3-1-2013

A Graph-Theoretical Approach to the Selection of the Minimum Tiling Path from a Physical Map

Serdar Bozdag
Marquette University, serdar.bozdag@marquette.edu

Accepted version. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 10, No. 2 (March 2013): 352-360. DOI.

A Graph-Theoretical Approach to the Selection of the Minimum Tiling Path from a Physical Map

Serdar Bozdag
Marquette University, Milwaukee, WI
Timothy J. Close
University of California Riverside, Riverside, CA
Stefano Lonardi
University of California Riverside, Riverside, CA

Abstract: The problem of computing the minimum tiling path (MTP) from a set of clones arranged in a physical map is a cornerstone of hierarchical (clone-by-clone) genome sequencing projects. We formulate this problem in a graph theoretical framework, and then solve by a combination of minimum hitting set and minimum spanning tree algorithms. The tool implementing this strategy, called FMTP, shows improved performance compared to the widely used software FPC. When we execute FMTP and FPC on the same physical map, the MTP produced by FMTP covers a higher portion of the genome, and uses a smaller number of clones. For instance, on the rice genome the MTP produced by our tool would reduce by about 11 percent the cost of a clone-by-clone sequencing project. Source code, benchmark data sets, and documentation of FMTP are freely available at http://code.google.com/p/fingerprint-basedminimal-tiling-path/ under MIT license.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

SECTION 1

Introduction

A *physical map* is a partial ordering of a set of genomic clones (usually bacterial artificial chromosomes or BACs) encompassing one or more chromosomes. A physical map can be represented by a set of unordered *contigs*, where each contig is a set of overlapping clones (see Fig. 1). A physical map is usually obtained by digesting BAC clones via restriction enzymes into DNA fragments and then measuring the length of the resulting fragments (*restriction fingerprints*) on an agarose gel. Since restriction enzymes cut clones at specific sites, overlapping clones in the genome will share a much larger number of fragments with similar length than nonoverlapping clones. By comparing all restriction fingerprints pairwise, overlaps between clones can be detected, and clones can be eventually "assembled" in a physical map. To minimize gaps, each genomic location is covered by many BACs in the physical map. This redundancy has to be removed before sequencing the clones. The smallest set of clones that spans the region represented by the physical map is called *minimum tiling path* (MTP) (see Fig. 1). The problem of computing the MTP can be approached at the level of each contig. Given a set of clones in a contig along with their restriction fingerprints, select the minimum set of clones that covers the same genomic region of the contig.

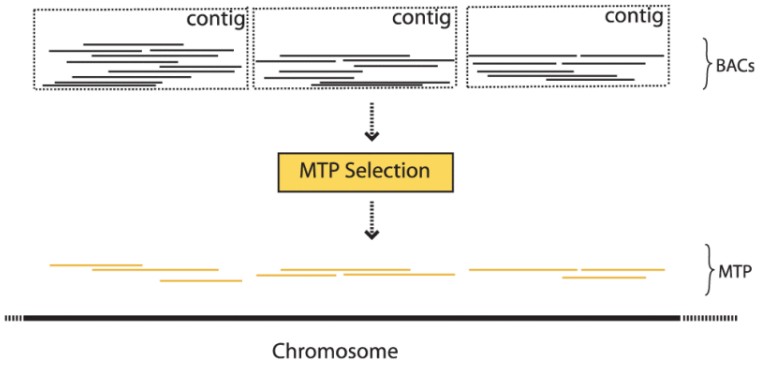

Fig. 1. Illustrating the MTP of a physical map. A physical map consists of a set contigs, where each contig is a set of overlapping BAC clones. The MTP is the smallest set of clones that cover the same genomic region covered by the physical map.

Determining the smallest set of clones that cover a genome is a critical step in *clone-by-clone* (or *hierarchical*) sequencing projects. In this protocol, first a physical map is constructed, then clones in the MTP are sequenced one by one.[1] The clone-by-clone sequencing method has been used to sequence several genomes including *A. thaliana*[2] and

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

H. sapiens[3] among others. Also, in several recent whole-genome shotgun sequencing projects, the MTP obtained from a physical map has been employed to validate and improve the quality of sequence assembly.[4] This validation step has been used, for example, in the assembly of *M. musculus*,[5] *R. norvegicus*,[6] and *G. gallus*[7]

While shotgun sequencing is currently the most popular protocol for whole genome sequencing because it does not require cloning, the clone-by-clone approach has some advantages in certain domain of applications. For instance, very large/repetitive genomes can be extremely challenging to assemble, in particular from short reads obtained from next-generation sequencing instruments. In the clone-by-clone protocol, sequenced reads are separated into sets that represent individual BACs, and some sequences that are repetitive in the context of the whole genome can have only a single copy in each BAC; this greatly simplifies the assembly. Another advantage of the clone-by-clone approach is the ability of sequencing only a portion of a genome, a strategy called *selective sequencing*. One can focus, for instance, on the gene-rich fraction of the genome. To identify gene-rich segments, one can take advantage of EST sequences to design probes for screening, use gene enrichment techniques like methyl-filtration or high C_o^t selection, or employ microarrays to capture the genomic DNA fragments that are enriched for genes.

As said, computing the MTP from restriction fingerprints is challenging. If the exact locations of all clones in the physical map were known, the problem would be straightforward; simply select the set of clones in the shortest path from the leftmost clone to the rightmost clone in the interval graph representing all the clones. This ideal scenario is, however, far from the reality because noise in the fingerprinting data can only provide us with approximate clone overlaps.

Although the problem of computing an MTP has been studied previously (see, e.g.,[2,8,9]), in practice there is only one commonly used software tool, namely, FingerPrinted Contigs (FPC).[10] FPC builds a physical map from fingerprinting data and it offers two methods to compute an MTP. One of these methods uses sequence comparison between a draft sequence and BAC end sequences (BESs), whereas the other method uses solely restriction fingerprint data. Since a draft sequence and BESs are usually not available in the early stages of the physical mapping projects, we only consider the second method in this paper (hereafter called FPC unless specified otherwise).

The FPC's MTP algorithm works as following: For each contig, an overlap score is computed for all pairs of clones that have a distance less than a user-specified maximum in the contig. To compute the overlap score, FPC incorporates restriction fingerprint data of

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

three extra clones, namely, a *spanner* clone that spans both clones in the pair and two flanking clones that extend to the left and right of the pair. The distance between clone pairs, and selection of spanner and flanking clones are based the coordinates of clones in the contig. The overlap score of the pair is computed as a weighted sum of number of matching fragments between clones in the pair, number of unmatching fragments in the spanner clone to either clone in the pair and number of unmatching fragments in the clone in the pair to either spanner or flanker clones. Once the overlap score between all pairs in the contig is computed, FPC uses clone coordinates in the contig to select the subset of clones that cover the entire contig by minimizing the total overlap score.[10]

The performance of FPC is good, but it can be improved. Experimental results will show that the MTP computed by FPC can be significantly far from the perfect MTP, which is the MTP we would compute if we knew the genomic coordinates of all clones. In general, FPC selects fewer clones than necessary that in turns reduces the overall coverage. By changing parameters, one can increase the coverage, but this comes at the cost of introducing redundant clones (i.e., clones that do not provide additional coverage).

1.1 Our Contribution

We propose a new algorithm called fingerprint-based MTP (FMTP) that computes the MTP of a physical map based purely on restriction fingerprint data. Our algorithm analyzes each contig separately and completely ignores the ordering of clones in the contig obtained during the construction of the physical map, because such ordering is often unreliable due to noise in restriction fingerprint data.

FMTP first computes a tiling path by selecting a set of clones that covers the genomic region that is covered by all clones in the contig. This preliminary tiling path may contain redundant clones. The problem of computing the tiling path is formulated as a minimum hitting set problem and solved heuristicly. In the second phase, FMTP orders the clones in the tiling path and computes the MTP by using a shortest path algorithm. In the earlier version of this work,[11] we used integer linear programming to solve the minimum hitting set problem optimally. One of the anonymous reviewers suggested an alternative greedy heuristic that produced MTPs with higher coverage. The greedy approach is what is described below.

We carried out an extensive set of experiments on physical maps of rice and barley based on real restriction fingerprint data and a *synthetic* physical map of rice based on *in silico* fingerprint data. For the rice data set, the actual coordinates for the clones are known

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

and, therefore, we could measure the accuracy of our algorithm. Experimental results show that although both tools are unable to achieve 100 percent coverage, the MTPs computed by FMTP for both real and synthetic physical maps for rice has higher coverage than the ones produced by FPC while using approximately the same number of clones. This suggests that a larger portion of the genome could be obtained for the same sequencing cost. Our experimental results also show that if one fixes a given MTP coverage, FMTP produces MTPs with about 11 percent fewer clones than FPC. This suggests that replacing FPC by FMTP would reduce the sequencing costs by the same amount.

SECTION 2

Methods

We use the term *clone fragment* (or *fragment*) to indicate a portion of a clone obtained by digesting it with a restriction enzyme. Let $b(u)$ be the size for clone fragment u. We say that two clone fragments u and v *match* if their corresponding sizes are within the *tolerance* T, i.e., if $|b(u) - b(v)| \leq T$. The tolerance parameter depends on the fingerprinting method, thus it should match the one used in the construction of the physical map [12]. Given a fragment u, let $N(u)$ be the unique ID of the clone that u belongs to. This notion can be extended to a set of fragments: given a set of fragments U,$N(U)$ represents the set of all clones that contain at least one fragment in U.

We say that two clones c_i and c_j are *overlapping* if their Sulston score S is lower than or equal to a user-defined *cutoff* threshold C, i.e., $S(c_i, c_j) \leq C$. The Sulston score measures the probability that two clones share a given number of fragments by chance according to a binomial probability distribution.[13] FPC's physical mapping construction module uses an analogous cutoff parameter.[12]

A *matching fragment graph* (MFG) of a contig is a weighted undirected graph $G=(V,E)$ in which V represents the set of all clone fragments and an edge $(u,v)\in E$ exists if fragment u matches fragment v and $S(N(u), N(v))\leq C$. The weight on the edge $(u,v)\in E$ is defined as the negative logarithm of $S(N(u),N(v))$. In an ideal MFG, each connected component of the MFG G should correspond to a unique region in the target genome. In practice, due to noise in the fingerprint data and falsely matching fragments, each unique region can be fragmented in several connected components in the MFG.

Given a connected component $H=(Y,F), Y\subseteq V, F\subseteq E$ we call $N(Y)$ a *connected component cloneset* (or *cloneset* if it is clear from the context). We say that a clone c *covers*

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

a connected component H of G if it belongs to the corresponding cloneset, i.e., $c \in N(Y)$. Given an ideal MFG G, the *MTP* of G is the smallest set M of clones that cover all connected components of G, i.e., $M \cap N(Y) \neq \emptyset$ for all connected components H of G.

A small example of an MFG is shown in Fig. 2a. Each node is labeled as <clone name>-<fragment size>-<copy number of the fragment>. Clones K, L, M, N, and O are overlapping. Shared fragments are represented by the connected component of the MFG (namely, fragments of approximate size of 1,870, 1,805, 1,255, and 1,400 bases, respectively). For example, given the connected component $H=(Y,F)$ where Y={L-1,803-1,M-1,807-1,N-1,802-1}, the cloneset $N(Y)$ is {L, M, N}. If we assume that this is an ideal MFG, the MTP would be {L, O} because by selecting these two clones we cover all the connected components of the graph.

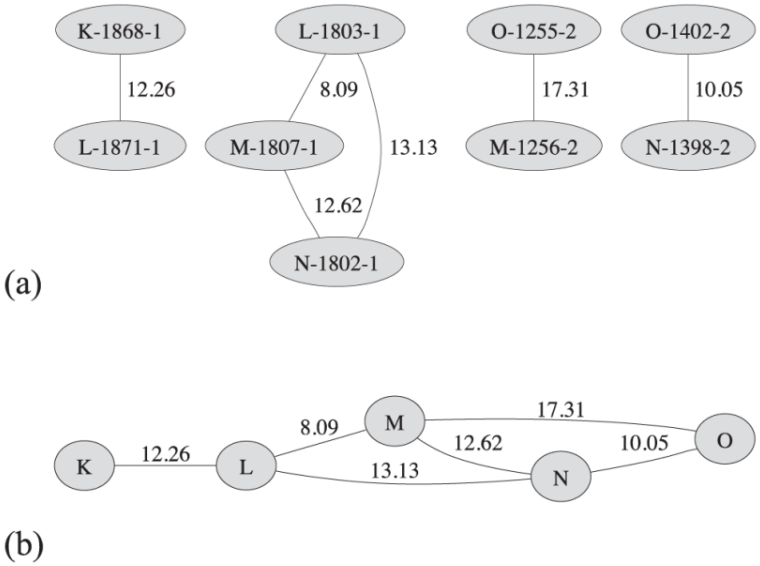

Fig. 2. (a) An example of an MFG. Each node represents a clone fragment and is labeled as the <clone name>-<fragment size>-<copy number of the fragment>. Each edge is weighted by the negative logarithm of the Sulston score between the clones that contain the incident node fragments. (b) Overlap graph of clones in this MFG. There is an edge between two clones if their fragments are together in at least one connected component in the MFG.

2.1 Algorithm

Before giving the details of our algorithm, we formally define the problem as follows:

Problem statement:

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

Input: Restriction fingerprints for a set of clones that belong to a contig. Each clone *covers* an unknown contiguous region (interval) of the genome; the coverage of a contig is the union of the intervals of its clones.

Output: The smallest subset of clones that cover all the bases covered by the contig.

The algorithm in FMTP is composed of two modules, namely the minimum hitting set module (MHS_MODULE) and the minimum spanning tree module (MST_MODULE) (see Fig. 3a). The MHS_MODULE aims to compute a tiling path without considering the possible redundancy by solving a minimum hitting set problem. The objective of MHS_MODULE is to cover all connected components in the MFG using the smallest set of clones. Since MFG is not ideal as mentioned above, the preliminary tiling path is expected to contain redundant clones. In the second phase, MST_MODULE removes redundant clones without affecting the coverage. The MST_MODULE computes the MTP based on an overlap graph: first it orders the clones in the tiling path, and then it runs a shortest path algorithm to compute the MTP.

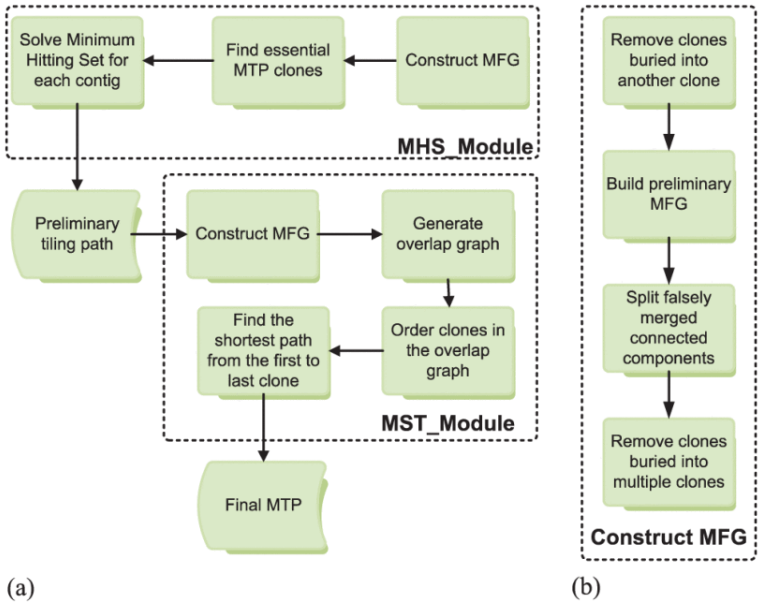

Fig. 3. (a) Flowchart of FMTP. First a preliminary tiling path is computed by MHS_MODULE. Then final MTP is computed by MST_MODULE. (b) Flowchart of MFG construction. This step is included in both modules.

2.1.1 Constructing the MFG

FMTP builds an MFG for each contig in the physical map so that each connected component in the MFG represents a unique region in the genome. To achieve this, for each

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

contig, after removing clones that are completely contained in others, FMTP performs a pairwise alignment between the remaining ones based on their restriction fingerprint data. The alignment produces the initial MFG. Then, some of the false-positive edges (i.e., falsely matched fragments) are removed and the final MFG is constructed, as illustrated in Fig. 3b and described below in more details.

Preprocessing: burying clones. A clone is defined *buried* if B percent or more of its fragments are matching fragments of another clone, where B is a user-defined parameter. Since a buried cloned should not be selected as MTP clone, in this preprocessing step, we aim to reduce the problem size and false-positive edges in the MFG by discarding buried clones. If two clones can be buried into each other, the smaller of the two (i.e., the one with fewer fragments) is buried into the other one. FPC also buries clones during the process of building the physical map, but it does not discard them during MTP computation.

Building the preliminary MFG. First, we align each pair of overlapping clones based on their restriction fingerprint data. For each clone pair (c_i, c_j) for which $S(c_i, c_j) \leq C$, we build a bipartite graph $G_{\{i,j\}} = (L_i \cup R_j, E_{\{i,j\}})$, where L_i and R_j consist of the fragments of c_i and c_j, respectively, and the set of edges is $E_{i,j} = \{(u, v) | u \in L_i, v \in R_j \text{suchthat} | b(u) - b(v)| \leq T\}$. To align clones c_i and c_j, we search for the maximum bipartite matching in $G_{\{i,j\}}$. The matching of maximum cardinality is found by solving maximum flow on the corresponding flow network.[14] Let $M_{\{i,j\}}$ be set of matched edges. For all clone pairs (c_i, c_j) for which $S(c_i, c_j) \leq C$, the matching edges in $M_{\{i,j\}}$ are used to create the (preliminary) MFG G. Specifically, for each edge $(u, v) \in M_{\{i,j\}}$, nodes u, v and edge (u, v) are added to G (unless they are already present). The weight of (u, v) is set to be the negative logarithm of the Sulston score between clone c_i and clone c_j.

The objective of the bipartite matching is to attempt to group together clone fragments that are located at the same location on the genome. Because of noise in the fingerprint data, some of the matched fragments might not represent the same region in the target genome. In the following steps, we try to eliminate as many false-positive matches as possible.

MFG pruning. In this step, some of the components of G that might represent more than one unique region in the target genome are split. The aim of this step to increase the accuracy of MFG so that later steps can compute an MTP with high coverage and less redundancy. Specifically, we examine all the connected components of G and mark those that satisfy at least one of the following conditions as candidates:

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

1. *Extra fragment.* Since each fragment of a clone maps to a unique location in the genome, two or more fragments of the same clone should not be in the same connected component. If a connected component contains multiple fragments of a clone, it is possibly falsely merged.
2. *Unmatched fragments.* Fragments that represent a common region must have similar lengths. If the difference between the length of the second shortest and the second longest fragment in the connected component is more than the tolerance value T, it is marked as a candidate to be split. We ignore the shortest and the longest fragments to allow two outliers per component.
3. *Weak overlap.* If a connected component contains at least one pair of clones that are very unlikely to overlap (i.e., have Sulston score of at least 1e-3 for MHS_MODULE or 1e-1 for MST_MODULE) then this connected component is likely to represent more than one unique region in the target genome.

For each candidate component, we further check whether the criteria above are met even though the component could represent a unique region in the genome. A connected component might be incorrectly marked to be split for various reasons. For instance, if a clone fragment is reported more than once in fingerprinting stage then these multiple fragments would appear in the same connected component, which then satisfies condition #1 above. We mark a candidate component as noncandidate if the average weight of edges that would be removed to split the component is much higher (i.e., 1.5-fold) than $-\log(C)$. We would like to split each remaining candidate components so that fragments of nonoverlapping clones are no longer in the same component (since they do not represent the same genomic region). We aim to split the components by removing the weakest set of edges (i.e., minimum total edge weight) as these edges probably connect fragments of nonoverlapping clones. Thus, each remaining candidate component is partitioned using a min-cut algorithm.[15] After a component is partitioned into two subgraphs, we check both subgraphs against the conditions above and we partition them recursively until no more components satisfy the conditions.

Postprocessing: removing clones that are buried into multiple clones. Recall that the goal of the first step in the construction of the MFG is to bury clones into other clones to reduce the problem size. In this step, we reduce the problem size further by removing clones that are buried into multiple clones. For example, in Fig. 4 clone C is buried into $B \cup D$, thus C can be removed.

Fig. 4. A layout of five clones in a contig based on their actual genome coordinates.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

For each clone in a contig, we compute the ratio between the total length of its fragments in the MFG and the total length of its fragments. If at least B' percent of the total fragment length is covered in the MFG then we declare that clone buried. However, there is a complication. It is possible that two clones are *mutually buried* and, therefore, we cannot remove both. To solve this problem, we first identify all possible candidates that can be buried into multiple clones. Then, we examine each connected component U of G, and save $N(U)$ in a list L if all clones in $N(U)$ are candidates to be buried.

All candidates that do not exist in any $N(U) \in L$ can be buried. However, we need to make sure that at least one clone in each $N(U) \in L$ is not buried; otherwise, the region in the target genome that is represented by U will not be covered. So the task is to remove as many candidate clones as possible with the constraint that at least one clone from each cloneset $N(U) \in L$ is kept. This problem is called *minimum hitting set* and it is NP-complete (polynomial reduction from the vertex cover problem[16]). Here, we solve it suboptimally using a greedy approach.[16] At each iteration, we

1. select the clone c that occurs in the maximum number of clone sets in L,
2. remove all clone sets that contain c,
3. save c into minimum hitting set H, and
4. repeat.

The iterative process terminates when the list L becomes empty.

Buried clones are removed from the MFG permanently. If this removal introduces components with only one node, they are also removed from the MFG permanently. The algorithm is summarized in Fig. 5.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

Algorithm 1

Input: Set of clones $\mathbb{C}$ in the contig
Input: $G = (V, E)$ {MFG of $\mathbb{C}$}
Output: G {Graph obtained by removing clones that are buried into multiple clones}
A={} {List of candidate clones}
for all clones $c \in \mathbb{C}$ **do**
  $m_c = 0$ {number of times c has a fragment in G}
  for all connected components $U \subseteq V$ **do**
    if $c \in N(U)$ **then**
      $m_c = m_c + 1$
  $r_c = m_c/|c|$ {$|c|$ is the number of fragments in c}
  if $r_c \geq B'$ **then**
    Add c to A
$L = \{\}$ {Set of clone sets}
for all connected components $U \subseteq V$ **do**
  if $N(U) \subseteq A$ **then**
    $L.insert(N(U))$
H =Solve Minimum Hitting Set(L)
for all clones $c \in A - H$ **do**
  for all nodes $v \in V$ **do**
    if $N(v) = c$ **then**
      Remove v from G
Remove connected components of size 1 from G

Fig. 5. Sketch of the algorithm that removes clones that are buried into multiple clones.

2.1.2 Selecting Essential MTP Clones

Clones that have to be selected as MTP clones are called *essential* clones. More specifically, if the total length of fragments that are not present in the MFG is at least 30 percent of the total length of all fragments of a clone then that clone is marked as essential. When a clone is marked as essential, it is immediately stored in the MTP and ignored for further analysis by removing all of its fragments from the MFG.

2.1.3 MHS_MODULE: Computing the Tiling Path by Solving the Minimum Hitting Set Problem

Recall that the MTP can be computed by selecting the smallest set of clones that cover all connected components of an ideal MFG. This problem is known as the *minimum*

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

hitting set problem. Since, we do not necessarily have an ideal MFG, set of clones that cover all connected components of the MFG will produce a preliminary tiling path. We solve the minimum hitting set problem suboptimally using a greedy approach as described above (see section *postprocessing: removing clones that are buried into multiple clones*).

2.1.4 MST_MODULE: Solving the MTP via Minimum Spanning Tree

From this point onward, all the clones in the physical map that do not belong to the preliminary tiling path are disregarded. A new MFG G=(V,E) is constructed only on the clones in the preliminary tiling path. An overlap graph is built from G, and then the minimum spanning tree (MST) of the overlap graph is computed to order the clones. Finally, the shortest path from the first clone to the last clone in the ordering is computed. Clones on this path constitute the final MTP. Here are the details.

First, MST_MODULE attempts to order the clones in the preliminary tiling path. For this purpose, a weighted overlap graph $G_o = (V_o, E_o)$ is constructed for each contig, where V_o is the set of preliminary tiling path clones in each contig and $E_o = \{(u, v) | \exists U \in G$ suchthat $(u, v) \subseteq N(U)\}$. The weight of edge is again the negative logarithm of the Sulston score between u and v.

Fig. 2b shows the overlap graph for the MFG in Fig. 2a. For example, there is an edge between clones K and L because their fragments occur together in at least one connected component in the MFG (see Fig. 2a).

By computing the MST of an edge-weighted overlap graph where the edge weights are proportional to the overlap size we obtain an ordering of nodes in the graph.[17] Although the correlation between Sulston score and overlap size is not perfect, the MST still gives very accurate ordering because the clones originate from the preliminary tiling path and not from the original physical map. Recall that in the preliminary tiling path a clone is not expected to overlap too many clones with similar overlap size.

According to our experiments, the MST of G_o is usually a path. When the MST is not a path, the relative ordering of some of the clones may not be determined. However, this is not a serious problem if we can detect a pair of overlapping clones that cover the group of clones whose order is undetermined. Because of this overlap, clones with unknown relative ordering do not need to be in the MTP; hence, they can be discarded in the analysis.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

Once the MST of G_o is computed, we pick the longest path P in the tree. If there is more than one such path, we select the path with the smallest total weight. The rationale is to minimize the total overlap size between consecutive clones, and thus select the path with highest coverage. The clones in P cover almost the whole contig, but they may not be an MTP. To find the MTP, the path P must be augmented with high confidence overlap edges and a shortest path must be found.

To minimize the possibility of adding false-negative and false-positive edges to P, we use several criteria based on the Sulston score and the MFG. The details of this algorithm are shown in Fig. 6. After augmenting P, the shortest path from u_1 to $u_{|P|}$ (in terms of number of hops) is computed. All nodes in this path are chosen as the MTP clones of this contig.

Algorithm 2

1: **Input:** The longest path $\mathbb{P}$ from the MST of G_O
2: **Input:** Clones $u_i \in \mathbb{P}$, $1 \leq i \leq |\mathbb{P}|$, where i is the order of u
3: **Input:** $G = (V, E)$ {MFG of the preliminary MTP clones in the contig}
4: **Output:** Augmented path $\mathbb{P}$
5: **for** $d = 2$ to $|\mathbb{P}| - 1$ **do**
6: **for** $i = 1$ to $|\mathbb{P}| - d$ **do**
7: Check if $S(u_i, u_{i+d-1}) \leq C$
8: Check if $S(u_{i+1}, u_{i+d}) \leq C$
9: Check if there exists a connected component U of G such that $\bigcup_{j=i}^{i+d} u_j \subseteq U$
10: Check if $S(u_i, u_{i+d}) \leq C$
11: **if** All conditions are true **then**
12: Add (u_i, u_{i+d}) to $\mathbb{P}$.

Fig. 6. Sketch of the algorithm that detects overlapping clones in an ordered clone list. At each iteration four conditions are checked to determine if u_i and $u_{\{i+d\}}$ are overlapping where u_i, $1 \leq i \leq |P|$, is the ith clone in P. All conditions have to be true to add edge $(u_i, u_{\{i+d\}})$ to P. In lines 7 and 8, we check whether the clone pairs $u_i, u_{\{i+d-1\}}$ and $u_{\{i+1\}}, u_{\{i+d\}}$ are overlapping. If at least one of these pairs do not overlap then u_i and $u_{\{i+d\}}$ cannot be overlapping (assuming that no clone is completely contained in another clone). In line 9, we check if clones $u_i, u_{\{i+1\}}, \ldots, u_{\{i+d\}}$ have fragments together in at least one connected component of G. If u_i and $u_{\{i+d\}}$ are overlapping then $u_i, u_{\{i+1\}}, \ldots, u_{\{i+d\}}$ have to be overlapping, and therefore they should share at least one fragment in G. At the end, we check if $S(u_i, u_{\{i+d\}}) \leq C$.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

2.2 Implementation

FMTP is implemented in C++ and Perl. It uses the Boost C++ library for graph-based functions. FMTP has been tested on Linux and Mac OS X platforms. The source and documentation of FMTP is freely available.[1]

2.3 Data Sets

Real fingerprint data for rice and barley. We used genomic data of two plants, namely, barley and rice, to compare our software to FPC.

First, we constructed the physical maps using FPC's physical mapping construction module and our compartmentalized assembler method[19] on real restriction fingerprints of rice[2] and barley[3] BACs.[18] The rice physical map that we used contains 22,486 clones, 2,070 contigs, and 2,593 singletons; and it is a subset of the publicly available rice physical map.[4] Specifically, our map contains only the subset of clones whose BESs could be uniquely mapped to the genome. Details of this procedure were given in.[19] The barley physical map contains 72,052 clones, 10,794 contigs, and 10,598 singletons.

Then, we ran FPC and FMTP on both physical maps and obtained their MTPs. To objectively measure the quality of the MTPs, we used genomic coordinates available for rice clones (but not for barley because its genome has not been sequenced yet).

Synthetic rice physical map based on in silico restriction fingerprint data. To compare the performance of both FPC and FMTP at ideal conditions, we generated a physical map of rice based on *in silico* restriction fingerprint data. We used the FSD tool[20] to generate *in silico* restriction fingerprints of rice BACs whose genomic coordinates are available. Then, we used FPC's physical mapping construction module to generate the physical map and pruned contigs that did not represent a contiguous region in the genome. For each contig, we kept only the clones that represent the longest contiguous region in the genome. In the synthetic rice physical map there are 26,469 clones, 1,132 contigs, and 1,004 singletons.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

SECTION 3

Results

Parameters. FPC and FMTP have several parameters. Depending on the fingerprinting method (i.e., agarose or high information content fingerprinting (HICF)), FMTP provides default values for its parameters. Using values for these parameters close to the defaults is crucial to obtain good performance. For example, by default, MHS_MODULE uses a low C value (1e-10 for agarose or 1e-40 for HICF). Since MHS_MODULE processes the original contigs, which usually contain many clones, a higher value of C would introduce many false-positive overlaps. For the synthetic rice physical map, reducing C to as low as 1e-24 improved the coverage. MST_MODULE uses a high C value (1e-2 for agarose or 1e-10 for HICF) to detect shorter overlaps between clones. According to our experiments, B should be at least 80 to avoid false-positive buried clones.

We have generated a large number of MTPs for each physical map using both tools with several parameter choices, however we only recorded the best possible MTP for a given size (i.e., number of clones). More specifically, given two MTPs M_i and M_k obtained by choosing different parameters and assuming without loss of generality that the size of M_i is greater than the size of M_k, we will keep M_i if the coverage of M_i is greater than coverage of M_k, and we will keep M_k if the number of redundant clones in M_k is smaller than the number of redundant clones in M_i. Consequently, as the size of the MTP increases, the coverage and number of redundant clones reported in the experimental result will always increase monotonically.

To have a fair comparison between FMTP and FPC, we used the same B and T values used during the construction of the physical map (B=90, T=7 for rice, and T=3 for barley). We set C in the range [1e-10, 1e-24], and [9e-2, 5e-4] for the MHS_MODULE and MST_MODULE, respectively. We set B' to values from 80 to 90 for MHS_MODULE and MST_MODULE. The parameter values for each run of FPC and FMTP are listed in Supplementary Tables S1 and S2, respectively.

"Perfect" and "Random" tiling path. To establish the range of possible coverage for a fixed number of clones in the computed MTPs, we declared the *random* tiling path (TP) as the lowest possible quality, and the *perfect* TP as the highest possible quality.

The "perfect" TP was obtained as follows, assuming that the genomic coordinates of all clones were known. For each contig, we first build a directed interval graph where each

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

node is a clone and there is an edge between two clones if their coordinates are overlapping. Buried clones are disregarded. The perfect TP is the set of clones in the shortest (unweighted) path from the leftmost clone to the rightmost clone. The number of clones of this TP and its genome coverage was recorded. To have a fair comparison between the computed MTP and the perfect TP, we also limited the number of clones in the perfect TP by the number of clones in the computed MTP for each contig.

A "random" TP was obtained by making completely random choices for each contig. The total number of TP clones selected randomly for each contig is again matched to the total number of clones selected by either FPC or FMTP for that contig. We compute ten random TPs for each computed MTP and report the average coverage.

Evaluation metrics. We evaluate each MTP based on two metrics, namely *contig-wise coverage* and *number of redundant clones*. The contig-wise coverage of a contig is the ratio between the coverage of its MTP clones and the coverage of all clones in the contig. The contig-wise coverage is computed for each contig and then an overall score is computed as the weighted average of contig-wise coverage, where the weight is the size of the contig. Contig-wise coverage is the essential evaluation metric because both FPC and FMTP aim to achieve 100 percent coverage with as smallest set of clones as possible.

Recall that an MTP clone is called *redundant* if it is completely covered by one or more MTP clones in the same contig. To determine the number of redundant clones, we compute the perfect TP of the contigs based on the original MTP clones in the contigs. The difference between the number of original MTP clones and the number of clones in the perfect TP gives the number of redundant clones.

The benchmark results show that FMTP outperforms FPC in terms of coverage (see Fig. 7) and number of redundant clones (see Fig. 8). For instance, for the synthetic rice physical map, the biggest MTP by FMTP has about 10.5 percent higher coverage than the biggest MTP by FPC. In the real map, the average coverage for the MTPs produced by FPC and FMTP are 86.2 and 91.7 percent, respectively. In the synthetic map, the average coverage for FPC and FMTP increases by about 0.2 and 3.3 percent, respectively (see Fig. 7). In the real map, the smallest MTP obtained by FMTP has about 1 percent higher coverage than the biggest MTP by FPC even though the size of the former MTP is about 11 percent smaller than the size of the latter MTP (see Fig. 7b).

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

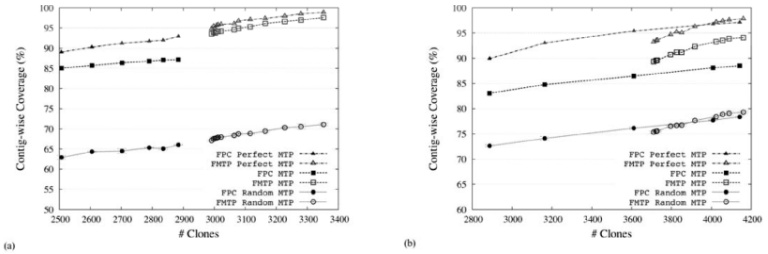

Fig. 7. Contig-wise coverage of several MTPs generated by FPC and FMTP, and their corresponding "perfect" and "random" tiling paths in the rice physical map. (a) MTPs of the synthetic map generated with *in silico* restriction fingerprints (b) MTPs of the map generated with real restriction fingerprint data.

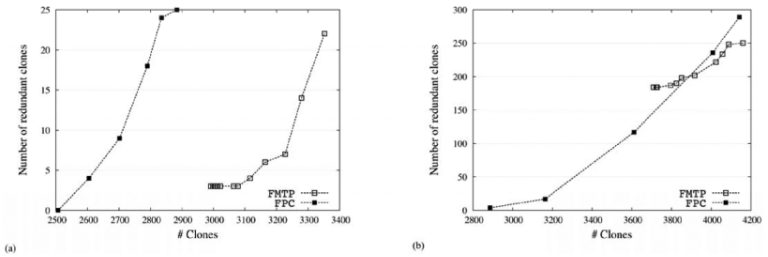

Fig. 8. Number of redundant clones in the MTPs in Fig. 7. (a) MTPs of the synthetic map generated with *in silico* restriction fingerprints (b) MTPs of the map generated with real restriction fingerprint data.

In general, FMTP tends to generate bigger MTPs than FPC. We also observe that the average MTP size for the synthetic map is lower than the average MTP size for the real map. The average MTP size by FPC is 2,721 for the synthetic map and 3,562 for the real map. MTPs generated by FMTP have an average size of 3,101 for the synthetic map and 3,898 for the real map (see Fig. 7).

The coverage for MTPs generated by FMTP is much closer to their corresponding perfect TP than FPC (see Fig. 7). FMTP generates MTPs whose coverage, on average, is 1.5 and 3.9 percent smaller than the coverage of perfect TP in the synthetic and real map, respectively. The coverage gap between perfect and actual MTPs by FPC is 4.8 and 8.3 percent in synthetic and real map, respectively.

Both tools generate MTPs with fewer redundant clones for the synthetic map than for the real map. The biggest MTP for the synthetic map has about 25 redundant clones, whereas the biggest MTP for the real map has about 250 redundant clones (see Fig. 8).

For the synthetic map, MTPs generated by FMTP have fewer redundant clones than FPC (see Fig. 8a). On the real physical map, small MTPs produced by FMTP have more redundant clones than similar size MTPs generated via FPC. However, medium-size and

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

large MTPs obtained with FMTP have fewer redundant clones than similar size MTPs generated via FPC (see Fig. 8b).

We also ran FMTP and FPC on the barley physical map generated by our group at the University of California, Riverside and several other institutions.[5] FPC generated MTPs that contain between 11,000 and 21,000 clones. When default values are used, FMTP generated MTPs that contain about 18,000 clones.

In terms of running time, both FMTP and FPC compute MTP in a few minutes.

SECTION 4

Discussion

FMTP generates MTPs with high coverage and a small number of redundant clones. When the physical map is accurate, FMTP can generate MTPs with coverage close to 100 percent. The key ingredient to have high-quality MTPs is to have high-quality restriction fingerprint data.

A few observations are in order. The average MTP size for the synthetic map is smaller than the average MTP size for the real map because there are fewer contigs in the synthetic map than in the real map. MTPs produced by FMTP are bigger than MTPs obtained via FPC for both the synthetic and the real map. Although we tried several combinations of parameter values, we could not reduce the MTP size computed by FMTP, probably because FMTP aims to maximize coverage by covering all the connected components in the MFG (i.e., regions in the contig).

Once a set of MTPs is generated with different parameters, one common question is how to decide which MTP to choose for further analysis. Our experimental results indicate that the coverage of the MTP increases linearly with its size, whereas the number of redundant clones in an MTP increases exponentially with its size. In general, to balance coverage and redundancy, one should select a medium-size MTP. If the quality of restriction fingerprint data is high, an MTP with maximum coverage can be selected because the expected redundancy from high-quality data will be low.

In the future, we plan to test the performance of FMTP on the HICF data. Since HICF allows to build high-resolution maps with less noise than from agarose fingerprinting,[21] we expect that FMTP would generate much higher quality MTPs.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

SECTION 5

Conclusions

In this paper, we presented FMTP, a novel tool to compute the MTP of a physical map that uses a two-step approach. In the first step, we solve a minimum hitting set problem on MFG of each contig heuristicly to generate a possibly redundant preliminary tiling path without compromising the coverage. In the second step, we order the clones in the preliminary tiling path by computing a minimum spanning tree of an overlap graph. Then, we employ a shortest path algorithm to compute the MTP. Our experimental results show that FMTP generates MTPs with significantly higher coverage than MTPs generated by the most commonly used software FPC, even using fewer MTP clones. As a consequence FMTP could substantially reduce the cost of clone-by-clone sequencing projects. FMTP runs under Linux and Mac OS X and is freely available.[6]

Acknowledgments

The authors thank Drs. Carol Soderlund and William Nelson for their suggestions about FPC. This project was supported in part by the US National Science Foundation under grants CAREER IIS-0447773, NSF DBI-0321756, and NSF ABI-1062301.

References

[1]E.D. Green, "Strategies for the Systematic Sequencing of Complex Genomes", *Nature Rev. Genetics*, vol. 2, pp. 573-583, 2001.

[2]M. Marra, T. Kucaba, M. Sekhon, L. Hillier, R. Martienssen, A. Chinwalla, J.m. Crockett, J. Fedele, H. Grover, C. Gund, W.R. McCombie, K. McDonald, J. McPherson, N. Mudd, L. Parnell, J. Schein, R. Seim, P. Shelby, R. Waterston, R. Wilson, "A Map for Sequence Analysis of the Arabidopsis Thaliana Genome", *Nature Genetics*, vol. 22, no. 3, pp. 265-270, 1999.

[3]J.D. McPherson, "A Physical Map of the Human Genome", *Nature*, vol. 409, pp. 934-941, 2001.

[4]R.L. Warren, D. Varabei, D. Platt, X. Huang, D. Messina, S.-P. Yang, J.W. Kronstad, M. Krzywinski, W.C. Warren, J.W. Wallis, L.W. Hillier, A.T. Chinwalla, J.E. Schein, A.S. Siddiqui, M.A. Marra, R.K. Wilson, S.J.M. Jones, "Physical Map-Assisted Whole-Genome Shotgun Sequence Assemblies", *Genome Research*, vol. 16, no. 6, pp. 768-775, June 2006.

[5]S.G. Gregory, M. Sekhon, J. Schein, S. Zhao, K. Osoegawa, C.E. Scott, R.S. Evans, P.W. Burridge, T.V. Cox, C.A. Fox, R.D. Hutton, I.R. Mullenger, K.J. Phillips, J. Smith, J. Stalker, G.J. Threadgold, E. Birney, K. Wylie, A. Chinwalla, J. Wallis, L. Hillier, J. Carter, T. Gaige, S. Jaeger, C. Kremitzki, D. Layman, J. Maas, R. McGrane, K. Mead, R. Walker, S. Jones, M. Smith, J. Asano, I. Bosdet, S. Chan, S. Chittaranjan, R. Chiu, C. Fjell, D. Fuhrmann, N. Girn, C. Gray, R. Guin, L. Hsiao, M. Krzywinski, R. Kutsche, S.S. Lee, C. Mathewson, C. McLeavy, S. Messervier, S. Ness, P. Pandoh, A.-L. Prabhu, P.

Saeedi, D. Smailus, L. Spence, J. Stott, S. Taylor, W. Terpstra, M. Tsai, J. Vardy, N. Wye, G. Yang, S. Shatsman, B. Ayodeji, K. Geer, G. Tsegaye, A. Shvartsbeyn, E. Gebregeorgis, M. Krol, D. Russell, L. Overton, J.A. Malek, M. Holmes, M. Heaney, J. Shetty, T. Feldblyum, W.C. Nierman, J.J. Catanese, T. Hubbard, R.H. Waterston, J. Rogers, P.J. de Jong, C.M. Fraser, M. Marra, J.D. McPherson, D.R. Bentley, "A Physical Map of the Mouse Genome", *Nature*, vol. 418, no. 6899, pp. 743-750, 2002.

[6]M. Krzywinski, J. Wallis, C. Gsele, I. Bosdet, R. Chiu, T. Graves, O. Hummel, D. Layman, C. Mathewson, N. Wye, B. Zhu, D. Albracht, J. Asano, S. Barber, M. Brown-John, S. Chan, S. Chand, A. Cloutier, J. Davito, C. Fjell, T. Gaige, D. Ganten, N. Girn, K. Guggenheimer, H. Himmelbauer, T. Kreitler, S. Leach, D. Lee, H. Lehrach, M. Mayo, K. Mead, T. Olson, P. Pandoh, A.-L. Prabhu, H. Shin, S. Tnzer, J. Thompson, M. Tsai, J. Walker, G. Yang, M. Sekhon, L. Hillier, H. Zimdahl, A. Marziali, K. Osoegawa, S. Zhao, A. Siddiqui, P.J. de Jong, W. Warren, E. Mardis, J.D. McPherson, R. Wilson, N. Hbner, S. Jones, M. Marra, J. Schein, "Integrated and Sequence-Ordered BAC- and YAC-Based Physical Maps for the Rat Genome", *Genome Research*, vol. 14, no. 4, pp. 766-779, Apr. 2004.

[7]C. Ren, M.-K. Lee, B. Yan, K. Ding, B. Cox, M.N. Romanov, J.A. Price, J.B. Dodgson, H.-B. Zhang, "A BAC-Based Physical Map of the Chicken Genome", *Genome Research*, vol. 13, no. 12, pp. 2754-2758, Dec. 2003.

[8]J.C. Venter, H.O. Smith, L. Hood, "A New Strategy for Genome Sequencing", *Nature*, vol. 381, no. 6581, pp. 364-366, 1996.

[9]Z. Frenkel, E. Paux, D. Mester, C. Feuillet, A. Korol, "LTC: A Novel Algorithm to Improve the Efficiency of Contig Assembly for Physical Mapping in Complex Genomes", *BMC Bioinformatics*, vol. 11, no. 1, pp. 584, 2010.

[10]W. Nelson, C. Soderlund, "Integrating Sequence with FPC Fingerprint Maps", *Nucleic Acids Research*, vol. 37, no. 5, pp. 36, Apr. 2009.

[11]S. Bozdag, T. Close, S. Lonardi, "Computing the Minimal Tiling Path from a Physical Map by Integer Linear Programming", *Proc. Eighth Int'l Workshop on Algorithms in Bioinformatics*, pp. 148-161, 2008.

[12]C. Soderlund, S. Humphray, A. Dunham, L. French, "Contigs Built with Fingerprints Markers and FPC V4.7", *Genome Research*, vol. 10, no. 11, pp. 1772-1787, Nov. 2000.

[13]J. Sulston, F. Mallett, R. Staden, R. Durbin, T. Horsnell, A. Coulson, "Software for Genome Mapping by Fingerprinting Techniques", *Computer Application Biosciences*, vol. 4, no. 1, pp. 125-132, Mar. 1988.

[14]J. Edmonds, R.M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems", *J. ACM*, vol. 19, no. 2, pp. 248-264, 1972.

[15]J. Hao, J.B. Orlin, "A Faster Algorithm for Finding the Minimum Cut in a Directed Graph", *J. Algorithms*, vol. 17, no. 3, pp. 424-446, 1994.

[16]M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness., 1979.

[17]Y. Wu, P.R. Bhat, T.J. Close, S. Lonardi, "Efficient and Accurate Construction of Genetic Linkage Maps from the Minimum Spanning Tree of a Graph", *PLoS Genetics*, no. 10, pp. 1000212, 2008.

[18]"A physical genetic and functional sequence assembly of the barley genome", *Nature*, vol. 491, no. 7426, pp. 711-716, 2012.

[19]S. Bozdag, T. Close, S. Lonardi, "A Compartmentalized Approach to the Assembly of Physical Maps", *BMC Bioinformatics*, vol. 10, no. 1, pp. 217, 2009.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.

[20]F.W. Engler, J. Hatfield, W. Nelson, C.A. Soderlund, "Locating Sequence on FPC Maps and Selecting a Minimal Tiling Path", *Genome Research*, vol. 13, no. 9, pp. 2152-2163, Sept. 2003.

[21]W.M. Nelson, A.K. Bharti, E. Butler, F. Wei, G. Fuks, H. Kim, R.A. Wing, J. Messing, C. Soderlund, "Whole-Genome Validation of High-Information-Content Fingerprinting", *Plant Physiology*, vol. 139, no. 1, pp. 27-38, Sept. 2005.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol 10, No. 2 (March 2013): pg. 352-360. DOI.