An efficient alignment-free method for finding genetic differences between pig races from individual whole genome sequencing data

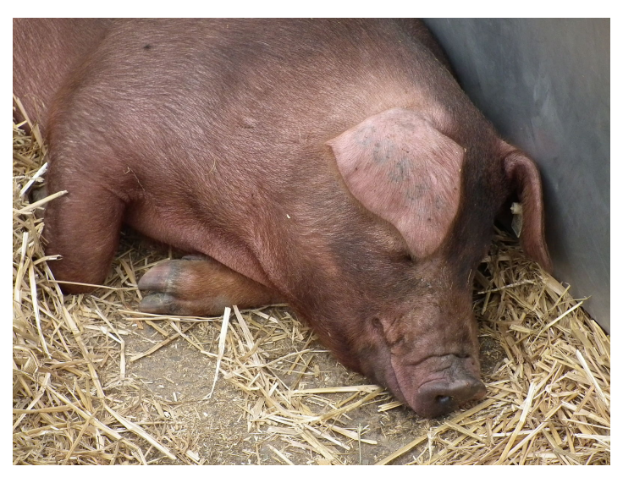
Jens Zentgraf[†‡] Sven Rahmann[†]

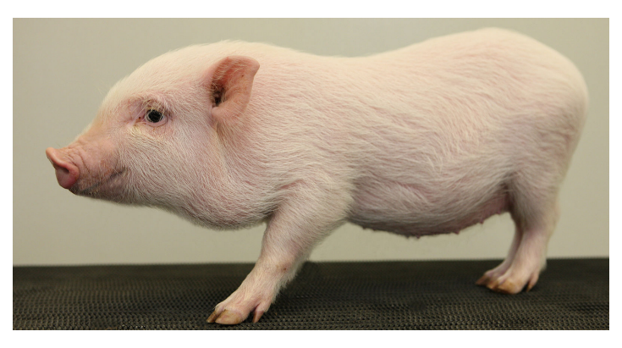
[†] Department of Computer Science and Center for Bioinformatics, Saarland University, Saarbrücken, Germany
[‡] Saarbrücken Graduate School of Computer Science, Saarland University, Saarbrücken, Germany

Goal

Find genes (genomic regions) that are:

- unaltered between Duroc pigs and Minipigs, lost or heavily altered in Ossabaw
- explain different physiology in Ossabaw pigs (e.g. obesity)


Duroc


Minipig

Ossabaw

Limitations and Hardware

- Only annotated Duroc reference
- No Minipig or Ossabaw reference with annotations
- Limited computing resources:
  - 16 cores (32 threads)
  - 64 GB RAM

Datasets

- **Duroc** refenrence genome (*Sus Scrofa* reference 11.1)
- 10 indiviudal sequenced **Minipigs** (NCBI SRA: PRJEB27654, ERR2744277 to ERR2744286)
- 8 individual sequenced **Ossabaw** pigs (Proprietary data, unpublished, PD Dr. Petra Kleinbongard, Institute of Pathophysiology, University Hospital Essen.)

Species	Samples	Reads (sum)	Gbp (sum)	Cov (sum)
Minipigs	10f	1 969 166 591	393.83	152.65x
Ossabaw	4f	3 349 435 217	1004.83	389.5x
Ossabaw	4m	2 870 736 879	861.22	333.8x

Classical approach: Alignment

Disadvantages

- Requires a lot of computing power
- Large BAM files

1: Workflow for one individual

For ear read in the sample file:

- Find most plausible origin in the Duroc reference
- Store alignment in BAM file (large)

2: Workflow for one race

For each position in Duroc reference:

- Find all reads aligned across this position
- Call race-specific variants

3: Compare Minipig and Ossabaw

List all locations with:

- Variants in Ossabaw race
- No variant in Minipig race

References

[1] D. Tang, D. Tan, W. Xiao, J. Lin, and J. Fu. KMC3 and CHTKC: best scenarios, deficiencies, and challenges in high-throughput sequencing data analysis. *Algorithms*, 15(4):107, 2022.

[2] J. Zentgraf and S. Rahmann. Fast lightweight accurate xenograft sorting. *Algorithms Mol. Biol.*, 16(1):2, 2021.

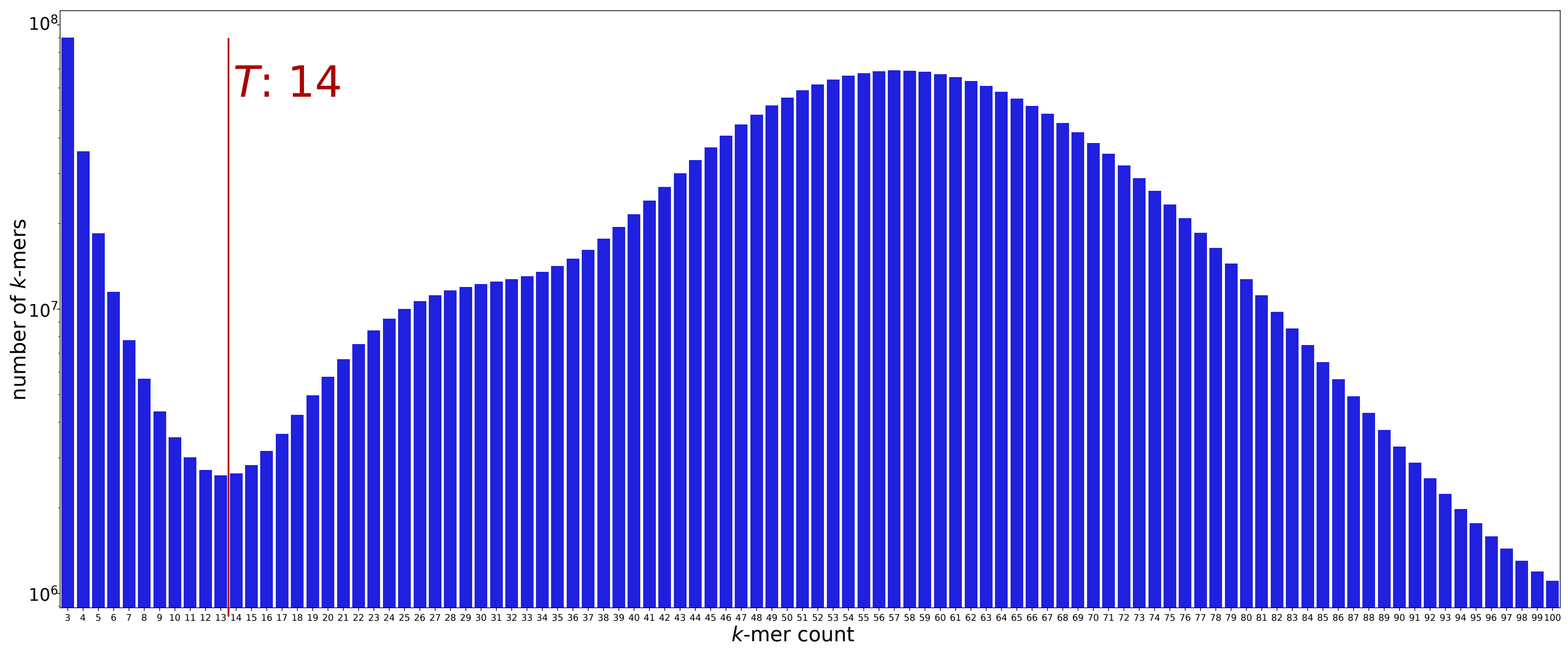
[3] J. Zentgraf and S. Rahmann. Fast gapped k-mer counting with subdivided multi-way bucketed cuckoo hash tables. In *22nd International Workshop on Algorithms in Bioinformatics, WABI*, 2022.

Alignment-free approach

1: *k*-mer Count Histrogram (Anarietta, Ossabaw)

Using 3-way bucketed Cuckoo hashing and blocked Bloom filters:

- Count all *k*-mers in the samples
- Get *k*-mer histogram
- Pick a count Threshold T:
  - Count $\geq T$: *k*-mer belong to the genome
  - Count $< T$: *k*-mer is not specific



2: *k*-mer Distribution

k-mer belongs to a species if it belongs to sufficently many individuals (≥ 4)

Minipig (columns), Ossabaw (rows)

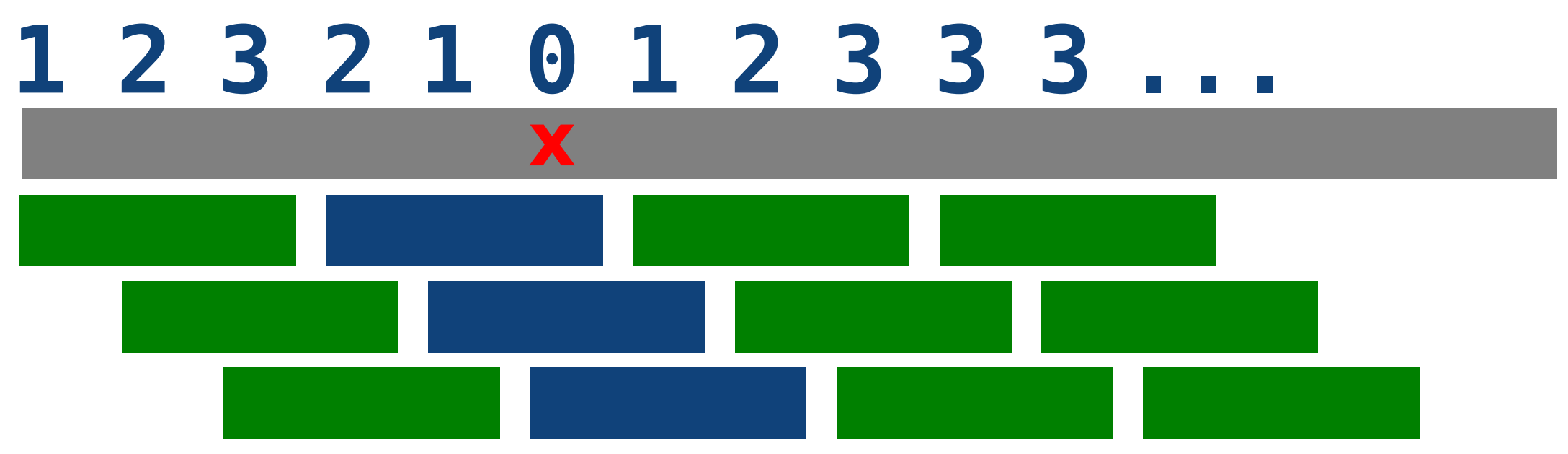
	0	1	2	3	4	5	6	7	8	9	10
0	0.00	2.05	1.40	1.26	1.13	1.00	0.96	0.93	0.91	0.94	1.70
1	0.87	0.08	0.07	0.06	0.06	0.06	0.06	0.06	0.07	0.08	0.15
2	0.72	0.09	0.08	0.07	0.07	0.07	0.07	0.08	0.09	0.10	0.22
3	0.80	0.12	0.09	0.10	0.09	0.09	0.09	0.10	0.11	0.13	0.28
4	1.44	0.21	0.17	0.17	0.17	0.16	0.16	0.17	0.19	0.23	0.52
5	0.56	0.10	0.09	0.09	0.08	0.08	0.09	0.10	0.11	0.14	0.34
6	0.61	0.12	0.10	0.09	0.10	0.10	0.11	0.12	0.13	0.15	0.37
7	0.77	0.18	0.16	0.16	0.16	0.17	0.18	0.20	0.22	0.30	0.85
8	2.94	1.18	1.15	1.23	1.34	1.48	1.69	2.11	2.91	5.98	**49.70**

3: Comparing Minipig and Ossabaw

For each race:

- Scan across Duroc reference, one chromosome at a time
- For each *k*-mer look up if:
  - *k*-mer is part of the race (**green**)
  - *k*-mer is **not** part of the race (**blue**)
- For each position:
  - How many green *k*-mers cover it (typically k)

Variants and lost basepairs



Variants:

- **Identical regions**: All counters are constant and equal to k
- **Single nucleotide substitution (1bp)**: Counter drop from k to 0, then go back up to k
- **Isolated deletion of length ℓ**:
  - Counter drops from k to 0
  - Stay at 0 for ℓ base pairs
  - Go back up to k

Lost basepair:

- Position with small counter (0 if strict)

Complications

- *k*-mers may appear in different places in the genome, not only at the current position
- Variants may not be isolated

⇒ Complex, hard to interpret counter pattern

- We do **not** need individual genome variants

Discovery Method

- Consider exons of genes of the Duroc reference
- For each gene: Count number of lost Duroc base-pairs in Minipig and Ossabaw genome
- Interesting regions depend on threshold T
  - High loss in Ossabaw
  - low (zero) loss in Minipig
- Analyze gene list

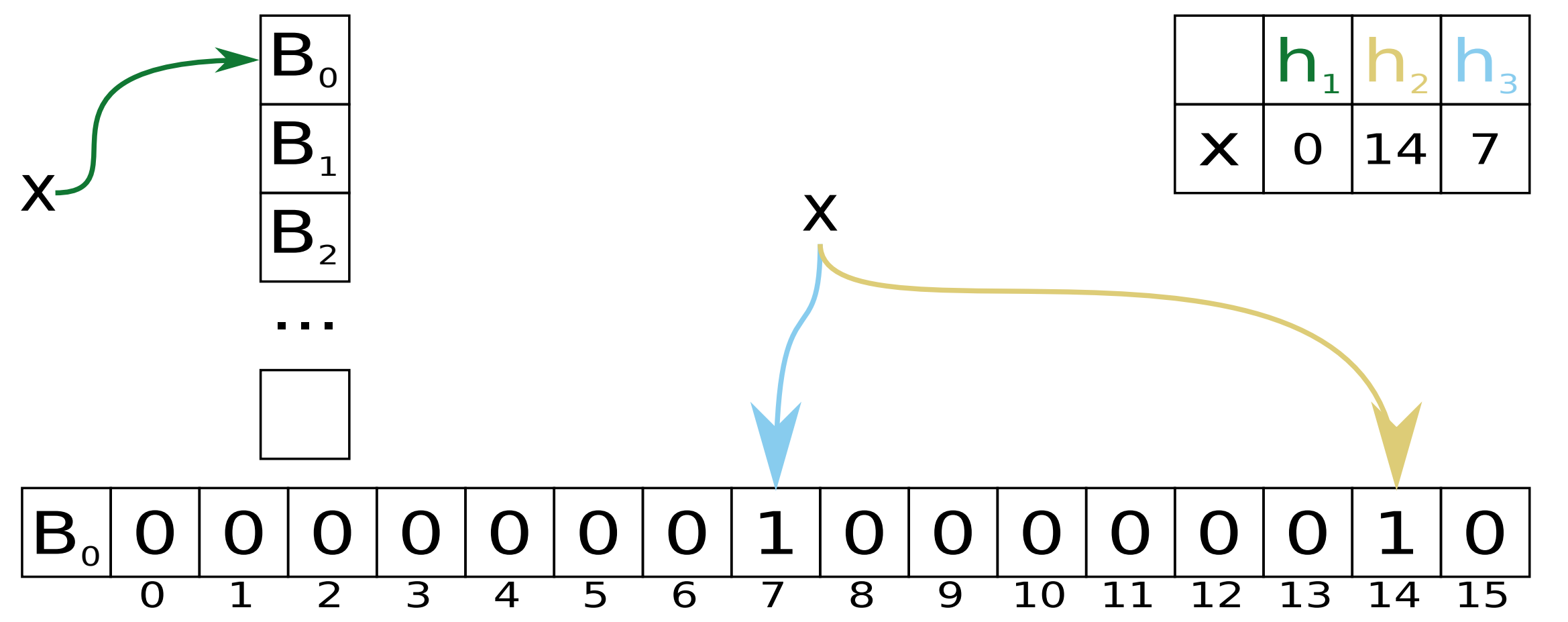
Blocked Bloom Filter

Problem: Very many rare *k*-mers

- In total 40 to 50 G *k*-mers occure only once or twice
- Only interested in the $\leq$ 2.5G frequent *k*-mers
- Only 64 GB of RAM to store *k*-mers and counters

Solution: Use multiple Bloom Filters

- 2 blocked Bloom filters
- First catches all *k*-mers that occure once
- Second catches all *k*-mers that occure twice



3-Way Cuckoo Hash Table

Cuckoo Hashing:

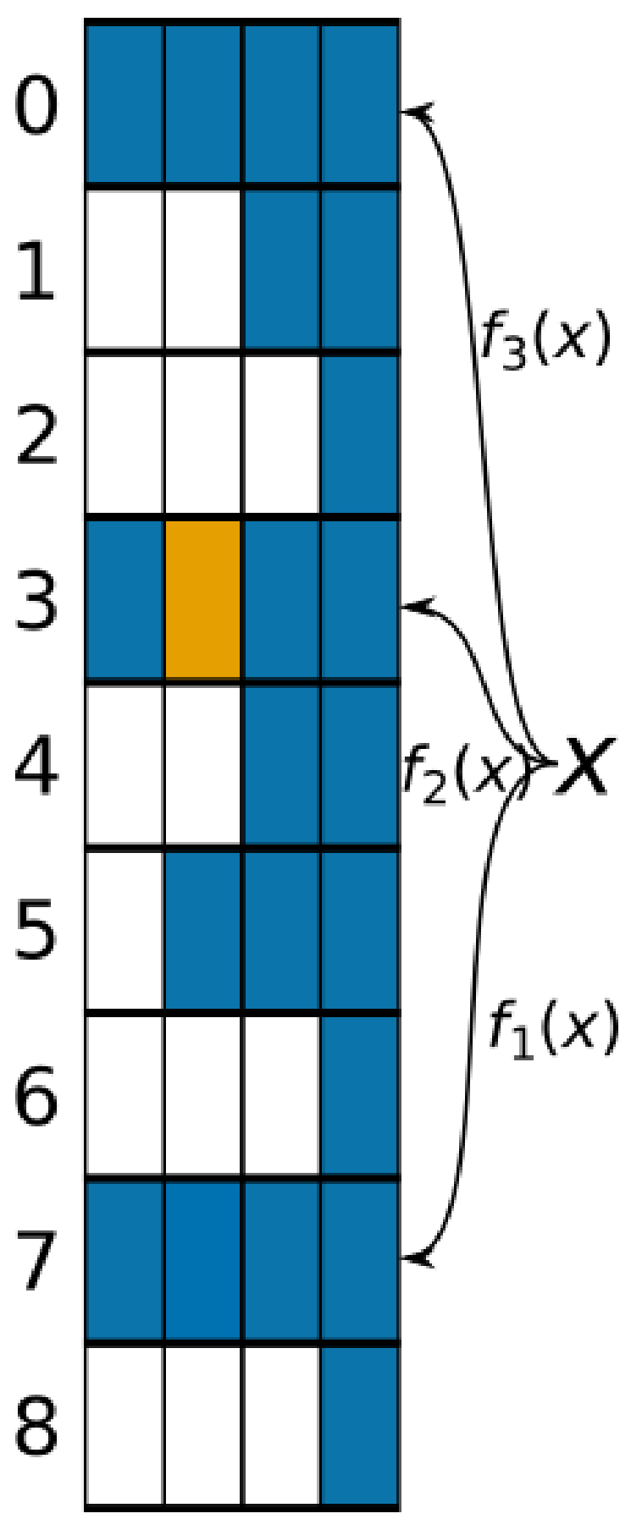
- $h = 3$ hash functions
- $b = 4$ bucket size
- A key has hb possible slots
- If full, replace random key and reinsert old key
- Repeat for fixed number of steps, FAIL if not possible

Save memory:

- High load $\geq$ 90% (up to 99.9%)
- Use word packing (only 54 bits for a 27-mer, not a full uint64)
- Only store quotient of a k-mer

Save time:

- Use multiple threads
- Write access to the same memory
  - Use independent sub-tables
  - All hash functions map to same subtable
  - Use producer-consumer model



Preliminary Results

High-ranked genes: mitochendrial proteins, inflammatory proteins from JAK-STAT pathway

zentgraf@cs.uni-saarland.de Department of Computer Science and Center for Bioinformatics, Saarland University Genome Informatics 2022, September 21–23, 2022