***n*PoRe:** *n*-Polymer Realigner for improved pileup variant calling

Tools and Technology Seminar Series

September 15th, 2022

Tim Dunn

UNIVERSITY OF MICHIGAN

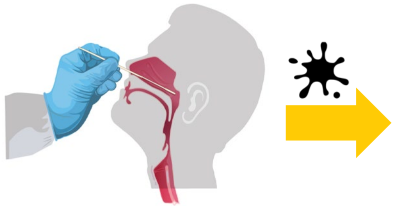
Overview

1. Background
    1. Whole Genome Sequencing
    2. Nanopore Sequencing
    3. Read Alignment
    4. Variant Calling
2. n-Polymer Realigner
    1. Motivation
    2. Algorithm
    3. Results

Overview

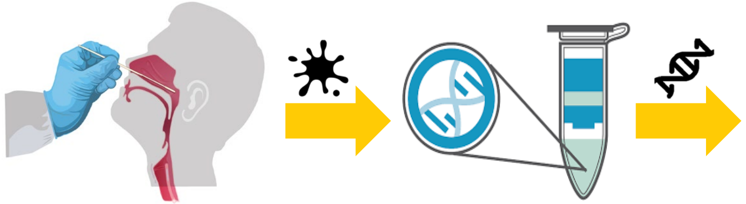
1. Background
    1. Whole Genome Sequencing
    2. Nanopore Sequencing
    3. Read Alignment
    4. Variant Calling
2. n-Polymer Realigner
    1. Motivation
    2. Algorithm
    3. Results

Whole Genome Sequencing: *Overview*



nasal swab

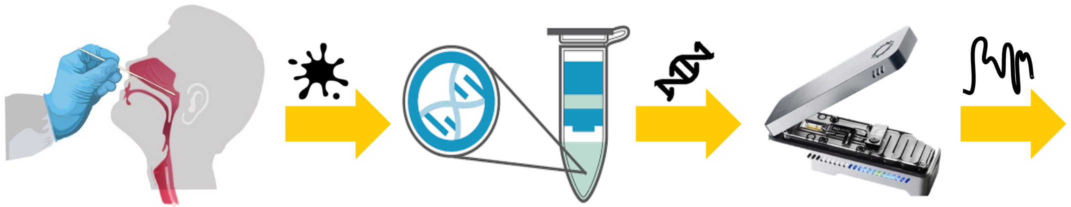
Whole Genome Sequencing: *Overview*



nasal swab

isolate DNA

Whole Genome Sequencing: *Overview*

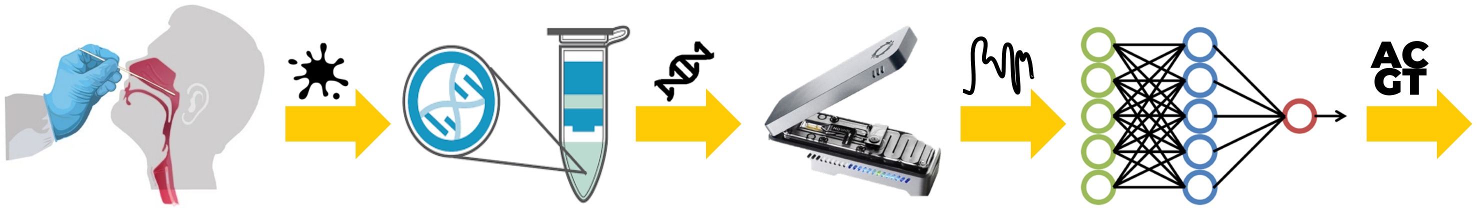


nasal swab

isolate DNA

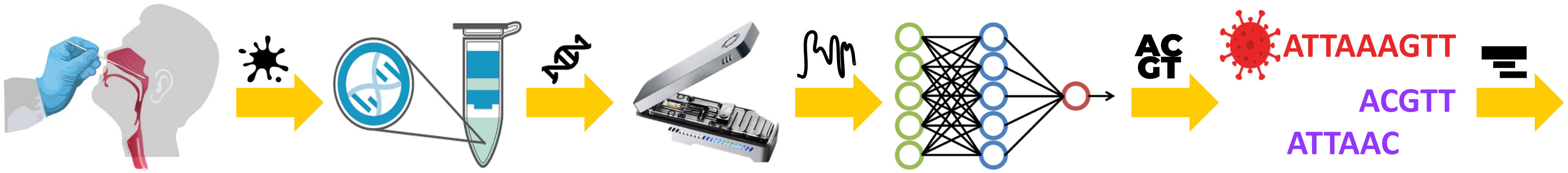
sequence

Whole Genome Sequencing: *Overview*



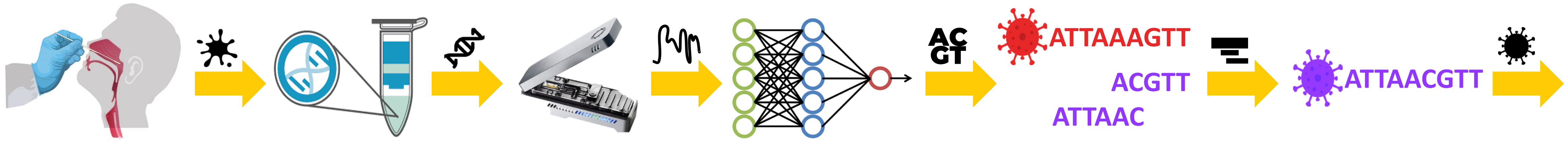
nasal swab → *isolate DNA* → *sequence* → *basecall* →

Whole Genome Sequencing: *Overview*



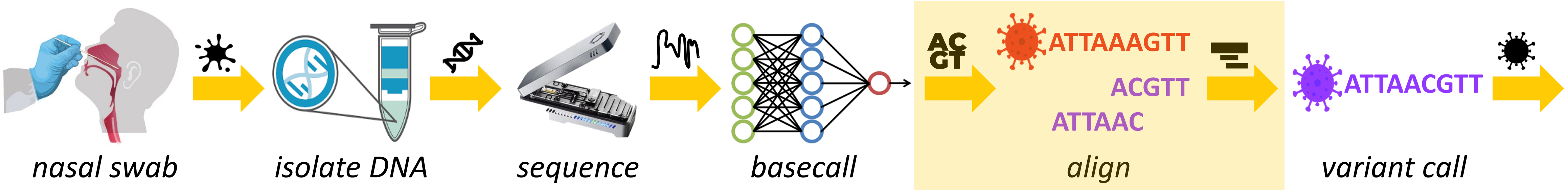
nasal swab → *isolate DNA* → *sequence* → *basecall* → *align* →

Whole Genome Sequencing: *Overview*



nasal swab → *isolate DNA* → *sequence* → *basecall* → *align* → *variant call*

Whole Genome Sequencing: *Overview*



nasal swab

isolate DNA

sequence

basecall

AC GT

ATTAAAGTT

ACGTT

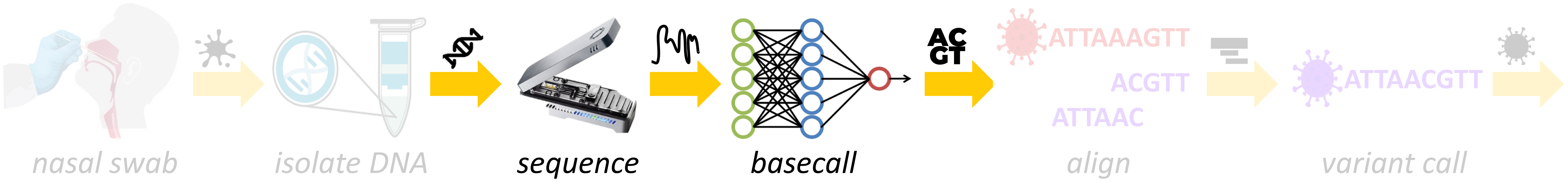
ATTAAC

align

ATTAACGTT

variant call

Whole Genome Sequencing: *Sequencing Tech*



Illumina

100-1,000 bases

99.9% accurate

Accuracy

Cost

illumina

Oxford Nanopore

1,000-1,000,000 bases

90-99.9% accurate

Mapping

Phasing

Oxford NANOPORE Technologies

Pacific Biosciences

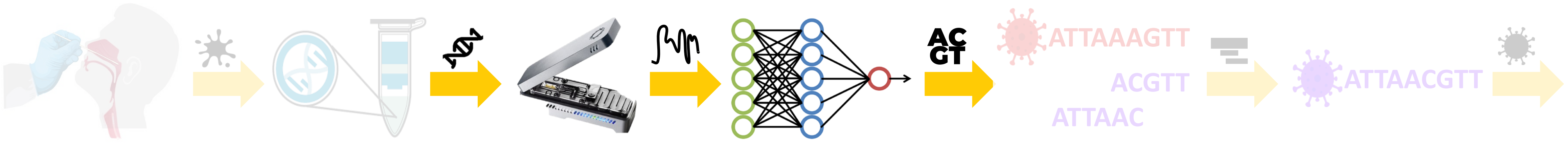
1,000-50,000 bases

99.9% accurate

Accuracy in repetitive regions

PacBio

Whole Genome Sequencing: *Sequencing Tech*



nasal swab → *isolate DNA* → *sequence* → *basecall* → *align* → *variant call*

Illumina

100-1,000 bases

99.9% accurate

Accuracy

Cost

illumina

Oxford Nanopore

1,000-1,000,000 bases

90-99.9% accurate

Mapping

Phasing

Oxford NANOPORE Technologies

Pacific Biosciences

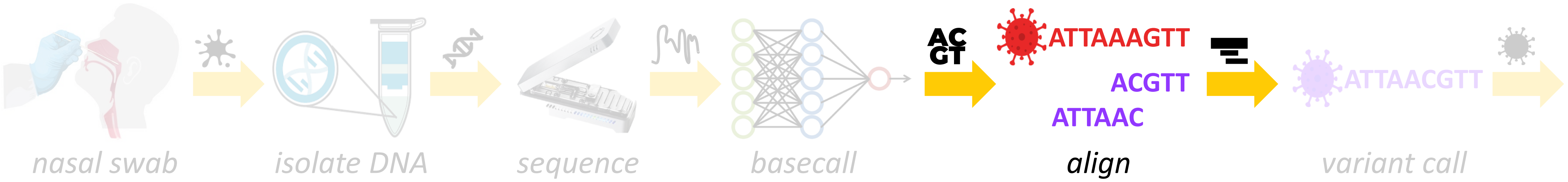
1,000-50,000 bases

99.9% accurate

Accuracy in repetitive regions

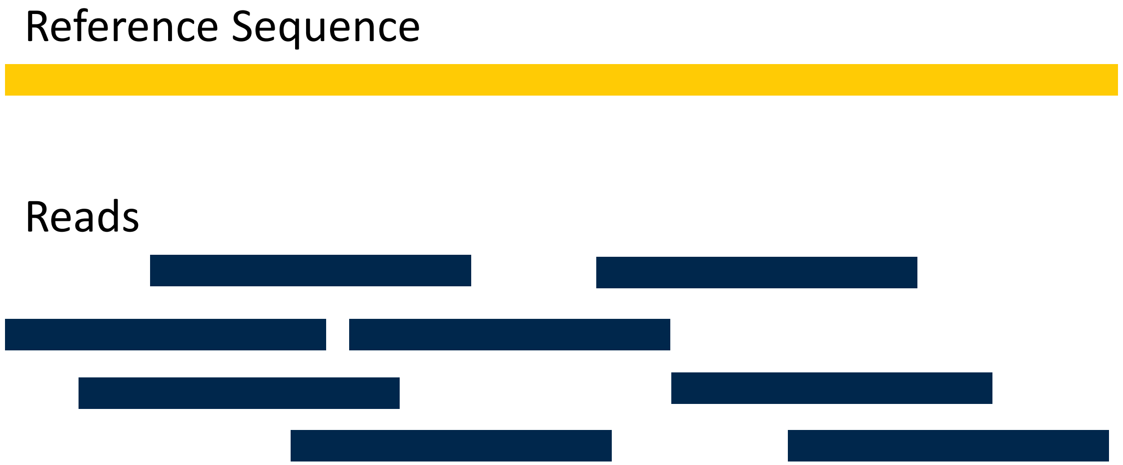
PacBio

Whole Genome Sequencing: *Alignment*



nasal swab *isolate DNA* *sequence* *basecall* *align* *variant call*

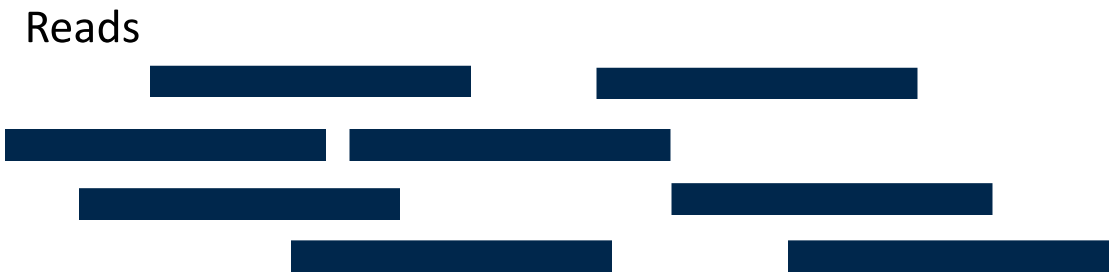
Reference Based



Reference Sequence

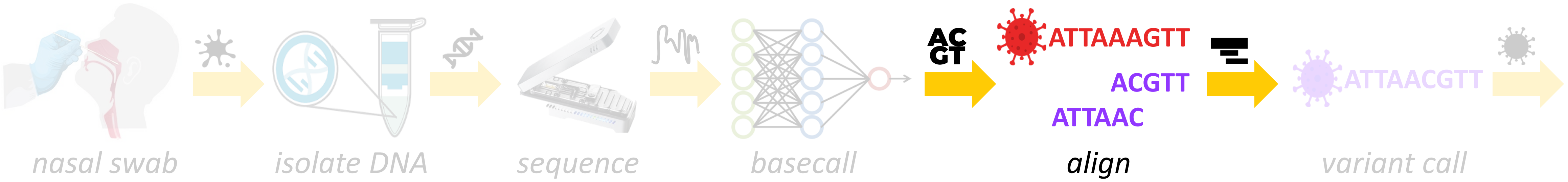
Reads

De Novo



Reads

Whole Genome Sequencing: *Alignment*



nasal swab

isolate DNA

sequence

basecall

AC GT

ATTAAAGTT

ACGTT

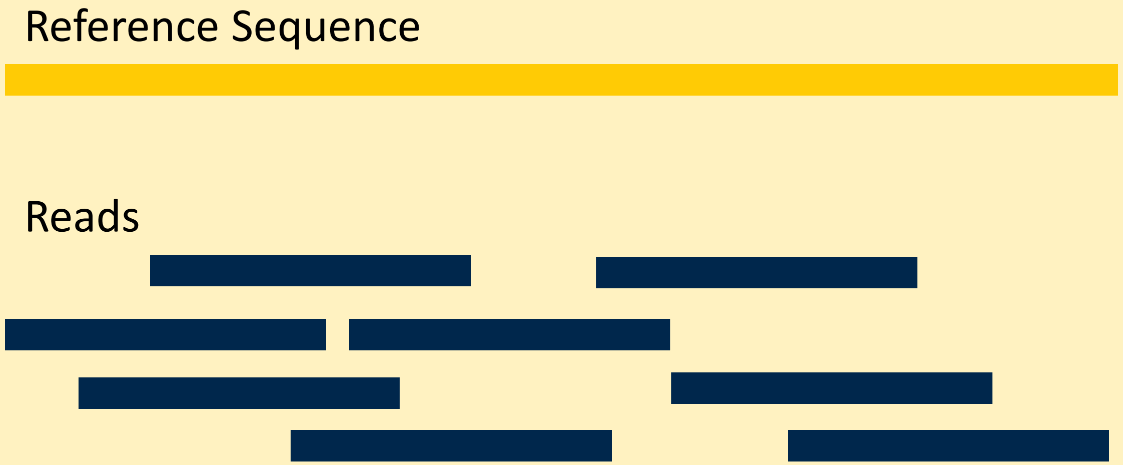
ATTAAC

align

ATTAACGTT

variant call

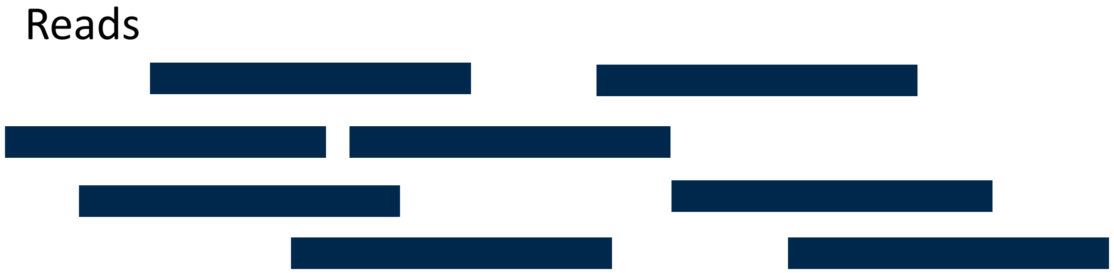
Reference Based



Reference Sequence

Reads

De Novo



Reads

Whole Genome Sequencing: *Applications*

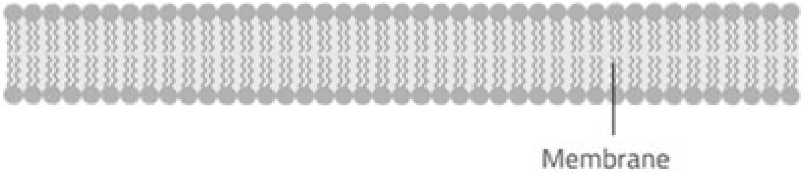
1. Viral/Bacterial Strain Typing
    1. Contact tracing
    2. Antibiotic resistance
2. Patient Diagnoses
    1. Cancer
    2. Genetic Diseases

3. Patient Treatment
    1. Predict drug response

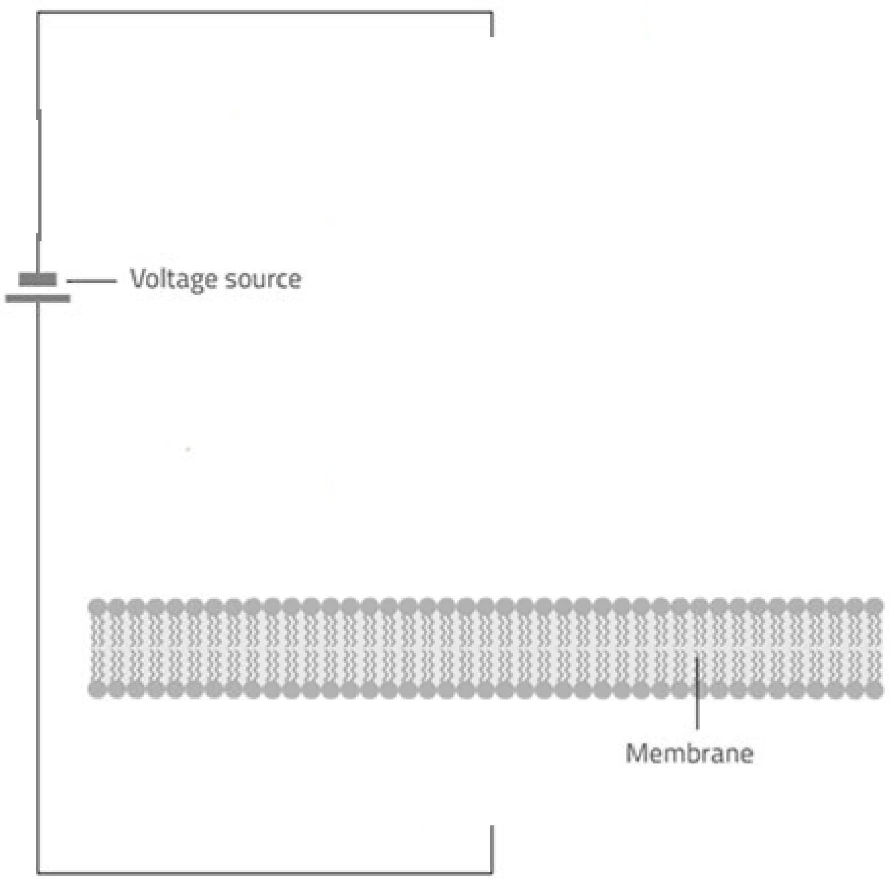
Overview

1. Background
    1. Whole Genome Sequencing
    2. Nanopore Sequencing
    3. Read Alignment
    4. Variant Calling
2. n-Polymer Realigner
    1. Motivation
    2. Algorithm
    3. Results

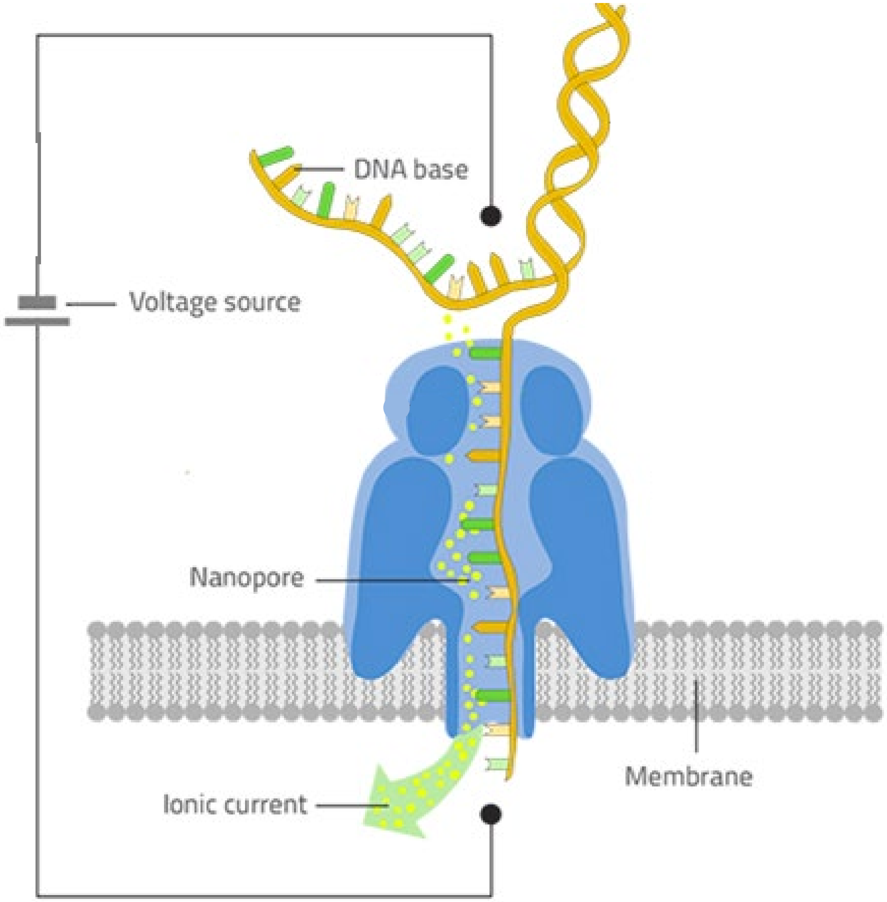
Nanopore Sequencing



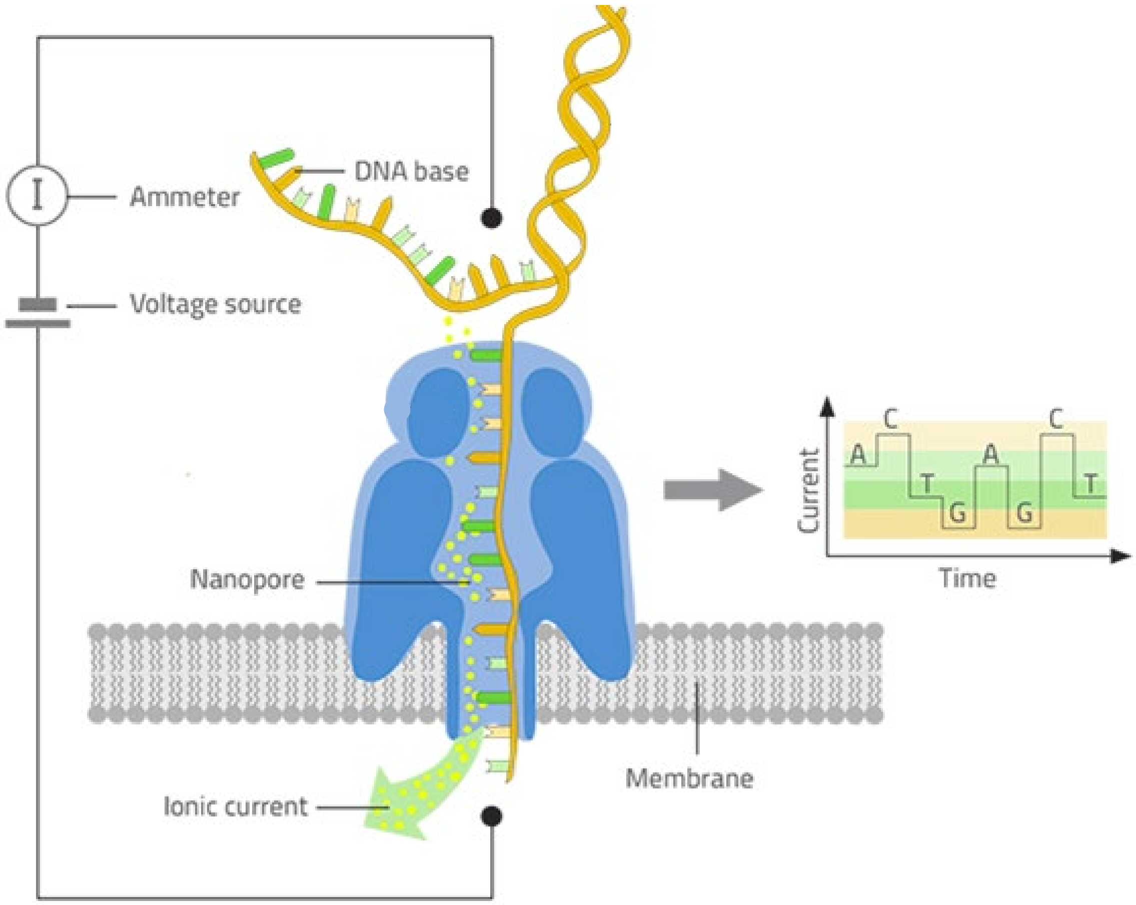
Nanopore Sequencing



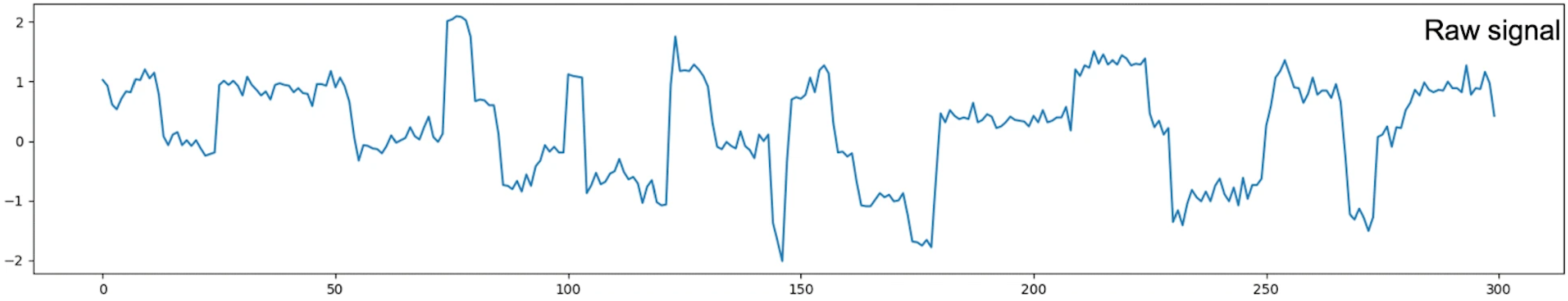
Nanopore Sequencing



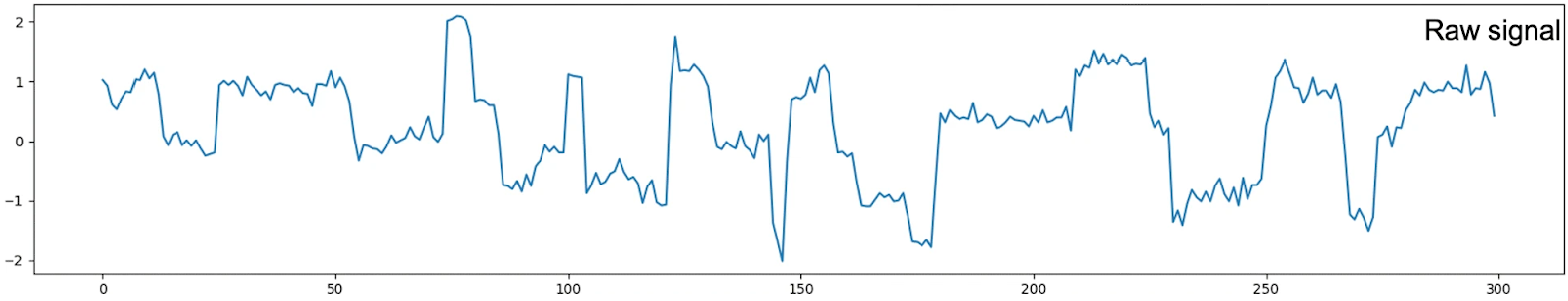
Nanopore Sequencing

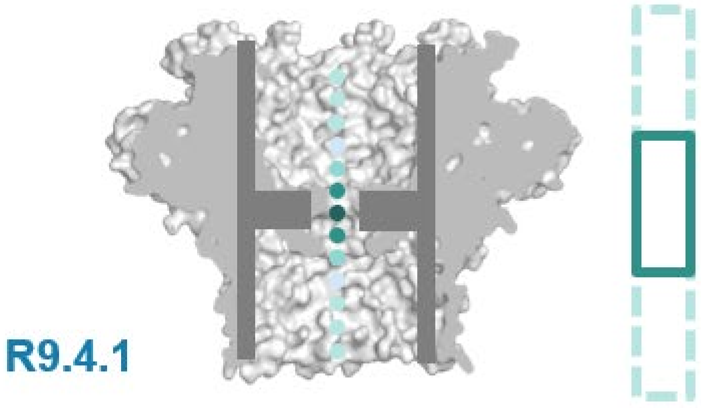


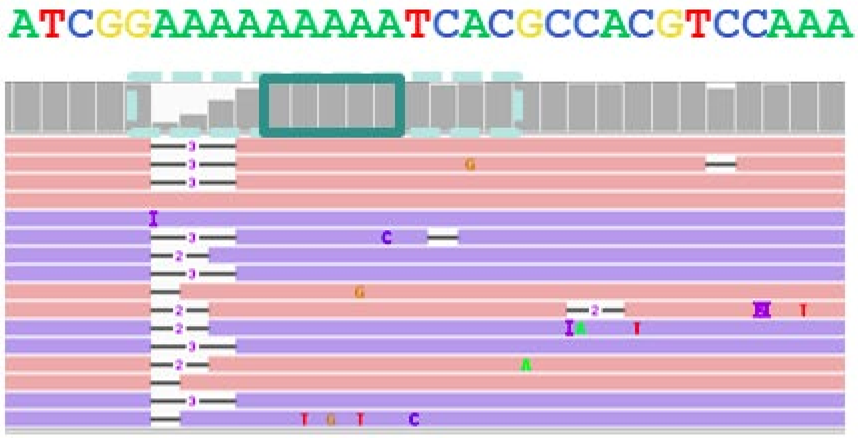
Nanopore Sequencing



Nanopore Sequencing





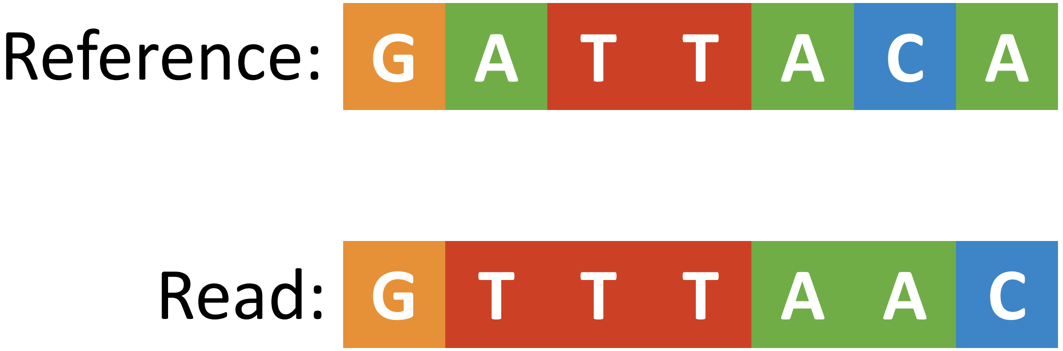


Overview

1. Background
   1. Whole Genome Sequencing
   2. Nanopore Sequencing
   3. Read Alignment
   4. Variant Calling
2. n-Polymer Realigner
   1. Motivation
   2. Algorithm
   3. Results

Alignment: *Edit Distance*

Minimum number of edits required to transform one string to another

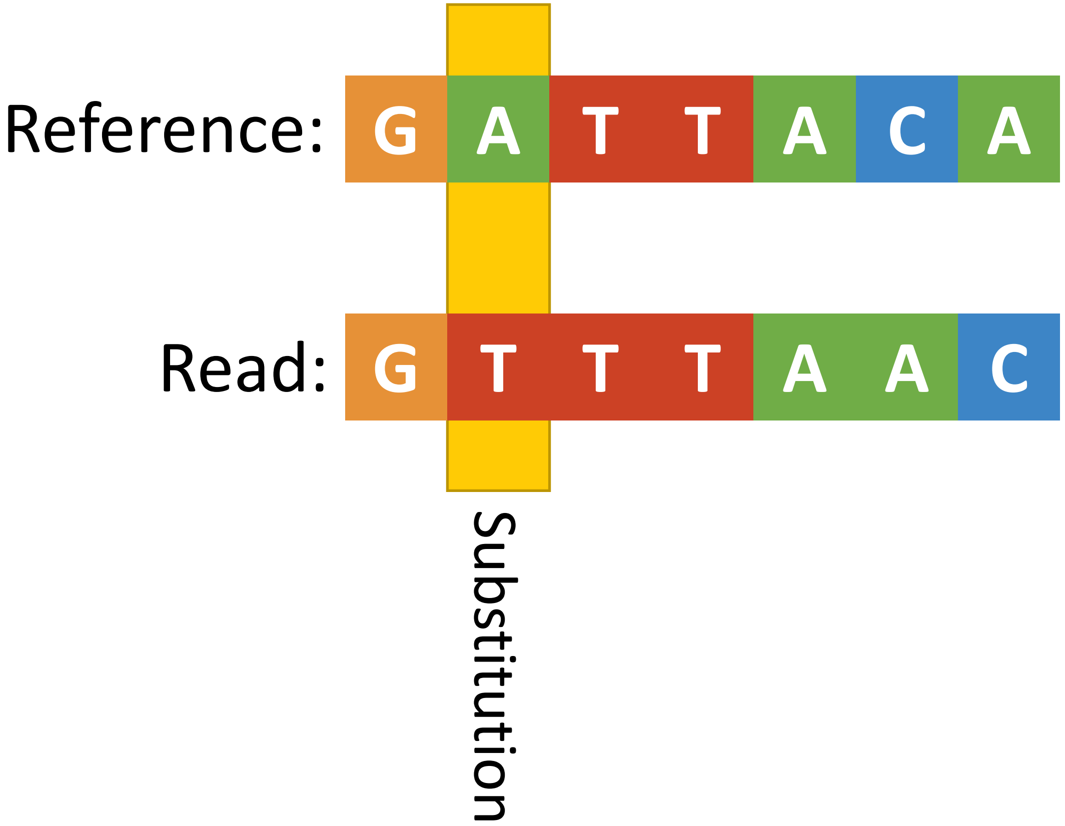


Reference: G A T T A C A

Read: G T T T A A C

Alignment: *Edit Distance*

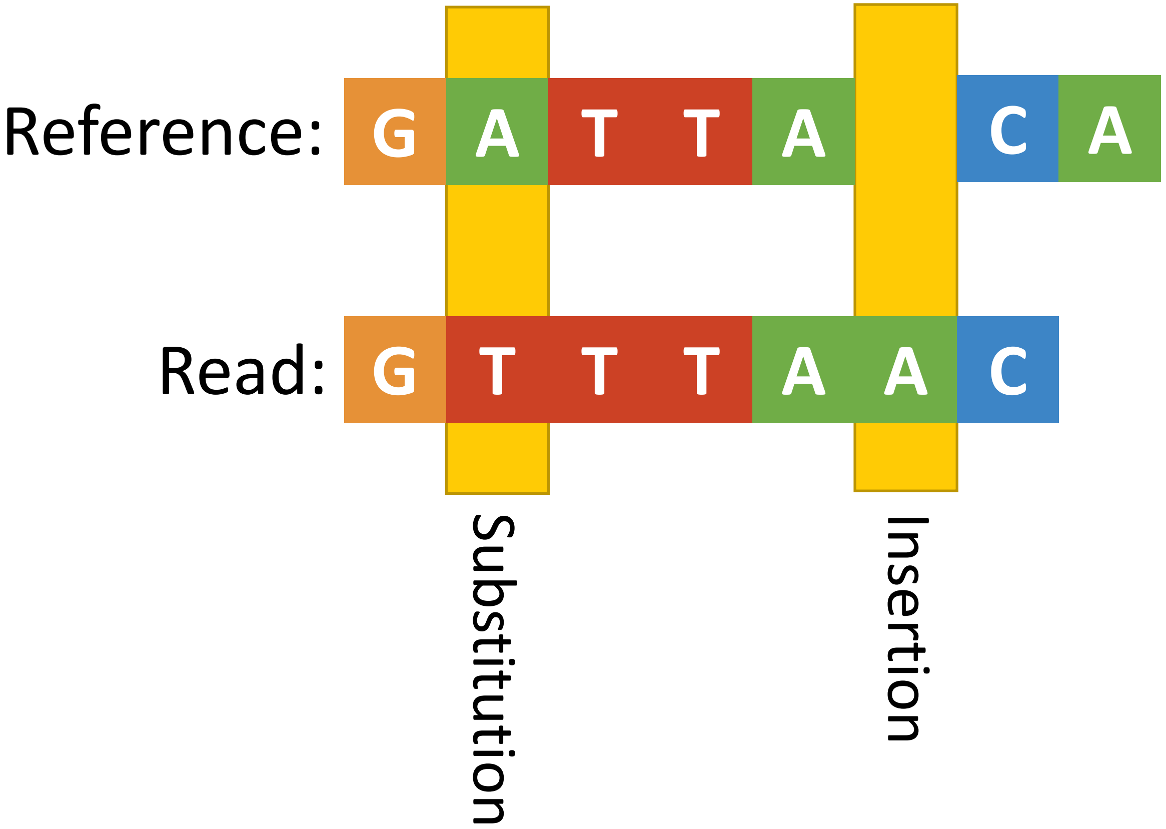
Minimum number of edits required to transform one string to another



Reference:	G	A	T	T	A	C	A
Read:	G	T	T	T	A	A	C

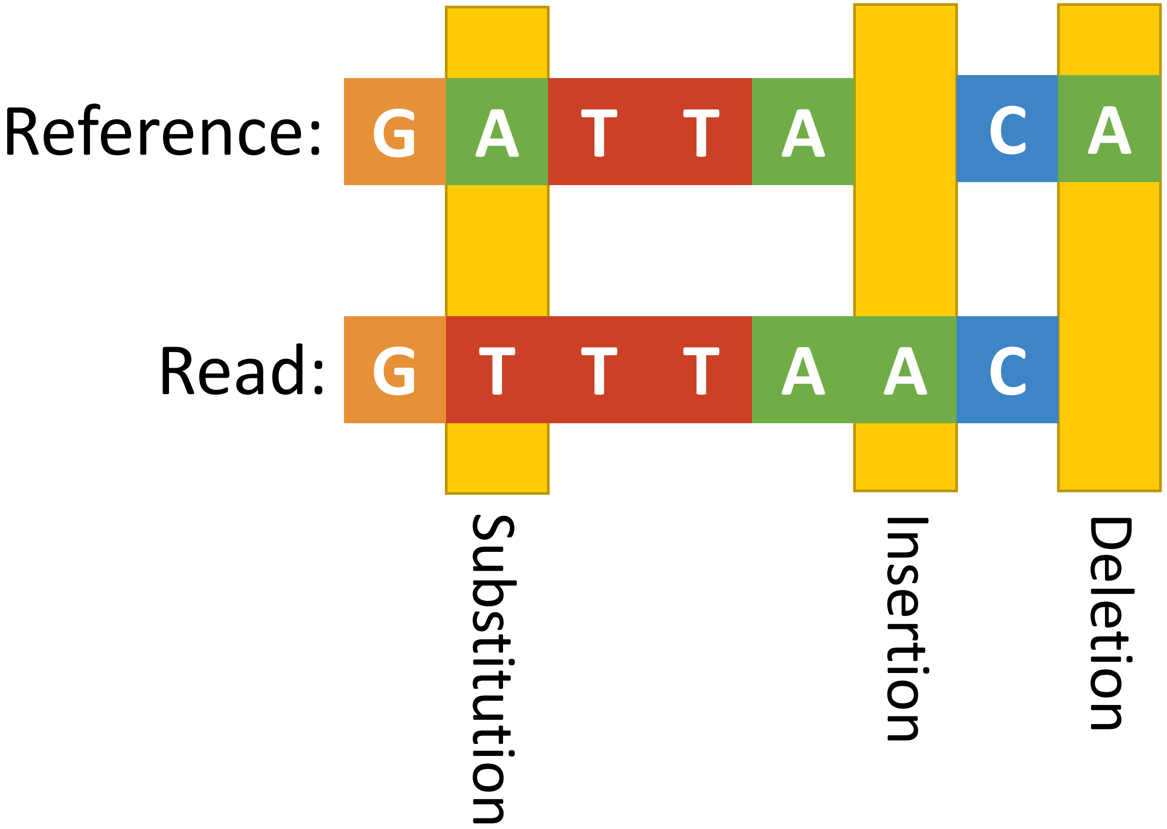
Alignment: *Edit Distance*

Minimum number of edits required to transform one string to another



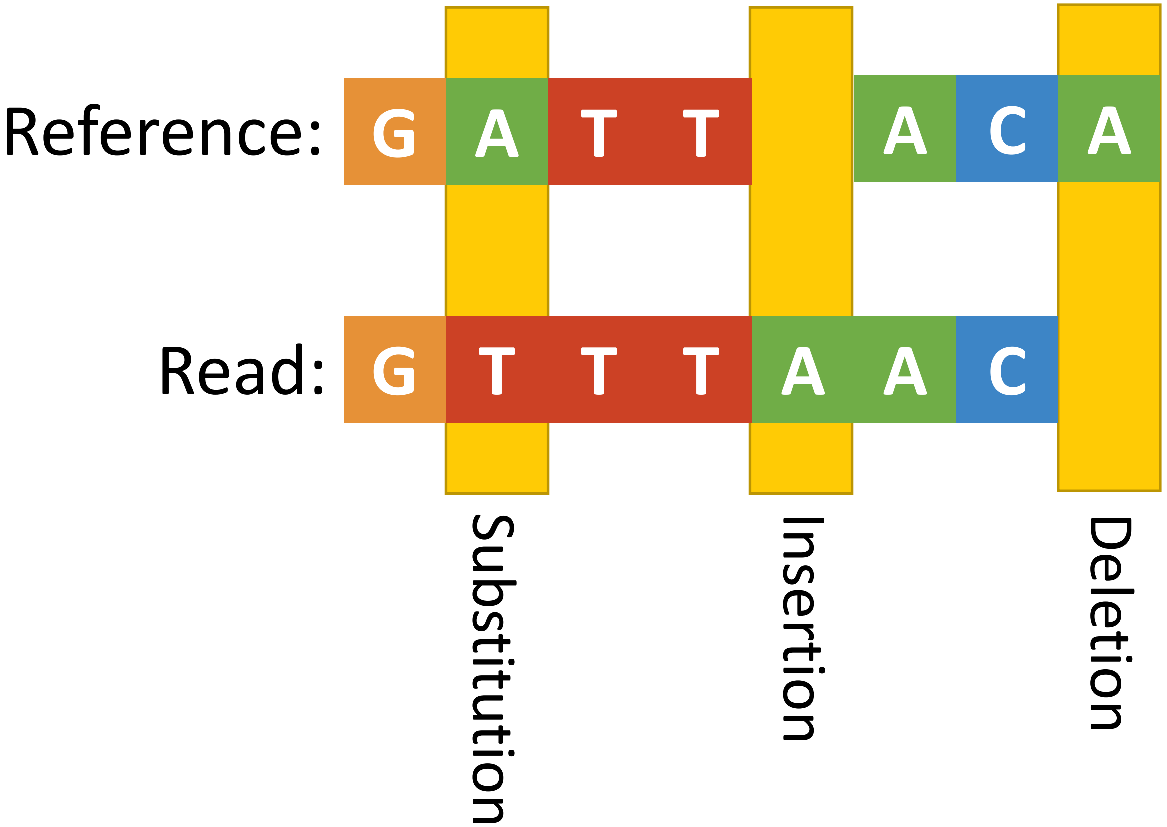
Alignment: *Edit Distance*

Minimum number of edits required to transform one string to another



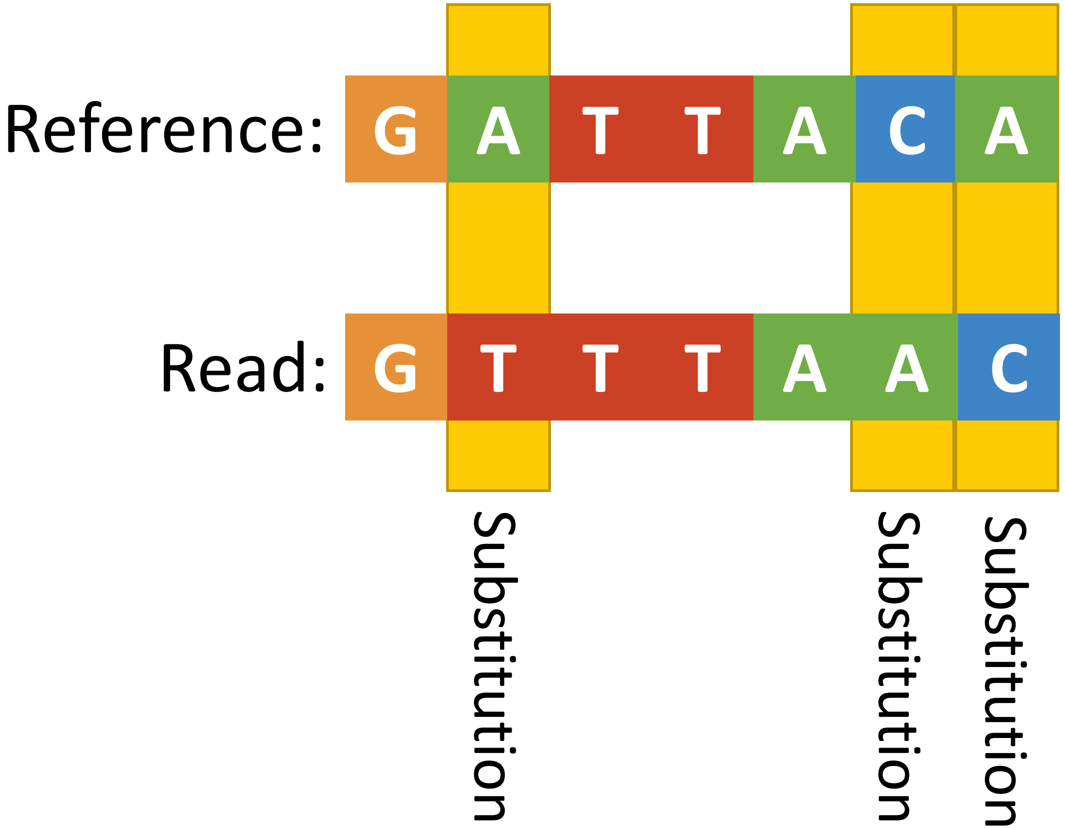
Alignment: *Edit Distance*

Minimum number of edits required to transform one string to another

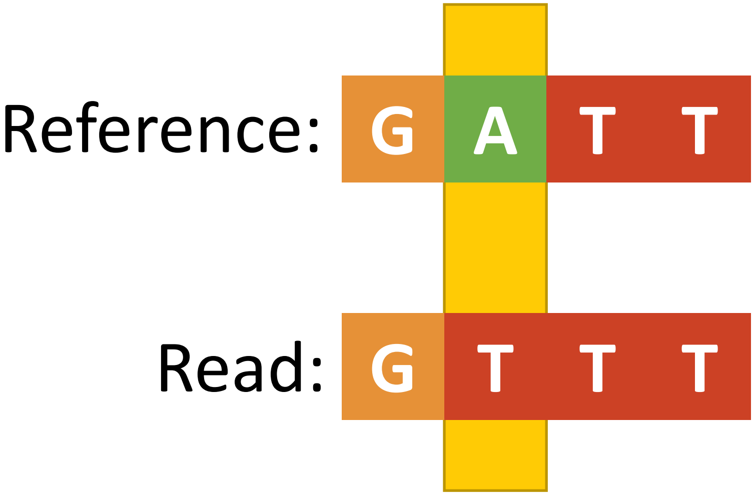


Alignment: *Edit Distance*

Minimum number of edits required to transform one string to another

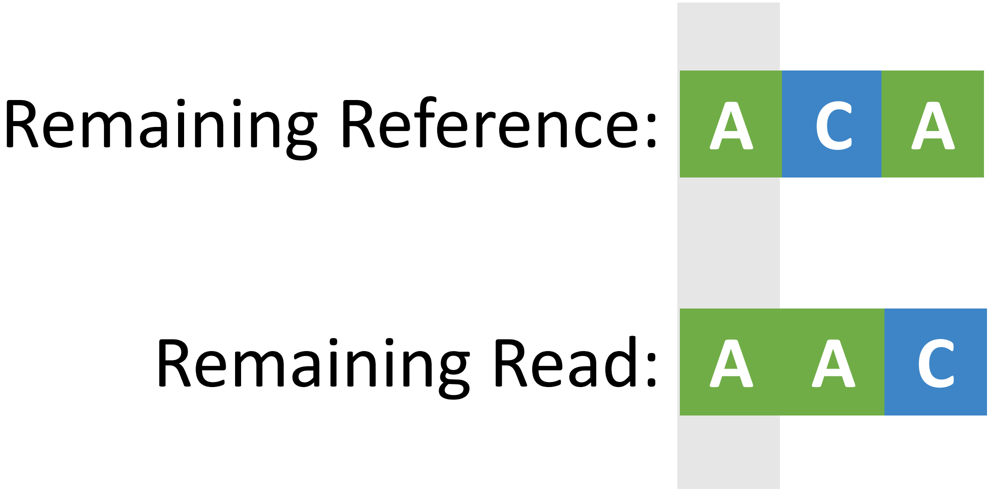


Alignment: *Edit Distance*



Reference: G A T T

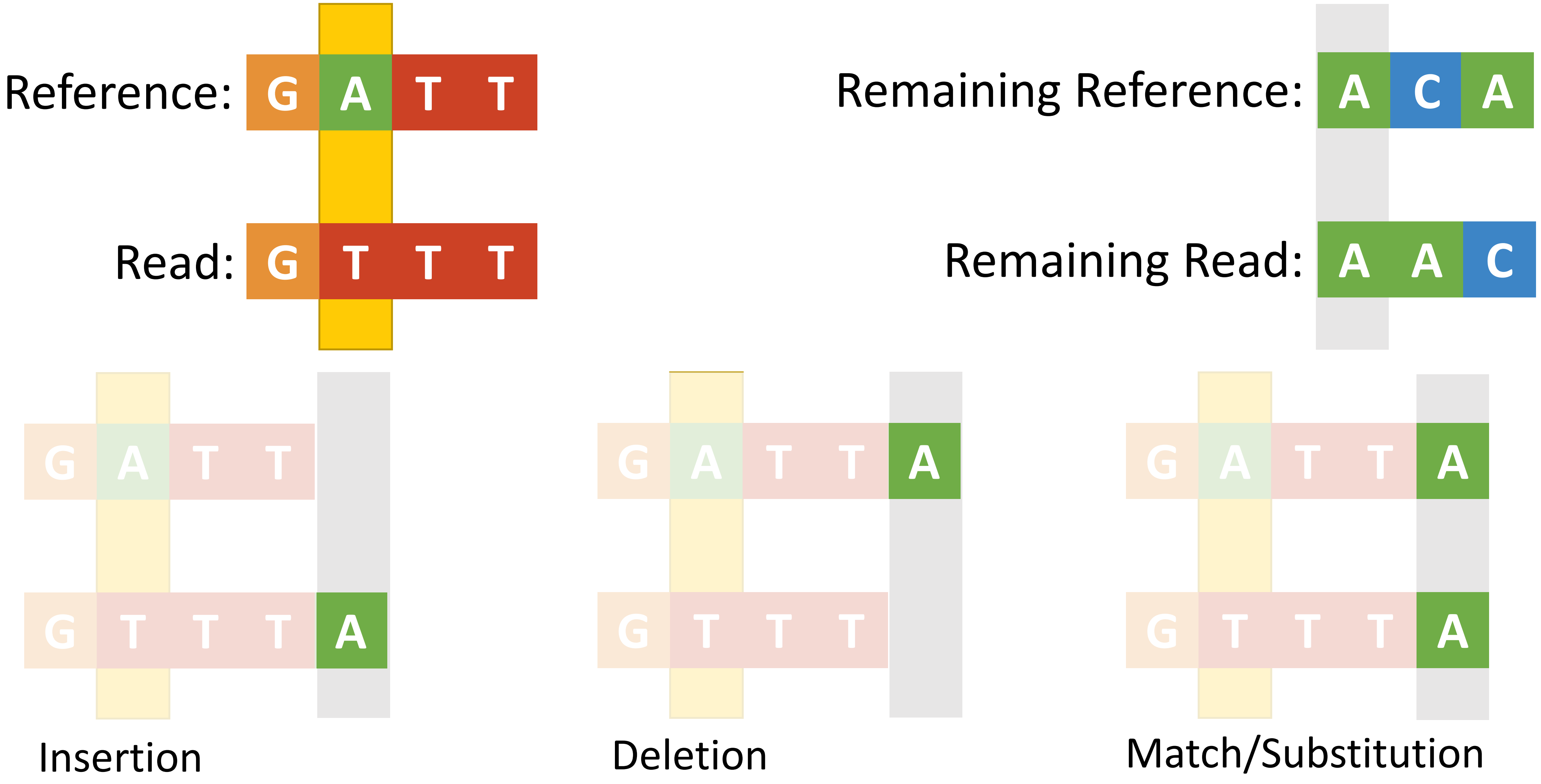
Read: G T T T



Remaining Reference: A C A

Remaining Read: A A C

Alignment: *Edit Distance*



Reference: G A T T

Read: G T T T

Remaining Reference: A C A

Remaining Read: A A C

Insertion

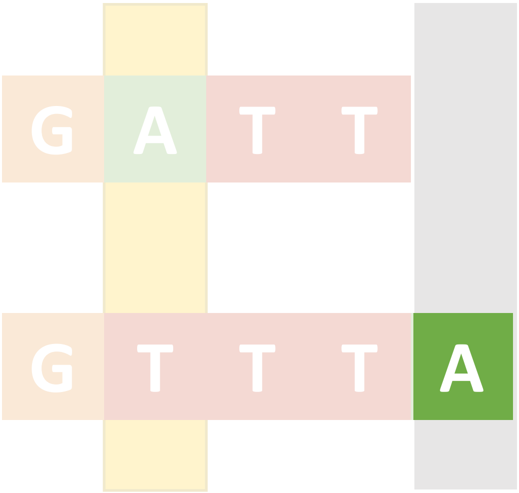
Deletion

Match/Substitution

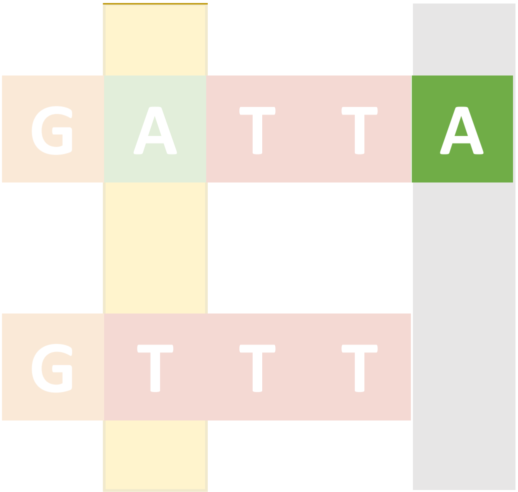
Alignment: *Edit Distance*

$O(3^{|R|})$ possible alignments!

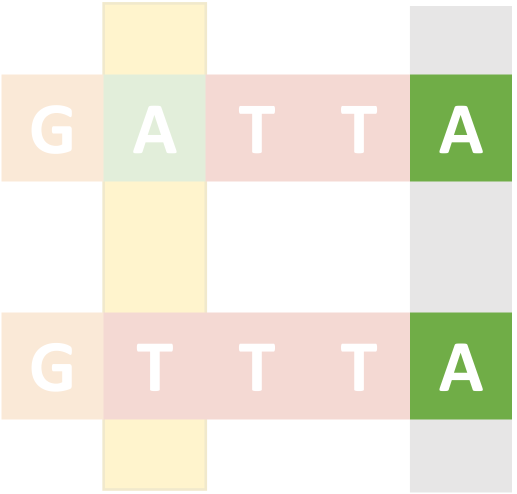
|R| = min(len(Read), len(Ref))



Insertion



Deletion

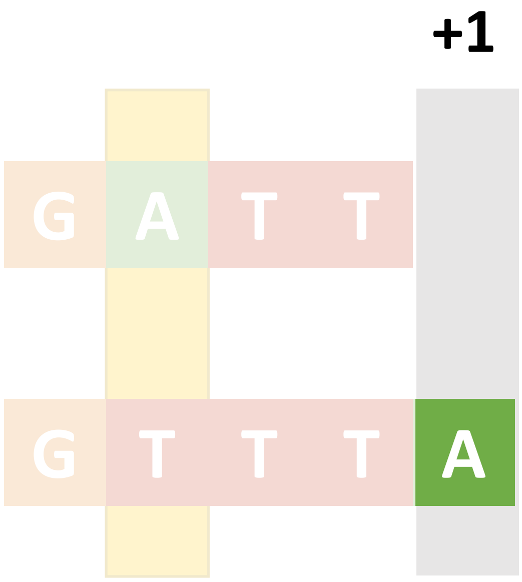


Match/Substitution

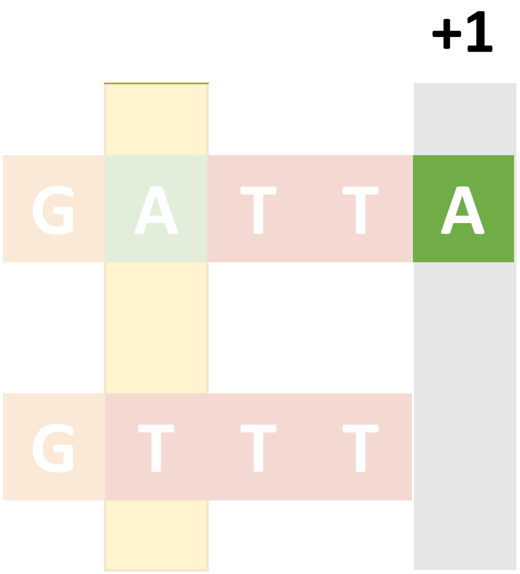
Alignment: *Edit Distance*

Solution: Dynamic Programming

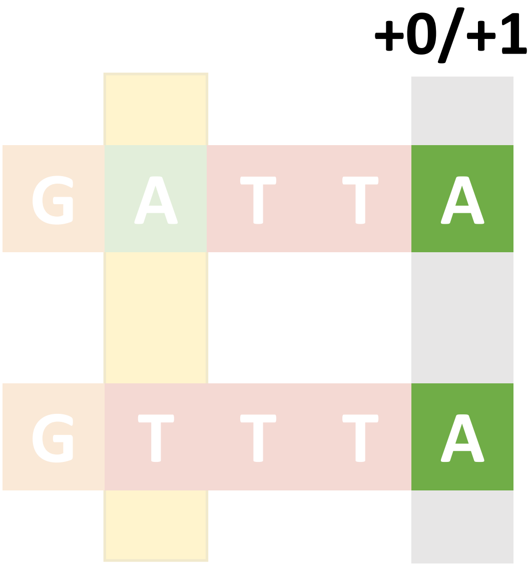
Edit distance is independent of prefix already aligned



Insertion

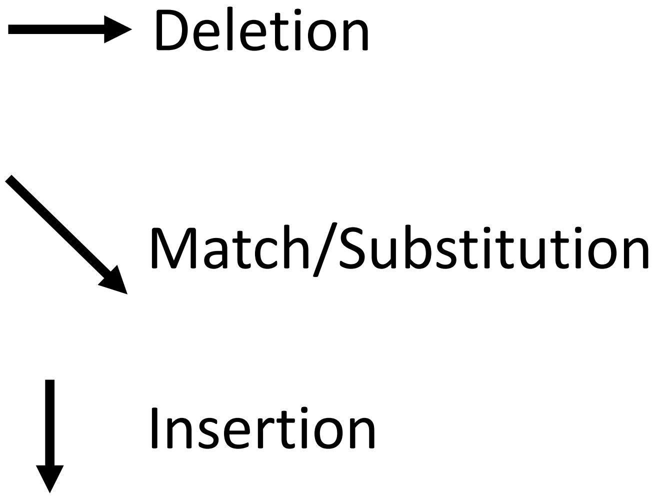


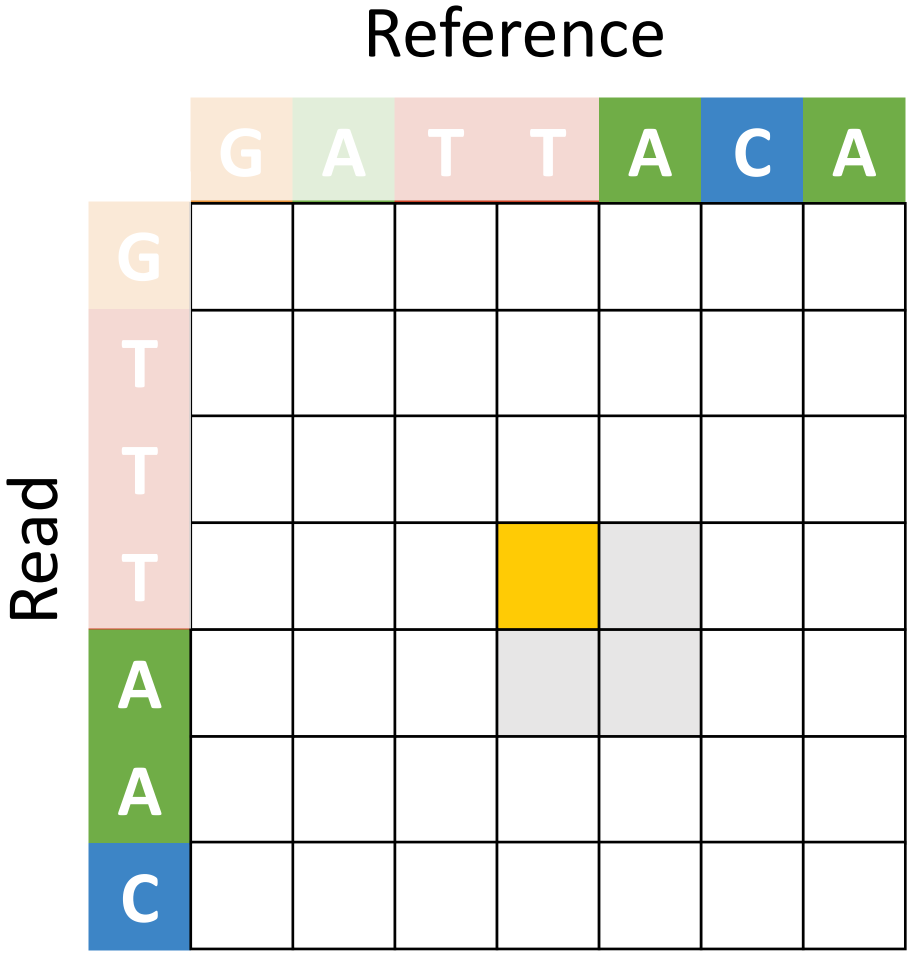
Deletion



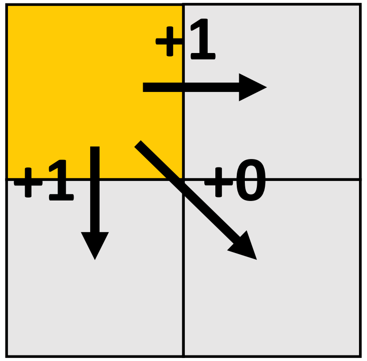
Match/Substitution

Alignment: *Edit Distance*





	G	A	T	T	A	C	A
G							
T							
T							
T							
A							
A							
C							

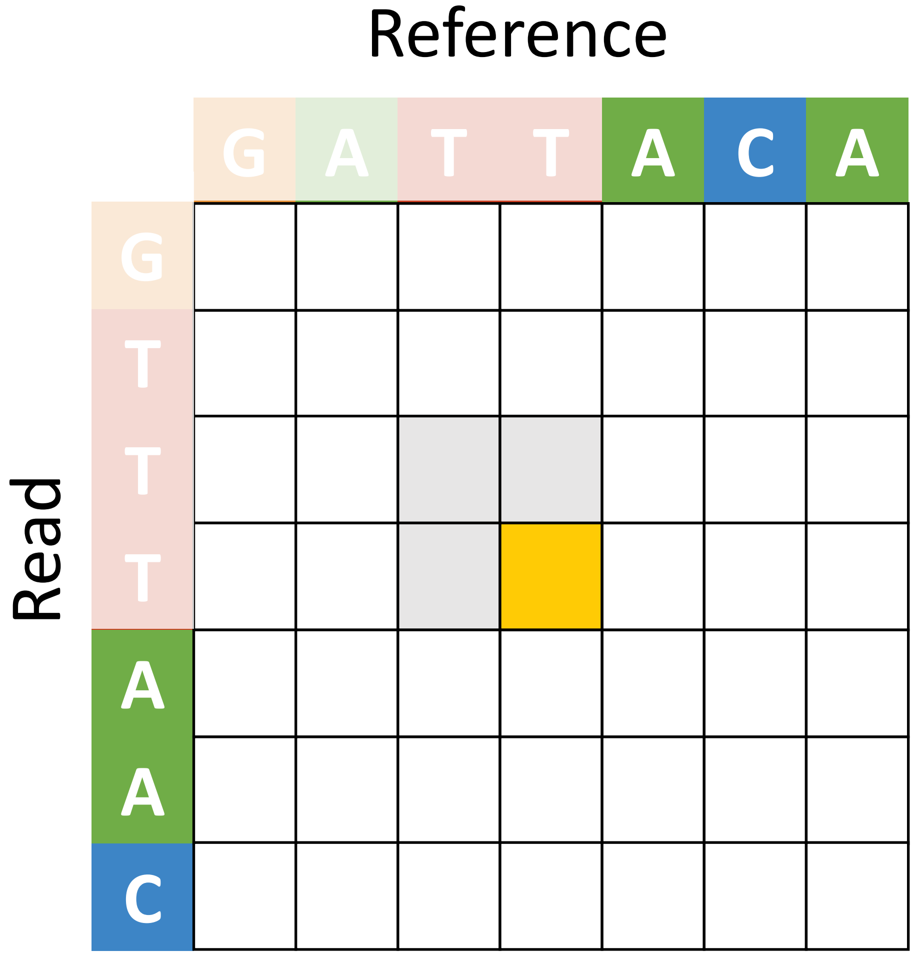


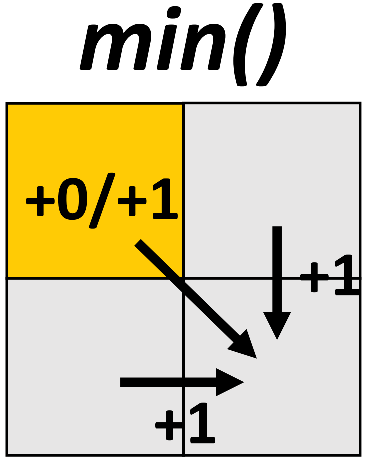
Alignment: *Edit Distance*

→ Deletion

↘ Match/Substitution

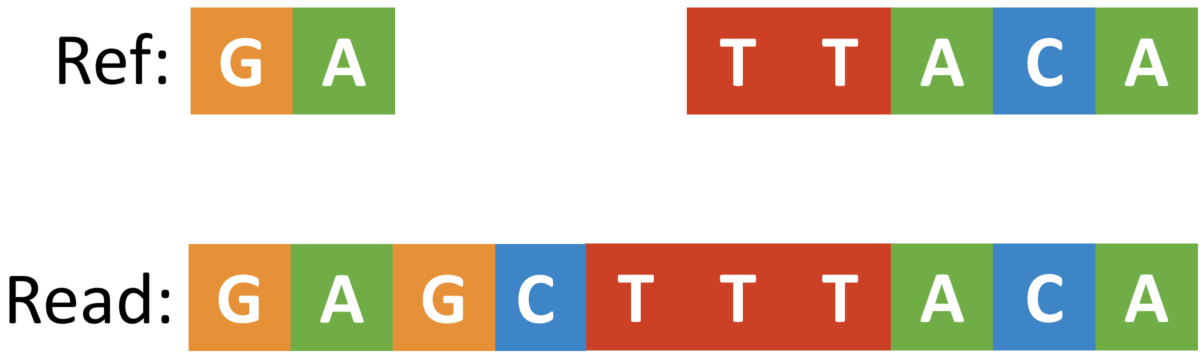
↓ Insertion



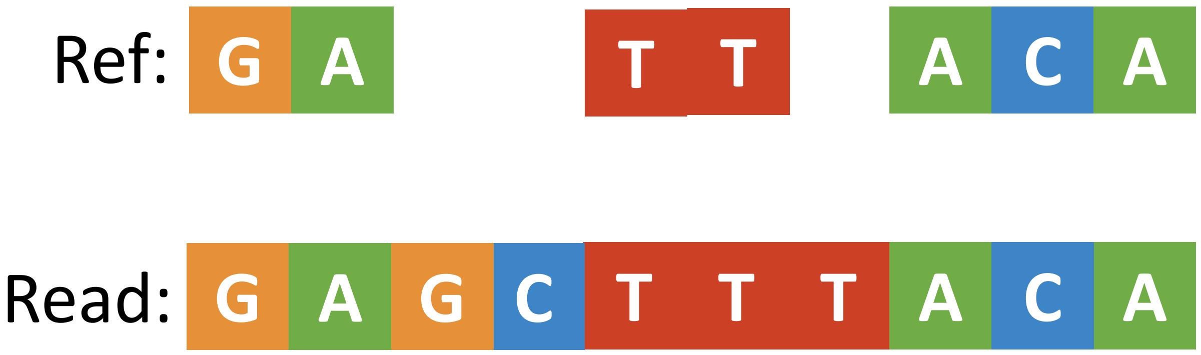


Alignment: *Affine Gap Scoring*

One larger insertion/deletion
is more likely than many small ones



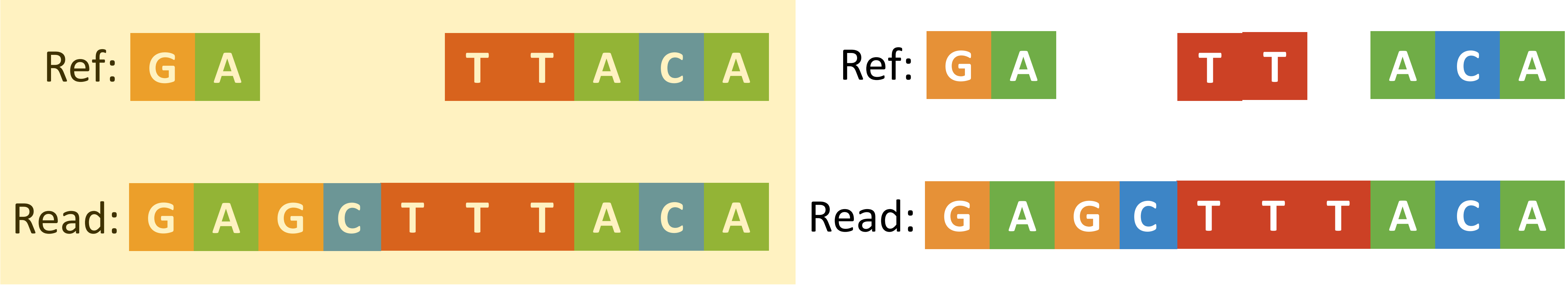
```
Ref:  G A - - - T T A C A
Read: G A G C T T T A C A
```



```
Ref:  G A - - T T - A C A
Read: G A G C T T T A C A
```

Alignment: *Affine Gap Scoring*

One larger insertion/deletion
is more likely than many small ones



Ref:	G	A				T	T	A	C	A
Read:	G	A	G	C	T	T	T	A	C	A

Ref:	G	A			T	T		A	C	A
Read:	G	A	G	C	T	T	T	A	C	A

Alignment: *Affine Gap Scoring*

Normal

	G	A	T	T	A	C	A
G							
T							
T							
T							
A							
A							
C							

Inserting

	G	A	T	T	A	C	A
G							
T							
T							
T							
A							
A							
C							

Deleting

	G	A	T	T	A	C	A
G							
T							
T							
T							
A							
A							
C							

Alignment: *Affine Gap Scoring*

Normal

	G	A	T	T	A	C	A
G							
T							
T							
T				🟨			
A					*		
A							
C							

Inserting

	G	A	T	T	A	C	A
G							
T							
T							
T							
A				5			
A							
C							

Deleting

	G	A	T	T	A	C	A
G							
T							
T							
T					5		
A							
A							
C							

* 0 if match
3 if mismatch

Alignment: *Affine Gap Scoring*

Normal

	G	A	T	T	A	C	A
G							
T							
T							
T				0			
A							
A							
C							

Inserting

	G	A	T	T	A	C	A
G							
T							
T							
T				🟨			
A				2			
A							
C							

Deleting

	G	A	T	T	A	C	A
G							
T							
T							
T							
A							
A							
C							

Alignment: *Affine Gap Scoring*

Normal

	G	A	T	T	A	C	A
G							
T							
T							
T				0			
A							
A							
C							

Inserting

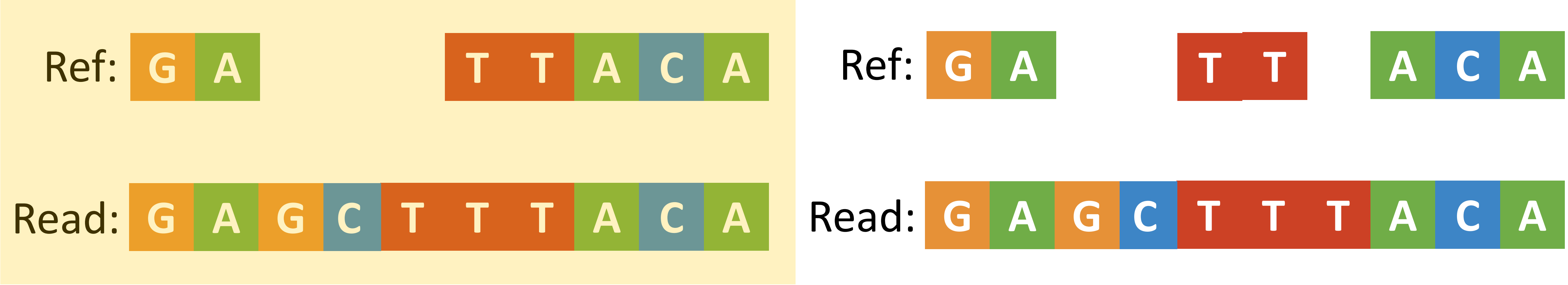
	G	A	T	T	A	C	A
G							
T							
T							
T							
A							
A							
C							

Deleting

	G	A	T	T	A	C	A
G							
T							
T							
T				🟨	2		
A							
A							
C							

Alignment: *Affine Gap Scoring*

One larger insertion/deletion
is more likely than many small ones



```
Ref:  G A       T T A C A        Ref:  G A     T T   A C A

Read: G A G C T T T A C A        Read: G A G C T T T A C A
```

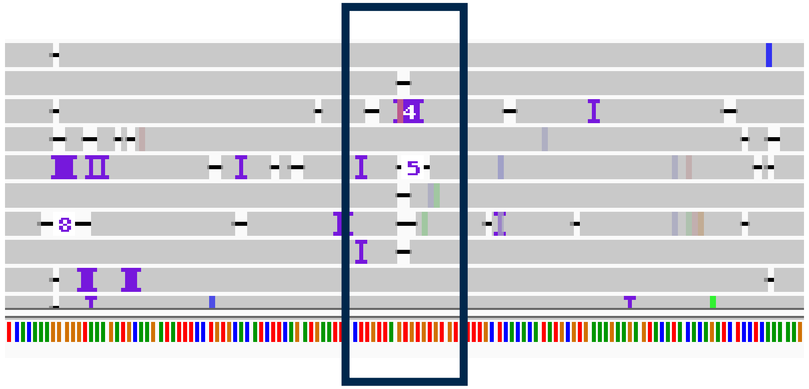
Overview

1. Background
    1. Whole Genome Sequencing
    2. Nanopore Sequencing
    3. Read Alignment
    4. Variant Calling
2. n-Polymer Realigner
    1. Motivation
    2. Algorithm
    3. Results

Variant Calling: *Overview*

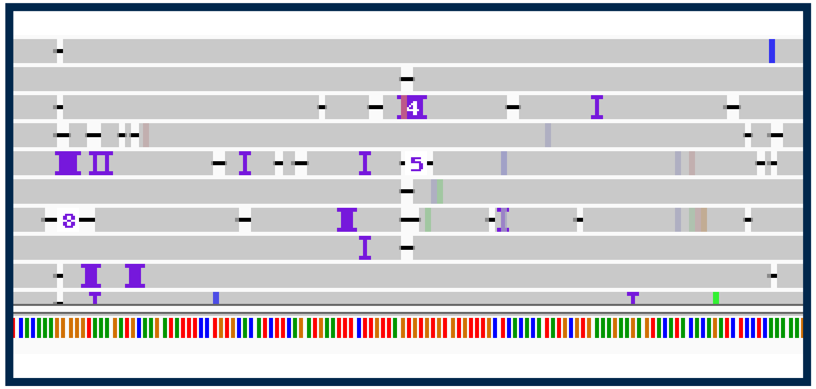
Pileup Calling

Identify Candidate Variants



Full-Alignment Calling

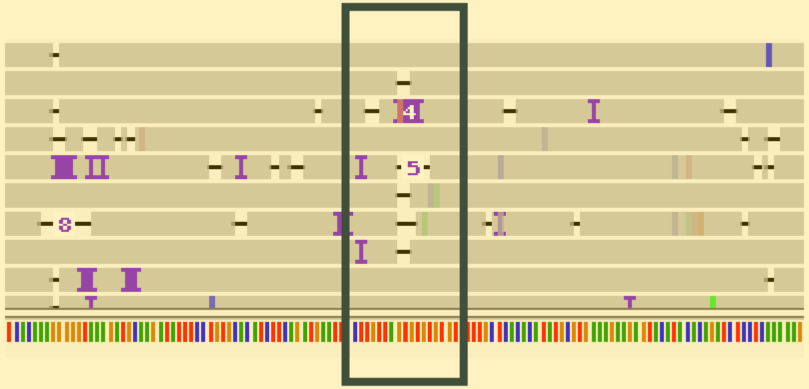
Final Variant Calls



Variant Calling: *Overview*

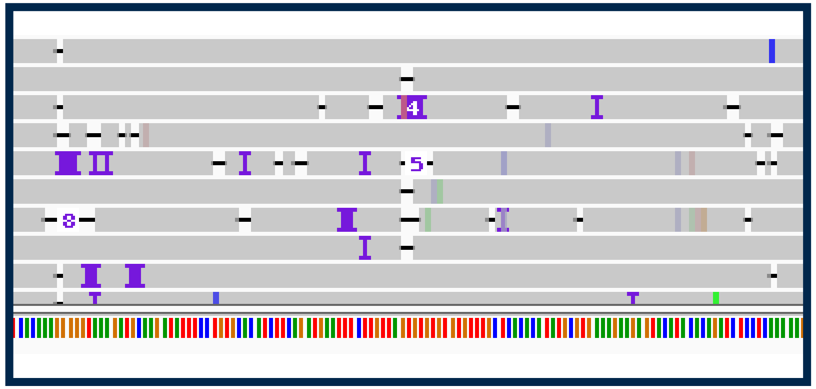
Pileup Calling

Identify Candidate Variants

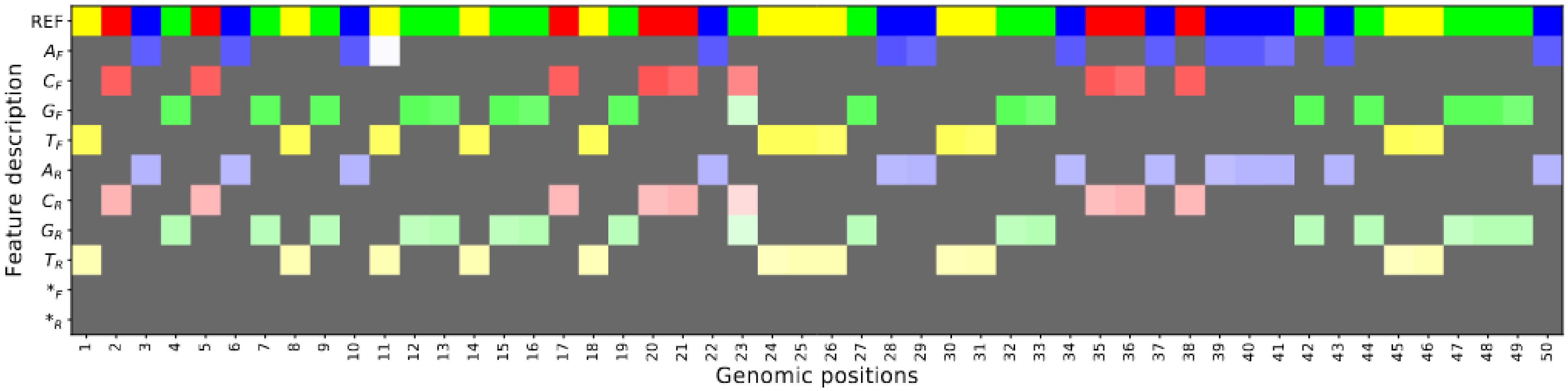


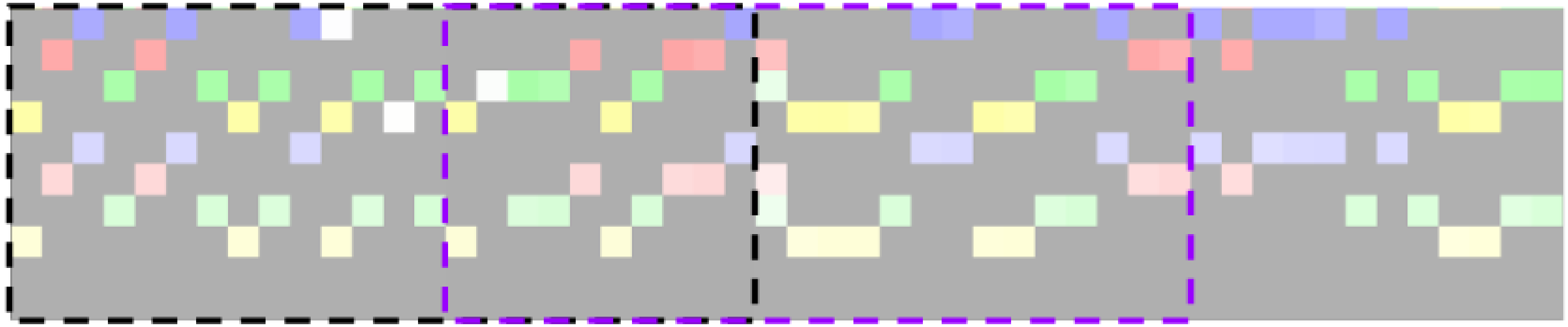
Full-Alignment Calling

Final Variant Calls



Variant Calling: *Pileup Calling*



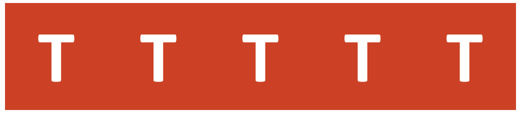


Sliding window over a long sequence

Overview

1. Background
    1. Whole Genome Sequencing
    2. Nanopore Sequencing
    3. Read Alignment
    4. Variant Calling
2. n-Polymer Realigner
    1. Motivation
    2. Algorithm
    3. Results

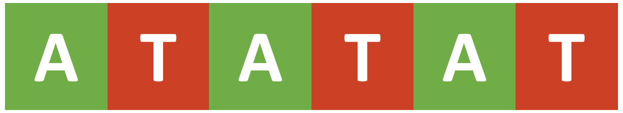
Definition: *n-Polymers are homopolymers and STRs*



T T T T T

1-polymer

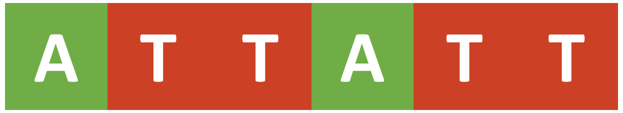
“homopolymer”



A T A T A T

2-polymer

“simple tandem repeat with repeat unit length 2”



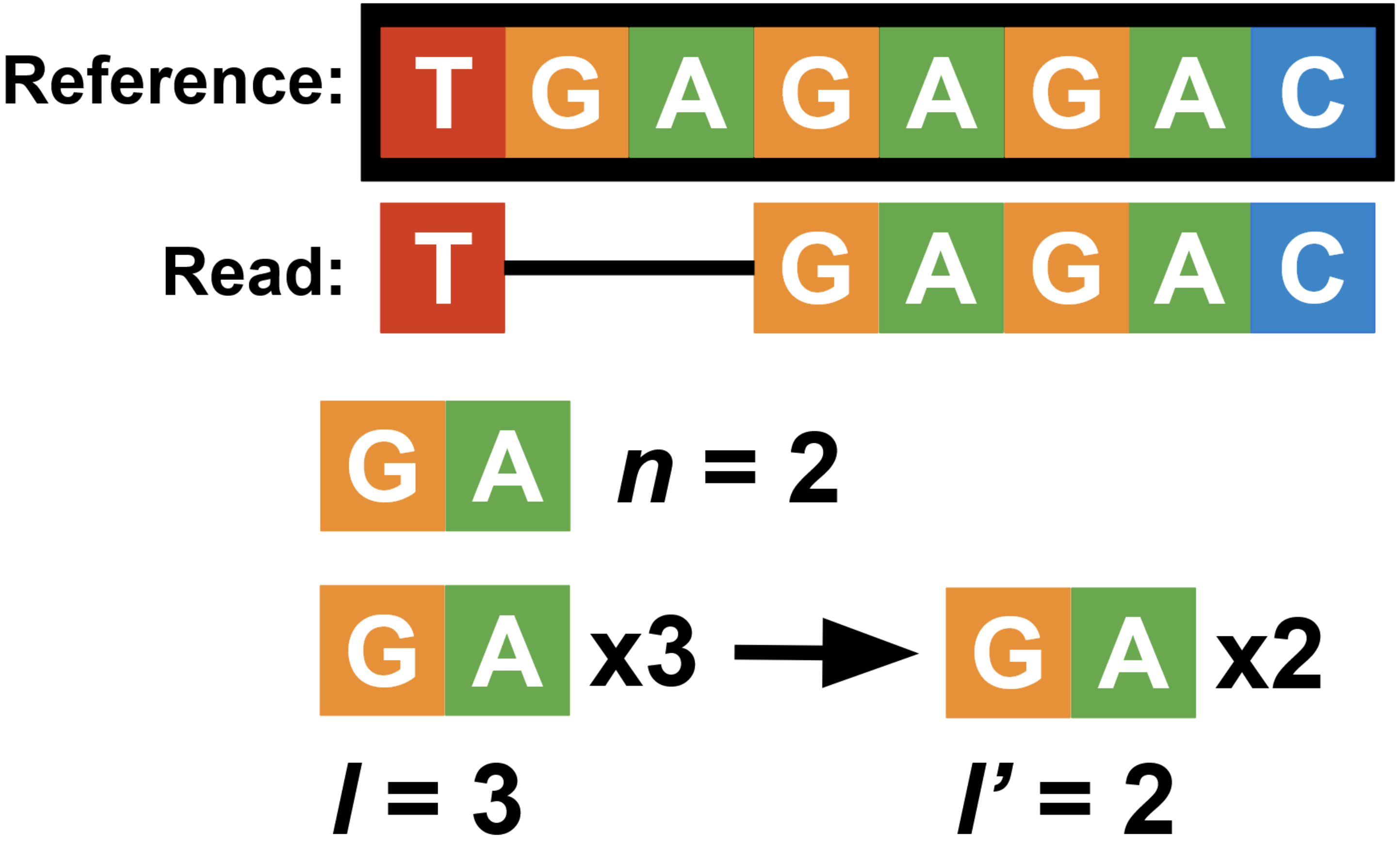
A T T A T T

3-polymer

…

…

Definition: *Copy number*



Reference: T G A G A G A C

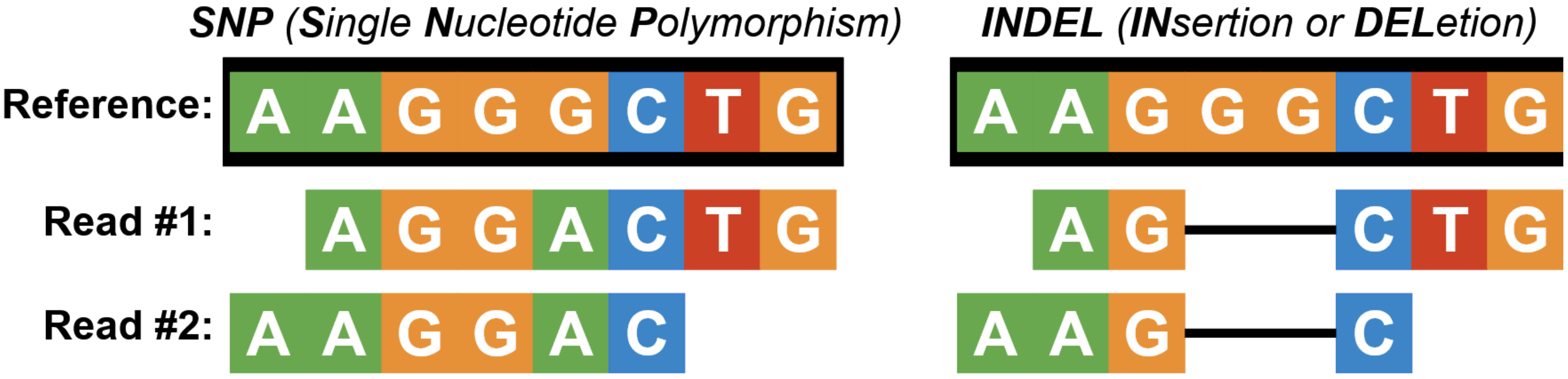
Read: T — G A G A C

G A $n = 2$

G A x3 → G A x2

$l = 3$ $l' = 2$

Motivation: *INDEL Variant Calling Accuracy*

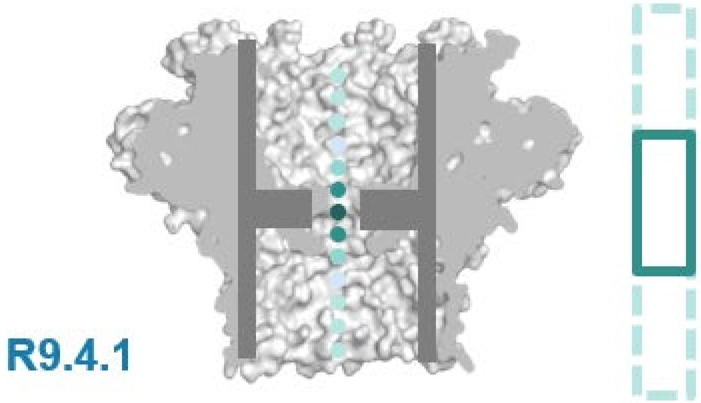


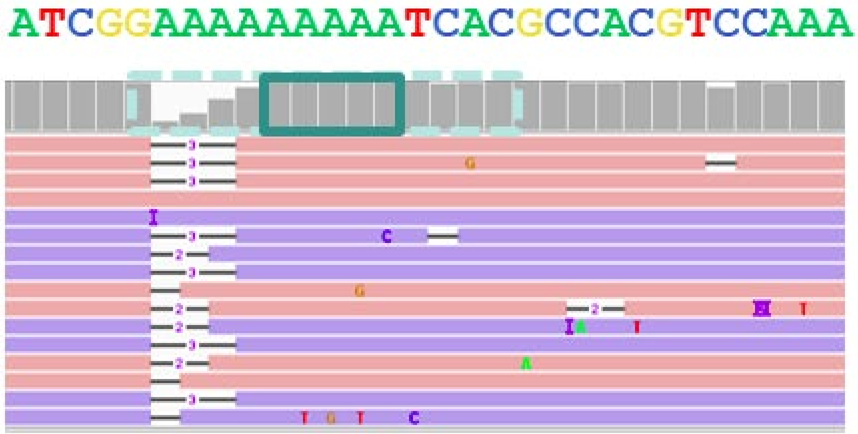
Nanopore R9.4.1 SOTA Accuracy

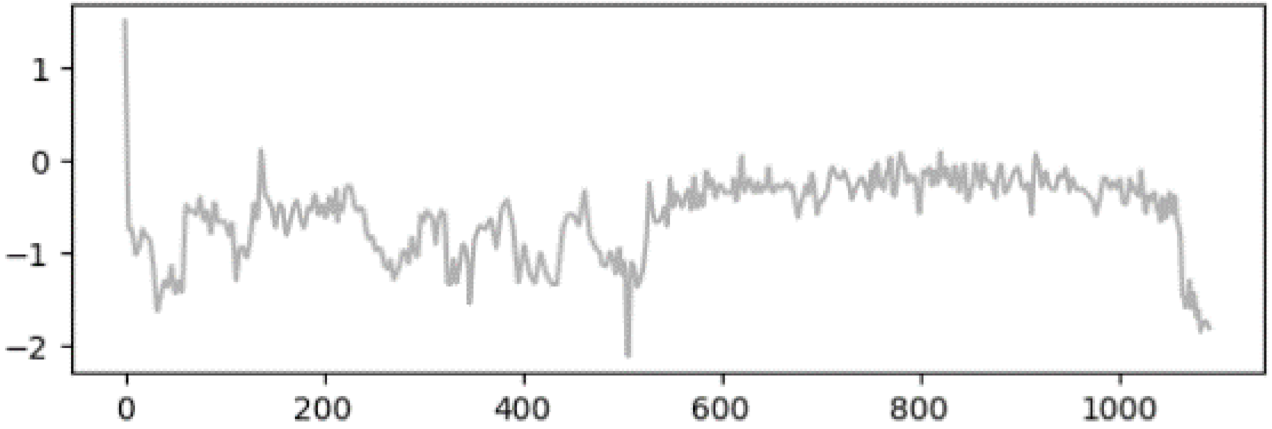
SNPs: 99.7% precision, 99.7% recall

INDELs: 92.8% precision, **76.0% recall**

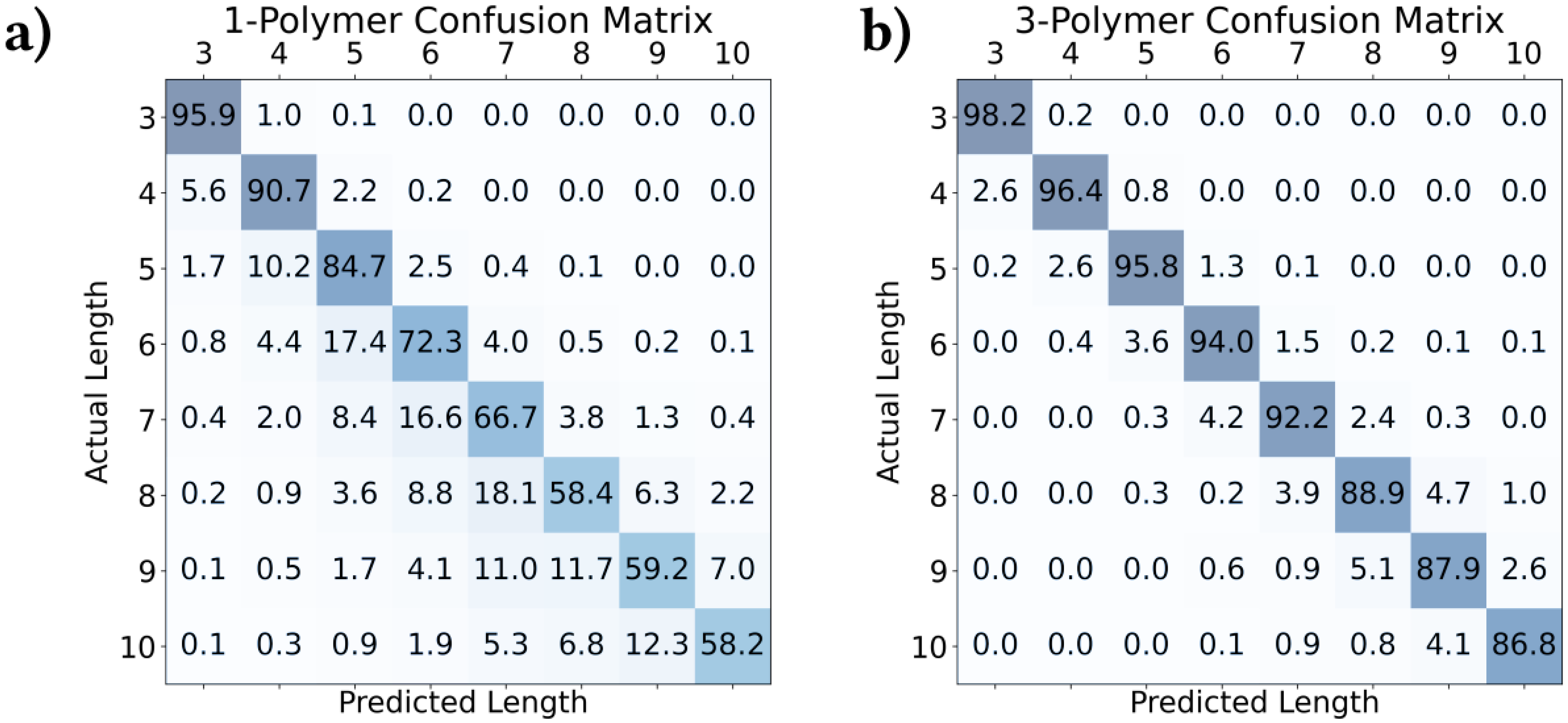
Motivation: *Inaccurate calling in repetitive regions*







Motivation: *Inaccurate calling in repetitive regions*



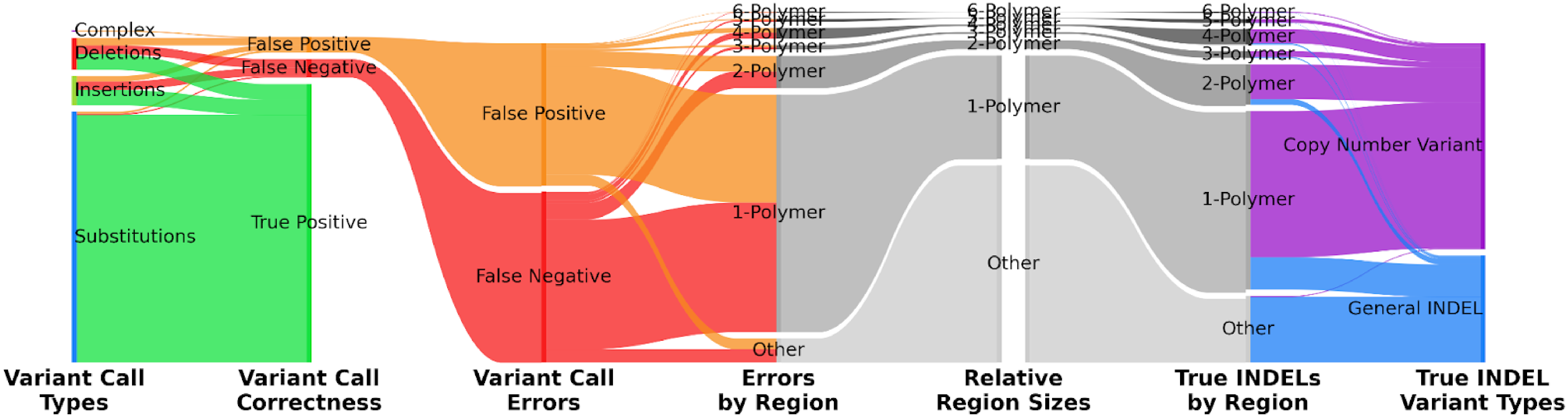
1-Polymer Confusion Matrix

Actual \ Predicted	3	4	5	6	7	8	9	10
3	95.9	1.0	0.1	0.0	0.0	0.0	0.0	0.0
4	5.6	90.7	2.2	0.2	0.0	0.0	0.0	0.0
5	1.7	10.2	84.7	2.5	0.4	0.1	0.0	0.0
6	0.8	4.4	17.4	72.3	4.0	0.5	0.2	0.1
7	0.4	2.0	8.4	16.6	66.7	3.8	1.3	0.4
8	0.2	0.9	3.6	8.8	18.1	58.4	6.3	2.2
9	0.1	0.5	1.7	4.1	11.0	11.7	59.2	7.0
10	0.1	0.3	0.9	1.9	5.3	6.8	12.3	58.2

3-Polymer Confusion Matrix

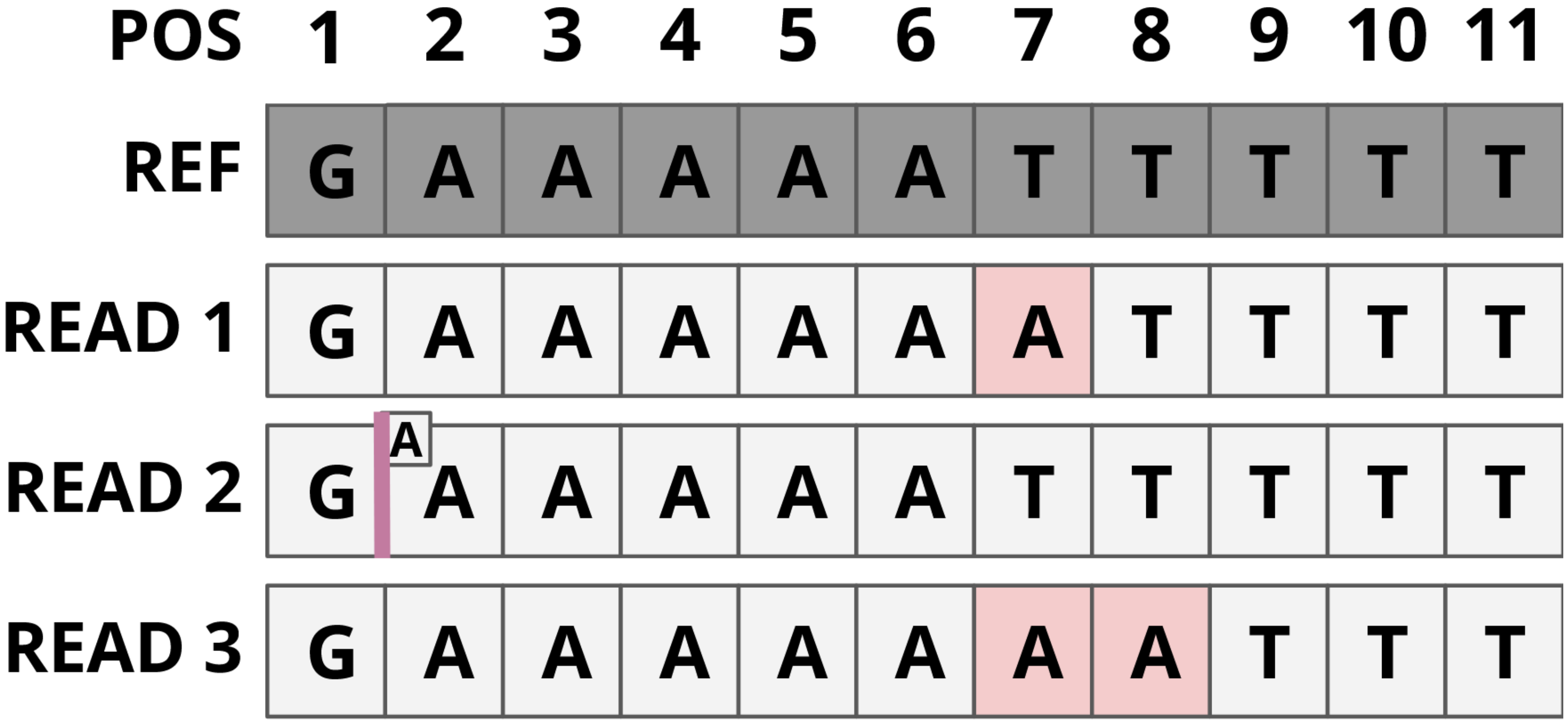
Actual \ Predicted	3	4	5	6	7	8	9	10
3	98.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0
4	2.6	96.4	0.8	0.0	0.0	0.0	0.0	0.0
5	0.2	2.6	95.8	1.3	0.1	0.0	0.0	0.0
6	0.0	0.4	3.6	94.0	1.5	0.2	0.1	0.1
7	0.0	0.0	0.3	4.2	92.2	2.4	0.3	0.0
8	0.0	0.0	0.3	0.2	3.9	88.9	4.7	1.0
9	0.0	0.0	0.0	0.6	0.9	5.1	87.9	2.6
10	0.0	0.0	0.0	0.1	0.9	0.8	4.1	86.8

Motivation: *n-Polymer INDELs are important*



Motivation: *Inconsistent alignments*

MiniMap2 BAM



POS	1	2	3	4	5	6	7	8	9	10	11
REF	G	A	A	A	A	A	T	T	T	T	T
READ 1	G	A	A	A	A	A	A	T	T	T	T
READ 2	G	A A	A	A	A	A	T	T	T	T	T
READ 3	G	A	A	A	A	A	A	A	T	T	T

Motivation: *Alignment doesn’t reflect probability*

Substitutions

	A	C	G	T
A	0	3	3	3
C	3	0	3	3
G	3	3	0	3
T	3	3	3	0

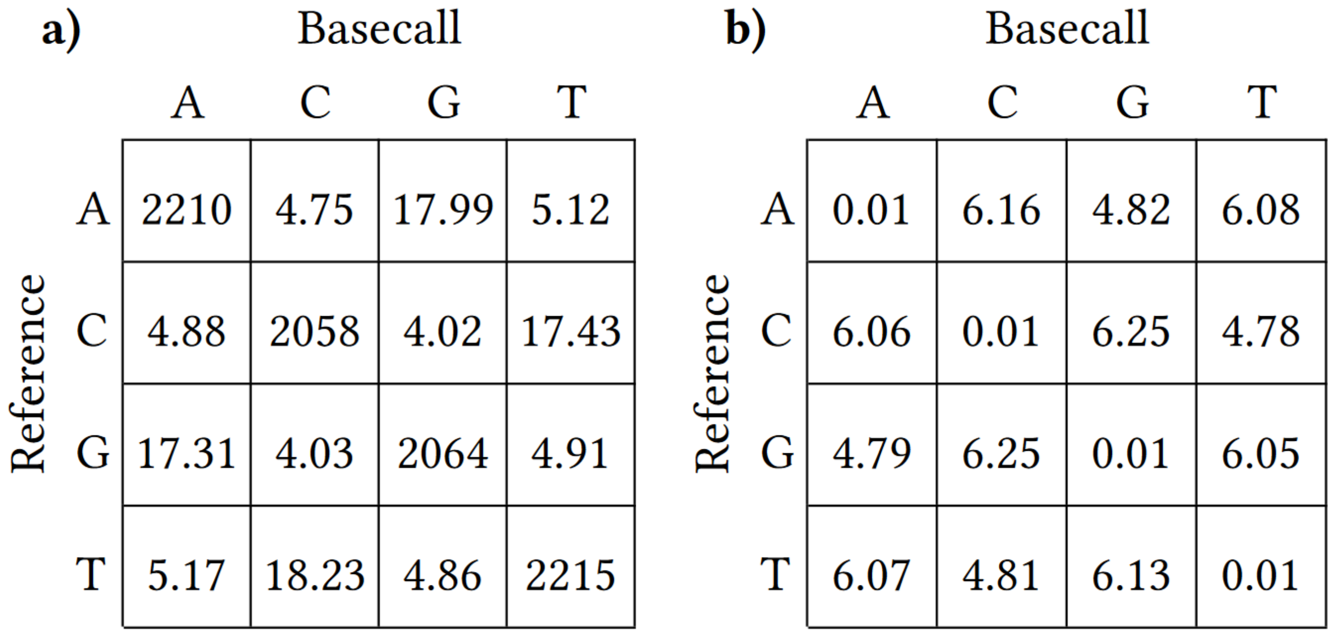
Insertions and Deletions

start: 5, extend: 2

Overview

1. Background
    1. Whole Genome Sequencing
    2. Nanopore Sequencing
    3. Read Alignment
    4. Variant Calling
2. n-Polymer Realigner
    1. Motivation
    2. Algorithm
    3. Results

Algorithm: Use actual probabilities



a)

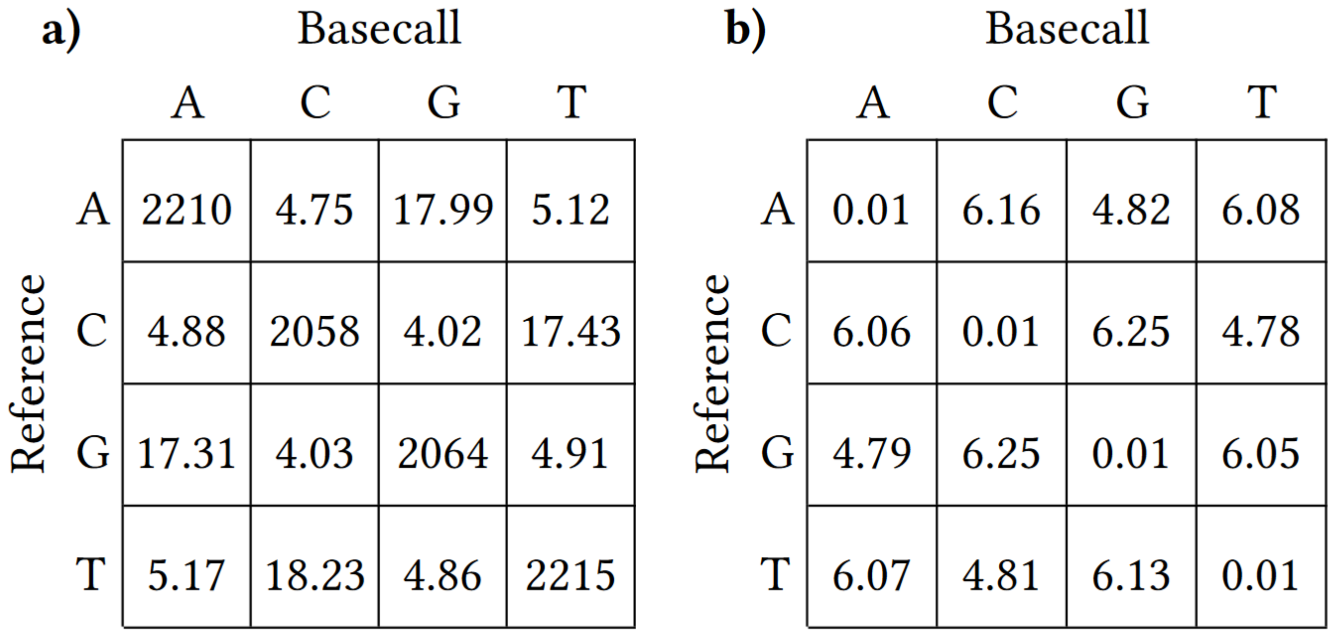
Reference \ Basecall	A	C	G	T
A	2210	4.75	17.99	5.12
C	4.88	2058	4.02	17.43
G	17.31	4.03	2064	4.91
T	5.17	18.23	4.86	2215

b)

Reference \ Basecall	A	C	G	T
A	0.01	6.16	4.82	6.08
C	6.06	0.01	6.25	4.78
G	4.79	6.25	0.01	6.05
T	6.07	4.81	6.13	0.01

Figure 5: a) substitution confusion matrix C_P, count in millions, and b) resulting penalty matrix P.

Algorithm: Use actual probabilities



a)

		Basecall			
		A	C	G	T
Reference	A	2210	4.75	17.99	5.12
	C	4.88	2058	4.02	17.43
	G	17.31	4.03	2064	4.91
	T	5.17	18.23	4.86	2215

b)

		Basecall			
		A	C	G	T
Reference	A	0.01	6.16	4.82	6.08
	C	6.06	0.01	6.25	4.78
	G	4.79	6.25	0.01	6.05
	T	6.07	4.81	6.13	0.01

Figure 5: a) substitution confusion matrix C_P, count in millions, and b) resulting penalty matrix P.

$$P[i,j] \approx -\log \mathbb{P}(x[i] \rightarrow x[j]) \approx -\log \frac{\mathrm{C_P}[\mathrm{i},\mathrm{j}] + \epsilon}{\mathrm{sum}(\mathrm{C_P}[\mathrm{i},:]) + \epsilon}$$

Algorithm: Context-dependent penalties

Original

Substitutions

	A	C	G	T
A	0	3	3	3
C	3	0	3	3
G	3	3	0	3
T	3	3	3	0

Insertions and Deletions

start: 5, extend: 2

nPoRe

Substitutions

	A	C	G	T
A	0	6	5	6
C	5	0	6	4
G	4	6	0	6
T	6	5	6	0

Insertions and Deletions

start: 7, extend: 2

n-Polymers

lookup table

Algorithm: Context-dependent penalties

Original

Substitutions

	A	C	G	T
A	0	3	3	3
C	3	0	3	3
G	3	3	0	3
T	3	3	3	0

Insertions and Deletions

start: 5, extend: 2

nPoRe

Substitutions

	A	C	G	T
A	0	6	5	6
C	5	0	6	4
G	4	6	0	6
T	6	5	6	0

Insertions and Deletions

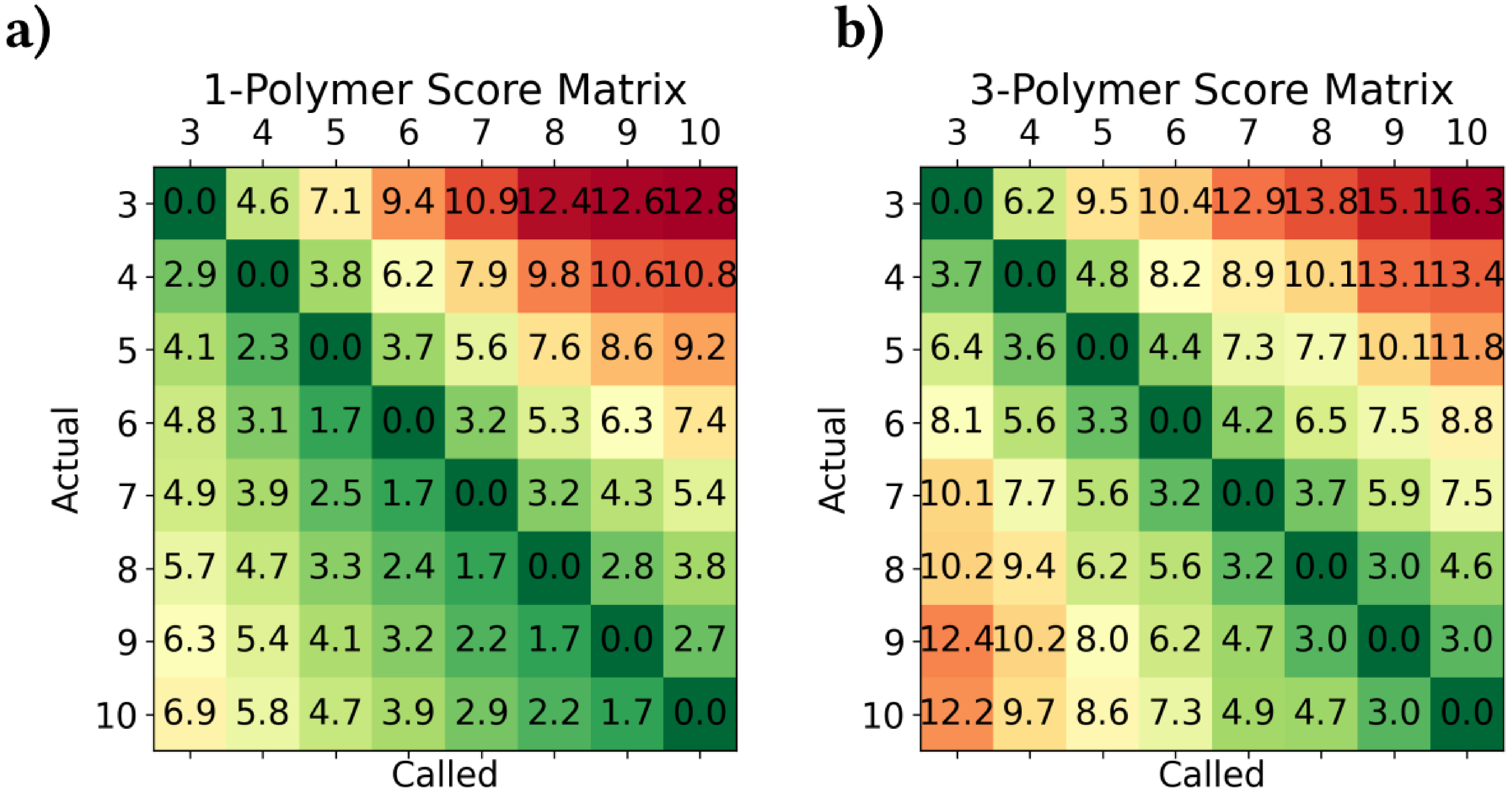
start: 7, extend: 2

n-Polymers

lookup table

$$N[n, i, j] \approx -\log \mathbb{P}(n, i, j) \approx -\log \frac{C_N[n, i, j] + \epsilon}{\mathrm{sum}(C_N[n, i, :]) + \epsilon}$$

Algorithm: *Lookup-based gap penalties*



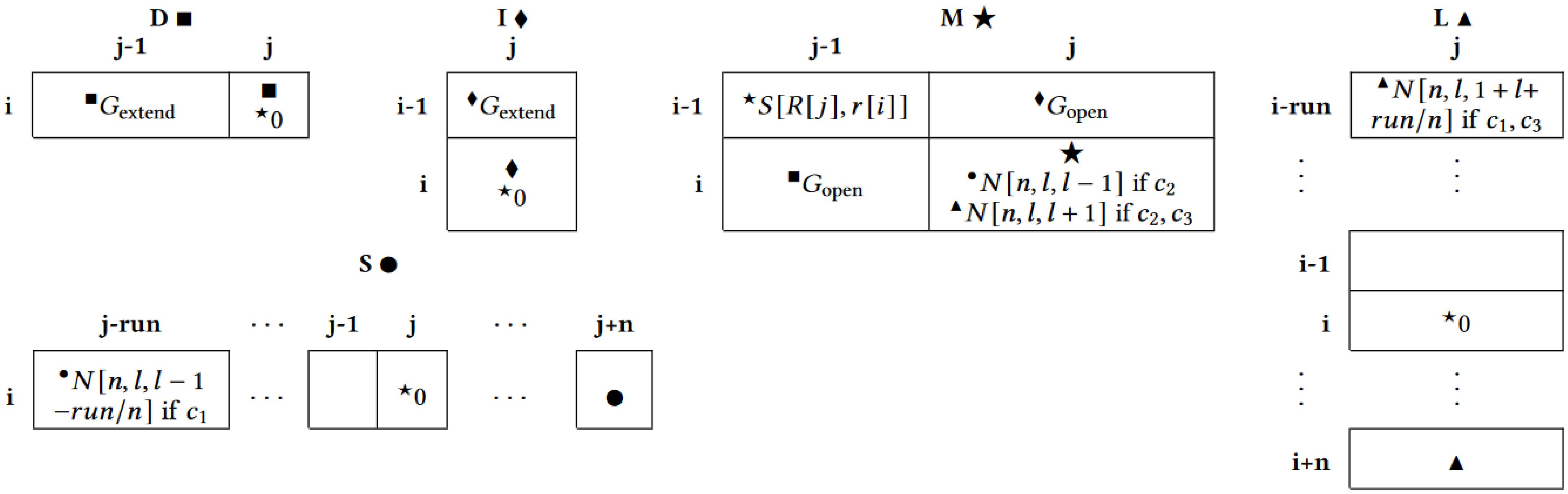
1-Polymer Score Matrix

	3	4	5	6	7	8	9	10
3	0.0	4.6	7.1	9.4	10.9	12.4	12.6	12.8
4	2.9	0.0	3.8	6.2	7.9	9.8	10.6	10.8
5	4.1	2.3	0.0	3.7	5.6	7.6	8.6	9.2
6	4.8	3.1	1.7	0.0	3.2	5.3	6.3	7.4
7	4.9	3.9	2.5	1.7	0.0	3.2	4.3	5.4
8	5.7	4.7	3.3	2.4	1.7	0.0	2.8	3.8
9	6.3	5.4	4.1	3.2	2.2	1.7	0.0	2.7
10	6.9	5.8	4.7	3.9	2.9	2.2	1.7	0.0

3-Polymer Score Matrix

	3	4	5	6	7	8	9	10
3	0.0	6.2	9.5	10.4	12.9	13.8	15.1	16.3
4	3.7	0.0	4.8	8.2	8.9	10.1	13.1	13.4
5	6.4	3.6	0.0	4.4	7.3	7.7	10.1	11.8
6	8.1	5.6	3.3	0.0	4.2	6.5	7.5	8.8
7	10.1	7.7	5.6	3.2	0.0	3.7	5.9	7.5
8	10.2	9.4	6.2	5.6	3.2	0.0	3.0	4.6
9	12.4	10.2	8.0	6.2	4.7	3.0	0.0	3.0
10	12.2	9.7	8.6	7.3	4.9	4.7	3.0	0.0

Algorithm: *Requires five state tables*



D ■

	j-1	j
i	$^{\blacksquare}G_{\text{extend}}$	■ $^{\star}0$

I ♦

	j
i-1	$^{\blacklozenge}G_{\text{extend}}$
i	♦ $^{\star}0$

M ★

	j-1	j
i-1	$^{\star}S[R[j], r[i]]$	$^{\blacklozenge}G_{\text{open}}$
i	$^{\blacksquare}G_{\text{open}}$	★ $^{\bullet}N[n, l, l-1]$ if c_2 $^{\blacktriangle}N[n, l, l+1]$ if c_2, c_3

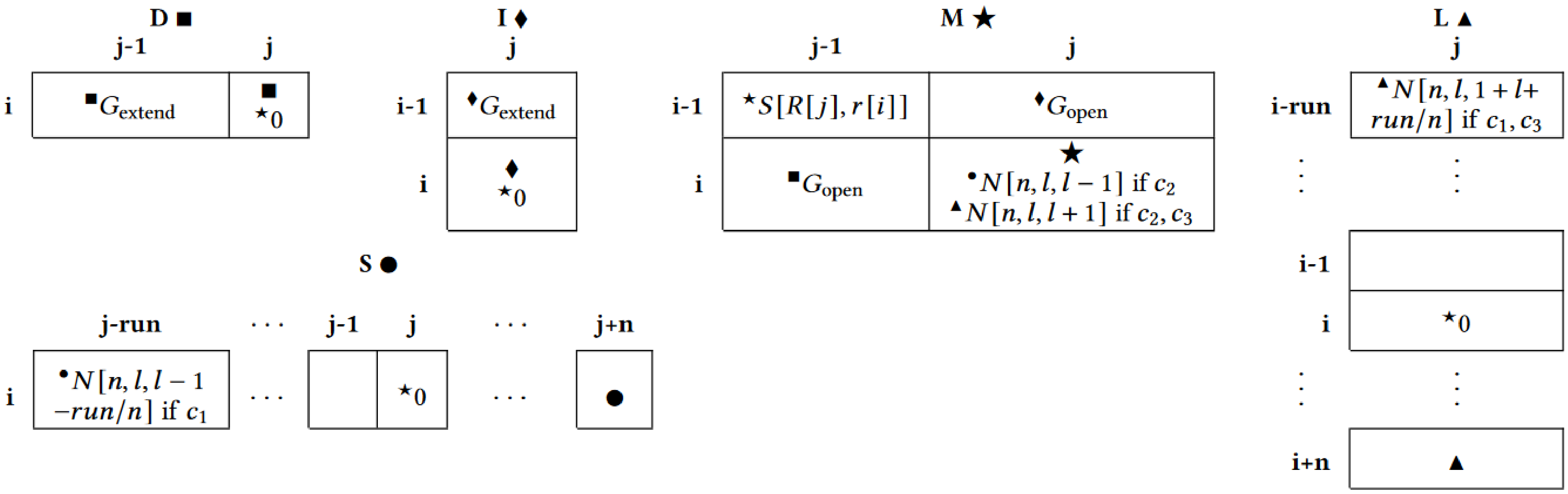
L ▲

	j
i-run	$^{\blacktriangle}N[n, l, 1+l+ run/n]$ if c_1, c_3
⋮	⋮
i-1	
i	$^{\star}0$
⋮	⋮
i+n	▲

S ●

	j-run	. . .	j-1	j	. . .	j+n
i	$^{\bullet}N[n, l, l-1 -run/n]$ if c_1	. . .		$^{\star}0$	. . .	●

Algorithm: *Requires five state tables*



D ■

	j-1	j
i	$^{\blacksquare}G_{\text{extend}}$	■ $^{\star}0$

I ♦

	j
i-1	$^{\blacklozenge}G_{\text{extend}}$
i	♦ $^{\star}0$

M ★

	j-1	j
i-1	$^{\star}S[R[j], r[i]]$	$^{\blacklozenge}G_{\text{open}}$
i	$^{\blacksquare}G_{\text{open}}$	★ $^{\bullet}N[n, l, l-1]$ if c_2 $^{\blacktriangle}N[n, l, l+1]$ if c_2, c_3

L ▲

	j
i-run	$^{\blacktriangle}N[n, l, 1+l+run/n]$ if c_1, c_3
⋮	⋮
i-1	
i	$^{\star}0$
⋮	⋮
i+n	▲

S ●

	j-run	. . .	j-1	j	. . .	j+n
i	$^{\bullet}N[n, l, l-1-run/n]$ if c_1	. . .		$^{\star}0$	. . .	●

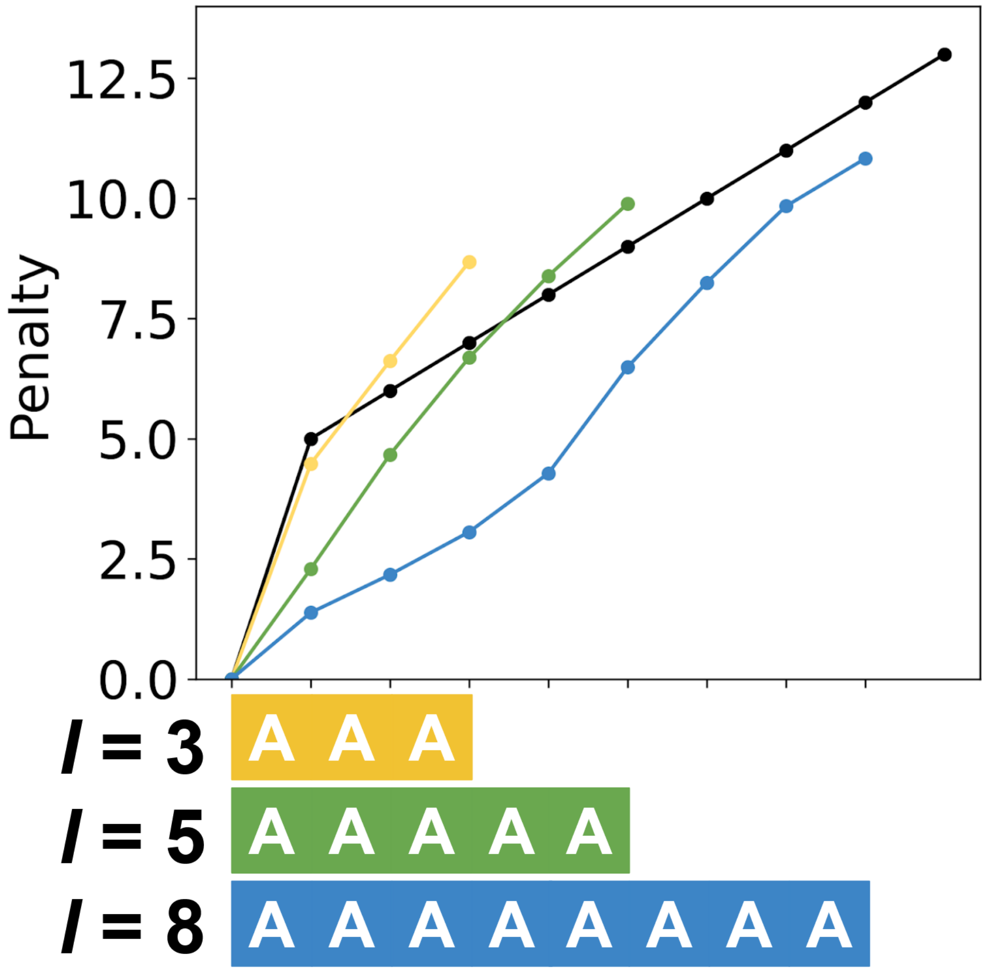
$c_1 = \quad l > 0$ start of repeat unit

$c_2 = \quad l > 0$ and $idx == 0$ start of n-polymer

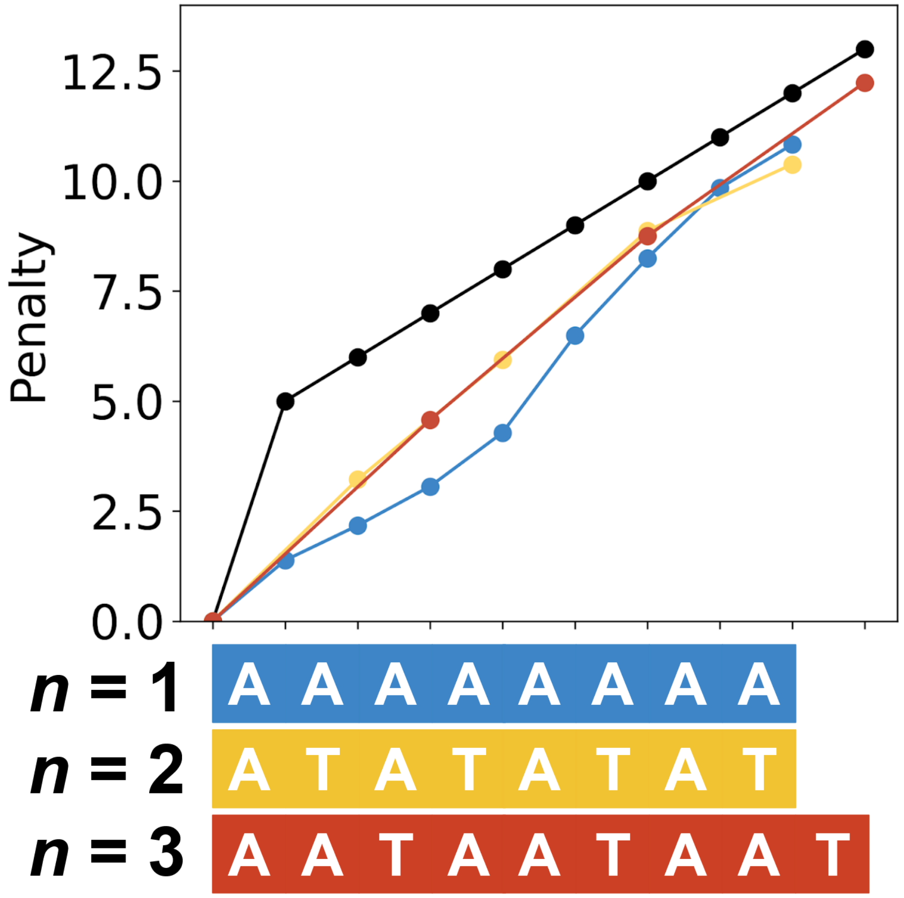
$c_3 = \quad R[j+1 : j+1+n] ==$
$r[i+1 : i+1+n]$ next n bases of r match R

Algorithm: *Properties*

Penalty depends on *l*, *l’*

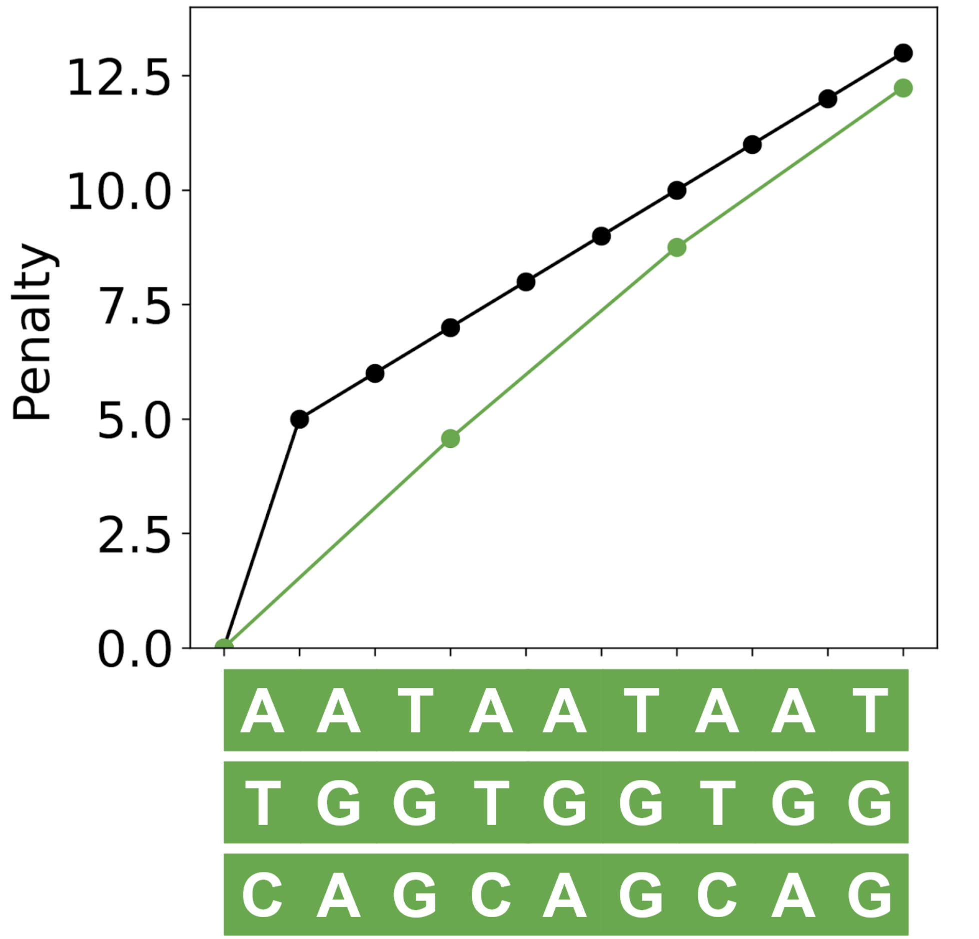


Penalty depends on *n*, only CNVs allowed

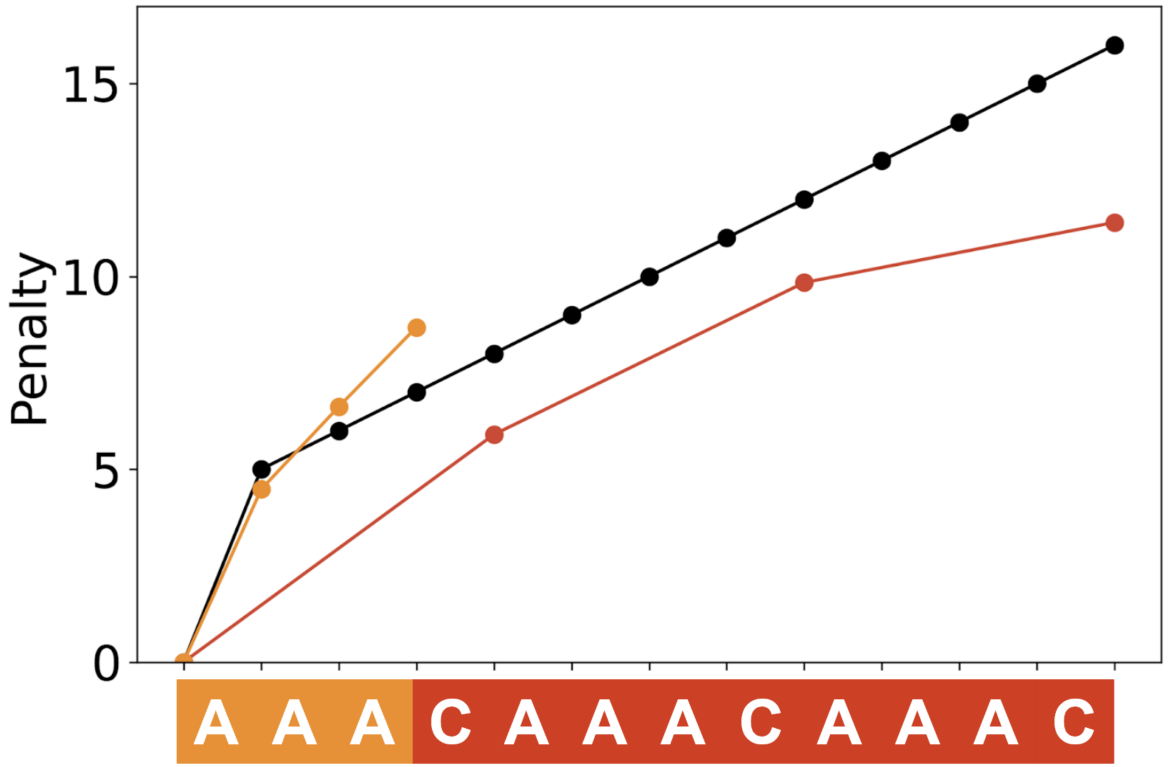


Algorithm: *Properties*

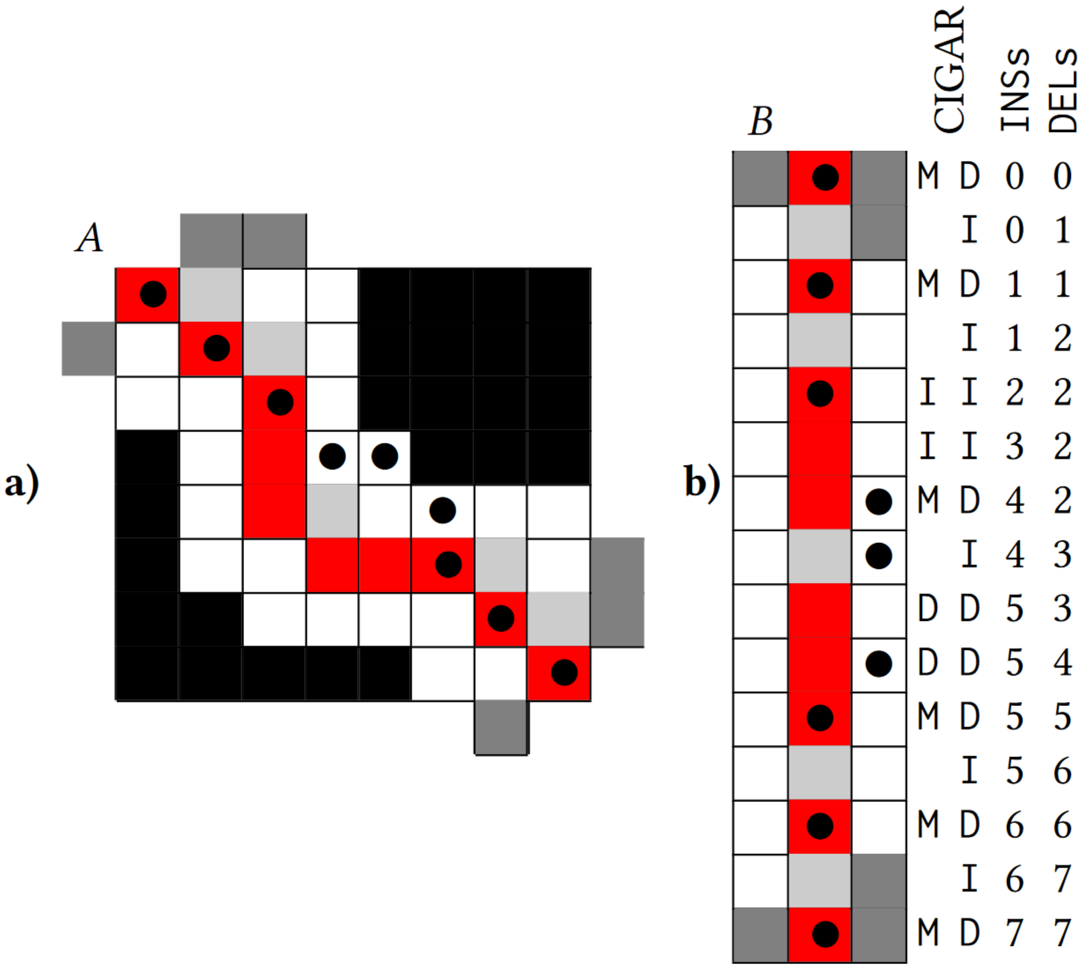
Sequence is not considered



Overlaps are allowed



Algorithm: *Banded alignment for efficiency*

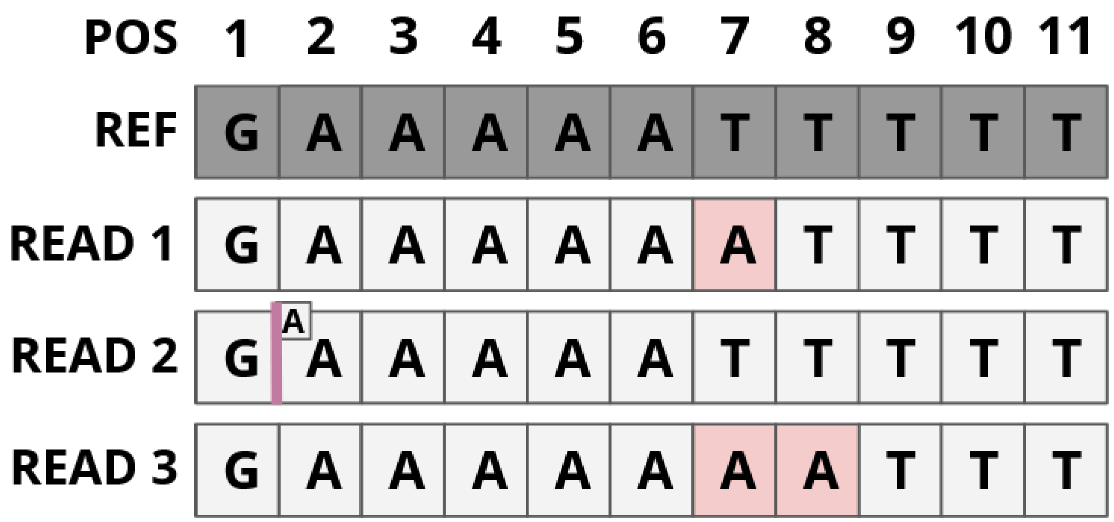


CIGAR		INSs	DELs
M	D	0	0
	I	0	1
M	D	1	1
	I	1	2
I	I	2	2
I	I	3	2
M	D	4	2
	I	4	3
D	D	5	3
D	D	5	4
M	D	5	5
	I	5	6
M	D	6	6
	I	6	7
M	D	7	7

Figure 7: Follow banding matrix transformation $A \rightarrow B$

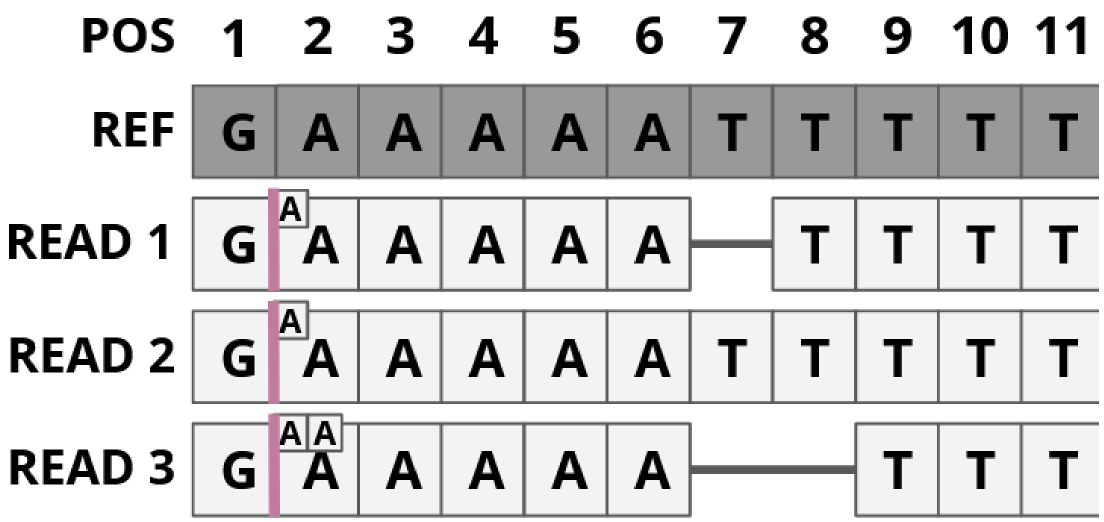
Algorithm: *Goal is improving alignment consistency*

MiniMap2 BAM



POS	1	2	3	4	5	6	7	8	9	10	11
REF	G	A	A	A	A	A	T	T	T	T	T
READ 1	G	A	A	A	A	A	A	T	T	T	T
READ 2	G	A A	A	A	A	A	T	T	T	T	T
READ 3	G	A	A	A	A	A	A	A	T	T	T

nPoRe BAM



POS	1	2	3	4	5	6	7	8	9	10	11
REF	G	A	A	A	A	A	T	T	T	T	T
READ 1	G	A A	A	A	A	A	—	T	T	T	T
READ 2	G	A A	A	A	A	A	T	T	T	T	T
READ 3	G	A A A	A	A	A	A	—	—	T	T	T

Overview

1. Background
    1. Whole Genome Sequencing
    2. Nanopore Sequencing
    3. Read Alignment
    4. Variant Calling
2. n-Polymer Realigner
    1. Motivation
    2. Algorithm
    3. Results

Results: *Alignment consistency*

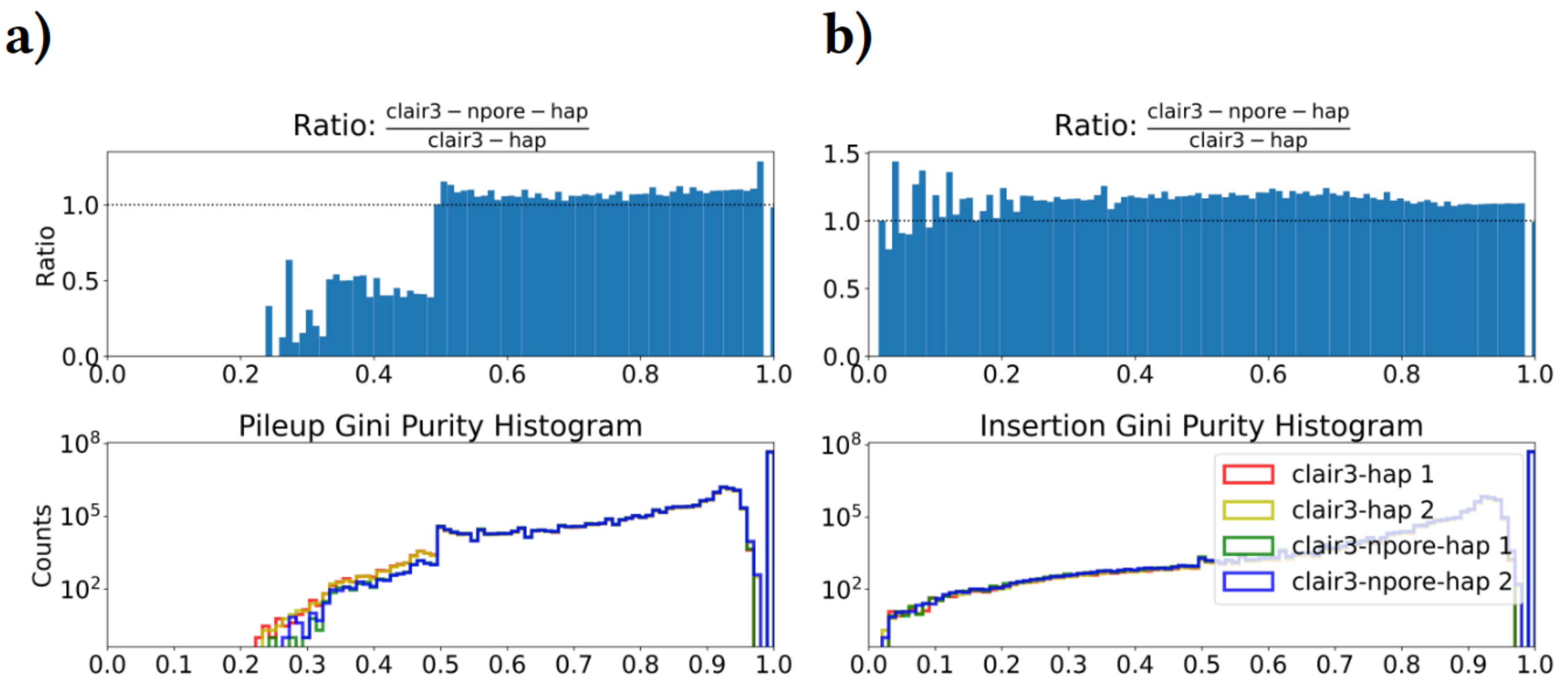
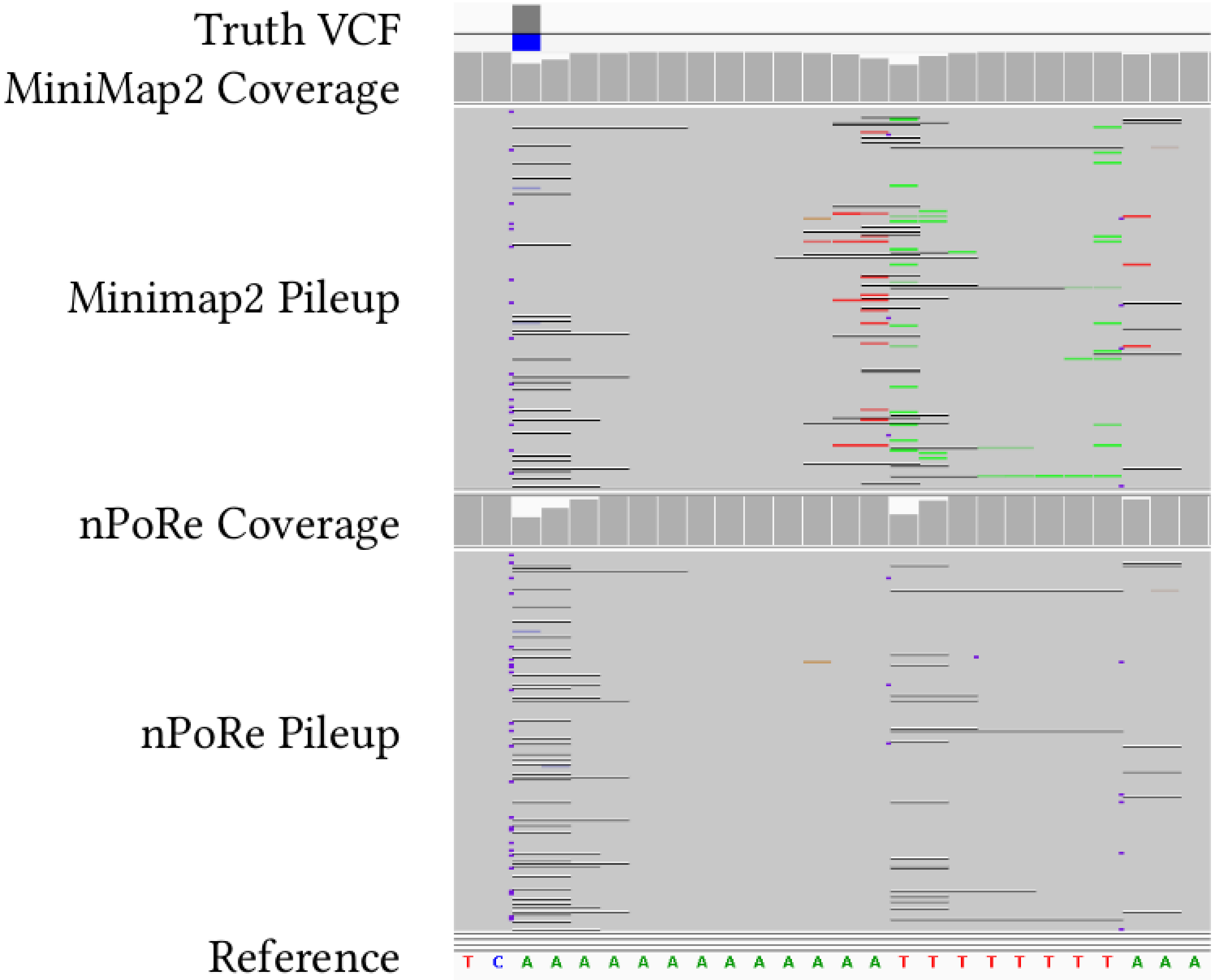


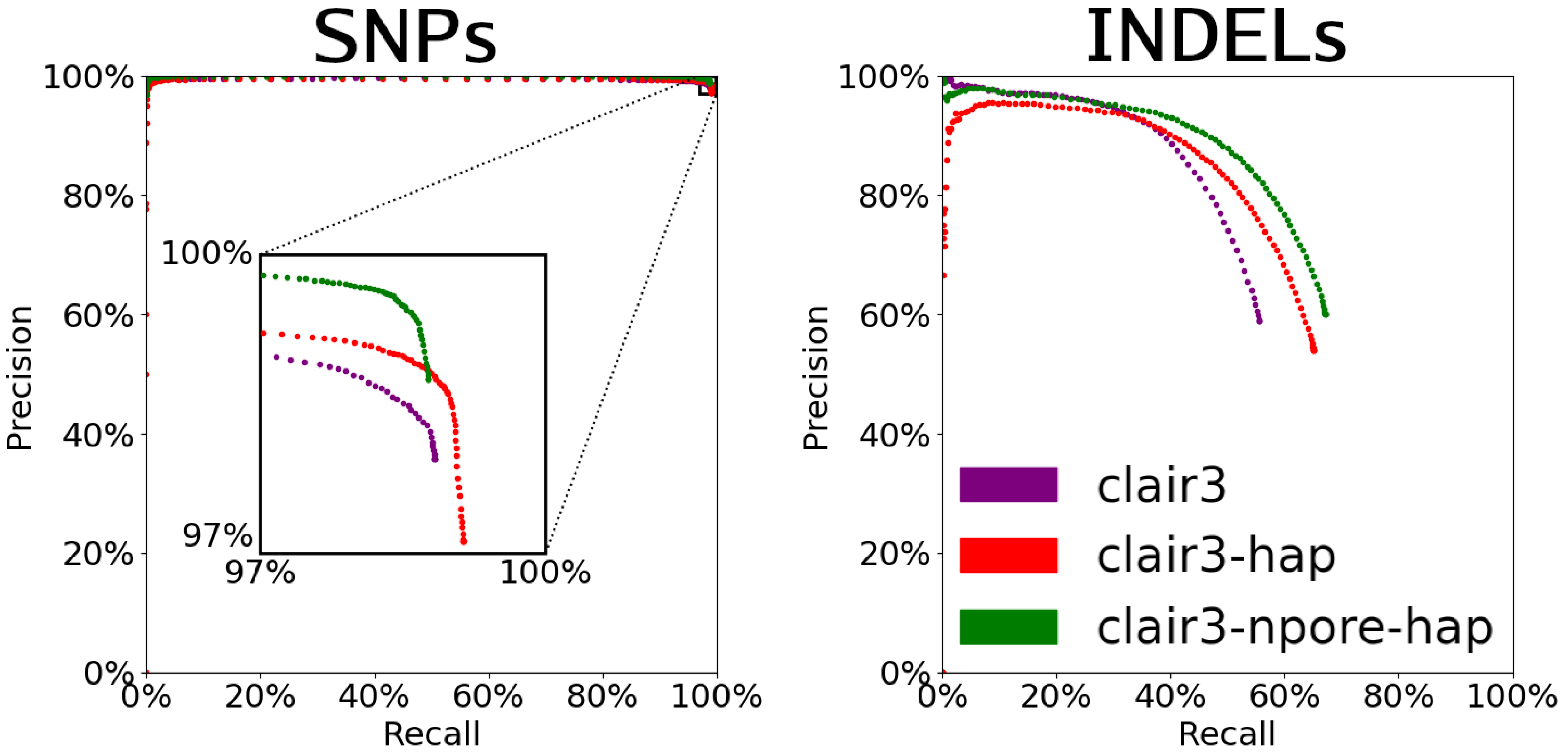
Figure 9: Read concordance: Gini purity histograms for a) pileup columns and b) insertions

$$GP = \sum_{i=1}^{n} P(i)^2$$

Results: *Alignment consistency example*

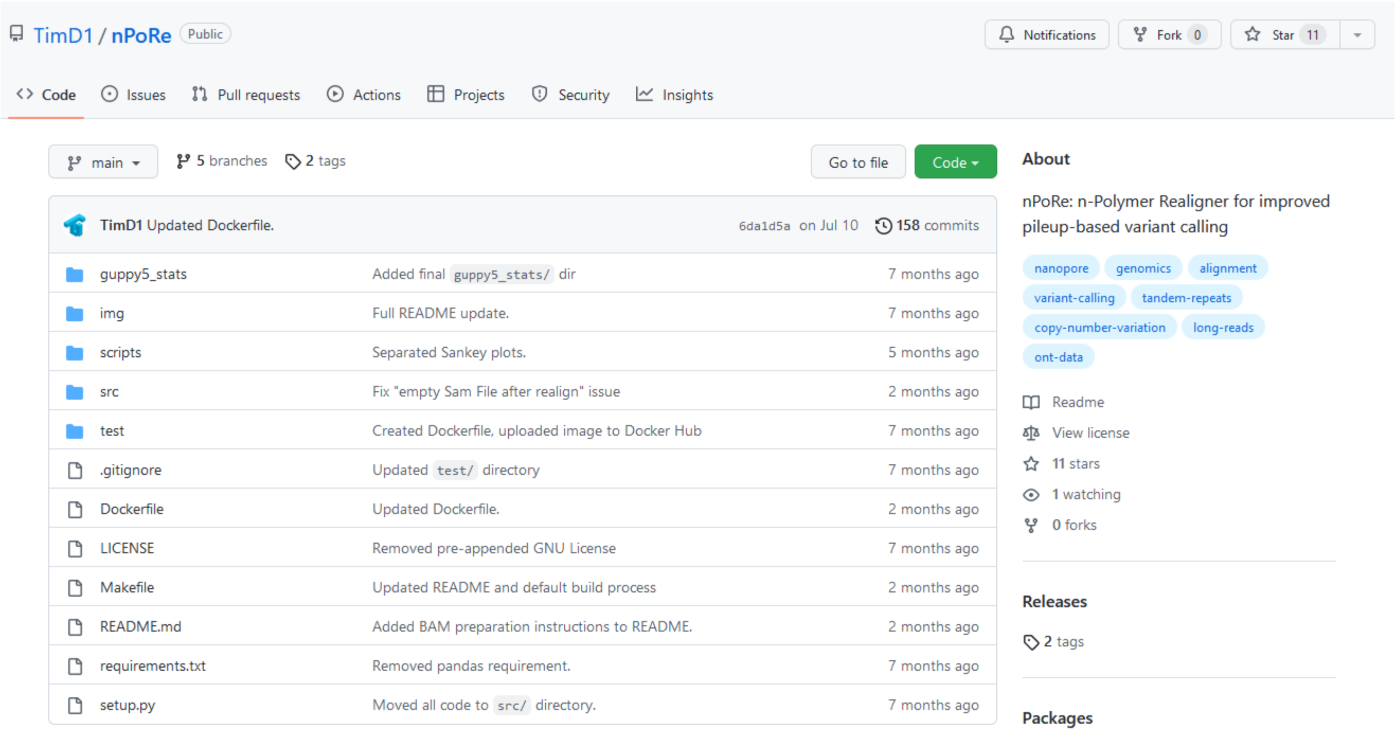


Results: *Accuracy improvements*



Results: *GitHub code*

https://github.com/TimD1/nPoRe



TimD1 / nPoRe Public

TimD1 Updated Dockerfile. 6da1d5a on Jul 10 158 commits

guppy5_stats	Added final `guppy5_stats/` dir	7 months ago
img	Full README update.	7 months ago
scripts	Separated Sankey plots.	5 months ago
src	Fix "empty Sam File after realign" issue	2 months ago
test	Created Dockerfile, uploaded image to Docker Hub	7 months ago
.gitignore	Updated `test/` directory	7 months ago
Dockerfile	Updated Dockerfile.	2 months ago
LICENSE	Removed pre-appended GNU License	7 months ago
Makefile	Updated README and default build process	2 months ago
README.md	Added BAM preparation instructions to README.	2 months ago
requirements.txt	Removed pandas requirement.	7 months ago
setup.py	Moved all code to `src/` directory.	7 months ago

About

nPoRe: n-Polymer Realigner for improved pileup-based variant calling

nanopore genomics alignment variant-calling tandem-repeats copy-number-variation long-reads ont-data

Readme

View license

11 stars

1 watching

0 forks

Releases

2 tags

Packages

Funding

This project was supported by the National Science Foundation Graduate Research Fellowship under Grant 1841052. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation

Questions?

Thanks for listening!

If you'd like to talk about research, please reach out!

Email: timdunn@umich.edu

Twitter: @T1MD1