730,727	54,026,538
total compression	433,708,310	**410,570,603**	495,357,773	**464,890,012**	1,028,570,763	**959,651,063**

- Comparison between the two best tools
- USTAR-CR consistently achieved the smallest *#color runs* and *total compression*

Results: time and memory





- GGCAT GM is slower for all k-mer lengths
- For $k = 15$, USTAR-CR is **51.89**× faster than GGCAT GM
- For $k = 31$, USTAR-CR is still **5.52**× faster while using about the same amount of memory

Conclusions

- Introduction of USTAR-CR, a tool for compressing a colored k-mers sets using a de Bruijn graph that optimizes color compression
- USTAR-CR achieved **compression ratios** surpass the state-of-the-art tool GGCAT GM
- USTAR-CR execution time is **faster** than its counterpart, up to 51× faster for small k-mer lengths, and requires **less memory**
- This tool offers an effective and resource-efficient solution for compressing colored k-mer sets, with potential for further performance enhancement through parallelization.

The End

Thanks for your attention!

Datasets

Dataset	Description	Read Length	#Reads	Size [GB]
SRR001665	Escherichia coli	36	20,816,448	9.304
SRR061958	Human Microbiome 1	101	53,588,068	3.007
SRR062379	Human Microbiome 2	100	64,491,564	2.348
SRR10260779	Musa balbisiana RNA-Seq	101	44,227,112	2.363
SRR11458718	Soybean RNA-seq	125	83,594,116	3.565
SRR13605073	Broiler chicken DNA	92	14,763,228	0.230
SRR14005143	Foodborne pathogens	211	1,713,786	0.261
SRR332538	Drosophila ananassae	75	18,365,926	0.683
SRR341725	Gut microbiota	90	25,479,128	1.254
SRR5853087	Danio rerio RNA-Seq	101	119,482,078	3.194
SRR957915	Human RNA-seq	101	49,459,840	3.671

USTAR2 pseudocode

Algorithm 1: USTAR2

```
Data: de Bruijn graph dBG
Result: SPSS S
begin
    S = ∅
    seed-nodes = sort nodes by Imb(node)
    for seed ∈ seed-nodes do
        if seed is not visited then
            visit(seed)
            contig = Extend(seed) to the right
            contig = Extend(contig) to the left
            S = S ∪ {contig}
    return S
Function Extend(contig):
    L = {non-visited neighbors of contig head}
    while L not empty do
        v = less connected node in L
        visit(v)
        contig = merge(v, contig)
        L = {non-visited neighbors of v}
    L = {neighbors of contig head}
    level = 1
    found new node = false;
    while level<=D and not found new node do
        L = {neighbors of all nodes in L}
        level=level+1
        L = Filter(L)
        L' = {non-visited nodes in L}
        if L' not empty then
            k = less connected node in L'
            visit(k)
            found new node = true;
            p = path from k to contig head
            contig = merge(p, contig)
    if found new node then
        return Extend(contig)
    else
        return contig
```

```
Function Filter(L):
    for v ∈ L do
        p = path from v to contig head
        if length(p) > 2k - 2 then
            remove v from L
    return L
```

UNIVERSITY OF PADUA

DEPARTMENT OF INFORMATION ENGINEERING

Fast and Succinct Compression of k-mer Sets with Plain Text Representation of Colored de Bruijn Graphs

Enrico Rossignolo Matteo Comin

ICCABS 2025