Construction of Physical Maps From Oligonucleotide Fingerprints Data

Guy Mayraz *　　　Ron Shamir †

Abstract

A new algorithm for the construction of physical maps from hybridization fingerprints of short oligonucleotide probes has been developed. Extensive simulations in high-noise scenarios show that the algorithm produces an essentially completely correct map in over 96% of trials. Tests for the influence of specific experimental parameters demonstrate that the algorithm is robust to both false positive and false negative experimental errors. The algorithm was also tested in simulations using real DNA sequences of *C. Elegans, E. Coli, S. Cerevisiae* and *H. Sapiens*. To overcome the nonrandomness of probe frequencies in these sequences, probes were preselected based on sequence statistics and a screening process of the hybridization data was developed. With these modifications, the algorithm produced very encouraging results on real DNA sequences.

1 Introduction

Physical mapping is the process of determining the relative position of landmarks along a genome section. The resulting maps are used as a basis for DNA sequencing, and for the isolation and characterization of individual genes or other DNA regions of interest (e.g., transcripts and regulatory sequences)[11]. The construction of a high-resolution sequencing-ready physical maps for *H. Sapiens* and other organisms is still one of the top priorities of the Human Genome Project. The mapping process involves the production of *clones*, i.e., cloned pieces of DNA, each representing a chromosomal segment. The mapping itself is the reconstruction of the order of and physical distance between the clones.

Physical mapping using hybridization fingerprints of short oligonucleotides was first suggested by Poustka et al. [13]. In this technique short labeled DNA sequences, or *probes*, attach, or *hybridize*, to positions along the target DNA matching their own DNA sequence. The probes are non-unique, i.e., they occur at many points along the genome, and typically hybridize with 10%-50% of the clones. The subset of the probes that hybridized with a clone is its *fingerprint*. Overlapping clones can be identified by their similar fingerprints, and this overlap information facilitates the construction of the clones physical map. Discussions and analysis of this approach, including the use of Bayesian statistics were given in [13, 10, 8]. Further analysis and algorithms were provided in [5] with an emphasis on avoiding gaps, and in [4] using annealing. The problem received a detailed formal analysis by Alizadeh et al. [1]. Local improvement methods were used to solve the problem in a noise free scenario. As far as using the method in practice, we are only aware of two studies: A cosmid map was constructed manually by Craig et al. [3] for the Herpes Simplex Virus (HSV-I) genome, but the probes were almost unique (this data was also analyzed in [4]). Hoheisel et al. [6] used short probes as a check for the order of unique probes in a map of the fission yeast *S. pombe*. The lack of good algorithms for map construction was a major obstacle so far for using the method in large scale experiments (H. Lehrach, private communication).

Physical mapping using non-unique probes shares with the popular mapping techniques that use unique probes the much simpler handling requirements, as well as the ability to analyze many clones in parallel. It also enjoys the insensitivity to repeat sequences of restriction digestion. Advantages over mapping with unique probes include: (1) Probe generation is straightforward and much cheaper, (2) The num-

*Department of Computer Science, Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978 ISRAEL. Current address: Gatsby Computational Neuroscience Unit, University College London, 17 Queen Square, London WC1N 3AR, U.K. gmz@gatsby.ucl.ac.uk URL: http://www.gatsby.ucl.ac.uk/~gmz.

†To whom correspondence should be addressed. Department of Computer Science, Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978 ISRAEL. shamir@math.tau.ac.il URL: http://www.math.tau.ac.il/~shamir. Parts of this work were performed while this author was on sabbatical at the Department of Computer Science and Engineering, University of Washington, Seattle.

To appear in the Third International Conference on Computational Molecular Biology (RECOMB '99)

ber of required probes is independent of the genome size[6], and (3) The information theoretic content of each experiment is much higher: The overwhelming majority of unique probe hybridizations to clones in a map are negative. (RH mapping [2] is an alternative recent approach that set out to overcome the latter drawback.) This makes possible a very natural and efficient probabilistic interpretation of the hybridization data, resulting in strong robustness to experimental errors.

In this paper we suggest a practical algorithm for the physical mapping problem using oligonucleotides. In Section 2 we present a statistical model for the problem. Perhaps the single most important idea in the paper is the Bayesian analysis of the overlap length between contigs, based on the analysis of the overlap length between two clones. Both are presented in Section 3. These overlap scores enable us to make much more efficient use of the hybridization data than was possible with a binary yes/no overlap decision used before [13, 1, 10]. In Section 4 we present the actual construction algorithm. Section 5 includes the results of simulations in a variety of experimental scenarios. In realistic high-noise scenarios, the maps produced were completely correct (up to very local reordering that have little significance to the map quality) in over 96% of the cases. Results on real DNA sequences are given in Section 6. Here the original algorithm fails as its underlying statistical assumptions do not hold. However, we provide additional procedures that restore the original quality of results almost completely. To the best of our knowledge, this study gives the first quantitative performance analysis in noisy experimental scenarios. We were therefore unable to include direct comparison with other papers. We close with a discussion of the consequences of our study and of potential extensions.

Software related to this research is available through the authors. More information can be found in a web site at http://www.math.tau.ac.il/~gmz/physical_mapping.

2 The statistical model

We first describe the statistical model that underlies our algorithm and our simulations. In the basic simulations we make the following assumptions: (1) Clones are uniformly and independently distributed along the target genome, (2) Clones are of equal length, (3) Probe occurrences along the genome are modeled by a Poisson process with the same rate for all probes. The rate λ is related to the probes' length $plen$ by the formula $\lambda = 1/4^{plen}$. Probe occurrences on clones are then determined by the clone position. These assumptions originate in [10], where they are also discussed in depth. The same assumptions were later adopted by other researchers [1, 12]. Assumptions (1) and (2) were also used in [7] in a general analysis of mapping scenarios, and were shown to produce clone distributions that mimic quite accurately real genomic clones.

Noise is introduced into the model via false positive and false negative errors. False positives are modeled by an independent Poisson process with the same rate on each clone. Each probe occurrence results in a hybridization with probability $1-\alpha$ independently of other occurrences of the same probe. The parameter α is a measure of false negatives errors.

Assumption (1) and the noise modeling are in good agreement with real experiments. Assumption (2) is a reasonable approximation for cosmid clones, which are the most likely clones to be used in physical mapping with oligonucleotides. There are nevertheless other interesting scenarios where larger clone length variability occurs. In Section 5 we examine the effect of variable clone lengths. Assumption (3), which we call *the same-rate assumption*, is the most problematic. In Section 6 we focus on mapping simulated data based on real genomic sequences of several organisms. We show that while the Poisson approximation appears to be valid, the same-rate assumption does not hold for randomly chosen probes. We present a solution to this problem based on judicious pre-selection of probes. With this improvement, our mapping algorithm obtains very good results on real sequence data.

Like any probabilistic model, ours is only a model, and as such does not fully reflect reality. In spite of the above caveats these assumptions have the advantage of simplicity allowing both analytical and simulation aided investigation. The insights gained from such an investigation shed invaluable light on the more complicated experimental conditions of a real lab.

3 The Bayesian overlap score

In this section we give the outline and key ideas of the overlap scores. A full description is given in Appendix A.

Consider a pair of clones with an overlap section of length l_γ. With a small l_γ it is quite likely for a probe to hybridize with one clone, but not with the other. With a larger l_γ the probe is more likely to hybridize with both clones, or with neither. This observation can be made precise, giving for every possible value of l_γ its likelihood given the data. Combined with *a priori* overlap probabilities, Bayes Theorem makes it possible to calculate the probability that two clones overlap by a certain length given their fingerprinting data.

Because of efficiency considerations, we make the simplifying assumption that l_γ can only take few quantized values, in steps of $1/d$ of a clone's length. With the aid of simple tables the complexity of calculating the overlap probabilities vector for a pair of clones is $O(n+d)$, where n is the number of probes. Typically, $d \ll n$ ($d = 10$ was used in our simulations), and so the total complexity is linear in the number of probes.

We finally note that in the calculation of the overlap length likelihood we assume that only a yes/no hybridization signal is available to the construction algorithm. Finer hybridization strength distinctions can be used to provide improved overlap probabilities estimates.

Contigs are ordered, contiguous clone sets. Extending the exact overlap probability calculation to contigs of more than a few clones is impractical due to the exponential growth in complexity as a function of the number of clones. We seek instead to use the information from clone pairs to find the relative placement between two contigs. This is done by taking all clone pairs from the two contigs (one clone from each contig) as independent sources of information on the contigs' relative placement. This approach runs the risk of counting the same evidence multiple times resulting in over sharp estimates. This problem is not severe, since the algorithm depends on a correct identification of the maximum likelihood overlap, rather than on the precise shape of the distribution of likely overlap lengths. In practice, the approximation was found to produce excellent results.

4 The construction algorithm

The following is the basic algorithm. Several changes described in Appendix B were found to improve performance.

1. Initialize a contig set, so that for each clone there is a corresponding contig consisting of that clone.
2. Calculate for all the initial contig pairs and for the two possible relative orientations their relative placement probabilities vector. The results are stored in a table.
3. Find for all contig pairs and for their two possible relative orientations their best relative placement, and its probability.
4. While more than one contig remains:
   (a) Find two contigs which have relative orientation and placement that attain the highest probability.
   (b) Merge the two contigs.
   (c) Change the table calculated in step 2 to reflect the last merger. Suppose contig b was merged into contig a, then the table entries for all contig pairs (a, x) need to be changed. The required change is a simple combination of the previous entries for (a, x) and for (b, x).
   (d) Find for contig pairs affected by the merger their new best relative placement.

Assume that there are n probes and m clones of length d, and the genome has length Ld. The bottleneck steps in the computation are steps 2 that takes $O(m^2(n + Ld))$ time and Steps 4(c) and 4(d) that take $O(mLd)$ time. Step 4 is repeated $m - 1$ times, giving a total time complexity of $O(m^2(m + Ld + n))$.

When construction is completed only one contig remains. In reality, however, there are often gaps (oceans) in the sections of the genome covered by the clones, and these subcontigs should not be linked together. Moreover, errors in the construction process can also result in incorrectly linked subcontigs. It is therefore highly desirable to be able in retrospect to pinpoint suspected points of incorrect merger. The overlap score between pairs of clones on opposite sides of a suspected point is highly indicative of its reliability. By making this observation precise (Appendix B.3) we were able to define a concept of *weak points*. When the number of errors is fairly small, a good choice of errors should make it possible to mark all errors and oceans as weak points with only a small number of false positive detections.

5 Simulation results

This section describes the simulations we have performed with the algorithm when clone positions, fingerprint data and errors were generated according to the statistical model described in Section 2.

Simulation setup: m clones of length l_c were positioned randomly with a uniform distribution along a genome of length L. The Poisson rate for probe occurrences was calculated from the probe length *plen* using the formula $\lambda = 1/4^{plen}$. Probe occurrences determined the probe-clone occurrence vector $\vec{A_i}$ for each clone C_i ($A_{i,j} = k$ iff probe j occurs k times on C_i). The probe hybridization probability was defined by $p(hyb_{i,j}) = p(\textit{false positive}) + (1 - p(\textit{false positive}))(1 - \alpha^{A_{i,j}})$, giving the expression $p(hyb_{i,j}) = 1 - e^{-\beta l_c}\alpha^{A_{i,j}}$. A binary hybridization pattern $\vec{B_i}$ was then chosen randomly according to the hybridization probabilities $p(hyb_{i,j})$.

The simulation program receives the following as instance generation parameters: L, l_c, n, α, *plen*, *f.p.prob* and the clone coverage. The number of clones is $m = \textit{coverage} \cdot L/l_c$. The false positives probability parameter *f.p.prob* is the probability of a false positive in a clone, and is related to β by the formula $\textit{f.p.prob} = 1 - e^{-\beta l_c}$.

The following combination of parameters was used as a *base scenario*: clone length $l_c = 40960$ bp (base pairs), genome length $L = 25$ clone lengths (approximately 1M bp), coverage of 10, false negative probability $\alpha = 0.2$, false positive probability $\textit{f.p.prob} = 0.05$, probe length $\textit{plen} = 8$, number of probes $n = 500$. These parameters correspond to an experiment with cosmid clones, many probes and a realistic high noise. Other scenarios were constructed by changing one of the parameters from its value in the base scenario.

Map accuracy measures: We developed several measures for quantifying the accuracy of maps produced by the algorithm. We report here on one measure that we have found most informative. It detects significant errors between the constructed and the correct map and ignores small differences that can be corrected by small shifts or local reordering. Sub-contigs of the constructed map between significant errors are essentially correct, but the relative order and orientation of these sub-contigs are incorrectly determined. We call such errors *big errors*. For example, the map in Figure 1(b) contains one big error, viewed as a big jump in the clone order as a function of the constructed order. A description of how big errors are calculated is included in Appendix C. When no big errors are found the map is essentially correct. A variable *any big* is defined to be 1 or 0 depending on whether there are any big errors. Its average is used to estimate the probability of having at least one big error.

Test examples: Refer to Figure 1 for a graphical illustration of the algorithm's results. In typical runs in the standard scenario there are only insignificant ordering errors between nearby clones. Increasing the genome length does not have a significant effect on performance. The last two subfigures illustrate what happens under very adverse conditions. The weak points detected by the algorithm (Section B.3) enable the experimenter to pinpoint possible errors. This information can be used for a judicious choice of additional hybridization experiments, minimizing cost and human effort.

Performance in the base scenario: 1000 separate experiments were performed in the base scenario in order to obtain accurate expectation and standard deviation values. In 0.961 ± 0.006 of the cases the algorithm made no big errors. Recall that the absence of big errors is a guarantee that the map is essentially correct, as every two clones that are consecutive in reality are placed by the algorithm within a clone length from each other.

The effect of separate parameters: The effect of separate parameters was tested by altering one parameter, while keeping the other parameters at their value in the base scenario. 1000 separate experiments were performed for each value, and the probability of having big errors was estimated by the average of *any big*.

The tested parameters were: $5 \leq L \leq 50$ (in clone lengths) in steps of $\Delta L = 5$, $7 \leq plen \leq 10$ in steps of $\Delta plen = 1$, $50 \leq n \leq 1000$ in steps of $\Delta n = 50$ up to $n = 200$ and $\Delta n = 100$ between 200 and 1000, $0 \leq \alpha \leq 0.8$ in steps of $\Delta\alpha = 0.1$, $0 \leq f.p.prob \leq 0.5$ in steps of $\Delta f.p.prob = 0.05$.

The results in Figure 2 demonstrate robustness to experimental noise. Even with $\alpha = 0.6$ the algorithm makes no big errors in the majority of the runs. The algorithm also copes well with relatively low coverage, but performance deteriorates rapidly below coverage 7. The algorithm is most sensitive to the frequency of probe occurrences, as it is mirrored in the probes length. Sensitivity to the number of probes becomes pronounced with less than 200 probes. The results for large map sizes indicate that the algorithm is still robust in a genome of about 50 clone lengths (2M bp). We expect acceptable performance even with larger map sizes. In the discussion section we comment on the implication of these results on the design of hybridization experiments.

We also tested the influence of variable clone lengths. Clones length were chosen from a uniform distribution, and the construction program estimated the lengths by the maximum likelihood length as a function of the total number of hybridizations with the clone. The results in Figure 2 show robustness to over 35% variability. Better performance can be expected with a normal distribution of clone lengths, which is probably more realistic.

Physical resources A PC with a 100Mhz Pentium Microprocessor and 32MB of memory (256K cache) was used. The implementation was written using GNU C++, and run under the LINUX Unix type operating system. An average run of the construction algorithm (excluding instance generation) took 10 seconds in the base scenario and required 3MB of memory (memory requirements vary somewhat between runs, as the clones are divided into groups differently).

6 Results on real DNA sequences

Our next step was to test the algorithm in simulations involving real DNA sequences. These sequences were taken from a variety of representative organisms: the nematode *C. Elegans*[1], bacterium (*E. Coli*)[2], yeast[3], and human[4]. The lengths of the sequences ranged from about 1.7MB to over 4MB. Non overlapping sections of length 1MB from each genome were used for the tests (1MB and 730MB for Homo Sapiens). An additional random DNA sequence was used for comparison.

Using randomly picked 8mers, our algorithm fared disastrously on real DNA sequences. The reason is that in real sequences the same-rate assumption made in our model completely collapses: the number of occurrences of quite a few probes is about average, but many probes occur very rarely, while others occur extremely frequently. And so, only some "good" probes fit out statistical model.

The obvious solution of fitting a separate Poisson model for each probe will not do, because most probes occur very infrequently, and will therefore provide very little informa-

[1] The C. Elegans data is from a Sanger Centre sequencing project. The file we have used is ftp://ftp.sanger.ac.uk/pub/databases/C.elegans_sequences/FINISHED_SEQUENCES/large_C._elegans_contig.

[2] The E. Coli DNA was taken from the E. Coli Genome Project, University of Wisconsin Madison. The file we have used is ftp://ftp.genetics.wisc.edu/pub/sequence/ecolim52.fas.gz.

[3] The yeast DNA we have used is chromosomes iv,vii, and xv taken from the Munich Information Centre for Protein Sequences (MIPS) accessed through the web site at http://www.mips.biochem.mpg.de/mips/programs/get_fragment.html.

[4] The human DNA was taken from two separate sources. Chromosome 7 data was taken from the University of Washington Genome Center (UWGC) at http://chimera.biotech.washington.edu/UWGC. The sequence itself, of length 1050106 bp is available through a link in the above web site into GenBank. Chromosome 17 data was taken from the Whitehead Institute/MIT Genome Sequencing Project at http://www-seq.wi.mit.edu/sequencing/public_release/human17.html. We have used the following files: L106,L107,L108,L117,L118,L127. The chromosome 7 data was used as the main sequence file. The chromosome 17 DNA contigs were concatenated into one sequence and used as an additional DNA file.

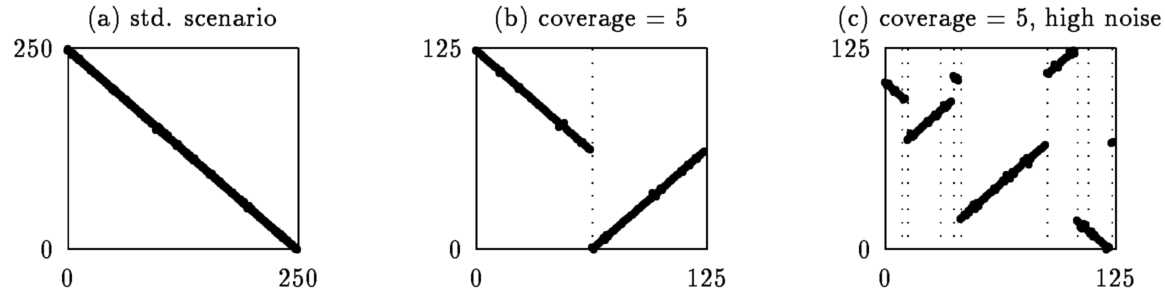

Figure 1: True clone order (y axis) vs. constructed order (x axis) in three scenarios. When applicable, weak points detected by the algorithm (cf. Appendix B) are shown as vertical dotted lines. The results are taken from the following scenarios: (a) base scenario, (b) a simulation with very low coverage (5), (c) a simulation with very low coverage (5) and very high noise ($\alpha = 0.3$ and $f.p.prob = 0.25$). Note that all big errors were detected as weak points, though some weak points incorrectly predicted additional big errors.

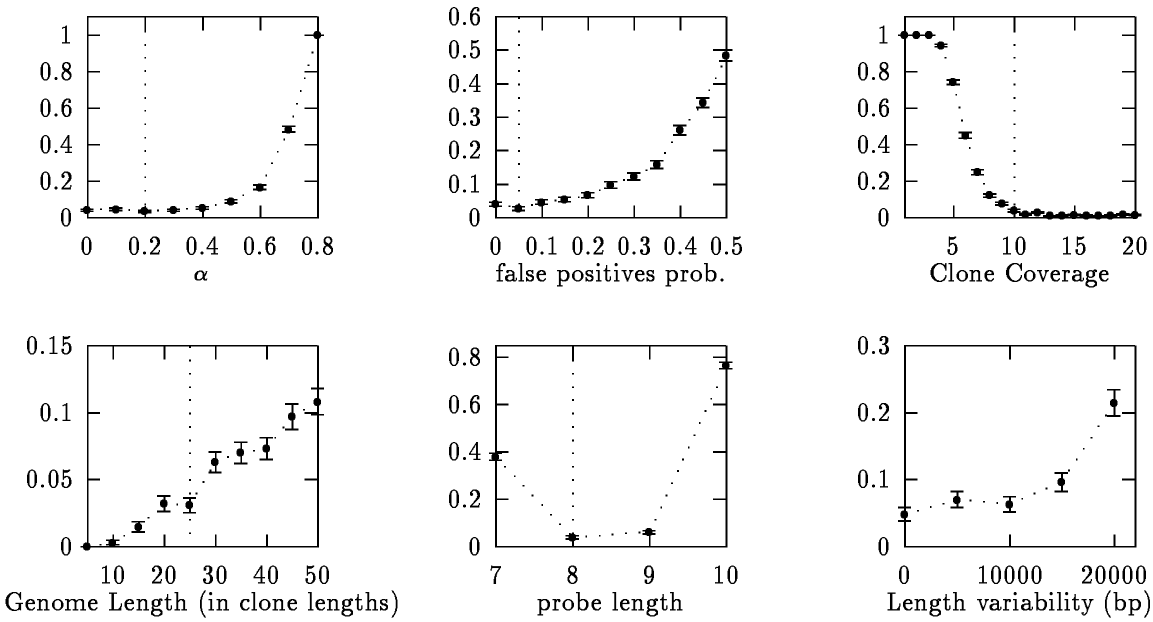

Figure 2: The influence of the various simulation parameters on the probability of having big errors. The vertical dotted line indicates the value of the parameter in the base scenario. The effect of a decrease in the number of probes is very similar to that of an increase in the experimental noise. This is because noise decreases the informational content of each probe, an effect that can be countered by an increase in the number of probes. It is also notable that the probe size has a very significant effect, resulting from its direct influence on the frequency of probe occurrences, and therefore on the informational content of the experiment. The results on variable clone length are for a uniform distribution about the average clone size. Good performance is achieved with a variability of up to about 15000 bp, or 37.5%.

tion. Because probe hybridizations are effort and cost intensive it is impractical to make very many experiments and then choose a small subset of "good" probes. However, if probes can be chosen before the actual experiment based on prior knowledge of the organism's sequence statistics (e.g. from other sequenced parts of its genome), better results can be achieved. We call this process *probe pre-selection.* The results show that the distribution of the number of probe occurrences for pre-selected probes is quite narrow around the average, with only few probes well outside this narrow region.

While pre-selection helps a great deal, some chosen probes still behave very differently from the statistical model. To overcome this problem we implemented a procedure that we call *probe post-screening* after the hybridizations are performed: only those probes which fit well the same-rate model (as assessed by the number of hybridizations) are subsequently used by the mapping algorithm. As hybridization data is obtained after probe pre-selection, there is only a small loss of usable hybridization information.

Table 1 shows the mapping results with probe pre-selection and with and without post-screening. The results clearly demonstrate the efficacy of using probe post-screening. In fact, the algorithm works better with screened probes even when the probes were Poisson distributed in the first place.

organism	test type	% maps with errors
Random	no post-screening	2.8 ± 0.7
Random	PS - 500 probes before	2.2 ± 0.7
Random	PS - 500 probes after	0.0 ± 0.0
Bacterium	no post-screening	60.6 ± 2.2
Bacterium	PS - 500 probes before	20.6 ± 1.8
Bacterium	PS - 500 probes after	10.8 ± 1.4
Yeast	no post-screening	20.4 ± 1.8
Yeast	PS - 500 probes before	4.0 ± 0.9
Yeast	PS - 500 probes after	1.2 ± 0.5
C. Elegans	no post-screening	34.6 ± 2.1
C. Elegans	PS - 500 probes before	7.4 ± 1.2
C. Elegans	PS - 500 probes after	0.0 ± 0.0
Human	no post-screening	88.2 ± 1.4
Human	PS - 500 probes before	12.6 ± 1.5
Human	PS - 500 probes after	0.0 ± 0.0

Table 1: Results of the mapping algorithm with probe preselection on real sequence data. The three lines for each organism correspond to: (1) No post-screening, (2) Post-screening on the 500 original probes (leaving about 300 screened probes), and (3) 500 post-screened probes (requiring more to begin with). The screened probes were chosen out of a pre-selected sample of probes occurring within $\pm 10\%$ of the mean on a different genome section of the same organism. The post-screened probes are estimated to occur within 10% of the mean frequency on the target genome section too. Averages and standard deviations are based on 1000 simulations in each scenario. PS: post-screening.

7 Concluding remarks

The basic simulations demonstrate robust performance in scenarios containing a large genome and highly noisy hybridizations. The algorithm is also robust to considerably more clone length variability than is the case with cosmid clones. With clones of widely variable lengths some length screening is likely to contribute significantly to performance. The basic simulations also show the importance of using many probes. The manufacturing of many probes involves high initial costs, but the same probes can later be used for the entire genome. The per-experiment costs are then much smaller than an experiment using unique probes.

A key issue emerging from the simulations is probe selection. The basic simulations demonstrate how crucial it is to choose probes with approximately 50% hybridizations. The simulations on real DNA show that random probes are a very poor choice with real DNA, but also show that by using pre-selection based on known sequence statistics specific to the organism, the algorithm could then apply post-screening and still have a sufficient number of probes to work well.

Short probes hybridization techniques are challenged by the difficulty to predict the strength of hybridization (and hence of the signal) between an oligo and a particular sequence: Factors such as GC content, mismatches and local fold structure influence the actual signal. A very important observation is that accurate prediction is not really necessary for the success of the algorithm. All that is required is that the hybridization strength between any probe and any point (a sequence segment) in the target will be *consistent* in all the clones that contain that point. The actual level of match between the sequence of the oligo and the sequence at that point in the target is immaterial. Any relation whatsoever can exist between the two sequences, provided that consistency is maintained. For example, a probe and a target sometimes hybridize well even when only a partial complementary match exists between them, but this is not a problem for the algorithm. This type of consistent deviation from a simple hybridization model has to be clearly differentiated from uncorrelated deviations of individual clone and probe pairs. The latter deviations are the false positive and the false negative errors.

The statistical inference approach to analyzing oligonucleotide hybridization data is not limited to physical mapping. In *cDNA clustering*, clone pairs can either fully overlap (one is contained in the other), or else come from different genes and have zero overlap. A likelihood ratio of these two possibilities can be used as input to the clustering algorithm. Initial experiments that we have done [9] show that it may give substantially better results than the currently used Euclidean norm or Hamming distance.

Another particularly promising field for applications is pre-selection of clones for shotgun sequencing. The problem is to select as small a subset as possible of clones covering the target genome, sequence only those first and thereby avoid re-sequencing the same target parts unnecessarily and save time and cost in sequencing [14]. Relatively inexpensive oligonucleotides fingerprinting is performed, and the resulting hybridization data can be used to produce a physical map, from which a near optimal clone subset can be chosen. In this problem the clone lengths are typically of the order of 1200bp, and so 8-mers are suboptimal as probes, as their occurrence probability is low. Using shorter probes raises a serious problem of hybridization detection fidelity. However, it has been suggested (Uwe Radelof - personal communication) that oligo pooling can be used, with a pool of several 8-mers serving as one probe, thus making it possible to achieve around 50% hybridization.

A final possible application is to restriction digest mapping. In this problem each clone is completely digested by a restriction enzyme and the input data for mapping is a list of fragment lengths for each clone. Our algorithm can be applied also to this seemingly very different type of data: On an abstract level, all the algorithm needs in order to work is features or landmarks along the DNA, where the appearance or absence of each feature is consistent in all the clones containing the corresponding DNA segment. If restriction fragment lengths serve as the features, the algorithm can be applied to restriction mapping too.

Acknowledgment

This research was supported in part by grants from the G.I.F., the German-Israeli Foundation for Scientific Research and Development, and from the Ministry of Science and Technology, Israel.

We thank an anonymous reviewer of a previous version of this paper for suggesting the test of the algorithm on real sequences, which led to a substantial improvement of the results.

References

[1] F. Alizadeh, R. M. Karp, L. A. Newberg, and D. K. Weisser. Physical mapping of chromosomes: A combinatorical problem in molecular biology. *Algorithmica*, 13:52–76, 1995.

[2] D.R. Cox, M. Burmeister, R. Price, S. Kim, and R.M. Myers. Radiation hybrid mapping: a somatic cell genetic method for reconstructing high-resolution maps of mammalian chromosomes. *Science*, 250:245–250, 1990.

[3] A. G. Craig, D. Nizetic, D. Hoheisel, G. Zehetner, and H. Lehrach. Ordering of cosmid clones covering the herpes simplex virus type I (HSV-I) genome: A test case for fingerprinting by hybridization. *Nucleic Acids Research*, 18:2653–2660, 1990.

[4] A. Cutticchia, J. Arnold, and W. Timberlake. ODS: ordering DNA sequences - a physical mapping algorithm based on simulated annealing. *CABIOS*, 9:215–219, 1993.

[5] Y.-X. Fu, W. E. Timberlake, and J. Arnold. On the design of genome mapping experiments using short synthetic oligonucleotides. *Biometrics*, 48:337–359, 1992.

[6] J.D. Hoheisel, E. Maier, R. Mott, L. McCarthy, A.V. Grigoriev, L.C. Schalkwyk, D. Nizetic, F. Francis, and H. Lehrach. High resolution cosmid and P1 maps spanning the 14 Mb genome of the fission yeast S. pombe. *Cell*, 73:109–120, 1993.

[7] E. S. Lander and M. S. Waterman. Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics*, 2:231–239, 1988.

[8] H. Lehrach, R. Drmanac, J. Hoheisel, Z. Larin, G. Lennon, A. P. Monaco, D. Nizetic, G. Zehetner, and A. Poustka. Hybridization fingerprinting in genome mapping and sequencing. In K. E. Davies and S. Tilghman, editors, *Genome analysis: genetic and physical mapping*, pages 39–81. Cold Spring Harbor, New York, 1990.

[9] G. Mayraz and R. Shamir. A bayesian score for cDNA clustering. Technical report, School of Mathematics, Tel-Aviv University, Tel-Aviv, Israel, 1997.

[10] F. Michiels, A. G. Craig, G. Zehetner, G. P. Smith, and H. Lehrach. Molecular approaches to genome analysis: a strategy for the construction of ordered overlapping clone libraries. *CABIOS*, 3(3):203–210, 1987.

[11] R.F. Mott, A.V. Grigoriev, E. Maier, J.D. Hoheisel, and H. Lehrach. Algorithms and software tools for ordering clone libraries: appplication to the mapping of the genome of Schizosaccharomyces pombe. *Nucleic Acids Research*, 21:1965–1974, 1993.

[12] Darren M. Platt and Trevor I. Dix. Comparison of clone-ordering algorithms used in physical mapping. *Genomics*, 40:490–492, 1997.

[13] A. Poustka, T. Pohl, D.P. Barlow, G. Zehetner, A. Craig, F. Michiels, E. Ehrich, A.M. Frischauf, and H. Lehrach. Molecular approaches to mammalian genetics. *Cold Spring Harb Symp Quant Biol*, 51:131–139, 1986.

[14] U. Radelof, S. Hennig, P. Seranski, F. Francis, A. Schmitt, J. Ramser, C. Steffens, A. Poustka, and H. Lehrach. Preselection of shotgun clones for sequencing by oligofingerprinting. In *Proceedings of the Human Genome Meeting (HGM '98), Turin, Italy, March 1998*, page 52. Nature Genetics, 1998.

Appendix

A The Mathematics of the Bayesian Score

This appendix gives the mathematical details of the Bayesian score used by the construction algorithm and the computation of the big errors used to assess the solution quality. A description of the clone pairs overlap score is followed by that of the contig pairs overlap score, which utilizes the results of the former. The discussion does not depend on the assumption of equal clone lengths. Finally the computation of big errors is presented.

A.1 Clone pairs overlap score

Consider a pair of clones C_a and C_b, viewed as intervals of lengths l_a and l_b respectively. Without loss of generality, assume $l_a \leq l_b$. Let $C'_\gamma = C_a \cap C_b$, $l_\gamma = |C'_\gamma|$. The overlap score uses the hybridization vectors $\vec{B}_a$ and $\vec{B}_b$ to produce a probabilities vector for the overlap length l_γ.

We first calculate $Prob(\vec{B}_a, \vec{B}_b | l_\gamma = t), 0 \leq t \leq l_a$. Let $C'_a = C_a \backslash C_b$, and $C'_b = C_b \backslash C_a$. Define $A_{a,j}$ as the number of occurrences of probe j in C_a, similarly defining $A_{b,j}$. Define $A'_{i,j}$ as the number of occurrences of probe j in C'_i for $i = a, b, \gamma$. Then:

$$\begin{aligned}
& Prob(B_{a,j}, B_{b,j} | l_\gamma = t) \qquad (1) \\
= & \sum_{k'_a=0}^{\infty} \sum_{k'_b=0}^{\infty} \sum_{k'_\gamma=0}^{\infty} Prob(B_{a,j} | A_{a,j} = k'_a + k'_\gamma) \\
& Prob(B_{b,j} | A_{b,j} = k'_b + k'_\gamma) Prob(A'_{a,j} = k'_a | l_\gamma = t) \\
& Prob(A'_{b,j} = k'_b | l_\gamma = t) Prob(A'_{\gamma,j} = k'_\gamma | l_\gamma = t)
\end{aligned}$$

(Here $A_{a,j} = A'_{a,j} + A'_{\gamma,j}$ and $A_{b,j} = A'_{b,j} + A'_{\gamma,j}$). The calculation of the probabilities inside the summation symbols is straightforward using the statistical model. Since hybridization is a virtual certainty if a probe occurs many times inside a clone, we can limit the summation to small values of k (say $0 \leq k \leq 5$), thereby making feasible the score's computation while introducing only a negligible error.

Considering each probe as an independent source of information, the conditional probability of the vector pair $(\vec{B}_a, \vec{B}_b)$ is

$$Prob(\vec{B}_a, \vec{B}_b | l_\gamma = t) = \prod_{j=1}^{n} Prob(B_{a,j}, B_{b,j} | l_\gamma = t) \qquad (2)$$

Assuming uniform parameters for the probes, the expression inside the product, $Prob(B_{a,j} = x, B_{b,j} = y | l_\gamma = t)$, is independent of j. Therefore, we can define $P^t_{x,y}$ for $0 \leq x, y \leq 1$ by $P^t_{x,y} = Prob(B_{a,j} = x, B_{b,j} = y | t)$. Denoting by $S_{x,y}$ the set of probes $1 \leq j \leq n$, such that $B_{a,j} = x$ and $B_{b,j} = y$, we can write:

$$Prob(\vec{B}_a, \vec{B}_b | t) = \prod_{x=0}^{1} \prod_{y=0}^{1} {P^t_{x,y}}^{|S_{x,y}|} \qquad (3)$$

Having computed $Prob(\vec{B}_a, \vec{B}_b | l_\gamma = t)$ we can use Bayes Theorem:

$$Prob(l_\gamma = t_0 | \vec{B}_a, \vec{B}_b) = \frac{Prob(\vec{B}_a, \vec{B}_b | l_\gamma = t_0) Prob(l_\gamma = t_0)}{\sum_{t=0}^{l_a} Prob(\vec{B}_a, \vec{B}_b | l_\gamma = t) Prob(l_\gamma = t)} \qquad (4)$$

The actual computation uses logarithms to avoid accuracy and underflow problems. We also use only quantized values for l_γ, so that $l_\gamma = 0, \frac{l_a}{d}, \frac{2l_a}{d}, \ldots, l_a$. Because the probability of each of the four possible hybridization combinations is either increasing with l_γ or else decreasing with l_γ, the maximal error introduced by the quantization is less than the quantization unit length d. The experiments in this paper use a quantization unit size of one tenth of a clone length.

A.1.1 Complexity

With the aid of simple tables, the calculation of the Bayesian score for a pair of clones consists of the following steps:

1. Calculate $|S_{x,y}|$ in equation 3, namely how are the probes distributed among the following possibilities: hybridization with both clones, with the first only, with the second only, or with neither. This calculation requires $O(n)$ steps.

2. Calculating $\log Prob(\vec{B}_a, \vec{B}_b | l_\gamma = t)$ using quantized values of t. The probabilities $P^t_{x,y}$ are calculated in advance and stored in a table. Such a table can be used for variable sized clones as well as for fixed size clones. Using this table the calculation takes $O(d)$ steps.

3. Normalization (Bayes Theorem), taking $O(d)$ steps.

The overall complexity is, thus, is $O(n+d)$, where n is the number of probes, and d is the number of possible quantized values for l_γ. Typically, $d \ll n$, so the total complexity is proportional to the number of probes.

A.2 Contig pairs overlap score

A *contig* is a set of clones covering a contiguous region of the genome together with their relative placements. Consider the problem of estimating the *a posteriori* probabilities for the relative placement of two contigs G_a and G_b Two subsets (B_a, B_b) of the rows of the clone hybridization matrix B correspond to the two contigs. Before applying Bayes Theorem we need to estimate $P(B_a, B_b | p_{a,b} = t)$ for a relative placement $p_{a,b}$ of G_b with respect to G_a (As the probabilities of contigs relative placements depend on their relative orientation, the overlap section length by itself does not determine the relative placement). Consider a pair of clones (C_i, C_j) from G_a and G_b respectively. The relative placement $p_{a,b}$ of the contigs G_a and G_b determines the overlap $o_{i,j}$ between the two clones. As we already know how to calculate $P(\vec{B}_i, \vec{B}_j|o_{i,j})$ we can calculate $P(\vec{B}_i, \vec{B}_j|p_{a,b})$. Going over all the possible relative placements of (G_a, G_b) and normalizing the resulting vector (Bayes Theorem) we get an estimate of $P(p_{a,b} = t|B_a, B_b)$ based on the information from the two clones C_i and C_j. Let $\vec{R}^{i,j}$ stand for this probabilities vector. Going over all the relevant clone pairs we make the following definition:

$$S_t \stackrel{\text{def}}{=} \prod_{(i,j)|C_i \in G_a, C_j \in G_b} R_t^{i,j} \tag{5}$$

According to this definition $\vec{S}$ is the vector of *a posteriori* probabilities: $S_t \approx Prob(p_{a,b} = t|B_a, B_b)$, approximated under the assumption that the information from each of the clone pairs is independent. This assumption is not true, and it tends to sharpen the distribution of the *a posteriori* probabilities in comparison with the true distribution. However, the approximation enables the construction algorithm to integrate information from many clone pairs, and arrive at a much more informed contig merger decision.

A.2.1 Complexity

The space taken by the two contigs is at most L, giving a complexity of $O(n + L)$ for calculating the contribution of each clone pair to the contig pairs score. In the worst case there are $m/2$ clones in each contig giving an overall complexity of $O((n + L)m^2)$.

B Improvements to the basic algorithm

We briefly describe below several ideas that were found to improve the performance of the basic algorithm. These ideas were incorporated into the final version of the algorithm used in the simulations.

B.1 Initial partition into groups

The clone pairs score is used to divide the initial clone set into groups of clones, such that in each group there is a clone that overlaps all the other clones in its group with a near certain probability. Construction is done at the group level before the ordered groups are combined. This process makes for a significant improvement in the average running time. More importantly, the weighted information of many clone pairs is available when the ordered groups are combined, thus helping avoid wrong mergers of distant clones with similar fingerprints.

B.2 Choosing the best overlapping relative placement

In every step of the basic algorithm the contig pair to be merged is the one with the highest *a posteriori* relative placement probability, thus minimizing the probability of making an error. In practice, however, errors are unavoidable, and some are far more costly than others. We therefore seek to avoid wrong mergers of non-overlapping contigs and other particularly large errors. One idea was to first identify the contig pairs that have the lowest probability of no overlap, and among those to pick the pair that has the highest relative overlap probability. Another idea was to apply a Gaussian filter on the probabilities vector, with the result that unimodal distributions are preferred over multi-peaked distributions, thereby decreasing the probability of large errors. Both processes decrease the probability of large errors at the cost of increasing that of smaller ones.

B.3 Weak points detection

For any clone pair (C_i, C_j) with an overlap $o_{i,j}$ and hybridization fingerprints $(\vec{B}_i, \vec{B}_j)$ define its clone pair "energy" as $E_{i,j} = -\log Pr(\vec{B}_i, \vec{B}_j|o_{i,j})$. The contig energy E is defined as $E = \sum_{i<j} E_{i,j}$. In analogy to the breaking-up of a chemical molecule, the separation of a contig into two non-overlapping parts should increase the energy substantially. However, if the two parts do not overlap in the real map, the separation energy should be quite small or even negative. We define ***weak points*** as points along the contig having separation energy below a threshold (determined experimentally). Such information can be used by the laboratory in order to pinpoint areas where additional hybridizations may be useful, or points of non-clonable DNA (oceans). In the limit of very many probes, weak points will exactly correspond to oceans. We also make use of the weak points in our algorithm to break-up a contig and to reassemble it again. In case an error was made at an early stage, this process enables the algorithm to correct its previous error with the benefit of the additional information from other clones added at a later stage. A similar process was reported in [12] to be slightly favorable, when only the clone order was sought.

B.4 Simulated annealing

Simulated annealing on the clones order was used before [4, 12] where its performance was validated on small data sets. The basic problem with this approach is that the natural neighborhood relation must allow a move and reversal of a contiguous group of clones. Thus, when there are many clones, the number of neighbors becomes prohibitively large. In addition, this method ignores the significant overlap between non consecutive clones. In order to overcome these problems we tried instead simulated annealing on the *contig* order after the intra-contig clone relative positioning is known. This method enjoys the fact that there are typically 10 clones in one contig (reducing the number of neighbors by a factor of 1000), and also, that there is usually no overlap between non consecutive contigs. We implemented this method, but have found it to perform no better than the basic algorithm of Section 4.

C Computing big errors

For simplicity the presentation below is done for fixed length clones, but the computation easily extends to variable lengths clones.

By a *location* of a clone we mean the coordinate of its left endpoint. We number of clones in non-decreasing order of the locations in the algorithmic solution. The solution is thus an ordered sequence $\mathcal{A} = A_1 \leq A_2 \leq \ldots \leq A_n$. Let $\mathcal{P}$ be the correct solution, with P_i the location of clone i, indexed using the same indices as in $\mathcal{A}$. Let $\overline{\mathcal{P}}$ be the reverse of solution $\mathcal{P}$. That is, $\overline{P_i} = S - P_i$ for some constant S.

We say that *an error with respect to* $\mathcal{P}$ occurred at position i of $\mathcal{A}$ if $|(A_{i+1} - A_i) - (P_{i+1} - P_i)| \geq L$, where L is some constant (in practice we used the average clone length). We similarly define an error with respect to $\overline{\mathcal{P}}$.

The determination of big errors is done as follows: We first check for each $1 \leq i \leq n$ whether an error occurred at position i w.r.t. $\mathcal{P}$. A similar check is done for errors w.r.t. $\overline{\mathcal{P}}$. Then, starting from position 1 we compute the position of the next error w.r.t. $\mathcal{P}$ and $\overline{\mathcal{P}}$, and pick the larger position i as the place of the next big error. The process is then repeated starting from position $i+1$. Table 2 and Figure 3 demonstrate the computation on a small example.

clone no.	1	2	3	4	5	6	7	8	9	10
location in $\mathcal{A}$	0	6	6	13	13	18	19	21	24	32
location in $\mathcal{P}$	6	13	13	18	-3	-10	-9	-9	23	30
location in $\overline{\mathcal{P}}$	24	17	17	12	33	40	39	39	7	0
errors w.r.t $\mathcal{P}$	-	-	-	+	+	-	-	+	-	
errors w.r.t $\overline{\mathcal{P}}$	+	-	+	+	-	-	-	+	+	
big errors	-	-	-	+	-	-	-	+	-	

Table 2: Example of the computation of big errors. The locations of the 10 clones in the algorithmic solution $\mathcal{A}$ is presented together with their location in the correct solution in both orientations $\mathcal{P}$ and $\overline{\mathcal{P}}$. Clone locations in $\overline{\mathcal{P}}$ were defined by $\overline{P}_i = 30 - P_i$, with 30 being an arbitrary constant. The value L is set to 10, the clones length. Errors with respect to $\mathcal{P}$ and $\overline{\mathcal{P}}$ are marked with $+$ signs. The last line in the table marks big errors, as determined from the previous two lines.

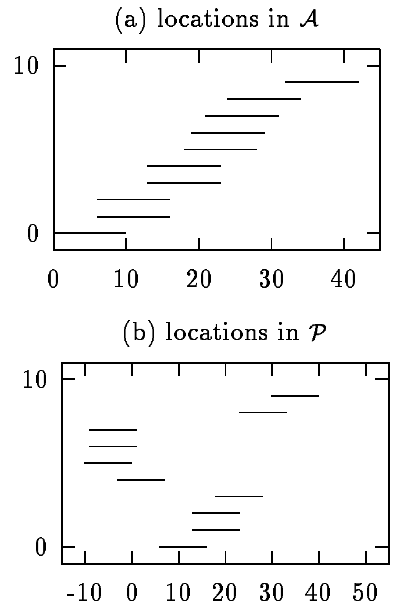

Figure 3: Physical maps for the data in Table 2. In both parts of the figure the y-axis corresponds to the clone indices in the algorithmic solution. In part (a) the x-axis denotes clone locations in algorithmic solution. In part (b) the x-axis denotes clone locations in the correct solution. Note that regardless of the correctness of the algorithmic solution, the clone locations in $\mathcal{A}$ form a non-decreasing series. It is in a graph such as (b), that the quality of the solution can be assessed. In this case, a division into three section is apparent, corresponding to the two big errors. Note also that in the first and third regions the orientation appears to be correct, whereas in the second region a reversed orientation fits $\mathcal{A}$ better.