BIOINFORMATICS DISCOVERY NOTE

Vol. 1 no. 1 2002
Pages 1–8

Microarray Synthesis through Multiple-Use PCR Primer Design

Rohan J. Fernandes[1] and Steven S. Skiena[1]

[1] Applied Algorithm Laboratory, Dept. of Computer Science, State University of New York, Stony Brook, NY, 11794-4400, USA

ABSTRACT

A substantial percentage of the expense in constructing full-genome spotted microarrays comes from the cost of synthesizing the PCR primers to amplify the desired DNA. We propose a computationally-based method to substantially reduce this cost. Historically, PCR primers are designed so that each primer occurs uniquely in the genome. This condition is unnecessarily strong for selective amplification, since only the *primer pair* associated with each amplification need be unique. We demonstrate that careful design in a genome-level amplification project permits us to save the cost of several thousand primers over conventional approaches.

Contact: skiena@cs.sunysb.edu

INTRODUCTION

Microarray technology (Maier *et al.*, 1997; Schena *et al.*, 1995) has revolutionized our understanding of gene expression. Microarrays are glass matrices on which DNA/RNA probes are affixed for the purpose of differentiation and identification of target material. Applications of microarrays include cell cycle analysis (Cho *et al.*, 1998; Spellman *et al.*, 1998), studying the response of cells to environmental stress (Gasch *et al.*, 2000), and the impact of gene knockouts in yeast (Hughes *et al.*, 2000).

The importance of this technology mandates that we study better ways to make microarrays. We are particularly interested in expanding the use of microarrays beyond the most popular model organisms to the literally thousands of species whose genomes have or will be sequenced in the near future. Indeed, the NCBI genomes web page (NCBI, 2002) now claims to contain whole genomic sequence for over 800 organisms! Most are small viruses, but almost 100 free-living species have already been sequenced. For most of these organisms, including many pathogenic bacteria and agricultural pathogens, we know relatively little of their biology except the sequence.

Fully exploiting this wealth of full-genome sequence requires developing custom microarray design/fabrication technologies which are inexpensive enough for typical individual investigators to pursue. In this paper, we develop a new microarray design approach which promises substantially reduced fabrication costs.

A substantial percentage of the expense in constructing full-genome spotted microarrays comes from the cost of synthesizing PCR primers to amplify the desired DNA. For example, in the NIH-funded Microarray equipment grant of Futcher and Leatherwood to build spotted microarrays for the yeasts S. cerevisiae and S. pombe, about $110,000 out of a total $220,000 budget was allocated to PCR primer synthesis.

In this paper, we describe an approach to minimizing the set of primers required to amplify a given set of interesting features in a genome. Historically, PCR primers are designed so that each primer occurs uniquely in the genome. This condition is unnecessarily strong for selective amplification, since only the *primer pair* associated with each amplification need be unique. Thus by careful design in a genome-level amplification project we can reuse literally thousands of primers in multiple roles, for a substantial reduction in cost.

This paper is organized as follows. First, we describe our general approach to genome-level PCR primer design, including a survey of related work. Next, we analyze the increase in total primer length necessary to overcome a given gap tolerance, and show that it is sufficiently small to support the method. We formalize two versions of the multiple-use design problem, and provide hardness results, most notably that it is hard to approximate the best primer design to within a logarithmic factor of optimality. None the less, we have developed heuristics which work reasonably well in practice. We then describe our experiences using conventional single-use primers to design full-genome arrays for two yeasts, *Saccharomyces Cerevisiae* and *Saccharomyces Pombe*. Finally, we present two heuristic for multiple-use primer design and simulation results on array designs for these yeasts. One heuristic is produces better results and a savings in number of primers required. The second heuristic is faster, in fact running in linear time, but gives smaller savings in number of primers used. However this heuristic may be more promising, particularly in designing full-genome arrays for organisms with larger number of genes. These demonstrate that our method can halve the PCR costs of microarray synthesis over conventional approaches.

GENOME-LEVEL PCR PRIMER DESIGN

The polymerase chain reaction (PCR) has revolutionized the practice of molecular biology, making it routine to create millions of copies of a single gene or any other portion of a genome. PCR requires the presence of two single-stranded DNA sequences called *primers*, which complement specific parts of either the forward or reverse strand of the double-stranded DNA and enable duplication of the region in-between.

Primer design is the problem of constructing these delimiting elements of the reproduced region. Important criteria in primer design include melting temperature, PCR product size, secondary structure, and the uniqueness of each designed primer. Traditionally, primer design programs such as Primer 3 (Rozen & Skaletsky, 1998) have focused on designing unique left and right primers for each gene in isolation.

We propose a more efficient approach for genome-level primer design. Amplifying a gene requires left and right primers which hybridize to nearby regions on the genome. Because the efficiency of PCR amplification falls off exponentially as the length of the product increases, PCR becomes ineffective for product sizes beyond 1200 bases or so. Although it is nice to have primers that hybridize to a unique region on the genome, this is not strictly necessary for successful PCR. Why? Hybridization outside the target region will not result in significant amplification unless both primers hybridize sufficiently closely to each other.

This creates the possibility of using common primers to amplify several genes, provided each *primer pair* is unique. The potential win from such a technique is enormous. Let n be the number of genes to be amplified and m be the minimum number of primers required to amplify all of them. Since m primers can result in $m(m+1)/2$ unique primer pairs, potentially $m = \sqrt{n}$ primers suffice for amplification instead of n with the conventional design. For a yeast-sized genome of 6,000 genes, this potentially reduces the number of primers needed from 12,000 to 78.

Although this lower bound is exceptionally optimistic, the potential is very compelling and worthy of further investigation. Consider cost of building a spotted microarray for a 20,000 gene organism, where 20-mer primers cost $4 each. The primers alone for such an array using conventional methods costs $160,000, far beyond the budget of a typical investigator. However, with careful multiple-use primer design we may hope to cover these genes with only 3,000 primers or so, for a quite manageable cost of only $12,000.

Previous work on genome-level PCR primer design has focused on the use of degenerate primers for multiplex PCR. There the goal is to fish out all similar or homologous proteins by picking primers in highly-conserved regions in representative family members (Rose *et al.*, 1998), or minimize the number of PCR reactions to express all of a given set of proteins which may then be separated by electrophoresis (Doi & Imai, 1999; Pearson *et al.*, 1996). Neither technology is relevant for spotted microarray design, because (1) typically we have fully sequenced the genome of an organism before seeking to build a spotted microarray for it and (2) the volume and purity of product needed for spotted microarrays requires individual amplification.

THE COST OF SPLIT ADDRESSING

We are interested in computing the probability that a given pair of strings will occur in close juxtaposition in a random string. More formally in a random string of length n, we want to know the probability that two strings of length α and β will occur in a given order and at no more than a distance of d from each other. The order is important since only the $3'$ end of a primer is chemically active. The problem of exactly computing such probabilities, is quite subtle, depending on the exact contents of the string (Guibas & Odlyzko, 1981; Wilf, 1987). For example, note that '00' occurs in only 3 three-bit binary strings while '01' occurs in 4 such strings.

To avoid such complications, we consider a simpler model where substrings of length l are assumed to occur uniformly at random in DNA with a probability of $1/4^l$. Under such a model, the probability $P(n, \alpha, \beta, d)$ that a primer pair of length α and β respectively occur by chance in a random n-base sequence, allowing a gap of length up to d is given by:

$$P(n, \alpha, \beta, d) = 1 - \left(1 - \frac{1 - \left(1 - 4^{-\beta}\right)^{d}}{4^{\alpha}}\right)^{-\alpha+n} \quad (1)$$

Figure 1 plots the total primer length l (assuming $\alpha = \beta = l/2$) required to define a region in a random sequence of length n as a function of gap length. We say a primer length is sufficient if the probability of occurrence in the random sequence is at most 0.01.

Figure 1 shows that gapped primer pairs require longer total length than an ungapped primer, but the additional length required grows very slowly with the length of the gap. The penalty for genome-scale lengths and realistic PCR gap lengths amounts to only an additional 3-4 bases of primer over ungapped matching. These results support the potential of multiple-user primers, provided we can algorithmically identify good primer sets.

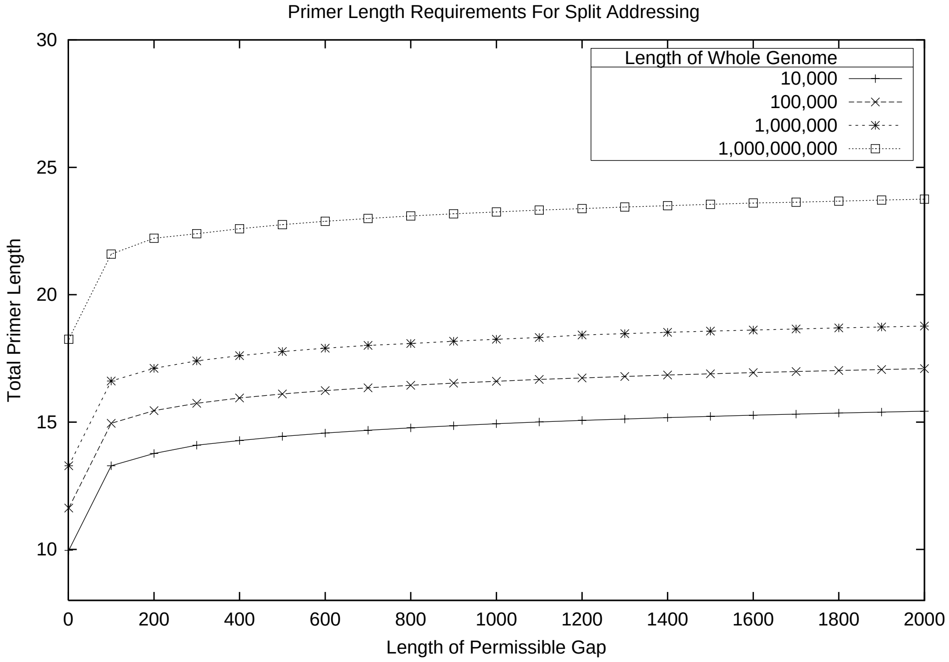

Fig. 1. Primer length vs. maximum gap length required for random binding to occur with probability $p = 0.01$. The numbers in the legend.

THE COMPLEXITY OF PRIMER DESIGN OPTIMIZATION

Multiple-use primer design requires the solution of a difficult combinatorial optimization problem. Given a set of k potentially amplifying primer pairs for n genes, $k \geq n$, find a minimal set of primers from these pairs such that we can amplify each gene using only combinations of primers from this set.

It is convenient to model this problem, which we call *minimum primer set*, as an edge-coloring problem on graphs. Represent each candidate primer sequence as a vertex of the graph. For every primer pair that uniquely amplifies a given gene, connect the two vertices with an edge and label (color) the edge with the name of the gene. Our problem is to find the smallest subset of vertices which induces a subgraph that contains edges with all possible gene colors.

An alternate formulation of the problem, called *budgeted primer set*, seeks the k primers (vertices) which cover the maximum number of different genes (colors) possible. We want to maximize our investment by selecting a set of primers that enables us to amplify as large a set of genes as possible. Thus k is a budget for how many primers we are willing to synthesize, and we seek to amplify as many genes as possible under this constraint. Formally:

Unfortunately, we have shown that both of these primer design problems are NP-complete. Indeed, we have proved that the minimum primer design problem is hard even to approximate within a logarithmic factor.

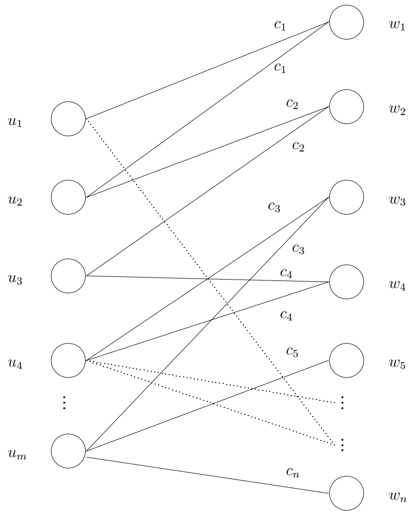

Fig. 2. The edge-colored subgraph constructed from an instance of set cover.

THEOREM 1. *The minimum primer set problem is inapproximable to less than a* $[1 - o(1)] \ln n - o(1)$ *factor.*

Proof: We demonstrate the hardness of minimum primer set by demonstrating an approximation-preserving reduction from set cover.

Consider an instance of set cover (S, X) where X is a set of n elements $\{x_i | 1 \leq i \leq n\}$ and S is a set of m subsets of X, $\{S_j | 1 \leq j \leq m\}$. We know from (Feige, 1996) that the best possible approximation factor to set cover is $(1 - o(1)) \ln n$. Approximating set cover to any better factor would have very surprising complexity-theoretic implications.

Now we construct a reduction from set cover to minimum primer set. Consider an edge-colored graph with a set of $m + n$ vertices $V = U \cup W$. Let the set of vertices $\{u_j | 1 \leq j \leq m\}$ be in one-to-one correspondence with the set of subsets S. Hence $|U| = m$. Similarly let the set of vertices $\{w_i | 1 \leq i \leq n\}$ be in one-to-one correspondence with the set of elements $\{x_i | 1 \leq i \leq n\}$. We refer to the elements of U and W as *set* and *element* vertices, respectively. Now we connect edges between u_i and w_j iff set S_i contains element x_j. We color this edge with a color c_j taken from a set C of size n and in one-to-one correspondence with the set of elements X. This completes the hardness proving gadget as shown in Figure 2.

Optimally solving minimum primer set on this graph requires selecting all the vertices in W, and selecting a set of vertices from U corresponding to a minimum set cover $T \subseteq S$. Any heuristic solution will include all vertices in W and a set of vertices from U, which will give us an approximation to the minimum set cover $T' \subseteq S$. However the best approximation ratio obtainable for set cover is $(1 - o(1)) \ln n$. Hence the lower bound on worst-case approximation ratio we can obtain for this instance of minimum primer set is $(((1 - o(1)) \ln n) \cdot OPT_{setcover} + n)/(OPT_{setcover} + n)$.

This demonstrates hardness, but does not tend to $\ln n$ in the limit. The remainder of our discussion demonstrates a tighter lower bound by using a stronger gadget to prove hardness of approximation.

Consider the instance of set cover (S, X). Let us multiply the size of this instance by a factor $p(n)$ where $p(n)$ is a polynomial in n. We do this by creating sets $\{S'_{i,j} | 1 \leq j \leq p(n)\}$ corresponding to each original set S_i. Each element $x_j \in S_i$ now maps to $p(n)$ elements $x'_{j,k}$ where $1 \leq k \leq p(n)$ and $x'_{j,k} \in S'_{i,k}$. Thus this new instance of set cover (S', X') has $m \cdot p(n)$ sets and $n \cdot p(n)$ elements. Clearly the optimal solution to this instance is simply a multiple of the solution to the original set cover problem instance (S, X) where the multiple is $p(n)$. Similarly any approximation obtained to the original set cover problem will similarly be multiplied by $p(n)$ to obtain a solution to this multiplied problem.

Now to convert this problem to an instance of minimum primer set. As before we obtain the set of vertices U' where $|U'| = m \cdot p(n)$. Each vertex $u'_{(i-1) \cdot p(n)+j}$ corresponds to a set $S'_{i,j}$. However $W' = W$ as defined earlier. The new set of vertices is $V' = U' \cup W'$. We connect two vertices $u'_{(i-1) \cdot p(n)+j}$ and w'_k by an edge iff $x_k \in S_i$. The edge will be colored (labelled) with color $c'_{(i-1) \cdot p(n)+j}$. The new set of colors is C' and of course $|C'| = n \cdot p(n)$. Now we have a new minimum primer set instance (V', E', C').

Using the same reasoning as before we conclude that any exact or approximate solution to minimum primer set must include all the vertices in set W'. Also from the approximation hardness of set cover we know that the lower bound on the worst case number of vertices we will pick from U' is $p(n)(1 - o(1)) \cdot \ln n \cdot OPT_{setcover}$ unless NP has a superpolynomial time algorithm. Thus

$$\begin{aligned} r &= \frac{p(n)[1 - o(1)] \ln n \, OPT_{setcover} + n}{p(n) \, OPT_{setcover} + n} \\ &= [1 - o(1)] \ln n + \frac{n \, (1 - [1 - o(1)] \ln n)}{p(n) \, OPT_{setcover} + n} \\ &\approx [1 - o(1)] \ln n - o(1) \end{aligned} \quad (2)$$

THEOREM 2. *The budgeted primer set problem is NP-complete.*

Proof: Clearly, the problem is in NP. To show hardness, we reduce the problem to finding the maximum clique in a graph.

A clique is a set of vertices in a graph which are connected to all other vertices in the set. An instance of the clique decision problem is given by (V, E, k) where V is the set of vertices, E is a set of edges and k is an integral parameter. We need to determine if there exists a subset $S \subseteq V$ of size k or more, such that the subgraph induced by this is a clique.

We define an instance budgeted primer set as (V, E, C, k, c) where V is a the set of vertices in a graph. E is the set of edges colored with colors from the set C, and k and c are integral parameters. We need to determine if there exists a subset of vertices $S \subseteq V$ in the graph such that $|S| = k$ and there are edges of at least c different colors induced by the set S.

Now to solve clique we just color the graph with a set of colors C such that $|C| = |E|$ and each edge has a different color. Now with this graph as input we can use $c = \binom{k}{2}$ and k the same value as for clique in the budgeted primer set decision problem to obtain a solution to clique. The budgeted primer set instance is solvable if and only if there is a clique in the graph of size k. Hence budgeted primer set is NP-complete. □

	Totals		By Category					
yeast	success	failure	I		II		III	
S. cerevisiae	5827	21	5572	18	240	2	15	1
S. pombe	5012	30	2783	22	1614	0	615	8

Table 1. Summary Statistics on Primer Designs for Cerevisiae and Pombe Arrays

YEAST MICROARRAY DESIGN

We initially designed genome-level PCR primers using the conventional approach for two yeasts, S. cerevisiae and S. pombe. After eliminating single copies of homologous genes and adding other sites of interest, we were left with 5848 genes in S. cerevisiae and 5042 in S. pombe.

Our program translated genome files in the EMBL database format into appropriate input for the primer generation program Primer 3 (Rozen & Skaletsky, 1998). We enforced strict conditions to identify the best region to amplify. Primers were partitioned into three classes, of increasing complexity and decreasing desirability. Category I primers amplify genes that have only one exon. Category II primers replicate a region of a gene lying on a single exon. Not all genes have category II primers, often because of insufficiently large exons. Category III primers are designed so the left and right primers complement different exons of the gene. However, we restrict search to bound the length of the enclosed introns. Table 1 shows our results. The number of failures and successes are shown for each category of primer, along with the totals of successes and failures across categories. By success we mean that a primer pair, satisfying all criteria was obtained for a given gene. Conversely by a failure we mean that a suitable primer pair was not found to exist for a given gene. Our results demonstrate that we were highly successful finding primers in all three cases.

Our careful primer design methodology appears to be a significant improvement over the primer design for most previous microarray projects. Indeed, we believe that poor primer design is the primary reason why some have experienced trouble using PCR to generate microarray DNA. With good primer design, PCR remains the best method for synthesizing spotted microarrays.

The primers we designed have performed very well in the lab. Our initial test of 4 arbitrarily selected 96-well plates (representing the primer pairs for 96 S. pombe and 96 S. cerevisiae genes) was completely successful, as *all* test pairs expressed gene-product of the appropriate length. Based on this success, we have now ordered the primers for the full complement of S. pombe and S. cerevisiae genes.

Certain simple post-processing steps in our software make it easier to evaluate the final design. A consistent labelling scheme in assigning left and right primers to 96-well plates minimizes the chance of human error during PCR. By sorting the genes on each plate by product length, visual inspection of the results after gel electrophoresis provides reassurance that the reactions performed successfully. Algorithmically assigning genes to plates in the optimal way maximizes visual discrimination of the electrophoresis product.

MULTIPLE-USE PRIMER DESIGN

Densest Subgraph Heuristic

Our initial heuristic for minimum primer set was inspired by Charikar's (Charikar, 2000) heuristic to approximate the densest subgraph problem (Gallo *et al.*, 1989), where the density of a subgraph is the ratio of number of edges induced by vertices in the subgraph to the number of vertices in the subgraph. The greedy algorithm successively strips the vertex with the minimum degree in the subgraph from it. In the end we select the intermediate subgraph with the maximum average degree, which can be shown to be at least half as dense as the densest possible subgraph. Our graph is edge-colored, however, and we seek the color-densest graph, which complicates the problem considerably.

We note that the budgeted version of the problem is related in some sense to the densest k-subgraph problem, which has been well studied (Asahiro *et al.*, 2000; Feige & Seltser, 1997), with approximation bounds on greedy approaches that are not very encouraging.

In our heuristic for minimum primer set, all the primers/vertices are initially weighted according to the number of genes/edge-colors they are incident to. At each iteration we discard the lowest weight vertex which does not result in any color/gene being lost in the subgraph. Specifically:

1. For each color-c in the graph compute the number of edges with this color, n_c.
2. Set the weight of each color c edge to $1/n_c$.
3. Set the weight of each vertex v to be the sum of weights of all edges incident on v.
4. Remove the minimum-weight non-critical vertex from the graph. If removing this vertex removes the last edges of a given color/gene, mark this vertex as critical.
5. Decrement n_c for each color-c edge removed from the graph by step 4.
6. Repeat steps 2-5 until no additional vertices can be removed from the graph.

To evaluate the performance of our proposed multiple-use primer heuristic, we prepared candidate primer sets for S. cerevisiae and S. pombe using Primer 3. The input criteria were as follows:

- The length of the primers ranged between 8 and 12 bases.
- Various ranges of melting temperatures were selected and evaluated, summarized in Table 2.
- PCR product size was allowed to range from 300 and 1200 bases, except for small genes where the product size was selected to be (7/10) of the length of the gene or greater. Further, all products were restricted to lie near the 3' end of the gene.
- For each gene, a maximum of 10,000 primer pairs were considered for further optimization.

These data sets were further preprocessed to eliminate duplicate pairs of primers that amplified different genes. This was necessary to ensure that only one gene is amplified by a pair of primers, and explains why different numbers of genes are amplified in each data set in Table 2. The appropriate edge-colored graph was then created from this data set for optimization.

From these graphs, we also derived edge-colored graphs of *degenerate* primers. Degenerate primers are a collection of similar sequences which can be simultaneously manufactured by substituting multiple bases at given positions. Significant amplification can result even with degeneracies of literally hundreds of thousands of sequences (Shamir & Linhart, 2001). However, for our initial investigations we limited primers to be degenerate at only at one sequence position, for a maximum degeneracy of 4.

As expected, the resultant degenerate primer graph is significantly larger and denser than the initial graph, which leads to significantly smaller primer sets.

Our simulation results on primer pair design appear in Table 2. The column T_m shows the melting temperature range of the primers designed. The lower bound is a number of primers provably no smaller than the minimum required for the given instance. We computed the lower bound as follows. First we calculated the colored degree of each vertex in the graph i.e. the number of different colored edges incident on a vertex. Then we sorted the vertices in the order of decreasing colored degree. The degrees of vertices were then added in order until the sum equaled or exceeded twice the number of colors in the graph. The number of vertices thus required gives the lower bound on number of vertices in a solution to minimum primer set.

The difference between lower bound and heuristic cost fields delimits the potential gains possible by using smarter heuristics.

For the three non-degenerate *S. Cerevisiae* data sets we obtain 27%, 42% and 55% improvements over the naive number of primers required to amplify all the genes having primers with the given melting temperatures. For the two degenerate *S. Cerevisiae* data sets we obtain more dramatic improvements of 51% and 61% respectively. For the three non-degenerate *S. Pombe* data sets we obtain improvements of 30%, 43% and 46% percent reductions in the primer set size required to obtain amplifications of all genes. For the only degenerate *S. Pombe* data set we could run our heuristic on we obtained a 54% per cent reduction in the number of primers required.

Two trends are clearly visible from the data. The first is that the solution size increases with larger size and density of the graph. This is a function of melting temperature, since more appropriate primers can be found at a lower melting temperature and hence redundancy can be better exploited. Second, introducing degeneracies also results in a reduction in solution size. This is again due to the increase in the density of the graph and correspondingly greater redundancy.

Even with our initial heuristic and low degrees of degeneracy, our designs save the cost of thousands of primers on every data set. Further, the substantial gaps between our lower bounds and heuristic costs suggest that there is substantial room for further optimization by better heuristics. The increased savings from even such small amounts of degeneracy bodes well for substantial future improvements.

Yeast-temperature pairs missing degenerate data entries from Table 2 indicates that the particular data sets were so large that solving them using our heuristic was impossible due to computational restrictions. The algorithm given above runs in time $O(|V| \cdot (|V| + |E|))$ and requires space $O(|V| + |E| + |C|)$. The next subsection deals with a faster heuristic to address this problem.

Linear-time Greedy Heuristic

The heuristic described in the previous subsection works well in practice. However it is computationally expensive. Because its running time is quadratic in $|V|$, it performs especially slowly on the larger data sets. Hence we explored the possibility of using simpler, linear-time heuristic. The heuristic described in this section satisfies both these requirements.

The new heuristic is much faster than the heuristic of the previous subsection, especially on the larger data sets. For the Cerevisiae, degenerate data set with melting temperature in the range 42–52 (an input graph with over 35 million edges) this heuristic produced a solution within 25 minutes on a Sun Ultra Sparc server with 3 GB of RAM running on SunOS 5.6. On the other hand, our implementation of the Densest Subgraph heuristic produced a solution only after two full days of computation.

The new heuristic uses the same candidate primer input graphs as the previous heuristic.

Input Graph Properties					Densest Subgraph Heuristic		Linear Time Heuristic	
Yeast	Graph Type	T_m	Amplified Genes	Lower Bound	Heuristic Cost	Primer Reduction	Heuristic Cost	Primer Reduction
cerevisiae	non-degenerate	47–57	3775	3065	5483	2067	5511	2039
cerevisiae	non-degenerate	42–52	2700	1344	3130	2270	3232	2168
cerevisiae	non-degenerate	40–50	5313	1241	4753	5863	5157	5469
pombe	non-degenerate	45–55	3583	2622	4987	2179	5058	2108
pombe	non-degenerate	43–53	4232	1988	4799	3665	4951	3513
pombe	non-degenerate	40–50	3400	1380	3651	3149	3852	2948
cerevisiae	degenerate	47–57	3775	1221	3638	3912	3940	3610
cerevisiae	degenerate	42–52	2700	475	2105	3295	2481	2919
pombe	degenerate	45–55	3583	1050	3283	3883	3598	3568

Table 2. Primer-Pair Optimized Designs for Cerevisiae and Pombe with Linear-time Greedy Heuristic

1. For each vertex in the graph, compute its colored degree, i.e. the number of different colored edges the vertex is adjacent to.
2. For each color, select an edge with that color as seed edge. Select the edge with the maximum sum of colored degrees of its inducing vertices. This can be performed for all vertices during one traversal of the graph. The set of seed edges make up the seed subgraph.
3. For each e edge in the graph, check whether replacing the current seed edge of e's color with e reduces the number of vertices in the subgraph. If so, replace the edge with e.
4. If e leave the vertex count unchanged, replace the current seed edge with e with probability $\frac{1}{2}$.
5. Repeat steps 3-4 until no more improvement can be obtained in the number of vertices in the seed subgraph.
6. Select all seed edges in the seed subgraph which do not have neighboring seed edges. This means that the vertices inducing the seed edge are not part of any other seed edge. Remove these seed edges from the seed subgraph.
7. Eliminate all colors of seed edges obtained in step 6 from the graph and go back to step 1 and repeat until there is no more improvement in the number of vertices in the seed subgraph.
8. The final seed subgraph is the color-dense subgraph returned as the result of our heuristic.

A few notes are in order regarding this heuristic. It above runs in time $O(|V| + |E| + |C|)$ if properly implemented. Step 2 enables us to make an intelligent guess about which regions of the graph are dense in colors and picking initial edges out of them promises a better solution.

Steps 3 and 4 are the main optimizing steps of the heuristic. With the right data structures step 3 takes $O(1)$ time for each edge of the graph. The introduction of randomization in step 4 contributed some improvement to the final solution.

Finally in step 7 we eliminate all colors costing us two vertices in our subgraph (since they are no better than any other primer pair for the gene) and look for a better solution outside of it.

Our primer-pair results for the linear-time heuristic are summarized in table 2. For the three non-degenerate *S. Cerevisiae* data sets we obtain 27%, 40% and 51% improvements over the naive number of primers required to amplify all the genes having primers with the given melting temperatures. For the two degenerate *S. Cerevisiae* data sets we obtain more dramatic improvements of 48% and 54% respectively. For the three non-degenerate *S. Pombe* data sets we obtain improvements of 29%, 41% and 43% percent reductions in the primer set size required to obtain amplifications of all genes. For the only degenerate *S. Pombe* data set we could run our heuristic on we obtained a 50% per cent reduction in the number of primers required.

The trends noted in the previous subsection remain true for the new heuristic. However in all cases we can see that the solutions computed with this heuristic are of slower lower quality that our initial greedy heuristic.

Even with our initial heuristics and low degrees of degeneracy, our designs save the cost of thousands of primers on every data set. Further, the substantial gaps between our lower bounds and heuristic costs suggest that there is substantial room for further optimization by better heuristics. The increased savings from even such small amounts of degeneracy bodes well for substantial future improvements. Improving the heuristics given to improve performance is the subject of our current work.

ACKNOWLEDGMENTS

We thank Bruce Futcher and Janet Leatherwood for their help and interest in this project.

REFERENCES

Asahiro, Y., Iwama, K., Tamaki, H. & T., T. (2000). Greedily finding a dense subgraph. *J. Algorithms*, **34**, 203–221.

Charikar, M. (2000). Greedy approximation algorithms for finding dense components in a graph. In *Proc. APPROX 2000*. Lecture Notes in Computer Science 1913, Springer Verlag, pp. 84–95.

Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D. & Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, **2**, 65–73.

Doi, K. & Imai, H. (1999). A greedy algorithm for minimizing the number of primers in multiple pcr experiments. *Japanese Society for Bioinformatics Journal*.

Feige, U. (1996). A threshold of $\ln n$ for approximating set cover. In *Proc. of the 28th Ann. ACM Symp. on Theory of Computing*.

Feige, U. & Seltser, M. (1997). On the densest k-subgraph problems. Technical Report CS97-16, The Weizmann Institute.

Gallo, G., Grigoriadis, M. & Tarjan, R. (1989). A fast parametric maximum flow algorithm and applications. *SIAM J. Computing*, **18**, 30–55.

Gasch, A., Spellman, P., Kao, C., Carmen-Harel, O., Eisen, M., Storz, G., Botstein, D. & Brown, P. (2000). Genomic expression programs in the response of yeast cells to environment changes. *Molecular Biology of the Cell*, **11**, 4241–4257.

Guibas, L. & Odlyzko, A. (1981). String overlaps, pattern matching, and non-transitive games. *J. Combinatorial Theory (Series A)*, **23**, 183–208.

Hughes, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Coffey, E., Dai, H., He, Y., Kidd, M., King, A., Meyer, M., Slade, D., P.Y., L., Stepaniants, S., D.D, S., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M. & Friend, S. (2000). Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Maier, E., Maier-Ewert, S., Bancroft, D. & Lehrach, H. (1997). Automated array technologies for gene expression profiling. *DDT*, **8**, 315–324.

NCBI (2002). genomes web page. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genome.

Pearson, W., Robins, G., Wrege, D. & Zhang, T. (1996). On the primer selection problem in polymerase chain reaction experiments. *Discrete Applied Mathematics*, **71**, 231–246.

Rose, T., Schultz, E., Henikoff, J., Pietrokovski, S., McCallum, C. & Henikoff, S. (1998). Consensus-degenerate hybrid oligonucleotide primers for amplification of distantly-related sequences. *Nucleic Acids Research*, **26**, 1628–1635.

Rozen, S. & Skaletsky, H. J. (1998). Primer3. Code available at http://www-genome.wi.mit.edu/genome_software/other/primer3.html.

Schena, M., Shalon, D., Davis, R. & Brown, P. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467–470.

Shamir, R. & Linhart, C. (2001). The degenerate primer selection problem. RECOMB poster session.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, **9**, 3273–3297.

Wilf, H. (1987). Strings, substrings, and the nearest integer function. *Amer. Math. Monthly*, **94**, 855–860.