

中華民國 第 50 屆中小學科學展覽會
作品說明書

高中組 生活與應用科學科

佳作

040818

我自己會唱歌

學校名稱：國立花蓮高級中學

作者： 高一 詹易衡 高一 徐愷 高一 楊銘恩	指導老師： 溫健順 白欣蓉
----------------------------------	---------------------

關鍵詞：DataMining、自動作曲

摘要

利用資料採礦 Data mining 的方式，找出流行歌曲常出現的音律和節奏形式，統計後依照音律的出現比例分配權重，使數千個的音律照著統計結果串接，完成自動作曲。再寫出一程式，讓使用者選擇所要的調性、數量、伴奏類型和速度，做出旋律和伴奏，最後由程式處理過後，以 midi 的音樂格式在播放軟體上播出。

研究動機

音樂是表達人類情感的一種藝術，當聽到快樂或悲傷的音樂時，心情很難不隨著音調的起伏而跟著波動起來。大部分的人都喜愛聽音樂，但要自己寫曲實在是件難事，有沒有可能讓電腦自動譜出一首首優美的樂曲呢？

壹、 研究目的

- 一、 研究在歌曲資料庫中，是否存在常出現之音律
- 二、 利用既有的常出現音律，加上作曲的理論，使程式能有自動作曲能力
- 三、 電腦自動產生歌曲

貳、 研究設備及器材

電腦 Windows XP：

- 一、 編譯程式：Dev-C++
- 二、 作曲程式：Noteworthy Composer 2 (NWC2)
- 三、 作曲程式：Overture
- 四、 音樂播放器：Windows Media Player
- 五、 文字文件：word pad

參、研究過程或方法

一、概念介紹：

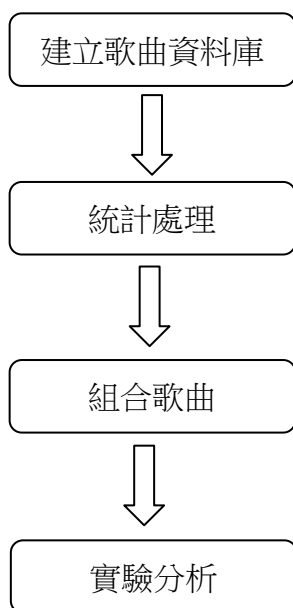
讓電腦自動作曲，應從作曲理論談起，構成一首歌所需的「**動機**」、「**樂句**」和「**樂段**」，希望由電腦自行產生。

音樂以「**動機**」為基礎，以「作曲」的技巧發展而成。「**動機**」是一個**樂段**的基本元素，決定歌曲的特色。**樂句**是旋律的組成部分，從動機發展，長短不一，由幾個小節組成，可從任一個和弦開始，結束於任一個和弦。**樂段**則是由數個**樂句**所組成。

在資訊課聽到資料採礦中群集化和推估等概念，聯想到應能應用至自動作曲，藉由資料採礦的理論及技巧找到作曲所需的元素。如果以職業作曲家寫出的歌曲當作資料庫，挑出上述的作曲要素，進行研究，應能完成期望的工作。

根據上述思維，訂出研究方向為：

首先假設現今流行樂曲中存在常常出現的音律，對流行樂曲進行分析及統計，試著找到常一起出現的音律(即為資料採礦中的 Frequency patterns)，並用程式搭配作曲的理論重新組合成新曲(圖 1)。



(圖 1) 概念圖

二、理論基礎

(一) 基本樂理簡介

一首歌曲包含旋律、節奏及和絃，並含有音名、調號、拍子等元素。

1. 音樂三要素：節奏、曲調、和聲。<參考 1>

(1) 節奏：

抑揚頓挫，由音之長短、強弱、快慢，組合而成。

(2) 曲調：

指音的連續進行，由音的高低組合而成。曲調沒有和聲，仍然可以構成音樂，但必須有節奏。

(3) 和聲：

兩個以上高低不同的音，同時或先後連續出現所產生的混合音。

2. 調號 (Key Signature)

意指音階具一定的高度時，第一音 (主音) 的位置，固定在某一音名上。例如，C 大調就是以 C (Do) 音當作主音的大音階調子。

3. 大小調 (Major, Minor)

調號大致分為大調和小調。

4. 模進

將動機、樂句做類比式的變化，會針對其樂曲形式模仿及加以變化，是作曲常用的技巧(圖 2)。

EX:  (圖 2) 模進

以上圖為例，第 2 小節的音均比第 1 小節高 1 度，產生類比式變化

(二) 資料採礦 Data Mining

資料採礦是為了在資料庫中發現有意義的模式或規則，以自動或半自動的方式，來勘查、分析大量資料所進行的流程。

換句話說，資料採礦所要處理的問題，是在龐大的資料庫中尋找出有價值的隱藏事件，並且加以分析。而其主要的功能在於從資料庫中獲取有意義的資訊以及對資料歸納出有結構的模式。

資料採礦的功能總共分為以下六個活動。每個活動目的，都是從資料中，取出有意義的新資訊。這些活動分別為：分類、推估、預測、同質分組(關聯規則)、群集化及描述視覺化。〈參考 2〉

研究希望在音樂資料庫中找到常出現的相同音律，進行統計及分析。這樣的概念，使用到 Data mining 的分類、推估及群集化的功能，詳細說明如下：

分類：按照分析對象的屬性分門別類加以定義，建立類組(class)
EX：將音樂資料庫分成 C 大調和 A 小調兩類。

推估：根據既有連續性數值之資料，以獲致某一之未知值。
EX：依照 pattern 與 threshold 關係推出洽當 theshold 值

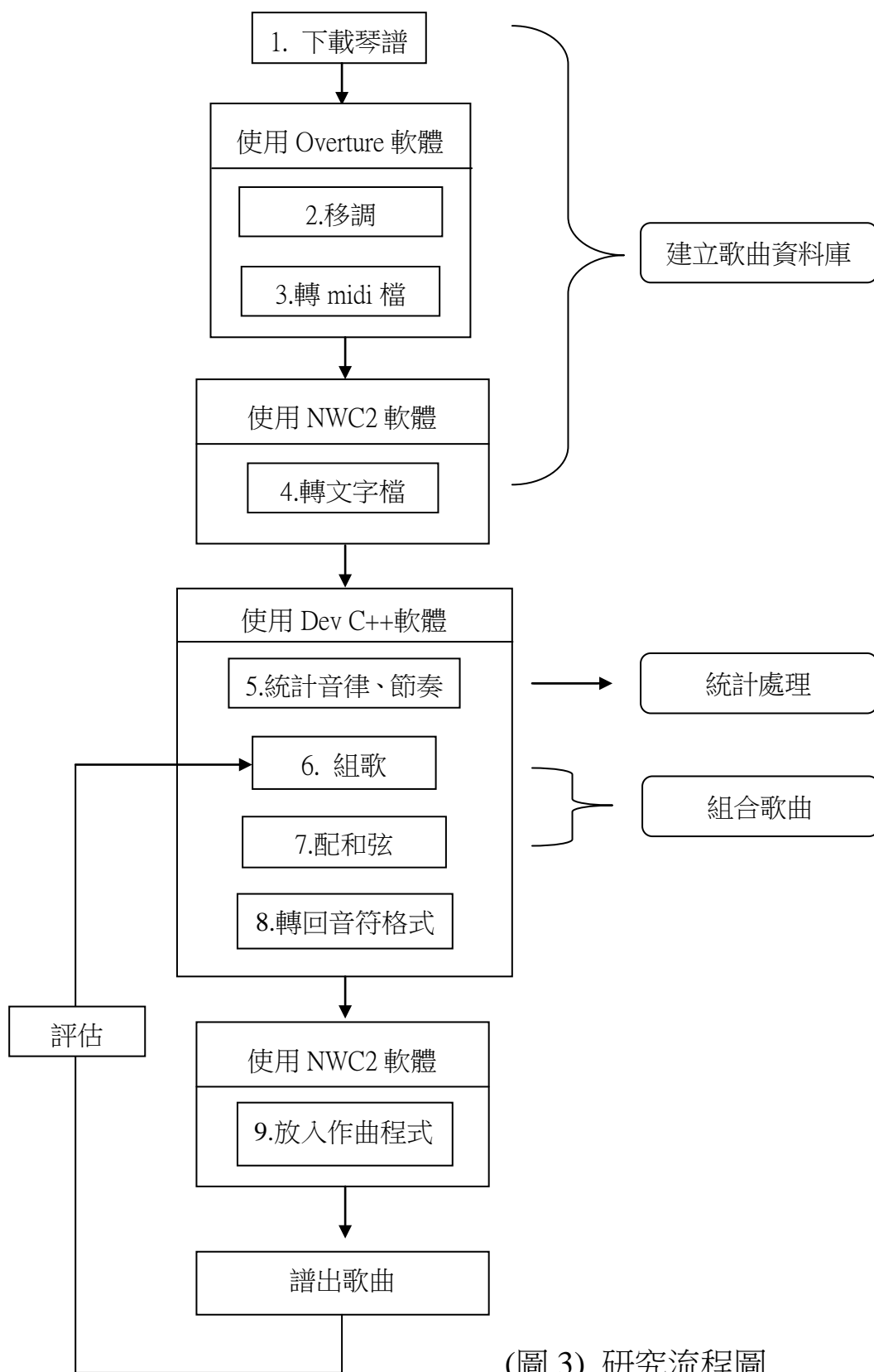
群集化：資料依據本身相似的特性自動群聚
EX:把相同 pattern 長度相互群集

(三) MIDI 簡介

電腦中音樂格式有 MIDI、MP3、WMA 等類型，副檔名為.mid 的音樂檔只記錄著一些類似音符的資訊，就像是一種樂譜。因此，MIDI 可以讓電子樂器設備互通訊息。〈參考 3〉

Data Mining 以文字做為分析，目前找到 NWC2 可將 MIDI 轉為文字檔，而 MIDI 檔記錄著要在甚麼時候使用第幾號樂器哪個頻率的音，顯然不會有歌詞，並且是絕對音準，這也是選擇 MIDI 檔的原因。雖然 MIDI 檔會因為各台電腦音源上的不同而有些微的差異，但這並不影響實驗。

三、研究過程：
流程圖(圖 3)



(圖 3) 研究流程圖

1. 下載琴譜：

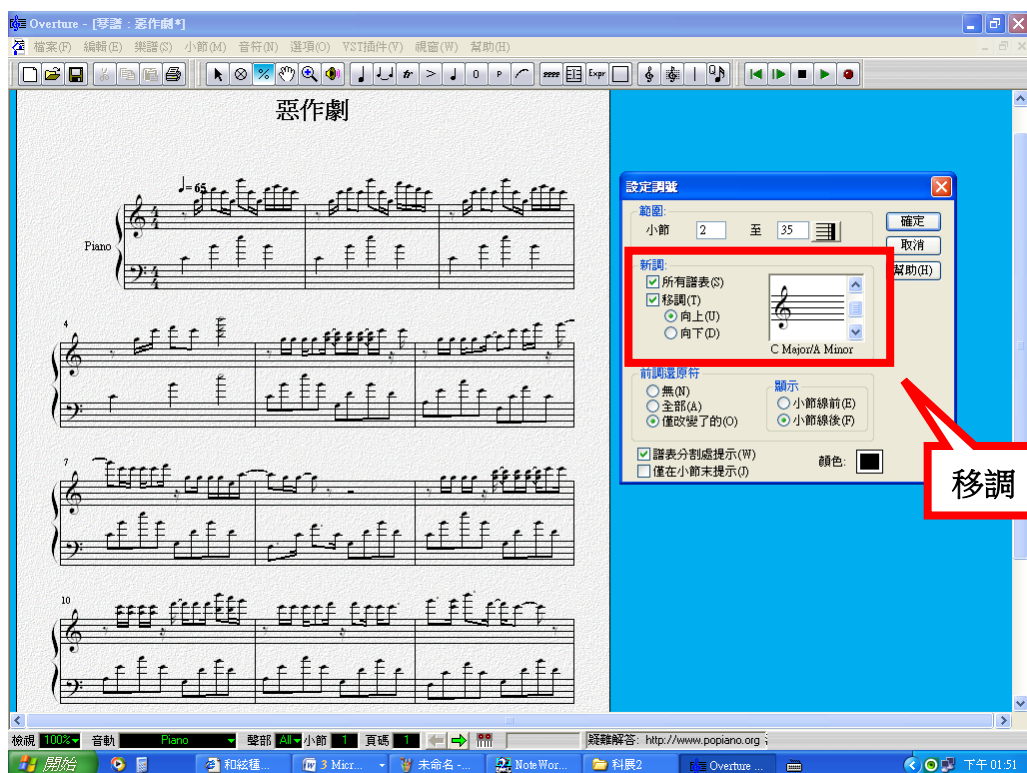
首先必須建一個龐大資料庫。於是先在網路上尋找現有琴譜下載，目前所建的資料庫共有 300 首的 C 大調的歌曲以及 37 首 A 小調歌曲。

<註>原本以 midi 檔作為資料庫的檔案類型，但發現會有太多音軌以致於不容易判斷主旋律。如果是琴譜檔，大多只有兩道音軌，方便作處理。

2. 移調：

一首歌可用不同調子表示，但只有音高上的不同，聽起來的感覺是一樣的。但統計時，如果不考慮調性，原本屬於同一個性質的音便會被當作不同的音看待，所以必須先將所有歌曲移成同調(正規化)。

利用 Overture 作曲程式，與手動方式判斷調性，將所有琴譜的調性移至 C 大調或 A 小調 藉此將格式統一(圖 4)。



(圖 4) 移調圖

3. 轉 midi 檔

因為目前無法以程式處理 OVE 檔，因此需要轉成文字檔，但 overture 也無法以文字檔輸出，所以必須先匯出 midi 檔，再由 Note Worthy Composer 2 做處理。

4. 轉文字檔：

以 Noteworthy Composer 2 作曲軟體匯入 midi 檔，並將譜上的主旋律複製，再以文字格式儲存(圖 5)。

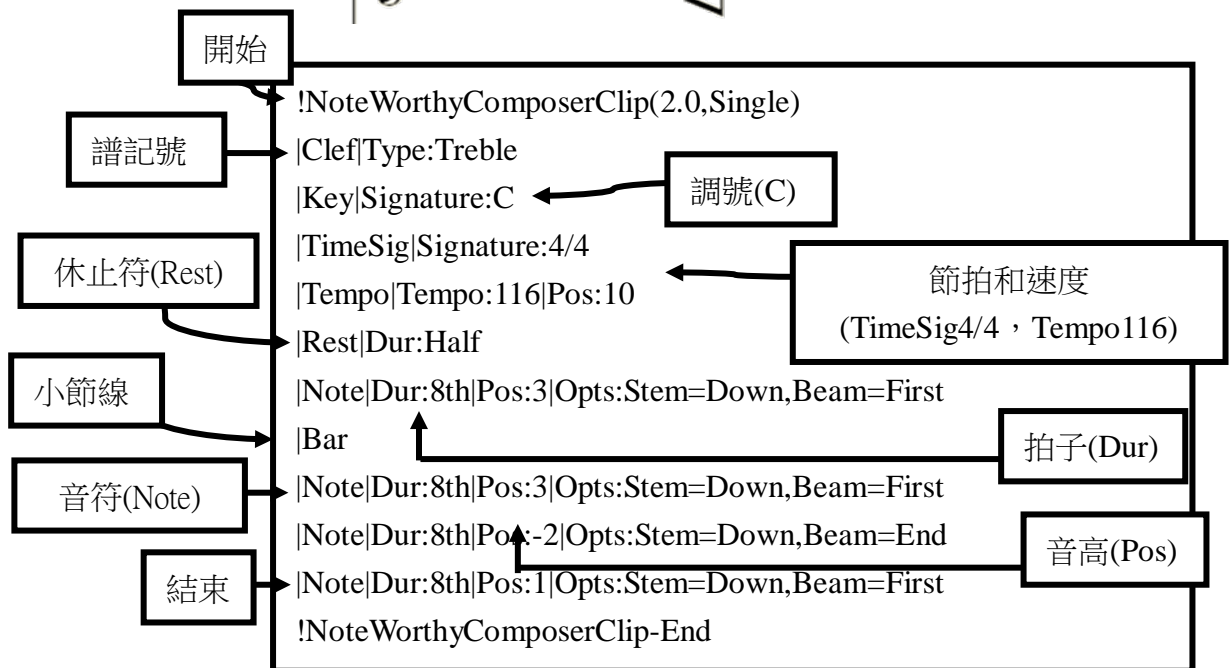


(圖 5) 主旋律和伴奏

Noteworthy Composer 2 的文字檔格式(圖 6、7)如下：



(圖 6) 範例樂曲



(圖 7) 範例文字檔

5. 統計音律、節奏

因為 Note Worthy Composer 2 的文字檔格式過於繁瑣不易作業，因此將轉成的文字檔以 read midi.exe(程式 1)簡化並儲存。

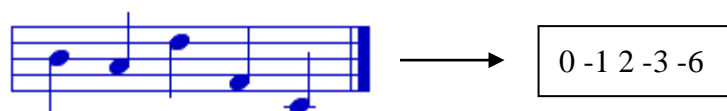
(1) 格式定義

A. 音高(Position)(表 1)：

符號	意義
i ($\forall i \in \mathbb{Z}$)	以高音 Do=1 為基準往上一音加 1，往下減 1。
^	延音，當有連結線時要將數個音符的拍數連在一起
#	升音記號，臨時升半音。
b	降音記號，臨時降半音。
n	還原記號，臨時升降後要還原成原本的音。
o	音桿連音的結束。
R	休止符
	小節線
50	一首曲子的結束

(表 1) 音高定義

EX:



(圖 8) 音高簡化

B. 節拍(Tempo)(表 2)：

為統一節奏的量，這次實驗中只取 4/4 拍子的歌做處理，休止符(Rest)和音符(Note)的拍量一同計算。

符號	意義
-50	小節線
50	一首曲子的結束
-3 ~ -96	休止符(節拍加負號)
3~96	音符節拍

(表 2) 節拍定義

原始定義(表 3)：

音符	自訂單位
全音符	32
2 分音符	16
4 分音符	8
8 分音符	4
16 分音符	2
32 分音符	1

(表 3) 原始定義

但三連音部分每個音會變為 8/3 而不易處理，因此將所有節拍定義均乘 3，使得三連音中每個音的節拍均為整數(如表 4)。

拍量	音符	NWC2 文字格式	自訂單位
4 拍	全音符	Whole	32x3 = 96
2 拍	二分音符	Half	16x3 = 48
1 拍	四分音符	4th	8x3 = 24
1 拍	三連音	4th	8x3 = 24
1/2 拍	八分音符	8th	4x3 = 12
1/4 拍	十六分音符	16th	2x3 = 6
1/8 拍	三十二分音符	32nd	1x3 = 3
附點	XX(上述)	XX, Dotted	上述*3/2

(表 4) 拍量轉換

因為每小節有 4 拍，故每小節節拍單位：96

(2) 簡化方式：將音律及節拍以上述方式換算輸出(圖 9)

```
!NoteWorthyComposerClip(2.0,Single)
|Clef|Type:Treble
|Key|Signature:C
|TimeSig|Signature:4/4
|Tempo|Tempo:116|Pos:10
|Rest|Dur:Half
|Note|Dur:8th|Pos:3|Opts:Stem=Down,Beam=First
|Bar
|Note|Dur:8th|Pos:3|Opts:Stem=Down,Beam=First
|Note|Dur:8th|Pos:-2|Opts:Stem=Down,Beam=End
```

音律：r 3 | 3 -2 1

節奏：-48 12 -50 12 12 12 50

音律文字檔存到 c1~c300,a1~a37
節拍文字檔存到 tc1~tc300,ta1~ta37

(圖 9) 範例文字檔

(3) 統計

將資料庫中的音律和節奏做統整的動作。特色在於，將音律和節奏拆開分別做處理，互不影響。應能避免和原本歌曲重複，組歌時再利用找到的 pattern 重新拼湊回去，變化性更大。

<註>在此以 C 大調的統計為例，A 小調比照辦理。

1) 音高

假設在曲子中能夠找到常常一起出現的音律，因此找尋 pattern 長度為 1~4 的音律。

單音：單個音符的出現比例

雙音：連續兩個音的出現比例

三音：連續三個音的出現比例

四音：連續四個音的出現比例

音高儲存方式：

統計音時以陣列記數，但陣列索引值恆 ≥ 0 ，且需包含升降音。

因此需找出音符的極值，整理後訂出：(如表 5)

音的種類	音高	處理	存取位址
負音	-19~-1	+20	陣列[1]~[19]
正音	0~ 20		陣列[20]~[40]
升音(包括正負)	#-19~#20	+60	陣列[41]~[80]

(表 5) 音高與陣列位置對照表

程式變數定義：(表 6)

n, m, p, q 為 1~80 的陣列索引值。

程式	變數	功能
1-C.exe	c[n]	音高為 n 的數量
2-C.exe	c[n][m]	Pattern 為 n-m 的數量
3-C.exe	c[n][m][p]	Pattern 為 n-m-p 的數量
4-C.exe	c[n][m][p][q]	Pattern 為 n-m-p-q 的數量

(表 6) 變數定義

在統計期間均使用此格式，雖然無法立即判斷音高，但方便電腦進行運算。(表 7)

Ex: 一段音符：5 6 6 5 1 6 6 5

Pattern 長度	音高組合	出現次數
單音	25 (5)	3
	26 (6)	4
雙音	25 26 (5 6)	2
	26 26 (6 6)	2
	26 25 (6 5)	2
三音	25 26 26 (5 6 6)	2
	26 26 25 (6 6 5)	2
	26 25 26 (6 5 6)	1
四音	25 26 26 25 (5 6 6 5)	2
	26 26 25 26 (6 6 5 6)	1
	26 25 26 26 (6 5 6 6)	1

(表 7) 範例組合

<註>小節線忽略，因為音律跨小節仍具有其連續性

音高統計過程：

程式執行順序(表 8)

功能	程式名稱	程式編號	讀入內容	讀出之內容
統計一首歌的單音	count1.exe	2	c1.txt~cn.txt	1-1.txt~1-n.txt
合併 1-1~1-n 的資料	1-C.exe	3	1-1.txt~1-n.txt	C-1.txt
統計一首歌的雙音	count2.exe	4	c1.txt~cn.txt	2-1.txt~2-n.txt
合併 2-1~2-n 的資料	2-C.exe	5	2-1.txt~2-n.txt	C-2.txt
統計一首歌的三音	count3.exe	6	c1.txt~cn.txt	3-1.txt~3-n.txt
合併 3-1~3-n 的資料	3-C.exe	7	3-1.txt~3-n.txt	C-3.txt
統計一首歌的四音	count4.exe	8	c1.txt~cn.txt	4-1.txt~4-n.txt
合併 4-1~4-n 的資料	4-C.exe	9	4-1.txt~4-n.txt	C-4.txt
合併 C-1~C-4 的資料	posC.exe	10	C-1 ~ C-4.txt	posC.txt

(表 8) 資料處理

2) 節奏：

程式變數定義：(表 9)

變數	功能
m[k]	第 k 小節之節奏，將 tc1~tcn 所有小節記錄於此

(表 9) 變數定義

目的	讀入內容	程式名稱	程式編號	讀出之內容
統計節奏	tc1.txt~tcn.txt	tempoC.exe	11	tempoC.txt

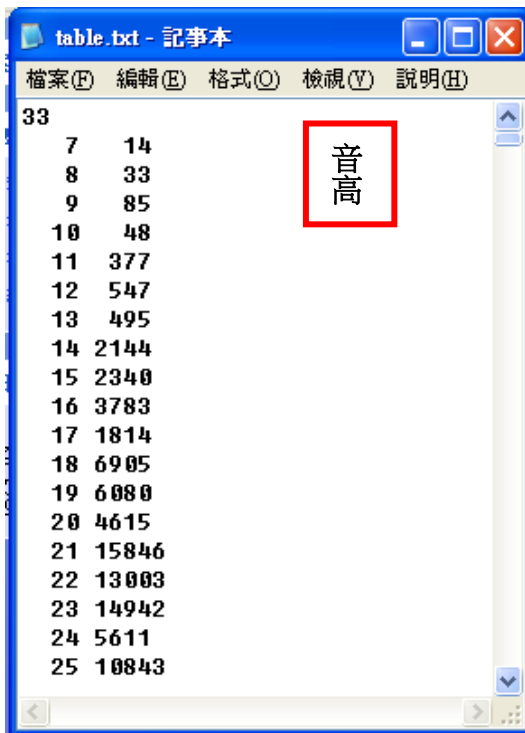
(表 10) 資料統計

3) 合併：

為了加快讀檔速度以及門檻值(Threshold)的制定

目的	讀入內容	撰寫之程式	讀出之內容
合併音高和節奏 門檻值的制定	posC.txt tempoC.txt	pos_tempo_thC.exe	tableC.txt

(表 11) 資料處理



(圖 10) tableC 音高部分



(圖 11) tableC 節奏部分

6. 組歌：

以 cost 程式將 tableC.txt 中的所有 pattern 存入。

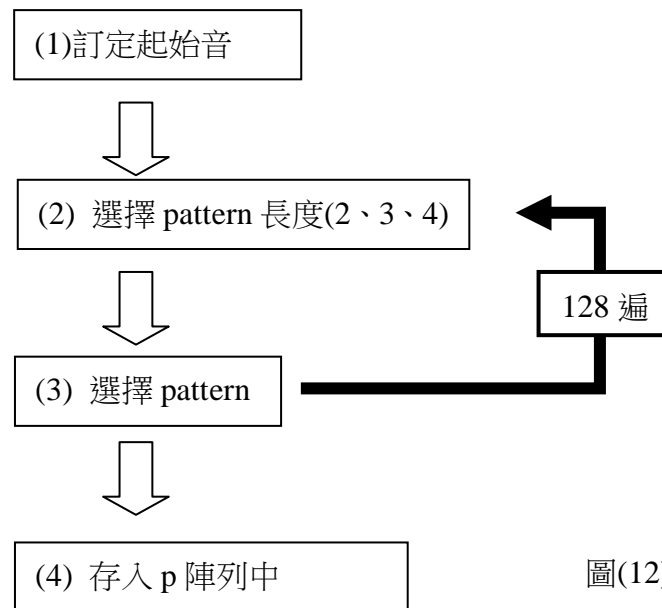
為了便於模進，使歌曲更為完整，因此寫出了三種函式以利使用。

函式必須承接一個或數個值，也就是參數(parameter)，在下列函式中的值皆為任意給訂。

變數 (value)	功能 (function)
pi 陣列 ($\forall i \in N$)	第 i 個主旋律
t ($\forall n \in N$)	產生 t 小節的拍子
Tl陣列 ($\forall l \in N$)	第 l 個節奏
k ($\forall k \in N$)	結尾的類型
q ($\forall q \in Z$)	模進之加減音程

(表 12) 變數定義

1) 旋律函式 Pos(int p1[])：



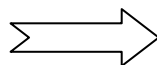
圖(12) 旋律函式流程圖

(1) C 大調以 C 和弦的組成音取亂數為起音，
A 小調以 Am 和弦的組成音取亂數為起音
ex: 假設選到 21(Do)，以 21 為起始音

(2) 以亂數決定要選擇 pattern 長度(2、3、4)。
ex: 假設選到雙音 pattern

(3) 以權重選擇 pattern

起音	尾音	數量(次)
21	29	11
21	30	16
21	32	12
22	14	17
22	15	23
22	16	17



起音	尾音	累計	範圍
21	29	11	0~10
21	30	27	11~26
21	32	39	27~38
22	14	17	0~16
22	15	40	17~39
22	16	57	40~56

(圖 13) 範例 tableC

(圖 14) 範例 tableC 權重

以上圖為例，21 只有接到 29、30、32 這三種候選音，以候選音在資料庫中的數量當作權重，並分配其出現機會。將亂數範圍定為 0~所有候選音的次數和-1，也就是(11+16+12)-1=38。

(註：因為包括 0，所以總量要減 1。)

假設取到 15，表示落在範圍 11~26 間，pattern 為 21 30

重複動作 128 遍：

以尾音當作下一個起音繼續連接 pattern

(註：重複 128 遍是因為無法預料整首歌所需的音，最大值為 16(音符)X16(小節)/2(最短 pattern)=128)

(4) 將製造後的主旋律存入 pi 陣列，待以後使用。

Ex: Pos (p1) → 建立 128 組 pattern 資料，存入 p1 陣列。

2) 節奏函式 Tem(int t, intT1[]):

讀入 table.txt 中的節奏，並以隨機方式選取 t 小節的節奏，存入 T1 陣列中。

Ex: Tem(2, T3) → 建立兩小節的節奏，存入 T3 陣列。

3) 結尾函式 tail(int k):

分成樂句的結束(k=0)和樂曲的結束(k=1)

K 值	處理
0	表示為一個樂句的結束，而樂句的尾音，應為一個拍子較長的音而較有結束的感覺，所以將原本產生的旋律尾段修成兩拍(以上)的音符。
1	將 C 大調音律的結束音強制回到 Do(21)、A 小調為 La(19)

(表 13) 尾音處理

4) 組曲 make(int t, int T1[], int pi, int q)

將 pi 與 t 個小節的 T1 搭配，為了有模進的效果，將 pi 中的每個音加 q，並分別存入 position 及 tempo 陣列中，讀出 s1~sn(使用者自訂幾首)。

Ex: make(2, T3, p1, -1) → 將 t3 節奏與 p1 中每個音加(-1)搭配，存入 position 及 tempo 陣列中

7. 配和弦：

只有主旋律稍嫌不足，因此配上和弦，能掩飾一些主旋律的不順暢，並增加音樂的厚度。

功能	使用程式	讀入文字檔	輸出文字檔
找出重音判斷和弦	配和弦.exe	s1~sn	ch-1~ch-n
將音名轉成數字	chord.exe	ch-1~ch-n	chord-1~chord-n

(表 14) 和弦處理

(1) 讀入已產生的主旋律 (配和弦)

樂曲理論中，重拍上的音為決定和弦的重要判斷條件，重拍為第一和第三拍，用程式主動配上和弦。

和弦樣式：(表 15)

C 大調	A 小調
主要和弦：	
C : 1、3、5	Am: 6、1、3
F : 4、6、1	Dm: 2、4、6
G : 5、7、2	E : 3、5#、7
G7 : 5、7、2、4	E7 : 3、5#、7、2
次要和弦：	
Dm : 2、4、6	F : 4、6、1
Em : 3、5、7	C : 1、3、5
Am : 6、1、3	G : 5、7、2

(表 15) 和弦樣式

配法：

在一個小節內有兩個重音，依照找出的重音屬於哪個和弦

情況一：兩個重音都屬同一個唯一的和弦→使用其和弦

例：重音為 1 和 4，只屬 F 和弦

情況二：兩個重音都屬同一個「不」唯一和弦→使用較常見的和弦

例：重音為 1 和 3，屬 C 和 Am，但 C 較常見，選之。

情況三：兩個重音屬不同和弦→選兩個各以重音為主音的和弦

例：重音為 1 和 7，各屬 C 和 G7

情況四：重音上為休止符

例：以休止符後第一個音為主音。

(2) 將和弦轉成數字，對照如下：(表 16)

C 大調		A 小調	
C	1	C	1
Dm	2	Dm	2
Em	3	E	3
F	4	F	4
G	5	G	5
Am	6	Am	6
G7	7	E7	7

(表 16) 和弦轉換

8. 轉回音符格式

執行後將得到一串音和節奏，但仍然是自定格式，所以要用 wtomusic.exe 讀入主旋律，wtovarychord.exe 讀入和弦，換回 Note Worthy Composer 的格式(圖 14、15)。

使用程式	功能	讀入文字檔	輸出文字檔
wtomusic.exe	主旋律轉換	s1~sn.txt	music.txt
wtovarychord.exe	和弦伴奏轉換	chord-1~chord-n	Chord_vary.txt

(表 17) 資料處理

(1) wtomusic.exe

依照對應之格式一一轉換回。處理完一首曲後以一全拍休止符代表一曲子的結束。

(2) wtovarychord.exe

為使和弦的節奏產生變化，且讓使用者決定喜歡的節奏類型，因此列出較普遍的種類以便使用。

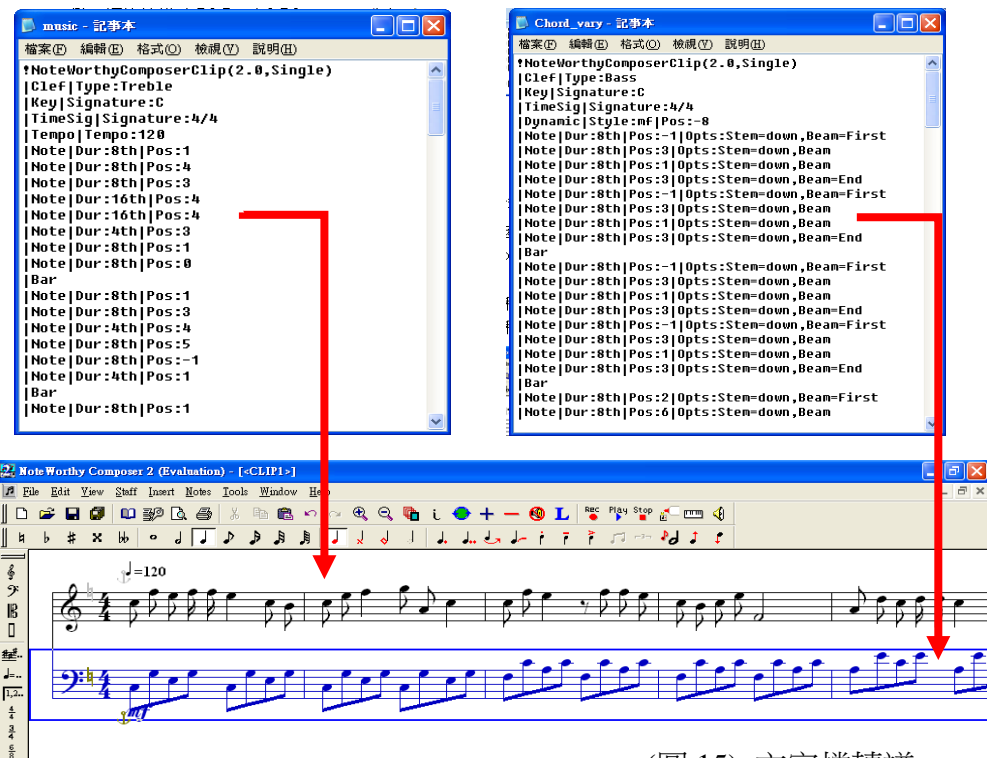
基本型：全音符之三和弦和半音符之三和弦(圖)

變奏型：不同風格之變化節奏

讓使用者選擇所喜歡的節奏風格，在歌曲製造出後產生。

9. 放入作曲程式：

曲子產生後仍為文件檔，所以必須手動的把 music.txt 及 Chord_vary.txt 的內容全選後貼上 Note worthy Composer 2 的軟體上，再按下播放鍵便完成了一首歌的播出！(圖 16、17)



(圖 15) 文字檔轉譜

10. 最終程式：

原先做出歌曲後還必須手動把文件檔中的資料貼至 NWC2，不夠人性化，造成使用者的麻煩，所以決定要寫出能自動轉換成 midi 檔的程式。透過老師的協助找到了一個能將文字文件轉換成 midi 檔的程式。MIDI 文件格式：

格式	起始時間(秒)	持續時間(秒)	音高(1~127)	音量(1~127)
例	0.0000	0.1849	67	100

唯音高部分較為複雜，此格式是以 48 為基準的中音 Do，每增加一個數字增加的是一度音，也就是包括了升音，和我們以往使用的格式不一樣，因此寫出了公式將以前的數值轉換。

音名	C	#C	D	#D	E	F	#F	G	#G	A	#A	B
唱名	Do	#Do	Re	#Re	Mi	Fa	#Fa	Sol	#Sol	La	#La	Si
自訂格式 X	1	11	2	12	3	4	14	5	15	6	16	7
代碼 Y	48	49	50	51	52	53	54	55	56	57	58	59

m[]：跟 Do 差幾度，EX: 2(Re)跟 Do 差 m[2]度

轉換公式：

If $X < 10$, $Y = 48 + ((X+13)/7-2)*12 + m[(X+13)\%7]$

Else , $Y = 48 + ((X+3)/7-2)*12 + m[(X+3)\%7] + 1$

例如：中音 Do (1) ==> 48

介面更加人性化

為使選取界面美觀，利用 Visual C++ 2008 寫一個程式，並和所有組歌的程式合併在一起，比較接近一個正式的軟體了。優點是這個程式檔案很小，不需要再多一個作曲軟體輔助，傳輸及使用方便。

肆、 研究結果：

一、可自動作出所指定條件之歌曲

條件：

1. 調性
2. 歌曲數
3. 歌曲快慢
4. 伴奏類型
5. 樂器種類

二、給定門檻值和連音的配法會有不同感覺的歌曲產生。

三、流行歌曲應有常出現的音律，雖種類繁多，但仍有明顯的比例分配。

四、使用成品方法

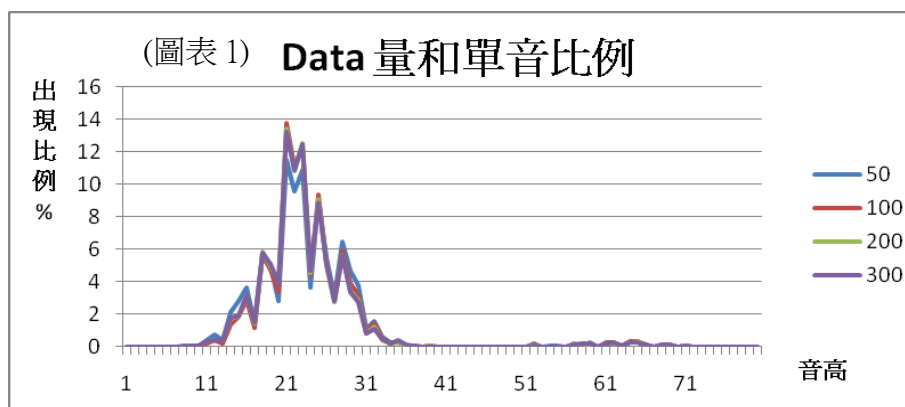
1. 點取“自動作曲”
2. 選取 C 大調或 A 小調
3. 輸入想要產生的歌數和速度
4. 選擇喜歡的和弦類型和樂器
5. 按下確定鍵
6. 享受美妙旋律

伍、 討論

實驗和比較

1. Data 量和音占有比例的關係(圖表 1)

音的使用頻度和資料庫內容密不可分，因此通常在建立資料庫會嚴格要求資料量要大於一定的值。此實驗要分析 Data 量的多寡會如何影響單音的出現比例。



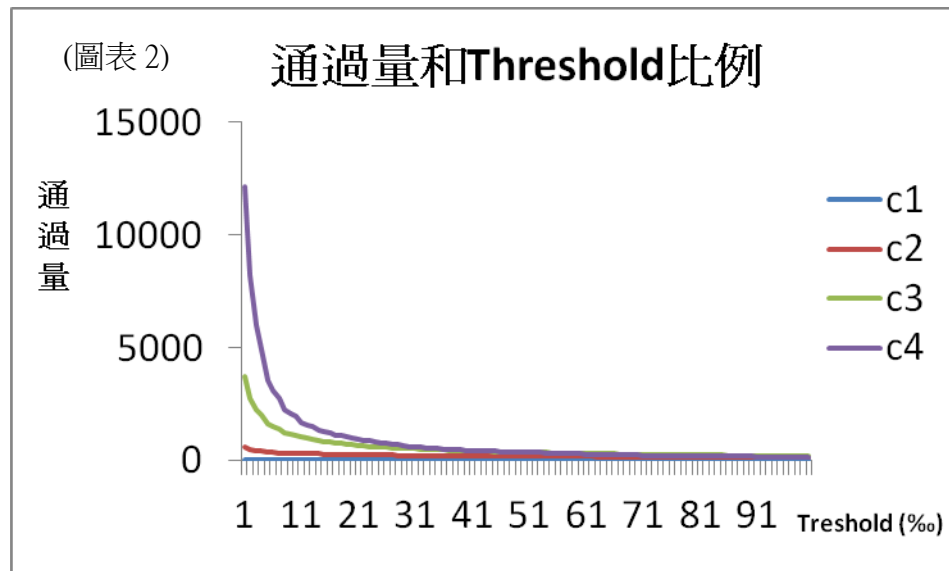
Data 量 50 首歌和其他三個較容易看出明顯差距，可見 Data 量必須要在高。來到兩百時，幾乎和 300 完全重疊，顯示到此 Data 量時應不太會對音的組成比例造成很大的影響了。

2. Threshold 的制定

數據整理後發現，有些 pattern 是太低音或太高音，甚至大跳，以致於取出的旋律非常不穩定。他們出現的原因可能是前奏的音位在高八度，而突然轉成主旋律時便跳下來。

這些 pattern 的數量通常都較少，若訂出一個 Threshold(門檻值)，即能將此次數過低的 pattern 排除，排除後便能使旋律較為平順。

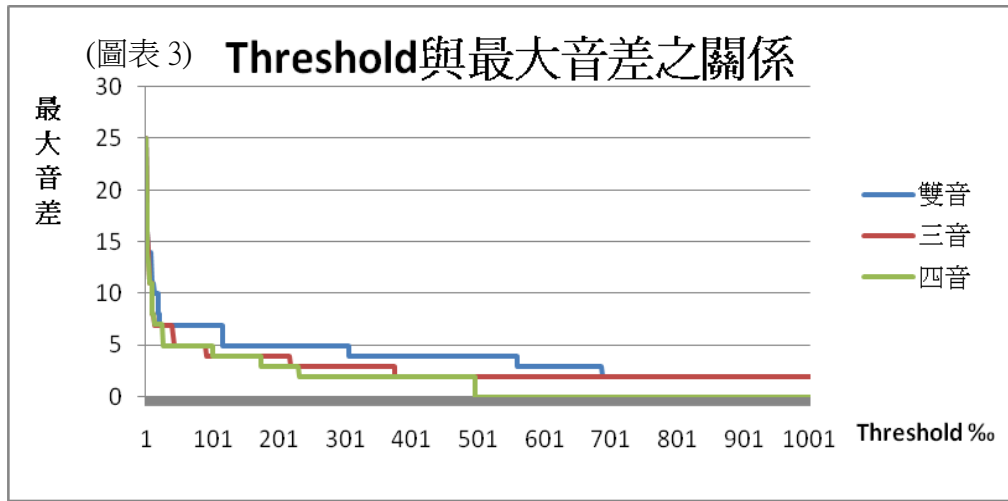
1) Threshold 和各長度 pattern 數的關係(圖表 2)



由上列圖表顯示，threshold 的遞增，通過的 pattern 量隨之遞減，屬於嚴格遞減之圖形，而其切線斜率由大至小，意即 Threshold 與通過量的變化率隨 Threshold 漸減，且 pattern 長度越高者愈明顯。

若 threshold 稍高，pattern 通過量與原量大相逕庭，最大音差也隨之降低，使得歌曲的變化性變小；而不訂 threshold，則會有最大音差 25 的情況，變化雖多，卻易使音律跑掉。

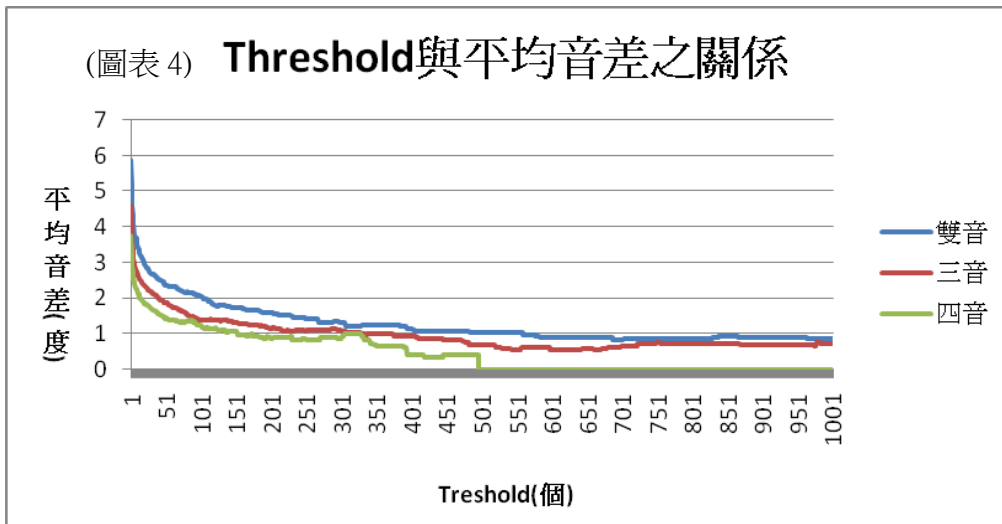
2) Threshold 與最大音差之關係(圖表 3)



上圖中顯示，當 Threshold 越高，而最大音差越小，意即在有 Threshold 的情況下，旋律將不會有大起大落之現象。

3) 平均音差和連音種類(圖表 4)

紀錄所有 pattern 頭尾音之差求平均，即可得到一種連音的平均音差。



由上圖可知，pattern 長度越短，平均音差恆大於較長之 pattern，意即較不穩定，音程容易大跳。

整理不同 threshold 後結果如下：

(1) 訂低(10~20)的 Threshold

- A. 音域穩定
- B. 變化較少
- C. 曲調順暢
- D. 風險較低

(2) 訂高(>100)的 Threshold

- A. 音域極窄
- B. 變化貧乏
- C. 容易重複同一個 pattern
- D. 不成調

(3) 不訂(0)Threshold

- A. 音域豐富
- B. 變化多
- C. 曲調起伏大
- D. 風險高，可能持續高(低)音，或大起大落

最後，決定將 Threshold 定為 10，所組出的曲目變化多且不會大起大落。

依上述結果，組歌時所使用的 pattern 長度會影響音律的起伏，所以依照不同的長度組合實驗，以找出最佳配對。

分成：

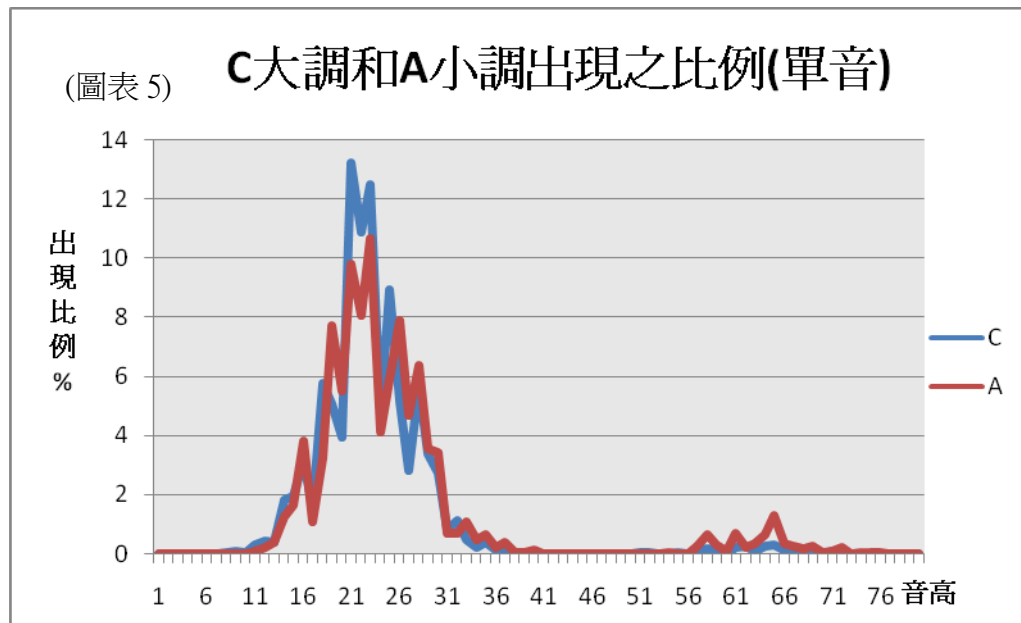
組合(combine)	描述(description)	使用程式(using program)
2-C	單組	cost-2.exe
3-C	單組	cost-3.exe
4-C	單組	cost-4.exe
2-C、3-C	二音和三音	cost-23.exe
3-C、4-C	三音和四音	cost-34.exe
2-C、4-C	二音和四音	cost-24.exe
2-C、3-C、4-C	全部	cost-234.exe

(表 18) 組合配對

研究發現，cost-4.exe 效果最佳，因其音域起伏適中，且仍具有變化性，而只要與 pattern 長度為 2 者搭配，就容易偏高或偏低，而且可能卡在某個音域，無法與下個音連接。-

3. 調性與音的比例(圖表 5)

計算 C 大調和 A 小調中每個音出現之比例，比較不同樂性之樂曲是否會出現不同比例的音。



由此圖表得知，不同樂性之樂曲會有不同的音出現。

整理後得出：

- I. A 小調的升音明顯比 C 大調多
- II. C 大調的 Do、Re、Sol 明顯較多
- III. A 小調的 La 明顯較多

推論：C 大調的 C 和 G7 和弦與 A 小調的 Am 和弦較常出現

探討：

- 一、原先 threshold 的制定上是在 mining 的過程中執行，是為了找到常出現的 pattern，但發現通過量太低，做出歌曲常常連續為同一個 pattern。因此必須在統計完後才訂定 threshold。因此主軸改成用權重比例，依照其出現頻度分配其串接的機率，也能達到相同的效果。而最後 threshold 只是做為刪除奇異 pattern 之用

二、 實驗發現 Threshold 與音域的廣度有很大的關係，有 Threshold 者音域較穩定，但變化性卻不足，而無 Threshold 者，音域較廣，變化較大，但容易有大起或大落，導致和諧性下降，因此，討論後決定將 Threshold 制定，但訂較低，使其達到兩者優點。在組曲方面，pattern 長度有 2~4 可以使用，討論後決定使用 pattern 長度 3、4 兩組，因為只要 pattern 長度有 2 就會不穩定，變化性過強，導致和諧性低。

三、 不論 threshold 大小，pattern 長度越長者，平均音差越小，意即不可使用過短的 pattern 組合，因此決定選擇 pattern 較長的配對。

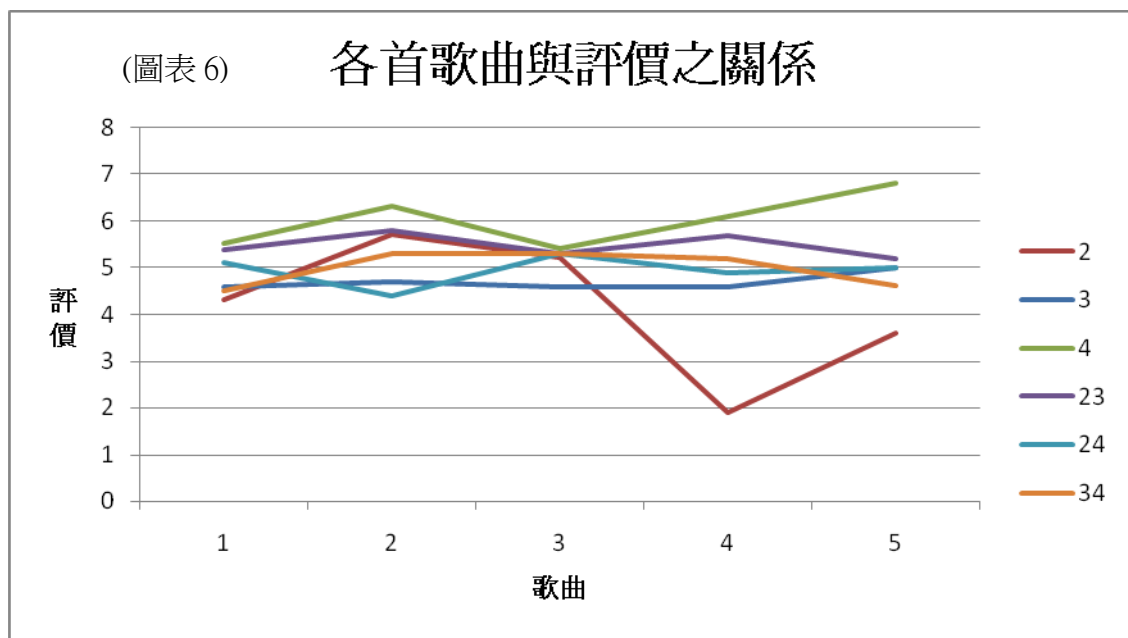
四、 初步分析完成之作曲

歌曲的好壞是因人而異，要比較出不同 pattern 做出之歌曲也不能以數據認定，因此對音樂班同學做了一個實驗，統計他們對於自動做出歌曲的評價。

實驗方法：

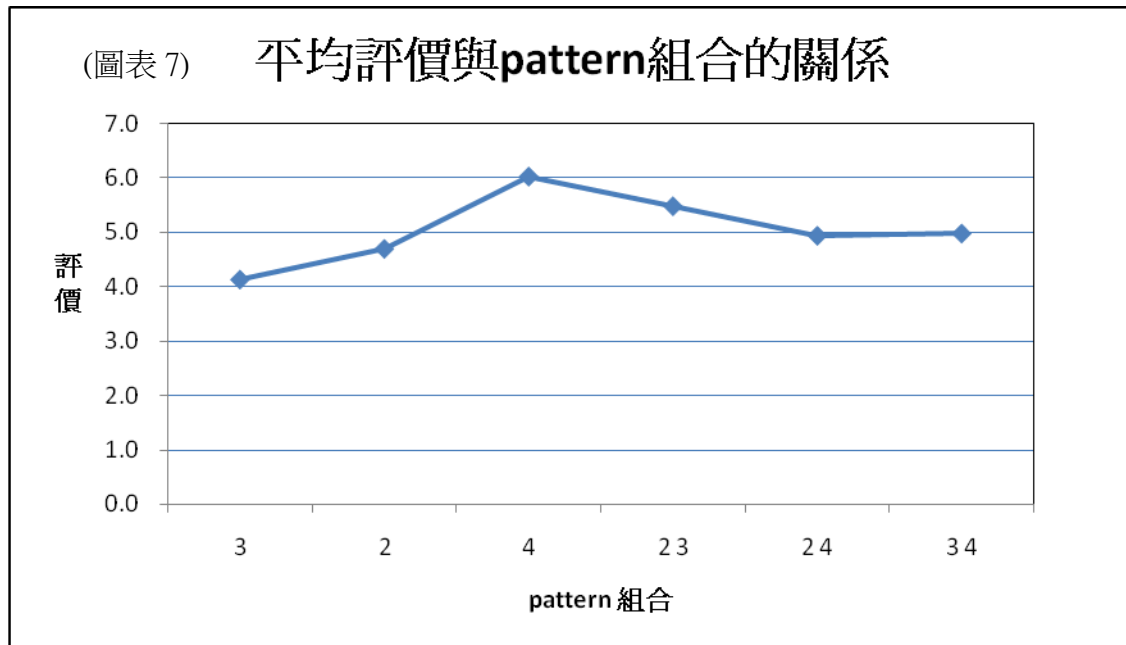
對每一種 pattern 類型各產生的五首歌，依照自己的感覺為歌曲評比 1~10 分

下圖 6 顯示，4 音及 2 音的歌曲評價差異較大，且兩種 pattern 長度組合的歌去評價較為平均。



利用 D&C 的方式，在 pattern 組合中挑出一首平均評價之歌曲，代表此 pattern 組合，並其播給受試者，比較大眾對不同 pattern 組合的看法。

下圖 7 顯示，4-pattern 的歌曲最為被大眾所接受，因此本組使用其組合方式。



五、 未來展望：

1. Data base 擴充：

- 甲、 曲風：中國，爵士等樂曲，研究是否能產生同類型的歌。
- 乙、 人物：當紅歌手歌曲分析，研究是否能產生個人特色之音樂。
- 丙、 TOP 金曲：歷年百大金曲，研究是否能譜出同樣流行之歌曲

2. 加入更多伴奏音軌：

- 甲、 打擊樂器
- 乙、 管弦樂器

3. 介面人性化：

- 甲、 輸入條件：可簡易/複雜輸入
- 乙、 美化介面：可搭配滑鼠使用

4. 程式鏈結：

不必手動複製貼上

5. 音樂評斷：

音樂的好壞是非常主觀的，無法量化或讓電腦決定，因此將以做問卷調查的方式，評定此程式做出之歌曲，是否與使用者之期待相符。

6. 應用至跨平台：

能應用至手機鈴聲，每次都能做出一首新曲。

陸、 結論

- 一、利用主旋律分割，可統計資料庫中所有音、pattern 以及節奏出現的次數及比例
- 二、將所有 pattern 依 Threshold 較小的方式進行權重組曲並搭配節奏組成一首歌
- 三、依照資料庫的形態，可做出 C 大調及 A 小調的樂曲
- 四、可依使用者所好決定速度之快慢
- 五、以 Data mining 的方式，並利用 pattern 組合的方式自動作曲應該是相當可行的，因為連接的音符是其他作曲家所用過的，且出現比率越高者，應是大眾所常使用且流行的。

柒、參考資料及其他

參考 1： http://www.cycschool.com.tw/new_page_74.htm

參考 2： <http://blog.xuite.net/sinner66/blog/5340022>

一、 網站資料

1. 流行鋼琴網： <http://www.popiano.org/big5/piano/>

2. 廷廷的鋼琴窩： <http://www.tintinpiano.com/>

3. ShellExecute 的用法：

<http://www.dev.idv.tw/mediawiki/index.php/ShellExecute%E7%9A%84%E7%94%A8%E6%B3%95>

4. 北一女科展： <http://nas.fg.tp.edu.tw/science/93/AA04.doc>

5. Songsmith網

址： <http://research.microsoft.com/en-us/um/redmond/projects/songsmith/index.html>

二、文獻參考

1. 中文：

i. 簡易作曲方法

作者：周惠萍，信宏出版社

ii. 實用歌曲作法

作者：李永康，全音樂譜出版社

iii. C&C++學習範本第二版

作者：吳明哲、黃世陽、林義証、蔡文龍，松崗出版社

2. 英文：

i. Data Mining: Concepts and Techniques

作者：Jiawei Han, Micheline Kamber，morgan kaufmann publisher 出版

【評語】 040818

應用作曲軟體，並從樂曲資料中整理出樂曲、曲調、節奏、和聲等類別，搭配作曲的理論，自行組合成新曲，是一個有趣的作品。只是本作品名稱「我自己會唱歌」應改為「我自己會作曲」，才比較符合作品的內容，因為作品中並沒有產生歌詞部分。