以遺傳演算法與最近鄰居分類法篩選遺傳疾病基因之研究

尹柏元[1] 黃代鈞 楊又潔 江素倩 紀曉燦 陳春賢[2]
長庚大學資訊管理學系、長庚生物資訊中心
E-mail: [1] m9144022@stmail.cgu.edu.tw, [2] cchen@mail.cgu.edu.tw

摘要

尋找基因與疾病的關係是醫學與生命科學致力追求的目標之一，也是生物資訊領域的研究重點。目前，微陣列(Microarray)技術的使用，有助於個別基因表現量的定量分析。醫學上使用基因表現資料針對遺傳疾病關鍵基因的搜尋與篩選，可借用特徵選擇(Feature Selection)的資訊技術來處理。一般我們所採用來解決此一問題的分析方法，除需要考慮精確度外，所需要的時間成本也是不可忽視的考慮因素。在許多實際的應用上，如果搜尋空間非常大，則耗竭式的搜尋顯的切不可行，主因其大量的運算時間需求所致。

遺傳演算法(Genetic Algorithm, GA)與最近鄰居分類法(K-Nearest Neighbor, KNN)的併用，可快速找到許多組具區別力的關鍵特徵，並可將這些關鍵特徵的區別力按在各組出現頻率的次數作一統計評分，根據出現頻率的高低作為判斷相關性高低的依據。然而，搜尋這些關鍵特徵組的時間需求凸顯了此一方法的弱點，因此，如何進一步改善與分析此方法來有效減少運算時間並兼顧精確度，是本文的探討重點。

關鍵字: 遺傳演算法(Genetic Algorithm, GA)，最近鄰居分類法(K-Nearest Neighbors, KNN)，特徵挑選(Feature Selection)，基因挑選(Gene Selection)，基因捐贈(Gene Donation)。

一、緒論

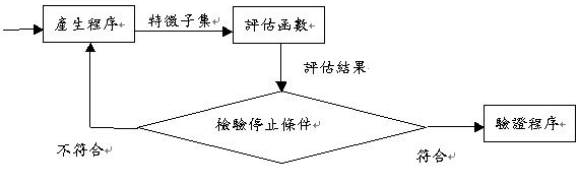
多種的微陣列技術已在過去數年間被逐漸發展，直至目前來看，該領域的技術發展仍維持穩定成長。cDNA 陣列的設計是著眼於基因表現量的觀察，cDNA 陣列可同時檢測數千或數萬個基因的表現量，因此生物學家可藉實驗產生出大量的原始資料，分析後可幫助了解生物的生命運作系統，包括基因的調控、細胞發展、生物演化，甚至是基因與疾病之間的複雜關係[1]。在癌症分子診斷的問題中，一般比較癌症患者與正常人的基因表現量作為診斷的基礎；但在一張基因數目有數千個的基因維陣列中，怎麼找出那些基因可能直接與該癌症相關，並可藉以將未知的樣本區分為正常人與癌症病患兩類呢? 據 Chiara Romualdi 等人的研究，以 cDNA 微陣列的基因表現量來鑑別一個病患是否患有某特定癌症，在以具區別力的基因作為辨識基礎的情況下，倘若不篩選掉與該癌症不相關的基因，將無法達到有效的識別正確率，可見基因篩選的重要性[21]。如果我們將一個基因視為一個特徵，搜尋的目標就是要找出與癌症或遺傳疾病有關的基因，因此篩選癌症的相關基因可用特徵挑選的技術來處理。在特徵挑選問題中的所有特徵，大部分有可能是不相干或相互依存的特徵，這些不相干或多餘的特徵不會對類別鑑識有所幫助[12]。

特徵選擇的定義曾經被許多學者從不同角度詮釋過，主要可分為以下四個部分[5]:

1 理想的(Idealized): 找出一組最小的特徵子集，使得該子集足以區別不同的類別。
2 古典的(Classical): 從 N 個特徵值中找出一組大小為 M 的子集合 (M<N)，使得這組在所有大小為 M 的解中為最佳者。
3 改良預測精確度(Improving Prediction Accuracy): 這種特徵選擇的方法主要在改進預測的精確度；或是在不影響精確度的情況下，降低目前子集合的大小。
4 近似原始資料分佈(Approximating Original Class Distribution): 以挑出的特徵子集作測試，產生出的類別分佈必須與資料的原始分佈盡可能相近，這是此方法的目的。

總結特徵挑選的目的雖然是嘗試找出一組最小的子集，但應兼顧三、四兩項原則。至於一般特徵選擇的方法，主要包含以下四個步驟(圖一):

1. 產生程序(Generation Procedure): 可以視為一個搜尋的程序，負責產生下一代的候選子集合。
2. 評估函數(Evaluation Function): 負責評估子集合的適合度，並與前一代的最佳值做比較。
3. 停止條件(Stop Criterion): 決定特徵選擇的終止條件，避免無止盡的搜尋動作。
4. 驗證程序(Validation Procedure): 當取得子集合時，用以評定該子集合是否合理的程序。



圖一、特徵選擇的主要程序

當我們在選擇特徵挑選方法來處理生物醫學的問題時，存在著幾個使我們必須要考慮該演算法限制的問題：

1. 資料規模： 一般的特徵挑選方法所處理的特徵數目最多只有 1000 個左右[24]，這些方法是否適合用於動輒數千個、甚至上萬個基因的基因微陣列分析上?舉例而言，若欲對一個大小為 N 的特徵空間做特徵選擇，根據以上的程序，期望能在 2^N 個子集合中找出最佳一組具區別力的特徵，若以耗竭式搜尋的產生程序則需要相當高的時間成本。舉例而言，若已知有 K 個特徵對某一分類的問題具有區別力，那又該如何找出這 K 個特徵?以本文所用的 2000 個基因為例[15]，假設 K=5，則組合約有 2.7×10^{14} 種，因此窮舉法(Exhaustive enumeration)並不可行。
2. 資料型態：方法所能處理的資料型態也是必須要考量的因素[5]，若方法只能處理不連續的資料，在基因微陣列的分析上也許就不適用。
3. 可能解的多樣性：就癌症的分子診療而言，被收集分析的癌症病患檢體數目通常非常珍貴且稀少，然而基因微陣列上的基因卻是非常的多，也就是說，樣本小而特徵多，因此容易找到許多組具區別力的特徵子集合[11]，這些基因子集合在輔助分類上具有統計的意義，但在生物意義上，不見得與某一癌症相關[17]；話雖如此，但真正在生物上具區別力的基因也仍有可能出現在這些具區別力的子集合中。

產生程序與評估函數是特徵選擇程序的兩個重要步驟，前者扮演子集合產生的角色，後者則扮演評估此子集合是否具分類區別力的角色；本文針對兩個步驟，分別依需求與時間成本選擇了遺傳演算法與最近鄰居分類法兩個方式[13,18,19]。遺傳演算法是由 John Holland 在 1970 年代中期所提出[10]，基本上是以達爾文(Charles Robert Darwin)的演化論為基礎所發展而成，目前已被證實且廣泛運用的最佳解搜尋方法；有別於其他許多演算法只能找出單一組的特徵[5,16]。遺傳演算法在基因篩選的問題中能找出多組符合停止條件，使這些具區別力的基因被統計後，其出現頻率有大小之別，亦是本論文採用遺傳演算法的原因。至於評估函數方面，雖然有多種定量的分類方法[2,7]，但所需的時間複雜度大小不同，為縮短運算時間，最近鄰居分類法是最簡單而可行的選擇[3, 4 , 14]。在遺傳演算法與最近鄰居分類法的合併使用中，我們試圖將兩個類別完整區隔開來，依基因出現在區別的基因組的高低加以排序，出現頻率愈高，表示一個基因與目標癌症愈相關，反之則愈低。

Leping Li 等人在 2001 年將 GA/KNN 的方法用在基因微陣列分析上[15]，文中針對不同染色體長度的靈敏性(Sensitivity)、再現性(Reproducibility)、穩定性(Stability)等部分作探討，並論述 GA/KNN 用在類別預測的準確度與不錯的結果。但在 Li 文中對於演算法的速率、取可能解的組數與適當的終止條件等相關議題上都還留有討論的空間，故本文將針對 GA/KNN 演算法的改進與可能解的組數加以探討。

二、遺傳演算法與最近鄰居分類法回顧

(一) 最近鄰居分類法(K-Nearest Neighbors)

最近鄰居分類法是一種以比較相似程度為基礎的方法，每個樣本可視為 n 維空間的一個點。當一個新樣本被讀入時，我們找尋與其最相近的 K 個點(本文中取 K=3)；而距離的定義在本文中是取兩點間的歐基里德距離(Euclidian Distance)[9]。舉例而言， $X=(x_1,x_2,\ldots,x_n)$與 $Y=(y_1,y_2,\ldots\ldots,y_n)$是兩個樣本點，X、Y 兩點間的歐基里德距離為：

$$\mathrm{dis}(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

倘若與一個待分類樣本最接近的三個點皆屬於同一類別 A，則分類辨識成功，此樣本被視為 A 類別；否則，若這四點分屬於兩個類別以上，我們視為未分類成功。

(二)遺傳演算法(Genetic Algorithm)

遺傳演算法利用電腦模擬地球上生物發展的演化過程，以適者生存的自然選擇方式找出一個問題的可能最佳解。生物產生後代, 並利用染色體紀錄其遺傳特徵, 使該物種的特性得以保持到下一世代。

1. 染色體表示方式 (Chromosome representation)

一般遺傳演算法用以表示染色體的方式是採二進位表示法[6]，但在我們的應用中，染色體內儲存的是十進位制的基因編號，如圖二，親代的染色體含有 1、12、6、8、9、2 等 6 個基因，一個染色體代表一組基因(特徵)，我們的目的即在針對某一

遺傳疾病，搜尋具區別力的基因(特徵)組。

2. 初始族群 (Initial population)

遺傳演算法包含了一個存有許多可能解的染色體族群，這樣的一個族群需要在遺傳演算法的程序開始時進行初始化。族群在面對不同的問題時有不同的變化，在本文中我們用隨機的方式挑選可能的染色體族群。

3. 適應值函數 (Fitness function)

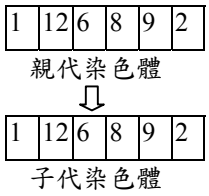
適應值函數在遺傳演算法中主要扮演評估染色體優劣的角色。通常我們會設定一個終止條件 T，在本文中設 T=(M-α)/M，M 為訓練樣本的組數，T 為染色體上這組基因的遺傳疾病辨識正確率，α/M 則為辨識錯誤率。本文採用 40 個訓練樣本，我們以辨識正確率 95%為終止條件,即α設定為 2。

4. 遺傳機制 (Genetic operator)

遺傳演算法的三個重要機制，再製、交配、突變分述如下：

4.1. 再製 (Reproduction)

親代將自己的染色體複製給子代，使子代仍然保有該物種的生物特性。



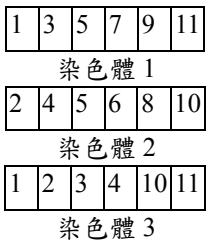
圖二、再製示意圖

4.2. 交配 (Crossover)

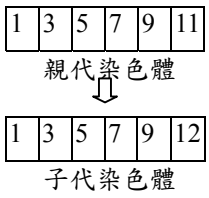
交配出現在兩個親代各交換他們相同位置的部分基因給子代, 當然有許多文獻探討交配的方式與結果分析[20,22 ,23]; 但基本的方式不外乎單點(one point)與雙點(two-point)兩種。 在本文中，我們像 Li 一樣未採此遺傳機制。

4.3. 突變 (Mutation)

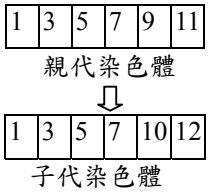
儘管再製與交配兩種方式已經可以有效的搜尋或再組現有的遺傳特徵，但在十進位的染色體編碼方式中，倘若親代的染色體沒有涵蓋所有的存在的基因時，子世代就非常有可能無法涵蓋所有的基因，以至於無法尋得最佳解的狀況。舉例而言，在十進位的染色體編碼方式中，假設目前共有 12 個基因可供挑選，但現有的 3 條染色體只包含 11 個基因(圖三)，無論進行再製或交配都無法使第 12 個基因被選入，因此突變機制有存在的必要[8]。有別於交配的演化方式，子代雖是由單一個親代產生，但又與再製略有不同；親代可針對本身作隨機單點突變或隨機多點突變。例如，圖四的親代單點突變自己的第 6 個基因成為子代，圖五則是多點突變本身的第 5、6 個基因成為子代。在本文中採用的方法為單點基因突變，每個基因突變的機率相等。



圖三、染色體配置圖



圖四、隨機單點突變示意圖



圖五、隨機多點突變示意圖

以上的機制是被用來建立新的與改善一組候選解所用，並盡可能與生物在群體中的生活類似，不同的程序是用來產生子世代(Offspring)並提昇其對自然的適應程度。遺傳演算法在初始族群中選擇親代進行再製、交配與突變三個機制，當所有親代均產生子代後，即更新原始族群。

三、改良式遺傳演算法與最近鄰居分類法

(一)演算法

就圖一而言，GA 被用來當作產生子集的"產生程序"，KNN 則被當成評估函數，故實驗結合 GA 與 KNN 兩個演算法作為基因挑選的方法，方法是找出 10000 組誤差在 5%以內的解，然後再以統計的方式分析個別基因在這 10000 組中的出現頻率，出現頻率愈高，表示該基因與問題中的遺傳疾病愈相關。以本文所用資料為例，當取 40 個訓練樣本時，必須有 38 個樣本都分類正確，這樣的基因組才符合區別力的需求。假設在 200 條染色體的族群中，以一般遺傳演算法的流程而言，要從 2000 個基因中挑出 5 個符合停止條件的解的機率為:

$$P(optimal\_five)=200\times 1/C(2000,5)\fallingdotseq 7.54\times 10^{-13} \quad (2)$$

根據式(2)的結果，以遺傳演算法找尋最佳解仍

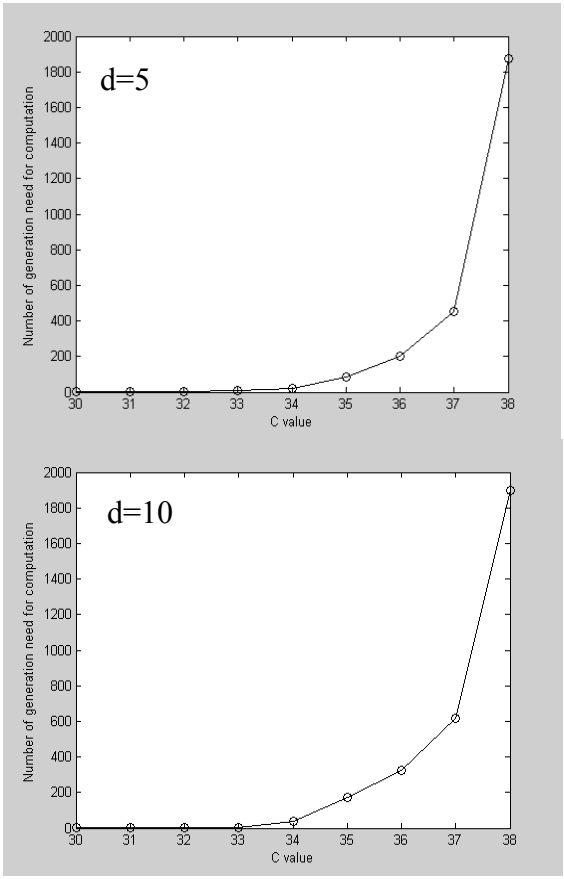
要耗費相當長的一段時間;若以目前的速度觀察，欲取得 10000 組解，時間成本仍然太高。因此，筆者嘗試提出可能改善的方式，演算法如下(流程圖附錄一):

1. 每1條染色體 (Chromosome)包含N個基因的方式，建立一個包含 150 條染色體的初始族群。
2. 利用 KNN 分類法計算出每條染色體的適應值 (Fitness value)。適應值的計算方法: 針對每個訓練樣本找出與其歐基里德距離最近的 3 個樣本,如果 3 個樣本與該訓練樣本皆屬於同一類別，則視為分類正確，並給予該訓練樣本 1 分，總共有 40 個訓練樣本，所以最高總分為 40 分。重複上述做法，計算每一條染色體的適應值。
3. 檢查是否有染色體的分數已達設定標準 (M-α)/M (本文中 M=40，α=2 分)，若有，則跳至步驟 7;若無，則繼續下面步驟。
4. 當代適應值最高的染色體對適應值較低者進行基因捐贈 (下文介紹)。
5. 適應值最佳的染色體進行記憶式調整(下文介紹)，其餘每一條染色體進行隨機選擇的突變。
6. 返回步驟 2。
7. 將該染色體中所包含的基因紀錄下來，並加以統計之後返回步驟一，直至取得 10000 組為止。取得 10000 組解之後，統計前 N 個最常出現的基因並加以紀錄(本文取 N=50)。

(二)基因捐贈(Gene donation)

每代除適應值最高的染色體之外，剩下的 199 條仍然處於隨機突變的狀況。在這 199 條染色體中，若有任何一條的適應值高於原最佳者，則將舊者取而代之。這樣的機制在 KNN 的分類方法之下，要將愈多的樣本全部分類正確，所需要的時間就愈久；因為染色體上的基因有 750 個，待測的基因共有 2000 個，在每一代的每條染色體只突變一個基因的情況下，至少要 9 個世代後，所有基因才有一次機會出現在染色體族群上，若要找出具有區別力的基因組合勢必要更多的世代。因此在此實驗中，我們發現當適應值<36/40 時，並不會佔去太多的運算時間;然而，當適應值超過 36/40 以後，適應值每攀高 1 分，所需等待的時間往往成倍數以上的增加。圖六是針對染色體長度 d(本文取 5 與 10)在搜尋 1 組具區別力的基因組合的代數比較，以這樣的情況看來，取得 10000 組解的時間成本恐怕是無法讓人接受的。

基因捐贈的想法來自於人類社會發展中，知識教育或經驗傳遞的概念，知識或經驗高者可將其知識或經驗在短時間內傳授給知識或經驗較低者，以避免長時間的自我摸索。可將適應值較低的染色體視為族群中經驗較不足者，適應值最高的染色體則視為族群中的智者，在族群發展的過程中，智者對於後進的指導扮演著重要的角色；此外，遺傳工程也普遍使用此一機制，例如研究者嘗試把抗病蟲害基因從一稻米品種植入另一稻米品種。在本文中依適應值的落差，對剩餘的 199 條染色體做 3 代一次的基因捐贈，並以 5 分為一個等級來捐 1 個基因。舉例而言，1 號染色體的適應值 36/40 是當代最佳的染色體，2 號染色體的適應值為 26/40，則 1 號染色體捐贈 2 個基因給 2 號染色體。此方法的重要目的在於幫助適應值最高者的組合有所變異，進而找尋出比目前更佳的組合。



圖六、取得一組具區別力的基因所需時間比較圖

(三) 記憶式調整(Memorial adaptation)

在遺傳演算法中，突變這個機制是早已存在的，但本機制與突變的不同在於幫助適應值最高的染色體在捐贈本身基因之餘，還可進行自身的突變。在原文作者所使用的方法中，突變的方法有可能使染色體的適應值在突變之後變低，故此法僅用於適應值最高的染色體。最高分染色體先從第一個基因開始進行隨機突變，倘若適應值變高，則以新的基因置換舊者；若分數變低，則保留原基因，並針對第二個基因進行突變，直至該染色體最後一個

基因為止，倘若適應值並無變化，則再製自己進入下一代。

(四)群體記憶式調整(Multiple memorial adaptation)

當然，在發展的過程中，適應值相等且最高的染色體可能不只一條(圖七)。這衍生出兩個可能的延伸方式:

1. 這些染色體中，何者應被設為目前的最佳解，進而執行基因捐贈的程序?
2. 這些與目前最佳解的適應值相同或只差 R(在本文中，R=1，這類似於一般遺傳演算法選擇 Top n 存活下來的方式)分的染色體是否有必要接受基因的捐贈?

染色體 1

1	3	5	7	9	11

染色體 2

2	4	6	8	10	12

染色體 3

17	27	38	49	56	6

圖七、適應值相等但內容不同的染色體

關於問題 1 或許有不同的解決方法，但本文所採行的方法是取染色體編號在前者，意味著假設目前適應值最高的染色體有編號為 1、2、3 的三條，則取編號 1 的染色體為目前的最佳解。儘管取編號在前的染色體為目前的最佳解，但我們還是無法確知究竟在三者中，誰可能在最短時間內再把適應值調升? 在問題 2 中，我們引入智力差距(即 R 值)的概念而產生群體記憶式調整，也就是與目前最佳適應值差距在 R 值以內的染色體，皆可不接受基因的捐贈，並自我進行調整的動作。

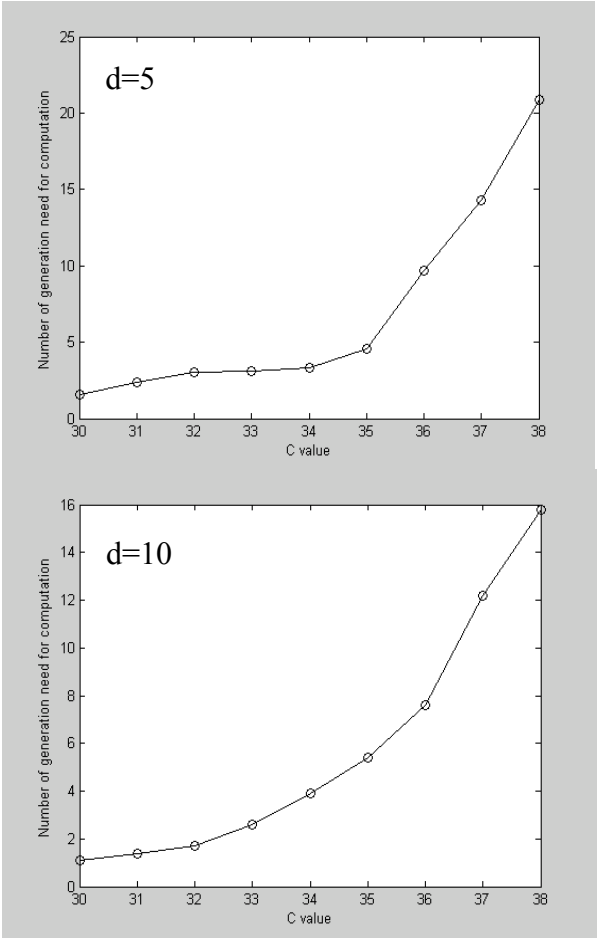
(五)合理的實驗次數(Reasonable solutions)

在評比的過程中，每條染色體的最高可能分數為 40 分，但在 95%的準確度下，本實驗的停止條件應設為 38 分，根據 Li 的實驗，求出 10000 組解之後，以統計方式找出前 50 個出現頻率最高的基因。但值得思考的是，是否一定要取得 10000 組解?9000、8000 組可以嗎?它們的前 50 個基因，是否變化不大?基於這樣的疑問，我們將實驗以每取得 1000 組解答為一分隔點，觀察前 50 個基因的名次是否有所變動，盼能找出可以節省時間成本且不失精確度的解答數目。

四、結果

上述方法的目的，希望能夠縮短遺傳演算法搜尋最佳解的時間成本，以利盡快取得實驗結果。基因捐贈搭配記憶式調整的兩種染色體長度比較列於下圖(圖八)。

另外，我們分析「10000 組解的數目」是否有降低的空間?本實驗將終止條件設為 38 分，以 10000 組解的前 50 個基因作為標準來觀察各時間點基因的出現頻率變化(表一)。根據表一的結果可以發現，當我們取得 1000 組解時，出現頻率最高的前 50 個基因與取得 10000 組時的相似度至少已達 86%。各時間點的前 40 個基因幾乎沒有太大的變動，真正有在改變的僅是 40-50 名的基因，這些基因的名次往往在 50 名的邊緣升降。以上這種情況也可以說明為何在 d=5 時，2000 組時的相似度是 96%，而 3000 組解的相似度卻僅有 94%的現象了。另一個觀察重點是，當可能解的組數到達 9000 組以上時，前 50 個出現頻率最高的基因已沒有變動，因此，可以嘗試將可能解的數目調整為 9000 組。



圖八、不同染色體長度的時間成本比較

d=5					
數量	相似度	相似度	相似度	相似度	相似度
10000	100%(50/50)	100%(40/40)	100%(30/30)	100%(20/20)	100%(10/10)
9000	100%(50/50)	98%(39/40)	100%(30/30)	100%(20/20)	100%(10/10)
8000	98%(49/50)	95%(38/40)	100%(30/30)	100%(20/20)	100%(10/10)
7000	98%(49/50)	95%(38/40)	97%(29/30)	100%(20/20)	100%(10/10)
6000	98%(49/50)	95%(38/40)	97%(29/30)	100%(20/20)	100%(10/10)
5000	96%(48/50)	95%(38/40)	93%(28/30)	95%(19/20)	100%(10/10)
4000	96%(48/50)	95%(38/40)	93%(28/30)	95%(19/20)	100%(10/10)
3000	94%(47/50)	98%(39/40)	93%(28/30)	95%(19/20)	90%(9/10)
2000	96%(48/50)	95%(38/40)	93%(28/30)	95%(19/20)	90%(9/10)
1000	92%(46/50)	95%(38/40)	93%(28/30)	90%(18/20)	90%(9/10)

d=10					
數量	相似度	相似度	相似度	相似度	相似度
10000	100%(50/50)	100%(40/40)	100%(30/30)	100%(20/20)	100%(10/10)
9000	100%(50/50)	100%(40/40)	97%(29/30)	100%(20/20)	100%(10/10)
8000	96%(48/50)	100%(40/40)	97%(29/30)	95%(19/20)	100%(10/10)
7000	94%(47/50)	100%(40/40)	97%(29/30)	100%(20/20)	100%(10/10)
6000	94%(47/50)	95%(38/40)	97%(29/30)	95%(19/20)	100%(10/10)
5000	94%(47/50)	98%(39/40)	97%(29/30)	90%(18/20)	100%(10/10)
4000	94%(47/50)	88%(36/40)	93%(28/30)	85%(17/20)	100%(10/10)
3000	90%(45/50)	93%(37/40)	97%(29/30)	85%(17/20)	100%(10/10)
2000	86%(43/50)	88%(36/40)	93%(28/30)	90%(18/20)	100%(10/10)
1000	86%(43/50)	83%(33/40)	93%(28/30)	90%(18/20)	90%(9/10)

表一、終止條件 38 的前 50 個基因變化情況

五、討論

Li 同樣以 GA/KNN 進行遺傳疾病的基因篩選，在 Li 的文章中較著重於基因挑選及預測的準確性，且對於染色體長度已經做了探討；但 GA/KNN 的方法在時間需求上是較高的,如何改善時間的需求是本文的研究重點。本文以嘗試在演算法、滿足條件的組數
等相關問題上討論，以求對整個 GA/KNN 的方法能夠更加詳盡。另外，本方法尚可點出以下幾個研究方向：

1. 不同的終止條件

本文針對 95%的準確度做了一系列的實驗,但在 85%或 75%是否也能取得相同的結果?若降低終止條件也能取得相同的結果，則更可減少運算時間。

2. 基因彼此的關聯性

對基因彼此的關聯性作分析,也許可以找出遺傳疾病與基因之間可能的關係。

3. K 值的改變

KNN 演算法中的 K 值若不同於本文中所設定的 3，而是 K>3，對結果會有何影響。

4. N-fold 分類方式

若只在 62 個樣本中取出 40 個樣本當作訓練樣本，所得到的結果是否會只對這 40 個樣本具有區別力，而對其他的樣本則不具區別力?N-fold 的方式可隨機抽樣多個樣本,並驗證各組結果的交集程度,以達到找出最有相關性的基因並供生物醫學研究。

在生物資訊的領域中經常遇見兩難的問題,因為大量資料運算所需要的成本,無論是運算時間或是記憶體空間的佔用都相當巨大。解決本領域的問題,可以從平行運算及演算法修改來著手。本文嘗試以數種方法來增進遺傳演算法的速度,進而求取在兼顧精確度的情況下,能使時間成本降至最低,以在短時間取得所需的資訊。

六、誌謝

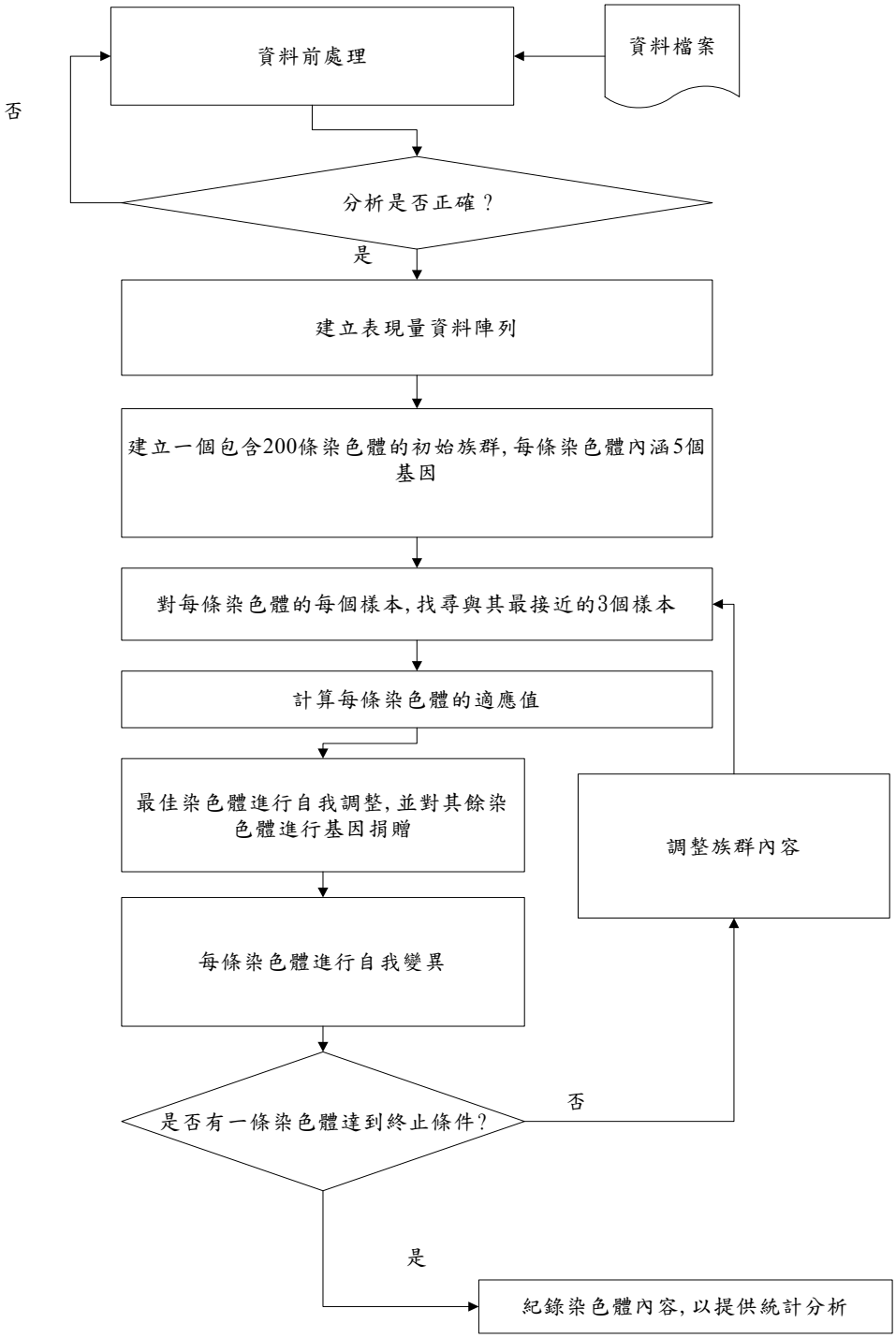
十分感謝長庚醫學研究計劃 CMRPD32002 與 CMRPD1008 的支持，提供我們此機會來研究此基因篩選方法。

七、參考文獻

[1] P. Baldi, G.W. Hatfied, DNA microarrays and gene expression, Cambridge University Press,

Cambridge, pp. 1-17, 2002.
[2] M. Ben-Bassat, Pattern recognition and reduction of dimensionality, Handbook of Statistics, North Holland, pp.773-791, 1982.
[3] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory, 13, pp.21-27, 1967.
[4] B. Dasarathy, Nearest neighbor (NN) norms: NN pattern classification techniques, Los Alamitos, CA: IEEE Computer Society Press, 1991.
[5] M. Dash, H. Liu, Feature selection for classification, Intelligent Data Analysis 1, pp.131-156, 1997.
[6] J. Devillers, Genetic algorithms in molecular modeling, Lyon, France, pp.38-42, 1996.
[7] J. Doak, An evaluation of feature selection methods and their application to computer security, Technical report, Davis, CA: University of California, Department of Computer Science, 1992.
[8] D.E. Goldberg, Genetic algorithm in search, optimization and machine learning, Addison-Wesley, p14, 1989.
[9] J. Han, M. Kamber, Data mining: concepts and techniques, Simon Fraser University, USA, pp.314-315, 2001.
[10] J. Holland, Adaptation in natural and artificial system , University of Michigan Press, Ann Arbor, MI, 1975.
[11] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.19, no2, pp.153-158, 1997.
[12] G. John, R. Kohavi and K. Pfleger, Irrelevant features and subset selection problem, Proceeding of the Eleventh International Conference on Machine Learning, pp.121-129, 1994.
[13] J. Kelly, L. Davis, A hybrid genetic algorithm for classification, Proceeding of the Twelfth International Joint Conference on Artificial Intelligence, pp.645-650, 1991.
[14] P. Langley, W. Iba, Average-case analysis of a nearest neighbor algorithm, Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chambery, France: Morgan Kaufmann, 1993.
[15] L. Li, Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, Bioinformatics, Vol.17, no.12, pp.1131-1142, 2001.
[16] C.B. Lucasius, G. Kateman, Understanding and using genetic algorithms, Chemon Intell Lab, Syst. 19, pp.1-33, 1993.
[17] Y. Lu and J. Han, Cancer classification using gene expression data, Information System 28, University of Illinois, pp.243-268, 2003.
[18] W. Punch, E. Goodman, M. Pei, C. Lai, P. Hovland and R. Enbody, Further research on feature selection and classification using genetic algorithm, Proceedings of fifth International Conference on Genetic Algorithms, pp.379-383, 1993.
[19] M. Raymer, W. Punch, E. Goodman, L. Kuhn, A. Jain, Dimensionality Reduction Using Genetic Algorithms, IEEE Transactions on Evolutionary Computation, vol. 4, no. 2, pp. 164-171, 2000.
[20] P. Robbins, The use of a variable length chromosome for permutation manipulation in genetic algorithm, Artificial Neural Net and Genetic Algorithms, Springer-Verlag, Wien, pp.144-147, 1995.
[21] C. Romualdi, S. Campanaro, D. Campagna, B. Celegato, N. Cannata, S. Toppo, G. Valle, G. Lanfranchi, Pattern recognition in gene expression profiling using DNA array: a comparative study of different statistical methods applied to cancer classification, Human Molecular Genetics, vol. 12, no. 8, pp.823-836, 2003.
[22] J.D. Schaffer, R.A. Caruana, L.J. Eshelman, A study of control parameters affecting online performance of genetic algorithm for function optimization. Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California, pp.51-60, 1989.
[23] J.D. Scaffer and L.J. Eshelman, On crossover as an evolutionarily viable strategy, Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California, pp.61-68, 1991.
[24] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems, Iowa State University, 1998.

八、附錄



附錄一. GA/KNN 流程圖