

Modeling, QBE-like Query and Processing for Metabolic Pathway Data¹

針對生化代謝途徑之資料模組、範例式查詢以及查詢處理演算法

Ching-Yang Chang(張青揚) and Jia-Lien Hsu(徐嘉連)²

Department of Computer Science and Information Engineering,
Fu Jen Catholic University, Taiwan, R.O.C.

alien@csie.fju.edu.tw

摘要

隨著生物科技邁入後基因時代，生物資訊研究的重心由基因體與蛋白質逐漸擴展到生化代謝途徑上。綜觀目前的系統，針對生化代謝途徑所提供的查詢功能，主要以關鍵字(keyword-based)的查詢方式為主，例如輸入參與反應的化合物(compound)或催化反應的酵素(enzyme)為查詢的條件。在本篇論文中，我們把多媒體資料內涵式查詢(multimedia content-based retrieval)的概念方法，延伸應用到生物代謝途徑的查詢上。我們提供另一種查詢方式，稱為範例式查詢(query by example)。首先，我們以關連性資料模組(relational data model)來儲存生化代謝途徑(metabolic pathway)。並且設計一範例式查詢(query-by-example)的查詢介面，讓使用者可依照範例式查詢到相關代謝途徑。對於這類資料的查詢，我們分別針對「化學反應網路(chemical networks)」和「酵素網路(enzyme networks)」歸納四種查詢型態(query type)。並分別針對這兩種網路提出索引結構(indexing structure)的索引結構表(inverted-list table)及針對每個查詢型態提出查詢處理演算法(query processing algorithm)。最後，我們以 KEGG XML 文件為資料來源，將文件經過 XML parser 存放在我們設計的關聯式資料表中。依據我們的索引結構與查詢處理演算法，實作出範例式查詢(query-by-example)的查詢系統。

關鍵字：範例式查詢(query-by-example)、生化代謝途徑(metabolic pathway)、索引結構表(inverted-list table)、查詢處理演算法(query processing algorithm)

1 序論

隨著新的生物技術的快速進展，生化反應途徑的角色日漸受到重視，許多生化反應途徑資料庫紛紛成立。就我們所知道的生化反應途徑資料庫有 KEGG、BioCyc、MetaCyc、What is There (WIT) 以及 CASE 等，KEGG 是日本京都大學化學研究所

的研究小組，所開發的一個生物資料庫[KEGG]。目前也是日本 GenomeNet 伺服器上最核心的一個資料庫。其中的 pathway 資料庫，記載相關的生化反應途徑等資訊。KEGG Markup Language (KGML) [KEGG][Kane00][Kane02]是一 KEGG 圖形物件的交換格式。KGML 能夠自動化的將 KEGG 代謝途徑呈現出，對於計算、分析酵素網路及化學反應網路都很有幫助。

XML 已成為資料表示的趨勢。文件資料在經過 XML[Bray00]標示後，而形成的 XML 文件，能夠容易的在網際網路上被傳遞、交換與使用。XML 對文件資料的描述能力遠比 HTML 強，因為從控制文件標籤名稱的定義中即可清楚的知道資料代表的意義。對於文件而言，XML 更具結構性能夠充滿描述資料本身的特性。KGML 利用 XML 語法描述 pathway 的資料，能夠在網際網路上作資料交換與分享。

KGML 利用 XML 語法表示複雜的生化代謝途徑，而查詢方式是關鍵字查詢。對於目前查詢 XML 資料的標準，是由 W3C 發佈的 XQuery[Boag]，然而 XQuery 對於 naive user 而言，太複雜且困難。針對查詢這麼複雜的生化代謝途徑(metabolic pathway)圖形，我們提出的解決方式就是設計一範例式查詢(Query by example, QBE)生化代謝途徑的功能。就是由使用者給予一個生化代謝途徑(metabolic pathway)樣本(sample)的方式來代表我們的查詢語言，也是一種所謂的「內涵式查詢」。

另外，KGML 儲存了一些與化學反應式或化學關係式無關的資料。例如，各 entry 的相關位置、屬性及生化代謝途徑(metabolic pathway) graphics 的分類等。對於查詢處理而言，KGML 的資料過於繁雜(redundant)，並且儲存生物資料格式與範例式查詢圖示無直接關係。針對 KGML 的資料，我們利用 XML 概念性模型(conceptual modeling)的概念將這些複雜的生化代謝途徑的文件轉成關聯式資料表。提出一個精簡且容易表示查詢的表示式(即 QBE 查詢圖示)，以及一友善的查詢介面。為了有效找到酵素(enzyme)、化合物(compound)、化學反應式及關係式，我們利用我們所提出的索引結構表

¹ 本論文研究為國科會補助之研究成果，計畫編號為 NSC-93-2213-E-030-005

² 本論文的相關聯絡人(To whom all correspondence should be sent).

(inverted-list table)，並針對使用者所下的範例式查詢作處理 (query processing)。

QBE 的查詢系統之系統架構圖如圖 1 所示。QBE 的查詢系統主要分為兩個部份：資料模型(data modeling)、索引和查詢處理(indexing and query processing)。

資料模型(data modeling)：主要是將 KEGG 之 XML 文件的資料，經由 XML parser module 處理後，分別儲存在我們設計的關聯式資料表中。

索引查詢處理(indexing and query processing)：將資料庫中的資料表載入至我們所設計的索引結構表中。根據使用者透過 Query interface module 給予 QBE 查詢，query processing module 利用索引結構 (indexing structure) 的索引結構表(inverted-list table) 來加速查詢處理，最後再透過 Query interface module 回傳相關結果給使用者。

本論文的架構如下：第 2 節將針對相關文獻做探討，第 3 節將對生化代謝途徑的基本觀念及我們所提出的儲存方式做一介紹，第 4 節針對範例式查詢，我們提供的查詢的型態，查詢處理演算法，第 5 節我們系統實作 (implementation) 的部份，第 6 節為本文結論與未來工作。

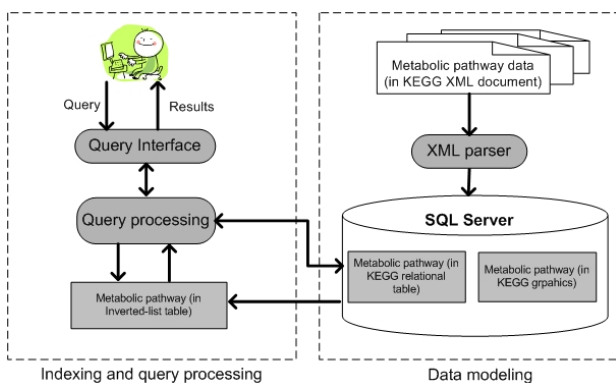


圖 1：系統架構圖

2 文獻探討

KGML 使用 XML 語法描述代謝途徑，能夠自動化將複雜的圖形呈現出。但，對於這麼複雜的代謝途徑圖形卻無提供一 QBE 查詢介面。BioCyc (<http://biocyc.org/>) 是專門收集生化反應途徑與基因組相關的知識庫。其中的 MetaCyc (<http://metacyc.org/>) 是一專門整理代謝生化反應途徑的資料庫，其查詢相關反應途徑的方式包含關鍵字的查詢、階層式的目錄查詢。網路上為人所熟知的代謝途徑資源 What is There (WIT; <http://wit.mcs.anl.gov/WIT2/>) 也是以關鍵字為基礎的查詢，WIT 一開始設計是針對生物專家所開發，其專門字彙相關說明對 naïve user 而言不能滿足所需，也就是說查詢 WIT 需先具備這些字彙的知識。另外一個有名的生物反應途徑資料庫 CASE Pathways Database System[CASE]，查詢方式也是以關鍵字及階層式的目錄 (tree-structured query) 查詢為主。另外，如 Scott[Scot04] 等人所提出的查詢介面，針對 pathway 原始 XML 資料，所作的下拉式

(drop-down box) 選單查詢方式。查詢介面會因為 XML 資料過於龐大，下拉式選單也相對過於龐大。

GXL (Graph eXchange Language) [Wint] 是一種圖形化為基礎 (graph-based) 標準交換格式的工具。據我們所知，雖然 GXL 利用 XML 語法將圖形架構表現出來，對於表示及儲存生化代謝途徑這類圖形是滿適合。不過此類圖形化表示語言與圖形化查詢語言，至目前為止並無一定標準，仍需依據目前對於 XML 的標準資料查詢語言 XQuery。XQBE[Brag03][Brag04] 為範例式查詢語言，查詢是依據 XML 原始階層式結構給予查詢樣本。不過對於查詢 network-like 的資料 (如生化代謝途徑等資料)，利用 XQBE 給以一 network-like 的圖形為查詢樣本，因為與 KGML 原始架構不符。再加上 XQBE 的查詢強制轉化成 XQuery 會使的錯誤更加嚴重。Krishnamurthy *et al.* 利用 adjacency-list[Keis03] 將原始 pathway 儲存，只考慮單一 pathway，其儲存結構對多個 pathway 則難以表示。

在 XML 文件的查詢處理方面，傳統查詢處理演算法如 Top-down 及 bottom-up 等方式，或結合以上兩種的 hybrid approach 方式[McHu99]。不管哪種方式都必須橫跨(traversal)整個 XML 資料庫一次以上。對於處理(processing)的效能而言太過於沉重。為了不必要橫跨(traversal)整個 XML 資料庫，在查詢處理演算法上，是利用作索引(index)的方式來加速查詢。如：B+ tree [Li01]；CTree [Zou04]；PathGuide [Chen03]；DataGuide [Gold97]；CADG [Weig04]；T-indexs [Milo99]；Index Fabric [Coop01]；APEX [Chun02]；ViST [Wang03]；R*-tree [Bech90]；XR-tree [Jian03] 等。對 XML 之樹狀結構的查詢，除了作索引來加速找到所需之路徑或元素。另外，增進 XML 查詢處理演算法效能的重要核心就是如何在資料庫中快速找到所有 parent-child 及 ancestor-descendant relationships。基於 numbering scheme，透過索引資料表(index table)找到元素(element)、屬性(attribute)或是路徑(path)之後的 merge join (*i.e.*, structural join)，也就是快速決定元素與元素之間，在階層式 XML 樹上關係，變得相當重要。例如：Quanzhong 所提，利用 EE-join、EA-join 及 KC-join 演算法[Li01]；Shurug 提出的 Tree-merge 及 Stack-tree 演算法[Alkh02]；Bruno 所提出的 TwingStack [Brun02] 的 join 演算法；Jiang 所提出的 Holistic Twig Joins 演算法[Jian03] 等相關方法都是為了解決 structural join 所設計之方法。以上之 structural join 演算法相當於從原始 XML 中找到符合條件的子圖形(sub-graph)。但是 KGML，其文件並不是一般 XML 文件。KGML 將一複雜代謝途徑拆解成一個個化學反應式 (或關係式) 來儲存。如果使用者是給以一生化代謝途徑(metabolic pathway) 樣本的查詢，查詢處理演算法並不是找出符合條件的子圖形(sub-graph) 這麼簡易，而是先找到查詢的 pathway 各片段化學式 (或關係式)，在將各片段化學式 (或關係式) join 成符合使用者的下的查詢。所以，對於 KGML 此儲存結構而言，並不適合利用

numbering scheme 的方法在 XML tree 上做編碼，或應用以上所提之各個 structural join 演算法。

3 生化代謝途徑 (METABOLIC PATHWAY)

在生物化學 (biochemistry) 中，生化代謝途徑 (metabolic pathway) 是指「一連串包含在一個 cell 中，需藉由酵素來催化的化學反應式」。

根據 Mehmet Koyuturk 等人在 Bioinformatics 期刊，所提出生化代謝途徑 (metabolic pathway) 的定義如下[Koyu04]：

定義 1：生物代謝途徑(化學網路)

一個生物代謝途徑 $P(M, Z, R)$ ，是代謝物 (metabolites; M)、酵素 (enzymes; Z) 和反應 (reactions; R) 的集合。其中每一個反應 (reactions; R) $r \in R$ ，是定義在酵素 $Z(r) \subseteq Z$ 、受質 (substrates) $S(r) \subseteq M$ 、產物 (products) $T(r) \subseteq M$ 的關聯。

請參考以下的英文定義：

Definition 1 : Metabolic pathway (chemical network)
“A metabolic pathway $P(M, Z, R)$ is a collection of metabolites M , enzymes Z , and reactions R , where each reaction $r \in R$ is associated with a set of enzymes $Z(r) \subseteq Z$, a set of substrates $S(r) \subseteq M$, and a set of products $T(r) \subseteq M$.” From[Koyu04].

生化代謝途徑，即是連接一連串的化學代謝物 (metabolites)，透過受質—產物 (substrate-product) 的關係來表示一連串的化學反應式 [Koyu04][Chen04][Sira02]。如附錄 (A) Pentose phosphate pathway 就是一生化代謝途徑的圖形。簡單來說，生化代謝途徑就是被定義成一連串反應式的組合。舉例來說，兩個化學反應式 (reaction) 為 $A \xrightarrow{e1} B$ 、 $B \xrightarrow{e2} C$ 。以 $A \xrightarrow{e1} B$ 為例，透過 substrate-product 的關聯性來表示即為 $A-B$ ， $B \xrightarrow{e2} C$ 即為 $B-C$ 。A、B 和 C 是化合物 (compound)，“e1”、“e2”分別代表 enzyme 1、enzyme 2，其中酵素 (enzyme) 的角色只是促進反應速度加速。其中 B 為 $A \xrightarrow{e1} B$ 反應式的產物，為 $B \xrightarrow{e2} C$ 反應式的受質，把 A、B 和 C 這三個代謝物透過受質—產物 (substrate-product) 的關聯性串連 (link) 起來，則 $A \xrightarrow{e1} B \xrightarrow{e2} C$ 就可稱為一生化代謝途徑 (metabolic pathway)，如圖 2 所示。這樣所形成的 graph 稱之為「化學反應網路」。在生化代謝途徑 (metabolic pathway) 的網路中，每一個節點 (node) (i.e., compound) 都是不同的，且路徑 (path) 也是獨一無二的 [Fors99]。

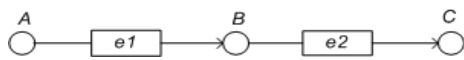


圖 2： $A \xrightarrow{e1} B \xrightarrow{e2} C$ 的生化代謝途徑

針對生化代謝途徑 (metabolic pathway) 的資料，我們先針對 KGML 的儲存方式做討論，以及我們所提出以關聯式網要來儲存 KGML XML 文件的方法，分別說明如下：

3.1 以 KGML 儲存代謝途徑 (Pathway in KGML)

在 KGML 的 pathway 圖形中，entry elements 視為節點 (node)，化學關係式 (relation) 及化學關係式元素 (reaction elements) 視為邊緣 (edge)，化學關係式 (relation) 與化學反應式 (reaction) 分別指的是連接到圖示中的方型與圓型之間連接的狀況。entry 與化學關係式元素 (reaction elements) 所形成的圖示被定義為蛋白質網路 (或是酵素網路)，entry 與化學關係式元素 (reaction elements) 所形成的圖示稱為化學反應網路 (chemical network)。所以生化代謝途徑被視為蛋白質網路 (亦稱酵素網路) 及化學反應網路 (如附錄 (A))，而 regulatory pathways 只被視為蛋白質網路 [KEGG]。本篇論文的重點將著重在生化代謝途徑上。

KGML 使用 XML 語法描述生化代謝途徑，其儲存的樹狀結構圖如圖 3 所示。圖 3 的 Part(A) 的部份是記錄 pathway 相關資訊如途徑名稱、圖形編號等。在 KGML 中，一個 pathway element 就是代表一個 pathway map 也是代表一份 KEGG XML 文件；Part(B) 記錄出現在此 pathway 中的所有 entry (i.e., 代謝途徑中的節點 (node))，其中包括化合物 (compound)、酵素 (enzyme)；graphic element 即是描述節點 (node) 的一些資訊以及在圖形中的位置；有些節點 (node) 較複雜可能是由數個節點 (node) 所組成一個節點 (node)，組成元素 (component element) 則是描述此節點 (node) 其中包含那些基因 (genes)。Part(C) 是描述化學反應式 (reaction)，Part(D) 描述化學關係式 (relation)。以下我們對於 Part(C) 與 Part(D) 的部分詳細說明。

我們以戊糖磷酸途徑 (Pentose phosphate pathway) 的文件 (document) 為例 [KEGG]，在文件 (document) 中，記錄了一連串複雜的化學反應式、關係式、代謝物及其屬性，如化學反應式這個元素其中包含反應式的名稱、型態、相關生成產物等。以下我們就針對 KGML 描述生化代謝途徑作簡易的說明。並且針對這些資料的性質，提出我們將如何儲存這些資料的模組。

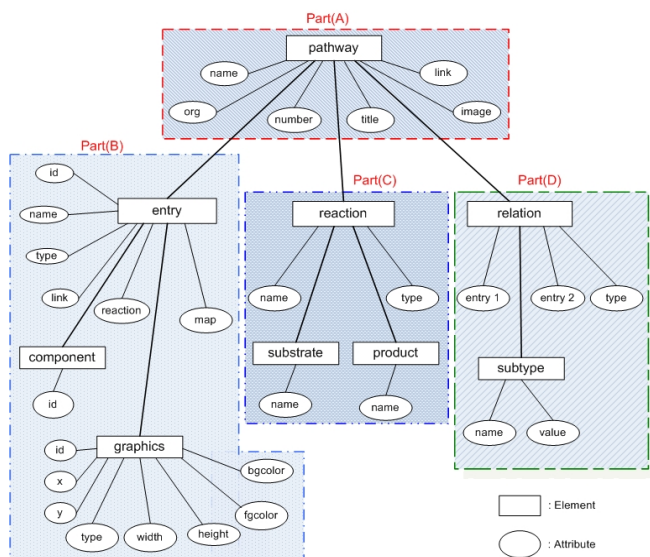


圖 3：KGML 的生化代謝途徑的樹狀結構(Metabolic pathway tree structure of KGML)

以下針對圖 3 part(C) 的化學關係式元素 (relation element) 做說明。舉例來說，範例 1 之表示式就是一化學反應表示式，我們藉此例子來說明化學反應式。

範例 1：

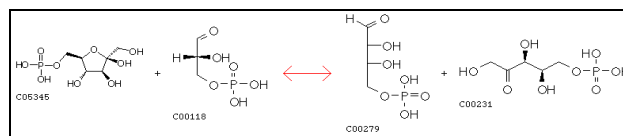


圖 4：C05345 + C00118 ↔ C00279 + C00231 (出自於 KEGG Pathway Database[KEGG])

KGML 利用 XML 語法描述此化學反應式，如圖 5 之文件。圖 6 是 KEGG 呈現此化學反應式所代表之圖形。其中 r n: R01830 為此化學反應的名稱，C05345、C00118、C00234 及 C00279 為化合物 (compound)，以小圓圈表示；ec 是酵素的分類，2.2.1.1 則為酵素的用矩形表示；C05345、C00118 為受質 (substrate)，C00234 及 C00279 為產物 (product)，因為此反應式可逆反應 (reversible) 所以以雙箭頭表示。

```
<entry id="10" name="ec:2.2.1.1" type="enzyme" reaction="rn:R01830"
link="http://www.genome.ad.jp/dbget-bin/www_bget?enzyme+2.2.1.1">
<graphics name="2.2.1.1" fgcolor="#000000" bgcolor="#FFFFFF" type="rectangle"
x="408" y="381" width="45" height="17" />
</entry>
<reaction name="rn:R01830" type="reversible">
<substrate name="cpd:C05345" />
<substrate name="cpd:C00118" />
<product name="cpd:C00279" />
<product name="cpd:C00231" />
</reaction>
```

圖 5：KGML 描述「化學反應式」“C05345+C00118 ↔ C00279+C00231” (節錄自 KEGG 之 XML 文件 [KEGG])

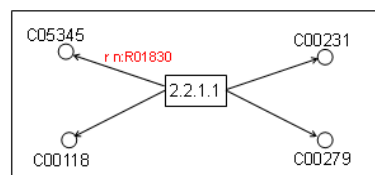


圖 6：反應式“C05345 + C00118 ↔ C00279 + C00231”之圖形

根據 Mehmet Koyuturk 等人在 Bioinfotmatics 期刊對生物代謝途徑與化學關係式之間的關係定義如下[Koyu04]：

定義二：生物代謝途徑(酵素網路)

給予一個生物代謝途徑 $P(M, Z, R)$ ， P 的關聯有向圖(associated directed graph) $G(V, E)$ 的建構如下：對於任何的酵素 $z_i \in Z$ ，再圖中都有一個對應的節點 $v_i \in V$ ，對於每一個 edge，都存在著如下的關聯： $(v_i, v_j) \in E$ if and only if $\exists r_1, r_2 \in R$, such that $z_i \in Z(r_1)$, $z_j \in Z(r_2)$, and $T(r_1) \cap S(r_2) \neq \emptyset$ 。”

請參考以下的英文定義：

DEFINITION 2：Metabolic pathway (enzyme network)

“Given metabolic pathway $P(M, Z, R)$, the associated directed graph $G(V, E)$ of P is constructed as follows: for any enzyme $z_i \in Z$, there is a node $v_i \in V$. There is an edge from v_i to v_j , i.e. $(v_i, v_j) \in E$ if and only if $\exists r_1, r_2 \in R$, such that $z_i \in Z(r_1)$, $z_j \in Z(r_2)$, and $T(r_1) \cap S(r_2) \neq \emptyset$.” From [Koyu04].

以上定義所提到的 P 的關聯有向圖 G ，其中圖 G 即代表化學關係式。這指的是在生化代謝途徑 (metabolic pathway) graph 中，存在一個邊緣，從一酵素連接至另一個酵素，這樣所形成的圖稱之為「酵素網路」。針對圖 3 part(D) 的化學關係式元素，我們再以範例 2 的例子來詳細說明，圖 8 即是描述一個化學關係式。

範例 2：

根據 KGML 描述化學關係式 (relation) 的文件如圖 7 所示，圖 8 為圖 7 之 KGML 描述關係式所呈現圖形。

³ International Union of Biochemistry and Molecular Biology (IUBMB) 1965 年命名系統化，將酶統一分類。以 Example 1 為例，其命名均以編號來表示，以下一化學反應式如：
beta-D-Fructose 6-phosphate + (2R)-2-Hydroxy-3-(phosphonoxy)-propanal ↔ D-Erythrose 4-phosphate + D-Xylulose 5-phosphate，即以 C05345 + C00118 ↔ C00279 + C00231 表示。

```
<relation entry1="25" entry2="42" type="ECrel">
  <subtype name="compound" value="48" />
</relation>
```

圖 7：KGML 描述「化學關係式」（節錄自 KEGG 之 document [KEGG]）

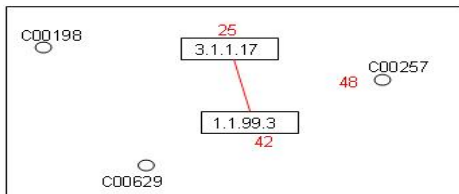


圖 8：KGML 描述圖 7「化學關係式」所呈現之圖形，其中 25、42、48 是 id identifier

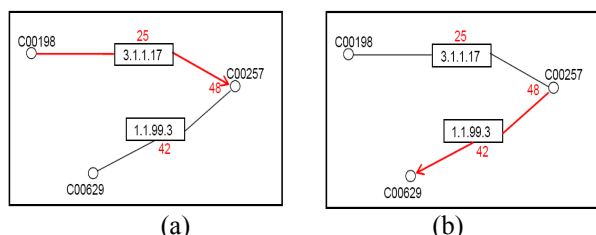


圖 9：(a)描述反應式“C00198 → C00257”圖示，其中加入 enzyme(ec:3.1.1.17)當催化劑。(b)描述反應式“C00257 → C00629”之圖示，其中加入 enzyme(ec:1.1.99.3)當催化劑。

化學關係式 (relation) 描述化學反應式 (reaction) 中化合物(compound)及酵素(enzyme)的關係。舉例來說，圖 9-(a) “C00198→C00257”此反應式中加入的酵素 (ec:3.1.1.17)，另一反應式圖 9-(b) “C00257→C00629”此反應式加入的酵素 (ec:1.1.99.3)。我們根據了 KGML 描述化學關係式，所以記錄了酵素 (ec:3.1.1.17)、酵素 (ec:1.1.99.3) 及化合物(compound) (cpd:C00257) 之間的關係，如圖 8 即描述此化學關係式。

3.2 以關聯性資料庫儲存代謝途徑 (Pathway in Relational Database)

在本節中，我們利用概念性模型 (conceptual modeling) 來建構生化代謝途徑資料文件的關聯式綱要 (relational schema)。將 XML 文件存在放關聯式資料庫。我們將每個元素當作一個資料表，元素中每個 attribute 當作關聯式綱要。

- 每一個元素變成一個關聯性資料表，所有具相同名稱的元素是泛指同一個資料表。
- 每一個 XML 屬性變成一個關連性 (relational attribute)。在關聯式資料表中，必須由設計者自定鍵值(key)來當鑑定人(identifier)，不過 XML 文件本身並沒有拿某個屬性當鍵值。所以將 XML 儲存在關聯式資料表時必須選擇哪些屬性當鍵值。
- 每個元素與子元素合併 (merge) 成一個資料

表。

不過由於 KGML 記錄了太多與查詢處理無關 (redundant) 的資料，例如 graphics 這個元素。Graphics element 是記錄化合物(compound)或酵素(enzyme)在生化代謝途徑圖形中的位置等相關資訊，對於查詢處理而言並不重要。所以我們只儲存一些我們認為重要的資料表以及擷取一些重要的屬性。以下我們以戊糖磷酸途徑的 XML 文件 (節錄自 KEGG) 為例，並將它轉成我們所需要的關聯式綱要及資料表。KEGG 之文件樹狀結構如圖 3，我們取出重要元素、子元素及重要屬性。我們以範例 3 來做說明：

範例 3：

我們利用 XML 概念性模型描述 XML 文件關聯性的概念，來說明戊糖磷酸途徑之片斷 XML 文件如圖 10 所示。

```
<pathway name="path:map00030" org="map" number="00030" title="Pentose phosphate pathway"
  image="http://www.genome.jp/kegg/pathway/map/map00030.gif"
  link="http://www.genome.jp/dbget-bin/show_pathway?map00030">
  <entry id="1" name="ec:2.7.1.15" type="enzyme" reaction="rn:R02750"
    link="http://www.genome.jp/dbget-bin/www_bget?enzyme+2.7.1.15">
    <graphics name="2.7.1.15" fgcolor="#000000" bgcolor="#FFFFFF" type="rectangle" x="180"
      y="591" width="45" height="17" />
  </entry>
  <entry id="3" name="ec:4.1.2.4" type="enzyme" reaction="rn:R01066"
    link="http://www.genome.jp/dbget-bin/www_bget?enzyme+4.1.2.4">
    <graphics name="4.1.2.4" fgcolor="#000000" bgcolor="#FFFFFF" type="rectangle" x="262"
      y="574" width="45" height="17" />
  </entry>
  <entry id="70" name="cpd:C00673" type="compound" link="http://www.genome.jp/dbget-
    bin/www_bget?compound+C00673">
    <graphics name="C00673" fgcolor="#000000" bgcolor="#FFFFFF" type="circle" x="262" y="613"
      width="8" height="8" />
  </entry>
  <relation entry1="1" entry2="3" type="ECrel">
    <subtype name="compound" value="70" />
  </relation>
  <reaction name="rn:R02750" type="reversible">
    <substrate name="cpd:C01801" />
    <product name="cpd:C00673" />
  </reaction>
</pathway>
```

圖 10：KEGG 的 XML 文件 (節錄自 Pentose phosphate pathway[KEGG])

就 KGML 資料格式，我們取出重要屬性轉成關聯式綱要如圖 11 所示。就以下關聯式綱要做討論。

```
Pathway (name, title)
Enzyme (pathway, id, name, reaction)
Compound (pathway, id, name)
Reaction (pathway, reaction name, substrate, product, type,
  enzyme name)
Relation (pathway, entry1, entry2, compound name)
```

圖 11：KEGG 的 XML 文件的關聯式綱要 (Relational schema for KEGG XML document)

```
Reaction (pathway, reaction name, substrate, product,
  type, Enzyme name)
```

表 1：化學反應式資料表的樣本(A sample of “Reaction” table)

pathway	reaction name	substrate	product	type	enzyme name
path:map00030	m:R02750	cpd:C01801	cpd:C00673	reversible	ec:2.7.1.18

圖 10 的化學反應式元素(reaction element)儲存資料如表 1 所示。在化學反應式中，受質與產物很重要之外，使用者也許更想知道此化學式需要加入那個酵素當催化劑。要知道此化學式所加入的酵素就必須先找到酵素資料表(Enzyme table)中相同的化學反應式名稱才能知道酵素的名稱。對於查詢處理而言，為了減少節省額外的成本，所以我們在 XML parser 階段（參考圖 1 之系統架構圖）將此化學反應中的催化劑名稱（enzyme name）這個也加入讓這化學反應式資料表中。

化學反應式資料表其綱要如下。在圖 10 中化學反應式元素的儲存資料如表 2 所示。

Relation (pathway, entry1, entry2, compound name)

表 2：化學關係式資料表的樣本(A sample of “Relation” table)

pathway	entry1	entry2	compound name
path:map00030	ec:2.7.1.18	ec:4.1.2.4	cpd:C00673
path:map00030	ec:4.1.2.4	ec:2.7.1.18	cpd:C00673

在表 2 中，entry1、entry2 以及化合物(compound) 名稱儲存的資料與 XML 文件的值所有不同。我們在 KEGG 文件經過 XML parser 階段（參考圖 1 之系統架構圖）時，將化學關係式元素(reaction element)中 entry1、entry2 的屬性的值(attribute value) (i.e., id value) 替換成名稱的值(name value)，化合物(compound) id 替換成化合物(compound) 名稱。另外，表示反應關係式的圖形，entry1 與 entry2 之間的邊緣並無方向性。如由 ec:2.7.1.18 與 ec:4.1.2.4 或 ec:4.1.2.4 與 ec:2.7.1.18 所形成之圖示，都是泛指同一個化學關係式。所以在 XML parser 階段，我們將一化學關係式儲存兩次如表 2。當使用者下一 QBE 查詢，將圖形轉換成 query set 時，一個化學關係式即不需要考慮拆解成那一種配對(pair)，對查詢處理而言每個化學關係式只需要比對一次。

entry 的部份，我們依據型態(type)這個屬性的屬性值將它分成兩個資料表。由於型態(type)屬性值為酵素與化合物(compound)，所儲存的性不盡相同，所以分別儲存在酵素資料表(Enzyme table)與化合物資料表(compound table)。其分別的關聯式綱要如下：

Enzyme (pathway, id, name, reaction)

Compound (pathway, id, name)

4 範例式查詢(QUERY BY EXAMPLE)

我們的查詢方式，稱為範例式查詢。就是由查詢者，給予一個生化代謝途徑(metabolic pathway)樣本的方法來代表我們的查詢語言。這種範例式查詢，





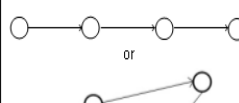
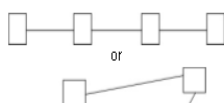


不但易表達容易學習、閱讀，提供使用者另一種查詢模式。

在本章節中，我們首先介紹我們所提供的查詢型態（章節 4.1），包含各圖像的意義及基本與複雜的查詢表示式。在章節 4.2 中，針對我們所提出的查詢型態，提出查詢處理演算法。基於我們的索引結構的索引結構表(inverted-list table)的索引方法，從關聯性資料表中，有效率的找到查詢的 pathway。

4.1 查詢型態

在我們的資料儲存結構下，我們使用圖形來表示使用者的查詢。我們分別針對「化學反應網路」與「酵素網路」歸納出 4 種範例式查詢型態（query type 1 to 4）以及 QBE 範例圖示，如表 3 所示。

表 3 查詢型態(Query type 1 to 4)

Query type	化學反應網路(reaction)	酵素網路(reaction)
QT1: 輸入單一 compound(enzyme)	 (a)	 (b)
QT2: 輸入一反應式(關係式)	 (c)	 (e)
QT3: complex query (輸入一連串反應式或關係式,且此查詢亦可包含cycle)		
QT4: complex query (steps unknown, 限定 K steps 內)	 (d)	 (f)

4.1.1 查詢圖像 (Query Icons)

在我們的查詢語言表示圖形中，主要有矩形、圓圈、箭頭以及破折號的箭頭，其各代表意義分述如下：

- 圓圈：表示代謝物（如表 3-(a)），其中包括受質以及產物。在生化代謝途徑(metabolic pathway)中就是表示化合物(compound)。
- 矩形：表示酵素（如表 3-(b)）。
- 箭頭：表示兩物件之間的關係（如表 3-(c)）。在化學反應網路中，是指兩物件（圓圈）可產生反應。箭頭的意義是代表起點連接受質物件，終點指向產物物件。
- 破折號的箭頭：表示兩物件之間的關係（如表 3-(d)）。由使用者輸入反應步驟的次數 K。此種查詢即在反應步驟次數 K 之內，找尋符合查詢條件的相關的反應式。
- 直線：表示兩物件之間的關係（如表 3-(e)）。在酵素網路中，第一個反應式生成的產物為第二個反應式之受質，則可將兩物件（矩形）連接。
- 破折號的直線（如表 3-(f)）：表示兩物件之間的關係。由使用者輸入反應步驟的次數 K。此種查詢即在反應步驟次數 K 之內，找尋是否存在符合查詢條件的相關的關係式。

4.1.2 基本查詢型態 (Basic Query Type)

- i. Query Type 1 (簡稱 QT1) : 輸入單一化合物 (compound) (或酵素)
QT1 即一般關鍵字查詢方式, 輸入欲查詢之化合物 (或酵素) 存在哪些 pathway 當中。
- ii. Query Type 2 (簡稱 QT2) : 輸入一反應式 (或關係式)
QT2 即查詢單一化學反應式 (或關係式)。可得知一反應式中化合物 (compound) 及化合物 (compound) 的關係, 或關係式中酵素與酵素之間的關係。

4.1.3 複雜查詢型態 (Complex Query Type)

- iii. Query Type 3 (簡稱 QT3) : 複雜查詢型態
QT3 為一連串複雜的反應式 (或關係式) 表示式的查詢, 其中此查詢可包含 cycle。此查詢即查詢文件中是否存在我們欲查詢之 pathway。
- iv. Query Type 4 (簡稱 QT4) : 複雜查詢型態 (steps unknown, 由 user define K steps 內) QT4 為查詢某反應式 (或關係式) 之代謝物 (或酵素) 之間的關係。K 值由使用者自訂反應步驟次數。此查詢即查詢在小於 K 步驟內, 所有符合查詢條件的可能途徑。

定義 3 : 化學關係式步驟 (step for relation)

給予一個生物代謝途徑 $P(M, Z, R)$, $G(V, E)$ 為 P 的關聯有向圖 (associated directed graph), 在關聯有向圖 (associated directed graph) G 裡, 存在著從 v_i 到 v_j 的路徑, 而我們把路徑的長度 (length) 稱為步驟的個數 (number of steps)。

請參考以下的英文定義 :

DEFINITION 3 : step for relation

Given metabolic pathway $P(M, Z, R)$, the associated directed graph $G(V, E)$ of P . In the associated graph G , exists the path from v_i to v_j . The length of the path is called the number of steps. [Koyu04].

定義 4 : 化學反應式步驟 (step for reaction)

在生物代謝途徑 $P(M, Z, R)$ 裡, 每存在一個反應 (reaction) 就稱為一個反應步驟, 所以反應的個數, 就是步驟的個數。

請參考以下的英文定義 :

DEFINITION 4 : step for reaction

In the metabolic pathway $P(M, Z, R)$, one reaction is called a (reaction) step. [Koyu04].

定義 3 中, 就 P 的關聯有向圖 $G(V, E)$ 而言, 簡單來說一個邊緣即是一個步驟。例如表 3-(e) 之圖示

即代表一個步驟。在定義 4 中, 一個化學反應式即是一個步驟。例如表 3-(c) 之圖示即代表一個步驟。

4.2 索引結構

針對我們所建構出來用來儲存 KEGG XML 文件的關聯式資料表, 我們將一般文件常利用的索引結構表加以延伸, 所形成的索引表, 如圖 12 及圖 13 的索引結構所示, 我們將之稱為索引結構表 (inverted-list table)。我們將基於索引結構表 (inverted-list table) 的索引表, 來加速找到相關 pathway。

4.2.1 索引結構表 (Inverted-list table)

索引結構表用來協助我們快速從生化途徑中快速找到反應式、關係式。我們的索引結構針對化學反應網路及酵素網路分別提供索引表。

- (A) 化學反應網路的索引結構表 (Inverted-list table for chemical network) (如圖 12) : 從化合物 (compound) 資料表中收集所有不同的化合物 (compound) 名稱, 儲存在我們實作之化合物 (compound) 列表的 B⁺-tree 索引表上。每一份生化代謝途徑 (metabolic pathway) 的 XML 文件都有一個獨特的名稱 (i.e., pathway name)。pathway list 上儲存所有不同的 pathway name。而 pathway name 則是用來表示化合物 (compound) 列表中的化合物 (compound) 出現在那個 XML 文件中。產物紀錄 (Product Record) 是記錄此化合物 (compound) 為受質時, 可反應生成出的產物, 以及一連接到資料庫化學反應式資料表 (Reaction table) 的指標。
- (B) 酵素網路的索引結構表 (Inverted-list table for enzyme network) (如圖 13) : 從酵素資料表 (Enzyme table) 中收集所有不同的催化劑名稱 (enzyme name), 儲存在我們實作之 Enzyme list 的 B⁺-tree 索引表上。每一份生化代謝途徑 (metabolic pathway) 的 XML 文件都有一個獨特的名稱。pathway list 上儲存所有不同的 pathway name。而 pathway name 則是用來表示酵素列表 (Enzyme list) 中的酵素, 出現在那個文件中。下一個 酵素紀錄 (Enzyme Record) 是記錄此酵素與哪些酵素有關係, 以及一連接到資料庫關聯性資料表 (Relation table) 的指標。

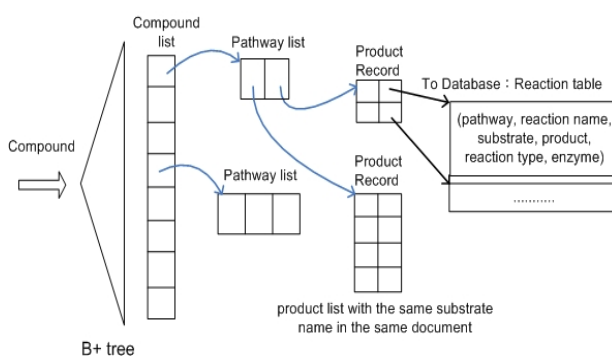


圖 12 : 化學反應網路的索引結構表 (Inverted-list table for chemical network)

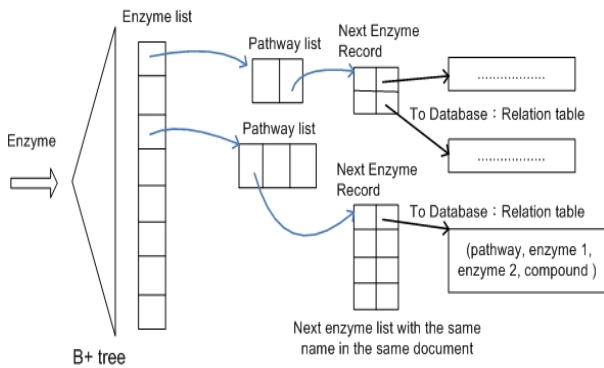


圖 13：酵素網路的索引結構表(Inverted-list table for enzyme network)

以下先舉個簡單的例子分別來說明這兩個網路的索引表如何建立。

範例 4：

pathway 名稱為 Pathway_s1 的化學反應網路如圖 14(a)所示，其文件如附錄(B)；pathway 名稱為 Pathway_s2 的化學反應網路如圖 14(b)所示，其文件如附錄(C)。以下化學反應式表示格式以“rn:反應式名稱 substrate $\xrightarrow{\text{enzyme}}$ product”表示。圖 14(a)中化學反應式包括：「rn:01 $A \xrightarrow{e1} B$ 」、「rn:02 $B \xrightarrow{e2} C$ 」、「rn:03 $C \xrightarrow{e3} D$ 」、「rn:04 $D \xrightarrow{e4} B$ 」、「rn:05 $B \xrightarrow{e5} E$ 」。圖 14(b)中化學反應式包括：「rn:06 $A \xrightarrow{e8} E$ 」、「rn:07 $E \xrightarrow{e7} G$ 」、「rn:05 $B \xrightarrow{e5} E$ 」、「rn:02 $B \xrightarrow{e2} C$ 」、「rn:03 $C \xrightarrow{e3} D$ 」、「rn:04 $D \xrightarrow{e4} B$ 」。

我們將附錄(B)、附錄(C)之 XML 文件在 XML parser 階段(參考圖 1 之系統架構圖)時，將文件轉成化合物(compound) 資料表與關聯性資料表如表 4 與表 5 所示。

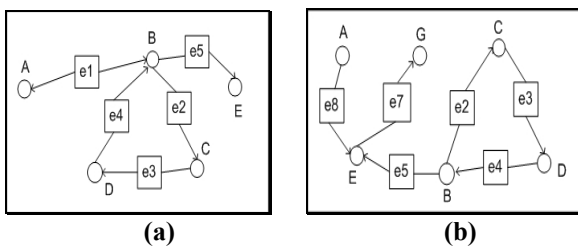


圖 14：(a) Pathway_s1 之化學反應網路；(b) Pathway_s2 之化學反應網路

表 4：化合物資料表(Compound table)

pathway	CPD id	CPD name
Pathway_s1	1	A
Pathway_s1	2	B
Pathway_s1	3	C
Pathway_s1	4	D
Pathway_s1	5	E
Pathway_s2	1	A
Pathway_s2	2	B
Pathway_s2	3	C
Pathway_s2	4	D
Pathway_s2	5	E
Pathway_s2	6	G

表 5：化學反應式資料表(Reaction table)

pathway	reaction name	substrate	product	reaction type	enzyme
Pathway_s1	m:01	A	B	reversible	e1
Pathway_s1	m:02	B	C	irreversible	e2
Pathway_s1	m:03	C	D	irreversible	e3
Pathway_s1	m:04	D	B	irreversible	e4
Pathway_s1	m:05	B	E	irreversible	e5
Pathway_s2	m:06	A	E	irreversible	e8
Pathway_s2	m:07	E	G	irreversible	e7
Pathway_s2	m:03	C	D	irreversible	e3
Pathway_s2	m:04	D	B	irreversible	e4
Pathway_s2	m:05	B	E	irreversible	e5
Pathway_s2	m:02	B	C	irreversible	e2

圖 15 的 part(A) 化合物資料表，由表 4 之 CPD name 欄位取出所有不同名稱的化合物所建出，而 pathway list 即代表此化合物出現在哪些 pathway 中。例如表 4 中，有 6 個不同的化合物資料表名稱，所以 part(A)部份即儲存 6 個值，這 6 個化合物分別出現在那些 pathway 中。

圖 15 的 part(B) 產物紀錄(Product Record)的部份是透過表 5 的 product 欄位所建出。例如在同一份文件 Pathway_s1 下，化合物 B 出現在表 5 之關連資料表的 substrate 欄位兩次，所以記錄了受質(substrate)為 B，產物紀錄(Product Record)中的 product 分別為 C、E 兩筆資料。另外，產物紀錄(Product Record) C、E 各有一指標指向資料庫之關聯性資料表。我們再以化合物 G 的例子來看，在表 4 中存在化合物 G，但表 5 中並不存在受質為 G 的反應式，所以 part(B) 的產物紀錄(Product Record)值為 NULL。

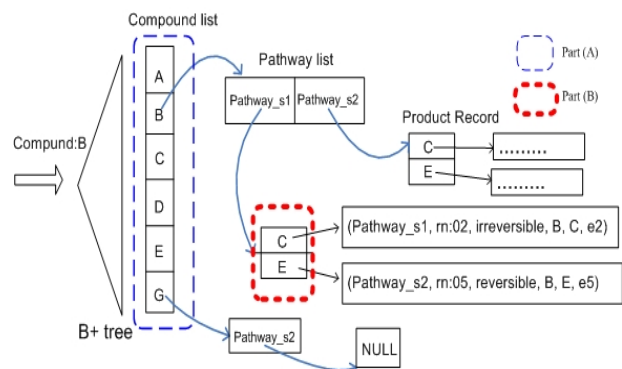


圖 15：化學反應網路的索引結構表(Inverted-list table for chemical network)

範例 5：

建構酵素網路的索引結構表的方式與化學反應網路的方式相同。我們將附錄(B)、附錄(C)之 XML 文

件，經 XML parser module (參考圖 1 之系統架構圖) 時，轉成表 6 之酵素資料表(Enzyme table)與表 7 之關聯性資料表。

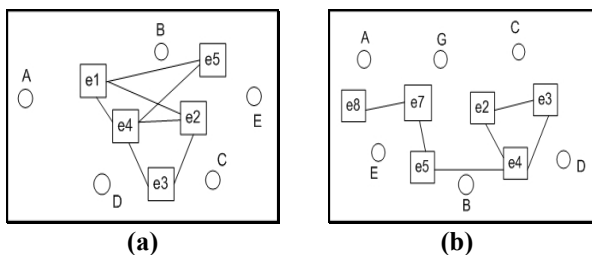


圖 16 : (a) Pathway_s1 之酵素網路 ; (b) Pathway_s2 之酵素網路

圖 17 的 part(A)的部份依據酵素資料表(Enzyme table)中欄位所有不同催化劑名稱(enzyme name)所建出。圖 17 的 part(B)的部份，在相同文件名稱且相同的 entry1 下，將不同的 entry2 的值(Value)儲存在下一個酵素記錄(Enzyme Record)欄位，以及儲存一個指標指向資料庫之關聯性資料表。

表 6 : 酵素資料表(Enzyme table)

pathway	enzyme id	enzyme name	reaction name
Pathway_s1	6	e1	rn:01
Pathway_s1	7	e2	rn:02
Pathway_s1	8	e3	rn:03
Pathway_s1	9	e4	rn:04
Pathway_s1	10	e5	rn:05
Pathway_s2	7	e2	rn:02
Pathway_s2	8	e3	rn:03
Pathway_s2	9	e4	rn:04
Pathway_s2	10	e5	rn:05
Pathway_s2	11	e7	rn:07
Pathway_s2	12	e8	rn:06

Table 7 : 關聯性資料表(Relation table)

pathway	entry1	entry2	compound
Pathway_s1	e1	e5	B
Pathway_s1	e1	e2	B
Pathway_s1	e1	e4	B
Pathway_s1	e2	e3	C
Pathway_s1	e2	e1	B
Pathway_s1	e2	e4	B
Pathway_s1	e3	e4	D
Pathway_s1	e3	e2	C
Pathway_s1	e4	e2	B
Pathway_s1	e4	e5	B
Pathway_s1	e4	e1	B
Pathway_s1	e4	e3	D
Pathway_s1	e5	e1	B
Pathway_s1	e5	e4	B
Pathway_s2	e2	e3	C
Pathway_s2	e2	e4	B
Pathway_s2	e3	e2	C
Pathway_s2	e3	e4	D
Pathway_s2	e4	e2	B
Pathway_s2	e4	e3	D
Pathway_s2	e4	e5	B
Pathway_s2	e5	e4	B
Pathway_s2	e5	e7	E
Pathway_s2	e7	e5	E
Pathway_s2	e7	e8	E
Pathway_s2	e8	e7	E

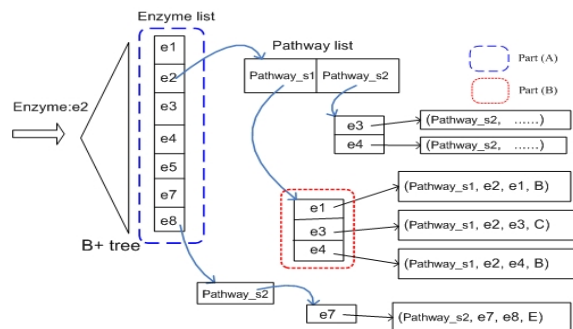


圖 17 : 酵素網路的索引結構表(Inverted-list table for enzyme)

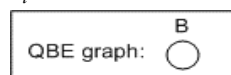
4.3 查詢處理演算法

根據表 3 的查詢型態，我們分別針對「化學反應網路」與「酵素網路」QBE 的查詢提出查詢處理演算法。我們以附錄(B)與附錄(C)之 XML 文件 (pathway 圖示如圖 14 所示) 的資料為例，根據 4 種查詢型態之查詢處理演算法，來說明查詢處理過程與輸出結果。以下 Query Processing for Query Type X 就以 QPTX 簡稱。先針對查詢化學反應網路作說明：

(A) QPT1：輸入單一化合物，即一般關鍵字(keyword-based)的查詢方式。簡單的說，就是查詢某化合物存在那些 pathway 中。

以範例 6 找化合物 B 為例。化合物 B 在 pathway 名稱為 Pathway_s1 與 Pathway_s2 有出現，即把相關資料圖示顯示給使用者。QT1 的 substrate-and-product join algorithm 如 Algorithm spQPT1 所示。

範例 6 : QT1 : $E_i = B$, QBE 的圖示如下



依照使用者 QT1 所下之查詢樣本，從化合物 list 中找到 B 元素。將 B 元素之 pathway list 回傳。QT1 的查詢結果以及圖 18 即是物件(object) B 在圖形所在位置，如下所示：

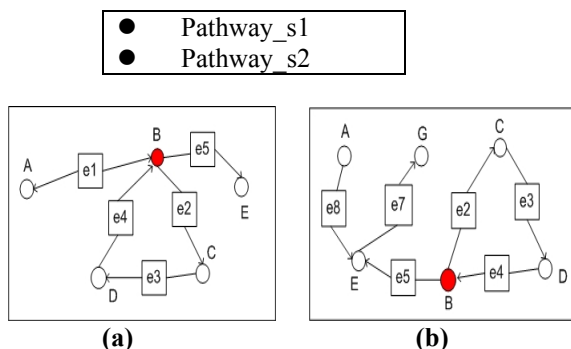


圖 18 : 物件(Object) B 在 (a) pathway_s1 ; (b) pathway_s2.

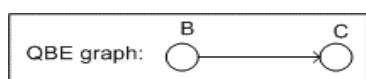
```

Algorithm spQPT1 : s_p_join_QPT1( $E_i$ )
// substrate-and-product join algorithm of QT1
/* input:  $E_i$ , where  $E_i$  is a compound. */
/* output: A list of pair of (pathway list). */
Begin
1. if ( $E_i$ ==element of compound list) then
2.     output (pathway list)
3. end if
End

```

(B) QPT2: 輸入一個反應式。簡單的說，就是查詢此化學反應式出現在那些文件中。首先，將 QT2 之 QBE 查詢圖示帶入 Algorithm GD 中。在將 Algorithm GD 的輸出輸入至 Algorithm spQPT2 中。

範例 7: QT2: $Q = \{(B, C)\}$, QBE 的圖示如下



根據使用者給予之 QT2 查詢樣本要尋找受質是 B，產物是 C 的化學反應式。我們將 QBE 樣本依據箭頭方向拆解成 (B, C) 這個 pair (經 Algorithm GD 得到)。得到結果 $Q = \{(B, C)\}$ 帶入 Algorithm spQPT2，依據化學反應網路之索引結構表 (inverted-list table)，在化合物列表 (compound list) 中先找到元素 B，再依據途徑列表 (pathway list) 中各途徑名稱 (pathway name) 做區分查閱產物紀錄 (Product Record) 中是否有元素 C。QT2 的查詢結果如下所示。

輸出結果 (Pathway_s2, B, C, rn:02, irreversible, e2)，各欄位代表的意思為 (pathway 名稱, 受質, 產物, 化學反應名稱, 可逆, 酵素名稱)。圖 19 顯示搜尋到的物件在圖形中的所在位置。

- (Pathway_s1, B, C, rn:02, irreversible, e2)
- (Pathway_s2, B, C, rn:02, irreversible, e2)

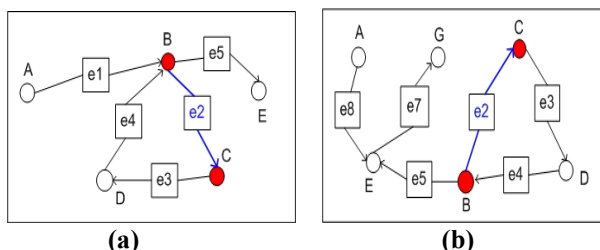


圖 19: QT2 在 (a) pathway_s1; (b) pathway_s2 的搜尋結果

```

Algorithm spQPT2 : s_p_join_QPT2(G)
/* input: G, QBE graph */
/* output: A set of (P_ID,  $E_1$ ,  $E_2$ , reaction name, type, enzyme) */
Begin
1.  $Q = \text{graph\_decomposition}(G)$ 
2. s_p_join_QPT2(Q)
End

```

```

s_p_join_QPT2(Q)
/* input: Q, Q is query set
QT2 :  $Q = \{(E_1, E_2)\}$ , is a pair of compounds. */
/* output: A set of (P_ID,  $E_1$ ,  $E_2$ , reaction name, type, enzyme) */
Begin
1. if ( $E_i$ ==element of compound list)
2.     if (Product Record of  $E_i=E_{i+1}$  with the same P_ID) then
3.         output (P_ID, S, P, reaction name, type, enzyme)
4.     end if
5. end if
End

```

```

Algorithm GD : graph_decomposition(G)
/* input: G, QBE graph
G1 :  $G = \{N, L\}$ , query samples, including  $N, L$ . Where
N is node set, L is edge set
G2 :  $G = \{N, L, K\}$ ,  $N$  is node set and  $L$  is edge set and
the label of  $L$  has a value  $K$ .
*/
/* output: The result of G1 or G2
the result of G1 :  $Q = \text{a set pair of (substrate, product)}$ 
the result of G2 :  $Q = \text{a set pair of (substrate, product, k)}$ 
*/
Begin
1. for each edge  $l \in L$ 
2.     if (edge  $l$  connect with node  $v$  to  $u$ )
3.         if (edge has a value) then
4.             output ( $v, u, k$ )
5.         else
6.             output ( $v, u$ )
7.         end if
8.     end if
9. loop
End

```

(C) QPT3: 複雜查詢型態 (complex query)。可輸入一連串反應式或具有週期性 (cycle) 之 pathway。

此種查詢，首先必須先針對使用者給予的樣本經由演算法 Algorithm GD 做分解，存儲成 (substrate, product) 的 pair 的集合。Algorithm GD 所得到的結果即代表 query set Q ，再帶入 QT3 的 substrate-and-product join algorithm 如 Algorithm spQPT3，所得到的結果即是查詢結果。

範例 8: QT3: $Q = \{(A, B), (B, C), (C, D)\}$, QBE 查詢圖示如圖 20。

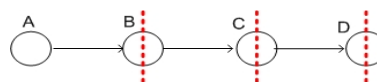


圖 20: QT3 的圖形分解表示法 (Decomposition of graph expression QT3)

首先針對查詢的圖形分解成我們一個個的 pair 如同範例 7 的方式。Algorithm GD 是依據邊緣 (edge) 來分割，在箭頭的尾端，將查詢分割成 $A \rightarrow B$ 、 $B \rightarrow C$ 以及 $C \rightarrow D$ ，即為 (A, B)、(B, C)、(C, D) 三個 pairs 所形成的集合，範例 8 之 Q 即是此 QBE 之

query set。再將得到的 query set Q，輸入 QT3 的 substrate-and-product algorithm 如 Algorithm spQPT3。判斷化合物 A 是否存在化合物列表中，在產物紀錄(Product Record)的欄位中的資料是否有 B，依序去檢查化合物 B 與 C 時的產物紀錄(Product Record)是否有 C 與 D，查詢與輸出結果如範例 8 所示。

QT3 查詢結果如下所示。各欄位代表的意思為 (pathway 名稱, 受質, 產物, 化學反應名稱, 可逆, 酵素名稱)。在 Pathway_s1 中找到符合{(A, B)、(B, C)、(C, D)}此查詢的結果，在 Pathway_s2 中只存在符合{(B, C)、(C, D)}的查詢。圖 21 即搜尋到的物件圖示。

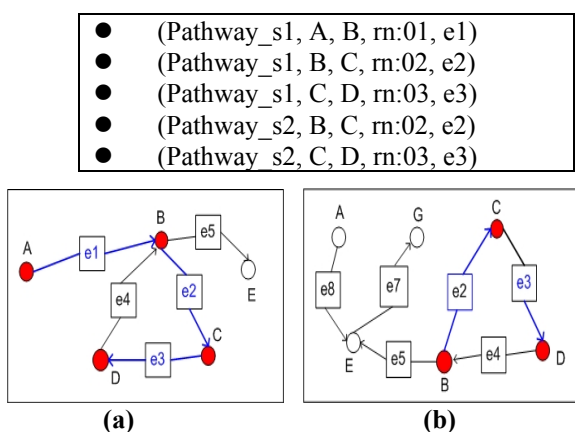


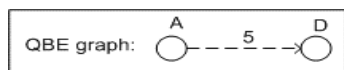
圖 21：QT3 在(a) pathway_s1；(b) pathway_s2 的搜尋結果

Algorithm spQPT3 : s_p_join_QPT3(G)
 /* input: G, QBE graph */
 /* output: A set of (P_ID, E_{n-1}, E_n, reaction name, type, enzyme) */
 Begin
 1. Q = graph_decomposition(G)
 2. for each pair (E_{n-1}, E_n) in Q do
 3. s_p_join_QPT2(Q)
 4. end for
 End

(D) QPT4：複雜查詢型態 (steps unknown)。此查詢的參數 K，是由使用者輸入。此類查詢即尋找是否存在某特定受質在 K 個步驟內，生成某產物的 pathway。假如在 K 步內找到相關反應式，則無任何結果回傳。

QT4 的 substrate-and-product join 演算法如 Algorithm spOPT4。QPT4 的查詢處理(query processing)，首先將 Algorithm GD 的輸出結果輸入 Algorithm spQPT4。此查詢型態在每一個步驟都必須有緩衝器(buffer)暫存相關的反應途徑。最後 Algorithm spOPT4 的輸出結果，即為查詢結果。

範例 9：QT4：Q = {(A, D, 5)}之查詢圖示如下



假設使用者 QBE 圖示查詢如下所示，經 Algorithm GD 得到 query set Q = {(A, D, 5)}，所代表我們的查詢。

將 Q 帶入 Algorithm spOPT4。首先，先將 A 儲存在緩衝器(buffer) 1，在判斷化合物 A 的產物紀錄(Product Record)的欄位中是否有 D。例如 Pathway_s1 中，A 的產物紀錄(Product Record)的欄位中只有 B，反應步驟未大於使用者定義的值 K，所以先將 B 暫存在緩衝器(buffer) 2 中。依次去檢查緩衝器(buffer) 2 中化合物 B，找到產物紀錄(Product Record)的欄位 C、E。緩衝器(buffer) 3 中，C 的產物紀錄(Product Record)存在我們與查詢的 D 產物(product)，所以緩衝器(buffer) 1 到緩衝器(buffer) 4 的 A→B→C→D，符合我們的查詢(A, D, 5)，也就是輸出結果之一，其他過程依此類推。緩衝器中暫存那些元素的過程如圖 22 所示。其中 0 表示此反應式無法繼續反應，為最終產物。

在 3 個步驟時，即找受質為 A，最終產物為 D 之 pathway。最後的 QT4 之輸出結果如下所示。

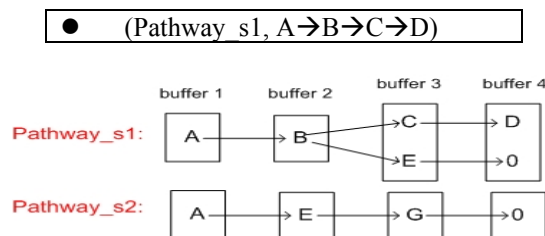


圖 22：QPT4 for Q = (A, D, 5)

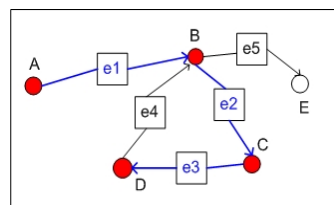


圖 23：符合 QT4 查詢的可能代謝途徑

Algorithm spQPT4 : s_p_join_QPT4(G) //substrate-product join
 /* input: G, QBE graph */
 /* output: A set of (P_ID, S, P) */
 Begin
 1. Q = graph_decomposition(G)
 2. for each pair (E_i, E_{i+1}) in Q do
 3. if (E_i and E_{i+1} in Compound list) //check whether E_i, E_{i+1} in Compound list
 4. l = 1 //initial value of buffer number
 5. put E_i into buffer l //store compound E_i to buffer l
 6. for j = 1 to K do //K is given by user
 7. for each element in buffer l with the same P_ID do
 8. if (buffer l elements in Product Record == E_{i+1}) then
 9. print the pathway from buffer 1 to l
 10. else
 11. put Product Record of buffer l into buffer l+1
 12. l++
 13. end if
 14. end for
 15. end for
 16. end if
 17. end for
 End

以下針對酵素網路的查詢處理演算法作說明，由於酵素網路的作法與化學反應網路的查詢處理演算法雷同。所以我們直接以 QBE 的查詢圖示為例，轉換成 Query set 的結果當作查詢，以查詢例子與輸出結果做說明。以下例子基於附錄(B)與附錄(C)之 XML 文件（酵素網路圖示如圖 16 所示）的資料為例來說明：

(A) QPT1：輸入單一酵素，即一般關鍵字(keyword-based)的查詢方式。簡單的說，就是查詢某酵素存在那些 pathway 中。QT1 的 enzyme-enzyme join 演算法如 Algorithm eeQPT1 所示。

範例 10：QT1： $E_i = e_2$

QBE 圖示與查詢處理過程與範例 6 相同，所以我們直接以 QBE 圖示與輸出結果來說明。

QBE graph: e2

QT1 查詢之結果如下：

- Pathway_s1
- Pathway_s2

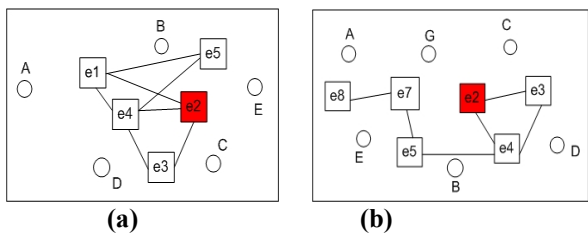


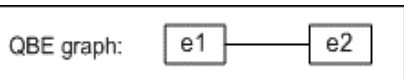
圖 24：物件 e_2 在 (a) pathway_s1；(b) pathway_s2, 的 pathway.

```

Algorithm eeQPT1 : en_en_join_QPT1( $E_i$ ) //enzyme
-enzyme join
/* input:  $E_i$ , where  $E_i$  is a enzyme. */
/* output: A list of pair of pathways. */
Begin
1. if ( $E_i$  ==element of enzyme list) then
2.   output (pathway list)
3. end if
End
    
```

(B) QPT2：輸入一個關係式。QBE 圖示經 Algorithm GD 演算法得到一 query set Q，將 Q 帶入 Algorithm eeQPT2 得到的結果即代表查詢結果。

範例 11：QT2： $Q = \{(e1, e2)\}$



QT2 查詢之輸出結果：

- (Pathway_s1, $e1, e2, B$)

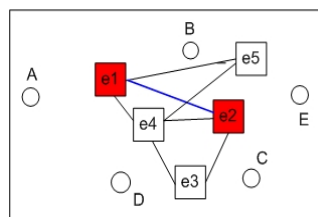


圖 25：QT2 在 pathway_s1 裡的搜尋結果

```

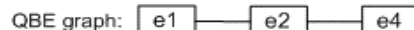
Algorithm eeQPT2 : en_en_join_QPT2(G)
/* input: G, QBE graph */
/* output: A set of (P_ID,  $E_i, E_{i+1}$ , compound) */
Begin
1. Q = graph_decomposition(G)
2. en_en_join_QPT2(Q)
End
    
```

```

en_en_join_QPT2(Q)
/* input: Q, QBE graph */
/* output: A set of (P_ID,  $E_i, E_{i+1}$ , compound) */
Begin
1. if ( $E_i$  ==element of enzyme list)
2.   if (Next Enzyme Record of  $E_i = E_{i+1}$  with the
      same P_ID) then
3.     output (P_ID,  $E_i, E_{i+1}$ , compound)
4.   end if
5. end if
End
    
```

(C) QPT3：複雜查詢型態。演算法如 Algorithm eeQPT3 所示。

範例 12：QT3： $Q = \{(e1, e2), (e2, e4)\}$



QT3 查詢之輸出結果：

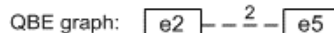
- (Pathway_s1, $e1, e2, B$)
- (Pathway_s1, $e2, e4, B$)
- (Pathway_s2, $e2, e4, B$)

```

Algorithm eeQPT3 : en_en_join_QPT3(G)
/* input: Q, where Q is query set */
/* output: A set of (P_ID,  $E_i, E_{i+1}$ , compound) */
Begin
1. Q = graph_decomposition(G)
2. for each pair ( $E_i, E_{i+1}$ ) in Q
3.   en_en_join_QPT2(Q)
4. end for
End
    
```

(D) QPT4：複雜查詢型態 (steps unknown，由使用者輸入 K 個步驟內)。QT4 之 enzyme-enzyme join 演算法如 Algorithm eeQPT4 所示。

範例 13：QT4： $Q = \{(e2, e5, 2)\}$



在 2 個步驟內符合 QT4 之查詢($e_2, e_5, 2$)，查詢處理過程如圖 26 所示。

QT4 查詢之輸出結果：

- (Pathway_s1, $e_2 \rightarrow e_1 \rightarrow e_5$)
- (Pathway_s1, $e_2 \rightarrow e_4 \rightarrow e_5$)
- (Pathway_s2, $e_2 \rightarrow e_4 \rightarrow e_5$)

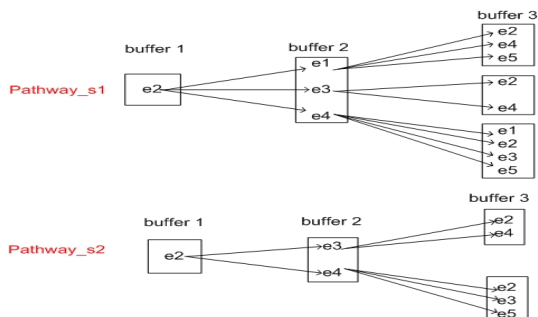


圖 26：QPT 在 QT4： $Q = (e_2, e_5, 2)$ 的查詢處理過程

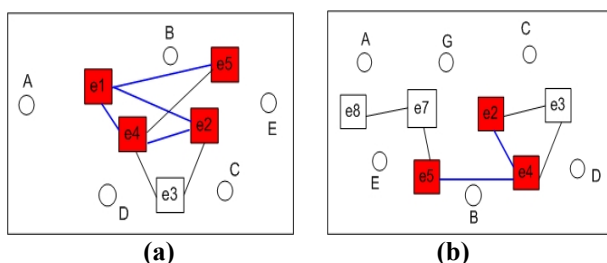


圖 27：(a)pathway_s1(b)pathway_s2，符合 QT4 查詢的可能代謝途徑

```

Algorithm eeQPT4 : en_en_join_QPT4(G) //enzyme-enzyme join
/* input: G, QBE graph */
/* output: the path which conformed to query condition, from  $E_1$  to  $E_{i+1}$ . */
Begin
1.  $Q = \text{graph\_decomposition}(G)$ 
2. for each pair( $E_i, E_{i+1}$ ) in  $Q$  do
3.   if ( $E_i$  and  $E_{i+1}$  in Enzyme list) //first, check  $E_i$  and  $E_{i+1}$  are in enzyme list.
4.      $l = 1$  //initial value of buffer number
5.     put  $E_i$  into buffer  $l$  //store enzyme  $E_i$  to buffer  $l$ 
6.     for  $j = 1$  to  $K_i$  do //The parameter "K" is given by user
7.       for each element in buffer  $l$  with the same P_ID do
8.         if (buffer  $l$  elements in Next Enzyme Record ==  $E_{i+1}$ ) then
9.           print the pathway from buffer 1 to  $l$ 
10.        else
11.          put Next Enzyme Record of buffer  $l$  into buffer  $l+1$ 
12.           $l++$ 
13.        end if
14.      end for
15.    end for
16.  end if
17. end for
End

```

5 系統實作

本章節描述我們實作的部份。所用電腦配備是 CPU P4-2.8G, RAM 1G, 作業系統為 Windows XP sp1, 撰寫使用軟體是 Borland C++ Builder 6.0, 使用之資料庫系統為 SQL Server 2000。資料來源 KEGG XML 文件包括： Glycolysis / Gluconeogenesis、

Glyoxylate and dicarboxylate metabolism、Urea cycle and metabolism of amino groups 等生化代謝途徑 (metabolic pathway) 資料[KEGG]。

5.1 系統架構

範例式查詢系統整個系統架構就如圖 28 所示，主要分成資料模型(data modeling)、索引和查詢處理(indexing and query processing)這兩部份。

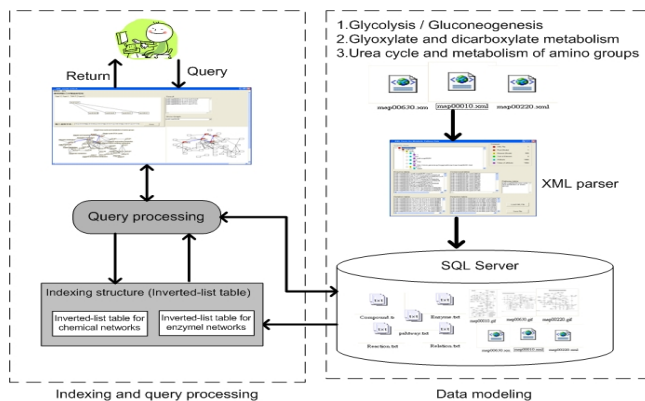


圖 28：系統架構圖(詳細版)

(A) 資料模型(Data modeling)：首先，將 KEGG 之生化代謝途徑(metabolic pathway) XML 文件經過 XML parser (如圖 34 所示)，儲存在我們設計了 5 個關聯式資料表中。這 5 個資料表分別為 Pathway table、化合物資料表、酵素資料表(Enzyme table)、化學反應式資料表(Reaction table)及化學關係式資料表(Relation table)。再將我們 parser 之後的資料，儲存至資料庫 SQL Server 2000 中。另外，我們將原始 KEGG XML 文件以及 KEGG 圖形也儲存在資料庫中。其中 parsing 出 XML tree 的部份 (XMLTree function) 是引用了“Delphi K.Top 論壇討論區 (<http://delphi.ktop.com.tw/default.asp>)”的程式碼。

(B) 索引和查詢處理(Indexing and query processing)：程式執行時，即把資料表載入我們所設計的索引結構中。我們利用化合物資料表與化學反應式資料表(Reaction table)建構出，針對查詢化學反應網路的索引結構的索引結構表(inverted-list table)；而酵素網路的索引表是由酵素資料表(Enzyme table)與學關係式資料表(Relation table)所建構出。我們分別針對「化學反應網路」與「酵素網路」歸納出 4 種查詢型態，分別針對兩種網路之 4 種查詢型態提出查詢處理演算法(Query processing algorithm)。當使用者給予之 QBE 查詢樣本，查詢處理模型(Query processing module)利用索引結構表，有效在的在資料庫中找到相關資料，最後將查詢結果以及搜尋到物件的圖示回傳給使用者。

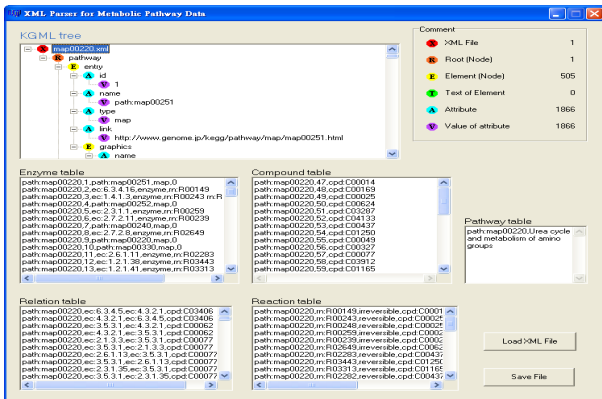


圖 29：生化代謝途徑資料的 MML parser (XML parser for metabolic pathway data)

5.2 查詢介面

本節我們描述 QBE query 的實作。使用者查詢生化代謝途徑方式可分兩種，針對化學反應網路，是利用輸入化合物的方式 (i.e. 查詢化學反應式)；另一種是針對酵素網路，為輸入酵素 (enzyme) (i.e. 查詢化學關係式)。

使用者首先選擇查詢的網路，且分別提供 4 種查詢型態。依照給予樣本 (sample) 來代表我們的查詢語言，如圖 30 之 alpha-D-Glucose \rightarrow alpha-D-Glucose 6-phosphate (葡萄糖分解反應)、圖 31 之 Urea Cycle (尿素循環) 查詢。圖形分解成 query set 的部份，直接由使用者輸入查詢樣型 (Query Pattern)。也就是說查詢方式由使用者輸入，產生代表 QBE 的圖形，其介面如圖 30、圖 31 左上方所示。查詢到的結果顯示在右上方，使用者可以點選 “Show Graph” 顯示查詢的物件圖形。下方會顯示兩張圖示，左邊的圖示為完整圖示，包括查詢到的途徑以及其他物件的名稱；右圖則只顯示查詢到的物件之名稱。

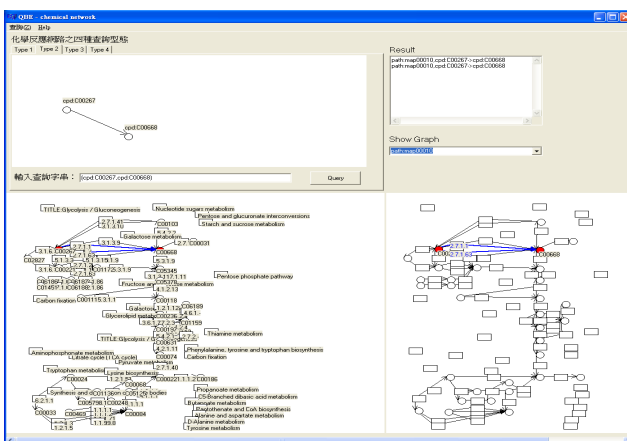


圖 30：化學反應網路之 QT2 查詢(葡萄糖分解反應)

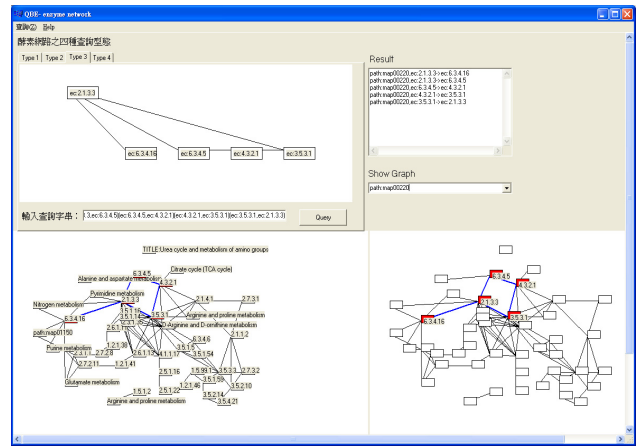


圖 31：酵素網路之 QT3 查詢(尿素循環)

另外，QT4 提供符合查詢條件的可能代謝途徑。如圖 32 的查詢，是查詢是否有符合 “cpd:C02867 在 1 個步驟內，生成中間產物 cpd:C00221，而在 3 個步驟內生成最終產物為 cpd:C05378” 條件的途徑。

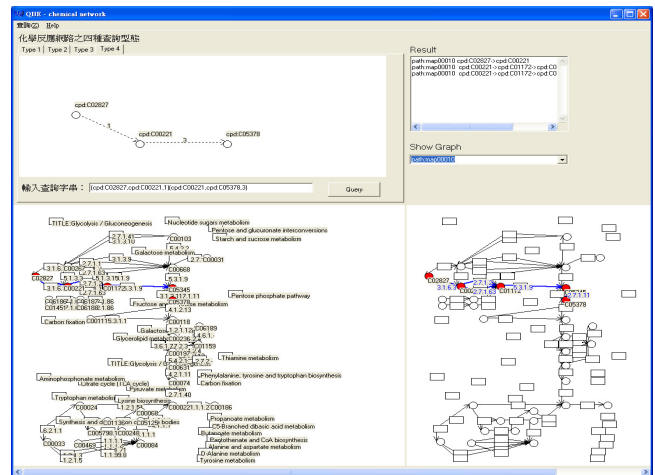


圖 32：化學反應網路之 QT4 查詢

6 結論與未來工作

目前的查詢生化途徑資料，大致以關鍵字為基礎的查詢方式為主。在本篇論文中，提出另一種查詢方式，稱為範例式查詢。範例式查詢對 naive user 而言是最精簡、容易的表達查詢方式。針對生化代謝途徑的 XML 文件，我們提出儲存這類資料的關聯式網要。我們針對「化學反應網路」與「酵素網路」，分別歸納出 4 種查詢型態、個別的查詢處理演算法，以及提供給使用者查詢生化代謝途徑的友善查詢介面 (user-friendly interface)。

目前查詢處理程序方面，我們是基於我們自定的索引結構表 (inverted-list table) 來做查詢處理。在未來的工作方面，我們將會增加實驗的資料，來產生出更多的實驗數據，以判斷系統的效率 and 好壞。另外，我們計劃將依據目前查詢方式研究並提出 unified query 及一些有效率的演算法來加速反應途徑，如利用 suffix tree 儲存所有可能途徑。另外，計畫先針對查詢進行相似性比對 (similarity

matching) , 提出 ranking algorithm。在來, 整合網路上現有的反應途徑資料庫, 預定針對兩個生化反應網路間的相似性比對做進一步的研究, 提供生物專家相關資訊。此外, 我們將依據相似性比對演算法, 研究如何將生化代謝途徑有效分類 (classification) 。

參考文獻

- [Alkh02] S. Al-Khalifa, H.V. Jagadish, N. Koudas, J. M. Patel, D. Srivastava, and Y. Wu, "Structural Joins: A Primitive for Efficient XML Query Pattern Matching," in *Proceedings of the 18th International Conference on Data Engineering*, 2002.
- [Beck90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-Tree: An efficient and robust access method for points and rectangles," *SIGMOD Record*, 1990, pages 322–331.
- [Boag] S. Boag, D. Chamberlin, M. F. Fernández, D. Florescu, J. Robie, and J. Siméon. XQuery 1.0: An XML Query Language for w3c draft. available at <http://www.w3.org/TR/xquery>.
- [Brag03] D. Braga and A. Campi, "A Graphical Environment to Query XML Data with XQuery," in *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03)*, 2003.
- [Brag04] D. Braga, A. Campi, and S. Ceri, "XQBE(XQuery by example): a visual interface to the standard XML query language," accepted by *ACM Transactions on Database systems*, (available at <http://www.elet.polimi.it/upload/braga/>).
- [Bray] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. "Extensible Markup language (XML) 1.0 (Second Edition)," available at <http://www.w3.org/XML/>.
- [Brun02] N. Bruno, N. Koudas, and D. Srivastava, "Holistic twig joins: optimal XML pattern matching," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2002.
- [Chen03] J. Cheng, G. Yu, G. Wang, and J. X. Yu, "PathGuide: An Efficient Clustering Based Indexing Method for XML Path Expressions," in *Proceedings of the Eighth International Conference on Database Systems for Advanced Applications (DASFAA'03)*, 2003.
- [Chen04] M. Chen and R. Hofestädt, "Web-Based Information Retrieval System for the Prediction of Metabolic Pathways," *IEEE Transactions on Nanobioscience*, Vol. 3, No. 3, September 2004.
- [Chun02] C. W. Chung, J. K. Min, and K. Shim, "APEX: An Adaptive Path Index for XML Data," in *Proceedings of ACM SIGMOD*, 2002.
- [Coop01] B. Cooper, N. Sample, M. J. Franklin, G. R. Hjaltason, and M. Shadmon. "A Fast Index for Semistructured Data," in *Proceedings of 27th International Conference on Very Large Data Bases*, January 2001.
- [Fors99] V. C. Forst and K. Schulten, "Evolution of metabolisms: A new method for the comparison of metabolic pathways using genomics information," *J. Comput. Biol.*, vol. 6, pp. 343–360, 1999.
- [Gold97] R. Goldman and J. Widom. "DataGuides: Enable query formulation and optimization in semistructured databases," in *Proceedings of 23rd International Conference on Very Large Data Bases*, 1997, pages 436-445.
- [Holt] R. Holt, A. Schürr, S. E. Sim, and A. Winter, "Graph eXchange Language," available at <http://www.gupro.de/GXL/>.
- [Jian03] H. Jiang, H. Lu, W. Wang and B. C. Ooi, "XR-Tree: Indexing XML Data for Efficient Structural Joins," in *Proceedings of the 19th International Conference on Data Engineering (ICDE'03)*, 2003.
- [Jian03] H. Jiang, W. Wang, H. Lu, and J. X. Yu, "Holistic Twig Joins on Indexed XML Documents," in *Proceedings of the 29th VLDB Conference*, Berlin, Germany, 2003.
- [Kane00] M. Kanenhisa and S. Goto, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucl. Acids. Res.*, Vol. 28, pp. 27-30, 2000.
- [Kane02] M. Kanenhisa, et al, "The KEGG databases at GenomeNet," *Nucl. Acids. Res.*, Vol. 30, pp. 42-46, 2002.
- [KEGG] <http://www.genome.ad.jp/kegg/xml/>.
- [Koyu04] M. Koyuturk, A. Grama and W. Szpankowski, "An Efficient Algorithm for Detecting Frequent Subgraphs in Biological Networks," *Bioinformatics* Vol. 20 (Suppl. 1), 2004, pages i200–i207.
- [Kris03] L. Krishnamurthy, J. Nadeau, G. Ozsoyoglu, M. Ozsoyoglu, G. Schaeffer, M. Tasan, W. Xu, "Pathways Database System: An Integrated Set of Tools for Biological Pathways," in *Proceedings of ACM SAC*, 2003.
- [Li01] Q. Li and B. Moon, "Indexing and Querying XML Data for Regular Path Expressions," in *Proceedings of the 27th VLDB Conference*, 2001.
- [Liu04] S. Liu, Q. Zou and W. W. Chu, "Configurable Indexing and Ranking for XML Information Retrieval," in *Proceedings of ACM SIGIR*, 2004.
- [McHu99] J. McHugh and J. Widom, "Query Optimization for XML," in *Proceedings of the 25th VLDB Conference*, pages 315–326, 1999.
- [Milo99] T. Milo, and D. Suciu, "Index Structures for Path Expression," in *Proceedings of 7th International Conference on Database Theory*, 1999, pages 277-295.
- [Newm04] S. Newman and Z. M. Ozsoyoglu, "A Tree-Structured Query interface for Query Semi-Structure Data," in *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04)*, 2004.
- [CASE] <http://nashua.case.edu/pathways/index.htm>.
- [Sira02] M. Sirava, T. Schafer, M. Eiglsperger, M. Kaufmann, O. Kohlbacher, E. Bornberg-Bauer and H. P. Lenhof, "BioMiner—modeling, analyzing, and visualizing biochemical pathways and networks," *Bioinformatics*, Vol. 18 (Suppl. 2), 2002, pages s219–s230.
- [Wang03] H. Wang, S. Park, W. Fan, and P. S. Yu, "ViST: A Dynamic Index Method for Querying

XML Data by Tree Structures,” in *Proceedings of SIGMOD 2003*.

[Weig04] F. Weigel, H. Meuss, F. Bry, and K. U. Schulz. “Content-Aware DataGuides: Interleaving IR and DB Indexing Techniques for Efficient Retrieval of Textual XML Data.,” in *Proceedings of 26th Europ. Conf. on IR*, 2004.

[Wint] A. Winter, B. Kullbach, and V. Riediger, “An Overview of the GXL Graph Exchange Language,” available at <http://www.gupro.de/winter>.

[Zou04] Q. Zou, S. Liu and W. W. Chu, “Ctree: A Compact Tree for Indexing XML Data,” in *Proceedings of ACM WIDM*, 2004.

附錄

(A) : “Pentose phosphate pathway”之圖形

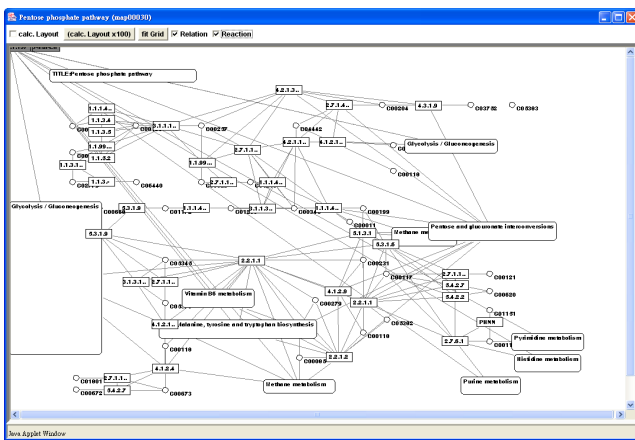


圖 33 : KEGG metabolic pathway - Pentose phosphate pathway (出自於 KEGG “Pentose phosphate pathway”之 graphics[KEGG])

(B) : pathway_s1.xml 之 XML document

```
<pathway name="pathway_s1" title="pathway example1">
  <entry id="1" name="A" type="compound" />
  <entry id="2" name="B" type="compound" />
  <entry id="3" name="C" type="compound" />
  <entry id="4" name="D" type="compound" />
  <entry id="5" name="E" type="compound" />
  <entry id="6" name="e1" type="enzyme" reaction="rn:01" />
  <entry id="7" name="e2" type="enzyme" reaction="rn:02" />
  <entry id="8" name="e3" type="enzyme" reaction="rn:03" />
  <entry id="9" name="e4" type="enzyme" reaction="rn:04" />
  <entry id="10" name="e5" type="enzyme" reaction="rn:05" />
  <relation entry1="6" entry2="9" type="ECrel">
    <subtype name="compound" value="2" />
  </relation>
  <relation entry1="7" entry2="9" type="ECrel">
    <subtype name="compound" value="2" />
  </relation>
  <relation entry1="8" entry2="9" type="ECrel">
    <subtype name="compound" value="4" />
  </relation>
  <relation entry1="9" entry2="10" type="ECrel">
    <subtype name="compound" value="2" />
  </relation>
  <reaction name="rn:01" type="reversible">
    <substrate name="A" />
    <product name="B" />
  </reaction>
  <reaction name="rn:02" type="irreversible">
    <substrate name="B" />
    <product name="C" />
  </reaction>
  <reaction name="rn:03" type="irreversible">
    <substrate name="C" />
    <product name="D" />
  </reaction>
  <reaction name="rn:04" type="irreversible">
    <substrate name="D" />
    <product name="E" />
  </reaction>
  <reaction name="rn:05" type="irreversible">
    <substrate name="B" />
    <product name="E" />
  </reaction>
</pathway>
```

圖 34 : pathway_s1.xml

(C) : pathway_s2.xml 之 XML document

```
<pathway name="pathway_s2" title="pathway example2">
  <entry id="1" name="A" type="compound" />
  <entry id="2" name="B" type="compound" />
  <entry id="3" name="C" type="compound" />
  <entry id="4" name="D" type="compound" />
  <entry id="5" name="E" type="compound" />
  <entry id="6" name="e1" type="enzyme" reaction="rn:01" />
  <entry id="7" name="e2" type="enzyme" reaction="rn:02" />
  <entry id="8" name="e3" type="enzyme" reaction="rn:03" />
  <entry id="9" name="e4" type="enzyme" reaction="rn:04" />
  <entry id="10" name="e5" type="enzyme" reaction="rn:05" />
  <entry id="11" name="e7" type="enzyme" reaction="rn:07" />
  <entry id="12" name="e8" type="enzyme" reaction="rn:08" />
  <relation entry1="7" entry2="8" type="ECrel">
    <subtype name="compound" value="3" />
  </relation>
  <relation entry1="7" entry2="9" type="ECrel">
    <subtype name="compound" value="2" />
  </relation>
  <relation entry1="8" entry2="9" type="ECrel">
    <subtype name="compound" value="4" />
  </relation>
  <relation entry1="9" entry2="10" type="ECrel">
    <subtype name="compound" value="2" />
  </relation>
  <relation entry1="10" entry2="11" type="ECrel">
    <subtype name="compound" value="5" />
  </relation>
  <relation entry1="11" entry2="12" type="ECrel">
    <subtype name="compound" value="5" />
  </relation>
  <reaction name="rn:06" type="irreversible">
    <substrate name="A" />
    <product name="E" />
  </reaction>
  <reaction name="rn:07" type="irreversible">
    <substrate name="E" />
    <product name="C" />
  </reaction>
  <reaction name="rn:08" type="irreversible">
    <substrate name="C" />
    <product name="D" />
  </reaction>
  <reaction name="rn:09" type="irreversible">
    <substrate name="D" />
    <product name="B" />
  </reaction>
  <reaction name="rn:05" type="irreversible">
    <substrate name="B" />
    <product name="E" />
  </reaction>
  <reaction name="rn:02" type="irreversible">
    <substrate name="B" />
    <product name="C" />
  </reaction>
</pathway>
```

圖 35 : pathway_s2.xml