Modeling short read experiments for reference-free appilcations

FABIO CUMBO

PROPOSAL 1 Represent an experiment as a bit vector

PROCEDURE

1. Be **`S`** a set of experiments
2. Be **`Ei`** an experiment with **`i = 1..N`** with **`N`** total number of experiments in the set **`S`**
3. For each experiment **`Ei`**
    - Extract all the minimizers with length **`p`**
        - Be **`M`** the set of minimizers
    - Extract the k-mers with length **`p`** with a **high coverage** (threshold)
        - Be **`K`** the set of k-mers
    - Build a set **`U`** equals to the intersection of **`M`** and **`K`**
        - The set **`U`** in composed of **relevant minimizers** only
    - Create a bit vector **`B`** with length **`4^p`**, one position for each k-mer/ minimizer
        - Set to **`1`** the positions in the set **`B`** related to the elements in **`U`**
        - Set all the other positions to **`0`**
    - A bit vector is a point in a **`N`**-dimensional space

PROPOSAL 1 Represent an experiment as a bit vector

DATASET With `p = 10` the bit vectors have a length of `1.048.576`

0	1	1	0	1	0	...	0
0	1	2	3	4	5	...	~1M

EXPERIMENT 1

1	1	1	0	0	0	...	1
0	1	2	3	4	5	...	~1M

EXPERIMENT 2

THE KMERS/MINIMIZERS ARE ALPHABETICALLY SORTED

0	0	1	1	0	1	...	0
0	1	2	3	4	5	...	~1M

EXPERIMENT n

PROPOSAL 2 Represent an experiment as a bit vectors matrix

PROCEDURE

1. Be **S** a set of experiments
2. Be **Ei** an experiment with **i = 1..N** with **N** total number of experiments in the set **S**
3. For each experiment **Ei**
    - For each read **Rw** in Ei with **w = 1..Z** with **Z** maximum number of reads in the experiment **Ei**
        - Extract the frequency of the nucleotides in **Rw: [f(A), f(C), f(G), f(T)]** and normalize them by the length of the read **Rw**
        - [optional] replace the previous step with the frequency of k-mers with **k > 1**
    - It will result in a matrix **M** with w rows and 4 columns (or more, accordingly to the chosen **k**)
        - Every row in **M** is a point in a **N**-dimensional space

PROPOSAL 2 Represent an experiment as a bit vectors matrix

DATASET

With `p = 3` the bit vectors have a length of **64**

A single experiment is composed of a set of vectors, one for each read

4	3	8	9	13	7	...	8
0	1	2	3	4	5	...	64

READ 1 - READ LENGTH: 200

24	6	2	43	2	1	...	9
0	1	2	3	4	5	...	64

READ 2 - READ LENGTH: 200

9	5	26	51	4	23	...	3
0	1	2	3	4	5	...	64

READ n - READ LENGTH: 200

THE KMERS/MINIMIZERS ARE ALPHABETICALLY SORTED

LET'S TRY TO CLUSTERIZE

Unsupervised Learning

We do not know a priori how many clusters we need.

Hierarchical clustering (AGNES), Density-Based Spatial Clustering (DBSCAN), etc.

CONSIDERATIONS

Clustering bit vectors for the proposal 1 would be relatively easy.

For the proposal 2 we have a huge amount of bit vectors.

Is it possible to reduce the number of bit vectors for the proposal 2?

INTRA-CLUSTERING

Proposal 2

For each experiment:

1. Cluster the bit vectors (reads)
2. For each cluster take trace of the centroids computed as the euclidean distances between all the points in the clusters

INTER-CLUSTERING

Proposal 1 and 2

An experiment is represented by:

1. A bit vector (Proposal 1)
2. A set of centroids (Proposal 2)

Cluster all the bit vectors and take trace of the centroids computed again as the euclidean distances between all the points in the clusters

HOW TO QUERY

Represent your query as a bit vector using the same procedure previously described for the Proposal 1.

Now the query is represented by a point in a N-dimensional space.

Compute the euclidean distance between this bit vector and the previously extracted centroids.

- **The nearest centroid represents a set of candidate experiments in which the query could appear.**
- **Sorting the centroids according to their distance to query allow us to assign a score (hit) to every experiment**

REFERENCE-FREE ALIGNMENT AS A SERVICE

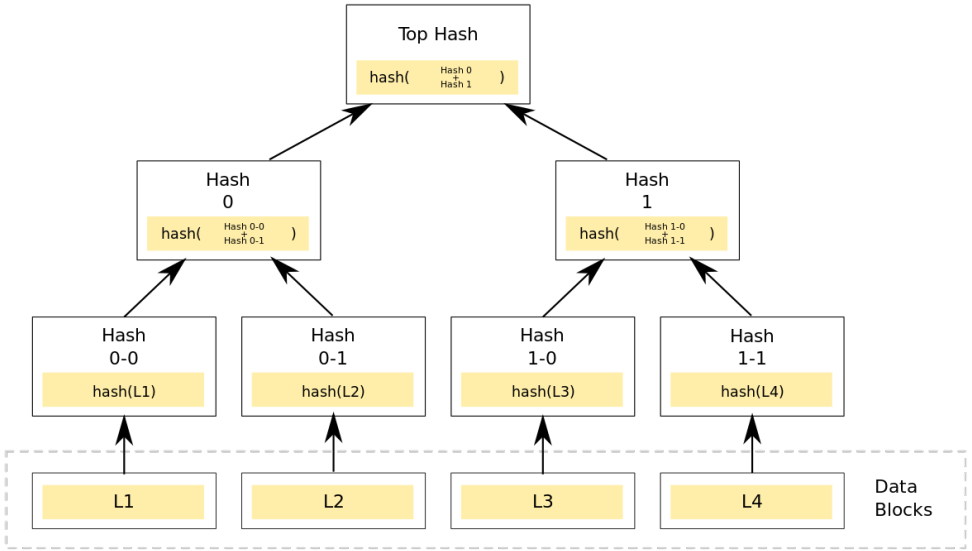
Thinking at providing a web service to make queries, the previously described approaches could be computationally expensive.

Using batches of queries could e a good idea but **is it possible to move the work load from the server to the client?**

Additionally, is it possible to make this work computationally feasible for the client?

MERKLE TREE

Blockchain



REFERENCE-FREE ALIGNMENT AS A SERVICE

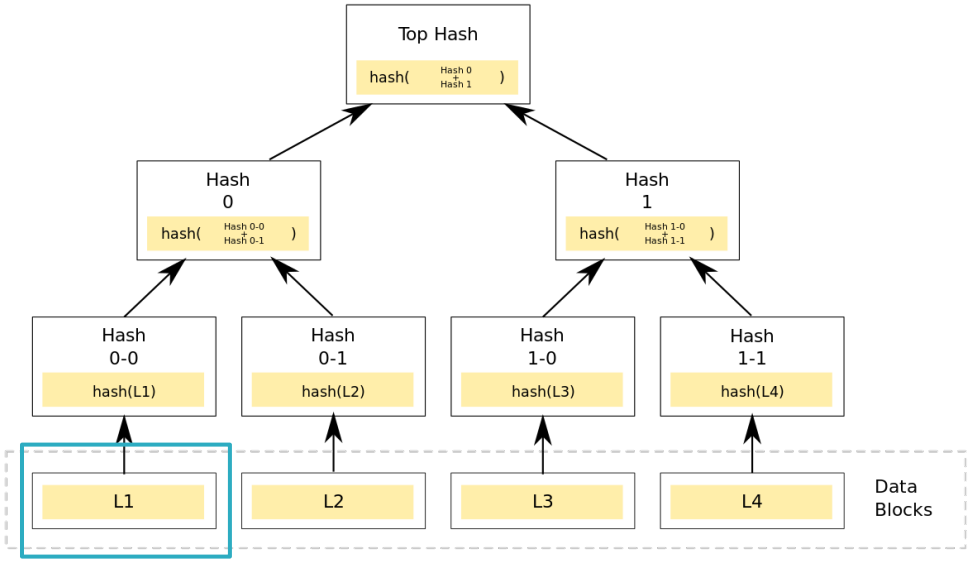
Thinking at providing a web service to make queries, the previously described approaches could be computationally expensive.

Using batches of queries could e a good idea but **is it possible to move the work load from the server to the client?**

Additionally, is it possible to make this work computationally feasible for the client?

MERKLE TREE

Blockchain



REFERENCE-FREE ALIGNMENT AS A SERVICE

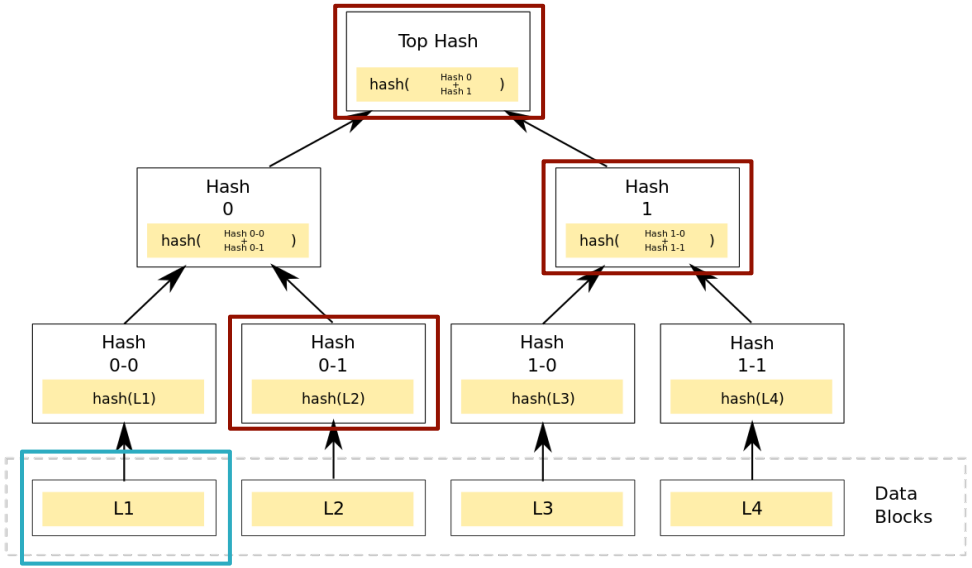
Thinking at providing a web service to make queries, the previously described approaches could be computationally expensive.

Using batches of queries could e a good idea but **is it possible to move the work load from the server to the client?**

Additionally, is it possible to make this work computationally feasible for the client?

MERKLE TREE

Blockchain



REFERENCE-FREE ALIGNMENT AS A SERVICE

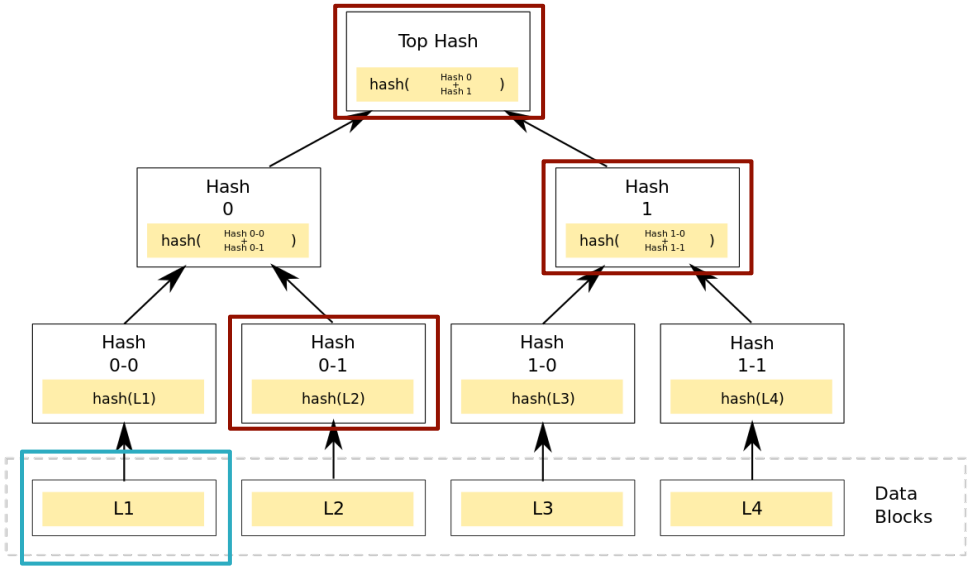
Thinking at providing a web service to make queries, the previously described approaches could be computationally expensive.

Using batches of queries could e a good idea but **is it possible to move the work load from the server to the client?**

Additionally, is it possible to make this work computationally feasible for the client?

MERKLE TREE

Blockchain



THIS SOLUTION IS FEASIBLE FOR SHORT QUERIES