Seed design framework for mapping SOLiD reads

Laurent Noé, **Marta Gîrdea**, Gregory Kucherov

LIFL (CNRS and Université Lille 1)
INRIA Lille - Nord Europe

RECOMB 2010, Lisbon
April 25-28, 2010

Seed design framework for mapping SOLiD reads

- Context and motivation
- Seed design
  - Background
  - Our contribution
    - Positioned seeds
    - Lossy seeds
    - Lossless seeds
- Experiments
- Conclusions

High-throughput sequencing technologies

High-throughput sequencing technologies

Source: http://www.massgenomics.org/2010/03/next-gen-sequencing-in-2010.html

- Sequencing human genome: >\$100 million in 2001, ... yesterday \$48,000, today \$4,400, tomorrow \$100 (?)
- "Reading" the genome by short reads of 25-250bp with redundancy

High-throughput sequencing technologies

High-throughput sequencing technologies

Source: http://www.massgenomics.org/2010/03/next-gen-sequencing-in-2010.html

Fact:

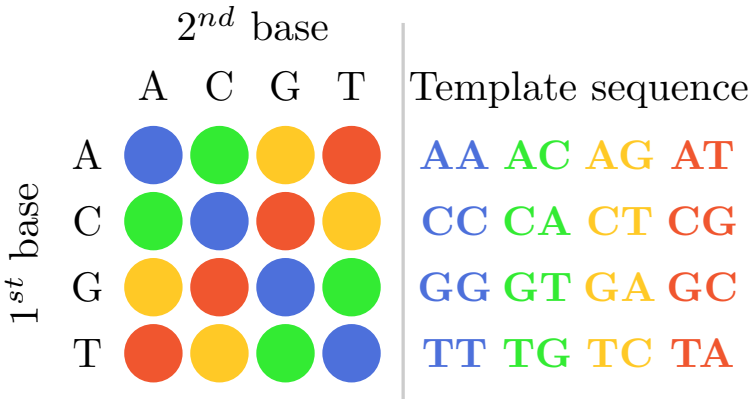
- All technologies produce artifacts

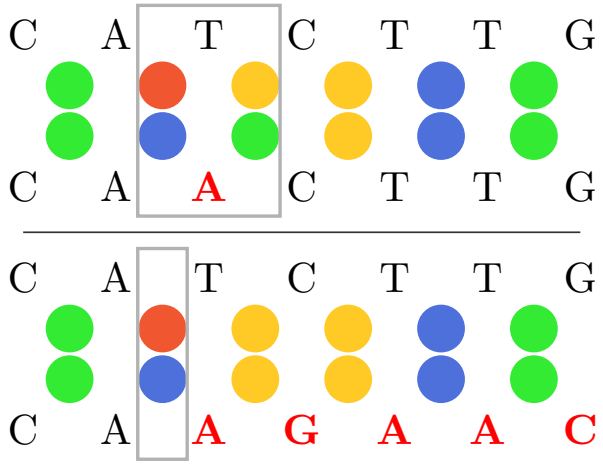
Our goal:

- Use known technology characteristics (including artifacts) to our advantage, to reduce noise effects and obtain more accurate results
- Target: SOLiD

SOLiD™ system (Applied Biosystems)

- 2-base encoding of 35bp (v2) - 50bp (v3) reads ⇒ error-correcting capability helping to reduce the error rate and to better distinguish between sequencing errors and SNPs
- Mappings of color sequences must be implicitly interpreted as nucleotide alignments



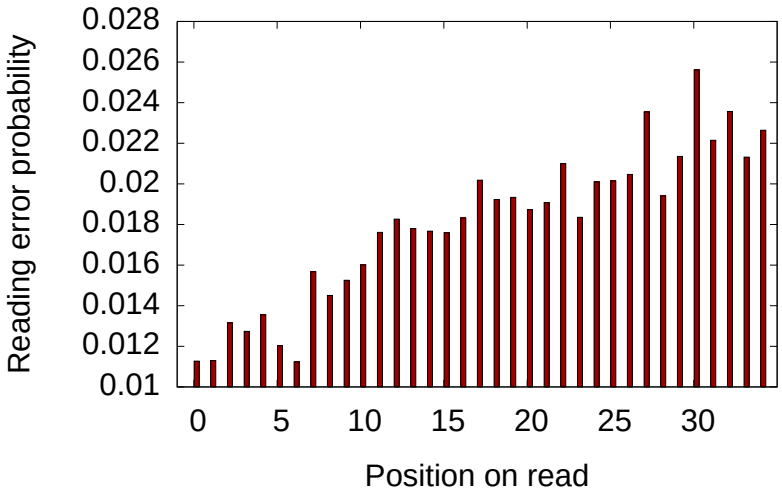


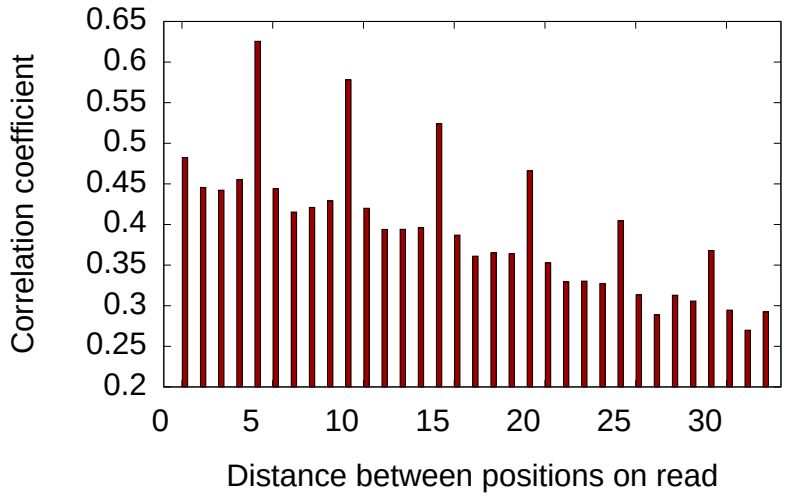
Properties and artifacts of SOLiD technology

- SNPs correspond to 2 adjacent mismatches

Properties and artifacts of SOLiD technology

- SNPs correspond to 2 adjacent mismatches
- The tendency for reading errors to occur
  - periodically at a distance of 5 positions
  - more often towards the end of the read





Read mapping software

- Numerous tools proposed since 2008:

  Eland, SOCS, PatMaN, Maq, ZOOM, SHRiMP, MOSAIK, PASS, PerM, RazerS, Bowtie, BWA, SOAP2, segemehl, MPSCAN, BFAST, ...

Read mapping software

- Numerous tools proposed since 2008:

  Eland, SOCS, PatMaN, Maq, ZOOM, SHRiMP, MOSAIK, PASS, PerM, RazerS, Bowtie, BWA, SOAP2, segemehl, MPSCAN, BFAST, ...

  many of them are based on seeding

Read mapping software

- Numerous tools proposed since 2008:

  Eland, SOCS, PatMaN, Maq, ZOOM, SHRiMP, MOSAIK, PASS, PerM, RazerS, Bowtie, BWA, SOAP2, segemehl, MPSCAN, BFAST, ...

  many of them are based on seeding

- Our "edge": using advanced seed design techniques finely tuned to statistical properties of SOLiD reads

Seed design framework for mapping SOLiD reads

- Context and motivation
- **Seed design**
  - Background
  - Our contribution
    - Positioned seeds
    - Lossy seeds
    - Lossless seeds
- Experiments
- Conclusions

Spaced seeds: background

- Seeds are patterns of matching characters defined to be an evidence of a significant alignment. (Ex: `##-#---#-##`)
- Spaced seeds are more likely to hit an alignment than contiguous seeds of the same weight (= nb of #) $\Rightarrow$ more *sensitive* search [PatternHunter 2002, Yass 2004, ...]
- Using seed families (several seeds simultaneously such that a hit of at least one of them is sufficient) further improves the performance [PatternHunter II 2003, Buhler&Sun 2004].
  Ex: {`###-##`,`#--##---#-#`}
  Price: multiplying memory for hash tables.
- Spaced seeds can (and should) be adapted to the search situation, depending on various statistical characteristics of searched sequences, technological artifacts, desired selectivity (directly affecting speed), etc.

Designing seeds: lossy vs. lossless

Lossy seeds The goal is to detect **most** of the target alignments (better seeds have higher **sensitivity**)

> Sensitivity of a seed (seed family) is defined to be the probability for at least one of the seeds to hit a read alignment with respect to a given probabilistic model of the alignment [Ma et al., 2002, Keich et al., 2004].

Lossless seeds The goal is to detect **all** the alignments with up to a given number of errors (or a given score threshold)

Both settings are used in practice, e.g.

- SHRiMP: lossy
- ZOOM, PerM, MAQ: lossless

Seed design framework

IEDERA software (`http://bioinfo.lifl.fr/yass/iedera`)

- Computes the seed sensitivity with a dynamic programming algorithm as described in [Kucherov et al., 2006, JBCB]
  - "Good" mappings are modeled by *Hidden Markov Models with emitting transitions*
  - A seed, or a seed family, is modeled by a *seed automaton*
- Generates seeds patterns and selects the most sensitive seed families

Seed design framework for mapping SOLiD reads

- Context and motivation
- **Seed design**
  - Background
  - Our contribution
    - Positioned seeds
    - Lossy seeds
    - Lossless seeds
- Experiments
- Conclusions

Our contribution to seed design for mapping SOLiD reads

- We design **positioned spaced seeds** for mapping SOLiD reads, both in the lossy and lossless settings
- For designing and evaluating seeds, we propose **models and algorithms that distinguish between SNPs/indels and reading errors** in order to obtain seeds that are properly adapted to the problem of mapping SOLiD reads

Positioned spaced seeds

Reminder:

- Reads are short sequences of **fixed length**
- The **reading error probability increases towards the end** of the read, implying that a search for similarity within the last positions of the read could lead to erroneous results or no results at all

Positioned seed: a seed π designed *jointly* with a set of positions P to which it is applied on the read.

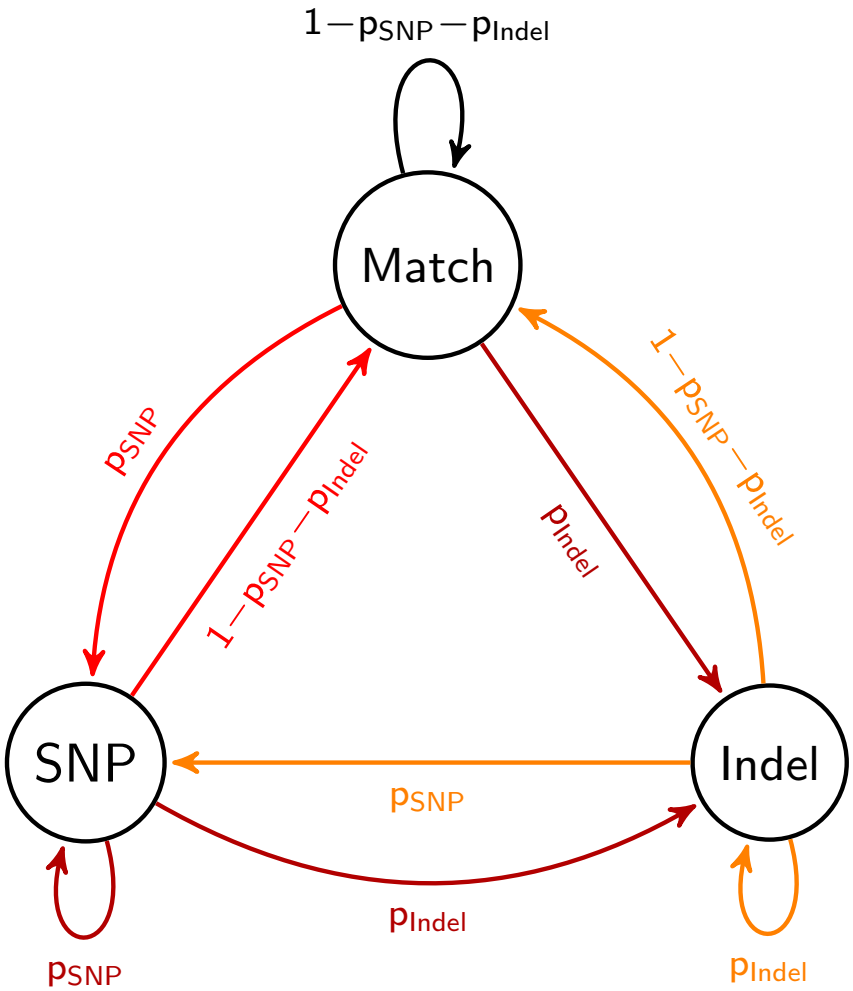
Example: $\pi =$ #-##, $P_\pi = \{0, 3, 9, 13, 18\}$

```
Alignment   111001110101110110110010 0
Positioned  #-##
   seeds       #-##
                    #-##
                         #-##
                              #-##
```

Lossy framework: seed design

- Select the most **sensitive** seeds w.r.t. "good" read mappings
- "Good" mappings are modeled by a combination of two HMMs representing the **biological variation** and the **reading errors** respectively

Lossy framework



States refer to **DNA alignment**

Emitted symbols refer to **color alignment**

Legend (transitions):

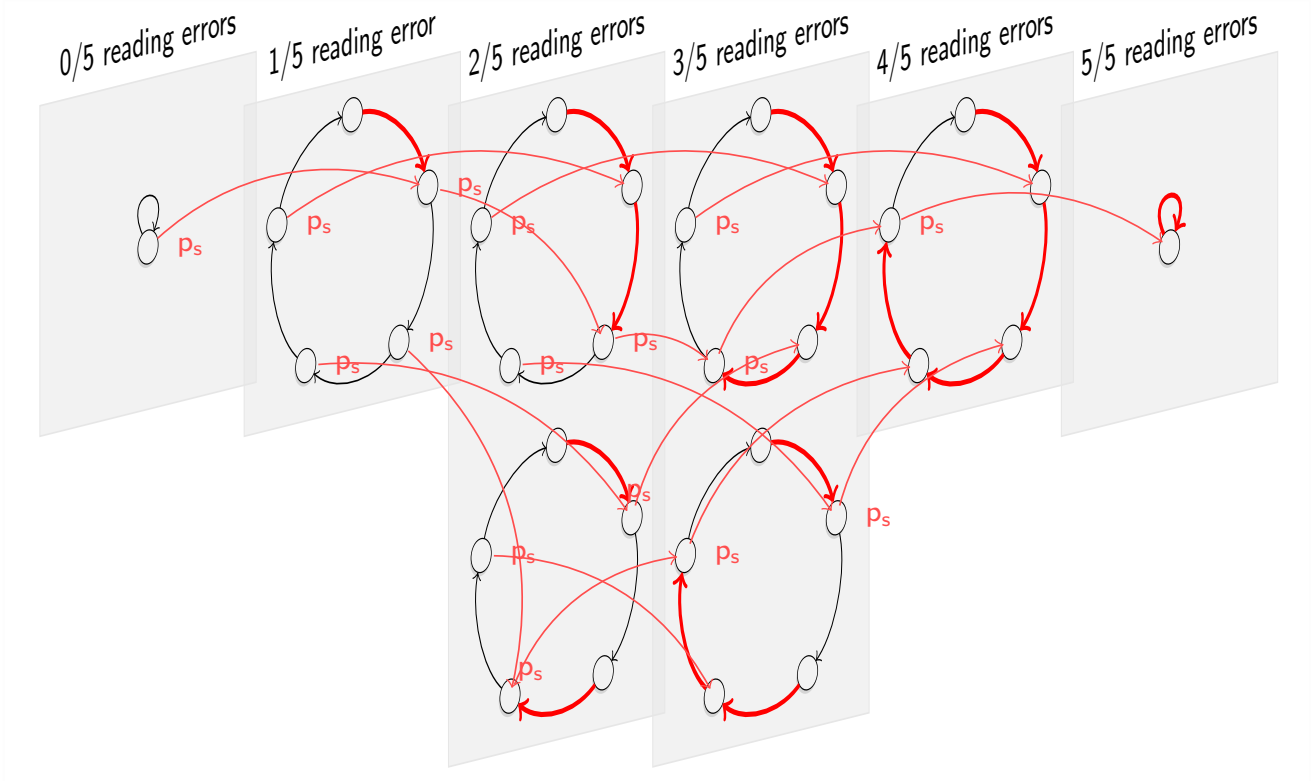
- color matches
- color mismatches
- 1/4 color matches + 3/4 mismatches
- color indels

Biological variations model

Lossy framework

Reminder: The reading error probability increases towards the end of the read

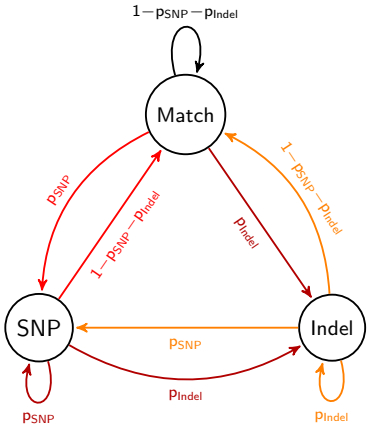
Reminder: Errors tend to appear with a periodicity of 5



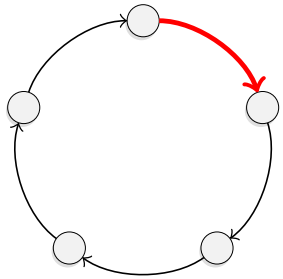
Reading errors model

Legend (transitions): **periodic errors, fixed error probability**; switching to a high error probability; small error probability

Lossy framework

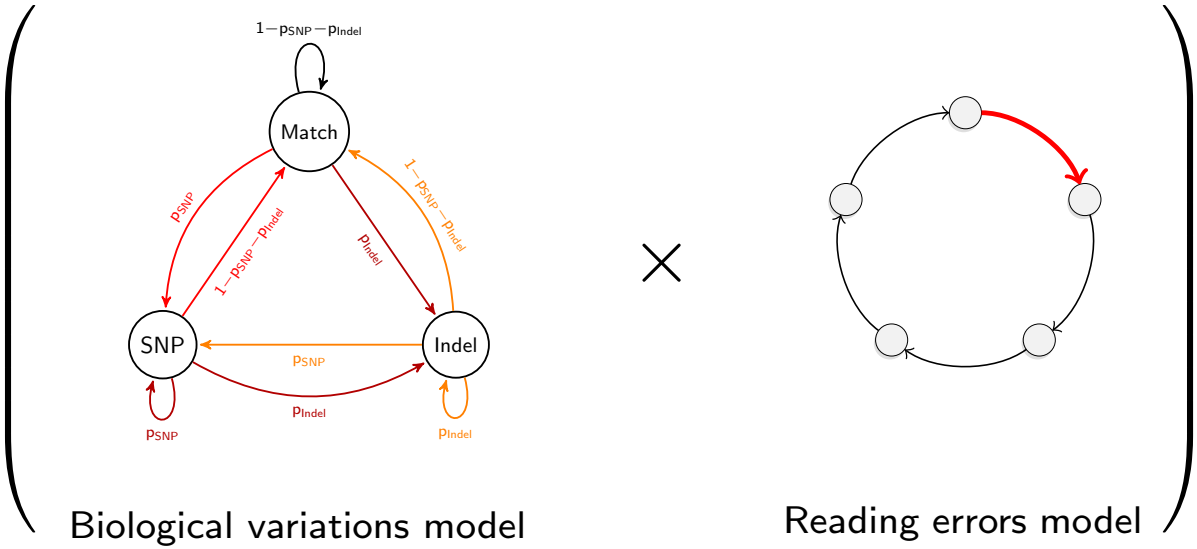


Biological variations model



Reading errors model

Lossy framework

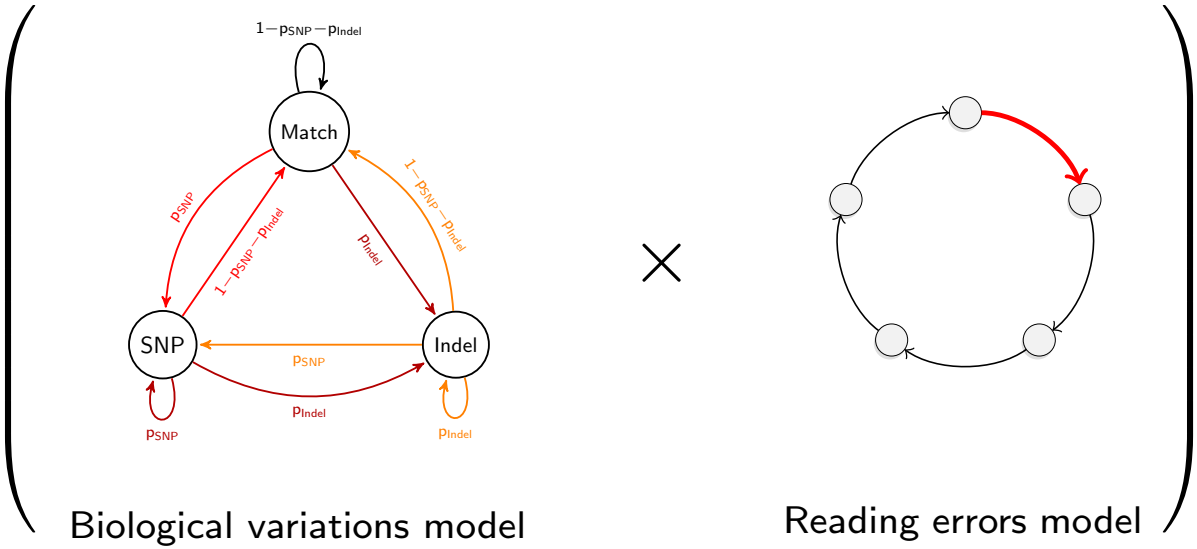


Biological variations model

×

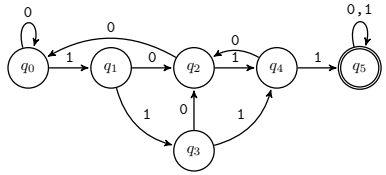
Reading errors model

Lossy framework



Biological variations model

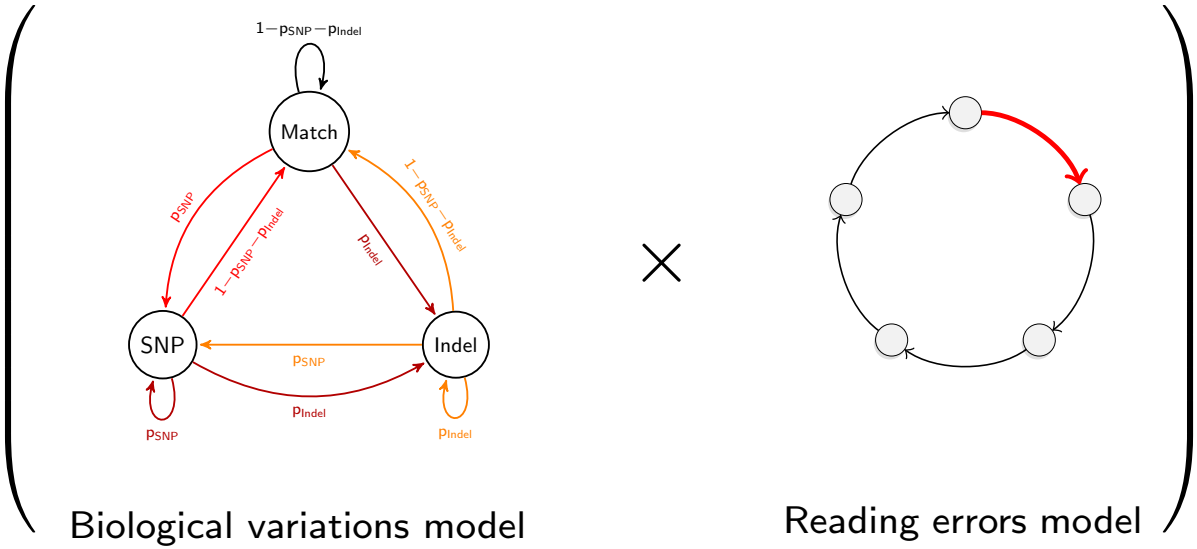
Reading errors model



Seed automaton

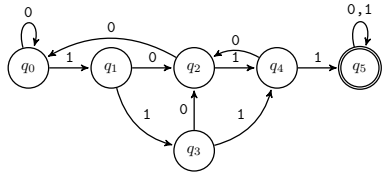
(#−##)

Lossy framework



Biological variations model

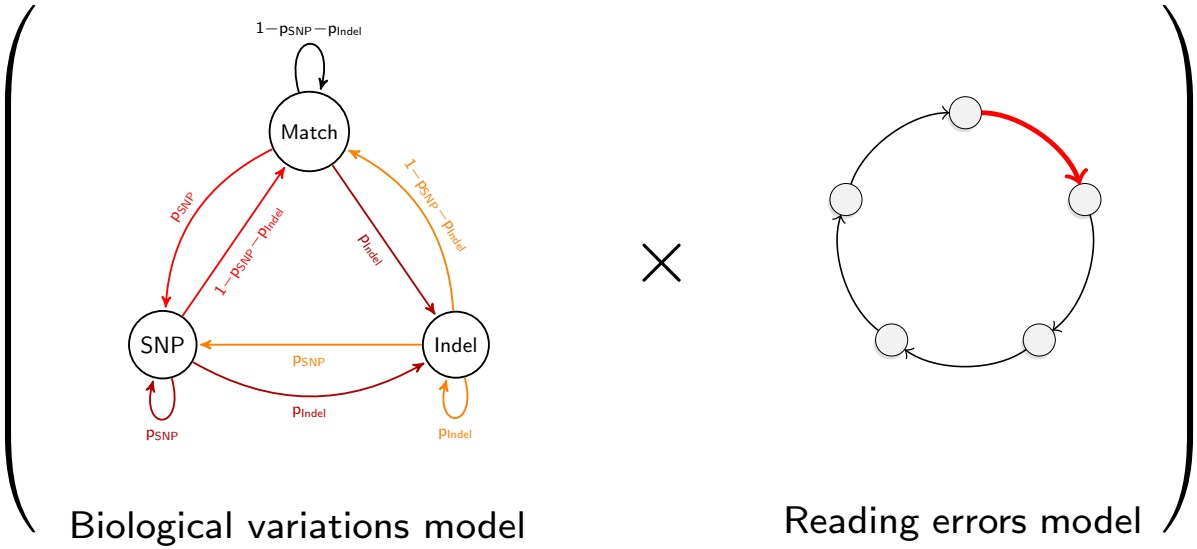
Reading errors model



Seed automaton

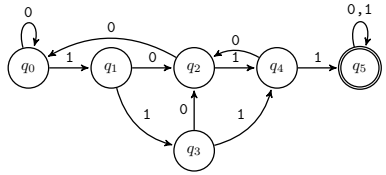
(#−##)

Lossy framework



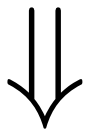
Biological variations model

Reading errors model



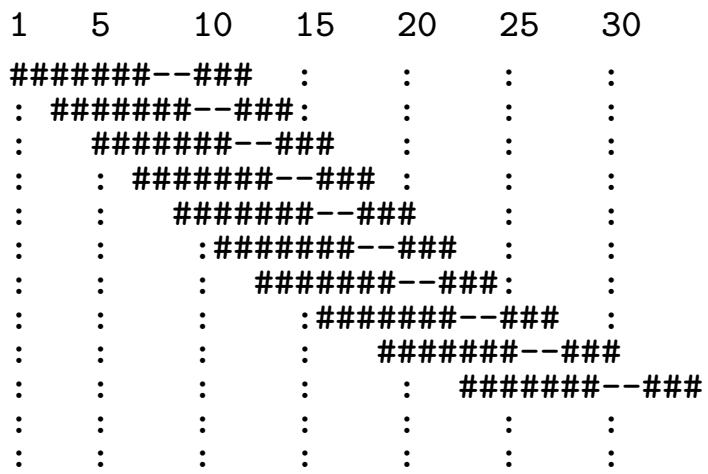
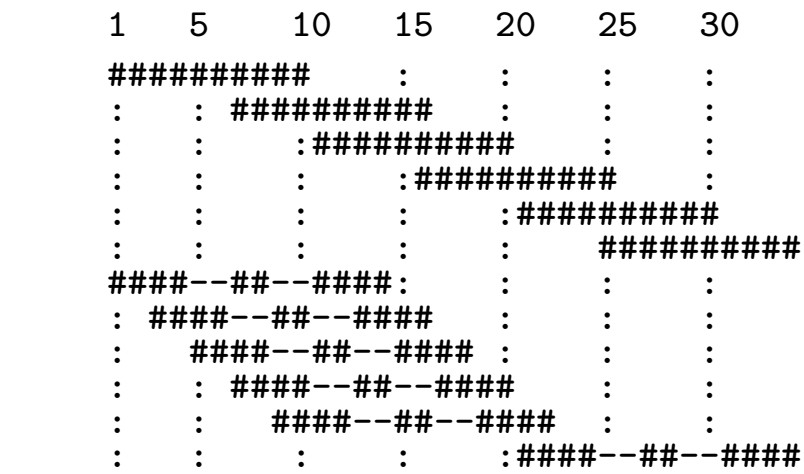
Seed automaton

(#−##)



Theoretical sensitivity of the seed (family) on target alignments

EXAMPLE: Lossy seeds restricted to 12 positions

1-Lossy-10p: sensitivity 0.9543	2-Lossy-12p: sensitivity 0.9685
	

Lossless framework

Lossless seeds have the capacity to hit **all** alignments containing up to an established number of errors
[Burkhardt and Kärkkäinen, 2003]

- lossless for 2 mismatches,
- lossless for 1 mismatch and 1 indel,
- lossless for 1 SNP and 3 reading errors,
- ...

Lossless framework: Verifying the lossless property

We propose **an efficient dynamic programming algorithm directly applied to** $\mathcal{Q}$ that can verify the **inclusion**.

ALGORITHM 1: VERIFICATION OF THE LOSSLESS PROPERTY

m = read length, $\mathcal{Q}$ = the seed automaton

k = the threshold for the number of mismatches

For each state q of $\mathcal{Q}$, **for each** iteration $i \in [1..m]$

compute the minimal number of mismatches needed to reach q at step i.

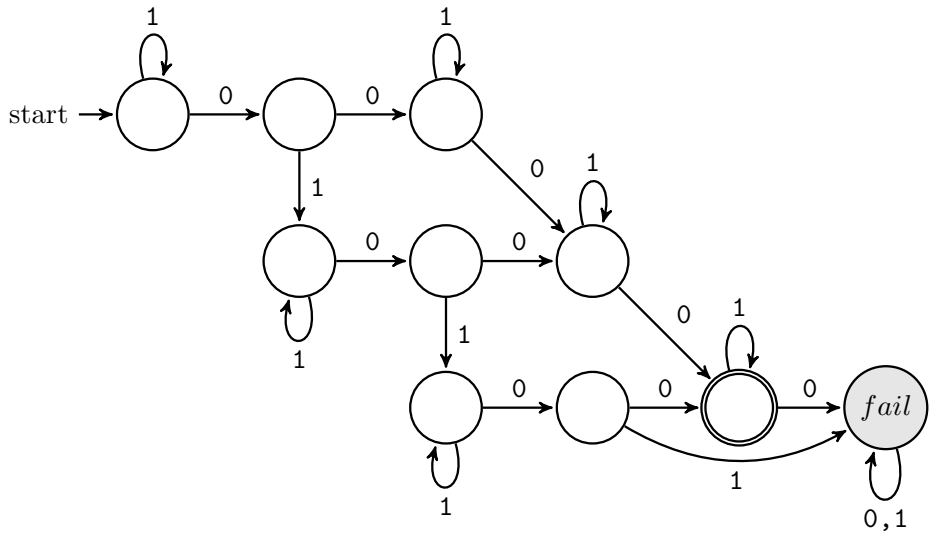
The lossless condition holds **iff** at step m, all non-final states have a number of mismatches greater than k.

Time complexity: $\mathcal{O}(|\mathcal{Q}| \cdot \mathit{readlength})$; Space complexity: $\mathcal{O}(|\mathcal{Q}|)$

Lossless framework: Separating reading errors and SNPs

- The method can be extended in order to split reading errors and SNPs.
- Product with an automaton that restricts the set of alignments recognized by the seed to those having an established combination of errors.

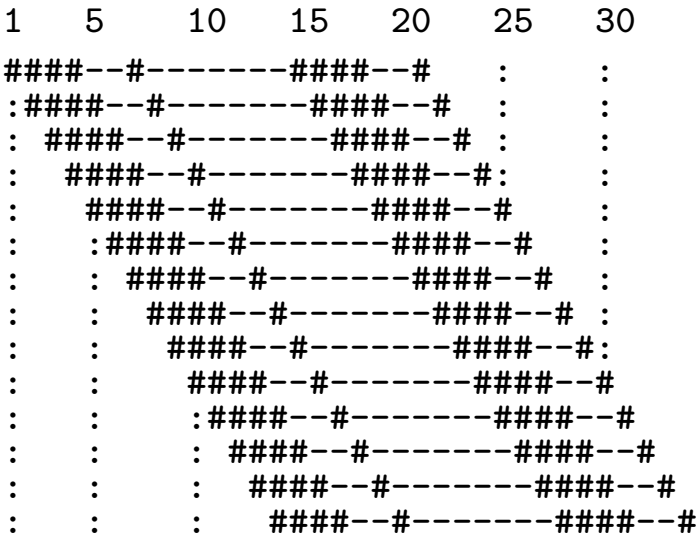
Example: automaton for 1 SNP and 2 color substitutions



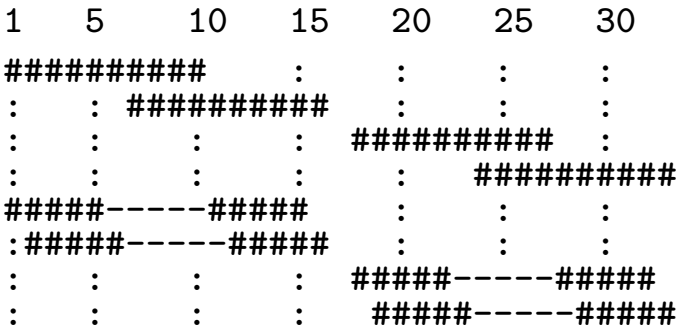
EXAMPLE 1: Lossless seeds

for 1 SNP and 2 reading errors

1-Lossless-14p



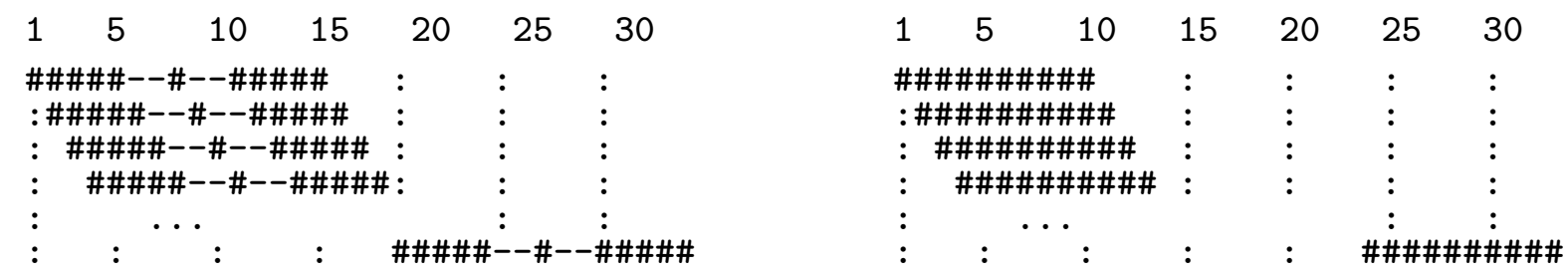
2-Lossless-8p



EXAMPLE 2: Lossless seed family

for 1 indel and 1 SNP, 1 indel and 1 reading error, 2 SNPs, or 1 SNP and 2 reading errors respectively

2-LOSSLESS-INDEL



(all possible placements)

Seed design framework for mapping SOLiD reads

- Context and motivation
- Seed design
  - Background
  - Our contribution
    - Positioned seeds
    - Lossy seeds
    - Lossless seeds
- Experiments
- Conclusions

Comparison with other software I

Comparison with popular read mapping tools, implementing various approaches

- **Bowtie** [Langmead et al, Genome Biology 2009]. Based on the Burrows-Wheeler transform.
- **BWA** [Li et al, Bioinformatics 2009]. Based on the Burrows-Wheeler transform.
- **MAQ** [Li et al, Genome Research 2008]. Uses few light-weight seeds allowed to hit in the initial part of the read.
- **SHRiMP** [Rumble et al, PLoS Comp Bio 2009]. Uses spaced seed family, multi-hit method, SIMD filter. Hashing reads rather than reference genome.
- **PerM** [Chen et al, Bioinformatics 2009]. Uses lossless "periodic" seeds.

vs

- **SToRM** – the implementation of our approach. Uses SIMD bandwidth alignment filter and spaced seeds designed as explained so far.

Experimental results I

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
Default setup for each tool

Program	Mapped reads	Execution time
Bowtie	553,140 (43.20%)	0m50s
BWA	422,550 (33.00%)	0m38s
MAQ	616,497 (48.14%)	1m20s
PerM	418,524 (32.68%)	0m31s
SHRiMP 1.3.2	663,923 (51.85%)	7m56s
SHRiMP 2.0	709,146 (55.38%)	1m22s
SToRM	839,633 (65.57%)	2m10s

Experimental results I

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
Default setup for each tool

Program	Mapped reads	Execution time
Bowtie	553,140 (43.20%)	0m50s
BWA	422,550 (33.00%)	0m38s
MAQ	616,497 (48.14%)	1m20s
PerM	418,524 (32.68%)	0m31s
SHRiMP 1.3.2	663,923 (51.85%)	7m56s
SHRiMP 2.0	709,146 (55.38%)	1m22s
SToRM	839,633 (65.57%)	2m10s

Experimental results I

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
Default setup for each tool

Program	Mapped reads	Execution time
Bowtie	553,140 (43.20%)	0m50s
BWA	422,550 (33.00%)	0m38s
MAQ	616,497 (48.14%)	1m20s
PerM	418,524 (32.68%)	0m31s
SHRiMP 1.3.2	663,923 (51.85%)	7m56s
SHRiMP 2.0	709,146 (55.38%)	1m22s
SToRM	839,633 (65.57%)	2m10s

Experimental results I

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
Default setup for each tool

Program	Mapped reads	Execution time
Bowtie	553,140 (43.20%)	0m50s
BWA	422,550 (33.00%)	0m38s
MAQ	616,497 (48.14%)	1m20s
PerM	418,524 (32.68%)	0m31s
SHRiMP 1.3.2	663,923 (51.85%)	7m56s
SHRiMP 2.0	709,146 (55.38%)	1m22s
SToRM	839,633 (65.57%)	2m10s

Experimental results I

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
Default setup for each tool

Program	Mapped reads	Execution time
Bowtie	553,140 (43.20%)	0m50s
BWA	422,550 (33.00%)	0m38s
MAQ	616,497 (48.14%)	1m20s
PerM	418,524 (32.68%)	0m31s
SHRiMP 1.3.2	663,923 (51.85%)	7m56s
SHRiMP 2.0	709,146 (55.38%)	1m22s
SToRM	839,633 (65.57%)	2m10s

Experimental results I

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
Default setup for each tool

Program	Mapped reads	Execution time
Bowtie	553,140 (43.20%)	0m50s
BWA	422,550 (33.00%)	0m38s
MAQ	616,497 (48.14%)	1m20s
PerM	418,524 (32.68%)	0m31s
SHRiMP 1.3.2	663,923 (51.85%)	7m56s
SHRiMP 2.0	709,146 (55.38%)	1m22s
SToRM	839,633 (65.57%)	2m10s

Comparison with other software II

Comparison of seeds designed with our approach with seeds proposed by other seed-based tools

- **Seeds of SHRiMP** [Rumble et al, PLoS Comp Bio 2009]: lossy spaced seed family.
- **Seeds of PerM** [Chen et al, Bioinformatics 2009]: lossless "periodic" seeds.

vs

- Lossy and lossless spaced seeds designed as explained so far.

Seed families used in the comparison

Seed set ID	Patterns	Positions

SHRiMP-DEFAULT	`#####---#######`
	`####--###--#--####`
	`###--#--#---###--####`
	`####--#----#---#--#--####`
		All

PERM-F3-S20	`###-#--#---###-#--#---##` `####--#----####--#----##`	All

3-LOSSY-12	`####-####-####`
	`####-###--#----####`
	`####----##--##-####`
		All

3-LOSSLESS-10-24P	`####-##-####`
	`#-########-#`
	`####-#-------#-####`
		0,1,2,3,4,5,6,7,8,18,19,20
		2,12,15,16,18,19,20,21
		0,1,11,14

Experimental results II

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
All seeds used within SToRM, to ensure identical setups

Seed family	Mapped reads	Execution time
PERM-F3-S20	768,732 (60.03%)	0m55s
SHRIMP-DEFAULT	836,899 (65.36%)	2m15s
3-LOSSY-12	839,633 (65.57%)	2m10s
3-LOSSLESS-10-24P	839,072 (65.53%)	2m06s

Experimental results II

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
All seeds used within SToRM, to ensure identical setups

Seed family	Mapped reads	Execution time
PERM-F3-S20	768,732 (60.03%)	0m55s
SHRIMP-DEFAULT	836,899 (65.36%)	2m15s
3-LOSSY-12	839,633 (65.57%)	2m10s
3-LOSSLESS-10-24P	839,072 (65.53%)	2m06s

Experimental results II

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
All seeds used within SToRM, to ensure identical setups

Seed family	Mapped reads	Execution time
PERM-F3-S20	768,732 (60.03%)	0m55s
SHRIMP-DEFAULT	836,899 (65.36%)	2m15s
3-LOSSY-12	839,633 (65.57%)	2m10s
3-LOSSLESS-10-24P	839,072 (65.53%)	2m06s

Experimental results II

Data: 1,280,536 reads from *Saccharomyces cerevisiae*
All seeds used within SToRM, to ensure identical setups

Seed family	Mapped reads	Execution time
PERM-F3-S20	768,732 (60.03%)	0m55s
SHRIMP-DEFAULT	836,899 (65.36%)	2m15s
3-LOSSY-12	839,633 (65.57%)	2m10s
3-LOSSLESS-10-24P	839,072 (65.53%)	2m06s

Seed design framework for mapping SOLiD reads

- Context and motivation
- Seed design
  - Background
  - Our contribution
    - Positioned seeds
    - Lossy seeds
    - Lossless seeds
- Experiments
- Conclusions

Contributions

- A seed design framework for mapping SOLiD reads to a reference genomic sequence
  - The concept of **positioned seeds**, particularly suitable for short alignments with non-uniform error distribution
  - **A model** that captures the **statistical characteristics of the SOLiD reads**, used for the evaluation of lossy seeds
  - An efficient dynamic programming **algorithm for verifying the lossless property** of seeds with the capacity to **distinguish between SNPs and reading errors** in seed design
- A selection of "ready-to-use" seeds (seed families) (cf `http://www.lifl.fr/yass/iedera_solid`)
- An experimental read mapping software STORM (to be released)

Thank you!

Acknowledgments

ANR project CoCoGen (BLAN07-1 185484) – funding for Laurent Noé.

Valentina Boeva and **Emmanuel Barillot** (*Institut Marie Curie* Paris)
– for helpful discussions and for providing the dataset of *Saccharomyces cerevisiae* reads that we used as a testset in our study.

Martin Figeac (*Institut national de la santé et de la recherche médicale*) – for sharing insightful knowledge about the SOLiD technology.

References I

Burkhardt, S. and Kärkkäinen, J. (2003).
Better filtering with gapped q-grams.
Fundamenta Informaticae, 56(1-2):51–70.
Preliminary version in Combinatorial Pattern Matching 2001.

Keich, U., Li, M., Ma, B., and Tromp, J. (2004).
On spaced seeds for similarity search.
Discrete Applied Mathematics, 138(3):253–263.
(preliminary version in 2002).

Kucherov, G., Noé, L., and Roytberg, M. (2006).
A unifying framework for seed sensitivity and its application to subset seeds.
Journal of Bioinformatics and Computational Biology, 4(2):553–570.

Ma, B., Tromp, J., and Li, M. (2002).
PatternHunter: Faster and more sensitive homology search.
Bioinformatics, 18(3):440–445.