

Improved Parameterized Algorithms for Network Query Problems

Ron Y. Pinter, Hadas Shachnai, and Meirav Zehavi

Department of Computer Science, Technion, Haifa 32000, Israel
{pinter,hadas,meizeh}@cs.technion.ac.il

Abstract. Biological networks are often explored through extensive use of network queries. *Partial Information Network Queries (PINQ)* address the major challenge of analyzing such networks in the absence of certain topological data. In the PINQ problem, we are given a host graph H , modeling a network, and a pattern \mathcal{P} , whose topology is only *partially* known. We seek a subgraph of H that *resembles* \mathcal{P} .

In this paper, we study a generalization of PINQ which allows *near resemblance* between H and \mathcal{P} . We obtain an exact parameterized algorithm as well as an FPT-approximation scheme for a wide class of inputs, where the parameter is the number of nodes in \mathcal{P} . Our algorithms substantially improve the best O^* running times in solving PINQ, as well as the special cases of the *Alignment Network Query* problem and the *Topology-Free Network Query* problem.

1 Introduction

Biological networks orchestrate the complex functions of the living cells. With the increasing amount of data on these networks available, the discovery of conserved patterns or signaling pathways has become of major importance. Such intrinsic structural properties can be identified through the extensive use of network queries, which compare the *graph* modeling the network with a given *pattern*. Indeed, the well-known *Alignment Network Query (ANQ)* and *Topology Free Network Query (TFNQ)* problems play a pivotal role in the analysis of biological networks (see, e.g., [15] and [33]). Given a pattern \mathcal{P} and an undirected graph H , these queries seek a subgraph of H that *resembles* \mathcal{P} .

TFNQ requires only the connectivity of the solution, while ANQ requires resemblance between the *topology* of \mathcal{P} and the solution. *Partial Information Network Queries (PINQ)*, introduced in [30], fit for the common scenario where we have only partial information on the topology of \mathcal{P} .

Since network query problems are often NP-hard, there is a growing body of literature studying their parameterized complexity (see Section 1.2). A problem is *fixed-parameter tractable (FPT)* with respect to a parameter k if it can be solved in time $O^*(f(k))$ for some function f , where O^* hides factors polynomial in the input size.

In this paper, we consider a natural generalization of PINQ, which allows *insertions and deletions (indels)* of nodes. We develop for *PINQ with indels (PINQ_I)* exact as well as approximation parameterized algorithms.

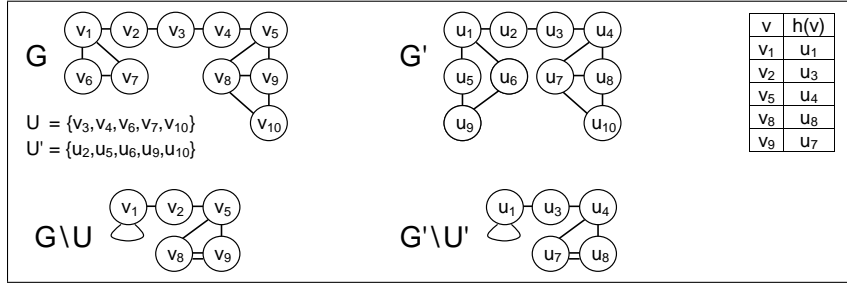


Fig. 1. An example of a homeomorphism h from G to G' .

1.1 Problem Statement

Given a graph H and a set of graphs \mathcal{P} , in PINQ_I we seek a disjoint collection of subgraphs of H , each resembling a different graph in \mathcal{P} , whose union is a connected graph. Each of these subgraphs is mapped to the graph it resembles in \mathcal{P} , by using a variant of isomorphism allowing to delete degree-2 nodes, called *homeomorphism* (defined below). For biological motivation, see, e.g., [28].

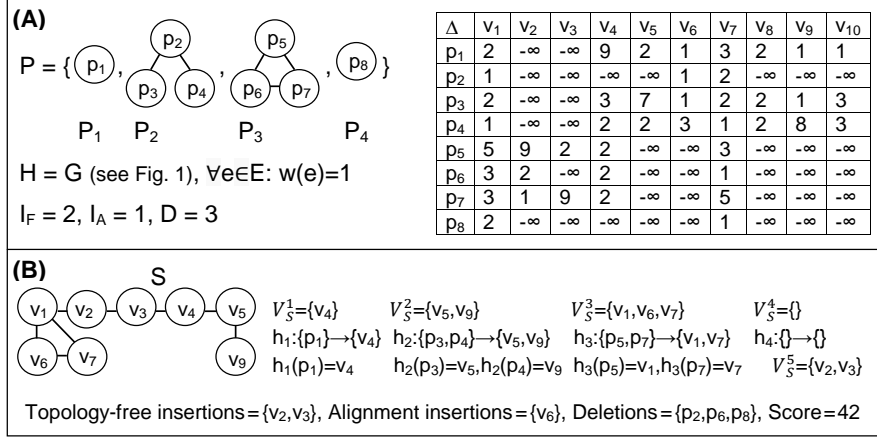
Homeomorphism: Given a graph $G = (V, E)$ and a subset U of degree-2 nodes in V , generate the multigraph $G \setminus U$ as follows (see Fig. 1). Delete from G the nodes in U and their adjacent edges. For every pair of nodes $v, u \in V \setminus U$ and every simple path in G connecting them, in which all other nodes belong to U , add an edge $\{v, u\}$. For every node $v \in V \setminus U$ and every simple cycle in G consisting of v and nodes in U (only), add a self-loop to v .

A homeomorphism from $G = (V, E)$ to $G' = (V', E')$ is defined as an isomorphism from $G \setminus U$ to $G' \setminus U'$, where U and U' are subsets of degree-2 nodes in V and V' , respectively. To simplify the presentation, we use the term homeomorphism also when referring to a function whose domain is empty.

Definition of PINQ_I : The input for PINQ_I consists of a set of graphs $\mathcal{P} = \{P_1, \dots, P_t\}$, where $P_i = (V_i, E_i)$, and a graph $H = (V, E)$ having real numbers as edge-weights, along with a *similarity score* table Δ . The table Δ contains an entry $\Delta(p, h) \in \mathbb{R} \cup \{-\infty\}$ for any pair of nodes p, h , where $p \in V_i$, $1 \leq i \leq t$ and $h \in V$ (an entry $\Delta(p, h) = -\infty$ indicates that p and h cannot be matched). The input contains also the nonnegative integers I_F, I_A and D . Let $k = \sum_{i=1}^t |V_i|$ denote the total number of nodes in \mathcal{P} (see Fig. 2(A)).

We now give the definition of a *solution* to PINQ_I , illustrated in Fig. 2(B). Let $\mathcal{S} = (S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$, where $S = (V_S, E_S)$ is a connected subgraph of H , $\{V_S^1, \dots, V_S^{t+1}\}$ is a partition of V_S , and h_i is a homeomorphism from P_i to the subgraph of S induced by V_S^i , for all $1 \leq i \leq t$. Let $\text{dom}(f)$ and $\text{ima}(f)$ denote the domain and image of a function f , respectively, and let $w(e)$ denote the weight of an edge e . The number of indels and score of \mathcal{S} are defined as follows.

- The number of *free insertions* is $|V_S^{t+1}|$. Informally, this is the number of nodes connecting the subgraphs of S that are mapped to graphs in \mathcal{P} .
- The number of *alignment insertions* is the number of unmapped nodes in $\bigcup_{i=1}^t V_S^i$, i.e., $\sum_{i=1}^t |V_S^i \setminus \text{ima}(h_i)|$. Informally, this is the number of nodes that are not mapped to nodes of graphs in \mathcal{P} , and yet belong to the subgraphs of S that are mapped to graphs in \mathcal{P} .



Reference	Weights	Indels	The topology of each P_i	Time Complexity
Pinter et al. [30]	\mathbb{R}	No	Tree	$O^*(6.75^{k+O(\log^2 k)} 3^t)$
This paper	Z	Yes	Bounded treewidth	$O^*(3.7^{k-D+I_A} \mathbf{W})$
This paper ^s	\mathbf{N}^0	Yes	Bounded treewidth	$O^*(3.7^{k-D+I_A} \lfloor \frac{1}{\epsilon} \rfloor)$

Table 1. Parameterized algorithms for PINQ_I .

Reference	Weights	Indels	The Topology of P_1	Time Complexity
Blin et al. [6]	\mathbb{R}	Yes	Bounded feedback vertex set	$O^*(8.2^{k+I_A})$
Dost et al. [12]	\mathbb{R}	Yes	Bounded treewidth	$O^*(8.2^{k+I_A})$
Shlomi et al. [32]	\mathbb{R}	Yes	Simple Path	$O^*(5.44^{k+I_A})$
Hüffner et al. [20]	\mathbb{R}	Yes	Simple Path	$O^*(4.32^{k+I_A})$
Pinter et al. [30]	\mathbb{R}	No	Tree	$O^*(6.75^k)$
Pinter et al. [30]	\mathbb{R}	No	Simple Path	$O^*(4^k)$
This paper	Z	Yes	Bounded treewidth	$O^*(2^{k-D+I_A} \mathbf{W})$
This paper ^s	\mathbf{N}^0	Yes	Bounded treewidth	$O^*(2^{k-D+I_A} \lfloor \frac{1}{\epsilon} \rfloor)$

Table 2. Parameterized algorithms for ANQ_I .

handling a wide class of inputs (see Theorem 1). We achieve improved running times of $O^*(2^{k-D+I_A} \mathbf{W})$ and $O^*(2^{k-D} \mathbf{W})$ for the special cases of ANQ_I and TFNQ_I , respectively. Algorithm Exact improves and unifies the previous results as follows.

- We extend the PINQ algorithm presented in [30], by considering indels and bounded treewidth graphs (see Table 1). Note that a graph with a bounded feedback vertex set has a bounded treewidth [8]. Thus, our results hold also for graphs with bounded feedback vertex sets.
- For inputs with polynomially bounded integral weights, we significantly improve the O^* running times of the best known algorithms for PINQ (due to [30]) and ANQ_I (due to [12] and [20]). For example, using the real data presented in [25], the weights in the table Δ can take integral values in $\{-\infty, 0, \dots, 4\}$. Applying the best known algorithm (of [12]) for ANQ_I , where P_1 has a bounded treewidth, we get a running time of $O^*(8.2^{k+I_A})$, whereas Exact solves ANQ_I on such inputs in time $O^*(2^{k-D+I_A})$. We note that both algorithms have the same dependency on the treewidth of P_1 .
- We extend the algorithm for TFNQ_I presented in [5] for unweighted instances to handle instances with integral weights.
- Algorithm Exact has the same O^* running time as the best known parameterized algorithms for Subgraph Isomorphism, in which the subgraph is a tree [24], or has a bounded treewidth [16]. The same holds for group Steiner tree [26], and for Min Connected Components [29]. Indeed, all of these problems are special cases of PINQ_I .

We also give an $O^*(3.7^{k-D+I_A} \lfloor \frac{1}{\epsilon} \rfloor)$ time FPT-AS for PINQ_I , which handles a wide class of inputs (see Theorem 2). We achieve improved running times of $O^*(2^{k+I_A-D} \lfloor \frac{1}{\epsilon} \rfloor)$ and $O^*(2^{k-D} \lfloor \frac{1}{\epsilon} \rfloor)$ for ANQ_I and TFNQ_I , respectively.

Notation: Let $V(\mathcal{P}) = \bigcup_{i=1}^t V_i$ and $E(\mathcal{P}) = \bigcup_{i=1}^t E_i$ denote the sets of nodes and edges in \mathcal{P} , respectively. Also, let \mathcal{P}^* be the set of single-node graphs in \mathcal{P} , and let $V(\mathcal{P}^*) = \bigcup_{P_i \in \mathcal{P}^*} V_i$ and $k^* = |\mathcal{P}^*|$. Let $V(G)$ and $E(G)$ denote the sets of nodes and edges of a graph G , respectively.

Reference	Weights	Indels	Time Complexity
Bruckner et al. [9]	\mathbb{R}	Yes	$O^*(k!3^k)$
Dondi et al. [11]	$\{0\}$	Yes	$O^*(32^{k-D})4^{O(k-D)}$
Fellows et al. [14]	$\{0\}$	No	$O^*(87^k)$
Pinter et al. [30]	\mathbb{R}	No	$O^*(20.25^{k+O(\log^2 k)})$
Betzler et al. [1]	$\{0\}$	Yes	$O^*(29.6^{k-D})$
Betzler et al. [1]	$\{0\}$	No	$O^*(10.88^k)$
Betzler et al. [2]	$\{0\}$	No	$O^*(4.32^k)$
Guillemot et al. [18]	\mathbb{N}^0	Yes	$O^*(4^{k-D}W^2)$
Koutis [23]	$\{0\}$	Yes	$O^*(2.54^{k-D})$
Pinter et al. [29]	\mathbb{N}^0	Yes	$O^*(2^k W)$
Björklund et al. [5]	$\{0\}$	Yes	$O^*(2^{k-D})$
This paper	Z	Yes	$O^*(2^{k-D}W)$
This paper ^s	\mathbb{N}^0	Yes	$O^*(2^{k-D} \lfloor \frac{1}{\epsilon} \rfloor)$

Table 3. Parameterized algorithms for TFNQ_I.

2 An Overview of Our Algorithms

Algorithm **Exact** uses a non-trivial combination of *narrow sieves* [4] (see also [22]) and *divide-and-color* [10], which are often applied as two independent tools in the development of parameterized algorithms.

Narrow sieves is an algebraic technique in which we express a parameterized problem by associating monomials with potential solutions. Each monomial either represents a unique correct solution, or an even number of incorrect solutions. Having a polynomial that is the sum of such monomials, we need to determine whether it has a monomial whose coefficient is odd.

Given an input for PINQ_I, our algorithm **Exact** examines different choices for the number of graphs in \mathcal{P}^* that appear in the solution. When this number is “small”, the algorithm calls the procedure **FewSingles**; else, it calls the procedure **ManySingles**. Both associate monomials with potential solutions, evaluate several sums of such monomials by using dynamic-programming, and then return the maximum score of the desired solution. Our novel application of narrow sieves, which replaces the standard single procedure by these two procedures, may be useful in solving other parameterized problems involving graph connectivity.

Divide-and-color is a combinatorial technique in which we have a set S_n of n elements, and we seek a certain subset S_k of k elements in S_n . We randomly partition S_n into two sets: S_n^1 and S_n^2 . Thus, we get the problem of finding a subset $S \subseteq S_k$ in S_n^1 , and another problem of finding the subset $S_k \setminus S$ in S_n^2 .

The correctness of **ManySingles** crucially relies on a preceding selection step, which determines the nodes in V that can be mapped only to graphs in \mathcal{P}^* , as well as the nodes in V that can be mapped only to other graphs in \mathcal{P} . The improved running times of **Exact** for PINQ_I, ANQ_I and TFNQ_I are achieved by using in this step divide-and-color.

Our FPT-AS uses scaling and rounding to manipulate the weights of a given instance (recall that these are the possible values for edge-weights and scores in the table Δ , excluding $-\infty$), and then calls algorithm **Exact**.

Organization: Section 3 presents procedures **FewSingles** and **ManySingles**. Then, Section 4 gives algorithm **Exact**, and Section 5 concludes our FPT-AS.

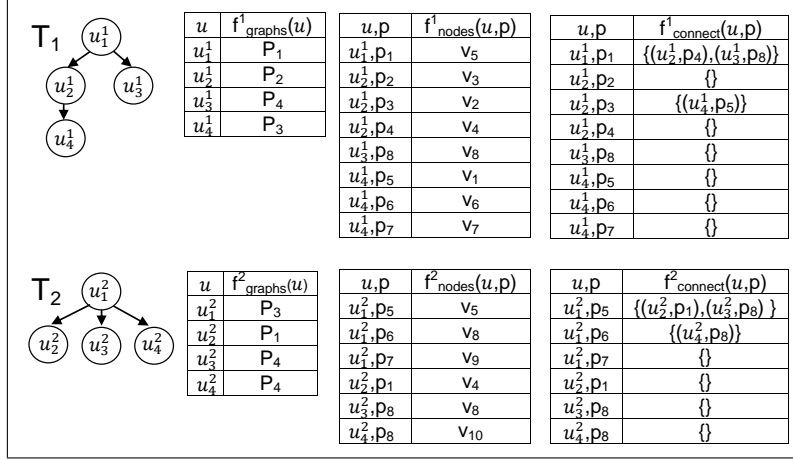


Fig. 3. Examples of v_5 -good quads (see Definition 1). The quads refer to the input in Fig. 2(A), with each score in Δ replaced by 10.

3 The Procedures FewSingles and ManySingles

In this section assume that \mathcal{P} is a set of bounded treewidth graphs, the number of input indels is minimal, and the weights are nonnegative integers. Algorithm Exact (described in Section 4) only needs the procedures to be correct under these assumptions. For the sake of clarity, we focus on a simpler version of FewSingles that cannot handle indels. Note that some proofs are given in Appendix A.

3.1 SimpleFewSingles: A Narrow Sieves Procedure

Assuming that $I_F = I_A = D = 0$, we present a narrow sieves procedure that efficiently finds solutions containing few graphs in \mathcal{P}^* . We first define the structure of a potential solution. We then describe the potential solutions, and associate them with monomials. We show how to evaluate some sums of such monomials, and finally, we present the procedure, which heavily relies on such evaluations.

3.1.1 The Structure of a Potential Solution

Recall that any cycle in a solution \mathcal{S} is contained in a subgraph induced by V_S^i , for some $1 \leq i \leq t$. Thus, by contracting each of the subgraphs into a single node, and choosing a node as a root, any solution for PINQ_I can be represented by a rooted tree. We study the mappings of such trees (into graphs in \mathcal{P}) below.

A quad $(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{con}})$ refers to a rooted tree $T = (V_T, E_T)$ on t nodes, $f_{\text{gra}} : V_T \rightarrow \mathcal{P}$, $f_{\text{nod}} : X \rightarrow V$ and $f_{\text{con}} : X \rightarrow 2^X$, where $X = \{(v, p) : v \in V_T, p \in V(f_{\text{gra}}(v))\}$. Informally, such a quad describes a structure for a solution as follows. T and f_{gra} specify which graphs to choose from \mathcal{P} and how to connect them; f_{nod} indicates how to map the nodes of graphs chosen from \mathcal{P} to nodes in V ; and f_{con} refines our information about how the chosen graphs are connected.

Next, we define the quads corresponding to structures of potential solutions for PINQ_I. We give in Fig. 3 examples of such quads.

Definition 1. Given $r \in V$, we say that a quad $(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{con}})$ is r -good if:

1. $|\{(v, p) \in \text{dom}(f_{\text{nod}}) : f_{\text{nod}}(v, p) = r\}| = |\{p \in V(f_{\text{gra}}(\text{root}(T))) : f_{\text{nod}}(\text{root}(T), p) = r\}| = 1$.

2. $\forall v \in V_T, \{p, p'\} \in E(f_{gra}(v)) : \{f_{nod}(v, p), f_{nod}(v, p')\} \in E$.
3. $\forall (v, p) \in \text{dom}(f_{nod}) : \Delta(p, f_{nod}(v, p)) \neq -\infty$.
4. $\forall (v, p) \in \text{dom}(f_{con}), (u, p') \in f_{con}(v, p) :$
 - (a) v is the father of u in T , and $\{f_{nod}(v, p), f_{nod}(u, p')\} \in E$.
 - (b) $\forall (u', p'') \in f_{con}(v, p) \setminus \{(u, p')\} : f_{nod}(u, p') \neq f_{nod}(u', p'')$.
5. $\forall u \in V_T \setminus \{\text{root}(T)\} : |\{(v, p, p') : (u, p') \in f_{con}(v, p)\}| = 1$.

Condition 1 states that we map only one node in $V(\mathcal{P})$ to r , and this node belongs to the graph mapped to the root of T . Condition 2 states that the mapping of the graphs in \mathcal{P} to subgraphs of H is correct (i.e., we map edges of graphs in \mathcal{P} to edges in E). Condition 3 states that we do not match a node in $V(\mathcal{P})$ with a node in V that cannot be matched according to Δ . Condition 4a states that f_{con} does not contradict the information provided by T on the edges connecting the graphs in \mathcal{P} . More precisely, a node v being a father of a node u in T states that $f_{gra}(v)$ and $f_{gra}(u)$ are connected by an edge. Only then f_{con} may provide information on the connecting edge, where $(u, p') \in f_{con}(v, p)$, for some $p \in V(f_{gra}(v))$ and $p' \in V(f_{gra}(u))$, states that p and p' are connected by an edge (which, by this condition, is mapped to an edge in E). Condition 4b avoids some quads in which several nodes in $V(\mathcal{P})$ are mapped to the same node in V . Finally, Condition 5 states that for each pair of a node u and its father v in T , f_{con} provides information on exactly one pair (p, p') , for some $p \in V(f_{gra}(v))$ and $p' \in V(f_{gra}(u))$, indicating that p and p' are connected by an edge.

We now define the score of an r -good quad by the mapping of the edges in $E(\mathcal{P})$, the pairs of matched nodes, and the edges connecting the graphs in \mathcal{P} . In Fig. 3 the score of $(T_1, f_{gra}^1, f_{nod}^1, f_{con}^1)$ is 88, and of $(T_2, f_{gra}^2, f_{nod}^2, f_{con}^2)$ is 66.

Definition 2. The score of an r -good quad $(T, f_{gra}, f_{nod}, f_{con})$ is

$$\sum_{v \in V_T, \{p, p'\} \in E(f_{gra}(v))} w(\{f_{nod}(v, p), f_{nod}(v, p')\}) + \sum_{(v, p) \in \text{dom}(f_{nod})} [\Delta(p, f_{nod}(v, p)) + \sum_{(u, p') \in f_{con}(v, p)} w(\{f_{nod}(v, p), f_{nod}(u, p')\})].$$

3.1.2 Potential Solutions

Let $L = \{1, \dots, k+t\}$ be the set of indices used in labeling r -good quads (recall that $t = |\mathcal{P}|$ and $k = \sum_{i=1}^t |V_i|$), defining potential solutions of the same score as follows.

Definition 3. Given an r -good quad $(T, f_{gra}, f_{nod}, f_{con})$ and $\ell : V_T \cup \text{dom}(f_{nod}) \rightarrow L$ satisfying $|\text{dom}(\ell)| = k+t$, we say that $(T, f_{gra}, f_{nod}, f_{con}, \ell)$ is an r -solution.

We now define two sets of potential solutions. $Sol(r, s)$ contains all r -solutions $(T, f_{gra}, f_{nod}, f_{con}, \ell)$ of score s such that ℓ is bijective; and $Cor(r, s) = \{(T, f_{gra}, f_{nod}, f_{con}, \ell) \in Sol(r, s) : f_{gra} \text{ and } f_{nod} \text{ are injective}\}$.

Informally, the next lemma implies that each set $Cor(r, s)$ includes enough potential solutions from $Sol(r, s)$, and all these potential solutions are *correct*.

Lemma 1. The input has a solution of score s iff $\bigcup_{r \in V} Cor(r, s) \neq \emptyset$.

Note that $(T, f_{gra}, f_{nod}, f_{con}, \ell), (T', f'_{gra}, f'_{nod}, f'_{con}, \ell') \in Sol(r, s)$ are equal iff there is an isomorphism *iso* between the rooted trees T and T' , such that

1. $\forall v \in V_T : f_{gra}(v) = f'_{gra}(iso(v))$, and $\ell(v) = \ell'(iso(v))$.
2. $\forall (v, p) \in \text{dom}(f_{nod}) : f_{nod}(v, p) = f'_{nod}(iso(v), p)$, $\ell(v, p) = \ell'(iso(v), p)$, and $[\forall (u, p') : (u, p') \in f_{con}(v, p) \text{ iff } (iso(u), p') \in f'_{con}(iso(v), p)]$.

3.1.3 Associating Monomials with Potential Solutions

Recall that, in the narrow sieves technique, a parameterized problem is solved via associating monomials with potential solutions. Towards defining these monomials, we introduce the variables x , $y_{e,h}$ for all $e \in V(\mathcal{P}) \cup V$ and $h \in V$, and $z_{e,l}$ for all $e \in \mathcal{P} \cup V$ and $l \in L$. Let ind denote the number of these variables, i.e., $ind = 1 + (k + |V|)|V| + (t + |V|)|L|$.

We next define the monomials associated with potential solutions. In defining a monomial for an r -solution $sol \in Sol(r, s)$, we store information about sol that allows reconstructing sol iff it is a correct solution (i.e., $sol \in Cor(r, s)$).

Definition 4. $m(T, f_{gra}, f_{nod}, f_{con}, \ell) = x^s \prod_{v \in V_T} z_{f_{gra}(v), \ell(v)}$

$$\prod_{(v,p) \in \text{dom}(f_{nod})} [y_{p, f_{nod}(v,p)} z_{f_{nod}(v,p), \ell(v,p)} \prod_{(u,p') \in f_{con}(v,p)} y_{f_{nod}(v,p), f_{nod}(u,p')}].$$

Given an r -solution, x tracks its score (as in [3]). The product $\prod_{v \in V_T} z_{f_{gra}(v), \ell(v)}$ specifies which graphs to choose from \mathcal{P} and how to label them. Then, the product $\prod_{(v,p) \in \text{dom}(f_{nod})} y_{p, f_{nod}(v,p)} z_{f_{nod}(v,p), \ell(v,p)}$ specifies how to map nodes in $V(\mathcal{P})$ to nodes in V and how to label nodes in V . Finally, $\prod_{(v,p) \in \text{dom}(f_{nod}), (u,p') \in f_{con}(v,p)} y_{f_{nod}(v,p), f_{nod}(u,p')}$ indicates how to connect the graphs chosen from \mathcal{P} .

The following lemma states that the monomials are defined in a manner compatible with the narrow sieves technique: Different correct solutions are associated with different monomials, and a monomial of an incorrect solution represents an even number of incorrect solutions.

Lemma 2. *All pairs $\{sol, sol'\}$ of different r -solutions in $Cor(r, s)$ satisfy $m(sol) \neq m(sol')$. Moreover, $Sol(r, s) \setminus Cor(r, s)$ can be partitioned into a set of pairs $\{sol, sol'\}$ satisfying $m(sol) = m(sol')$.*

3.1.4 Evaluating the Sum of the Monomials

For each $r \in V$, define $P(r) = \sum_{s \in \{0, \dots, (|V|+|E|)W\}, sol \in Sol(r, s)} m(sol)$. We next evaluate these polynomials over the field \mathbb{F}_q (i.e., the finite field of order q), where $q = 2^{\lceil \log_2(10(2(k+t)+t)) \rceil}$.

By Lemmas 1 and 2, the input has a solution of score s iff there exists a node $r \in V$ such that $P(r)$ has a monomial with an odd coefficient in which the degree of x is s . Since \mathbb{F}_q has characteristic 2, we have the following result.

Lemma 3. *The input has a solution of score s iff there is a node $r \in V$ such that $P(r)$ has a monomial in which the degree of x is s .*

Given $A \subseteq L$, let $P_A(r) = \sum_{sol \text{ is an } r\text{-solution in which } \text{ima}(\ell) \subseteq A} m(sol)$. Using inclusion-exclusion, and since \mathbb{F}_q has characteristic 2, we have that $P(r) = \sum_{A \subseteq L} P_A(r)$. Thus, we can evaluate $P(r)$ by using the following lemma.

Lemma 4. *Let $A \subseteq L$ and $a_1, \dots, a_{ind-1} \in \mathbb{F}_q$. For all $r \in V$, $P_A(r)(x, a_1, \dots, a_{ind-1})$ (assign values to all variables excluding x) can be evaluated in $O(W \log W |V|^{tw+O(1)} k^{O(1)})$ time and $O(W |V|^{tw+O(1)} k^{O(1)})$ space by using dynamic programming.*

3.1.5 Concluding Procedure SimpleFewSingles

The pseudocode of SimpleFewSingles is given below. First, SimpleFewSingles chooses values from the field \mathbb{F}_q , to be assigned to all the variables, excluding x , of polynomials of the form $P_A(r)$, where $r \in V$ and $A \subseteq L$. Then, it evaluates these polynomials, and thus evaluates polynomials of the form $P(r)$, where $r \in V$, as explained in Section 3.1.4. Finally, it determines the maximum score s of a solution by verifying that at least one evaluation of a polynomial of the form $P(r)$, where $r \in V$, resulted in a polynomial (whose only variable is x) of degree s .

Lemma 4 implies the time and space complexities of SimpleFewSingles, while correctness follows from Lemma 3 and the next lemma, proved in [31, 34].

Lemma 5. *Let $p(x_1, \dots, x_n) \neq 0$ be a polynomial of degree d over a finite field F . Then, for $a_1, \dots, a_n \in F$ i.u.d. random variables, $\Pr(p(a_1, \dots, a_n) \neq 0) \geq 1 - d/|F|$.*

We summarize in the following result.

Lemma 6. *If there is a solution, then SimpleFewSingles returns OPT with probability $\geq \frac{9}{10}$, and does not return a higher score otherwise; else, it rejects. It uses $O(2^{k+t} W \log W |V|^{tw+O(1)} k^{O(1)})$ time and $O(W |V|^{tw+O(1)} k^{O(1)})$ space.*

Procedure SimpleFewSingles(\mathcal{P}, H, Δ)

- 1: **forall** $r \in V$ **do** $Sum[r] \leftarrow 0$. **end for**
 - 2: select $a_1, \dots, a_{ind-1} \in \mathbb{F}_q$ independently and uniformly at random.
 - 3: **forall** $A \subseteq L, r \in V$ **do** $Sum[r] \leftarrow Sum[r] + P_A(r)(x, a_1, \dots, a_{ind-1})$. **end for**
 - 4: return the maximum value s such that (there exists $r \in V$ for which $Sum[r]$ is a nonzero polynomial of degree s), where if no such s exists – reject.
-

3.2 The Procedure FewSingles

Procedure FewSingles extends SimpleFewSingles to handle indels. Now the input is of the form $(n_E, n_{\mathcal{P}}, \mathcal{P}, H, \Delta, I_F, I_A, D)$, where n_E and $n_{\mathcal{P}}$ are nonnegative integers. Informally, n_E and $n_{\mathcal{P}}$ indicate that we seek solutions containing exactly n_E edges from E , and $n_{\mathcal{P}}$ graphs in \mathcal{P} that are not entirely deleted.

We briefly note how we obtain FewSingles from SimpleFewSingles (the details are given in Appendices C and D). First, we modify the quads presented in Section 3.1.1 to comply with n_E and $n_{\mathcal{P}}$, allow f_{nod} to delete nodes in $V(\mathcal{P})$, add a function mapping paths whose internal nodes are deleted to walks in H (resulting in alignment insertions), and allow f_{con} to insert nodes between the graphs it connects (resulting in free insertions). Second, we use the set $\{1, \dots, k - D + I_A + n_{\mathcal{P}}\}$ (instead of $\{1, \dots, k + t\}$) in defining the potential solutions. This set allows labeling the graphs chosen from \mathcal{P} (using $n_{\mathcal{P}}$ indices), the nodes in V that are mapped to nodes in these graphs (using $k - D$ indices), and the alignment insertions (using I_A indices). We summarize in the following result.

Lemma 7. *If there is a solution s.t. $|E_S| = n_E$, where $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ nonempty sets, then FewSingles returns the maximum score of such a solution with probability $\geq \frac{9}{10}$, and not a higher score otherwise; else, it rejects. It uses $O(2^{k-D+I_A+n_{\mathcal{P}}} W \log W |V|^{tw+O(1)} k^{O(1)})$ time and $O(W |V|^{tw+O(1)} k^{O(1)})$ space.*

3.3 The Procedure ManySingles

The procedure ManySingles efficiently finds solutions that contain many graphs

from \mathcal{P}^* . Its input is of the form $(n_E, n_{\mathcal{P}^*}, n_{\mathcal{P}}, \mathcal{P}, H, \Delta, I_F, I_A, D)$, where $n_{\mathcal{P}^*} \in \mathbb{N}^0$ indicates that we seek solutions containing exactly $n_{\mathcal{P}^*}$ graphs from \mathcal{P}^* .

Algorithm **Exact** only needs the procedure to be correct under the assumption that there is a set $U \subseteq V$ satisfying $[\forall h \in U : \text{If } p \in V(\mathcal{P}) \setminus V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty]$ and $[\forall h \in V \setminus U : \text{If } p \in V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty]$ (see Section 4). This allows us to avoid labeling nodes in V that can only be mapped to nodes in $V(\mathcal{P}^*)$, and thus use the set $\{1, \dots, k - D + I_A + n_{\mathcal{P}} - n_{\mathcal{P}^*}\}$ (instead of $\{1, \dots, k - D + I_A + n_{\mathcal{P}}\}$) in defining potential solutions. The details are given in Appendices C and E. We summarize in the following result.

Lemma 8. *If there is a solution without alignment insertions from U , satisfying $|E_S| = n_E$, in which $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ nonempty sets and $n_{\mathcal{P}^*}$ one-node sets, then **ManySingles** returns the maximum score of such a solution with probability $\geq \frac{9}{10}$, and not a higher score otherwise; else, it rejects. It uses $O(2^{k-D+I_A+n_{\mathcal{P}}-n_{\mathcal{P}^*}} W \log W |V|^{tw+O(1)} k^{O(1)})$ time and $O(W |V|^{tw+O(1)} k^{O(1)})$ space.*

4 An Exact Algorithm

We now describe our main algorithm (see the pseudocode below). **Exact** first checks if the number of input indels is minimal (Step 1). Then it manipulates the weights to be nonnegative (Step 2). The variable s , initialized to $-\infty$, holds the highest score found so far, corresponding to the original weights. **Exact** iterates over all choices for $n_E, n_{\mathcal{P}^*}$ and $n_{\mathcal{P}}$, specifying the number of edges, graphs from \mathcal{P}^* and graphs from \mathcal{P} , respectively, in the currently searched solution (Step 3).

For each choice, **Exact** uses a calculation which determines whether $n_{\mathcal{P}^*}$ is “small” or “large” (Step 4), indicating whether it is now preferable (in terms of running time) to call **FewSingle** or **ManySingles**. If $n_{\mathcal{P}^*}$ is “small”, **Exact** calls **FewSingles** to compute the maximum score of a solution complying with $n_E, n_{\mathcal{P}^*}$ and $n_{\mathcal{P}}$ (Step 5). In this step, the term $v(n_E + k - D)$ is used to correctly compare between the score returned by **FewSingles** and s , since only s concerns the original weights. Next assume that $n_{\mathcal{P}^*}$ is “large”. Before calling **ManySingles** (Step 12), **Exact** uses divide-and-color (Steps 7–11) to examine several choices concerning which nodes in V should be used in mapping graphs in \mathcal{P}^* , and which should be used in mapping graphs in $\mathcal{P} \setminus \mathcal{P}^*$. In particular, the number of iterations of Step 7 ensures that, with good probability, **Exact** examines such a choice that complies with a solution of maximum score. Finally, **Exact** returns the score s , unless no solution was found, in which case it rejects (Step 16).

We summarize in the next result, proved in Appendix B.

Theorem 1. ***Exact** solves PINQ_I in $O(3.698^{k-D+I_A} W \log W |V|^{tw+O(1)} k^{O(1)})$ time and $O(W |V|^{tw+O(1)} k^{O(1)})$ space, handling instances with integer weights, where \mathcal{P} is a set of bounded treewidth graphs. Its running time for ANQ_I is $O^*(2^{k+I_A-D} W)$, and for TFNQ_I , $O^*(2^{k-D} W)$.*

5 An FPT-Approximation Scheme

In this section we present **Approx**, an FPT-AS for PINQ_I . Using scaling and rounding, we manipulate the weights of a given instance, and then call algorithm **Exact**. We give the details in Appendix F, and summarize in the next result.

Theorem 2. ***Approx** is an FPT-AS for PINQ_I , handling instances with non-*

Algorithm 1: $\text{Exact}(\mathcal{P}, H, \Delta, I_F, I_A, D)$

```

1: if there exist  $i_F \leq I_F, i_A \leq I_A, d \leq D$  such that  $i_F + i_A + d < I_F + I_A + D$  and
    $\text{Exact}(\mathcal{P}, H, \Delta, i_F, i_A, d)$ , skipping Step 1, returns  $s'$  then reject. end if
2: subtract  $v = \min(\text{weights})$  from every weight, and initialize  $s \leftarrow -\infty$ .
3: for  $n_E = 0, \dots, |E|$ ,  $n_{\mathcal{P}^*} = \max\{0, k^* - D\}, \dots, \min\{k^*, k - D\}$ ,
    $n_{\mathcal{P}} = n_{\mathcal{P}^*}, \dots, \min\{t, n_{\mathcal{P}^*} + (k - D + I_A - n_{\mathcal{P}^*})/2\}$  do
4:   if  $2^{n_{\mathcal{P}^*}} \leq \frac{10(k - D + I_A)^{k - D + I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k - D + I_A - n_{\mathcal{P}^*})^{k - D + I_A - n_{\mathcal{P}^*}}}$  then
5:     if  $\text{FewSingles}(n_E, n_{\mathcal{P}}, \mathcal{P}, H, \Delta, I_F, I_A, D)$  returns  $s' > s - v(n_E + k - D)$  then
6:        $s \leftarrow s' + v(n_E + k - D)$ . end if
7:     else
8:       for  $\frac{10(k - D + I_A)^{k - D + I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k - D + I_A - n_{\mathcal{P}^*})^{k - D + I_A - n_{\mathcal{P}^*}}}$  times do
9:         initialize  $U \leftarrow \emptyset$  and  $\lambda \leftarrow \Delta$ .
10:        forall  $h \in V$ , with probability  $\frac{n_{\mathcal{P}^*}}{(k - D + I_A)}$  do add  $h$  to  $U$ . end for
11:        forall  $p \in V(\mathcal{P}) \setminus V(\mathcal{P}^*), h \in U$  do  $\lambda(p, h) \leftarrow -\infty$ . end for
12:        forall  $p \in V(\mathcal{P}^*), h \in V \setminus U$  do  $\lambda(p, h) \leftarrow -\infty$ . end for
13:        if  $\text{ManySingles}(n_E, n_{\mathcal{P}^*}, n_{\mathcal{P}}, \mathcal{P}, H, \lambda, I_F, I_A, D)$  returns  $s' > s - v(n_E + k - D)$ 
14:          then  $s \leftarrow s' + v(n_E + k - D)$ . end if
15:        end for
16:      end if
17:    end for
18:  if  $s \neq -\infty$  then return  $s$ . else reject. end if

```

negative integer weights, where \mathcal{P} is a set of bounded treewidth graphs. Approx uses $O(3.698^{k-D+I_A} \lfloor \frac{1}{\epsilon} \rfloor \log(\lfloor \frac{1}{\epsilon} \rfloor) |V|^{tw+O(1)} k^{O(1)})$ time and $O(\lfloor \frac{1}{\epsilon} \rfloor |V|^{tw+O(1)} k^{O(1)})$ space. Its running time for ANQ_I is $O^*(2^{k+I_A-D} W)$, and for TFNQ_I , $O^*(2^{k-D} W)$.

References

1. Betzler, N., Bevern, R., Fellows, M.R., Komusiewicz, C., Niedermeier, R.: Parameterized algorithmics for finding connected motifs in biological networks. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 8(5), 1296–1308 (2011)
2. Betzler, N., Fellows, M.R., Komusiewicz, C., Niedermeier, R.: Parameterized algorithms and hardness results for graph motif problems. In: *CPM*. pp. 31–43 (2008)
3. Björklund, A.: Determinant sums for undirected hamiltonicity. In: *FOCS*. pp. 173–182 (2010)
4. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Narrow sieves for parameterized paths and packings. *CoRR* abs/1007.1161 (2010)
5. Björklund, A., Kaski, P., Kowalik, L.: Probably optimal graph motifs. In: *STACS*. pp. 20–31 (2013)
6. Blin, G., Sikora, F., Vialette, S.: Querying graphs in protein-protein interactions networks using feedback vertex set. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 7(4), 628–635 (2010)
7. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* 25(6), 1305–1317 (1996)
8. Bodlaender, H.L., Koster, A.M.C.A.: Combinatorial optimization on graphs of bounded treewidth. *Comput. J.* 51(3), 255–269 (2008)
9. Bruckner, S., Hüffner, F., Karp, R.M., Shamir, R., Sharan, R.: Topology-free querying of protein interaction networks. *J. Comput. Biol.* 17(3), 237–252 (2010)

10. Chen, J., Kneis, J., Lu, S., Molle, D., Richter, S., Rossmannith, P., Sze, S., Zhang, F.: Randomized divide-and-conquer: Improved path, matching, and packing algorithms. *SIAM J. on Computing* 38(6), 2526–2547 (2009)
11. Dondi, R., Fertin, G., Vialette, S.: Maximum motif problem in vertex-colored graphs. In: *CPM*. pp. 388–401 (2011)
12. Dost, B., Shlomi, T., Gupta, N., Ruppim, E., Bafna, V., Sharan, R.: Qnet: a tool for querying protein interaction networks. *J. Comput. Biol.* 15(7), 913–925 (2008)
13. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness ii: on completeness for $w[1]$. *Theor. Comput. Sci.* 141(1-2), 109–131 (1995)
14. Fellows, M.R., Fertin, G., Hermelin, D., Vialette, S.: Sharp tractability borders for finding connected motifs in vertex-colored graphs. In: *ICALP*. pp. 340–351 (2007)
15. Fionda, V., Palopoli, L.: Biological network querying techniques: Analysis and comparison. *J. Comput. Biol.* 18(4), 595–625 (2011)
16. Fomin, F.V., Lokshtanov, D., Raman, V., Saurabh, S., Rao, B.V.R.: Faster algorithms for finding and counting subgraphs. *J. Com. Sys. Sci.* 78(3), 698–706 (2012)
17. Garey, M.R., Johnson, D.S.: *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, New York (1979)
18. Guillemot, S., Sikora, F.: Finding and counting vertex-colored subtrees. In: *MFCS*. pp. 405–416 (2010)
19. Harvey, D., Roche, D.S.: An in-place truncated fourier transform and applications to polynomial multiplication. In: *ISSAC*. pp. 325–329 (2010)
20. Hüffner, F., Wernicke, S., Zichner, T.: Algorithm engineering for color-coding with applications to signaling pathway detection. *Algorithmica* 52(2), 114–132 (2008)
21. Kellerer, H and, P.U., Pisinger, D.: *Knapsack problems*. Springer, Berlin (2004)
22. Koutis, I.: Faster algebraic algorithms for path and packing problems. In: *ICALP*. pp. 575–586 (2008)
23. Koutis, I.: Constrained multilinear detection for faster functional motif discovery. *Inf. Process. Lett.* 112(22), 889–892 (2012)
24. Koutis, I., Williams, R.: Limits and applications of group algebras for parameterized problems. In: *ICALP*. pp. 653–664 (2009)
25. Lacroix, V., Fernandes, C.G., Sagot, M.F.: Motif search in graphs: Application to metabolic networks. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 3(4), 360–368 (2006)
26. Misra, N., Philip, G., Raman, V., Saurabh, S., Sikdar, S.: FPT algorithms for connected feedback vertex set. *J. Comb. Optim.* 24(2), 131–146 (2012)
27. Nederlof, J.: Fast polynomial-space algorithms using möbius inversion: Improving on steiner tree and related problems. In: *ICALP*. pp. 713–725 (2009)
28. Pinter, R.Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, M.: Alignment of metabolic pathways. *Bioinformatics* 21(16), 3401–3408 (2005)
29. Pinter, R.Y., Zehavi, M.: Algorithms for topology-free and alignment queries. *Technion Technical Reports CS-2012-12* (2012)
30. Pinter, R.Y., Zehavi, M.: Partial information network queries. In: *IWOCA*. pp. 362–275 (2013)
31. Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.* 27(4), 701–717 (1980)
32. Shlomi, T., Segal, D., Ruppim, E., Sharan, R.: Qpath: a method for querying pathways in a protein-protein interaction networks. *BMC Bioinform.* 7, 199 (2006)
33. Sikora, F.: An (almost complete) state of the art around the graph motif problem. *Université Paris-Est Technical reports* (2012)
34. Zippel, R.: Probabilistic algorithms for sparse polynomials. In: *EUROSAM*. pp. 216–226 (1979)

A Proofs Omitted from Section 3.1

A.1 Proof of Lemma 1

Let s be the score of a solution $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ for the input. We define a quad $(T, f_{gra}, f_{nod}, f_{con})$ as follows.

1. $V_T = \mathcal{P}$.
2. $E_T = \{\{P_i, P_j\} : 1 \leq i < j \leq t, \exists h \in V_S^i, h' \in V_S^j \text{ such that } \{h, h'\} \in E_S\}$.
3. $T = (V_T, E_T)$ rooted at P_1 . Note that T is a tree since $(V_S^1, \dots, V_S^{t+1})$ is a partition of V_S , and any cycle in S is completely contained in the subgraph induced by V_S^i , for some $1 \leq i \leq t$.
4. $f_{gra} : \mathcal{P} \rightarrow \mathcal{P}$ is the identity function.
5. $\forall i \in \{1, \dots, t\}, p \in V_i: f_{nod}(P_i, p) = h_i(p)$.
6. $\forall i \in \{1, \dots, t\}, p \in V_i: f_{con}(P_i, p) = \{(P_j, p') : 1 \leq j \leq t, P_i \text{ is the father of } P_j \text{ in } T, \{h_i(p), h_j(p')\} \in E_S\}$.

Let r be a node in V_S^1 . We get that f_{gra} is injective, and the first part of Condition 4a in Definition 1 is fulfilled. Since S is a subgraph of H , we get that Condition 2 and the second part of Condition 4a in Definition 1 are fulfilled. Since h_i complies with Δ , for all $1 \leq i \leq t$, we get that Condition 3 in Definition 1 is fulfilled. Since $(V_S^1, \dots, V_S^{t+1})$ is a partition of V_S , and h_i is an isomorphism from P_i to the subgraph of H induced by V_S^i , for all $1 \leq i \leq t$, we get that Conditions 1 and 4b in Definition 1 are fulfilled and f_{nod} is injective. Moreover, since any cycle in S is completely contained in the subgraph induced by V_S^i , for some $1 \leq i \leq t$, we get that Condition 5 in Definition 1 is fulfilled. Thus, $(T, f_{gra}, f_{nod}, f_{con})$ is an r -good quad such that f_{gra} and f_{nod} are injective. By Definition 2, $(T, f_{gra}, f_{nod}, f_{con})$ has the same score s as the solution. Since $|V_T \cup \text{dom}(f_{nod})| = |L|$, we can choose a bijection $\ell : V_T \cup \text{dom}(f_{nod}) \rightarrow L$, and get that $(T, f_{gra}, f_{nod}, f_{con}, \ell) \in \text{Cor}(r, s)$.

Now, consider some $r \in V$ and $s \in \mathbb{N}^0$ such that there exists $(T, f_{gra}, f_{nod}, f_{con}, \ell) \in \text{Cor}(r, s)$. We define $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ as follows.

1. $\forall i \in \{1, \dots, t\}$:
 - (a) Since $|V_T| = t$, and f_{gra} is injective, we can let v denote the unique node in V_T satisfying $f_{gra}(v) = P_i$.
 - (b) $\forall p \in V_i: h_i(p) = f_{nod}(v, p)$.
2. $V_S^{t+1} = \emptyset$.
3. $V_S = \text{ima}(f_{nod})$.
4. $E_S = \{\{h, h'\} : \exists (v, p), (u, p') \in \text{dom}(f_{nod}) \text{ such that } f_{nod}(v, p) = h, f_{nod}(u, p') = h' \text{ and } [(u, p') \in f_{con}(v, p) \text{ or } (v = u \text{ and } \{p, p'\} \in E(f_{gra}(v)))]\}$.
5. $S = (V_S, E_S)$.

By Condition 2 and the second part of Condition 4a in Definition 1, we get that S is a subgraph of H . Since f_{nod} is injective, we get that $(V_S^1, \dots, V_S^{t+1})$ is a partition of V_S . Since f_{nod} is injective and by Condition 2 in Definition 1, we get that h_i is an isomorphism from P_i to the subgraph of S induced by V_S^i , for all $1 \leq i \leq t$. By Condition 3 in Definition 1, we get that h_i complies with Δ , for all $1 \leq i \leq t$. By the first part of Condition 4a and Condition 5 in Definition 1, we get that any cycle in S is completely contained in the subgraph induced by V_S^i , for some $1 \leq i \leq t$. We get that $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ is a solution for the input, which, by Definition 2, has the same score s as $(T, f_{gra}, f_{nod}, f_{con}, \ell)$. \square

A.2 Proof of Lemma 2

We now prove the first part of the lemma, stating that different correct solutions are associated with different monomials.

Proof. Consider some $(T, f_{gra}, f_{nod}, f_{con}, \ell), (T', f'_{gra}, f'_{nod}, f'_{con}, \ell') \in Cor(r, s)$ satisfying $(T, f_{gra}, f_{nod}, f_{con}, \ell) \not\equiv (T', f'_{gra}, f'_{nod}, f'_{con}, \ell')$.

We first assume that $(T, f_{gra}) \not\equiv (T', f'_{gra})$.¹

If $f_{gra}(\text{root}(T)) \neq f'_{gra}(\text{root}(T'))$, then there are *different* nodes $p^* \in V(f_{gra}(\text{root}(T)))$ and $p^{*'} \in V(f'_{gra}(\text{root}(T')))$, such that $f_{nod}(\text{root}(T), p^*) = f'_{nod}(\text{root}(T'), p^{*'}) = r$. Since f_{nod} and f'_{nod} are injective, we get that $(\prod_{(v,p) \in \text{dom}(f_{nod})} y_{p, f_{nod}(v,p)}) \neq \prod_{(v,p) \in \text{dom}(f'_{nod})} y_{p, f'_{nod}(v,p)}$, and thus $m(T, f_{gra}, f_{nod}, f_{con}, \ell) \neq m(T', f'_{gra}, f'_{nod}, f'_{con}, \ell')$. Thus, we next also assume that $f_{gra}(\text{root}(T)) = f'_{gra}(\text{root}(T'))$.

Since f_{gra} and f'_{gra} are injective, $|V_T| = |V_{T'}| = t$, $(T, f_{gra}) \not\equiv (T', f'_{gra})$ and $f_{gra}(\text{root}(T)) = f'_{gra}(\text{root}(T'))$, we can assume WLOG that there is $v^* \in V_T$ which has a son u^* in T , such that the unique node $v^{*'} \in T'$ satisfying $f_{gra}(v^*) = f'_{gra}(v^{*'})$ does not have a son $u^{*'}$ satisfying $f_{gra}(u^*) = f'_{gra}(u^{*'})$. Moreover, we can denote by $u^{*'}$ the unique node in $V_{T'}$ satisfying $f_{gra}(u^*) = f'_{gra}(u^{*'})$, and denote by p^* and $p^{*'}$ the unique nodes in $V(\mathcal{P})$ satisfying $p^* \in V(f_{gra}(v))$, $p^{*'}$ $\in V(f_{gra}(u^*))$ and $(u^*, p^*) \in f_{con}(v^*, p^*)$.

Since f_{nod} and f'_{nod} are injective, we have that if $f_{nod}(v^*, p^*) \neq f'_{nod}(v^{*'}, p^{*'})$ or $f_{nod}(u^*, p^{*'}) \neq f'_{nod}(u^{*'}, p^{*'})$, then $(\prod_{(v,p) \in \text{dom}(f_{nod})} y_{p, f_{nod}(v,p)}) \neq \prod_{(v,p) \in \text{dom}(f'_{nod})} y_{p, f'_{nod}(v,p)}$, and thus $m(T, f_{gra}, f_{nod}, f_{con}, \ell) \neq m(T', f'_{gra}, f'_{nod}, f'_{con}, \ell')$. Thus, we next also assume that $f_{nod}(v^*, p^*) = f'_{nod}(v^{*'}, p^{*'})$ and $f_{nod}(u^*, p^{*'}) = f'_{nod}(u^{*'}, p^{*'})$. We get that $(\prod_{(v,p) \in \text{dom}(f_{nod}), (u,p') \in f_{con}(v,p)} y_{f_{nod}(v,p), f_{nod}(u,p')}) \neq \prod_{(v,p) \in \text{dom}(f'_{nod}), (u,p') \in f'_{con}(v,p)} y_{f'_{nod}(v,p), f'_{nod}(u,p')}$, and thus again we have that $m(T, f_{gra}, f_{nod}, f_{con}, \ell) \neq m(T', f'_{gra}, f'_{nod}, f'_{con}, \ell')$.

We now assume that $(T, f_{gra}) \equiv (T', f'_{gra})$, which, WLOG, allows us to assume that $T = T'$ and $f_{gra} = f'_{gra}$.

If $f_{nod} \neq f'_{nod}$, then $(\prod_{(v,p) \in \text{dom}(f_{nod})} y_{p, f_{nod}(v,p)}) \neq \prod_{(v,p) \in \text{dom}(f'_{nod})} y_{p, f'_{nod}(v,p)}$, and thus $m(T, f_{gra}, f_{nod}, f_{con}, \ell) \neq m(T, f_{gra}, f'_{nod}, f'_{con}, \ell')$. Thus, we next also assume that $f_{nod} = f'_{nod}$.

If $f_{con} \neq f'_{con}$, then $(\prod_{(v,p) \in \text{dom}(f_{nod}), (u,p') \in f_{con}(v,p)} y_{f_{nod}(v,p), f_{nod}(u,p')}) \neq \prod_{(v,p) \in \text{dom}(f'_{nod}), (u,p') \in f'_{con}(v,p)} y_{f'_{nod}(v,p), f'_{nod}(u,p')}$ (since f_{nod} is injective), and thus $m(T, f_{gra}, f_{nod}, f_{con}, \ell) \neq m(T, f_{gra}, f_{nod}, f'_{con}, \ell')$. Thus, we next also assume that $f_{con} = f'_{con}$.

Since $(T, f_{gra}, f_{nod}, f_{con}, \ell) \not\equiv (T, f_{gra}, f_{nod}, f_{con}, \ell')$, we get that $\ell \neq \ell'$. Then, since f_{gra} and f_{nod} are injective, we get that $(\prod_{v \in V_T} z_{f_{gra}(v), \ell(v)} \prod_{(v,p) \in \text{dom}(f_{nod})} z_{f_{nod}(v,p), \ell(v,p)}) \neq \prod_{v \in V_T} z_{f_{gra}(v), \ell'(v)} \prod_{(v,p) \in \text{dom}(f_{nod})} z_{f_{nod}(v,p), \ell'(v,p)}$, and thus $m(T, f_{gra}, f_{nod}, f_{con}, \ell) \neq m(T, f_{gra}, f_{nod}, f_{con}, \ell')$. \square

We next prove the second part of the lemma, stating that a monomial of an incorrect solution represents an even number of incorrect solutions.

¹ Note that $(T, f_{gra}) \equiv (T', f'_{gra})$ iff there is an isomorphism iso between the *rooted* trees T and T' , such that $[\forall v \in V_T : f_{gra}(v) = f'_{gra}(iso(v))]$.

Proof. We prove this part by showing that there is a fixed-point-free involution (i.e., a permutation that is its own inverse) $inv : Sol(r, s) \setminus Cor(r, s) \rightarrow Sol(r, s) \setminus Cor(r, s)$ satisfying $m(sol) = m(inv(sol))$ for all $sol \in Sol(r, s) \setminus Cor(r, s)$. Note that given $a, b, c, d \in L$, we have that $(a, b) < (c, d)$ iff $a < c$ or $(a = c \wedge b < d)$.

Let $(T, f_{gra}, f_{nod}, f_{con}, \ell) \in Sol(r, s) \setminus Cor(r, s)$.

First, assume that f_{gra} is not injective. Define $swap(\ell)$ as the set $\{v^*, u^*\}$ in $\{\{v, u\} : v, u \in V_T, v \neq u, f_{gra}(v) = f_{gra}(u)\}$ minimizing $(\min\{\ell(v^*), \ell(u^*)\}, \max\{\ell(v^*), \ell(u^*)\})$ (this choice is well-defined since ℓ is bijective). Also, define $new(\ell)$ as ℓ except that $new(\ell)(v^*) = \ell(u^*)$ and $new(\ell)(u^*) = \ell(v^*)$, and $inv(T, f_{gra}, f_{nod}, f_{con}, \ell) = (T, f_{gra}, f_{nod}, f_{con}, new(\ell))$.

Note that $inv(T, f_{gra}, f_{nod}, f_{con}, \ell) \in Sol(r, s) \setminus Cor(r, s)$ and $m(T, f_{gra}, f_{nod}, f_{con}, \ell) = m(inv(T, f_{gra}, f_{nod}, f_{con}, \ell))$. Since $swap(\ell) = swap(new(\ell))$, we get that $(T, f_{gra}, f_{nod}, f_{con}, \ell) = inv(inv(T, f_{gra}, f_{nod}, f_{con}, \ell))$.

Suppose, by way of contradiction, that $(T, f_{gra}, f_{nod}, f_{con}, \ell) \equiv inv(T, f_{gra}, f_{nod}, f_{con}, \ell)$. Thus, there is an isomorphism iso between the rooted tree T and itself, such that

1. $\forall v \in V_T : f_{gra}(v) = f_{gra}(iso(v))$ and $\ell(v) = new(\ell)(iso(v))$.
2. $\forall (v, p) \in \text{dom}(f_{nod}) : f_{nod}(v, p) = f_{nod}(iso(v), p)$, $\ell(v, p) = new(\ell)(iso(v), p)$ and $\forall (u, p') : (u, p') \in f_{con}(v, p)$ iff $(iso(u), p') \in f_{con}(iso(v), p)$.

By Condition 1 and the definition of $new(\ell)$, and since ℓ is bijective, we get that $(\forall v \in V_T \setminus \{v^*, u^*\} : iso(v) = v)$, $iso(v^*) = u^*$ and $iso(u^*) = v^*$. Thus, by Condition 2, we have that there is $p^* \in V(f_{gra}(v^*))$ such that $\ell(v^*, p^*) = new(\ell)(iso(v^*), p^*) = new(\ell)(u^*, p^*) = \ell(u^*, p^*)$. Since ℓ is bijective, this is a contradiction.

Now, assume that f_{gra} is injective, and thus f_{nod} is not injective. Define $swap(\ell)$ as the set $\{(v^*, p_{v^*}), (u^*, p_{u^*})\}$ in $\{(v, p), (u, p')\} : (v, p), (u, p') \in \text{dom}(f_{nod}), (v, p) \neq (u, p'), f_{nod}(v, p) = f_{nod}(u, p')\}$ minimizing $(\min\{\ell(v^*, p_{v^*}), \ell(u^*, p_{u^*})\}, \max\{\ell(v^*, p_{v^*}), \ell(u^*, p_{u^*})\})$ (this choice is well-defined since ℓ is bijective). Also, define $new(\ell)$ as ℓ except that $new(\ell)(v^*, p_{v^*}) = \ell(u^*, p_{u^*})$ and $new(\ell)(u^*, p_{u^*}) = \ell(v^*, p_{v^*})$, and $inv(T, f_{gra}, f_{nod}, f_{con}, \ell) = (T, f_{gra}, f_{nod}, f_{con}, new(\ell))$.

Note that $inv(T, f_{gra}, f_{nod}, f_{con}, \ell) \in Sol(r, s) \setminus Cor(r, s)$ and $m(T, f_{gra}, f_{nod}, f_{con}, \ell) = m(inv(T, f_{gra}, f_{nod}, f_{con}, \ell))$. Since $swap(\ell) = swap(new(\ell))$, we get that $(T, f_{gra}, f_{nod}, f_{con}, \ell) = inv(inv(T, f_{gra}, f_{nod}, f_{con}, \ell))$.

Suppose, by way of contradiction, that $(T, f_{gra}, f_{nod}, f_{con}, \ell) \equiv inv(T, f_{gra}, f_{nod}, f_{con}, \ell)$. Again, consider a function iso defined as in the case where f_{gra} is not injective. By Condition 1 and the definition of $new(\ell)$, and since ℓ is bijective, we get that $(\forall v \in V_T : iso(v) = v)$. Thus, by Condition 2, we get that $\ell(v^*, p_{v^*}) = new(\ell)(iso(v^*), p_{v^*}) = new(\ell)(v^*, p_{v^*}) = \ell(u^*, p_{u^*})$. Since ℓ is bijective, this is a contradiction. \square

A.3 Proof of Lemma 4

Let $A \subseteq L$ and $r \in V$. In this proof, when we refer to a variable that is not x , we refer to the *value* that is assigned to this variable. We also assume an order $<$ on $\{0\} \cup V$, such that 0 is the smallest element.

The proof is organized as follows. We first give some definitions used only in this proof. We then present the matrices of the dynamic programming, and show how to compute their cells. Finally, we conclude the correctness of the lemma.

A.3.1 Definitions

Treewidth is a well-studied parameter which intuitively indicates how close a graph is to being a tree. For example, a tree has treewidth 1 and an n -node clique has treewidth $n-1$. Formally, the treewidth of a graph is defined as follows.

Definition 5. A tree decomposition of a graph $G = (V_G, E_G)$ is a tree $T = (V_T, E_T)$, such that

1. Each $X \in V_T$ represents a subset of V_G , which we denote by $\text{set}(X)$.
2. For each $v \in V_G$, there is $X \in V_T$ such that $v \in \text{set}(X)$.
3. For each $\{v, u\} \in E_G$, there is $X \in V_T$ such that $\{v, u\} \subseteq \text{set}(X)$.
4. Let $X, Y \in V_T$. If both $\text{set}(X)$ and $\text{set}(Y)$ contain a node $v \in V_G$, then all nodes $Z \in V_T$ on the path between X and Y satisfy $v \in \text{set}(Z)$.

The width of T is $\max_{X \in V_T} \{|\text{set}(X)|\} - 1$. The treewidth of G is the minimum width among all possible tree decompositions of G .

A nice tree decomposition is a form of a tree decomposition that is easier to handle in dynamic programming-based algorithms. It is defined as follows.

Definition 6. A tree decomposition $T = (V_T, E_T)$ of a graph G , rooted at some node, is nice if each node $X \in V_T$ is of one of the following types.

1. Leaf: X is a leaf of T and $|\text{set}(X)| = 0$.
2. Join: X has exactly two sons Y and Z , and $\text{set}(X) = \text{set}(Y) = \text{set}(Z)$. We say that X joins Y and Z .
3. Introduce: X has exactly one son Y , and there is a node v such that $\text{set}(X) = \text{set}(Y) \cup \{v\}$. We say that X introduces v .
4. Forget: X has exactly one son Y , and there is a node v such that $\text{set}(X) = \text{set}(Y) \setminus \{v\}$. We say that X forgets v .

For any graph in \mathcal{P} , we can compute a minimum width tree decomposition in $O(k)$ time [7]. Then, for any graph in \mathcal{P} , we can compute a nice tree decomposition with the same width, such that $\text{set}(\text{root}(T)) = \emptyset$, in $O(k)$ time [8]. Thus, for all $1 \leq i \leq t$, we next assume that we have a minimum width tree decomposition $TD_i = (VD_i, ED_i)$ of P_i that is nice.

The following two definitions will be immediately used in defining partial solutions for the dynamic programming.

Definition 7. Let $1 \leq i \leq t, p^* \in V_i, h^* \in V$ and $X \in VD_i$. We say that a function $\alpha: \text{set}(X) \rightarrow V$ is a (p^*, h^*, X) -mapping if:

1. If $p^* \in \text{set}(X) : \alpha(p^*) = h^*$.
2. $\forall p \in \text{set}(X) \setminus \{p^*\} : \alpha(p) \neq r$.
3. $\forall p \in \text{set}(X) : \Delta(p, \alpha(p)) \neq -\infty$.
4. $\forall p, p' \in \text{set}(X)$ such that $\{p, p'\} \in E_i : \{\alpha(p), \alpha(p')\} \in E$.

Definition 8. Let $1 \leq i \leq t$ and $X \in VD_i$. Define $P_i(X)$ as the subgraph of P_i induced by $\{p \in V_i : X \text{ forgets } p \text{ or has a descendant in } TD_i \text{ that forgets } p\}$.

We now define the partial solutions. For the sake of clarity, we divide the conditions into three sets. Informally, the first set (i.e., Conditions 1–4) modifies our definition of a quad (see the second paragraph in Section 3.1.1); the second set (i.e., Conditions 5–11) modifies Definition 1 (specifically, Conditions 5–6 modify Condition 1 in Definition 1, Conditions 7–8 modify Condition 2 in Definition 1, and Conditions 9–11 are Conditions 3–5 in Definition 1); and the third set (i.e., Condition 12) modifies Definition 3.

Definition 9. Let $1 \leq i \leq t$, $p^* \in V_i$, $h^* \in V$, $X \in VD_i$, a (p^*, h^*, X) -mapping α , $1 \leq t^* \leq t$ such that $(h^* = r \rightarrow t^* = t)$, and $1 \leq k^* \leq k$.

A pentuple $(T, f_{gra}, f_{nod}, f_{con}, \ell)$ is a $(p^*, h^*, X, \alpha, t^*, k^*)$ -solution if:

1. $T = (V_T, E_T)$ is a rooted tree on t^* nodes.
2. $f_{gra} : V_T \rightarrow \mathcal{P} \cup P_i(X)$, and $k^* = \sum_{v \in V_T \setminus \text{root}(T)} |V(f_{gra}(v))| + |V_i|$.
3. If $P_i \neq P_i(X) : (\forall v \in V_T : f_{gra}(v) = P_i(X) \text{ iff } v = \text{root}(T))$.
4. $f_{nod} : Y \rightarrow V$, $f_{con} : Y \rightarrow 2^Y$, where $Y = \{(v, p) : v \in V_T, p \in V(f_{gra}(v))\}$.
5. If $(\text{root}(T), p^*) \in \text{dom}(f_{nod}) : f_{nod}(\text{root}(T), p^*) = h^*$.
6. $\forall (v, p) \in \text{dom}(f_{nod}) \setminus \{(\text{root}(T), p^*)\} : f_{nod}(v, p) \neq r$.
7. $\forall v \in V_T, \{p, p'\} \in E(f_{gra}(v)) : \{f_{nod}(v, p), f_{nod}(v, p')\} \in E$.
8. $\forall p$ s.t. $(\text{root}(T), p) \in \text{dom}(f_{con}), p' \in \text{set}(X)$ s.t. $\{p, p'\} \in E_i : \{f_{nod}(\text{root}(T), p), \alpha(p')\} \in E$.
9. $\forall (v, p) \in \text{dom}(f_{nod}) : \Delta(p, f_{nod}(v, p)) \neq -\infty$.
10. $\forall (v, p) \in \text{dom}(f_{con}), (u, p') \in f_{con}(v, p) :$
 - (a) v is the father of u in T , and $\{f_{nod}(v, p), f_{nod}(u, p')\} \in E$.
 - (b) $\forall (u', p'') \in f_{con}(v, p) \setminus \{(u, p')\} : f_{nod}(u, p') \neq f_{nod}(u', p'')$.
11. $\forall u \in V_T \setminus \{\text{root}(T)\} : |\{(v, p, p') : (u, p') \in f_{con}(v, p)\}| = 1$.
12. $\ell : (V_T \setminus \{\text{root}(T)\}) \cup \text{dom}(f_{nod}) \rightarrow A$.

The score and the monomial of a $(p^*, h^*, X, \alpha, t^*)$ -solution are defined as in Definitions 2 and 4.

A.3.2 The Matrices

We use the following two matrices.

1. $\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*]$ for all $1 \leq i \leq t, p^* \in V_i, h^* \in V, X \in VD_i$, a (p^*, h^*, X) -mapping α , $1 \leq t^* \leq t$ such that $(r = h^* \rightarrow t^* = t)$, and $1 \leq k^* \leq k$.
2. $\text{CON}[h_0, h^*, t^*, k^*]$ for all $h_0 \in \{0\} \cap V, h^* \in V, 1 \leq t^* \leq t$, and $0 \leq k^* \leq k$.

$\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*]$ holds the polynomial $\sum_{(p^*, h^*, X, \alpha, t^*)\text{-solution } sol} m(sol)$, whose

only variable is x , and $\text{CON}[h_0, h^*, t^*, k^*]$ holds the following polynomial whose only variable is x .

$$\begin{aligned}
& \sum_{n=1}^{t^*-1} \sum_{\text{index}(1), \dots, \text{index}(n) \in \{1, \dots, t\}} \sum_{p_1, \dots, p_n \text{ s.t. } (\forall i \in \{1, \dots, n\}: p_i \in V_{\text{index}(i)})} \\
& \sum_{h_1, \dots, h_n \text{ s.t. } [\forall i \in \{1, \dots, n\}: \{h^*, h_i\} \in E \text{ and } h_{i-1} < h_i]} \\
& \sum_{1 \leq t_1, \dots, t_n \text{ s.t. } \sum_{i=1}^n t_i = t^* - 1} \sum_{1 \leq k_1, \dots, k_n \text{ s.t. } \sum_{i=1}^n k_i = k^*} \\
& \sum_{\text{sol}_1, \dots, \text{sol}_n \text{ s.t. } (\forall i \in \{1, \dots, n\}: \text{sol}_i \text{ is a } (p_i, h_i, \text{root}(TD_{\text{index}(i)}), f: \emptyset \rightarrow \emptyset, t_i, k_i) \text{-solution})} \\
& \prod_{i=1}^n [x^{w(\{h^*, h_i\})} \cdot (\sum_{l \in A} z_{P_{\text{index}(i)}, l}) \cdot y_{h^*, h_i} \cdot m(\text{sol}_i)].
\end{aligned}$$

A.3.3 Computation

Base Cases – MAP:

1. If X is a leaf, $t^* = 1$, and $k^* = |V_i|$: $\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*] = 1$.
2. If X is a leaf, and $t^* > 1$ or $k^* \neq |V_i|$: $\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*] = 0$.

Steps – MAP:

1. If X joins Y and Z : $\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*] =$

$$\sum_{t'=1}^{t^*} \sum_{k'=|V_i|}^{k^*} (\text{MAP}[i, p^*, h^*, Y, \alpha, t', k'] \cdot \text{MAP}[i, p^*, h^*, Z, \alpha, t^* - t' + 1, k^* - k' + |V_i|]).$$
2. If X introduces p :
 - (a) Denote the (only) son of X in TD_i by Y .
 - (b) Denote α restricted to $\text{set}(Y)$ by β .
 - (c) $\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*] = \text{MAP}[i, p^*, h^*, Y, \beta, t^*, k^*]$.
3. If X forgets p :
 - (a) Denote the (only) son of X in TD_i by Y .
 - (b) Denote $M = \{(p^*, h^*, Y) \text{ - mapping } \beta : (\forall p' \in X : \alpha(p') = \beta(p'))\}$.
 - (c) Denote $N = \{p' \in \text{set}(X) : \{p, p'\} \in E_i\}$.
 - (d) $\forall \beta \in M$: $\text{poly}(\beta) = x^{\Delta(p, \beta(p)) + \sum_{p' \in N} w(\{\beta(p), \beta(p')\})} y_{p, \beta(p)} (\sum_{l \in A} z_{\beta(p), l})$.
 - (e) $\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*] =$

$$\sum_{\beta \in M} [\text{poly}(\beta) \sum_{t'=1}^{t^*} \sum_{k'=|V_i|}^{k^*} (\text{MAP}[i, p^*, h^*, Y, \beta, t', k'] \cdot \text{CON}[0, \beta(p), t^* - t' + 1, k^* - k'])].$$

Base Cases – CON:

1. $t^* = 1$, and $k^* = 0$: $\text{CON}[h_0, h^*, t^*, k^*] = 1$.
2. $t^* = 1$, and $k^* > 0$: $\text{CON}[h_0, h^*, t^*, k^*] = 0$.

Step – CON:

$$\begin{aligned}
\text{CON}[h_0, h^*, t^*, k^*] &= \sum_{i=1}^t \sum_{p \in V_i} \sum_{h_0 < h \neq p \text{ s.t. } \{h^*, h\} \in E} \sum_{t'=1}^{t^*-1} \sum_{k'=|V_i|}^{k^*} [x^{w(\{h^*, h\})} \\
& (\sum_{l \in A} z_{P_i, l}) \cdot y_{h^*, h} \cdot \text{MAP}[i, p, h, \text{root}(TD_i), f: \emptyset \rightarrow \emptyset, t', k'] \cdot \text{CON}[h, h^*, t^* - t', k^* - k']].
\end{aligned}$$

Order:

1. For $t^* = 1, \dots, t$, $k^* = 0, \dots, k$, $i = 1, \dots, t$, each $p^* \in V_i$ and $h^* \in V$:
 - (a) For $h_0 = \max(V), \dots, 0$: Compute $\text{CON}[h_0, h^*, t^*, k^*]$.
 - (b) For each $X \in VD_i$, where TD_i is traversed by using a postorder:
 - i. For each (p^*, h^*, X) -mapping α : Compute $\text{MAP}[i, p^*, h^*, X, \alpha, t^*, k^*]$.

A.3.4 Concluding the Lemma

It is straightforward to verify the correctness of the computation of the matrices. We get that $P(r) = \sum_{1 \leq i \leq t, p \in V_i, l \in A} z_{P_i, l} \text{MAP}[i, p, r, \text{root}(TD_i), f : \emptyset \rightarrow \emptyset, t, k]$.

Each cell in the matrices holds a polynomial of degree at most $W|V|^{O(1)}$ and coefficients polynomial in the input size. Since the matrices contain $O(|V|^{tw+O(1)} k^{O(1)})$ cells, we get that the space complexity of the entire computation is $O(W|V|^{tw+O(1)} k^{O(1)})$. Since we can compute a multiplication of two polynomials, such that each has a degree of at most $W|V|^{O(1)}$ and coefficients polynomial in the input size, in $O(W \log W|V|^{O(1)} k^{O(1)})$ time [19], we get that the running time of the computation of each cell is $O(W \log W|V|^{O(1)} k^{O(1)})$. Thus, the running time of the entire computation is $O(W \log W|V|^{tw+O(1)} k^{O(1)})$. \square

B Proof of Theorem 1

We prove a slightly stronger theorem, since this will be necessary for the proof of Theorem 2 (see Section F).

Theorem 3. *Algorithm Exact solves PINQ_I in $O(3.698^{k-D+I_A} W \log W|V|^{tw+O(1)} k^{O(1)})$ time and $O(W|V|^{tw+O(1)} k^{O(1)})$ space, handling inputs with integer weights, where \mathcal{P} is a set of bounded treewidth graphs. Its success probability is $\frac{9}{10}$. Its running time for ANQ_I is $O^*(2^{k+I_A-D} W)$, and for TFNQ_I , $O^*(2^{k-D} W)$.*

If there is no solution, algorithm Exact rejects (with probability 1). Else, if the number of input indels is minimal, it does not return a score higher than OPT (with probability 1).

First, assume that the number of input indels is minimal. By Lemmas 7 and 8, all the calls to `FewSingles` and `ManySingles` executed by the calls to `Exact` in Step 1 reject, and thus `Exact` does not reject in Step 1. If there is no solution, then by Lemmas 7 and 8, all the calls to `FewSingles` and `ManySingles` reject, and thus `Exact` rejects. Thus, next also assume that there is a solution.

Let $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ be some solution of maximum score OPT . By Lemmas 7 and 8, for any $n_E \in \{0, \dots, |E|\}$, no call to `FewSingles` and `ManySingles` in an iteration corresponding to n_E returns a score higher than $\text{OPT} - v(n_E + k - D)$. Consider the iteration of Step 3 that corresponds to $n_{\mathcal{P}}, n_{\mathcal{P}^*}$ and n_E that are the number of nonempty sets in $\{V_S^1, \dots, V_S^t\}$, the number of one-node sets in $\{V_S^1, \dots, V_S^t\}$ and the number of edges in E_S , respectively. If $2^{n_{\mathcal{P}^*}} \leq \frac{(k-D+I_A)^{k-D+I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}}$, then by Lemma 7, the call to `FewSingles` returns $\text{OPT} - v(n_E + k - D)$ with probability $\geq \frac{9}{10}$, and thus `Exact` returns OPT with probability $\geq \frac{9}{10}$. Next assume that $2^{n_{\mathcal{P}^*}} > \frac{(k-D+I_A)^{k-D+I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}}$.

Denote $A = \{h \in V_S : \exists i \in \{1, \dots, t\}, p \in V(\mathcal{P}^*) \text{ s.t. } h_i(p) = h\}$ and $B = V_S \setminus (A \cup V_S^{t+1})$. Note that $|A| = n_{\mathcal{P}^*}$ and $|B| = k - D + I_A - n_{\mathcal{P}^*}$. The probability that there is an iteration of Step 7 where we choose U such that $(A \subseteq U \text{ and } B \subseteq V \setminus U)$ and then `ManySingles` returns the maximum score of a solution for its input is

$$1 - \left(1 - \frac{9}{10} \left(\frac{n_{\mathcal{P}^*}}{(k-D+I_A)}\right)^{|A|} \left(1 - \frac{n_{\mathcal{P}^*}}{(k-D+I_A)}\right)^{|B|}\right)^{\frac{10(k-D+I_A)^{k-D+I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}}} =$$

$$1 - \left[\left(1 - \frac{9n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}}{10(k-D+I_A)^{k-D+I_A}}\right)^{\frac{10(k-D+I_A)^{k-D+I_A}}{9n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}}}\right]^9$$

$$\geq 1 - e^{-9} \geq \frac{9}{10}$$

In such an iteration, the score returned by `ManySingles` is $OPT - v(n_E + k - D)$, and thus `Exact` returns OPT with probability $\geq \frac{9}{10}$.

Now assume that there are $i_F \leq I_F$, $i_A \leq I_A$ and $d \leq D$ s.t. $i_F + i_A + d < I_F + I_A + D$, and $\mathcal{T}' = (\mathcal{P}, H, \Delta, i_F, i_A, d)$ has a solution. Choose those that minimize $i_F + i_A + d$ (if there are several possibilities, choose one arbitrarily). By the previous case that we proved, with probability $\geq \frac{9}{10}$ the call `Exact`($\mathcal{P}, H, \Delta, i_F, i_A, d$) executed in Step 1 returns some score, and thus `Exact` rejects.

Next, we prove the running time and space complexity of `Exact`. Since the space complexity of `FewSingles` and `ManySingles` is $O(W|V|^{tw+O(1)}k^{O(1)})$, and the space complexity of all the other computations of `Exact` is $O(|V|^{O(1)}k^{O(1)})$, we get that the space complexity of `Exact` is $O(W|V|^{tw+O(1)}k^{O(1)})$. By the pseudocode of `Exact` and the running times of `FewSingles` and `ManySingles`, we get that the running time of `Exact` is $O(xW \log W|V|^{tw+O(1)}k^{O(1)})$, where

$$x = 2^{k-D+I_A} \cdot \max_{n_{\mathcal{P}^*} = \max\{0, k^*-D\}, \dots, \min\{k^*, k-D\}} \left\{ 2^{\min\{t, (k-D+I_A+n_{\mathcal{P}^*})/2\}-n_{\mathcal{P}^*}} \cdot \frac{(k-D+I_A)^{k-D+I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}} \right\}$$

For `TFNQI`, $t = k = k^*$ and $I_A = 0$, and for `ANQI`, $t = 1$ and $k^* = 0$. In both these cases, it is clear that $x = O(2^{k-D+I_A})$.

For $n_{\mathcal{P}^*} \leq 0.7729(k-D+I_A)$, we have that

$$\begin{aligned} & 2^{\min\{t, (k-D+I_A+n_{\mathcal{P}^*})/2\}-n_{\mathcal{P}^*}} \min\left\{ 2^{n_{\mathcal{P}^*}}, \frac{(k-D+I_A)^{k-D+I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}} \right\} \\ & \leq 2^{\frac{k-D+I_A+n_{\mathcal{P}^*}}{2}-n_{\mathcal{P}^*}} \leq 2^{\frac{1.7729(k-D+I_A)}{2}} \leq 1.84863^{k-D+I_A} \end{aligned}$$

For $n_{\mathcal{P}^*} \geq 0.7729(k-D+I_A)$, we have that

$$\begin{aligned} & 2^{\min\{t, (k-D+I_A+n_{\mathcal{P}^*})/2\}-n_{\mathcal{P}^*}} \min\left\{ 2^{n_{\mathcal{P}^*}}, \frac{(k-D+I_A)^{k-D+I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}} \right\} \\ & \leq 2^{(k-D+I_A-n_{\mathcal{P}^*})/2} \frac{(k-D+I_A)^{k-D+I_A}}{n_{\mathcal{P}^*}^{n_{\mathcal{P}^*}} (k-D+I_A-n_{\mathcal{P}^*})^{k-D+I_A-n_{\mathcal{P}^*}}} \\ & \leq 2^{0.11355(k-D+I_A)} \frac{(k-D+I_A)^{k-D+I_A}}{(0.7729(k-D+I_A))^{0.7729(k-D+I_A)} (0.2271(k-D+I_A))^{0.2271(k-D+I_A)}} \\ & \leq 1.84863^{k-D+I_A} \end{aligned}$$

We get that $x = O(2^{k-D+I_A} \cdot 1.84863^{k-D+I_A}) = O(3.69726^{k-D+I_A})$. \square

C FewSingles and ManySingles: Potential Solution Structure

In this section we give some definitions used in determining the potential solutions of both `FewSingles` and `ManySingles`.

C.1 Notation

Assume an order $<$ on V . We refer to a walk w in H as a function, where $w(i)$ is the i^{th} node on w . Given $h, h' \in V$, $\text{walks}(h, h')$ includes all walks w from $\min(h, h')$ to $\max(h, h')$ in H , such that $|\text{dom}(w)| \leq I_A + 2$ and $(h = h' \rightarrow 4 \leq |\text{dom}(w)|)$. Define $\text{walks} = \bigcup_{h, h' \in V} \text{walks}(h, h')$. Given $p, p' \in V_i$, for some $1 \leq i \leq t$, $\text{paths}(p, p')$ includes all simple paths or cycles in P_i between p and p' , having only degree-2 (in P_i) internal nodes. An example for this notation is given in Fig. 4.

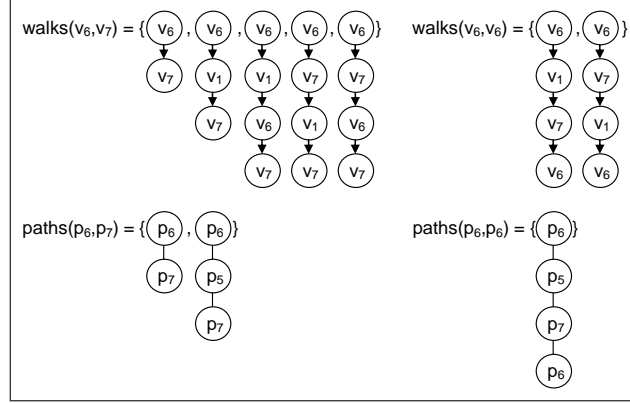


Fig. 4. An illustration of the notation given in Section C.1, referring to the input in Fig. 2(A), where $I_A = 1$ is replaced by $I_A = 2$.

C.2 An r -Good Pentuple

We now extend the r -good quads defined in Section 3.1.1 to allow indels, resulting in r -good pentuples. This section presents the details concerning the first main modification noted in Section 3.2.

A pentuple $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ refers to a rooted tree $T = (V_T, E_T)$, and functions $f_{gra}, f_{nod}, f_{pat}$ and f_{con} as follows.

1. $f_{gra} : V_T \rightarrow \mathcal{P}$.
2. $\text{dom}(f_{nod}) \subseteq \{(v, p) : v \in V_T, p \in V(f_{gra}(v))\}$, and $\text{ima}(f_{nod}) \subseteq V$.
3. $f_{pat} : \{(v, \{p, p'\}, path) : (v, p), (v, p') \in \text{dom}(f_{nod}), path \in \text{paths}(\{p, p'\})\}$, there is no internal node p'' on $path$ such that $(v, p'') \in \text{dom}(f_{nod})\} \rightarrow \text{walks}$.
4. $\text{dom}(f_{con}) = \text{dom}(f_{nod}) \cup \{(v, (\{p, p'\}, path, i)) : (v, \{p, p'\}, path) \in \text{dom}(f_{pat}), i \in \{2, \dots, |\text{dom}(f_{pat}(v, \{p, p'\}, path))| - 1\}\}$.

Informally, such a pentuple refers to a structure for a solution as follows. T and f_{gra} specify which graphs to choose from \mathcal{P} , and how to connect them; f_{nod} indicates how to map some of the nodes of the graphs chosen from \mathcal{P} to nodes in V (where unmapped nodes are deleted); f_{pat} indicates which nodes in V are alignment insertions; f_{con} refines our information about how the chosen graphs are connected; in particular, it indicates which nodes in V are free insertions.

To simplify the presentation, define $f_{nod\&pat}(v, x)$ for all $(v, x) \in \text{dom}(f_{con})$ as follows. $f_{nod\&pat}(v, x) = f_{nod}(v, x)$ if $(v, x) \in \text{dom}(f_{nod})$, and $f_{nod\&pat}(v, x) = (f_{pat}(v, \{p, p'\}, path))(i)$ for $x = (\{p, p'\}, path, i)$ otherwise.

The following definition concerns free insertions. It is related to the concept of *branching walks* [27], and will be immediately used in defining r -good pentuples.

Definition 10. Given $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$, $h \in V$, a rooted tree $T^{tf} = (V^{tf}, E^{tf})$, a function $f_{nod}^{tf} : V^{tf} \rightarrow V$ and an injection $f_{lea}^{tf} : \text{leaves}(T^{tf}) \rightarrow \text{dom}(f_{con})$,² we say that $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf})$ is an h -walk tree for $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ if:

1. $f_{nod}^{tf}(\text{root}(T^{tf})) = h$.

² We denote by $\text{leaves}(T^{tf})$ the set of leaves in T^{tf} excluding its root (even if $|V^{tf}| = 1$).

2. $\forall \{v, u\} \in E^{tf} : \{f_{nod}^{tf}(v), f_{nod}^{tf}(u)\} \in E$.
3. $\forall v \in V^{tf} : \text{There is no sibling } u \text{ of } v \text{ in } T^{tf} \text{ such that } f_{nod}^{tf}(v) = f_{nod}^{tf}(u)$.
4. $\forall v \in \text{leaves}(T^{tf}) : f_{nod}^{tf}(v) = f_{nod\&pat}(f_{lea}^{tf}(v))$.

We now define the pentuples corresponding to structures of potential solutions for PINQ₁. In this definition, assume that n_E and $n_{\mathcal{P}}$ are parameters given as input to FewSingles and ManySingles (see Sections 3.2 and 3.3).

Definition 11. *Given $r \in V$, a pentuple $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ is r -good if:*

1. $|\{(v, x) \in \text{dom}(f_{con}) : f_{nod\&pat}(v, x) = r\}| = |\{p \in V(f_{gra}(\text{root}(T))) : f_{nod}(\text{root}(T), p) = r\}| = 1$.
2. $\forall (v, \{p, p'\}, path) \in \text{dom}(f_{pat}) : f_{pat}(v, \{p, p'\}, path) \in \text{walks}(\{f_{nod}(v, p), f_{nod}(v, p')\})$.
3. $\forall (v, p) \in \text{dom}(f_{nod}) : \Delta(p, f_{nod}(v, p)) \neq -\infty$.
4. $\forall (v, x) \in \text{dom}(f_{con}) : f_{con}(v, x) = (T^{tf}, f_{nod}^{tf}, f_{lea}^{tf})$ such that
 - (a) $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf})$ is a $f_{nod\&pat}(v, x)$ -walk tree for $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$.
 - (b) $\forall (u, y) \in \text{ima}(f_{lea}^{tf}) : v$ is the father of u in T .
5. $\forall v \in V_T \setminus \{\text{root}(T)\} : |\{(u, y) \in \text{dom}(f_{con}) : \exists x \text{ s.t. } (v, x) \in \text{ima}(f_{lea}^{tf}), \text{ for } f_{con}(u, y) = (T^{tf}, f_{nod}^{tf}, f_{lea}^{tf})\}| = |\{x : \exists (T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \in \text{ima}(f_{con}) \text{ s.t. } (v, x) \in \text{ima}(f_{lea}^{tf})\}| = 1$.
6. $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ complies with $n_{\mathcal{P}}, n_E, I_F, I_A$ and D as follows.
 - (a) $n_{\mathcal{P}} = |V_T|$.
 - (b) $n_E = \sum_{x \in \text{dom}(f_{pat})} (|\text{dom}(f_{pat}(x))| - 1) + \sum_{(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \in \text{ima}(f_{con})} |E^{tf}|$.
 - (c) $I_F = \sum_{(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \in \text{ima}(f_{con})} |V^{tf} \setminus (\{\text{root}(T^{tf})\} \cup \text{leaves}(T^{tf}))|$.
 - (d) $I_A = \sum_{x \in \text{dom}(f_{pat})} (|\text{dom}(f_{pat}(x))| - 2)$.
 - (e) $D = k - |\text{dom}(f_{nod})|$.

Finally, we define the score of an r -good pentuple.

Definition 12. *The score of an r -good pentuple $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ is*

$$\sum_{x \in \text{dom}(f_{pat}), i \in \{1, \dots, |\text{dom}(f_{pat}(x))| - 1\}} w(\{f_{pat}(x)(i), f_{pat}(x)(i+1)\}) + \sum_{(v, p) \in \text{dom}(f_{nod})} \Delta(p, f_{nod}(v, p)) + \sum_{(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \in \text{ima}(f_{con}), \{v, u\} \in E^{tf}} w(\{f_{nod}^{tf}(v), f_{nod}^{tf}(u)\}).$$

D FewSingles: A Narrow Sieves Procedure

We now present the details concerning the procedure FewSingles (see Section 3.2). Recall that we assume that \mathcal{P} is a set of bounded treewidth graphs, the number of input indels is minimal, and the weights are nonnegative integers. Moreover, recall that the input for FewSingles is of the form $(n_E, n_{\mathcal{P}}, \mathcal{P}, H, \Delta, I_F, I_A, D)$.

D.1 Potential Solutions

Let $L = \{1, \dots, k - D + I_A + n_{\mathcal{P}}\}$ denote the set of indices used in labeling r -good pentuples, resulting in potential solutions of the same score as follows.

Definition 13. *Given an r -good pentuple $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ and $\ell : V_T \cup \text{dom}(f_{con}) \rightarrow L$, we say that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$ is an r -solution.*

We now define two sets of potential solutions. $Sol(r, s)$ contains all r -solutions $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$ of score s such that ℓ is bijective; and $Cor(r, s) = \{(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in Sol(r, s) : f_{gra} \text{ and } f_{nod\&pat} \text{ are injective}\}$.

Lemma 9. *The input has a solution of score s , such that $|E_S| = n_E$ and $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ nonempty sets, iff $\bigcup_{r \in V} Cor(r, s) \neq \emptyset$.*

Proof. Let s be the score of a solution $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ for the input, such that $|E_S| = n_E$ and $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ non-empty sets. We define a pentuple $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ as follows.

1. $V_T = \{P_i \in \mathcal{P} : V_S^i \neq \emptyset\}$.
2. $E_T = \{\{P_i, P_j\} : P_i, P_j \in V_T, i \neq j, \exists h \in V_S^i, h' \in V_S^j \text{ s.t. there is a path between } h \text{ and } h' \text{ in } S \text{ whose internal nodes are in } V_S^{t+1} \text{ (only)}\}$.
3. $T = (V_T, E_T)$ rooted at a node chosen arbitrarily from V_T .
4. $f_{gra} : \mathcal{P} \rightarrow \mathcal{P}$ is the identity function.
5. $\text{dom}(f_{nod}) = \{(P_i, p) : P_i \in V_T, p \in \text{dom}(h_i)\}$.
6. $\forall (P_i, p) \in \text{dom}(f_{nod}) : f_{nod}(P_i, p) = h_i(p)$.
7. $\forall P_i \in V_T, p, p' \in \text{dom}(h_i)$: For each $path \in \text{paths}(\{p, p'\})$ on which there is no internal node p'' s.t. $(P_i, p'') \in \text{dom}(f_{nod})$, choose a different simple path or cycle $w \in \text{walk}(\{f_{nod}(P_i, p), f_{nod}(P_i, p')\})$ in S satisfying $[\forall j \in \{2, \dots, |\text{dom}(w)| - 1\} : w(j) \in V_S^i \setminus \text{ima}(h_i)]$.
8. $\forall (P_i, x) \in \text{dom}(f_{con})$ (this is well-defined since we have already defined f_{nod} and f_{pat}): Define $f_{con}(P_i, x) = (T^{tf}, f_{nod}^{tf}, f_{lea}^{tf})$ as follows.
 - (a) $V^{tf} = \{h \in V : h \text{ belongs to a path between } f_{nod\&pat}(P_i, x) \text{ and } f_{nod\&pat}(P_j, y) \text{ in } S, \text{ for some } (P_j, y) \in \text{dom}(f_{con}), \text{ whose internal nodes are in } V_S^{t+1}\}$.
 - (b) $E^{tf} = \{\{h, h'\} \in E_S : h, h' \in V^{tf}\}$.
 - (c) $f_{nod}^{tf} : V^{tf} \rightarrow V$ is the identity function.
 - (d) $\forall h \in \text{leaves}(T^{tf}) : f_{lea}^{tf}(h) = f_{nod\&pat}^{-1}(h)$ ($f_{nod\&pat}^{-1}$ is well-defined since $f_{nod\&pat}$ is injective).

Let r be a node in $\{h \in V : \exists p \text{ s.t. } f_{nod}(\text{root}(T), p) = h\}$. It is straightforward to verify that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ is an r -good pentuple of score s . Note that f_{gra} and $f_{nod\&pat}$ are injective. Since $|V_T \cup \text{dom}(f_{con})| = |L|$, we can choose some bijection $\ell : V_T \cup \text{dom}(f_{con}) \rightarrow L$, and get that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in Cor(r, s)$.

Now, consider some $r \in V$ and $s \in \mathbb{N}^0$ such that there exists a pentuple $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$ in $Cor(r, s)$. We define $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ as follows.

1. $\forall i \in \{1, \dots, t\}$:
 - (a) If there is no $v_i \in V_T$ s.t. $f_{gra}(v_i) = P_i$, then define $V_S^i = \emptyset$, $E_S^i = \emptyset$ and $h_i : \emptyset \rightarrow \emptyset$. Else, note that there is only one such v_i , and follow Steps 1b–1e.
 - (b) $V_S^i = \{h \in V : \exists x \text{ s.t. } h = f_{nod\&pat}(v_i, x)\}$.
 - (c) $E_S^i = \{\{w(j), w(j+1)\} : \exists \{p, p'\}, path \text{ s.t. } (v, \{p, p'\}, path) \in \text{dom}(f_{pat}) \text{ and } j \in \{1, \dots, |\text{dom}(w)| - 1\}, \text{ where } w = f_{pat}(v, \{p, p'\}, path)\}$.
 - (d) $\text{dom}(h_i) = \{p : (v_i, p) \in \text{dom}(f_{nod})\}$.
 - (e) $\forall p \in \text{dom}(h_i) : h_i(p) = f_{nod}(v_i, p)$.

2. V_S^{t+1}' denotes the *multiset* including an occurrence of $f_{nod}^{tf}(v)$ for each $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \in \text{ima}(f_{con})$ and $v \in V^{tf} \setminus (\{\text{root}(T^{tf})\} \cup \text{leaves}(T^{tf}))$.
3. V_S^{t+1} denotes the *set* of different nodes in V_S^{t+1}' , excluding those in $\bigcup_{i=1}^t V_S^i$.
4. E_S^{t+1}' denotes the *multiset* that includes an occurrence of $\{f_{nod}^{tf}(v), f_{nod}^{tf}(u)\}$ for each $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \in \text{ima}(f_{con})$ and $\{v, u\} \in E^{tf}$. As long as there is a cycle in the subgraph of H containing the edges in E_S^{t+1}' and their endpoints, choose an edge of a cycle and delete it. Denote the resulting edge *set* by E_S^{t+1} .
5. $S = (\bigcup_{i=1}^{t+1} V_S^i, \bigcup_{i=1}^{t+1} E_S^i)$.

Since $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in \text{Cor}(r, s)$, it is straightforward to verify that $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ is a solution for the input with I_A alignment insertions and D deletions, such that $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ non-empty sets. If $V_S^{t+1}' \neq V_S^{t+1}$, then $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ has less than I_F free insertions, which is a contradiction (since the number of input indels is minimal). Thus, V_S^{t+1}' is a *set* equal to V_S^{t+1} , which, by the definition of $\text{Cor}(r, s)$, implies that E_S^{t+1}' is a *set* equal to E_S^{t+1} . Therefore $|E_S| = n_E$, and the score of $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ is s . \square

This proof implies the following corollary.

Corollary 1. *Let $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in \text{Cor}(r, s)$. The multiset including an occurrence of $f_{nod}^{tf}(v)$, for each $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \in \text{ima}(f_{con})$ and $v \in V^{tf} \setminus (\{\text{root}(T^{tf})\} \cup \text{leaves}(T^{tf}))$, is a set that does not contain nodes from $\text{ima}(f_{nod\&pat})$.*

D.2 Associating Monomials with Potential Solutions

Introduce the following variables.

- w .
- $\forall h, h' \in V: x_{h, h'}$.
- $\forall p \in V(\mathcal{P}), h \in V: y_{p, h}$.
- $\forall i \in \{1, \dots, t\}, p, p' \in V(P_i), \text{path} \in \text{paths}(\{p, p'\}), j \in \{2, \dots, I_A + 1\}, h \in V: y_{(\{p, p'\}, \text{path}, j), h}$.
- $\forall x \in \mathcal{P} \cup V, l \in L: z_{x, l}$.

Let ind denote the number of these variables. Note that $\text{ind} \leq 1 + (|V| + k + k^3 I_A + |L|)|V| + t|L|$. We now define the monomials of the r -solutions in $\text{Sol}(r, s)$.

Definition 14. $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = w^s \prod_{v \in V_T} z_{f_{gra}(v), \ell(v)}$.

$$\prod_{(v, x) \in \text{dom}(f_{con})} [y_{x, f_{nod\&pat}(v, x)} z_{f_{nod\&pat}(v, x), \ell(v, x)} \cdot \prod_{(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) = f_{con}(v, x), u \in V^{tf}, u' \text{ is a son of } u \text{ in } T^{tf}} x_{f_{nod}^{tf}(u), f_{nod}^{tf}(u')}].$$

Lemma 10. *If $\text{sol}, \text{sol}' \in \text{Cor}(r, s)$ are different, then $m(\text{sol}) \neq m(\text{sol}')$.*

Proof. Let $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell), (T', f'_{gra}, f'_{nod}, f'_{pat}, f'_{con}, \ell') \in \text{Cor}(r, s)$ such that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \neq (T', f'_{gra}, f'_{nod}, f'_{pat}, f'_{con}, \ell')$.

If there is $(v^*, x^*) \in \text{dom}(f_{con})$ (resp. $(v^*, x^*) \in \text{dom}(f'_{con})$) for which there is no

$v^{*'} \in V_{T'}$ (resp. $v^* \in V_T$) such that $(v^{*'}, x^*) \in \text{dom}(f'_{con})$ (resp. $(v^*, x^*) \in \text{dom}(f_{con})$) and $f_{nod\&pat}(v^*, x^*) = f'_{nod\&pat}(v^{*'}, x^*)$, then $(\prod_{(v,x) \in \text{dom}(f_{con})} y_{x, f_{nod\&pat}(v,x)} \neq \prod_{(v,x) \in \text{dom}(f'_{con})} y_{x, f'_{nod\&pat}(v,x)})$. Then, $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \neq m(T', f'_{gra}, f'_{nod}, f'_{pat}, f'_{con}, \ell')$, and thus we next assume that

Assumption 1 *Let $(v^*, x^*) \in \text{dom}(f_{con})$ (resp. $(v^{*'}, x^*) \in \text{dom}(f'_{con})$). Then, there is $v^{*'} \in V_{T'}$ (resp. $v^* \in V_T$) such that $(v^{*'}, x^*) \in \text{dom}(f'_{con})$ (resp. $(v^*, x^*) \in \text{dom}(f_{con})$) and $f_{nod\&pat}(v^*, x^*) = f'_{nod\&pat}(v^{*'}, x^*)$.*

First assume that $(T, f_{gra}) \not\equiv (T', f'_{gra})$.

If $f_{gra}(\text{root}(T)) \neq f'_{gra}(\text{root}(T'))$, then there are *different* nodes $p^* \in V(f_{gra}(\text{root}(T)))$ and $p^{*'} \in V(f'_{gra}(\text{root}(T')))$, such that $f_{nod}(\text{root}(T), p^*) = f'_{nod}(\text{root}(T'), p^{*'}) = r$. Since f_{nod} and f'_{nod} are injective, this contradicts Assumption 1.

Since f_{gra} and f'_{gra} are injective, $(T, f_{gra}) \not\equiv (T', f'_{gra})$ and $f_{gra}(\text{root}(T)) = f'_{gra}(\text{root}(T'))$, we can assume WLOG that there is $v^* \in V_T$ which has a son u^* in T , such that the (unique) node $v^{*'}$ in T' satisfying $f_{gra}(v^*) = f'_{gra}(v^{*'})$ (whose existence is implied by Assumption 1) does not have a son $u^{*'}$ satisfying $f_{gra}(u^*) = f'_{gra}(u^{*'})$. Moreover, we can denote by $u^{*'}$ the (unique) node in $V_{T'}$ satisfying $f_{gra}(u^*) = f'_{gra}(u^{*'})$ (whose existence is implied by Assumption 1); and denote by x^* and y^* the unique elements satisfying $(v^*, x^*) \in \text{dom}(f_{con})$ and $(u^*, y^*) \in \text{ima}(f_{lea}^{*'})$, where $(T^{*'}, f_{nod}^{*'}, f_{lea}^{*'}) = f_{con}(v^*, x^*)$.

By Assumption 1, we have that $(v^{*'}, x^*), (u^{*'}, y^*) \in \text{dom}(f'_{con})$, $f_{nod\&pat}(v^*, x^*) = f'_{nod\&pat}(v^{*'}, x^*)$ and $f_{nod\&pat}(u^*, y^*) = f'_{nod\&pat}(u^{*'}, y^*)$. Since u^* is a son of v^* in T , $u^{*'}$ is not a son of $v^{*'}$ in T' and $f_{nod\&pat}$ and $f'_{nod\&pat}$ are injective, Assumption 1 and Corollary 1 imply the following claim. There is a node $w^* \in V^{*}$ and a son s^* of w^* in T^{*} , for which there are no $(T^{*'}, f_{nod}^{*'}, f_{lea}^{*'}) \in \text{ima}(f'_{con}), w^{*'} \in V^{*'}$ and a son $s^{*'}$ of $w^{*'}$ in $T^{*'}$ such that $f_{nod}^{*'}(w^*) = f_{nod}^{*'}(w^{*'})$ and $f_{nod}^{*'}(s^*) = f_{nod}^{*'}(s^{*'})$. We get that

$$\prod_{(v,x) \in \text{dom}(f_{con}), (T^{*'}, f_{nod}^{*'}, f_{lea}^{*'}) = f_{con}(v,x), u \in V^{*}, u' \text{ is a son of } u} x_{f_{nod}^{*'}(u), f_{lea}^{*'}(u')} \neq \prod_{(v,x) \in \text{dom}(f'_{con}), (T^{*'}, f_{nod}^{*'}, f_{lea}^{*'}) = f'_{con}(v,x), u \in V^{*'}, u' \text{ is a son of } u} x_{f_{nod}^{*'}(u), f_{lea}^{*'}(u')}.$$

Thus, $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \neq m(T', f'_{gra}, f'_{nod}, f'_{pat}, f'_{con}, \ell')$.

Now assume that $(T, f_{gra}) \equiv (T', f'_{gra})$, which, WLOG, allows assuming that $T = T'$ and $f_{gra} = f'_{gra}$. By Assumption 1 and since f_{gra} is injective, we get that $f_{nod} = f'_{nod}$ and $f_{pat} = f'_{pat}$. Thus, Corollary 1 implies that if $f_{con} \neq f'_{con}$, then there are $(T^{*'}, f_{nod}^{*'}, f_{lea}^{*'}) \in \text{ima}(f_{con}), w^* \in V^{*}$ and a son s^* of w^* in T^{*} , for which there are no $(T^{*'}, f_{nod}^{*'}, f_{lea}^{*'}) \in \text{ima}(f'_{con}), w^{*'}$ and a son $s^{*'}$ of $w^{*'}$ in $T^{*'}$ such that $f_{nod}^{*'}(w^*) = f_{nod}^{*'}(w^{*'})$ and $f_{nod}^{*'}(s^*) = f_{nod}^{*'}(s^{*'})$. This again leads to $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \neq m(T, f_{gra}, f_{nod}, f_{pat}, f'_{con}, \ell')$. Thus, we next also assume that $f_{con} = f'_{con}$, which implies that $\ell \neq \ell'$.

Note that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ and $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$ (resp. $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell')$) uniquely determines ℓ (resp. ℓ'). Thus, since $\ell \neq \ell'$, we get that $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \neq m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell')$. \square

Lemma 11. *We can partition $Sol(r, s) \setminus Cor(r, s)$ into a set of pairs $\{sol, sol'\}$ such that $m(sol) = m(sol')$.*

Proof. We prove the lemma by showing that there is a fixed-point-free involution (i.e., a permutation that is its own inverse) $inv : Sol(r, s) \setminus Cor(r, s) \rightarrow Sol(r, s) \setminus Cor(r, s)$ such that $m(sol) = m(inv(sol))$ for all $sol \in Sol(r, s) \setminus Cor(r, s)$. Note that given $a, b, c, d \in L$, we have that $(a, b) < (c, d)$ iff $a < c$ or $(a = c \wedge b < d)$.

Let $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in Sol(r, s) \setminus Cor(r, s)$.

First, assume that f_{gra} is not injective. Define $swap(\ell)$ as the set $\{v^*, u^*\}$ in $\{\{v, u\} : v, u \in V_T, v \neq u, f_{gra}(v) = f_{gra}(u)\}$ minimizing $(\min\{\ell(v^*), \ell(u^*)\}, \max\{\ell(v^*), \ell(u^*)\})$ (this choice is well-defined since ℓ is bijective). Also, define $new(\ell)$ as ℓ except that $new(\ell)(v^*) = \ell(u^*)$ and $new(\ell)(u^*) = \ell(v^*)$, and $inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = (T, f_{gra}, f_{nod}, f_{pat}, f_{con}, new(\ell))$.

Note that $inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in Sol(r, s) \setminus Cor(r, s)$ and $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = m(inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$. Since $swap(\ell) = swap(new(\ell))$, we get that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = inv(inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$.

Suppose, by way of contradiction, that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \equiv inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$. Thus, there is an isomorphism iso between the rooted tree T and itself, such that

1. $\forall v \in V_T : f_{gra}(v) = f_{gra}(iso(v))$ and $\ell(v) = new(\ell)(iso(v))$.
2. $\forall (v, x) \in \text{dom}(f_{con}) :$
 - (a) $f_{nod\&pat}(v, x) = f_{nod\&pat}(iso(v), x)$.
 - (b) $\ell(v, x) = new(\ell)(iso(v), x)$.
 - (c) Denoting $f_{con}(v, x) = (T^{tf}, f_{nod}^{tf}, f_{lea}^{tf})$ and $f_{con}(iso(v), x) = (T^{tf'}, f_{nod}^{tf'}, f_{lea}^{tf'})$, we have that $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \equiv (T^{tf'}, f_{nod}^{tf'}, f_{lea}^{tf'})$.³

By Condition 1, the definition of $new(\ell)$ and since ℓ is bijective, we get that $(\forall v \in V_T \setminus \{v^*, u^*\} : iso(v) = v)$, $iso(v^*) = u^*$ and $iso(u^*) = v^*$. Thus, by Condition 2b, there is x such that $(v^*, x) \in \text{dom}(f_{con})$ and $\ell(v^*, x) = new(\ell)(iso(v^*), x) = new(\ell)(u^*, x) = \ell(u^*, x)$. Since ℓ is bijective, this is a contradiction.

Now, assume that f_{gra} is injective, and thus $f_{nod\&pat}$ is not injective. Let $swap(\ell)$ be the set $\{(v^*, x_{v^*}), (u^*, x_{u^*})\}$ in $\{(v, x), (u, x') : (v, x), (u, x') \in \text{dom}(f_{con}), (v, x) \neq (u, x'), f_{nod\&pat}(v, x) = f_{nod\&pat}(u, x')\}$ minimizing $(\min\{\ell(v^*, x_{v^*}), \ell(u^*, x_{u^*})\}, \max\{\ell(v^*, x_{v^*}), \ell(u^*, x_{u^*})\})$ (this choice is well-defined since ℓ is bijective). Also, define $new(\ell)$ as ℓ except that $new(\ell)(v^*, x_{v^*}) = \ell(u^*, x_{u^*})$ and $new(\ell)(u^*, x_{u^*}) = \ell(v^*, x_{v^*})$, and $inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = (T, f_{gra}, f_{nod}, f_{pat}, f_{con}, new(\ell))$.

³ Note that $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \equiv (T^{tf'}, f_{nod}^{tf'}, f_{lea}^{tf'})$ iff there is an isomorphism iso' between the rooted trees T^{tf} and $T^{tf'}$, such that

- i. $\forall v \in V^{tf} : f_{nod}^{tf}(v) = f_{nod}^{tf'}(iso'(v))$.
- ii. $\forall v \in \text{leaves}(T^{tf}) : f_{lea}^{tf}(v) = (u, x)$, where $(iso(u), x) = f_{lea}^{tf'}(iso'(v))$.

Note that $\text{inv}(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}}, \ell) \in \text{Sol}(r, s) \setminus \text{Cor}(r, s)$ and $m(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}}, \ell) = m(\text{inv}(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}}, \ell))$. Since $\text{swap}(\ell) = \text{swap}(\text{new}(\ell))$, we get that $(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}}, \ell) = \text{inv}(\text{inv}(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}}, \ell))$.

Suppose, be way of contradiction, that $(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}}, \ell) \equiv \text{inv}(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}}, \ell)$. Again, consider a function iso defined as in the case where f_{gra} is not injective. By Condition 1, the definition of $\text{new}(\ell)$ and since ℓ is bijective, we get that $(\forall v \in V_T : \text{iso}(v) = v)$. Thus, by Condition 2b, we have that $\ell(v^*, x_{v^*}) = \text{new}(\ell)(\text{iso}(v^*), x_{v^*}) = \text{new}(\ell)(v^*, x_{v^*}) = \ell(u^*, x_{u^*})$. Since ℓ is bijective, this is a contradiction. \square

D.3 Evaluating the Sum of the Monomials

For each $r \in V$, let $P(r) = \sum_{s \in \{0, \dots, (|V|+|E|)W\}, \text{sol} \in \text{Sol}(r, s)} m(\text{sol})$. We next evaluate these polynomials over the field \mathbb{F}_q , where $q = 2^{\lceil \log_2(10(4k+2I_A+I_F)) \rceil}$.

By Lemmas 9–11, the input has a solution of score s , such that $|E_S| = n_E$ and $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ nonempty sets, iff there is $r \in V$ such that $P(r)$ has a monomial with an odd coefficient in which the degree of w is s . Since \mathbb{F}_q has characteristic 2, we get that

Lemma 12. *The input has a solution of score s , such that $|E_S| = n_E$ and $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ nonempty sets, iff there is $r \in V$ such that $P(r)$ has a monomial in which the degree of w is s .*

Given $A \subseteq L$, define $P_A(r) = \sum_{\text{sol is an } r\text{-solution in which } \text{ima}(\ell) \subseteq A} m(\text{sol})$. Using inclusion-exclusion, and since \mathbb{F}_q has characteristic 2, we get that $P(r) = \sum_{A \subseteq L} P_A(r)$. Thus, we can evaluate $P(r)$ by using the following lemma.

Lemma 13. *Let $A \subseteq L$ and $a_1, \dots, a_{\text{ind}-1} \in \mathbb{F}_q$. For all $r \in V$, the polynomial $P_A(r)(w, a_1, \dots, a_{\text{ind}-1})$ can be evaluated in $O(W \log W |V|^{tw+O(1)} k^{O(1)})$ time and $O(W |V|^{tw+O(1)} k^{O(1)})$ space by using dynamic programming.*

Proof. The lemma can be shown by using a straightforward extension of the dynamic programming described in Section A.3. We thus relegate the details to the full version of the paper. \square

D.4 The Procedure

$\text{FewSingles}(n_E, n_{\mathcal{P}}, \mathcal{P}, H, \Delta, I_F, I_A, D)$ performs the same steps as SimpleFewSingles , though now they refer to the new definition of $P_A(r)$. Thus, Lemmas 5, 12 and 13 imply the correctness of Lemma 7.

E ManySingles: A Narrow Sieves Procedure

We now present procedure ManySingles (see Section 3.3). Recall that we assume that \mathcal{P} is a set of bounded treewidth graphs, the number of input indels is minimal, the weights are nonnegative integers, and there is a set $U \subseteq V$ such that $(\forall h \in U : \text{If } p \in V(\mathcal{P}) \setminus V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty)$ and $(\forall h \in V \setminus U : \text{If } p \in V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty)$. Moreover, recall that the input for ManySingles is of the form $(n_E, n_{\mathcal{P}^*}, n_{\mathcal{P}}, \mathcal{P}, H, \Delta, I_F, I_A, D)$.

In this section we say that a pentuple $(T, f_{\text{gra}}, f_{\text{nod}}, f_{\text{pat}}, f_{\text{con}})$ is r -good only if it satisfies the conditions: $U \cap (\bigcup_{w \in \text{ima}(f_{\text{pat}})} \text{ima}(w)) = \emptyset$ and $|\{v \in V_T : f_{\text{gra}}(v) \in \mathcal{P}^*\}| = n_{\mathcal{P}^*}$, in addition to the conditions in Definition 11.

E.1 Potential Solutions

Let $L = \{1, \dots, k - D + I_A + n_{\mathcal{P}} - n_{\mathcal{P}^*}\}$ denote the set of indices used for labeling r -good pentuples, resulting in potential solutions of the same score as follows.

Definition 15. *Given an r -good pentuple $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ and $\ell: V_T \cup \{(v, x) \in \text{dom}(f_{con}) : f_{gra}(v) \notin \mathcal{P}^*\} \rightarrow L$, $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$ is an r -solution.*

We now define two sets of potential solutions. $Sol(r, s)$ contains all r -solutions $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$ of score s such that ℓ is bijective; and $Cor(r, s) = \{(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in Sol(r, s) : f_{gra} \text{ and } f_{nod\&pat} \text{ are injective}\}$.

Lemma 14. *The input has a solution of score s without alignment insertions from U , such that $|E_S| = n_E$ and $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ nonempty sets and $n_{\mathcal{P}^*}$ one-node sets, iff $\bigcup_{r \in V} Cor(r, s) \neq \emptyset$.*

Proof. We describe how to modify the proof of Lemma 9 to obtain this lemma.

In the first direction, we now consider a solution satisfying two additional conditions: It does not contain alignment insertions from U , and its $\{V_S^1, \dots, V_S^t\}$ set includes exactly $n_{\mathcal{P}^*}$ one-node sets. We change the definition of ℓ to correspond to the new required domain and L . Thus, ℓ is bijective, and the constructed pentuple satisfies the additional conditions $U \cap (\bigcup_{w \in \text{ima}(f_{pat})} \text{ima}(w)) = \emptyset$ and $|\{v \in V_T : f_{gra}(v) \in \mathcal{P}^*\}| = n_{\mathcal{P}^*}$.

In the second direction, we now consider the new definition of $Cor(r, s)$, and thus get that the constructed solution satisfies the required additional conditions: it does not contain alignment insertions from U and its $\{V_S^1, \dots, V_S^t\}$ set includes exactly $n_{\mathcal{P}^*}$ one-node sets. \square

E.2 Associating Monomials with Potential Solutions

We use the variables introduced in Section D.2, and define the monomial an r -solution in $Sol(r, s)$ as follows.

Definition 16. $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = w^s \prod_{v \in V_T} z_{f_{gra}(v), \ell(v)} \cdot$

$$\prod_{(v,x) \in \text{dom}(f_{con}) \text{ s.t. } f_{gra}(v) \notin \mathcal{P}^*} z_{f_{nod\&pat}(v,x), \ell(v,x)} \prod_{(v,x) \in \text{dom}(f_{con})} [y_{x, f_{nod\&pat}(v,x)} \cdot$$

$$\prod_{(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) = f_{con}(v,x), u \in V^{tf}, u' \text{ is a son of } u \text{ in } T^{tf}} x_{f_{nod}^{tf}(u), f_{nod}^{tf}(u')}].$$

Lemma 15. *If $sol, sol' \in Cor(r, s)$ are different, then $m(sol) \neq m(sol')$.*

Proof. The new definitions of $Cor(r, s)$ and the monomial of an r -solution of score s only effect the last claim in the proof of Lemma 9 (i.e., the monomial uniquely determines ℓ). Yet, since this claim clearly holds, we get the lemma. \square

Lemma 16. *We can partition $Sol(r, s) \setminus Cor(r, s)$ into a set of pairs $\{sol, sol'\}$ such that $m(sol) = m(sol')$.*

Proof. Some parts of this proof are similar to the proof of Lemma 11. For the sake of clarity, and since this is the only proof showing the necessity of assuming that there is a set $U \subseteq V$ such that $(\forall h \in U : \text{If } p \in V(\mathcal{P}) \setminus V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty)$ and $(\forall h \in V \setminus U : \text{If } p \in V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty)$, we give the full proof.

We prove the lemma by showing that there is a fixed-point-free involution (i.e., a permutation that is its own inverse) $inv : Sol(r, s) \setminus Cor(r, s) \rightarrow Sol(r, s) \setminus Cor(r, s)$ such that $m(sol) = m(inv(sol))$ for all $sol \in Sol(r, s) \setminus Cor(r, s)$. Note that given $a, b, c, d \in L$, we have that $(a, b) < (c, d)$ iff $a < c$ or $(a = c \wedge b < d)$.

Let $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in Sol(r, s) \setminus Cor(r, s)$.

First, assume that f_{gra} is not injective. Define $swap(\ell)$ as the set $\{v^*, u^*\}$ in $\{\{v, u\} : v, u \in V_T, v \neq u, f_{gra}(v) = f_{gra}(u)\}$ minimizing $(\min\{\ell(v^*), \ell(u^*)\}, \max\{\ell(v^*), \ell(u^*)\})$ (this choice is well-defined since ℓ is bijective). Also, define $new(\ell)$ as ℓ except that $new(\ell)(v^*) = \ell(u^*)$ and $new(\ell)(u^*) = \ell(v^*)$, and $inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = (T, f_{gra}, f_{nod}, f_{pat}, f_{con}, new(\ell))$.

Note that $inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in Sol(r, s) \setminus Cor(r, s)$ and $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = m(inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$. Since $swap(\ell) = swap(new(\ell))$, we get that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = inv(inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$.

Suppose, by way of contradiction, that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \equiv inv(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$. Thus, there is an isomorphism iso between the rooted tree T and itself, such that

1. $\forall v \in V_T : f_{gra}(v) = f_{gra}(iso(v))$ and $\ell(v) = new(\ell)(iso(v))$.
2. $\forall (v, x) \in \text{dom}(f_{con})$ s.t. $f_{gra}(v) \notin \mathcal{P}^* : \ell(v, x) = new(\ell)(iso(v), x)$.
3. $\forall (v, x) \in \text{dom}(f_{con}) :$
 - (a) $f_{nod\&pat}(v, x) = f_{nod\&pat}(iso(v), x)$.
 - (b) Denoting $f_{con}(v, x) = (T^{tf}, f_{nod}^{tf}, f_{lea}^{tf})$ and $f_{con}(iso(v), x) = (T^{tf'}, f_{nod}^{tf'}, f_{lea}^{tf'})$, we have that $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \equiv (T^{tf'}, f_{nod}^{tf'}, f_{lea}^{tf'})$.⁴

By Condition 1, the definition of $new(\ell)$ and since ℓ is bijective, we get that $(\forall v \in V_T \setminus \{v^*, u^*\} : iso(v) = v), iso(v^*) = u^*$ and $iso(u^*) = v^*$. Thus, by Condition 3b, we get that there are (unique) x_{v^*}, x_{u^*} and $(w^*, x_{w^*}) \in \text{dom}(f_{con})$ s.t. $(v^*, x_{v^*}), (u^*, x_{u^*}) \in \text{ima}(f_{lea}^{tf*})$, where $(T^{tf*}, f_{nod}^{tf*}, f_{lea}^{tf*}) = f_{con}(w^*, x_{w^*})$; and there is an isomorphism iso' between the rooted tree T^{tf*} and itself, such that

1. $\forall v \in V^{tf*} : f_{nod}^{tf*}(v) = f_{nod}^{tf*}(iso'(v))$.
2. $\forall v \in \text{leaves}(T^{tf*}) : f_{lea}^{tf*}(v) = (u, x)$, where $(iso(u), x) = f_{lea}^{tf*}(iso'(v))$.

Denote by l_{v^*} and l_{u^*} the (unique) leaves in T^{tf*} s.t. $f_{lea}^{tf*}(l_{v^*}) = (v^*, x_{v^*})$ and $f_{lea}^{tf*}(l_{u^*}) = (u^*, x_{u^*})$. By Condition 2, we get that $(\forall v \in \text{leaves}(T^{tf*}) \setminus \{l_{v^*}, l_{u^*}\} : iso'(v) = v), iso'(l_{v^*}) = l_{u^*}$ and $iso'(l_{u^*}) = l_{v^*}$. Let w^* be the lowest common ancestor of l_{v^*} and l_{u^*} in T^{tf*} , and let s_{v^*} and s_{u^*} be the sons of w^* in T^{tf*} on the simple paths from w^* to l_{v^*} and l_{u^*} , respectively. Since iso' is an isomorphism between the rooted tree T^{tf*} and itself, we get that $iso'(s_{v^*}) = s_{u^*}$, and by Condition 1, we get that $f_{nod}^{tf*}(s_{v^*}) = f_{nod}^{tf*}(s_{u^*})$. We thus have a contradiction, since $(T^{tf*}, f_{nod}^{tf*}, f_{lea}^{tf*})$ does not fulfill Condition 3 in Definition 10.

⁴ Note that $(T^{tf}, f_{nod}^{tf}, f_{lea}^{tf}) \equiv (T^{tf'}, f_{nod}^{tf'}, f_{lea}^{tf'})$ iff there is an isomorphism iso' between the rooted trees T^{tf} and $T^{tf'}$, such that

- i. $\forall v \in V^{tf} : f_{nod}^{tf}(v) = f_{nod}^{tf'}(iso'(v))$.
- ii. $\forall v \in \text{leaves}(T^{tf}) : f_{lea}^{tf}(v) = (u, x)$, where $(iso(u), x) = f_{lea}^{tf'}(iso'(v))$.

Now, assume that f_{gra} is injective, and thus $f_{nod\&pat}$ is not injective. Let $A = \{(v, x) \in \text{dom}(f_{con}) : f_{gra}(v) \in \mathcal{P}^*\}$ and $B = \text{dom}(f_{con}) \setminus A$. We have the following cases.

1. If $\{(v, x), (u, x')\} : (v, x), (u, x') \in B, (v, x) \neq (u, x'), f_{nod\&pat}(v, x) = f_{nod\&pat}(u, x')\} \neq \emptyset$:

Define $\text{swap}(\ell)$ as the set $\{(v^*, x_{v^*}), (u^*, x_{u^*})\}$ in $\{(v, x), (u, x') \in B, (v, x) \neq (u, x'), f_{nod\&pat}(v, x) = f_{nod\&pat}(u, x')\}$ that minimizes $(\min\{\ell(v^*, x_{v^*}), \ell(u^*, x_{u^*})\}, \max\{\ell(v^*, x_{v^*}), \ell(u^*, x_{u^*})\})$ (this choice is well-defined since ℓ is bijective). Also, define $\text{new}(\ell)$ as ℓ except that $\text{new}(\ell)(v^*, x_{v^*}) = \ell(u^*, x_{u^*})$ and $\text{new}(\ell)(u^*, x_{u^*}) = \ell(v^*, x_{v^*})$, and $\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = (T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \text{new}(\ell))$.

Note that $\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in \text{Sol}(r, s) \setminus \text{Cor}(r, s)$ and $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = m(\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$. Since $\text{swap}(\ell) = \text{swap}(\text{new}(\ell))$, we get that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = \text{inv}(\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$.

Suppose, be way of contradiction, that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \equiv \text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$. Again, consider a function iso defined as in the case where f_{gra} is not injective. By Condition 1, the definition of $\text{new}(\ell)$ and since ℓ is bijective, we get that $(\forall v \in V_T : \text{iso}(v) = v)$. Thus, by Condition 2, we have that $\ell(v^*, x_{v^*}) = \text{new}(\ell)(\text{iso}(v^*), x_{v^*}) = \text{new}(\ell)(v^*, x_{v^*}) = \ell(u^*, x_{u^*})$. Since ℓ is bijective, this is a contradiction.

2. Else if $\{(v, x), (u, x')\} : (v, x) \in A, (u, x') \in B, (v, x) \neq (u, x'), f_{nod\&pat}(v, p) = f_{nod\&pat}(u, p')\} \neq \emptyset$:

This case cannot be true since $(T, f_{gra}, f_{nod}, f_{pat}, f_{con})$ is an r -good pentuple, and there is a set $U \subseteq V$ such that $(\forall h \in U : \text{If } p \in V(\mathcal{P}) \setminus V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty)$ and $(\forall h \in V \setminus U : \text{If } p \in V(\mathcal{P}^*) \text{ then } \Delta(p, h) = -\infty)$, and $U \cap (\bigcup_{w \in \text{ima}(f_{pat})} \text{ima}(w)) = \emptyset$.

3. Else:

Define $\text{swap}(f_{gra}, \ell)$ as the set $\{(v^*, x_{v^*}), (u^*, x_{u^*})\}$ in $\{(v, x), (u, x')\} : (v, x), (u, x') \in A, v \neq u, f_{nod\&pat}(v, x) = f_{nod\&pat}(u, x')\}$ that minimizes $(\min\{\ell(v^*), \ell(u^*)\}, \max\{\ell(v^*), \ell(u^*)\})$ (this choice is well-defined since ℓ and f_{gra} are injective and $f_{gra}(v^*), f_{gra}(u^*) \in \mathcal{P}^*$). Define $\text{new}(f_{gra})$ as f_{gra} except that $\text{new}(f_{gra})(v^*) = f_{gra}(u^*)$ and $\text{new}(f_{gra})(u^*) = f_{gra}(v^*)$. Define $\text{new}(f_{nod})$ as f_{nod} except that $\text{new}(f_{nod})(v^*, x_{v^*}) = f_{nod}(v^*, x_{v^*})$ and $\text{new}(f_{nod})(u^*, x_{u^*}) = f_{nod}(u^*, x_{u^*})$. Also, define $\text{new}(f_{con})$ as f_{con} except that $\text{new}(f_{con})(v^*, x_{v^*}) = f_{con}(v^*, x_{v^*})$ and $\text{new}(f_{con})(u^*, x_{u^*}) = f_{con}(u^*, x_{u^*})$. Finally, define $\text{new}(\ell)$ as ℓ except that $\text{new}(\ell)(v^*) = \ell(u^*)$ and $\text{new}(\ell)(u^*) = \ell(v^*)$, and $f(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = (T, \text{new}(f_{gra}), \text{new}(f_{nod}), f_{pat}, \text{new}(f_{con}), \text{new}(\ell))$.

Note that $\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \in \text{Sol}(r, s) \setminus \text{Cor}(r, s)$ and $m(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = m(\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$. Since $\text{swap}(f_{gra}, \ell) = \text{swap}(\text{new}(f_{gra}), \text{new}(\ell))$ (and neither of the previous two cases is satisfied by $\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$), we get that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) = \text{inv}(\text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell))$. Following arguments similar to those presented in the case where f_{gra} is not injective, we get that $(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell) \not\equiv \text{inv}(T, f_{gra}, f_{nod}, f_{pat}, f_{con}, \ell)$.

□

E.3 Evaluating the Sum of the Monomials

For each $r \in V$, let $P(r) = \sum_{s \in \{0, \dots, (|V|+|E|)W\}, \text{sol} \in \text{Sol}(r,s)} m(\text{sol})$. We evaluate the polynomials over \mathbb{F}_q (see Section D.3). Then, by Lemmas 14–16, we get that

Lemma 17. *The input has a solution of score s without alignment insertions from U , such that $|E_S| = n_E$ and $\{V_S^1, \dots, V_S^t\}$ includes exactly $n_{\mathcal{P}}$ nonempty sets and $n_{\mathcal{P}^*}$ one-node sets, iff there is $r \in V$ such that $P(r)$ has a monomial in which the degree of w is s .*

Given $A \subseteq L$, define $P_A(r) = \sum_{\text{sol} \text{ is an } r\text{-solution in which } \text{ima}(\ell) \subseteq A} m(\text{sol})$. Using inclusion-exclusion, and since \mathbb{F}_q has characteristic 2, we get that $P(r) = \sum_{A \subseteq L} P_A(r)$. Thus, we can evaluate $P(r)$ by using the following lemma.

Lemma 18. *Let $A \subseteq L$ and $a_1, \dots, a_{\text{ind}-1} \in \mathbb{F}_q$. For all $r \in V$, the polynomial $P_A(r)(w, a_1, \dots, a_{\text{ind}-1})$ can be evaluated in $O(W \log W |V|^{tw+O(1)} k^{O(1)})$ time and $O(W |V|^{tw+O(1)} k^{O(1)})$ space by using dynamic programming.*

Proof. The lemma can be shown by using a straightforward extension of the dynamic programming described in Section A.3. We thus relegate the details to the full version of the paper. \square

E.4 The Procedure

ManySingles($n_E, n_{\mathcal{P}^*}, n_{\mathcal{P}}, \mathcal{P}, H, \Delta, I_F, I_A, D$) performs the same steps as SimpleFewSingles, though now they refer to the new definition of $P_A(r)$. Thus, Lemmas 5, 17 and 18 imply the correctness of Lemma 8.

F An FPT-Approximation Scheme (Cont.)

Finally, we describe our FPT-AS for PINQ_I (see the pseudocode below). Algorithm **Approx** first ignores the weights and only checks if a solution exists, in which case it examines several choices for the maximum weight, M , used by a solution of score OPT . Using scaling and rounding (see, e.g., [21]), algorithm **Approx** then manipulates the weights in the given instance to be small enough, so that our algorithm **Exact** can be implemented efficiently, with a small loss in accuracy.

Algorithm 2: Approx($\epsilon, \mathcal{P}, H, \Delta, I_F, I_A, D$)

- 1: change each $v \in$ weights to 0.
 - 2: **if** Exact($\mathcal{P}, H, \Delta, I_F, I_A, D$) rejects **then** reject. **end if**
 - 3: restore the original input weights.
 - 4: initialize $s \leftarrow 0$.
 - 5: **for all** $M \in$ weights $\setminus \{0\}$ **do**
 - 6: change each $v \in$ weights to $\lfloor \frac{(|V|+|E|)v}{\epsilon M} \rfloor$ if $v \leq M$, and to $-\infty$ otherwise.
 - 7: **if** Exact($\mathcal{P}, H, \Delta, I_F, I_A, D$) returns $s' > s \frac{(|V|+|E|)}{\epsilon M}$ **then** $s \leftarrow \frac{\epsilon M}{(|V|+|E|)} s'$. **end if**
 - 8: restore the original input weights.
 - 9: **end for**
 - 10: Return s .
-

We now turn to the proof of Theorem 2.

Proof. By Theorem 3, if there is no solution or the number of input indels is not minimal, **Exact** rejects with probability $\geq \frac{9}{10}$ in Step 2, and thus **Approx** rejects.

Thus, next assume that this is not the case. If there is no iteration of Step 5, then $OPT = 0$ and **Approx** returns 0. Thus, next assume that this is also not the case, and denote $N = (|V| + |E|)$.

Consider some iteration of Step 5, and let $(S^*, V_S^{1^*}, \dots, V_S^{t+1^*}, h_1^*, \dots, h_t^*)$ be a maximum score solution for its new input. We have that

$$\begin{aligned} OPT &\geq \sum_{i=1}^t \sum_{p \in \text{dom}(h_i^*)} \Delta(p, h_i^*(p)) + \sum_{e \in E_{S^*}} w(e) \\ &\geq \frac{\epsilon M}{N} \left(\sum_{i=1}^t \sum_{p \in \text{dom}(h_i^*)} \lfloor \frac{N \Delta(p, h_i^*(p))}{\epsilon M} \rfloor + \sum_{e \in E_{S^*}} \lfloor \frac{N w(e)}{\epsilon M} \rfloor \right) \\ &= \frac{\epsilon M}{N} \text{score}(S^*, V_S^{1^*}, \dots, V_S^{t+1^*}, h_1^*, \dots, h_t^*) \end{aligned}$$

Thus, by Theorem 3, we do not assign a score greater than OPT to s (in any iteration). This implies that if $OPT = 0$, then $s = 0$, and thus next assume that $OPT > 0$.

Let $(S, V_S^1, \dots, V_S^{t+1}, h_1, \dots, h_t)$ be a solution of score OPT . Next, consider the iteration corresponding to $M = \max(\{w(e) : e \in E_S\} \cup \{\Delta(p, h) : p \in V(\mathcal{P}), h \in V\})$. Let $(S^*, V_S^{1^*}, \dots, V_S^{t+1^*}, h_1^*, \dots, h_t^*)$ be a maximum score solution for the input with whom we call **Exact** in this iteration. Note that $OPT \geq M$. By Theorem 3, with probability $\geq \frac{9}{10}$, **Exact** returns $\text{score}(S^*, V_S^{1^*}, \dots, V_S^{t+1^*}, h_1^*, \dots, h_t^*)$. We have that

$$\begin{aligned} &\frac{\epsilon M}{N} \text{score}(S^*, V_S^{1^*}, \dots, V_S^{t+1^*}, h_1^*, \dots, h_t^*) = \\ &\frac{\epsilon M}{N} \left(\sum_{i=1}^t \sum_{p \in \text{dom}(h_i^*)} \lfloor \frac{N \Delta(p, h_i^*(p))}{\epsilon M} \rfloor + \sum_{e \in E_{S^*}} \lfloor \frac{N w(e)}{\epsilon M} \rfloor \right) \geq \\ &\frac{\epsilon M}{N} \left(\sum_{i=1}^t \sum_{p \in \text{dom}(h_i)} \lfloor \frac{N \Delta(p, h_i(p))}{\epsilon M} \rfloor + \sum_{e \in E_S} \lfloor \frac{N w(e)}{\epsilon M} \rfloor \right) \geq \\ &\sum_{i=1}^t \sum_{p \in \text{dom}(h_i)} \Delta(p, h_i(p)) + \sum_{e \in E_S} w(e) - \left(\sum_{i=1}^t |\text{dom}(h_i)| + |E_S| \right) \frac{\epsilon M}{N} \geq \\ &OPT - \epsilon M \geq (1 - \epsilon)OPT \end{aligned}$$

Thus, we have proved that

1. If there is no solution or the number of input indels is not minimal: **Approx** rejects with probability $\geq \frac{9}{10}$.
2. Else if $OPT = 0$: **Approx** returns 0 with probability $\geq \frac{9}{10}$.
3. Else (i.e., $OPT > 0$): **Approx** returns a value in $[(1 - \epsilon)OPT, OPT]$ with probability $\geq (\frac{9}{10})^2 > \frac{4}{5}$.

In each iteration, the maximum value of a weight of the new input is $\lfloor \frac{N}{\epsilon} \rfloor$. Thus, by the time and space complexities of **Exact**, and since we execute at most $|E| + k|V|$ iterations of Step 5, we get that the time and space complexities of **Approx** are as stated in the theorem. \square