

# Clone-array pooled shotgun mapping and sequencing: design and analysis of experiments

**Miklós Csűrös**<sup>1</sup>      **Bingshan Li**<sup>2</sup>      **Aleksandar Milosavljevic**<sup>3</sup>  
csuros@iro.umontreal.ca      bli1@bcm.tmc.edu      amilosav@bcm.tmc.edu

<sup>1</sup> Département d'informatique et de recherche opérationnelle, Université de Montréal, CP 6128 succ. Centre-Ville, Montréal, Québec H3C 3J7, Canada.

<sup>2</sup> Human Genome Sequencing Center, Department of Molecular and Human Genetics, Baylor College of Medicine, Houston, Texas 77030, USA.

<sup>3</sup> Bioinformatics Research Laboratory, Department of Molecular and Human Genetics, and Program in Structural and Computational Biology and Molecular Biophysics, Baylor College of Medicine.

## Abstract

This paper studies sequencing and mapping methods that rely solely on pooling and shotgun sequencing of clones. First, we scrutinize and improve the recently proposed Clone-Array Pooled Shotgun Sequencing (CAPSS) method, which delivers a BAC-linked assembly of a whole genome sequence. Secondly, we introduce a novel physical mapping method, called *Clone-Array Pooled Shotgun Mapping* (CAPS-MAP), which computes the physical ordering of BACs in a random library. Both CAPSS and CAPS-MAP construct subclone libraries from pooled genomic BAC clones.

**Keywords:** sequencing, physical mapping, pooled shotgun sequencing.

## 1 Introduction

In a *hierarchical approach* to large genome sequencing, one first breaks many genome copies into random fragments. A *library* is constructed by cloning the fragments, typically as *Bacterial Artificial Chromosome* inserts (BACs). Some BACs in the library are selected for complete sequencing. Each selected BAC sequence is assembled individually using the shotgun method: a *subclone* library is prepared by cloning short fragments of the BAC. Subsequently, sequence *reads* are produced from a sufficient number of randomly chosen subclones. The reads are assembled algorithmically into the BAC sequence. An alternative to the hierarchical, or *clone-by-clone*, strategy is the *whole-genome shotgun* approach [12], which employs a few (essentially 1–3) subclone libraries prepared from the entire genome, without resorting to an intermediate BAC library. The main advantage of the whole-genome approach is that it eliminates the need to prepare tens of thousands of subclone libraries to sequence a mammalian genome. However, it is generally an inadequate strategy for finishing the assembly of such large repeat-rich genomes. For a review of contemporary sequencing methodologies, see, e.g., [8].

A new BAC-based sequencing strategy, called Clone-Array Pooled Shotgun Sequencing (CAPSS), was proposed recently [4]. CAPSS assembles the complete sequences of individual BACs as does the clone-by-clone approach, but requires a much smaller number of subclone library preparations. The strategy is currently being applied for the first time on a genome scale in the context of sequencing the honey bee genome. This paper provides the theory for the design and analysis of pooling-based genome projects. It also introduces the CAPS-MAP method for physical mapping, and transversal pooling designs for both CAPSS and CAPS-MAP, thereby laying the theoretical foundation for pooling-based genome-scale sequencing projects.

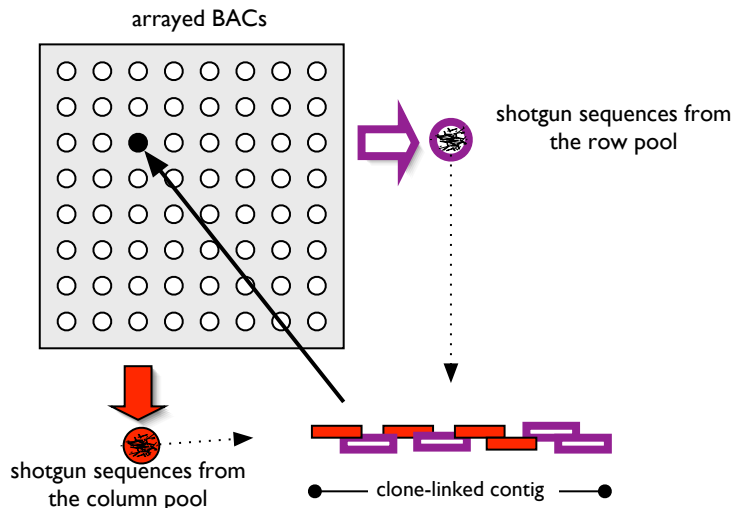


Figure 1: CAPSS strategy for arrayed BACs. DNA extracted from each clone is pooled together with other clones in the same row and column. Subclone libraries are prepared from the pools, and shotgun sequences are collected from the sublibraries. Sequences are assembled into contigs. If a contig contains sequences from a row and a column pool’s sublibrary, the contig is assigned to the BAC at the intersection of the row and the column.

In a clone-by-clone approach, BACs are sequenced independently: one subclone library is constructed for every clone. In contrast, DNA from BACs are pooled together in a CAPSS approach, and subclone libraries are prepared from the pools. A CAPSS experiment is designed so that the number of subclone libraries is much smaller than the number of clones, yet the pooling design enables the assembly of individual clone sequences. In what follows, by *pooled shotgun* (CAPS) sequences we mean shotgun sequence reads collected from a subclone library that was constructed using pooled BACs. For the computational aspects of sequence assembly, pooled shotgun sequences are random subsequences originating from a set of clone sequences.

The original CAPSS proposal [4] relied on a simple rectangular design defined by an array layout of BACs (Figure 1). The pools correspond to the rows and columns. An array layout reduces the number of shotgun library preparations to the square root of the number of BACs when compared to clone-by-clone sequencing. This reduction can be important in case of a mammalian genome, for which even a minimally overlapping tiling path contains between twenty and thirty thousand clones [10].

This paper has two goals. First, after pointing out some shortcomings of the original CAPSS proposal, we propose algorithmic and experimental improvements that make CAPSS a viable option for sequencing a set of BACs. Specifically, we apply transversal pooling designs to increase the accuracy of CAPSS, which we previously developed for the PGI method of comparative physical mapping that also uses pooled shotgun sequencing [6]. We provide the first probabilistic model of CAPSS sequencing progress. The model leads to theoretical results supporting previous, less formal arguments on the practicality of CAPSS.

The paper’s second goal is to introduce the *Clone-Array Pooled Shotgun Mapping (CAPS-MAP)* method to detect clone overlaps in a random BAC library. The information on clone overlaps is used to compute the physical ordering of clones in the library, without requiring additional clone fingerprinting. CAPS-MAP operates in the same experimental framework as CAPSS. It needs only shotgun sequences, which makes it a cost-effective method that can be seamlessly integrated into a sequencing project with very little experimental overhead. We demonstrate the usefulness of CAPS-MAP for clone overlap detection with a probabilistic analysis. In addition to the theoretical results, we illustrate the method’s performance in a simulated project using the *Drosophila* genome assembly.

## 2 Transversal designs

It was proposed in [4] that CAPSS be used in hybrid projects, combining whole-genome shotgun (WGS) and pooled shotgun (CAPS) sequences. The motivation is that the pooled shotgun sequences can provide the localization information for the whole-genome shotgun sequences so that the latter can be used for a clone-linked assembly. After WGS and CAPS sequences from a set of pools are assembled into contigs, the contigs need to be mapped to individual BACs. There are a few challenges to contig mapping. We mention here three main problems: false negatives, ambiguities, and false mapping. A false negative refers to a situation where a BAC is not sampled in a pool it is included in, due to the low number of CAPS sequences collected. A false negative for a simple rectangular design means that no contigs can be mapped to the BAC. Ambiguities and false mappings are caused by overlapping clones, or more generally, by clones that have highly similar regions. The mapping of a contig is ambiguous if it is not possible to decide which clones the contig should be assigned to, in cases where two or more clone sets are equally likely choices for the mapping. False mapping occurs when an insufficient number of CAPS sequences are collected, and a contig that covers overlapping BACs gets assigned to the wrong clone or clone set.

One strategy used to overcome the mapping problems involves transversal pooling designs [6, 7]. For a transversal design with  $n$  pool sets, every clone is included in exactly one pool of each pool set, and any subset with two of those pools uniquely identifies the clone. Half of the pool sets are designated as column pools, and the other half as row pools to realize the design with the help of BAC arrays. Using a transversal double-array design (i.e., one with four pool sets), the same set of BACs is independently arrayed twice. Each of the two resulting arrays contains the same set of BACs. Thus, each BAC ends up being sampled in two column-pools and two row-pools. One of the arrays contains an arbitrary arrangement of BACs, while the other is “reshuffled” relative to the first. More generally, clones can be arranged on  $d$  reshuffled arrays using a transversal pooling design with  $n = 2d$  pool sets.

The number of arrays in a transversal design may be adjusted to allow unambiguous and correct contig mapping for any redundancy in a BAC library. Specifically, it can be shown [6, 7] that a  $d$ -array transversal design can accurately resolve BACs at up to  $(2d-1)X$  redundancy. We previously described and analyzed transversal designs in the context of pooled shotgun experiments [6] and compared their performance to other designs. Even though our analysis was performed for the Pooled Genomic Indexing (PGI) method in the context of comparative physical mapping, the results are generally valid for CAPSS and CAPS-MAP as well. Specifically, our results indicate that transversal designs reduce the frequency of false negatives and false mappings when compared to a simple rectangular design. Furthermore, when compared to other more complicated designs, they achieve an optimal balance between the number of shotgun library preparations and the frequency of contig mapping problems. Transversal designs also enjoy a practical advantage over more complicated combinatorial designs, in that they are readily implemented using existing automated clone arraying technologies.

## 3 Sequence assembly

This section analyzes CAPSS progress in a hybrid project that uses whole-genome and pooled shotgun sequences. CAPS sequences are collected using a transversal design with  $n$  pool sets, i.e.,  $n/2$  arrays. In order to derive a probabilistic model for such experiments, we introduce the following notations along with some standard simplifying assumptions. Assume that every clone has the same length  $L$  (100–200 thousand base pairs in practice), and that each shotgun sequence has the same length  $\ell$  (e.g., 500 bp). The WGS and CAPS sequences are combined and compared to each other to find overlaps between them. Overlapping sequences form *islands*. Islands with two or more sequences are *contigs*. An overlap between two shotgun sequences is detected if it is at least of length  $\vartheta\ell$  where  $0 < \vartheta \leq 1$ . Statistics for islands, and gaps between islands are well known [11, 13]. We are interested in statistics for *clone-linked contigs*, those that are assigned to BACs using the pooling information.

Let  $a$  be the coverage by CAPS sequences, i.e., if  $F_p$  CAPS sequences are collected, then  $a = \frac{F_p \ell}{NL}$  where  $N$  is the total number of clones. Let  $w$  denote the coverage by WGS sequences, i.e., if  $F_w$  WGS sequences are collected, then  $w = \frac{F_w \ell}{G}$  where  $G$  is the genome length. Notice that  $w = 0$  is possible. Here we consider the simplest case of assembling the sequence of a single clone that does not overlap with any other clone. Such a clone is covered by a total coverage of  $(a + w)$ . Although we concentrate on sequencing a particular clone, the transversal design allows the simultaneous sequencing of multiple, possibly overlapping clones by combining WGS sequences with CAPS sequences from many (or even all) pools. Regions of overlapping clones have higher coverage since they are covered by more CAPS sequences than a single clone. The sequencing of overlapping regions progresses thus faster than what is suggested by the statistics for a single clone. We examine the case of assigning contigs to overlapping BACs in §4. Two shotgun sequences from different pools suffice to assign a contig to a single BAC. In a practical setting, it may be advantageous to require more stringent criteria in order to avoid false mappings. Theorem 1 can be readily adapted for such criteria, albeit resulting in bulkier formulas.

Figure 2 compares different experimental designs based on Theorem 1 and simulations. Figure 2a shows that while more arrays increase the sequencing success, the improvements are very small after the second array. Notice that if the clones are selected from a minimally overlapping tiling path, then no part of the genome is covered by more than two BACs, and thus two arrays suffice for the unambiguous mapping of all contigs that cover clone overlaps. Figure 2b plots the N50 values. The N50 contig length is the value  $l$  such that half of the sequenced nucleotides belong to contigs of length at least  $l$ . The statistics for all designs converge to those of a non-pooled sequencing project as the coverage increases. In other words, the negative effects of pooling diminish and the project progresses just as without pooling: for example, at total coverage 4–5X, 99% of the clone is sequenced.

**Theorem 1.** *Let  $\sigma = 1 - \vartheta$  where  $\vartheta$  is the fraction of length two shotgun sequences must share in order for the overlap to be detected. Consider a BAC that does not overlap with other clones. Define  $c = w + a$ , the total coverage. Let  $X_1 = \frac{w+a}{c}$ ,  $X_2 = \frac{w}{c}$ , and  $Y_i = 1 - (1 - e^{-c\sigma})X_i$  for  $i = 1, 2$ .*

(i) *The expected number of clone-linked contigs covering the clone equals  $\frac{L}{\ell} c e^{-c\sigma} p_{\text{link}}$ , where*

$$p_{\text{link}} = \begin{cases} 1 - e^{-c\sigma} \left( n \frac{X_1}{Y_1} - (n-1) \frac{X_2}{Y_2} \right) & \text{if } w > 0; \\ \frac{1 - e^{-a\sigma}}{1 + \frac{1}{n-1} e^{-a\sigma}} & \text{if } w = 0. \end{cases} \quad (1)$$

(ii) *The expected number of shotgun sequences in a clone-linked contig is*

$$F_{\text{link}} = \begin{cases} \frac{e^{c\sigma}}{p_{\text{link}}} \left( 1 - e^{-2c\sigma} \left( n \frac{X_1}{Y_1} - (n-1) \frac{X_2}{Y_2} \right) \right) & \text{if } w > 0; \\ e^{a\sigma} + \frac{1 + \frac{1}{n-1}}{1 + \frac{e^{-a\sigma}}{n-1}} & \text{if } w = 0. \end{cases} \quad (2)$$

(iii) *Define  $F_{\text{no link}} = \frac{n \frac{X_1}{Y_1} - (n-1) \frac{X_2}{Y_2}}{n \frac{X_1}{Y_1} - (n-1) \frac{X_2}{Y_2}}$ , and  $\lambda_{\text{CBC}} = \frac{e^{c\sigma} - 1}{c} + \vartheta$ . The expected length of a clone-linked contig can be written as  $\ell \lambda_{\text{link}}$  where  $\lambda_{\text{link}}$  is bounded as*

$$\frac{\lambda_{\text{CBC}} - (F_{\text{no link}} \sigma + \vartheta)(1 - p_{\text{link}})}{p_{\text{link}}} \leq \lambda_{\text{link}} \leq \frac{\lambda_{\text{CBC}}}{p_{\text{link}}}. \quad (3)$$

Furthermore, when  $\mu = a/c$  is kept constant,  $F_{\text{no link}}$  increases monotonically with  $c$  and

$$\lim_{c \rightarrow \infty} F_{\text{no link}} = \begin{cases} \mu^{-1} \frac{(3n^2 - 3n + 1) - \mu(2n^2 - 3n + 1)}{(2n^2 - 3n + 1) - \mu(n^2 - 2n + 1)} & \text{if } w > 0; \\ \frac{n}{n-1} & \text{if } w = 0. \end{cases} \quad (4)$$

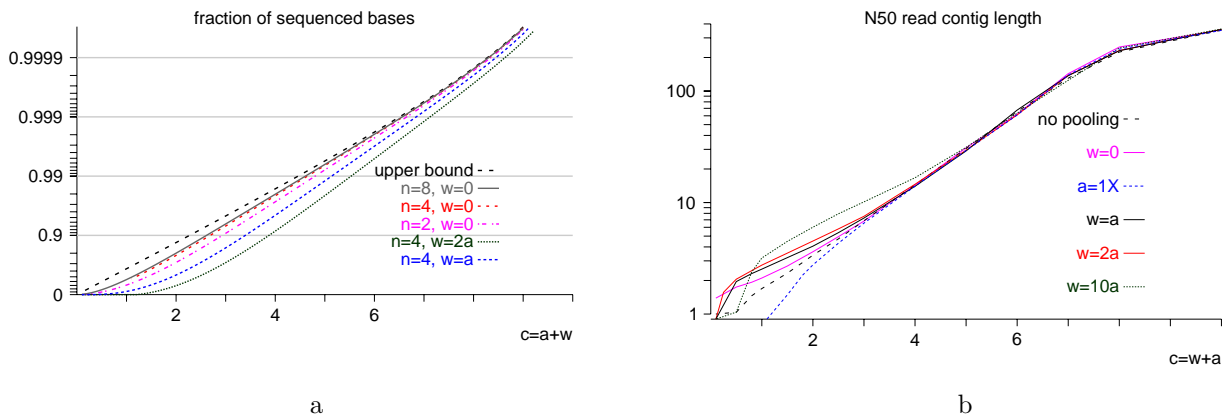


Figure 2: CAPSS (Theorem 1): sequencing progress. The left-hand side plots the fraction of bases covered by clone-linked contigs as a function of total coverage ( $c = a + w$ ) for different designs. Notice that the improvement from two arrays to four arrays ( $n = 4$  vs.  $n = 8$ ) is marginal. The right-hand side plots the N50 values for different designs with two arrays, as multiples of  $\ell$ . All values were calculated with shotgun sequence overlap detection  $\vartheta = 0.1$ . The N50 plot was obtained from simulation: each point is an average of 200 measurements.

The value  $\lambda_{\text{CBC}}$  in the theorem is the expected island length in a non-pooled sequencing project. It follows from Equation (3) that  $\lim_{c \rightarrow \infty} \lambda_{\text{link}} = \lambda_{\text{CBC}}$  when the ratio of CAPS sequences is kept constant, since  $\lim_{c \rightarrow \infty} p_{\text{link}} = 1$  by (1) and  $\lim_{c \rightarrow \infty} F_{\text{no link}} < \infty$  by (4). This limit result is not surprising given that every island can be linked to a clone with near certainty when the CAPS coverage is large.

## 4 Clone overlap detection

The key observation for this section is that a transversal design makes it possible to map a contig unambiguously to more than one BAC at once. Now, a contig that is mapped to two clones simultaneously can be viewed as evidence that the two clones overlap. Taking the idea further, an entire set of BACs can be tested for overlaps in this manner, which leads us to the Clone-Array Pooled Shotgun Mapping (CAPS-MAP) method that is described as follows. A redundant collection of random BACs covering a large genome is grouped into subsets of size  $q^2$ . Pooled shotgun sequence reads are collected from each clone group using a transversal design with  $d$  arrays of size  $q \times q$ . Partitioning into subsets may be dictated by the practical concerns of chemistry, biology and robotic automation. For array sizes that are multiples of 8 or 12 or both (yielding standard dimensions of a 96-well microtiter plate), such as  $q = 24$ , or  $q = 48$ , there exist known [5] transversal designs. A pooling design with a few ( $d = 2, 3, 4$ ) arrays suffice to compute the physical ordering of BACs in the library, depending on the library's redundancy and the array sizes. In addition to the CAPS sequences, WGS sequences are used to increase read contig lengths. The shotgun sequences are compared to each other to find the overlaps between them, and are assembled into contigs. Contigs that map unambiguously to more than one clone are taken as evidence that the clones overlap. See Figure 3 for an illustration. The clone overlap information can then be used to compute the physical ordering of the BACs in the library, and to select a minimal tiling path for complete sequencing, just as if the overlaps were detected using a fingerprinting scheme.

Theorem 2 considers the case of detecting an overlap between two clones in different clone groups. Similar analyses can be carried out for more general cases with more overlapping clones, or clones in the same clone group, resulting in more cumbersome formulas.

Figure 4 plots the overlap detection probabilities in a few scenarios with different amounts of CAPS and WGS sequences. Based on the figure, the probability of detecting an overlap increases exponentially toward 1 with the overlap length. The same exponential behavior is characteristic

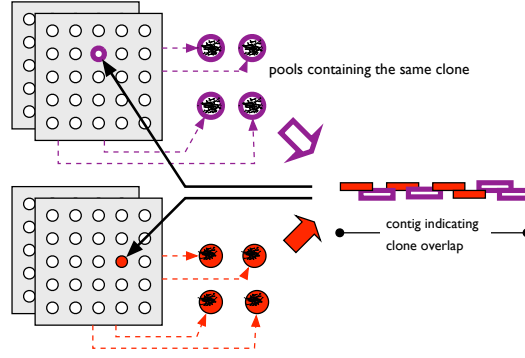


Figure 3: CAPS-MAP detects overlaps between clones by identifying situations where a read contig maps simultaneously to two clones. This figure illustrates a transversal pooling design with two clone groups and two arrays per group. The transversal design guarantees that the intersection of any two pools out of four possible for each BAC (two row and two column pools) uniquely identifies the BAC. Note that overlaps between clones on the same array can also be detected by a transversal design.

of clone anchoring methods for overlap detection [2]. Consequently, clone contig statistics for CAPS-MAP can be calculated using a clone anchoring model with an appropriate anchoring process intensity. Clone contig statistics can also be estimated using a fingerprinting model [11] by noticing that clone overlaps above a certain length are detected with near certainty. Figure 4 indicates that using 1X CAPS coverage and 2–5X WGS coverage, BAC overlaps of more than 20000 bp are detected almost certainly. While CAPS-MAP uses only the fact that a contig is mapped to multiple BACs, and not the actual contig sequence, the sequence information is used in the ensuing sequencing phase, and thus CAPS-MAP represents very little overhead in a genome sequencing project.

**Theorem 2.** *Let two clones from different clone groups share an overlap. Define  $c_2 = 2a + w$ , the total shotgun sequence coverage for the overlap. Define*

$$\beta_1 = \frac{w + (1 + \frac{1}{n})a}{c_2} \quad \beta_2 = \frac{w + a}{c_2} \quad \beta_3 = \frac{w + \frac{2a}{n}}{c_2} \quad \beta_4 = \frac{w + \frac{a}{n}}{c_2} \quad \beta_5 = \frac{w}{c_2};$$

$$\gamma_i = 1 - (1 - e^{-c_2\sigma})\beta_i \quad \text{for } i = 1, \dots, 5.$$

(i) *An apparent island in the overlap consisting of  $j > 0$  shotgun sequences is mapped to the two clones simultaneously with probability  $1 - q(j)$  where*

$$q(j) = 2n\beta_1^j - 2(n-1)\beta_2^j - n^2\beta_3^j + 2n(n-1)\beta_4^j - (n-1)^2\beta_5^j < 2n\beta_1^j. \quad (5)$$

(ii) *An apparent island covering the overlap is mapped to the two clones simultaneously with probability*

$$p_2 = 1 - e^{-c_2\sigma} \left( 2n \frac{\beta_1}{\gamma_1} - 2(n-1) \frac{\beta_2}{\gamma_2} - n^2 \frac{\beta_3}{\gamma_3} + 2n(n-1) \frac{\beta_4}{\gamma_4} - (n-1)^2 \frac{\beta_5}{\gamma_5} \right). \quad (6)$$

It is difficult to derive useful closed formulas for the probability of overlap detection. For example, based on Equation (6), the number of contigs in the overlap that are simultaneously mapped to the clones can be modeled as arrivals in a Poisson process with intensity  $c_2 e^{-c_2\sigma} p_2$ . For practical values of  $c_2$ , this model seriously underestimates the probability of overlap detection. The problem is similar to the one of using Lander-Waterman statistics [11] at high coverage levels (see [13] for a discussion). For a more suitable model, let  $G$  be the number of gaps entirely contained in the overlap, and number the islands from 0 to  $G$ . Let  $j_0, j_2, \dots, j_G$  denote the number of shotgun sequences in the islands. The probability that none of the islands can be mapped simultaneously to the two clones can be calculated

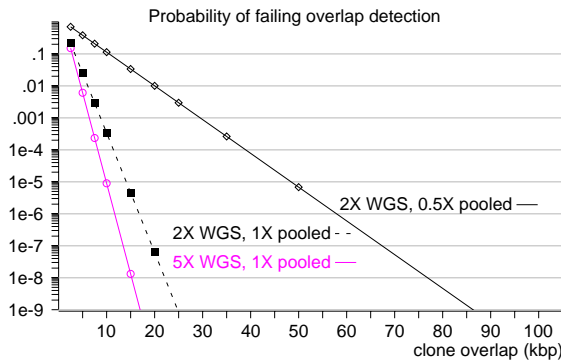


Figure 4: Clone overlap detection. The graph shows the probability of not detecting an overlap between two clones, as a function of the overlap size. The plots were calculated by a Monte-Carlo method using Theorem 2. All plots use  $\vartheta = 0.1$  for shotgun sequence overlap detection, and  $\ell = 500$  for shotgun sequence length.

as  $p_{\text{nomap}}(j_0, \dots, j_G) = \prod_{i=0}^G q(j_i)$ , where  $q(j)$  is defined by Equation (5). (Notice that  $G$  and the  $j_i$  are random variables.) We are interested in the expected value  $p_{\text{nomap}} = \mathbb{E}p_{\text{nomap}}(j_0, \dots, j_G)$ . In order to get a good assessment of CAPS-MAP performance, we found that it is best to use a Monte-Carlo estimation<sup>1</sup> of this expected value; see Figure 4. For an alternative, observe that the inequality of Equation (5) implies  $p_{\text{nomap}} < \mathbb{E}[\beta_1^R (2n)^{G+1}]$  where  $R$  is the number of sequences in the overlap, and thus  $R = \sum_{i=0}^G j_i$ . Based on this observation, we derived bounds (not shown here) that are useful for large values of  $c_2$  (e.g.,  $c_2 = 7$ ), but at lower coverages, this approach also underestimates the overlap detection probabilities significantly.

## 5 Ordering the BACs by using the BAC-contig incidence matrix

Our analyses so far have focused on detecting BAC overlaps via CAPS-MAP. This localized perspective was partly adopted to ease the theoretical analysis. In practice, mapping is performed based on a global clone-contig incidence matrix. The global approach exploits the dependencies in the collected data for increased accuracy. The algorithmic issues are very similar to those encountered in the context of STS-based physical mapping [9]. Define the mapping matrix  $\mathbf{M}$  in which the rows correspond to the BACs, the columns correspond to the contigs, and  $\mathbf{M}[i, j] = 1$  if contig  $j$  is linked to clone  $i$ , otherwise  $\mathbf{M}[i, j] = 0$ . We want to find the true ordering of the rows and columns, defined by their physical locations on the genome. Assume for a moment that the matrix is completely error-free, i.e., all contigs are correctly assembled, and all contig-clone overlaps are detected. It is not hard to see that the row and column permutations corresponding to the correct ordering result in a matrix  $\mathbf{M}'$  that satisfies the *consecutive ones* property (C1P) in the rows and the columns: for every row  $i$ , there exist  $a \leq b$  with  $\mathbf{M}'[i, j] = 1$  if and only if  $a \leq j \leq b$ , and the same property holds for the columns. (A sufficient condition ensuring row-wise (or column-wise) C1P is that if the left endpoint of a contig (or a clone) precedes the left endpoint of another one, the same holds for their right endpoints.) Finding such permutations is a well-known problem [3], and can be done in linear time. When the matrix is not error-free, one can use techniques introduced for STS-based physical mapping. In §6 we detail a method that relies on traveling salesperson tours.

<sup>1</sup> Specifically, for every overlap size considered, we carried out a number of simulated experiments. Each experiment used a fixed number of shotgun sequences  $R$  placed randomly in the overlap, and produced an instance of a  $(j_0, \dots, j_G)$  vector, for which  $p_{\text{nomap}}(j_0, \dots, j_G)$  was calculated using Equation (5). The average of these values was used to estimate  $p_{\text{nomap}}$ . The average was weighted with the probabilities of different  $R$  values, given by a Poisson distribution. The set of  $R$  values was chosen so that it provided a sufficient accuracy for the weighted average estimate. For every  $R$ , ten thousand experiments were performed.

Minimum bactig size	Genome covered	Number of BACs in bactigs
2	97.1%	2758
5	94.9%	2714
10	88.5%	2565
20	77.9%	2284
30	65.0%	1945
51	50.9%	1521
60	40.3%	1195

Table 1: Statistics for simulated *Drosophila* assembly. This table details the genome and BAC library coverage by bactig sizes. More than half of the genome is covered by bactigs with at least 51 BACs in them, defining the N50 statistic for the clone map.

## 6 CAPS-MAP simulation of *Drosophila* assembly

We tested the CAPS-MAP approach by simulating the assembly of the *Drosophila melanogaster* genome. One of the main goals of the simulated assembly was to predict the performance a hybrid approach combining WGS and CAPS sequences in a project that closely resembles the setup of the honey bee genome project’s (<http://www.hgsc.bcm.tmc.edu/projects/honeybee/>), currently pursued at the Human Genome Sequencing Center (HGSC) of Baylor College of Medicine.

Concatenating all the *Drosophila* genome sequence (Release 2.5, 112.6 million bases), 2880 BAC sequences were generated by randomly picking their locations and lengths. The mean BAC insert length was 150 kbp, and its standard deviation was 500 bp. The resulting random BAC library provides 3.6X coverage of the genome. BACs were arrayed by first partitioning them into 5 groups, and then using a two-array  $24 \times 24$  transversal design for each group. Every BAC was covered by 1.2X CAPS sequences: 0.4X per pool on the first array and 0.2X per pool on the second (reshuffled) array. In addition, WGS sequences were produced at 4X genome coverage. The shotgun sequences were generated using the program `wgs-simulator` (written by K. James Durbin), which mimics shotgun sequence collection realistically by relying on sequence quality files produced in sequencing projects.

Shotgun sequences were assembled into contigs using the Atlas suite of genome assembly tools (developed at the HGSC) and Phrap (<http://www.phrap.org/>). A contig was mapped to a clone if it contained sequences from all four clone pools. Contigs that mapped to more than one BACs provided the evidence of BAC overlaps. BACs were grouped into maximal overlapping sets, or *bactigs*.

We compared the overlap graphs to assess CAPS-MAP overlap detection. The vertices of the overlap graphs are the BACs, and two BACs are connected if there is an overlap between them. The true overlap graph for the original BACs contains 2880 vertices, and 10992 edges in 66 graph components. The overlap graph calculated from the bactigs has 9193 edges in 110 components. Among its edges, 8527 (93%) are correct, and 2465 (22%) of the true overlaps are not discovered. The median length of detected overlaps is 87 kbp, and the median length of undetected overlaps is 42 kbp. There are 666 edges that correspond to no real overlaps. The vast majority of these “false positives” are instances when a long read contig links several BACs, which do not always overlap pairwise. All but two of the CAPS-MAP bactigs are true overlapping sets of BACs. CAPS-MAP links the assembled contigs to BACs correctly even in these two bactigs: the source of the error is the read contig assembly. Table 1 shows statistics on the bactig sizes and genome coverage.

BACs were ordered within each bactig. For every bactig, an overlap matrix  $\mathbf{M}$  was calculated, in which the rows correspond to the bactig’s clones, the columns correspond to the contigs linked to at least one bactig clone, and  $\mathbf{M}[i, j] = 1$  if contig  $j$  is linked to clone  $i$ , otherwise  $\mathbf{M}[i, j] = 0$ . The following traveling salesperson (TSP) formulation is used to find the correct column permutation. We search for a tour in a graph, in which every vertex corresponds to a contig (and thus a column), with an additional vertex  $u_0$ . The weight of an edge between vertices  $u$  and  $u'$ , corresponding to contigs  $j$



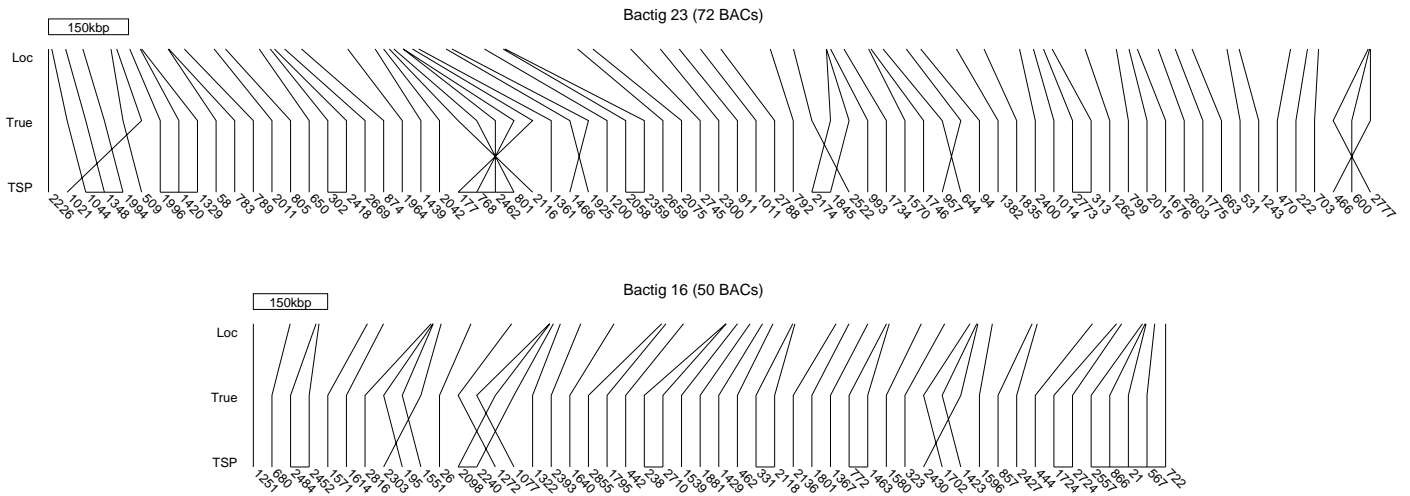


Figure 5: Correctness of BAC ordering in *Drosophila* simulation. The top (**Loc**) of each graph shows the relative physical location of each BAC, the middle (**True**) shows the correct BAC order, and the bottom (**TSP**) shows the TSP order, and the BAC identifiers. Identical BACs are connected in order to display the differences between the two permutations. The order of BACs at the bottom is not resolved when they are connected with a horizontal line. By resolving them optimally, *bactig 23* produces the order of 72 BACs with 12 breakpoints and *bactig 16* orders 50 BACs with 9 breakpoints. (Breakpoints are neighbors in the TSP order that are not neighbors in the true order.)

and  $j'$ , is the number of rows in which they differ:  $w(u, u') = \sum_i \chi\{\mathbf{M}[i, j] \neq \mathbf{M}[i, j']\}$ , where  $\chi\{\cdot\}$  is the indicator function. The weight of an edge between  $u$  and  $u_0$  is the sum of ones in the column  $j$  that corresponds to  $u$ :  $w(u, u_0) = \sum_i \mathbf{M}[i, j]$ . Now, a Hamilton path with the minimum weight in this graph gives the best column permutation in the sense that it minimizes the number of gaps between blocks of ones within rows [9]. The best row ordering could be found in an analogous manner, but we used a simpler method which worked better in practice. Clones are ordered relatively to the contig order by placing clone  $B$  before  $B'$  if the first contig  $B$  is linked to is before the first contig  $B'$  is linked to, or if their first contigs are identical but  $B$  has its last contig before  $B'$ .

We used the `concorde` program [1] to solve the TSP instances. The resulting row permutation is then further analyzed to find clones, for which the permutation arbitrarily enforces an order. Specifically, if consecutive rows of the permuted matrix  $\mathbf{M}'$  are identical, then the order of the corresponding clones is not resolved. Subsequently, we compared the TSP orders to the true orders, which is known since the BAC sequences are generated artificially. Figure 5 shows the outcome of the comparison for two *bactigs*. The TSP order is very close to the true order.

## 7 Discussion

The experimental expedience of shotgun sequencing has been essential for the success of genome-scale sequencing projects in the past decade. The power of the concept comes from the now established fact that the loss of information about read localization incurred by random subcloning can be largely recovered in the assembly step using sequence information. Clone pooling is similar in spirit to shotgun sequencing in that it introduces experimental expedience by dramatically reducing the number of subclone library preparations. The clone pooling step leads to a temporary loss of information about localization of shotgun sequences on individual BAC clones. We have demonstrated that sequence information can be used to successfully recover most of the information lost in pooling.

Our analyses presented here indicate the theoretical feasibility of the CAPS-MAP method and provide guidance for the design of genome-scale CAPS-MAP experiments. In particular, our analysis indicates that transversal pooling designs can accommodate high levels of clone redundancy and perform well even at low levels of shotgun sequence coverage of clone pools.

Practical biological and technical considerations may set a limit to the array size. In case of large genomes, the limitations may imply that the set of BACs is partitioned and that pooling is applied separately to individual subsets. This results in a lower clone redundancy within individual arrays and a larger number of pools. Our analysis allows for the partitioning of clones. It also allows for the possibility of including whole-genome shotgun sequence reads. It thus covers realistic and practical scenarios of the CAPSS and CAPS-MAP methods' application.

ACKNOWLEDGEMENTS. We are grateful to Richard Gibbs and George Weinstock for sharing pre-publication information on CAPSS and for useful comments. Our discussion of computing CAPS-MAP overlap detection probabilities has greatly benefited from conversations with Luc Devroey and Michael Waterman. This work was supported by grants RO1 HG02583-01 from NHGRI at the NIH, U01 RR18464 from the NCCR, and 250391-02 from the NSERC.

REMARK. The proofs for Theorems 1 and 2 were omitted because of space constraints. They can be found in a more detailed version of this paper at <http://www.br1.bcm.edu/>.

## References

- [1] Applegate, D., R. Bixby, V. Chvátal, and W. Cook. Concorde 99.12.15 release, 1999. <http://www.math.princeton.edu/tsp/concorde.html>.
- [2] Arratia, R., E. S. Lander, S. Tavaré, and M. S. Waterman. Genomic mapping by anchoring random clones: A mathematical analysis. *Genomics*, 11:806–827, 1991.
- [3] Booth, K. S. and G. S. Lueker. Testing for the Consecutive Ones Property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comp. Sys. Sci.*, 13:335–379, 1976.
- [4] Cai, W.-W., R. Chen, R. A. Gibbs, and A. Bradley. A clone-array pooled strategy for sequencing large genomes. *Genome Res.*, 11:1619–1623, 2001.
- [5] Colbourn, C. J. and J. H. Dinitz (Eds.) *The CRC Handbook of Combinatorial Designs*. CRC Press, 1996.
- [6] M. Csűrös and A. Milosavljevic. Pooled genomic indexing (PGI): mathematical analysis and experiment design. In *Proc. Workshop on Algorithms in Bioinformatics*, 10–28. Springer-Verlag, 2002.
- [7] Du, D.-Z. and F. K. Hwang. *Combinatorial Group Testing and Its Applications*. World Scientific, 2000.
- [8] Green, E. D. Strategies for the systematic sequencing of complex genomes. *Nat. Rev. Genet.*, 2:573–583, 2001.
- [9] Gusfield, D. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.
- [10] IHGSC. Initial sequencing and analysis of the human genome. *Nature*, 609(6822):860–921, 2001.
- [11] Lander, E. S. and M. S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2:231–239, 1988.
- [12] Weber, J. L. and E. W. Myers. Human whole-genome shotgun sequencing. *Genome Res.*, 7:401–409, 1997.
- [13] Wendl, M. C. and R. H. Waterston. Generalized gap model for bacterial artificial chromosome clone fingerprint mapping and shotgun sequencing. *Genome Res.*, 12:1943–1949, 2002.