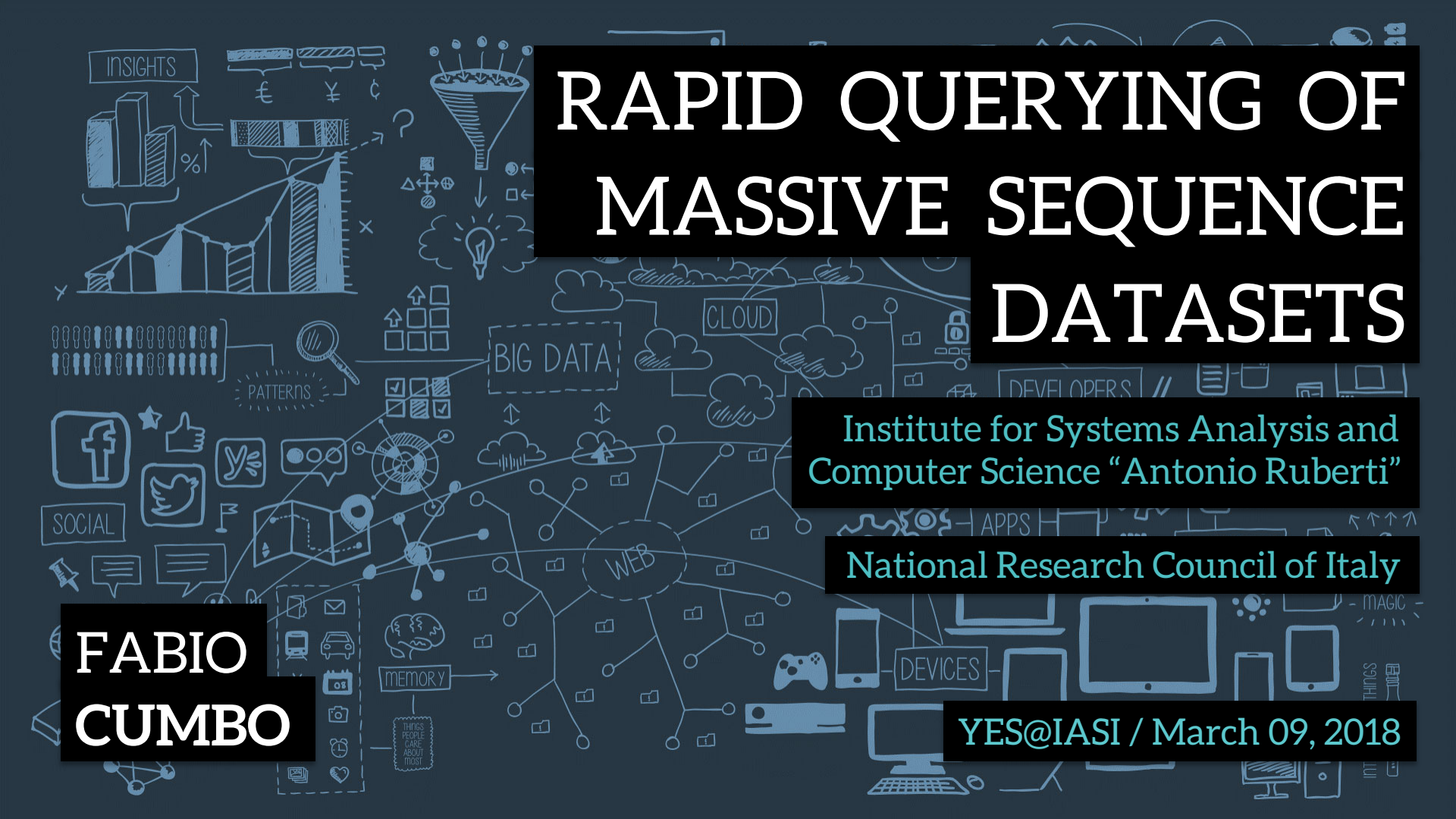


RAPID QUERYING OF MASSIVE SEQUENCE DATASETS

Institute for Systems Analysis and Computer Science "Antonio Ruberti"

National Research Council of Italy

FABIO CUMBO

YES@IASI / March 09, 2018

SUMMARY

While advances of sequencing technologies have opened unprecedented opportunities for all of Life Sciences, these developments have inadvertently created new challenges. Initially ignored, they continued to accumulate and today these challenges are complex enough to block the progress of biomedical research.

In this proposal we are addressing one of these roadblocks–ability to **rapidly query already existing and future (bigger) sequence data**.

We argue that effective use of existing data can:

1. produce new knowledge without the need for additional experiments,
2. guide smarter design of future experiments, and
3. open existing datasets to other disciplines beyond the life sciences, fostering truly cross-disciplinary collaborations.

To achieve this, we will leverage the latest developments in data structure and algorithm research to create a general framework for processing and querying very large datasets using **Sequence Bloom Trees** [1].

ROMA TRE
UNIVERSITÀ DEGLI STUDI
DEPARTMENT OF ENGINEERING

National Research Council of Italy
Institute for Systems Analysis and Computer Science "Antonio Ruberti"

PENNSTATE
1855
NEKRUTENKO LAB AND
MEDVEDEV LAB

1. Solomon, Brad, and Carl Kingsford. *"Fast search of thousands of short-read sequencing experiments."* Nature biotechnology 34.3 (2016): 300.

Basics

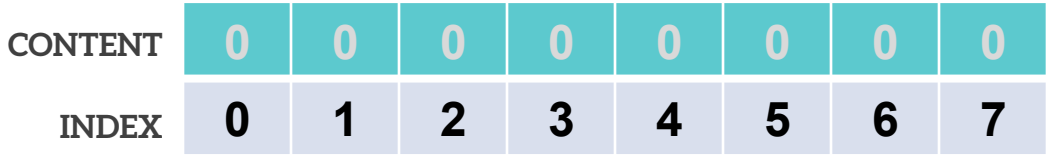
Bloom Filter

BLOOM FILTER

A Bloom filter [1, 2] is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set.

It is a **probabilistic data structure**: it tells us that the element either ***definitely is not*** in the set or ***may be*** in the set.

BIT VECTOR



HASH FUNCTIONS

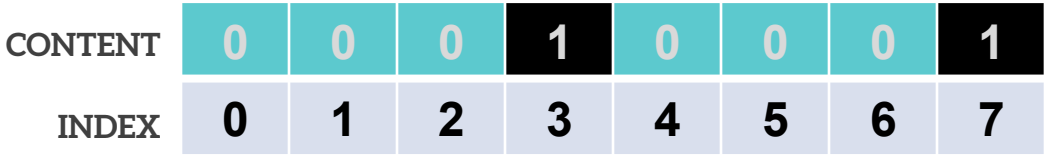
1. Bloom, Burton H. *"Space/time trade-offs in hash coding with allowable errors."* Communications of the ACM13.7 (1970): 422-426.
2. Broder, Andrei, and Michael Mitzenmacher. *"Network applications of bloom filters: A survey."* Internet mathematics1.4 (2004): 485-509.

BLOOM FILTER

A Bloom filter [1, 2] is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set.

It is a **probabilistic data structure**: it tells us that the element either ***definitely is not*** in the set or ***may be*** in the set.

BIT VECTOR



HASH FUNCTIONS

ADD TO BLOOM FILTER

GTGGAAGTTCTTAGGGCATGGCAAAGAGTCAGAATTTGACACTG

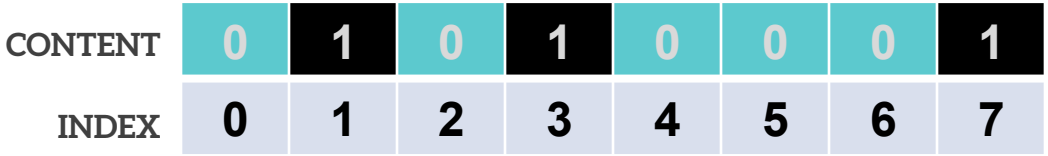
1. Bloom, Burton H. *"Space/time trade-offs in hash coding with allowable errors."* Communications of the ACM13.7 (1970): 422-426.
2. Broder, Andrei, and Michael Mitzenmacher. *"Network applications of bloom filters: A survey."* Internet mathematics1.4 (2004): 485-509.

BLOOM FILTER

A Bloom filter [1, 2] is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set.

It is a **probabilistic data structure**: it tells us that the element either ***definitely is not*** in the set or ***may be*** in the set.

BIT VECTOR



HASH FUNCTIONS

ADD TO BLOOM FILTER

`ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCCGA`

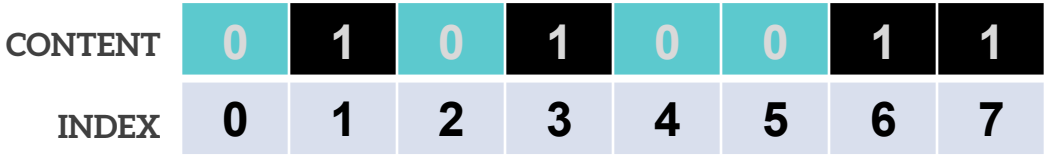
1. Bloom, Burton H. *"Space/time trade-offs in hash coding with allowable errors."* Communications of the ACM13.7 (1970): 422-426.
2. Broder, Andrei, and Michael Mitzenmacher. *"Network applications of bloom filters: A survey."* Internet mathematics1.4 (2004): 485-509.

BLOOM FILTER

A Bloom filter [1, 2] is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set.

It is a **probabilistic data structure**: it tells us that the element either ***definitely is not*** in the set or ***may be*** in the set.

BIT VECTOR



HASH FUNCTIONS

ADD TO BLOOM FILTER

`CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATAGCTC`

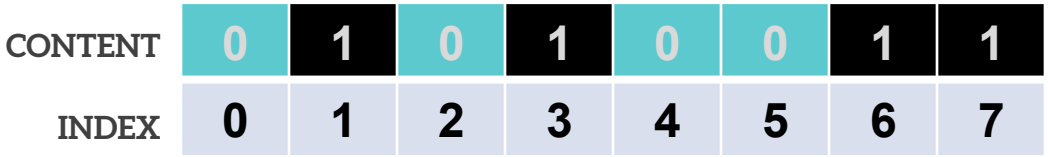
1. Bloom, Burton H. *"Space/time trade-offs in hash coding with allowable errors."* Communications of the ACM13.7 (1970): 422-426.
2. Broder, Andrei, and Michael Mitzenmacher. *"Network applications of bloom filters: A survey."* Internet mathematics1.4 (2004): 485-509.

BLOOM FILTER

A Bloom filter [1, 2] is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set.

It is a **probabilistic data structure**: it tells us that the element either ***definitely is not*** in the set or ***may be*** in the set.

BIT VECTOR



HASH FUNCTIONS

REMOVE FROM BLOOM FILTER

CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATAGCTC

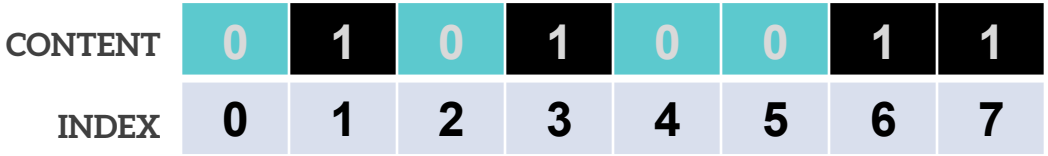
1. Bloom, Burton H. *"Space/time trade-offs in hash coding with allowable errors."* Communications of the ACM13.7 (1970): 422-426.
2. Broder, Andrei, and Michael Mitzenmacher. *"Network applications of bloom filters: A survey."* Internet mathematics1.4 (2004): 485-509.

BLOOM FILTER

A Bloom filter [1, 2] is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set.

It is a **probabilistic data structure**: it tells us that the element either ***definitely is not*** in the set or ***may be*** in the set.

BIT VECTOR



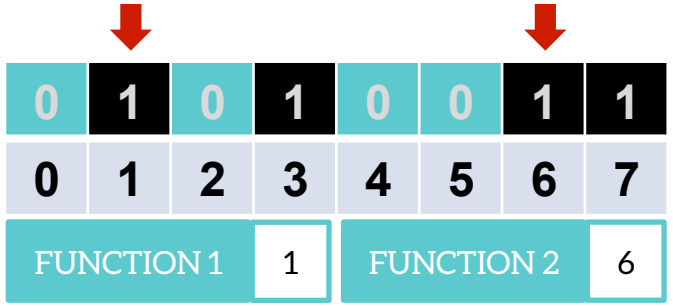
HASH FUNCTIONS

REMOVE FROM BLOOM FILTER NOT ALLOWED

CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATAGCTC

1. Bloom, Burton H. *"Space/time trade-offs in hash coding with allowable errors."* Communications of the ACM13.7 (1970): 422-426.
2. Broder, Andrei, and Michael Mitzenmacher. *"Network applications of bloom filters: A survey."* Internet mathematics1.4 (2004): 485-509.

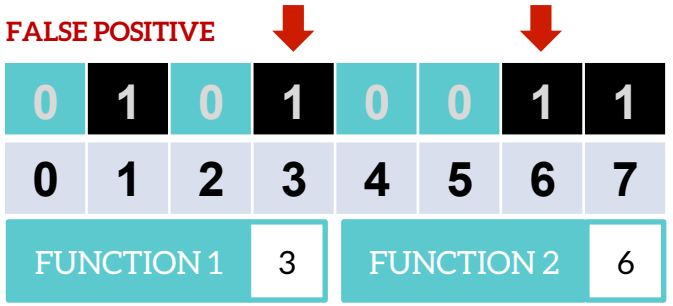
MEMBERSHIP



CCTGGAGGGTGGCCCCACCGGCCGAGACA
GCGAGCATATAGCTC

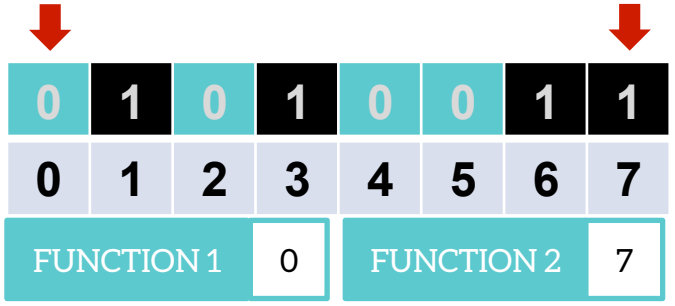
PREVIOUSLY INSERTED → *MAY BE*

FALSE POSITIVE



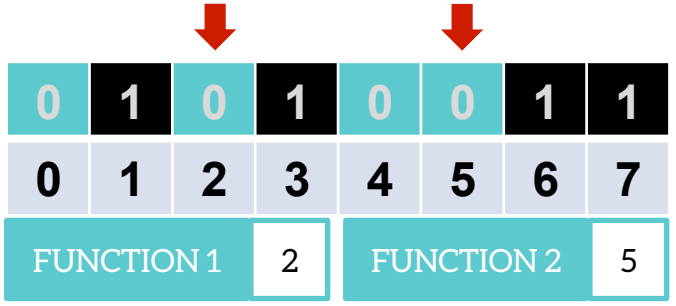
AAGCTCGGGAGGTGGCCAGGCGGCAGGAA
GGCGCACCCCCCCAG

NOT INSERTED BEFORE → *MAY BE*



CTGCAGGAACTTCTTCTGGAAGACCTTCT
CCTCCTGCAAATAAA

NOT INSERTED BEFORE → *DEFINITELY IS NOT*



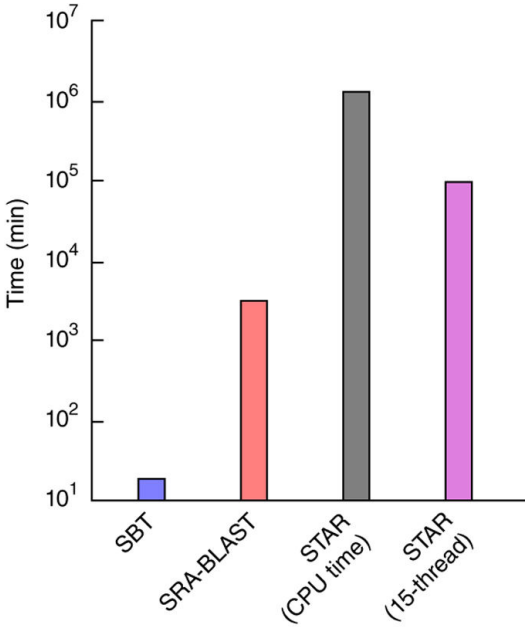
ACCCAGTCCTTCTTCTGGAAGACCTTGCT
CCAACTGCCAATAAT

NOT INSERTED BEFORE → *DEFINITELY IS NOT*

Methods

Sequence Bloom Tree

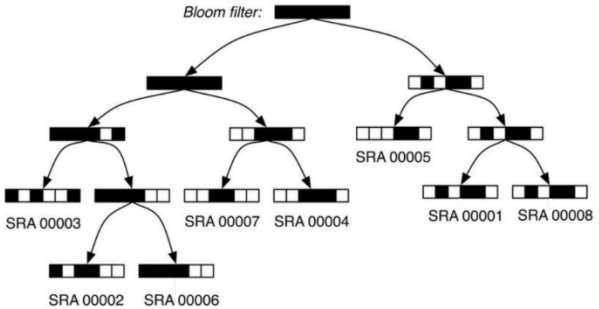
SEQUENCE BLOOM TREE



Estimated running times of search tools for one transcript
SBT is 162 times faster than existing approaches

Sequence Bloom Tree (SBT) [1] is a method for querying thousands of short-read sequencing experiments by sequence. It create a hierarchy of **compressed bloom filters** [2], which efficiently store a set of items. Each bloom filter contains the **set of k-mers present within a subset of the sequencing experiments**.

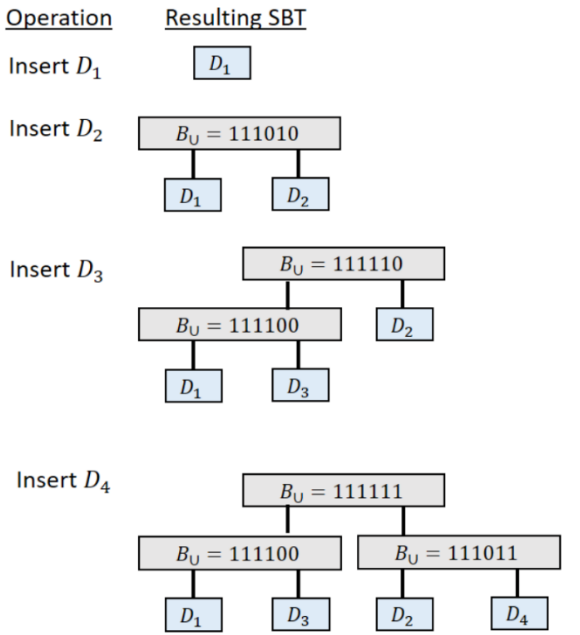
SBTs are binary trees in which the sequencing experiments are associated with leaves, and each node ***v*** of the SBT contains a bloom filter that contains the set of k-mers present in any read in any experiment in the subtree rooted at ***v***.



1. Solomon, Brad, and Carl Kingsford. *"Fast search of thousands of short-read sequencing experiments."* Nature biotechnology 34.3 (2016): 300.
2. Raman, Rajeev, Venkatesh Raman, and S. Srinivasa Rao. *"Succinct indexable dictionaries with applications to encoding k-ary trees and multisets."* Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2002.

SBT CONSTRUCTION

https://github.com/Kingsford-Group/bloomtree
https://github.com/medvedevgroup/bloomtree-allsome



{D1 = 111000, D2 = 111010, D3 = 000100, D4 = 000011}

A SBT [1] is a binary tree that is built by repeated insertion of sequencing experiments.

Given a (possibly empty) SBT ***T***, a new sequencing experiment ***s*** can be inserted into ***T*** by first computing the bloom filter ***b(s)*** of the k-mers present in ***s*** and then walking from the root along path to the leaves and inserting ***s*** at the bottom of ***T*** in the following way:

When at node u:

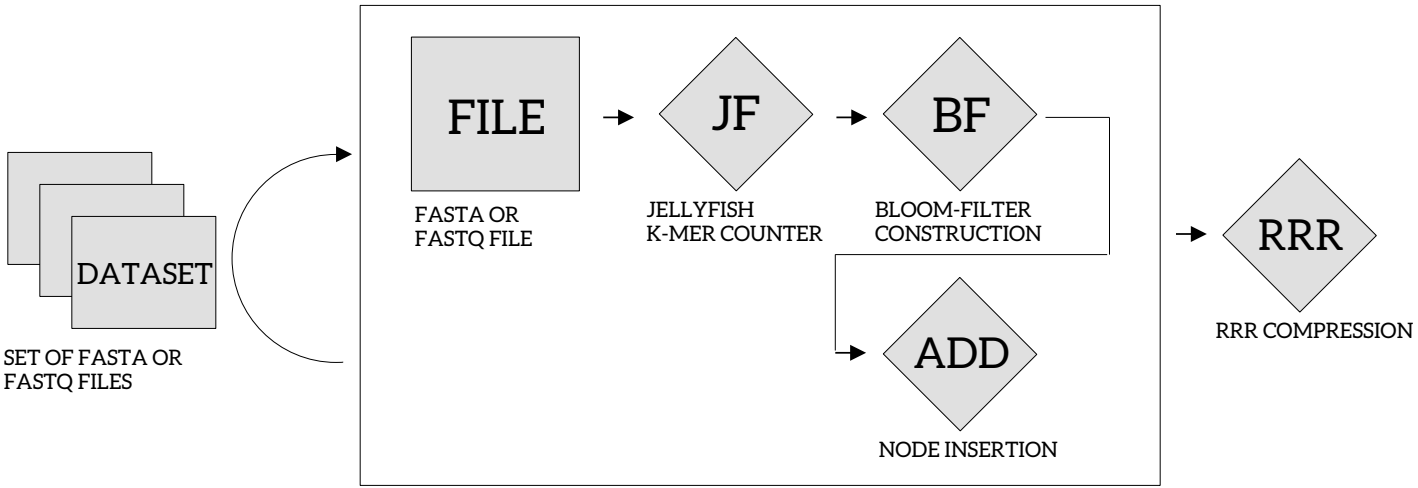
- if ***u*** has a single child, a node representing ***s*** (and containing ***b(s)***) is inserted as ***u***'s second child.
- if ***u*** has two children, ***b(s)*** is compared among against the bloom filters ***b(left(u))*** and ***b(right(u))*** of the ***left(u)*** and ***right(u)*** children of ***u***.
    - the child with the more similar filter under the Hamming distance between the filters becomes the current node, and the process is repeated.
- if ***u*** has no children, ***u*** represents a sequencing experiment ***s'***. In this case, a new union node ***v*** is created as a child of ***u***'s parent.
    - this new node has two children: ***u*** and a new node representing ***s***.

1. Solomon, Brad, and Carl Kingsford. *"Fast search of thousands of short-read sequencing experiments."* Nature biotechnology 34.3 (2016): 300.

TREE CONSTRUCTION SCHEMA

The construction of the SBT involves three major tasks:

1. creation of the bloom filters for each of the experiments included at its leaves;
2. construction of the tree and internal bloom filters;
3. RRR compression [1] of each of the filters.



1. Raman, Rajeev, Venkatesh Raman, and S. Srinivasa Rao. *"Succinct indexable dictionaries with applications to encoding k-ary trees and multisets."* Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2002.

QUERYING THE TREE

Given a query sequence q and a SBT T, it is possible to query the SBT in the following way:

- the sequencing experiments (at the leaves) that contain q can be found by breaking q into its constituent set of k-mers K_q and then flowing these k-mers over T starting from the root;

- at each node u, the bloom filter $b(u)$ at that node is queried for each of the k-mers in K_q.

    - If more than $\theta |K_q|$ k-mers are reported to be present in $b(u)$, the search proceeds to all of the children of u, where θ is a cutoff between 0 and 1 governing the stringency required of the match. **The parameter θ governs a query's tolerance to errors**.

Note:
A general SBT query with N k-mers and k-mer size k tolerates at least $N(1 - \theta)/k$ k-mer mismatches, between the query and the stored data.

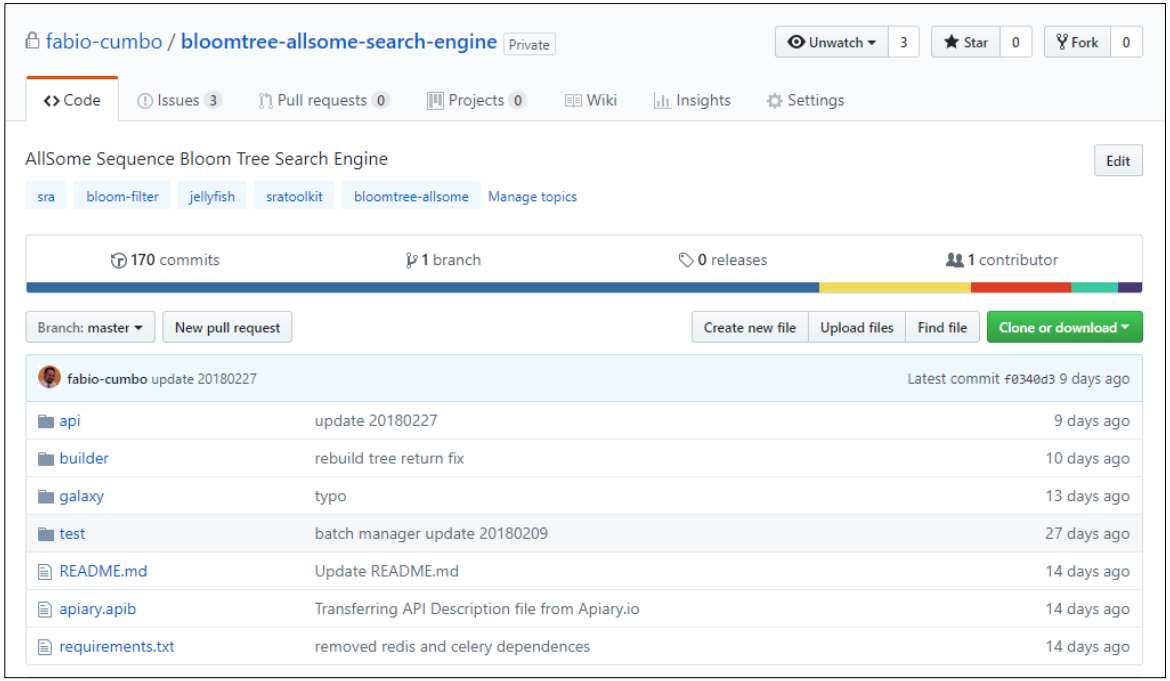
Solomon, Brad, and Carl Kingsford. *"Fast search of thousands of short-read sequencing experiments."* Nature biotechnology 34.3 (2016): 300.

Results

SBT Builder, APIs, Galaxy

GITHUB

GitHub repository activity
September 22 – Ongoing

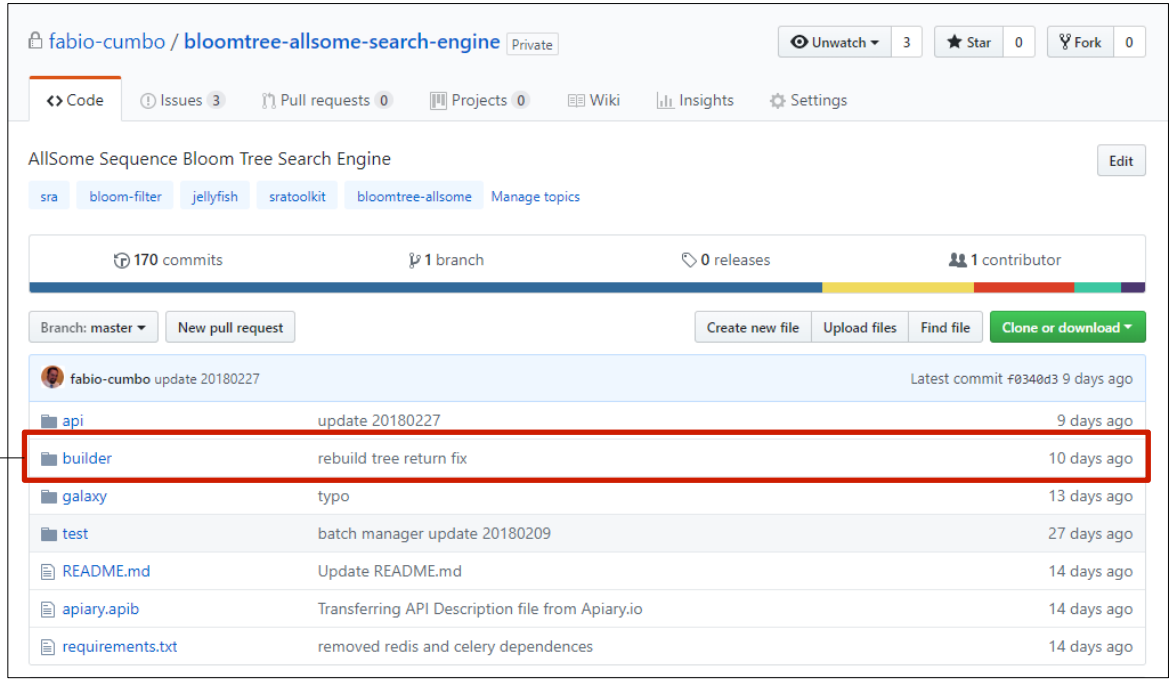


This GitHub repository is composed of two components plus a tool for Galaxy [1]

1. Afgan, Enis, et al. *"The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update."* Nucleic acids research 44.W1 (2016): W3-W10.

GITHUB

GitHub repository activity
September 22 – Ongoing



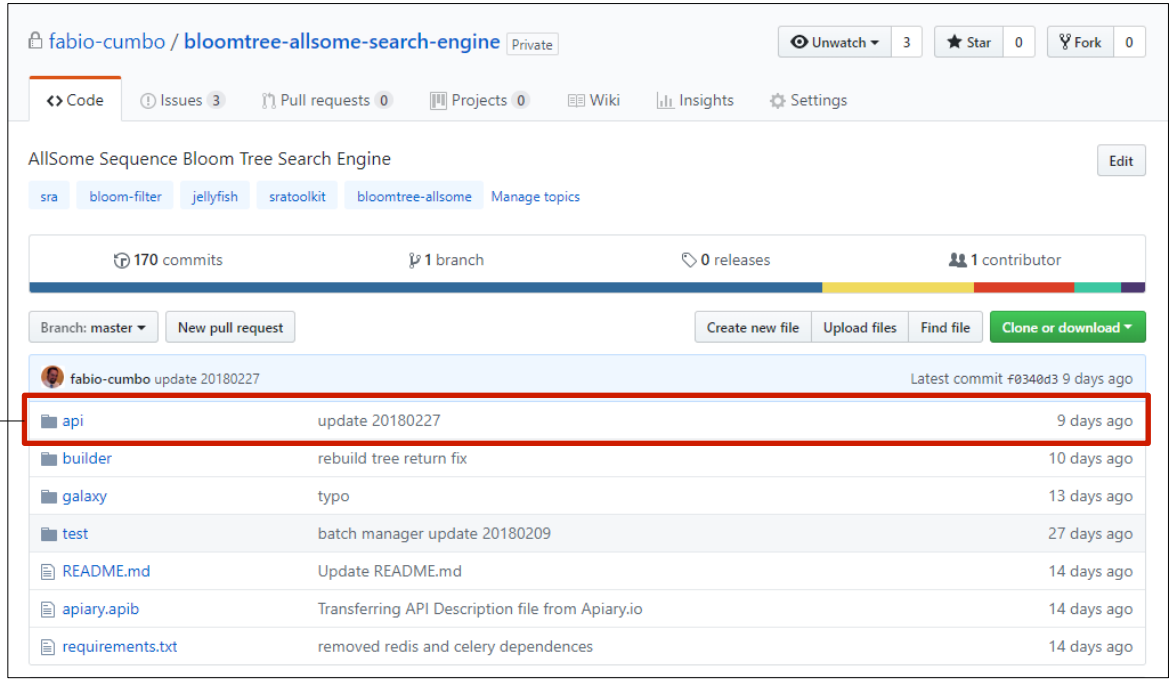
This GitHub repository is composed of two components plus a tool for Galaxy [1]

Builder: package to automatically download a list of predefined experiments (accession numbers) from SRA (Sequence Read Archive – NCBI) or ENA (European Nucleotide Archive – EBI) and build an SBT.

1. Afgan, Enis, et al. *"The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update."* Nucleic acids research 44.W1 (2016): W3-W10.

GITHUB

GitHub repository activity
September 22 – Ongoing



This GitHub repository is composed of two components plus a tool for Galaxy [1]

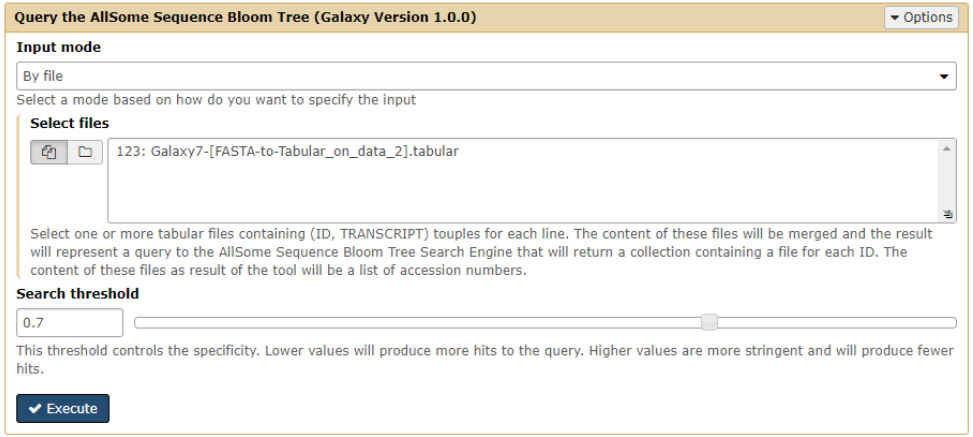
API: REST APIs (FLASK) to interact (submit queries and retrieve the corresponding results) with a list of SBTs http://nn14.galaxyproject.org:8080/

Documentation: https://sbtas.docs.apiary.io/

1. Afgan, Enis, et al. *"The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update."* Nucleic acids research 44.W1 (2016): W3-W10.

GALAXY TOOL

https://usegalaxy.org/
https://test.galaxyproject.org/
https://toolshed.g2.bx.psu.edu/view/fabio/sbtas_se/



The AllSome Sequence Bloom Tree Search Engine is a fast querying tool to identify all publicly available sequenced samples which express a transcript of interest.

Example

The input for this tool is a list of (ID, TRANSCRIPT) touples, one for each line, in a tab delimited format:

```
id0  CCAACCAAAGGGAAAACTTTTTTCCGACTTTGGCCTAAAGGGTTTAACGGCCAAGTCAGAAGGGAAAAAGTTGCGCCA
id1  TTAATGACAGGGCCACATGATGTGAAAAAAAATCAGAAACCGAGTCAACGTGAGAAGATAGTACGTACTACCGCAAAT
...
idn  CAATTAATGATAAATATTTTATAAGGTGCGGAAATAAAGTGAGGAATATCTTTTAAATTCAAGTTCAATTCTGAAAGC
```

The ID can contain alphanumeric characters in addition to spaces, dots, dashes, and round and square brackets. Any additional characters will be trimmed out.

The output of the tool is a collection that contains a file for each ID with a list of accession numbers representing the samples that express one particular transcript.

Notes

This Galaxy tool has been developed by Fabio Cumbo.

Please visit this GithHub repository for more information about the AllSome Sequence Bloom Tree Search Engine

Citations Show BibTeX

Sun, Chen and Harris, Robert S. and Chikhi, Rayan and Medvedev, Paul (2016). AllSome Sequence Bloom Trees. [doi:10.1101/090464][Link]

1. Afgan, Enis, et al. *"The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update."* Nucleic acids research 44.W1 (2016): W3-W10.

CONCLUSIONS

EXPERIMENTS

- XSEDE accepted our proposal and allocated us **50TB** of storage and **1 year of computational time**.
- We started building a SBT from **3474 RNA-Seq samples of Drosophila Melanogaster** retrieved from the **Sequence Read Archive**.
- The size of the original samples is between **1 and 10 GB**
- The final amount of storage space for the compressed tree is **340GB**
    - Considering also the uncompressed tree, the total amount of used space is **~2TB**
- The computational time to query this SBT is **~10 minutes**

FUTURE STEPS

- Create additional SBTs for other organisms / experiments.
- Deletion operation is not allowed in bloom filters.
    - Do we really need this operation?
- The structure of an AllSome Sequence Bloom Tree [1] can not be modified.
    - How to update the tree without rebuilding its nodes and its topology?
- The result of a query is a list of experiments.
    - How to sort the result and associate a score for each experiment?

1. Sun, C., Harris, R. S., Chikhi, R., & Medvedev, P. (2017, May). *"Allsome sequence bloom trees."* In International Conference on Research in Computational Molecular Biology (pp. 272-286). Springer, Cham.

Fabio Cumbo
http://cumbo.me/
fabio.cumbo@iasi.cnr.it

Thank You

« *This work will make it possible to search against all available sequence data. The implications of this simple idea are enormous. Just as modern search engines allow rapidly navigate the web our system will unify sequence data across all branches of Life Sciences to make it universally accessible.* »