Genomics via Optical Mapping IV: Sequence Validation via Optical Map Matching*†

Marco Antoniotti † Thomas Anantharaman §†
Salvatore Paxia † Bud Mishra †
† Courant Bioinformatics Group, New York University
New York, NY, U.S.A.
§ Biotechnology Center, University of Wisconsin
Madison, WI, U.S.A.

NYU-CIMS-TR-811

September 27, 2000

Abstract

This paper describes the underlying mathematical model and the dynamic programming algorithm technology for the *validation* of a (DNA) *sequence* against a (DNA) *map*. The sequence can be obtained from a variety of sources (e.g. GenBank, Sanger's Lab, or Celera P.E.) and it is assumed to be written out as a string of nucleotides. The map is an *ordered restriction map* obtained through an *optical mapping* process and is augmented with statistical information which will be used to place (or not) the sequence in the genome.

Our approach has many other applications beyond validation: e.g. map-based sequence assembly, phasing sequence contigs, detecting and closing gaps and annotation of partially sequenced genomes to find open reading frames, genes, and synteny groups.

We tested our system by checking various maps against publicly available sequence data for *Plasmodium falciparum* (*cf.* [10, 11]).

1 Object Definitions and Processes Definitions

The objects we will be dealing with in the following are of three kinds: nucleotides *sequences*, ordered restriction *consensus maps* obtained mainly through a genome-wide shotgun optical mapping process, and *molecule maps* obtained via an "in silico" simulated digestion process.

We will define these objects in an rather abstract way, while keeping in mind various implementation issues. The description will refer to other sources for a mathematically sound treatment of the same objects.

1.1 Sequences

A *sequence* is simply a *string* of letters drawn from the set

$$\{A, C, G, T, N, X\}.$$

The letters have the standard meaning in the bioinformatics literature, A, C, G, T are DNA bases; N is "unknown", and X is a "gap".

*This research was conducted under the Department of Energy Grant DoE-25-74100-F1799 and under the National Cancer Institute Grant NCI 5 RO1 CA79063-03.

†We thank David C. Schwartz, University of Wisconsin, Madison, U.S.A. for his help and continous encouragement, and Alberto Policriti, Università di Udine, ITALY, for his useful comments.

1.2 Consensus (Optical) Map

A *consensus map* (optical) is genome-wide, ordered restriction map, which is represented as a structured item consisting of some *identification* data and a variable length vector of *fragments*.

The consensus map is represented as a vector of fragments, where each fragment is a triple of positive real numbers.

$$\langle c_i,\, l_i,\, \sigma_i \rangle \in R^3$$

where c_i is the *cut probability* associated with a Bernoulli trial, l_i is the *fragment size*, a random variable with Gaussian distribution with an estimated mean $\mu = l_i$, and estimated *standard deviation* σ_i[1]. We will indicate the total length of the fragment vector as N. Also we will actually index the vector of fragments from 0 to $N-1$.

1.3 Sequence Map

A *sequence map* is an "in silico" ordered restriction map obtained from a *sequence* by simulating a restriction enzyme digestion process. Hence each sequence map has some piece of *identification* data plus a vector of *fragments*, whose elements encode exactly the size in base-pairs.

The sequence map fragment vector j-th element is simply a number a_j which is the size of the fragment. We will indicate the total length of the sequence map fragment vector as M. Also we will index the fragment vector from 0 to $M-1$.

Note that our consensus maps are created from several long genomic single molecule maps, where each one is obtained from the images of molecules stretched on a surface and further combined by a *Bayesian* algorithm implemented in the gentig program. The objects described in Section 1.2 are used and/or constructed by the gentig program. gentig constructs consensus maps by considering the local variations among the aligned single molecule maps. For more details see [2, 3].

Finally it must be noted that the "in silico" digestion process is extremely simple and, to the best of our knowledge, does not introduce "errors" in the map it produces.

1.4 The Validation Process

The validation process *matches* a DNA sequence ordered restriction map (obtained via an "in silico" digestion) against a consensus map. The match is computed using a *Bayesian* cost function described in Section 2. Figure 1 summarizes the overall flow of the validation process.

2 Statistical Description of the Problem

Figure 2 shows a simple setup of the matching problem involving a sequence map and a consensus map. The sequence map is considered to be *correct* and it is viewed as the *hypothesis* $\mathcal{H}$ of a Bayesian problem to be analyzed, while the consensus map is considered to be a piece of *data* $\mathcal{D}$ to be *validated* against $\mathcal{H}$.

All in all this amounts to maximize the probability density function

$$\Pr(\mathcal{D},\, \ldots \mid \mathcal{H}(\hat{\sigma}, p_c, p_f)),$$

where $\hat{\sigma}$ is a standard deviation (which summarizes maps wide standard deviation data, i.e. $\hat{\sigma} = \mathrm{f}\sigma_i$ for some function 'f'), and the overall function depends on other parameters as well: p_c, the *cut probability*, and p_f, the *false positive cut probability*.

2.1 Ideal Case

We can formulate the problem in the ideal case where we have *known orientation* of the sequence map, *no false cuts*, and *no missing cuts*. I.e. $p_c = 1$, and $p_f = 0$ (in this case the terms associated with these parameters just vanish, as it will be shown later).

Let's now consider a position h in the consensus map and the consensus map fragment sub-vector from h to $N-1$. Let's also consider the full sequence map fragment vector from 0 to $M-1$. For the sake of simplicity, we

[1] There are really more pieces of information that are present in the consensus map data as stored in the system database.

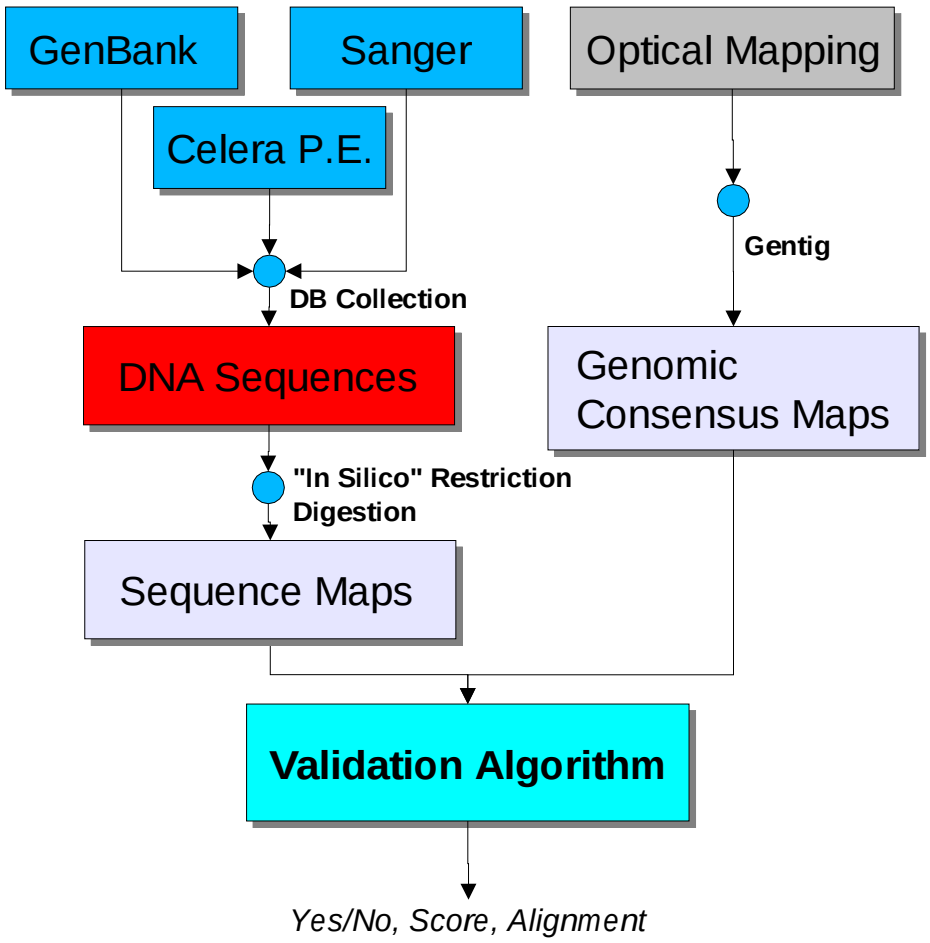

Figure 1: The validation process overall flow.

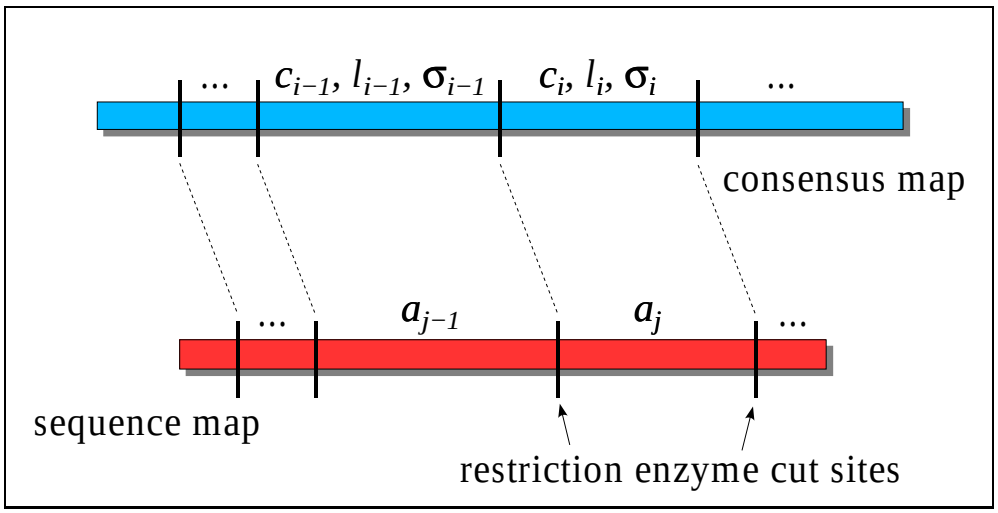

Figure 2: A simple matching of a sequence map against a consensus map. In this case, the sizes $\langle \ldots, l_i, \ldots \rangle$ from the consensus map, and the sizes $\langle \ldots, a_j, \ldots \rangle$ match. the remaining parameters are used to compute the "match" function, which actually *minimizes* a sum-of-squares expression.

drop the h term and count the consensus map fragments from 0 as well, so that we can write expressions like l_i instead of l_{h+i}.

Let's now consider the "match" between the i-th fragments of the consensus map and of the sequence map. We want to evaluate how much the hypothesis consensus map deviates from the "correct" sequence map. We assume a Gaussian distribution, therefore for the i-th fragment we will need to evaluate the following expression:

$$\frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(l_i - a_i)^2}{2\sigma_i^2}}.$$

Given this expression, and the assumption that the sequence map is correct[2] the overall $\Pr(\mathcal{D}, \ldots \mid \mathcal{H}(\hat{\sigma}, \ldots))$ function can be written as

$$\Pr(\mathcal{D}, \ldots \mid \mathcal{H}(\hat{\sigma}, \ldots)) \quad = \quad \prod_{i=0}^{n} \left(\frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(l_i - a_i)^2}{2\sigma_i^2}} \right).$$

Maximizing Likelihood. Now we can take the logarithm of the simplified expression and obtain:

$$\ln(\Pr(\mathcal{D}, \ldots \mid \mathcal{H}(\hat{\sigma}, \ldots))) \quad = \quad \sum_{i=0}^{n} \ln \left(\frac{1}{\sqrt{2\pi}\sigma_i} \right) - \sum_{i=0}^{n} \left(\frac{(l_i - a_i)^2}{2\sigma_i^2} \right).$$

This expression maximizes *log likelihood*, therefore it provides a Max. Likelihood Estimate (MLE).

Minimizing "Weighted Sum-of-Squares". Since we can assume that the first term of the MLE does not vary much from location to location, we simplify the problem by minimizing a "weighted sum-of-error-square" *cost function.*

$$\mathbf{F}(\mathcal{D}, \ldots) \quad = \quad \sum_{i=0}^{n} \left(\frac{(l_i - a_i)^2}{2\sigma_i^2} \right)$$

Minimizing $\mathbf{F}(\mathcal{D}, \ldots)$ yields the "best match" of the sequence map represented as $\mathcal{H}$, against the consensus map represented as $\mathcal{D}$.

2.2 Orientation, False Cuts, and Missing Cuts

First of all we need to take into account the two possible *orientations* of the sequence map with respect to the consensus map. Next we must take into account *false cuts* and *missing cuts* in the consensus map[3].

2.2.1 Orientation

Since the sequence map can be evaluated against the consensus map by "reversing" its orientation, we rewrite the expression for $\Pr(\mathcal{D}, \hat{\sigma}, \ldots \mid \mathcal{H})$ as:

$$\Pr(\mathcal{D}, \hat{\sigma}, \ldots \mid \mathcal{H}) \quad = \quad \max \left[\Pr_1(\mathcal{D}, \hat{\sigma}, \ldots \mid \mathcal{H}),\ \Pr_2(\mathcal{D}, \hat{\sigma}, \ldots \mid \mathcal{H}^R) \right],$$

where $\mathcal{H}^R$ represents the reversed sequence map. Proceeding as before, we construct the function $\mathbf{F}$ as

$$\mathbf{F}(\mathcal{D}, \ldots, \mathcal{H}) \quad = \quad \max \left[\mathbf{F}_1(\mathcal{D}, \ldots, \mathcal{H}),\ \mathbf{F}_2(\mathcal{D}, \ldots, \mathcal{H}^R) \right].$$

The form for $\mathbf{F}_2(\mathcal{D}, \ldots, \mathcal{H}^R)$ will turn out to be

$$\mathbf{F}_2(\mathcal{D}, \ldots, \mathcal{H}^R) \quad = \quad \sum_{i=0}^{n} \left(\frac{\left(l_i - a_{(n-i)}\right)^2}{2\sigma_i^2} \right).$$

[2] I.e. $\Pr(\mathcal{H}) = 1$.

[3] We still make the simplifying assumption that the sequence map is correct.

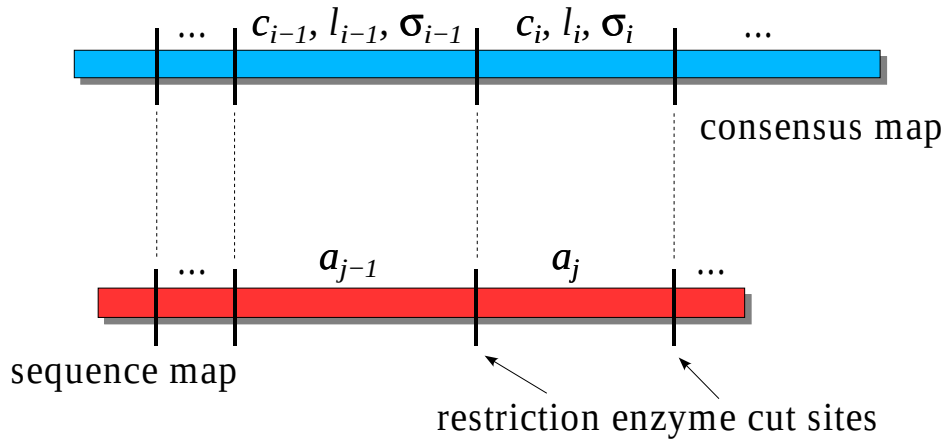

Figure 3: Straight Match

2.2.2 False Cuts and Missing Cuts

In order to correctly model errors in the matching process, we need to take into account *false cuts* and *missing cuts*. As in [3], we model them as with two parameters.

- Missing restriction sites in the sequence map are modeled by a probability p_c ("cut" probability). $p_c = 1$ means that the restriction sites are actually present in the map, $0 \leq p_c < 1$ means that there are some missing cuts.
- False restriction sites in the consensus map are modeled by a rate parameter p_f ("false" cut probability). In this case $0 < p_f \leq 1$ means that the consensus map may have some false cuts.

These parameters must be included in the expression describing $\Pr(\ldots)$ and, therefore, $\mathbf{F}(\ldots)$. The details follow.

Case 1: No missing cuts and no false cuts. In this case the term for the matching of the i-th fragment of the sequence map against the i-th fragment of the consensus map must take into account the *cut probability* p_c. The situation is depicted in Figure 3 (with indexes i and j). The expression becomes

$$p_c \times \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(l_i - a_j)^2}{2\sigma_i^2}},$$

yielding (after taking the negative log likelihood) the cost function

$$\ln\left(\frac{\sqrt{2\pi}\sigma_i}{p_c}\right) + \frac{(l_i - a_i)^2}{2\sigma_i^2}.$$

Case2: Missing cuts and no false cuts. In this case, our model considers a cut in the sequence map that has no corresponding cut in the consensus map. The situation is depicted in Figure 4. We try to match the i-th consensus map fragment against the aggregation of the i and $i-1$ fragments in the sequence map. I.e. the computation of the Gaussian expression must be "penalized" by taking into account the missing cut. We model the main term as

$$p_c \times \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{\left(l_i - \left(a_i + a_{(i-1)}\right)\right)^2}{2\sigma_i^2}} \times (1 - p_c).$$

yielding a cost function:

$$\ln\left(\frac{\sqrt{2\pi}\sigma_i}{p_c}\right) + \frac{\left(l_i - \left(a_i + a_{(i-1)}\right)\right)^2}{2\sigma_i^2} + \ln\left(\frac{1}{1 - p_c}\right).$$

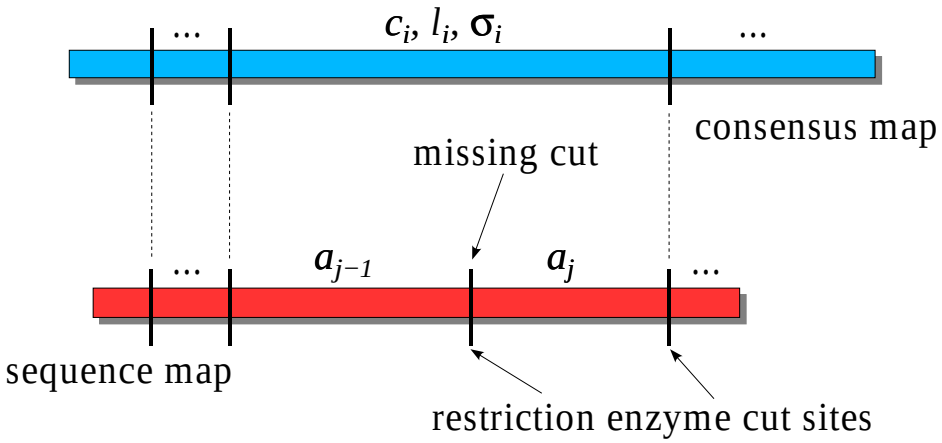

Figure 4: Missing Cut Match

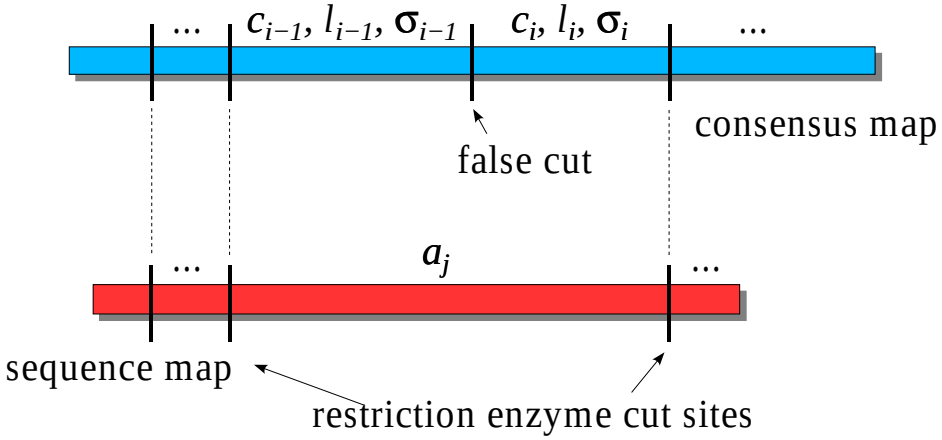

Figure 5: False Positive Match

Case 3: No missing cuts and some false cuts. In this case, we are considering the converse case of Case 2. The consensus map false cut event is modeled as a Bernoulli trial with probability p_f. This case is depicted in Figure 5. I.e. the full term for this matching will aggregate fragments i and $i-1$ of the consensus against the i-th fragment of the sequence map.
The full term will become

$$p_c \times \frac{1}{\sqrt{2\pi\left(\sigma_i^2+\sigma_{(i-1)}^2\right)}} e^{-\frac{\left(\left(l_i+l_{(i-1)}\right)-a_i\right)^2}{2\left(\sigma_i^2+\sigma_{(i-1)}^2\right)}} \times p_f.$$

Taking the negative log likelihood again, we obtain

$$\ln\left(\frac{\sqrt{2\pi\left(\sigma_i+\sigma_{(i-1)}\right)}}{p_c}\right)+\frac{\left(\left(l_i+l_{(i-1)}\right)-a_i\right)^2}{2\left(\sigma_i^2+\sigma_{(i-1)}^2\right)}+\ln\left(\frac{1}{p_f}\right).$$

It must be noted that for the current data obtained from the optical mapping process, $p_f \approx 10^{-5}$, and often dominates the complete expression.

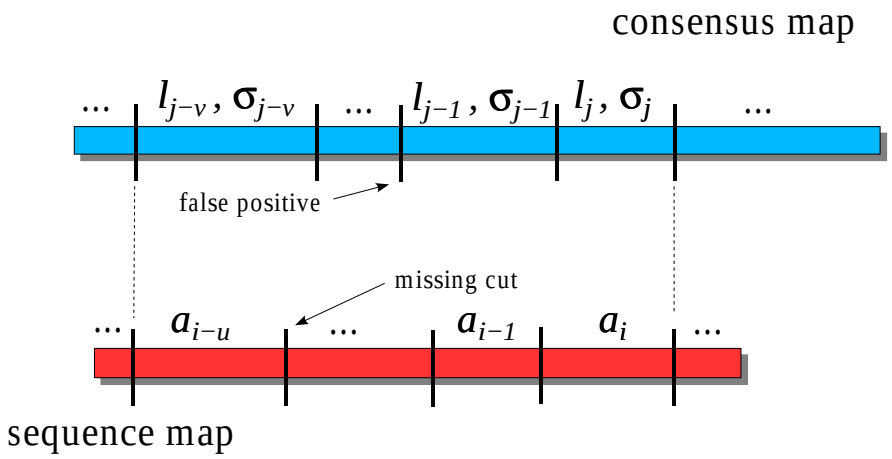

Figure 6: General Match

Putting it all together. Of course, missing cuts and false cuts may happen together. Suppose that we were able to *correctly* match (i.e. *align*) the $i-u$ cut in the sequence map against the $i-v$ cut in the consensus map. Now we can match correctly the $(i+1)$-th cut[4] in the two maps only by properly treating all the intervening missing cuts in sequence map and all the intervening false cuts in the consensus map. This situation is depicted in Figure 6 (again, with indices j and i). In this case the "matching term" takes its most general form.

$$\begin{aligned} p_c \quad \times \quad & \frac{1}{\sqrt{2\pi\left(\sigma_i^2+\sigma_{(i-1)}^2+\ldots+\sigma_{(i-v)}^2\right)}} \\ \times \quad & e^{-\frac{\left(\left(l_i+l_{(i-1)}+\ldots+l_{(i-v)}\right)-\left(a_i+a_{(i-1)}+\ldots+a_{(i-u)}\right)\right)^2}{2\left(\sigma_i^2+\sigma_{(i-1)}^2+\ldots+\sigma_{(i-v)}^2\right)}} \\ \times \quad & (1-p_c)^{(u-1)} \\ \times \quad & p_f^{(v-1)}. \end{aligned}$$

Taking the negative log likelihood, we obtain:

$$\begin{aligned} -\ln(p_c) \quad + \quad & \ln\left(\sqrt{2\pi\left(\sigma_i^2+\sigma_{(i-1)}^2+\ldots+\sigma_{(i-v)}^2\right)}\right) \\ + \quad & \frac{\left(\left(l_i+l_{(i-1)}+\ldots+l_{(i-v)}\right)-\left(a_i+a_{(i-1)}+\ldots+a_{(i-u)}\right)\right)^2}{2\left(\sigma_i^2+\sigma_{i-1}^2+\ldots+\sigma_{i-u}^2\right)} \\ + \quad & (u-1)\ln\left(\frac{1}{1-p_c}\right) \\ + \quad & (v-1)\ln\left(\frac{1}{p_f}\right). \end{aligned}$$

3 Dynamic Programming Procedure

The *validation* of a sequence map against a (optical) map can be set up as a simple *dynamic programming procedure* (DPR) (*cf.* [6, 8]). The overall validation procedure is shown in Figure 7. Let's start with the dynamic programming procedure *main recurrence* formulation of the sequence map vs. consensus map matching problem.

[4] I.e. the cut right after the i-th fragment in both the consensus map and the sequence map.

```
Procedure sequence-map-validate(sequence-map, consensus-map)
/* Other parameters will specified ... E.g. p_f, p_c, k, etc. */
begin
    run DPR on consensus-map and sequence-map;
    run DPR on consensus-map and reversed sequence-map;
    collect the k "best" alignments by examining the last row
        of both DPR tables and "return" them;
end
```

Figure 7: The overall sequence map validation procedure. We are running the *dynamic programming procedure* (*DPR*) twice as a simplification. We deem improbable that two alignments for a sequence map and for its reversed version will have equivalent scores.

3.1 Dynamic Programming "Main" Recurrence

Let's assume that the consensus map has m fragments and that the sequence map has n fragments. The DPR will use a $n \times m$ matching table T. Let's now consider the entry $\mathtt{T}[i, j]$. This entry will contain the (partially computed) value of the matching function $\mathbf{F}(\ldots)$: i.e. $\mathbf{F}(\ldots)$ is incrementally computed from "left" to "right", by considering all possible fragment by fragment matches.

Note. From now on we will use the index i to indicate a fragment in the consensus map, and the index j to indicate a fragment in the sequence map.

The main recurrence for entry $\mathtt{T}[i, j]$ is the following.

$$\mathtt{T}[i,j] := \min_{\substack{0<u\le i\\ 0<v\le j}} \left(\begin{array}{l} \mathtt{T}[i-u,\, j-v] \\ +\ln\left(\frac{\sqrt{2\pi\left(\sigma_i^2+\sigma_{(j-1)}^2+\ldots+\sigma_{(i-v)}^2\right)}}{p_c}\right) \\ +\frac{\left(\left(l_j+l_{(j-1)}+\ldots+l_{(j-v)}\right)-\left(a_i+a_{(i-1)}+\ldots+a_{(i-u)}\right)\right)^2}{2\left(\sigma_j^2+\sigma_{(j-1)}^2+\ldots+\sigma_{(j-v)}^2\right)} \\ +(u-1)\,\ln\left(\frac{1}{1-p_c}\right) \\ +(v-1)\,\ln\left(\frac{1}{p_f}\right) \end{array} \right).$$

Of course, we now must ask how big u and v should be. The correct answer is that u and v depend on the σ_i's[5]. However, a pragmatic bound could be around 3. This will become a parameter of the DPR. In this way, the computation for each entry $\mathtt{T}(\cdot,\cdot)$ must consider about 9 nearby entries.

3.1.1 Boundary Conditions

A simple model for the initial conditions will be

$$\begin{array}{lcl} \mathtt{T}[i,0] & := & \infty, \quad \text{for } i \in [1,N]. \\ \mathtt{T}[0,j] & := & 0, \quad \text{for } j \in [0,M]. \end{array}$$

In this model we never match (i.e. we strongly *penalize* a match of) the first fragments of the consensus map against an "inner" fragment of the sequence map (*cf.* first column ∞ value). Also, we make the match of any fragment of the consensus map against the first fragment of the sequence map rather neutral (*cf.* the first row 0 value).

A more complex model would initialize the first row of the dynamic programming table by taking into account only the size of the i-th fragment. In Section 3.2 we show a complete model for the boundary conditions.

[5] In an even more accurate model, u and v would also depend also on the *digestion rate* of the "in vivo" experiment that breaks up the DNA molecule.

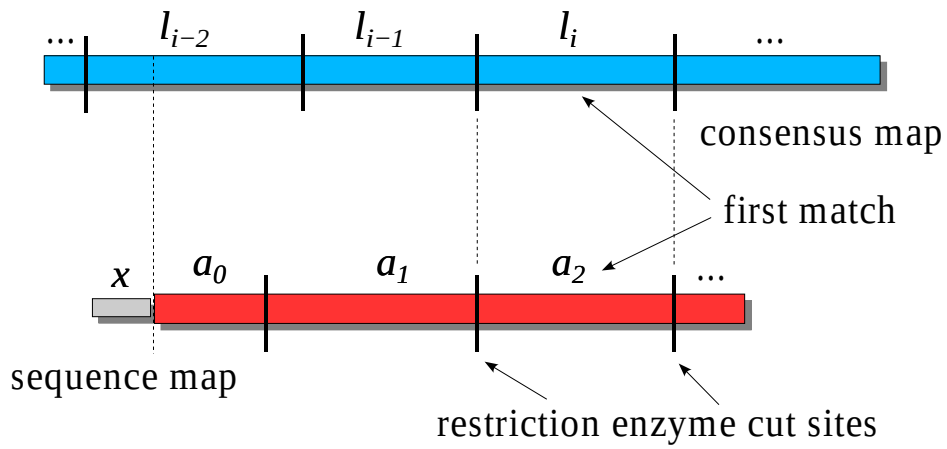

Figure 8: Left end fragments mismatch (the σ_j's are not shown).

3.2 Left and Right End Fragment Computations.

We can devise more sophisticated and correct models for the left fragments and right fragments calculations (i.e. for the initial and final conditions). Both models take into account the case when some fragments on either the left or the right of the sequence map do not "properly match" any fragment in the consensus map.

3.2.1 Left End Penalty Computation.

Consider the case shown in Figure 8. The first "matching fragments" in the figure are a_2 from the sequence map and l_j from the consensus map, identified by their size. The general case is for fragment i of the sequence map to match fragment j of the consensus map.

Let's consider what happens at fragment a_0 of the sequence map. Most of the times, its left end – which we can assume not to be corresponding to an actual restriction site[6] – will fall within the boundaries of fragment $i-n$ of the consensus map (for $0 \leq n \leq i$).

Within this framework, the minimum value that we can assign to a "match" of the left end fragments of the sequence map, corresponds to one of three cases.

1. Match by *extension* of the first left end fragment of the sequence map.
2. Bad matches until fragment i of the sequence map matches fragments j of the consensus map.
3. Match without extension to some fragment in the consensus map.

Case 1: extending a_0 by x leads to a match. Suppose that we "extended" a_0 by an extra size x (as shown in Figure 8. In this case we extend x as far left as we can in order to match the cut on the left of fragments $i-n$ (i.e. of fragment of size l_{i-2} in Figure 8).

The value of this match (built on top of the derivation done for the "regular case") is given by the following expression:

$$
\begin{aligned}
&\ln\left(\frac{\sqrt{2\pi\left(\sigma^2_{(i-n)}+\sigma^2_{(i-(n-1))}+\ldots+\sigma^2_i\right)}}{p_c}\right)\\
&+\frac{\left(\left(l_{(i-n)}+l_{(i-(n-1))}+\ldots+l_i\right)-\left(x+a_0+a_1+\ldots+a_j\right)\right)^2}{2\left(\sigma^2_{(i-n)}+\sigma^2_{(i-(n-1))}+\ldots+\sigma^2_i\right)}\\
&+\tfrac{x}{L}\\
&+(n-1)\,\ln\left(\tfrac{1}{p_f}\right)\\
&+j\,\ln\left(\tfrac{1}{1-p_c}\right).
\end{aligned}
$$

[6] We can guarantee that also by a careful encoding of the boundary fragments in the underlying data structure.

This expression depends on two parameters which did not appear in the regular case:

- x: the *size extension* (please note it in the second and the third term),
- L: the molecule map *average fragment size.*

The second sub-term is the regular "sizing error" penalty which takes into account the extension x. The third sub-term adds an extra penalty based on the amount we are stretching the end fragment with respect to the overall structure of the problem.

To use the expression we must find where its minimum respect to x lies. By *differentiating* we find that the expression is minimized at

$$\begin{aligned} x &= \left(\left(l_{(i-n)} + l_{(i-(n-1))} + \ldots + l_i\right) - (a_0 + a_1 + \ldots + a_j)\right) \\ &\quad - \frac{\sigma^2 + \sigma^2 + \ldots + \sigma^2}{L}. \end{aligned}$$

By substituting this value for x in the original expression we obtain the following, simplified, form:

$$\begin{aligned} &\ln\left(\frac{\sqrt{2\pi\left(\sigma^2_{(i-n)}+\sigma^2_{(i-(n-1))}+\ldots+\sigma^2_i\right)}}{p_c}\right) \\ &+\frac{\left(\left(l_{(i-n)}+l_{(i-(n-1))}+\ldots+l_i\right)-(a_0+a_1+\ldots+a_j)\right)}{L} \\ &+\left(-\tfrac{1}{2L^2}\right)\left(\sigma^2_{(i-n)} + \sigma^2_{(i-(n-1))} + \ldots + \sigma^2_i\right) \\ &+n\ln\left(\tfrac{1}{p_f}\right) \\ &+j\ln\left(\tfrac{1}{1-p_c}\right). \end{aligned}$$

Again, the last two sub-terms account for false cuts and for missing cuts. Note that we assume that there is at least one "good" cut in the sequence map.

Case 2: no extension and bad matches until i and j. In this case we are considering the simple case where the first "good match" is between fragment i of the sequence map matches fragments j of the consensus map. The term corresponding to this case is simply

$$n\ln\left(\frac{1}{p_f}\right) + (j+1)\ln\left(\frac{1}{1-p_c}\right).$$

This term takes care of all the missing matches and the false matches in both maps (the $j+1$ term takes into account the 0-th cut as a missing one).

Case 3: match without extension to some fragment in the consensus map. In this case we have the following situation. Remember that we assume to have a "good match" between fragment i of the consensus map and fragments j of the sequence map.

Suppose now (as in Case 1) that the fragment from the consensus map, within which the end of fragment 0 (size a_0) of the sequence map lies, is indexed $i-n$.

We will try to match fragment 0 of the sequence map to any of the n fragments up to fragment i of the consensus map. We will take into account all possible missing cuts and false cuts along the way. The attempt minimizing the following expression (dependent on k) will compete against the terms in Case 1 and Case 2 for the best left end match.

$$\min_{0\le k\le i}\left(\begin{array}{l} \frac{\left(\left(l_{(i-k)}+l_{(i-(k-1))}+\ldots+l_i\right)-(x+a_0+a_1+a_j)\right)^2}{2\left(\sigma^2_{(i-n)}+\sigma^2_{(i-(n-1))}+\ldots+\sigma^2_i\right)} \\ +(k-1)\ln\left(\frac{1}{p_f}\right) \\ +j\ln\left(\frac{1}{1-p_c}\right). \end{array}\right)$$

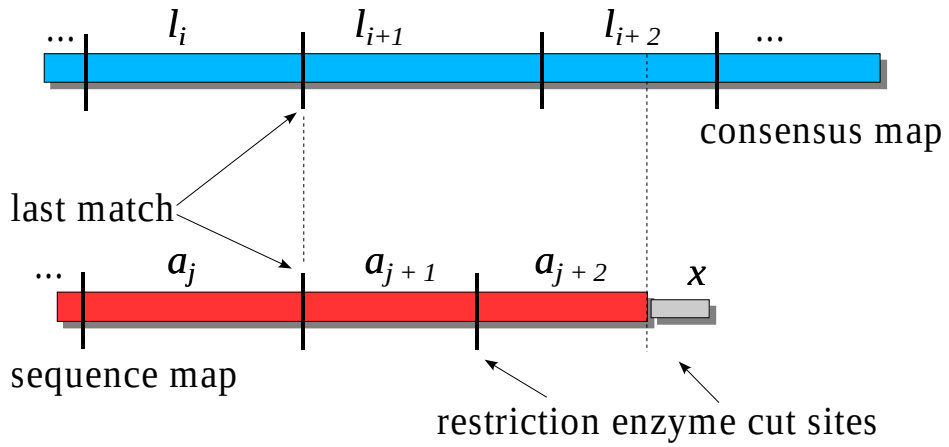

Figure 9: Left end fragments mismatch (the σ_j's are not shown).

3.2.2 Right End Penalty Computation.

The computation for the right end penalty for fragments trailing the end of the sequence and or of the consensus map is *almost* specular to the the left end penalty computation. Figure 9 depicts the situation for one simple case.

There is a difference to be taken into account for the right end computation, which makes it asymmetrical with respect to the left end case. When we consider the "last good match" between fragment i of the consensus map and fragment j of the sequence map, we must also consider what is the score of the match up to that point: i.e. we must consider the value $\mathtt{T}[j, i]$ (thus assumed to be available at this time).

Hence, as per the left end computation, we have three terms to be considered. They are analogous to the three terms for the left end computation, but they must be augmented with $\mathtt{T}[j, i]$ to be meaningful.

3.2.3 Complete Algorithm Description.

We can finally summarize the overall architecture of the validation algorithm. The architecture is a rather complex composition of simple applications of the "Dynamic Programming" paradigm. Figure 10

The overall computation uses three tables[7]:

1. the $\mathtt{T}[\cdot, \cdot]$ for the "middle computation", which we have already seen;
2. the $\mathtt{TL}[\cdot, \cdot]$ for the *left* end penalty computation;
3. the $\mathtt{TR}[\cdot, \cdot]$ for the *right* end penalty computation.

The flow of control produces the content of each table in turn and the final resulting table can be examined to reconstruct the alignment traceback.

3.3 Complexity Claims

Naïvely filling the whole $\mathtt{T}(\cdot, \cdot)$ table takes order 4 time $O(N^2 M \min(N, M))$, where N is the size of the sequence map and M is the size of the consensus map. However, we actually optimize it down to $O(NM \min(N, M))$ thanks to the limiting argument on the computation done for each entry $\mathtt{T}(i, j)$. Because of the limit on u and v, the computation time for each entry can be considered "constant".

In a simple setup, the table takes up $O(NM)$ space, hence it too is quadratic in the worst case, even when considering extra "backtrace recording" (*cf.* [8]).

[7] The actual implementation could be done in a smarter way and save memory by reusing some tables.

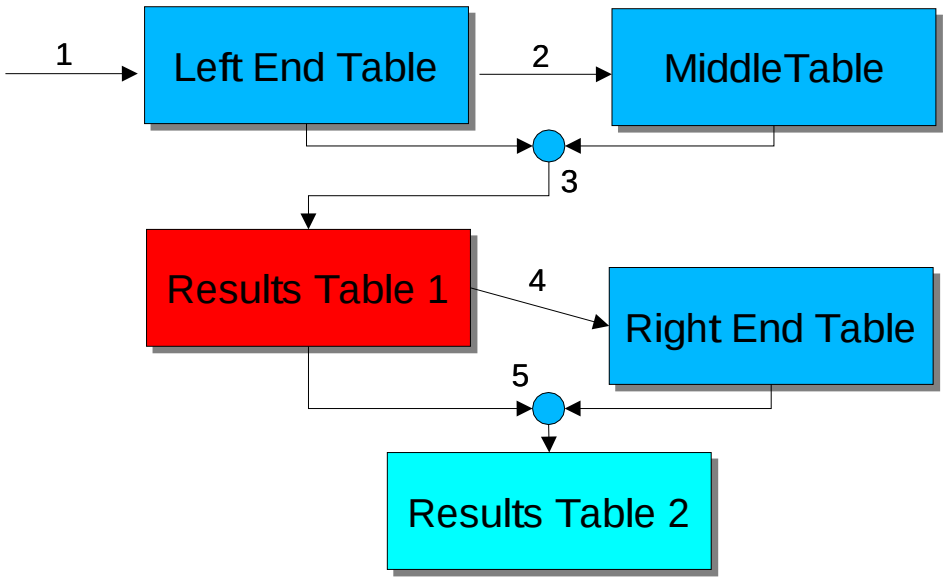

Figure 10: The overall architecture of the validation computation. Each box represents the solution of a "dynamic programming" like problem. The result tables (the intermediate tagged 1, and the final, tagged 2), are computed using the scores contained n the other ones. The arrows illustrate the flow of control of the computation.

Chromosome	Number of Fragments	
	from DB	*reversed*
chr 2	30	23
chr 3	36	28

Table 1: Number of fragments obtained from the *Plasmodium falciparum*'s chromosomes 2 and 3 by the "in silico" digestion process. The enzyme used is `BamHI`.

Further Optimizations Although we have not yet implemented them, we can add further optimizations to the running time via a hashing scheme as the one used in gentig. In this case the time complexity can be reduced by another order of magnitude.

4 Experimental Results

We used the software we developed based on the mathematical model described in the previous sections to run several experiments. Our first experiments checked "in silico" maps obtained from *Plasmodium falciparum* sequence data against optical ordered restriction maps for the same organism.

4.1 *Plasmodium falciparum* Sequence Data

We obtained the sequences for the *Plasmodium falciparum*'s 14 chromosomes from the Sanger Institute database (`www.sanger.ac.uk`) and from the TIGR database (`www.tigr.org`). Our experiment cuts the sequences "in silico" using the `BamHI` restriction enzyme. The resulting maps are fed to the validity checking program alongside appropriate optical ordered restriction maps.

Here, we report results of our extensive experiments on chromosome 2 and chromosome 3. In the full paper, we describe the full experiment on all the chromosomes, and with other enzymes (e.g. `NheI`).

We produce two "in silico" maps for the chromosome 2 and chromosome 3 sequences. Table 1 shows the values we obtained. The molecule maps thus produced are then sent to the validation checker alongside various consensus maps.

Chromosome 2 Validation Summary A

rank	matches	score	map id	# missing cuts	# false cuts
1	29	80.869	1302	0	1
2	28	105.861	1302	2	1
3	18	126.956	1326	12	4
4	22	127.488	1305	8	4
5	18	132.890	1414	12	2

Table 2: The data reported shows the best "matches" found by the validation checker in the case of *Plasmodium falciparum* chromosome 2. The "in silico" sequence map was obtained from the TIGR database sequence. The sequence map (as well as its reversed) was checked against 75 (optical) consensus maps produced by gentig. The 75 optical maps cover the entire *Plasmodium falciparum* genome. The validity checker found its best matches against the map tagged 1302.

4.2 *Plasmodium falciparum* Optical Ordered Restriction Maps

For our experiment we used optical ordered restriction map published in [10, 11] and the maps generated by the gentig program.

The gentig program gave us an indication of the overall standard deviation to be used for each fragment of the consensus map. The parameter used was

$$\hat{\sigma} \quad = \quad 4.4754 \text{ Kbps},$$

and each fragment was assigned a standard deviation of

$$\hat{\sigma}\sqrt{\frac{l}{L}} \text{ Kbps},$$

where l is the *fragment size* and L is the *average* consensus map fragment size.

4.3 Validation Check Results

We ran the validity checker on chromosome 2 and chromosome 3. The dynamic programming procedure ran with the following limitations.

- The u and v parameters for the main recurrence formula were set to 3.
- The procedure for matching the left and right ends of the sequence maps using the special computations described in Section 3.2 were not used.

We produced the results reported in Tables 2, 3, and 4. Table

Tables 2 and 4 show the match of the sequence maps for chromosomes 2 and 3 against the consensus maps generated by gentig. Table 3 show the match of the sequence maps against the consensus map published in [7].

The data we report here is a summary of the data we actually produced. In particular we also have the position of the matches of the sequence maps against the consensus maps.

Comment. By reading Tables 2, 3, and 4 we can get a sense of the goodness of the sequences, the maps, and the tools we used to run our experiments.

Speed. Our implementation is not really optimized. However, speed did not turn out to be an issue. The system runs the $75 \times 4 = 300$ DPR instances in about 5 minutes[8].

[8]The system may appear slow, but in reality it keeps track of all the intermediate results and makes them available for interactive inspection after the actual run. Also, the sequence, the sequence map, and the consensus maps, are always available for inspection and manipulation.

Chromosome 2 Validation Summary B

rank	matches	score	map id	# missing cuts	# false cuts
1	29	77.308	NYU-WISC	1	0
2	22	125.088	NYU-WISC	8	2
3	22	130.866	NYU-WISC	8	4
4	24	131.475	NYU-WISC	6	1
5	24	132.838	NYU-WISC	6	4

Table 3: The data reported shows the best "matches" found by the validation checker in the case of *Plasmodium falciparum* chromosome 2. The "in silico" sequence map was obtained from the TIGR database sequence. The sequence map (as well as its reversed) was checked against the map published in [7].

Chromosome 3 Validation Summary

rank	matches	score	map id	# missing cuts	# false cuts
1	35	108.360	1365	1	0
2	32	117.571	1365	4	1
3	32	119.956	1365	4	2
4	35	121.786	1296	1	3
5	31	125.265	1365	5	1

Table 4: The data reported shows the best "matches" found by the validation checker in the case of *Plasmodium falciparum* chromosome 3. The "in silico" sequence map was obtained from the Sanger Institute database sequence. The sequence map (as well as its reversed) was checked against 75 (optical) consensus maps produced by gentig. The 75 optical maps cover the entire *Plasmodium falciparum* genome. The validation checker found its best matches against the map tagged 1365.

VALIS. The validation checker is an integral part of the VALIS system being developed in NYU Bioinformatics Group [4]. The VALIS system aims to be an integrated environment targeted at biologists and bioinformatics practitioners. Within VALIS it will be possible to develop algorithms and perform "in silico" experiments with ease.

5 Conclusion

We described the underlying mathematical model and the dynamic programming algorithm for the *validation* of a (DNA) *sequence* against a (DNA) *map*.

The statistical model is essentially a formulation of a maximum likelihood problem which is solved by minimizing a weighted sum-of-square-error score. The solution is computed by constructing a "matching table" using a dynamic programming approach whose overall complexity is of the order $O(NM \min(N, M))$ (for our non optimized solution), where N is the length of the sequence map and M is the length of the consensus map.

We ran our program using publicly available data for chromosomes 2 and 3 of *Plasmodium falciparum*. Our very preliminary results, point out how our technology can be useful in assessing the goodness of various sequence and map data currently being published in a variety of formats from a variety of sources.

References

[1] T. Anantharaman, B. Mishra, and D. C. Schwartz. A Probabilistic Analysis of False Positives in Optical Map Alignment and Validation. submitted to ISMB 2001.

[2] T. Anantharaman, B. Mishra, and D. C. Schwartz. Genomics via Optical Mapping II: Ordered Restriction Maps. *Journal of Computational Biology*, 4(2):91–118, 1997.

[3] T. Anantharaman, B. Mishra, and D. C. Schwartz. Genomics via Optical Mapping III: Contiging Genomic DNA and Variations. In *7th International Conference on Intelligent Systems for Molecular Biology: ISMB 99*, volume 7, pages 18–27. AAAI Press, 1999.

[4] M. Antoniotti, B. Mishra, and S. Paxia. VALIS, an Integrated Development Environment for Bioinformatics. WEB site at `http://galt.mrl.nyu.edu/valis`, 2000.

[5] C. Aston, B. Mishra, and D. C. Schwartz. Optical Mapping and Its Potential for Large-Scale Sequencing Projects. *Trends in Biotechnology*, 17:297–302, 1999.

[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press and McGraw-Hill, 1990.

[7] M. J. Gardner et al. Chromosome 2 sequence of the human malaria parasite *Plasmodium Falciparum. Science*, 282:1126–1132, 1998.

[8] D. Gusfield. *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, 1997.

[9] X. Huang and M. S. Waterman. Dynamic Programming Algorithms for Restriction Map Comparison. *Comp. Appl. Bio. Sci.*, 8:511–520, 1992.

[10] J. Jing, Z. Lai, C. Aston, J. Lin, D. J. Carucci, M. J. Gardner, B. Mishra, T. Anantharaman, H. Tettelin, L. M. Cummings, S. L. Hoffman, J. C. Venter, and D. C. Schwartz. Optical Mapping of *Plasmodium Falciparum* Chromosome 2. *Genome Research*, 9:175–181, 1999.

[11] Z. Lai, J. Jing, C. Aston, V. Clarke, J. Apodaca, E. T. Dimalanta, D. J. Carucci, M. J. Gardner, B. Mishra, T. Anantharaman, S. Paxia, S. L. Hoffman, J. C. Venter, E. Huff, and D. C. Schwartz. A shotgun optical map of the entire *Plasmodium Falciparum* genome. *Nature Genetics*, 23:309–313, 1999.

[12] X. Su, M. T. Ferdig, Y. Huang, C Q. Huynh, A. Liu, J You, J. C. Wootton, and T. E. Wellems. A Genetic Map and Recombination Parameters of the Human Malaria Parasite *Plasmodium falciparum. Science*, 286, 1999.