



# 特色課程說明: 生物資訊課程 (BIOINFORMATICS LECTURES)

劉佳鑫 (CHIA-HSIN LIU) 助理教授  
高雄醫學大學 醫學院 臨床醫學研究所  
[martin@kmu.edu.tw](mailto:martin@kmu.edu.tw)

## 碩士

【碩一上】分子及細胞生物學導論(必)

【碩一上】生物醫學研究技術(EMI)(必)

【碩一下】生物資訊及其應用

【碩二上】高通量基因資料分析概論

【碩二上】資料科學於臨床醫學上的應用：多體學資料分析簡介

# 高手是這樣練成的!

## 博士

【博一上】高級分子及細胞生物學(必)

【博一下】高級分子及細胞生物學(必)

【博一上】臨床醫學資料分析/ R程式語言實作

【博二上】進階生物資訊分析與實作 (I)

【博二下】進階生物資訊分析與實作 (II)

## 選課地圖 (生物資訊)

## Basic concepts

### 【碩】高通量基因資料分析概論 Introduction to High-throughput Genomic Data Analysis

- ✓ 高通量基因資料與精準醫學
- ✓ 高通量定序技術簡介
- ✓ 高通量基因資料分析流程介紹
- ✓ DNA-seq資料分析簡介
- ✓ 基因功能注釋及分析
- ✓ RNA-seq資料分析簡介
- ✓ 基因差異表達分析
- ✓ 基因功能和富集分析
- ✓ 蛋白質交互作用分析
- ✓ 基因變異及藥物敏感性分析
- ✓ 機器學習與高通量基因資料
- ✓ 如何建構疾病風險模型
- ✓ 高通量基因資料研究及應用

### 專家講座

## Hands-on practices

### 【博】進階生物資訊分析與實作 Advanced Bioinformatics Analysis and Practice

- ✓ 機器學習簡介
- ✓ 深度學習簡介
- ✓ R軟體簡介及安裝
- ✓ R語言及資料結構
- ✓ 基因資料庫簡介
- ✓ 機器學習分析與實作
- ✓ 深度學習演算法分析與實作
- ✓ 高通量基因資料分析流程簡介
- ✓ Linux系統介紹
- ✓ 生物資訊分析環境建置
- ✓ DNA-seq資料分析實作
- ✓ RNA-seq資料分析實作

(I)

(II)

# 課程總覽 COURSE OVERVIEW

# 臨床醫學資料分析/ R程式語言實作

INTRODUCTION TO DATA ANALYSIS IN BIOMEDICAL DATA: R-LANGUAGE

## 課程特色

- 基礎程式語言邏輯撰寫
- Vide coding
  - 判別與修正AI所產生程式碼
  - 利用AI撰寫網頁可視化報告
- 建立自己的分析程式庫

## 課程資訊

- Thursday 13:10-15:00  
(Office hour: 12:10-13:00)
- IR204 暫定
- 博士班 (不限系所、人數)

## 課程要求

- 準時出席、參與課堂討論
- 缺席達四次者，不授予學分
- 需自備筆電
- 建立網頁可視化報告
- 建立自己的程式庫(GitHub)

## 評量方式

- 上課出席率 (35%)
- 期末報告 (65%)

# 資料科學於臨床醫學上的應用:多體學資料分析簡介

## INTRODUCTION TO MULTI-OMICS DATA ANALYSIS AND APPLICATIONS IN CLINICAL MEDICINE: APPLICATION OF DATA SCIENCE IN CLINICAL MEDICINE

### 課程特色

- 應用體學資料的研究設計
- 資料庫使用
- 體學分析方法與概念

### 課程資訊

- Wed 10:10-12:00 PM  
(Office hour: 12:10-13:00)
- 未定
- 碩士班 (不限系所、人數)

### 課程要求

- 準時出席、參與課堂討論
- 缺席達四次者，不授予學分
- Homework:心得報告x6
- One-page proposal

### 評量方式

- 上課出席率與作業 (35%)
- 期末報告 (65%)

# 高通量基因資料分析概論

## INTRODUCTION TO HIGH-THROUGHPUT GENOMIC DATA ANALYSIS

### 課程特色

- 高通量基因資料基礎結構
- 常見基因資料分析流程及方法
- 專家學者分享研究成果及應用

### 課程資訊

- Thursday 13:10-15:00  
(Office hour: 12:10-13:00)
- IR729 (國研7F) 暫定
- 碩士班 (不限系所、人數)

### 課程要求

- 準時出席、參與課堂討論
- 期末心得報告繳交

### 評量方式

- 上課出席率 (70%)
- 期末報告 (30%)

# 進階生物資訊分析與實作I/II

## ADVANCED BIOINFORMATICS ANALYSIS AND PRACTICE I/II

### 課程特色

- 基因定序資料分析實作
- 機器學習演算法分析流程
- 機器學習模型分析實作.
- 基因資料視覺化

### 課程資訊

- Wednesday 09:10-12:00  
(Office hour: 12:10-13:00)
- IR729 (國研7F) 暫定
- 博士班 (限6-10人，開放旁聽)

### 課程要求

- 缺席達四次者，不授予學分
- 需自備電腦
  - 作業系統需求：Windows (Win 10或以上)/MacOS/Linux
- 需繳交課程作業及報告

### 評量方式

- 上課出席率、作業繳交 (50%)
- 期末報告 (50%)

# EXAMPLES(I): MACHINE LEARNING IN R

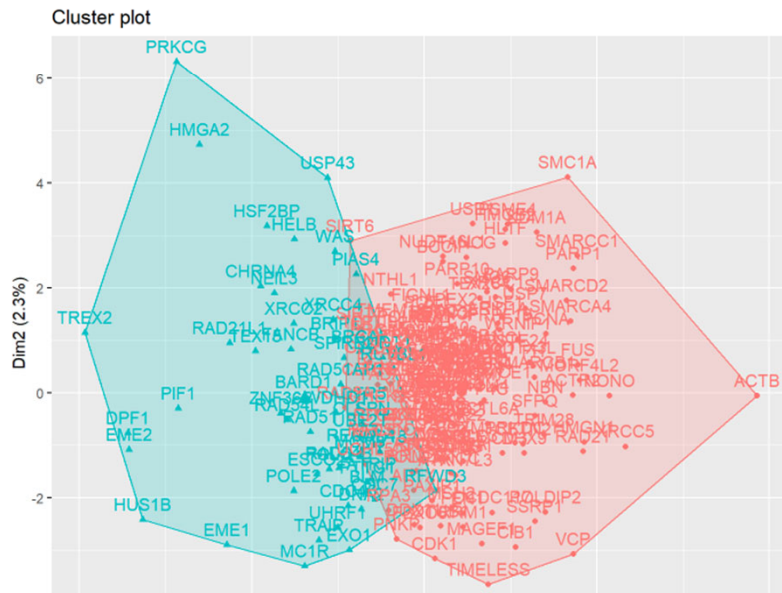
## 3. Hierarchical Clustering (hc\_analyzer)

- Patient-level
- Gene-level to Patient-level

```
# cluster.single = hc_analyzer(data_logfc, level="patient", kmax=20);

# Gene-level to patient-level
cluster.multiple = hc_analyzer(data_logfc, level="multiple", kmax=10);
```

```
## [1] "==Multi-level clustering=="
## [1] "Optimal cluster no. = 2"
## hc.clusters
##      1  2
## 168 52
```



## 5. Linked to survival outcome

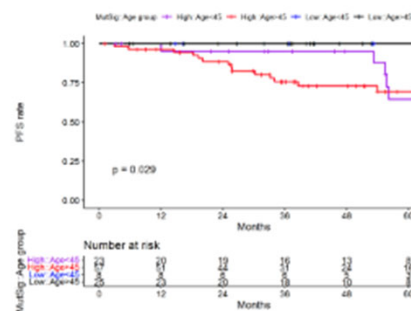
1. Lasso model: Mutation signature (MutSig)
2. Hierarchical clustering: RNA expression (RNA exp)
3. Age group

```
# KM plot ###
library(survival)
library(survminer)
df_km =
  df_lm %>%
  mutate(Mut_sig = case_when("PFS-LMS" >> LMS_recOptimal_cutpoint == "High LMS",
                             TRUE ~ "Low LMS"),
         mRNA_exp = case_when(X1 == 0 ~ "C1", X1 == 1 ~ "C2"),
         Age_group = case_when(AGE < 45 ~ "A5", TRUE ~ ">45"));

fit_c1 = survfit(Surv(PFS_MONTHS, PFS) ~ Mut_sig + Age_group, data = df_km);
fit_c2 =

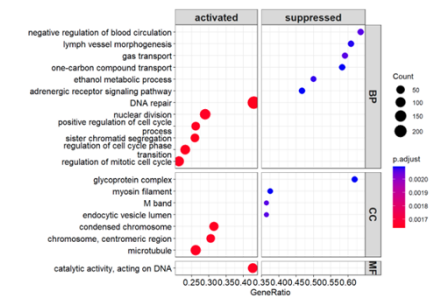
## Call: survfit(formula = Surv(PFS_MONTHS, PFS) ~ Mut_sig + Age_group,
## data = df_km)
##
## 因為不存在 1 個觀察值被刪除了
##              n events median 0.95LCL 0.95UCL
## Mut_sig=High LMS, Age_group<45 23      7  74.7    56.2    NA
## Mut_sig=High LMS, Age_group>45 57     17 120.6    87.9    NA
## Mut_sig=Low LMS, Age_group<45   8       0   NA      NA     NA
## Mut_sig=Low LMS, Age_group>45  25       0   NA      NA     NA

ggsurvplot(fit_c1, data = df_km,
  pval = T,
  risk.table = T, risk.table.y.text = T, censor = T,
  xlab = "Months", break.time.by = 12, xlim = c(0,60),
  ylab = "PFS rate",
  legend.title = "MutSig:Age group",
  legend.labs = c("High:Age<45", "High:Age>45",
                  "Low:Age<45", "Low:Age>45"),
  palette = c("purple","red","blue","black"))
);
```



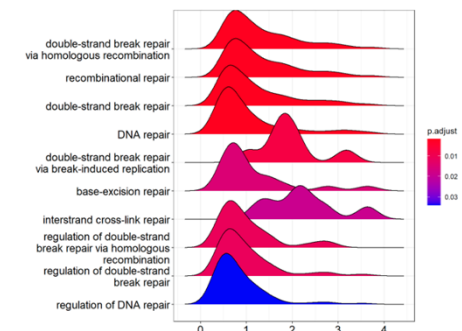
## Select Key Pathways

```
# 5.2. Identified Target Pathways
# 5.2.1. Dotplot ###
dplot_sig =
  dotplot(gse_sig, showCategory=10, label.format=0, split="sign") +
  facet_grid(ONTOLOGY~sign, scale="free", space="free") +
  theme(strip.text.x = element_text(size = 14, face = "bold"),
        strip.text.y = element_text(size = 14, face = "bold"));
print(dplot_sig);
```



## Pathway comparison: Ridge plot

```
gse_sig.target = gse_sig[gse_sig$result$ID %in% gse.target.ids, asis=T]
edo = gse_sig.target;
rplot = ridgeplot(edo);
rplot;
```



MOI-LAB

# EXAMPLES(II): NGS DATA ANALYSIS PIPELINE

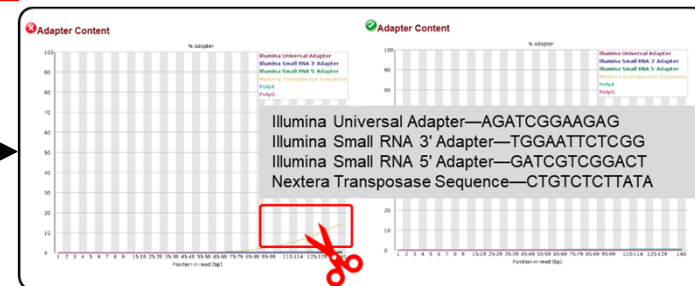
FASTQ

BAM

VCF

1. Quality Control (QC)
2. Reads Mapping (Alignment)
  - Reference genome (hg19, hg38)

Trimmed adapter



3. Marked Duplication

4. Recalibration BAM QC → BQSR

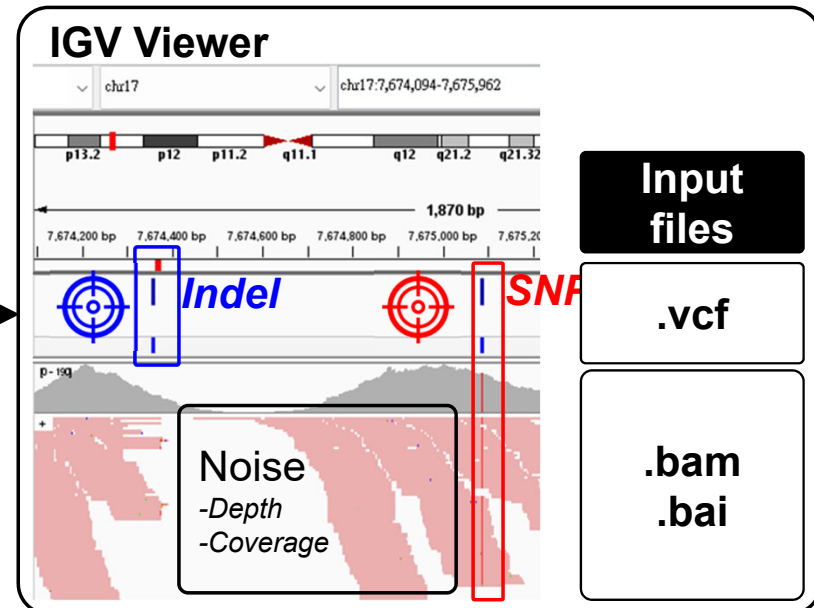
- (Indels) Mills and 1000G gold standard
- (SNPs) 1000G phase1 high confidence SNPs, dbsnp

5. Somatic Calling

- (Germline) af-only-gnomad

6. Annotation

- Functional consequence, ClinVar, ...



## EXAMPLES(III): VIBE CODING

### Vibe Coding Log-Rank Tests

**Prompt to AI:** “Run a log-rank test on *tcga\_surv* to compare survival by *stage\_simple*. Then, run another one for *lymphovascular\_invasion\_present*. Drop NAs for those columns first.”

```
1 # Stage Test
2 tcga_surv |>
3   tidyr::drop_na(stage_simple) |>
4   dplyr::filter(stringr::str_detect(stage_simple, "Stage")) |>
5   survival::survdif(Surv(surv_time, surv_status) ~ stage_simple, data = _)
```

Call:

```
survival::survdif(formula = Surv(surv_time, surv_status) ~ stage_simple,
  data = dplyr::filter(tidyr::drop_na(tcga_surv, stage_simple),
    stringr::str_detect(stage_simple, "Stage")))
```

|                       | N   | Observed | Expected | (O-E) <sup>2</sup> /E | (O-E) <sup>2</sup> /V |
|-----------------------|-----|----------|----------|-----------------------|-----------------------|
| stage_simple=Stage I  | 190 | 13       | 48.0     | 25.484                | 27.499                |
| stage_simple=Stage IA | 215 | 12       | 55.6     | 34.163                | 37.301                |
| stage_simple=Stage IB | 321 | 34       | 83.4     | 29.295                | 33.567                |

# EXAMPLES(III): VIBE CODING

## Exercise 2: Fixing AI File Loading Code

You prompted an AI to write code to load the `TCGA_clinical_patient.xlsx` file. It gave you the following code, but it gives an error when you run it!

### AI Generated Code (Broken):

▼ Code

```
tcga_excel <- read_excel("TCGA_clinical_patient.xlsx")
```

**Question:** What did the AI forget to do? Fix the code so it works properly, using the strict `package::function()` notation we learned.

💡 Solution

>

## Exercise 3: The Missing `na.rm`

You want to calculate the average survival time of the patients. You give the AI this prompt: “Calculate the mean of the `days_to_death` column in the `tcga_data` dataframe using `dplyr`.”

### AI Generated Code:

▼ Code

```
# Assume tcga_data is already loaded via read_tsv()
library(dplyr)
tcga_data |>
  dplyr::summarise(mean_survival = mean(days_to_death))
```

When you run this code, the result is `NA`.

**Question:** Why is the result `NA`? How do you fix the code?

💡 Solution

>

## Exercise 4: Write Your Own Prompt

You want to create a new column called `survival_years` by dividing `days_to_death` by 365.25. Then, you want to keep only the `patient_id` and `survival_years` columns.

**Task:** Write an English or Chinese prompt that you would feed to an AI to get the correct R code. Be as specific as possible (remember the Role, Task, Data Context, and Requirements).

💡 Solution

>

## Exercise 5: Verify AI Visualization Code

You used a good prompt to generate a density plot of `survival_years`. The AI gave you the following code.

**Task:** Run this code in your RStudio. Does it work? Does the plot make clinical sense?

▼ Code

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(readr)

# 1. Load data
tcga_data <- readr::read_tsv("TCGA_clinical_patient.txt")

# 2. Process data & plot
tcga_data |>
  dplyr::mutate(survival_years = days_to_death / 365.25) |>
  tidyr::drop_na(survival_years) |>
  ggplot2::ggplot(ggplot2::aes(x = survival_years)) +
  ggplot2::geom_density(fill = "coral", alpha = 0.5) +
  ggplot2::theme_bw() +
  ggplot2::labs(title = "Density Plot of Patient Survival (Years)", x = "Survival Time (
```

💡 Solution

>

**THANK YOU  
FOR YOUR  
LISTENING!**

