Seed design framework for mapping SOLiD reads

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

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

SHD 2010, ENS Paris,
March 24, 2010

Seed design framework for mapping SOLiD reads

- Background and motivation
- Seed design
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

High-throughput sequencing technologies

High-throughput sequencing technologies

- 454 Life Sciences, Illumina/Solexa, Applied Biosystems (SOLiD), ..., Helicos (Heliscope), ..., IBM, DNA Nanoarrays, ...
- Sequencing human genome: >\$100 million in 2001, ... yesderday \$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

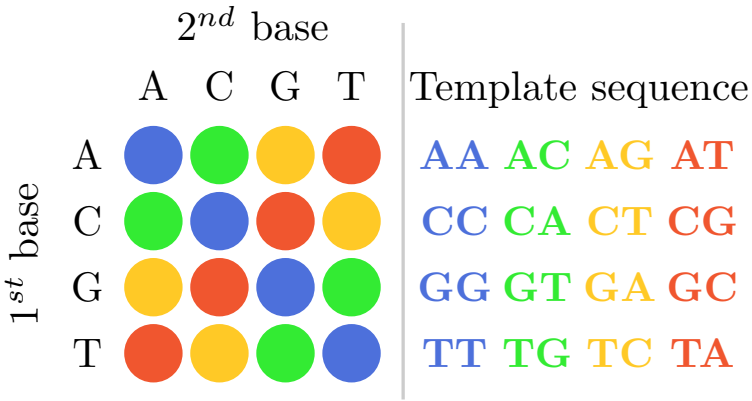
- 454 Life Sciences, Illumina/Solexa, Applied Biosystems (SOLiD), ..., Helicos (Heliscope), ..., IBM, DNA Nanoarrays, ...
- Sequencing human genome: >\$100 million in 2001, ... yesderday \$48,000, today \$4,400, tomorrow \$100 (?)
- "Reading" the genome by short reads of 25-250bp with redundancy

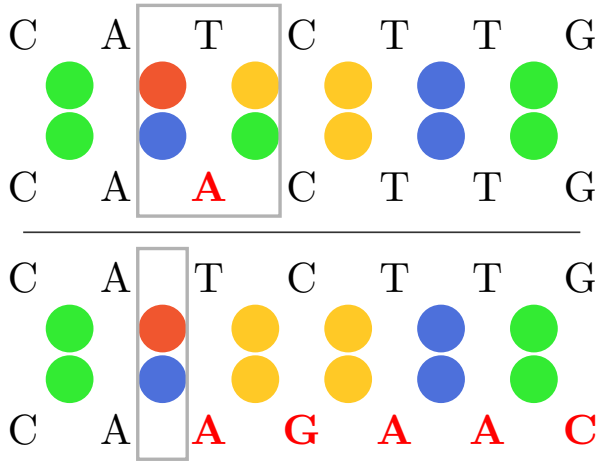
Central problem of this talk:

- Mapping reads to a reference genomic sequence

SOLiD^{TM} system (Applied Biosystems)

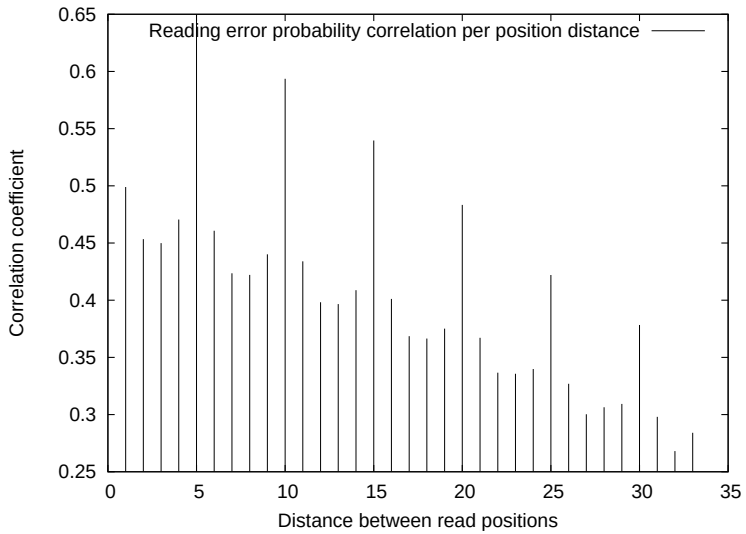
- 2-base encoding of 35bp reads $\Rightarrow$ error-correcting capability helping to reduce the error rate and to better distinguish between sequencing errors and SNPs
- Mappings of color sequences must be implicitly interpreted as nucleotide alignments

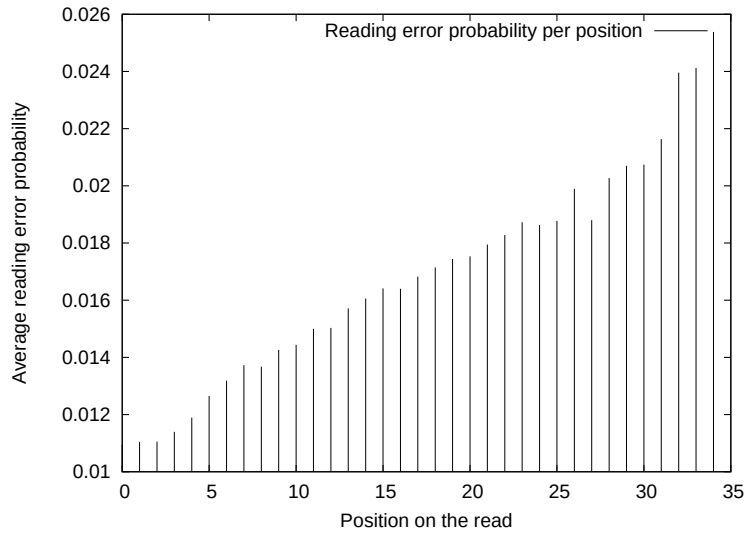




Properties and artifacts of SOLiD technology

- SNPs correspond to 2 adjacent mismatches
- 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 properites of SOLiD reads

Seed design framework for mapping SOLiD reads

- Background and motivation
- **Seed design**
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

Spaced seeds: background

Seed = pattern of matching characters which is defined to be an evidence of a significant alignment

Spaced seeds: background

Seed = pattern of matching characters which is defined to be an evidence of a significant alignment

Ex: seed ##### does not hit this alignment

```
ATCAGTGCAATGCTCAAGA
||.||.||.||||:||.||
ATTAGCGCGATGCGCAGGA
```

Spaced seeds: background

Seed = pattern of matching characters which is defined to be an evidence of a significant alignment

Ex: spaced seed ##-##-#

```
##-##-#
ATCAGTGCAATGCTCAAGA
||.||.||.||||:||.||
ATTAGCGCGATGCGCAGGA
```

Spaced seeds: background

Seed = pattern of matching characters which is defined to be an evidence of a significant alignment

Ex: spaced seed ##-##-#

```
   ##-##-#
ATCAGTGCAATGCTCAAGA
||.||.||.||||:||.||
ATTAGCGCGATGCGCAGGA
```

Spaced seeds: background

Seed = pattern of matching characters which is defined to be an evidence of a significant alignment

Ex: spaced seed ##-##-#

```
     ##-##-#
ATCAGTGCAATGCTCAAGA
||.||.||.||||:||.||
ATTAGCGCGATGCGCAGGA
```

Spaced seeds: background

Seed = pattern of matching characters which is defined to be an evidence of a significant alignment

Ex: spaced seed ##-##-#

```
                    ##-##-#
ATCAGTGCAATGCTCAAGA
||.||.||.||||:||.||
ATTAGCGCGATGCGCAGGA
```

Spaced seeds: background

- 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, ...]

Spaced seeds: background

- 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: background

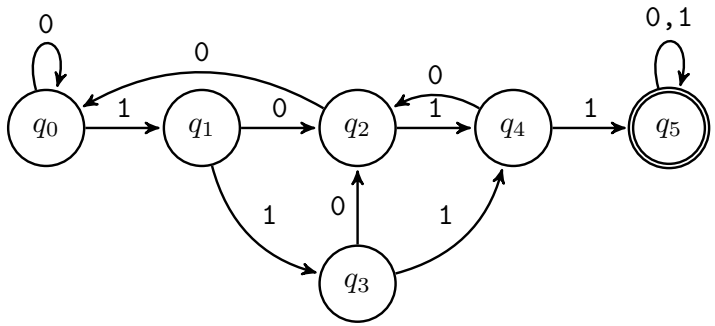
- Spaced seeds are more likely to hit an alignment than contiguous seeds of the same weight (= nb of #) $\Rightarrow$ more *sensitive* search [PatternHunter 2002, Yass 2004, ...]
- Using seed families (several seeds simultaneously such that a hit of at least one of them is sufficient) further improves 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.

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]
  - "Good" mappings are modeled by a *Hidden Markov Models with emitting transitions*
  - A seed, or a seed family, is modeled by a *seed automaton*

    Example: The automaton $\mathcal{Q}$ of the spaced seed $\pi =$ `#-##`



- Generates seeds patterns and selects the most sensitive seed families

Seed design framework for mapping SOLiD reads

- Background and motivation
- **Seed design**
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

Position-restricted seeds

Motivation:

- Reads are short sequences of **fixed length**
- *Reminder:* 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

Idea:

- Favor hits on the positions of the read where matches are more likely to be significant

Position-restricted seeds

Position-restricted 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        #–##
                    #–##
                         #–##
                              #–##
```

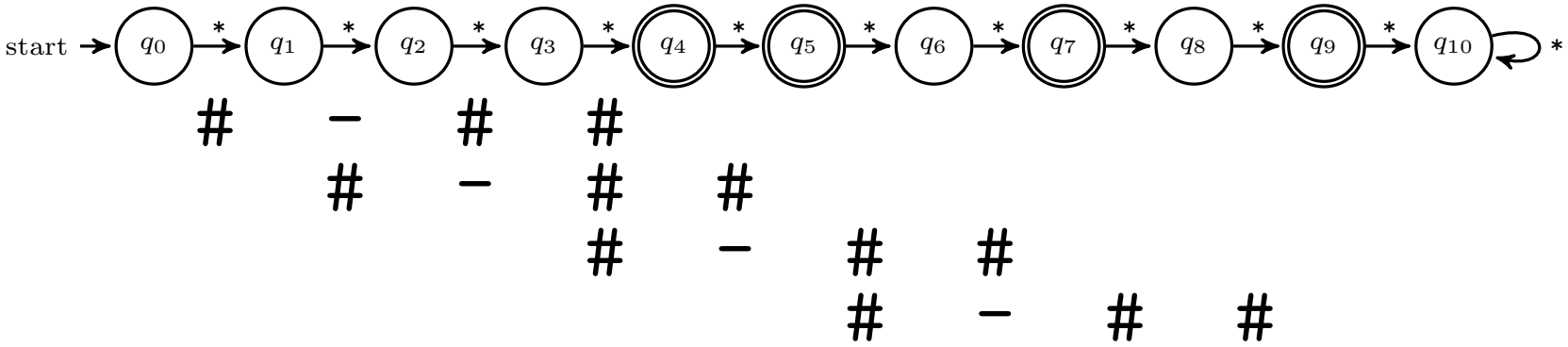
Position-restricted seeds

Restricting the seed

To take into account the set P of allowed positions, we compute the **product** of $\mathcal{Q}$ with **an automaton** λ_P

- consisting of **a linear chain** of $m+1$. (m = read length)
- whose final states are: $F = \{q_i : i - s \in P\}$ (s = the span of the concerned seed π).

Example: $\pi =$ `#-##` (the span $s = 4$), $P_\pi = \{0, 1, 3, 5\}$, $m = 10$



Seed design framework for mapping SOLiD reads

- Background and motivation
- **Seed design**
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

Designing seeds for SOLiD read mapping: lossy vs. lossless

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

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

Designing seeds for SOLiD read mapping: challenges

- There are two independent sources of errors in reads with respect to the reference genome:
  - **reading errors** (misread colors)
  - **SNPs/indels**, i.e., *bona fide* differences between the reference genome and sequenced data
- Both error types must be handled, and their *superposition* considered in the design process

Our contribution to seed design for mapping SOLiD reads

- We design **position-restricted seeds** for mapping SOLiD reads, both in the lossy and lossless settings

Our contribution to seed design for mapping SOLiD reads

- We design **position-restricted seeds** for mapping SOLiD reads, both in the lossy and lossless settings

- In the *lossy* framework:
  - We represent **each of the two error sources** (SNPs and reading errors) by a **separate Hidden Markov Model**, **combined in a model which allows all error types to be cumulated in the resulting sequences**
  - We design sensitive seeds w.r.t. this combined model

Our contribution to seed design for mapping SOLiD reads

- We design **position-restricted seeds** for mapping SOLiD reads, both in the lossy and lossless settings

- In the *lossy* framework:
  - We represent **each of the two error sources** (SNPs and reading errors) by a **separate Hidden Markov Model**, **combined in a model which allows all error types to be cumulated in the resulting sequences**
  - We design sensitive seeds w.r.t. this combined model

- In the *lossless* framework:
  - We are allowed to **distinguish between reading errors and SNPs** of a seed (e.g: lossless for 1 SNP and 2 reading errors)
  - This distinction is possible thanks to an automaton that restricts the set of alignments to those with the established number of errors
  - We apply a **fast algorithm for verifying the lossless property directly on the seed automaton** to design lossless seeds

Seed design framework for mapping SOLiD reads

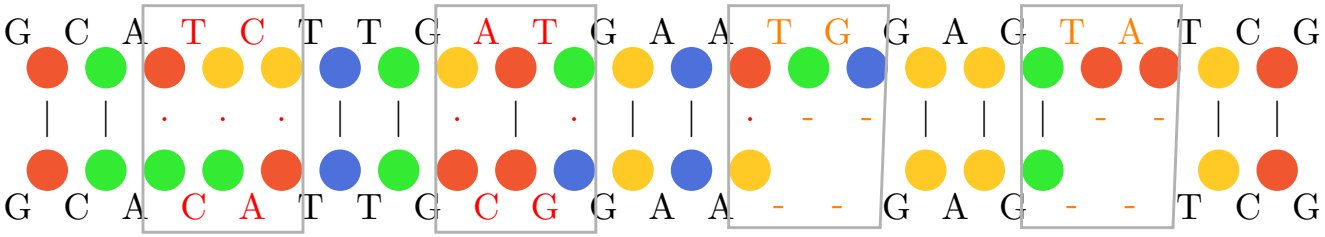
- Background and motivation
- **Seed design**
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

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: Biological variations model

DNA modifications reflected in the color sequence:



Consecutive mutations

$$\begin{array}{ccccccccc} B_1 & & B_2 & & B_3 & & B_4 & & B_5 \\ | & c_1 & & c_2 & & c_3 & & c_4 & | \\ | & c'_1 & & c'_2 & & c'_3 & & c'_4 & | \\ B_1 & & B'_2 & & B'_3 & & B'_4 & & B_5 \end{array}$$

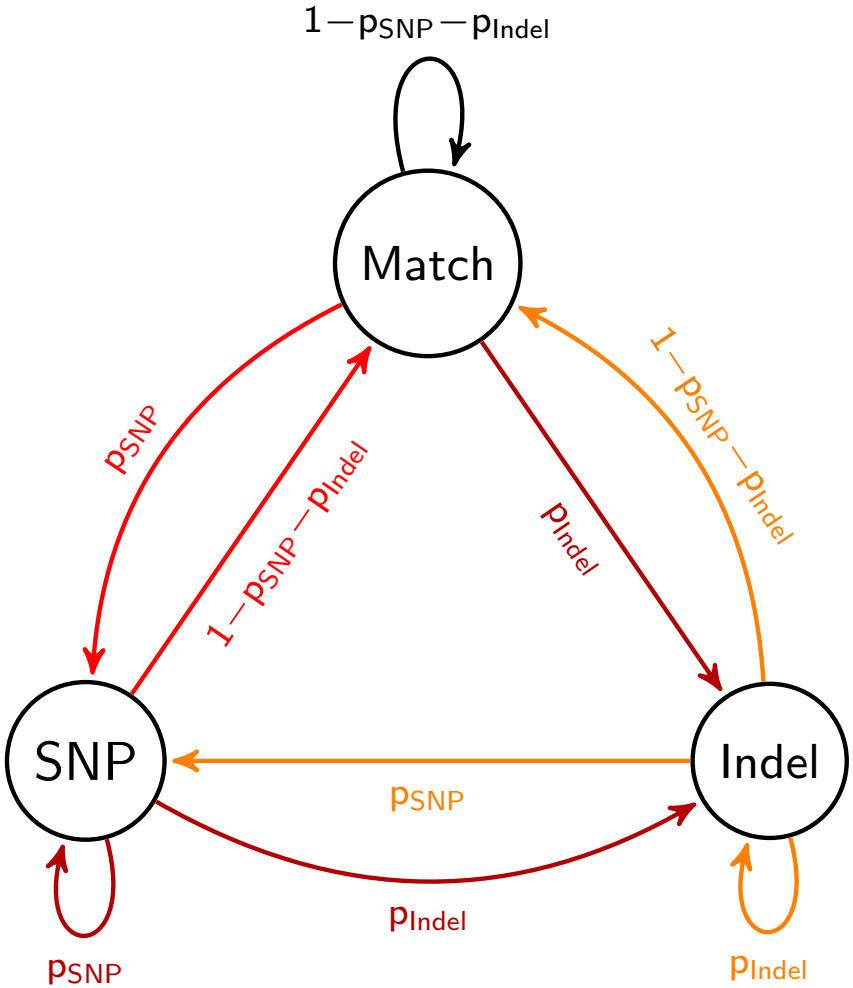
$c_1 \neq c'_1$, $c_4 \neq c'_4$
for $i = 2, 3$, $c_i \neq c'_i$ in $3/4$ cases

Consecutive indels

$$\begin{array}{ccccccccc} B_1 & & B_2 & & B_3 & & B_4 & & B_5 \\ | & c_1 & & c_2 & & c_3 & & c_4 & | \\ | & c'_1 & & & & & & & \\ B_1 & & - & & - & & - & & B_5 \end{array}$$

$c'_1 \neq c_1$ in $3/4$ cases

Lossy framework: Biological variations model ($M_{SNP/I}$)



States refer to **DNA alignment**

Emitted symbols refer to **color alignment**

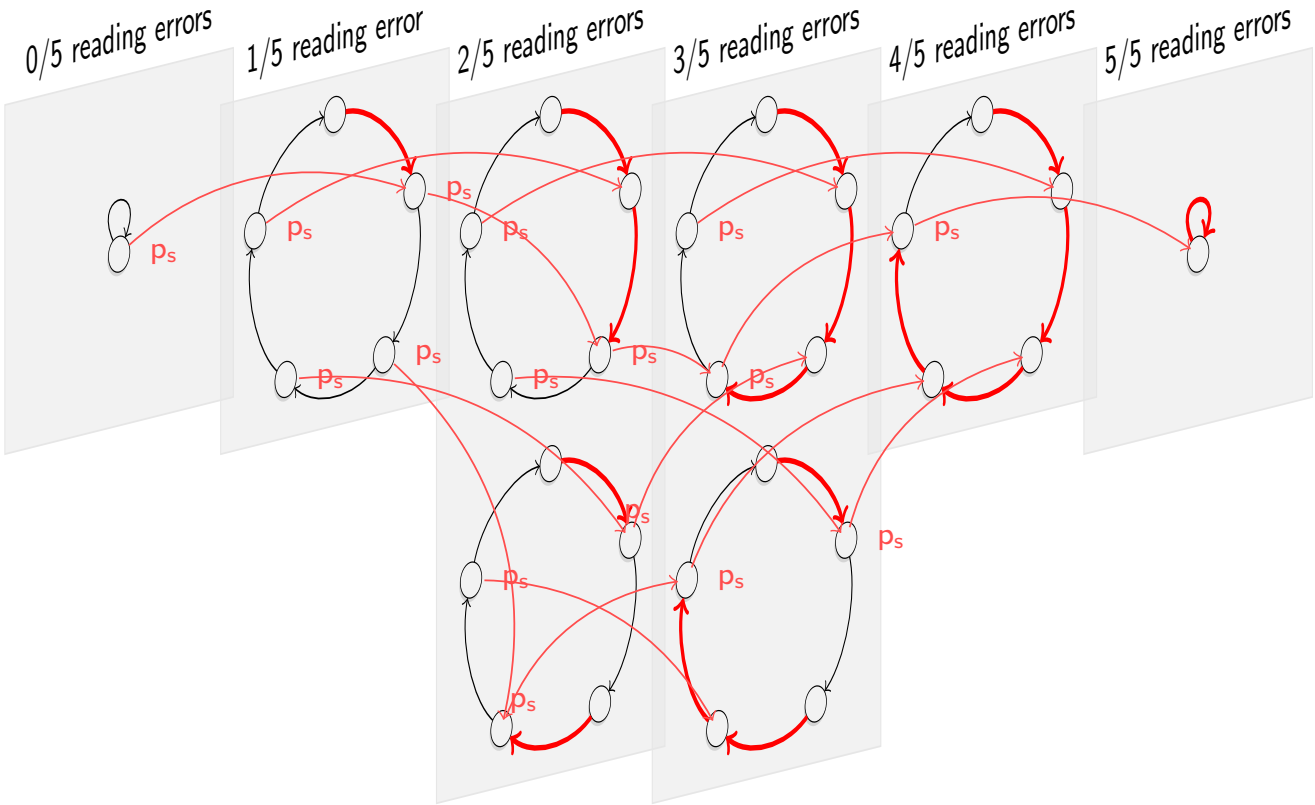
Legend (transitions):

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

Lossy framework: Reading errors model (M_{RE})

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

Reminder: Errors tend to appear with a periodicity of 5



Legend (transitions): **periodic errors, fixed high error probability**; switching to a high error probability; small error probability, increasing towards the end of the read.

Lossy framework: Combined model

The model which combines both error sources is **the product of** $M_{SNP/I}$ **and** M_{RE}.

How are errors cumulated (**example**):

$M_{SNP/I}$	M M M E E M M M E E M M M E I M M M M M I M M M M M M
$\times$	
M_{RE}	M M M M M M M M M E M M M M E M M M M E M E M E M E E
$=$	
$M_{(SNP/I)\times RE}$	M M M E E M M M E E M M M E I M M M M E I E M E M E E

Lossy framework: Seed selection according to 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].

Using the dynamic programming technique of [Kucherov et al., 2006] within IEDERA, we select **the most sensitive seeds w.r.t. the specified model**.

Seed design framework for mapping SOLiD reads

- Background and motivation
- **Seed design**
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

Lossless framework: Computing the lossless property

Lossless seeds have the capacity to hit **all** alignments containing up to an established number of errors.

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

Lossless framework: Computing the lossless property

Straightforward way: construct a deterministic automaton recognizing the set of all target alignments and test if the language of this automaton is included in the language of the automaton $\mathcal{Q}$ of the seed – **unfeasible in practice**.

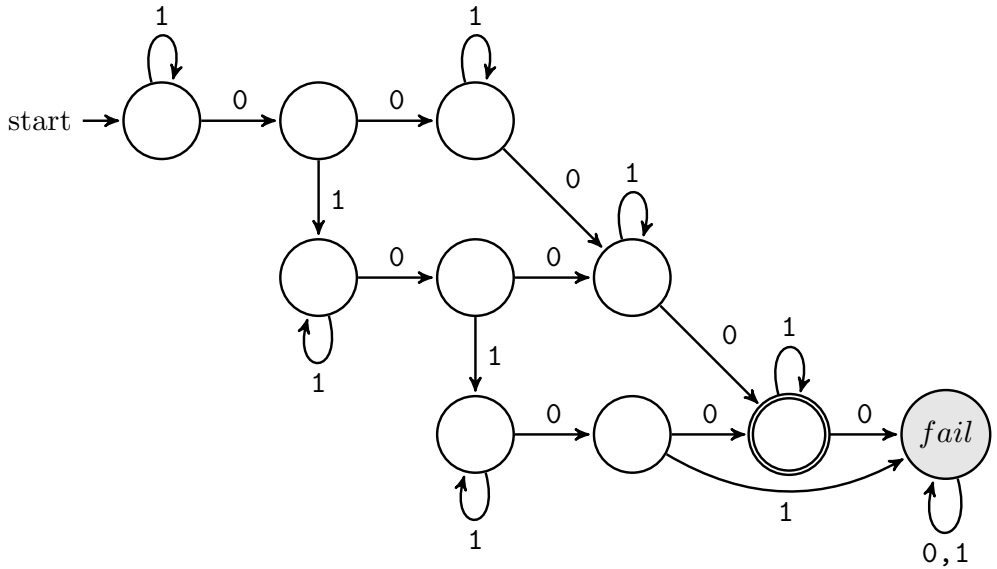
We propose **an efficient dynamic programming algorithm directly applied to** $\mathcal{Q}$ that can verify the **inclusion**:

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.

Example: automaton for 1 SNP and 2 color substitutons



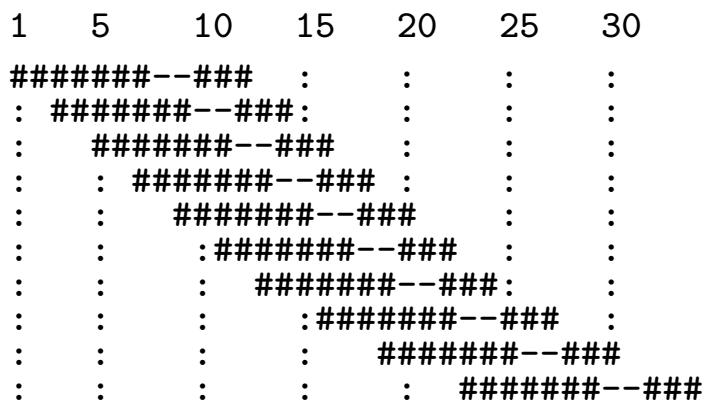
Designing lossless seeds for k SNPs and h color substitutions

Seed design framework for mapping SOLiD reads

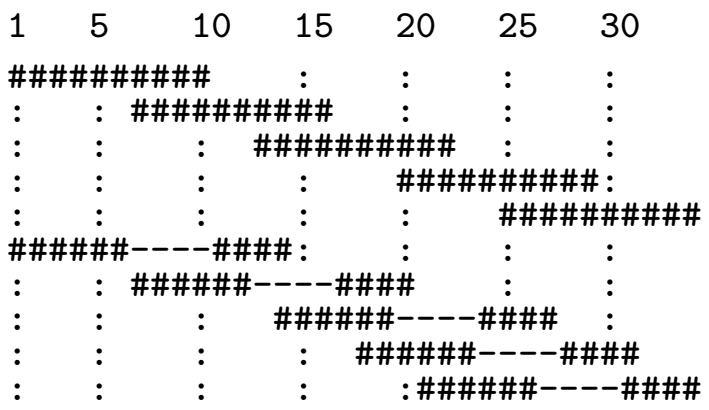
- Background and motivation
- Seed design
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

Lossy seeds restricted to 10 positions

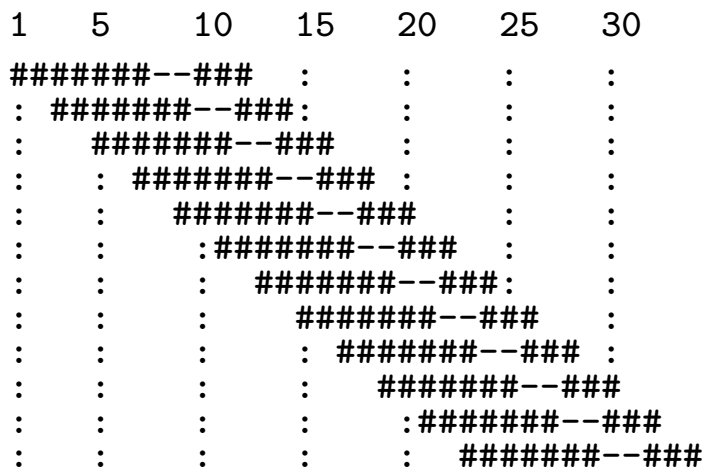
1-LOSSY-10P: sensitivity 0.9543



2-LOSSY-10P: sensitivity 0.9627

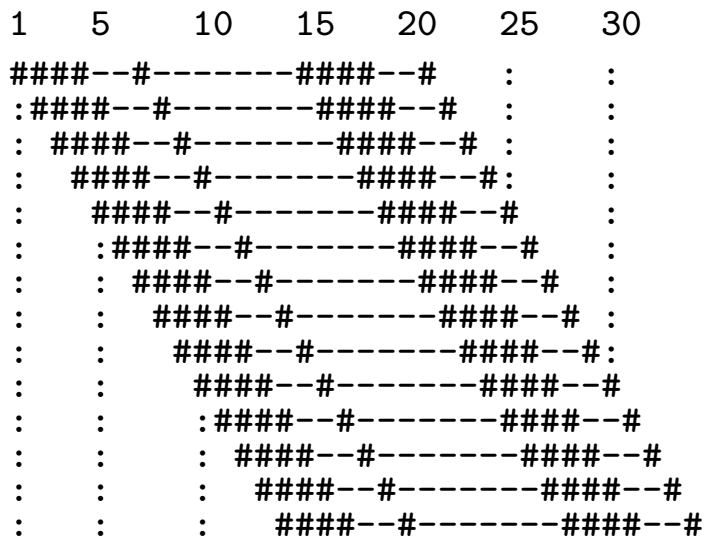


Lossy seeds restricted to 12 positions

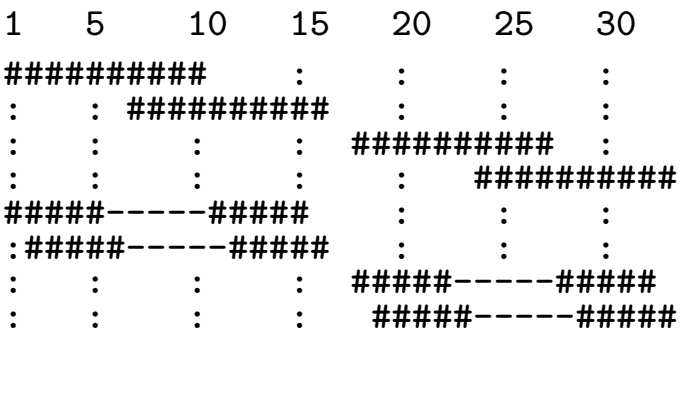
1-Lossy-12p: sensitivity 0.9626	2-Lossy-12p: sensitivity 0.9685
	

Lossless seeds for 1 SNP and 2 reading errors

1-Lossless-14p

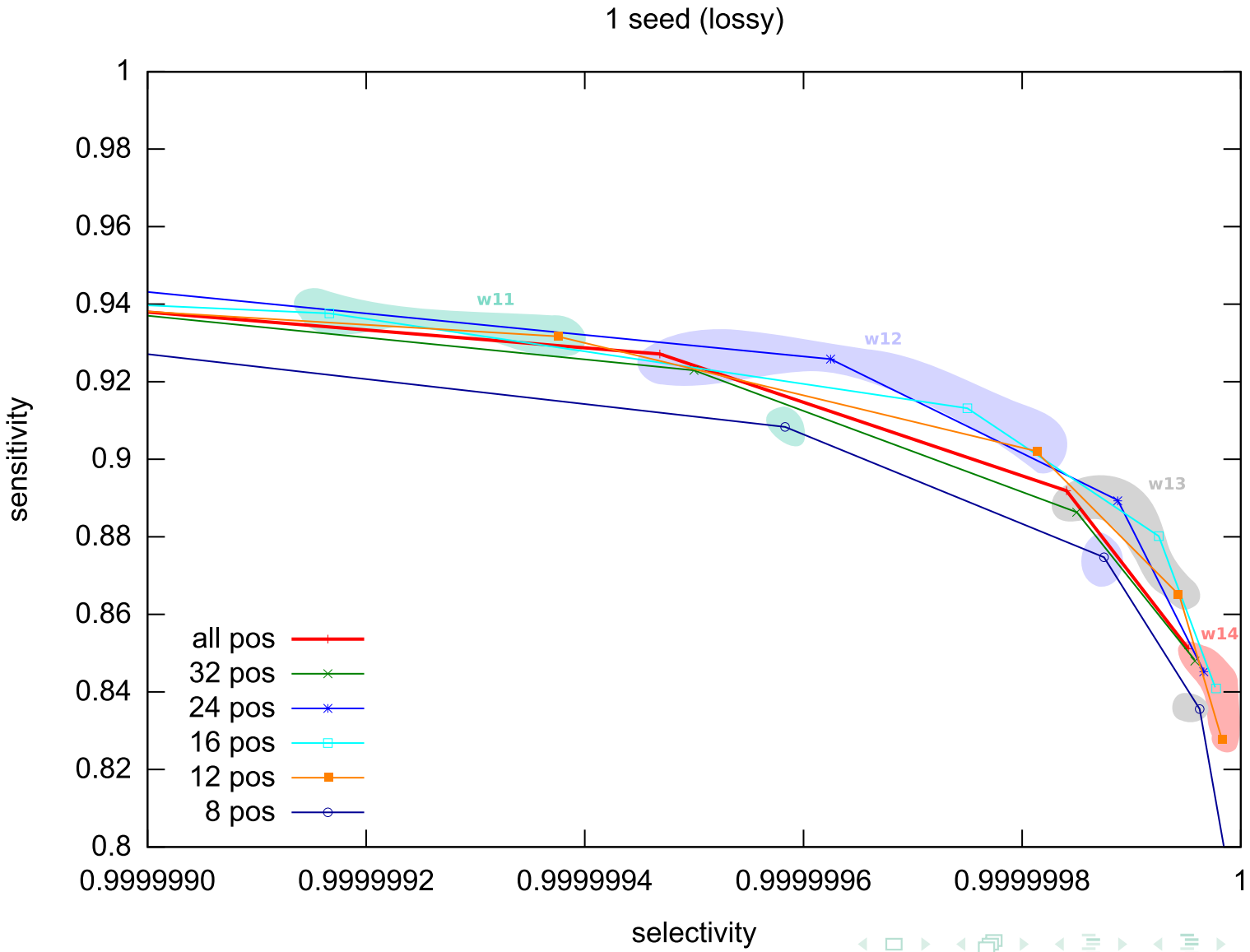


2-Lossless-8p



Comparative performance of position-restricted seeds

Theoretical sensitivity/selectivity of single seeds

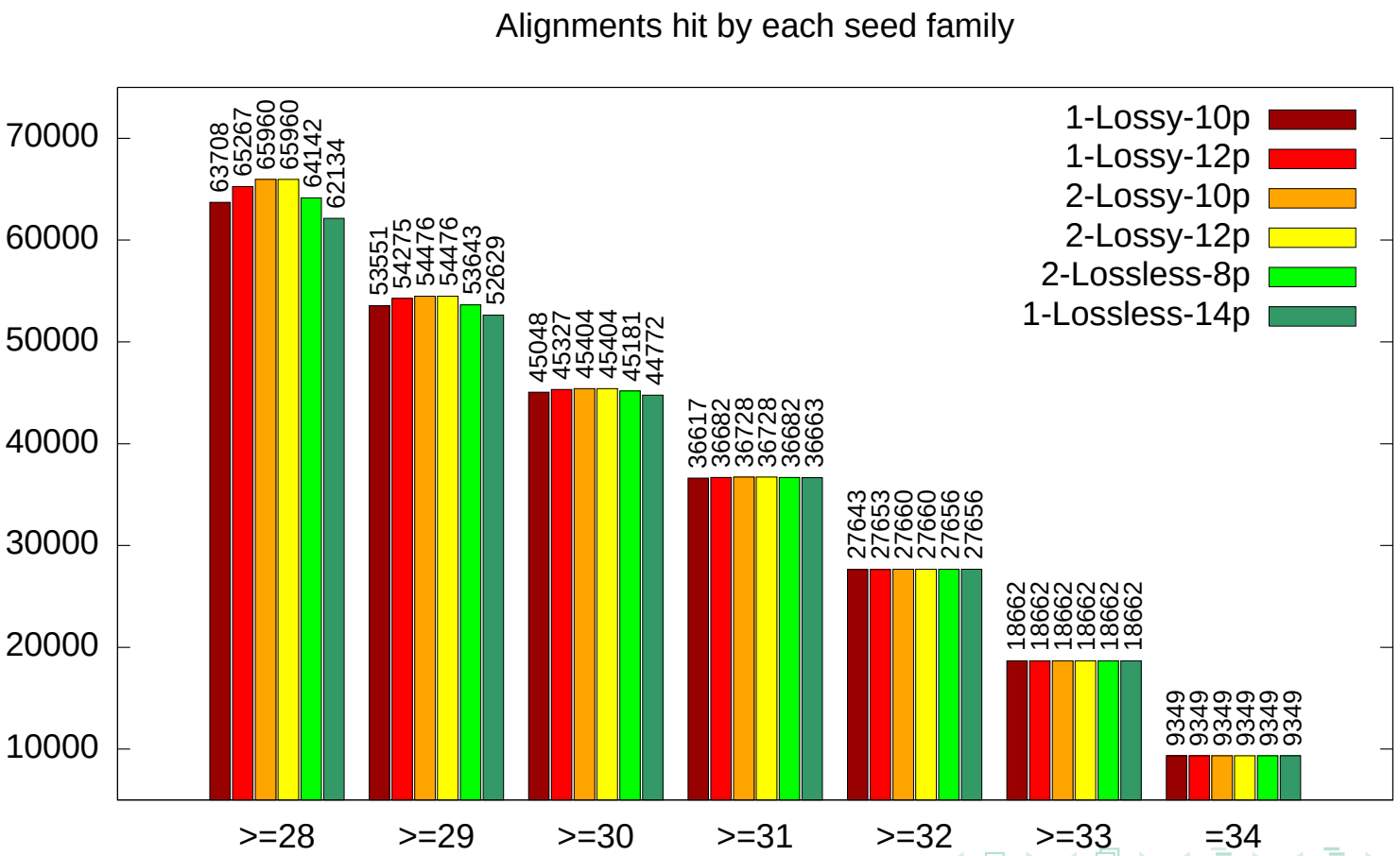


Seed comparison

Data: 100000 reads of length 34 from *S. cerevisiae*

Scoring scheme: +1 for match, 0 for color mismatch or SNP, -2 for gaps

Results: The number of read/reference alignments hit by each (single or double) seed with scores varying from 28 to 34



Comparison with other software

- our read mapping software. Uses SIMD bandwidth alignment filter.
- 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.

Experimental results

Comparable setup for the three programs. Cut-off score 46 under scoring system $(2, -3, -7, -4)$.

Program	Seed set	Mapped reads	Execution time
SHRiMP	SHRiMP-DEFAULT	663,923 (51.85%)	31m07s
PerM	LOSSLESS 5 MISMATCHES	618,554 (48.30%)	**0m25s**
our tool	PERM-F3-S20	539,772 (42.15%)	**2m02s**
our tool	SHRiMP-DEFAULT	675,308 (52.74%)	5m06s
our tool	3-LOSSY-12	677,043 (52.87%)	**4m40s**
our tool	4-LOSSY-12	**678,455 (52.98%)**	6m02s
our tool	4-Lossy-10	**679,802 (53.09%)**	30m17s

Table 1. Comparison with SHRiMP and PerM. Dataset: *S. cerevisiae* (1,280,536 reads)

Seed design framework for mapping SOLiD reads

- Background and motivation
- Seed design
  - Background
  - Position-restricted seeds
  - General approach
  - Lossy seeds
  - Lossless seeds
- Experiments
- Conclusions and perspectives

Contributions

- A seed design framework for mapping SOLiD reads to a reference genomic sequence
  - The concept of **position-restricted 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 (to be released)

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.

Thank you!

Questions?

More details in:

Noé, L., Gîrdea, M., and Kucherov, G. (2010).
Seed design framework for mapping SOLiD reads.
Proceedings of the 14th Annual International Conference on Computational Molecular Biology (RECOMB) (accepted).

References I

Biosystems, A. (2008).

A Theoretical Understanding of 2 Base Color Codes and Its Application to Annotation, Error Detection, and Error Correction. Methods for Annotating 2 Base Color Encoded Reads in the SOLiD^{TM}System.

Brejova, B., Brown, D., and Vinar, T. (2003).

Optimal spaced seeds for Hidden Markov Models, with application to homologous coding regions.

Lecture Notes in Computer Science, 2676:42–54.

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.

Sun, Y. and Buhler, J. (2005).

Designing multiple simultaneous seeds for DNA similarity search.

Journal of Computational Biology, 12:847–861.

References II

Zhou, L., Stanton, J., and Florea, L. (2008).
Universal seeds for cDNA-to-genome comparison.
BMC bioinformatics, 9(1):36.