Navtalks. November 6, 2015. Lisbon, PT.

Genomes in the cloud: Dependable please, but affordable!

Vinicius V. Cogo[1]
Alysson Bessani[1]

[1]LaSIGE, University of Lisbon

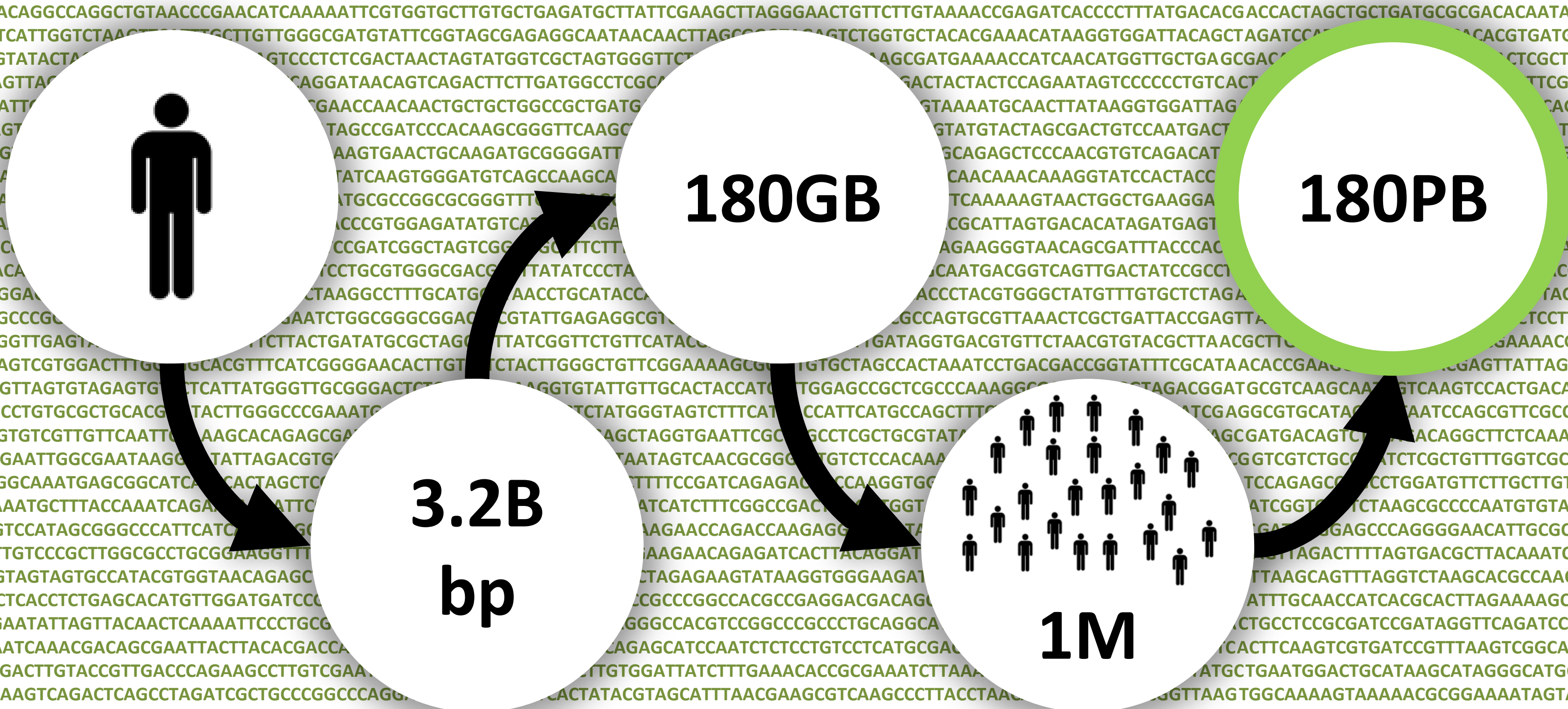
1. Motivation

Problem statement

- Human genomes are **big** an **critical**
- Require **scalability** and **dependability**

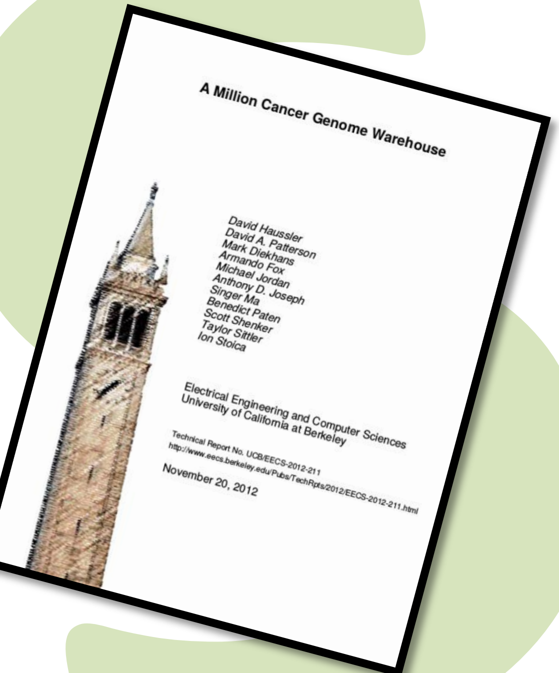
1. Motivation

Problem size



1. Motivation

Problem cost


A Million Cancer Genome Warehouse

David Haussler
David A. Patterson
Mark Diekhans
Armando Fox
Michael Jordan
Anthony D. Joseph
Singer Ma
Benedict Paten
Scott Shenker
Taylor Sittler
Ion Stoica

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-211
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-211.html

November 20, 2012

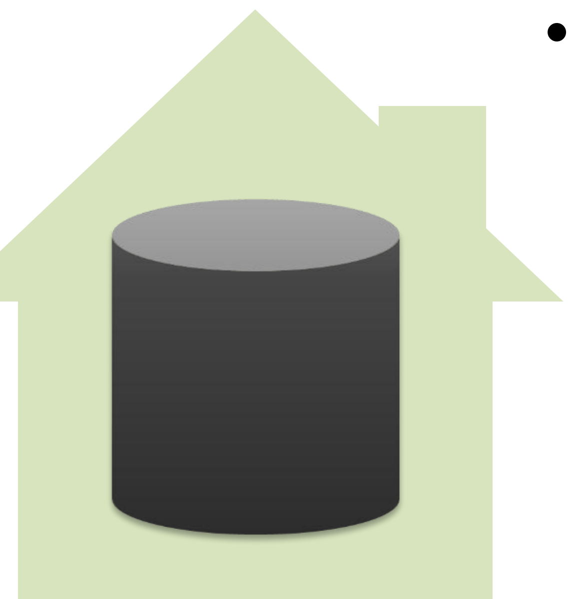
- Costs of a private infrastructure:
  - 2 sites
  - **125PB + processing**

  - $30M → Initial CAPEX
  - $25M → OPEX / year
  - $10M → CAPEX / year for upgrades

  $65M → 1st year

  $35M / year → next years

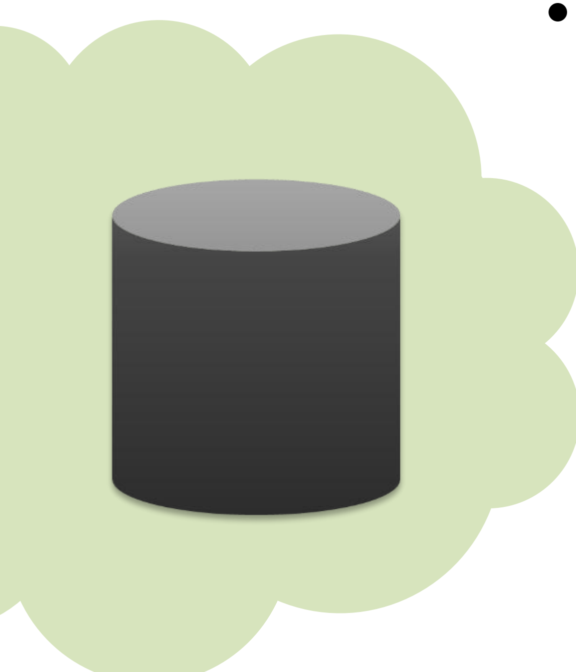
Why genomes in the cloud?



- Private infrastructures:
  - **Privacy-preserving**
  - **Limited scalability**
  - **High maintenance costs**

1. Motivation

Why genomes in the cloud?



- Single public cloud:
  - **Infinite scalability**
  - **Pay-as-you-go**
  - **Security and availability incidents**

Why genomes in the cloud?

- S3 costs:

Storage Type	1TB/year	125PB/year	180PB/year
Standard	$330	**$41.25M**	$59.4M
Infrequent access	$150	**$18.75M**	$27M
Glacier	$84	**$10.5M**	$15.12M

Why genomes in the cloud?

- Multiple public clouds (e.g., DepSky):
  - **Dependable**
  - **50% of space overhead**
  - **Time overhead** **(secret sharing and erasure codes)**

1. Motivation

Why genomes in the cloud?

- Private infrastructures do not scale
- Cloud is an economical alternative
- Cloud scales infinitely
- Cloud dependability is feasible

How to make the dependability viable?

- Adding **privacy-aware methods**
  Prioritizing dependability efforts
  on privacy-sensitive portions of data

- Using **cost-effective techniques**
  Data reduction techniques
  (e.g., compression and deduplication)

Outline

2. Privacy-awareness

DNA-privacy detector

3. Cost-effectiveness

Genome compression

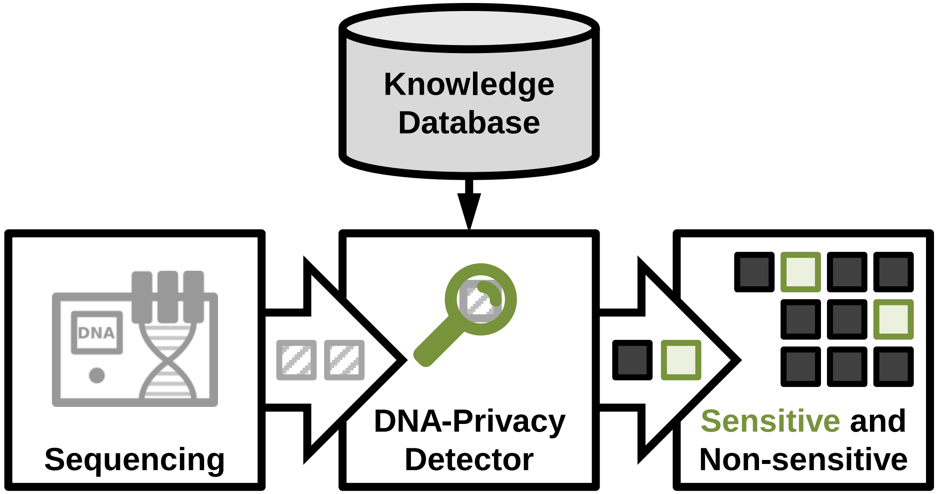
Genome deduplication

Standard compression

4. Combining the solutions

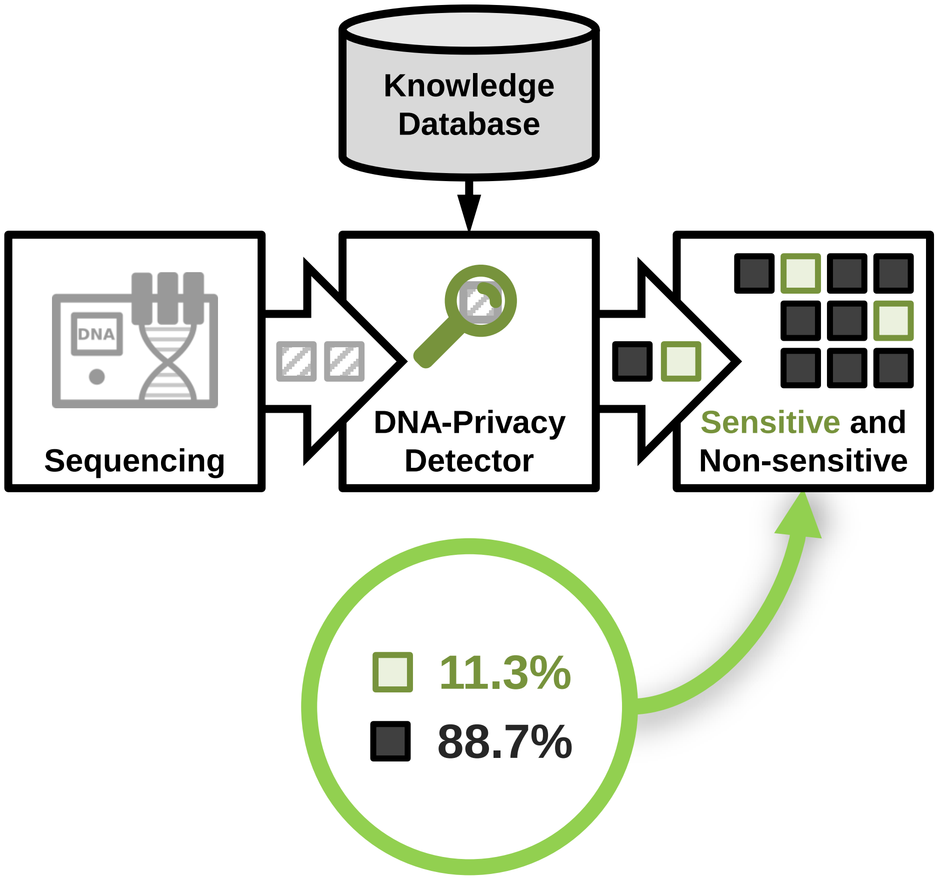
2. Privacy-awareness

DNA-privacy detector



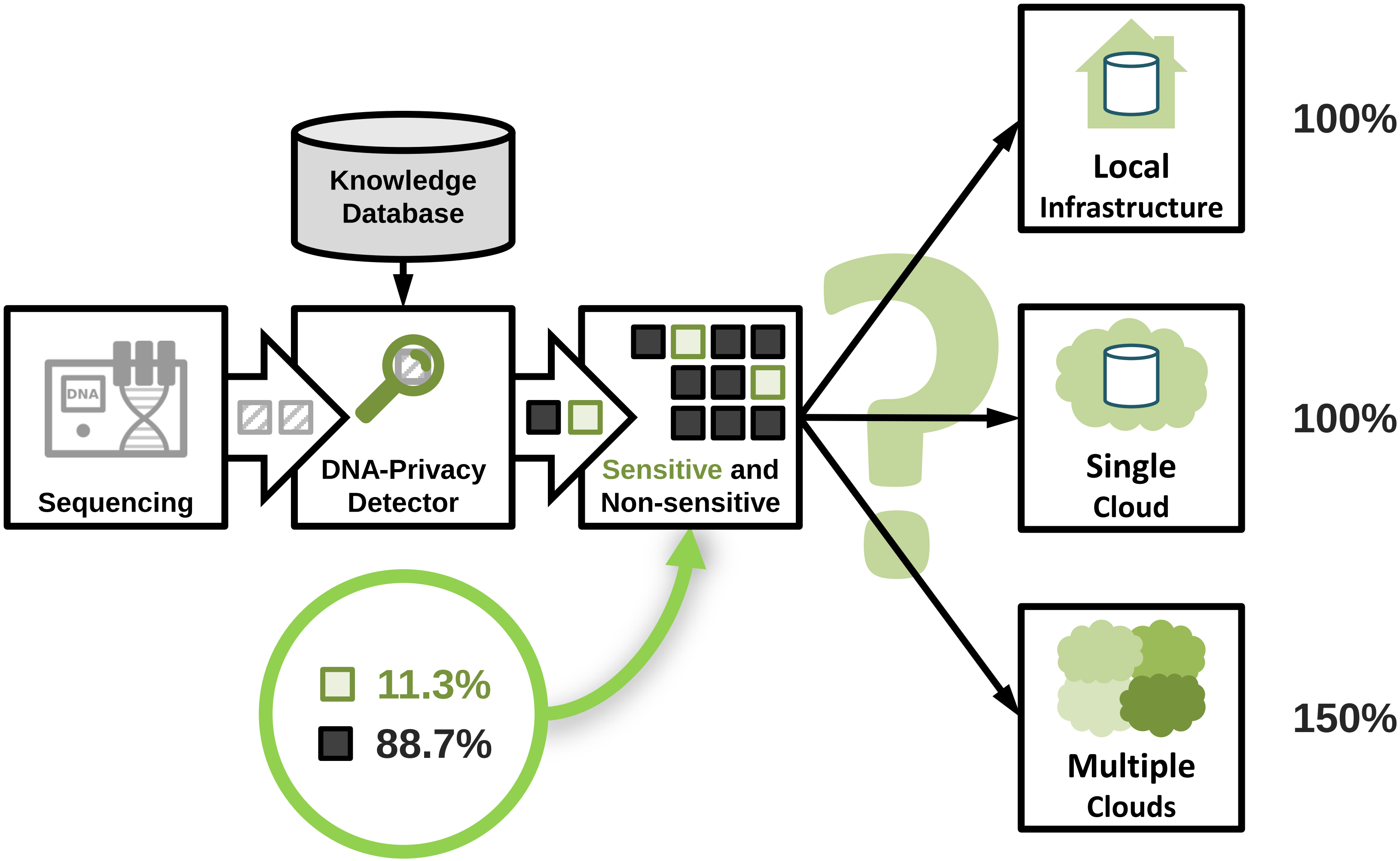
2. Privacy-awareness

DNA-privacy detector



2. Privacy-awareness

DNA-privacy detector



2. Privacy-awareness

Space on each combination

Sensitive	Non-sensitive	Space (S)
Private	Private	100%
Private	Single cloud	100%
Private	Cloud-of-clouds	144.35%
Cloud-of-clouds	Single cloud	105.65%
Cloud-of-clouds*	Cloud-of-clouds	150%

* One may increase the resilience by configuring the secret sharing between 3+ clouds, instead of 2.

3. Cost-effectiveness

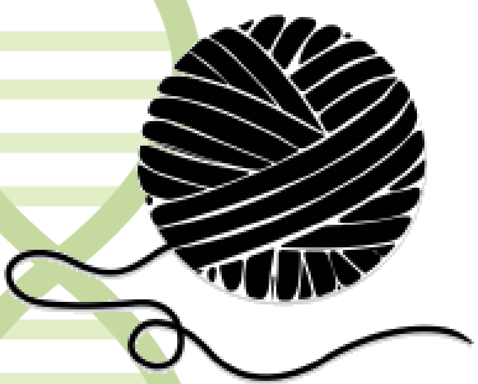
WARNING! Ideas only

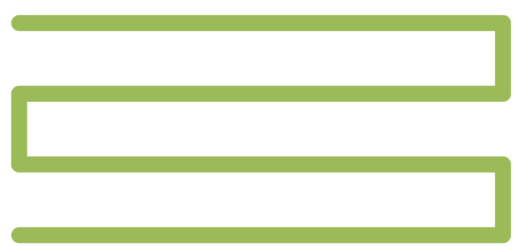
3. Cost-effectiveness

Genome compression

- Compression reduces the size of data
- Referential compression → differences from file A to B
- [Alves2015] compresses ≈3GB to ≈4MB (700:1) in 45s
- Referential compression is good for assembled genomes
- Example:

```
@SRR001666.chr1
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACCGGGTGATGGCCGCTGCCGA
TGGCGTCAAATCCCACCGGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACCGG
GTGATGGCCGCTGCCGATGGCGTCAAATCCCACC…
```





FASTA (contiguous)

[Alves2015] Alves, F., Cogo, V., Wandelt, S., Leser, U., & Bessani, A. (2015).
On-Demand Indexing for Referential Compression of DNA Sequences. PLoS ONE 10(7), e0132460.

3. Cost-effectiveness

Genome compression

- Raw genomic data is unaligned
- Referential compression is inefficient for FASTQ
  Reduces ≈30% the file size
- Example:

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
TGGCGTCAAATCCCACCGGGTGATGGCCGCTGAATC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
%R#$8)'X)*B,ed]`]_Ba_^_3_[YB7+9IG9IC
```

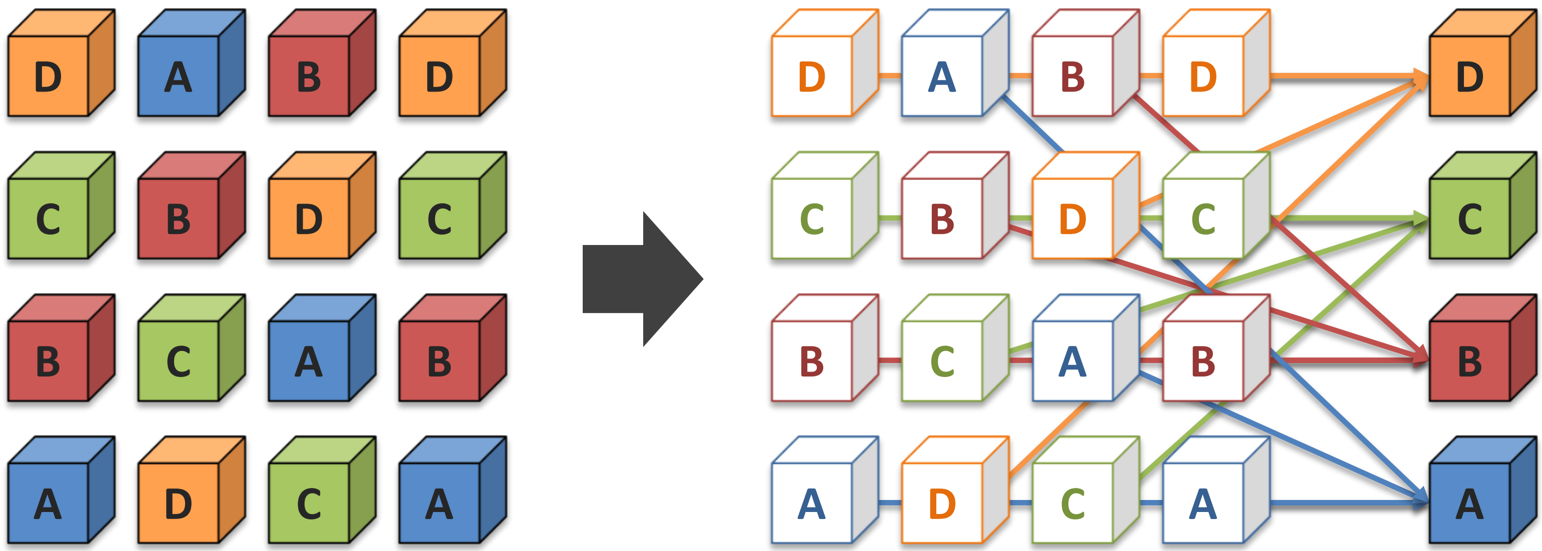
FASTQ (unaligned)

3. Cost-effectiveness

Is deduplication the answer to cost-efficiency in genome storage?

3. Cost-effectiveness

Deduplication



16GB

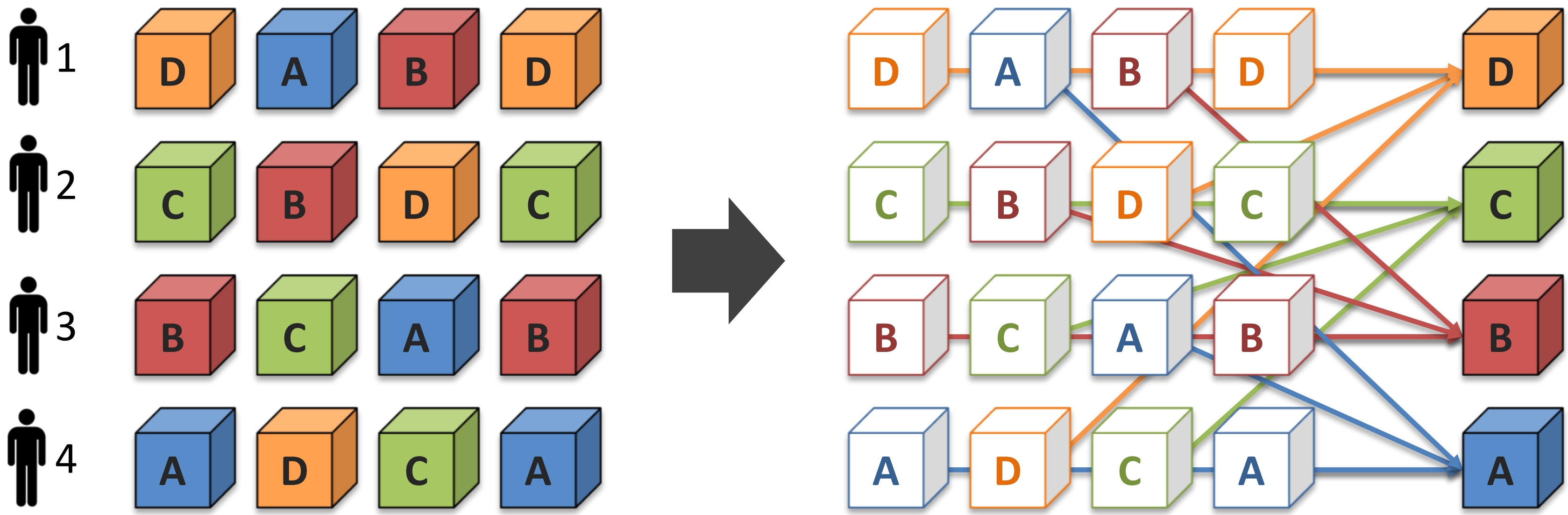
X 1GB block

X Data pointer

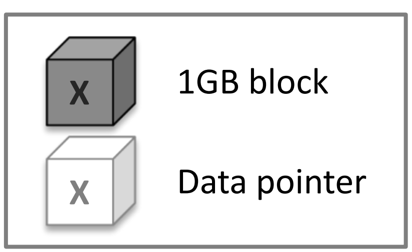
Hash	URI
H(A)	517af899c
H(B)	3bde572c
H(C)	f66876c24
H(D)	3fe71be07

4GB (4:1)
+16 pointers
+ index

Deduplication



16GB



Hash	URI
H(A)	517af899c
H(B)	3bde572c
H(C)	f66876c24
H(D)	3fe71be07

4GB (4:1)
+16 pointers
+ index

3. Cost-effectiveness

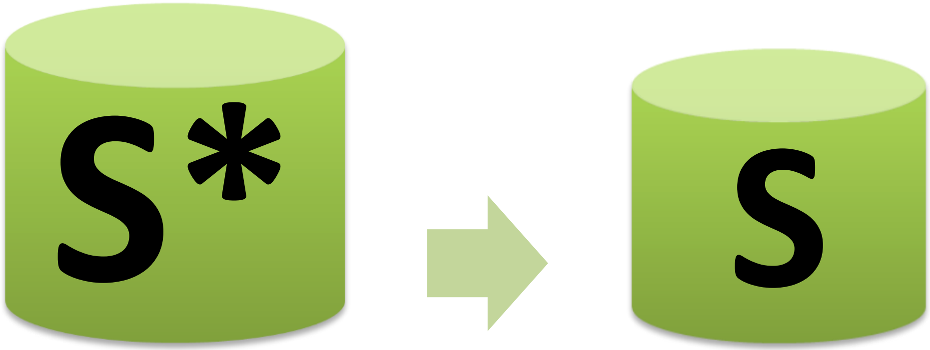
Deduplication

Classification:

- **Granularity:** files or <u>blocks</u> (<u>fixed</u> or <u>variable-size</u>)
- **Locality:** spatial, temporal, spatio-temporal, or <u>none</u>
- **Timing:** <u>online</u> or offline
- **Indexing:** full-comparison, <u>full</u>, sparse, or summary indexes
- **Technique:** <u>aliases</u> or deltas
- **Distribution:** local or <u>global</u> view + <u>centralized</u> or <u>distributed</u>
- **User:** <u>intra-</u> or <u>inter-user</u> (<u>trusted</u> or <u>untrusted</u>)

3. Cost-effectiveness

↓ 5.65% = 100% (original size)

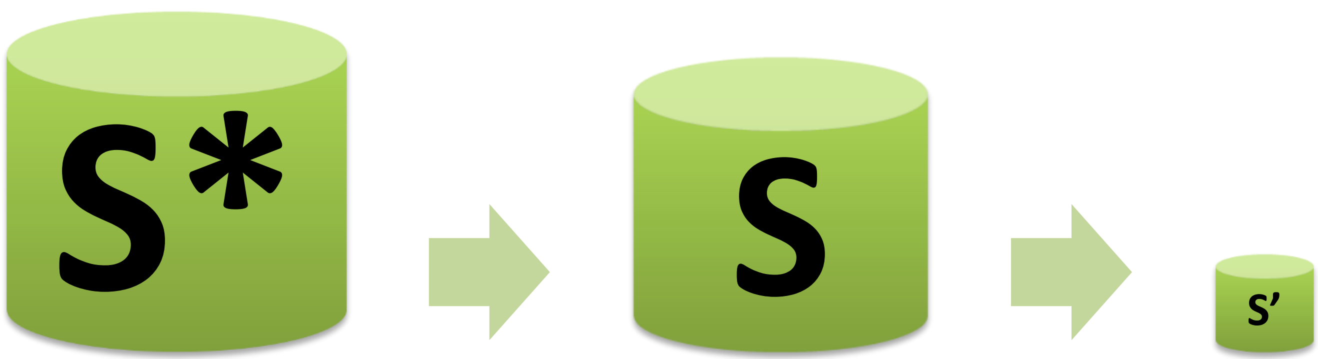


105.65%

100%
(original)

3. Cost-effectiveness

But, we want to reduce more!



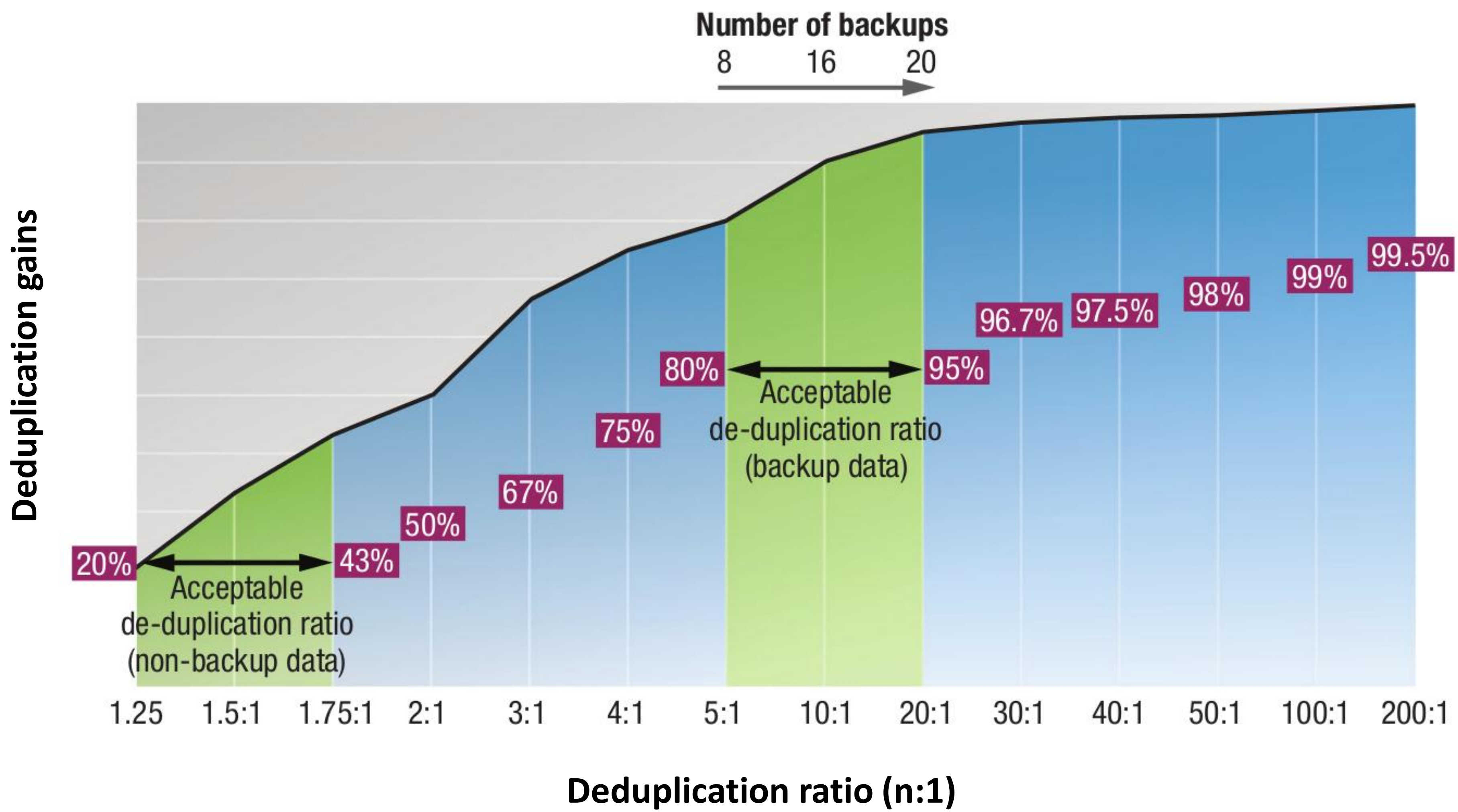
105.65%

100%
(original)

?

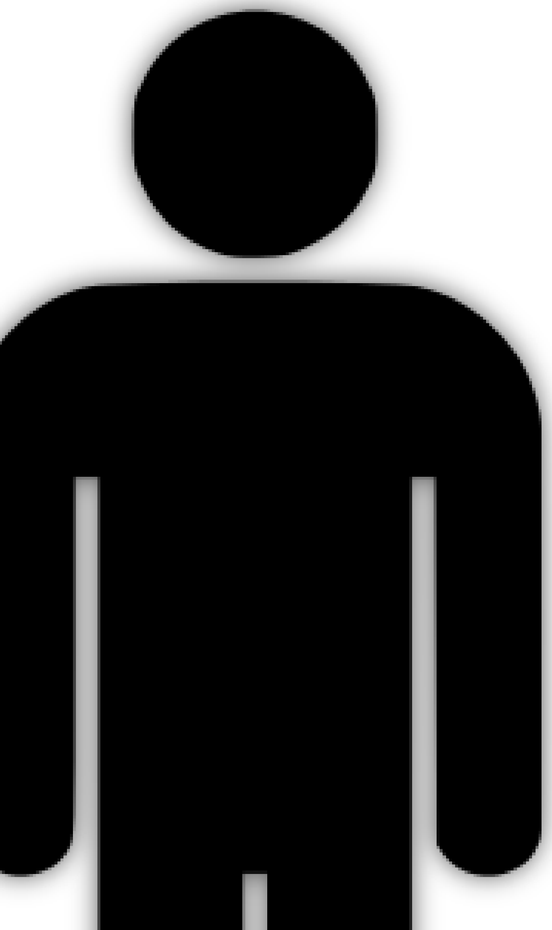
3. Cost-effectiveness

Deduplication



3. Cost-effectiveness

Genome similarity in humans…



>99.5%

3. Cost-effectiveness

Anatomy of a FASTQ read

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
%R#$8)'X)*B,ed]`]_Ba_^_3_[YB7+9IG9IC
```

Anatomy of a FASTQ read

Comment line: id of the individual, run id, id of the read, and others

The read id is never equal from one read to another

@ initiates the first comment line (before the sequence)

+ initiates the second comment line (third line, before the quality scores)

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
%R#$8)'X)*B,ed]`] Ba ^ 3 [YB7+9IG9IC
```

Sequence: nucleotides (50 to 3000)

Quality: scores of each sequenced nucleotide

Anatomy of a FASTQ read

Comment line: stored only once and compressed

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
%R#$8)`X)*B,ed]`]_Ba_^_3_[YB7+9IG9IC
```

Sequence: interesting for deduplication

Quality: interesting for deduplication (same algorithm)

3. Cost-effectiveness

Anatomy of a FASTQ file

Read 345

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
%R#$8)'X)*B,ed]`]_Ba_^_3_[YB7+9IG9IC
```

Read 346

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
```

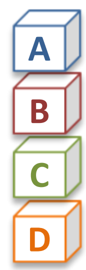
Read 347

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
IIGIGEGG??BFEC?9CFFE9DDGAGDDDF<@HGGH
```

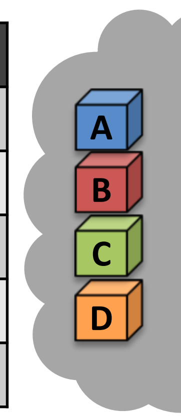
3. Cost-effectiveness

Genome deduplication

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
%R#$8)'X)*B,ed]`]_Ba_^_3_[YB7+9IG9IC
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
IIGIGEGG??BFEC?9CFFE9DDGAGDDDF<@HGGH
```



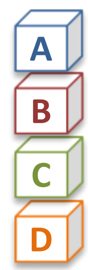
Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**ATTTGGGGTTCA**)	3bde572c
H(**GGGGTGATGGCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…



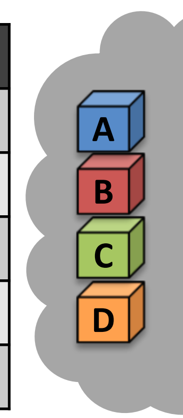
3. Cost-effectiveness

Genome deduplication

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
%R#$8)'X)*B,ed]`]_Ba_^_3_[YB7+9IG9IC
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
IIGIGEGG??BFEC?9CFFE9DDGAGDDDF<@HGGH
```



Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**ATTTGGGGTTCA**)	3bde572c
H(**GGGGTGATGGCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…

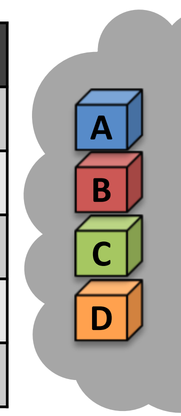


3. Cost-effectiveness

Genome deduplication

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
%R#$8)'X)*B,ed]`]_Ba_^_3_[YB7+9IG9IC
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
IIGIGEGG??BFEC?9CFFE9DDGAGDDDF<@HGGH
```

Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**ATTTGGGGTTCA**)	3bde572c
H(**GGGGTGATGGCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…

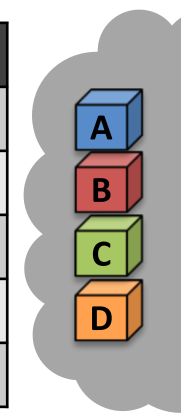


3. Cost-effectiveness

Genome deduplication

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
IIGIGEGG??BFEC?9CFFE9DDGAGDDDF<@HGGH
```

Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**ATTTGGGGTTCA**)	3bde572c
H(**GGGGTGATGGCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…



3. Cost-effectiveness

Genome deduplication

@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345

+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345

@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346

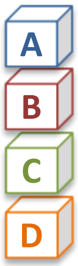
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346

@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347

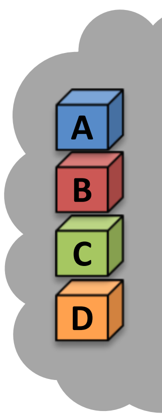
ATTTGGGGTTCAAAGCAGTATCGATCCCATTTGTTG

+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347

IIGIGEGG??BFEC?9CFFE9DDGAGDDDF<@HGGH



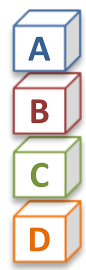
Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**ATTTGGGGTTCA**)	3bde572c
H(**GGGGTGATGGCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…



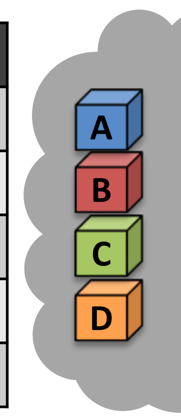
3. Cost-effectiveness

Genome deduplication

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
C
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
Q
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
B
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
Q
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
B
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
Q
```



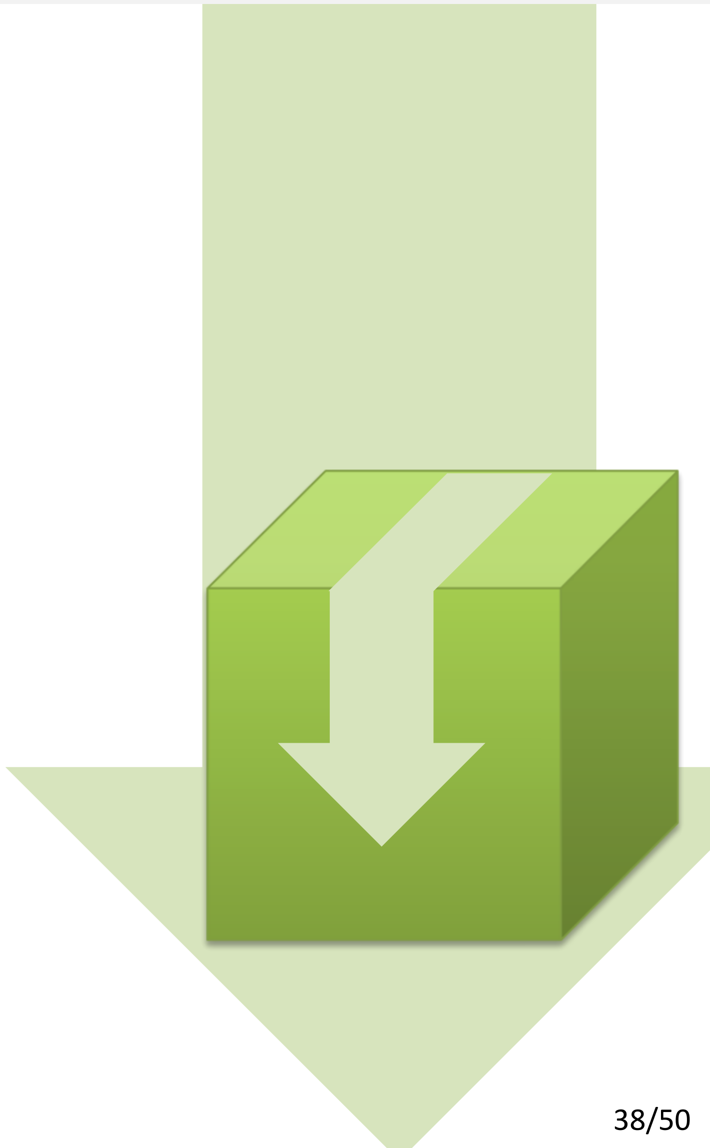
Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**ATTTGGGGTTCA**)	3bde572c
H(**GGGGTGATGGCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…



3. Cost-effectiveness

FASTQ Compression

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
```
C
```
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
```
Q
```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
```
B
```
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:346
```
Q
```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
```
B
```
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:347
```
Q



3. Cost-effectiveness

FASTQ Compression

`@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:`**`345`**

C

Q

346

B

Q

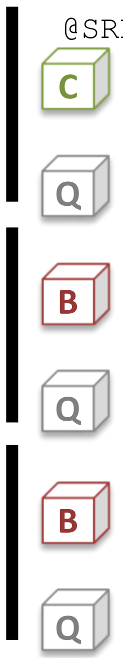
347

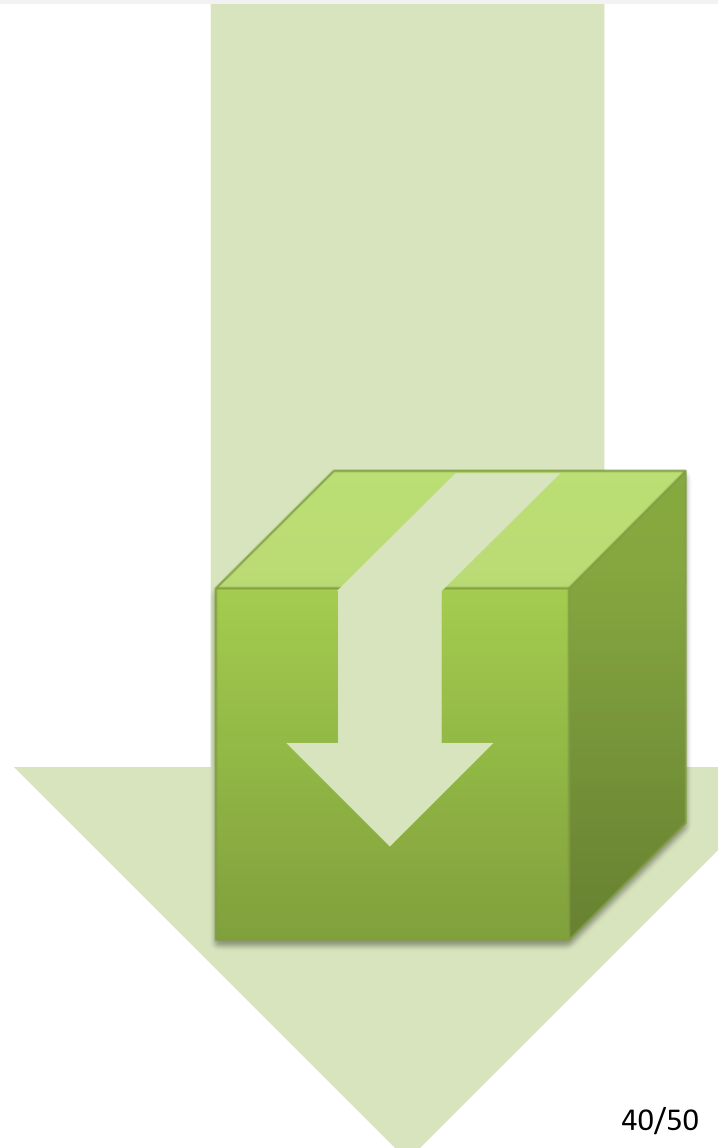
B

Q

FASTQ Compression

@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345

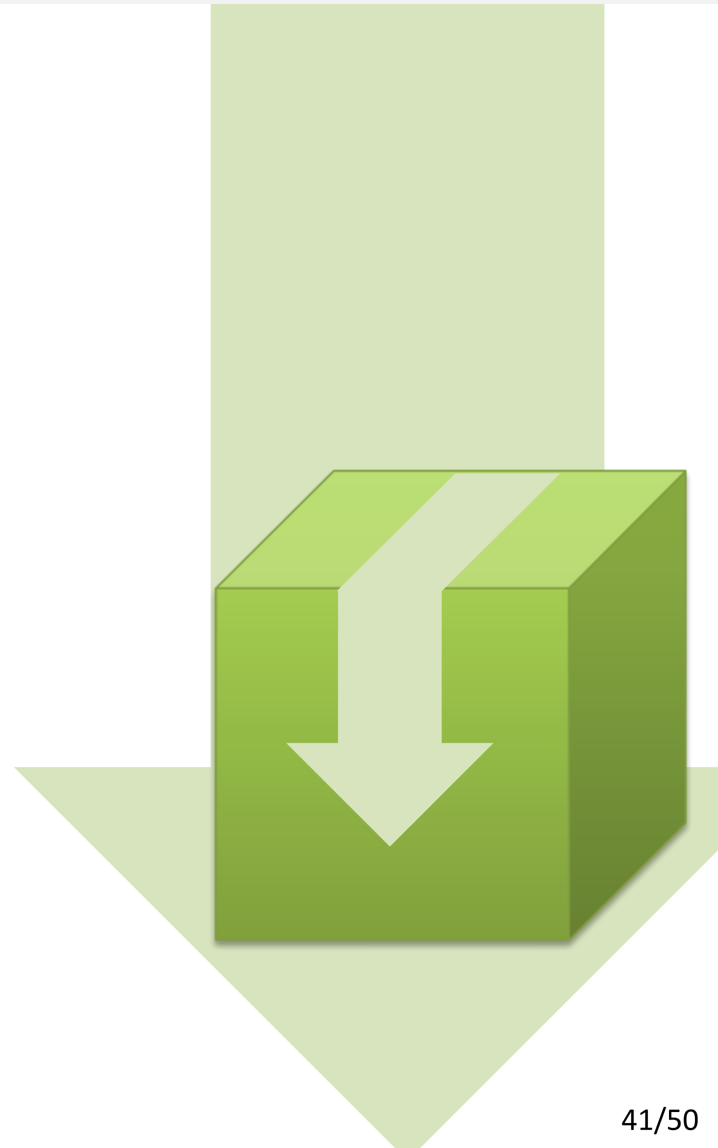




3. Cost-effectiveness

FASTQ Compression

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
```



3. Cost-effectiveness

FASTQ Compression

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
```

CQ2BQ

3. Cost-effectiveness

Sequence vs. hash size

↑ **sequence** = ↓duplicates, ↑ dedupe gains per hit
↑ **hash** = ↓ dedupe gains per hit, ↓ dedupe gains

Hash algorithm	Length (bits)	Gains (100bp)	Gains (1000bp)
MD5	128	84%	98.4%
SHA-1	160	80%	98%
SHA-256	256	68%	96.8%
SHA-512	512	36%	93.6%

3. Cost-effectiveness

Index size

?

Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**CTGCCGATGGTT**)	3bde572c
H(**GGCGTCAAATCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…

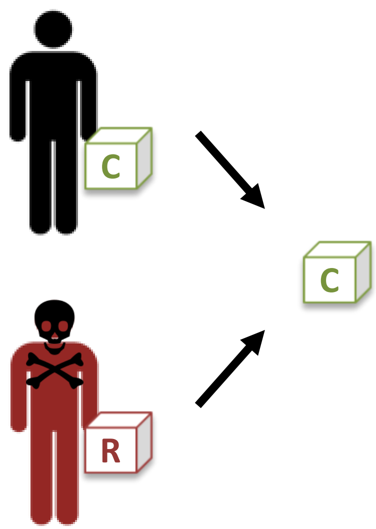
- Simulation: sequences with 100bp
- 50k samples → ≈100% of hits
- index.size() after 50k samples → ≈1 543 567 158 110 (**1.54T**)
- 1.54×10^{12} → 2^{41} (hashes with 41-bits) would be enough
- Amazon S3 and DynamoDB support unlimited number of objects

Fixed or variable-size blocks

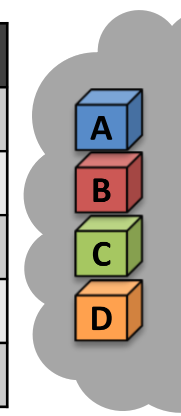
Fixed-size blocks	Variable-size blocks
All lookups have only one size	Lookups may have different sizes
If *e* not found: insert *e*	If *e* not found: split *e* in N *e'* and test all If none found: insert the N *e'*
Smaller index	Bigger index
Lower probability of collisions	Higher probability of collisions
Faster (only one lookup)	Slower (several lookups)
Do not detect translocations nor indels	May detect translocations and indels
Smaller deduplication ratios	Bigger deduplication ratios

3. Cost-effectiveness

Side-channel attack

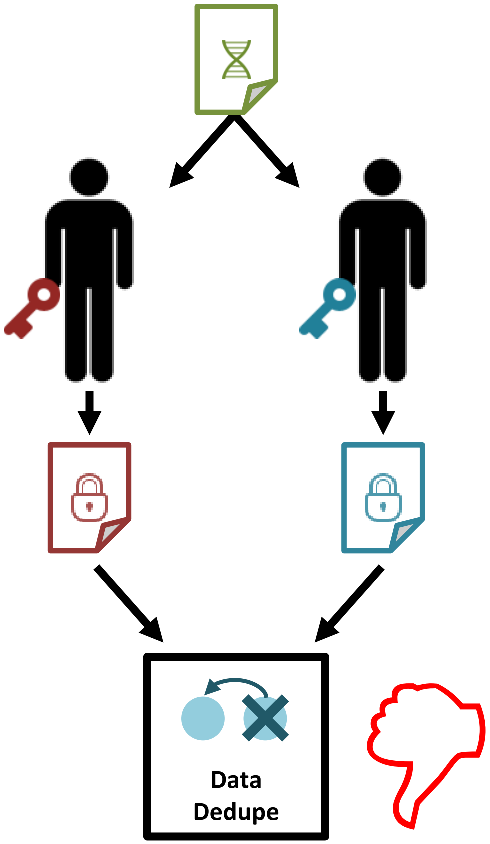


Hash	URI
H(**GATGGCCGCTGC**)	517af899c
H(**ATTTGGGGTTCA**)	3bde572c
H(**GGGGTGATGGCC**)	f66876c24
H(**GCCGATGGCGTA**)	3fe71be07
…	…

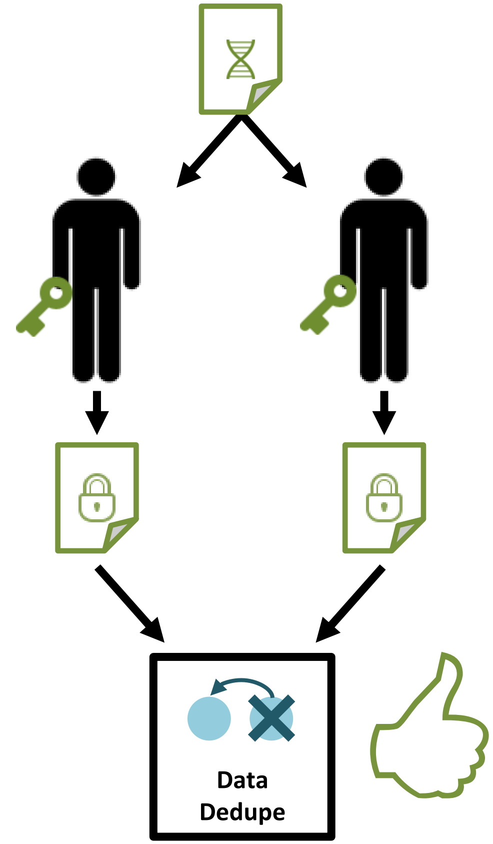


Convergent encryption

Divergent encryption

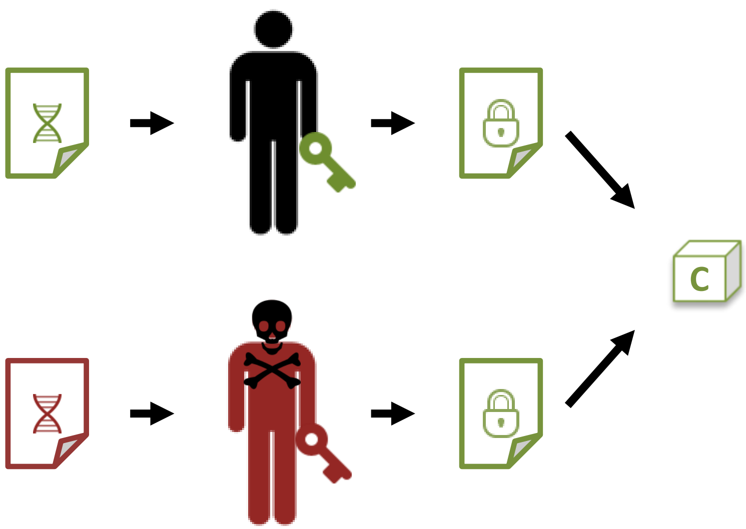


Convergent encryption

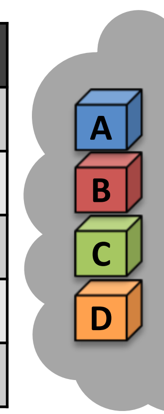


Malicious users

Side-channel attack

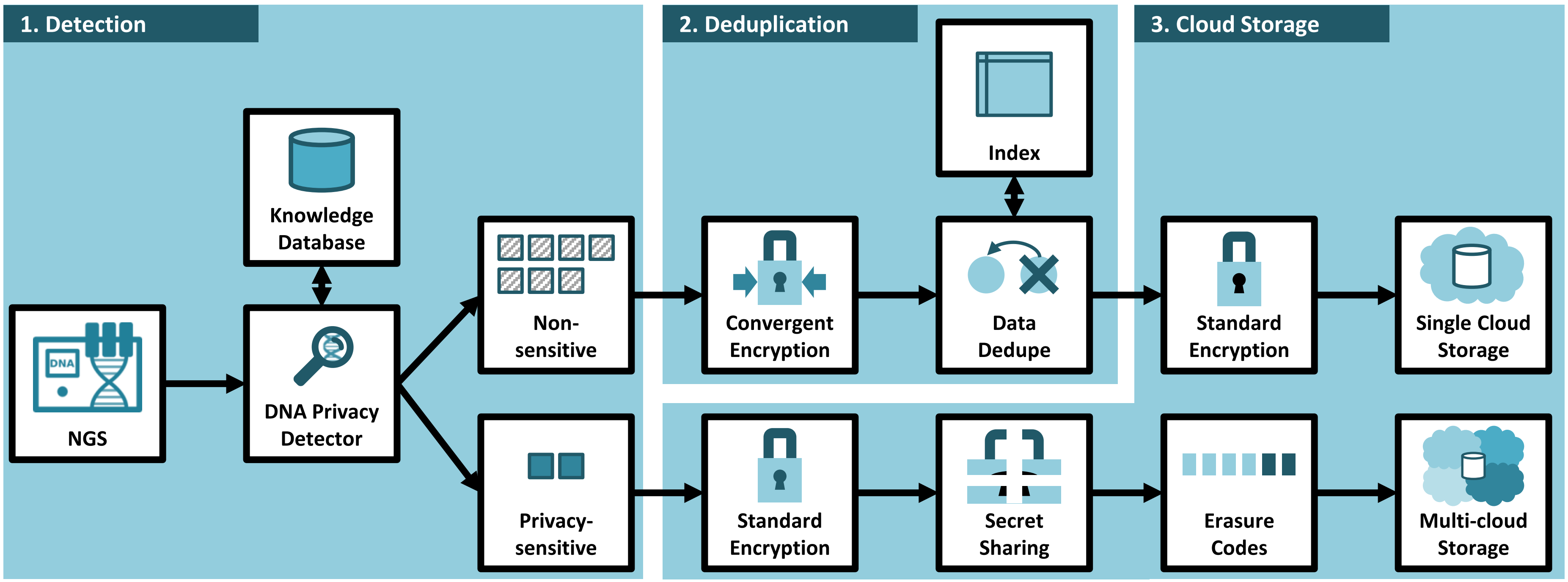


Hash	URI
H(GATGGCCGCTGC)	517af899c
H(ATTTGGGGTTCA)	3bde572c
H(GGGGTGATGGCC)	f66876c24
H(GCCGATGGCGTA)	3fe71be07
…	…



4. Combining the solutions

4. Combining the solutions



Navtalks. November 6, 2015. Lisbon, PT.

Thank you.

Vinicius V. Cogo[1]
Alysson Bessani[1]

[1]LaSIGE, University of Lisbon

[1]Scenarios		Complexity Cost		Dependability			
		Space (S)	Time (T)	Availability	Resilience	Fault Tolerance	Privacy Perception
[2]Local Private Infrastructure	In clear	1x	1x	1	1	0	High
	Encrypted	1x	1x + encrypt	1	1 + decrypt	0	Very High
Single Public Cloud	No replication	1x	1x + encrypt + internet	1	1 + decrypt	0	Low
Multiple Clouds	Replicate in 2 clouds	2x	2x + encrypt + internet	2	1 + decrypt	1 Crash	Very Low
	Replicate in 4 clouds	4x	4x + encrypt + internet	4	1 + decrypt	1 Byzantine	Very Low
	Cloud-of-Clouds	1.5x	1.5x + encrypt + secret sharing + erasure coding + internet	4	2 + decrypt	1 Byzantine	High

[1] We consider attackers have the same probability of compromising the security in any scenario.
[2] Note the real costs from hardware and maintenance are high (not show in the table).

[1]Scenarios		Complexity Cost		Dependability			
		Space (S)	Time (T)	Availability	Resilience	Fault Tolerance	Privacy Perception
[2]Local Private Infrastructure	In clear	1x	1x	1	1	0	High
	Encrypted	1x	1x + encrypt	1	1 + decrypt	0	Very High
Single Public Cloud	No replication						
Multiple Clouds	Replicate in 2 clouds	2x	2x + encrypt + internet	2	1 + decrypt	1 Crash	Very Low
	Replicate in 4 clouds	4x	4x + encrypt + internet	4	1 + decrypt	1 Byzantine	Very Low
	Cloud-of-Clouds						

[1] We consider attackers have the same probability of compromising the security in any scenario.
[2] Note the real costs from hardware and maintenance are high (not show in the table).