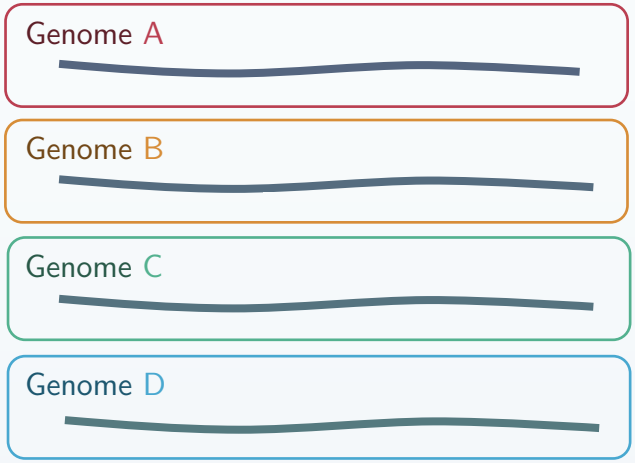
UNIVERSITÄT
BIELEFELD
Technische Fakultät

# Developing a standard interface for sets of *k*-mer sets index structures

Andreas Rempel

# Software for Computational Pangenomics

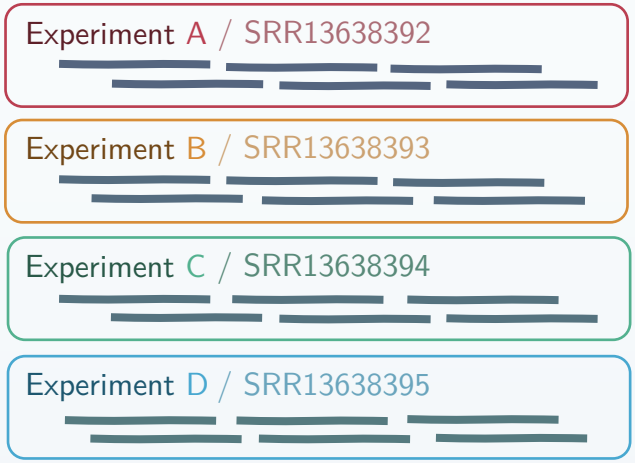
## ❖ Data structures




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

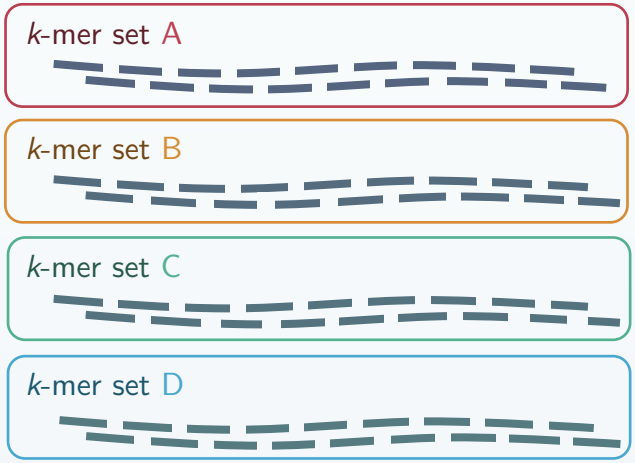
## ❖ Data structures




Computational Pangenomics

- 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

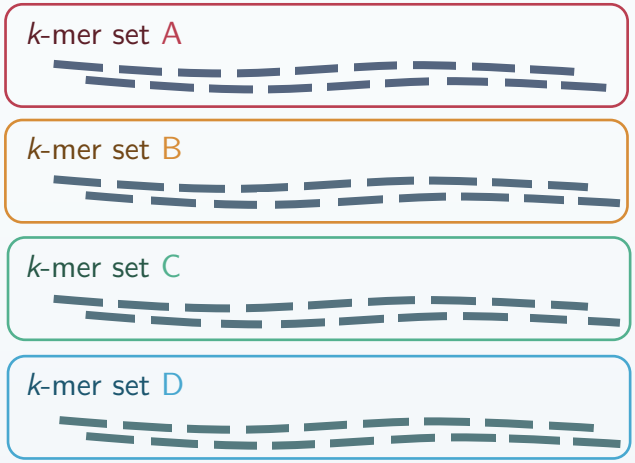
- Requires advanced algorithms and data structures

→ Sets of $k$-mer sets

**Idea: *k*-mers**
Split each 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

## ❖ Available tools

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

## ❖ Motivation

```
#my_bioinformatics_pipeline
./path/to/programA index ...
...
```

```
#my_bioinformatics_pipeline
./path/to/programB index ...
...
./path/to/programB query ...
...
```

switch between tools without having to rewrite your whole pipeline

# Software for Computational Pangenomics

## ❖ SeqOthello

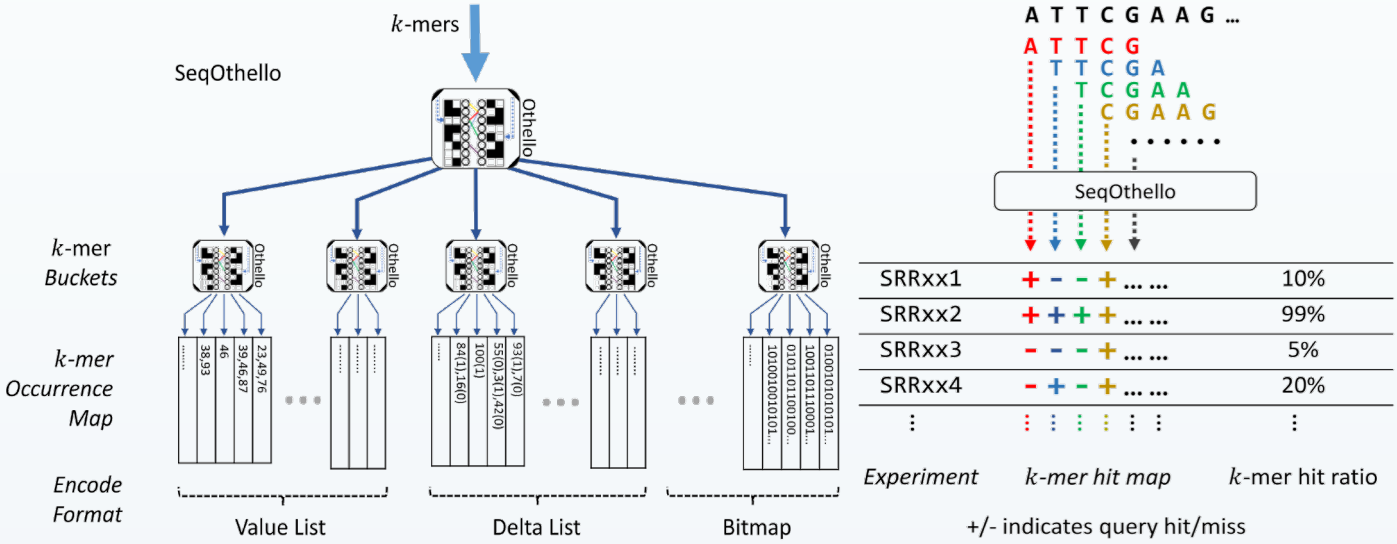



Figure: Lu, Liu, Zhang, Magner, Qian et. al.

## ❖ SeqOthello

```
jellyfish count
./build/bin/PreProcess
./build/bin/Group
./build/bin/Build
```

} construct index

```
./build/bin/Query
```

→ transcripts query

ultra-fast and memory-efficient indexing structure for RNA-seq experiments

https://github.com/LiuBioinfo/SeqOthello

## ❖ Mantis

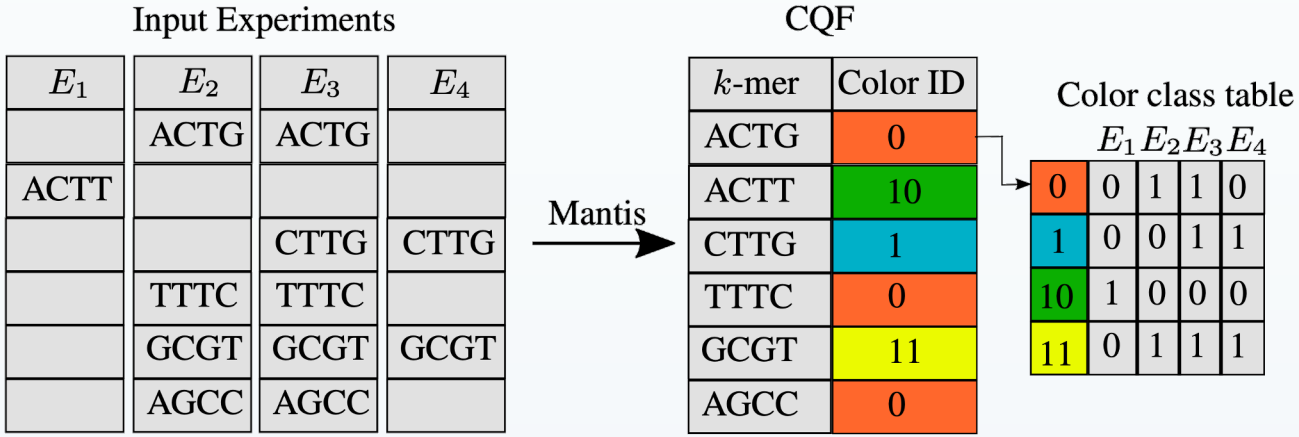



Figure: Pandey, Almodaresi, Bender, Ferdman, Johnson et. al.

## ❖ Mantis

```
squeakr count
./bin/mantis build        } construct index
./bin/mantis mst          → compress index
./bin/mantis query        → sequence query
```

fast, small & exact large-scale sequence-search index for raw-read experiments

https://github.com/splatlab/mantis

## ❖ BIGSI

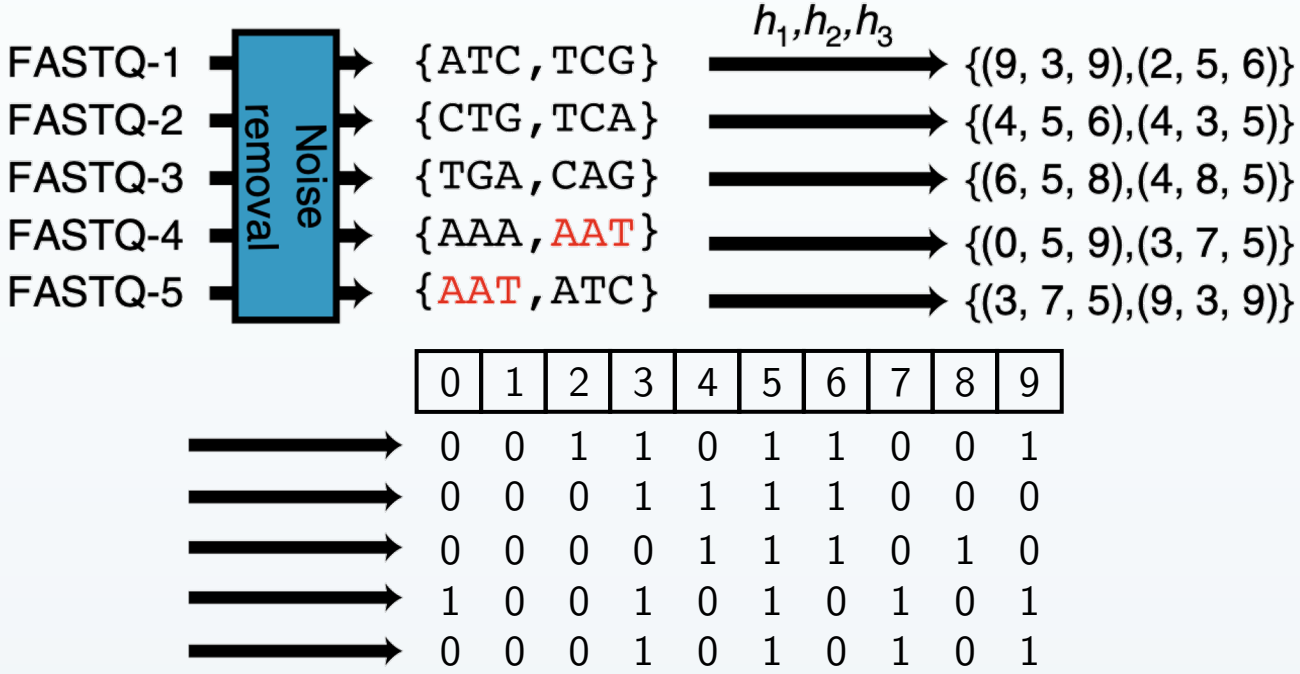



Figure: Bradley, Bakker, Rocha, McVean et. al.

## ❖ BIGSI

```
mccortex build
./bigsi bloom
./bigsi build
```

} construct index

```
./bigsi search
```

→ sequence query

bit-sliced signature index for ultra-fast search in bacterial and viral genomic data

https://github.com/iqbal-lab-org/bigsi

## ❖ COBS

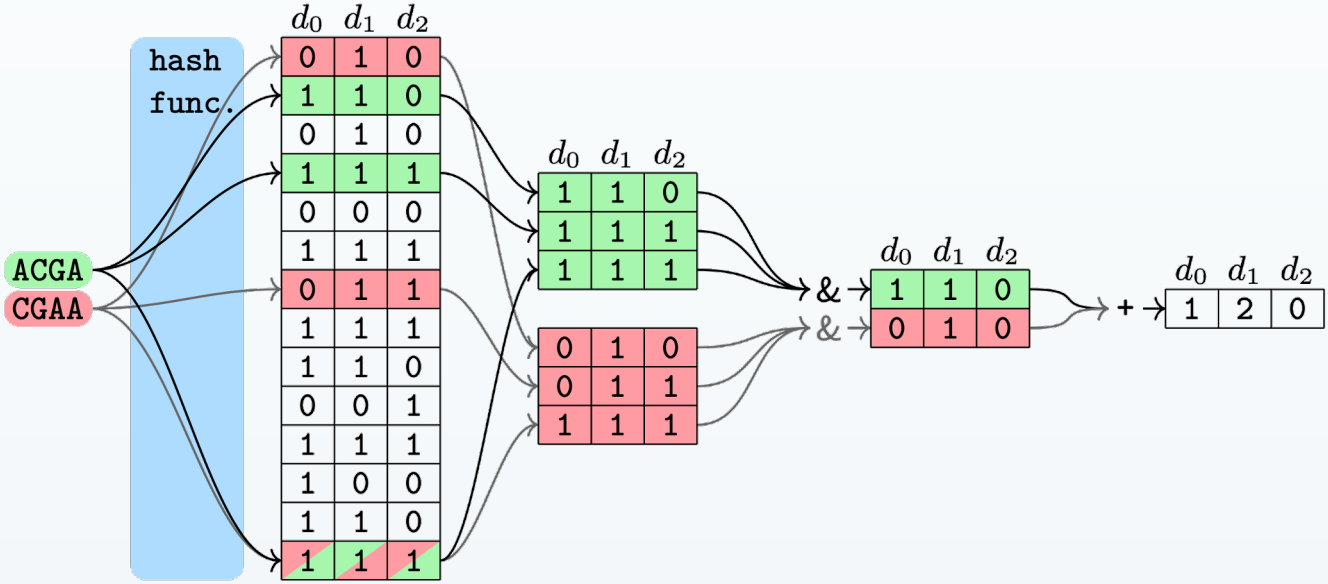



Figure: Bingmann, Bradley, Gauger et. al.

## ❖ COBS

| | | |
|---|---|---|
| `./src/cobs compact_construct` | | construct index |
| `./src/cobs query` | → | sequence query |

compact bit-sliced signature index for ultra-fast search in large-scale genomic data
https://github.com/bingmann/cobs

## ❖ Sequence Bloom Tree

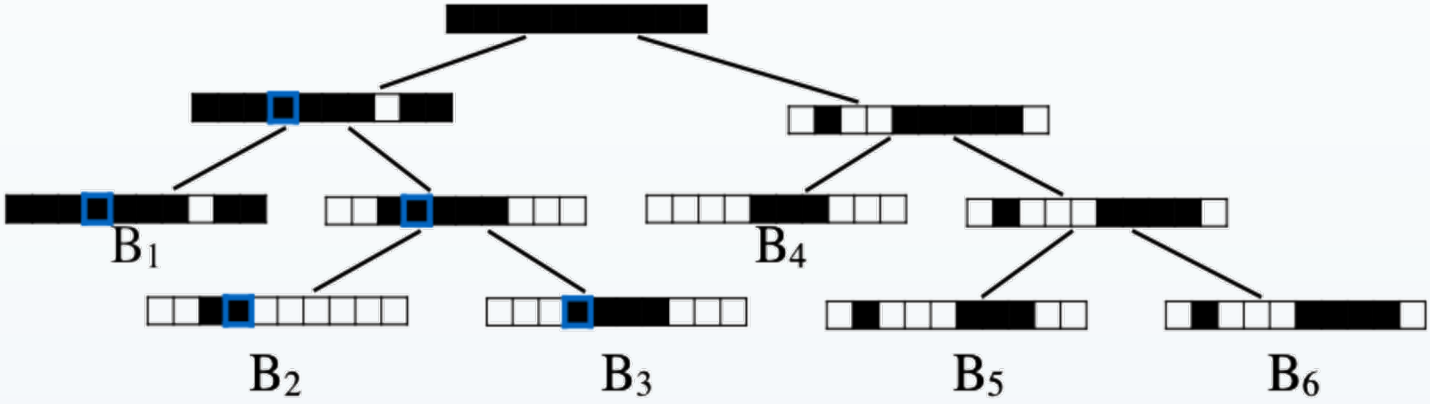



Figure: Solomon and Kingsford

## ❖ SBT/SSBT

```
./src/bt hashes
./src/bt count
./src/bt build
```

} construct index

```
./src/bt compress
```

→ compress index

```
./src/bt query
```

→ sequence query

bloom-tree based index and search for short-read sequencing experiments

https://github.com/Kingsford-Group/bloomtree | splitsbt

## ❖ HowDeSBT

```
./howdesbt makebf
./howdesbt cluster
./howdesbt build
```

} construct index

```
./howdesbt query
```

→ sequence query

improved index and search for short-read sequencing experiments

https://github.com/medvedevgroup/HowDeSBT

## ❖ Bifrost

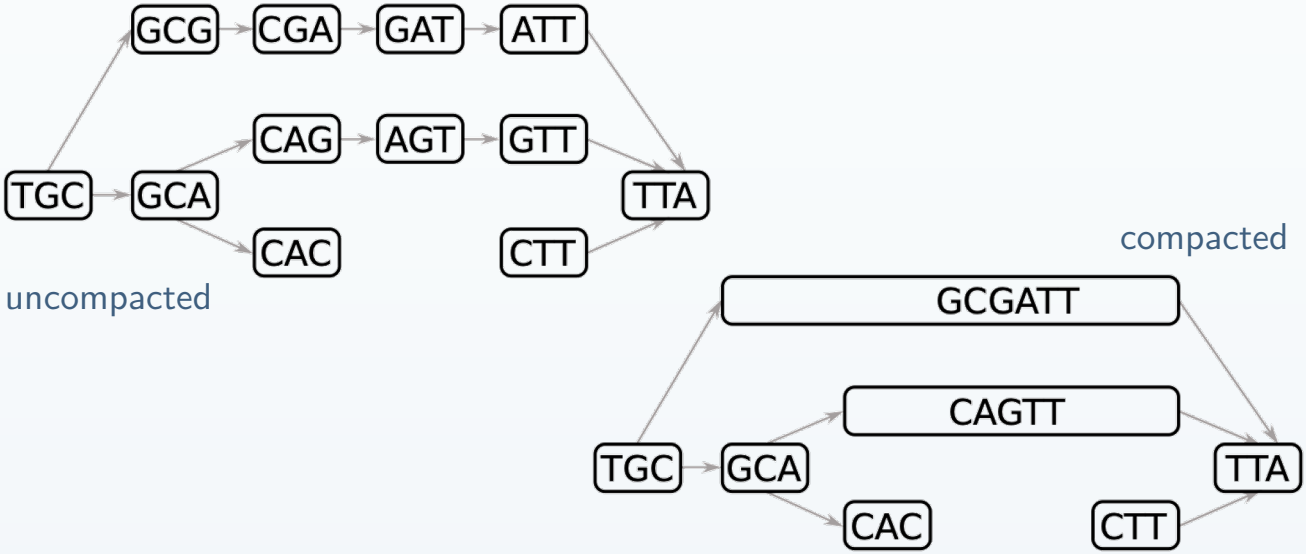



Figure: Holley and Melsted

## ❖ Bifrost

| | | |
|---|---|---|
| `./bifrost build` | → | construct index |
| `./bifrost update` | → | add sequences<br>update index |
| `./bifrost query` | → | sequence query |

parallel construction, indexing & querying of colored compacted de Bruijn graphs

https://github.com/pmelsted/bifrost

## ❖ COSMO/VARI-merge

| | | |
|---|---|---|
| `./cosmo-build` | → | construct index |
| `./pack-color` | → | compress index |
| `./vari-merge`<br>`./color-merge` | → | merge graphs<br>update index |

fast construction and merging of succinct colored de Bruijn graphs

https://github.com/cosmo-team/cosmo/tree/VARI | VARI-merge

## ❖ Cortex

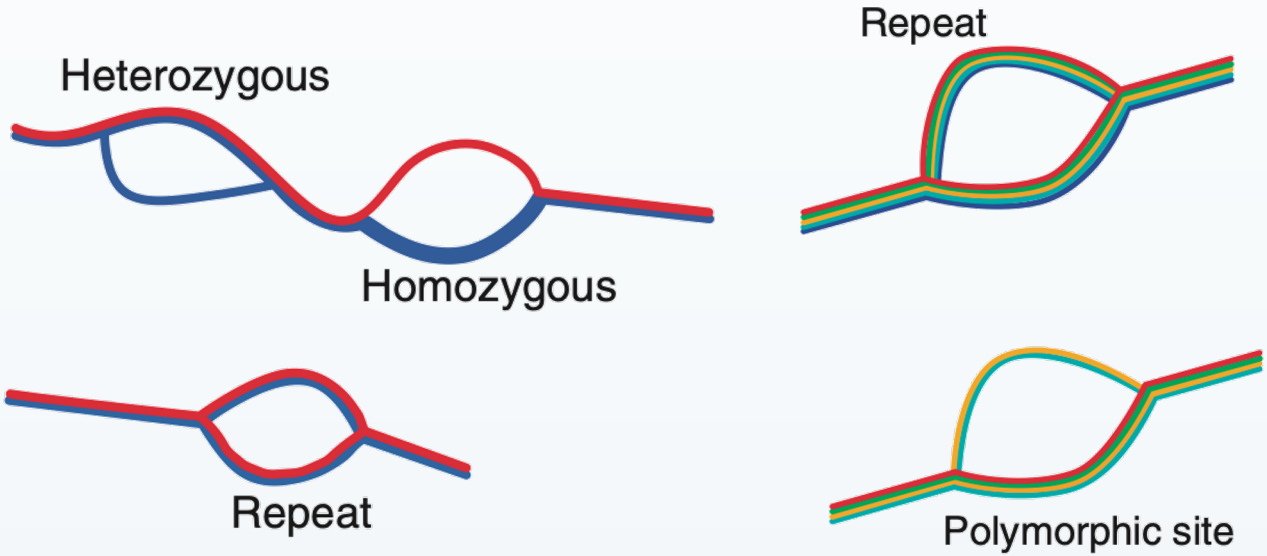



Figure: Iqbal, Caccamo, Turner, Flicek et. al.

## ❖ Cortex

```
./bin/cortex_var
      --se_list/pe_list
      --colour_list
```

construct index

```
./bin/cortex_var
      --detect_bubbles
      --output_bubbles
```

detect bubbles
call variants

efficient & low-memory software for consensus assembly and variation analysis

https://github.com/iqbal-lab/cortex

## ❖ McCortex

```
./bin/mccortex31
      - build
      - index
      - clean
      - correct
      - unitigs
      - contigs
```

construct index

clean/correct errors

output unitigs
assemble contigs

population de novo assembly and variant calling using linked de Bruijn graphs

https://github.com/mcveanlab/mccortex

## ❖ Motivation

```
#my_bioinformatics_pipeline
./path/to/programA index ...
...
```

```
#my_bioinformatics_pipeline
./path/to/programB index ?? build ??
...
```

switch between tools without having
to rewrite your whole pipeline
would be nice...

# Software for Computational Pangenomics

❖ **Step I**: Define a common standard interface

- Contact authors of existing tools for exchange of ideas
- Decide on a set of core features and consistent naming
- Update tools or write wrapper implementing the interface

| | | |
|---|---|---|
| **AllSomeSBT** | **Cortex** | **VARI** |
| **HowDeSBT** | **McCortex** | **VARI-merge** |
| **Themisto** | **BIGSI** | **REINDEER** |
| **kmtricks** | **COBS** | **GATB-core** |

# Software for Computational Pangenomics

Core interface:

- construct index
  - input: set of sequences[1], each with a distinct color
  - output: index
- add | remove
  - input: set of sequences, index
  - output: index
- read/write index
  - input: index
  - output: proprietary format | gfa | fasta | *k*-mer matrix
- *k*-mer query
  - input: *k*-mer
  - output: set of colors
- color query
  - input: color
  - output: all *k*-mers with this color

[1] sequence = assembled genome | read set | set of *k*-mers

# Software for Computational Pangenomics

Optional interface:

- general query
  - input: set of sequences, set of colors | quorum[2]
  - output: all hits matching the query
- forward/backward extension
  - input: $k$-mer / unitig
  - output: neighboring $k$-mers / unitigs in De Bruijn Graph
- abundance query
  - input: $k$-mer or longer sequence (e.g. transcript), color
  - output: abundance of the sequence in this color

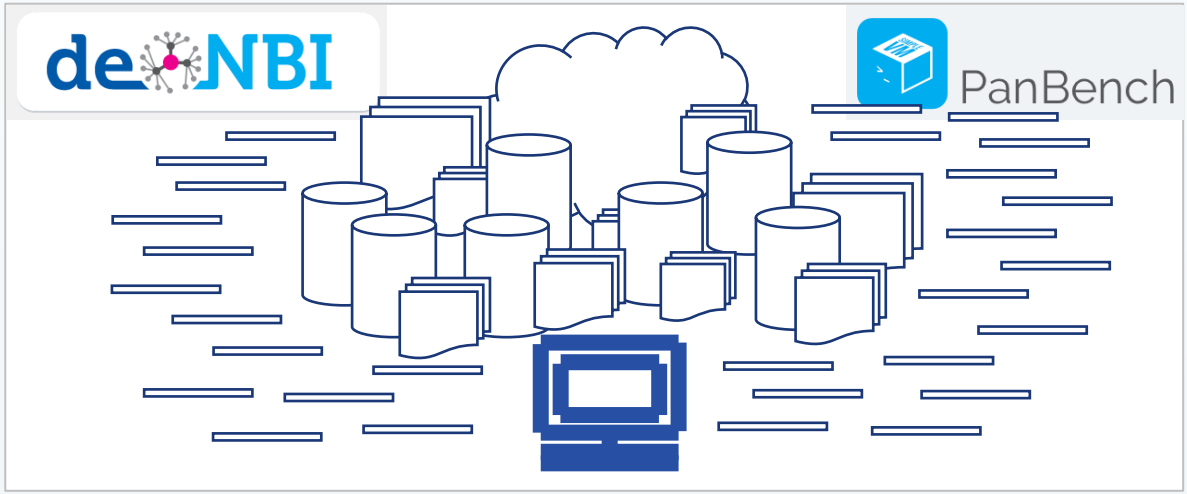
Additional interface:

- use case specific features
  - genome assembly, e.g. filter erroneous $k$-mers, assemble contigs
  - phylogenomics, e.g. output list of splits for the $k$-mer/color sets
  - and more ...

[2] quorum = required min. ratio of colors that have a match

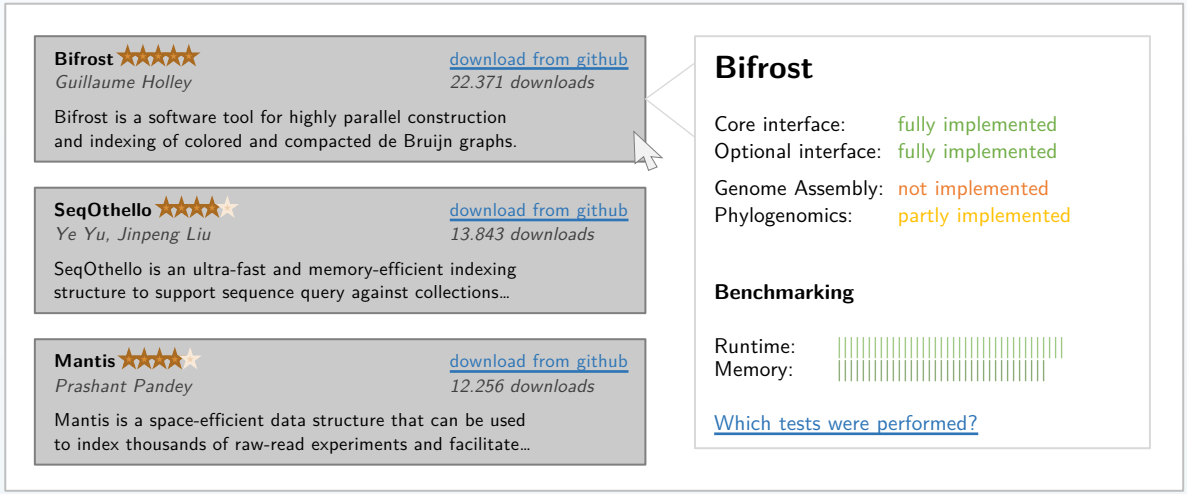
❖ **Step II**: Set up an automated test environment

- Select suitable test data sets and criteria for an evaluation
- Implement a benchmarking platform on the de.NBI cloud



## ❖ **Step III**: Perform the actual benchmarking study

- Contact authors to participate in a benchmark of tools
- Provide an open catalog of evaluated tools with results



## ❖ Conclusion

- Pangenomes can hold a variety of information
- Many different software tools are available
- Establishing a standard interface has the potential to enrich comparative genomics and genome research

## ❖ Special Thanks

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

**Thank you for your attention!**

Contact: andreas.rempel@uni-bielefeld.de