International Journal of Hybrid Information Technology
Vol.8, No.2 (2015), pp.355-366
http://dx.doi.org/10.14257/ijhit.2015.8.2.33

Study of Hybrid DNA Physical Mapping Based on Approximation Algorithm with Errors

Zhenglong Liu[1], Yanmei Yang[2], Yujun Luo[1*] and Hongping Wang[3]

[1]*Department of Mathematics and Computer of North-SiChuan Medical College, NanChong, SiChuan, 637007, China*
[2]*Mathematics & Information College of West China Normal Unviersity NanChong, SiChuan, 637003, China*
[3]*Department of information management, Nanchong Professional Technic College, NanChong, SiChuan,China 637007)*
nsmclzl@163.com

Abstract

A human chromosome is a DNA molecule with approximately 10^8 base pairs. The techniques developed to date for sequencing are restricted to pieces of DNA with up to tens of thousands of base pairs. This means that when a piece is sequenced, only an extremely small part of a chromosome can be seen. Molecular biologists use special techniques to deal with DNA molecules comparable in size to a chromosome. These techniques enable them to create maps of an entire chromosome or of significant fractions of chromosomes. Computational techniques were studied that could potentially aid biologists in the map-generation process. An algorithm that solves the consecutive 1s problem was studied. Such a problem is a good model of hybridization mapping when there are no errors and when probes are unique. If errors are present, another approach is needed, and the approximation algorithm is a prospective problem solver for hybridization physical mapping of DNA with errors.

Keywords: *Physical Mapping, DNA, Hybridization, Probes matrix, Greedy TSP, Consecutive 1s Problem (C1P)*

1. Introduction

A physical map of a piece of DNA shows the location of certain markers in a molecule. These markers are small but precisely defined sequences. These maps help molecular biologists explore genomes further. For example, if a certain stretch of DNA has been completely sequenced, revealing a sequence S, and if it is known which chromosome S came from, and a physical map of this chromosome is available, an attempt can be made to find one of the map's markers in S. If successful, S has been located in the chromosome [1-2].

How are these maps made? The first task is to obtain several copies of the DNA molecule that is to be mapped. Each copy must then be broken into several fragments, using restriction enzymes. Mapping is done by carefully comparing the subsequent fragments and carefully observing overlap [3]. For the most part, a fragment of a DNA piece is still too long to be sequenced, so overlap information is obtained by generating fingerprints of the fragments. A fingerprint describes part of the information contained in a fragment in a unique way, just like our fingerprints uniquely describe a part of ourselves [4]. Two popular ways of acquiring DNA fingerprints are restriction site analysis and hybridization.

Just as in other problems from molecular biology, possible lack of information and the presence of numerous experimental errors make the physical mapping problem especially

* Corresponding Author

ISSN: 1738-9968 IJHIT

hard. In particular, it may not be possible with a given collection of fragments to obtain one contiguous physical map. This may happen simple because the fragmentation process did not produce fragments covering certain section of the target DNA. When this happens, the physical map pieces are called contigs. This is a feature common to all physical mapping processes.

We studied an algorithm that solves the consecutive 1s problem saw such a problem is good model of hybridization mapping when there are no errors and when probes are unique. If errors are present, some other approach is needed, and that is the subject of this section.

First examine the effect errors can have on a clones x probes binary matrix M. Suppose M is presented to use with the true column permutation. Given one row, if there are no errors, all its 1s will be consecutive [1, 5]. If a row corresponds to a chimera clone, where two fragments were joined, we will see two blocks of 1s separated by some number of 0s(assuming no other errors are present in this row).We will call a consecutive block of 0s bordered by 1s a gap. Notice that this is different from the use of the term gap in other researchers. We can thus say that a gap was created in this row because of the chimera fragment [2]. If, on another row there is false negative, the corresponding 0 mat separate two blocks of 1s,creating another gap, as shown below:

0 1 1 0 1 1 1 1 0 0

↑

A false negative

The gap will not be created if the probe was leftmost and rightmost for this clone. Finally, a false positive may split a block of 0s in two, thus possibly creating yet another gap. In this way we see that there is a close correspondence between errors and gaps in the matrix. Given the basic assumption that we want to avoid explaining gaps by experimental error as much as possible, a reasonable approach is to try to find a permutation where the total number of gaps in the matrix is minimum, such an approach has the desirable property that, if there is a C1P permutation, it will have the minimum number if gaps. In other words, gap minimization can be seen as a generalization of the consecutive 1s problem we have mentioned in paper that some extensions to the C1P are NP-hard [3-4]. Such is the case also with the gap minimization problem just sketched. However, for this particular NP-hard problem, we can use many special techniques to get approximate solutions that we can expect to be reasonably good, as we show next.

2. TSP Graph Model

2.1. Build DNA Physical Mapping Model

It is believed that gap minimization is equivalent to solving a well-known graph problem, the traveling salesman problem (TSP).

The input to this version of the TSP is a complete undirected weighted graph G. The vertices of G correspond to columns of the clones x probes binary matrix M; that is, they correspond to probes. For reasons that will soon become clear, we also have to add an extra column to M filled with zeros, and G must have the corresponding vertex [6]. The weight on each edge of G is the number of rows where the two corresponding columns differ (this is also known as the hamming distance between the rows), For example, in table I we have an example of binary matrix, and in figure 1 we see the corresponding graph [5]. We will now argue that a minimum-weight cycle in G corresponds to a column permutation in M with the least number of gaps.

Table 1. Clones x Probe Matrix with Added Column p_6^*

clones	p_1	p_2	p_3	p_4	p_5	p_6*
c_1	1	1	1	0	0	0
c_2	0	1	1	1	0	0
c_3	1	0	0	1	1	0
c_4	1	1	1	1	0	0

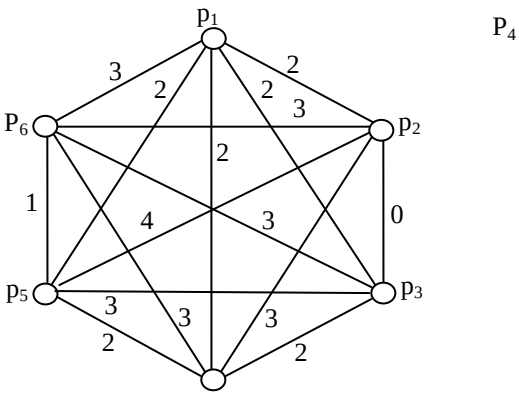

Figure 1. TSP Graph for Matrix of Table 1

2.2. Minimum-weight Cycle Computation of G Graph

Note that given a permutation of columns, a gap in a row means that at a certain point there is a transition from 1 to 0 and further on, a transition from 0 to 1. Therefore, for each gap there are two transitions and each gap contributes exactly two to the weights of the cycle corresponding to the given column permutation [7]. However, edge weights may also be increased by external transitions. That is, there may be transitions between elements in external (1 or m) columns, and these do not correspond to gaps. To ensure that every row has a pair of external transitions, an extra column of zeros is included in column m+1. Without such a column, cycles in the graph correspond to permutations where consecutive 1s are allowed to wrap around in each row, and this should not happen. So, the relationship between cycles and permutations now becomes:

Cycle weight=number of gap transitions+2n

This means that for a given n the minimizing cycle weight is the same as minimizing the number of gaps.

2.3. NP-hard Problem

The previous exercise showed how to reduce the gap minimization issue to the TSP. It is well known that the TSP is an NP-hard problem, so in principle this does not accomplish much. However, a wide array of techniques is available to solve or approximate the traveling salesman problem, and these techniques can be used in this paper [8-9]. The mere existence of such techniques is not enough to provide confidence that solving traveling salesman problems will deliver the true probe permutation answer. A guarantee is needed that the solutions gained are in some sense close to the correct solutions. This paper will show algorithm feasibility.

Before that, let us go back to the gap minimization problem and present an example of the ideas outlined at the end of paper. We have defined above a function that, given an input matrix, returns the total number of gaps in the matrix. Return the total number of gaps in the matrix. We have further argued that obtaining a permutation that has the minimum value for this function (or approximately minimum) will help us find the true column permution. Since we do not have a guarantee that the true permutation will be among the solution we find, we should look for other functions that might also be helpful. The idea is that by carefully defining several such optimization functions and developing algorithms for them we will increase the likelihood of hitting upon the true solution. In particular, it is reasonable to expect that the true solution will be in the intersection of all solution sets. This will only be the case. however, if each function represents one property that true solution do have or are likely to have.

Here is an example of another optimization function. One possible drawback of gap minimization is that in a permutation with a minimum value for this function one or a few rows may have many gaps, while others may have none [9]. Having many gaps in one row is undesirable, since it would mean that one clone was subject to many more errors than other clones, which contradicts laboratory experience. Therefore we could try to minimize the number of gaps per row. We leave as exercise 15 how to show that we can still use the preceding graph model. The resulting graph problem is known as the bottleneck traveling salesman problem, which is also NP-hard.

3. Main Title Algorithm Feasibility Analysis

3.1. DNA Physical Mapping Probe Permutation

In this paper proof will be presented that the TSP approach outlined in the last section will provide, with a high probability, the true permutation. The proof depends on two basic assumptions: that the number of probes is sufficiently large, and that the mapping process obeys a certain mathematical model [7]. This model appears to be a good representation of what occurs in large mapping projects. The model is described next.

First we assume that the DNA molecule we are dealing with is so long that we may think of it as an interval on the real line, extending from 0 to N. The clones are subintervals of this long interval, and we assume that all of them have the same length; for convenience each clone is one unit long. To simplify the exposition, will speak of clone permutation rather than probe permutation. The unit length assumption makes them equivalent. This means that we will be looking for consecutive 1s in columns [8]. Not in rows, and that each vertex in the associated TSP will correspond to a clone. We use each clone's left endpoint as a clone locator, We, of course, do not know the precise position of each clone along the molecule; and because we are dealing with hybridization, all we will be able to determine is relative clone order.

A critical feature of this model is clone distribution along the target DNA. We will assume that each clone's position is an independent random variable, That clone locators are distributed uniformly over [0, N-1],and that the clones cover the interval [0,N](that is, for every subinterval I of [0,N],there always exists at least one clone C such that $C \cap I \neq \phi$).

Another important aspect of the model is probe distribution. We will not assume that each probe is unique; instead, we will assume that each probe occurs rarely along the target DNA. More formally, we will say that the occurrences of a given probe obey a Poisson process with rare λ .Moreover; the Poisson process of may one probe is independent of all the others. This

part of the model lets us immediately obtain an expression for the probability of a specific probe hybridizing to a clone. The expression is:

$$\Pr\{\text{a given probe occurs k times in a given clone}\} = e^{-\lambda}(\lambda^{k} / k!) \quad (1)$$

This expression can be obtained from any textbook formula for Poisson processes, using the fact that our clones are unit length intervals.

3.2. Clones x Probes Binary Matrix

This completes the model description. We shall now argue that given a clones x probes binary matrix, the row permutation given by solving the associated TSP is a good approximation to the true permutation, in the following precise sense: The probability that both permutations are the same tends to 1 as the number of probes increases. Note that the number of probes is fixed for a given instance if the problem. We are just claiming that in larger and larger instances of the problem (and we are measuring size here by number of probes) the TSP permutation will be the same as the true permutation with higher and higher probability.

To prove this claim, we must argue in term of graph weights, or more appropriately, clone distances. As we mentioned ,the weight of each edge of the graph associated with input matrix M is called the Hamming distance between its two endpoints(clones).We denote by h_{ij} the Hamming distance between clone i and clone j.We can also think of the true distance between clones. Do noting a clone's coordinates by l (left) and r (right) we can define this distance to be given that clones are all of the same size.

$$t_{ij} = \left|l_j - l_i\right| + \left|r_j - r_i\right| = 2\left|l_j - l_i\right| \quad (2)$$

Suppose now that we knew all true distances. Then it is clear that the largest such distance would give us the clones that ate farthest apart, which is to say, the clones that occur at opposite ends of the interval[0.N].The next largest such distance gives us another similar pair that occurs between the precious two, and so on. This means that given the true distance we ate able to obtain the true clone permutation, which is not surprising [9]. However, we have distances h_{ij} and not distances t_{ij} ,But because we are trying to obtain only the true relative order of clones, it would suffice if we could say that, given any four clones i, j, r and $s, h_{ij} < h_{rs}$ implies that $t_{ij} < t_{rs}$ and vice versa. The reason is that some notion of order between clone distances was all we needed to place clones relative to each order using the true distances. If we prove that the probability that $h_{ij} < h_{rs} \Leftrightarrow t_{ij} < t_{rs}$ tends to 1 as the number of probes increases, we will have the result we need.

Let us look at a pair of clones i and j , and let us find the probability that a certain probe p contributes to their Hamming distance. This will happen if probe p hybridizes to i but not to j or vice versa. Referring to Equation(1)we see that the probability that probe p does not occur on clone j is $e^{-\lambda}$,since $k = 0$ .On the other hand, the probability of probe p occurring at least once in clone i is the same as the complement of the probability of probe p not occurring in that paper of i that does not overlap j .if this overlap is z_{ij} ,we obtain the following result:

$$\Pr\{p \text{ hybridizes to } i \text{ and not to } j \text{ or vice versa} = p_{ij} = 2e^{-\lambda}(1 - e^{-\lambda(1-z_{ij})}).\} \quad (3)$$

3.3. Hybridization Probability Computational

This essentially means that there is a well-defined probability of this event happening. We can now take into account all probes, and each of them will have a certain probability of hybridizing to clones i and j .Assume we have m probes, and consider h_{ij} / m this represents the mean contribution of each probe to the Hamming distance between clone i and j .(Note that a probe's contribution to the Hamming distance is either 0 or 1)If the number of probes is large, we can invoke the law of large numbers and say that h_{ij} / m approaches p_{ij} .Or, for any fixed small positive real number ε ,that

$$\Pr\{\left|\frac{h_{ij}}{m} - p_{ij}\right| > \varepsilon\} \to 0 \quad (4)$$

As $m \to \infty$ .Note now that since clones are unit length, the true distance between two overlapping clones i and j with and intersection measuring z_{ij} is given by (see Equation 2).This allows us to substitute t_{ij} for z_{ij} in Equation(3),effectively showing us that $t_{ij} > t_{rs}$ if and only if $p_{ij} > p_{rs}$,for pairs of clones i, j and r, s .But if this is the case, then we can say that $\Pr\{\left|t_{ij} - h_{ij}\right| > \varepsilon\} \to 0$

This implies that $h_{ij} < h_{rs} \Leftrightarrow t_{ij} < t_{rs}$, which is what we waned to prove.

4. Computational Practice

4.1. Hybridization Mapping Algorithms

In this paper we present some actual results of computational tests with algorithms for the hybridization mapping problem. Before presenting these results, however, we must make considerations regarding the way the results of such tests can be interpreted. Some of these considerations are valid for many problems in computational biology [10].

Initially we shall look at the input data using another graph. By doing so it will become clearer what we can expect of any algorithm that tries to obtain the true probe permutation. This graph will also be used in the paper.

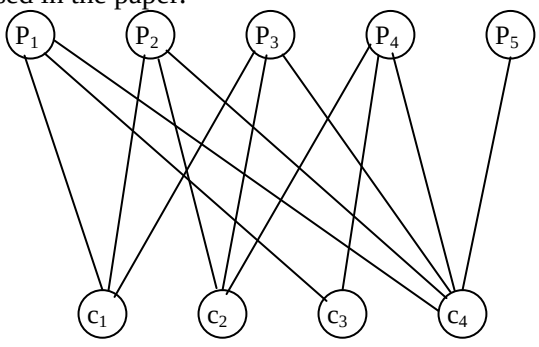

Figure 2. H Hybridization Graph H Corresponding to Hybridization Matrix from Table I, without the Added Column

We define the hybridization graph H as a bipartite graph (U,V,E)that is built using information from the hybridization matrix: Clones are the vertices of U partition, and probes

are the vertices of the V partition. There is an edge between two vertices if the corresponding probe hybridized to the corresponding clone. In Figure 2 we can see the bipartite graph H that was built based on matrix M shown in Table I, but excluding the all-zeros column.

The firs thing to notice is that H may not be connected, even if all entries in the hybridization matrix are correct. If this is the case, then no matter how good our algorithm is, we will not be able to tell the relative order between probes that belong to different connected component; the information to do so is simply not present in the hybridization matrix [3]. A connected component may be as simply as a singleton vertex, meaning that there is a probe that did not hybridize to any clone or a clone that was not hybridized to any probe. Another observation is that there may be redundant probes, or probes that hybridize to exactly the same set of clones. This could happen if the probes, although different, hybridize to parts of the target DNA that are close together. It could also happen if certain clones for that particular DNA stretch are missing, leacing some probes without any positive hybridization result.

4.2. Consecutive 1s Problem Feature

Connected components of H show up when we solve the corresponding consecutive 1s problem. Redundant probes can also be easily seen in the hybridization matrix: They are columns that have exactly the same 1s and 0s.But if there are errors we may get wrong information regarding the number and structure of connected components of H and whether a probe pair is redundant. We say that errors may mask these properties [10]. So notice the difficult situation that errors creat: The input matrix without errors may lack information necessary to find the true permutation (for example, the errorless H has several components), and we may have a lot of trouble just to recognize that lack of information (because our errorful H has only one component). Assuming matrix in table I is errorless, probes p_2 and p_3 are redundant, but if there were a false positive between clone c_3 and probe p_3 we would not be able to recognize that.

Given this situation, it is clear that evaluation of a mapping algorithm is a difficult task (in addition to mapping itself).We will now take a look at how we can evaluate such algorithms assuming that we somehow know the correct answer to any mapping problem. This can be done, for example, if we use a computer program to generate artificial instances of mapping problems, simulating experimental errors. If such instances, are faithful to real instances, it should be clear from the above discussion that the input matrix mat lack information to enable an algorithm to determine the true probe order. Therefore we should try to evaluate how "close" the solution found by a particular mapping algorithm is to the true probe order. The question now becomes: How should we define closeness in this context? At the moment there is no consensus on how to do this. However, to give some idea of the performance of current mapping algorithms in practice, we will present as an example one definition that has appeared in the literature. It is a reasonable definition, but even if becomes widely accepted, it may still undergo some refinements.

We will measure a mapping algorithm by the fraction of strong adjacencies reported by it. Strong adjacencies are defined in terms of the number b of blocks of consecutive 1s present in a hybridization matrix with a given probe permutation $\pi = p_1, p_2, \cdots, p_m$ .We analyze the effect of translocation, which are operations that reverse the order of a set of consecutive probes. We say that two adjacent probes p_i and p_{i+1} represent a strong adjacency if placing these probes apart by any translocation increase b in each row when such as increase takes

place we have some evidence(albeit not conclusion) that probes p_i and p_{i+1} should stay adjacent in all solution.

Based on this concept, we can define the strong adjacency cost of a given permutation. This is given by the formula

$$100(\frac{1}{m-1}\sum_{i=0}^{m-2}\sigma_i) \quad (5)$$

Where $\sigma_i = 1$ if p_i and p_{i+1} is a strong adjacency in the true permutation but these probes are not adjacent in the proposed permutation, and $\sigma_i = 0$ otherwise. Note that the cost is given as a percentage. Good permutations should have low strong adjacency cost.

4.3. Greedy TSP and Random Algorithm Compare

With this definition we are finally able to give the reader an idea of algorithm performance. Table Ⅱ presents the strong adjacency cost of two algorithms. One of them is "random": A random probe permutation is selected. The other is based on the TSP approach; this is how it works [11]. Given the TSP graph the algorithm builds a cycle by choosing pairs of vertices and making them adjacent on path. The pair (u, v) chosen at each iteration must fulfill the following conditions: If both u and v already belong to paths, and these paths are different, the paths can be joined only if u and v are endpoints in their respective paths(that is, they each have just one neighbor);and they must be the closest among all qualifying pair. After all vertices are on the same path, that path is closed forming a cycle. This solution is then submitted to another algorithm, which tries to improve the solution by applying another heuristic.

In table Ⅱ we can see that the TSP-based approach (called "greedy") performs well compared to "random." The paper from which these results were obtained presents the performances of three other, much more sophisticated algorithms, and the results are similar to those shown above for "greedy TSP."this can be seen as a point in favor of the TSP approach, but in a sense it is yet another measure of how difficult the mapping problem is. The table also shows that in the presence of false negatives the solution of "greedy TSP" was fairly poor. It is fair to assume that results would be even worse if all kinds of errors were combined in the same instance. This motivates our next paper, in which wee present a heuristic that appears to be robust in the presence of false negative.

Table 2. Two Algorithms Efficiency Compare

Algorithm	C 1P	Chimera	False Positives	False Negatives
Greedy TSP	1.9	0.9	16.0	28.3
Random	84.6	89.7	94.4	94.9

Strong adjacency costs for two algorithms on matrices with different kinds for errors. Error rate are indicated in the heading of each column (only one type of error per column). Coverage in all cases in 10, where coverage is the ratio between the total length of all clones and target DNA length.

4.4. Heuristics for Hybridization Mapping

As the previous papers have shown, mapping is a difficult problem, and no general and good algorithms for it have been found. As a consequence, what we see in practice is that researchers resort to various heuristics to help them arrive at a solution. In this paper we present two such heuristics that have yielded good results in hybridization mapping projects.

As noted above, chimera clones occur with high frequency ion clone libraries, and their presence brings serious problems to any mapping algorithm [8]. In this paper we present a simple heuristic that tire to split chimera clones into fragments. Such a heuristic is very useful as a screening procedure, which can be used as a preprocessing step before employing more sophisticated techniques.

This idea is very simple: if a clone is chimera (and let us assumes it is composed of only two fragments), the probes that hybridize to one of its fragments should not be related to the probes hybridizing to the other fragment. The key here is of course the concept of "relatedness". This is made concrete by looking at the following describes [9].

Take clone i and the set of probes that hybridize to it, P_i .We crate a graph $H_i = (P_i, E)$ for every clone i . We create an edge between two probes from P_i if they hybridize to a clone other than i .If the resulting graph is connected, we say that i is not chimera. If it has more than one component, we say that i is chimera and we replace i by new "artificial" clone, each new clone given by a connected component of H_i . This method can be refined by requiring that an edge exists between probes p and q in H_i only if p and q hybridize to at least k other clones, where k is a parameter depending on the particular problem at hand. See Figure 3 for an example.

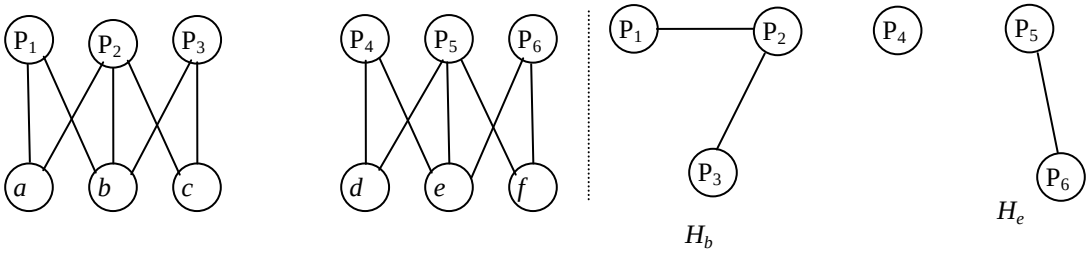

Figure 3. Hybridization Graph and Connect Graph

Above the line we show the hybridization graph H for some clones × probes matrix. Below the line at left is graph Hb for clone b. This clone is probably not chimera, because Hb is connected. At right is graph He for clone e. This clone could be chimera, because He is not connected.

Experience has shown that this heuristic behaves well. On the other hand it may consider a clone chimera when in fact it is not. Therefore, another useful heuristic would be one that combines two clones that are actually one.

4.5. Obtaining a Good Probe Ordering

The heuristic presented here is more ambitious than the one we saw in the previous paper. It aims at actually solving the problem that is, obtaining a permutation of the probes. The idea is to estimate for every probe p the number of probes to its left and the number of probe to its

right by looking at the hybridization graph H . We can then sort the probes using this estimate and thus obtain one "good" permutation.

Given a probe we will be able to count the number of probes to its left and to its right if we can somehow split the other probes into two separate components: one left component and one right component. Probes near the ends of the target DNA may not split the others in two; therefore, the probes for which we can obtain two components will be called splitters [6]. The method ofr detecting splitters is described next.

4.6. DNA Physical Mapping Algorithm Design

Give a probe p build a set of vertices s_p (whose elements are clones and probes)by including p in s_p and every other probe that shares clone with p .Include in s_p all clones that are incident to any probe in s_p .Now remove all vertices in s_p from H. We will say that p is a splitter if the resulting graph has exactly two components. Using this method. the heuristic for obtaining a good probe ordering is described in algorithm.

Algorithm probe permutation heuristic
Input: clones x probes hybridization matrix
Output: a "good" permutation for the probes
for every splitter I do determine the components Ai and Bi
for each probe p do initialize l_p and r_p with zero
select a pair(A_k , B_k)arbitrarily
for each probe p do
for each pair of components(A_i , B_i) do
if $p \in A_i$ then
if $|A_i \cap A_k| > |A_i \cap B_k|$ then
increment l_p
else
increment r_p
else if $p \in B_i$ then
if $|B_i \cap A_k| > |B_i \cap B_k|$ then
increment l_p
else
increment r_p
sort probes in decreasing order of $l_p - r_p$
heuristic to obtain a "good" probe permutation.

For each probe p , the algorithm keeps two counter, l_p and r_p , recording the nember of left and right regions that contain p . These counters are initialized with zero and are incremented using the splitters. It is not obvious how to do this, because although we know the two components A_i and B_i of a splitter i , we cannot tell which one goes to the left or to

the right [12]. To solve this difficulty we rely on a fixed arbitrary splitter k that we use as a reference. We assume that A_k lies to the left of B_k. Then, given another splitter i, a component X of i (either A_i or B_i)is the leftmost one when $|X \cap A_k| > |X \cap B_k|$. Once we have the final counts, we sort probes so that the ones with higher left bias $l_p - r_p$ come first. A simple improvement to this heuristic is in the choice of splitter k. The more "central" this splitter, the better should the results be.

4.7. Algorithm Implications

Before describing algorithmic technique for obtaining physical maps, the constraints of the difficulty of the problem can be explored. The first consideration is that this is an attempt to solve a real-life problem, not an abstract mathematical problem [13]. The reality is that the true order of the clones in a target DNA is a goal that many seek. It is a combinatorial problem: There is an infinite number of possible orders, but only one of them is the true order. Discovering the true order by means of abstract models that give rise to the optimization problem is one way to attempt to solve this issue [10], but these problems are very difficult and they are abstractions of an even more difficult problem. To say the least, optimal solutions will not be discovered quickly.

As with the heuristic of the previous paper, experience has shown that this one performs well. In particular, it is relatively robust with respect to false negatives. In addition, one can envisage this heuristic also used as delivering its result to other algorithms or heuristics that could try to improve the solution.

5. Conclusion

This study presented two techniques that yield data for DNA mapping: digestion by restriction enzymes, and hybridization experiences. The first method employed restriction enzyme data to measure and compare the corresponding fragment lengths. However, this led to the double digest and partial digest problems. The double digest problem was NP-complete. The second method involved data regarding fragments that can be reconstructed by determining overlaps between the fragments based on "fingerprints". This can be modeled in various ways by interval graphs, although most models result in a NP-complete problem. DNA mapping is a quandary that many pursue the solution for. Although there are good techniques and algorithms that shed a little light on this issue, the perfect instrument to definitively solve it continues to be sought.

Acknowledgements

A Project Supported by Scientific Research Fund of SiChuan Provincial Education Department (Grant No: 15ZA0217 & 12ZB040).

References

[1] A. Coulson, J. Sulston, S. Brenner and J. Kam, "Toward a physical map of the genome of the nematode Caenorhabditiselegans [J]", Proc. Nat1. Acad. Sci., USA, vol. 83, no. 20, **(1986)**, pp. 7821–7825.
[2] R. M. Karp, "Mapping the genome: some combinational problems arising in molecular biology [c]", In Proceedings of the twenty-fifth annual ACM symposium theory of computing, **(1993)**, pp.278–285.
[3] M. C. Golumbic, H. Kaolan and R. Shamir, "On the complexity of physical mapping [J]",Advance in applied Mathematics, vol. 15, **(1994)**, pp.251–261.
[4] D. Greenberg and S. Istrail, "Physical mapping by STS Hybridization: Algorithmic strategies and the challenge of software [J]", Journal of computational biology, vol. 2, no. 2, **(1995)**, pp. 219–274.

[5] R. M. Karp, C. Kenyon and O. Waarts, "Error resilient DNA computation [C]", In Proceedings of the seventh annual ACM-SIMAM Symposium on Discrete Algorithms, ,**(1996),** pp. 458–467.
[6] H. B. Zhang and R. A. Wing, "Physical mapping of the rice genome with BACs", Plant Molecular Biology, vol. 35, no. 12, **(1997)**, pp. 115-12.
[7] L. Goldstein and M. S. Waterman, "Mapping DNA by stochastic relaxation [J]", Advances in Applied Mathematics, vol. 8, **(1997)**, pp.194–207.
[8] U. Hohmann, G. Jacobs, A. Telgmann, R. M. Gaafar, S. Alam and C. Jung, "A bacterial artificial chromosome (BAC) library of sugar beet and a physical map of the region en compassing the bolting gene [J]", Mo1. Gen. Genomics, vol. 269, no. 1, **(2003)**, pp. 126–136.
[9] F. Tang and T. Zhang, "Construction and Application of Large insert Clones based Physical Map [J]", Genomics and Applied Bio1ogy, vol. 28, no. 1, **(2009)**, pp. 195–201.
[10] Z. Xu, R. J. Kohel, G. Song, J. Cho, J. Yu, S. Yu, J. Tomkins and J. Z. Yu, "An integrated genetic and physical map of homologous chromosomes 12 and 26 in upland cotton (G. hitsutum L.) [J]", BMC Genomics, **v**ol. 9, **(2008)**, pp. 108–124.
[11] P. A. Perzner, "DNA physical mapping and alternating Eulerian cycles in colored graphs [J]", Algorithmica, vol. 13, no. 1 /2, **(1995)**, pp.77–85.
[12] W.-P. Chen, C.-L. Hung, S.-J. Jan, E. Tsai and Y.-L. Lin, "Novel and efficient tag SNPs selection algorithms [J]", Bio-Medical Materials and Engineering, vol. 24, no. 1, **(2013)**, pp. 1383-1389.
[13] T. Dlugosz, "Uncertainity Analysis of Selected Sources of Errors in Bioelectromagnetic Investigations [J]", Bio-Medical Materials and Engineering, vol. 24, no. 1, **(2013)**, pp. 609-617.

Authors

Zhenglong Liu, he was born in 1976, he is an associate professor of Department of Computer and mathematics, Nort-Sichuan Medical University, Nanchong, China. He received his B.S. degree from Sichuan Normal University in 2000. He received his M. S. degree from Sichuan University in 2008. His main areas of research include biological Informatics, artificial intelligence, Computational molecular biology.

Yanmei Yang, she was born in 1977, she is an associate professor of College of mathematics and information, China west normal University, Nanchong, China. She received his B.S. degree from Southwest Petroleum University in 1999. She received his M. S. degree from Southwest Petroleum University in 2005. Her current research interests include Technical and economic management, decision analysis, asset evaluation.

Yunjun Luo, he was born in 1972, he is an associate professor of the Department of Computer and mathematics, Nort-Sichuan Medical University, Nanchong, China. He received his B.S. degree from Sichuan Normal University in 1996 He received his M. S. degree from Chengdu University of Technology in 2002. His teaching interests focus on the application of computer, signal processing.

