A resource-frugal probabilistic dictionary and applications in (meta)genomics

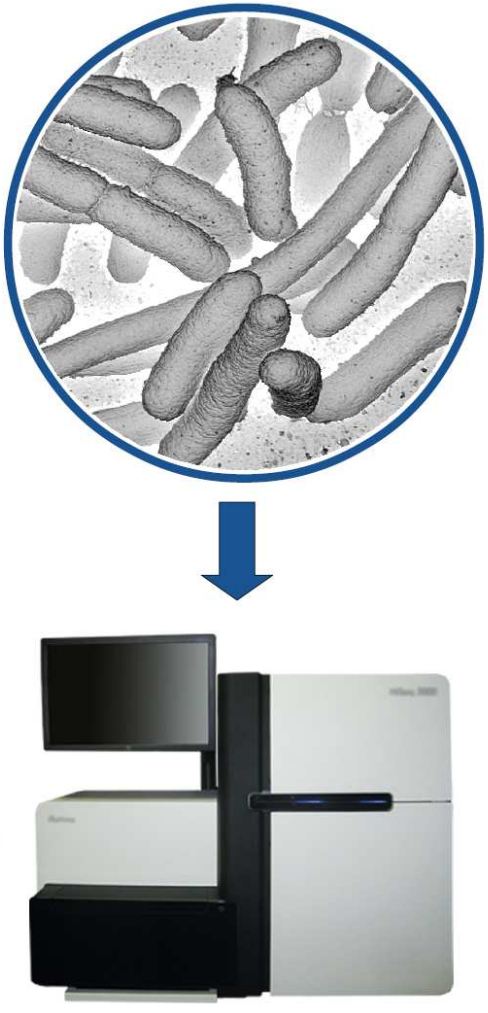
Camille Marchet, **Antoine Limasset**, Lucie Bittner, Pierre Peterlongo

Prague Stringology Conference

August, 2016

Sequencing

```
>read1
ACGACGACGTAGACGACTAGC
AAAACTACGATCGACTAT
>read2
ACTACTACGATCGATGGTCGC
GCTGCTCGCTCTCTCGCT
...
>read10.000.000
TCTCCTAGCGCGGCGTATACG
CTCGCTAGCTACGTAGCT
...
```



Dataset comparison

Read set to references sequences

Huge set of tool

Read set to Read set

Some tools but that index the dataset in a very roughly manner

sequence **ATGGAAGTCGCGGAATC**

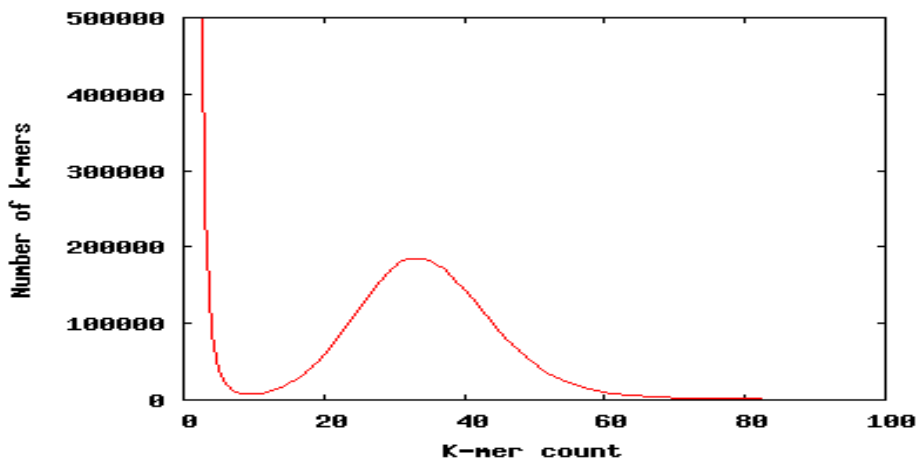
7mers

ATGGAAG
TGGAAGT
GGAAGTC
GAAGTCG
AAGTCGC
AGTCGCG
GTCGCGG
TCGCGGA
CGCGGAA
GCGGAAT
CGGAATC

Our proposal

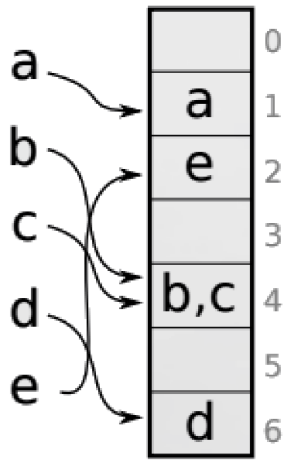
The idea

- Filter erroneous kmers
- Index those k-mers using an adequate hash function (MPHF)
- Associate information to the k-mers according to the task

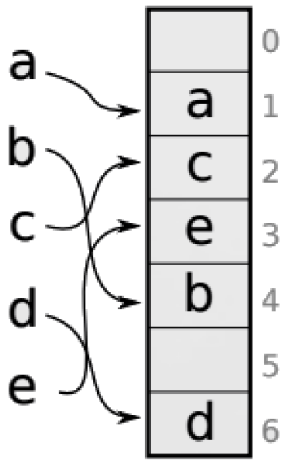


Hash functions

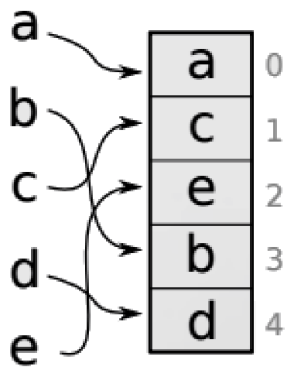
Classical hashing



Perfect hashing
(no collisions)



Minimal perfect hashing
(no collisions, |image|=|input|)



a,b,c,d,e hashable elements (e.g. strings, integers, etc..)

hash function

image [0;m] of hash function
(e.g. indices of buckets in a hash table)

BBhash: our MPHF library

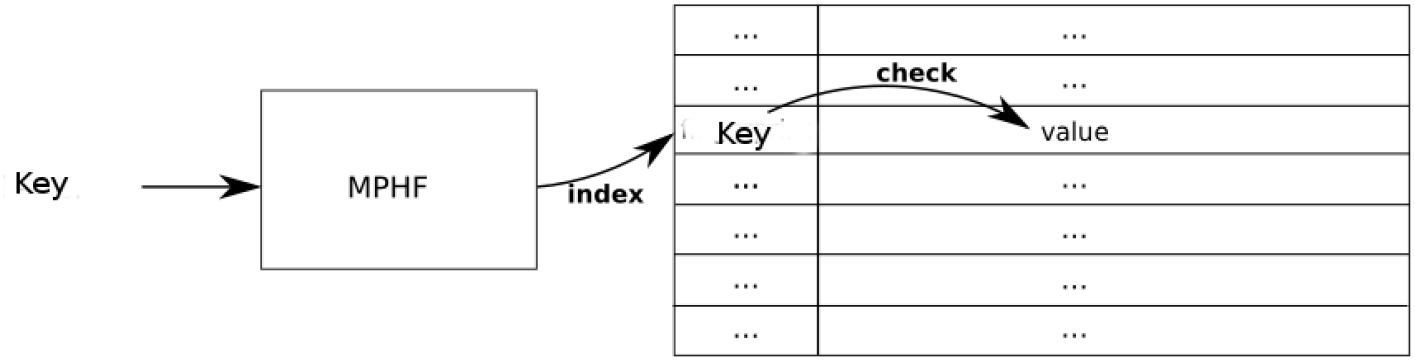
Pros

- Memory efficient (3 or 4 bits by key)
- Fast query (hundreds of ns)
- Fast to construct even for billions keys

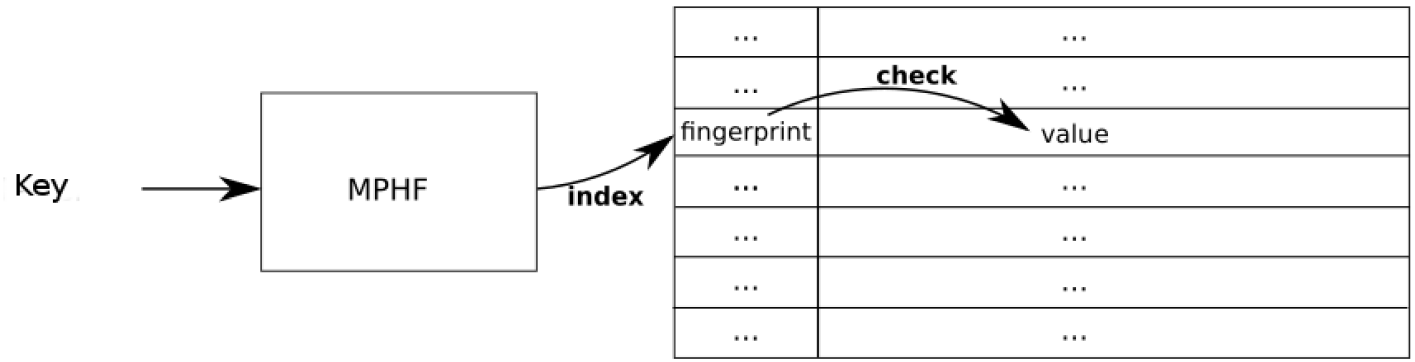
Cons

- Static
- No membership operation
- A stranger key can be associated to a value

Quasi-dictionary



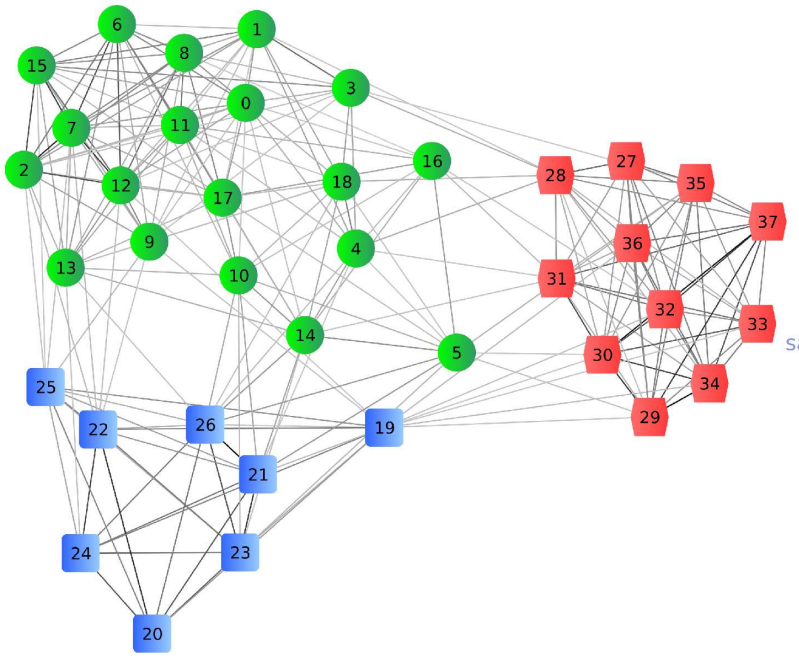
Can be reduced into

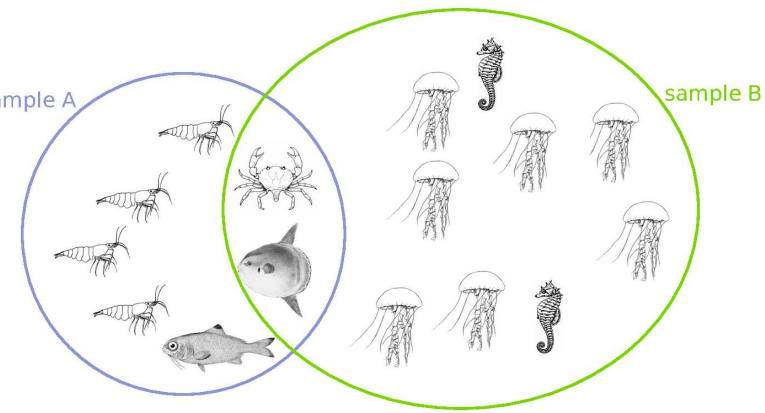


$$\text{FP rate} \approx 1/2^{fingerprintsize}$$

Biological question

How to detect similar reads inter set or intra set ?





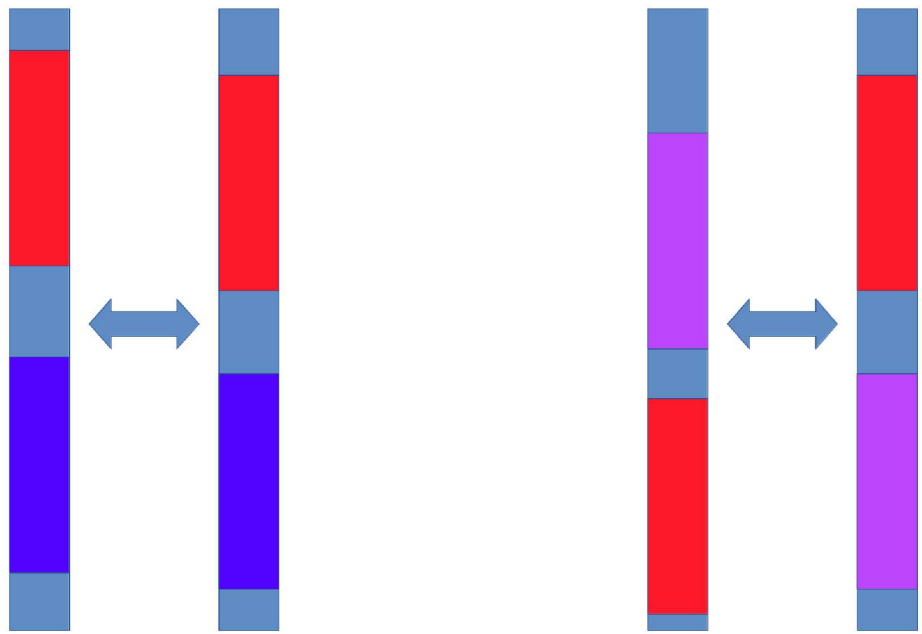
State of the art

Indexed Dataset	Time(s)				Memory(GB)			
	Blast	Bowtie2	BWA	Starcode	Blast	Bowtie2	BWA	Starcode
10K	4	3	6	2	0.7	0.29	0.04	11.36
100K	52	51	106	29	18.5	0.77	0.49	12.06
1M	795	10,644	3,155	1,103	24.5	5.54	3.4	18.18
10M	X	X	62,912	131,139	X	X	5.9	73.5
100M	X	X	X	X	X	X	X	X

SRC Linker

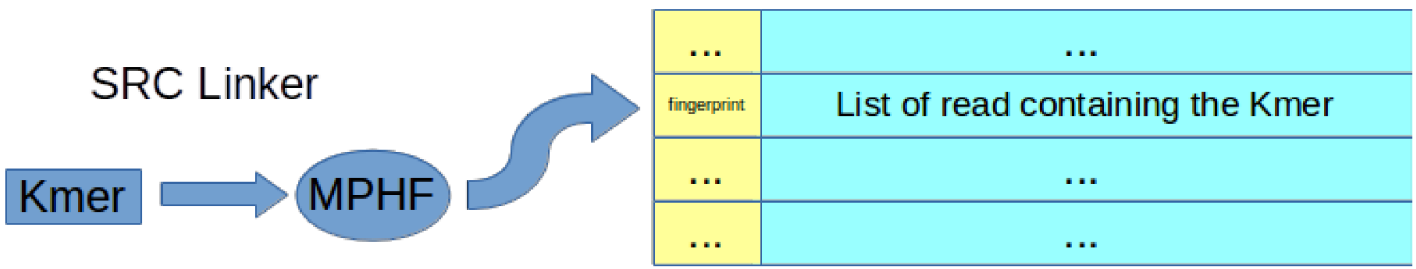
What we want

Find reads of A that share more than t k-mers with a read of B.



SRC Linker

Structure



Comparison

Indexed Dataset	Time(s)					Memory(GB)				
	Blast	Bowtie2	BWA	starcode	SRC_linker	Blast	Bowtie2	BWA	starcode	SRC_linker
10K	4	3	6	2	1	0.7	0.29	0.04	11.36	1.01
100K	52	51	106	29	5	18.5	0.77	0.49	12.06	1.07
1M	**795**	**10,644**	**3,155**	**1,103**	**45**	**24.5**	**5.54**	**3.4**	**18.18**	**1.28**
10M	X	X	62,912	131,139	587	X	X	5.9	73.5	3.61
100M	X	X	X	X	14,748		X	X	X	44.37
Full	X	X	X	X	40,828		X	X	X	110.84

Memory usage

Bottleneck

More than 100GB for indexing a complete dataset.
Values » indexing structure

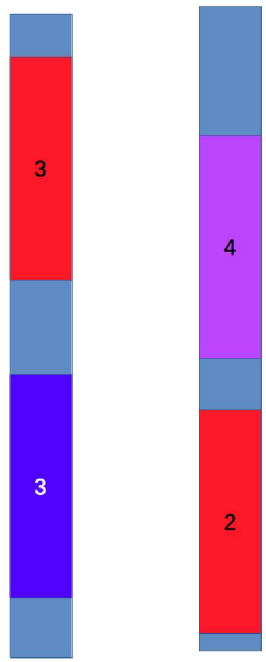
Disk version is less expensive but slow

	Indexation Time (s)		Query Time (s)		Memory
	One thread	20 threads	One thread	20 threads	(GB)
RAM Full	18,067	1,768	17,558	992	110
Disk Full	106,766	28,471	24,873	1,736	19

SRC counter

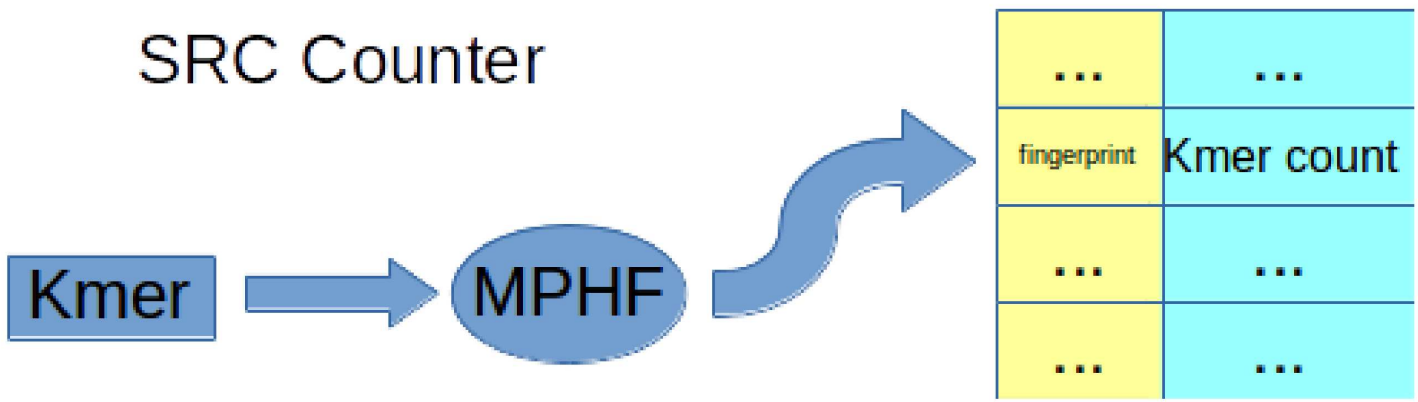
What we want

Find reads of A that share t kmers with the reads of B and estimate its coverage in B



SRC counter

Structure



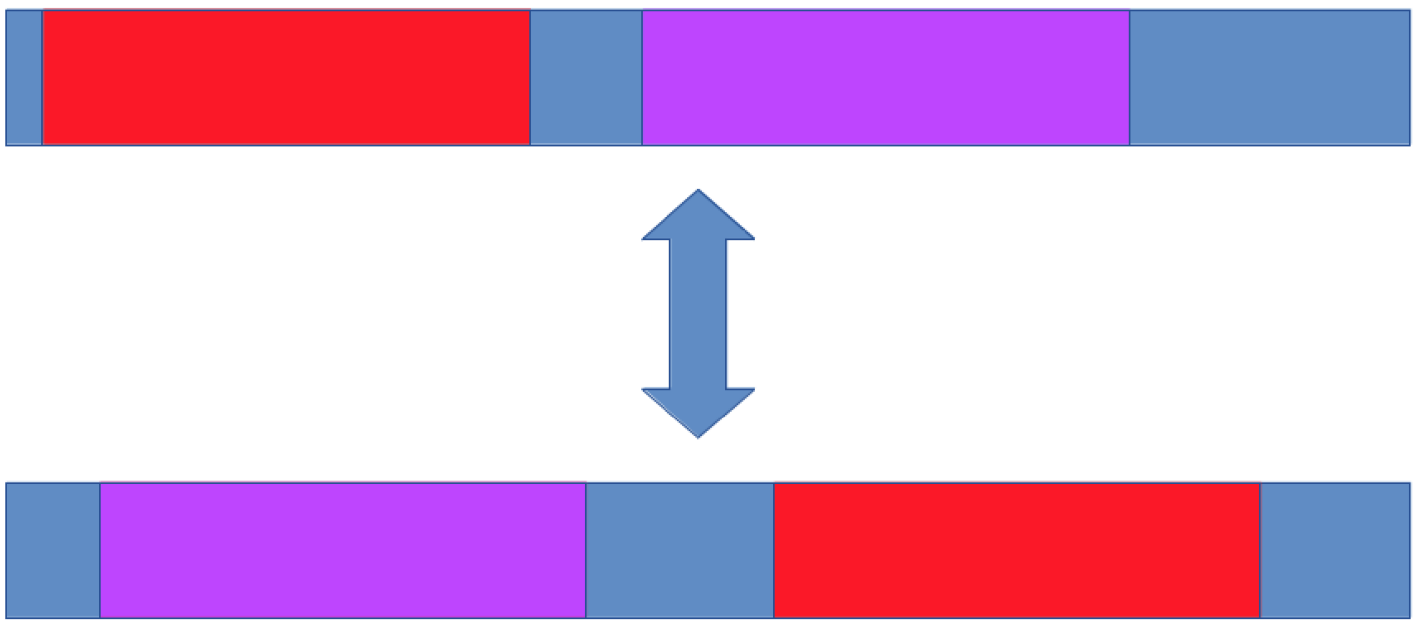
Comparison

Indexed Dataset (nb solid k-mers)	k-mer count time (s)	Construc. time (s)		Memory (GB)			Query Time(s)	
		QD	Hash	QD	QD62	Hash	QD	Hash
1M (64,321,167)	2	1	106	0.25	2.45	2.46	10	13
10M (621,663,812)	15	7	1091	1.80	5.45	23.58	11	17
50M (2,812,637,134)	72	77	5027	8.00	16.37	106.25	11	19
100M (5,191,190,377)	196	220	9335	14.71	44.93	202.91	13	19
Full (8,783,654,120)	486	532	X	24.83	75.96		15	X

Experimental validation to come

Warning

We still have to assess the qualitative aspect of our methods !



Similar in k-mer content but not in the alignment sense

Conclusion

Take home message

We have entered a new world in wich we can index billions of objects.
Many highly comsuming applications (for instance bioinformatics) could benefit of such stuctures in a straighforward way.

Bonus slide BBhash