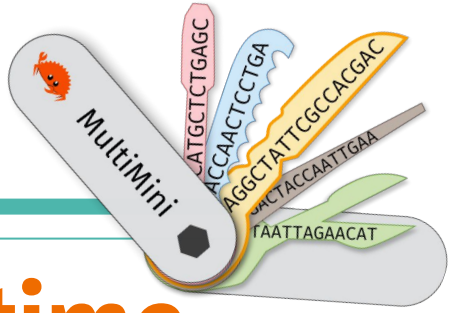


Multiminimizers: trading time for space in sampling schemes

Florian Ingels, Lucas Robidou, Igor Martayan,
Camille Marchet & Antoine Limasset

CRIStAL
Centre de Recherche en Informatique, Signal et Automatique de Lille

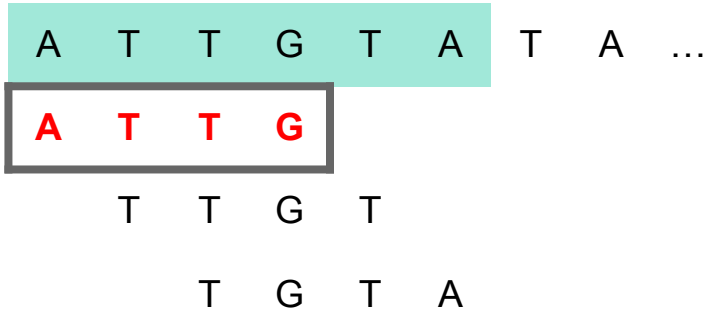
Thessaloniki | 2026
RECOMB

cnrs

Definition: local schemes, minimizers and sampling

A T T G T A T A …

Definition: local schemes, minimizers and sampling



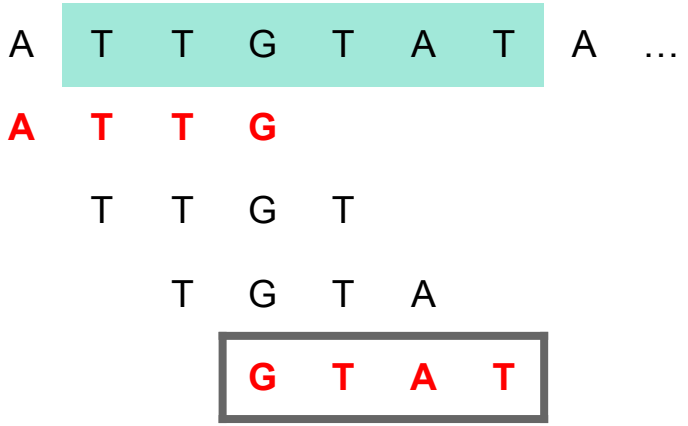
Local schemes
in each *k*-mer (word of length *k*), choose one m-mer

Minimizers
choose the smallest one (for some order, e.g. random)

k=6, *m*=4, *w*=3

window size ***w*** = *k*-*m*+1
of *m*-mers / *k*-mer

Definition: local schemes, minimizers and sampling



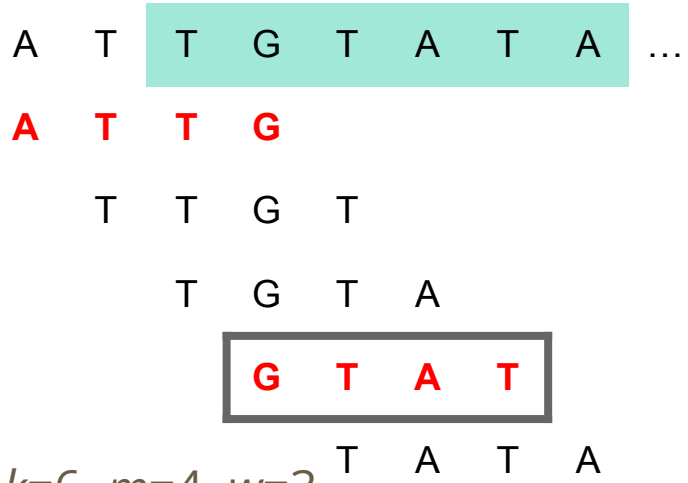
k=6, *m*=4, *w*=3

window size ***w*** = *k*-*m*+1
of *m*-mers / *k*-mer

Local schemes
in each *k*-mer (word of length *k*), choose one m-mer

Minimizers
choose the smallest one (for some order, e.g. random)

Definition: local schemes, minimizers and sampling



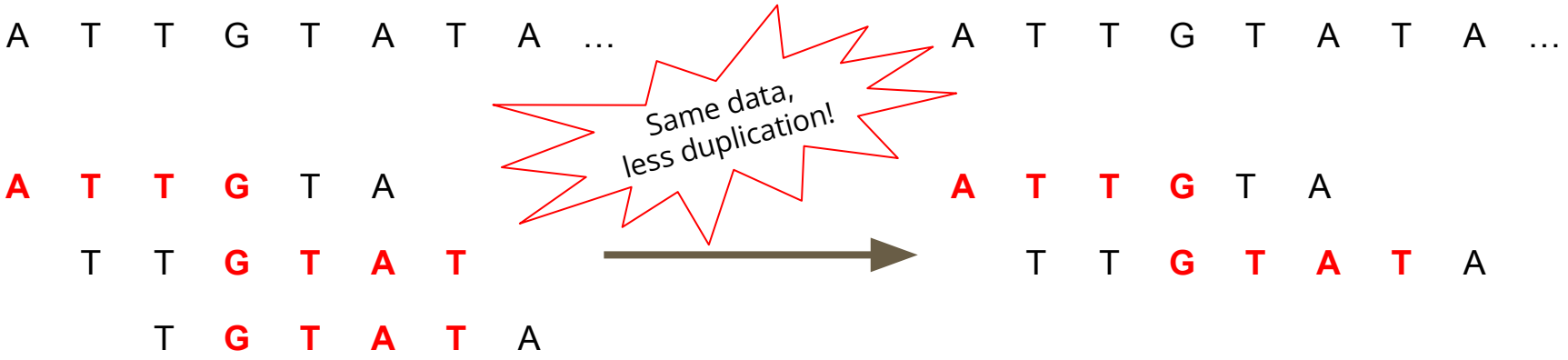
k=6, *m*=4, *w*=3

window size ***w*** = *k*-*m*+1
of *m*-mers / *k*-mer

Local schemes
in each *k*-mer (word of length *k*), choose one m-mer

Minimizers
choose the smallest one (for some order, e.g. random)

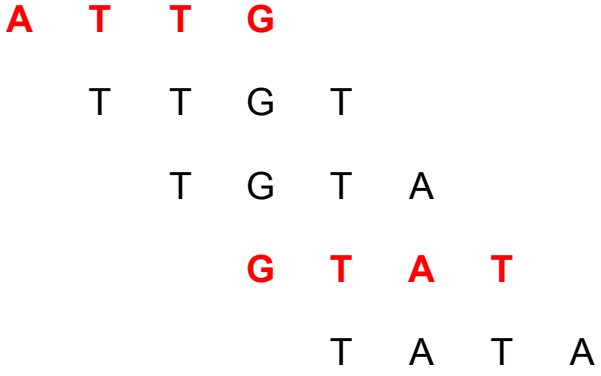
Definition: super-*k*-mer



Definition: density

The density is the proportion of selected positions in a random string *S* of infinite size.

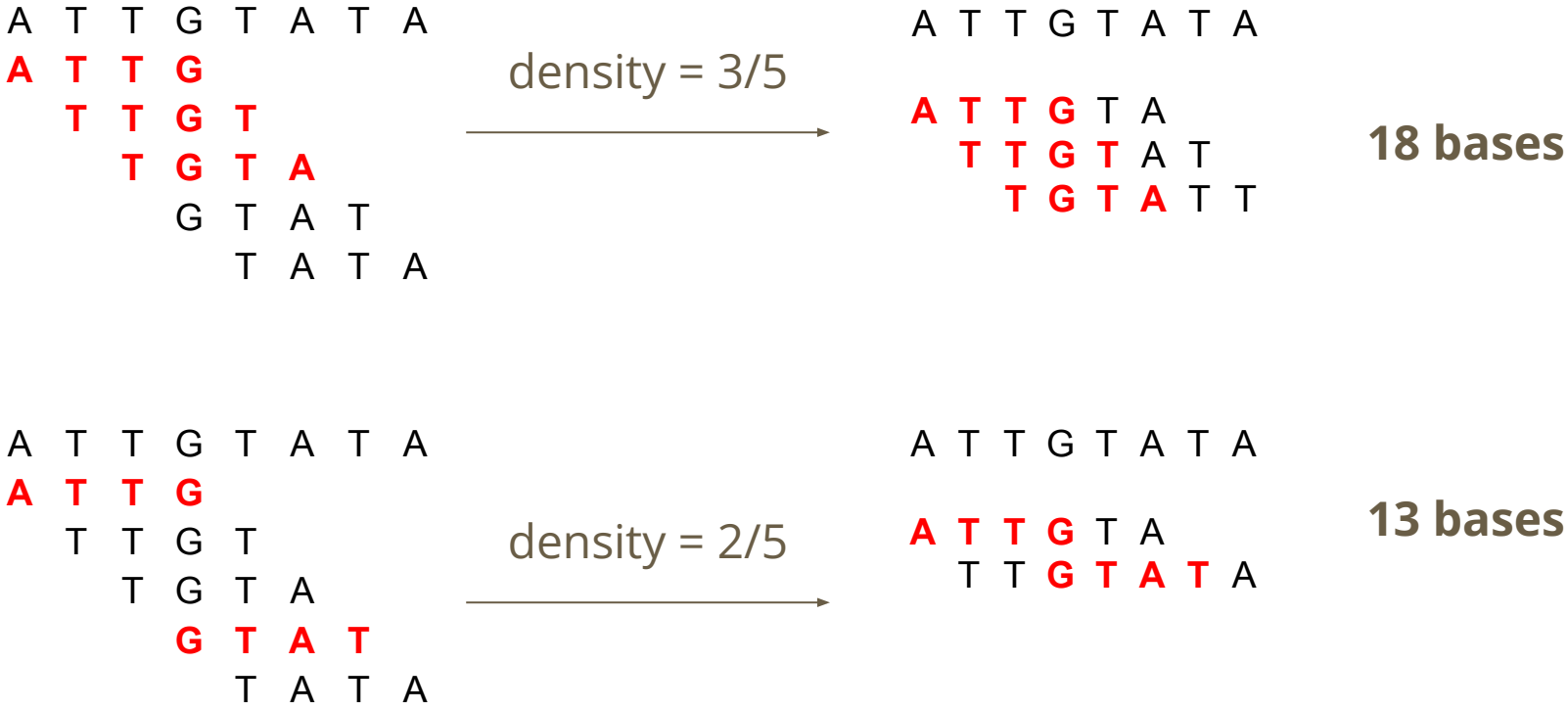
The specific density is defined for a given string.



specific density

$$\frac{2 \text{ selected positions}}{5\ m\text{-mers}} = \frac{2}{5}$$

The density drives the cost of a scheme



Theoretical state-of-the-art on local schemes

We have lower bounds: $$d \geq \frac{1}{\sigma^{w+k}} \sum_{p|(w+k)} M_\sigma(p) \left\lceil \frac{p}{w} \right\rceil \geq \frac{1}{w}$$

Best known bound
[Kille and Groot Koerkamp et al., 2024]

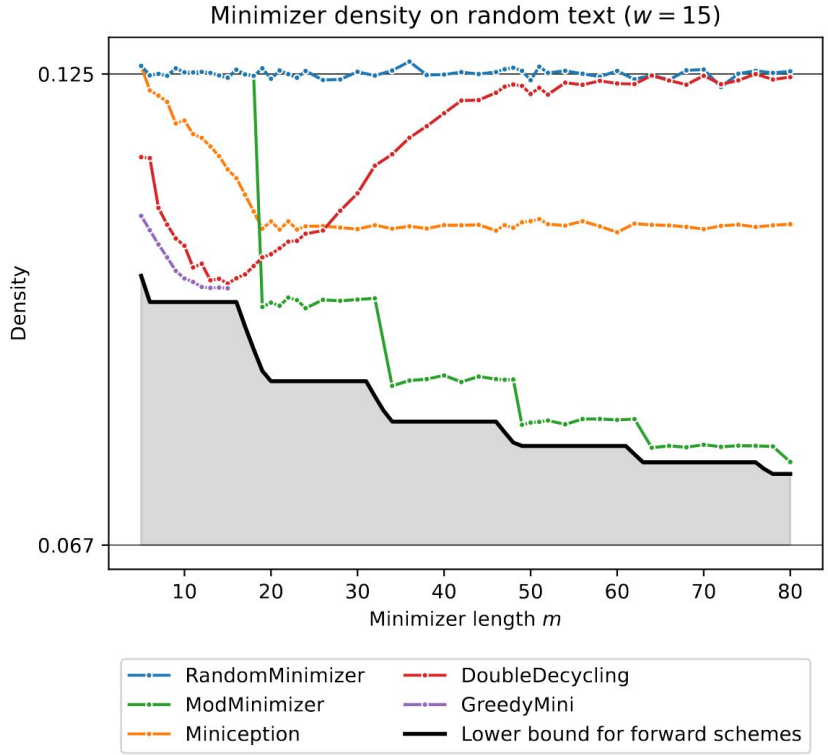
Trivial lower bound

We only know the theoretical density of **random minimizers:** $\frac{2}{w+1}$

[Schleimer et al., 2003]
[Zheng et al., 2020]

Existing local schemes are close to their lower bound

We are likely to face diminishing returns.



Contribution 1: density ↔ expected gap

Let ***d*** be the density of the selection scheme.

Let $\mu > 0$ be the expected distance between selected positions.

Then, $d \cdot \mu = 1$

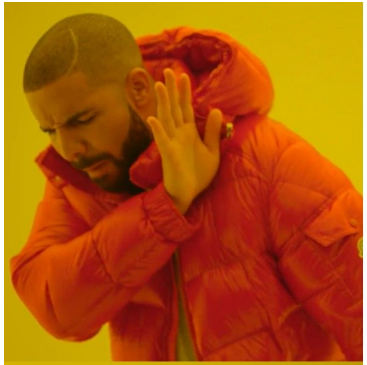
- **density** and **expected gap between selected positions** are equivalent concepts
- new insight : to decrease the density, **increase the expected gap**

Contribution 2: multiminimizers, a new framework

Goal: maximizing the gap between consecutive selected positions.

Key idea:

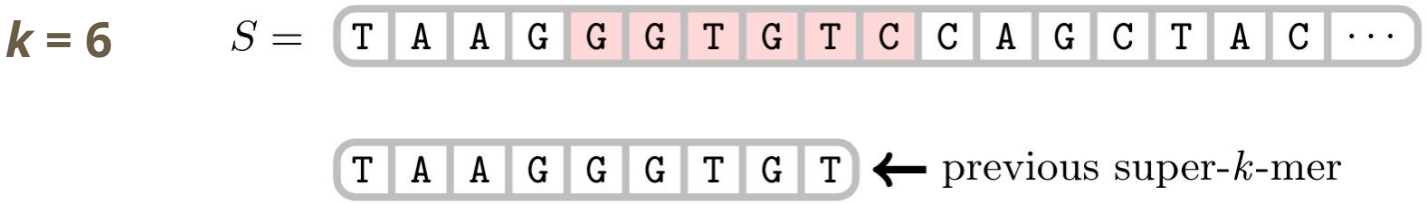
- we generate **multiple** minimizer candidates
- we select the candidate generating the super-*k*-mer ending the furthest in the read



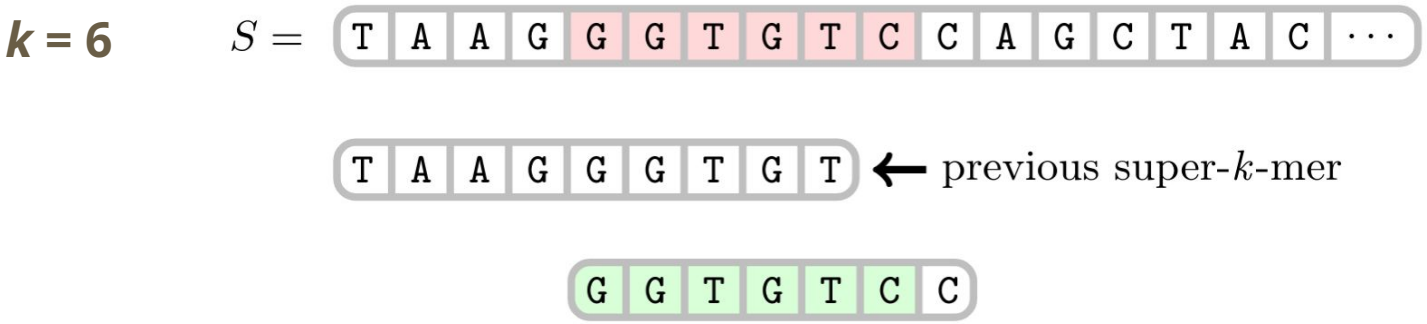
Proposing a new scheme

Improving all the existing schemes at once

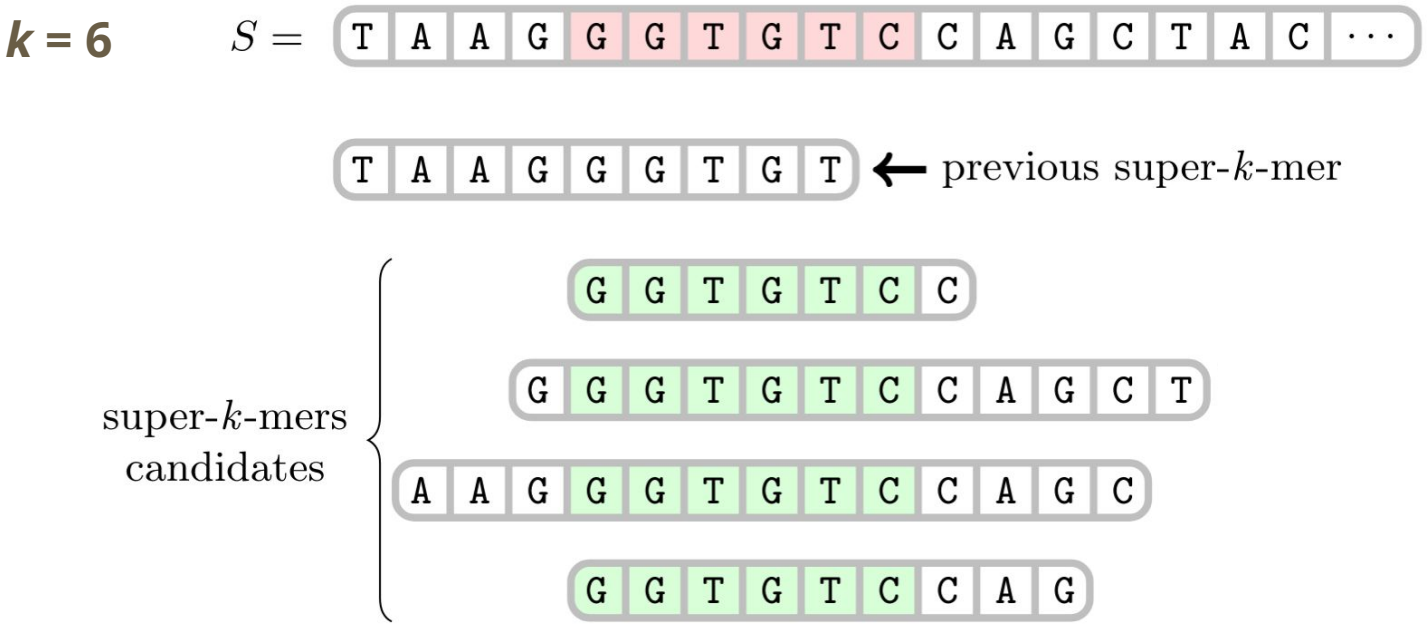
Contribution 2: multiminimizers, a new framework



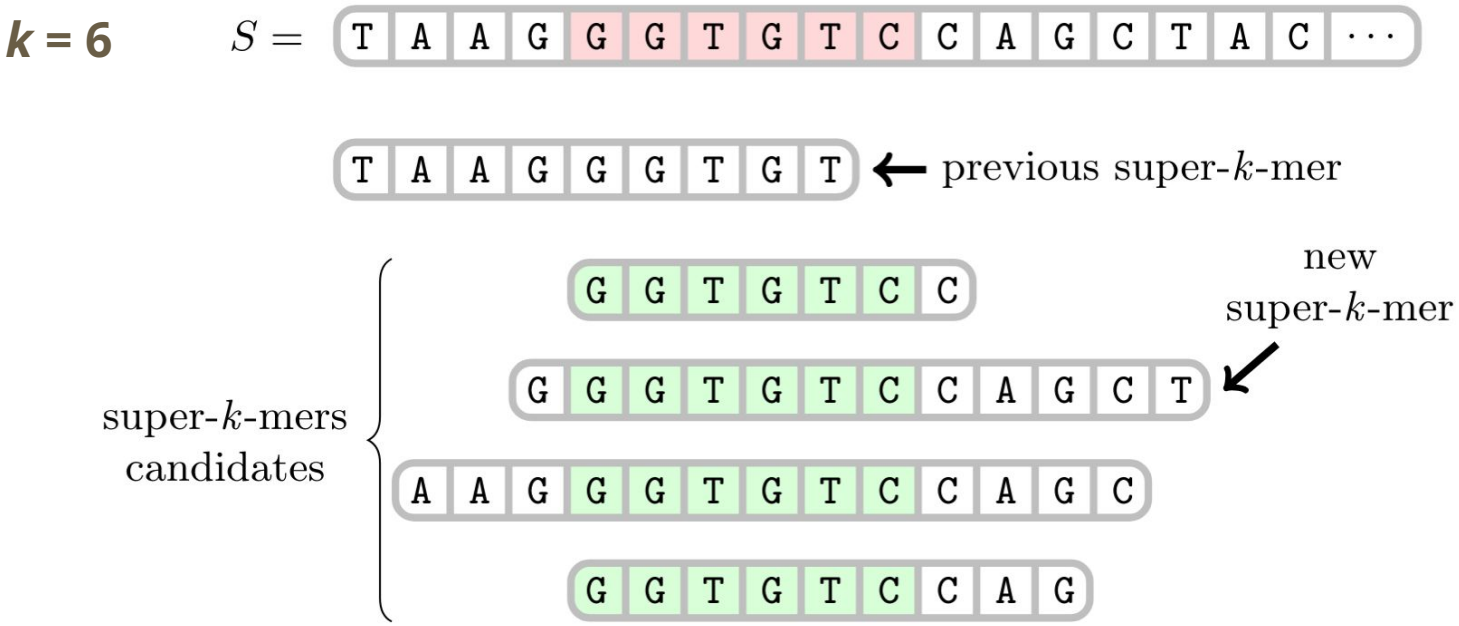
Contribution 2: multiminimizers, a new framework



Contribution 2: multiminimizers, a new framework



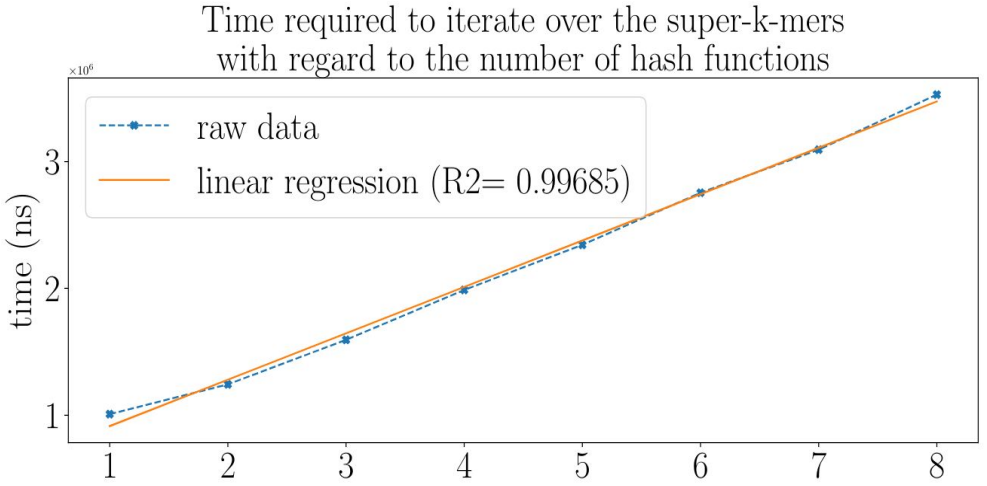
Contribution 2: multiminimizers, a new framework



Contribution 2: Results (speed)

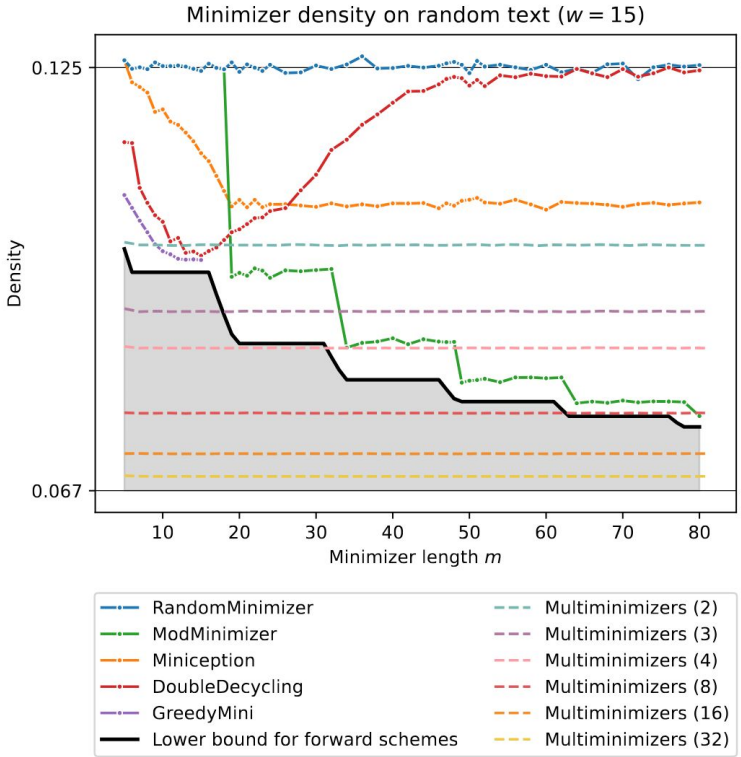
Our method needs to compute *N* distinct schemes.

This leads to a linear increase of computation time.



Contribution 2: Results (density)

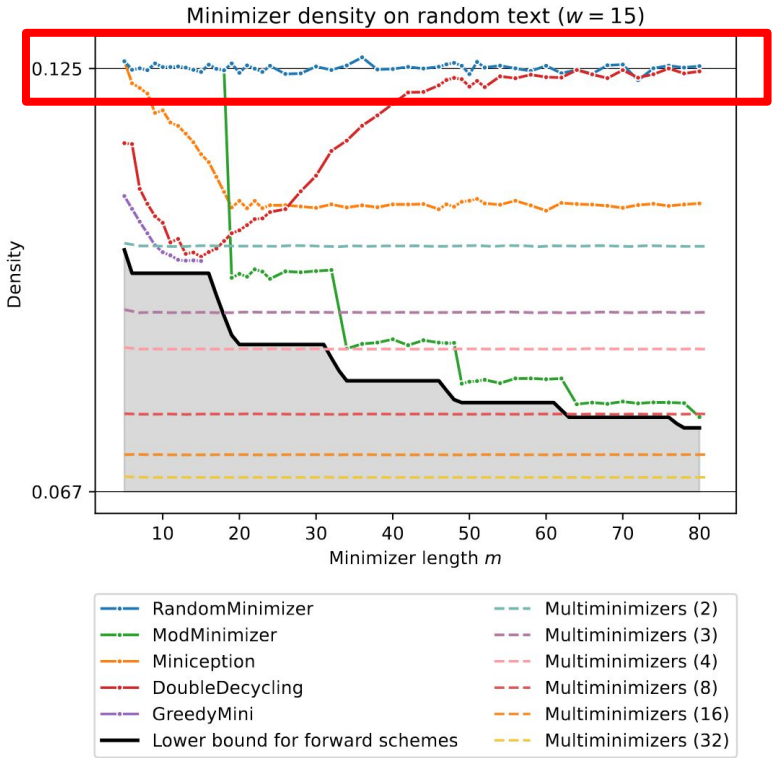
Using random minimizers, our approach converges to densities smaller than **any other scheme**.



Contribution 2: Results (density)

Using random minimizers, our approach converges to densities smaller than **any other scheme**.

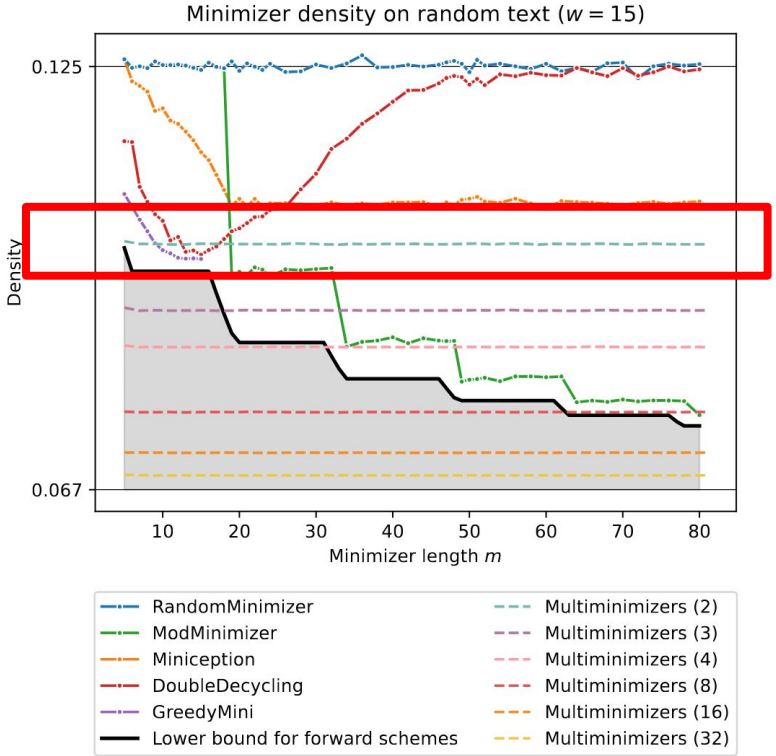
1 hash



Contribution 2: Results (density)

Using random minimizers, our approach converges to densities smaller than **any other scheme**.

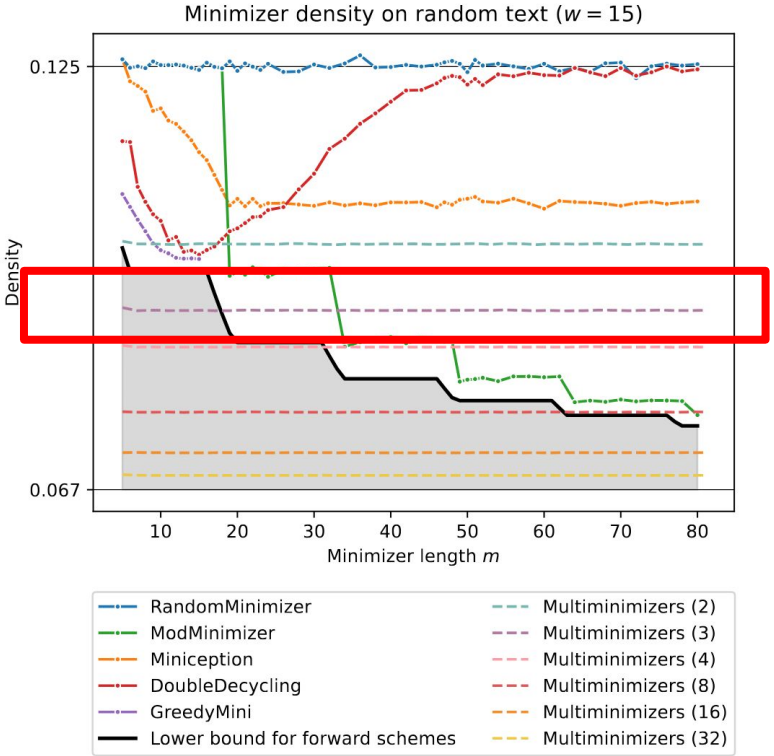
2 hashes



Contribution 2: Results (density)

Using random minimizers, our approach converges to densities smaller than **any other scheme**.

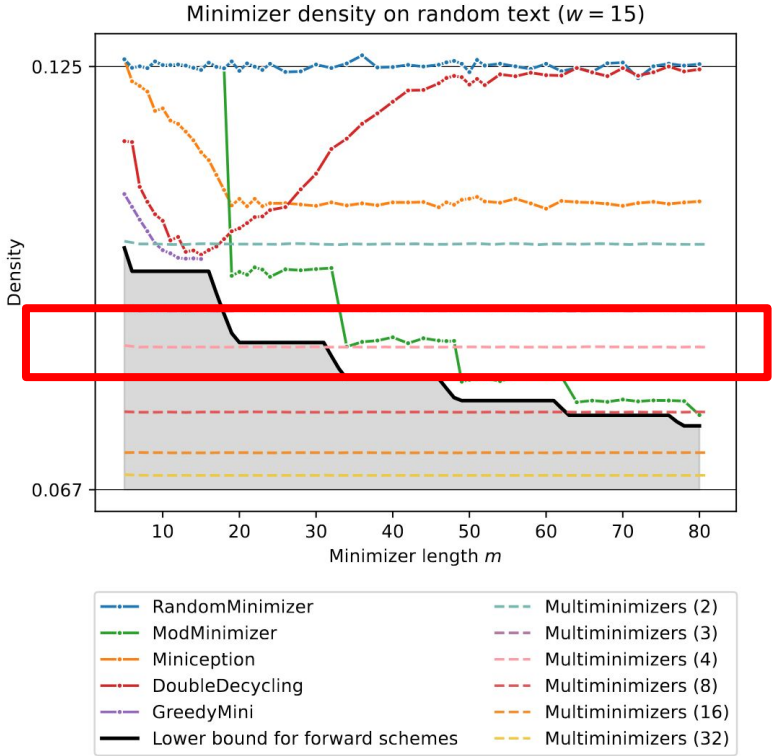
3 hashes



Contribution 2: Results (density)

Using random minimizers, our approach converges to densities smaller than **any other scheme**.

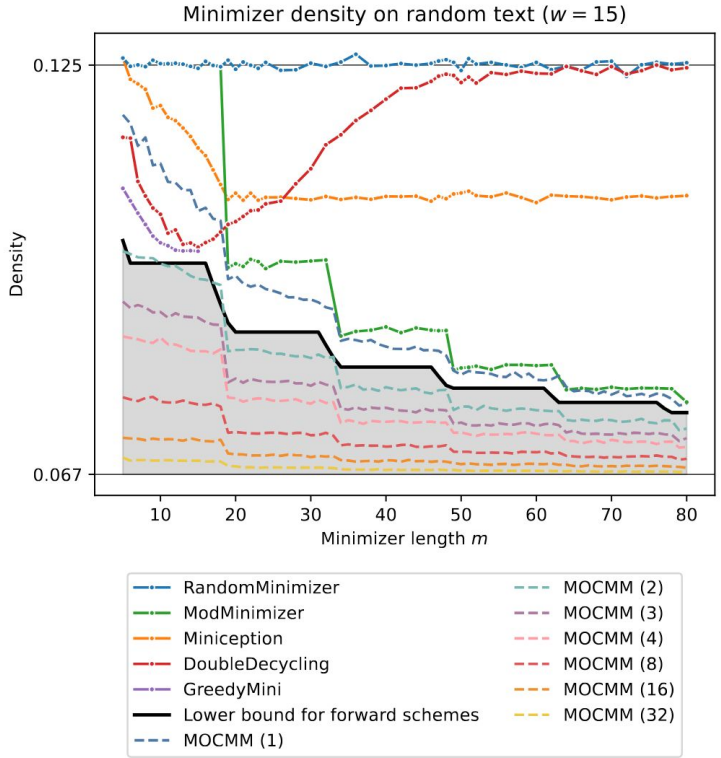
4 hashes



Contribution 2: Multi Open-Closed Mod-Minimizers

Our approach gets even better if the underlying scheme gets better.

Here, we switch from random minimizers to open-closed mod-minimizers.

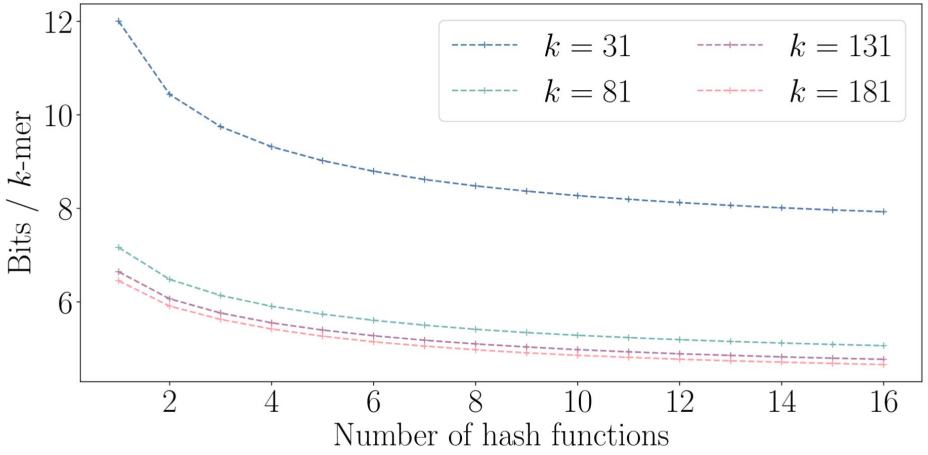


Contribution 2: Application to super-k-mers

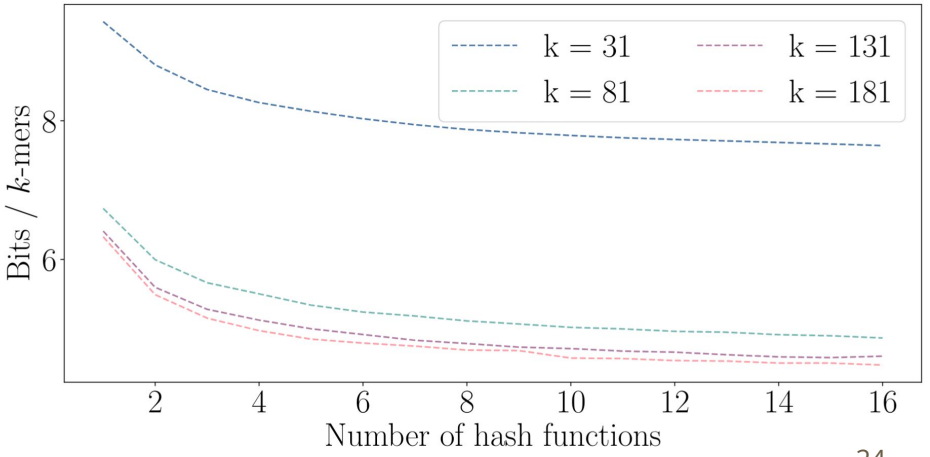
Space usage of super-*k*-mers:

- random minimizers: 6 bits / *k*-mer
- our multiminimizers: 4 bits / *k*-mer

Using random minimizers



Using open-closed mod-minimizers

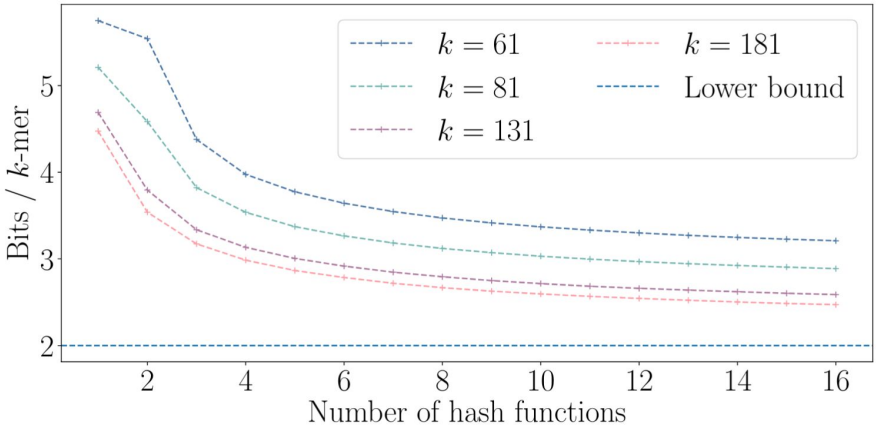


Contribution 2: Application to hyper-k-mers

Space usage of hyper-*k*-mers (2025):

- random minimizers: 4 bits / *k*-mer
- our multiminimizers: 2 bits / *k*-mer

This 2 bits / *k*-mer means ***we converge to the size of the sequence itself***.



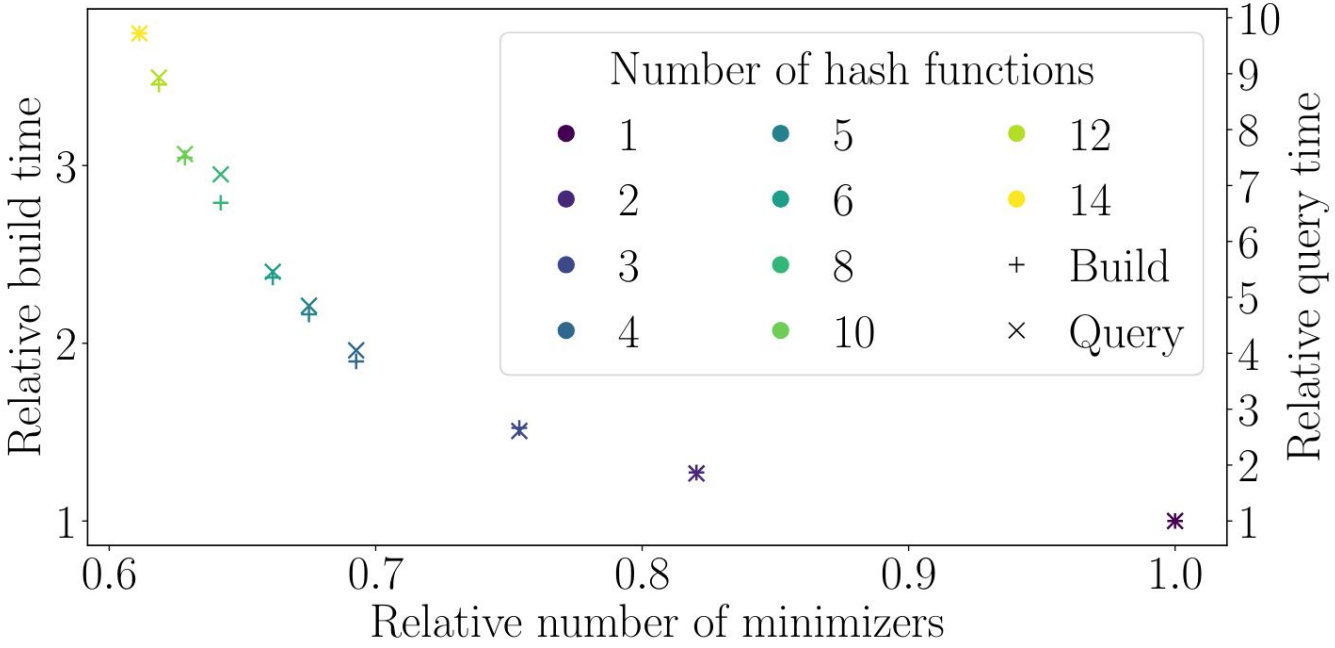
Contribution 3: Pin, a needle-like index

Needle ≈ index of minimizers (forgetting *k*-mers) to estimate quantifications of transcripts

Pin = index of minimizers using the multiminimizer trick

1. each *k*-mer has N minimizer candidates
2. we greedily select the minimizer that covers a maximum of *k*-mers
3. delete those *k*-mers and repeat

Contribution 3: Pin, a needle-like index

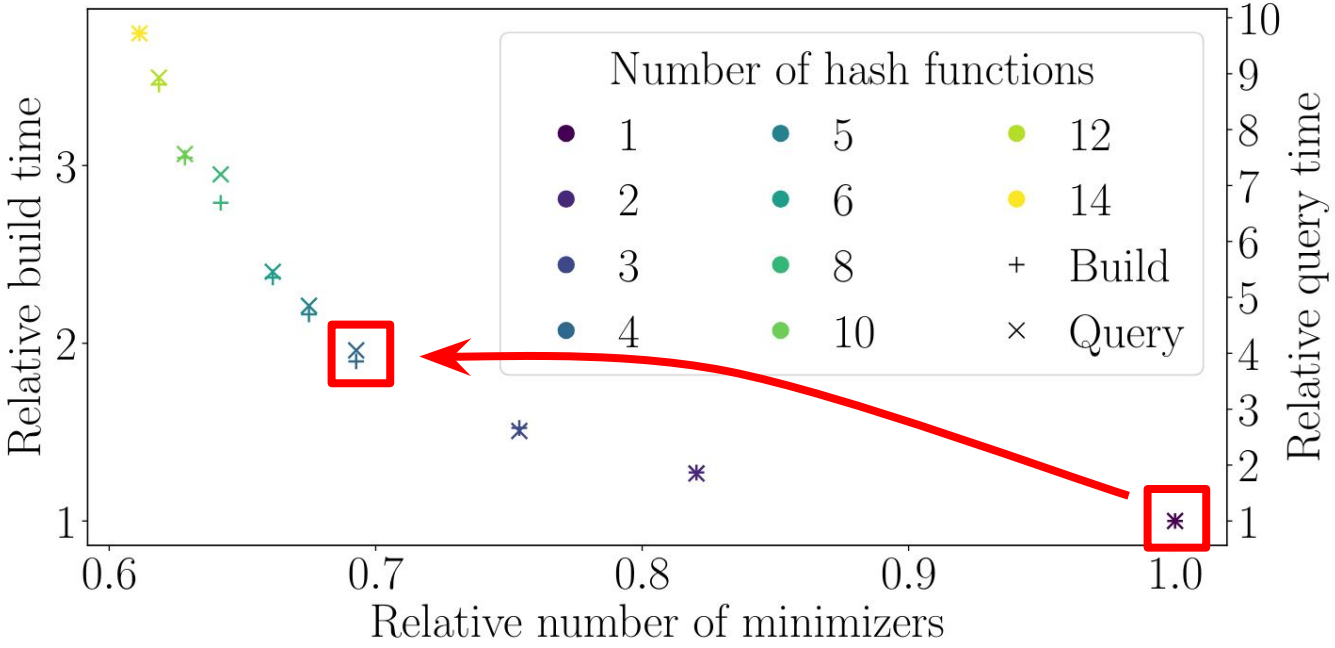


For **N=1**:

- 850 million minimizers
- index construction : 234s
- query time : 0.04s

Human HiFi accession SRR11292123 (~24Gb)

Contribution 3: Pin, a needle-like index



For **N=1**:

- 850 million minimizers
- index construction : 234s
- query time : 0.04s

Human HiFi accession SRR11292123 (~24Gb)

Future works

Theory

Model the density of the multiminizers as a function of:

- the number of hash functions
- the density of the underlying scheme

Practice

Strengthen our implementation:

- optimize supported schemes
- support more schemes

Apply our approach to more tools to reduce their space usage.

Thank you !

Preprint

doi.org/10.1101/2025.11.21.689688

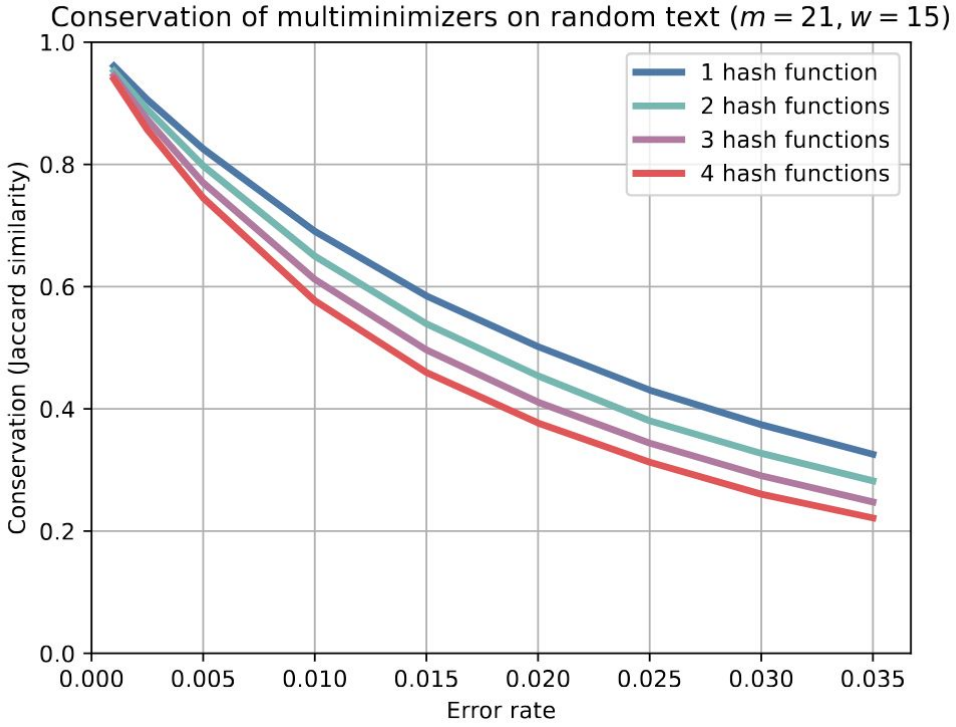
Code

github.com/lrobidou/multiminimizers

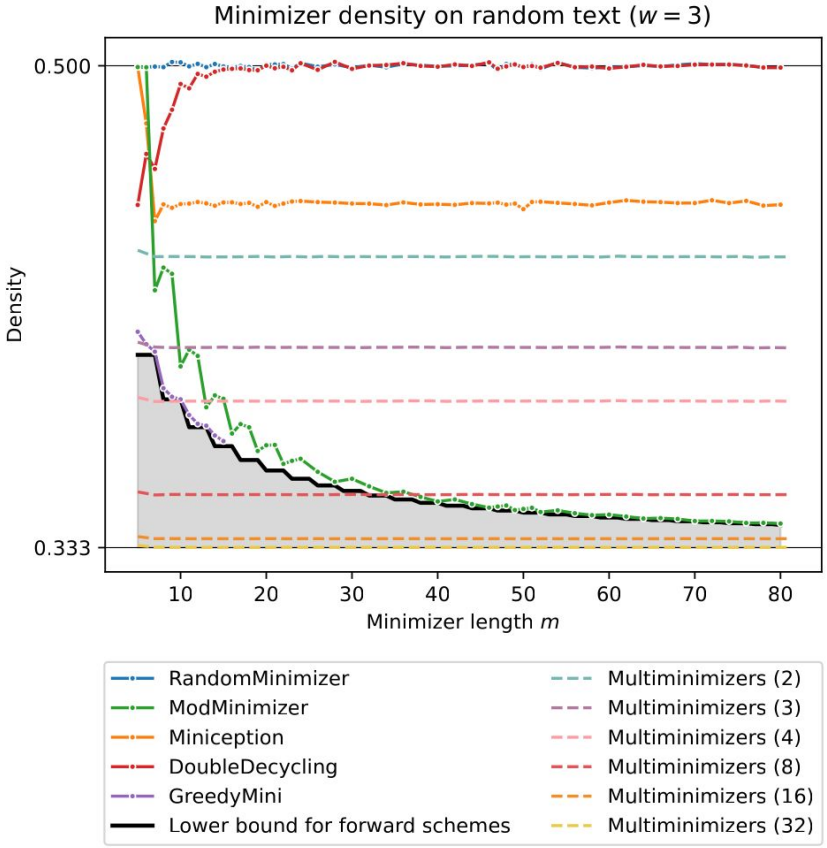
github.com/Malfoy/Pin

Come and get your sticker!

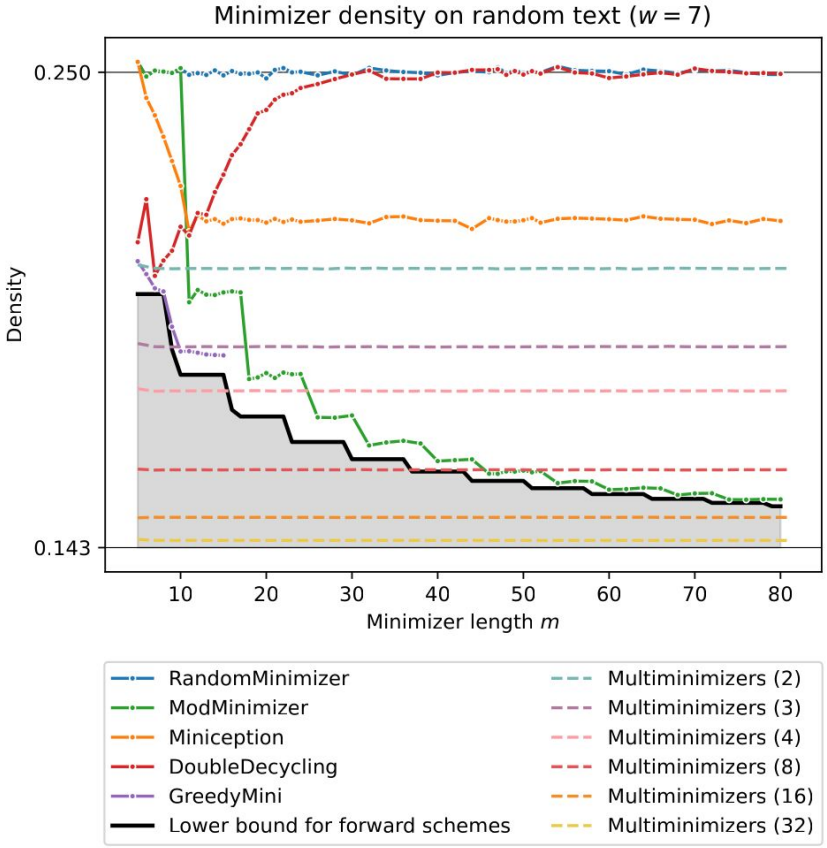
Appendix: Like minimizers, higher error rate reduces the conservation of multiminimizers



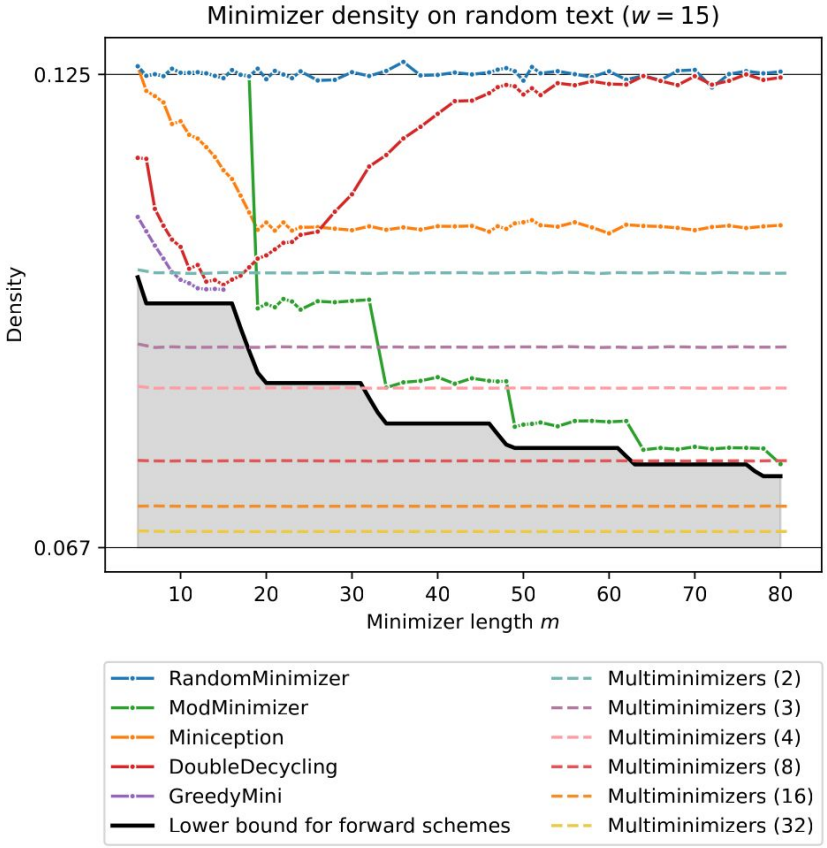
Appendix: multiminimizers for more w values



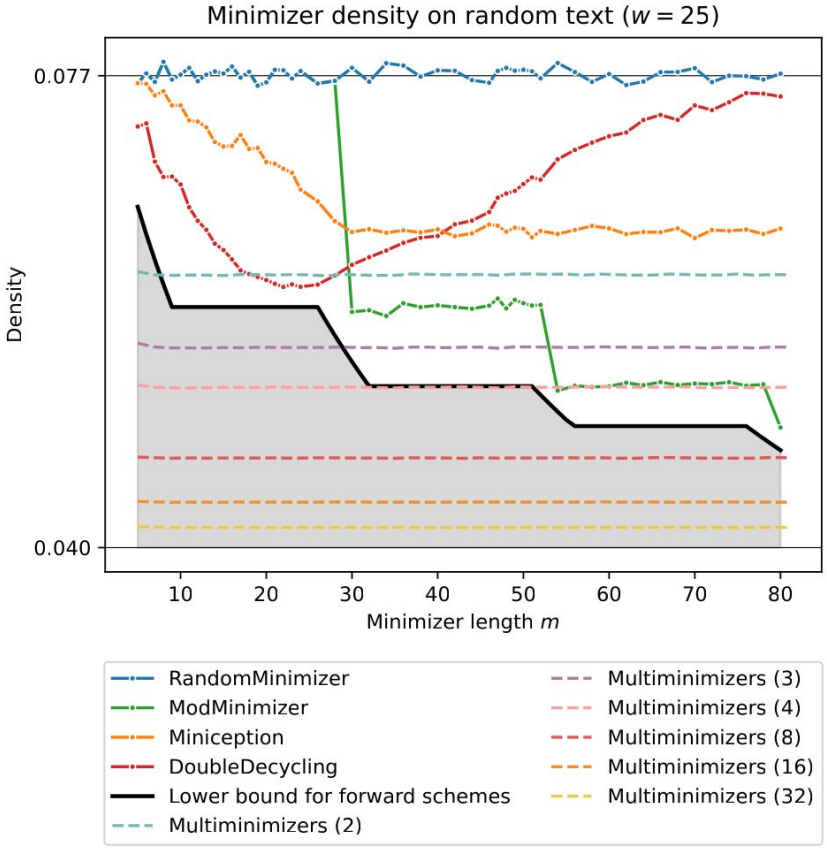
Appendix: multiminimizers for more w values



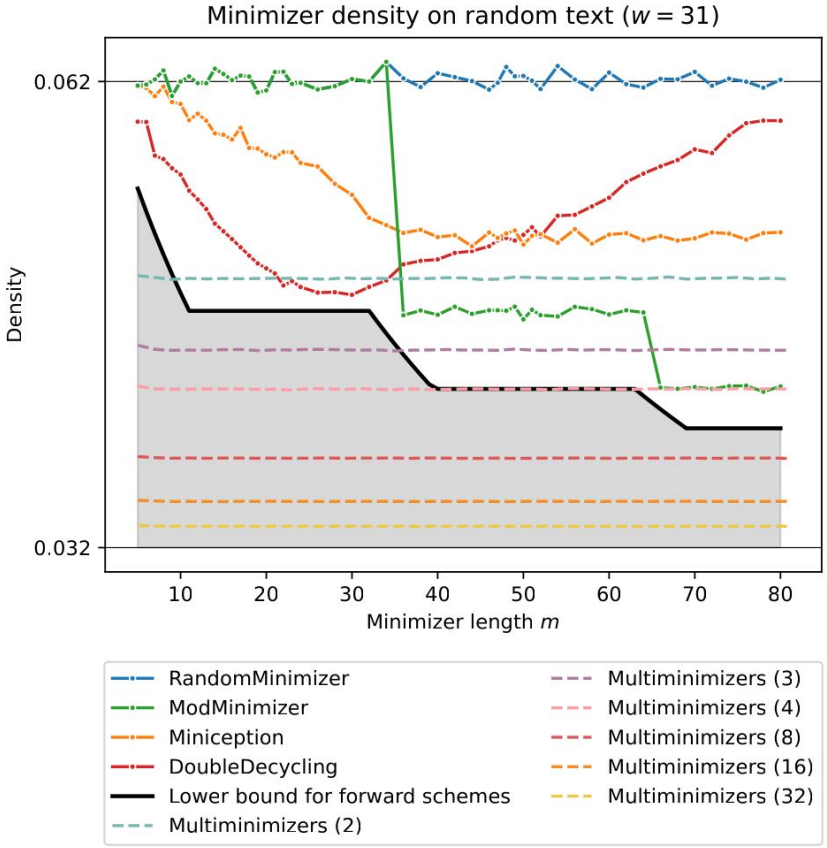
Appendix: multiminimizers for more w values



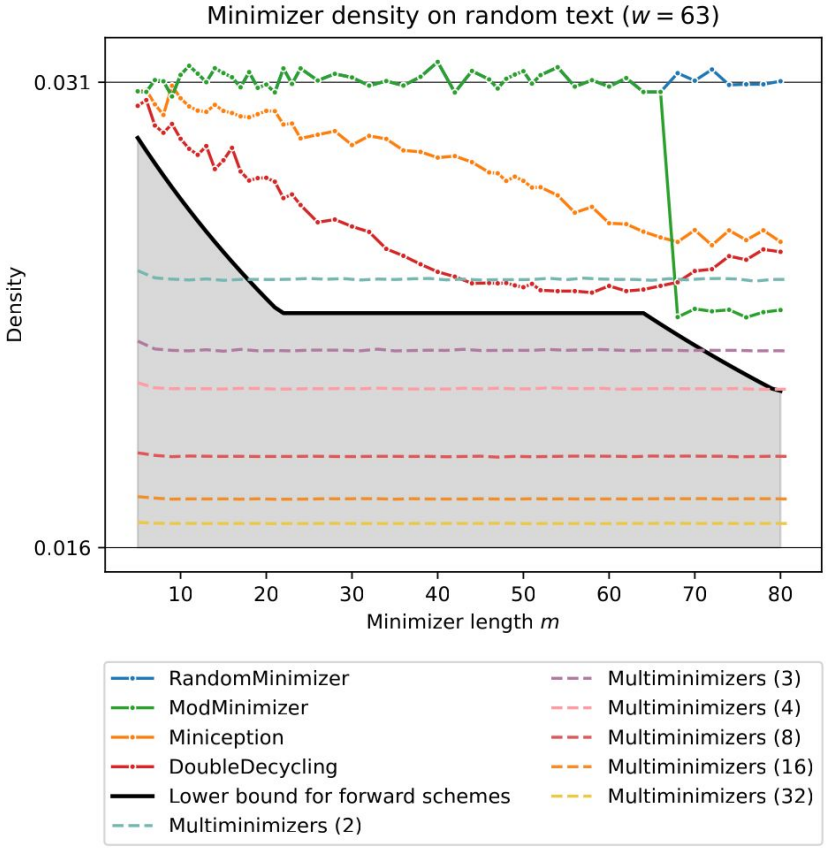
Appendix: multiminimizers for more w values



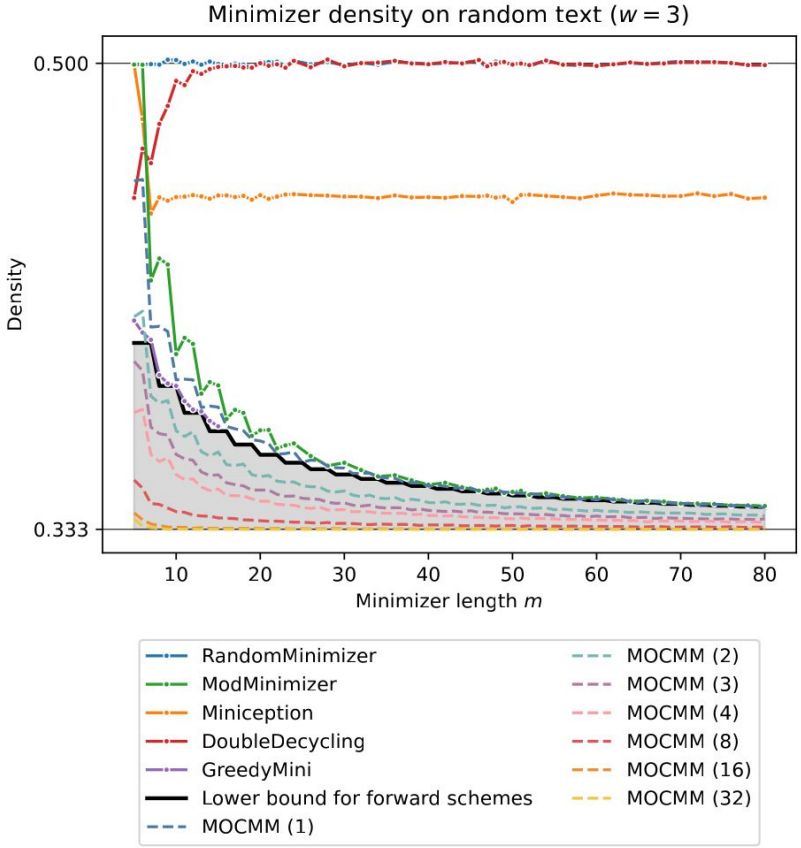
Appendix: multiminimizers for more w values



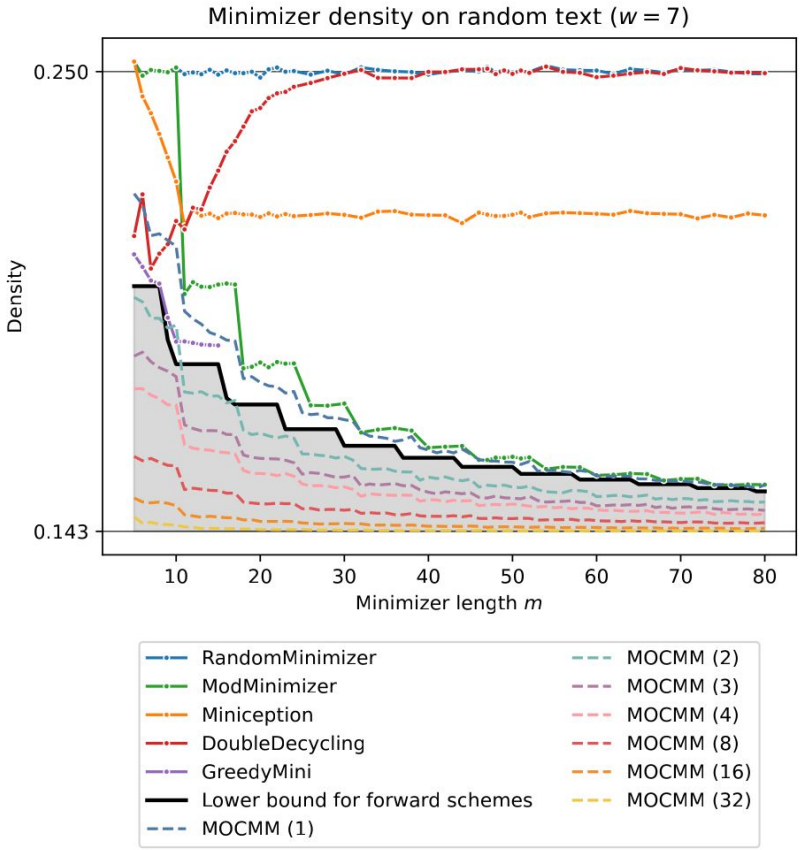
Appendix: multiminimizers for more w values



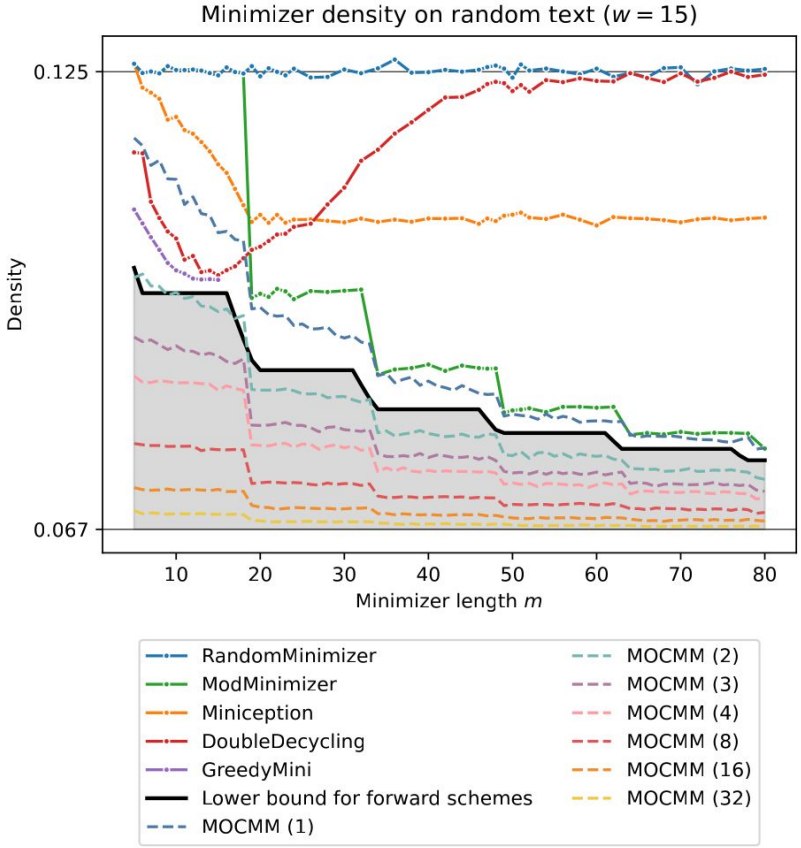
Appendix: MOCMM for more w values



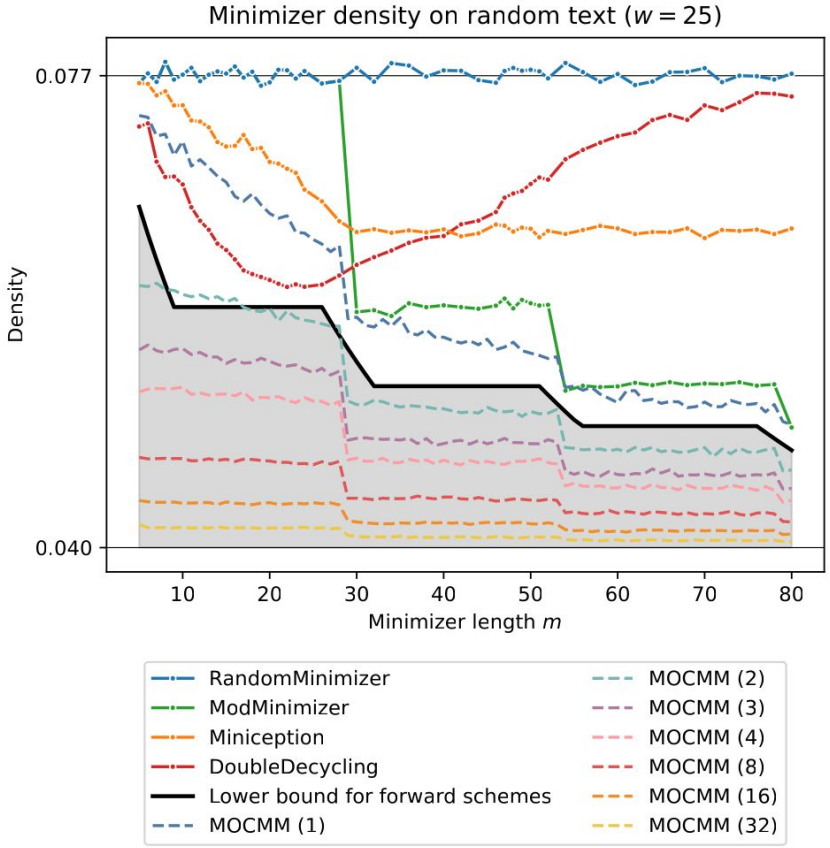
Appendix: MOCMM for more w values



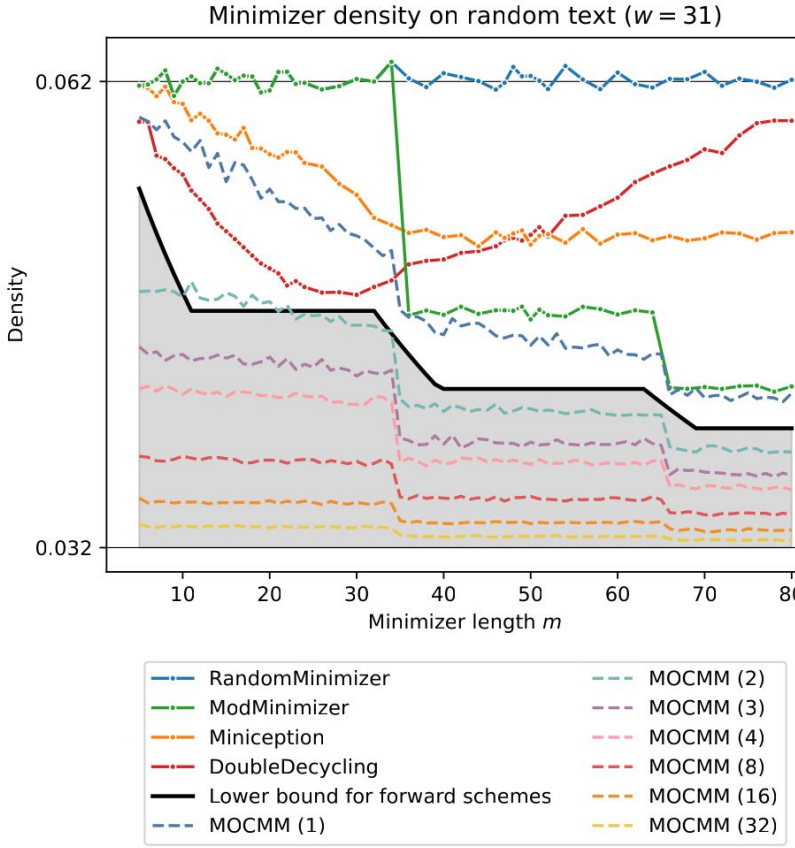
Appendix: MOCMM for more w values



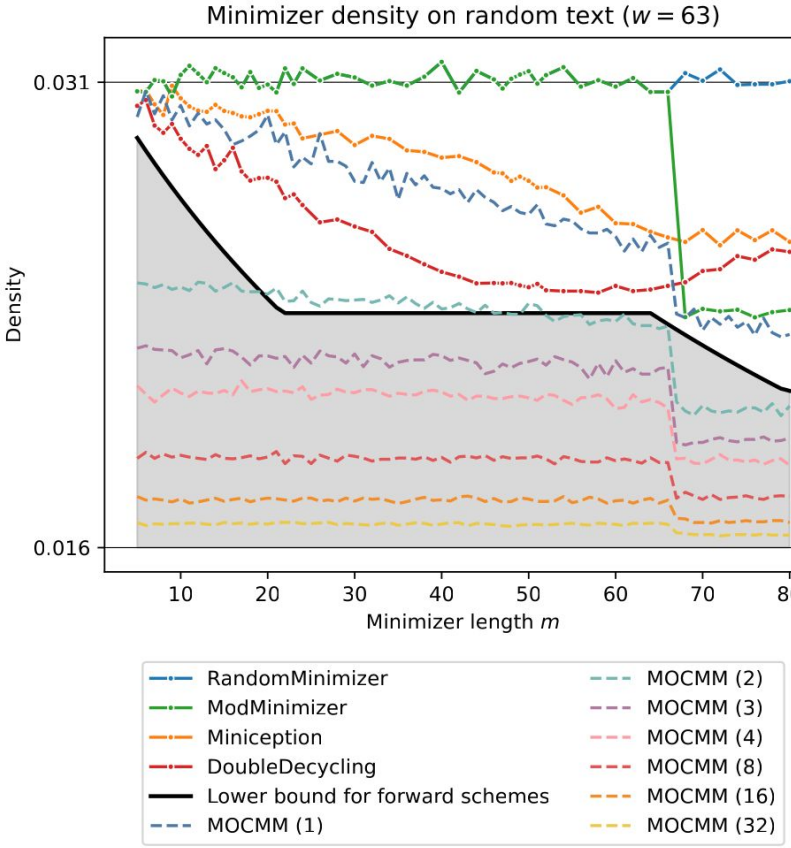
Appendix: MOCMM for more w values



Appendix: MOCMM for more w values



Appendix: MOCMM for more w values



Appendix: MOCMM for more w values