Multiple Partial Order Alignment as a Graph Problem[*]

D. Stott Parker[†] Christopher J. Lee[‡]

September 18, 2003

Abstract

Multiple Sequence Alignment (MSA) is a fundamental tool of bioinformatics. Row-Column MSA (RC-MSA) methods such as CLUSTALW [12] produce tabular alignments that are now familiar. However, these methods have a number of shortcomings, including difficulty of understanding the result, high computational complexity, questionable assumptions, and other artifacts (such as poor handling of prefix- and suffix-alignment).

Partial Order Alignment was proposed recently as an alternative approach to MSA. Partial Order MSA (PO-MSA) methods produce a partial order — a labeled directed acyclic graph — that includes the input sequences as subgraphs. The approaches differ in their strengths and weaknesses as well as their assumptions.

In this paper, we formalize PO-MSA as a graph problem, show that it corresponds to finding a Minimal Common Supergraph for a set of partial order graphs, and characterize how such a supergraph can be derived. This formalization offers some perspective on MSA generally, and also on particular tradeoffs between RC-MSA and PO-MSA.

1 Introduction

Multiple Sequence Alignment (MSA) is a basic tool of bioinformatics, with a history of development spanning several decades (e.g., [7, 11]). Given a set of input sequences, a Row-Column MSA (RC-MSA) algorithm derives a table summarizing their similarities and differences from a consensus sequence. A sample RC-MSA table is shown in Figure 1. Though widely used, traditional RC-MSA algorithms have a number of shortcomings, including the following list reviewed in [14]:

complexity Classical dynamic programming algorithms for MSA do not scale well. Although pairwise alignment algorithms have acceptable execution time of $O(L^2)$ where L is the length of the sequences, alignment of N sequences requires time $O(L^N)$. For even modest values of N this becomes excessive: even the small example in Figure 1 is intractible, as it has $N = 23$ sequences with $L = 50$.

assumptions Generally, ‘aligning a set of sequences’ has meant something like ‘finding minimal edit distances of the sequences from some consensus sequence’, where the consensus sequence may or may not be also an input. MSA and its tabular format rest on the assumption that there is a consensus sequence.

artifacts Both leading (prefix) and trailing (suffix) differences among sequences are kept in the alignment, and affect scoring. However, these differences may not be meaningful in the alignment. As another example, the ordering of the input sequences can determine the quality of the alignment result.

Partial Order Alignment (POA) [9, 13, 14, 16, 18] is a recently-developed alternative to traditional sequence alignment methods [7, 11]. Rather than a tabular arrangement of sequences, alignments with POA consists of a partial order (labeled directed acyclic graph) that subsumes the sequences, as shown in Figure 2.

Recently an independent evaluation by Lassman and Sonnhammer [15] reported the POA approach to have some advantages over traditional alignment methods. Several examples of applications of POA (explained elsewhere [14]) are reproduced in Figures 1–4. The diversity of applications suggests that partial order structures are good models for many aspects of biological sequences that are not as well captured by tabular arrangements.

POA is of interest in its own right, since it gives a global, graph-like perspective that can be used in concert with the local perspectives afforded by dynamic programming when attacking alignment problems. This paper attempts

[*] Work supported by NSF grant IIS 0082964. Project home page: `http://www.bioinformatics.ucla.edu/leelab/db`

[†] Computer Science Department, University of California, Los Angeles, California 90095-1596, `stott@cs.ucla.edu`

[‡] Departments of Chemistry & Biochemistry, University of California, Los Angeles, California 90095-1569, `leec@mbi.ucla.edu`

```
CONSENSUS      a.gttcctgc.tgcgtttgctggactgatgtactt.gtttgtgagg.caa
Hs#S663801     a.gttcctgc.tgcgtttgctggacttatgtactt.gtttgtgagg.caa
Hs#S337687     aagttcctgc.tgcgtttgctggactgatgtacttggtttgtgnaggcaa
Hs#S629177     a.gttcctgc.tgcgtttgctggactgatgtactt.gtttgtnagg.caa
Hs#S672957     a.gttcctgc.tgcgtttgct.............................
Hs#S672182     a.gttcctgc.tgcgtttgctggactgatgtactt.gttt..........
Hs#S674099     a.gttcctgc.tgcgtttgctggactgatgtactt.gtttgtgagg.caa
Hs#S196113     a.gttnctgn.tgngtttgctggactgatgtactt.gtttgtgagg.caa
Hs#S994400     ............................gtacnt.gtttgtgagg.cta
Hs#S80460      a.gttcctgc.tgcgtttgctggactgatgtactt.gtttgtgagg.caa
Hs#S1988018    a.gttcctgc.tgcttttgctggactgatgtactt.gattgtgagg.caa
Hs#S1794113    a.gttcctgc.tgcgcttgctggactgatgtactt.gtttgtgagg.caa
Hs#S4698       a.gttcctgc.tgcgtttgctggactgatgtactt.gtttgtgcgg.caa
Hs#S813765     a.gt.cctgc.g.cgtttgc.ggacggatgtactt.gtt.gtgagg.caa
Hs#S1184845    ...........................................g.caa
Hs#S1577463    ..........................................gg.caa
Hs#S914987     .......................ctgatgtactt.gtt.gtgagggcaa
Hs#S1985364    a.gttcctgc.tgcgtttgctggactgatgtactt.gtttgtgagg.caa
Hs#S1465644    ..gttc.tgcctgcgtttgctgaactgatgtactt.gttagt.aag.caa
Hs#S1850471    c.gttactgc.ggggtttgctggactcatg.actttgttngt.agg.caa
```

Figure 1: *RC-MSA for EST sequences from UniGene cluster Hs.100194*

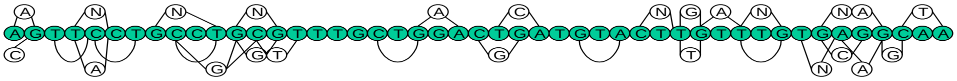

Figure 2: *POA summarizing the sequences shown in Figure 1*

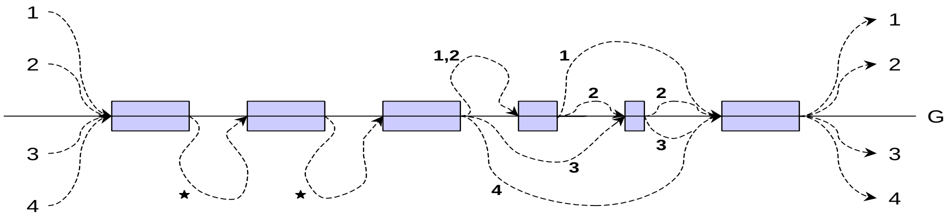

Figure 3: *Partial Order Alignment showing four different observed alternatively spliced mRNA forms, and the genomic sequence, of human gene HLA-DM β from Unigene cluster Hs.1162. Exons are shown as rectangles; asterisks indicate that all four forms share an indicated sequence.*

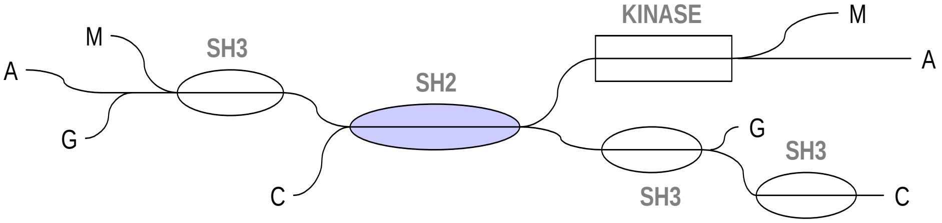

Figure 4: *POA showing recombination among the 4 multidomain protein sequences with Swissprot identifiers MATK_HUMAN (*M*), ABL1_HUMAN (*A*), GRB2_HUMAN (*G*), and CRK1_HUMAN (*C*); protein domains are shown as ovals and rectangles.*

to formalize what POA is, in graph-theoretic terms; frame computational issues for POA; and set a foundation for implementations. This paper is specifically focused on *multiple sequence alignment* with POA. A companion paper [20] analyzes the special case of pairwise alignment.

We begin by considering a view of alignment as a 'fusion' process that integrates sequences into a model (here: a partial order graph). The resulting intuition about fusion is then formalized as POA in graph-theoretic terms. We show the pairwise POA problem to be NP-complete. As a result (or nevertheless), we then develop a generic algorithm for POA that works by enumerating sets of possible fusions, or equivalently certain subgraphs of a derived *fusion graph*. We conclude by considering applications, and avenues for future work.

1.1 Alignment as Graph Fusion

In what follows sequences are viewed as graphs in which nodes are labeled with the letters of the sequences, as shown in Figure 6.

When sequences are represented as graphs, *alignment* of sequences corresponds naturally to *fusion* of their nodes. That is, nodes (letters) that are aligned can be integrated into a single node. Figure 6 shows a simple example, in which 7 fusions are made (for the subsequences of nodes corresponding to the letters KMVRNET).

This fusion process has the effect of integrating multiple sequence emission models into a single model. Usually there is more than one possible result. The shape of the resulting graph depends on the specific fusions that are permitted (e.g., we are permitted to fuse nodes having identical letters), and on the specific choices of fusions made. See Figure 5.

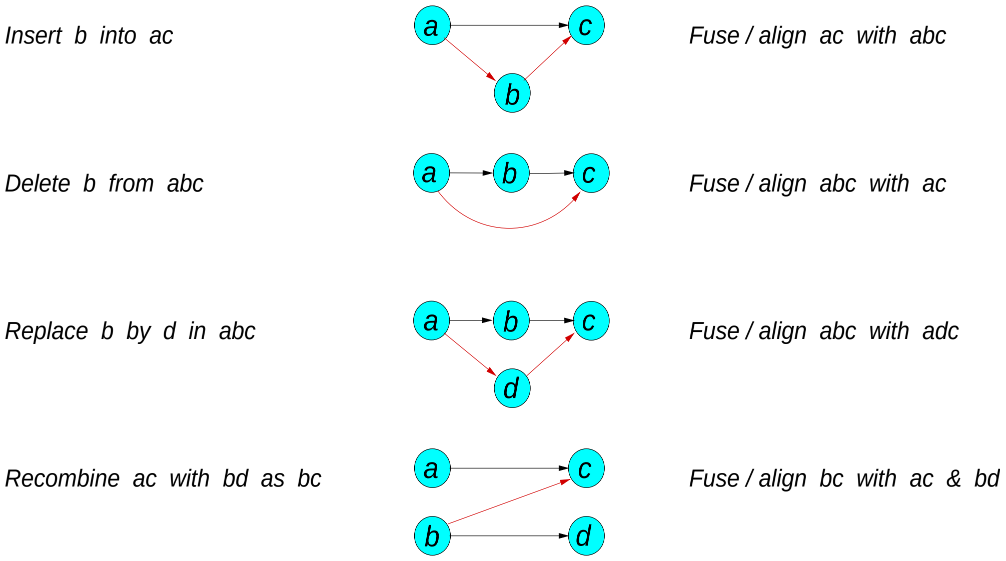

Figure 5: *Sequence edit operations differ from sequence fusion/alignment operations in several ways. Edit operators possess a notion of 'history' that fusion operators lack. Edit operators also produce sequences, while fusion operators yield sequence emission models. Traditionally, edit operators also do not cover the homologous recombination operator above, which is a natural aspect of fusion/alignment.*

Generally, as a Parsimony Principle, we seek a *minimal model* — a model that is as concise as possible. Intuitively, for example, the final graph in Figure 6 is a minimal model for the two sequences shown. To obtain a minimal model, we seek a set of fusions that minimizes the size of the resulting graph.

Definition 1 *Each node and edge in a graph has a* **size***: an associated numeric value. The* **size of a graph** *is the sum of the sizes of its nodes and the sizes of its edges.*

This definition is general enough to cover many notions of size. It can handle statistical notions such as log-odds measures. It can also handle graph-theoretic notions such as *node+edge graph size*, discussed later, which assigns high weight to the number of nodes and lower weight to the number of edges.

Definition 2 *This are two kinds of node fusion:*

- **identity fusion***: fusion of identically-labeled nodes.*
- **substitution fusion***: fusion of differently-labeled nodes.*

The node resulting from the fusion is labeled with a set containing all labels from the fused nodes. This set is called an **align ring** *[9].*

For example, in Figure 6, only identity fusion is shown. The two final nodes labeled with the letters I and V could be fused into a single node labeled with align ring $\{$V,I$\}$. Labeling nodes with sets of letters differs in a minor way from the initial POA paper [14], which used dashed circles around nodes to denote align rings.

In this paper, alignment is fusion. Furthermore fusion is limited to a predefined set of *possible fusions*, from which we seek optimal subsets.

Definition 3 *A* **set of possible fusions** *PF for labeled graphs $G_1 = (V_1, E_1, \ell_1)$ and $G_2 = (V_2, E_2, \ell_2)$ is a subset PF $\subseteq V_1 \times V_2$ of the cartesian product of their nodes.*

Some natural sets of possible fusions include the set of all identity fusions, the set of fusions for pairs of letters for which the probability of point mutation is high (e.g., as measured in a PAM/BLOSUM matrix [7]), or the set of fusions for nodes whose codon triplets differ by at most a single base change (such as L ↔ I, L ↔ V, etc. [7]).

$$\begin{array}{lcl} PF & = & \{\, (u,v) \in V_1 \times V_2 \mid \ell_1(u) = \ell_2(v), \text{ i.e., } u \text{ and } v \text{ have identical labels} \,\} \\ PF & = & \{\, (u,v) \in V_1 \times V_2 \mid \text{the } [\ell_1(u), \ell_2(v)] \text{ entry in the BLOSUM62 matrix is positive} \,\} \\ PF & = & \{\, (u,v) \in V_1 \times V_2 \mid \text{codons for the amino acid labels of } u \text{ and } v \text{ differ by at most a single base change} \,\}. \end{array}$$

This paper focuses on problems in which the set of possible fusions *PF* has nontrivial structure, such as when the set of labels (alphabet) is large and substitutions are held to a high standard. In trivial problems where *PF* $= V_1 \times V_2$ the alignment problem becomes less a graph problem, emphasizing the model structure we are interested in, and more a numerical optimization problem. The purpose of this paper is to investigate basic properties of a graph-theoretic formulation.

2 Partial Order Alignment

This section shows how POA arises naturally when emission models are represented as graphs. In this situation, combining models (*aligning models*) corresponds to combining graphs (*fusing graphs*). We clarify how the graphs of PO-MSA and tables of RC-MSA are inequivalent for representing alignments.

2.1 Partial Order Alignment as a graph problem

Any alignment that we obtain by fusing nodes in a set of graphs will ultimately give us a *supergraph* of these graphs: a single graph that contains each as a subgraph. This suggests the following definition.

Definition 4 *For this paper, we define:*

- *A* **PO (Partial Order)** *is a directed acyclic graph whose nodes are labeled (with letters). The PO for a sequence of letters is called a* **linear graph***.*
- *A* **PO-MSA (Partial Order Multiple Sequence Alignment)** *for a set of sequences S is a PO G such that every sequence in S is a linear subgraph of G.*
- *The* **Multiple PO Alignment Problem***:*

  Given an input set $S = \{G_1, \ldots, G_N\}$ of POs, find a **minimal common supergraph** *G for all of the graphs in S. Specifically, find G such that:*

  - *G is a PO that includes each input $G_i \in S$ as a subgraph;*
  - *G is* **minimal** *(among all such supergraphs, it has minimal size).*

.	.	P	K	M	I	V	R	P	Q	K	N	E	T	V	.
T	H	.	K	M	L	V	R	.	.	.	N	E	T	I	M

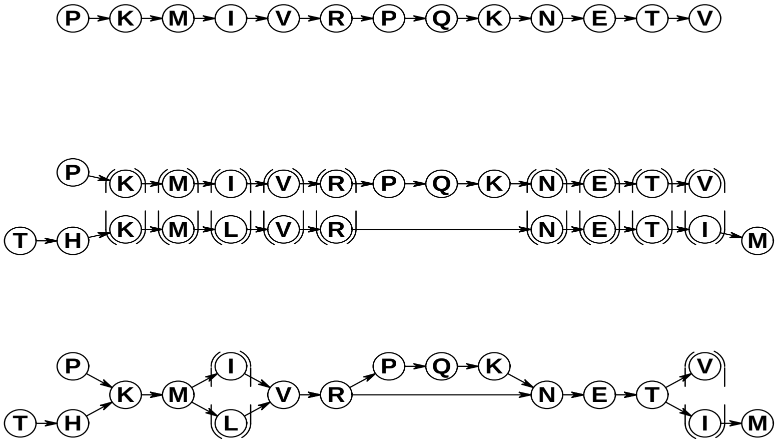
P K M I V R P Q K N E T V
P K M I V R P Q K N E T V
T H K M L V R N E T I M
P T H K M I L V R P Q K N E T V I M

Figure 6: *Partial Order Alignment for the two input sequences* PKMIVRPQKNETV, THKMLVRNETIM. *The first line gives a tabular RC-MSA for these sequences. The second line is a linear graph — the partial order (labeled directed acyclic graph) corresponding to the sequence* PKMIVRPQKNETV. *The third line (ignoring the dashed ovals) includes* only *the input sequences as paths, while the fourth line is a partial order that includes many other sequences as paths. However, if nodes can contain only one letter, the fourth line has the minimal number of nodes among all partial orders that include the input sequences as paths, while the third has the maximal number (the sum of all input sequence lengths). Each of the graphs shown can be viewed as an* emission model *— a HMM- or grammar-like generator of sequences. If the size of an emission model is its number of nodes, and each node can contain only a single letter, then the final graph shown is the minimal model for the two sequences.*

Example 1 *Each partial order alignment in Figure 6 and Figure 2 is also a PO. Also, if the size of a graph is proportional to its number of nodes, so that minimality of a graph means it has as few nodes as possible, the fourth line of Figure 6 shows the unique solution of the Multiple PO Alignment problem for the sequences* PKMIVRPQKNETV *and* THKMLVRNETIM.

A companion paper [20] studies the special case $N = 2$ of *pairwise* alignments of partial order graphs.

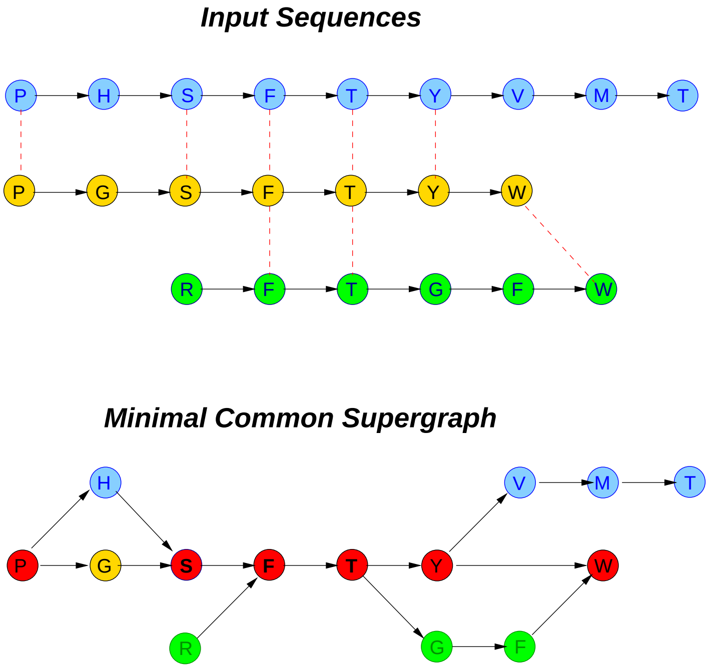

Figure 7: *Three linear graphs for the sequences* PHSFTYVMT, PGSFTYW, RFTGFW, *and a common supergraph. Dashed edges among the sequences identify the fusions used to obtain this supergraph; for clarity in this example only nodes with identical labels are fused. Under the node+edge graph cost, this is their Minimal Common Supergraph.*

3 Algorithmic Perspectives on PO-MSA: finding an optimal set of fusions

A PO-MSA implementation has been available for several years at `www.bioinformatics.ucla.edu/poa`. As described in [14], it uses a dynamic programming algorithm generalizing directly on classical sequence alignment algorithms, to optimally fuse a sequence with an existing PO. Using this it implements MSA by assimilating input sequences one by one into a PO-MSA. The resulting PO-MSA is therefore not globally optimal, but the approach has proven surprisingly effective in a variety of large-scale alignment problems. A recent refinement that gives still better results performs optimal linear fusion of a PO with a PO and progressive MSA, as described in [9].

The foregoing suggests PO-MSA can be implemented as a search for an optimal set of fusions on the set of input sequences. Although it is difficult to see what kinds of graph algorithms are needed, it is easy to see that brute-force approaches — such as enumerating all possible candidate sets of fusions, or all possible models — will be too slow to work in practice, given the size of the search space.

In this section we show that the set of possible fusions can be represented neatly in a graph, which we call the *fusion graph*. Compatible sets of fusions correspond directly to acyclic subgraphs in this graph. The fusion graph can therefore significantly limit the search for good sets of fusions, and be a basis for PO-MSA algorithms.

3.1 Some intuition about optimal sets of fusions

As just formalized, a PO-MSA for a set of sequences is a supergraph of the linear graphs corresponding to these sequences, and an optimal PO-MSA is a supergraph of minimal size. The supergraph of a set of labeled graphs $G_1, \ldots, G_N$ can be obtained by fusing pairs of nodes among these graphs.

Definition 5 *The* **PO Fusion Problem***:*
Given a PO G and a set of possible fusions PF, find a subset $F \subseteq$ PF of fusions that, when applied to G, minimize the size of the resulting graph.

An important insight here is that the PO-MSA problem for a set of graphs is equivalent to the PO Fusion problem for their union (i.e., the PO-MSA problem for disjoint graphs $G_1, \ldots, G_N$ is equivalent to the PO Fusion problem for $G = G_1 \cup \cdots \cup G_N$). This insight greatly simplifies the kinds of algorithms needed for PO-MSA. It also generalizes the approach for pairwise alignment recently developed in [20], since it allows fusions of pairs of nodes within a single graph, rather than between different graphs.

With this change from a problem involving N graphs $G_1 = (V_1, E_1)$, ..., $G_N = (V_N, E_N)$ to a problem involving 1 graph $G = (V, E)$, there is only one set of nodes $V = \cup_{i=1}^{N} V_i$ to be concerned about. Thus possible fusions (u, v) in the set *PF* are perhaps best viewed as sets $(\{u, v\})$ of size 2 — the ordering of u and v loses significance. Since it simplifies some of the analysis, in what follows we view *PF* this way.

Two other insights about fusions are useful:

- Only *compatible* sets of fusions — fusions that create no cycles — yield a partial order.
- Roughly speaking, in order to obtain a minimal supergraph, one must perform as many fusions as possible.

These insights form the basis of an algorithm for finding a minimal supergraph (and finding an optimal PO-MSA): enumerate all large compatible sets of fusions, and find a set that minimizes size.

3.2 Fusion graphs: graphical representations of compatible sets of fusions

We can define a *fusion graph* that is useful for identifying compatible sets of fusions.

Definition 6 *The* **fusion graph of** G **defined by the possible fusions** *PF is a directed graph $FG = (PF, CF)$. The edges CF specify the compatible fusions, defined as follows.*

There is a directed **precedence edge** *in CF from $(\{u, v\})$ to $(\{u', v'\})$ if these are distinct nodes in PF, and fusing u with v, and u' with v', yields a graph in which there is a path from (either u or v) to (either u' or v').*

There is a bidirected **equivalence edge** *in CF between $(\{u, v\})$ and $(\{u', v'\})$ if these pairs intersect (i.e., $\{u, v\} \cap \{u', v'\} \neq \emptyset$). Equivalence edges between $(\{u, v\})$ and $(\{u', v'\})$ in CF reflect the fact that if we fuse u with v, and u' with v', then — since these pairs share a node $x = \{u, v\} \cap \{u', v'\}$ — the result is a* three-way fusion *that renders u, v, u', v' all* equivalent*.*

A **fusion precedence cycle** *in a fusion graph is a sequence of edges $w_1 \to w_2 \to \cdots \to w_n \to w_1$ where each node $w_i = (\{u_i, v_i\})$ denotes a possible fusion in PF (of nodes u_i and v_i in G), each edge is in CF, and not all edges are equivalence edges (there is at least one precedence edge).*

An **incompatible set of fusions** *is any set of fusions that includes a subset $\{(\{u_1, v_1\}), \ldots, (\{u_n, v_n\})\}$ defining a fusion precedence cycle. Otherwise the fusions are* **compatible***.*

It is possible for there to be a fusion precedence cycle involving only the nodes $(\{u, v\})$ and $(\{u', v'\})$. Two such nodes are called **pairwise incompatible fusions***; otherwise they are* **pairwise compatible***.*

Theorem 1 *Performing a set of fusions $F \subseteq PF$ on the PO G yields a cycle in the resulting supergraph if and only if F defines a corresponding precedence cycle in the fusion graph of G defined by PF.*

We prove this by induction on the number n of nodes in F. The basis $n = 1$ is trivial, since no single fusion can yield a cycle. For the induction step, assume $F = \{(\{u_1, v_1\}), \ldots, (\{u_n, v_n\})\}$ with $n > 1$.

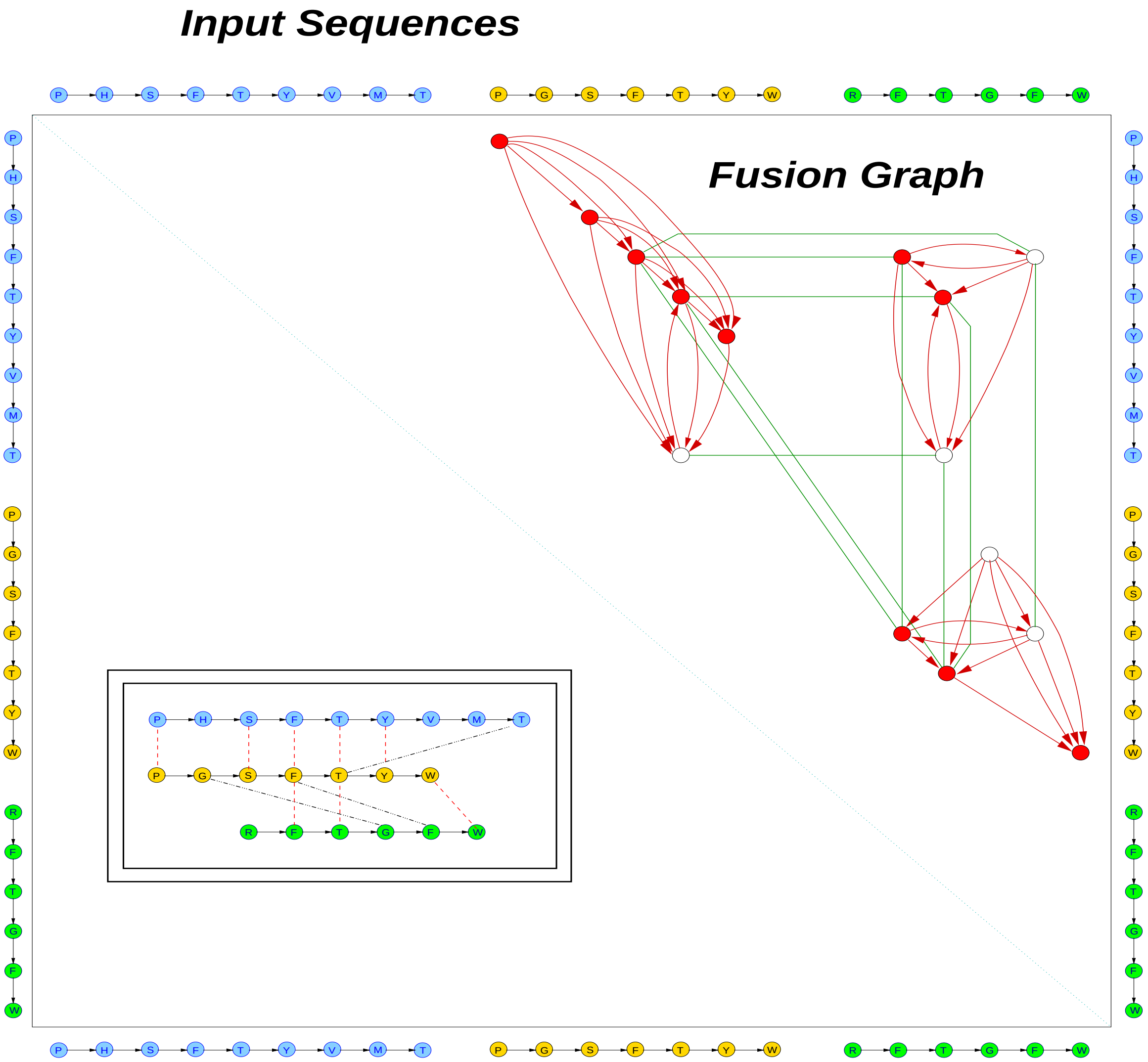

Figure 8: *The fusion graph for the graphs shown in Figure 7, which correspond to the sequences* PHSFTYVMT, PGSFTYW, RFTGFW. *All nodes in the fusion graph are* possible fusions, *and as mentioned in Figure 7 these have been restricted for clarity to node pairs* $\{u, v\}$ *that have identical letters. These node pairs are represented as points inside the grid shown above, at coordinates* (x, y) *that give the positions of nodes* u *and* v *among the sequences. By ordering* u *and* v *so that* $x < y$*, the fusion graph can be limited to the upper-triangular part of the grid. Undirected (green) edges shown are* equivalence edges*; directed (red) edges are* preference edges. *Highlighted nodes identify a maximal set of compatible fusions; performing any further fusions (non-highlighted nodes) would make the graph resulting from these fusions cyclic. Pairwise incompatible nodes are connected by a cycle of two preference edges. All 15 fusion possible fusion nodes are shown, but for clarity some edges have been omitted. For this graph, the maximal set of compatible fusions is unique, and produce a unique minimal common supergraph (for node+edge graph size). The grid layout used in constructing the fusion graph here can be used for any number of input sequences (linear graphs), and in fact also generalizes for the case where the inputs are partial order graphs — see [20].*

(⇒) Suppose the fusions of F produce a cycle in G. This cycle must include fused nodes in some order, since without the fused nodes there is no cycle. We can assume no proper subset of F has this property, since otherwise the rest follows by induction. Thus we can assume the cycle is of the form

$$w_1 \xrightarrow{*} w_2 \xrightarrow{*} \cdots \xrightarrow{*} w_n \xrightarrow{*} w_1$$

where each node w_i denotes the fusion of two nodes u_i and v_i in G, and each arrow denotes a path that is in G. The fusion graph definition requires that, for each path between w_i and w_{i+1}, there must be a corresponding edge $((\{u_i, v_i\}) \to (\{u_{i+1}, v_{i+1}\}))$ in *FG*. So F defines a cyclic subgraph of the fusion graph *FG*.

It remains to show this cycle contains a precedence edge. Any edge $((\{u_i, v_i\}) \to (\{u_{i+1}, v_{i+1}\}))$ is an equivalence edge if, and only if, $\{u_i, v_i\} \cap \{u_{i+1}, v_{i+1}\}$ is nonempty. If all edges in the cycle are equivalence edges, then the result of performing the fusions $F = \{(\{u_1, v_1\}), \ldots, (\{u_n, v_n\})\}$ will be a single node. This is not a cycle, as was assumed. So at least one edge in the cycle must be a precedence edge, and it is a fusion precedence cycle.

(⇐) Suppose that F defines a fusion precedence cycle in *FG*. We can assume no proper subset of F does also, since otherwise the rest follows by induction. Thus, we can assume (by renumbering if necessary) that *FG* contains the edges

$$(\{u_1, v_1\} \to \{u_2, v_2\}) \, , \ \cdots , \ (\{u_{n-1}, v_{n-1}\} \to \{u_n, v_n\}) \, , \ (\{u_n, v_n\} \to \{u_1, v_1\}) \, ,$$

and at least one of these edges is a precedence edge. By the fusion graph definition, the i-th edge corresponds to a path in G from either u_i or v_i to either u_{i+1} or v_{i+1}. With the fusions of F, these paths define a cycle in G.

3.3 Implementing Partial Order Alignment as search on a Fusion Graph

Theorem 1 shows that implementation of POA requires finding good sets of node fusions. Specifically, the fusions must be *compatible* — i.e., performing them does not create a cycle, and destroy the partial ordering property. (Some time ago by Morgenstern et al. [19] mentioned a similar idea of using partial order as a consistency check for proposed MSAs.) Beyond this, the fusions should be as extensive (size-reducing) as possible, so as to yield a minimal supergraph.

If a set of fusions F is compatible, then all of its elements must be pairwise compatible. Construct a graph *FG*′ that has the same nodes *PF* as the fusion graph *FG*, but has (undirected) edges between pairwise compatible fusions. If F is a compatible set of fusions, then, F will be a clique of *FG*′.[1] So, we can implement POA by finding large cliques F of *FG*′, checking that they are also compatible subsets of *FG*, and if they are, then determining the size of the supergraph that results by performing all the fusions in F. Any smallest such supergraph we obtain is a PO-MSA.

There are many algorithms for finding maximal cliques [5]. A popular choice has been the *Bron-Kerbosch clique-finding algorithm* [3], a clique enumeration method that includes a branch-and-bound technique for avoiding rediscovery of cliques and heuristics to improve search performance, and has the virtue that it can be modified to return 'all reasonable cliques'. In order to implement POA, we implemented an extension of the Bron-Kerbosch algorithm that finds maximal fusion sets. Our program consists of about 2500 lines of C, of which 200 lines is the Bron-Kerbosch algorithm, 200 lines is acyclicity checking, 650 lines is Minimal Common Supergraph construction, and most of the remainder is concerned with building, manipulating, and reading and printing of graphs. The POA output from the program is rendered as a graph, using a layout obtained from the *dot* program in the *graphviz* package available from AT&T Research [8]. A simple PO-MSA example is shown in Figure 9.

The Bron-Kerbosch algorithm has been used to find common subgraphs and supergraphs in bioinformatics applications in the past. For example, for over a decade it has been used in comparing molecular structure, particularly protein structure alignment [10, 23], and protein threading, which involves the alignment of protein sequences with protein structures [17]. VAST, the NCBI Vector Alignment Search Tool [21] adopted this approach, and it is used in MMDB [6] to relate protein domains in PDB.

Clique finding is computationally intensive, and is particularly useful for *large grain* search, which can then be refined down to *fine grain* alignment using a randomized search that can make more precise measures of alignment quality. Again, this is the strategy taken by VAST. For POA in general, it can be useful for aligning large-grain segments (blocks, HSPs, domains, ...) of sequences, and this alignment can then be extended to smaller-grain (e.g. residue) levels using other methods. Because the nucleotide alphabet is small, naively-constructed fusion graphs for nucleotide sequences can easily have thousands of nodes and millions of edges. Heuristics and ingenuity can sometimes be used to limit the size of the fusion graph to hundreds of nodes and thousands of edges, and clique

[1] A **clique** is a completely connected subgraph, i.e., a subgraph every pair of whose nodes is connected.

```
CYYH GJQSHDE I
LWPH GJQSHDE I
LWPH GJQSNDE I
LWPHTGJQSNDE I
LFPHTGJQSNDE I
LFPHTGJQSNDEJI
LFPHTGRVANDEJI
LWPH GJQSNDD I
```

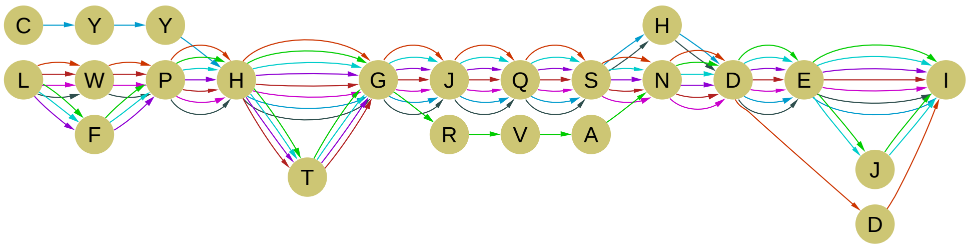

Figure 9: *A set of sequences and their Minimal Common Supergraph under the node+edge graph size measure, with identity fusions. Each of the input sequences is shown in the graph as a single-color path. As noted in Figure 5, the graph structure gives an alternative representation of sequence edit operations — such as the insertion of* T *and the introduction of* RVA *by recombination.*

```
UT
AAT
GUT
UUT
CAADT
EAAAT
FAAAT
UAAAT
AAAABT
```

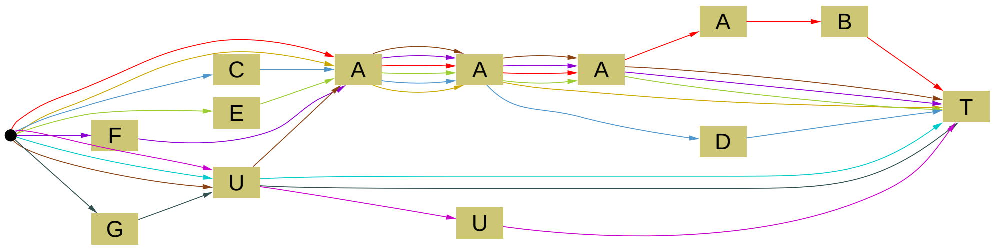

Figure 10: *A set of BAliBASE 2.0 repeat sequences (from [22]) and their resulting Minimal Common Supergraph. The sequences are again shown as single-color paths.*

Hs#S337687	aagttcctgctgcgtttgctggactgatgtacttggtttgtgnaggcaa
Hs#S813765	a.g.tcctgc.gcgtttgc.ggacggatgtacttg..ttgtg.aggcaa
Hs#S1794113	a.gttcctgctgcgcttgctggactgatgtacttg.tttgtg.aggcaa
Hs#S674099	a.gttcctgctgcgtttgctggactgatgtacttg.tttgtg.aggcaa
Hs#S4698	a.gttcctgctgcgtttgctggactgatgtacttg.tttgtg.cggcaa
Hs#S629177	a.gttcctgctgcgtttgctggactgatgtacttg.tttgt.naggcaa
Hs#S672182	a.gttcctgctgcgtttgctggactgatgtacttg.ttt..........
Hs#S663801	a.gttcctgctgcgtttgctggacttatgtacttg.tttgtg.aggcaa
Hs#S1988018	a.gttcctgctgcttttgctggactgatgtacttg.attgtg.aggcaa
Hs#S196113	a.gttnctgntgngtttgctggactgatgtacttg.tttgtg.aggcaa

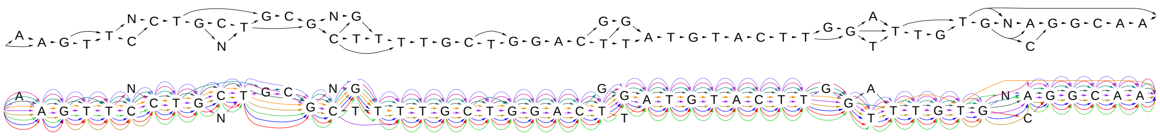

Figure 11: *Two PO-MSAs generated for the indicated set of Hs.100194 sequence segments, from Figure 1, like the graph constructed by hand in Figure 2. The two graphs shown here were actually identical, but were rendered using different* dot *program parameters.*

methods can then be used effectively. However, in general, the clique methods shown here will probably be most effective for large-grain problems.

To illustrate, Figure 10 shows a POA for sequences of protein repeats, which is one of the MSA benchmarks in the BAliBASE 2.0 suite [22]. The graph shown is the Minimal Common Supergraph. The POA diagram suggests a mixture of two or more models, one yielding repeats of A segments, and another repeats of U segments. By considering alignment at a block level like this, the clique search involved becomes tractable.

Although finding an optimal supergraph is computationally difficult, it can be practical to find near-optimal supergraphs using these methods. The program mentioned above allows specification of search termination criteria such as time limits and number of candidate cliques found. Also, iterative alignment techinques may be used to find a good supergraph; a PO-MSA generated by iterative pairwise *MCS* for Hs.100194 sequence fragments from Figure 1 is shown in Figure 11. This POA was produced in 11 seconds on a 750 MHz SUNW UltraSPARC-III CPU. For perspective, consider the problem of aligning the first two sequences shown. If each nucleotide letter is represented as a graph node, the resulting fusion graph has 549 vertices and 227862 edges: because the nucleotide alphabet is tiny, there are a large number of possible identity fusions. Even despite this brute-force approach to fusion, for which an exhaustive search will require lots of time, the Bron-Kerbosch algorithm finds a maximal clique of size 41 in less than a second. In fact, this clique is the largest possible — it is not possible to fuse more than 41 nodes among the two sequences. Alternatively, by taking a more intelligent approach, we can limit the search dramatically: if we represent subsequences of 3 nucleotides as single graph nodes, then the resulting fusion graph then has only 56 vertices and 1979 edges, and there are only 6 maximal cliques with 29 fusions each. These cliques are all found by Bron-Kerbosch in less than a second, and the resulting alignment again turns out to be optimal. The POA shown in Figure 11 was produced this way. With greater ingenuity the search can be reduced still further. The more the searches can be limited to fusion of features like blocks or segments, rather than fusion of individual residues, the larger the problems that can be handled.

4 Conclusion

Partial Order Alignment (POA) [9, 13, 14, 16, 18] presents an alternative to conventional Multiple Sequence Alignment methods. Rather than a tabular arrangement of sequences, alignments with POA can be a graph that contains them as paths. We have argued that this approach casts alignments as *sequence emission models* — HMM- or grammar-like graphs that can generate sequences — and that this perspective has some advantages.

This paper has concentrated on presenting the POA concept in terms of graphs, leaving open many specific questions about implementation. The aim has been to develop a foundation for further exploration, keeping in mind specific concerns about Multiple Sequence Alignments. A PO-MSA implementation is available for experimentation at `www.bioinformatics.ucla.edu/poa`.

Acknowledgement

We gratefully acknowledge the many contributions of Catie Grasso to this work. A conversation with Pavel Pevzner was important in improving the exposition. This research was supported by NSF grant IIS 0082964 and NIH grant MH065166.

References

[1] S.F. Altschul, “The Statistics of Sequence Similarity Scores”, NCBI BLAST online tutorial.
`http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html`

[2] NCBI BLAST, home page:
`http://www.ncbi.nlm.nih.gov/BLAST/`

[3] C. Bron, J. Kerbosch, “Algorithm 457: Finding all cliques of an undirected graph”, *Comm. ACM* 16, 575–577, 1973.

[4] H. Bunke et al., “On the Minimum Common Supergraph of Two Graphs”, *Computing* 65, 13–25, 2000.
`http://www.iam.unibe.ch/~fki/publications/papersOnGraphMatching/CommonSupergraphV2.ps.gz`

[5] I. M. Bomze, M. Budinich, P. M. Pardalos, M. Pelillo. “The maximum clique problem”, in D.-Z. Du and P. M. Pardalos, eds., *Handbook of Combinatorial Optimization*, volume 4, Kluwer, 1999.
`http://citeseer.nj.nec.com/bomze99maximum.html`

[6] J. Chen et al., “MMDB: Entrez’s 3D-structure database”, *Nucleic Acids Research* 31:1, 474–477, January 1 2003.
`http://www.ncbi.nlm.nih.gov/Structure/MMDB/mmdb.shtml`

[7] R. Durbin et al., *Biological sequence analysis*, Cambridge University Press, 1998.

[8] E.R. Gansner, E. Koutsofios, S.C. North, and K.P. Vo, “A technique for drawing directed graphs,” *IEEE Trans. Soft. Eng.* 19, 214–230, 1993.
graphviz home: `http://www.research.att.com/sw/tools/graphviz`

[9] C. Grasso, C. Lee, “Applying Partial Order Alignment to Progressive Multiple Sequence Alignment of Proteins”, *Bioinformatics*, to appear, 2003.

[10] H.M. Grindley, P.J. Artymuik, D.W. Rice, P. Willett, “Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm”, *J. Mol. Biol.* 229, 707–721, 1993.

[11] D. Gusfield, *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, 1997.

[12] D. Higgins, J. Thompson, T. Gibson, J.D. Thompson, D.G. Higgins, T.J. Gibson, “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.” *Nucleic Acids Res.* 22:4673-4680 (1994).
`http://www.ebi.ac.uk/clustalw/`

[13] K. Irizarry, et al., C. Lee, “Genome-wide analysis of single-nucleotide polymorphisms in human expressed sequences,” *Nature Genetics* 26, 233–236, October 2000.

[14] C. Lee, C. Grasso, and M. Sharlow, “Multiple sequence alignment using partial order graphs”, *Bioinformatics* 18, 452–464, 2002.
`http://www.bioinformatics.ucla.edu/poa/`
Online tutorial at `http://www.bioinformatics.ucla.edu/poa/Poa_Tutorial.htm`

[15] T. Lassman, E. Sonnhammer, “Quality assessment of multiple alignment programs”, *FEBS Letters* 529, 126–130, 2002.

[16] C. Lee, “Generating Consensus Sequences from Partial Order Multiple Sequence Alignment Graphs,” *Bioinformatics*, 2003, in press.

[17] T. Madej, J.-F. Gibrat, S.H. Bryant, “Threading a database of protein cores”, *Proteins*, 23, 356–369, 1995.

[18] B. Modrek, A. Resch, C. Grasso, C. Lee, “Genome-wide detection of alternative splicing expressed sequences of human genes,” *Nucleic Acids Research* 29:13, 2850–2859, October 2001.

[19] B. Morgenstern, A. Dress, T. Werner, “Multiple DNA and protein sequence alignment based on segment-to-segment comparison”, *Proc Natl Acad Sci U S A* 93, 12098-12103, 1996.

[20] D.S. Parker, C. Lee, “Pairwise Partial Order Alignment as a Graph Problem — Aligning Alignments Revisited”, submitted for publication, August 2003.

[21] VAST: the NCBI Vector Alignment Search Tool.
`http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml`

[22] J.D. Thomson, F. Plewniak, O. Poch, “BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs”, *Bioinformatics*, 15:1, 87–88, 1999.
`http://bess.u-strasbg.fr/BioInfo/BAliBASE2`
Kringle1 (repeats with an additional conserved Trypsin domain) data used in Figure 10 is at:
`http://bess.u-strasbg.fr/BioInfo/BAliBASE2/ref6/test_3/kringle_1_ref6_schema.html`

[23] P. Willett, “Matching of chemical and biological structures using subgraph and maximal common subgraph isomorphism algorithms”, in: D.G. Truhlar et al. (eds.), *Rational drug design*, NY: Springer-Verlag, 11–38, 1999.