Inapproximability of Flip-Cut, Shift-Cut and other problems from Optical Mapping

Laxmi Parida*

August 14, 1997

Abstract

Optical Mapping is an emerging technology for constructing ordered restriction maps of DNA molecules. The study of the complexity of the problems arising in Optical Mapping has generated considerable interest amongst computer science researchers [9], [10], [11], [12], [13]. In this paper we examine the complexity of these problems.

Optical Mapping leads to various computational problems such as the Binary Flip Cut (BFC) problem [9, 10], the Weighted Flip Cut (WFC) problem [10] the Exclusive Binary Flip Cut (EBFC) problem [9, 10], the Binary Shift Cut (BSC) problem [12, 11], the Binary Partition Cut (BPC) [12] problem and others. The complexity and the hardness of the BFC problem [9], the WFC problem [10] were not known. Using the technique of *gap-preserving* reduction (see [7, 8]) of the max-cut problem, we show that BFC and WFC problems are MAX SNP-hard [6] and achieving an approximation ratio $1 - \Upsilon/7$ for these problems is NP-hard, where Υ denotes the upper bound on the polynomial time approximation factor of the well-known max cut problem. A slight variation of BFC, $\mathrm{BFC}_{\max K}$, had been shown to be NP-hard [12]; we improve the result to show that $\mathrm{BFC}_{\max K}$ is MAX SNP-hard and achieving an approximation ratio $(1 - \Upsilon/7)\frac{p_{max}}{p_{min}}$ for $\mathrm{BFC}_{\max K}$ is NP-hard, where $p_{\min}$ and $p_{\max}$ are the minimum and maximum of the digestion rates in the given problem. The EBFC problem was shown to be NP-Complete [11]; we improve this result to show that EBFC is MAX SNP-hard and achieving an approximation ratio $1 - \Upsilon/7$ for EBFC is NP-hard. However, a dense instance of the EBFC problem does have a PTAS [9].

The Binary Partition Cut (modeling spurious molecules) problem has been shown to be NP-Complete [12]: we show, in this paper, that a (reasonable) unrestrained version of it has an efficient polynomial time algorithm. A variation of the Binary Shift Cut (modeling missing fragments) $\mathrm{BSC}_{\max K}$, had been shown to be NP-hard [12]; we show both the versions of this problem to be MAX SNP-hard and achieving an approximation ratio $1 - \Upsilon/6$ for BSC and a ratio $(1 - \Upsilon/6)\frac{p_{max}}{p_{min}}$ for $\mathrm{BSC}_{\max K}$ is NP-hard. In addition, we show that d-wise Match (dM) problem [13] is MAX SNP-hard and achieving an approximation ratio $1 - \Upsilon$ is NP-hard.

*Dept. of Computer Science, Courant Inst. of Math. Sciences, NYU, USA, parida@cs.nyu.edu.

1 Introduction

The ultimate goal of many efforts in Molecular Biology, including the Human Genome Project, is to determine the entire sequence of Human DNA and to extract genetic information from it. In this context an important step is to build *restriction maps* of portions of the DNA [1]. A restriction enzyme cleaves or cuts a DNA molecule at some fixed site called the *restriction site*. An ordered restriction map specifies the location of these identifiable markers or restriction sites along a DNA molecule. Construction of ordered restriction maps for eukaryotic chromosomes is laborious and difficult, in part because many of the current procedures for mapping and sequencing DNA were originally designed to analyze genes rather than genomes. A microscope-based technique called Optical Mapping [2], [3], [4] is a very promising emerging technology for rapid production of ordered restriction maps. The Optical Mapping process fixes elongated DNA molecules onto polylysine-treated glass surfaces. The fixation conditions are carefully controlled (to minimize DNA coil relaxation effects but allow enough relaxation) at restriction sites for their detection by flourescence microscopy. The size of the resulting individual restriction fragments is determined by relative fluorescence intensity and apparent molecular contour length measurements and other parameters. The fragment size information is used to obtain the ordered restriction maps.

Various combinatorial problems arising from Optical Mapping model the different kinds of errors that arise in the experimental and/or pre-processing stages. Some of these problems had been shown to be NP-Complete; in this paper we show that most of these problems are MAX SNP-hard [6], that is, they do not admit a Polynomial Time Approximation Scheme (PTAS), unless P=NP [8]. One exception is the Binary Partition Cut (BPC) problem that models spurious molecules, and can be solved in polynomial time under a reasonable model; this may be of particular interest to chemists who generate the data, since it may provide guidelines to keep the related computational problems simple and tractable. However, it is important to bear in mind that the hardness proofs are for the problems in their total generality. In real life, the data arise from a well-controlled (benevolent) process. Hence (alternate) efficient, practical solutions have been proposed– use of mean field annealing techniques [9], Exclusive Binary Flip Cut (EBFC) based approximation [10], branch-and-bound heuristics [12]. Also, the dense (number of 1's in a column is at least τ times the number of rows, for every column, $0 < \tau < 1$) EBFC has been shown to have a PTAS [10].

The sections are organized as follows: Section 2 defines the various errors used in the models, Section 3 describes the notation used and Section 4 describes the BFC, EBFC, WFC, dM and related problems. Sections 5 and 6 deal with the Binary Partition Cut (spurious molecules) and the Binary Shift Cut (missing fragments) problems respectively. Section 7 discusses the k-populations problem and also gives a polynomial time algorithm to the 2-populations problem under one of the models.

2 The Ordered Restriction Map Problem

We will define the problem informally as follows. Let us view this as a game played by Ann and John. John has a string S, of length n, of 0's and 1's. He makes m copies of this string and, using some process, *alters* the m copies in some controlled manner.

John assures Ann that the number of these alterations is not very large. Now, this altered set of m strings, called the *data set*, is made available to Ann and she is required to guess the original string S John started with. Ann makes a (reasonable) guess by providing an S'. The problem that Ann solves is the Ordered Restriction Map problem.

Let us now look at the (reasonable) alterations John can make.

1. **False Positives:** John can change some 0's to 1's in the m copies. But he must assure Ann that the number of such changes is very small.

   In practice, these may be due to actual false cuts or due to errors in the pre-processing stage.

2. **False Negatives:** John can change some 1's to 0's in the m copies. But he must assure Ann that the number of such changes is no more than mc_j for each column j. Note that in the absence of this restraint on John (and with False Positives), Ann will have no way of guessing a reasonable S'.

   $(1 - c_j) = p_j$ is the *digestion rate* of the experiment or the minimum number of 1's required for a column j to be designated a *consensus site* [1].

3. **Spurious Molecules:** John can throw out some, say k, molecules from this data set and throw in k random strings of 0's and 1's in its place.

   In practice, some "bad" molecules get into the sample population; these need to be invalidated and not used in the map computation.

4. **Sizing Errors:** John moves the positions of some 1's in a *small* neighborhood, that is, for some $\delta > 0$, he can move the position of a 1 in the molecule at j to anywhere between $j - \delta$ and $j + \delta$.

   This corresponds to the possible sizing errors of the fragments. The input data does not depict the location of restriction sites accurately because of the error inherent in measuring the lengths of fragments that remain after digestion by the restriction enzyme. Thus a 1 at some site in the molecule might in fact signal a restriction site in one of its neighbors. This fuzziness is the result of coarse resolution and discretization, other experimental errors, or

[1] It may be noted that if the number of false positives per column j is q_j, then Ann cannot make a reasonable guess, if the following holds: $p_j + q_j \approx 1$, for any j.

errors in preprocessing the data prior to constructing physical maps such as in the image processing phase.

5. Orientation Uncertainties: John flips some of the strings: if $s = x_1 x_2 \ldots x_{n-1} x_n$ is a string with $x_i = 0$ or $1, i = 1, 2, \ldots, n$, the flipped string is $x_n x_{n-1} \ldots x_2 x_1$.

   When the molecule is laid out on a surface, the left-to-right or right-to-left order is lost. However, the orientation information may be given in the data (using a more elaborate chemical protocol) with a vector arm on one fixed side of the molecule [2]. The model can view this as a consensus cut site at one end of the map. Notwithstanding this, there is a non-zero probability of the orientation of the molecule being still unknown.

6. Missing Fragments: John can remove some fragments of the string.

   This corresponds to fragments that get washed away during the experiment, which is common for BAC DNA, although not for cosmids and λDNA [12].

Circular Ordered Restriction Map Problem. If John take the string S and glues the two free ends producing a "seamless" ring, the corresponding problem is the circular DNA problem. In this version John makes m altered *rings* (instead of linear molecules as in the previous case) available to Ann. The seamlessness refers to Ann not having any information about where John glued the ends.

Ann provides an S' that minimizes the number of certain alterations that John has to make to produce the m data set from this S'. The alterations for which Ann has to pay a "price" are: (1) changing a 1 to 0, or (2) a 0 to 1 [2]. Any other change comes for free. Other ways of counting the cost will, of course, give rise to other cost functions. [13] gives a characterization of the various models.

3 Notation

The correspondence of the Ann & John game (see Section 2) to the Ordered Restriction Map problem is as follows: a string is a molecule, the length of the string corresponds to the number of sites on each molecule, the 1's in the string refer to cuts and a 0's refer to no-cuts at that site. The string S is called the map, and the 1's on S are the consensus cuts. The changes that John makes correspond to the various experimental and/or pre-processing inaccuracies that creep in at various stages.

[2] These two changes are usually not symmetric, since the former denotes missing a true cut and the latter denotes a false cut.

In the rest of the paper we will deal with problems which take into account subsets of the possible errors described above. Almost all the models deal with false positive and negative errors.

Let the data be represented in a $m \times n$ binary matrix $[M_{ij}]$ with each entry as either 0 or 1. Each row represents a molecule and each column refers to a site on the molecule: thus there are m molecules and n sites. A 1 at position (i, j) means that the j^{th} site (column) of the i^{th} molecule (row) is a cut. A 0 indicates the absence of a cut.

Alignment of the rows/molecules refers to assignment of the following:

1. Labeling a molecule as *spurious* or not.
2. Labeling the orientation of the molecule as *flipped* or not.
3. Assigning a *left-flushed* or *right-flushed* or any other positioning of each molecule.

A *map* is an n-length string that designates each site as a *consensus cut* site or not.

Cost of an alignment is a function (measure) of the alignment which we optimize. [13] discusses various cost functions. In this paper we deal with the following cost functions:

1. Given an alignment, maximize the total number of 1's in the consensus cut columns of the aligned molecules.
2. Given an alignment, maximize the the number of consensus cuts K.

We summarize the results that we present in the rest of the paper in Table 1.

4 The Binary Flip Cut (BFC) Problem

The BFC problem as formalized in [9], takes the following errors into account: (1) false positives, (2) false negatives and (3) orientations.

Given m molecules with n sites each, and, p_j as the digestion rate for column j, obtain an alignment of the molecules such that the following holds.

1. BFC problem [9]: The total number of 1's in the consensus cut columns, J, which is at least mp_J in each, is maximized.

Problem	Complexity class	Approx factor (upper bound)	Errors modeled
Exclusive Binary Flip Cut† (EBFC)	MAX SNP-hard	$1 - \Upsilon/7$	1) False positives 2) False negatives 3) Orientation Uncertainties
Binary Flip Cut* (BFC)	MAX SNP-hard	$1 - \Upsilon/7$	
$\mathrm{BFC}_{\max K}$†	MAX SNP-hard	$(1 - \Upsilon/7)\frac{p_{max}}{p_{min}}$	
d-wise Match* (dM)	MAX SNP-hard	$1 - \Upsilon$	
Consistency Graph* (CG)& Weighted Consistency Graph* (WCG)	MAX SNP-hard	$1 - \Upsilon$	
Weighted Flip Cut* (WFC)	MAX SNP-hard	$1 - \Upsilon/7$	1) False positives 2) False negatives 3) Orientation Uncertainties 4) Sizing errors
$\mathrm{WFC}_{\max K}$†	MAX SNP-hard	$(1 - \Upsilon/7)\frac{p_{max}}{p_{min}}$	
Binary Partition Cut* (BPC)	$\mathcal{P}$	–	1) False positives 2) False negatives 3) Good/spurious molecule
$\mathrm{BPC}_{\max K}$†	$\mathcal{P}$	–	
Binary Shift Cut* (BSC)	MAX SNP-hard	$1 - \Upsilon/6$	1) False positives 2) False negatives 3) Missing fragments
$\mathrm{BSC}_{\max K}$†	MAX SNP-hard	$(1 - \Upsilon/6)\frac{p_{max}}{p_{min}}$	

Table 1: Computational Problems from Optical Mapping: Problem* denotes unknown complexity until this paper and Problem† denotes the best known result for the hardness of this problem was that it was NP-complete. The Binary Partition Cut (BPC) problem has been modified (slightly) to admit a polynomial time solution. $p_{min} = \min_j p_j$, and $p_{max} = \max_j p_j$ are defined by the given problem. (Υ denotes the upper bound on the polynomial time approximation factor of the well-known max cut problem.)

2. $\text{BFC}_{\max K}$ [12]: The total number of consensus cut columns, K, which is at least mp_J in each consensus cut column, is maximized. This has been formalized by Anantharaman et al (problem 1 in [12]) and shown to be NP-Complete. We will show at the end of this section that this problem is MAX SNP-hard and give an upper bound on the polynomial time approximation factor of the problem.

3. dM Problem [13]: This is an alternate view to the problem modeling the same errors as BFC and is discussed at the end of this section (in Section 4.3).

We show a simple example, in Figure 1, that shows BFC and $\text{BFC}_{\max K}$ problems give rise to different optimal alignments and maps.

$$
\begin{array}{c}
\text{Input Problem} \\
\hline
\begin{array}{cccccc}
1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1
\end{array}
\end{array}
\qquad
\begin{array}{c}
\text{BFC} \\
\hline
\begin{array}{cccccc|c}
1 & 1 & 0 & 0 & 0 & 0 & \\
1 & 1 & 0 & 1 & 0 & 0 & \\
1 & 1 & 0 & 1 & 0 & 0 & \\
1 & 1 & 0 & 0 & 0 & 0 & * \\
\hline
1 & 1 & 0 & 1 & 0 & 0 & S'
\end{array}
\end{array}
\qquad
\begin{array}{c}
\text{BFC}_{\max K} \\
\hline
\begin{array}{cccccc|c}
1 & 1 & 0 & 0 & 0 & 0 & \\
1 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 1 & 0 & 1 & 1 & * \\
0 & 0 & 0 & 0 & 1 & 1 & \\
\hline
1 & 1 & 0 & 0 & 1 & 1 & S'
\end{array}
\end{array}
$$

Figure 1: An example to show different optimal configurations for the two different cost functions, BFC and $\text{BFC}_{\max K}$. It is assumed that $p_j = 1/2$ for all j. The optimal cost for the BFC problem is 10 (number of 1's in the consensus cut columns) with 3 consensus cut columns. The optimal cost for $\text{BFC}_{\max K}$ is 4 (the number of consensus cut columns). Note that the maps corresponding to the optimal configurations are different. The rows/molecules that are flipped are marked by asterisks.

Let us associate indicator variables X_i, $i = 1, 2, \ldots, m$, with every row which takes a value 1 if the molecule is flipped and 0 otherwise. Let Y_j, $j = 1, 2, \ldots, n$, be an indicator variable associated with every column that takes on a value of 1 if it is a consensus cut and 0 otherwise.

Define *conjugate* of column j as $\overline{j} = n - j + 1$.

BFC can be modeled as the following optimization problem:

$$\max \left\{ \sum_{j=1}^{n} Y_j \left(\sum_{i=1}^{m} \left(M_{ij}(1 - X_i) + M_{i\overline{j}} X_i \right) - mp_j \right) \right\}. \tag{1}$$

Note that the term mp_j is used to ensure that the number of 1's along a consensus cut site j (with the rows flipped, if required) is at least mp_j. In other words, for a given alignment (which is an assignment of boolean values to X_i, $i = 1, 2, \ldots, m$, and Y_j, $j = 1, 2, \ldots, n$) we count the number of 1's in every column j, that has $Y_j = 1$, less mp_j.

4.1 The Weighted Flip Cut (WFC) Problem

The WFC problem as formalized in [10], takes the following errors into account: (1) false positives, (2) false negatives (3) orientations and (4) sizing errors.

The WFC problem: Given m molecules with n sites each, p_j as the digestion rate, and, with δ_j as the "sizing error" for column j, obtain an alignment of the molecules such that the total number of 1's in the consensus cut columns and its neighborhood defined by δ_j, which is at least mp_j in each, is maximized [10].

The WFC problem is modeled by modifying the BFC model so that an observation of a cut in one location supports a cut in a nearby location. For this we modify the binary matrix $[M_{ij}]$ as follows:

$$\tilde{M}_{ij} = M_{ij} + \sum_{k=1}^{\delta} \alpha_k (M_{i(j+k)} + M_{i(j-k)}). \tag{2}$$

δ denotes the neighborhood in which the 1 is likely to be seen and α_k denotes the weight of the nearby cuts. The matrix, $[\tilde{M}_{ij}]$, is no longer binary and this new problem, which is the optimization problem of equation (1) on $\tilde{M}_{ij}$ instead of M_{ij}, is termed the *Weighted Flip Cut* (WFC) Problem [10].

4.2 The EBFC Problem

Formally, the exclusive BFC problem is as follows. Given m binary molecules of length n each, determine the flip for each molecule and an assignment of either j or $\bar{j}$ as a cut (but not both) for j, $1, \leq j \leq n/2$, such that the total number of 1's in the cut sites is maximized. Note that we can assume without loss of generality that n is even since otherwise, we can remove the middle site, that is, the site $(n+1)/2$, and the problem remains unchanged.

EBFC is another view of the BFC problem: notice that by flipping a molecule, a cut (or a no-cut) at position j can only move to its *conjugate* position given by $\bar{j} = n + 1 - j$. Thus we can view the problem as efficiently distributing the cuts along a column j and its conjugate $\bar{j}$.

We prove the following lemma about the EBFC problem.

Lemma 1 *EBFC is a special case of the BFC problem.*

Proof: Let

$$S_j = |\{i | M_{ij} = 1 \text{ AND } M_{i\bar{j}} = 1\}|,$$
$$\bar{S}_j = |\{i | M_{ij} = 1 \text{ XOR } M_{i\bar{j}} = 1\}|,$$

where $\overline{j} = n - j + 1$. Further, let

$$p_j = p_{\overline{j}} = \frac{\bar{S}_j + 2S_j}{2m}. \quad (3)$$

Note that S_j is the count of the number of symmetric cuts and $\bar{S}_j$ is the total number of non-symmetric cuts in columns j and $\overline{j}$. Irrespective of the assignment of orientations to the molecules/rows, j and $\overline{j}$ will always have at least S_j 1's. The 1's corresponding to $\bar{S}_j$, will get distributed between j and $\overline{j}$ depending on the alignment. We claim that under this definition of p_j for the BFC problem, it is the same as the EBFC problem. It can be verified that under these conditions that $Y_j + Y_{\overline{j}} = 1$ holds for all j, since the the definition of p_j ensures that only one of j or $\overline{j}$ is a consensus cut in the optimal alignment (and that is the one with the higher number of 1's). If the number of 1's is equal in both, we can arbitrarily pick only one without changing the cost. □

4.2.1 EBFC is MAX SNP-hard

For the sake of completeness we give the following definitions.

Max Cut (MC) problem: Given a graph, find a partition of the vertices into disjoint sets, S_1 and S_2, such that the number of edges with one vertex in S_1 and the other in S_2 is maximized.

Bipartite Max Cut (BMC) problem: Given a bi-partite weighted graph with edge weights $\in \{+1, -1\}$, find a partition of the vertices into disjoint sets, S_1 and S_2, such that the sum of the weights of edges with one vertex in S_1 and the other in S_2 is maximized.

Theorem 1 *EBFC is NP-hard. Further, there exists a constant $\epsilon > 0$ such that approximating EBFC within a factor of $1 - \epsilon$ is NP-hard.*

Proof. To prove the inapproximability of the EBFC problem, we use the recent technique of giving a *gap-preserving reduction* of a Max SNP-hard problem, the MC problem, to our problem [7], [8].

Outline of the Proof: The proof has four steps. Let C^*_X denote the cost of the optimal solution and C_X denote the cost of any solution of the problem X.

Step 1 . We show the reduction of an instance of the MC problem with e edges to an instance of the BMC problem with

(**1.1**) correspondence between the two solutions,
(**1.2**) $C^*_{MC} = C^*_{BMC}/2$, $C_{MC} \geq C_{BMC}/2$, and,
(**1.3**) the number of negative edges in the BMC is $6e$.

Step 2 . We show the reduction of an instance of the BMC problem to an instance of the EBFC problem with

(**2.1**) correspondence between the two solutions, and,
(**2.2**) $C_{EBFC} - e^- = C_{BMC}$,
where e^- is the number of edges with negative weights in BMC.

Step 3 . We relate the solution of the EBFC and a BMC that was constructed from an MC.

Step 4 . Finally, we show that the reduction is *gap-preserving*.

For some $\epsilon > 0$, let C^* denote the optimal solution and $\tilde{C}$ denote an approximate solution with $\tilde{C}_{EBFC} \geq (1-\epsilon)C^*_{EBFC}$.

$$\begin{array}{rcll}
\tilde{C}_{MC} & \geq & \frac{\tilde{C}_{BMC}}{2} & \text{(using Step 1.2)} \\
 & = & \frac{\tilde{C}_{EBFC}-6e}{2} & \text{(using steps 1.3 \& 2.2)} \\
 & \geq & \frac{(1-\epsilon)C^*_{EBFC}-6e}{2} & \text{(by defn of } \tilde{C}_{EBFC}) \\
 & = & \frac{(1-\epsilon)(C^*_{BMC}+6e)-6e}{2} & \text{(using Step 2.2)} \\
 & = & \frac{(1-\epsilon)C^*_{BMC}-6e\epsilon}{2} & \\
 & = & (1-\epsilon)\frac{C^*_{BMC}}{2} - 3e\epsilon & \\
 & \geq & (1-\epsilon)C^*_{MC} - (3\epsilon)2C^*_{MC} & \text{(since } C^*_{MC} \geq e/2) \\
 & = & (1-7\epsilon)C^*_{MC} &
\end{array} \tag{4}$$

This shows that given a PTAS for EBFC, we can construct a PTAS for MC, which is a contradiction, hence EBFC does not have a PTAS.

Now, we prove each of the steps from 1 to 3.

Step 1. MC to BMC reduction (see Figure 2).
Consider an MC problem with vertices and edges $(V, E), n = |V|, e = |E|$. Let a solution be of size K, and, the partition of the vertices induced by this solution be S_1 and S_2.

Reduction: Construct an instance of BMC with $(\tilde{V}, \tilde{E})$ as follows: For each $v_i \in V$, with degree d_i, construct $2(d_i + 1)$ vertices, $V_{gadget_i} = \{v'_{i0}, v'_{i1}, \ldots, v'_{id_i}, v''_{i0}, v''_{i1}, \ldots, v''_{id_i}\}$. Further, $wt(v'_{ij}, v''_{ij}) = wt(v'_{i0}, v''_{ij}) = wt(v'_{ij}, v''_{i0}) = -1, j = 1, 2, \ldots, d_i$. Thus, v_i gives rise to $3d_i$ edges with negative weight. Also if $v_1v_2 \in E$ then $wt(v'_{10}v''_{20}) = wt(v'_{20}v''_{10}) = +1$. It can be seen that this construction gives a bipartite graph with $\tilde{V} = V' \cup V''$ where $v'_x \in V', v''_x \in V''$.

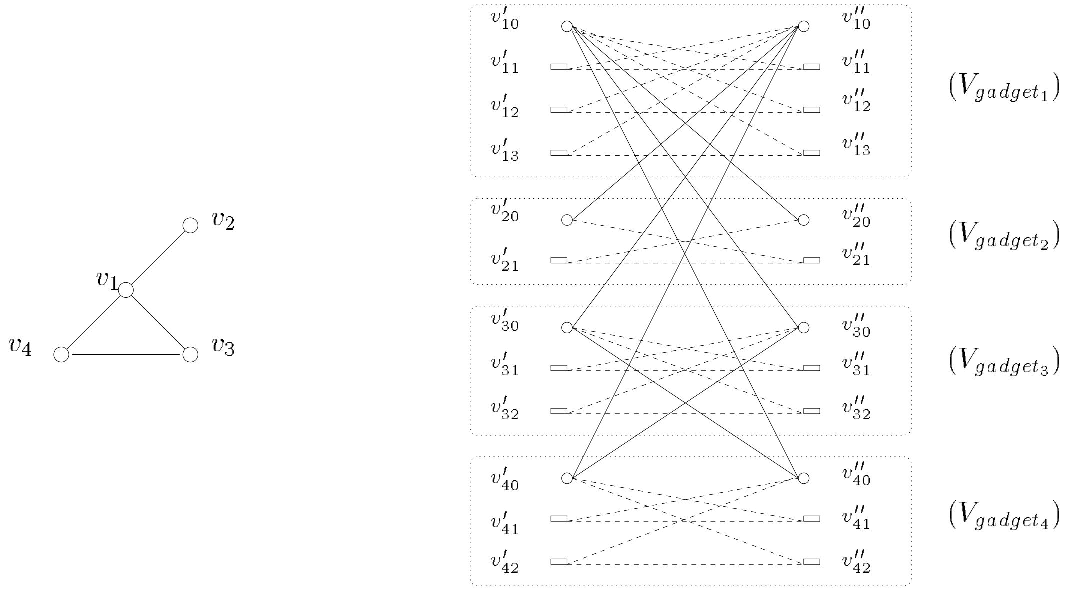


Figure 2: An example to show the reduction of an MC instance to a BMC instance. The bipartite graph is shown on the right: the solid edges have $+1$ weight and the dashed edges have weight -1. The "gadgets" corresponding to each vertex v_i, $i = 1 \ldots 4$ of the original graph, is shown enclosed in a dotted box.

Thus the BMC has $2n + 6e$ vertices, and, $2e$ edges with weights $+1$, and, $6e$ edges with negative weights. Recall for any graph $\sum_i d_i = 2e$.

<u>Observations</u>: We make the following observations.

1.1 In a solution of the BMC, the two sets $\tilde{S}_1, \tilde{S}_2$, are such that $V_{gadget_i} \subset \tilde{S}_1$ or $\tilde{S}_2$, $\forall i$.

    If this does not hold, that is $V_{gadget_i} \not\subset \tilde{S}_1$ then the solution can be modified as follows that only improves the solution. By the construction, $|V_{gadget_i}| = 2(d_i + 1)$. Thus $|V_{gadget_i} \cap \tilde{S}_1|$ or $|V_{gadget_i} \cap \tilde{S}_2| \geq d_i + 1$. Without loss of generality, let, $|V_{gadget_i} \cap \tilde{S}_2| \geq d_i + 1$. If $v_{ki} \in S_1, k \neq 0$, then by including v_{ki} in S_2 the cost increases by 2. If $v_{i0} \in S_1$, then it has *exactly* d_i negative edges incident on it with the other ends being in S_2 while *at most* d_i positive edges incident on it with the other ends in S_2. Hence by including v_{i0} in S_2 the "cut" will not suffer a loss.

1.2 All the edges that contribute to a solution to the BMC have positive weights, since, by observation 1.1, all the negative weight edges must be either in S_1 or in S_2.

1.3 In a solution to the BMC, if $v_1'v_2''$ is in the cut, so must $v_1''v_2'$ (called the *image* of $v_1'v_2''$). This follows from observation 1.1, as V_{gadget_1} and V_{gadget_2} are in the sets $\tilde{S}_1$ and $\tilde{S}_2$ respectively (without loss of generality).

1.4 Given a solution to the BMC, the solution to the corresponding MC is constructed as follows: if $v_1'v_2''$ (and its image) is in the solution to the BMC, then v_1v_2 is in the solution to the MC.

<u>Claim (C1.1)</u>: MC has solution of size K iff BMC has solution of size $2K$.
Proof: Let the MC have a solution of size K. Assume the BMC has a solution of $2(K + x)$, for some $x > 0$. Let the x edges be $v'_{i_10}v''_{j_10}, v'_{i_20}v''_{j_20}, \ldots, v'_{i_x0}v''_{j_x0}$ and their images (see observations 1.2 and 1.3). Then the solution to MC can be the edges corresponding to K, augmented by $v_{i_1}v_{j_1}, _{i_2}v_{j_2}, \ldots, v_{i_x}v_{j_x}$, thus giving a solution of size $K + x$ to the MC problem, which is a contradiction.

Let the BMC have a solution of size $2K$. Assume the MC has a solution of size $K + x$, for some $x > 0$. Let the x edges be $v_{i_1}v_{j_1}, v_{i_2}v_{j_2}, \ldots, v_{i_x}v_{j_x}$ in the solution of the MC. Now, the solution of the BMC, can be augmented by $v'_{i_10}v''_{j_10}, v'_{i_20}v''_{j_20}, \ldots, v'_{i_x0}v''_{j_x0}$ and their images (see observation 1.3), giving a solution of size $2(K + x)$ for the BMC, which is a contradiction. $\square$

<u>**Step 2.**</u> BMC to EBFC reduction (see Figure 3).
Consider a BMC $((V_1, V_2), E)$, $V_1 = \{v_1^1, v_2^1, \ldots, v_m^1\}$, $V_2 = \{v_1^2, v_2^2, \ldots, v_n^2\}$, and, number of edges with negative weights be e^-. Let a solution be of size K and partition of vertices, $V_1 \cup V_2$, induced by this solution be S_1 and S_2.

	v''_{10}	v''_{11}	v''_{12}	v''_{13}	v''_{20}	v''_{21}	v''_{30}	v''_{31}	v''_{32}	v''_{40}	v''_{41}	v''_{42}	v''_{42}	v''_{41}	v''_{40}	v''_{32}	v''_{31}	v''_{30}	v''_{21}	v''_{20}	v''_{13}	v''_{12}	v''_{11}	v''_{10}	
v'_{10}	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	$\surd$
v'_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	$\surd$
v'_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	$\surd$
v'_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	$\surd$
v'_{20}	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	$\times$
v'_{21}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	$\times$
v'_{30}	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	$\times$
v'_{31}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	$\times$
v'_{32}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	$\times$
v'_{40}	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	$\times$
v'_{41}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	$\times$
v'_{42}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	$\times$
	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$	

Figure 3: The EBFC matrix corresponding to the BMC problem of Figure 2. Note that the 1's in the left half of the matrix correspond to the positive edge weights of the BMC and the 1's on the right correspond to the negative edge weights. [In the solution, the rows/molecules marked with $\surd$ are flipped, and, the columns marked with $\surd$ are the consensus cut columns.]

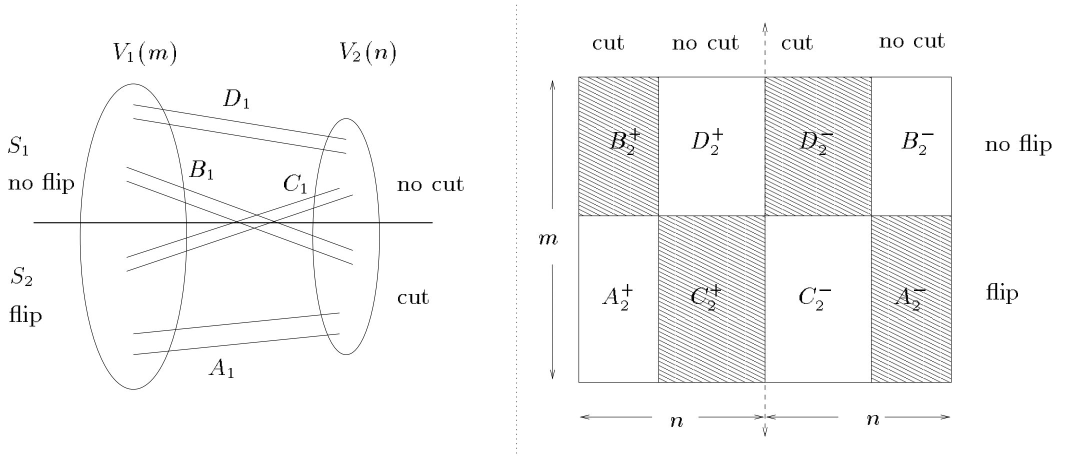
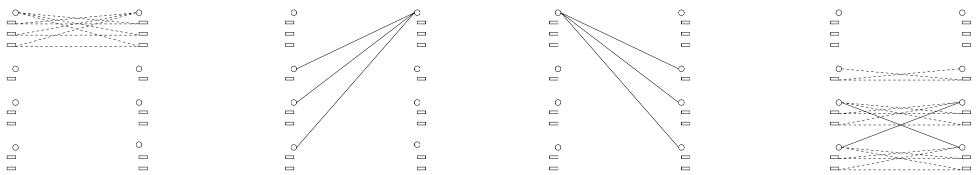


Figure 4: A schematic representation of the BMC to EBFC reduction: The left shows the BMC problem and the right shows the EBFC problem. See the text on the reduction for other details.

Reduction: Construct an instance of EBFC $[M_{ij}]$ with m rows and $2n$ columns as follows. If $wt(v_i^1 v_j^2) = 1$, then $M_{ij} = 1, M_{i\bar{j}} = 0$. If $wt(v_i^1 v_j^2) = -1$, then $M_{ij} = 0, M_{i\bar{j}} = 1$. If $v_i^1 v_j^2$ is not an edge in the BMC, then $M_{ij} = M_{i\bar{j}} = 0$.

	v''_{1*}	v''_{2*}	v''_{3*}	v''_{4*}	$\overline{v''_{4*}}$	$\overline{v''_{3*}}$	$\overline{v''_{2*}}$	$\overline{v''_{1*}}$
v'_{10}	$A^+A^+A^+A^+$	$\mathbf{C}^+C^+$	$\mathbf{C}^+C^+C^+$	$\mathbf{C}^+C^+C^+$	$C^-C^-C^-$	$C^-C^-C^-$	C^-C^-	$\mathbf{A}^-\mathbf{A}^-\mathbf{A}^-A^-$
v'_{11}	$A^+A^+A^+A^+$	C^+C^+	$C^+C^+C^+$	$C^+C^+C^+$	$C^-C^-C^-$	$C^-C^-C^-$	C^-C^-	$\mathbf{A}^-A^-\mathbf{A}^-A^-$
v'_{12}	$A^+A^+A^+A^+$	C^+C^+	$C^+C^+C^+$	$C^+C^+C^+$	$C^-C^-C^-$	$C^-C^-C^-$	C^-C^-	$\mathbf{A}^-\mathbf{A}^-A^-A^-$
v'_{13}	$A^+A^+A^+A^+$	C^+C^+	$C^+C^+C^+$	$C^+C^+C^+$	$C^-C^-C^-$	$C^-C^-C^-$	C^-C^-	$\mathbf{A}^-A^-A^-A^-$
v'_{20}	$\mathbf{B}^+B^+B^+B^+$	D^+D^+	$D^+D^+D^+$	$D^+D^+D^+$	$D^-D^-D^-$	$D^-D^-D^-$	$\mathbf{D}^-D^-$	$B^-B^-B^-B^-$
v'_{21}	$B^+B^+B^+B^+$	D^+D^+	$D^+D^+D^+$	$D^+D^+D^+$	$D^-D^-D^-$	$D^-D^-D^-$	$\mathbf{D}^-D^-$	$B^-B^-B^-B^-$
v'_{30}	$\mathbf{B}^+B^+B^+B^+$	D^+D^+	$D^+D^+D^+$	$\mathbf{D}^+D^+D^+$	$D^-D^-D^-$	$\mathbf{D}^-\mathbf{D}^-D^-$	D^-D^-	$B^-B^-B^-B^-$
v'_{31}	$B^+B^+B^+B^+$	D^+D^+	$D^+D^+D^+$	$D^+D^+D^+$	$D^-D^-D^-$	$\mathbf{D}^-\mathbf{D}^-D^-$	D^-D^-	$B^-B^-B^-B^-$
v'_{32}	$B^+B^+B^+B^+$	D^+D^+	$D^+D^+D^+$	$D^+D^+D^+$	$D^-D^-D^-$	$\mathbf{D}^-D^-D^-$	D^-D^-	$B^-B^-B^-B^-$
v'_{40}	$\mathbf{B}^+B^+B^+B^+$	D^+D^+	$\mathbf{D}^+D^+D^+$	$D^+D^+D^+$	$\mathbf{D}^-\mathbf{D}^-D^-$	$D^-D^-D^-$	D^-D^-	$B^-B^-B^-B^-$
v'_{41}	$B^+B^+B^+B^+$	D^+D^+	$D^+D^+D^+$	$D^+D^+D^+$	$\mathbf{D}^-\mathbf{D}^-D^-$	$D^-D^-D^-$	D^-D^-	$B^-B^-B^-B^-$
v'_{42}	$B^+B^+B^+B^+$	D^+D^+	$D^+D^+D^+$	$D^+D^+D^+$	$\mathbf{D}^-D^-D^-$	$D^-D^-D^-$	D^-D^-	$B^-B^-B^-B^-$

Grouping of the matrix elements.



A edges (empty A^+ & dashed edges in A^-) B edges (empty B^- & solid edges in B^+) C edges (empty C^- & solid edges in C^+) D edges (solid edges in D^+ & dashed edges in D^-)

Corresponding grouping of the (BMC) graph edges.

Figure 5: The grouping of the elements of the EBFC matrix, and its corresponding edges, due to a solution (not necessarily optimal) shown in Figure 3.

Observations: We make the following observations.

2.1 In a given alignment of the EBFC, the elements of M_{ij} can be grouped into the following sets, as shown in Figure 4:

$$A_2^+ = \{(i,j) | i \text{ has been flipped, } j \text{ is a cut, } j \leq n\}, A_2^- = \{(i,\bar{j}) | (i,j) \in A_2^+\},$$
$$B_2^+ = \{(i,j) | i \text{ has not been flipped, } j \text{ is a cut, } j \leq n\}, B_2^- = \{(i,\bar{j}) | (i,j) \in B_2^+\},$$

$$C_2^+ = \{(i,j)|i \text{ has been flipped, } j \text{ is not a cut, } j \leq n\}, C_2^- = \{(i,\overline{j})|(i,j) \in C_2^+\},$$
$$D_2^+ = \{(i,j)|i \text{ has not been flipped, } j \text{ is not a cut, } j \leq n\}, D_2^- = \{(i,\overline{j})|(i,j) \in D_2^+\}.$$

Let $\sum A_2^+ = \sum_{(i,j)\in A_2^+} M_{ij}$. Similarly define $\sum A_2^-$, $\sum B_2^+$, $\sum B_2^-$, $\sum C_2^+$, $\sum C_2^-$, $\sum D_2^+$, $\sum D_2^-$. Figure 5 shows an illustrative example.

Recall that the cost in the EBFC is the number of 1's in the cut columns with the rows flipped appropriately. Thus the cost is $\sum A_2^- + \sum B_2^+ + \sum C_2^+ + \sum D_2^-$ (corresponding to the shaded rectangular region shown in Figure 4).

2.2

$$\sum A_2^- + \sum B_2^- + \sum C_2^- + \sum D_2^- = \sum_{i=1}^{m} \sum_{j=n+1}^{2n} M_{ij} = e^-. \tag{5}$$

2.3 Given an alignment for the EBFC with cost C_{EBFC} as

$$C_{EBFC} = \sum A_2^- + \sum B_2^+ + \sum C_2^+ + \sum D_2^-, \tag{6}$$

(see observation 2.1), a solution for the BMC is constructed as follows. Define the sets as below:

$$A_1^+ = \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in A_2^+\}, A_1^- = \{v_i^1 v_j^2 | M_{i\overline{j}} \neq 0, (i,j) \in A_2^-\},$$
$$B_1^+ = \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in B_2^+\}, B_1^- = \{v_i^1 v_j^2 | M_{i\overline{j}} \neq 0, (i,j) \in B_2^-\},$$
$$C_1^+ = \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in C_2^+\}, C_1^- = \{v_i^1 v_j^2 | M_{i\overline{j}} \neq 0, (i,j) \in C_2^-\},$$
$$D_1^+ = \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in D_2^+\}, D_1^- = \{v_i^1 v_j^2 | M_{i\overline{j}} \neq 0, (i,j) \in D_2^-\}.$$

Let $A_1 = A_1^+ \cup A_1^-$, $B_1 = B_1^+ \cup B_1^-$, $C_1 = C_1^+ \cup C_1^-$, $D_1 = D_1^+ \cup D_1^-$. Then S_1 and S_2, the partition of the vertices, are defined as follows:

$$S_1 = \{v_i^1 | v_i^1 v_j^2 \in B_1, \text{ for some } j\} \cup \{v_j^2 | v_i^1 v_j^2 \in C_1, \text{ for some } i\} \cup \{v_i^1, v_j^2 | v_i^1 v_j^2 \in D_1\}, \tag{7}$$
$$S_2 = \{v_j^2 | v_i^1 v_j^2 \in B_1, \text{ for some } i\} \cup \{v_i^1 | v_i^1 v_j^2 \in C_1, \text{ for some } j\} \cup \{v_i^1, v_j^2 | v_i^1 v_j^2 \in A_1\}. \tag{8}$$

Notice that $|A_1^+| = \sum A_2^+$, $|A_1^-| = \sum A_2^-$ and so on. Also notice that B_1^+ is the set of edges with positive weights and B_1^- is the set of edges with negative weights. Similarly for the other sets. Thus the corresponding cost, C_{BMC} for the BMC is,

$$C_{BMC} = |B_1^+| - |B_1^-| + |C_1^+| - |C_1^-|. \tag{9}$$

2.4 It can be seen from the above that given a partition of the vertices in the BMC, an alignment (assignments of flips/no-flips to rows and cuts/no-cuts to columns) can be obtained for the EBFC, and, vice-versa.

2.5 If C_{EBFC} denotes the cost for an alignment in the EBFC problem, and, if C_{BMC} denotes the cost for the corresponding alignment in the BMC problem, the following holds (using equations (5), (6), (9)):

$$C_{EBFC} - e^{-} = C_{BMC}. \tag{10}$$

Recall that e^{-} is the number of edges with negative weights in the BMC problem.

Claim (C2.1): BMC has an optimal solution of size K iff EBFC has an optimal solution of size $K + e^{-}$.

Proof: It can be verified from the above construction that, improving the solution for the EBFC by $x > 0$, results in improving the BMC by x and vice-versa. Hence using equation (10) we have the required result. □

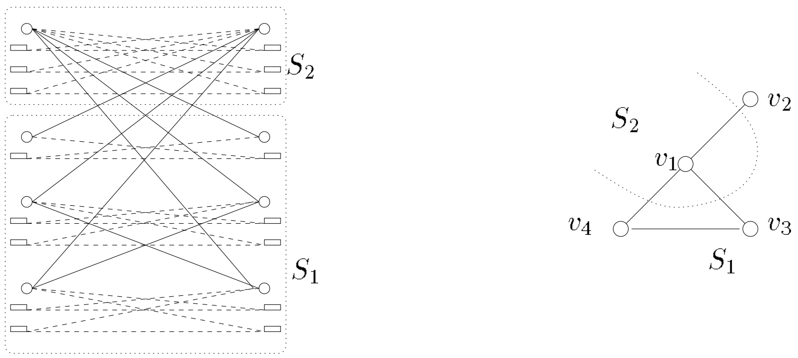


Figure 6: The solution to the BMC problem obtained from Figure 5 using equations (7), (8) and the corresponding solution in the MC problem, introduced in Figure 2. S_1 and S_2 are the partition of the vertices in the graphs.

Step 3. (Arguments about MC to EBFC):
We make the following observations about the solution to an EBFC, that has been constructed from a BMC that was in turn constructed from an MC problem.

3.1 The EBFC matrix is of size $L \times 2L$ where $L = v + 2e$. Recall that v is the number of vertices and e the number of edges for the MC problem.

3.2 See Figure (7) which shows the rows and columns associated with a vertex v_i of the MC problem in the EBFC matrix. Let $X_{v_{i0}}$ denote the variable associated with the row that corresponds to the vertex v'_{i0}, and, Y_{v_i0} denote the variable associated with the column that corresponds to the vertex v''_{i0} of the BMC. Similarly $X_{v_{i0}}, X_{v_{i1}}, X_{v_{i2}}, \ldots, X_{v_{id_i}}, Y_{v_{i1}}, Y_{v_{i2}}, \ldots, Y_{v_{id_i}}$.

		v'_{i0}	v'_{i1}	v'_{i2}	v'_{i3}			$\overline{v''_{i3}}$	$\overline{v''_{i2}}$	$\overline{v''_{i1}}$	$\overline{v''_{i0}}$	
		$\vdots$	$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$	$\vdots$	$\vdots$	
		0	0	0	0			0	0	0	0	
		1	0	0	0			0	0	0	0	
		0	0	0	0			0	0	0	0	
		$\vdots$	$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$	$\vdots$	$\vdots$	
		0	0	0	0			0	0	0	0	
		1	0	0	0			0	0	0	0	
		0	0	0	0			0	0	0	0	
		$\vdots$	$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$	$\vdots$	$\vdots$	
v'_{i0}	$\ldots 010 \ldots 010 \ldots$	0	0	0	0	$\ldots 010 \ldots$	$\ldots 0 \ldots$	**1**	**1**	**1**	0	$\ldots 0 \ldots 0 \ldots$
v'_{i1}	$\ldots 000 \ldots 000 \ldots$	0	0	0	0	$\ldots 000 \ldots$	$\ldots 0 \ldots$	**1**	0	**1**	0	$\ldots 0 \ldots 0 \ldots$
v'_{i2}	$\ldots 000 \ldots 000 \ldots$	0	0	0	0	$\ldots 000 \ldots$	$\ldots 0 \ldots$	**1**	**1**	0	0	$\ldots 0 \ldots 0 \ldots$
v'_{i3}	$\ldots 000 \ldots 000 \ldots$	0	0	0	0	$\ldots 000 \ldots$	$\ldots 0 \ldots$	**1**	0	0	0	$\ldots 0 \ldots 0 \ldots$
		$\vdots$	$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$	$\vdots$	$\vdots$	
		0	0	0	0			0	0	0	0	
		1	0	0	0			0	0	0	0	
		0	0	0	0			0	0	0	0	
		$\vdots$	$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Figure 7: For clarity of exposition, let $d_i = 3$ in the MC problem. Abusing notation, let the conjugates of the columns corresponding to $v''_{i0}, v''_{i1}, v''_{i2}, v''_{i3}$ be $\overline{v_{i0}}'', \overline{v_{i1}}'', \overline{v_{i2}}'', \overline{v_{i3}}''$ respectively. The above table shows the rows and the columns corresponding to v_i of the MC. Note that the number of 1's in the column v'_{i0} is 3 and the number of 1's in the row v'_{i0} is $3+3$, where, half of the 1's are due to positive weight edges (left half of matrix) and the other half due to the negative weight edges (right half of matrix). The 1's shown in bold correspond to the negative weight edges on the vertices of the gadget corresponding to v_i of the MC problem. Notice that $v'_{i1}, v'_{i2}, v'_{i3}$ will have the same flip as v'_{i0} without any conflict with other vertices. Also, since the column due to v''_{i0} has no 1's at all, v''_{i0} is a cut if v'_{i0} is flipped and is not a cut if v'_{i0} is not flipped.

The reader can verify (see Figure (7)) that one can always obtain a solution for which the following holds: $X_{v_{i0}} = X_{v_{i1}} = X_{v_{i2}} = \ldots = X_{v_{id_i}} = Y_{v_{i1}} = Y_{v_{i2}} = \ldots = Y_{v_{id_i}}$. This is important since it implies that in the BMC, $\tilde{V}_{gadget_i} \subset \tilde{S}_1$ or $\tilde{S}_2$. Thus the corresponding solution of size $2K$ in the BMC corresponds directly to a solution of size K of the MC. Thus for any solution of the EBFC, all the 1's corresponding to the negative edges are counted in the solution. If not, the solution can be altered, in linear time, without reducing the cost so that this condition holds. Thus

$$C_{EBFC} \geq 6e, \tag{11}$$

where e is the number of edges of the MC.

This concludes the proof of the inapproximability of the EBFC problem. □

Corollary 1 *Achieving an approximation ratio* $1 - \Upsilon/7$ *for EBFC is NP-hard.*

Proof: Since it is known that achieving an approximation ration of Υ for the MC problem is NP-hard, our results follow from our reduction (see Step 4 of the proof) presented in the theorem. □

We prove the following theorem about the hardness of the BFC problem.

Theorem 2 *BFC is NP-hard. Further, there exists a constant* $\epsilon > 0$ *such that approximating BFC within a factor of* $1 - \epsilon$ *is NP-hard.*

Proof. Using Lemma 1 and Theorem 1 we conclude that BFC is MAX SNP-hard. □

As a direct consequence of Theorem 2 we have the following hardness result for the WFC problem.

Theorem 3 *WFC is NP-hard. Further, there exists a constant* $\epsilon > 0$ *such that approximating WFC within a factor of* $1 - \epsilon$ *is NP-hard.*

Corollary 2 *Achieving an approximation ratio* $1 - \Upsilon/7$ *for WFC is NP-hard.*

Theorem 4 *$BFC_{\max K}$ [12] is NP-hard. Further, there exists a constant* $\epsilon > 0$ *such that approximating $BFC_{\max K}$ within a factor of* $1 - \epsilon$ *is NP-hard.*

Proof. Under the definition of p_j's as in equation (3) the $\text{BFC}_{\max K}$ is the same as the EBFC problem (the number of consensus cuts is always $n/2$, when the molecules have n sites), hence $\text{BFC}_{\max K}$ is NP-hard.

Next, we will show that if we have a PTAS for $\text{BFC}_{\max K}$, we will have a PTAS for BFC, which would be a contradiction. Given a BFC let $p_{min} = \min_j p_j$, and $p_{max} = \max_j p_j$. Let $\tilde{X}$ denote an approximate solution and X^* denote the optimal solution. Then if $\text{BFC}_{\max K}$ has a PTAS let $\frac{\tilde{N}}{N^*} \geq \epsilon$ for some $0 < \epsilon \leq 1$. Note that N^* is the number of consensus cuts. Let $\tilde{C} \geq \tilde{N} p_{min}$, then $C^* \leq N^* p_{max}$. Hence we have

$$\frac{\tilde{C}}{C^*} \geq \frac{\tilde{N} p_{min}}{N^* p_{max}} \geq \epsilon \frac{p_{min}}{p_{max}}. \tag{12}$$

□

Corollary 3 *Achieving an approximation ratio $(1 - \Upsilon/7)\frac{p_{max}}{p_{min}}$ for $BFC_{\max K}$ is NP-hard.*

4.3 d-wise Match (dM) Problem

The d-wise Match problem has been identified in [13] which takes the following errors into account: (1) false positives, (2) false negatives and (3) orientations.

d-wise Match (dM) problem: Given m molecules with n sites each, with false positive and negative errors and orientation uncertainties, and a fixed $1 < d \leq m$, $\delta > 0$, find an alignment to the molecules so that it has the maximum *d-wise match*. Given a set of d molecules, $i_1, i_2, \ldots, i_d$, d-wise match, $A^X(i_1, i_2, \ldots, i_d)$, is defined as

$$A^X(i_1, i_2, \ldots, i_d) = \text{ \# of cut sites that are within } \delta \text{ of each other in \textit{all} the mols, given } X,$$

where X denotes and alignment [3] and the match is made respecting this alignment. In principle, this alignment could model other errors such as spurious molecules and others. Thus it is the following optimization problem:

$$\max_{\text{(over all alignments)}} \left\{ \sum_{i_1=1}^{m} \sum_{i_2=i_1}^{m} \sum_{i_3=i_2}^{m} \cdots \sum_{i_d=i_{d-1}}^{m} A^X(i_1, i_2, \ldots, i_s) \right\} \tag{13}$$

where alignment refers to an assignment of orientation to each molecule. Informally, we are looking for d simultaneous agreements between molecules. In particular if $d = 2$, then the task is to maximize the pairwise match.

Theorem 5 ([13]) *The matching with d-wise agreement is equivalent to the following optimization problem.*

$$\max_{\textit{(over all alignments)}} \sum_{l} \binom{T_l}{d},$$

where T_l represents the number of cuts in the position l in that alignment.

[3] In the case where the alignment involves only the orientation uncertainty, X is one of the 2^{d-1} possibilities.

Weighted Consistency Graph (WCG) problem. This is the d-wise match problem when $d = 2$. We will first define some related concepts. Given a 2-wise match problem, a corresponding graph $\mathcal{G}$ is constructed with every vertex v_i corresponding to a molecule i. Let $X = S$ denote an alignment where both the molecules i and j are taken as-is, and $X = O$ denote the alignment where one of them is flipped. Every edge $e_{ij} = v_i v_j$ is labeled and weighted as follows. Label $L(v_i v_j)$ is defined as:

$$L(e_{ij}) = \begin{cases} Same & A^S(i,j) \geq A^O(i,j), \\ Opposite & \text{otherwise.} \end{cases}$$

Vertices v_i, v_j, v_k are **consistent** if the following holds: either all the three or exactly one edge is labeled *Same*. Note that this implies that all the three molecules corresponding to the three edges can be uniquely aligned with the maximum match cost. A labeled graph $\mathcal{G}$ is said to be **consistent** if every three vertices v_i, v_j, v_k is consistent.

Weight $Wt(e_{ij})$ is defined as:

$$\begin{aligned} Wt(e_{ij}) &= \min(A^S(i,j), A^O(i,j)) - \max(A^S(i,j), A^O(i,j)) \\ &= -|A^S(i,j) - A^O(i,j)| \end{aligned} \tag{14}$$

The weight corresponds to the "loss" suffered if the relative alignment is forced to change.

Weighted Consistency Graph problem (WCG): Given the match problem with the corresponding graph $\mathcal{G}$, the problem is to obtain a set of edges S with new labels $\tilde{L}(e_{ij})$ such that

1. $\mathcal{G}$ is consistent under the new labels $\tilde{L}()$, and,
2. the sum of the weights on the edges with new labels is minimum.

Consistency Graph problem (CG): The Consistency Graph problem is defined as follows: given a labeled graph $\mathcal{G}$, find the minimum number of changes to the labels (*Same* to *Opposite* or vice-versa) required to get a consistent graph. This is the WCG problem under the assumption that all the edges are of equal weight.

Theorem 6 *The CG problem is NP-hard. Further, there exists a constant $\epsilon > 0$ such that approximating CG within a factor of $1 + \epsilon$ is NP-hard.*

Proof: We give the proof in two steps. In step 1 we give the MC to CG reduction and show that $C_{MC} = e - C_{CG}$ where C_X is a solution to the problem X and e is the number of edges in the MC problem. In step 2 we show that the reduction is *gap-preserving*.

Step 1

Construction: Given an MC with n vertices and e edges, we construct an instance of CG by simply labeling every edge as *Opposite*.

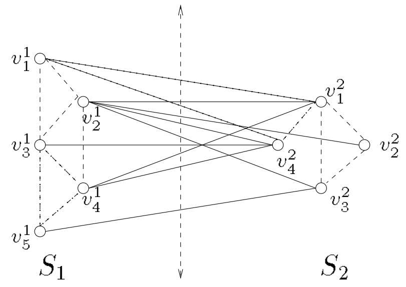


Figure 8: A consistent graph: the solid edges are labeled *Opposite* and the dashed edges are labeled *Same*. The vertices are partitioned into sets S_1 and S_2 as shown (all edges on vertices only in S_1 or S_2 are labeled *Same*; all edges with one end in S_1 and the other in S_2 are labeled *Opposite*). The only two kinds of triangles are: (1) $v_3^1 v_4^1 v_5^1$, where all edges are labeled *Same*, and, (2) $v_1^1 v_1^2 v_4^2$, where exactly one edge is labeled *Same*.

We make the following observations:

1. A consistent graph is such that the vertices can be partitioned into two sets S_1 and S_2 such that $\forall v_i, v_j \in S_1$ (or S_2) , $L(e_{ij}) = Same$, and, $\forall v_i \in S_1, v_j \in S_2, L(e_{ij}) = Opposite$.

   There are only two kinds of consistent triangles: (1) all labels are *Same* or (2) exactly one label is *Same*. It can be verified that only these two kinds of triangles (and no other) exist for the consistent graph whose vertices are given by $S_1 \cup S_2$. See Figure 8 for an example.

2. Given a solution of size e' to the CG, which is the number of changes from label *Opposite* to *Same* (since there was no edge with label *Same*), we can show that the solution to MC is of size $e - e'$. It can also be verified that decreasing the solution to CG by $x > 0$, increases the solution to the MC by x.

Step 2

Let $\tilde{C}_{CG}$ denote an approximate solution and C^*_{CG} denote the optimal solution. Then $\tilde{C}_{CG} \leq (1+\epsilon)C^*_{CG}$.

$$\begin{aligned}
\tilde{C}_{MC} &= e - \tilde{C}_{CG} && \text{(using observation 2)} \\
&\geq e - (1+\epsilon)C^*_{CG} && \text{(by defn of } \tilde{C}_{MC}) \\
&\geq e - (1+\epsilon)(e - C^*_{MC}) && \text{(by observation 2)} \\
&= (1+\epsilon)C^*_{MC} - e\epsilon \\
&\geq (1-\epsilon)C^*_{MC} && \text{(since } C^*_{MC} \geq e/2).
\end{aligned} \tag{15}$$

This concludes the proof. □

Corollary 4 *Achieving an approximation ratio* $1 - \Upsilon$ *for CG is NP-hard.*

Theorem 7 *WCG problem is NP-hard. Further, there exists a constant* $\epsilon > 0$ *such that approximating WCG within a factor of* $1 + \epsilon$ *is NP-hard.*

Corollary 5 *Achieving an approximation ratio* $1 - \Upsilon$ *for WCG is NP-hard.*

Theorem 8 *dM problem is NP-hard. Further, there exists a constant* $\epsilon > 0$ *such that approximating dM within a factor of* $1 + \epsilon$ *is NP-hard.*

Corollary 6 *Achieving an approximation ratio* $1 - \Upsilon$ *for dM is NP-hard.*

5 The Binary Partition Cut Problem

The main error the Binary Partition Cut (BPC) problem takes into account is the presence of spurious or bad molecules (along with false positive and negative errors).

We are given m molecules with n sites each, and, p_j as the digestion rate for column j, obtain an alignment of the molecules such that the following holds.

1. BPC problem: In the optimal solution, the number of 1's in the consensus cut columns, J, which is at least mp_J in each, is maximized.

   We get a rather surprising result that this indeed has a polynomial time algorithm.

2. $\text{BPC}_{\max K}$: In the optimal solution the number of consesnsus cut columns, K, is maximized. Anantharaman et al [12] shows a variation of this problem, where the information that the number of bad molecules is known *exactly*, is NP-complete.

Let the input binary $m \times n$ matrix be $[M_{ij}]$. Let us associate indicator variables X_i, $i = 1, 2, \ldots, m$, with every row which takes a value 1 if the molecule is good and 0 if it is spurious. Let Y_j, $j = 1, 2, \ldots, n$, be an indicator variable associated with every column that takes on a value of 1 if it is a consensus cut and 0 otherwise. Let the digestion rate be p_j for column j, $j = 1, 2, \ldots, n$. BPC problem can then be modeled as the following optimization problem:

$$\max \left\{ \sum_{j=1}^{n} Y_j \left(\sum_{i=1}^{m} X_i M_{ij} - p_j m \right) \right\}. \tag{16}$$

We prove the following theorem for the above problem.

Theorem 9 *The BPC problem has a polynomial time solution.*

Proof: Consider the corresponding minimization problem (equation 16)

$$\min \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} -M_{ij} X_i Y_j + \sum_{j=1}^{n} m p_j Y_j \right\} \tag{17}$$

which is a submodular function [15], hence has a polynomial time solution. □

Corollary 7 *The BPC$_{\max K}$ problem has a polynomial time solution.*

Proof: It can be verified that the following never occurs: if n_o is the number of consensus cuts in the optimal solution then there is a sub-optimal solution with $n > n_o$. This is because the new optimal alignment can be obtained from this sub-optimal giving a larger n_o. Thus the solution to the BPC problem gives a solution to this problem. □

6 The Binary Shift Cut (BSC) Problem

In this problem it is assumed that apart from the false positive and negative errors, the only error present is that some fragments of the molecules are missing. It is assumed that all the orientations are known correctly.

Given m molecules with at most n sites each, and, p_j as the digestion rate for column j, obtain an alignment (which identifies the missing fragment of each molecule with respect to the map with a total on n sites) of the molecules such that the following holds.

1. Binary Shift Cut problem (BSC): The total number of 1's in the consensus cut sites is maximized, where each cut site has at least mp_j number of 1's. In Dancik et al [11], this problem has $p_j = p = m/2$, $\forall j$.
2. BSC$_{\max K}$ [12]: This is the missing fragments problem in [12]. The total number of consensus cut sites is maximized, where each cut site has at least mp_j number of 1's. Anantharaman et al [12] have shown BSC$_{\max K}$ to be NP-Complete, we will show that this problem is MAX SNP-hard, and, give an upper bound on the polynomial time approximation factor of the problem.

We show a simple example, in Figure 9, that shows BSC and BSC$_{\max K}$ problems give rise to different optimal alignments and maps.

We show that BSC is MAX SNP-hard in the following theorem.

Input Problem

0	1	0	1	0
1	0	1	0	
0	1	0	1	0
1	0	1	0	

BSC

0	1	0	1	0	
	1	0	1	0	
0	1	0	1	0	
	1	0	1	0	
0	1	0	1	0	S'

$\mathrm{BSC}_{\max K}$

0	1	0	1	0	
1	0	1	0		
0	1	0	1	0	
1	0	1	0		
1	1	1	1	0	S'

Figure 9: An example to show different optimal configurations for the two different cost functions, BSC and $\mathrm{BSC}_{\max K}$. It is assumed that $p_j = 1/2$ for all j. The optimal cost for the BSC problem is 8 (number of 1's in the consensus cut columns) with 2 consensus cut columns. The optimal cost for $\mathrm{BSC}_{\max K}$ is 4 (the number of consensus cut columns). Note that the maps corresponding to the optimal configurations are different.

Theorem 10 *BSC problem is NP-hard. Further, there exists a constant $\epsilon > 0$ such that approximating it within a factor of $1 - \epsilon$ is NP-hard.*

Proof We will prove the result for a special case of the BSC problem where every molecule is such that either the left or the right fragment (not both) is missing; the missing fragment is exactly one unit in all the molecules. In the aligned configuration, a column j is in a cut only if the number of cuts is at least $p_j m$, which is defined in the proof of step 2.

Outline of the Proof: The proof has four steps. Let C^*_X denote the cost of the optimal solution and C_X denote the cost of any solution of the problem X.

Step 1 . We show a reduction of an instance of the MC problem with n vertices and e edges to an instance of the BMC problem with

(**1.1**) correspondence between the two solutions,
(**1.2**) $4C^*_{MC} = C^*_{BMC} - 4e - 3n$, $4C_{MC} \geq C_{BMC} - 4e - 3n$, and,
(**1.3**) the number of edges with positive weights in the BMC is $8e + 2n$.

Step 2 . We show the reduction of an instance of the BMC problem to an instance of the BSC problem with

(**2.1**) correspondence between the two solutions, and,
(**2.2**) $2C_{BSC} - c = C_{BMC}$,
where c is the number of 1's in the BSC matrix.

Step 3 . We relate the solution of the BSC and a BMC that was constructed from an MC.

Step 4 . Finally, we show that the reduction is *gap-preserving*.

For some $\epsilon > 0$, let C^* denote the optimal solution and $\tilde{C}$ denote an approximate solution with $\tilde{C}_{BSC} \geq (1-\epsilon)C^*_{BSC}$.

$$\begin{array}{rcll} \tilde{C}_{MC} & \geq & \frac{\tilde{C}_{BMC}-4e-3n}{4} & \text{(using Step 1.2)} \\ & = & \frac{2\tilde{C}_{BSC}-12e-5n}{4} & \text{(using Steps 1.3 \& 2.2)} \\ & \geq & \frac{(1-\epsilon)2C^*_{BSC}-12e-5n}{4} & \text{(by defn of } \tilde{C}_{BSC}\text{)} \\ & = & \frac{(1-\epsilon)(C^*_{BMC}+8e+2n)-12e-5n}{4} & \text{(using Step 2.2)} \\ & = & \frac{(1-\epsilon)C^*_{BMC}-4e-3n}{4} - \frac{(8e+2n)\epsilon}{4} & \\ & \geq & (1-\epsilon)C^*_{MC} - 2.5e\epsilon & \text{(using Step 1.2)} \\ & \geq & (1-\epsilon)C^*_{MC} - (2.5\epsilon)2C^*_{MC} & \text{(since } C^*_{MC} \geq e/2\text{)} \\ & = & (1-6\epsilon)C^*_{MC}. & \end{array} \tag{18}$$

This shows that given a PTAS for BSC, we can construct a PTAS for MC, which is a contradiction, hence BSC does not have a PTAS.

Now, we prove each of the steps from 1 to 3.

Step 1. MC to BMC reduction (see Figure 10).
Consider an MC problem with vertices and edges $(V, E), n = |V|, e = |E|$. Let a solution be of size K inducing a partition of the vertices.

Reduction: Construct an instance of BMC with $(\tilde{V}, \tilde{E})$ as follows For each $v_i \in V$, with degree d_i, construct $3d_i + 6$ vertices, v_0^{i-}, v_0^{i*}, v_0^i, v_0^{i+}, v_0^{l-}, v_0^{l+}, $v_0^{i'}$, u_0^i, u_l^i, $l = 1, 2, \ldots, d_i$. Thus the total number of vertices are $6e + 6n$.

The edges have $+1$ or -1 weights which are constructed as shown in the table (Again see Figure 10).

Edges with weights $+1$			Edges with weights -1		
edges ($l = 1, 2, \ldots, d_i$)	count per vertex	total count	edges ($l = 1, 2, \ldots, d_i$)	count per vertex	total count
$u_0^i v_0^l$, $u_0^l v_0^i$	d_i	$2e$	$u_0^i v_0^{l*}$, $u_0^l v_0^{i*}$	d_i	$2e$
$u_0^i v_l^{i+}$	d_i	$2e$	$u_0^i v_l^{i-}$	d_i	$2e$
$v_0^{i+} u_l^i$	d_i	$2e$	$v_0^{i-} u_l^i$	d_i	$2e$
$u_l^i v_l^{i+}$	d_i	$2e$	$u_l^i v_l^{i-}$	d_i	$2e$
$u_0^i v_0^{i+}$	2	$2n$	$u_0^i v_0^{i-}$	2	$2n$

Thus the number of edges with weight $+1$ is the same as the number with weight -1 which is $8e + 2n$. Further, it can be seen that this construction gives a bipartite graph with $\tilde{V} = V' \cup V''$ where $v_x^y \in V'$, $u_x^y \in V''$.

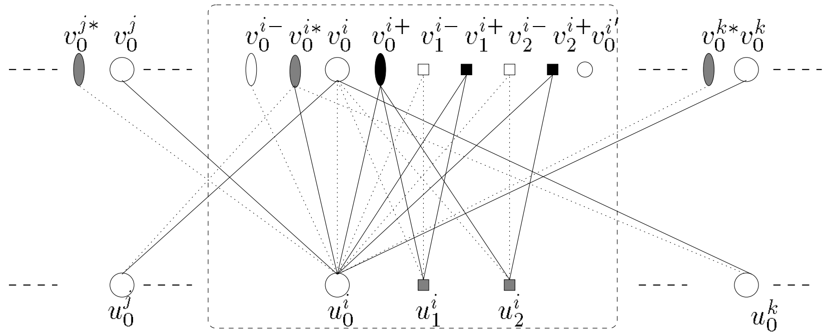


Figure 10: The MC to BMC reduction: Let the degree of the vertex numbered i in the MC be 2; its neighbors are the vertices numbered j and k. Here we show the "gadget" that is constructed for the vertex numbered i (in the dotted rectangle). The hollow circles correspond to the two copies of the vertex i of the MC problem, u_i^0 and v_i^0. The solid lines denote edges with weight $+1$ and the dotted lines denote edges with weight -1.

Observations: We make the following observations. Let

$$\begin{aligned} V_i^{in} &= \{v_0^{i-}, v_0^i, v_1^{i-}, v_2^{i-}, \ldots, v_{d_i}^{i-}, u_0^i, u_1^i, u_2^i, \ldots, u_{d_i}^i\}, \\ V_i^{out} &= \{v_0^{i+}, v_1^{i+}, v_2^{i+}, \ldots, v_{d_i}^{i+}\}, \\ V_i^{any} &= \{v_0^{i*}, v_0^i, u_0^i\}. \end{aligned}$$

1.1 It can be verified that in a solution of the BMC, the two sets S_1, S_2, are such that if $V_i^{in} \subset S_1$, then $V_i^{out} \subset S_2$ (or vice-versa). Further, if this does not hold, the solution can be modified, without decreasing the cost, so that the above condition holds.

1.2 Thus, we get a partition, that contains both v_0^i and u_0^i; we use this to construct the solution for the MC. The partition corresponding to the MC is the one that of the BMC, where we replace the set V_i^{in} by v_i (and, removing V_i^{out} and $\{v_0^{i*}\}$). Thus if $u_0^j v_0^i$ is a cut, then so is $v_0^j u_0^i$. Thus if K is the solution to the MC, then the solution to the BMC is at least $2K$.

1.3 The cost due to the vertices $V_i^{in} \cup V_i^{out} - \{v_0^i, u_0^i\}$ in a gadget is $3d_i + 2$, for every vertex, as can be verified from the construction. Notice that we are able to make such a claim since these vertices have only "local" connectivity.

Thus the total cost due to the gadgets in all the vertices is $C_{gadget} = 6e + 2n$.

1.4 Once we have a solution, and condition of observation 1.1 is satisfied, we can modify it to move the vertex v_0^{i*} around as follows. Using observation 1.2, the neighbors of every vertex

v_i (of the MC) can be partitioned into two sets. Let $V_i^{in} \in S_1$. Let C_i denote the neighbors of v_i in the MC that are in S_2, and $c_i = |C_i|$. Then $d_i - c_i \leq c_i$, where d_i is the degree of the vertex v_i in the MC problem. If not, V_i^{in} can be moved to S_2, that can only increase the cost. Thus the cost due to the vertex v_0^{i*} is $1 - (d_i - c_i)$ (recall that there is a positive edge between u_0^i and v_0^{i*}, hence the 1). If K is the solution to the maxcut, note that $2K = \sum_i c_i$. Thus the total contribution due to $v_0^{l*}, l = 1, 2, \ldots, n$, is $\sum_l (1 - (d_l - c_l)) = n - 2e + 2K$.

Note that the cost for any solution $K \geq e/2$, and the cost excluding the gadgets, $C_{\overline{gadget}}$, is as follows:

$$\begin{aligned} C_{\overline{gadget}} &= n - 2e + 2K + 2K \\ &= n - 2e + 4K \\ &\geq 0 \end{aligned} \tag{19}$$

1.4 Thus, noting that $K = C_{MC}$, we have,

$$\begin{aligned} C_{BMC} &= C_{gadget} + C_{\overline{gadget}} \\ &= (6e + 2n) - (n - 2e + 4K) \\ \Rightarrow \quad K &= C_{MC} \\ &= \frac{C_{BMC} - 4e - 3n}{4} \end{aligned} \tag{20}$$

using observations 1.2, 1.3 and 1.4.

Claim (C1.1): MC has solution K iff BMC has solution $4K + 4e + 3n$.
Proof: It can be verified from the above construction that, improving the solution for the BMC by $x > 0$, results in improving the MC by x and vice-versa. □

Step 2. BMC to BSC reduction (see Appendix for an example).
Let the incidence matrix of the BMC be $[\tilde{M}_{ij}]$. Define the matrix $[M_{ij}]$ for the BSC problem satisfying the invariance $\tilde{M}_{ij} = M_{ij} - M_{i(j-1)}, j > 1$.

Observations: We make the following observations.

2.1 In a given alignment of the BSC, the elements of M_{ij} can be grouped into the following sets:

$$\begin{aligned} A_2^+ = \{(i,j) | i \text{ is right aligned, } j \text{ is a cut}\}, A_2^- = \{(i, j-1) | (i,j) \in A_2^+, j > 1\}, \\ B_2^+ = \{(i,j) | i \text{ is left aligned, } j \text{ is a cut}\}, B_2^- = \{(i, j-1) | (i,j) \in B_2^+, j > 1\}, \\ C_2^+ = \{(i,j) | i \text{ is right aligned, } j \text{ is not a cut}\}, C_2^- = \{(i, j-1) | (i,j) \in C_2^+, j > 1\}, \\ D_2^+ = \{(i,j) | i \text{ is left aligned, } j \text{ is not a cut}\}, D_2^- = \{(i, j-1) | (i,j) \in D_2^+, j > 1\}, \end{aligned}$$

Note that $A_2^+ = C_2^-$, $A_2^- = C_2^+$, $B_2^+ = D_2^-$, $B_2^- = D_2^+$. Let $\sum A_2^+ = \sum_{(i,j) \in A_2^+} M_{ij}$. Similarly define $\sum A_2^-$, $\sum B_2^+$, $\sum B_2^-$, $\sum C_2^+$, $\sum C_2^-$, $\sum D_2^+$, $\sum D_2^-$.

(BMC)

	…	v_0^{j-}	v_0^{j*}	v_0^{j}	v_0^{j+}	…	v_0^{i-}	v_0^{i*}	v_0^{i}	v_0^{i+}	v_0^{1-}	v_0^{1+}	v_0^{2-}	v_0^{2+}	$v_0^{i'}$	…	v_0^{k-}	v_0^{k*}	v_0^{k}	v_0^{k+}	…
										$\vdots$											
u_0^j	…	-1	$+1$	-1	$+1$	…	0	-1	$+1$	0	0	0	0	0	0	…	0	0	0	0	…
										$\vdots$											
u_0^i	…	0	-1	$+1$	0	…	-1	$+1$	-1	$+1$	-1	$+1$	-1	$+1$	0	…	0	-1	$+1$	0	…
u_1^i	…	0	0	0	0	…	0	0	-1	$+1$	-1	$+1$	0	0	0	…	0	0	0	0	…
u_2^i	…	0	0	0	0	…	0	0	-1	$+1$	0	0	-1	$+1$	0	…	0	0	0	0	…
										$\vdots$											
u_0^k	…	0	0	0	0	…	0	-1	$+1$	0	0	0	0	0	0	…	-1	$+1$	-1	$+1$	…
										$\vdots$											

(BSC)

	…	v_0^{j-}	v_0^{j*}	v_0^{j}	v_0^{j+}	…	v_0^{i-}	v_0^{i*}	v_0^{i}	v_0^{i+}	v_0^{1-}	v_0^{1+}	v_0^{2-}	v_0^{2+}	$v_0^{i'}$	…	v_0^{k-}	v_0^{k*}	v_0^{k}	v_0^{k+}	…
										$\vdots$											
u_0^j	…	1	0	1	0	…	0	1	0	0	0	0	0	0	0	…	0	0	0	0	…
										$\vdots$											
u_0^i	…	0	1	0	0	…	1	0	1	0	1	0	1	0	0	…	0	1	0	0	…
u_1^i	…	0	0	0	0	…	0	0	1	0	1	0	0	0	0	…	0	0	0	0	…
u_2^i	…	0	0	0	0	…	0	0	1	0	0	0	1	0	0	…	0	0	0	0	…
										$\vdots$											
u_0^k	…	0	0	0	0	…	0	1	0	0	0	0	0	0	0	…	1	0	1	0	…
										$\vdots$											

Figure 11: Portions of incidence matrix of the BMC and the corresponding BSC matrix, for the graph shown in Figure 10.

We define the p_j's as follows:

$$p_{v_l^{i+}} = p_{v_l^{i-}} = 2/m, p_{v_0^{i-}} = 1/m, p_{v_0^{i+}} = d_i/m, p_{v_0^{i*}} = p_{v_0^i} = (d_i+1)/2m. \tag{21}$$

where $l = 1, 2, \ldots, d_i$, $\forall i$. This definition of p_j's ensures that for any two consecutive columns j and $j+1$, if by the alignment, the number of 1's in column $j+1$ is larger than in column j, then $Y_{j+1} = 1$, that is column $j+1$ is a consensus cut. This is similar to the EBFC problem where if column $\overline{j} = n - j + 1$ has larger number of 1's in the alignment than j then $Y_{\overline{j}} = 1$ and vice-versa.

2.3 Given an alignment for the BSC problem, with cost C_{BSC} as

$$C_{BSC} = B_2^+ + C_2^-, \tag{22}$$

a solution for the BMC is constructed as follows (see Figure 12). Define the sets as below:

$$\begin{aligned}
A_1^- &= \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in A_2^+\}, A_1^+ = \{v_i^1 v_j^2 | (i,(j+1)) \in A_1^-\}, \\
B_1^- &= \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in B_2^+\}, B_1^+ = \{v_i^1 v_j^2 | (i,(j+1)) \in B_1^-\}, \\
C_1^- &= \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in C_2^+\}, C_1^+ = \{v_i^1 v_j^2 | (i,(j+1)) \in C_1^-\}, \\
D_1^- &= \{v_i^1 v_j^2 | M_{ij} \neq 0, (i,j) \in D_2^+\}, D_1^+ = \{v_i^1 v_j^2 | (i,(j+1)) \in D_1^-\},
\end{aligned}$$

Let $A_1 = A_1^+ \cup A_1^-$, $B_1 = B_1^+ \cup B_1^-$, $C_1 = C_1^+ \cup C_1^-$, $D_1 = D_1^+ \cup D_1^-$. Then S_1 and S_2, the partition of the vertices, are defined as follows:

$$\begin{aligned}
S_1 &= \{v_i^1 | v_i^1 v_j^2 \in B_1, \text{ for some } j\} \cup \{v_j^2 | v_i^1 v_j^2 \in C_1, \text{ for some } i\} \cup \{v_i^1, v_j^2 | v_i^1 v_j^2 \in D_1\}, \\
S_2 &= \{v_j^2 | v_i^1 v_j^2 \in B_1, \text{ for some } i\} \cup \{v_i^1 | v_i^1 v_j^2 \in C_1, \text{ for some } j\} \cup \{v_i^1, v_j^2 | v_i^1 v_j^2 \in A_1\}.
\end{aligned}$$

Notice that $|A_1^+| = \sum A_2^-$, $|A_1^-| = \sum A_2^+$ and so on. Also notice that B_1^- is the set of edges with positive weights and B_1^+ is the set of edges with negative weights. Similarly for the other sets. Thus the corresponding cost, C_{BMC} for the BMC is,

$$C_{BMC} = |B_1^-| - |B_1^+| + |C_1^-| - |C_1^+| = |C_1^+| - |C_1^+|. \tag{23}$$

since $|B_1^+| = |B_1^-|$ by the construction.

2.4 It can be seen from the above that given a partition of the vertices in the BMC, an alignment (assignments of left/right-aligns to rows and cuts/no-cuts to columns) can be obtained for the BSC, and, vice-versa.

2.5 Let C_{BSC} denote the cost for an alignment in the BSC problem, and, let C_{BMC} denote the cost for the corresponding alignment in the BMC problem. Further let $L =$

$|\{(i,j)|i \text{ is left aligned}\}|$ and $R = |\{(i,j)|i \text{ is right aligned}\}|$. Then,

$$\begin{aligned} 2C_{BSC} - (L+R) &= (2B_2^+ - L) + (2C_2^- - R) \quad \text{using eqn(22)} \\ &= B_2^+ - B_2^- + C_2^- - C_2^+ \\ &= C_2^- - C_2^+ \end{aligned} \tag{24}$$

Thus,

$$2C_{BSC} - c = C_{BMC}. \tag{25}$$

where c $(= L + R = 8e + 2n)$ is the number of 1's in the BSC matrix.

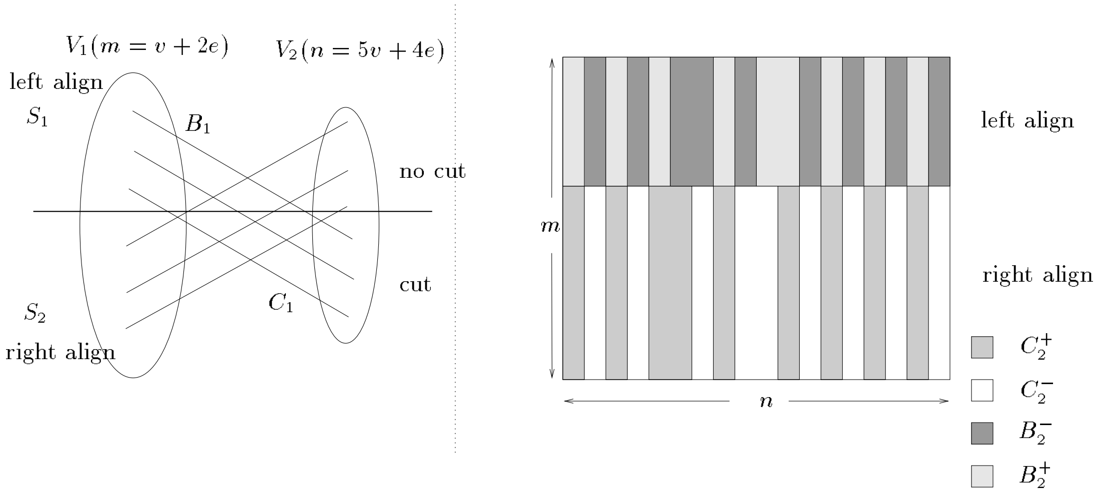


Figure 12: A schematic representation of the BMC to BSC reduction: The left shows the BMC problem and the right shows the BSC problem. See the text on the reduction for other details.

Claim (C2.1): BSC has an optimal solution of size K iff BMC has an optimal solution of size $2K - c$.

Proof: It can be verified from the above construction that, improving the solution for the BSC by $x > 0$, results in improving the BMC by x and vice-versa.

Step 3. (Arguments about MC to BSC).

It can be verified (see Figure 13 and the Appendix for examples) that, given an arbitrary solution to BSC, it can be modified, without decreasing the cost, so that the following holds (let Y_x denote the indicator variable (cut or no-cut) associated with vertex x of the BMC):

$$Y_{v_0^{i*}} = Y_{v_0^{i+}} = Y_{v_l^{i+}},\ Y_{v_0^{i'}} = 0, Y_{v_0^i} = Y_{v_0^{i-}} = Y_{v_l^{i-}}, \text{ and, } Y_{v_0^{i*}} \neq Y_{v_0^i}.$$

where $l = 1, 2, \ldots, d_i$, $\forall i$.

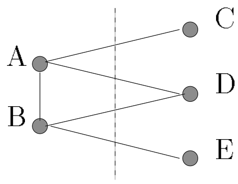
This concludes the proof of the inapproximability of the BSC problem. $\square$

Corollary 8 *Achieving an approximation ratio* $1 - \Upsilon/6$ *for BSC is NP-hard.*

Theorem 11 *$BSC_{\max K}$ problem [12] is NP-hard. Further, there exists a constant $\epsilon > 0$ such that approximating this problem within a factor of $1 - \epsilon$ is NP-hard.*

Proof This follows the lines of the proof of theorem 4. $\square$

Corollary 9 *Achieving an approximation ratio* $(1 - \Upsilon/6)\frac{p_{max}}{p_{min}}$ *for* $BSC_{\max K}$ *is NP-hard.*



A	1010...0100...0100...0100...0000	A*	-101...0010...0010...0010...00000
B	0100...1010...0000...0100...0100	B*	-010...0101...0000...0010...00100
C	0100...0000...1010...0000...0000	C	0100...0000...1010...0000...0000-
D	0100...0100...0000...1010...0000	D	0100...0100...0000...1010...0000-
E	0000...0100...0000...0000...1010	E	0000...0100...0000...0000...1010-

Figure 13: An example of MC to BSC reduction. Note that only the columns corresponding to vertices v_0^{i-}, v_0^{i*}, v_0^{i}, v_0^{i+} and rows corresponding to u_i^0, for each of the vertex A, B, C, D, E in the MC, is shown for the sake of clarity (The complete example is described in the appendix). The rows that are right aligned are marked by asterisk. Notice how the 1's in the columns align when the first 2 rows are right aligned.

7 k-Populations problem

Given m molecule sample, the k-Populations problem is the one of finding $k \geq 1$ restriction maps from the sample, where each sample has at least s_l molecules, $l = 1, 2, \ldots k$.

$$
\begin{array}{c|ccccc|}
 & V & W & X & Y & Z \\
\hline
a & 1 & 0 & 0 & 1 & 0 \\
b & 0 & 1 & 1 & 0 & 0 \\
c & 0 & 0 & 1 & 0 & 0 \\
d & 1 & 0 & 0 & 1 & 1 \\
\hline
\end{array}
$$

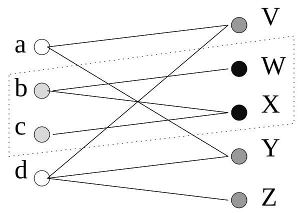


Figure 14: The matrix on the left is the input to the 2-populations problem. The corresponding bipartite graph for the Cut Classification (CC) problem, which is a min-cut problem, is as shown. The min-cut is of size 0 with the partitions $\{b, c, W, X\}$ and $\{a, d, V, Y, Z\}$. Thus the 2 populations are $\{a, d\}$ and $\{b, c\}$ and the cut classifications are $\{V, Y, Z\}$ and $\{W, X\}$.

The reader may note that all the problem considered up until now may be regarded as 1-population problem: hence hardness of the k-populations problem follows from the $k = 1$ case, wherever applicable.

The problem, whose complexity is of interest is that of the Binary Partition Cut (BPC) (discussed in Section 5) problem which was shown to have a polynomial time solution.

We will show that for a 2-populations problem using the BPC model, there is a polynomial time solution.

Cut Classification (CC) problem is defined as follows. Given a map with n (consensus) cuts, and m molecules with n sites each, the task is to classify each (consensus) cut, and, each molecule as belonging to k distinct populations, such that the total number of 0's corresponding to a no-cut column in a population is minimized.

The CC problem with $k = 2$ can be viewed as a min-cut problem, which has a polynomial time solution [5] under the condition, $s_1 > 0, s_2 > 0$.

The corresponding (bi-partite) graph is constructed as follows (see Figure 14): (1) vertex v_i, $i = 1, 2, \ldots, m$ corresponding to every molecule i, (2) vertex u_j, $j = 1, 2, \ldots, n$ corresponding to every site j, and, (3) an edge between v_i and u_j iff molecule i has the cut j (or has a 1 in location j). It is easy to see that the CC problem here corresponds to the min-cut problem.

7.1 Polynomial time algorithm for 2-populations problem

Now, we present a polynomial time algorithm for the 2-populations problem under the BPC model. This is carried out in two steps:
Step 1: Using the approach described in Section 5, compute all the consensus cuts and discard the bad molecules in polynomial time.

Step 2: Using the results of the last step, classify the (consensus) cuts into the 2-populations using the min-cut formulation stated above in polynomial time.

The min-cut formulation is NP-complete when s_1 and s_2 define specific lower bounds of the two populations [5]. But the relaxation of this condition is general enough to admit real problems. Thus under the BPC model, 2-populations problem has a polynomial time (exact) algorithm.

Conclusion

The ultimate goal of many efforts in Molecular Biology, including the Human Genome Project, is to determine the entire sequence of Human DNA and to extract genetic information from it. In this context an important step is to build *restriction maps* of portions of the DNA [1]. Various computational problems have been identified [9], [10], [12], [13]. The study of the complexity of the problems is important in the context of efficient algorithm design. In this paper, we first show that (most) of the the computational problems that have been identified in Optical Mapping [9],[10], [12],[13], are inapproximable and then obtain theoretical upper bounds on the polynomial time approximation factors of the problems that are hard.

One of the problems, the Binary Partition Cut problem, has been shown to have a polynomial time algorithm under a reasonable model: this may be of particular interest to the chemists to provide guidelines for the kinds of errors that do not make the related computational problem hard or intractable.

References

[1] N. G. Cooper (editor), *The Human Genome Project - Deciphering the Blueprint of Heredity*, University Science Books, Mill Valley, California, 1994.

[2] Y. Wang, E. Huff, D. Schwartz, *Optical Mapping of site-directed cleavages on single DNA molecules by the RecA-assisted restriction endonuclease technique*, Proc. Nat. Acad. Sci., 92, pp 165-169, January 1995.

[3] X. Meng, K. Benson, K. Chada, E. Huff, D. Schwartz, *Optical mapping of lambda bacteriophage clones using restriction endonucleases*, Nature Genetics, 9, pp 432-438, April 1995.

[4] D. Schwartz, X. Li, L. Hernandez, S. Ramnarain, E. Huff, Y. Wang. *Ordered Restriction Map of Saccharomyces cerevisiae Chromosomes Constructed by Optical Mapping*, In Proc. Natl. Acad. Sci. USA, 92:165-169, 1995.

[5] M. Garey, D. Johnson, *Computers and Intractability: A Guide to the theory of NP-Completeness*, page 210, Freeman Press, 1979.

[6] C. Papadimitriou, M. Yannakakis, *Optimization, approximation and complexity classes*, Journal of Computer and System Sciences 43, pp 425-440, 1991.

[7] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegdy, *Proof Verification and the hardness of approximation problems*, STOC, 1994.

[8] S. Arora, C. Lund, *Hardness of Approximations*, To appear in Approximation Algorithms for NP-hard Problems, PWS Publishing , 1996. (http://www.cs.princeton.edu/ arora/publist.html)

[9] D. Geiger, L. Parida, *A Model and Solution to the DNA Flipping String Problem*, Courant Inst. of Math. Sciences, TR1996-720, May, 1996.

[10] S. Muthukrishnan, L. Parida, *Towards Constructing Physical Maps by Optical Mapping: An Effective, Simple, Combinatorial Approach*, Procc of the International Conference on Computational Molecular Biology (RECOMB 97), ACM Press, Santa Fe, 1997.

[11] V. Dancik, S. Hannenhalli, S. Muthukrishnan, *Hardness of Flip-Cut Problems from Optical Mapping*, Journal of Computational Biology, 1997.

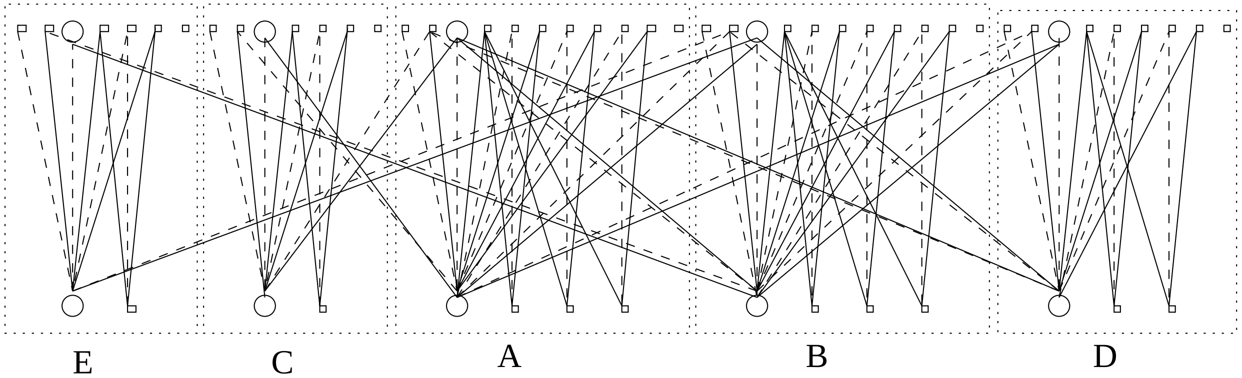
[12] T. Anantharaman, B. Mishra B, D. Schwartz, *Genomics via Optical Mapping II: Ordered Restriction Maps*, Journal of Computational Biology, 1997.

[13] L. Parida, *A Uniform Framework for Ordered Restriction Map Problems*, Courant Inst. of Math. Sciences, TR, 1997.

[14] S. Arora, D. Karger, M. Karpinski. Polynomial time approximation schemes for dense instances of NP-Hard problems. *STOC*, 1996.

[15] G. Nemhauser, L. Wolsey, *Integer and Combinatorial Optimization*, Wiley Interscience Series in Discrete Math and Optimization, 1988.

Appendix



The BMC graph for the problem in Figure 13. The 5 dotted boxes are the "gadgets" corresponding to the 5 vertices of the graph. The solid edges denote weight +1 and the dashed edges denote weight -1.

A											B											C							D									E							
1	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	√
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	√
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	√
0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	√
0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	√
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	√
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	√
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	√
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	×
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	×
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	×
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	×
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	×
×	√	×	√	×	√	×	√	×	√	×	×	√	×	√	×	√	×	√	×	√	×	√	×	√	×	√	×	×	√	×	√	×	√	×	√	×	×	√	×	√	×	√	×	×	S'
2	4	4	6	4	4	4	4	4	4	4	2	4	4	6	4	4	4	4	4	4	4	2	2	2	2	4	4	4	2	3	3	4	4	4	4	4	4	2	3	3	4	4	4	4	$30p_j$

The BSC matrix is constructed from the graph shown above. The rows/molecules that are marked by $\surd$ denote the ones that are shifted to the right by one unit. The consensus map S' is shown at the bottom with 1 denoting a consensus cut site and 0 its absence. The p_j for each column is shown in the bottom line: x denotes probability $x/30$.