三、序列比對

3.1 比對方法

由於熱效應，很難控制到讓每次通過的核苷酸只有一顆，因此，便需要一套比對的方法，能夠在核苷酸分子鏈穿透過程中，有遺失訊號的情況下，所得到不完整的序列，透過蒐集整理這些不完整的序列，也就是利用這些不完整的資訊，將其拼湊成為完整的訊息，即得到完整的序列。

我們假設已得知整個核苷酸分子鏈的長度 N，且選擇模擬所得的序列長度為 n(n ≦N)，因此遺失的核苷酸個數為 $N-n=N_{missing}$，並且將這 N 個遺失的核苷酸以 B 的符號隨機插入得到的不完整序列中，以之前的序列為例，N=100，n=77，$N_{missing}$=23，遺失率 $R_{missing}$=23%即如下所示:

TCATBTTBTCTGABGATTGCGCGBACBTAGCTBCTGGATABGCAGGTGABTTACBCCBB

CABABATBCBGATGCBBBGBBTTBCATGCGBGBTTAACCAA

對於其他獲得的不完整序列，重複上述步驟，則可將他們依序排成如下:

第 第
一 二……
行 行

序列 1. ……B……B……B……B……

序列 2. …B…B……B……B……

序列 3. ……B……B……B……B……

. ……

. ……

透過移動 B 的位置，可使每一行的核苷酸滿足下列規則:

1. 同一行中，不會出現兩種以上的核苷酸。

2. 同一行中，可全為 B。

接著，再將每一行的核苷酸種類或 B 依序紀錄下，即為比對後的序列。比對的組數愈少，出現的簡併組數愈多，隨著比對的組數增加，出現的簡併組數會減少，最後只剩下一組，而這組序列便是最後定序出的序列。

在得到的不完整序列當中，當然也會有部分是核苷酸在奈米洞口停留超過一個快、慢頻切換週期，當這種情形發生時，就會使得取到的不完整序列中，產生多餘的訊號，而這多餘的訊號被轉譯成核苷酸，即原本序列中，假設有 3 個連續的核苷酸 AAA，會變成 3 個以上，這會造成比對中出現下列情形:

1. 無論如何，都無法有滿足前述規則的序列被比對出來。

2. 有滿足前述規則的序列被比對出來，但被比對出來的序列無法保證其正確性。

針對第二點，可以在相同組數的情況下，多次比對不同的序列，

比較比對出的序列，選擇序列出現機率較大者，當作定序出的結果。如此，可以將不正確的序列剔除，但是，這樣的前提是，在所有被比對的序列中，有多餘訊號的序列數不可過多，否則，仍會導致定序出的序列出現錯誤。

3.2 比對的演算法

上述方法的實作，我們選擇了利用蒙地卡羅法，其中最主要的目的，是要使抽樣的效率增加，使得計算速度得以加快，以便更快得到結果。

根據前述的規則，我們可以設定得分規則如下:

1. 每兩對中，只要同一行的核苷酸序列不一致，便扣分。

2. 同一行中，B 不影響得分。

然後對每一次比對前，改變 B 的位置，再計算其分數，而這分數便是蒙地卡羅法中，一般稱為能量的部分，很明顯地，當得分為零時，比對出來的序列，便是本方法定出的序列了。

舉例來說，假設我們有下列三組序列，每一組有 13 個核苷酸，配對不成功便扣一分:

1. ACBGTTCBCGAAA

2. CGBBBTCBBGTBA

3. ABGTBABBCBTAA

則計算分數時，先兩兩分配成 1-2、1-3、2-3，然後將所有分數相加，製表如下:

	1-2	1-3	2-3	score
Col.1	-1	0	-1	-2
Col.2	-1	0	0	-1
Col.3	0	0	0	0
Col.4	0	0	0	0
Col.5	0	0	0	0
Col.6	0	-1	-1	-2
Col.7	0	0	0	0
Col.8	0	0	0	0
Col.9	0	0	0	0
Col.10	0	0	0	0
Col.11	-1	-1	0	-2
Col.12	0	0	0	0
Col.13	0	0	0	0
總得分				-7

因此總得分為-7 分，這樣便完成一次嘗試比對，並且在負分時，不輸出比對的結果。接著若移動 B 可使得上述三組的排列組合如下:

1. ACGTBTCCGBAAA

2. BCGBBTCBGTBBA

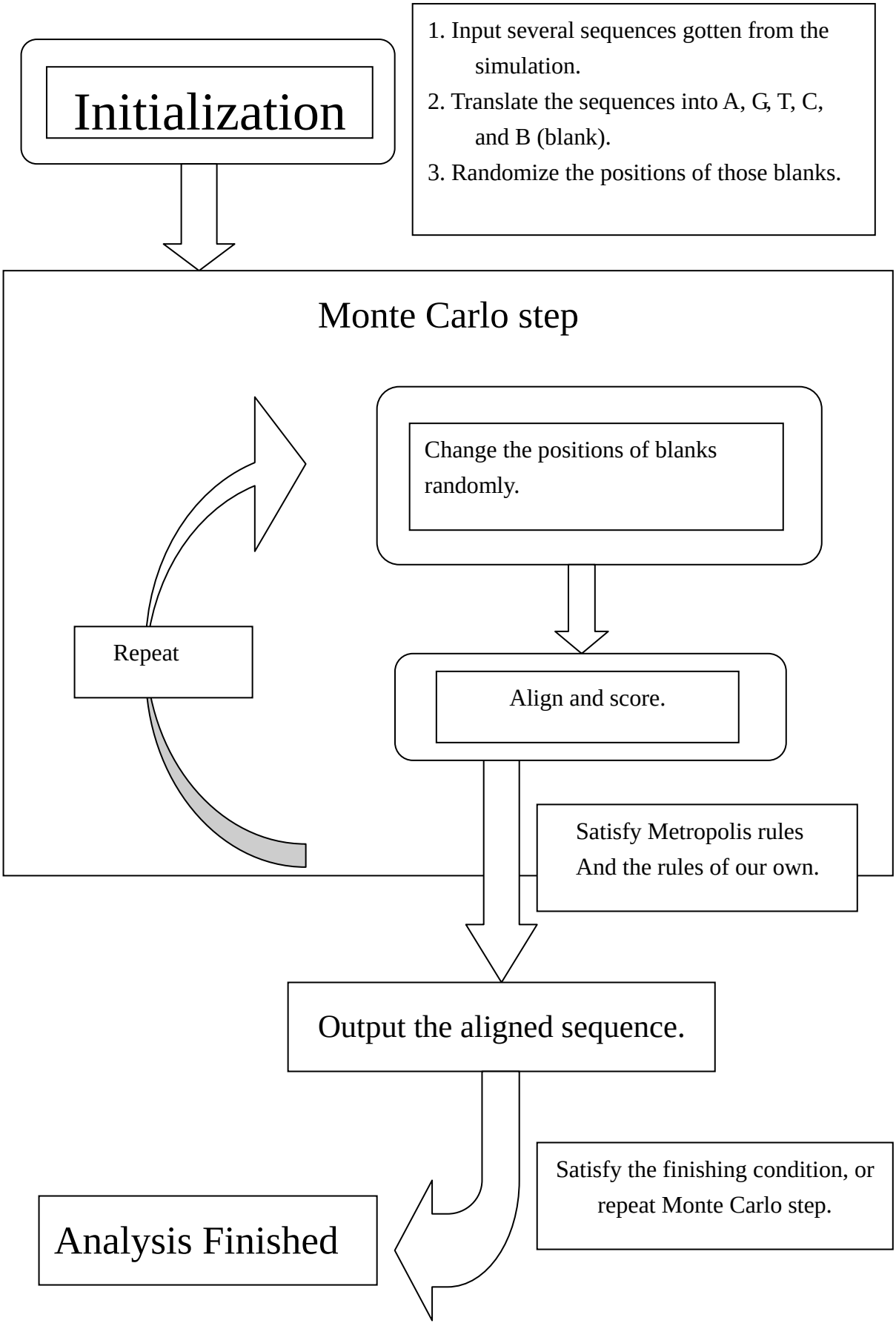
3. ABGTABBCBTAAB

，則很明顯的上述組合比對後的分數為 0，其結果為下面的序列:

ACGTATCCGTAAA

，這也就是待定序列。

在規則中，我們並沒有加入『若是每兩對中，同一行的核苷酸種類相同便得分』，這是因為，當我們加入此條規則時，最高的得分便會大於零，而且無法得知哪個分數才是目標分數，另外一點，加入上述的那條規則，會導致被比對的序列們，其核苷酸會盡量排在同一行，將 B 排在其他地方。因此，得分數雖是最高，但是某一行或是某幾行卻存在著不一致的核苷酸分佈，這是因為犧牲某幾行的不一致，卻可以使其他行擁有更多的相同的核苷酸，造成得分更多，但這並非我們所預期的，因此，排除這樣的規則。


Initialization
1. Input several sequences gotten from the simulation.
2. Translate the sequences into A, G, T, C, and B (blank).
3. Randomize the positions of those blanks.
Monte Carlo step
Change the positions of blanks randomly.
Repeat
Align and score.
Satisfy Metropolis rules
And the rules of our own.
Output the aligned sequence.
Satisfy the finishing condition, or repeat Monte Carlo step.
Analysis Finished

前頁為比對演算法的基本流程圖，另外，我們在此演算法加入了ELP(Energy Landscape Paving)[23、24]的技巧，以避免某些排列組合會將整個蒙地卡羅過程卡住，無法搜尋到更高的分數。

另一方面，針對取樣的B，這裡採取一次移動多個作為測試的組合，這樣的好處是可以在初期比對，排列較亂時，能快速將大部分的序列排列整齊。也就是說，排列速度會較一次只移動一個B有效率。

每一次移動的個數是隨機取的，但是限制在某個數目以下，即假設將移動個數最大值設為M，則每一次取樣時，嘗試移動的B的個數m，會滿足m≦M，m⊂正整數，這樣的選法，會使得每次嘗試移動能有較靈活的選擇。