TOTALRECALLER: Improved Accuracy and Performance via Integrated Alignment & Base-Calling

Fabian Menges

New York, 2011/3/10

Overview

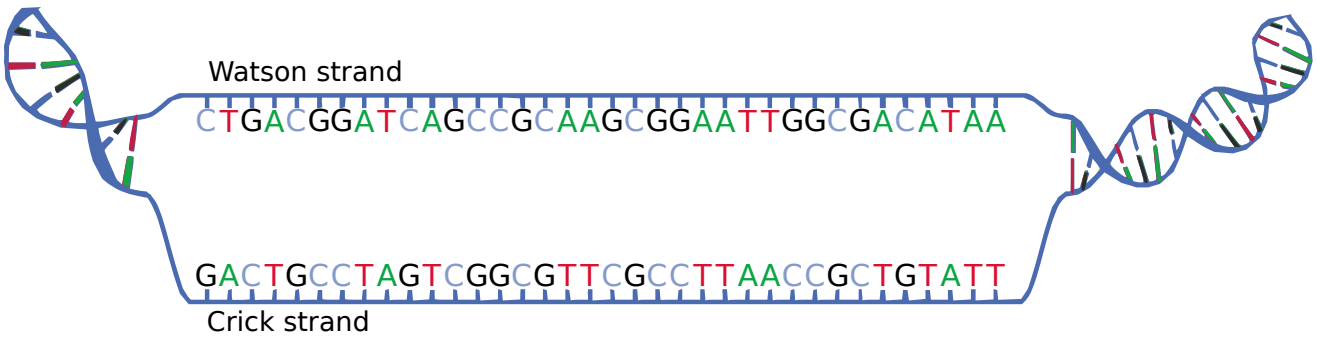
Introduction

Basecalling with alingment
- Sequencing technology
- Linear error model
- Sequence Alignment
- Branch and Bound

Results

Overview

DNA


Watson strand
CTGACGGATCAGCCGCAAGCGGAATTGGCGACATAA
GACTGCCTAGTCGGCGTTCGCCTTAACCGCTGTATT
Crick strand

DNA

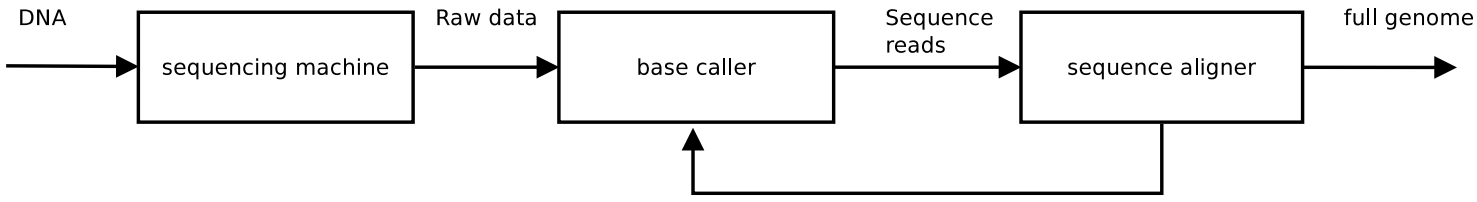
DNA of different organisms:

Organism type	Organism	Genome size (base pairs)
Virus	Bacteriophage MS2	3,569
Virus	SV40	5,224
Bacterium	Haemophilus influenzae	1,830,000
Plant	Populus trichocarpa	480,000,000
Yeast	Saccharomyces cerevisiae	12,100,000
Fungus	Aspergillus nidulans	30,000,000
Insect	Apis mellifera (honey bee)	1,770,000,000
Fish	Tetraodon nigroviridis	385,000,000
Fish	Protopterus aethiopicus	130,000,000,000
Mammal	Homo sapiens	3,200,000,000

Re-sequencing Pipeline

Conventional *re-sequencing* pipeline:

Proposed *re-sequencing* pipeline:



Motivation: Avoiding errors in *sequence reads*.

Overview

Introduction

Basecalling with alingment

- Sequencing technology
- Linear error model
- Sequence Alignment
- Branch and Bound

Results

Overview

Illumina sequencing machine

sequence reads with a length of up to 125BP.

Flow Cell:

Genome Analyzer IIe:

Input: Cluster of DNA-Fragments
Output: Raw sequence intensities

Source: `http://www.illumina.com`

Illumina Intensities

Sequencing machine:

- Input: DNA-fragments
- Output: Intensities

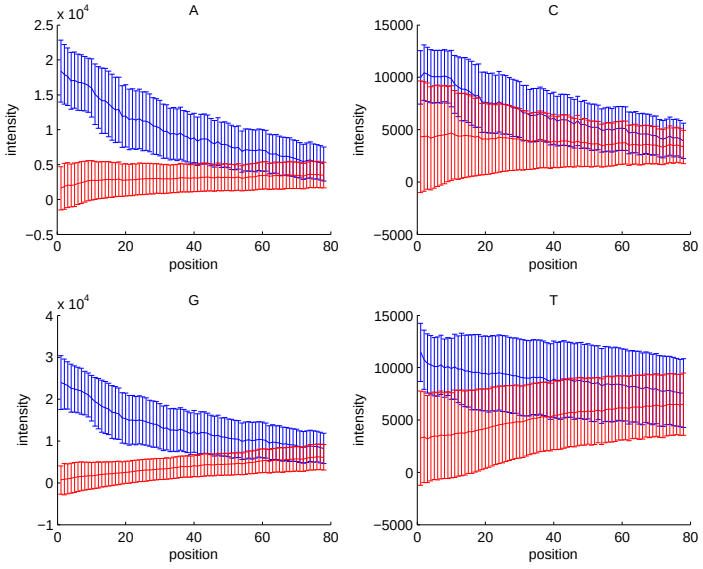
Cycle	Channel A	Channel C	Channel G	Channel T
1	15.7	−19.5	**3812.9**	1398.9
2	−29.0	41.6	365.5	**1200.5**
3	14.4	36.6	379.1	**1447.0**
⋮	⋮	⋮	⋮	⋮
76	837.4	549.4	1098.8	841.7
77	633.0	491.8	1280.7	901.4
78	602.9	558.2	1036.2	860.9

Basecaller:

⇒ *sequence read:* GTT...???

Illumina Intensities

Level distance: High-level(blue) , Low-level(red)



⇒ The level distance between low and high signal levels is too small in order to reconstruct the correct sequence read.

Overview

Introduction

Basecalling with alingment

- Sequencing technology
- Linear error model
- Sequence Alignment
- Branch and Bound

Results

Causes of Errors

- **Fading:** Signal strength decreases with the number cycles.
- **Crosstalk:** The channels are not independent form one another.
- **Lagging:** There is interference between two consecutive cycles.

Linear error model

Crosstalk and fading

- Cycle: $k \in \mathbb{N}$:
- **Input:** Raw intensities: $\boldsymbol{I}_k = \begin{pmatrix} I_A^k & I_C^k & I_G^k & I_T^k \end{pmatrix}^\top$
- **Output:** Filtered intensities: $\boldsymbol{X}_k = \begin{pmatrix} X_A^k & X_C^k & X_G^k & X_T^k \end{pmatrix}^\top$
- Crosstalk matrix: $\boldsymbol{A}_k \in \mathbb{R}^{4\times 4}$
- Model: $\boldsymbol{I}_k = \boldsymbol{A}_k \cdot \boldsymbol{X}_k$

$$\boldsymbol{A}_k = \begin{pmatrix} \mu_{k,A}^A & \mu_{k,C}^A & \mu_{k,G}^A & \mu_{k,T}^A \\ \mu_{k,A}^C & \mu_{k,C}^C & \mu_{k,G}^C & \mu_{k,T}^C \\ \mu_{k,A}^G & \mu_{k,C}^G & \mu_{k,G}^G & \mu_{k,T}^G \\ \mu_{k,A}^T & \mu_{k,C}^T & \mu_{k,G}^T & \mu_{k,T}^T \end{pmatrix}$$

- **Filter:** Filtered intensities: $\boldsymbol{X}_k = \boldsymbol{A}_k^{-1} \cdot \boldsymbol{I}_k$

Linear error model

Crosstalk, fading and lagging

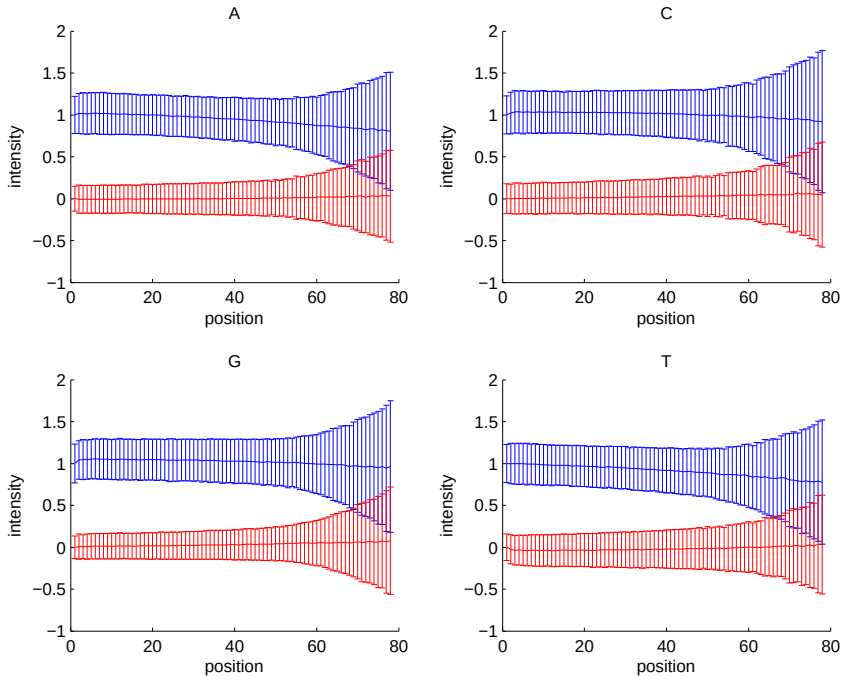
- Cycle: $k \in \mathbb{N}$:
- **Input:** Raw intensities: $\boldsymbol{I}_k = \begin{pmatrix} I_A^k & I_C^k & I_G^k & I_T^k \end{pmatrix}^\top$
- Crosstalk matrix: $\boldsymbol{A}_k \in \mathbb{R}^{4\times 4}$
- Lagging matrix: $\boldsymbol{\Upsilon}_k \in \mathbb{R}^{4\times 4}$
- **Output:** Filtered intensities:

$$\begin{pmatrix} \boldsymbol{X}_{k-1} \\ \boldsymbol{X}_k \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{A}_{k-1} & 0 \\ \boldsymbol{\Upsilon}_k & \boldsymbol{A}_k \end{pmatrix}^{-1}}_{\boldsymbol{G}_k \in \mathbb{R}^{8\times 8}} \cdot \begin{pmatrix} \boldsymbol{I}_{k-1} \\ \boldsymbol{I}_k \end{pmatrix}$$

Basecalling with alingment

Linear error model

Result



⇒ The level distance between the high and low level allows to separate the channels up to the 60th cycle.

Goal: Correct reconstruction of all bases in a sequence read!

Overview

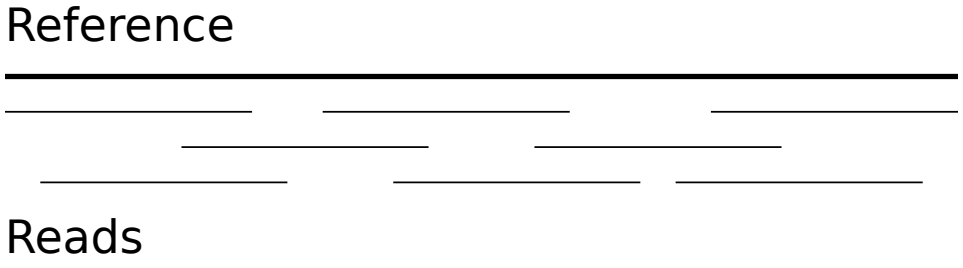
Introduction

Basecalling with alingment

- Sequencing technology
- Linear error model
- Sequence Alignment
- Branch and Bound

Results

Sequence Alignment

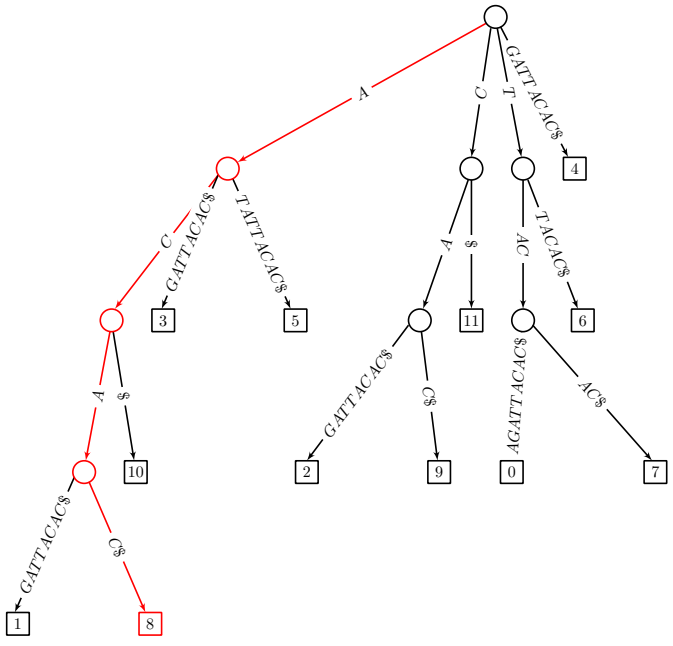


- Discover the correct position for each *sequence reads* in a given reference genome.
- Can be described a search of *sequence reads* in the reference genome considering (*mismatches*).

Problem: Fast search with mismatches in large reference genome ($\sim 3 \cdot 10^9$BP).

Suffix Tree

Example: Reference T = "TACAGATTACAC$"



- Advantage: Allow to search for a *read* with length n in O(n).
- Disadvantage: Requires for a reference with length m $O(m^2)$ memory.

Burrows-Wheeler Transformation

- Reversible transformation.
- Was developed for a compression algorithm

Example:

TACAGATTACAC**$**		**$**TACAGATTACA**C**		
ACAGATTACAC**$**T		AC**$**TACAGATTA**C**		
CAGATTACAC**$**TA		ACAC**$**TACAGAT**T**		
AGATTACAC**$**TAC		ACAGATTACAC**$T**		
GATTACAC**$**TACA		AGATTACAC**$**TA**C**		
ATTACAC**$**TACAG		ATTACAC**$**TACA**G**		
TTACAC**$**TACAGA	$\Longrightarrow$	C**$**TACAGATTAC**A**	$\Longrightarrow$	**CCTTCGAAAAT$A**
TACAC**$**TACAGAT		CAC**$**TACAGATT**A**		
ACAC**$**TACAGATT		CAGATTACAC**$**T**A**		
CAC**$**TACAGATTA		GATTACAC**$**TAC**A**		
AC**$**TACAGATTAC		TACAC**$**TACAGA**T**		
C**$**TACAGATTACA		TACAGATTACAC**$**		
$TACAGATTACAC		TTACAC**$**TACAG**A**		

Ferragina-Manzini search

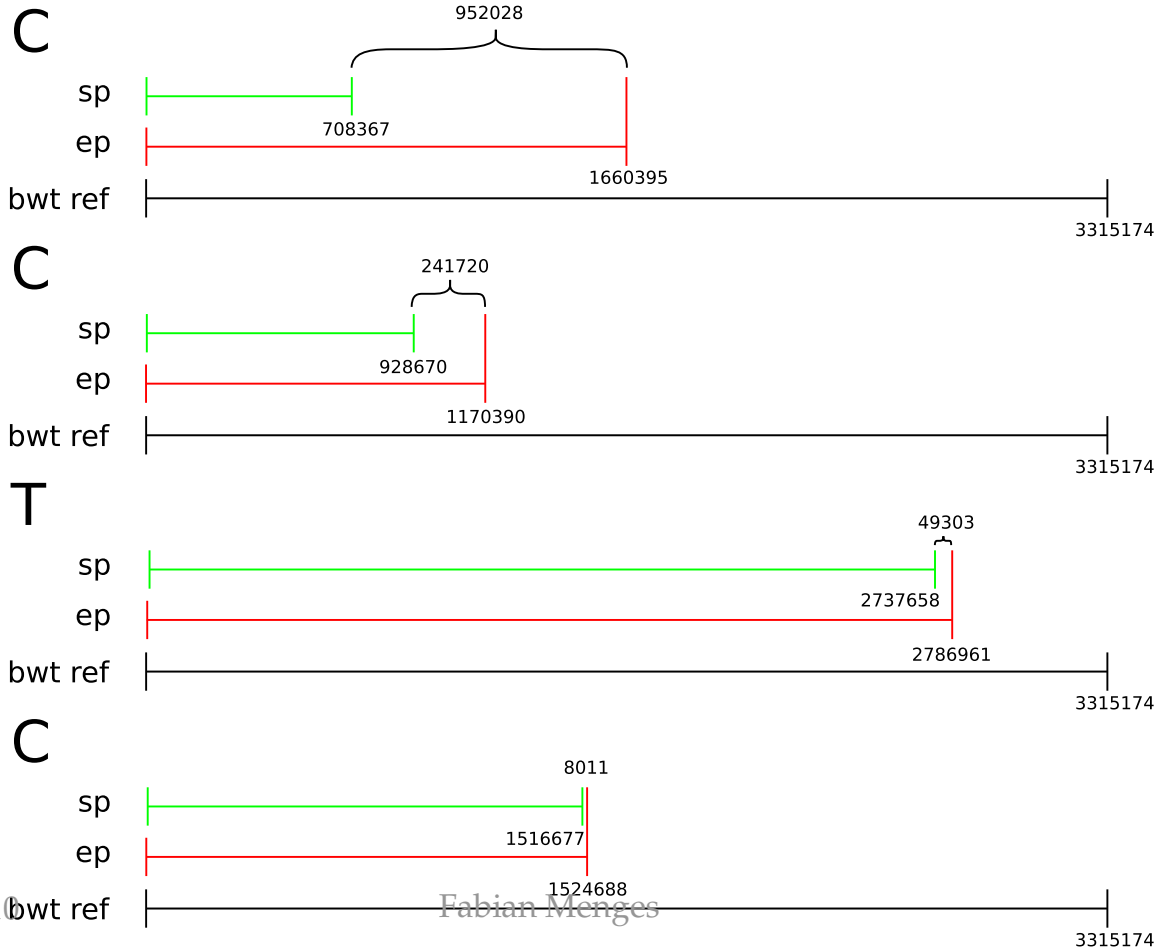
Allows to search the suffix tree through the BWT.

Algorithm 1: FM_search

```
/* Ferragina-Manzini search algorithm                                   */
          P search pattern
input :   n length of P
          F table of accumulated character frequencies
output: ep − sp + 1 number of occurrences of P in T
i = n − 1;
/* i is a index to the last position of the pattern                     */
c = P[i], /* set c to the last character of the pattern                */
sp = F[c − 1] + 1;
/* set the start index                                                 */
/* c-1 represents the lexicographical previous character               */
ep = F[c];
/* set the end index                                                   */
while (sp ≤ ep) and (i ≥ 1) do
    i = i − 1;
    /* decrement the position                                          */
    c = P[i];
    /* get the next character                                          */
    sp = F[c − 1] + C(c, sp − 1) + 1;
    /* update the start index                                          */
    ep = F[c − 1] + C(c, ep);
    /* update the end index                                            */
if (ep < sp) then
    return "pattern not found"
else
    return "found (ep − sp + 1) occurrences"
```

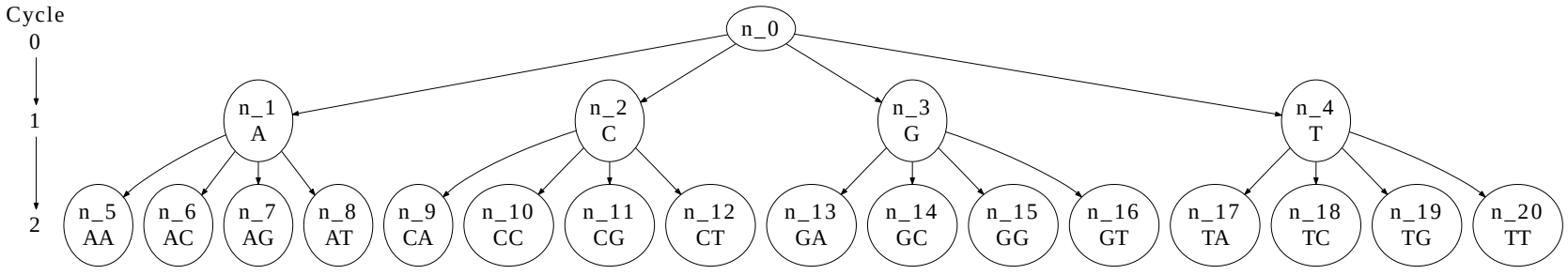
Basecalling with alingment

Ferragina-Manzini search



Overview

Basecalling by building a tree



- Every cycle the tree grows in depth. At cycle k the tree has 4^k leaves and $\sum_{i=1}^{k} 4^i$ nodes.
- Every node represents a possible correct *sequence read*.
- Every node has specific probability of being the correct *sequence read*.
- The node with the highest probability represents the best solution.

Problem: Too many nodes! How is the probability for correctness computed?

Branch and Bound

The branch and bound algorithm allows to reduce the number of nodes that need to be considered. Every cycle the following three steps are performed:

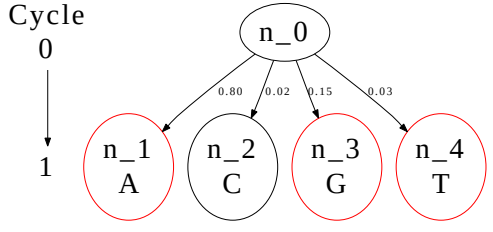
- **Branch:** All nodes in the set of possible solutions are expanded.
- **Bound:** The new child nodes are weighted according to there probability for correctness: a score function.
- **Pruning:** Only the best b nodes are kept in the set of possible solutions.

$\Rightarrow$ Branch and Bound reduces the number of nodes that need to be considered by building the tree only partially.

$\Rightarrow$ The correct/best solution can not be guaranteed!

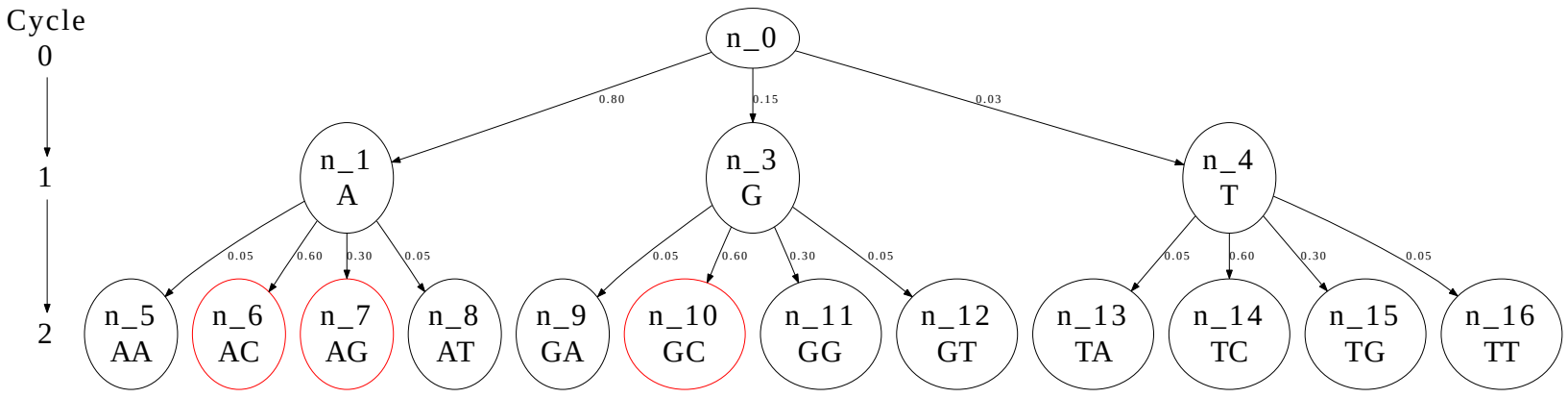
Example

Branch and Bound: b = 3 possible solutions are considered.



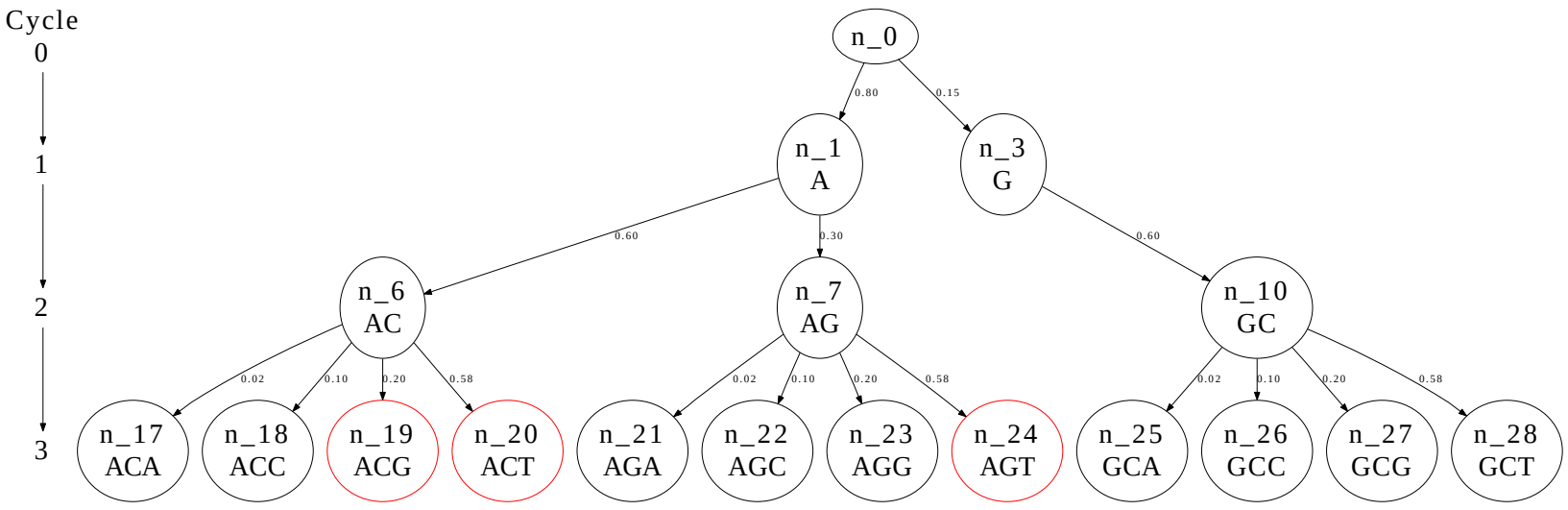
Basecalling with alingment

Example



Basecalling with alingment

Example



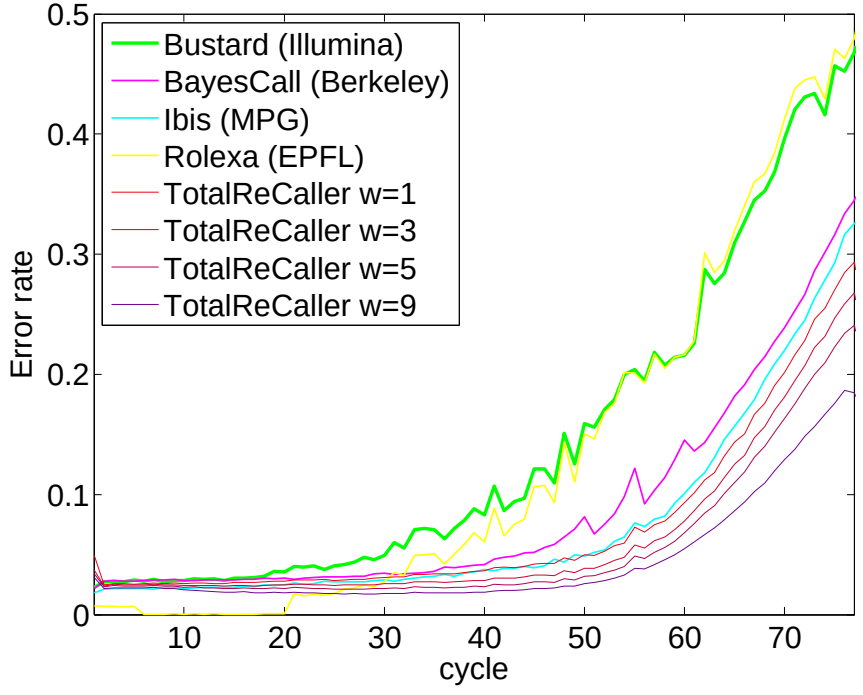
Basecalling with alingment

Score function

$$
\begin{aligned}
P_k(B \mid \boldsymbol{X}_k) &= \frac{P_k(\boldsymbol{X}_k \mid B)P_k(B)}{P_k(\boldsymbol{X}_k)} \qquad \text{with } B \in \{A, C, G, T\} \\
&= \frac{P_k(\boldsymbol{X}_k \mid B)P_k(B)}{P_k(\boldsymbol{X}_k \mid B)P_k(B) + P_k(\boldsymbol{X}_k \mid \neg B)P_k(\neg B)} \\
&= \frac{1}{1 + \frac{P_k(\boldsymbol{X}_k \mid \neg B)P_k(\neg B)}{P_k(\boldsymbol{X}_k \mid B)P_k(B)}} \\
&= \frac{1}{1 + \underbrace{\frac{P_k(\boldsymbol{X}_k \mid \neg B)}{P_k(\boldsymbol{X}_k \mid B)}}_{\text{Intensities}} \cdot \underbrace{\frac{P_k(\neg B)}{P_k(B)}}_{\text{Sequence alignment}}}
\end{aligned}
$$

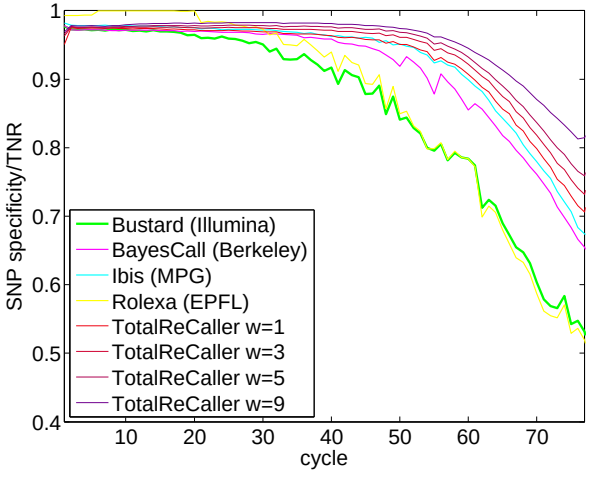
Overview

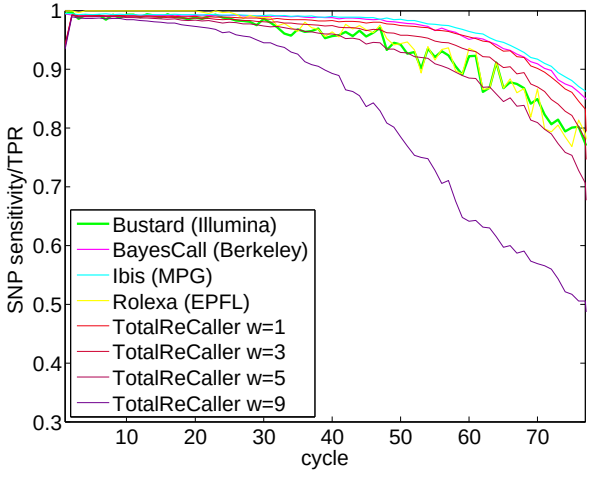
Phi-X


Bustard (Illumina)
BayesCall (Berkeley)
Ibis (MPG)
Rolexa (EPFL)
TotalReCaller w=1
TotalReCaller w=3
TotalReCaller w=5
TotalReCaller w=9
Error rate
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50
60
70
cycle

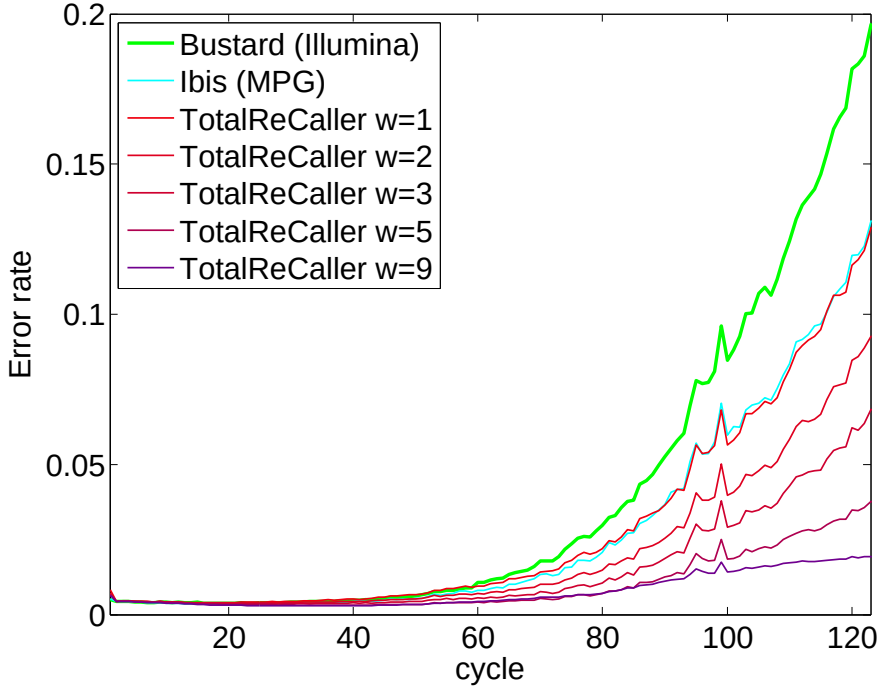
Results

Phi-X

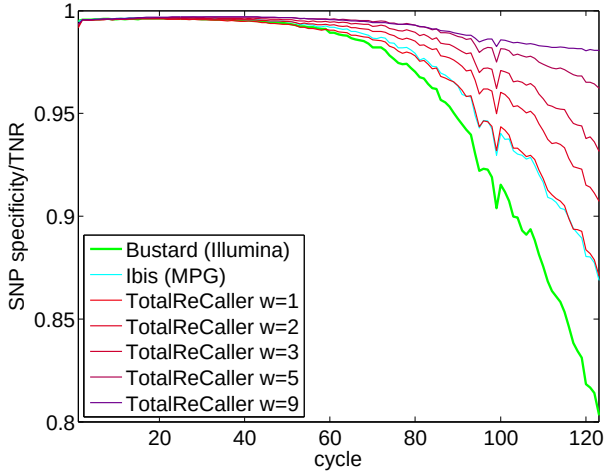


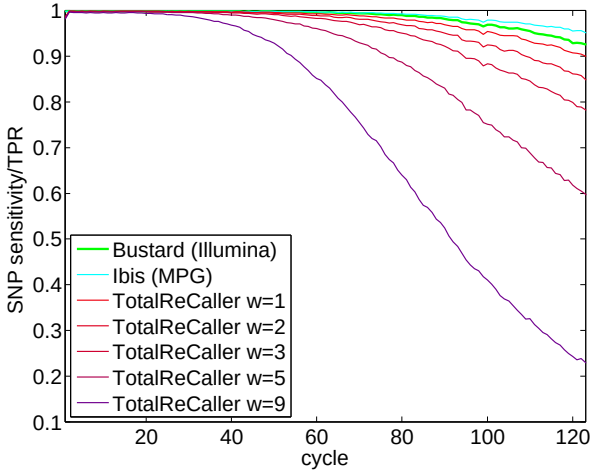


E.Coli


Bustard (Illumina)
Ibis (MPG)
TotalReCaller w=1
TotalReCaller w=2
TotalReCaller w=3
TotalReCaller w=5
TotalReCaller w=9
Error rate
0.2
0.15
0.1
0.05
0
20
40
60
80
100
120
cycle

E.Coli


SNP specificity/TNR
cycle
Bustard (Illumina)
Ibis (MPG)
TotalReCaller w=1
TotalReCaller w=2
TotalReCaller w=3
TotalReCaller w=5
TotalReCaller w=9


SNP sensitivity/TPR
cycle
Bustard (Illumina)
Ibis (MPG)
TotalReCaller w=1
TotalReCaller w=2
TotalReCaller w=3
TotalReCaller w=5
TotalReCaller w=9

Literatur

M. Burrows and D.J. Wheeler.
A block-sorting lossless data compression algorithm, 1994.

P. Ferragina and G. Manzini.
Opportunistic data structures with applications.
ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE, 41:390–398, 2000.

M. Kircher, U. Stenzel, and J. Kelso.
Improved base calling for the Illumina Genome Analyzer using machine learning strategies.
Genome Biology, 10(8):R83, 2009.

W.C. Kao, K. Stevens, and Y.S. Song.
BayesCall: A model-based base-calling algorithm for high-throughput short-read sequencing.
Genome Research, 19(10):1884, 2009.

Michael L Metzker.
Emerging technologies in DNA sequencing.
Genome research, 15(12):1767–76, 2005.