Shimizu et al. BMC Bioinformatics 2011, 12(Suppl 14):S3
http://www.biomedcentral.com/1471-2105/12/S14/S3

BMC Bioinformatics

PROCEEDINGS Open Access

Enumerating tree-like chemical graphs with given upper and lower bounds on path frequencies

Masaaki Shimizu[1], Hiroshi Nagamochi[1*], Tatsuya Akutsu[2]

From 22nd International Conference on Genome Informatics
Busan, Korea. 5-7 December 2011

Abstract

Background: Enumeration of chemical graphs satisfying given constraints is one of the fundamental problems in chemoinformatics and bioinformatics since it leads to a variety of useful applications including structure determination of novel chemical compounds and drug design.

Results: In this paper, we consider the problem of enumerating all tree-like chemical graphs from a given set of feature vectors, which is specified by a pair of upper and lower feature vectors, where a feature vector represents the frequency of prescribed paths in a chemical compound to be constructed. This problem can be solved by applying the algorithm proposed by Ishida *et al.* to each single feature vector in the given set, but this method may take much computation time because in general there are many feature vectors in a given set. We propose a new exact branch-and-bound algorithm for the problem so that all the feature vectors in a given set are handled directly. Since we cannot use the bounding operation proposed by Ishida *et al.* due to upper and lower constraints, we introduce new bounding operations based on upper and lower feature vectors, a bond constraint, and a detachment condition.

Conclusions: Our proposed algorithm is useful for enumerating tree-like chemical graphs with given upper and lower bounds on path frequencies.

Introduction

Development of novel drugs is one of the major goals in chemoinformatics and bioinformatics. To achieve this purpose, it is important not only to investigate common chemical properties over chemical compounds having common structural patterns [1-3] but also to study methods of enumerating chemical structures satisfying given constraints. The enumeration of chemical structures has a long history. Actually, Cayley [4] considered the enumeration of structural isomers of alkanes in the 19th century. Applications for the enumeration of chemical compounds include structure determination using mass-spectrum and/or NMR-spectrum [5,6], virtual exploration of chemical universe [7,8], reconstruction of molecular structures from their signatures [9,10], and classification of chemical compounds [11].

In the field of machine learning, the *pre-image problem* [12,13] has been studied. In this problem, a desired object is computed as a feature vector in a feature space, and then the feature vector is mapped back to the input space, where this mapped back object is called a pre-image. The definition of the feature vectors based on the frequency of labeled paths [14,15] or small fragments [11,16] has been widely used. Akutsu and Fukagawa [17] formulated the graph pre-image problem as the problem of inferring graphs from the frequency of paths of labeled vertices, which corresponds to the pre-image problem, and proved that the problem is NP-hard even for planar graphs with bounded degrees [17]. Nagamochi [18] proved that a graph determined by frequency of paths with length 1 can be found in polynomial time if any.

To enumerate tree-like chemical graphs, Fujiwara *et al.* [19] proposed a branch-and-bound algorithm

* Correspondence: nag@amp.i.kyoto-u.ac.jp
[1]Graduate School of Informatics, Kyoto University, Yoshida, Kyoto 606-8501, Japan
Full list of author information is available at the end of the article

which consists of a branching procedure based on the tree enumeration algorithm due to Nakano and Uno [20,21] and bounding operations designed by the path frequency and the atom-atom bonds. In addition, to reduce the size of search trees, Ishida *et al.* [22] introduced a new bounding operation, called the *detachment-cut*, based on the result by Nagamochi [18]. Implementations of the algorithm proposed by Ishida *et al.* [22] are available at a web server (*http://sunflower.kuicr.kyoto-u.ac.jp/tools/enumol/*) for enumerating tree-like chemical graphs with given path frequency. However, an instance with constraint which is specified by one feature vector admits no solution in many cases. Therefore, it is needed to introduce a more relaxed constraint than a single feature vector to obtain some solutions in the tree-like chemical graph enumeration problem.

In this paper, we are given a set of feature vectors, which is specified by a pair of upper and lower feature vectors, and enumerate all tree-like chemical graphs satisfying one of the vectors. It seems that this can be done by simply applying the algorithm proposed by Ishida *et al.* to each single feature vector in the given set. However, this method will take much computation time because in general there are many feature vectors in a given set. We propose a new exact branch-and-bound algorithm for the problem so that all the feature vectors in a given set are handled directly.

Methods

Preliminaries and problem formulation

A graph is called a *multigraph* if multiple edges (i.e., edges with the same end vertices) are allowed; otherwise it is called *simple*. A *path* P is a sequence $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ of distinct vertices v_i ($i = 0, \ldots, k$) and edges e_j that join v_{j-1} and v_j ($j = 1, \ldots, k$). Without confusion we may write $P = (v_0, v_1, \ldots, v_k)$. The length $|P|$ of path P is defined to be k, i.e., the number of edges. Assume that a set $\Sigma = \{\ell_1, \ell_2, \ldots, \ell_s\}$ (i.e., chemical elements) is given. Let each label ℓ be associated with a valence $val(\ell) \in \mathbb{Z}_+$. A multigraph G is called Σ-*labeled* if each vertex v has a label $\ell(v) \in \Sigma$, and is called (Σ, *val*)-*labeled* if, in addition, the degree of each vertex v is $val(\ell(v))$, i.e., the valence of the element $\ell(v)$. We regard chemical compounds as (Σ, *val*)-labeled, self-loopless, and connected multigraphs, where vertices and labels represent atoms and elements, respectively. For a path $P = (v_0, v_1, \ldots, v_k)$, we call $\ell(P) = \ell(v_0), \ell(v_1), \ldots, \ell(v_k)$ the *label sequence* of P. Given a label sequence t, let $\#t$ denote the number of paths P with $\ell(P) = t$ in a graph, where multiple edges with the same end-vertices are treated as a single edge and paths are considered to be "directed." The *feature vector* $f_K(G)$ *of level* K ($\in \mathbb{Z}_+$) of G is defined to be the vector whose entry $f_K(G)[t]$ ($|t| \le K$) represents $\#t$. See Fig. 1 for an example.

Figure 1 A chemical compound and its feature vector. An illustration of a (Σ, *val*)-labeled multitree G and its feature vector $f_1(G)$. Notice that multiple edges with the same end-vertices are treated as one edge, where $\#OC = \#CO = 2$.

Let $deg(v; G)$ denote the degree of a vertex v in a graph G. The tree-like chemical graph enumeration problem with given one feature vector can be formulated as follows [19].

Enumeration of Tree-like chemical graphs with given Path Frequency (ETPF)

Given a set Σ of labels, a valence function $val : \Sigma \to \mathbb{Z}_+$ and a feature vector g of level K, find all (Σ, *val*)-labeled multitrees T such that $f_K(T) = g$ and $deg(v;T) = val(\ell(v))$ for all vertices $v \in V(T)$.

Observe that a large number of chemical compounds contain a high proportion of hydrogens. Based on this fact, another model can be considered in the problem ETPF by removing all hydrogen atoms. These two different models were proposed by Fujiwara *et al.* [19] and Ishida [23].

In this paper, we consider the problem of enumerating all tree-like chemical graphs based on given upper and lower feature vectors because we want to relax the feature vector constraint in the problem ETPF. For feature vectors g_1 and g_2 of level K, we define $g_1 \le g_2$ to be $g_1[t] \le g_2[t]$ for any label sequence t ($|t| \le K$). The problem of enumerating tree-like compounds from given two feature vectors can be formulated based on the problem ETPF as follows (see Fig. 2 for an illustration).

Enumeration of Tree-like chemical graphs with given Upper and Lower bounds on path Frequencies (ETULF)

Given a set Σ of labels, a valence function $val : \Sigma \to \mathbb{Z}_+$ and feature vectors g_U and g_L of level K ($g_L \le g_U$), find all (Σ, *val*)-labeled multitrees T such that $g_L \le f_K(T) \le g_U$ and $deg(v;T) = val(\ell(v))$ for all vertices $v \in V(T)$.

For the problem ETULF, we assume that $g_L(\ell) = g_U(\ell)$ for an atom type $\ell \in \Sigma$, where $g(L)$ denotes the entry in g that corresponds to a label sequence L (thus $g(\ell)$ specifies the number of vertices of label ℓ) and that $g_L(L) \le g_U(L)$ for any label sequence L ($|L| \ge 2$).

Note that the number n of vertices is given by $\sum_{\ell \in \Sigma} g(\ell)$. To solve the problem ETULF, we start with an empty graph, and repeatedly extend the current tree T by appending a new vertex with each label $\ell \in \Sigma$ to obtain a *valid* tree (a tree that does not violate any constraints on output trees) one by one until we get n vertices. In order

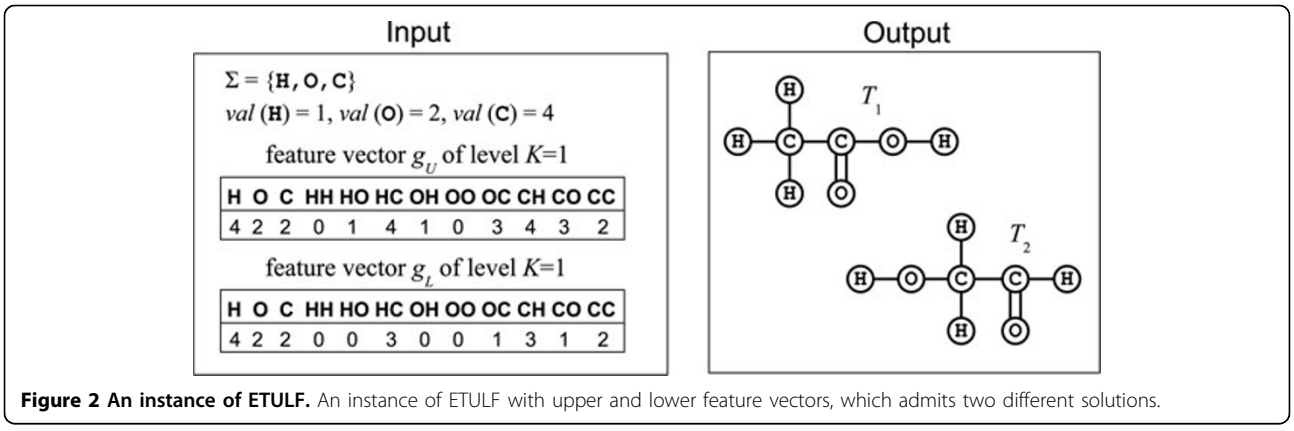

Figure 2 An instance of ETULF. An instance of ETULF with upper and lower feature vectors, which admits two different solutions.

to avoid duplicate outputs, we follow the branch-and-bound framework of Fujiwara *et al*. [19], which first defines a canonical representation for isomorphic trees, and then lists them using the algorithm of Nakano and Uno [20,21] (the branching operation) discarding invalid trees with some bounding operations. Since we cannot directly use the bounding operation proposed by Ishida *et al*. [22] due to upper and lower constraints, we introduce some new bounding operations.

Canonical representation of trees and the branching operation

In this section, we explain a canonical representation of trees introduced by Fujiwara *et al*. [19] and the branching operation based on the canonical representation.

First of all, we introduce a root of a tree based on the following theorem.

Theorem 1 (Jordan [24]**)** *For any tree with n' vertices, either there exists a unique vertex v* such that each subtree obtained by removing v* contains at most* $\left\lfloor \frac{n'-1}{2} \right\rfloor$ *vertices, or there exists a unique edge e* such that both of the subtrees obtained by removing e* contain exactly* $\frac{n'}{2}$ *vertices.*

Such a vertex v^* and an edge e^* in Theorem 1 are called *unicentroid* and *bicentroid*, respectively. Either unicentroid or bicentroid is called as *centroid*. Note that there exists a bicentroid only for an even n'. Since a case of bicentroid is similar to a case of unicentroid, now we only explain a case of unicentroid.

Next we introduce a canonical representation of trees that must be unique up to isomorphism. Let T be a tree of n vertices rooted at a vertex v_0 (which is not necessarily its unicentroid). Suppose that it is embedded in the plane as an ordered tree, where v_0 is located at the top part. Without loss of generality, let $v_0, v_1, \ldots, v_{n-1}$ be indexed by the depth-first search (DFS) that starts from v_0 and visits vertices from the left to the right. Define the *depth* $d(v)$ of a vertex v to be the length of the (unique) path from v_0 to v in T. The *depth-label sequence* of T ($L(T)$) is defined to be

$$L(T) = (d(v_0), \ell(v_0), d(v_1), \ell(v_1), \ldots, d(v_{n-1}), \ell(v_{n-1})).$$

Given an arbitrary order of labels, we define the order of depth-label sequences as follows. For any T_1 and T_2, we denote $L(T_1) > L(T_2)$ if $L(T_1)$ is *lexicographically* larger than $L(T_2)$. Then the *canonical representation* of a rooted tree is defined by the *largest* depth-label sequence among all its plane embeddings. Actually this is equivalent to the *left-heavy* plane embedding [20,21].

Thus our branching task is to list all centroid-rooted left-heavy trees with n vertices and m (= $|\Sigma|$) labels. Following the scheme [20,21], we define a *parent-child* relation between two left-heavy trees. The *parent* $P(T)$ of a left-heavy tree T is obtained from T by removing its *rightmost* leaf. Clearly $P(T)$ is still left-heavy In this way, we can define a *family tree* $\mathcal{F}(n, m)$ of left-heavy trees whose leaves are exactly what we want to obtain.

Therefore we only need to enumerate the (leaf) nodes of $\mathcal{F}(n, m)$. This can be done by starting from the empty tree (the root node of $\mathcal{F}(n, m)$) and repeatedly appending a new leaf to some appropriate place on the rightmost path of the current tree. Our branching operation employs the algorithm of Nakano and Uno [20,21], which extends the current tree T (i.e., finds a child of T) in *constant* time [19].

Bounding operations

In this section, we explain how to check the validity of the current tree T. If we can conclude that T and all its descendants are not valid, then we can discard T. Our bounding operation discards T if at least one of the following criteria is violated:

(C1) The root of T remains the centroid of an output (the centroid constraint);

(C2) $deg(v;T) \leq val(l(v))$ for all $v \in V(T)$ (the valence constraint);

(C3) $f_K(T) \leq g_U$, and $|T| = n$ and $g_L \leq f_K(T)$ (the feature vector constraint);

(C4) T can be extended to a connected and loopless tree with n vertices (the detachment constraint);

(C5) T can have a descendant which has an appropriate number of multiple bonds (the multiplicity constraint).

(C1) and (C2) are the same as the work by Fujiwara *et l.* [19] and not difficult to check. (C3) and (C4) are different from the work by Fujiwara *et al.* [19] and Ishida *et al.* [22] due to upper and lower constraints. (C5) is a new bounding operation that we propose in this paper. In the following three subsections, we will discuss three bounding operations resulting from (C3), (C4), and (C5), called as *feature-vector-cut*, *detachment-cut*, and *multiplicity-cut*, respectively.

Feature-vector-cut procedure

In the problem ETULF, we cannot use the bounding operation proposed by Fujiwara *et al.* [19] directly due to upper and lower feature vectors, but we can introduce a bounding operation based on upper and lower feature vectors by modifying Fujiwara *et al.*'s work slightly.

Let T denote a current tree, $f_K(T)$ denote the feature vector of T, g_u denote a given upper feature vector, and g_L denote a given lower feature vector. By the feature vector constraints in the problem ETULF, we check the following condition.

$$f_K(T) \leq g_U. \tag{1}$$

If T violates (1), then we discard T.

In addition, if $|T| = n$, then we check the following condition based on the constraint of upper and lower feature vectors.

$$g_L \leq f_K(T) \leq g_U. \tag{2}$$

If T violates (2), then we discard T.

Detachment-cut procedure

This subsection describes the definition of detachment [18] and a new bounding operation based on it for the problem ETULF. Let G be a multigraph that may have self-loops, which represents the graph obtained from a chemical graph H by contracting the vertices with the same label into a single vertex, where each vertex in G corresponds a label in H (note that we do not eliminate any edges in H in contracting vertices to obtain G). A process of regaining H from G is described as follows. Given a function $r : V(G) \to \mathbb{Z}_+$, an *r-detachment* H of G is a multigraph obtained from G by splitting each vertex $v \in V(G)$ into a set of $r(v)$ copies of v, denoted by $W_v = \{v^1, v^2 \ldots, v^{r(v)}\}$, so that each edge $\{u, v\} \in E(G)$ joins some vertices $u^i \in W_u$ and $v^j \in W_v$. Hence an r-detachment H of G is not unique in general. A self-loop $\{u, u\}$ in G may be mapped to a self-loop $\{u^i, u^i\}$ or a non-loop edge $\{u^i, u^j\}$ in a detachment H of G. Note that, for all vertex pairs $\{u, v\} \in V(G)$, the number of edges between subsets W_u and W_v in H is equal to that of edges between vertices u and v in G.

To obtain a chemical graph H as an r-detachment H of G, we need to specify the degree of vertices (with the same label) in H. For a function $r : V(G) \to \mathbb{Z}_+$, an *r-degree specification* is a set ρ of vectors $\rho(v) = (\rho_1^v, \rho_2^v, \ldots \rho_{r(v)}^v)$ for $v \in V(G)$ such that

$$\sum_{1 \leq i \leq r(v)} \rho_i^v = deg(v; G),$$

which is necessary for all the edges incident to vertex v in G to be assigned to split vertices $v^i \in W_v$ completely. An r-detachment H of G is called a *ρ-detachment* if each $v \in V$ satisfies

$$\deg(v^i; H) = \rho_i^v \text{ for all } v^i \in W_v = \{v^1, v^2, \ldots v^{r(v)}\},$$

which is a requirement that each vertex v_i in H must have the prescribed degree ρ_i^v. Figure 3 illustrates a ρ-detachment H for a graph $G = (V, E)$ with $V = \{a, b, c\}$, a function r with $r(a) = 4$, $r(b) = 3$, $r(c) = 1$, and a degree specification ρ with $\rho(a) = (2, 2, 3, 2)$, $\rho(b) = (2, 3, 1)$, $\rho(c) = (3)$. The next theorem gives a characterization of a multigraph G that admits a connected and loopless ρ-detachment.

Theorem 2 (Nagamochi [18]) *Let $G = (V, E)$ be a multigraph, $r : V \to \mathbb{Z}_+$ and $\rho : V \to \mathbb{Z}_+^{r(v)} (v \in V)$. Then G has a connected and loopless ρ-detachment H if and only if the following hold:*

$$r(X) + c(G - X) - d(X, V; G) \leq 1 \quad (\forall X \subseteq V, X \neq \ ),$$
$$1 \leq \rho_i^v \leq d(v; G) + d(\{v\}, \{v\}; G) \quad (\forall v \in V, i = 1, 2, \ldots, r(v)),$$

where $r(X) = \Sigma_{v \in X} r(v)$, $c(G')$ denotes the number of connected components of a graph G', $G - X$ denotes the graph obtained from a graph G by removing the vertices in X together with all edges incident to vertices in X, and

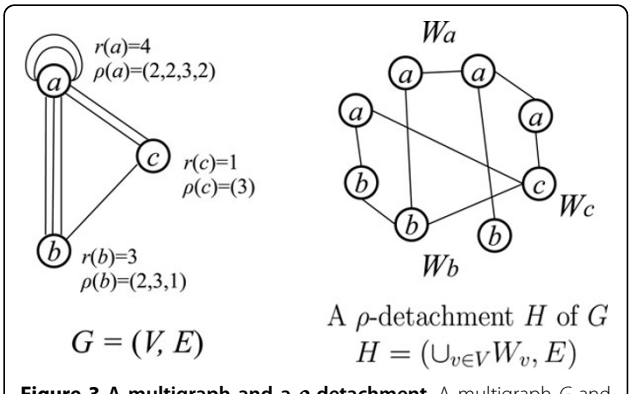

Figure 3 A multigraph and a *ρ*-detachment. A multigraph G and a ρ-detachment H of G.

d(*A*, *B*; *G*) *denotes the number of edges* $(u, v) \in E$ *with* $u \in A$ *and* $v \in B$.

Ishida *et al*. [22] proposed a bounding operation for the problem ETPF based on Theorem 2. However, we cannot use the bounding operation proposed by Ishida *et al*. for the problem ETULF due to upper and lower constraints. We now describe our new bounding operation based on detachments for the problem ETULF. The new bounding operation, called *detachment-cut* tests whether the current multitree T has a multitree that is consistent with given path frequencies among its descendants in the family tree, based on the difference between the feature vector $f_K(T)$ and the input feature vectors g_U and g_L.

Let $\ell_1, \ell_2, \ldots, \ell_s$ be input labels and $g_U, g_L : \Sigma^{\leq K+1} \to \mathbb{Z}_+$ be feature vectors. Let $r_0, \ldots, r_h$ be the vertices in the rightmost path to which a new leaf can be appended and $n_i^R (1 \leq i \leq s)$ denote the number of vertices r_j $(0 \leq j \leq h)$ with $\ell(r_j) = \ell_i$. For each label sequence t, $\#t$ denotes the number of paths P in T with $\ell(P) = t$. From g_U, g_L, and T, we define new feature vectors g'_U and g'_L of level $K = 1$ to be

$$g'_U(\ell_i) = \begin{cases} g_U(\ell_i) - \#\ell_i + n_i^R & (1 \leq i \leq s), \\ 1 & (i = s+1), \end{cases}$$

$$g'_U(\ell_i\ell_j) = \begin{cases} g_U(\ell_i\ell_j) - \#\ell_i\ell_j & (1 \leq i, j \leq s), \\ n_i^R & (1 \leq i \leq s, j = s+1), \end{cases}$$

$$g'_L(\ell_i) = \begin{cases} g_L(\ell_i) - \#\ell_i + n_i^R & (1 \leq i \leq s), \\ 1 & (i = s+1), \end{cases}$$

$$g'_L(\ell_i\ell_j) = \begin{cases} g_L(\ell_i\ell_j) - \#\ell_i\ell_j & (1 \leq i, j \leq s), \\ n_i^R & (1 \leq i \leq s, j = s+1). \end{cases}$$

We next introduce a vertex with a new label ℓ_{s+1} of valence $h + 1$ (for example, label A in Fig. 4), a graph $G_U = (V_U, E_U)$ with a vertex set $V_U = \{v_1, \ldots, v_s, v_{s+1} \mid \ell(v_i) = \ell_i, 1 \leq i \leq s+1\}$ and edge set $E_U = \{e_{ij} \mid e_{ij} = \{v_i, v_j\}, d(\{v_i\}, \{v_j\}; G_U) = g'_U(\ell_i\ell_j), 1 \leq i, j \leq s+1\}$, and a graph $G_L = (V_L, E_L)$ with a vertex set $V_L = \{v_1, \ldots, v_s, v_{s+1} \mid \ell(v_i) = \ell_i, 1 \leq i \leq s+1\}$ and edge set $E_L = \{e_{ij} \mid e_{ij} = \{v_i, v_j\}, d(\{v_i\}, \{v_j\}; G_L) = g'_L(\ell_i\ell_j), 1 \leq i, j \leq s+1\}$. Note that $d(\{v_i\}, \{v_j\}; G)$ means a multiplicity of the edge $\{v_i, v_j\}$ in a graph G. The function r and degree specification ρ are defined to be

$$r(v) = g'_U(\ell_i) \quad (1 \leq i \leq s+1),$$

$$\rho_i^v = \begin{cases} val(\ell(v_i)) & (v_i \notin \{r_0, \ldots, r_h\}, 1 \leq j \leq r(v)), \\ val(\ell(v_i)) - deg(v_i; T) + 1 & (v_i \in \{r_0, \ldots, r_h\}, 1 \leq j \leq r(v)). \end{cases}$$

Using G_U, G_L, r, and ρ, we can check if a current multitree T violates (C4). We need to check whether none of the following two conditions is violated.

(a) $deg(v; G_L) \leq \sum_{1 \leq i \leq r(v)} \rho_i^v \ (\forall v \in V_L)$.

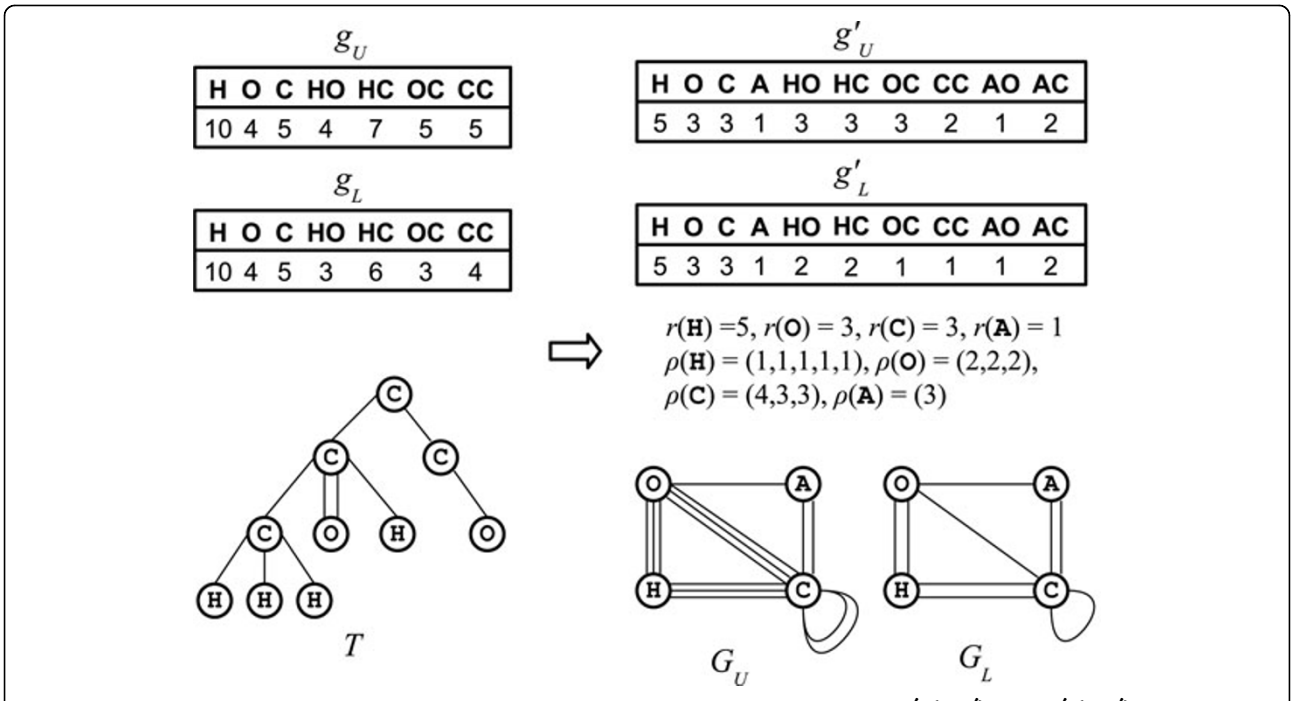

Figure 4 Detachment-cut. Bounding operation by detachment-cut, where vectors $g_U(\ell, \ell')$, $g_L(\ell, \ell')$, $g'_U(\ell, \ell')$, and $g'_L(\ell, \ell')$ are defined for unordered pairs $\{\ell, \ell'\}$ and those with value=0 are omitted in the tables.

(b) $r(X) + c(G_U - X) - d(X, V_U; G_U) \leq 1$ ($\forall X \subseteq V_U$, $X \neq \emptyset$).

In the first condition, we check whether the number of the rest of bonds is large enough to satisfy the lower feature vector constraint. In the second condition, we check whether T has a connected and loopless descendant based on G_U and Theorem 2.

Multiplicity-cut procedure

This subsection describes a new bounding operation based on multiplicity for the problem ETULF. Let $g(\ell)$ be the number of vertices with label $\ell \in \Sigma$ that are obtained from given the feature vector. Now we assume that $g(\ell)$ for all $\ell \in \Sigma$ are fixed in the problem ETULF. Then we can calculate the number of edges in output trees in the problem ETULF. Let n be the number of vertices in output trees. If we treat a multiple edge as a set of single edges, the number of edges e_m in an output tree is given by:

$$e_m = \frac{1}{2}\sum_{\ell \in \Sigma} val(\ell) g(\ell).$$

On the other hand, if we treat a multiple edge as a simple one, the number of edges e_s in an output tree is equal to $n - 1$ due to the tree-like constraint. Now we consider

$$M = e_m - e_s,$$

which means that only M edges are used to construct multiple bonds in an output tree. Note that $M \geq 0$. We calculate M from an input of the problem ETULF before the enumeration algorithm starts.

Let $T = (V, E)$ be a multitree, and m_e denote the multiplicity of $e \in E$. *The multiplicity* $M(T)$ *of* T is defined to be

$$M(T) = \sum_{e \in E} (m_e - 1).$$

Now we describe the *multiplicity-cut* based on $M(T)$ and M.

Let T be the current rooted multitree in the branching operation, $M(T)$ be the multiplicity of T, $RP(T) = (r_0, r_1, \ldots, r_k)$ be the rightmost path of T, T_i be the new rooted multitree obtained by appending a new leaf p to a vertex r_i ($0 \leq i \leq k$), and $RP(T_i)$ be the rightmost path of T_i . The rightmost path $RP(T_i)$ of T_i is updated by appending p to the end of $RP(T)$ when a new leaf p is appended to r_i, that is, $RP(T_i) = (r_0, r_1, \ldots, r_i, p)$. Then we can determine the multiplicities of the edges $\{(r_j, r_{j-1}), j = k, k-1, \ldots, i+1\}$ due to the valence constraint, at the same time, we update $M(T_i)$. We denote the multiplicity of an edge (r_j, r_{j-1}) in T_i by $Mul(r_j, r_{j-1} \mid T_i)$. When we update the multiplicity of the edge (r_j, r_{j-1}), $M(T_i)$ is updated as follows:

$$M(T_i) := \begin{cases} M(T) + Mul(r_k, r_{k-1} \mid T_i) - 1 & (j = k) \\ M(T_i) + Mul(r_j, r_{j-1} \mid T_i) - 1 & (i + 1 \leq j \leq k - 1). \end{cases}$$

By the definition of M, a valid multitree T_i satisfies

$$M(T_i) \leq M. \tag{3}$$

If T_i violates (3), then we discard T_i . See Fig. 5 for an illustration of this.

Results

This section reports the experimental results of our algorithm. First of all, we mention that the problem ETULF can be solved by applying the algorithm proposed by Ishida *et al.* [22] to each single feature vector in a given set of feature vectors, i.e., the problem ETULF can regard as a set of the problem ETPF. Then we call an algorithm for the problem ETULF based on the algorithm proposed by Ishida *et al.* RepEnum (Repeated Enumeration). On the other hand, we call our algorithm SimEnum (Simultaneous Enumeration). It is to be noted that RepEnum is one of the fastest tools to enumerate tree-like chemical

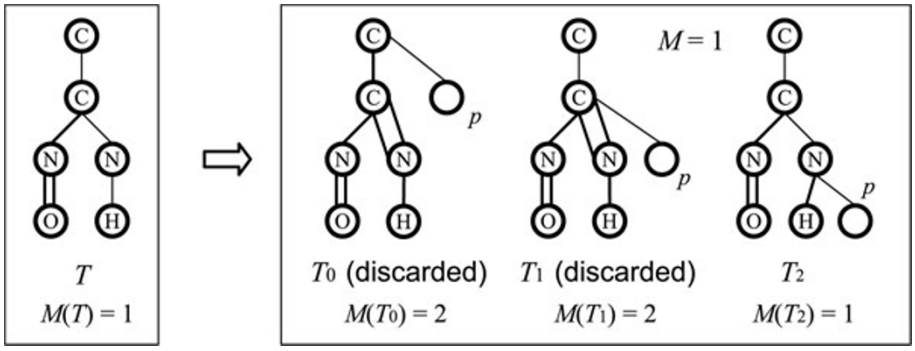

Figure 5 Multiplicity-cut. An illustration of the multiplicity-cut procedure, where $M = 1$.

structures from a given molecular formula (i.e., feature vector with $K = 0$) [22] and, to our knowledge, there does not exist any other available tool to enumerate chemical structures from a given feature vector based on path frequency (i.e., feature vector with general K).

Now we compare the performances of two algorithms, SimEnum and RepEnum, and we also compare the performances of two algorithms, SimEnum including multiplicity-cut and SimEnum not including multiplicity-cut. We have tested the algorithm SimEnum for some widths between upper and lower feature vectors. Tests were carried out on a PC with CPU AMD Athlon Dual Core Processor 5050e using instances based on some chemical compounds selected from the KEGG LIGAND database [25] (*http://www.genome.jp/ligand/*). Note that we treat a benzene ring contained in these compounds as a new virtual atom of valence six.

We define $w \in \mathbb{Z}_+$ to be a *width* between upper and lower feature vectors. From a feature vector g, we construct two feature vectors g_U and g_L as follows. For each entry $a > 0$ of g, let g_U be the upper feature vector, where each entry a_U is given by $a + w$ and g_L be the lower one,

Table 1 Comparison of previous method and our method

Entry Formula					SimEnum			RepEnum			
	n	K	w	f_v	time (s)	nodes	solutions	time (s)	nodes	solutions	solved
		1	1	3^{6}	1037.04	177,074,686	414,890	163.32	44,340,488	414,890	729
		2	1	3^{18}	2.97	392,246	44	T.O.	2,381,360,000	N.F.	65,909,572
		3	1	3^{34}	1.22	145,213	2	T.O.	3,293,260,000	N.F.	96,860,588
C00062	26	4	1	3^{53}	0.33	34,539	1	T.O.	2,780,050,000	N.F.	81,766,176
$C_6H_{14}N_2O_4$		5	1	3^{71}	0.24	20,361	1	T.O.	1,561,230,000	N.F.	45,918,529
		6	1	3^{85}	0.25	15,166	1	T.O.	569,590,000	N.F.	16,752,647
		7	1	3^{96}	0.18	14,547	1	T.O.	79,870,000	N.F.	2,349,117
		1	1	3^{6}	T.O.	377,260,000	N.F.	T.O.	413,000,000	N.F.	460
		2	1	3^{18}	7.24	845,760	25	T.O.	1,442,760,000	N.F.	70,175,902
		3	1	3^{31}	2.81	307,151	7	T.O.	3,316,970,000	N.F.	195,115,882
C03343	37	4	1	3^{47}	1.03	99,945	1	T.O.	2,494,780,000	N.F.	146,751,764
$C_{16}H_{22}O_4$		5	1	3^{64}	0.98	87,600	1	T.O.	1,050,480,000	N.F.	61,792,941
		6	1	3^{82}	0.76	60,194	1	T.O.	315,820,000	N.F.	18,577,647
		7	1	3^{99}	0.57	42,538	1	T.O.	41,450,000	N.F.	2,438,235
		1	1	3^{8}	T.O.	157,320,000	N.F.	T.O.	200,490,000	N.F.	1,388
		2	1	3^{26}	37.59	1,940,295	238	T.O.	2,911,390,000	N.F.	66,167,954
		3	1	3^{48}	1.71	60,792	3	T.O.	2,673,940,000	N.F.	60,771,363
C07178	46	4	1	3^{71}	0.35	14,248	1	T.O.	1,925,490,000	N.F.	43,761,136
$C_{21}H_{28}N_2O_5$		5	1	3^{92}	0.27	10,866	1	T.O.	743,940,000	N.F.	16,907,727
		6	1	3^{110}	0.27	10,680	1	T.O.	93,880,000	N.F.	2,133,636
		7	1	3^{125}	0.24	9,276	1	T.O.	19,270,000	N.F.	437,954
		1	1	3^{5}	T.O.	382,470,000	N.F.	T.O.	552,290,000	N.F.	61
		2	1	3^{16}	T.O.	211,800,000	N.F.	T.O.	530,930,000	N.F.	10,451,912
		3	1	3^{27}	1395.13	144,244,042	206	T.O.	3,314,260,000	N.F.	194,956,470
C03690	61	4	1	3^{41}	121.36	11,332,363	4	T.O.	2,392,530,000	N.F.	140,737,058
$C_{24}H_{38}O_4$		5	1	3^{57}	83.70	6,978,557	2	T.O.	958,650,000	N.F.	56,391,176
		6	1	3^{75}	40.11	2,923,819	1	T.O.	298,600,000	N.F.	17,564,705
		7	1	3^{92}	16.50	1,096,128	1	T.O.	38,670,000	N.F.	2,274,705

Comparison of SimEnum and RepEnum for the problem ETULF.

Note:

(1) C00062, C03343, C07178, and C03630 are the chemical compounds in the KEGG LIGAND database, respectively;

(2) n is the number of vertices in an instance preprocessed by replacing each benzene ring with a new atom having six valences;

(3) K is the level of given feature vectors;

(4) w is the width for constructing upper and lower feature vectors;

(5) f_v is the number of feature vectors in a given set;

(6) "time (s)" is the CPU time in seconds;

(7) T.O. means "time over" (the time limit is set to be 1,800 seconds);

(8) "nodes" is (the sum of) the number of nodes of family trees that are traversed;

(9) "solutions" is the number of all possible solutions;

(10) "solved" is the number of feature vectors which the algorithm RepEnum solved in the time limit; and (11) N.F. means "not found."

where each entry a_L is given by $\max\{0, a - w\}$. Note that if $w = 0$, then an instance for the problem ETULF is equivalent for the problem ETPF.

Table 1 and Additional file 1 show the results of the comparison. We find that the algorithm RepEnum cannot solve all the problems with $K = 2$ within the time limit since the number of feature vectors in a given set is exponentially increasing with K. On the other hand, Table 1 shows that the algorithm SimEnum can solve the problem much faster for a larger K. This shows that the algorithm SimEnum runs significantly faster than the algorithm RepEnum. It is also seen that RepEnum can only examine a very small portion of feature vectors in most cases. Additional file 1 shows that the algorithm SimEnum including multiplicity-cut runs faster than the algorithm SimEnum not including multiplicity-cut for almost all of the instances. This shows that the multiplicity-cut operation works well to improve enumeration efficiency.

Table 2 shows the results on the performance for varying width w for the problem ETULF. The search space in the problem ETULF is exponentially increasing with w. However, it seems that the number of search nodes and computation time are not exponentially increasing with w. This suggests that the algorithm SimEnum works efficiently for the large search space in the problem ETULF.

Here, we briefly discuss practical values on K and w though we do not have concrete evidence and these values depend on target classes of chemical compounds. It is suggested from the results on similar feature vectors [9,10,15] that K between 3 to 10 should be used. Though there is no previous result on w, it is seen from Table 2 that w cannot be large because there may exist too many solutions. Therefore, w less than 4 should be used.

Table 2 Comparison of varying width

Entry Formula				SimEnum		
	n	K	w	time (s)	nodes	solutions
		2	0	0.51	55,196	6
		2	1	3.58	400,501	44
		2	2	7.58	835,509	503
C00062	26	2	3	10.84	1,163,548	2,351
$C_6H_{14}N_2O_4$		2	4	12.55	1,349,057	5,430
		2	5	13.29	1,431,075	9,852
		2	50	14.31	1,537,496	25,425
		2	0	0.34	35,952	9
		2	1	8.39	845,760	25
		2	2	48.27	4,815,369	41
C03343	37	2	3	149.83	14,781,738	305
$C_{16}H_{22}O_4$		2	4	377.01	37,435,878	40,732
		2	5	639.68	63,459,180	106,870
		2	50	1118.75	110,703,034	510,079
		2	0	2.33	111,781	16
		2	1	46.81	2,246,578	238
		2	2	96.52	4,715,072	1,375
C07178	46	2	3	152.18	7,420,060	6,824
$C_{21}H_{28}N_2O_5$		2	4	179.42	8,744,563	19,180
		2	5	199.66	9,677,513	29,891
		2	50	255.01	12,292,587	54,861
		5	0	19.50	1,482,017	2
		5	1	220.14	16,063,569	5
		5	2	439.12	33,037,741	32
C03690	61	5	3	684.88	52,207,745	178
$C_{24}H_{38}O_4$		5	4	1024.96	78,509,554	349
		5	5	1285.55	98,762,291	615
		5	50	T.O.	136,835,134	N.F.

Comparison of the performance for varying w for the problem ETULF.

Conclusions

We considered the problem of enumerating all tree-like chemical graphs from a given set of feature vectors, which is specified by upper and lower feature vectors based on frequencies of paths, and proposed a new exact branch-and-bound algorithm. Our experimental results show that our algorithm outperforms the naive algorithm based on a previous method. In comparison to the algorithm based on Ishida *et al.* [22], our algorithm can greatly reduce the number of search nodes and the computation time and enumerate all the feasible solutions in many instances.

However, the search space of the problem ETULF is much larger than that of the problem ETPF due to upper and lower constraints and in fact there are many search nodes for solving the problem ETULF by our algorithm. One of the future works is to improve the bounding operations, or introduce a new bounding operation. Actually, in the feature-vector-cut mentioned in subsection , information of a lower feature vector g_L is only used if $|T| = n$. Another future work is to develop a web server that implements our proposed algorithm. Generalization of the proposed techniques for other types of kernel functions and other problems is also left as a future work.

Additional material

Additional file 1: Comparison of multiplicity-cut Comparison of SimEnum including multiplicity-cut and SimEnum not including multiplicity-cut for the problem ETULF. Note: (1) "add multiplicity-cut" is the algorithm SimEnum including multiplicity-cut; and (2) "no multiplicity-cut" is the algorithm SimEnum not including multiplicity-cut.

Acknowledgements
This work was partially supported by Grant-in-Aid #22240009 from Mext, Japan.

This article has been published as part of *BMC Bioinformatics* Volume 12 Supplement 14, 2011: 22nd International Conference on Genome Informatics: Bioinformatics. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/12?issue=S14.

Author details
[1]Graduate School of Informatics, Kyoto University, Yoshida, Kyoto 606-8501, Japan. [2]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan.

Authors' contributions
HN gave the basic idea based on discussions with TA and MS. MS developed and implemented the algorithms, and carried out the experiments. MS, HN, and TA authored and approved the manuscript.

Competing interests
The authors declare that they have no competing interests.

Published: 14 December 2011

References

1. Bytautas L, Klein DJ: **Chemical combinatorics for alkane-isomer enumeration and more.** *Journal of Chemical Information and Computer Sciences* 1998, **38**:1063-1078.
2. Bytautas L, Klein DJ: **Formula periodic table for acyclic hydrocarbon isomer classes: combinatorially averaged graph invariants.** *Physical Chemistry Chemical Physics* 1999, **1**:5565-5572.
3. Bytautas L, Klein DJ: **Isomer combinatorics for acyclic conjugated polyenes: enumeration and beyond.** *Theoretical Chemistry Accounts* 1999, **101**:371-387.
4. Cayley A: **On the analytic forms called trees with applications to the theory of chemical combinations.** *Reports British Association for the Advancement of Science* 1875, **45**:257-305.
5. Buchanan BG, Feigenbaum EA: **DENDRAL and Meta-DENDRAL: their applications dimension.** *Aritificial Intelligence* 1978, **11**:5-24.
6. Funatsu K, Sasaki S: **Recent advances in the automated structure elucidation system, CHEMICS. Utilization of two-dimensional NMR spectral information and development of peripheral functions for examination of candidates.** *Journal of Chemical Information and Computer Sciences* 1996, **36**:190-204.
7. Fink T, Reymond JL: **Virtual exploration of the chemical universe up to 11 atoms of C, N, O, F: assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery.** *Journal of Chemical Information and Computer Sciences* 2007, **47**:342-353.
8. Mauser H, Stahl M: **Chemical fragment spaces for de novo design.** *Journal of Chemical Information and Computer Sciences* 2007, **47**:318-324.
9. Faulon JL, Churchwell CJ, Jr DPV: **The signature molecular descriptor. 2. Enumerating molecules from their extended valence sequences.** *Journal of Chemical Information and Computer Sciences* 2003, **43**:721-734.
10. Hall LH, Dailey ES: **Design of molecules from quantitative structure-activity relationship models. 3. Role of higher order path counts: path 3.** *Journal of Chemical Information and Computer Sciences* 1993, **33**:598-603.
11. Deshpande M, Kuramochi M, Wale N, Karypis G: **Frequent substructure-based approaches for classifying chemical compounds.** *IEEE Transactions on Knowledge and Data Engineering* 2005, **17**:1036-1050.
12. Bakir GH, Weston J, Schölkopf B: **Learning to find pre-images.** *Advances in Neural Information Processing Systems* 2003, **16**:449-456.
13. Bakir GH, Zien A, Tsuda K: **Learning to find graph pre-images.** *Lecture Notes in Computer Science* 2004, **3175**:253-261.
14. Kashima H, Tsuda K, Inokuchi A: **Marginalized kernels between labeled graphs.** *Proceedings of the Twentieth International Conference on Machine Learning, AAAI Press* 2003, 321-328.
15. Mahé P, Ueda N, Akutsu T, Perret JL, Vert JP: **Graph kernels for molecular structure-activity relationship analysis with support vector machines.** *Journal of Chemical Information and Modeling* 2005, **45**:939-951.
16. Byvatov E, Fechner U, Sadowski J, Schneider G: **Comparison of support vector machine and artificial neural network systems for drug/nondrug classification.** *Journal of Chemical Information and Computer Sciences* 2003, **43**:1882-1889.
17. Akutsu T, Fukagawa D: **Inferring a graph from path frequency.** *Lecture Notes in Computer Science* 2005, **3537**:371-392.
18. Nagamochi H: **A detachment algorithm for inferring a graph from path frequency.** *Algorithmica* 2009, **53**:207-224.
19. Fujiwara H, Wang J, Zhao L, Nagamochi H, Akutsu T: **Enumerating treelike chemical graphs with given path frequency.** *Journal of Chemical Information and Modeling* 2008, **48**:1345-1357.
20. Nakano S, Uno T: **Generating colored trees.** *Lecture Notes in Computer Science* 2005, **3787**:249-260.
21. Nakano S, Uno T: **Efficient generation of rooted trees.** *NII Technical Report NII-2003-005E* 2003.
22. Ishida Y, Zhao L, Nagamochi H, Akutsu T: **Improved algorithms for enumerating tree-like chemical graphs with given path frequency.** *Genome Informatics* 2008, **21**:53-64.
23. Ishida Y: **Improved algorithms for enumerating tree-like chemical graphs with given path frequency.** *Master thesis of Graduate School of Informatics in Kyoto University* 2008.
24. Kvasnicka V, Pospichal J: **Constructive enumeration of acyclic molecules.** *Collect Czech Chem Commun* 1991, **56**:1777-1802.
25. Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M: **KEGG for representation and analysis of molecular networks involving diseases and drugs.** *Nucleic Acids Res* 2010, **36**:D355-D360.

doi:10.1186/1471-2105-12-S14-S3
Cite this article as: Shimizu *et al.*: Enumerating tree-like chemical graphs with given upper and lower bounds on path frequencies. *BMC Bioinformatics* 2011 **12**(Suppl 14):S3.