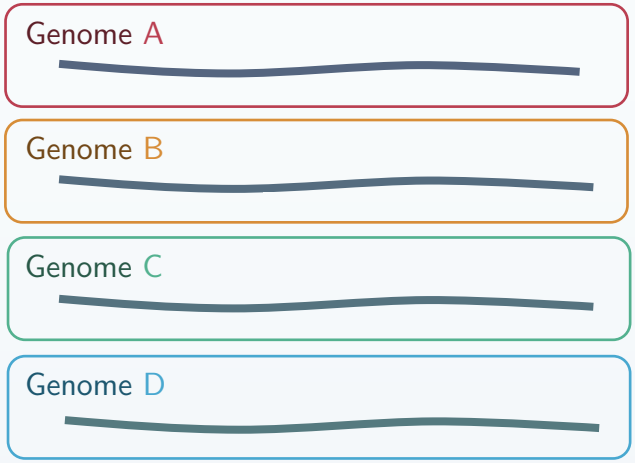
UNIVERSITÄT BIELEFELD
Technische Fakultät

CeBiTec
Center for Biotechnology

Developing a standard interface for *k*-mer-based index structures (update)

Andreas Rempel · June 13, 2022

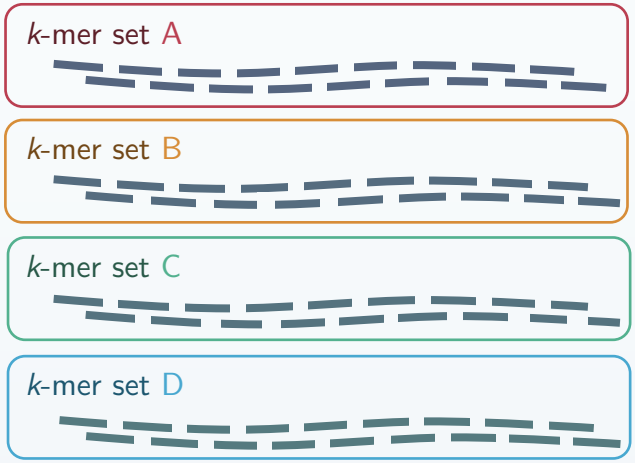
❖ Data structures



Computational Pangenomics

- Thousands of input sequences
- Process large number of strings
  - in a reasonable time
  - within the limits of RAM
- Construct a compact index that allows fast queries

❖ Data structures



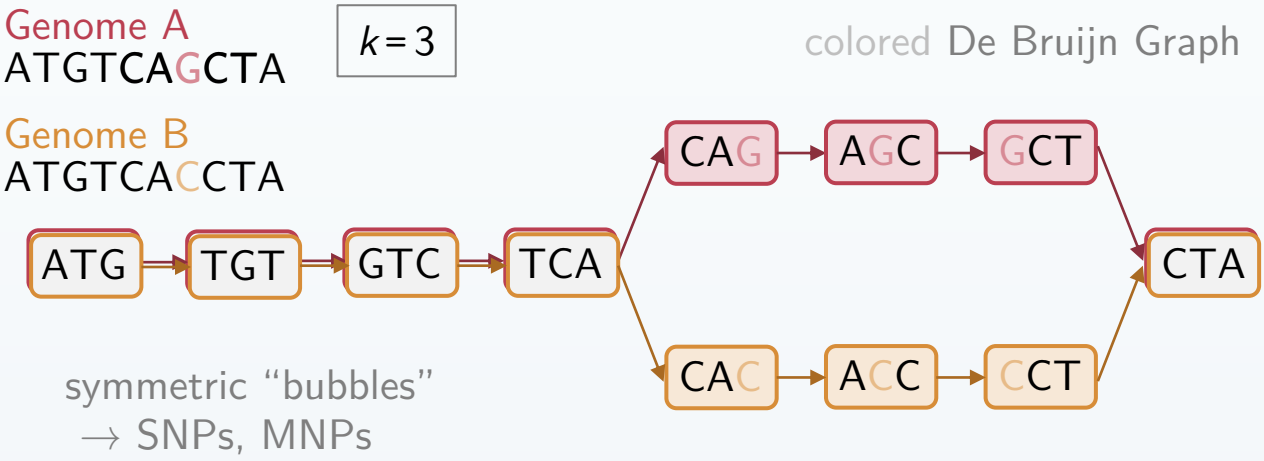
Computational Pangenomics

- Requires advanced algorithms and data structures

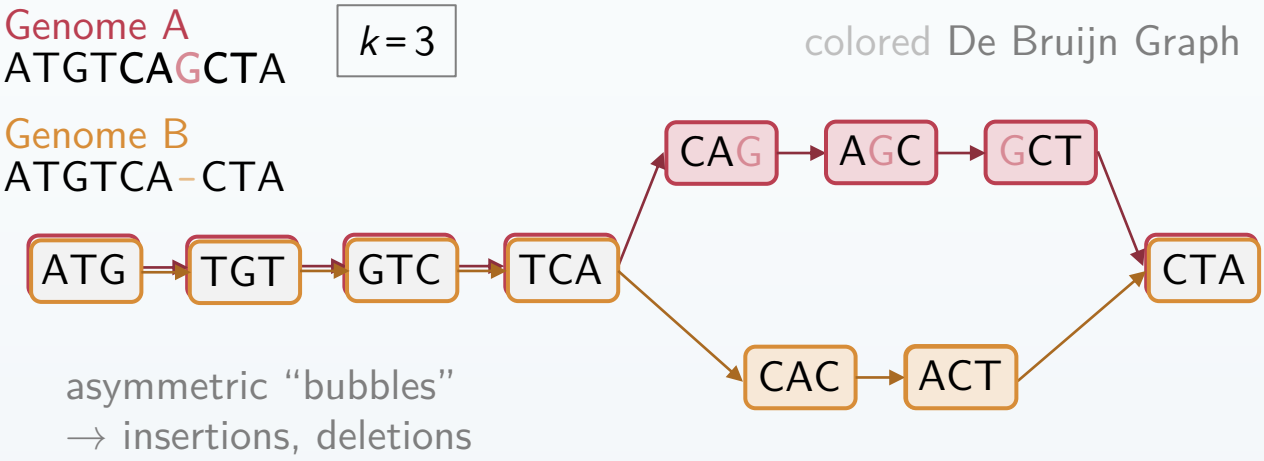
→ **colored *k*-mer sets**

Idea: *k*-mers & colors
Split each genomic sequence into overlapping substrings of length k

❖ Data structures



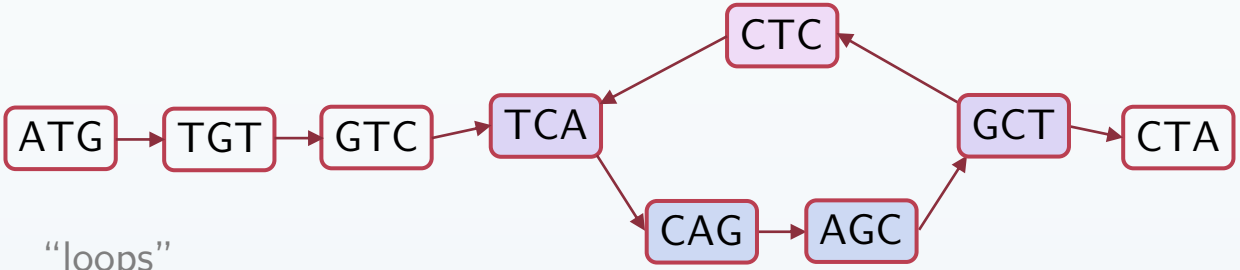
❖ Data structures



❖ Data structures

Genome A
ATGTCAGCTCAGCTA

colored De Bruijn Graph



"loops"
→ repeats

and many more complex structures...

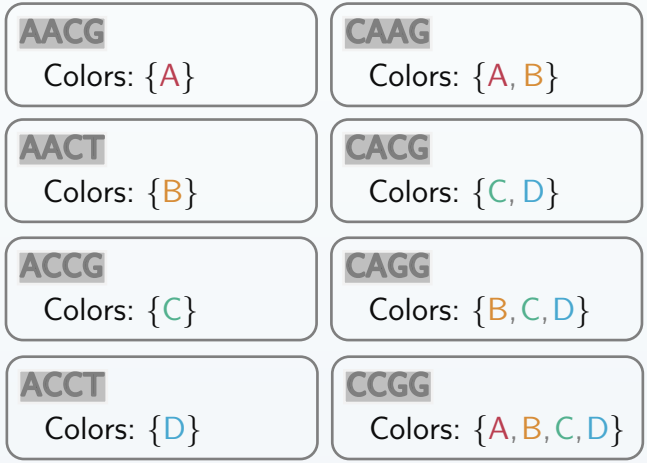
❖ Data structures

Computational Pangenomics

- Requires advanced algorithms and data structures

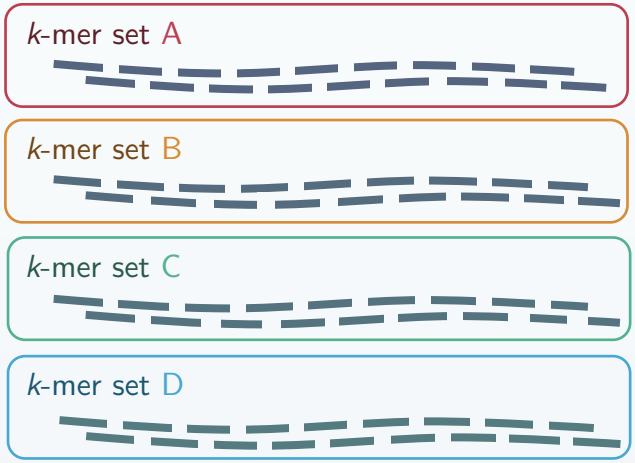
→ **colored *k*-mer sets**

Idea: *k*-mers & colors
Split each genomic sequence into overlapping substrings of length k



❖ Data structures

For each input, store the set of *k*-mers



AACG
Colors: {A}

CAAG
Colors: {A, B}

AACT
Colors: {B}

CACG
Colors: {C, D}

ACCG
Colors: {C}

CAGG
Colors: {B, C, D}

ACCT
Colors: {D}

CCGG
Colors: {A, B, C, D}

For each *k*-mer, indicate the input data
in which the *k*-mer is present

❖ Data structures

For each input, store the set of *k*-mers

Genome A
ATGTCAGCTA

$k = 3$ $h = 3$

ATG
TGT
GTC
TCA
CAG
AGC
GCT

h_1 h_2 h_3

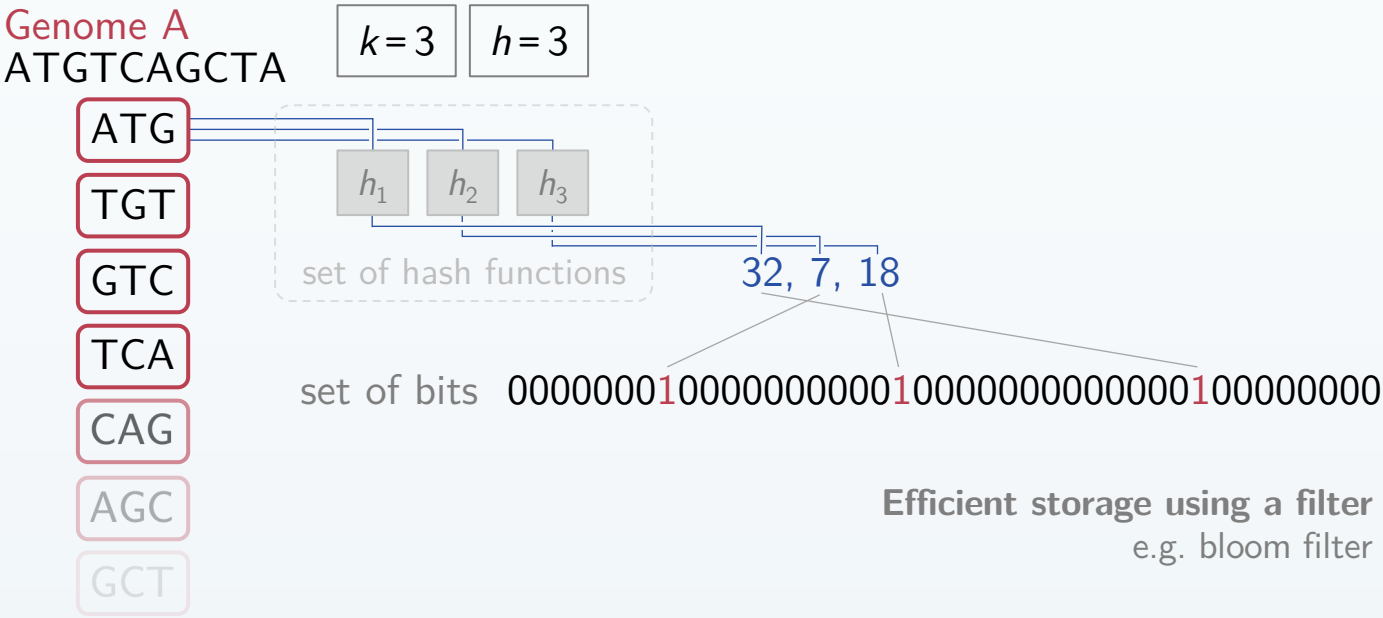
set of hash functions

set of bits 00000000000000000000000000000000000000000000

Efficient storage using a filter
e.g. bloom filter

❖ Data structures

For each input, store the set of *k*-mers



Efficient storage using a filter
e.g. bloom filter

❖ Data structures

For each input, store the set of *k*-mers

Genome A
ATGTCAGCTA 0011111011010100011111100010010011000 10000

Genome B
ACGTTAGCTG 0101011101101011010000011110111000111100

Genome C
CGGTAACGAG 0001011101001101100100101100001111001 0011

Genome D
GTATTACCAT 0010110011001111001101110111000100001 0100

Efficient storage using a filter
e.g. bloom filter

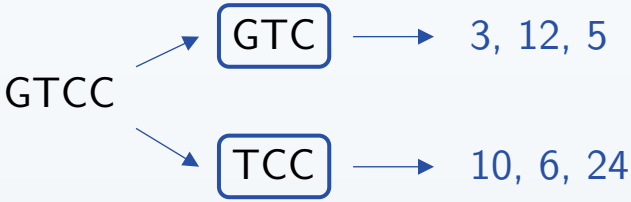
❖ Data structures

For each input, store the set of *k*-mers

Index

Genome A / SRA Exp.	00111101101010001111110001001001100010000
Genome B / SRA Exp.	01010111011010110100000011110111000111100
Genome C / SRA Exp.	00010111010011011001001011000011110010011
Genome D / SRA Exp.	00101100110011110011011101110001000010100

Query



is probably in Genome B
definitely not in A, C, D

false positive rate due to collisions,
is controlled by *h* & number of bits

❖ Available tools

SeqOthello	**Cortex**	**kmtricks**
Mantis	**McCortex**	**kcollections**
SBT/SSBT	**BIGSI**	**Cosmo|VARI**
AllSomeSBT	**COBS**	**Rainbowfish**
HowDeSBT	**RAMBO**	**Cuttlefish**
BFT	**Raptor**	**REINDEER**
Bifrost	**Themisto**	**GATB-core**

... and many more

❖ Available tools

SeqOthello	Mantis	SBT/SSBT	HowDeSBT	BIGSI
PreProcess	count	hashes	makebf	build
Group	build	count	cluster	bloom
Build	mst	build	build	build
Query	query	query	query	search

❖ Available tools

SeqOthello	Mantis	SBT/SSBT	HowDeSBT	BIGSI
PreProcess	count	hashes	makebf	build
Group	build	construct index	cluster	bloom
Build	mst	build	build	build
Query	query	query sequence	query	search

Developing a common standard interface

- **Software for colored *k*-mer sets (SOCKS)**


Knut Reinert


Nadia Pisanti


Carl Kingsford


Brad Solomon


Gaurav Gupta


Pierre Peterlongo


Rayan Chikhi


Yoann Dufresne


Paul Medvedev


Sven Rahmann

❖ SOCKS interface

- construct index
  - input: set of sequences, each with a distinct color
  - output: index

input: plain text file containing the sequence file names
- list of assembled genomes/contigs, multiple fasta
- list of read sets, single-end or paired-end fastq
- one line per color, e.g. whitespace-separated

output: one or multiple files for the constructed index
- proprietary/binary format, compact, fast read/write
- interoperable format, e.g. gfa | fasta | *k*-mer matrix

❖ SOCKS interface

- `build` : construct an index from a set of sequences
  - **input**: plain text file containing the sequence file names, e.g.:

```
COLOR_NAME_1: /PATH/TO/GENOME.FASTA
COLOR_NAME_2: /PATH/TO/READ_1.FASTQ /PATH/TO/READ_2.FASTQ
```

  - **output**: index in binary or interoperable format, e.g. kmer file format
    (at least one of the options, binary or interoperable format, should be provided)
    (if both options are provided, it should be possible to switch using a parameter)

❖ SOCKS interface

- query sequence
  - input: *k*-mer(s)
  - output: set of colors

input: plain text file containing the *k*-mers
- one line per *k*-mer query

output: plain text file listing the color sets
- list of positive hits, e.g. whitespace-separated
- hit/miss vector, e.g. present = '1', absent = '0'
- one line per *k*-mer query result

❖ SOCKS interface

- `lookup-kmer` : find the color sets for a list of *k*-mers
  - **input**: plain text file containing the *k*-mers, one per line, e.g.:

```
ACGTACGT
ACCTAGGT
```

  - **output**: plain text file listing the color set for each *k*-mer, e.g.:

    (as list of positive hits)

```
ACGTACGT: COLOR_NAME_1 COLOR_NAME_4 COLOR_NAME_7 ...
ACCTAGGT: COLOR_NAME_1 COLOR_NAME_5 COLOR_NAME_8 ...
```

❖ SOCKS interface

- `lookup-kmer` : find the color sets for a list of *k*-mers
  - **input**: plain text file containing the *k*-mers, one per line, e.g.:

```
ACGTACGT
ACCTAGGT
```

  - **output**: plain text file listing the color set for each *k*-mer, e.g.:

(or as a binary vector)

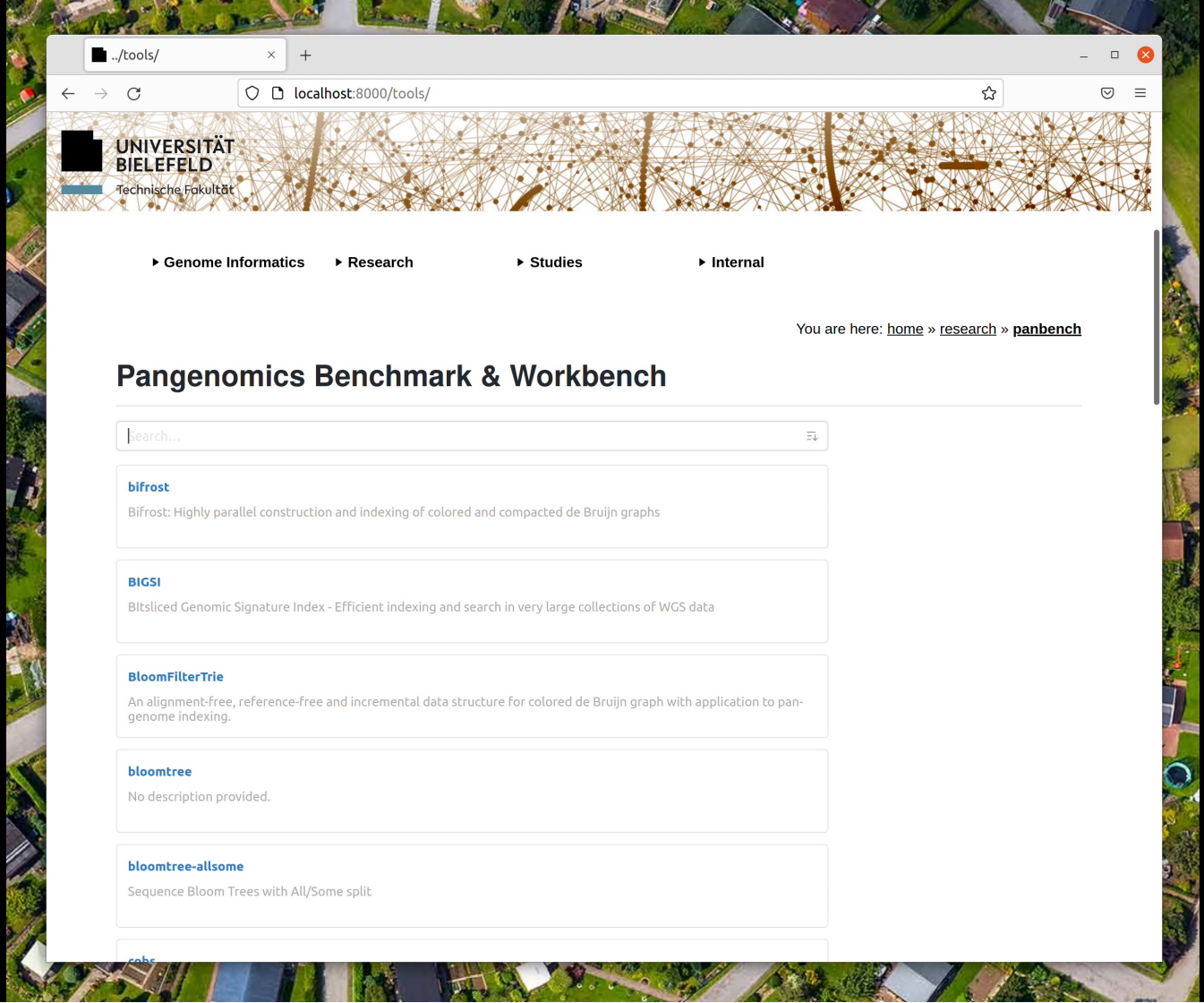
```
ACGTACGT: 10010010...
ACCTAGGT: 10001001...
```

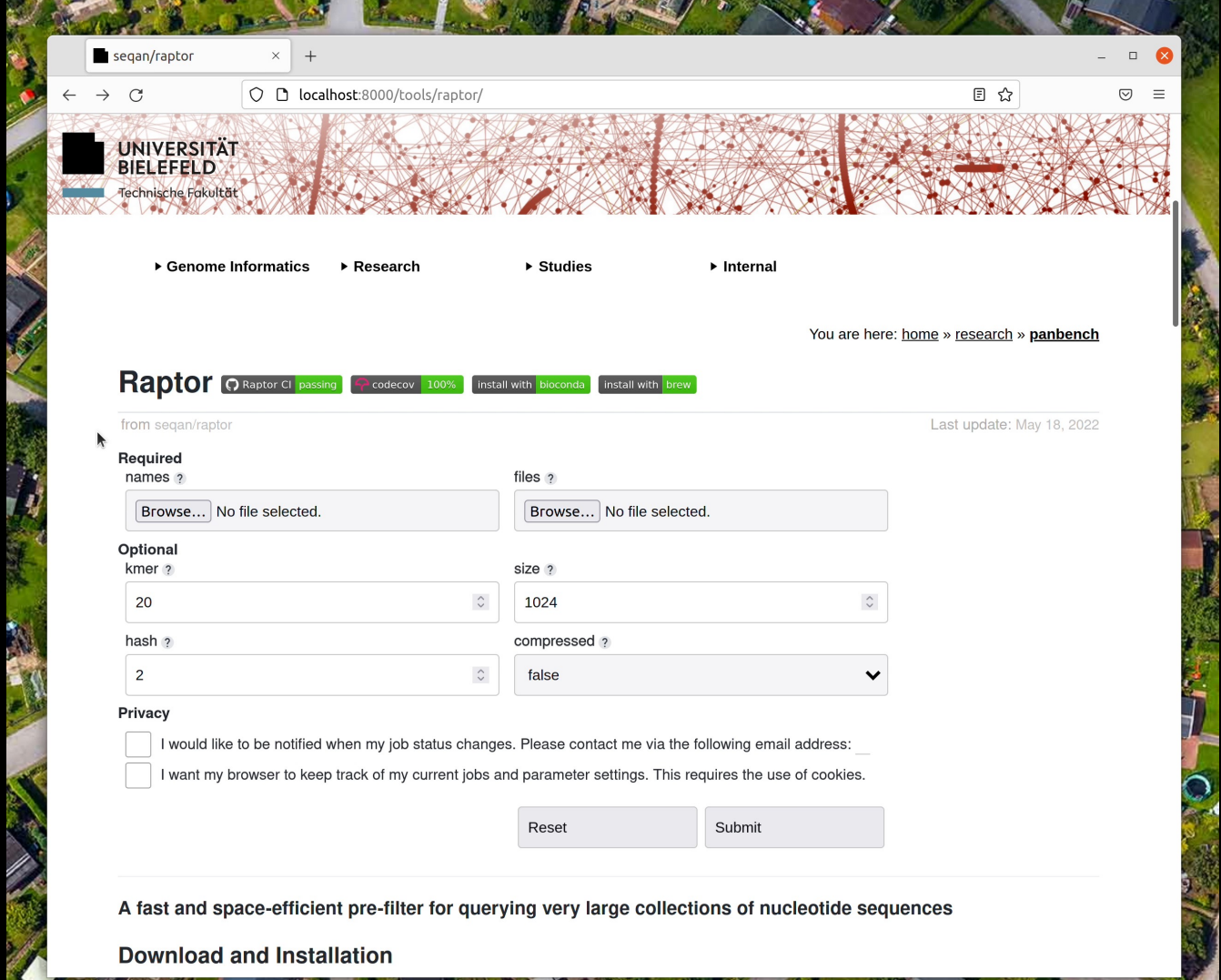
Software for colored k-mer sets (SOCKS)

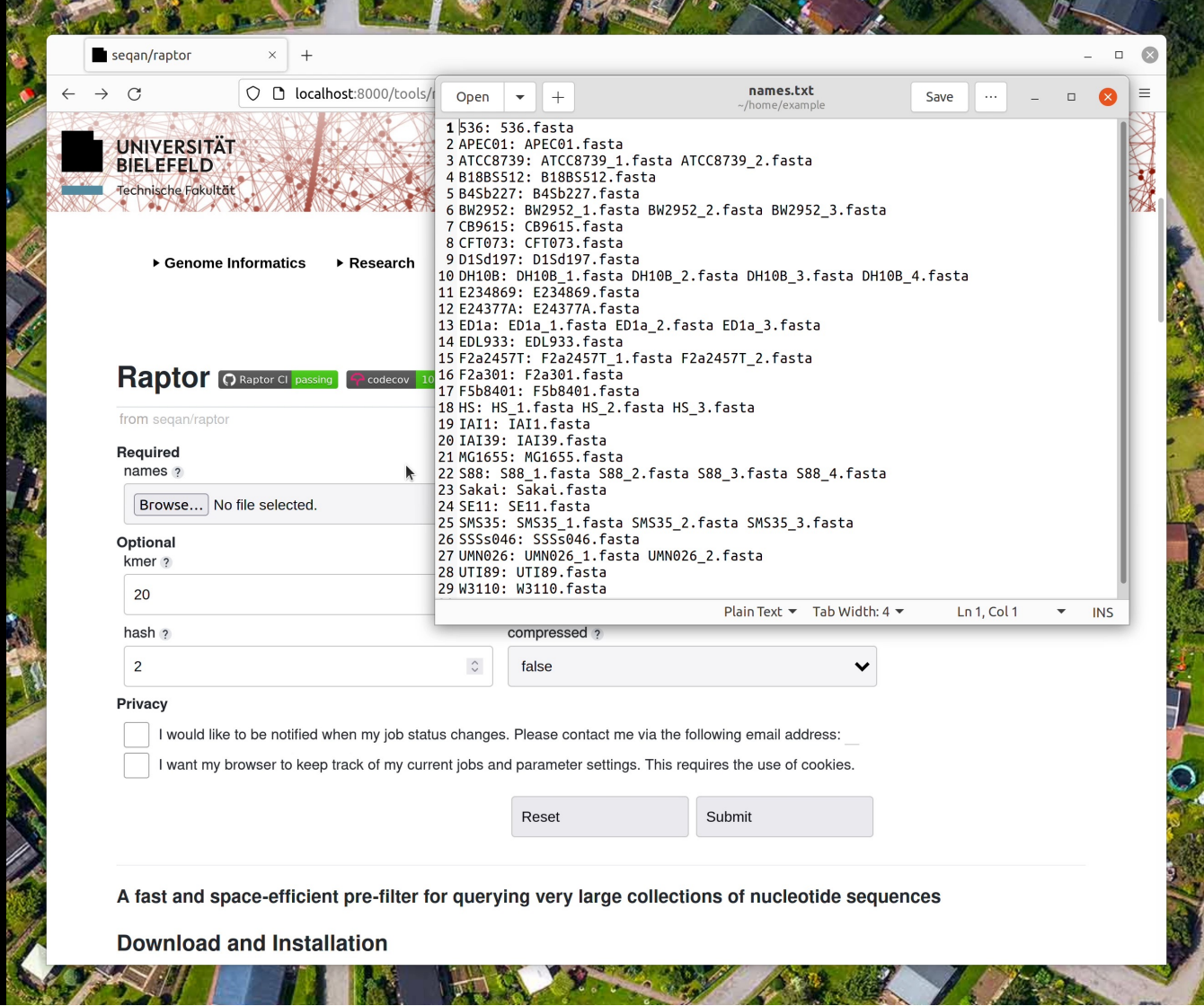
❖ Available tools

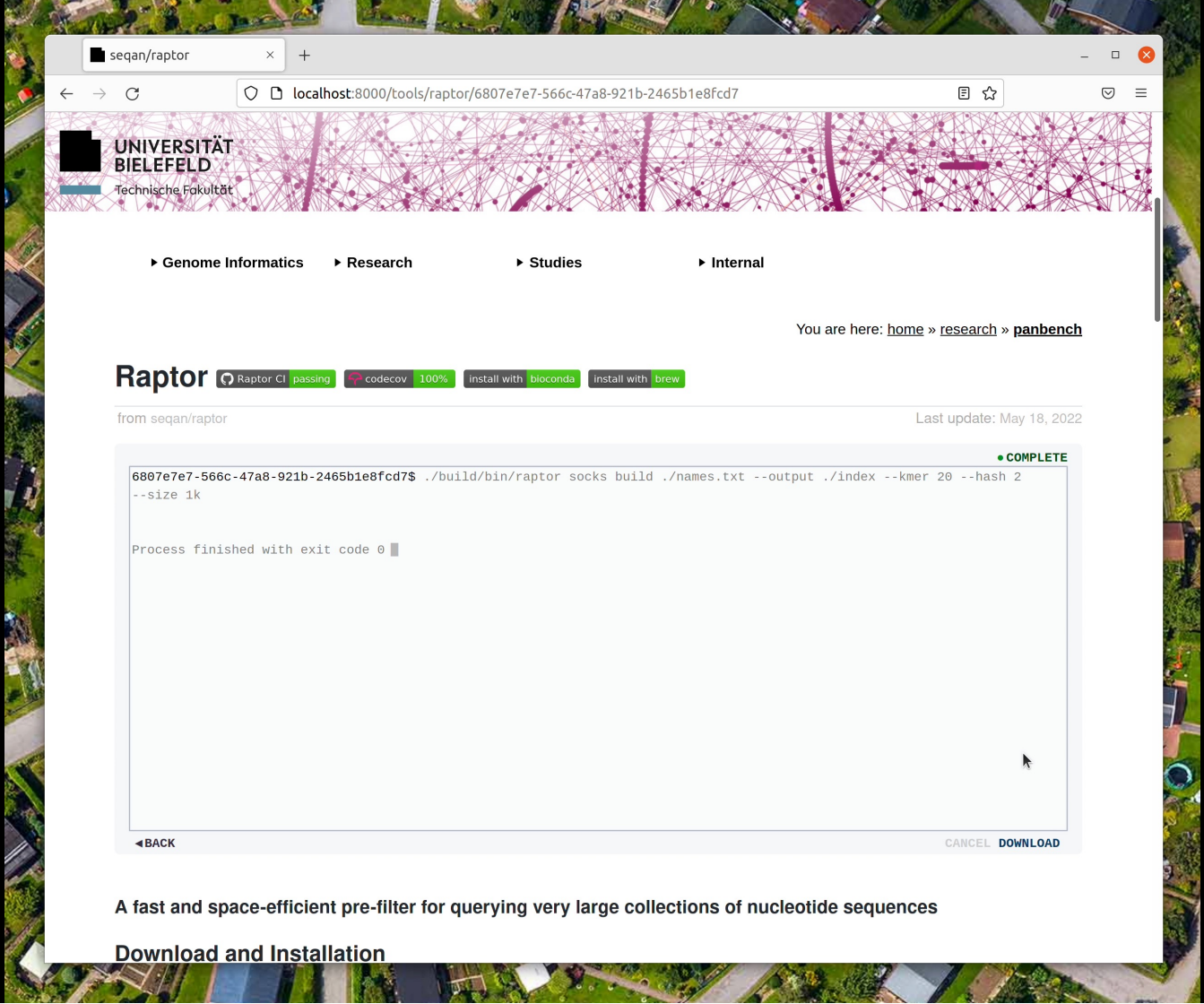
SeqOthello	**Cortex**	**kmtricks**
Mantis	**McCortex**	**kcollections**
SBT/SSBT	**BIGSI**	**Cosmo|VARI**
AllSomeSBT	**COBS**	**Rainbowfish**
HowDeSBT	**RAMBO**	**Cuttlefish**
BFT	**Raptor**	**REINDEER**
Bifrost	**Themisto**	**GATB-core**

... and many more


../tools/
localhost:8000/tools/
UNIVERSITÄT
BIELEFELD
Technische Fakultät
Genome Informatics
Research
Studies
Internal
You are here: home » research » panbench
Pangenomics Benchmark & Workbench
Search...
bifrost
Bifrost: Highly parallel construction and indexing of colored and compacted de Bruijn graphs
BIGSI
BItsliced Genomic Signature Index - Efficient indexing and search in very large collections of WGS data
BloomFilterTrie
An alignment-free, reference-free and incremental data structure for colored de Bruijn graph with application to pan-genome indexing.
bloomtree
No description provided.
bloomtree-allsome
Sequence Bloom Trees with All/Some split

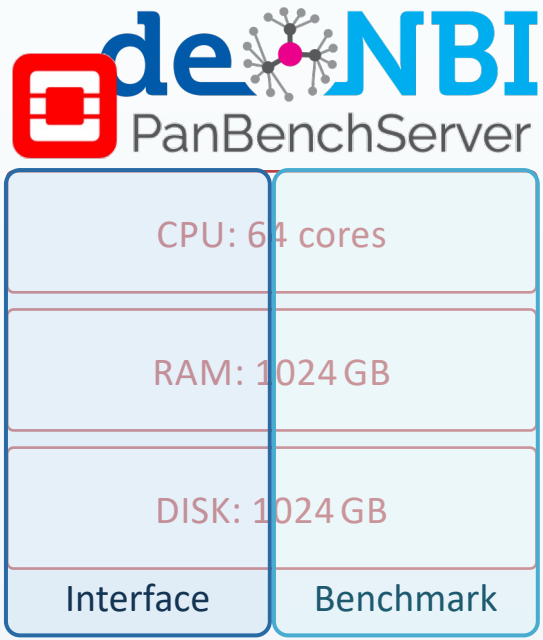

seqan/raptor
localhost:8000/tools/raptor/
UNIVERSITÄT BIELEFELD
Technische Fakultät
Genome Informatics
Research
Studies
Internal
You are here: home » research » panbench
Raptor
Raptor CI passing
codecov 100%
install with bioconda
install with brew
from seqan/raptor
Last update: May 18, 2022
Required
names
Browse... No file selected.
files
Browse... No file selected.
Optional
kmer
20
size
1024
hash
2
compressed
false
Privacy
I would like to be notified when my job status changes. Please contact me via the following email address:
I want my browser to keep track of my current jobs and parameter settings. This requires the use of cookies.
Reset
Submit
A fast and space-efficient pre-filter for querying very large collections of nucleotide sequences
Download and Installation


seqan/raptor
localhost:8000/tools/
UNIVERSITÄT BIELEFELD
Technische Fakultät
Genome Informatics
Research
Raptor
Raptor CI passing
codecov
from seqan/raptor
Required
names
Browse... No file selected.
Optional
kmer
20
hash
2
compressed
false
Privacy
I would like to be notified when my job status changes. Please contact me via the following email address:
I want my browser to keep track of my current jobs and parameter settings. This requires the use of cookies.
Reset
Submit
A fast and space-efficient pre-filter for querying very large collections of nucleotide sequences
Download and Installation
Open
names.txt
~/home/example
Save
1 536: 536.fasta
2 APEC01: APEC01.fasta
3 ATCC8739: ATCC8739_1.fasta ATCC8739_2.fasta
4 B18BS512: B18BS512.fasta
5 B4Sb227: B4Sb227.fasta
6 BW2952: BW2952_1.fasta BW2952_2.fasta BW2952_3.fasta
7 CB9615: CB9615.fasta
8 CFT073: CFT073.fasta
9 D1Sd197: D1Sd197.fasta
10 DH10B: DH10B_1.fasta DH10B_2.fasta DH10B_3.fasta DH10B_4.fasta
11 E234869: E234869.fasta
12 E24377A: E24377A.fasta
13 ED1a: ED1a_1.fasta ED1a_2.fasta ED1a_3.fasta
14 EDL933: EDL933.fasta
15 F2a2457T: F2a2457T_1.fasta F2a2457T_2.fasta
16 F2a301: F2a301.fasta
17 F5b8401: F5b8401.fasta
18 HS: HS_1.fasta HS_2.fasta HS_3.fasta
19 IAI1: IAI1.fasta
20 IAI39: IAI39.fasta
21 MG1655: MG1655.fasta
22 S88: S88_1.fasta S88_2.fasta S88_3.fasta S88_4.fasta
23 Sakai: Sakai.fasta
24 SE11: SE11.fasta
25 SMS35: SMS35_1.fasta SMS35_2.fasta SMS35_3.fasta
26 SSSs046: SSSs046.fasta
27 UMN026: UMN026_1.fasta UMN026_2.fasta
28 UTI89: UTI89.fasta
29 W3110: W3110.fasta
Plain Text
Tab Width: 4
Ln 1, Col 1
INS

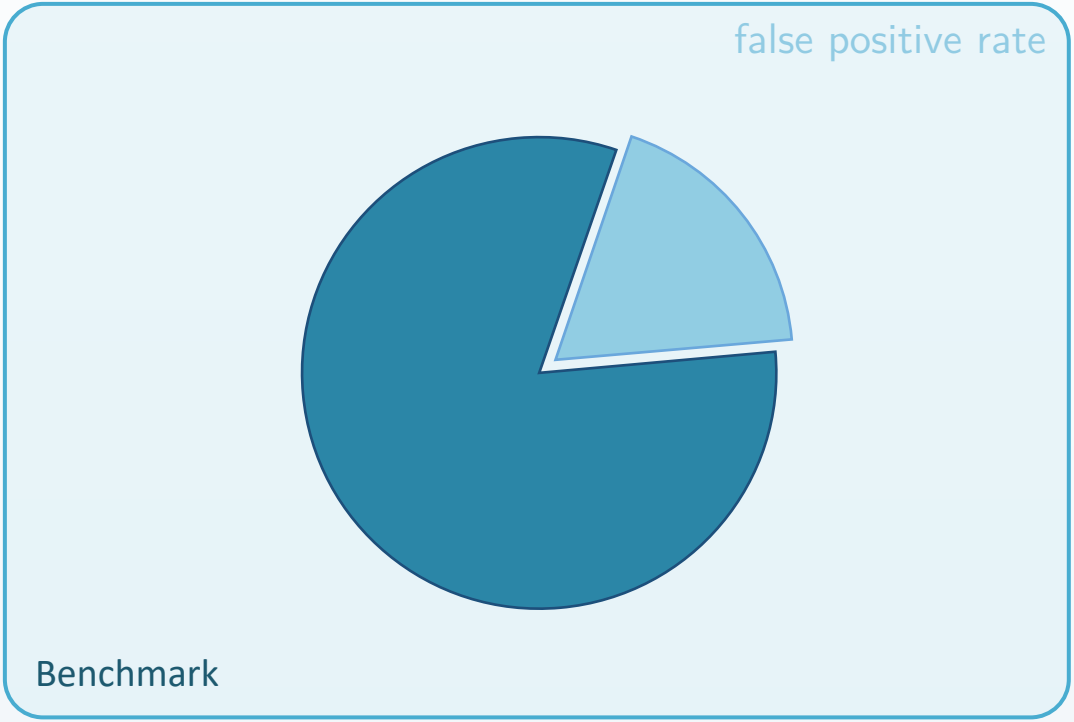

seqan/raptor
localhost:8000/tools/raptor/6807e7e7-566c-47a8-921b-2465b1e8fcd7
UNIVERSITÄT BIELEFELD
Technische Fakultät
Genome Informatics
Research
Studies
Internal
You are here: home » research » panbench
Raptor
Raptor CI passing
codecov 100%
install with bioconda
install with brew
from seqan/raptor
Last update: May 18, 2022
COMPLETE
6807e7e7-566c-47a8-921b-2465b1e8fcd7$ ./build/bin/raptor socks build ./names.txt --output ./index --kmer 20 --hash 2
--size 1k
Process finished with exit code 0
BACK
CANCEL
DOWNLOAD
A fast and space-efficient pre-filter for querying very large collections of nucleotide sequences
Download and Installation

Pangenomics Benchmark and Workbench

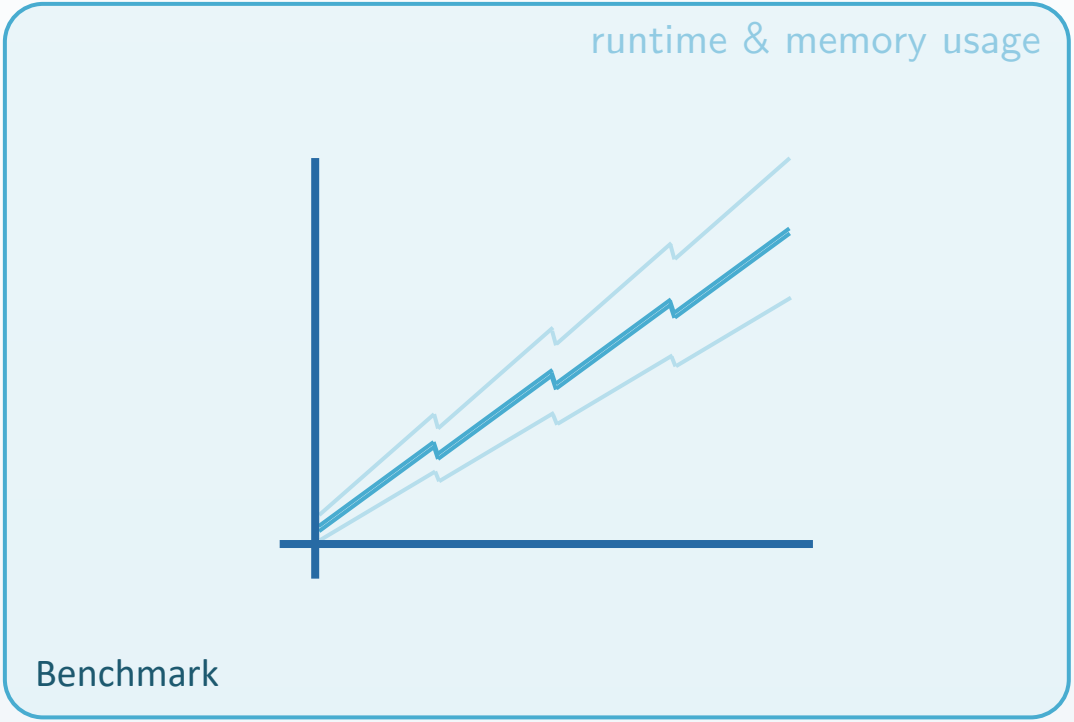
panbench.eu



Pangenomics Benchmark and Workbench



Pangenomics Benchmark and Workbench



❖ Special Thanks

Prof. Dr. Jens Stoye
Genome Informatics group
Graduate School “DILS”

Thank you for your attention!

andreas_rempel
andreas.rempel@uni-bielefeld.de

PANGAIA ALPACA