Fast mapping and precise alignment of AB SOLiD color reads to reference DNA

Miklós Csűrös[1], Szilveszter Juhos[2], and Attila Bérces[2]

[1] Department of Computer Science and Operations Research, University of Montréal, Canada. csuros@iro.umontreal.ca.
[2] Omixon www.omixon.com, Chemistry Logic Kft, Budapest, Hungary.

Abstract. Applied Biosystems' SOLiD system offers a low-cost alternative to the traditional Sanger method of DNA sequencing. We introduce two main algorithms of mapping SOLiD's color reads onto a reference genome. The first method performs mapping by adapting a greedy alignment framework. In such an alignment, reads are mapped to approximate genome positions, allowing for a pre-specified bound on sequence difference that combines nucleotide mismatches, gaps, and sequencing errors. The second method for precise alignment relies on a pair hidden Markov model framework, combining a DNA sequence evolution model and sequencing errors (from read quality files).

1 Introduction

Next-generation sequencing (NGS) methods [1] provide economical alternatives to the traditional Sanger method of DNA sequencing. Various commercially available platforms can generate large amounts of information which enable important biological and medical applications [2], including, perhaps most notably, the sequencing of personal and somatic genomes [3, 4], or even entire ecosystems [5]. In a typical genome analysis pipeline, NGS reads are mapped to reference sequences, and the alignments are further examined to detect variations within the target DNA sample, and with respect to the reference.

Currently available software for large-scale NGS mapping [6] use indexing techniques in order to speed up the search for similarities. The underlying algorithms rely either on hashtable-based indexes (*seed-and-extend*), or on compressed indexes exploiting the Burrows-Wheeler Transformation (BWT). BWT-based methods use little memory, and have an impressive computing speed [7, 8]. Seed-and-extend has an increasing advantage with higher sequence divergences, due to flexible tailoring choices for seeding methods [9].

The AB SOLiD sequencing platform from Applied Biosystems, Inc. (Foster City, Cal.) poses even greater challenges for bioinformatics than other widely used NGS technologies, due to the sheer size of the produced data (up to about a billion 35bp or 50bp reads in one production run), and the employed dinucleotide encoding by "colors." We introduce algorithmic solutions to different problems encountered when mapping AB SOLiD reads to a reference genome. First, we

propose a seed-and-extend framework for mapping color reads to locations along a reference DNA. The novelty of the framework is a greedy extension procedure employed in filtering the hits, which combines sequencing errors and DNA sequence differences. The seeding and the extension use the same "phase" representation of the color sequences, in order to minimize the number of executed arithmetic operations. The mappings are immediately useful for inferring structural variations [10] or phylogenetic classifications [11] (when multiple reference genomes are considered). Our second algorithmic solution addresses fine-scale alignments in a statistical framework. A notable feature of the approach is that color read quality values (sequencing error probabilities) are incorporated into a pair hidden Markov model. The statistical framework helps inferring the alignment with maximum expected accuracy or alignment metric accuracy (AMAP). The model assigns posterior probabilities to all target sequence variations, which can be used directly to deduce the consensus between overlapping reads without a multiple alignment.

2 Methods

2.1 Sequences and numerical encoding

The AB SOLiD system relies on the ligase-driven synthesis of PCR-amplified target DNA fragments. The sequencing read is produced in "color" encoding, where colors correspond to the dinucleotides sampled by fluorescently labeled probes in iterated synthesis cycles, arranged in their physical order along the target fragment. In the rest of the paper, we use a convenient numerical encoding for nucleotides and colors (or fluorescent dies):

$$\mathtt{A} = 0, \mathtt{C} = 1, \mathtt{G} = 2, \mathtt{T} = 3$$
$$\text{FAM/blue} = 0, \text{Cy3/green} = 1, \text{TXR/orange} = 2, \text{Cy5/red} = 3.$$

With this encoding, the mapping between colors and dinucleotides is simply the bitwise exclusive OR operation, denoted by $\oplus$: dinucleotide xy is encoded by the color $c = x \oplus y$.

The error-free color encoding for a DNA sequence $\mathbf{t} = t_{0..m}$ is the sequence $\mathbf{s} = c_{1..m}$ where $c_i = t_i \oplus t_{i-1}$. Notice that the same $\mathbf{c}$ translates into four possible $\mathbf{t}$ determined by t_0. The *read alignment* problem is that of aligning an unknown *target sequence* $\mathbf{t}$ to a known reference DNA sequence $\mathbf{s} = s_{1..n}$, using a color sequence $\mathbf{c}$ that encodes $\mathbf{t}$ but may contain sequencing errors. The alignment is evaluated with respect to the implied nucleotide mismatches and gaps, as well as the implied sequencing errors. Figure 1 illustrates this concept. An alignment is composed of column types M1–M4, D and I1–I2, where each column contains three cells: a reference cell s, a color cell c and a target cell t. For all three, $s, c, t \in \{0, 1, 2, 3, \square\}$, where $\square$ is the indel character. Concatenated non-indel characters in the color cells give the complete sequence $c_{1..m}$, and those in the reference cells yield a reference region $s_{i..i'}$. Indel characters may not occupy all three cells, and indels appear together in the color and target cells.

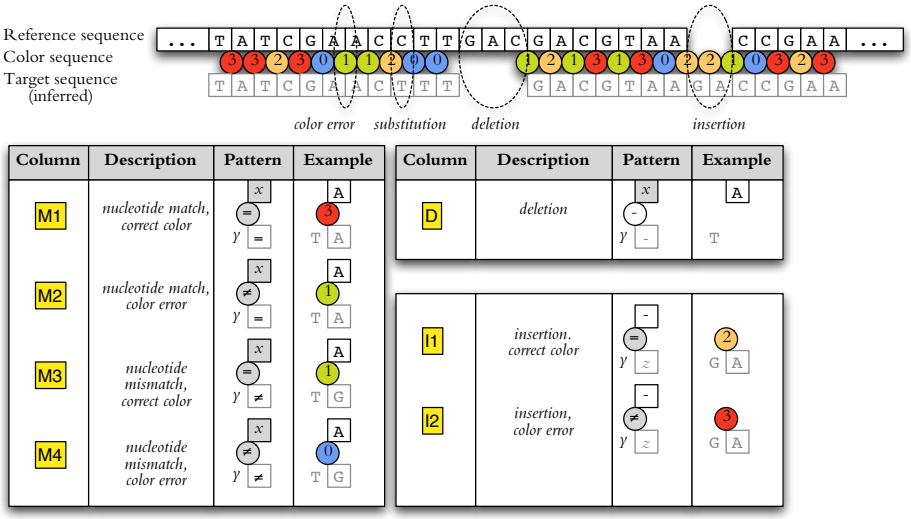


Fig. 1. Alignment between reference DNA, color sequence, and target sequence. The tables on the **bottom** enumerate possible alignment columns. Every column is annotated by the preceding nucleotide y in the target, defining sequencing errors.

Here we consider the simplest alignment scoring system, called the *edit distance*, which is computed by penalizing columns of type M2, M3, D and I1 with 1, and columns of type M4 and I2 with 2. Columns of type M1 are not penalized. The classic Smith-Waterman-Gotoh alignment [12, 13] is readily adaptable to find an optimal alignment [14, 15]. In order to track sequencing errors, it is necessary to include dinucleotide information in the formulas. Formally, there is a color error in a non-D column that is not the leftmost such column, if $t \oplus t' \neq c_j$ where t' is in the target cell, c_j is in the color cell, and t is the target cell content in the closest preceding non-D column.

The following lemma (proof omitted) shows that there is an optimal alignment that contains no columns with penalty 2.

Lemma 1. *There is an alignment with minimum edit distance that contains neither* M4 *nor* I2 *columns.*

In a run of ℓ consecutive perfect matches (M1) between $s_{i..i+\ell-1}$ and $c_{j..j+\ell-1}$,

$$c_j = y \oplus s_i \tag{1a}$$

$$c_{j+k} = s_{i+k-1} \oplus s_{i+k} \qquad \{1 \leq k < \ell\}, \tag{1b}$$

where y denotes the last aligned target nucleotide preceding the run. For convenience, we introduce the *phase representation* $\phi_{0..m}$ of the color sequence: $\phi_0 = 0$, and $\phi_k = c_1 \oplus c_2 \oplus c_3 \oplus \cdots \oplus c_k = \phi_{k-1} \oplus c_k$ for $k > 0$. From (1),

$$s_{i+k} = y \oplus c_j \oplus c_{j+1} \oplus \cdots \oplus c_{j+k} = y \oplus \phi_{j-1} \oplus \phi_{j+k} \tag{2}$$

for all $k = 0, \ldots, \ell - 1$. In other words, there exists an u (in particular, $u = y \oplus \phi_{j-1}$) with which $s_{i+k} = u \oplus \phi_{j+k}$ holds for all $k < \ell$. The phase representation $\phi_{0..m} = t_{0..m}$ is thus the translation of the color read into DNA, assuming that the target sequence starts with $t_0 = \phi_0 = 0$ (A).

2.2 Color read indexing

In a *seed-and-extend* framework [9], local alignments between two DNA sequences R, T are found by using a seeding function $h\colon \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^\ell \mapsto \mathcal{H}$, which filters the (i, j) position pairs where local alignments are worth being looked for. Specifically, an index table is built for R which gives the set of positions $h_R^{-1}(x) = \{i\colon h(R_{i..i+\ell-1}) = x\}$ for all $x \in \mathcal{H}$. A pair (i, j) is *hit* when $h(T_{j..j+\ell-1}) = h(R_{i..i+\ell-1})$, or $i \in h_R^{-1}(h(T_{j..j+\ell-1}))$. Hits are found by sliding a window along T and consulting the index table for $h(T_{j..j+\ell-1})$ in each position j. Hits are *extended* by performing a local alignment in a region around (i, j).

In the simplest case, h is the identity function, and hits correspond to matching ℓ-mers. Other widely used seeding functions rely on so-called *spaced seeds*. An (ℓ, w) spaced seed is defined by a set $\{\delta_1, \delta_2, \ldots, \delta_w\} \subseteq \{1, 2, \ldots, \ell\}$ of sampled positions, corresponding to the seeding function $h(x_{1..\ell}) = x_{\delta_1} \cdots x_{\delta_w}$. Accordingly, (i, j) pairs are hit when $R_{i+\delta_k-1} = T_{j+\delta_k-1}$ for all $k = 1, \ldots, w$. Spaced seeds perform theoretically and practically better [9] than ℓ-mers as seeding functions.

Seeding is not straightforward with color reads, because $\mathbf{s}$ and $\mathbf{c}$ do not encode DNA in the same way. Equation (2) suggests a possible way of adapting spaced seeds to indexing color reads. For a hit, $s_{i+\delta_k-1} = t_{j+\delta_k-1}$ holds in all sample positions $k = 1, \ldots, w$. Assuming no sequencing errors in $c_{j..j+\ell-1}$, Eq. (2) implies that $s_{i+\delta_1-1} \oplus s_{i+\delta_k-1} = \phi_{j+\delta_1-1} \oplus \phi_{j+\delta_k-1}$ for all $k = 2, \ldots, w$. Consequently, the hits can be found by indexing the reads in the phase representation: the seeding function is $h(x_{1..\ell}) = y_{1..w-1}$ with $y_k = x_{\delta_1} \oplus x_{\delta_{k+1}}$. For a corresponding hit, $h(s_{i..i+\ell-1}) = h(\phi_{j..j+\ell-1})$. (Existing tools like [14] translate instead the reference sequence into color space, so that for an (i, j)-hit, $s_{i+\delta_k-2} \oplus s_{i+\delta_k-1} = c_{j+\delta_k-1} = \phi_{j+\delta_k-2} \oplus \phi_{j+\delta_k-1}$ at all k, which corresponds to a seeding function $h(x_{0..\ell}) = y_{1..w}$ with $y_k = x_{\delta_k-1} \oplus x_{\delta_k}$ in our notation.)

2.3 Greedy alignment between color read and DNA sequence

Hits are extended by adapting the classic greedy procedure of Wu et al. [16]. An (i, j) hit between the reference DNA $s_{1..n}$ and color read $c_{1..m}$ is extended by computing the longest prefix of the color sequence that can be aligned starting at reference position $(i - j + 1)$ within prespecified bounds on the edit distance. Specifically, the procedure uses an argument $d_{\max}$ bounding the number of allowed indels between the reference and the inferred target sequence, and an argument $e_{\max}$ that bounds the edit distance. The procedure is explained best in terms of the *edit graph*. The edit graph's vertices are $\{(i, j, t)\colon 0 \le i \le n; 0 \le j \le m; 0 \le t \le 3\}$. The edges are weighted, and correspond to alignment

columns of Fig. 1. By Lemma 1, it suffices to consider column types M1–M3, D and I1. The t component of the vertex triple contains phase information on the color sequence. An edge of type M1 has weight 0, and by (2), connects (i, j, t) to $(i + 1, j + 1, t)$ where $t = s_{i+1} \oplus \phi_{j+1}$. All other edge types have weight 1: M2 $(i, j, t) \to (i+1, j+1, s_{i+1} \oplus \phi_{j+1})$ with $t \neq s_{i+1} \oplus \phi_{j+1}$, M3 $(i, j, t) \to (i+1, j+1, t)$ with $t \neq s_{i+1} \oplus \phi_{j+1}$, I1 $(i, j, t) \to (i, j + 1, t)$, and D $(i, j, t) \to (i + 1, j, t)$.

A path in the edit graph corresponds to an alignment. Given a bound $e_{\max}$, we restrict our attention to paths from any of the $(i, 0, t)$ vertices reach some (i', j, t') with maximum $j \leq m$, and have at most $e_{\max}$ non-M1 edges. In other words, we are searching for the longest alignable prefix within the bound. Define the *diagonal* $d = 0, 1, \ldots, n$ as the vertex set $\{(i, i - d, t)\}$. Our greedy algorithm considers paths along diagonals $0, \ldots, 2d_{\max}$ only. Let $R_t^d(e) = j$ if $(j + d, j, t)$ is the farthest reachable vertex from any $(i, i - d, t')$ on a path with vertices on diagonals $d \leq 2d_{\max}$, and with edge weight sum at most $e \leq e_{\max}$. Algorithm GREEDY computes all $R_t^d(e)$.

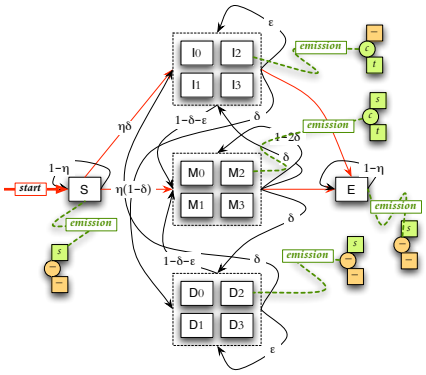
```
Algorithm GREEDY(s_{1..n}, φ_{0..m}, d_max)
Output: longest prefix of φ alignable within e_max errors on diagonals 0, ... 2d_max.
G1  for t ← 0,...,3 and d = 0,...,2d_max do R_t^d(0) ← 0; ∀e > 0: R_t^d(e) ← −∞
G2  for e ← 0,...,e_max do
G3    for t ← 0,...,3 and d = 0,...,2d_max do
G4      j ← R_t^d(e); i ← j + d
G5      if j ≠ −∞ then
G6        while i + 1 < n and j + 1 < m and s_{i+1} ⊕ φ_{j+1} = t do
G7          i ← i + 1; j ← j + 1                          ▷ run of M1 edges
G8        if j ≥ m − (e_max − e) then return m else R_t^d(e) ← j
G9    if e ≠ e_max then
G10     for t ← 0,...,3 and d = 0,...,2d_max do
G11       j ← R_t^d(e); i ← j + d
G12       if j ≠ −∞ then
G13         if d ≠ 2d_max then UPDATE(d + 1, t, e + 1, j)        ▷ D edge
G14         if d ≠ 0 then UPDATE(d − 1, t, e + 1, j + 1)         ▷ I1 edge
G15         if i < n then
G16           UPDATE(d, t, e + 1, j + 1)                       ▷ M3 edge
G17           UPDATE(d, s_{i+1} ⊕ φ_{j+1}, e + 1, j + 1)       ▷ M2 edge
G18 return max_{t=0,...,3;d=0,...,2d_max} {R_t^d(e_max)}

Algorithm UPDATE(d, t, e, j)
U1  if R_t^d(e) < j then R_t^d(e) ← j
```

When extending a hit at (i, j) for the reference sequence $\mathbf{s}$ and the phase sequence $\phi_{0..m}$, Algorithm GREEDY$(s_{i'..i'+m+2d_{\max}-1}, \phi_{0..m}, d_{\max})$ is called, where $i' = i - j + 1 - d_{\max}$ is the starting position of the region within which the extension is performed. By an analogous argument to [16], the running time is $O(m + d_{\max} e_{\max})$ on average (for random sequences), and $O(m d_{\max})$ in the worst case. The greedy framework can be adapted to slightly more general scoring systems (match/mismatch penalties), but it is unclear whether it could accommodate symbol-dependent scoring and affine gap penalization [17]. Therefore, GREEDY is more useful for filtering hits than for retrieving optimal alignments.

2.4 Statistical alignment for color reads

We perform statistical alignment by using a pair hidden Markov model [18], or pair-HMM. A pair-HMM defines a probability distribution over alignments. The advantages of having a well-defined probabilistic model are manifold [19]. Likelihoods can be used to recognize unrelated sequence pairs, or to optimize model parameters. Posterior probabilities quantify discrepancies between the two sequences in a statistically principled manner.



Transition	Transition prob.	Emission	Emission prob.
$\mathsf{S} \to \mathsf{S}$	$1-\eta$	$(s, \square)$	π_s
$\mathsf{S} \to (\mathsf{M}, t)$	$\eta(1-\delta)\pi_t$	$(\square, \square)$	1
$\mathsf{S} \to (\mathsf{I}, t)$	$\eta\delta\pi_t$	$(\square, \square)$	1
$(\mathsf{M}, t) \to (\mathsf{M}, t')$	$(1-2\delta)\pi_{t'}$	$(s, t \oplus t')$	$p(t' \to s)$
$(\mathsf{M}, t) \to (\mathsf{D}, t)$	δ	$(s, \square)$	π_s
$(\mathsf{M}, t) \to (\mathsf{I}, t')$	$\delta\pi_{t'}$	$(\square, t \oplus t')$	1
$(\mathsf{M}, t) \to \mathsf{E}$	1[after m colors]	$(\square, \square)$	1
$(\mathsf{D}, t) \to (\mathsf{M}, t')$	$(1-\epsilon-\delta)\pi_{t'}$	$(s, t \oplus t')$	$p(t' \to s)$
$(\mathsf{D}, t) \to (\mathsf{D}, t)$	ϵ	$(s, \square)$	π_s
$(\mathsf{D}, t) \to (\mathsf{I}, t')$	$\delta\pi_{t'}$	$(\square, t \oplus t')$	1
$(\mathsf{I}, t) \to (\mathsf{M}, t')$	$(1-\delta-\epsilon)\pi_{t'}$	$(s, t \oplus t')$	$p(t' \to s)$
$(\mathsf{I}, t) \to (\mathsf{D}, t)$	δ	$(s, \square)$	π_s
$(\mathsf{I}, t) \to (\mathsf{I}, t')$	$\epsilon\pi_{t'}$	$(\square, t \oplus t')$	1
$(\mathsf{I}, t) \to \mathsf{E}$	1[after m colors]	$(\square, \square)$	1
$\mathsf{E} \to \mathsf{E}$	$1-\eta$	$(s, \square)$	π_s

Fig. 2. Pair HMM for alignment of color reads and the reference DNA. Only the correct colors are shown in the **Emission** column, i.e., the error probability is 0 in this table.

For the alignment of color reads to a reference DNA, we introduce a pair-HMM with state set $\mathcal{Q} = \{\mathsf{S}, \mathsf{E}\} \cup \big(\{\mathsf{M}, \mathsf{I}, \mathsf{D}\} \times \{0, 1, 2, 3\}\big)$. The HMM generates a state sequence $q_0, \ldots, q_\ell \in \mathcal{Q}^\ell$ as a random Markov chain determined by transition probabilities between the states. A transition is followed by the random emission of a pair $w = (s, c)$ where $s \in \{0, 1, 2, 3, \square\}$ is a numerically encoded nucleotide and $c \in \{0, 1, 2, 3, \square\}$ is a numerically encoded color. A *run* of the hidden Markov model [20] consists of a random state sequence $q_0, \ldots, q_\ell$ coupled with the random emitted pairs $w_1, \ldots, w_\ell$. States S and E emit unaligned prefixes and suffixes of the reference sequence. States (M, t), (D, t), (I, t) encode the rightmost inferred target nucleotide t, and correspond to match, deletion, and insertion. A transition from (x, t) to (x', t') with $x' \in \{\mathsf{M}, \mathsf{I}\}$ entails the emission of a color character c: the color is correct if $t \oplus t' = c$. The SOLiD sequencing system provides error estimates in so-called quality files that encode the error probability ν on an integer scale using a formula originally introduced for Sanger sequencing in the phred program [21]: $\mathsf{qual} = \lfloor -10 \cdot \log_{10} \nu \rfloor$. We thus assume that a sequence of error probabilities $\nu_{1..m}$ is available with the color sequence $c_{1..m}$. Subsequently to a state transition $(x, t) \to (x', t')$, the emission of the color character c_j occurs with probability $\gamma_j(t \oplus t')$, where

$$\gamma_j(c_j) = 1 - \nu_j \quad \text{and} \quad c \neq c_j \colon \gamma_j(c) = \nu_j / 3. \tag{3}$$

The emission of reference nucleotides is dictated by an assumed Markov model of DNA sequence evolution [22], like the F84 model [23]. In general, we assume that the nucleotide substitutions between reference and target happen according to a Markov model that specifies the stationary distribution π and the substitution probabilities $p(s \to t)$, and that the model is reversible ($\pi_s p(s \to t) = \pi_t p(t \to s)$). Transitions to states (x, t) with different $t \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ thus happen by probabilities proportional to π_t. The emission of a reference nucleotide $s \neq \square$ occurs with probability $p(t \to s)$ on arrival to state (M, t).

Transition probabilities determine the expected lengths of unaligned prefixes and suffixes, as well as the frequency and length of gaps. In particular, we assume that the prefix and suffix regions have a geometric prior length distribution with mean $1/\eta$, that insertions and deletions start with a probability δ, and that gaps have a geometric prior length distribution with mean $1/(1-\epsilon)$. When aligning a color sequence of length m, we are interested in state sequences with exactly m states emitting color characters (M and I). For that reason, we impose the non-emitting state transition $M \to E$ and $I \to E$ after emitting m color characters. The transition out of state S to (M, t) or (I, t), which sets the first target nucleotide $t_0 = t$, is also non-emitting. Figure 2 summarizes the state transitions and the emissions.

2.5 Likelihood and posterior probabilities

A run of the pair-HMM in Fig. 2 produces an alignment, but the indels cannot be observed, only the produced sequences. Given a reference $s_{1..n}$ and a color sequence $c_{1..m}$, we can compute the likelihood that such a pair is generated by the model, while admitting color errors by known probabilities $\nu_{1..m}$. In order to compute the likelihood (and various posterior probabilities later), we use *forward* and *backward* probabilities [18, 20]. The forward probabilities are denoted by $S[i] = S[i,0], E[i] = E[i,m], M_t[i,j], I_t[i,j], D_t[i,j]$ with $i = 0, \ldots, n$ and $j = 0, \ldots m$. The quantity $q[i,j]$ denotes the probability that the pair-HMM generates the prefixes $s_{1..i}$ and $c_{1..j}$ in a run that ends with state q. Forward probabilities can be computed in a recursive manner, as shown in Table 1

The backward probabilities $S'[i] = S'[i,0], E'[i] = E'[i,m], M'_t[i,j], I'_t[i,j], D'_t[i,j]$ capture a symmetric concept. The quantity $q'[i,j]$ is the probability that the pair-HMM produces the suffixes $s_{i+1..n}$ and $c_{j+1..n}$ in a run starting with state q. The backward probabilities are calculated by analogous recursions to those in Table 1.

Now, given the color error probabilities $\nu_{1..m}$, the likelihood for the observed sequences is $L(s_{1..n}, c_{1..m}) = E[n] = S'[0]$. The forward and backward probabilities are combined to calculate posterior probabilities for visiting various states. The posteriors are $\psi(q)[i,j] = \frac{q[i,j] \cdot q'[i,j]}{L(s_{1..n}, c_{1..m})}$ for $q = M_t, I_t, D_t$ and $\psi(q)[i] = \frac{q[i] \cdot q'[i]}{L(s_{1..n}, c_{1..m})}$ for $q = S, E$. The posterior probabilities can be used to assign confidence to a triple alignment column. A state transition to $q = (\mathsf{M}, t)$, followed by the emission of (s, c) corresponds to an alignment column (s, c, t) of type $\mathsf{M1}$–$\mathsf{M4}$. Hence, $p(i \diamond j, t) = \psi(M_t)[i,j]$ is the probability that such a column aligning $s = s_i$ and $c = c_j$ is correct. The probability that s_i is

Table 1. Recursions for forward probabilities.

$$
\begin{aligned}
S[0] &= 1; \qquad E[0] = 0 \\
S[i] &= \pi_{s_i} \cdot (1-\eta) \cdot S[i-1] && \{i > 0\} \\
M_t[i,0] &= \pi_t \cdot \eta(1-\delta) \cdot S[i]; \quad I_t[i,0] = \pi_t \cdot \eta\delta \cdot S[i]; \quad D_t[i,0] = 0 && \{i \geq 0\} \\
M_t[i,j] &= \pi_t p(t \to s_i) \sum_{t'} \Big(\gamma_j(t' \oplus t) && \{i,j > 0\} \\
&\qquad \times \ \Big((1-2\delta) \cdot M_{t'}[i-1,j-1] \\
&\qquad\quad +(1-\delta-\epsilon) \cdot \big(I_{t'}[i-1,j-1] + D_{t'}[i-1,j-1] \big) \Big) \Big) \\
I_t[i,j] &= \pi_t \sum_{t'} \gamma_j(t' \oplus t) \ \Big(\epsilon \cdot I_{t'}[i,j-1] && \{i \geq 0, j > 0\} \\
&\qquad\qquad +\delta \cdot \big(M_{t'}[i,j-1] + D_{t'}[i,j-1] \big) \Big) \\
D_t[i,j] &= \pi_{s_i} \Big(\epsilon \cdot D_t[i-1,j] + \delta \cdot \big(M_t[i-1,j] + I_t[i-1,j] \big) \Big) && \{i,j > 0\} \\
E[i] &= \pi_{s_i}(1-\eta) \cdot E[i-1] + \sum_t \Big(M_t[i,m] + I_t[i,m] \Big) && \{i > 0\}
\end{aligned}
$$

deleted in the target sequence is $p(i \diamond \cdot) = \sum_{j,t} \psi(D_t)[i,j] = 1 - \sum_{j,t} p_t(i \diamond j, t)$. The probability that a column of type $\mathsf{I1}$ or $\mathsf{I2}$ containing $(\square, c_j, t)$ should appear in the alignment is $p(\cdot \diamond j, t) = \sum_i \psi(I_t)[i,j]$. Finally, the probability that reference nucleotide i is part of the skipped prefix or suffix is $\alpha(i) = \psi(S)[i] + \psi(E)[i] - \sum_t \big(\psi(M_t)[i,m] + \psi(I_t)[i,m] \big)$, where the non-emitting transitions into E are taken into account.

With the posterior probabilities at hand, we can find the so-called *AMAP alignment* that maximizes metric accuracy [24]. Consider an alignment with ℓ columns $\big((s_k, c_k, t_k) \colon k = 1, \ldots, \ell \big)$ Let $T(k)$ be the type of column k, and let $s_k^{\#}$, $c_k^{\#}$ denote the number of non-indel reference and color characters emitted in columns $1, \ldots, k$. Using a gap-factor $G \in [0,1]$, the alignment maximizes the score $(1-G) \cdot \sum_{k \colon T(k) \in \mathcal{M}} p(s_k^{\#} \diamond c_k^{\#}, t_k) + G \cdot \Big(\sum_{k \colon T(k) = \mathsf{D}} p(s_k^{\#} \diamond \cdot) + \sum_{k \colon T(k) \in \mathcal{I}} p(\cdot \diamond c_k^{\#}, t_k) + \sum_{k \colon T(k) \in \mathcal{S}} \alpha(s_k^{\#}) \Big)$, where $\mathcal{M} = \{\mathsf{M1}, \mathsf{M2}, \mathsf{M3}, \mathsf{M4}\}$, $\mathcal{I} = \{\mathsf{I1}, \mathsf{I2}\}$ and $\mathcal{S} = \{\mathsf{S}, \mathsf{E}\}$. The gap-factor sets a tradeoff between specificity and sensitivity: $G = 0$ corresponds to the alignment with maximum expected accuracy [18], and $G = 1/3$ provides a neutral setting. Computing the AMAP alignment is straightforward by dynamic programming after the posterior probabilities are calculated.

Small-scale variations such as nucleotide substitutions and short gaps can be readily identified with statistical confidence. The probability that s_i is aligned with a target nucleotide $t \in \{0,1,2,3\}$ is $p(i \sim t) = \sum_j p(i \diamond j, t)$. The proba-

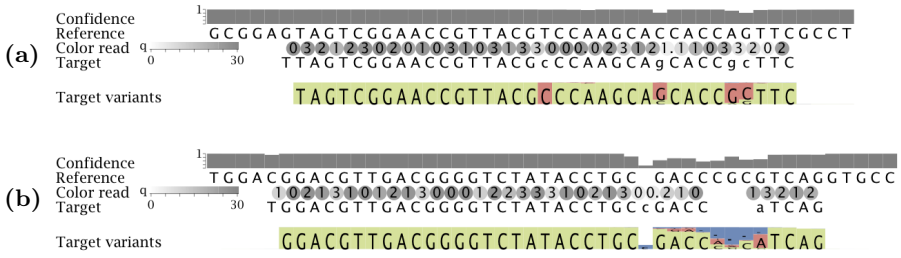


Fig. 3. AMAP alignments and sequence variations. "Confidence" is the probability of the column being correct. Shading indicates the quality values along the color sequence; a dot '.' denotes a color error. Sequence variants are shown by the logos. The height of each logo box is proportional to the probability $1 - \alpha(i)$ that the nucleotide is covered by the alignment; posterior probabilities for homology statements are shown by the relative symbol height. **(a)** Mismatches with different credibility. **(b)** Homology statements may be stronger than alignment confidence (see `GACC` before the deletion).

bility that the reference nucleotide s_i is aligned with a gap is $p(i \diamond \cdot)$. Figure 3 illustrates AMAP alignments and sequence variants. The probabilities of the homology statements can be combined across different reads that align to the same reference region, in order to infer sequence variations in the target DNA.

3 Experiments

We implemented the algorithms in a Java software package called Crema, and used it on sequencing reads for *Escherichia coli* DH10B. The reads (35bp long reads, no mate pairs) were downloaded from the Applied Biosystems website (`http://download.solidsoftwaretools.com/frag/R1a007_20080307_2_EG017_F3.csfasta.zip`), with the accompanying quality file. We selected 1 million reads randomly, and mapped them against the genome of *Shigella flexneri* 2a str. 301 (Genbank accession number NC_004337.1). In the experiments, we compared our implementation with Bowtie [8] version 0.12.5, and SHRiMP [14] version 1.3.2. All programs were tested on an ordinary Linux machine (Amazon Elastic Compute Cloud, Standard Instance).

Read mapping. Table 2 shows the mapping results. In the greedy extension, we mapped the reads by retaining hits where all 35 positions be aligned within an edit distance of $e_{\max} = 6$, along a band of ± 3 diagonals. At comparable sensitivities, the greedy extension (with a platform-independent implementation) is faster than Bowtie, or SHRiMP.

Read alignment. We computed the alignments for uniquely mapped reads by first optimizing the pair-HMM parameters using a random subset of 100 thousand reads. We employed the F84 model [23] of DNA sequence evolution, with

Table 2. Mapping DH10B sequencing reads to *S. flexneri*. Mappings with different seeds (numbers denote length and weight) are compared with other tools at parameter settings resulting in comparable sensitivities. "Unique" reads are mapped to a single locus with maximal alignment score.

Method	CPU time	Mapped reads	Unique
SHRiMP (`-M 35bp,fast`)	263 s	593785	561301
SHRiMP (`-M 35bp,sensitive`)	717 s	605348	572317
Bowtie (`--best`)	216 s	488137	
Crema (19,17)-seed + greedy	118 s	511263	492230
Crema (16,14)-seed + greedy	192 s	569865	546495
Crema (14,12)-seed + greedy	581 s	605938	576419

equal base frequencies (GC-content of *E. coli* is close to 50%), and a transition/transversion ratio of 2. The sequence divergence, and the gap open/extend probabilities were set in an Expectation-Maximization procedure by computing the expected numbers of substitutions and indels: convergence was achieved after four iterations with a divergence of 0.0145, gap open probability $\delta = 0.00025$ and gap extension probability $\epsilon = 0.5$. Instead of directly using the Phred formula for transforming quality scores into probabilities, we used our own mapping based on the expected number of color errors at different scores, as computed by the pair-HMM model.

Table 3. Alignments of DH10B sequencing reads with *S. flexneri*. "Validated" reads and nucleotides appear in BLAST alignments to the DH10B reference. "Incorrect" nucleotides differ from the DH10B genome sequence.

	Reads		All inferred nucleotides		Substitutions		Insertions	
	unique	validated	validated	incorrect	validated	incorrect	validated	incorrect
Bowtie (best)	465 024	463 785	15 274 713	13 862 (0.09%)	70 110	2 783 (4.0%)	*(does not infer indels)*	
SHRiMP (sensitive)	572 317	570 107	19 569 714	42 863 (0.22%)	184 254	28 364 (15.4%)	290	10 (3.4%)
Crema (AMAP alignment)	576 419	574 490	19 962 920	40 308 (0.20%)	249 107	27 162 (10.9%)	1 108	72 (6.5%)

In order to validate alignment results, we used `blastn` [25] to align the inferred target sequences to the assembled DH10B genome (Genbank accession number NC_010473.1), with default parameters and an E-value cutoff of 10^{-6}. BLAST found an alignment for 99.7–99.6% of the reads. The alignments (as reported in SAM [`http://samtools.sourceforge.net/`] format's CIGAR strings) of uniquely mapped reads were scanned to validate the inferred target nucleotides. Table 3 shows the results. Bowtie, designed to map human sequence variants, captures only very similar sequences, with an overall error rate of 0.09%. SHRiMP and Crema are much more sensitive, but have a similar 0.2% overall

error. Crema is, however, better than SHRiMP at finding actual sequence differences: about 35% more substitutions are predicted, with 30% fewer errors. The framework is especially useful in annotating the computed alignments. The posterior probabilities for the inferred nucleotides can be encoded in the QUAL field of the SAM format using the phred transformation [21]. Figure 4 illustrates that high-scoring positions have a much lower error level. For instance, inferred nucleotides with a quality score at least 20 (96% of positions) are wrong only 0.045% of the time. The plot also shows that quality values under 30 are predicted fairly accurately (Bowtie quality values are underestimated by more than 20 on the same interval — data not shown).

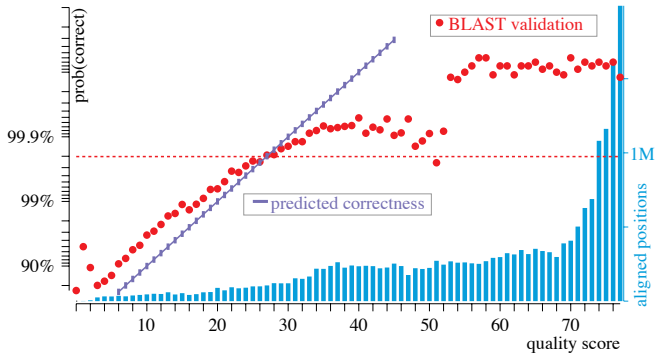


Fig. 4. Quality scores for inferred nucleotides and actual correctness ("BLAST validation") in validating BLAST hits. The horizontal dashed line shows the overall fraction of correctly inferred nucleotides. "Predicted correctness" uses the Phred formula with small bars denoting rounding errors. Vertical bars plot the frequency of quality scores with scaling shown on the right.

Conclusion

We presented a seed-and-extend framework for efficient color read mapping, and a statistical alignment framework for precise alignments. The experiments demonstrate that they offer valuable options in the comparative sequencing of bacterial genomes.

References

1. Shendure, J., Li, H.: Next-generation DNA sequencing. Nat. Biotechnol. **26**(10) (2008) 1135–1145
2. Shendure, J., Mitra, R.D., Varma, C., Church, G.M.: Advanced sequencing technologies: Methods and goals. Nat. Rev. Genet. **5** (2004) 335–344

3. Wheeler, D.A., et al.: The complete genome of an individual by massively parallel DNA sequencing. Nature **452** (2008) 872–876
4. Pleasance, E.D., et al.: A comprehensive catalogue of somatic mutations from a human cancer genome. Nature **463** (2010) 191–196
5. Venter, J.C., et al.: Environmental genome shotgun sequencing of the Sargasso Sea. Science **304** (2004) 66–74
6. Flicek, P., Birney, E.: Sense from sequence reads: methods for alignment and assembly. Nat. Methods **6**(11s) (2009) S6–S12
7. Li, H., Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics **25**(14) (2009) 1754–1760
8. Langmead, B., Trapnell, C., Pop, M., Salzberg, S.L.: Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. **10** (2009)
9. Brown, D.G., Li, M., Ma, B.: A tutorial of recent developments in the seeding of local alignment. J. Bioinform. Comput. Biol. **2**(4) (2004) 819–842
10. Medvedev, P., Stanciu, M., Brudno, M.: Computational methods for discovering structural variation with next-generation sequencing. Nat. Methods **6**(11s) (2009) S13–S20
11. Huson, D.H., Auch, A.F., Qi, J., Schuster, S.C.: MEGAN analysis of metagenomic data. Genome Res. **17** (2007) 377–386
12. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. **147** (1981) 195–197
13. Gotoh, O.: An improved algorithm for matching biological sequences. J. Mol. Biol. **162**(3) (1982) 705–708
14. Rumble, S.M., Lacroute, P., Dalca, A.V., Fiume, M., Sidow, A., Brudno, M.: SHRiMP: Accurate mapping of short color-space reads. PLoS Comput. Biol. **5**(5) (2009) e1000386
15. Homer, N., Merriman, B., Nelson, S.F.: Local alignment of two-base encoded DNA sequence. BMC Bioinformatics **10** (2009) 175
16. Wu, S., Manber, U., Myers, G., Miller, W.: An $O(NP)$ sequence comparison algorithm. Inform. Process. Lett. **35**(6) (1990) 317–323
17. Zhang, Z., Schwartz, S., Wagner, L., Miller, W.: A greedy alignment for aligning DNA sequences. J. Comput. Biol. **7**(1/2) (2000) 203–214
18. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, UK (1998)
19. Lunter, G., Drummond, A.J., Miklós, I., Hein, J.: Statistical alignment: Recent progress, new applications, and challenges. In Nielsen, R., ed.: Statistical Methods in Molecular Evolution. Springer-Verlag, Heidelberg (2005)
20. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proc. IEEE **77**(2) (1989) 257–286
21. Ewing, B., Green, P.: Base-calling of automated sequencer traces using *phred*: II. error probabilities. Genome Res. **8** (1998) 186–194
22. Liò, P., Goldman, N.: Models of molecular evolution and phylogeny. Genome Res. **8** (1998) 1233–1244
23. Felsenstein, J., Churchill, G.A.: A Hidden Markov Model approach to variation among sites in rate of evolution. Mol. Biol. Evol. **13**(1) (1996) 93–104
24. S. Schwartz, A., Pachter, L.: Multiple alignment by sequence annealing. Bioinformatics **23**(2) (2007) e24–29
25. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. **215**(3) (1990) 403–410