楓城新聞與評論
Maple News and Commentary

如何學習生物資訊?

臺大生物機電工程學系 陳倩瑜教授

學習生物資訊之所以不容易，是因為涵蓋範圍太廣，從基因體 (genome)、表觀遺傳體 (epigenome)、轉錄體 (transcriptome)、蛋白體 (proteome) 到代謝體 (metabolome)；從計算生物學 (computational biology)、統計 (statistics) 到機器學習 (machine learning)；還有各式各樣的資料庫 (database)，從 DNA 序列、RNA 序列與結構，到蛋白質的序列、結構與功能，進而到蛋白質與小分子之間或蛋白質之間的交互作用等，不同複雜度 (complexity) 的問題需要的計算資源差異很大；隨著產生資料的技術日新月異，從短序列 (short read) 到長序列 (long read)；從單細胞 (single cell) 到空間 (space) 到影像 (image)，如果沒有選定學習目標，很容易失焦，不知道所學為何，碰到問題的時候還是不知道從何下手。

我曾經嘗試過很多種開課方式，也多次以次世代定序 (next-generation sequencing)為主軸開課，其中最成功的一門是：【次世代定序、生物資訊學與基因體醫學】(與基因體暨蛋白體醫學研究所陳沛隆老師/許書睿老師合授)

我覺得這門課之所以成功，是因為我們同時將這幾個元素放入課程中：

* 應用場域：基因體醫學
* 背景知識：遺傳學與基因體學
* 產生資料的技術：次世代定序
* 分析方法：生物資訊演算法與統計檢測
* 視覺化操作介面：TAIGenomics 與 NTU Galaxy

在這門課中，要被分析的資料種類很明確，要回答的問題也相對明確，因此，需要學習的分析工具與方法有哪些，就很清楚；換言之，學習動機很重要，要學習生物資訊，我建議同學們要先為自己訂定學習生物資訊的短期、中期與長期目標，也就是目標導向的學習，從目標來決定學習內容。以下是我經常拋給來登門拜訪的同學思考的問題：

你手上有資料嗎？哪一個物種？哪一種體學的資料？樣本來自有獨特性狀的品系或變異株嗎？過程中，會進一步有以下問題：

問：一定要學 Linux 嗎？

答：不一定，但如果想當工程師，我覺得一定要。

問：一定要學寫程式嗎？

答：不一定，但如果想自己做資料分析，很多工具之間的串接，需

要整理資料格式，若不會寫程式，會綁手綁腳。

問：一定要學機器學習嗎？

答：不一定，但有一些觀念是好的，特別是如果想建立一些預測模型的話。

問：一定要學統計嗎？

答：我覺得要。

問：一定要學資料結構嗎？

答：不一定，但如果想開發演算法，實作程式並釋出給別人使用，就一定要。

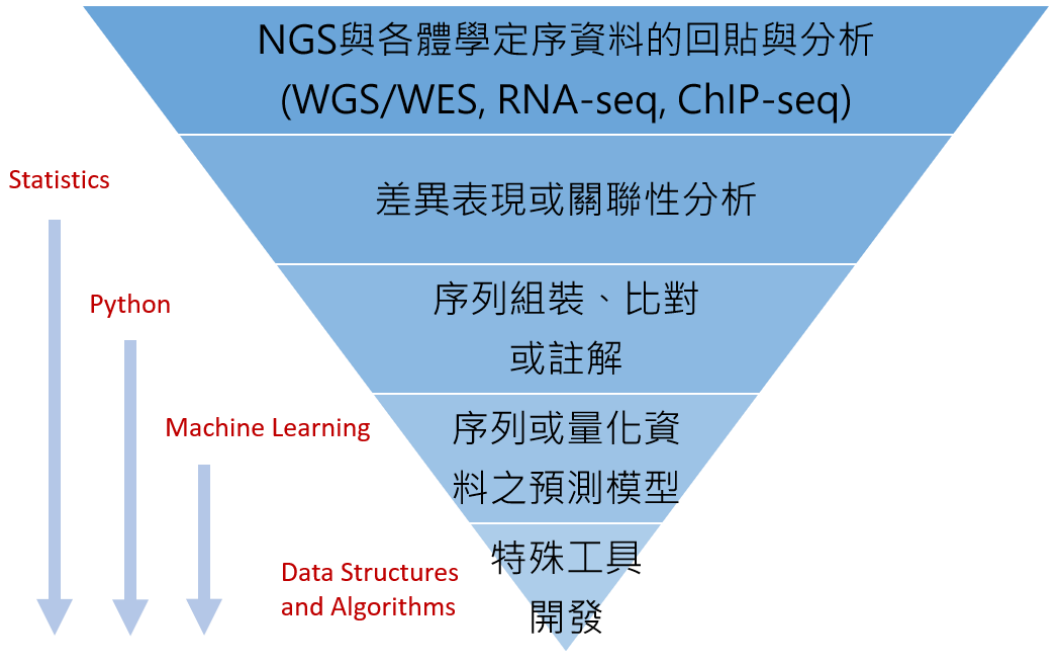
我覺得比較重要的分水嶺是：**你要作資料分析還是工具開發**？如果是後者，那麼上述五個問題的答案都是：YES!!

這並不是說做**資料分析**相對簡單，很多人低估了資料分析的困難性，這其實是跨領域的學習，不僅要對問題的本質有所瞭解，也要試圖理解資料的特性，瞭解產生資料的生物技術，瞭解分析方法的理論，瞭解工具中參數的意義，學習如何串接工具進而回答問題! 我會建議想走這條學習路徑的同學，先選擇主題，例如：癌症? 遺傳疾病? 模式生物? 如果有鎖定某一種資料類型，學習就會很容易

聚焦，例如：基因表現？蛋白質表現？基因變異？是否使用 single-cell 定序技術？等等。

如果想成為**工具開發**者，其專業能力之養成和一般軟體工程師沒有太大差異，但生物資訊工具的開發，其困難性比大家想的高很多，這個領域大部分的工具都是開源(open-source)，想發表新的工具，其準確性或效能一定要比現有的工具好，才有機會被廣為使用，簡單地說，就是要以世界第一為目標。

在培養生物資訊人才的過程，還有一個值得注意的問題：不論是產業界或學術界，對資料分析者的需求，一定遠比對工具開法者的需求來的大很多，畢竟，每一個專題計畫只要產生新的資料，就需要資料分析；反觀，一個創新的計算方法一旦被提出，它就會被不同的研究團隊重覆使用。大多數的時候，工具開法者不會每次都提出一個全新的方法，反而是比較經常地去調整現有方法，讓它們更適合用來分析新型態的資料。而資料分析者有時候也會因為一些新的需求，利用現有的工具組裝出全新的解決方案。



因此，生物資訊的學習，我推薦從各**體學** (omics) 的資料開始，NGS 是目前各體學資料產生方式的大宗，所以首先瞭解 NGS 原理，並學習如何回貼序列是第一步驟；基因體定序資料回貼後，會進行變異偵測 (variant calling)，有時也會進行 CNV (copy number variation) 或 SV (structural variation) 分析；轉錄體定序資料回貼後，會進行基因表現定量 (expression quantification)；表觀遺傳體的定序資料回貼後通常會進行峰值偵測 (peak calling)。這一系列的分析，都可以在開源的生物資訊平台 Galaxy 上進行，工具之間的格式串接得宜，使用者不太需要自己寫程式處理。

在一個研究計畫中，研究人員通常會想瞭解兩種情況 (condition) 下，各體學資料是否有明顯差異，例如：癌症細胞 vs.

正常細胞，用藥前 vs. 用藥後，所以不論上述哪一種體學資料，通常會有多個樣本分組比較的需求，這邊統稱為差異表現分析 (differential analysis) 或關聯性分析 (association study)，像是變異點有或無的差異，基因表現高或低的差異，染色質上各種修飾的峰值差異等等，這一系列的分析，需要使用不同的**統計**方法，因此我覺得統計方法在這個環節很重要。

近年來**單細胞定序** (single-cell sequencing) 技術成熟，各體學都可以進行單細胞資料分析，由於單細胞定序能在一次的實驗中，將各細胞的數據拆分開來，過往 bulk sequencing 要收集多個樣本才能進行差異分析，現在一次的單細胞定序實驗就能得到上千個細胞各自的數據，針對細胞進行分群，再進一步探討組間差異，這使得單細胞定序的資料分析難度一下子拉高好幾個等級，最好有一些**機器學習**的基礎再來學習，會比較容易上手。

在進入多樣本或多細胞的預測模型建立之前，有一些複雜的序列分析值得探討，特別是當研究的物種不再是資源豐富的模式生物 (model organism) 時，我們常會需要序列組裝 (assembly) 搭配序列比對 (alignment) 來進行未知物的辨識與註解，菌相 (microbiome) 分析即是其中一種範例，一些尚未有基因體參考序列的物種，會利

用類似的流程來組裝轉錄體定序資料，再搭配序列比對進行註解。這一系列的工具，彼此串接時可能需要處理資料格式，此時如果不會**撰寫程式**，就可能會卡住。

那什麼時候開始應該學習 **Linux** 呢？等你開始覺得視窗介面沒有效率的時候，或是當你想使用的工具沒有 GUI 的時候，或是當你需要強大的運算資源的時候，自然而然就會尋求更理想的作業環境。

在累積越來越多資料後，利用**機器學習**來挖掘資料中未知的關聯性或是建立預測模型，漸漸成為一種顯學，這裡會建議同學透過課程建立一些機器學習和深度學習的基本觀念，以便理解各種生物資訊預測工具背後的邏輯。在分子層次的預測，有像是蛋白質功能預測或蛋白質結構預測，也有更進一步預測蛋白質可能結合的對象，例如：轉錄因子會結合哪些特定的 DNA 序列？MHC 蛋白質會結合哪些特定的胺基酸片段 (peptide)？在基因變異的層次，有很多計算工具能預測變異點的致病性，或是對蛋白質功能的影響力；在個體的層次，我們會想評估一個人罹患特定疾病的風險 (risk score)，或是預測癌症病人的用藥反應或存活率。在這個階段的學習，有一些**線性代數**的基礎可能是有幫助的，例如：有很多生物資

訊工具都會使用 PCA (principal component analysis)，在高維度的資料分析中很常見。

HMM (hidden Markov model) 也是分析生物序列時常見的方法，在變異偵測 (variant calling)、單體辨識 (haplotyping or allele typing) 或基因預測 (gene prediction)都會看到 HMM 的影子；資料結構中的 graph，也在許多生物資訊演算法中出現，從序列組裝到泛基因組 (pangenomes)，從基因調控網路到蛋白質交互作用網路，都會用到圖論；除此之外，近幾年一些新穎的**深度學習**方法，例如：transformer、VAE (variational auto encoder)等，也都常在生物資訊的方法中看到，想從事工具開發的同學，建議要認真學習這些資訊領域的常用方法，才能適時將這些厲害的方法運用於生物資訊工具的開發中。

結語：

生物資訊的學習，建議先從對多體學資料瞭解開始，可以先嘗試模仿一些相關文獻中的分析，利用其公開的資料，仿照論文中描述的研究方法進行學習。工具的選擇應盡量先選用使用者多的方法，參考其發表論文的引用數是一個方式；從許多生物資訊工具的高引用數可以看出，一個好的生物資訊工具對生醫領域的影響非常

大，想學習資料分析的同學，可以先試著掌握這些工具的原理和操作；想從事工具開發的同學，則建議先將資訊工程技術的基礎打好，再以下列這些生物資訊工具為標竿，尋找現有工具在解決當今重要生醫問題時是否有什麼盲點，思考如何善用最新資訊技術加以突破。就像是 DeepMind 用深度學習的技術解決了蛋白質結構預測的問題，是非常了不起的突破，讓我們期待更多 AlphaFold 2 的誕生吧！