SuperSampler: Efficient and Lightweight Genomic Data Sketching tool

Timothé Rouzé [1] Igor Martayan [2] Camille Marchet [1] Antoine Limasset [1]

[1]Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France [2]ENS Rennes, Univ. Rennes, France

Context

The exponential increase in publicly available sequencing data and genomic resources necessitates the development of highly efficient methods for data processing and analysis. Locality-sensitive hashing techniques have successfully transformed large datasets into smaller, more manageable sketches while maintaining comparability using metrics such as Jaccard and containment indices. However, fixed-size sketches encounter difficulties when applied to divergent datasets. Other methods based on sketches of adaptative size are needed (called scaled size sketching). An example being Sourmash, an implementation of FracMinHash for k-mer subsampling. All the work presented here is available on GitHub. A preprint about the method is available by scanning the QR code at the bottom of this poster!

Fixed size VS Scaled size sketches

When studying metagenomes or reads sets, datasets can be very different in content and in sizes. This makes fixed size sketching such as implemented in Mash, unreliable for Jaccard index approximation. Other methods relying on scaled size sketching technique are necessary.

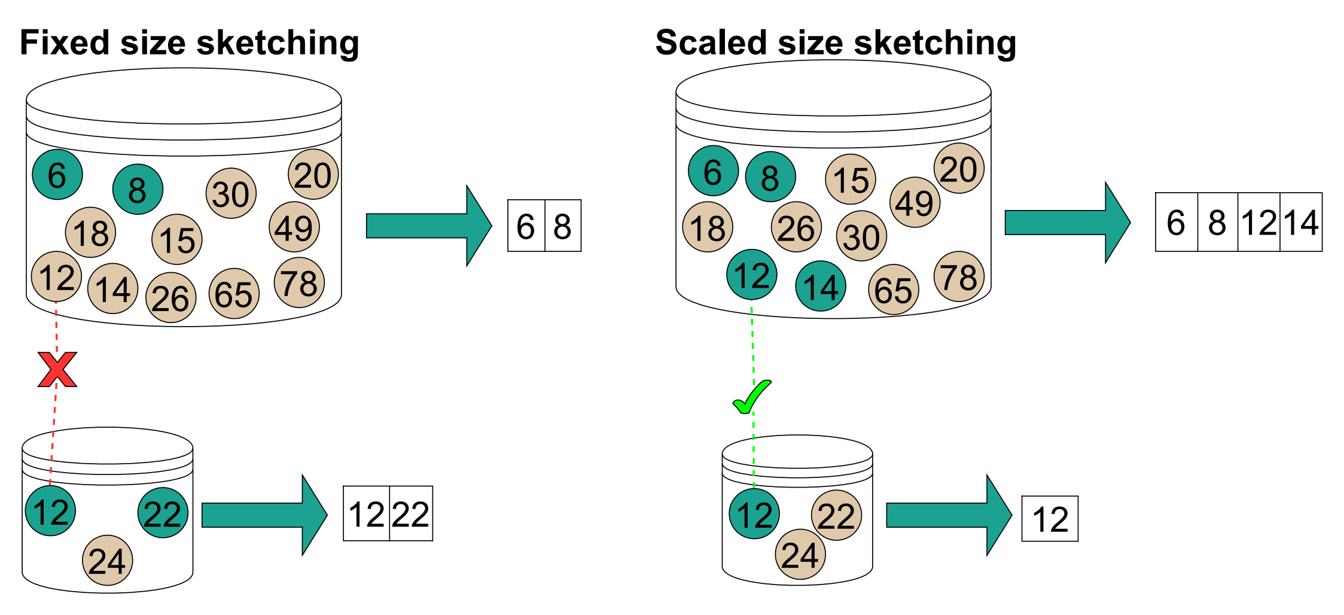

Figure 1. Toy example of the differences between fixed (left) and scaled size sketching (right).

super-k-mers

super-k-mers

Super-k-mers are an aggregation of consecutive k-mers around a minimizer.

Minimizers

Minimizers are the smallest m-mers of a k-mer according to some order (here lexicographical).

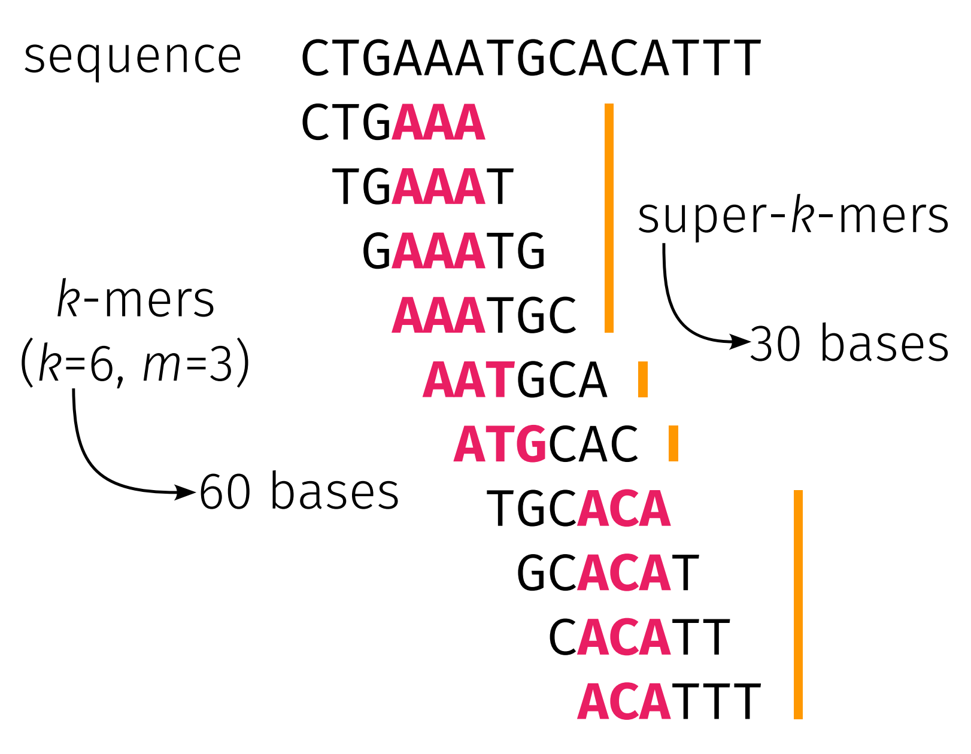

Figure 2. Toy example of a super-k-mers representation for a small sequence.

More information on the theory behind super-k-mers sketching can be found on Igor Martayan's Poster named "Fractional Hitting Sets"!

Sketch construction

SuperSampler constructs sketches by aggregating super-k-mers sharing common minimizers under specific buckets. This allows several improvements for both disk memory cost and computational time during comparisons. Obtained sketches are an exact representation of selected k-mers.

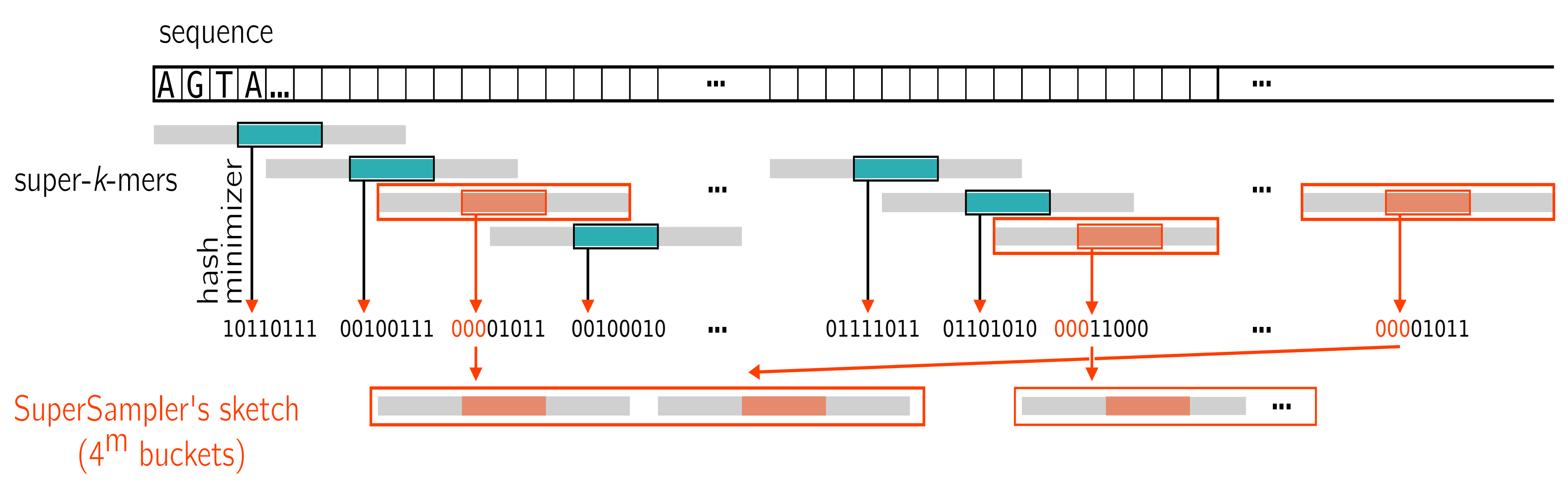

Figure 3. Toy example of a SuperSampler sketch construction.

Sketch comparison

Thanks to the sketch construction, SuperSampler can skip entire buckets during comparisons in divergent datasets. As compared buckets are generally small, they improve cache coherence.

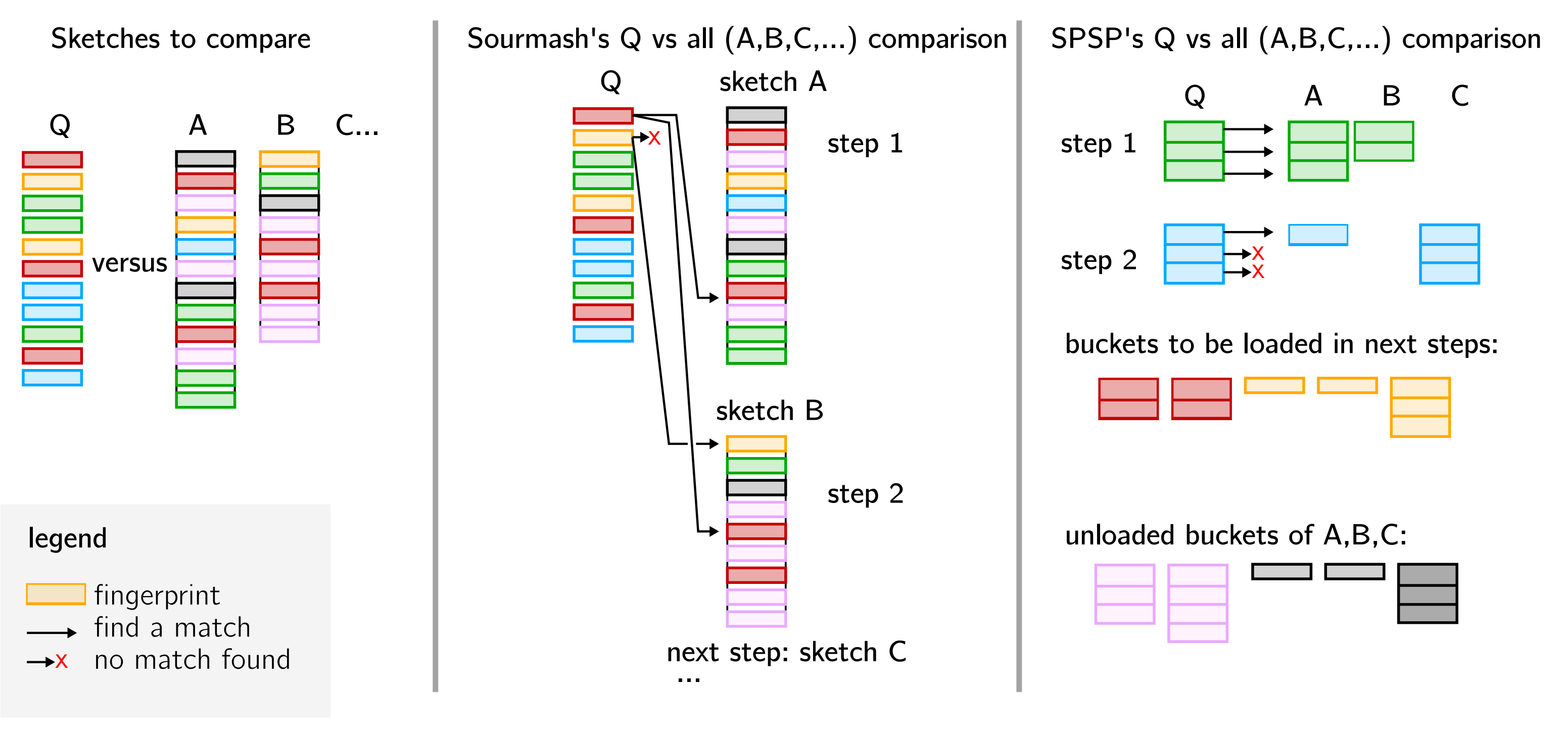

Figure 4. Comparison of workflow for Sourmash and SuperSampler's sketch comparisons.

Results

We compared SuperSampler against Sourmash, the state-of-the-art tool for scaled size sketching using FracMinHash.

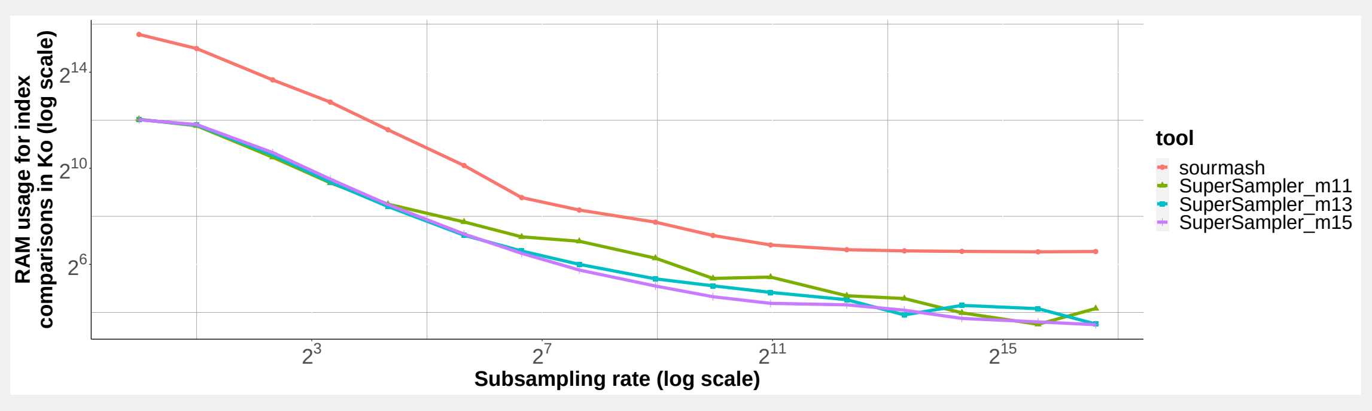

Figure 5. RAM usage for 1024 RefSeq genomes comparison, k = 63.

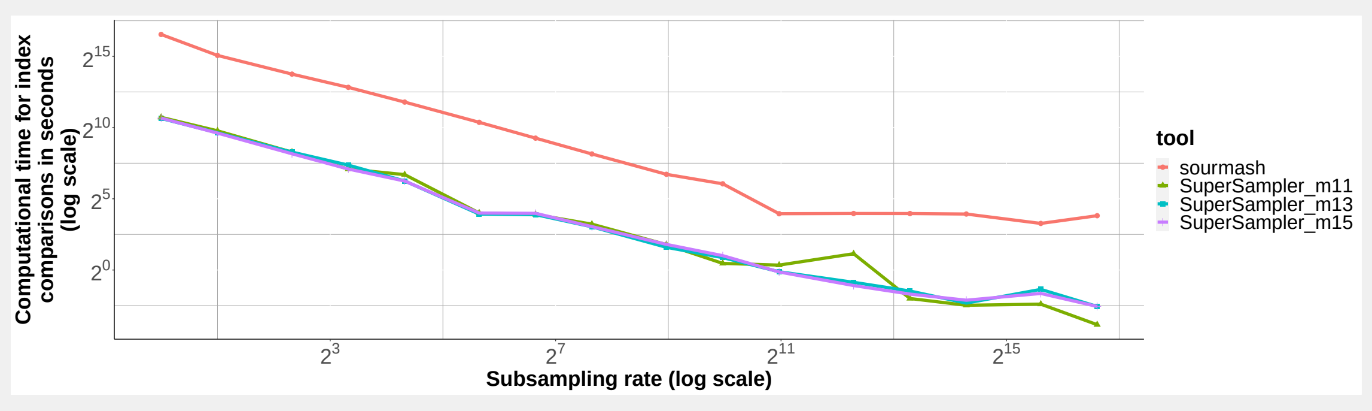

Figure 6. Computational time for 1024 RefSeq genomes comparison, k = 63.

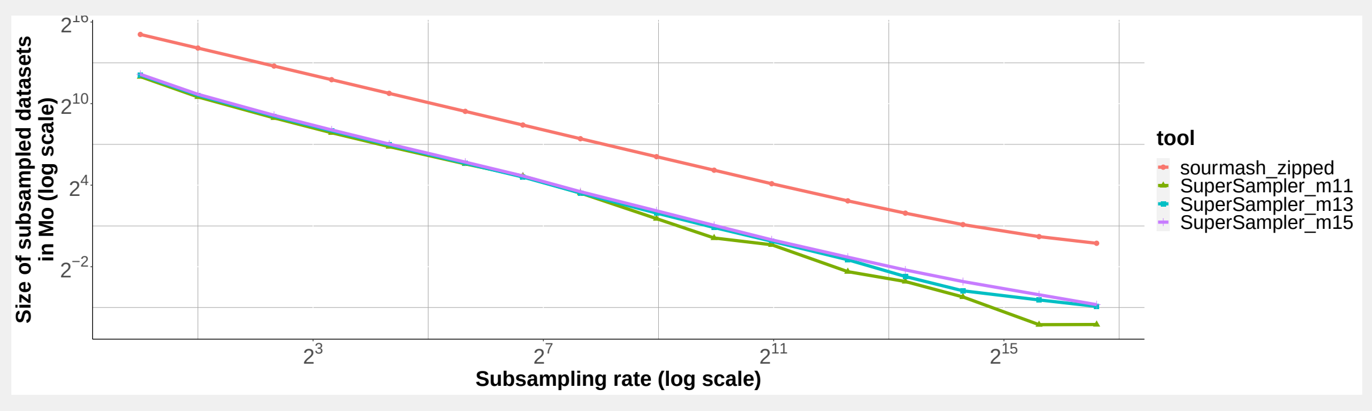

Figure 7. Disk usage for 1024 RefSeq genomes comparison, k = 63.

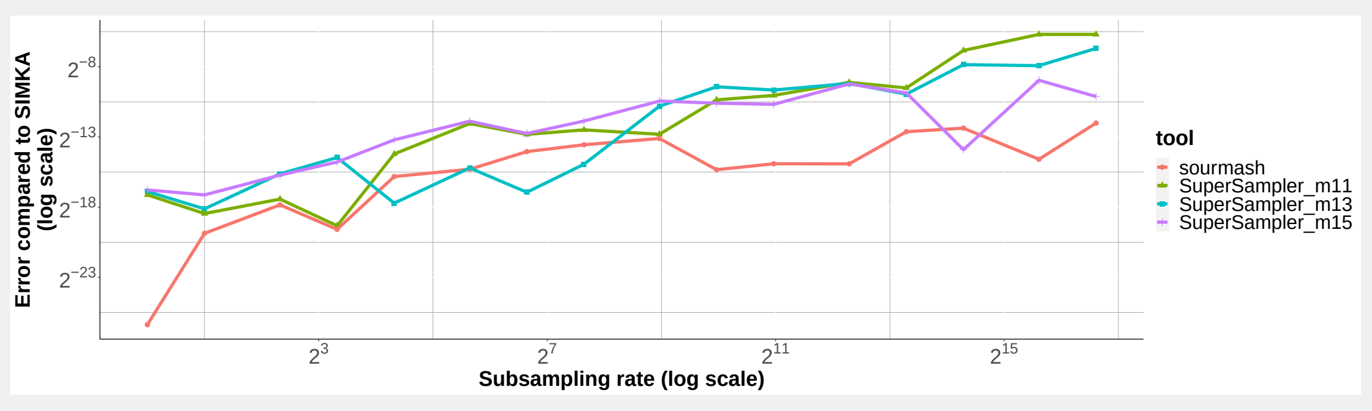

Figure 8. Error for containment index approximation for 1024 RefSeq genomes comparison, k = 63.

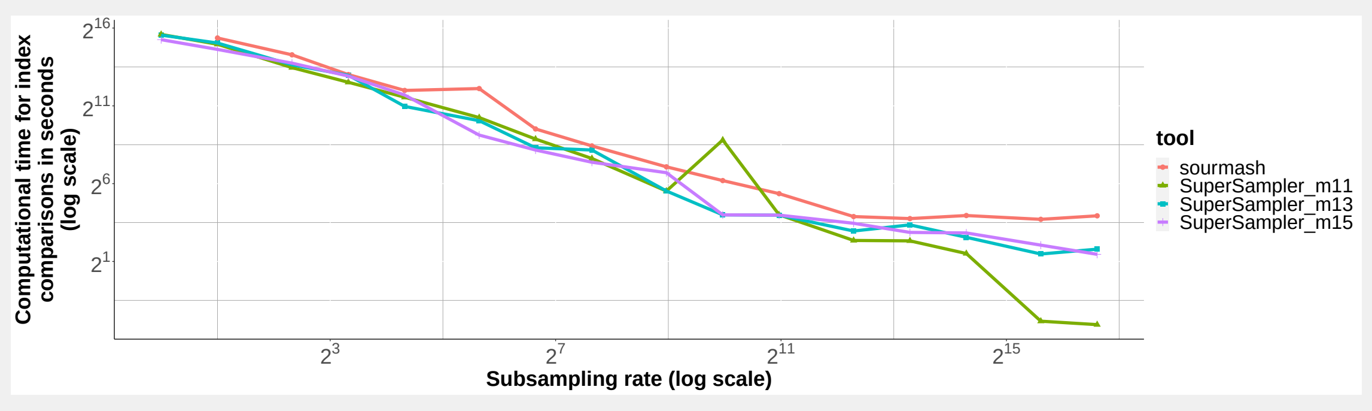

Figure 9. Computational time for 1024 Salmonella genomes comparison, k = 63.

Conclusion

- New sketching scheme
- Faster and Lighter implementation
- Comparable accuracy

Correspondence to:
Timothé Rouzé
timothe.rouze.etu@univ-lille.fr