A Convex Atomic-Norm Approach to Multiple Sequence Alignment and Motif Discovery

Ian E. H. Yen [1] * IANYEN@CS.UTEXAS.EDU
Xin Lin [1] * JIMMYLIN@UTEXAS.EDU
Jiong Zhang [2] ZHANGJIONG724@UTEXAS.EDU
Pradeep Ravikumar [1,2] PRADEEPR@CS.UTEXAS.EDU
Inderjit S. Dhillon [1,2] INDERJIT@CS.UTEXAS.EDU

* Both authors contributed equally.

[1] Department of Computer Science, University of Texas at Austin, TX 78712, USA.
[2] Institute for Computational Engineering and Sciences, University of Texas at Austin, TX 78712, USA.

Abstract

Multiple Sequence Alignment and Motif Discovery, known as NP-hard problems, are two fundamental tasks in Bioinformatics. Existing approaches to these two problems are based on either local search methods such as Expectation Maximization (EM), Gibbs Sampling or greedy heuristic methods. In this work, we develop a convex relaxation approach to both problems based on the recent concept of atomic norm and develop a new algorithm, termed Greedy Direction Method of Multiplier, for solving the convex relaxation with two convex atomic constraints. Experiments show that our convex relaxation approach produces solutions of higher quality than standard tools widely-used in Bioinformatics community on Multiple Sequence Alignment and Motif Discovery problems.

1. Introduction

Multiple Sequence Alignment (MSA) Given an alphabet Σ, let $\hat{\Sigma} = \Sigma \cup \{*, \#\}$ be its extension with start-end symbols and $\Sigma_{\#}$ the extension with end symbol only. An *alignment* between two sequences $x_1 \in \hat{\Sigma}^{\ell_1}$, $x_2 \in \hat{\Sigma}^{\ell_2}$ of length ℓ_1, ℓ_2 can be defined as a path of state transitions, where each state $(i, j) \in \mathcal{S} = [\ell_1] \times [\ell_2]$ is a pair of read positions on x_1, x_2, and the set of possible state transitions $\mathcal{T} \subset \mathcal{S} \times \mathcal{S}$ are

$$\mathcal{T}^I = \{((i,j),(i+1,j)) \mid i \in [\ell_1 - 1], j \in [\ell_2]\}$$
$$\mathcal{T}^D = \{((i,j),(i,j+1)) \mid i \in [\ell_1], j \in [\ell_2 - 1]\}$$
$$\mathcal{T}^M = \{((i,j),(i+1,j+1)) \mid i \in [\ell_1 - 1], j \in [\ell_2 - 1]\},$$

termed as insertion, deletion, and matching transition respectively. A collection of transitions $\{t_k\}_{k=1}^K$ is a *path* iff any state s involved has exactly one incoming transition and one outgoing transition, except a begin state s_B of only outgoing transition and an end state s_E of only incoming transition. An alignment a is a path with begin state $s_B = (1, 1)$ and end state $s_E = (\ell_1, \ell_2)$. To evaluate quality of an alignment, a scoring function of form

$$d(t; x_1, x_2) = \begin{cases} d_I & , t \in \mathcal{T}^I \\ d_D & , t \in \mathcal{T}^D \\ d_M(x_1[i+1], x_2[j+1]) & , t \in \mathcal{T}^M \end{cases}$$

is given that specifies the penalty given for each insertion d_I, deletion d_D and mismatch $d_M(x_1[i+1], x_2[j+1])$ transition where $d_M(a, b) = 0$ if $a = b$. The cost of an alignment a is then defined by

$$d(a; x_1, x_2) := \sum_{t \in a} d(t; x_1, x_2).$$

The problem of finding the best alignment $a^* = arg\min_a \; d(a; x_1, x_2)$ between two sequences, known as *Pairwise Alignment*, can be solved within the time complexity of $O(\ell_1 \ell_2)$ via the *Needleman-Wunsch* algorithm (or the *Smith-Waterman* algorithm for the scenario of finding local alignments).

Given a collection of sequences $\mathcal{D} = \{x_n\}_{n=1}^N$ of different lengths $\{\ell_n\}_{n=1}^N$ with total length $\ell = \sum_{n=1}^N \ell_n$, the problem of *Multiple Sequence Alignment* can be defined in a number of ways. For this work, we consider

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

the formulation named *MSA with Consensus*, in which one aims at finding a consensus sequence y and its alignments $a = (a_n)_{n=1}^{N}$ w.r.t. each of N sequences jointly. Formally,

$$(y^*, a^*) = \underset{y,a}{argmin} \sum_{n=1}^{N} d(a_n; x_n, y), \tag{1}$$

where optimal solution y^* is also known as *Steiner string* with $d(.)$ being edit distance. The above objective (1) is also known as the *Star Alignment* (in distinction to the *Sum-of-Pairs* objective, for which a convex relaxation approach has been considered in (Alayrac et al., 2015)). However, Star Alignment is proved to be NP-hard just as any other formulation of MSA (Elias, 2003). When using *Profile-HMM* to model a collection of sequences, the *Maximum-Likelihood* estimation gives an objective similar to (1), where y is replaced with parameters $\boldsymbol{\pi}$ that specifies a distribution of consensus sequence, known as *Profile* (Durbin et al., 1998). The standard method for estimating Profile-HMM from *un-aligned* sequences is the *Baum-Welch* (or *Expectation Maximization (EM)*) algorithm that maximizes log-likelihood w.r.t. the alignments a and profile $\boldsymbol{\pi}$ alternately. However, it has been shown that this approach barely outperforms (heuristic) progressive algorithm as implemented in, for example, *Clustal-W* (Notredame, 2002; Karplus & Hu, 2001).

Motif Discovery (MD) The Motif Discovery problem can be interpreted as a generalization of MSA that aims to find K *consensus motifs* y_1, y_2, ...,y_K jointly that are aligned to *different segments* of sequences $\{x_n\}_{n=1}^{N}$, so the characters being matched are as many as possible. If there is a solution s.t. every character is matched, the problem can be also interpreted as a *decipher problem* where sequences $\{x_n\}_{n=1}^{N}$ are the encoded text and $\{y_k\}_{k=1}^{K}$ is the unknown coding book to find. Formally, the MD problem aims to solve

$$(y^*, a^*) := \underset{y,a}{argmin} \sum_{n=1}^{N} d(a_n; x_n, y_1, ..., y_K) \tag{2}$$

where $y = (y_k)_{k\in[K]}$ are K motifs to estimate and $a = (a_n)_{n\in[N]}$ are alignments that consider the set of possible transitions $\mathcal{T}_n = \mathcal{T}_n^M \cup \mathcal{T}_n^S$ for each sequence to be aligned

$$\mathcal{T}_n^M = \left\{ ((i,j,k),(i+1,j+1,k)) \;\middle|\; \begin{array}{l} i \in [\ell_n - 1], \\ j \in [L_k - 1], \\ k \in [K] \end{array} \right\}$$

$$\mathcal{T}_n^S = \left\{ ((i,L_k,k),(i+1,1,k')) \;\middle|\; \begin{array}{l} i \in [\ell_n - 1], \\ k,k' \in [K] \cup \{U\} \end{array} \right\},$$

termed as the *match* and *switch* transition respectively. Note that the MD problem, in its simplest form, does not consider insertion-deletion noise presumably due to its intrinsic difficulty, and therefore, finding alignment a becomes equivalent to finding the *start positions* of motifs on the data sequences (Das & Dai, 2007). Besides the K motifs, an additional dummy motif y_U of length $L_U = 1$ is added to align with any segment that does not match any motif under an *unmatch penalty* d_U. For any transition $t = ((i,j,k),(i',j',k')) \in \mathcal{T}_n$, the cost function is defined as

$$d(t; x_n, y_1, ..., y_K) = \begin{cases} d_U, & k' = U \\ d_M(x_n[i'], y_{k'}[j']), & o.w. \end{cases}$$

where $d_M(a,b) = 0$ if $a = b$. Among many Motif Discovery algorithms, the probabilistic modelling approaches have been the most widely-used (Bailey et al., 2006; Siddharthan et al., 2005; Lawrence et al., 1993), where an objective similar to (2) is derived from maximum likelihood or Bayesian criteria with (y,a) replaced by their probabilistic counterparts. However, the algorithms being used for fitting those models are based on local-search methods such as EM (Bailey et al., 2006) or Gibbs sampling (Siddharthan et al., 2005; Lawrence et al., 1993) that does not find solution of guaranteed quality.

In this work, we develop convex relaxation for the MSA and MD problem based on recent concept of atomic norm and convex atomic constraint. The problem of minimizing convex, smooth function subject to a feasible domain being *convex hull of atomic set* has been widely studied in the recent years (Jaggi, 2013; Tewari et al., 2011), where one can exploit the specific structure of atomic set $\mathcal{S}$ to find search direction efficiently (Jaggi, 2013). However, the objective developed in this work is constrained by the intersection of two convex hulls of atomic sets, which makes standard algorithms for convex atomic domain not directly applicable to our case. In section 3, we propose a new algorithmic framework that combines *Augmented Lagrangian* with a variant of *Frank-Wolfe* (Lacoste-Julien & Jaggi, 2015) to efficiently solve the convex relaxation of MSA/MD problems with convergence guarantee.

2. Problem Formulation

2.1. A Convex Relaxation for Multiple Sequence Alignment (MSA)

To define a convex relaxation of (1), we define the ambient space that can be used to represent any consensus sequence and its alignment w.r.t. the N sequences [1]. Since the consensus sequence is unknown, we will consider L as the maximum possible length of it, and use $(i,q) \in \mathcal{S}_n = [\ell_n] \times \mathcal{Q}$ to represent the state of alignment between n-th sequence and y, where $q = (j,c) \in \mathcal{Q} = [L] \times \hat{\Sigma}$ is a read on the j-th position of y with any candidate symbol c. The states $\mathcal{S}_n$ we are considering can be illustrated as a $\ell_n \times L \times |\hat{\Sigma}|$ cube in Figure 1. Denoting $\mathcal{Q}_{-1} = \mathcal{Q} \setminus \{(j,c)|j = L-1\}$, the set of valid transitions

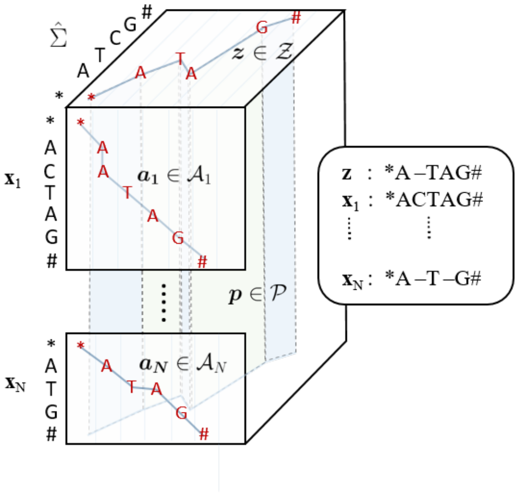

Figure 1. Illustration of $W \in \mathcal{M}$ for Multiple Sequence Alignment as an $\ell \times L \times |\hat{\Sigma}|$ cube, where $\boldsymbol{x}_n$ is data sequence, $\boldsymbol{z}$ is consensus sequence, $\boldsymbol{a}_n$ is the alignment of $\boldsymbol{x}_n$ to $\boldsymbol{z}$, and $\boldsymbol{p}$ is a *consensus atom* that has support within $\boldsymbol{z}$ when projected onto the upper surface of the cube.

$\mathcal{T}_n = \mathcal{T}_n^I \cup \mathcal{T}_n^D \cup \mathcal{T}_n^M \subset \mathcal{S}_n \times \mathcal{S}_n$ are defined as

$$\mathcal{T}_n^I = \{((i,q),(i+1,q)) \mid i \in [\ell_n - 1], q \in \mathcal{Q}\}$$

$$\mathcal{T}_n^D = \left\{ ((i,q),(i,q')) \;\middle|\; \begin{array}{l} q = (j,c) \in \mathcal{Q}_{-1} : c \neq \# \\ q' = (j+1,c'), \forall c' \in \Sigma_\# \end{array} \right\}$$

$$\mathcal{T}_n^M = \left\{ ((i,q),(i+1,q')) \;\middle|\; \begin{array}{l} q = (j,c) \in \mathcal{Q}_{-1} : c \neq \# \\ q' = (j+1,c'), \forall c' \in \Sigma_\# \end{array} \right\}.$$

where the mismatch/match transition $\mathcal{T}_n^M$ and the deletion transitions $\mathcal{T}_n^D$ going from j to $j+1$-th position of consensus sequence can read any symbol $c \in \Sigma_\#$ at the new position. An alignment is then defined as a path inside the cube illustrated in Figure 1, with begin state $s_B = (1,(1,*))$ and end state $s_E \in \mathcal{S}_n^E = \{(\ell_n,(j,\#)) \mid j \in [L]\}$. Note one can now evaluate the cost of a transition by:

$$d(t;x_n) = \begin{cases} d_I & ,t \in \mathcal{T}_n^I \\ d_D & ,t \in \mathcal{T}_n^D \\ d_M(x_n[i+1], c'(t)) & ,t \in \mathcal{T}_n^M. \end{cases}$$

where $c'(t)$ denotes the symbol of consensus sequence used by the destination state of t.

Then we define the ambient space to be $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 ... \times \mathcal{M}_N$ where $\mathcal{M}_n := \{0,1\}^{|\mathcal{T}_n|}$. Let $a(n,t) \in \{0,1\}$ be the indicator of whether transition t is used in the alignment of n-th sequence, and $\boldsymbol{a}_n := (a(n,t))_{t \in \mathcal{T}_n}$, $D_n := (d(t,x_n))_{t \in \mathcal{T}_n}$. The cost of an alignment a_n can be written as

$$\langle D_n, \boldsymbol{a}_n \rangle = \sum_{t \in \mathcal{T}_n} d(t;x_n) a(n,t) \qquad (3)$$

and the constraint for $W_n \in M_n$ to be a valid alignment can be expressed by $W_n \in \mathcal{A}_n$, where $\mathcal{A}_n$ denotes the set of all valid alignments between sequence x_n to another sequence. Note $W_n \in \mathcal{A}_n$ does not constrain x_1,...,x_N to align to the same consensus sequence.

To impose the *consensus constraint*, denote $\mathcal{T}_Q \subset \mathcal{Q} \times \mathcal{Q}$ as the set of valid transitions on $\mathcal{Q}$ of the form $((j,c),(j+1,c'))$ and $\mathcal{M}_Q = \{0,1\}^{|\mathcal{T}_Q|}$ as the ambient space of consensus sequence. We can then define consensus sequence $\boldsymbol{z} \in \mathcal{M}_Q$ as indicator vector that has value 1 on a path of transitions beginning with $q_B = (1,*)$, ending with $q_E \in \mathcal{Q}_E = \{(j,\#) | j \in [L]\}$, and value 0 on all other entries, as illustrated on top surface of the cube in Figure 1. Let $\mathcal{Z} \subset \mathcal{T}_Q$ be the set of all possible consensus sequences. For all $\boldsymbol{z} \in \mathcal{Z}$, we define

$$\mathcal{P}_{\boldsymbol{z}} = \{W \in \mathcal{M} \mid supp(\mathbf{proj}_{\mathcal{M}_Q}(W)) \subseteq \boldsymbol{z}\} \qquad (4)$$

as the set of any W with support in $\boldsymbol{z}$ when projected to $\mathcal{M}_Q$, and $\mathcal{P} = \bigcup_{\boldsymbol{z} \in \mathcal{Z}} \mathcal{P}_{\boldsymbol{z}}$ as the set of all W with only one supporting consensus sequence. Then the Multiple Sequence Alignment problem can be equivalently formulated as

$$\begin{aligned} \min_{W \in \mathcal{M}} \quad & \langle D, W \rangle \\ s.t. \quad & W_n \in \mathcal{A}_n, \; n \in [N] \\ & W \in \mathcal{P}. \end{aligned} \qquad (5)$$

where $D_{n,t} = d(t;x_n)$. One can verify that the feasible set of (5) exactly profiles the search space of the MSA problem (1): any W that corresponds to N alignments to one consensus sequence. To define a convex relaxation of (5), we relax the integer domain M_n to domain of real values between 0 and 1: $\mathcal{M}_\mathbb{R} := Conv(\mathcal{M})$ and solve

$$\begin{aligned} \min_{W \in \mathcal{M}_\mathbb{R}} \quad & \langle D, W \rangle \\ s.t. \quad & W_n \in Conv(\mathcal{A}_n), \; n \in [N] \\ & W \in Conv(\mathcal{P}). \end{aligned} \qquad (6)$$

where $Conv(\mathcal{V})$ denotes the convex hull formed by *atoms* $\boldsymbol{a} \in \mathcal{V}$. Note the tractability of problem (6) lies in the domain $Conv(\mathcal{A}) \bigcap Conv(\mathcal{P})$, which is a superset of a more straightforward relaxation $Conv(\mathcal{A} \bigcap \mathcal{P})$. The latter leads to an NP-hard problem since it always produces an optimal solution of (5) by adding perturbation to D to induce unique integer solution, which leads to contradiction.

The relaxation (6) is not tight in general. Solution of (6) could be fractional and lies in the interior of convex hulls. Fortunately, the structure of (6) permits simple rounding scheme for transferring any solution of the relaxed problem (6) to a feasible solution of the combinatorial problem (5), which we will describe in section 3.3. In our

[1] We overload several notations, such as $\mathcal{S}, \mathcal{T}, \mathcal{Q}$, in different scenarios as long as their meanings are clear from the context.

experiment, we found rounding from (6) yields solution that outperforms standard MSA tools and is sometimes as good as ground truth. We also observed cases when (6) yields integer solution W^* directly, in which case the solution W^* is also optimal to (5) since any integer solution in $Conv(\mathcal{A}) \bigcap Conv(\mathcal{P})$ must satisfy $W^* \in \mathcal{A}$ and $W^* \in \mathcal{P}$.

2.2. Convex Relaxation for Motif Discovery (MD)

To define the convex relaxation for (2), let L_m, L_M be the minimum and maximum possible lengths of a motif, and

$$\mathcal{Y} = \left(\bigcup_{l=L_m}^{L_M} \Sigma^l \right) \cup \{y_U\} \tag{7}$$

be the set of all possible motifs and let $\bar{K} = |\mathcal{Y} \setminus \{y_U\}|$. We denote L_k as the length of motif y_k with $L_U = 1$. Instead of allowing arbitrary symbol c to be aligned as in previous section, we specify the symbol being read by motif index k and position j, denoted as $y_j[k]$. Defining $q \in \mathcal{Q} = \{(j,k) \mid j \in [L_k], k \in [\bar{K}] \cup \{U\}\}$ as the read state on the motif, and $s = (i,q) \in \mathcal{S}_n = [\ell_n] \times \mathcal{Q}$ as the state of alignment on the n-th sequence, the set of possible transitions between states can be defined as $\mathcal{T}_n = \mathcal{T}_n^M \cup \mathcal{T}_n^S$ where

$$\mathcal{T}_n^M = \left\{ ((i,q),(i+1,q')) \;\middle|\; \begin{array}{l} i \in [\ell_n - 1], \\ q = (j,k), q' = (j+1,k) \\ j \in [L_k - 1], k \in [\bar{K}] \end{array} \right\}$$

$$\mathcal{T}_n^S = \left\{ ((i,q),(i+1,q')) \;\middle|\; \begin{array}{l} i \in [\ell_n - 1] \\ q = (L_k,k), q' = (1,k'), \\ k,k' \in [\bar{K}] \cup \{U\} \end{array} \right\}.$$

Suppose each data sequence $\boldsymbol{x}_n$ has a start symbol $*$ appended at position 0 that does not require to be matched. The alignment of n-th sequence is defined as a path of transitions from any begin state $s_B \in \mathcal{S}_B$ to any end state $s_E \in \mathcal{S}_E$ where

$$\begin{aligned} \mathcal{S}_B &= \{(0,q) \mid q = (1,U),\} \\ \mathcal{S}_E &= \{(\ell_n,q) \mid q = (L_k,k), k \in [\bar{K}] \cup \{U\}\}, \end{aligned}$$

and each transition $t = ((i,(j,k)),(i',(j',k'))) \in \mathcal{T}_n$ incurs a cost given by

$$d(t;x_n) = \left\{ \begin{array}{ll} d_U, & k' = U \\ d_M(x_n[i'], y_{j'}[k']), & o.w. \end{array} \right.$$

After the above modification of state and transition sets, we can define the ambient space $\mathcal{M}$, alignment space $\mathcal{A}$ and cost vector $\mathcal{D}$ in the same way of section 2.1 such that the alignment cost is given by $\langle D, W \rangle$, and the constraint for W_n being a valid alignment is $W_n \in \mathcal{A}_n$. To constrain number of motifs learned from sequences, let $\mathcal{T}_Q \subset \mathcal{Q} \times \mathcal{Q}$ denotes the transitions (q,q') on $\mathcal{Q}$ that can be induced from any transition $t \in \mathcal{T}$ and $\mathcal{M}_Q = \{0,1\}^{|\mathcal{T}_Q|}$ as the ambient space for representation of motifs. We can then define the indicator vector of a motif as $\boldsymbol{z}_k \in \mathcal{M}^{|\mathcal{T}_Q|}$ that has value 1 on all transitions (q,q') involving k-th motif and value 0 on all other transitions $t \in \mathcal{T}_Q$. Then let $\mathcal{P}_k = \{W \in M \mid supp(\mathbf{proj}_{M_Q}(W)) \subseteq \boldsymbol{z}_k\}$ be the group of $W \in \mathcal{M}$ of support within $\boldsymbol{z}_k$ when projected to $\mathcal{M}_Q$, and $\mathcal{P} = \bigcup_{k \in [\bar{K}]} \mathcal{P}_k$ be the group of W with support within one motif when projected to $\mathcal{M}_Q$. The Motif Discovery problem (2) can be reformulated as

$$\begin{array}{ll} \min\limits_{W \in \mathcal{M}} & \langle D, W \rangle \\ s.t. & W_n \in \mathcal{A}_n, \; n \in [N] \\ & \Omega_{\mathcal{P}_C}(W) = K. \end{array} \tag{8}$$

where K is the desired number of motifs, and $\Omega_{\mathcal{P}_C}(.)$ is the atomic norm defined on the convex hull $\mathcal{P}_C = Conv(\mathcal{P})$. Note in the integer domain $\mathcal{M}$, W satisfying $\Omega_{\mathcal{P}_C}(W) = K$ can be written as $W = \sum_{k \in \mathcal{B}} \boldsymbol{p}_k$ for some set $\mathcal{B}$ of K distinct motifs and $\boldsymbol{p}_k \in \mathcal{P}_k$. So any feasible W involves exactly K motifs and is also a valid alignment (since $W \in \mathcal{A}$). Therefore, the feasible domain of (8) exactly reflects the search space of (2), giving equivalence between the two problems. The convex relaxation of (8) can be written as

$$\begin{array}{ll} \min\limits_{W \in \mathcal{M}_{\mathbb{R}}} & \langle D, W \rangle \\ s.t. & W_n \in Conv(\mathcal{A}_n), \; n \in [N] \\ & \Omega_{\mathcal{P}_C}(W) \leq K, \end{array} \tag{9}$$

which simply replaces $\mathcal{M}$, $\mathcal{A}_n$ by the convex hulls of them. Note although the ambient space $\mathcal{M}_{\mathbb{R}}$ considers exponentially large number of variables proportional to $|\mathcal{Y}|$, only a small data-dependent subset $\mathcal{Y}_{\boldsymbol{x}}$ will be considered by the algorithm we will introduce in section 3.

Although (9) in general does not have the same solution as (8), when (9) has unique solution and there exists a perfect solution (coding book) $\{y_k^*\}_{k=1}^K$ to the Motif Discovery problem (2) that incurs no unmatch d_U or mis-match cost d_M, the relaxation (9) finds solution W^* equivalent to the optimal solution of (2), depicted as the following theorem.

Theorem 1 (Realizable Case)**.** *Suppose there exists a perfect solution $\{y_k^*\}_{k=1}^K$ to problem (2) with $\sum_{n \in [N]} d(a_n^*; x_n, y_1^*, ..., y_K^*) = 0$ and (9) has a unique solution W^*. Then W^* is optimal solution to the Motif Discovery problem (8).*

Proof. This follows directly from the fact that $\langle D, W^* \rangle$ cannot have loss lower than 0 but the relaxation (9) should achieve a loss not higher than that of (8), so $\langle D, W^* \rangle = 0$ and by uniqueness W^* is the optimal solution of (8). □

Algorithm 1 Greedy Direction Method of Multiplier

```
0: Initialize $\boldsymbol{v}_0 := arg\min_{\boldsymbol{v}\in\mathcal{A}}\langle D, \boldsymbol{v}\rangle$, $W_1^0 = \boldsymbol{v}_0$, $W_2^0 = \boldsymbol{0}$, $Y^0 = \boldsymbol{0}$ and active set $\mathcal{B}^0 = \{\boldsymbol{v}_0\}$.
   for $t = 0, 1, 2, ..$ do
1:     Perform Algorithm 2 on (13) until a non-drop step.
2:     Update Lagrangian multipliers via (14).
   end for
```

Algorithm 2 Away-Steps Frank-Wolfe (one non-drop step)

```
   for $s = 0, 1, 2, ...$ do
1:     $\boldsymbol{v}_{FW} = \underset{\boldsymbol{v}\in\mathcal{A}\times\mathcal{G}}{argmin}\ \langle\nabla\mathcal{L}, \boldsymbol{v}\rangle$ ; $\boldsymbol{d}_{FW} = \boldsymbol{v}_{FW} - W^{(t,s)}$.
2:     $\boldsymbol{v}_A = \underset{\boldsymbol{v}\in\mathcal{B}^{(t,s)}}{argmax}\ \langle\nabla\mathcal{L}, \boldsymbol{v}\rangle$; $\boldsymbol{d}_A = W^{(t,s)} - \boldsymbol{v}_A$.
       if $\langle\nabla\mathcal{L}, \boldsymbol{d}_{FW}\rangle \le \langle\nabla\mathcal{L}, \boldsymbol{d}_A\rangle$ then
3:         $\boldsymbol{d} := \boldsymbol{d}_{FW}$, $\gamma_{\max} := 1$,
           $\mathcal{B}^{(t,s+\frac{1}{2})} := \mathcal{B}^{(t,s)} \cup \{\boldsymbol{v}_{FW}\}$
       else
4:         $\boldsymbol{d} := \boldsymbol{d}_A$, $\gamma_{\max} := \beta_{\boldsymbol{v}_A}^{(t,s)}/(1-\beta_{\boldsymbol{v}_A}^{(t,s)})$.
       end if
5:     $\gamma^* := \underset{\gamma\in[0,\gamma_{\max}]}{argmin}\ \mathcal{L}\left((W_1^{(t,s)}, W_2^{(t,s)}) + \gamma\boldsymbol{d}, Y^{(t)}\right)$.
6:     $(W_1^{(t,s+1)}, W_2^{(t,s+1)}) := (W_1^{(t,s)}, W_2^{(t,s)}) + \gamma^*\boldsymbol{d}$.
7:     $\mathcal{B}^{(t,s+1)} := \mathcal{B}^{(t,s+\frac{1}{2})} \setminus \{\boldsymbol{v} \mid \beta_{\boldsymbol{v}}^{(t,s)} = 0\}$.
       Break if $\gamma^* < \gamma_{\max}$.
   end for
```

3. Algorithm

In this section, we propose an algorithmic framework, termed Greedy Direction Method of Multiplier (GDMM), for solving problem of form

$$\begin{aligned} \min_{W\in\mathcal{M}_{\mathbb{R}}} \quad & \langle D, W\rangle \\ s.t. \quad & W \in Conv(\mathcal{A}) \\ & W \in Conv(\mathcal{G}), \end{aligned} \tag{10}$$

that subsumes both the convex relaxation of Multiple Sequence Alignment (6) and Motif Discovery (9). We will first state the general algorithm in section 3.1, and in section 3.2 we describe ways to realize the *Linear Minimization Oracles (LMO)* for atomic sets $\mathcal{A}_n$, $\mathcal{G}$ in (6) and (9) respectively.

3.1. Greedy Direction Method of Multiplier

Note problem of form (10) is easy to solve when there is only *alignment constraints* $W_n \in Conv(\mathcal{A}_n)$ or only *consensus constraint* $W \in Conv(\mathcal{G})$. In the former case, the problem can be solved via a Dynamic Programming (DP) method similar to the *Smith-Waterman* algorithm (Smith & Waterman, 1981), while in the latter case, a DP method (introduced in section 3.2) similar to the *Viterbi* algorithm can find solution directly. To decouple the two atomic constraints in (10), we solve the equivalent problem

$$\begin{aligned} \min_{W_1, W_2\in\mathcal{M}_{\mathbb{R}}} \quad & \langle D, W_1\rangle + \frac{\rho}{2}\|W_1 - W_2\|^2 \\ s.t. \quad & W_1 \in Conv(\mathcal{A}) \\ & W_2 \in Conv(\mathcal{G}) \\ & W_1 = W_2 \end{aligned} \tag{11}$$

via an *Augmented Lagrangian (AL)* method. Define the AL function $\mathcal{L}(W_1, W_2, Y)$ as

$$\langle D, W_1\rangle + \langle Y, W_1 - W_2\rangle + \frac{\rho}{2}\|W_1 - W_2\|^2 \tag{12}$$

where Y is the Lagrangian Multipliers. At each iteration, the GDMM algorithm performs few iterates of *Away-Steps Frank-Wolfe (AFW)* algorithm on the AL subproblem

$$\min_{(W_1,W_2)\in Conv(\mathcal{A})\times Conv(\mathcal{G})} \mathcal{L}(W_1, W_2, Y^{(t)}), \tag{13}$$

followed by the update of Lagrangian Multipliers

$$Y^{(t+1)} = Y^{(t)} + \eta\left(W_1^{(t+1)} - W_2^{(t+1)}\right) \tag{14}$$

where $0 < \eta \le 1$ is a constant step size.

AFW is a variant of Frank-Wolfe method that achieves linear convergence rate on problem of form (12) (Lacoste-Julien & Jaggi, 2015). A Frank-Wolfe (FW) method each iteration moves towards an atom obtained by the *Linear Minimization Oracle (LMO)*

$$\boldsymbol{v}_{FW} \in \underset{\boldsymbol{v}\in Conv(\mathcal{A})\times Conv(\mathcal{G})}{argmin} \langle\nabla\mathcal{L}(W_1, W_2, Y^{(t)}), \boldsymbol{v}\rangle. \tag{15}$$

And different to vanilla FW method, the AFW algorithm maintains an active set $\mathcal{B}$ of atoms with non-zero coefficients $\{\beta_{\boldsymbol{v}}\}_{\boldsymbol{v}\in\mathcal{B}}$ explicitly, so for any iterate (t, s) the variable (W_1, W_2) of interest can be expressed as

$$(W_1^{(t,s)}, W_2^{(t,s)}) = \sum_{\boldsymbol{v}\in\mathcal{B}^{(t,s)}} \beta_{\boldsymbol{v}}^{(t,s)}\boldsymbol{v}, \tag{16}$$

and the AFW algorithm performs an *away step* (step 2, 4 of Algorithm 2) whenever ”moving away” from an atom $\boldsymbol{a} \in \mathcal{B}$ leads to more progress than moving towards atom $\boldsymbol{v}_{FW}$ in terms of the linear approximation. We sketch the AFW algorithm in Algorithm (2). Note the LMO (15) decomposes into subproblems

$$\boldsymbol{v}_{FW}^{(1,n)} := \underset{W_{1,n}\in Conv(\mathcal{A}_n)}{argmin} \langle\nabla_{W_{1,n}}\mathcal{L}, W_{1,n}\rangle, \forall n \in [N] \tag{17}$$

$$\boldsymbol{v}_{FW}^{(2)} := \underset{W_2\in Conv(\mathcal{G})}{argmin} \langle\nabla_{W_2}\mathcal{L}, W_2\rangle. \tag{18}$$

In section 3.2, we will discuss how to efficiently compute (17), (18) for MSA (6) and MD (9) problem.

Algorithm 3 Extended Smith Waterman Algorithm (ESW)

```
input : $G := -\nabla_{W_{1,n}}\mathcal{L}(.)$ , $\mathcal{T}_n$.
output : $\boldsymbol{v}_{FW}^{(1,n)}$ satisfying (17).
  Initialize $R(s) = 0, s \in \mathcal{S}_B$.
  for $s' \in \mathcal{S} \setminus \mathcal{S}_B$ in a topological order w.r.t. $\mathcal{T}_n$ do
    $R(s') := \max_{(s,s') \in \mathcal{T}_n}\ R(s) + G((s,s'))$.
    $S(s') \in arg\max_{s:(s,s') \in \mathcal{T}_n}\ R(s) + G((s,s'))$.
  end for
  Let $s_E \in arg\max_{s \in \mathcal{S}_E} R(s)$.
  Find path $a$ by traceback from $s_E$ to $s_B \in \mathcal{S}_B$ via $S(.)$.
  $\boldsymbol{v}_{FW} :=$ indicator vector of path $a$ (in domain $\mathcal{M}_n$).
```

Note if solving (13) exactly, Algorithm 1 is equivalent to standard *Augmented Lagrangian Method*, which however is prohibitively expensive. As one technical contribution of this work, we show in the following theorem that it suffices to perform one "non-drop step" of AFW to ensure convergence of Algorithm 1 to optimum with a $O(1/t)$-type rate, where a non-drop step refers to a step that has $\gamma^* < \gamma_{\max}$. Since any step with $\gamma^* = \gamma_{\max}$ will result in removal of atom from the active set $\mathcal{B}$, the number of non-drop step is at least half of the total number of AFW steps (Lacoste-Julien & Jaggi, 2015). Then the number of AFW steps is at most $2t$ for t GDMM iterations. Note that linear convergence behavior of AFW plays a crucial role in the convergence analysis of GDMM—a vanilla Frank-Wolfe method would not be sufficient to obtain the following result.

Theorem 2 (Convergence of GDMM). *Let* $d(Y) = \min_{W_1,W_2} \mathcal{L}(W_1, W_2, Y)$ *be the dual objective of* (12) *and define* $\Delta_d^t := d^* - d(Y^t)$ *and* $\Delta_p^t := \mathcal{L}(W_1^{t+1}, W_2^{t+1}, Y^t) - d(Y^t)$ *as the dual and primal suboptimality. Then the iterates* $\{(W_1^t, W_2^t, Y^t)\}_{t=1}^{\infty}$ *produced by Algorithm 1 has*

$$\Delta_p^t + \Delta_d^t \leq \frac{\omega}{t}$$

with $\omega = 4(1 + \frac{1}{\kappa}) \max\left(\Delta_p^0 + \Delta_d^0, 2R_Y^2/\rho\right)$, *where* κ *is a constant depending on the smoothness and (generalized) strong convexity constant of* (12), *pyramidal width of polyhedral domain* $Conv(\mathcal{A}) \times Conv(\mathcal{G})$. R_Y *is an upper bound on the distance of dual iterate* Y^t *to the optimal solution set of* $d(Y)$.

Note in Theorem (2), the constant κ depends on quantities such as pyramidal width of $Conv(\mathcal{A}) \times Conv(\mathcal{G})$ and generalized strong convexity constant of (12), whose relationships to input dimensions N, T are still research issues. In practice, we observe that performing AFW on W_1, W_2 in a sequential manner gives faster convergence than performing AFW on (W_1, W_2) jointly. The sequential approach is however more difficult for analysis, and is only considered as heuristics in our implementation.

Algorithm 4 Adapted Viterbi Algorithm for MSA

```
input : $G = -\nabla_{W_2}\mathcal{L}(.)$.
output : $\boldsymbol{v}_{FW}^{(2)}$ satisfying (18).
  $G_Q(q, q') = \sum_{t=((i,q),(i',q'))} [G(t)]_+,\ \forall (q, q') \in \mathcal{T}_Q$.
  $G_Q(q) = \sum_{t=((i,q),(i',q))} [G(t)]_+, , \forall q \in \mathcal{Q}$.
  $R(q_B) = 0$.
  for $q' \in \mathcal{Q} \setminus \{q_B\}$ in a topological order w.r.t. $\mathcal{T}_Q$ do
    $R(q') := G_Q(q') + \max_{(q,q') \in \mathcal{T}_Q}\ R(q) + G_Q(q, q')$.
    $S(q') := arg\max_{(q,q') \in \mathcal{T}_Q}\ R(q) + G_Q(q, q')$.
  end for
  Let $q_E \in arg\max_{q \in \mathcal{Q}_E}\ R(q)$.
  Find path $y$ by traceback from $q_E$ to $q_B \in \mathcal{Q}_B$ via $S(.)$.
  $\boldsymbol{v}_{FW} :=$ indicator vector of $y$ (in domain $\mathcal{M}_Q$).
```

3.2. Linear Minimization Oracle (LMO)

Alignment Subproblem (17) Firstly, we give an extension of Smith-Waterman algorithm for the LMO subproblem (17) for MSA formulation, as shown in Algorithm 3. It essentially performs dynamic programming following the topological ordering given by the transition set $\mathcal{T}_n$ of n-th sequence.

Note for MD problem, the set of possible transitions $\mathcal{T}_n$ is exponentially large due to consideration of all possible motifs $\mathcal{Y}$. However, to find the minimizer of LMO, we only need to consider a much smaller subset. Note the Augmented Lagrangian (12) has gradient of the form

$$G := -\nabla_{W_{1n}}\mathcal{L}(.) = -D_n - \rho P_n \tag{19}$$

where D_n gives same cost for different motifs of the same character at the same position, and $P_n = W_{1n} - W_{2n} + Y_n/\rho$ has non-zero entries only on motif y that ever in an alignment returned by the LMO. Then suppose there are k atoms ever returned by the LMO. We can perform a k-best version of Algorithm 3 with a transition set $\mathcal{T}_n$ of same size to MSA, which records the k-best solutions of each state s in $R(s)$ and $S(s)$ for computing the k-best solutions of other states s', so one can explicitly compare the k paths with non-zero entries in P_n with other paths at any state s.

Consensus Subproblem (18) The LMO subproblem (18), in the MSA case, has decomposition

$$\begin{aligned}
\max_{\boldsymbol{p} \in \mathcal{P}} \langle G, \boldsymbol{p} \rangle &= \max_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{p} \in \mathcal{P}_{\boldsymbol{z}}} \langle G, \boldsymbol{p} \rangle \\
&= \max_{\boldsymbol{z} \in \mathcal{Z}} \sum_{q:z(q)=1} G_Q(q) + \sum_{(q,q'):z(q,q')=1} G_Q(q, q')
\end{aligned}$$

where $\mathcal{P}_{\boldsymbol{z}}$ is the set of all $W \in \mathcal{M}$ supported only by consensus sequence $\boldsymbol{z}$, and $G_Q(q)$, $G(q, q')$ are defined at step 2, 3 of Algorithm 4. The second equality is given by the minimization w.r.t. $\boldsymbol{p} \in \mathcal{P}_{\boldsymbol{z}}$, which simply set $p(t) = 1$

for $G(t) > 0$ and $p(t) = 0$ otherwise. Then the maximization w.r.t. $\boldsymbol{z}$ can be done by an adapted Viterbi algorithm (Algorithm 4).

On the other hand, in MD, the constraint $Conv(G) := \{W_2 \in \mathcal{M}_{\mathbb{R}} \mid \Omega_{\mathcal{P}}(W_2) \leq K\}$ is the convex hull of atomic set $\{\sum_{k\in\mathcal{B}}^{K} \boldsymbol{p}_k \mid |\mathcal{B}| = K, \boldsymbol{p}_k \in \mathcal{P}_k\} \cup \{\boldsymbol{0}\}$, where $\boldsymbol{p}_k \in \mathcal{P}_k$ is any vector of single supporting motif $y_k \in \mathcal{Y}$. Therefore, the LMO simply finds K distinct motifs of highest score given by

$$\max_{\boldsymbol{p}_k \in \mathcal{P}_k} \langle G, \boldsymbol{p}_k \rangle = \sum_{q \in y_k} G_Q(q) + \sum_{(q,q') \in y_k} G_Q(q, q') \quad (20)$$

Note in MD, each state q belongs to a single motif, so we can compute (20) directly for each motif k of non-zero entries in G and pick top-K motifs of highest scores. Note since $G = -\nabla \mathcal{L}(.) = \rho(W_1 - W_2 + Y/\rho)$ and each LMO on subproblem (17) can only generate ℓ/L_m motifs of nonzero entries in G, the number of motifs of non-zero score is bounded by $2(\ell/L_m)t$, where ℓ is the total length of data sequences and t is the number of GDMM iterations.

3.3. Rounding Scheme

Note for any GDMM iteration t, W_2^t can be expressed as a convex combination of atoms based on the active atom set maintained by the AFW algorithm

$$W_2^t = \sum_{\boldsymbol{g} \in \mathcal{B}^t} \beta_{\boldsymbol{g}} \boldsymbol{g} \quad (21)$$

In MSA, each atom $\boldsymbol{g}$ belongs to a possible consensus sequence $P_{\boldsymbol{z}}$, and thus by randomly pick an atom $\boldsymbol{g}$ from active set $\mathcal{B}^t$ with probability $\beta_{\boldsymbol{g}}$, we can obtain an consensus sequence $\boldsymbol{z}$, and since pairwise alignment can be easily achieved via SW algorithm, we can construct an integer solution based on $\boldsymbol{z}$ that is guaranteed a feasible solution to the original MSA problem.

In MD, by applying the same sampling procedure on (21), we can obtain an atom $\boldsymbol{g}$ comprises K atoms $\boldsymbol{p}_1, ..., \boldsymbol{p}_K$ representing K distinct motifs. Therefore, we can recover an integer solution to the MD problem by simply finding the optimal alignment of data sequences to the K motifs. In practice, we can keep track of the quality given by the rounded solution during iterates, which can often stop program earlier with the same solution.

In both MSA and MD, it is possible to have multiple optimal integer solutions which lead to fractional solution of the convex relaxation. In such case, we add small perturbation to the matrix D to induce an unique solution.

4. Experiments

4.1. Multiple Sequence Alignment (MSA)

Following conventions of the bioinformatics community, we adopt *Sum-of-Pairs Score* and *Star Alignment Score* as the metrics to assess the quality of resulted alignments. Given a set of sequences g_1, ..., g_N in the format of *Global Multiple Sequence Alignment*, and a distance function $d(\cdot, \cdot)$ defining the penalty assigned to input pairs of acids, the Sum-of-Pairs score function S_{SP} can be expressed as the sum of all pairwise penalties over all acid columns (Durbin et al., 1998):

$$S_{SP} = \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j>i}^{N} d(g_{il}, g_{jl}) \quad (22)$$

where g_{il} denotes the acid at l-th column of the sequence g_i. And the Star Alignment Score function (S_{Star}) penalizes all applied operations along the way to figure out alignments. It can be formulated the same as the objective of the linear program (5).

In this work, both synthetic datasets and realistic datasets are experimented. To produce DNA sequences as a qualified synthetic dataset, we employ TKF1 evolutionary models (Thorne et al., 1991; 1992) to simulate the generation of insertion and deletion. The TKF1 model have three parameters: substitution rate α, insertion rate λ, and deletion rate β, with which every site of a sequence evolves independently. An illustrative example of the TKF1 model can be found at Fig.2. At any infinitesimal time interval, an acid at one position is either deleted with a Poisson rate β, or substituted with a Poisson rate α. Furthermore, an insertion occurs between two sites with a rate λ. In this work, we experiment synthetic datasets of three different levels of mutations: (I) Syn01 and Syn02: $\alpha = \beta = 0.005$, (II) Syn03: $\alpha = \beta = 0.010$, and (III) Syn04: $\alpha = \beta = 0.015$, where, in terms of TKF1 model, the insertion rate λ can be determined by $\frac{\beta L}{1+L}$ (L is the length of the ancestral sequence).

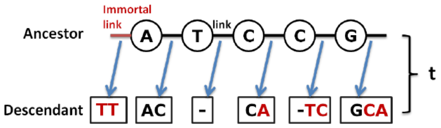

Figure 2. A TKF1 model: each link evolves independently.

Our experiments compare the ConvexMSA program with five broadly adopted solvers in the community of bioinformatics. These solvers are respectively *Clustal-Omega* (Larkin et al., 2007), *Kalign* (Lassmann & Sonnhammer, 2005), *T-COFFEE* (Notredame et al., 2000), *MAFFET*

(Katoh et al., 2002), and *MUSCLE* (Edgar, 2004). The method of *Multi-dimensional Dynamic Programming*, due to its exponential complexity, is not considered as one competing candidate.

From Table 1, it can be observed that our ConvexMSA solver succeeds to figure out the optimal alignment solution in both synthetic and real data sets and outperforms other solvers significantly. Another remarkable thing is that the ConvexMSA program recovers a solution of better score than the ground truth in cases with high mutation rates (e.g. Syn03 and two realistic datasets).

4.2. Motif Discovery (MD)

In this section, we experiment on a synthetic decipher data set that comprises 0-1 sequences generated by mapping each character of a sentence to a binary string of length between $[L_m, L_M]$. We use a well-known saying by Julius Caesar, *veni vidi vici* (*I came, I saw, I conquered*), as the source to produce the encoded text. We compare our MD algorithm with a well-known *Multiple EM for Motif Elicitation* (MEME) method (Bailey et al., 2006). MEME allows gaps between motifs but excludes insertion and deletion within a motif. The algorithm takes number of motifs K and range of length $[L_m, L_M]$ as inputs.

The following two frames give Motif Discovery result of MEME and ConvexMD. In either one, ConvexMD algorithm finds motifs with perfect matching, while MEME can only reach about 75% matching rate. On the first encoded case, ConvexMD even discovers an unexpected perfect-matching solution.

Text Encoded Case 1: $L_m = 4, L_M = 6$

010110000101001110001011111101011000100001000010
111111010110001000000110010

ConvexMD sol-1: matching rate 100.0%

$$\underbrace{010110}_{v}\ \underbrace{000101}_{e}\ \underbrace{001110}_{n}\ \underbrace{0010}_{i}\ \underbrace{111111}_{}\ \underbrace{010110}_{v}\ \underbrace{0010}_{i}\ \underbrace{000100}_{d}\ \underbrace{0010}_{i}$$

$$\underbrace{111111}_{}\ \underbrace{010110}_{v}\ \underbrace{0010}_{i}\ \underbrace{000011}_{c}\ \underbrace{0010}_{i}$$

ConvexMD sol-2: matching rate 100.0%

$$\underbrace{0101}_{1}\ \underbrace{1000}_{2}\ \underbrace{0101}_{1}\ \underbrace{0011}_{3}\ \underbrace{100010}_{4}\ \underbrace{111111}_{5}\ \underbrace{0101}_{1}\ \underbrace{1000}_{2}\ \underbrace{1000}_{2}\ \underbrace{0100}_{6}\ \underbrace{0010}_{7}$$

$$\underbrace{111111}_{5}\ \underbrace{0101}_{1}\ \underbrace{1000}_{2}\ \underbrace{1000}_{2}\ \underbrace{0011}_{3}\ \underbrace{0010}_{7}$$

MEME sol: matching rate 75.7%

$$\underbrace{0101}_{7}\ \underbrace{100001}_{3}\ 01\ \underbrace{0011}_{4}\ 1\ \underbrace{0001}_{6}\ 0\ \underbrace{111111}_{1}\ 0\ \underbrace{1011}_{2}\ 000\ \underbrace{100001}_{3}\ 0\ \underbrace{0001}_{6}\ 0$$

$$\underbrace{111111}_{1}\ 0\ \underbrace{1011}_{2}\ 0\ \underbrace{0010}_{5}\ 00\ \underbrace{0011}_{4}\ 0010$$

Text Encoded Case 2: $L_m = 5, L_M = 8$

01110110011001010110111001001111111110111011001001
01100100010011111111101110110010010110001101001

ConvexMD sol: matching rate 100.0%

$$\underbrace{01110110}_{v}\ \underbrace{01100101}_{e}\ \underbrace{01101110}_{n}\ \underbrace{01001}_{i}\ \underbrace{11111111}_{}\ \underbrace{01110110}_{v}\ \underbrace{01001}_{i}$$

$$\underbrace{01100100}_{d}\ \underbrace{01001}_{i}\ \underbrace{11111111}_{}\ \underbrace{01110110}_{v}\ \underbrace{01001}_{i}\ \underbrace{01100011}_{c}\ \underbrace{01001}_{i}$$

MEME sol: matching rate 72.2%

$$0111\ \underbrace{01100}_{3}\ 1\ \underbrace{10010}_{6}\ \underbrace{10110}_{6}\ 111\ \underbrace{00100}_{2}\ \underbrace{11111}_{7}\ 11\ \underbrace{11011}_{4}\ 1011\ \underbrace{00100}_{2}$$

$$1011\ \underbrace{00100}_{1}\ \underbrace{01001}_{5}\ \underbrace{11111}_{7}\ 1\ \underbrace{11011}_{4}\ 1011\ \underbrace{00100}_{1}\ 1\ \underbrace{01100}_{3}\ 011\ \underbrace{01001}_{5}$$

Acknowledgements We appreciate an anonymous meta reviewer for his/her detailed, constructive comments that lead to significant improvement of this paper. This research was supported by NSF grants CCF-1320746 and IIS-1546459, IIS-1149803, IIS-1320894, IIS-1447574, and DMS-1264033, ARO grants W911NF-12-1-0390, and NIH grant R01 GM117594-01 as part of the Joint DMS/NIGMS Initiative to Support Research at the Interface of the Biological and Mathematical Sciences.

References

Alayrac, Jean-Baptiste, Bojanowski, Piotr, Agrawal, Nishant, Sivic, Josef, Laptev, Ivan, and Lacoste-Julien, Simon. Learning from narrated instruction videos. *arXiv preprint arXiv:1506.09215*, 2015.

Bailey, Timothy L, Williams, Nadya, Misleh, Chris, and Li, Wilfred W. Meme: discovering and analyzing dna and protein sequence motifs. *Nucleic acids research*, 34 (suppl 2):W369–W373, 2006.

Das, Modan K and Dai, Ho-Kwok. A survey of dna motif finding algorithms. *BMC bioinformatics*, 8(Suppl 7): S21, 2007.

Durbin, Richard, Eddy, Sean R, Krogh, Anders, and Mitchison, Graeme. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

Edgar, Robert C. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797, 2004.

Elias, Isaac. *Settling the intractability of multiple alignment*. Springer, 2003.

Jaggi, Martin. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 427–435, 2013.

Karplus, Kevin and Hu, Birong. Evaluation of protein multiple alignments by sam-t99 using the balibase multiple alignment test set. *Bioinformatics*, 17(8):713–720, 2001.

Katoh, Kazutaka, Misawa, Kazuharu, Kuma, Kei-ichi, and Miyata, Takashi. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, 30(14):3059–3066, 2002.

Lacoste-Julien, Simon and Jaggi, Martin. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pp. 496–504, 2015.

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Larkin, Mark A, Blackshields, Gordon, Brown, NP, Chenna, R, McGettigan, Paul A, McWilliam, Hamish, Valentin, Franck, Wallace, Iain M, Wilm, Andreas, Lopez, Rodrigo, et al. Clustal w and clustal x version 2.0. *Bioinformatics*, 23(21):2947–2948, 2007.

Lassmann, Timo and Sonnhammer, Erik LL. Kalign–an accurate and fast multiple sequence alignment algorithm. *BMC bioinformatics*, 6(1):298, 2005.

Lawrence, Charles E, Altschul, Stephen F, Boguski, Mark S, Liu, Jun S, Neuwald, Andrew F, and Wootton, John C. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *science*, 262 (5131):208–214, 1993.

Notredame, Cédric. Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1):131–144, 2002.

Notredame, Cédric, Higgins, Desmond G, and Heringa, Jaap. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.

Siddharthan, Rahul, Siggia, Eric D, and Van Nimwegen, Erik. Phylogibbs: a gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput Biol*, 1(7):e67, 2005.

Smith, Temple F and Waterman, Michael S. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

Tewari, Ambuj, Ravikumar, Pradeep K, and Dhillon, Inderjit S. Greedy algorithms for structurally constrained high dimensional problems. In *Advances in Neural Information Processing Systems*, pp. 882–890, 2011.

Thorne, Jeffrey L, Kishino, Hirohisa, and Felsenstein, Joseph. An evolutionary model for maximum likelihood alignment of dna sequences. *Journal of Molecular Evolution*, 33(2):114–124, 1991.

Thorne, Jeffrey L, Kishino, Hirohisa, and Felsenstein, Joseph. Inching toward reality: an improved likelihood model of sequence evolution. *Journal of molecular evolution*, 34(1):3–16, 1992.

5. Appendix A: Proof for Theorem 2

Recall that the Augmented Lagrangian $\mathcal{L}(W_1, W_2, Y)$ is of the form

$$\langle D, W_1\rangle + \langle Y, W_1 - W_2\rangle + \frac{\rho}{2}\|W_1 - W_2\|^2.$$

Then let $X = [W_1; W_2]$ be the primal variables and denote

$$\mathcal{X}(Y) := \{X | X = arg\min_X \mathcal{L}(X, Y)\}$$

with

$$\bar{X}^t := \underset{\bar{X}\in\mathcal{X}(Y^t)}{argmin} \|\bar{X} - X^t\|,$$

and let

$$\mathbf{A}X = \left[\begin{array}{cc} I & -I \end{array}\right]\left[\begin{array}{c} W_1 \\ W_2 \end{array}\right] = W_1 - W_2 \tag{23}$$

and

$$\langle C, X\rangle = \left[\begin{array}{c} D \\ O \end{array}\right]^T \left[\begin{array}{c} W_1 \\ W_2 \end{array}\right] = \langle D, W_1\rangle \tag{24}$$

The Augmented Lagrangian can be re-written as

$$\mathcal{L}(X, Y) = \langle C, X\rangle + \langle Y, \mathbf{A}X\rangle + \frac{\rho}{2}\|\mathbf{A}X\|^2. \tag{25}$$

The dual function is

$$d(Y) = \min_{X\in Conv(\mathcal{A})\times Conv(\mathcal{G})} \mathcal{L}(X, Y)$$

and

$$d^* = \max_Y d(Y)$$

is the optimal dual function value. Then we measure the sub-optimality of iterates $\{(X^t, Y^t)\}_{t=1}^T$ given by GDMM in terms of dual function difference

$$\Delta_d^t = d^* - d(Y^t)$$

and the primal function difference for a given dual iterate Y^t:

$$\Delta_p^t = \mathcal{L}(X^{t+1}, Y^t) - d(Y^t)$$

yielded by X^{t+1} obtained from AFW steps. Then we have following lemma.

Lemma 1 (Dual Progress)**.** *Each iteration of GDMM (Algorithm 1) has*

$$\Delta_d^t - \Delta_d^{t-1} \leq -\eta(\mathbf{A}X^t)^T(\mathbf{A}\bar{X}^t). \tag{26}$$

Proof.

$$\begin{aligned}\Delta_d^t - \Delta_d^{t-1} &= (d^* - d(Y^t)) - (d^* - d(Y^{t-1})) \\ &= \mathcal{L}(\bar{X}^{t-1}, Y^{t-1}) - \mathcal{L}(\bar{X}^t, Y^t) \\ &\leq \mathcal{L}(\bar{X}^t, Y^{t-1}) - \mathcal{L}(\bar{X}^t, Y^t) \\ &= \langle Y^{t-1} - Y^t, \mathbf{A}\bar{X}^t\rangle \\ &= -\eta\langle \mathbf{A}X^t, \mathbf{A}\bar{X}^t\rangle\end{aligned}$$

where the first inequality follows from the optimality of $\bar{X}^{t-1}$ for the function $\mathcal{L}(X, Y^{t-1})$ defined by Y^{t-1}, and the last equality follows from the dual update in GDMM (14). □

On the other hand, the following lemma gives an expression on the primal progress that is independent of the algorithm used for minimizing Augmented Lagrangian

Lemma 2 (Primal Progress)**.** *Each iteration of GDMM (Algorithm 1) has*

$$\begin{aligned}\Delta_p^t - \Delta_p^{t-1} \leq &\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t) \\ &+ \eta\|\mathbf{A}X^t - \mathbf{A}\bar{X}^t\|^2 - \eta\langle \mathbf{A}X^t, \mathbf{A}\bar{X}^t\rangle\end{aligned}$$

Proof.

$$\begin{aligned}&\Delta_p^t - \Delta_p^{t-1} \\ =&\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^{t-1}) - (d(Y^t) - d(Y^{t-1})) \\ =&\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t) + \mathcal{L}(X^t, Y^t) - \mathcal{L}(X^t, Y^{t-1}) \\ &+ (d(Y^{t-1}) - d(Y^t)) \\ \leq&\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t) + \eta\|\mathbf{A}X^t\|^2 - \eta\langle \mathbf{A}X^t, \mathbf{A}\bar{X}^t\rangle\end{aligned}$$

where the last inequality uses Lemma 1 on $d(Y^{t-1}) - d(Y^t) = \Delta_d^t - \Delta_d^{t-1}$. □

By combining results of Lemma 1 and 2, we can obtain a joint progress of the form

$$\begin{aligned}&\Delta_d^t - \Delta_d^{t-1} + \Delta_p^t - \Delta_p^{t-1} \\ &\leq \mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t) + \eta\|\mathbf{A}X^t - \mathbf{A}\bar{X}^t\|^2 \\ &- \eta\|\mathbf{A}\bar{X}^t\|^2\end{aligned} \tag{27}$$

Note the only term that can be positive in (27) is the second. To guarantee descent of the joint progress, we bound the second term with the primal gap $\mathcal{L}(X^t, Y^t) - d(Y^t)$ given by the following lemma

Lemma 3.

$$\|\mathbf{A}X^t - \mathbf{A}\bar{X}^t\|^2 \leq \frac{2}{\rho}(\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t)) \tag{28}$$

Proof. Let

$$\tilde{\mathcal{L}}(X, Y) = h(X) + g(\mathbf{A}X),$$

where

$$g(\mathbf{A}X) = \frac{\rho}{2}\|\mathbf{A}X\|^2$$

and

$$h(X) = \langle C, X\rangle + \langle Y, \mathbf{A}X\rangle + \boldsymbol{I}_{X\in\mathcal{C}}$$

, where $\boldsymbol{I}_{X\in\mathcal{C}} = 0$ if $X \in \mathcal{C}$ and $\boldsymbol{I}_{X\in\mathcal{C}} = \infty$ otherwise, and

$$\mathcal{C} = \{(W_1, W_2) \mid W_1 \in Conv(\mathcal{A}), W_2 \in Conv(\mathcal{G})\}. \tag{29}$$

Note we have $\tilde{\mathcal{L}}(\bar{X}^t, Y^t) = \mathcal{L}(\bar{X}^t, Y^t)$, $\tilde{\mathcal{L}}(X^t, Y^t) = \mathcal{L}(X^t, Y^t)$ due to feasible iterates. According to the definition of $d(Y)$, we know that

$$0 \in \partial_X \tilde{\mathcal{L}}(\bar{X}^t, Y) = \partial h(\bar{X}^t) + \mathbf{A}^T \nabla g(\mathbf{A}(\bar{X}^t))$$

And by the convexity of $h(\cdot)$ and the strong convexity of $g(\cdot)$, we have

$$h(X^t) - h(\bar{X}^t) \geq \langle \partial h(\bar{X}^t), X^t - \bar{X}^t \rangle$$

and

$$\begin{aligned} &g(\mathbf{A}(X^t)) - g(\mathbf{A}(\bar{X}^t)) \\ \geq &\langle \mathbf{A}^T(\nabla g(\mathbf{A}(\bar{X}^t))), X^t - \bar{X}^t \rangle + \frac{\rho}{2}\|\mathbf{A}(X^t)) - \mathbf{A}(\bar{X}^t)\|^2 \end{aligned}$$

The the above two together implies

$$\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t) \geq \frac{\rho}{2}\|\mathbf{A}(X^t)) - \mathbf{A}(\bar{X}^t)\|^2$$

which leads to our conclusion. □

Then to guarantee significant descent of (27) relative to the current sub-optimality, we need to lower bound the magnitude of first term $\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t)$ and last term $-\eta\|\mathbf{A}\bar{X}^t\|^2$. Note by Danskins theorem, we have

$$\nabla d(Y^t) = \mathbf{A}\bar{X}^t$$

and we have the following lower bound on $\|\mathbf{A}\bar{X}^t\|$ by the concavity of $d(Y)$

$$\begin{aligned} d^* - d(Y^t) &\leq \langle \mathbf{A}\bar{X}^t, Y^{t*} - Y^t \rangle \\ &\leq \|\mathbf{A}\bar{X}^t\|\|Y^{t*} - Y^t\| \\ &\leq \|\mathbf{A}\bar{X}^t\| R_Y \end{aligned}$$

where Y^{t*} is the maximizer of $d(Y)$ that is closest to Y^t and R_Y is an upper bound on the distance (in ℓ_2 norm) of dual iterates $\{Y^t\}_{t=0}^T$ to its projection to the set of maximizer of $d(Y)$. Therefore, the progress (27) can be lower bounded as

$$\begin{aligned} &\Delta_d^t - \Delta_d^{t-1} + \Delta_p^t - \Delta_p^{t-1} \\ \leq &\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t) \\ &+ \frac{2\eta}{\rho}(\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t)) - \frac{\eta}{R_Y^2}\Delta_d^{t2} \end{aligned} \tag{30}$$

The remaining thing to do is show that one good step of Away-Step Frank-Wolfe iterate suffices to give descent amount $\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t)$ lower bounded by some constant multiple of the primal sub-optimality $\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t)$. Then by selecting GDMM step size η small enough, the RHS of (30) leads to a positive descent amount. Note this can be achieved by leveraging recent result from (Lacoste-Julien & Jaggi, 2015), who shows a linear-type convergence of AFW, even for non-strongly convex function of form (25). We thus provide the following lemma.

Lemma 4. *The AFW (Algorithm 2) performed on $X = (W_1, W_2)$ gives descent amount*

$$\begin{aligned} &\mathcal{L}(X^{t+1}, Y^t) - \mathcal{L}(X^t, Y^t) \\ &\leq -\frac{\kappa}{1+\kappa}(\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t)) \end{aligned} \tag{31}$$

where $\kappa := \mu_f/(8C_f^A)$, μ_f is the generalized geometric strong convexity constant for function $\mathcal{L}(.)$ in domain $\mathcal{C}$, and C_f^A is the corresponding smoothnesss constant.

Proof. Note the AL (25) is of the form

$$F(X) = \mathcal{L}(X, Y) = \langle C, X \rangle + f(\mathbf{A}X) \tag{32}$$

where $f(\mathbf{A}X) = \frac{\rho}{2}\|\mathbf{A}X + Y/\rho\|^2 + const.$ is a ρ-strongly convex function w.r.t. to $\mathcal{A}X$, and we are minimizing the function subject to a polyhedral domain $\mathcal{C}$ (defined at (29)). Therefore, by Theorem 10 of (Lacoste-Julien & Jaggi, 2015), we have the *generalized geometrical strong convexity* constant μ_f for function $\mathcal{L}(.)$ in domain $\mathcal{C}$ that has

$$\mu_f \geq \mu(PWidth(\mathcal{C}))^2 \tag{33}$$

where $PWidth(\mathcal{C}) > 0$ is the pyramidal width of polyhedron $\mathcal{C}$ and μ is the generalized strong convexity constant of function (32) defined in Lemma 9 of (Lacoste-Julien & Jaggi, 2015). By definition of the geometric strong convexity constant, we have

$$F(X) - F^* \leq \frac{g_X^2}{2\mu_f} \tag{34}$$

from (28) in (Lacoste-Julien & Jaggi, 2015), where $g_X = \langle \nabla F(X), \boldsymbol{v}_{FW}(X) - \boldsymbol{v}_A(X) \rangle$ for any FW atom $\boldsymbol{v}_{FW}(X)$ and away atom $\boldsymbol{v}_A(X)$ at point X. Note, since the convex polyhedron $\mathcal{C}$ is separable w.r.t. W_1, W_2, we have

$$\boldsymbol{v}_{FW}(X) = \begin{bmatrix} \boldsymbol{v}_{FW}^{(1)} \\ \boldsymbol{v}_{FW}^{(2)} \end{bmatrix} \tag{35}$$

and

$$\boldsymbol{v}_A(X) = \begin{bmatrix} \boldsymbol{v}_A^{(1)} \\ \boldsymbol{v}_A^{(2)} \end{bmatrix} \tag{36}$$

Then consider the progress given by a non-drop (”good”) step at iterate s of the AFW. We have

$$\begin{aligned} F(X^{s+1}) - F(X^s) &\leq -\frac{\gamma}{2} g_s + \frac{C_f^A}{2}\gamma^2 \\ &\leq -\frac{g_s^2}{16C_f^A} \\ &\leq -\frac{\mu_f(F(X^s) - F^*)}{8C_f^A} \end{aligned} \tag{37}$$

assuming $\gamma^* = g_s/(2C_f^A) < 1$, where $g_s = \langle -\nabla F, \boldsymbol{v}_{FW}(X^s) - \boldsymbol{v}_A(X^s)\rangle$, C_f^A is the curvature constant of $F(X)$ on domain $\mathcal{C}$ (eq. (26) in (Lacoste-Julien & Jaggi, 2015)). The first inequality follows from the fact that AFW chooses the smaller one between $\langle \nabla F, \boldsymbol{d}_{FW}\rangle$ and $\langle \nabla F, \boldsymbol{d}_A\rangle$ as the descent direction. The second inequality is given by minimizing RHS w.r.t. $\gamma \in [0, 1]$. And the third inequality is from (34). In case $\gamma^* = g_s/(2C_f^A) > 1$, we have $\gamma = 1$ and

$$\begin{aligned} F(X^{s+1}) - F(X^s) &\leq -\frac{\gamma}{2} g_s + \frac{C_f^A}{2}\gamma^2 \\ &\leq -g_s/4 \leq -(F(X^s) - F^*)/4 \\ &\leq -\frac{\mu_f(F(X^s) - F^*)}{8C_f^A} \end{aligned} \tag{38}$$

which leads to the same result.

Then let $\kappa = \mu_f/(8C_f^A)$. We have

$$\begin{aligned} F(X^{t+1}) - F(X^t) &\leq F(X^{s+1}) - F(X^s) \\ &\leq -\kappa(F(X^s) - F^*) \\ &\leq -\kappa(F(X^{t+1}) - F^*) \end{aligned}$$

where the first inequality is due to $F(X^t) \geq F(X^s)$ (since AFW is a descent algorithm). Through rearrangement we have

$$F(X^{t+1}) - F^* \leq \frac{1}{1+\kappa}(F(X^t) - F^*)$$

which then leads to the conclusion. □

Now we provide proof of the main theorem 2 as follows.

Proof. By lemma 4 and (30), we have

$$\begin{aligned} &\Delta_d^t - \Delta_d^{t-1} + \Delta_p^t - \Delta_p^{t-1} \\ \leq &\frac{-\kappa}{1+\kappa}\left(\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t)\right) \\ &+ \frac{2\eta}{\rho}(\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t)) - \frac{\eta}{R_Y^2}\Delta_d^t. \end{aligned} \tag{39}$$

Then by choosing $\eta < \frac{\kappa\rho}{2(1+\kappa)}$, we have guaranteed descent on $\Delta_p + \Delta_d$ for each GDMM iteration. By choosing $\eta \leq \frac{\kappa\rho}{4(1+\kappa)}$, we have

$$\begin{aligned} &(\Delta_d^t + \Delta_p^t) - (\Delta_d^{t-1} + \Delta_p^{t-1}) \\ \leq &\frac{-\kappa}{2(1+\kappa)}\left(\mathcal{L}(X^t, Y^t) - \mathcal{L}(\bar{X}^t, Y^t)\right) - \frac{\eta}{R_Y^2}\Delta_d^{t2} \\ \leq &\frac{-\kappa}{2(1+\kappa)}\Delta_p^t - \frac{\kappa\rho}{4(1+\kappa)R_Y^2}\Delta_d^{t2} \\ \leq &\frac{-\kappa}{2(1+\kappa)(\Delta_p^0 + \Delta_d^0)}\Delta_p^{t2} - \frac{\kappa\rho}{4(1+\kappa)R_Y^2}\Delta_d^{t2} \\ \leq &-\left(\frac{\kappa}{4(1+\kappa)}\min(\frac{1}{\Delta_p^0 + \Delta_d^0}, \frac{\rho}{2R_Y^2})\right)(\Delta_p^t + \Delta_d^t)^2 \end{aligned}$$

where the third inequality is by non-increasing of $\{\Delta_p^t + \Delta_d^t\}_{t=1}^{\infty}$. Then recursion of the form $\Delta^t - \Delta^{t-1} \leq c\Delta^{t2}$ leads to the conclusion. □

Table 1. Comparisons of Sum-of-Pair Score and Star Alignment Score between MSA Solvers. Each dataset is characterized by the number of sequences N, the length L of its ancestral sequence, and one mutation triplet. The mutation triplet (I, D, M) denotes the number of insertions I, deletions D, and substitutions M respectively. Score of any match event is 0 whereas any non-match event is 1.

Settings	Synthetic Datasets				Realistic Datasets	
Data Solvers	Syn01 N=10, L=30 (3, 2, 4)	Syn02 N=30, L=50 (12, 11, 7)	Syn03 N=30, L=50 (19, 18, 9)	Syn04 N=30, L=50 (24, 24, 19)	sDicF N=6, L=15 (3, 4, 16)	sHairpin N=20, L=30 (9, 7, 44)
ClustalOmega	311 / 47	3295 / 126	6671 / 274	5946 / 240	119 / 27	1225 / 77
Kalign	88 / 10	1440 / 51	2003 / 71	2612 / 93	104 / 24	874 / 54
T-COFFEE	99 / 12	1031 / 36	1492 / 53	2120 / 75	104 / 24	868 / 53
MAFFET	87 / 10	1196 / 42	1856 / 66	2843 / 103	103 / 27	874 / 54
MUSCLE	87 / 10	1060 / 37	1649 / 59	2311 / 83	105 / 24	874 / 54
ConvexMSA	**79 / 9**	**863 / 30**	**1285 / 45**	**1903 / 67**	**98 / 23**	**853 / 50**
Ground Truth	*79 / 9*	*863 / 30*	*1310 / 46*	*1903 / 67*	*103 / 23*	*974 / 60*