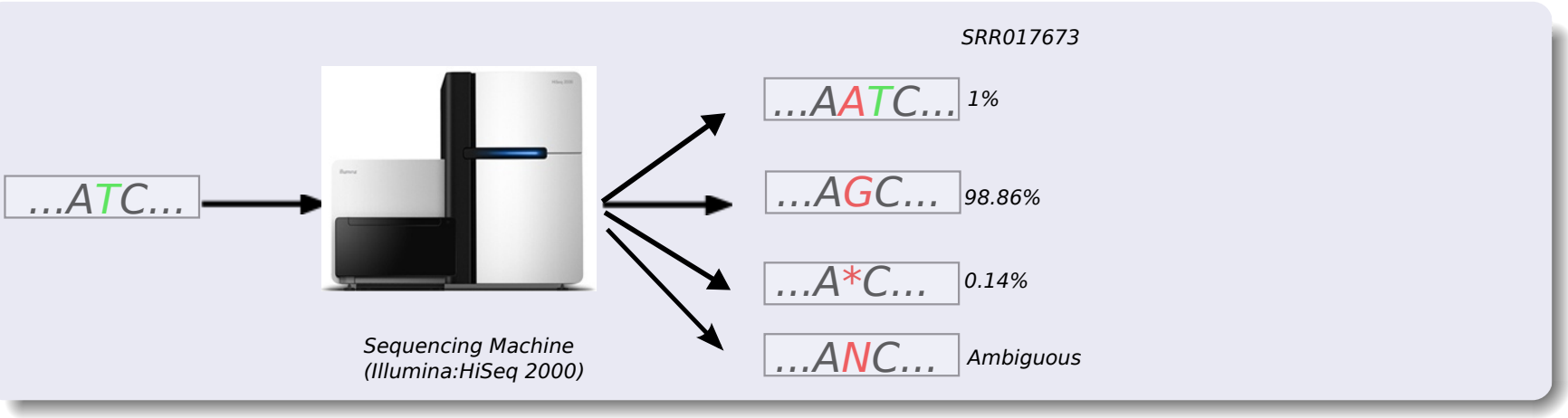
Error Correction Algorithms for Next-Generation Sequencing

Srinivas Aluru

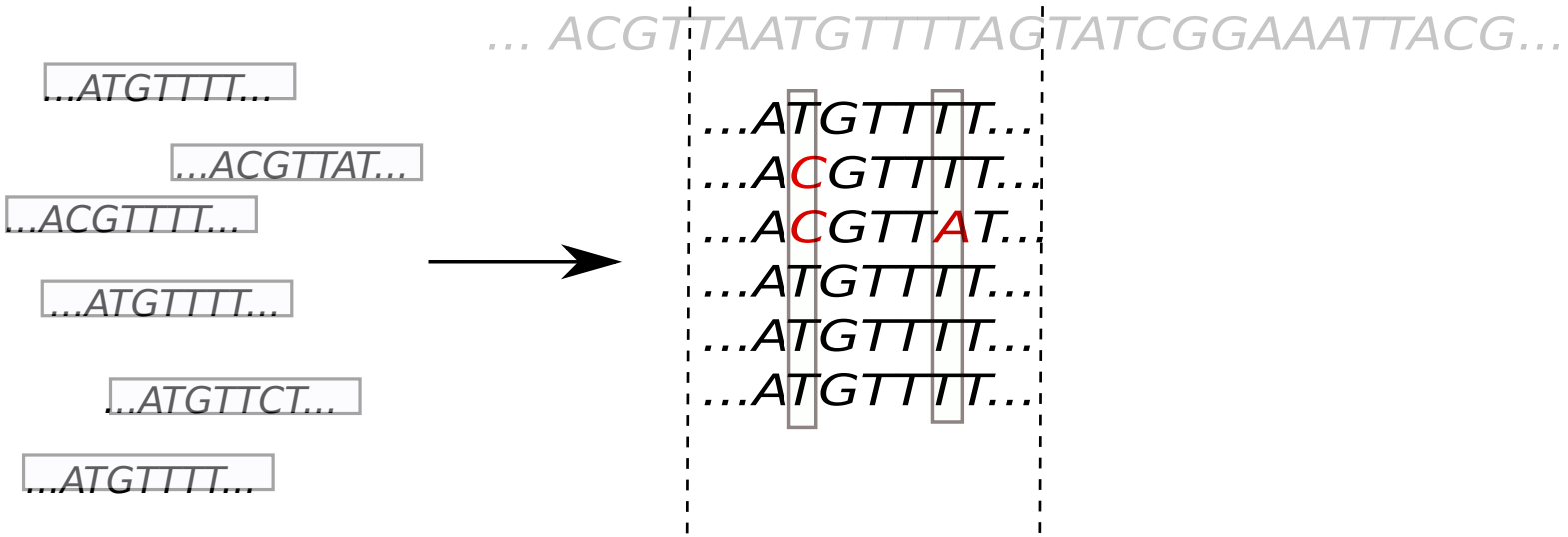
Electrical and Computer Engineering
Iowa State University

Computer Science and Engineering
Indian Institute of Technology Bombay

Error Correction



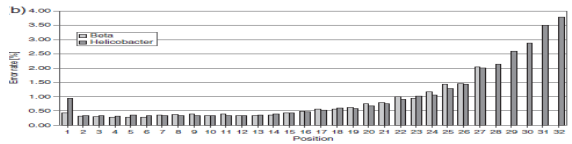
How? - Ideally


... ACGTTAATGTTTTAGTATCGGAAATTACG...
...ATGTTTT...
...ACGTTAT...
...ACGTTTT...
...ATGTTTT...
...ATGTTCT...
...ATGTTTT...
...ATGTTTT...
...ACGTTTT...
...ACGTTAT...
...ATGTTTT...
...ATGTTTT...
...ATGTTTT...

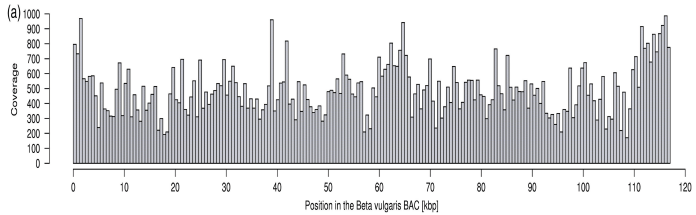
Challenges

- unknown reference genome
- massive number of reads
- non-uniform error distribution
- non-uniform genome sampling
- polymorphisms
- repeats

... ACGTTTAAAAACGTACCAAGTTACGT...



Dohm et al, 2008



465x

Error Correction Approaches

- SAP-formulation – Chaisson *et al.*, 2004, 2008; Chin *et al.*, 2009; *Quake* (Kelley *et al.*, 2010); *Reptile* (Yang *et al.*, 2010)
- Suffix-trie – *SHREC* (Schröder *et al.*, 2009), *Hybrid SHREC* (Salmela and Schröder, 2010)
- Alignment based – *CORAL* (Salmela, 2011)

k-Spectrum Approach

Formulation

- k-Spectrum: Set of k-length substrings from reads ($k$mers)
- Valid $k$mer $\rightarrow$ frequency $\geq M$
- Error-free read $\rightarrow$ contains no invalid $k$mers
- Goal: edit each erroneous read to make all $k$mers valid

k-Spectrum Approach

Formulation

- k-Spectrum: Set of k-length substrings from reads ($k$mers)
- Valid $k$mer $\rightarrow$ frequency $\geq M$
- Error-free read $\rightarrow$ contains no invalid $k$mers
- Goal: edit each erroneous read to make all $k$mers valid

Strategy

- Edit invalid $k$mers to valids within short Hamming distance
- Hamming graph: (u, v) if $hd(u, v) \leq d$
- Constant time retrieval/memory intensive.

Choosing M

Chin *et al.* derived optimum M (minimizing FP+FN) assuming

- uniform genome sampling
- uniform error distribution
- equal mutation rate, e.g., $A \rightarrow \{C, G, T\}$

Choosing M

Chin *et al.* derived optimum M (minimizing FP+FN) assuming

- uniform genome sampling
- uniform error distribution
- equal mutation rate, e.g., $A \rightarrow \{C, G, T\}$

Quake (Kelley *et al.*, 2010)

- calculate the weight W of each $k$mer K, let K_i be an instance:
  $W(K) = \sum_i W(K_i) = \sum_i \prod_{j=0}^{k-1} \Pr(\text{quality score}(K_i[j]))$
- histogram of the $k$mer weights $\rightarrow$ threshold M

Key Challenges

Technical Challenges

- Multiple correction choices leading to ambiguity
- Small $k \rightarrow$ high frequency; Large $k \rightarrow$ less ambiguity.
- Combinatorially explosive search space
- Large memory & run-time

Key Challenges

Technical Challenges

- Multiple correction choices leading to ambiguity
- Small $k \rightarrow$ high frequency; Large $k \rightarrow$ less ambiguity.
- Combinatorially explosive search space
- Large memory & run-time

Human Follies

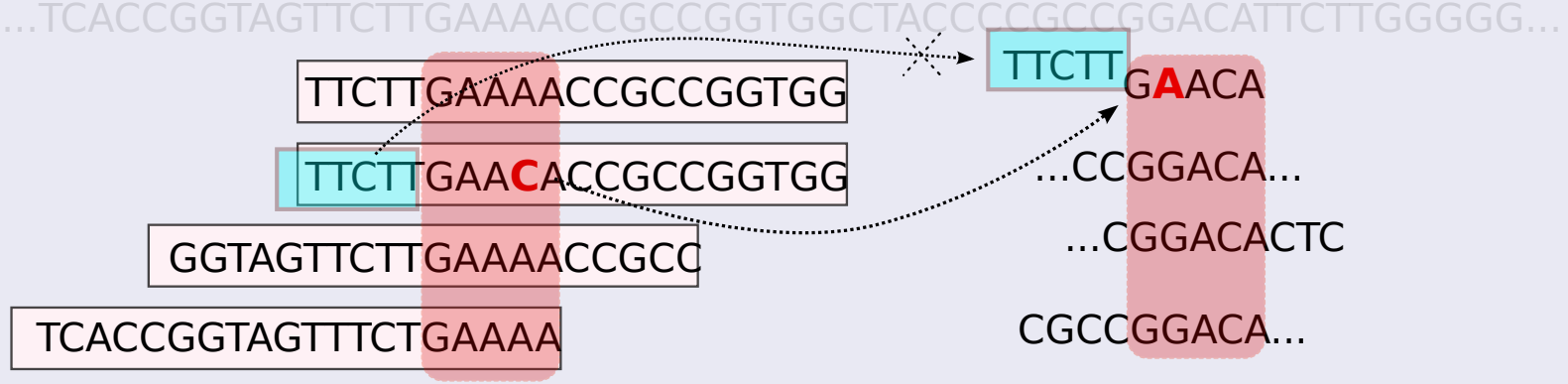
- Validation – Choose parameters with knowledge of answer
- Wrong metrics
  - Claim victory for flagging errors
  - Claim victory for correcting errors but ignore errors introduced

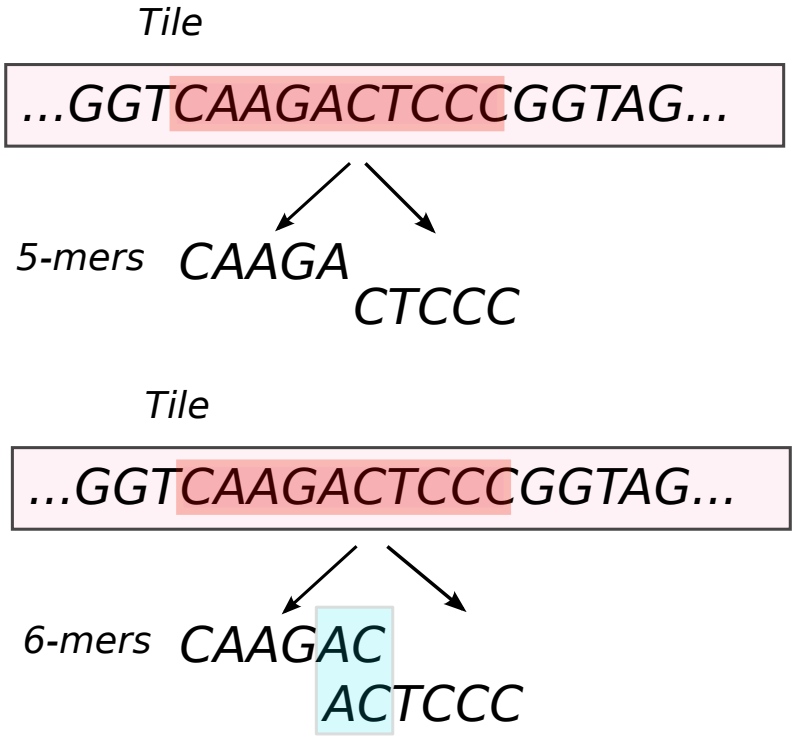
Reptile – Key Ideas

Idea 1: Using context to resolve ambiguity

Reptile – Key Ideas

Idea 1: Using context to resolve ambiguity

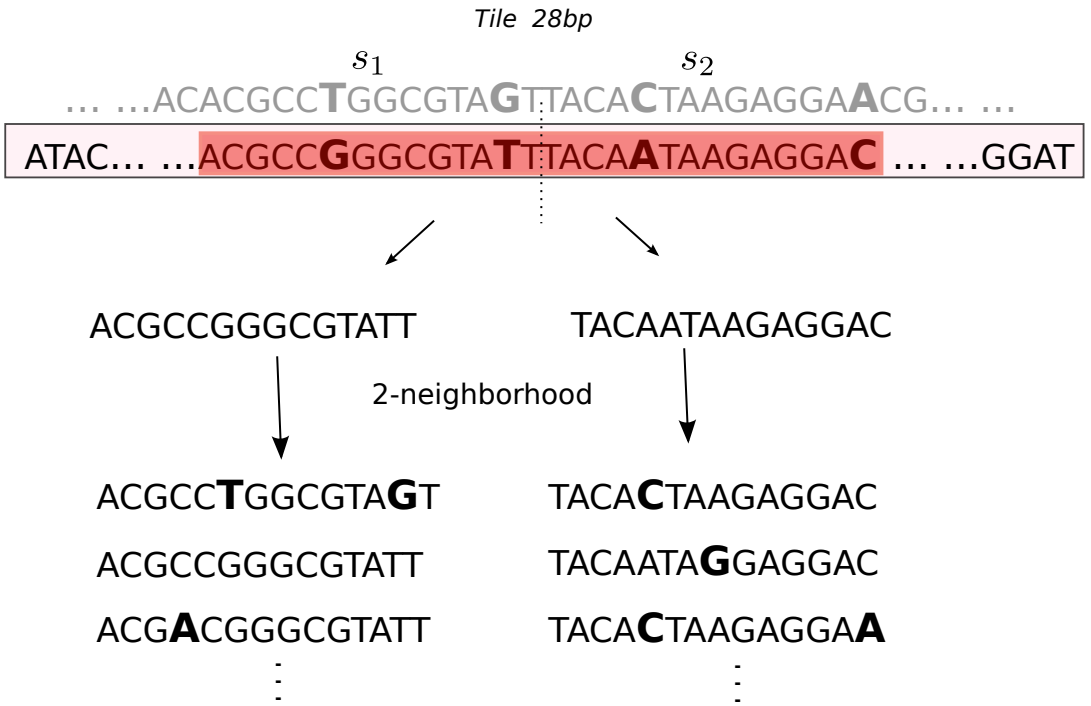




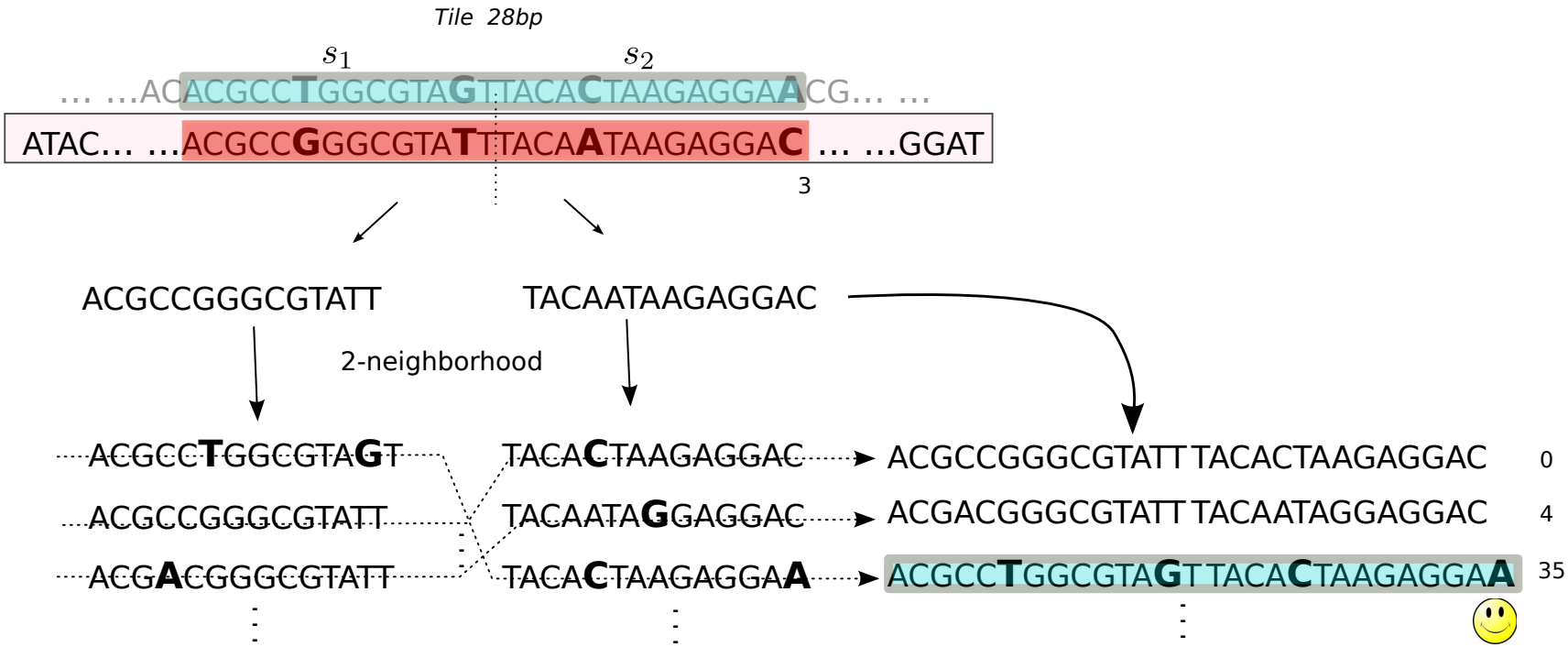
- Let α and β be two strings such that $\alpha[(|\alpha| - l) : (|\alpha| - 1)] = \beta[0 : (l - 1)]$ for some $0 \leq l < \min(|\alpha|, |\beta|)$.
- The *l-concatenation* $\gamma = \alpha||_l \beta$ satisfies:
  - $\gamma[0 : (|\alpha| - 1)] = \alpha$
  - $\gamma[(|\gamma| - |\beta|) : (|\gamma| - 1)] = \beta$.
- $t = \alpha ||_l \beta$ $(0 \leq l < k)$ is a *tile* of read r

  if t is a substring of r, and $|\alpha| = |\beta| = k$.

The d-neighborhood of s: $\{s' \mid hd(s, s') \leq d\}$.



The d-neighborhood of s: $\{s' \mid hd(s, s') \leq d\}$.



Idea 2 – Space reduction for d-neighborhood retrieval

Bucketing Strategy

- Divide the k indices of a $k$mer into $c > d$ blocks
- Sort k-spectrum by ignoring indices from d blocks for each of the $\binom{c}{d}$ choices
- Two d-neighbors differ in at most d blocks; they fall in the same bucket in at least one of the sorted lists
- Randomize the k indices to improve uniformity in bucket sizes

Complexity

- Run time
  - Expected number of elements in a bucket is $h \leq |K_s|/4^{k-d\lceil k/c \rceil}$
  - $\binom{c}{d} h \log |K_s|$ expected time to retrieve d-neighbors
- Memory: $\binom{c}{d}|K_s|$

A case study

E. coli: 20.8 Million 36bp reads 160x
Space Reduction: 9 GB to 560 MB

Idea 3 – Read decomposition

tile: 10mer, 2 x 5mers; $d = 2$, 1 in each 5mer

...AAGTCGTTCGAATCTCAGTAG**T**AACAACC**G**C**GA**GGCGAAAA**TT**GTGTGGAAA**T**TTAAATTC...

r AATCTCAGTAG**C**AACAACC**C**C**TG**GGCGAAAA**GC**GTGTGGAAA**G**TT

t_0 AATCTCAGTA

Idea 3 – Read decomposition

tile: 10mer, 2 x 5mers; $d = 2$, 1 in each 5mer

...AAGTCGTTCGAATCTCAGTAGTAACAACCGCGAGGCGAAAATTGTGTGGAAATTTAAATTC...

r AATCTCAGTAGCAACAACCCCTGGGCGAAAAGCGTGTGGAAAGTT

t_0 AATCTCAGTA

t_1 CAGTAGCAAC

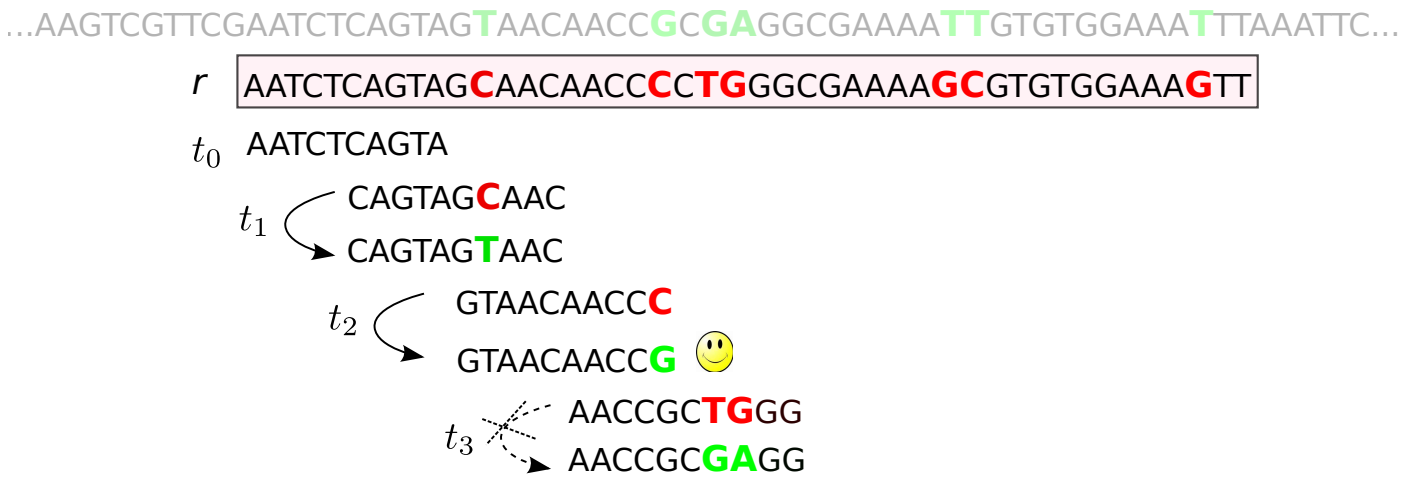
CAGTAGTAAC

t_2 GTAACAACCC

GTAACAACCG 

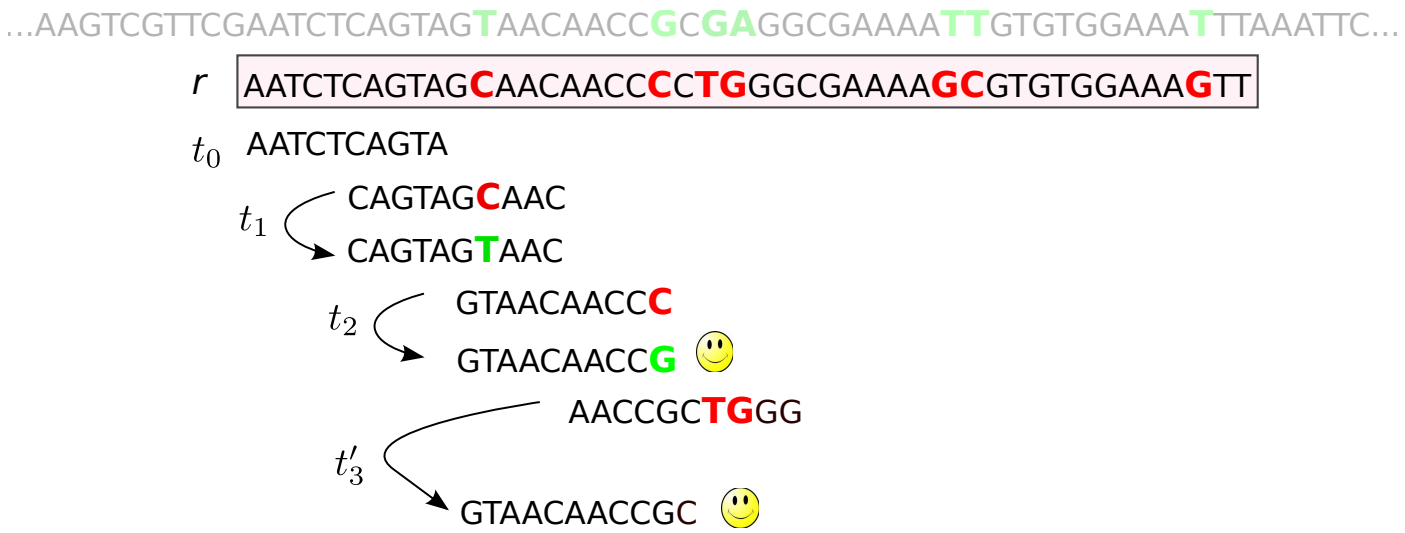
Idea 3 – Read decomposition

tile: 10mer, 2 x 5mers; $d = 2$, 1 in each 5mer



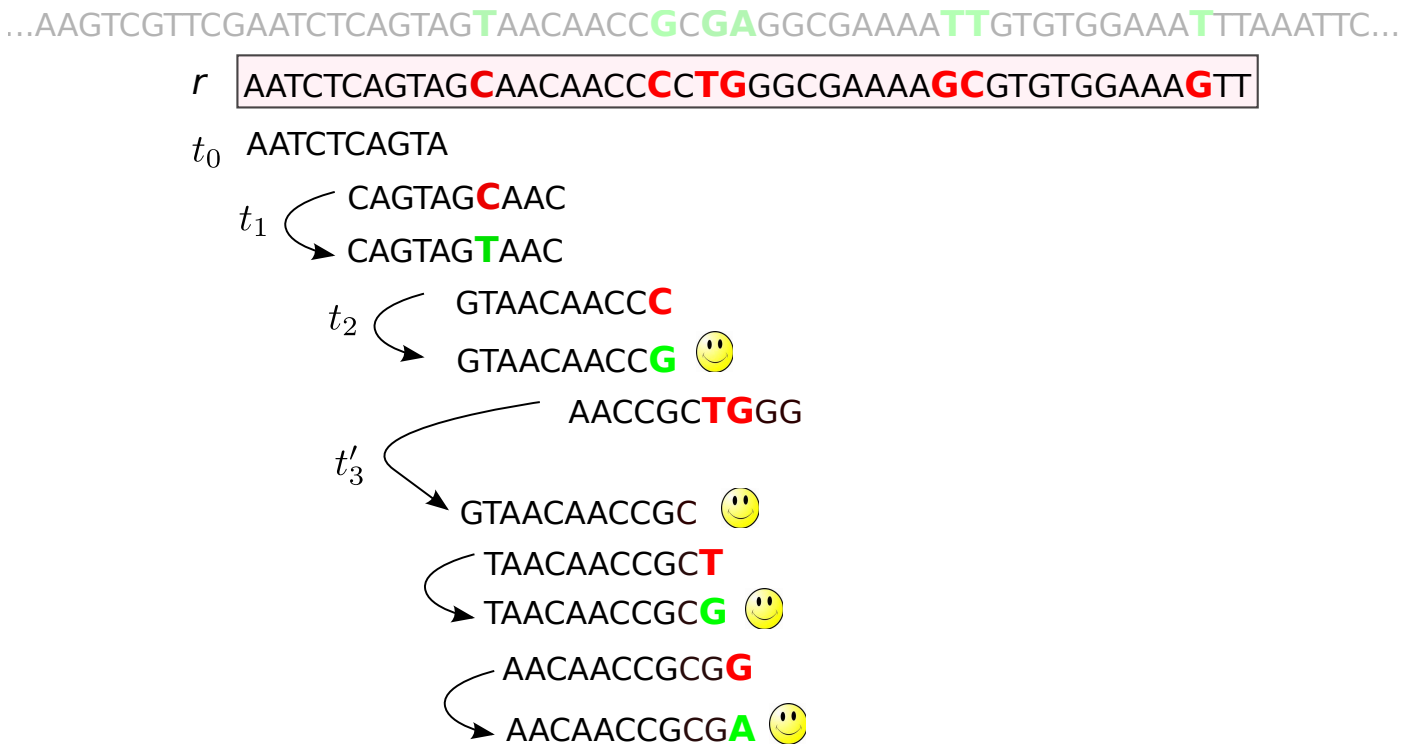
Idea 3 – Read decomposition

tile: 10mer, 2 x 5mers; $d = 2$, 1 in each 5mer



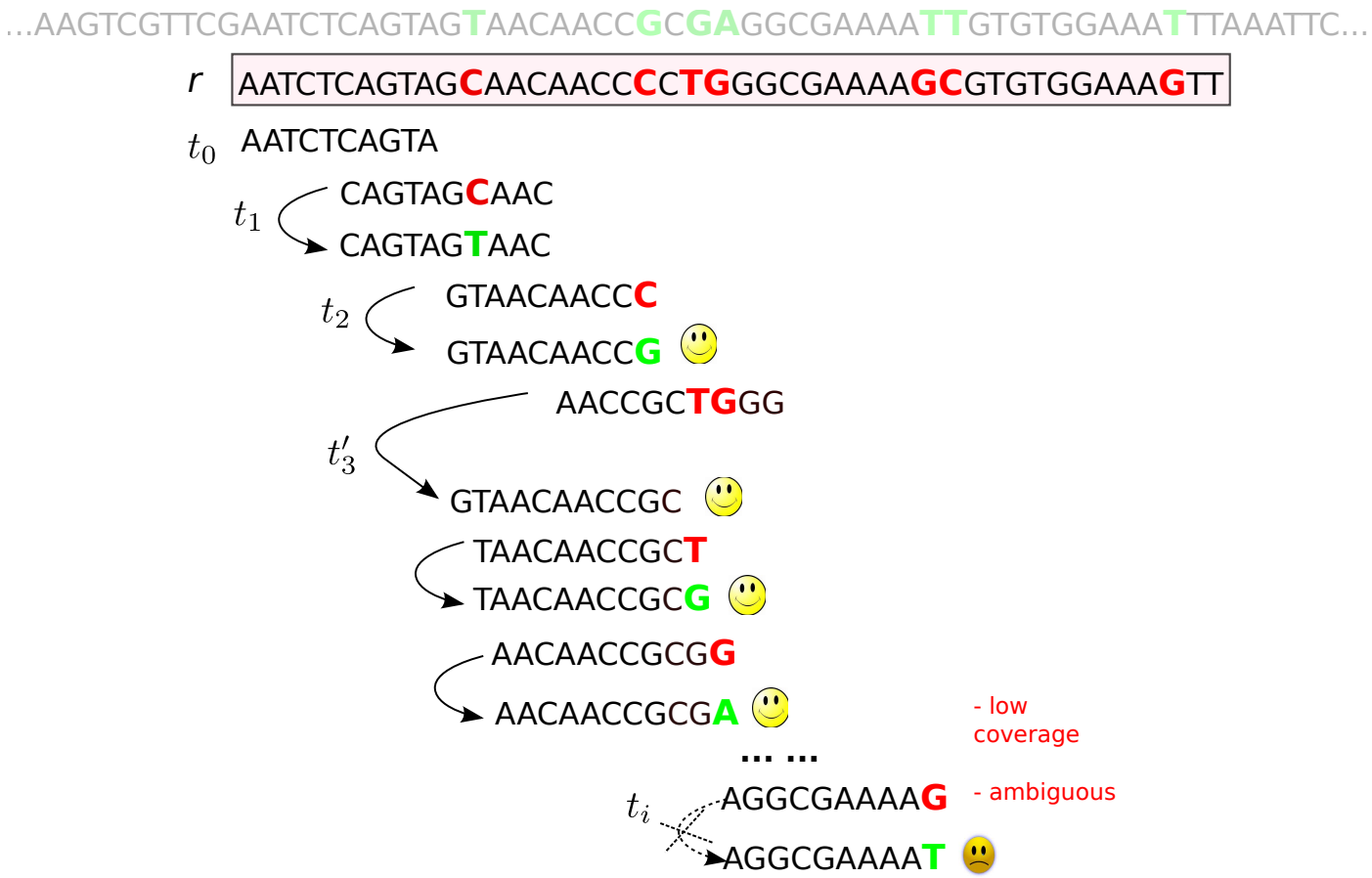
Idea 3 – Read decomposition

tile: 10mer, 2 x 5mers; $d = 2$, 1 in each 5mer



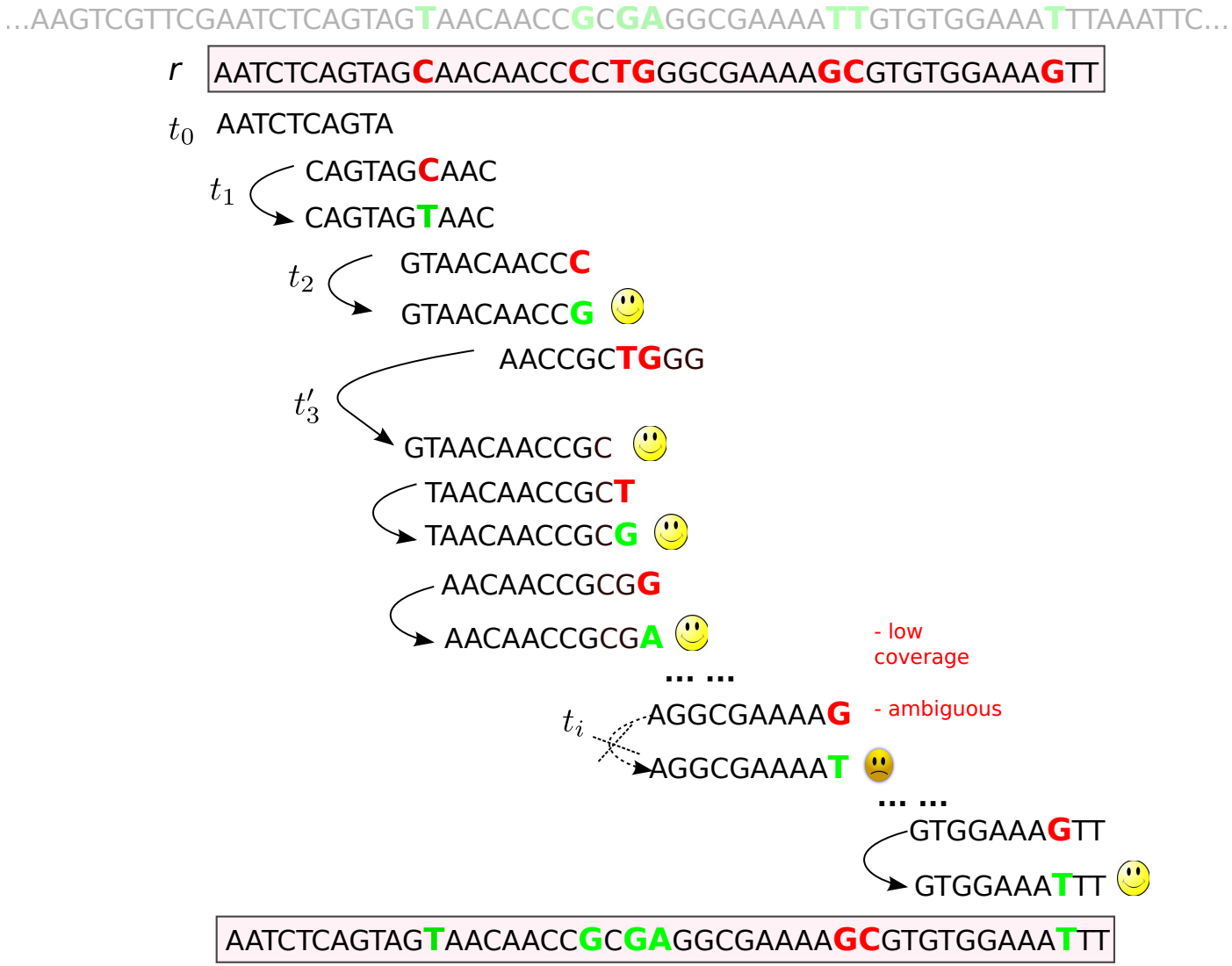
Idea 3 – Read decomposition

tile: 10mer, 2 x 5mers; $d = 2$, 1 in each 5mer



Idea 3 – Read decomposition

tile: 10mer, 2 x 5mers; $d = 2$, 1 in each 5mer



Validation

- *True Positive* (TP): an erroneous nucleotide (nt) is changed to the true nt;
- *False Positive* (FP): a true nt is changed to an erroneous nt;
- *True Negative* (TN): a true nt is unchanged;
- *Flase Negative* (FN): a erroneous nt is unchanged;
- *Sensitivity*: $\frac{TP}{TP+FN}$; *Specificity*: $\frac{TN}{TN+FP}$.

We propose

- *Gain*: $\frac{TP-FP}{TP+FN}$ – Percentage of errors removed
- *Erroneous Base Assignment* (EBA): $EBA = \frac{N_e}{TP+N_e}$
  N_e: correctly identified, wrongly changed

Datasets

Data	Genome	Read Length	Number of Reads	Discarded Reads	Cov.	Error rate
$D1$	*E. coli*	36bp	20.8M	107.7K	160x	0.6%
$D2$	*E. coli*	36bp	10.4M	48.3K	80x	0.6%
$D3$	*A. sp.*	36bp	17.7M	456K	173x	1.5%
$D4$	*A. sp.*	36bp	4.0M	0	40x	1.5%
$D5$	*E. coli*	47bp	7.0M	32.7K	71x	3.3%
$D6$	*E. coli*	101bp	8.9M	1.44M	193x	2.2%

Quality Comparison with SHREC (Schröder *et al.*, 2009) 2010 version

Data (Cov)	Method	EBA (%)	Sensitivity	Specificity	Gain	CPU Hrs	Memory (GB)
$D1$	SHREC	2.27	25.6%	99.7%	−24.1%	31.8	7.7
(160x)	Reptile	**0.028**	86.4%	99.9%	**80.2%**	2.49	1.1
$D2$	SHREC	1.094	72.4%	99.9%	60.6%	9.5	5.1
(79.5x)	Reptile	**0.042**	76.2%	99.9%	**70.9%**	1.23	0.84
$D3$	SHREC	-	-	-	-	-	-
(172.5x)	Reptile	**0.013**	75.1%	99.8%	63.2%	1.66	2.2
$D4$	SHREC	1.063	53.4%	99.7%	29.8%	4.2	4.1
(40x)	Reptile	**0.091**	71%	99.8%	**59.9%**	0.26	0.66
$D5$	SHREC	3.53	21.7%	99.1%	-21.7%	-	> 8
(71x)	Reptile	**0.017**	52.7%	99.7%	**38.1%**	0.94	1.9
$D6$	SHREC	-	-	-	-	-	> 12
(193x)	Reptile	**0.01**	85.3%	99.9%	**78.9%**	2.76	4.6

Parallel Framework for Spectrum Based Approaches

Graph Construction

- Compute k-spectrum.
- Computing Hamming graph and label nodes.

Read Error Correction

Correct each read independently.

- Given a $k$mer, is it valid or invalid?
- Given an invalid $k$mer, find its valid graph neighbors.

$\Rightarrow$ Parallelize graph construction and interface with read error correction.

Performance of Parallel Reptile with $d = 1$

Number of Processors	Dataset D1			Dataset D6		
	k-spectrun Construction Time(s)	Error Correction Time(s)	Total Time(s)	k-spectrum Construction Time(s)	Error Correction Time(s)	Total Time(s)
1	261.93	859.42	1121.35	296.28	4923.35	5219.63
2	133.48	504.52	638.00	161.82	2480.73	2642.55
4	80.44	306.99	387.43	84.04	1349.21	1433.25
8	36.7	149.32	186.02	52.73	698.91	751.64
16	18.18	77.54	95.72	30.62	364.81	395.43
32	10.69	40.76	51.45	13.54	187.6	201.14
64	7.08	21.79	28.87	10.13	96.19	106.32
128	5.19	11.28	16.47	5.20	51.17	56.37
256	5.75	5.76	11.51	6.17	27.92	34.09
512	8.39	2.90	11.29	8.47	14.43	22.9

Larger datasets

Illumina reads from *Drosophila Melanogaster*

Read length	Reads in millions	Coverage
95 bp	37.9	30x
35 bp	41.5	12x
75 bp	18.8	12x

Number of Processors (d)	k-spectrun Construction Time(s)	Error Correction Time(s)	Total Time(s)
128 (1)	34.77	469.99	504.76
256 (1)	24.17	225.76	249.93
512 (1)	53.29	116.06	169.35
512 (2)	54.29	24660.80	24779.20

Open Problems

- Extension to indels (edit distance graph?)
- Hybrid read error correction
- Extract error model from training data and use in error correction
- Error correction $\Leftrightarrow$ Genome assembly

Reptile Availability

- `http://aluru-sun.ece.iastate.edu/doku.php?id=reptile`
- Assembly Pipeline
  `http://code.google.com/p/ngopt/wiki/FastQtoDraftAssembly`