運用 CUDA 平行處理技術實現蛋白質表面測地線特徵比對

Geodesic distance features for protein surface comparison by using CUDA parallel computing architecture

陳天盛、羅英倉、白敦文*
國立臺灣海洋大學資訊工程學系
*Email:twp@ntou.edu.tw

摘要—蛋白質結構的構型與功能息息相關，透過比對不同蛋白質間拓樸結構或表面形態的相似程度，已經成為結構生物學者及生物資訊學者進行驗證或預測蛋白質功能最直接的技術方法。本研究專注於蛋白質結構表面形狀特徵的分析，使用測地線距離作為比對技術的主要特徵，該特徵在結構表面發生形變後，仍能保持原有表面構型特徵不變的特性。然而為解決該特徵運算繁雜且執行費時的缺點，本研究亦導入 CUDA 平行處理技術，以提昇運算效能。系統驗證是使用具有顯著構型變化的 3D 交換域蛋白質資料集進行辨識與分群，實驗結果顯示本系統不僅可以精準地完成蛋白質表面結構比對及正確分類，應用 CUDA 的加速技術可使整體運算效能提昇 52.4 倍的優異表現。

關鍵詞—統一計算架構、測地線距離、結構表面比對

ABSTRACT—**The conformation of a protein structure is strongly related to its biological functions. For bioinformaticians and structure biologists, comparing topologies or surface conformations among structures is the most intuitive approach to verify or predict protein functions. Geodesic distance is one of the best deformation invariant features for protein surface comparison, which allows two deformed structures possessing conserved shape descriptors. However, the computational complexity for creating geodesic distance features is still a challenging and time-consuming work. Hence, this study applied CUDA parallel computing architecture to reduce required computational time. By combining geodesic distance characteristics and CUDA technologies, the proposed system is able to provide a robust and efficient system for protein surface comparison and classification. A testing dataset from 3D domain swapping proteins was employed in this study for system verification. According to the experimental results, the proposed shape features not only facilitated the identification of deformed proteins but also achieved a 52-fold improvement by utilizing the CUDA algorithms.**

Keyword: **CUDA; Geodesic Distance; Surface Comparison**

一、背景介紹

蛋白質巨分子的研究在近幾十年來被列為生命科學領域中最重要的研究課題之一，如今研究蛋白質的方法越來越多元化，例如蛋白質序列或結構的排比與分類、蛋白質表面資訊的

分析比對及蛋白質交互作用的研究等。至今為止，雖然各類型蛋白質比對方法皆有不錯的研究成果，但相對於蛋白質序列的技術研究，實際上使用蛋白質結構資訊的研究成果仍屬相對少數，而著重在蛋白質表面結構的研究分析更是少有，主要是因為蛋白質結構常因化學鍵結及分子間作用力的影響，使胺基酸分子於幾何空間的結合構型具動態性及變異性，增加研究的困難度及不確定性。

大多數生物體內的交互反應中，蛋白質的互動扮演著延續生命不可或缺的角色，因此蛋白質的研究在生物學上有著舉足輕重的地位。研究發現蛋白質的功能與其結構構型息息相關[13]，故針對結構資訊進行研究分析蛋白質的特性是最直接的辦法。由美國 Brookhaven 實驗室所建立的蛋白質三維結構資料庫(Protein Data Bank，PDB)[2]，是國際上最重要的蛋白質巨分子結構數據庫，目前已記錄的分子結構已經超過七萬五千筆的資料，且仍然逐年在快速成長當中。

PDB 所收錄之蛋白質分子結構資訊是以核磁共振(NMR)、X 光繞射及低溫電顯(cryo-EM)技術所解析，結構的數量又以前兩種技術佔了大部分。但前兩種方法所分析的作法相異，NMR 利用電子自旋特性，分析蛋白質電子雲圖，預測可能的結構構型；而 X 光繞射則是針對結晶的蛋白質進行 X 光繞射，分析所得到的圖譜並解析出結構的相關資訊，雖然大部分結構所解析的蛋白質形狀相去不遠，但在不同實驗環境下所解析的蛋白質，由於表面靜電勢、疏水性質或極性等表面化學性質隨環境或不同的結合物而變動，加上運用不同分析技術，相同的蛋白質卻可能會產生不同之三維結構定位。除此之外，當兩個蛋白質進行交互作用時，或是為了形成最穩定的複合體，某些蛋白質會改變其結構及表面構形，因此，不同狀態下所解析出的蛋白質結構，也可能具有差異性極大的立體結構。

本研究主軸是針對蛋白質立體結構之動態表面進行比對研究，開發一套可以容忍表面構型變化的分析系統，提供生物研究人員一個精準及有效率的生物資訊研究工具。一般而言，使用蛋白質立體結構資訊的比對分析技術常需要運用龐大的硬體資源與較長的運算時間才能得到分析結果，有鑑於此，本研究亦導入 NVIDIA 公司所開發的 CUDA(Compute Unified Device Architecture)平行運算架構，即圖形顯示晶片平行處理技術，將大量的重複計算過程進行平行化的處理，在不改變分析結果的原則下，減少數十倍的程式運算時間，達到高效率及高效能的預期成果。

- 文獻回顧與討論

自從 PDB 資料庫的建立後，結構生物學家便提出不同的技術對蛋白質結構進行分析研究[20]，例如針對蛋白質結構的排比技術如 DALI(Distance-matrix ALIgnment)[7]、CE(Combinatorial extension)[8, 16]、SSAP (Sequential Structure Alignment Program)[15, 19]等；或是針對蛋白質表面結構的特性進行分析如 SES(Solvent Excluded Surface)[17]、SAS(Solvent Accessible Surface)[9]等。更有一些應用是分析蛋白質的表面物理化學特徵進行預測蛋白質之間的交互作用[1]。早期的研究都是將蛋白質結構假設為「剛體」物件，以方便進行蛋白質結構及功能的比對分析，相關的蛋白質結構分析演算法及應用可參考回顧性文獻[6]。主要分為全域比對及局部比對等兩大類蛋白質結構排比演算法，近年來，結構生物學家開始注重蛋白質結構及表面為動態的特性，並開始使用具有彈性的表面結構特徵進行蛋白質結構之間的比較[5, 12]，由發表的結果證明加上具有動態表面特性的蛋白質結構，可以強化分析蛋

白質間互動結合的正確性或進一步提昇正確的結構比對分析。

由於蛋白質動態表面結構的概念興起，尋求在表面結構形狀改變後仍然可以保持其原有構形的特徵遂成為重要的研究項目，其中測地線距離特徵正具有形變不變性(deformation invariant)的特性，此特徵已逐漸為結構生物學家、電腦視覺及電腦圖學工程師所接受，最近有關測地線特徵應用在蛋白質結構上的研究成果正陸續發表中[10, 11, 21]，每種應用固然都以測地線性質為核心，但所使用之應用方法卻有所不同。然而，唯一美中不足的是測地線特徵需計算每一點到其他所有表面點的表面最短距離，所以運算過程繁雜且執行費時。因此，本研究提出一套以測地線特徵為主的蛋白質表面結構比對分析系統，並結合 CUDA 平行處理之雙重概念，以期獲得強健穩定的表面結構比對結果並達到高效能的執行效率。

二、研究技術介紹

(一) 測地線距離特徵(Geodesic Distance)

本論文所探討的對象是鎖定在發生構形改變的蛋白質結構，若採用現今一般研究中常用的蛋白質表面結構幾何特性，如立體角(solid angle)、表面凹凸曲率(curvature)、空間體積密度(volume density)及表面法向量(normal vector)之分佈等特徵，當蛋白質的表面構型改變時，上述各類特徵值，都將造成重大變化，因而大幅降低該特徵值的正確性及可靠性。所以為了維持表面特徵在形變後仍能保持不變的特性，本研究採用測地線距離特徵作為蛋白質動態表面比對的核心依據。測地線距離定義為空間中兩點沿著扭曲表面上空間最短行進距離的路徑(如圖 1)，即使表面發生變化，在拓樸結構不變的情況下，各點間相對的測地線距離依然相等。

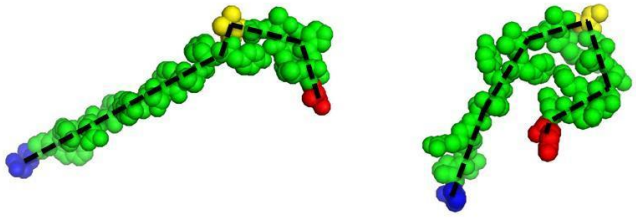

圖 1、測地線距離(虛線部分)不會因蛋白質結構的動態改變而影響其特性

(二) CUDA 平行運算技術

於西元 2007 年，全球最知名的 GPU 供應商 NVIDIA 公司開發一套針對大量平行運算的技術架構－統一計算架構(Compute Unified Device Architecture，CUDA)，它是一種利用圖形運算單元(GPU)來實現平行運算加速器(General-Purpose GPU)的一種程式開發技術，CUDA 程式亦可使用 FROTRAN、C、C++等延伸的程式語言，再透過 NVCC 來編譯 CUDA 的程式[14]。

相較於一般循序式的 CPU 執行，使用 CUDA 大量平行處理的技術能有效地使原有程式加速，其原因有以下主要兩點：(1) CPU 需要管理電腦運作大大小小的工作事項，而 GPU 較能專注在程式的執行上；(2) 一般的 CPU 程式通常較為複雜，並使用較慢的分枝(branch)與效率較低的循序執行，而 GPU 則是使用大量且單純的多核心來取代複雜的程式分枝，在大量平行運算處理技術方面更是遠勝於 CPU。

CUDA 的運作流程如圖 2 所示，分成四個主要的步驟：(1) 將運算所需的資料由主記憶體(Main Memory)複製到 GPU 的記憶體；(2) 利用 CPU 平行處理的程式碼指令來驅動 GPU，告知 GPU 記憶體所有的分配情況與各個核心所要執行的工作；(3) GPU 使用多核心來進行大量平行處理並算出結果；(4) 將 GPU 內部已算好的結

果複製回主記憶體，接著 CPU 繼續向下執行。

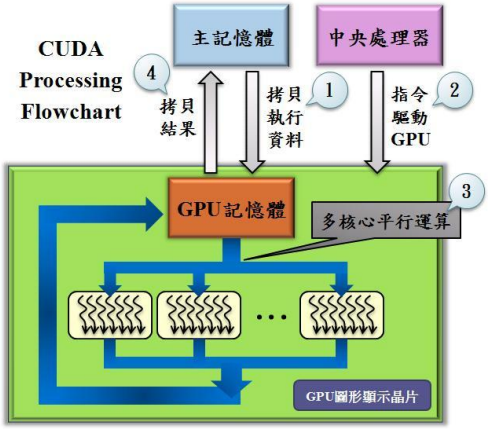

圖 2、CUDA 程序流程圖

至今，CUDA 運算幾乎成為軟體業界與學術界中非官方的標準，大量平行處理的技術，已被廣泛應用到各類研究領域，例如：資訊安全與密碼學、網際網路通訊協定、自然語言與資料檢索、生物資訊等，研究成果亦陸續發表在各個重要國際會議及期刊論文。

三、研究方法及步驟

本論文的研究方法是採用區域性的測地線距離為主要特徵值，以該特徵進行蛋白質結構表面的比對分析，並且應用 CUDA 圖形顯示晶片平行處理技術來增加比對的速率。系統流程圖如圖 3 所示，主要步驟如下：

1. 輸入 PDB 檔(PDB file importing)
2. 擷取骨幹(Backbone atom retrieval)
3. 擷取蛋白質結構表面點(Protein surface point retrieval)
4. 表面點縮減取樣(Surface point downsampling)
5. 計算測地線距離(GD value calculation)
6. 計算平均測地線距離(Averaging GD values)
7. 編碼測地線特徵值(GD value encoding)
8. 測地線特徵向量排比與排名(GD vector alignment and ranking)

其中 CUDA 平行處理技術是應用於「計算測地線距離值(GD value calculation)」與「測地線特徵向量排比(GD vector alignment)」兩步驟，以提昇整體的運算效率。

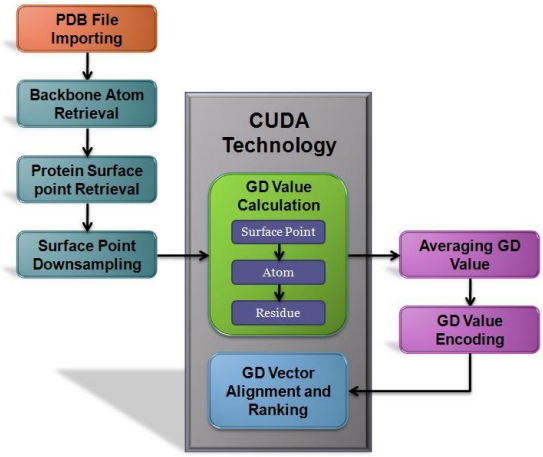

圖 3、系統架構流程圖，系統共分為 8 個步驟完成表面結構比對，其中 CUDA 技術應用於各胺基酸的測地線距離及排比計算模組。

各步驟細節說明如下：

(一) 輸入 PDB 檔

PDB 檔內容記載經實驗解析出的蛋白質結構幾何資訊，可由 PDB 官方網站下載相關資料(http://www.pdb.org/ 或 http://www.pdbj.org/)。經由輸入 PDB 檔案或檔名資訊，可取得組成該蛋白質所有原子在三維空間的座標資訊、胺基酸內容及二級結構資訊，這些資訊經過分析運算與篩選後，便能於後續蛋白質表面結構的比對過程進行下一步驟的分析與研究。

(二) 擷取骨幹(backbone atom retrieval)

蛋白質是由多個不同類型的胺基酸所組成，每個胺基酸又分別由多個不同的原子所組成，目前常見有 20 種不同類別的胺基酸，每一個胺基酸都是由 C、CA、N 及 O 四種原子組成主要

架構並且串接而成蛋白質結構的骨幹，而由骨幹中 CA 原子向外連接的部分稱為支鏈(Side Chain)，不同的支鏈的化學分子組合決定不同的胺基酸種類。若忽略支鏈部分的元素，僅針對骨幹的原子進行研究分析，可以簡化蛋白質的整體架構，減少計算分析所需的系統時間。

(三) 擷取蛋白質結構表面點

使用 PDB 檔內所記錄的各原子於三維空間中的座標，在記憶體空間先宣告整個蛋白質的體積形狀及大小，並藉由水分子沿著整個蛋白質結構表面進行滾動，形成可接觸面積(accessible surface area, ASA)的定義，並以基本單元的正立方體將蛋白質進行數位網格化，最後再透過數學形像處理學(mathematical morphology)技術定義蛋白質的表面區域並記錄所有表面點的座標位置，表面區域的正確定義是下一步計算測地線距離的重要依據。

(四) 表面點縮減取樣

為加快表面結構分析技術，在三維空間的結構分別對各個平面的表面點進行縮減取樣的動作，於同一平面中以 3x3 的網格為單位，取出其中 1 點表面點做為該範圍內的表面點代表(如圖 4)，以達到簡化表面點的目的，並大幅降低運算量，縮短計算時間。

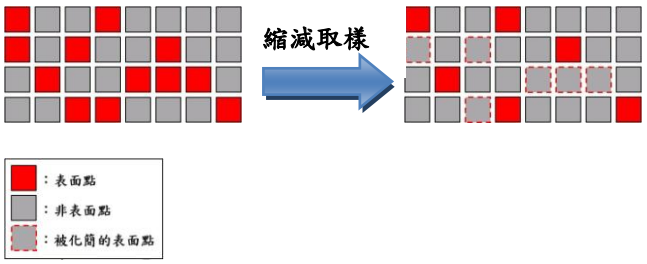

圖 4、化簡表面點示意圖，三維空間中某一平面的區塊，12 個表面點經過縮減取樣之後，僅剩下 6 個表面點。

(五) 計算測地線距離(GD value calculation)

為比較分析不同蛋白質的表面結構，需要先進行胺基酸序列的測地線特徵值計算，該特徵值的計算共分為三個步驟：

● 表面點間之測地線距離：

應用 Floyd-Warshall 演算法[4]計算出表面點之間的測地線距離。初始相鄰矩陣(Adjacency Matrix)的定義，需要先設定一個門檻值來確認各表面點之間是否有連通，為了滿足測地線必須沿著表面路徑的規範，本研究以表面點為中心，尋找同一平面在 3x3 網格範圍內的最遠距離，原先取$\sqrt{2}$作為連通的門檻值，但因為有簡化的步驟，使得定義連通的門檻值重新調整為$2\sqrt{2}$(如圖 5)。

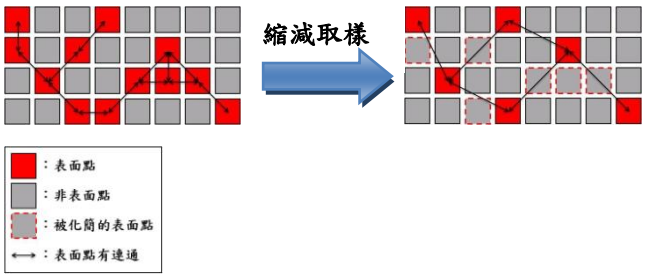

圖 5、表面點連通示意圖，化簡前的連通門檻值為$\sqrt{2}$，化簡後則須調整為$2\sqrt{2}$。

● 原子之間測地線距離：

將測地線的範圍由表面點擴大到原子之間，利用上一步所得到的表面點距離，藉由兩兩原子之間的所有相關表面點距離的平均計算，定義兩個原子之間的測地線距離，圖 6 舉一個範例說明兩個原子間測地線距離的計算方式。

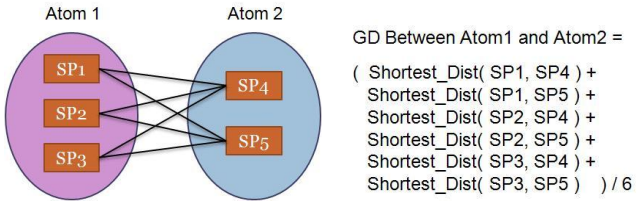

圖 6、假設第一個原子由 3 個表面點構成：SP1、SP2、SP3，第二個原子包含兩個表面點：SP4、SP5，則兩個原子之間的測地線距離，即為此兩組表面點之間所有組合交叉距離的平均。

● 胺基酸之間測地線距離：

胺基酸之間的測地線距離是由個別所屬原子間測地線距離的延伸，計算的方法與原子間的距離計算類似，採用上一步所得到的原子距離，藉由兩兩胺基酸之間的所有原子對的交叉測地線距離進行平均值計算，以獲得在結構表面上胺基酸之間的測地線距離，圖 7 是範例說明兩個胺基酸測地線距離的計算方式。

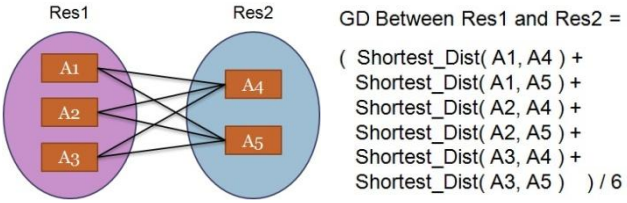

圖 7、假設第一個胺基酸包含 3 個原子：第二個胺基酸包含兩個原子，則兩個胺基酸之間的測地線距離，即為此兩組原子之間的個別測地線距離的平均。

(六) 計算平均測地線距離

在取得任兩兩之間胺基酸的測地線距離後，此步驟需要完成每個胺基酸相對應的測地線特徵值，並完成該蛋白質相對應的特徵向量檔。每一個胺基酸的測地線特徵值計算是以胺基酸在蛋白質序列的編號順序為基礎，先給定一個視窗範圍(本系統預設視窗範圍是設定前後各 10 個胺基酸)，依據序列編號，計算座落於目標胺基酸前後範圍內所有表面胺基酸測地線距離平均，作為各表面胺基酸的測地線特徵值。若範圍內存在有胺基酸為非表面胺基酸(亦即該胺基酸座落於蛋白質內部，沒有包含任何表面原子點的胺基酸)，則逕行跳過不列入平均的計算，圖 8 舉例說明如何對每一目標胺基酸在視窗範圍內進行測地線距離特徵值的計算方式。

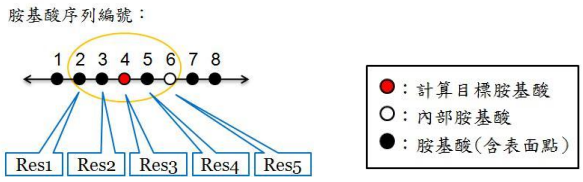

圖 8、計算區域測地線特徵的示意圖。假設有胺基酸序列編號 1 到 8 號共 8 個胺基酸，若視窗範圍為前後各 2 個胺基酸，欲計算序列編號第 4 號的測地線特徵值，則應加總其前後各兩個胺基酸的測地線距離再取平均，但因為編號第 6 號的胺基酸為非表面胺基酸，因此只需對編號第 2、3 及 5 號的胺基酸進行測地線距離的運算即可。

(七) 編碼測地線特徵值

由上一步驟取得各表面胺基酸的區域測地線特徵值後，為加速兩個蛋白質表面結構的匹配分析，本研究提出另一個編碼技術將測地線特徵值向量轉換成對應的英文字母，並形成一測地線特徵值序列。將表面結構的分佈以測地線特徵值序列表示，再經由動態規劃演算法進行字串排比的分析，檢索出表面結構相似的結構，圖 9 舉例說明區域測地線特徵值編碼的範例，實際視窗範圍為 21 個胺基酸，其中編碼查閱表是經由一個測試用結構資料集並統計分析測地線距離的實際數據分佈後，使用 26 個字母(A 到 Z)做為表示編碼的字元代號。

	Res1	Res2	Res3	…
GD特徵值(Å)	1.25	5.15	2.5	…
序列	A	C	B	…

圖 9：區域測地線特徵值編碼的範例。序列範圍值為 2.5Å，將區域測地線特徵值轉成對應的英文字母關係如左表，實際範例如右表格序列所示。

(八) 測地線特徵向量排比與排名

為確認所提出形狀特徵向量的穩定性並加速表面結構的比對，本研究使用 3D domain swapping 的蛋白質結構做為測試資料集，該結

構檔案的內容是由論文[3]中所蒐錄。資料集可分為33群不同功能的蛋白質結構群，總計有402個 PDB 檔，同一群的蛋白質結構皆存在不同變形程度的表面結構變異性。本系統事先將所有的蛋白質結構進行上述七個步驟的運算，每一個結構皆產生一個對應的區域測地線特徵序列，以利後續比對與排名計算。當系統輸入一個待查詢的蛋白質結構代碼後，本系統立刻執行區域測地線特徵序列的轉換分析，並與各資料群進行兩兩序列間相似程度的比對運算。序列比對主要是使用動態規劃演算法實做，並使用區域比對技術完成排比。採用區域式排比主要原因是為了避免序列間長短差異太大而產生過多的縫隙，因而影響整體比對結果的表現。

四、演算法說明

本論文使用許多各種不同的演算法達成表面結構的比對分析，各種演算法的細節描述於下列段落。

(一) Smith-Waterman 演算法

Smith-Waterman 演算法的內容如圖 10 所示[18]，定義 a 與 b 為兩序列且字串長度分別為 m 和 n，H(i,j)紀錄 a[1…i]與 b[1…j]於字尾(Suffix)處之間的最大相似分數，於一開始必須初始 H(0~n, 0)與 H(0, 0~m)為 0，若 a_i 等於 b_j，則 w(a_i, b_j)為正確比對所得到的分數，若 a_i 不等於 b_j，則 w(a_i, b_j)為錯誤比對所得到的分數。本論文中，正確比對設定為+1 分，錯誤比對為-1 分。經過局部比對演算法後，記錄相似分數的 H 陣列中的最大值，即為兩序列經比對後的相似分數。

$$H(i,j) = \max\{\ 0,\ H(i-1,j-1) + w(a,b),\ H(i-1,j) + w(a,-),\ H(i,j-1) + w(-,b)\ \}$$
$$(\ 1 \le i \le m,\ 1 \le j \le n\ )$$

圖 10、Smith-Waterman 演算法內容

將所有特徵值序列進行交叉局部比對，並根據局部比對所得到的分數當作排名的依據，分數由高到低進行排序，排序後的結果即為排名的輸出結果，排名越前面代表此兩個蛋白質的測地線特徵值序列相似的部分越長，也就是兩個蛋白質的表面結構越相似，因此藉由排名的結果，可以清楚的分辨出各個蛋白質之間的表面結構相似程度。

(二) CUDA 平行處理技術應用

本論文的 CUDA 平行處理技術主要應用於計算量較大的兩個部分：計算表面點之間的測地線距離及交叉比對測地線特徵值序列。

- CUDA 加速表面點間的測地線距離計算

利用 Floyd-Warshall 演算法，計算表面點之間的測地線距離，其演算法複雜度為$O(n^3)$，n 為表面點的個數，蛋白質經過本研究方法的前三個步驟：擷取骨幹原子、數位網格化及表面點縮減取樣後，每一個蛋白質結構的表面點個數仍然是超過數千個點資訊，甚至有超過萬點的結構資訊。此外，使用者可能同時有大量比較的需求，需要一次輸入多個 PDB 檔，所以計算量需求之龐大可以預期。因此，改善這部分的運算效能便成為 CUDA 技術應用的主要焦點，Floyd-Warshall 演算法如虛擬碼(1)所示：

(1) Floyd-Warshall 演算法

```
for k ← 1 to n do
    for i ← 1 to n do
        for j ← 1 to n do
            if (D_{i,k} + D_{k,j} < D_{i,j}) then
                D_{i,j} ← D_{i,k} + D_{k,j};
```

其中 n 為表面點的個數，$D_{i,j}$ 表示表面點 i 到表面點 j 所需的距離，若 i → j 的距離較 i → k → j 的距離更遠，則更新為較近的距離，但因為表面點之間的連線沒有方向性，$D_{i,j}$ 等同於 $D_{j,i}$ ，利用此特性，只需要計算一半的相鄰矩陣即可，再利用矩陣對稱的特性，將索引 i、j 對調，即可於一次運算完成兩次距離的更新，改良後的演算法如虛擬碼(2)：

(2) 利用對稱性改良 Floyd-Warshall 演算法

```
for k ← 1 to n do
    for i ← 1 to n do
        for j ← 1 to i do
            if (D_{i,k} + D_{k,j} < D_{i,j}) then
            begin
                D_{i,j} ← D_{i,k} + D_{k,j};
                D_{j,i} ← D_{i,j};
            end
```

針對改良後的 Floyd-Warshall 演算法進行 CUDA 加速，CPU 執行的部分如虛擬碼(3)：

(3) CUDA 加速 Floyd-Warshall 演算法 (CPU 部分)

```
for k ← 1 to n do
    floydGPU(k);
```

CPU 部分迴圈呼叫 floydGPU 函式，此函式為 CUDA 平行處理的主要函式，並稱為核心函式。當呼叫此函式時，顯示卡便會開始規劃所有的執行緒去執行該核心函式，依照不同的執行緒辨識碼，各自產生不同的對應索引值，於核心函式內分配各自的工作，故核心函式又稱為每執行緒函式(per thread function)。核心函式 floydGPU 可參考虛擬碼(4)：

(4) CUDA 加速 Floyd-Warshall 演算法 (GPU 部分-核心函式)

```
Procedure floydGPU(k)
begin
    不同執行緒辨識碼，產生不同索引值 i 和 j
    if (D_{i,k} + D_{k,j} < D_{i,j}) then
    begin
        D_{i,j} ← D_{i,k} + D_{k,j};
        D_{j,i} ← D_{i,j};
    end
end
```

主要鎖定原演算法的 i 與 j 兩層迴圈，運用大量的執行緒平行處理迴圈，每條執行緒利用執行緒辨識碼各自產生索引值 i 和 j，並計算出 $D_{i,j}$ 與 $D_{j,i}$。

- CUDA 加速交叉比對測地線特徵值序列

交叉比對的演算法如虛擬碼(5)：

(5) 交叉比對演算法

```
for i ← 1 to n do
    for j ← 1 to n do
        S_{i,j} ← localAlignment( seq_i , seq_j );
排序(排名)局部比對分數陣列 S
```

其中 n 為查詢比對的 PDB 檔案個數，seq_i 為第 i 個 PDB 檔的蛋白質測地線特徵值序列，$S_{i,j}$ 記錄 seq_i 與 seq_j 之間的局部比對分數，交叉比對完所有的蛋白質並記錄其間的比對分數於 S 陣列

後，將陣列分數由大到小排序，即可得到蛋白質表面結構比對的最終結果及相似度的排名。

本研究將針對比對的部分進行 CUDA 加速，CPU 的部分如虛擬碼(6)：

(6) CUDA 加速交叉比對演算法(CPU 部分)

```
for i ← 1 to n do
        seqCmpGPU(i);
```

鎖定原演算法的 *j* 迴圈，利用大量執行緒平行處理。CPU 虛擬碼的部分是透過迴圈呼叫 seqCmpGPU 核心函式，其函式如虛擬碼(7)：

(7) CUDA 加速交叉比對演算法
(GPU 部分-核心函式)

```
Procedure seqCmpGPU (i)
begin
        不同的執行緒辨識碼，產生不同索引值 j
        S_{i,j} ← localAlignment( seq_i , seq_j );
end
```

於函式內，同樣利用不同執行緒辨識碼製造出不同的索引值 *j*，依照不同的索引值 *j*，提供每個執行緒運算不同的比對分數 $S_{i,j}$。

五、實驗結果

本論文的測試環境描述如下：作業系統是 64 位元的 Windows 7 系統、記憶體 12GB、中央處理器是採用 Intel Core i7 950 3.06G 及顯示卡的規格是 NVIDIA GeForce GTX 580。測試樣本為 3D domain swapping 的資料集，其中分為 33 個不同的蛋白質結構群，總共包含 402 個 PDB 檔，同一群的蛋白質結構皆存在不同變形程度的表面結構變異性。測試過程是採用去一交叉驗證法(Leave-One-Out cross verification)，每一個蛋白質皆由原資料群中取出，並進行與剩餘的 401 個表面結構進行比對辨識，再以整體比對結果進行分析。

(一) 比對結果排名與統計

● 平均群排名

在所有蛋白質經過個別取出並執行區域測地線特徵比對排名後，依照分類群的排名方式進行實驗結果分析，即使用分類群排名出現的先後順序決定排比結果，例如：第 1 群第 1 個蛋白質與其他 401 個不同的蛋白質進行比對並排序後，若前六名最相似的蛋白質分別屬於第 1、2、2、1、4、3 群，則表示第 1 群的排名為第 1 名，第 2 群的排名列為第 2 名，第 4 群排名列為第 3 名而第 3 群的排名結果列為第 4 名。各群的排名排序後，將同一群的排名加總並平均，便能得知交叉比對後，同一群結構的相似度排名分布情形，例如：第 1 群的各個蛋白質經分別與其他蛋白質交叉比對後，則加總第 1 群的排名結果，第 2 群至第 33 群以此類推。

演算法的預設參數經過多次不同參數的組合測試後，最終選擇序列相鄰範圍為前後各 10 個胺基酸，用以計算平均測地線距離的視窗範圍，而測地線距離編碼表的設定則是選擇 3.4 Å 當作編碼的範圍值。經過交叉比對排名，所得到的平均排名為 1.21 名，表示各群內的蛋白質一一經過個別比對後，同一群的排名幾乎是平均排列在第 1.21 名(共 33 群)，即經過比對分析表面結構的測地線距離特徵後，可以正確將變形後的蛋白質歸類到同一群的正確性是非常高的。

● PR(Precision-Recall) Curve 統計曲線圖

本研究定義不同分數的門檻值，分別進行蛋白質表面結構相似性的分析及分類，其中 P(Positive)是指表面結構相似，N(Negative)則表示表面結構不相似。根據兩蛋白質進行局部排比後的分數當作判斷是否相似(P 或 N)的依據，判斷相似度分數的門檻值由滿分的 10%、12.5%、

15%等持續增加到 100%為止，每一個門檻值進行一次辨識並記錄實驗結果。比對時若該蛋白質可以正確分類至同一群，預測結果與實際分類相符並正確歸類時則為 T(True)，反之則為 F(False)，TP(True Positive)表示實際運算後為正確分類的蛋白質總數，FP(False Positive)則為非同一群的蛋白質結構卻被預測為同一群的蛋白質總個數；同理 TN(True Negative) 表示非同一群的蛋白質結構可以正確判斷為非同一群結構的總數；FN(False Negative)則表示同一群的蛋白質結構卻錯誤判斷為非同一群結構的總數。PR(Precision-Recall) Curve 統計曲線的取得是經由下列統計公式獲得：

(1) Precision = TP / (TP + FP)

(2) Recall = TP / (TP + FN)

由 PR Curve 曲線圖(如圖 11)可知，曲線下的面積越大，代表演算法的準確性越高。本研究的測試結果獲得的面積區域約為 0.954，可以歸納由測地線特徵值比對蛋白質表面結構的相似性是具有高效能的辨識特徵。

(二) CUDA 加速成果

經由計算測地線特徵值到產生特徵值序列所需要完成的時間曲線如圖 12 所示。曲線 CPU 代表使用純 CPU 計算的時間曲線，曲線 CPU+GPU 則為 CPU 搭配 CUDA 技術 GPU 平行處理的時間曲線，由圖可見，單純利用 CPU 計算所需的時間隨著運算的表面點越多而形成指數性成長。當加入 CUDA 平行處理技術後，曲線即進步為線性成長狀態。以資料集中最大的蛋白質為例，其表面點於縮減樣本後還剩有 9750 個點，原本需要約 37 分鐘的運算時間，在應用 CUDA 技術後，僅需要 34.05 秒即可計算完畢，時間加速可達 65 倍之多，至於計算全部 402 個蛋白質，使用純 CPU 計算一共需要 30896 秒，CPU+GPU 的計算僅需 577 秒，平均每個蛋白質的計算效率約加速了 54 倍。

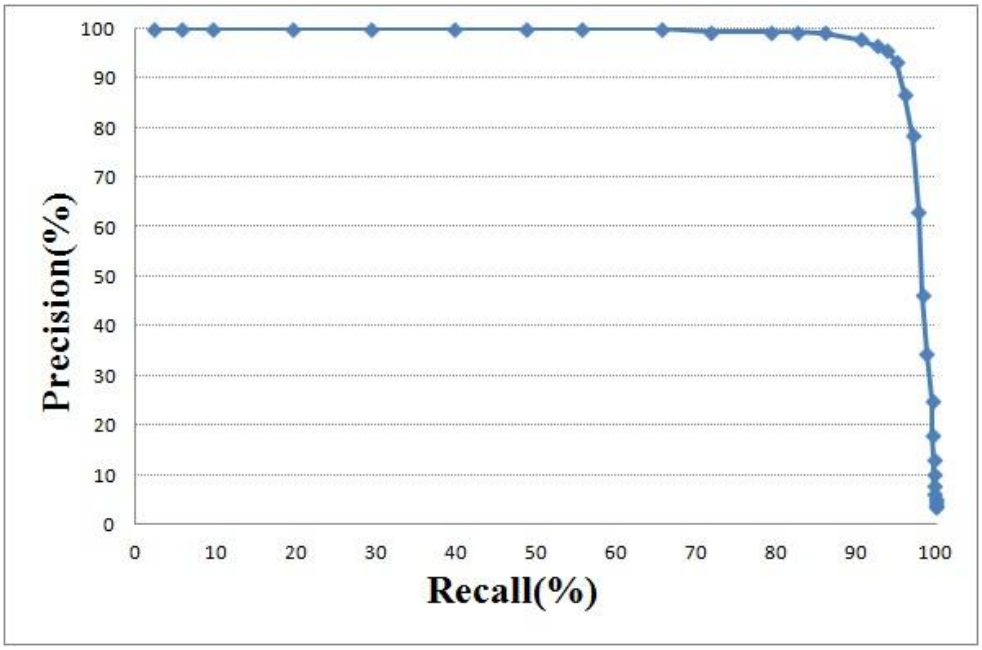

圖 11、PR Curve 曲線圖

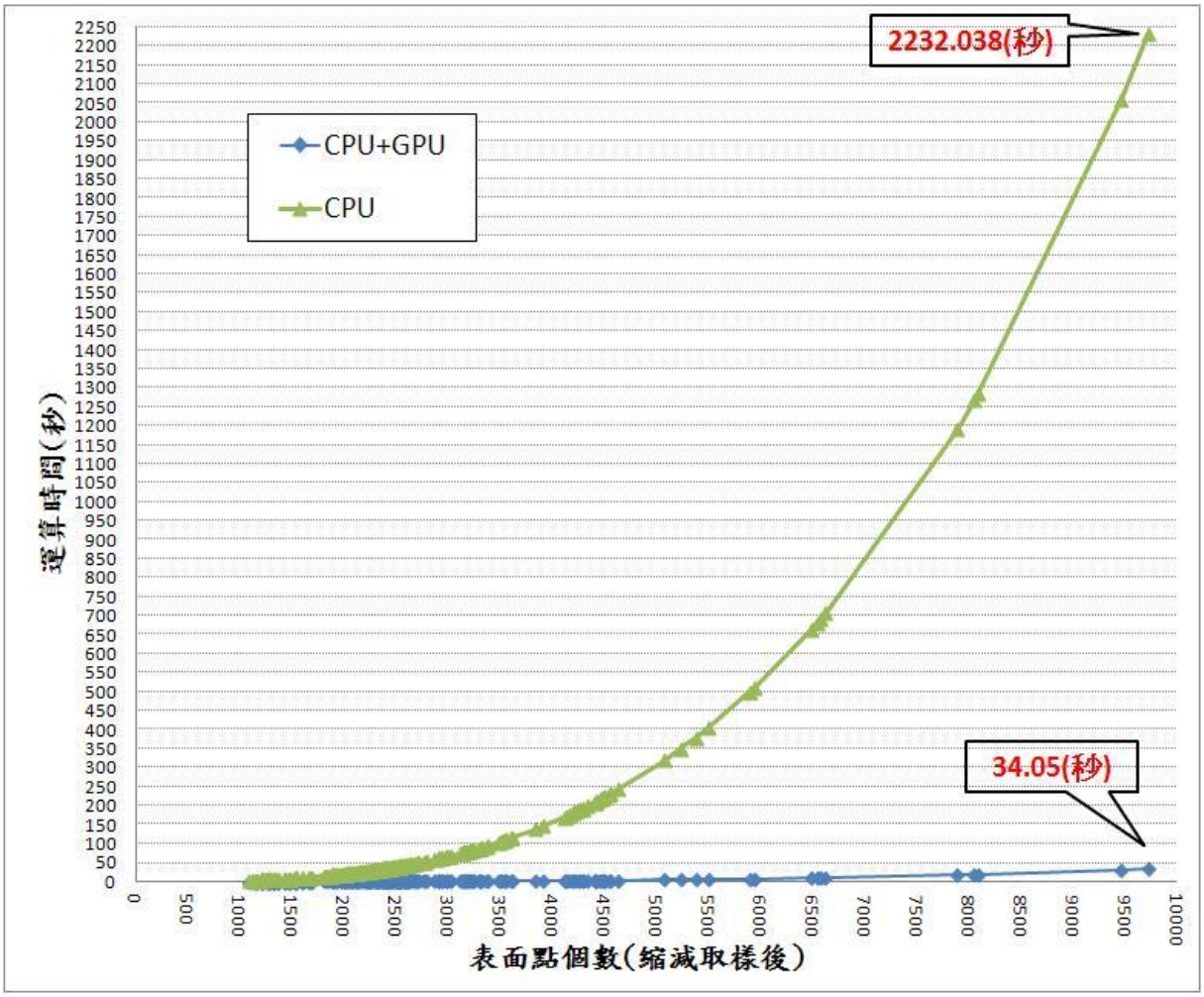

圖 12：純 CPU 與 CPU 搭配 GPU 加速，計算 402 個 PDB 檔分別產生特徵值序列所需的時間曲線圖

對於後續 402 條序列進行交叉比對至排名的部分，單純以 CPU 執行共需 1237 秒，改由 CPU 搭配 GPU 加速後，運算時間可降低為 36 秒，兩者相較之下，運算時間也相差約 34 倍。最後，由輸入檔案至排名結束的運算時間，使用單純 CPU 計算一共需要 32134 秒，CPU 搭配 GPU 計算一共需要 613 秒，整體平均加速約 52.4 倍。運用 CUDA 平行處理技術於蛋白質表面結構比對的演算法，其改進的效能可見一斑。

致謝：本論文感謝國科會(計畫編號：NSC98-2221-E-019-031-MY2 及 NSC100-2321-B-019-004)及國立台灣海洋大學海洋生物科技及環境生態中心之計畫經費贊助。

六、參考文獻

[1] A. S. Aytuna, *et al.*, "Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces," *Bioinformatics*, vol. 21, pp. 2850-5, Jun 15 2005.

[2] H. M. Berman, *et al.*, "The Protein Data Bank," *Nucleic Acids Res*, vol. 28, pp. 235-42, Jan 1 2000.

[3] C. H. Chu, *et al.*, "Detection and alignment of 3D domain swapping proteins using angle-distance image-based secondary structural matching techniques," *PLoS One*,

vol. 5, p. e13361, 2010.

[4] R. W. Floyd, "Algorithm 97: Shortest path," *Communications of the ACM*, vol. 5, p. 345, 1962.

[5] P. Francis-Lyon, *et al.*, "Sampling the conformation of protein surface residues for flexible protein docking," *BMC Bioinformatics*, vol. 11, p. 575, 2010.

[6] P. F. Gherardini and M. Helmer-Citterich, "Structure-based function prediction: approaches and applications," *Brief Funct Genomic Proteomic*, vol. 7, pp. 291-302, Jul 2008.

[7] L. Holm and C. Sander, "Mapping the protein universe," *Science*, vol. 273, pp. 595-603, Aug 2 1996.

[8] R. Kolodny and N. Linial, "Approximate protein structural alignment in polynomial time," *Proc Natl Acad Sci U S A*, vol. 101, pp. 12201-6, Aug 17 2004.

[9] B. Lee and F. M. Richards, "The interpretation of protein structures: estimation of static accessibility," *J Mol Biol*, vol. 55, pp. 379-400, Feb 14 1971.

[10] W. Liu, *et al.*, "Protein structure alignment using elastic shape analysis," presented at the Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology, Niagara Falls, New York, 2010.

[11] W. Liu, *et al.*, "A mathematical framework for protein structure comparison," *PLoS Comput Biol*, vol. 7, p. e1001075, 2011.

[12] Y. S. Liu, *et al.*, "Using diffusion distances for flexible molecular shape comparison," *BMC Bioinformatics*, vol. 11, pp. -, Sep 24 2010.

[13] Z. P. Liu, *et al.*, "Predicting gene ontology functions from protein's regional surface structures," *BMC Bioinformatics*, vol. 8, p. 475, 2007.

[14] NVidia, "CUDA Programming Guide Version 4.0," *http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf*.

[15] C. A. Orengo, *et al.*, "CATH--a hierarchic classification of protein domain structures," *Structure*, vol. 5, pp. 1093-108, Aug 15 1997.

[16] A. Prlic, *et al.*, "Pre-calculated protein structure alignments at the RCSB PDB website," *Bioinformatics*, vol. 26, pp. 2983-5, Dec 1 2010.

[17] F. M. Richards, "Areas, volumes, packing and protein structure," *Annu Rev Biophys Bioeng*, vol. 6, pp. 151-76, 1977.

[18] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J Mol Biol*, vol. 147, pp. 195-7, Mar 25 1981.

[19] W. R. Taylor, *et al.*, "Multiple protein structure alignment," *Protein Sci*, vol. 3, pp. 1858-70, Oct 1994.

[20] J. D. Watson, *et al.*, "Predicting protein function from sequence and structural data," *Curr Opin Struct Biol*, vol. 15, pp. 275-84, Jun 2005.

[21] S. Y. Yin, *et al.*, "Fast screening of protein surfaces using geometric invariant fingerprints," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, pp. 16622-16626, Sep 29 2009