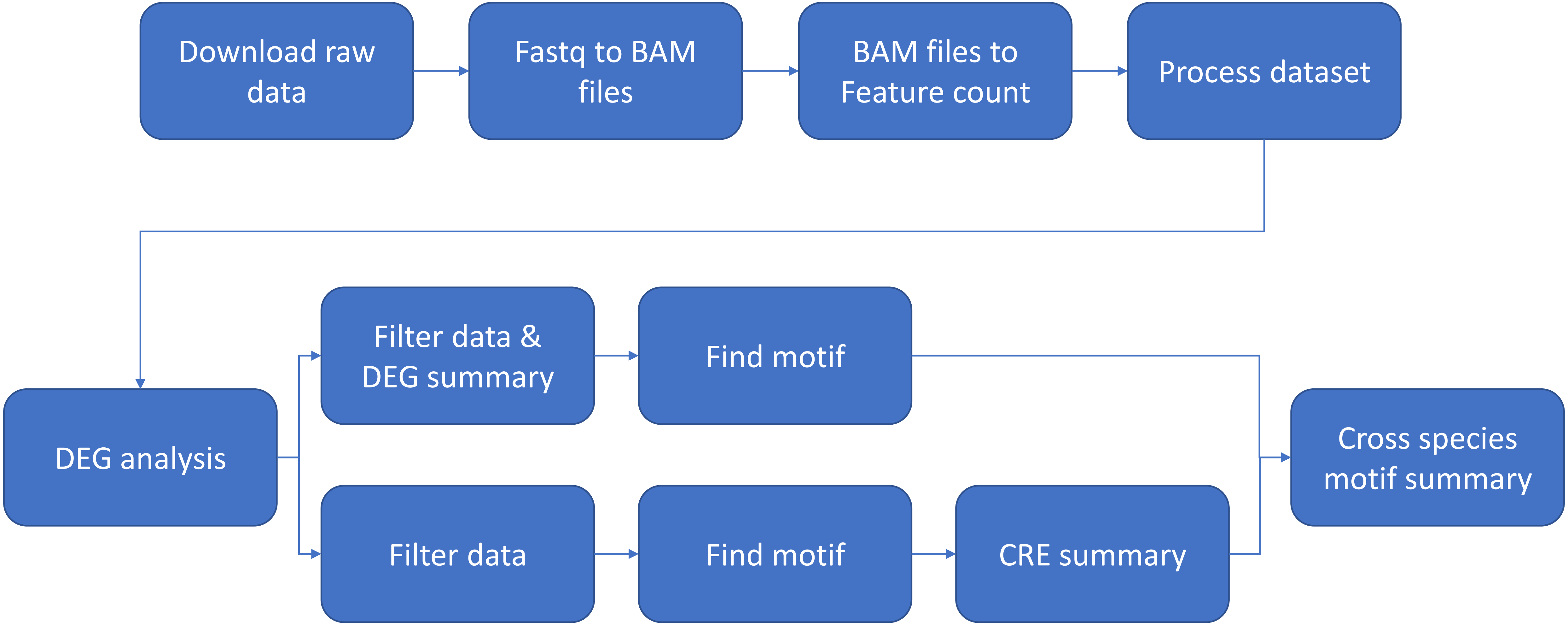
Script workflow



Download raw data

Fastq to BAM files

BAM files to Feature count

Process dataset

DEG analysis

Filter data & DEG summary

Find motif

Filter data

Find motif

CRE summary

Cross species motif summary

Installation

Create environment:

```
git clone https://github.com/brent88310070/CRE_finding.git
cd CRE_finding
conda env create -f environment.yml
conda activate cre
```

Install MEME package:

```
wget https://meme-suite.org/meme/meme-software/5.5.8/meme-5.5.8.tar.gz
tar zxf meme-5.5.8.tar.gz
cd meme-5.5.8
./configure --prefix=$HOME/meme --enable-build-libxml2 --enable-build-libxslt
make
make install
echo 'export PATH=$HOME/meme/bin:$HOME/meme/libexec/meme-5.5.8:$PATH' >
~/miniconda3/envs/cre/etc/conda/activate.d/env_vars.sh
```

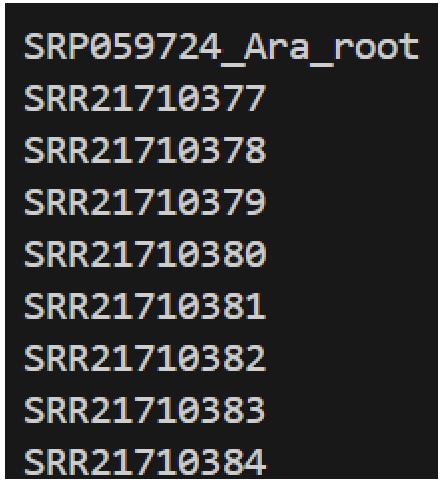
下載SRR檔案 (Fastq)

Download raw data:

```
./download.sh
```

```
# Parameter and threshold
SRR_LIST="srr_list.txt“        # SRR ID 清單檔案
BASE_DIR="raw_data“            # 儲存資料夾
THREADS=8                      # fasterq-dump 使用的執行緒數
KEEP_SRA=false                 # 若要保留 .sra 檔，改成 true
```

srr_list.txt



```
SRP059724_Ara_root
SRR21710377
SRR21710378
SRR21710379
SRR21710380
SRR21710381
SRR21710382
SRR21710383
SRR21710384
```

Project name

SRR ID of files you want to download

Fastq 轉成 BAM files

Fastq to BAM files:

```
./toBAM.sh
```

```
# Parameter and threshold
INPUT_PREFIX="SRP551342_tomato"                        # 專案/實驗前綴
TISSUE="root"                                          # 子資料夾名稱，可留空
BASE_DIR="raw_data"                                    # 儲存主資料夾
GENOME_DIR="./ref/S_lycopersicum/star_index“           # STAR 索引路徑
THREADS=12                                             # STAR 執行緒數
PAIRED=true                                            # true=雙端定序；false=單端定序
GZIP_FASTQ=false                                       # 若為 *.fastq.gz 請設 true
```

若沒有star_index，需從gtf轉star_index:

```
STAR   --runThreadN 8 \
  --runMode genomeGenerate \
  --genomeDir star_index \
  --genomeFastaFiles S_lycopersicum_chromosomes.4.00.fa \
  --sjdbGTFfile ITAG4.1_gene_models.gtf \
  --sjdbGTFtagExonParentTranscript Parent \
  --sjdbOverhang 99
```

BAM files 轉成 Feature count files

BAM files to Feature count files:

```
./to_featureCounts.sh
```

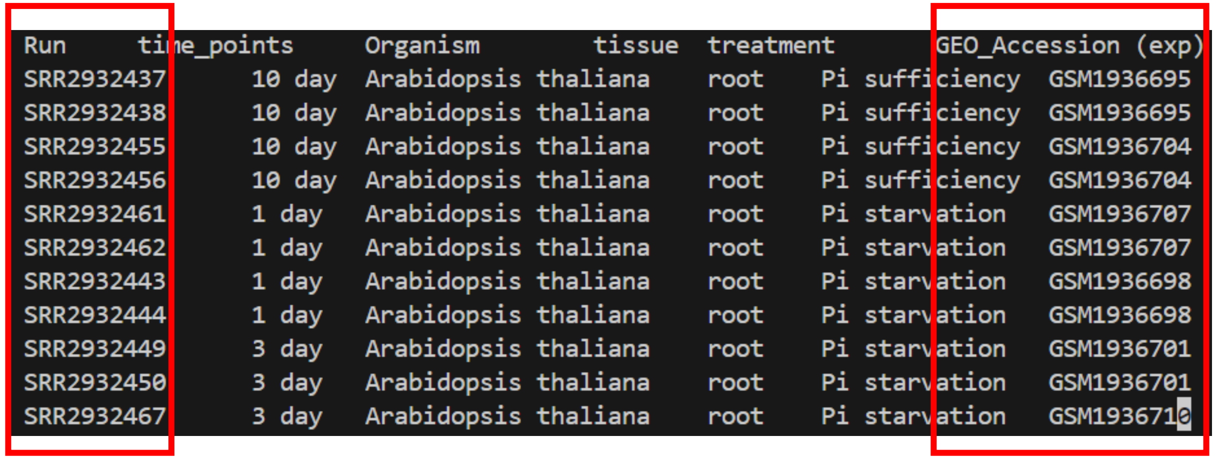
```
# Parameter and threshold
INPUT_PREFIX="SRP551342_tomato"                  # 專案名稱
TISSUE="root"                                    # 子資料夾名稱
THREADS=12                                       # 使用的執行緒數
GTF="ref/S_lycopersicum/S_lycopersicum.gtf"      # 註解檔案（GTF）
IS_PAIRED=true                                   # 是否為 paired-end 資料
```

SRR ID轉成GEO Accession ID & 合併相同的GEO Accession ID

Process same project dataset & Combine same GEO data:

```
python combine_geo_data.py
```

```
# Parameter and threshold
file_name = './SRP399644_tomato_root_count/SRP399644_tomato_counts.txt’     # featureCounts 輸出
output_name = './exp_files/SRP399644_tomato_root_exp.tsv'                  # 最終輸出檔案
meta_path = 'run_info.txt'                                                  # 需包含 SRR 對 GEO ID 的對照表
```



```
Run     time_points     Organism        tissue  treatment       GEO_Accession (exp)
SRR2932437      10 day  Arabidopsis thaliana    root    Pi sufficiency  GSM1936695
SRR2932438      10 day  Arabidopsis thaliana    root    Pi sufficiency  GSM1936695
SRR2932455      10 day  Arabidopsis thaliana    root    Pi sufficiency  GSM1936704
SRR2932456      10 day  Arabidopsis thaliana    root    Pi sufficiency  GSM1936704
SRR2932461      1 day   Arabidopsis thaliana    root    Pi starvation   GSM1936707
SRR2932462      1 day   Arabidopsis thaliana    root    Pi starvation   GSM1936707
SRR2932443      1 day   Arabidopsis thaliana    root    Pi starvation   GSM1936698
SRR2932444      1 day   Arabidopsis thaliana    root    Pi starvation   GSM1936698
SRR2932449      3 day   Arabidopsis thaliana    root    Pi starvation   GSM1936701
SRR2932450      3 day   Arabidopsis thaliana    root    Pi starvation   GSM1936701
SRR2932467      3 day   Arabidopsis thaliana    root    Pi starvation   GSM1936710
```

每個Project 需要跑一次，ex: Project name SRP399644
run_info.txt須包含Run & GEO_Accession (exp) 這兩行
其他行，可有可無

DEG analysis: 將不同實驗的Control & Treatment進行DEG分析

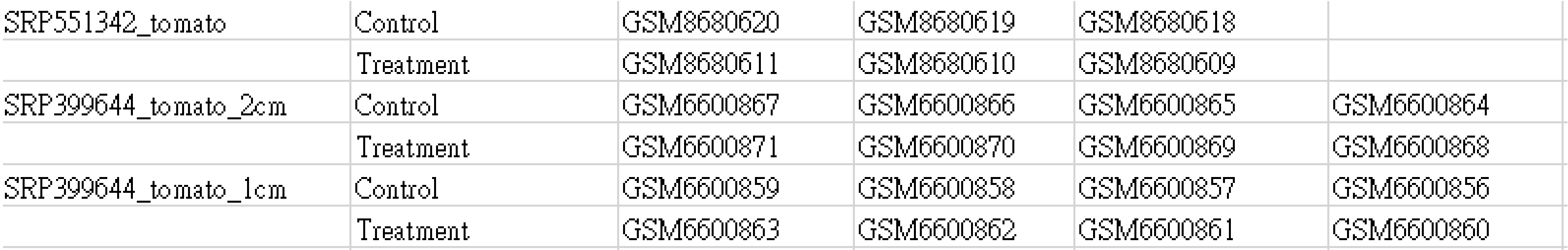
After merging the expression data of the same species, DEG analysis was performed:

```
python deg_analysis.py
```

```
# Parameter and threshold
# --- 合併 expression 檔 ---
input_folder   = "./exp_files/tomato"                    # 單實驗 expression 檔資料夾
file_pattern   = "*.tsv"                                 # 檔案格式
index_col      = "Geneid"                                # 基因欄位名稱
merged_counts  = "./exp_files/Tomato_all_exp.tsv"

# --- DESeq2 相關 ---
sample_info_path = "./read_treat_control_list.txt"       # Treatment/Control 配對設定
deg_output_dir   = "./tomato_deg_results"                # DESeq2 輸出資料夾
```

可以先在Excel檔編輯:
Excel檔中的格式如右，然後複製
到read_treat_control_list.txt



SRP551342_tomato	Control	GSM8680620	GSM8680619	GSM8680618	
	Treatment	GSM8680611	GSM8680610	GSM8680609	
SRP399644_tomato_2cm	Control	GSM6600867	GSM6600866	GSM6600865	GSM6600864
	Treatment	GSM6600871	GSM6600870	GSM6600869	GSM6600868
SRP399644_tomato_1cm	Control	GSM6600859	GSM6600858	GSM6600857	GSM6600856
	Treatment	GSM6600863	GSM6600862	GSM6600861	GSM6600860

DEG summary

DEG summary & get promoter: 整合不同DEG lists的結果

Combine several DEG lists (DEG summary) & get promoter:

```
./deg_summary_and_get_promoter.sh
```

```
# Parameter and threshold
# — Step-1：合併各樣本 DEG 結果 ————————————————
INPUT_DIR="./tomato_deg_results"      # 存放多個 *.tsv
PADJ_TH=0.05                          # FDR 門檻
FC_TH=1                               # |log2FC| 門檻
# — Step-2：篩選嚴謹 DEG / Non-DEG ———————————————
SPECIES="tomato"                      # 也會作為檔名前綴
SIG_COUNT=2                           # sig_count ≥ ?
DEG_FC_TH=1                           # meta_log2FC > ?
NON_P_TH=0.1                          # Non-DEG 的 meta_p > ?
NON_FC_NEAR0_TH=0.1                   # |meta_log2FC| ≤ ?
# — Step-3：擷取啟動子序列 ————————————————————
PROMOTER_UP_BP=1000                   # 1000 或 2000
NEG_MULTIPLIER=3                      # 每個 positive 配幾個 negative，3代表neg是pos數量的三倍
NEG_MIN=1000                          # 至少多少條 negative
GFF_PATH="ref/S_lycopersicum/ITAG4.1_gene_models.gff"
FASTA_PATH="ref/S_lycopersicum/S_lycopersicum_chromosomes.4.00.fa"
```

使用STREME來找motifs

Find motif by STREME:

```
./run_motif.sh
```

```
POS=arabidopsis_DEG_promoter_1kb.fa          # Positive sample
NEG=arabidopsis_nonDEG_promoter_1kb.fa       # Negative sample
OUT=motif_out                                # output director
OUT_DIR=Arabidopsis                          # sub director
MINW=6                                       # short motif length
MAXW=15                                      # longer motif length
N_MOTIFS=20                                  # motif number
```

streme_xml_to_html 出問題，需安裝: sudo apt-get install libxml-parser-perl

CRE summary

取得每個實驗 DEG analysis的 DEG以及non DEG的 Promoter

Get and filter DEG & non DEG in each experiments and promoter:

```
./extract_multi_expt_DEG_and_promoter.sh
```

```
# Parameter and threshold
# 1) DEG / non-DEG GeneID 清單 (extract_multi_expt_DEG_and_nonDEG.py)
INPUT_DIR="./tomato_deg_results"                    # 存放 DESeq2 .tsv 的目錄
OUTPUT_DIR="./multi_exp_tomato"                     # 產出清單的根目錄
FILE_PATTERN="*.tsv"                                # 要讀取的檔案格式
BASEMEAN_DEG_MIN=10                                 # DEG 條件
LOG2FC_DEG_MIN=1
PADJ_DEG_MAX=0.05
LOG2FC_NON_MAX=0.1                                  # non-DEG 條件
PADJ_NON_MIN=0.1
PROMOTER_UP_BP=1000                                 # 1000 或 2000
NEG_MULTIPLIER=3     # 每個 positive 配幾個 negative
# 2) Promoter 擷取 (extract_multi_expt_promoter.py)
GFF_PATH="ref/S_lycopersicum/ITAG4.1_gene_models.gff"
FASTA_PATH="ref/S_lycopersicum/S_lycopersicum_chromosomes.4.00.fa"
PROMOTER_UP_BP=1000                                 # 1000 = −1 kb，2000 = −2 kb …
DEG_FILENAME="DEG.txt"                              # 若改名請同步修改
NONDEG_FILENAME="nonDEG.txt"
```

使用STREME來找每個實驗的motifs

Find motif by STREME:

```
./run_multi_expt_motif.sh
```

```
ROOT_DIR="./multi_exp_tomato"          # ← 放各 sample 子資料夾的根目錄
PROM_SIZE="1kb"                        # "1kb" / "2kb"... (檔名必須含此字串)
MINW=5                                 # 最短 motif 長度
MAXW=15                                # 最長 motif 長度
N_MOTIFS=20                            # 要找幾個 motif
VERBOSITY=1                            # streme --verbosity
```

streme_xml_to_html 出問題，需安裝: sudo apt-get install libxml-parser-perl

整合不同實驗的motif，並移除冗餘motifs

Summary CREs & remove redundant motifs:

```
python cre_integrate.py
```

```
# Parameter and threshold for cre_integrate.sh
# — 資料來源與去冗餘後輸出 —
DATA_DIR       = "./multi_exp_tomato"                          # 多個實驗資料夾
FILTERED_MEME   = f"{DATA_DIR}/filtered.meme"
KEPT_ID_TSV     = f"{DATA_DIR}/kept_motif_ids.tsv"
REDUNDANT_TSV   = f"{DATA_DIR}/redundant_motif_reps.tsv"
# — 參數設定 —
EVALUE_FILTER   = 10.0                                          # STREME motifs：保留 E ≤ 10
TOMTOM_THRESH   = 0.05                                          # Tomtom q-value 閾值
# — Logo 繪圖 —
N_LOGO_MOTIFS   = 5                                             # filtered.meme 前 N 個
LOGO_OUT_DIR    = f"{DATA_DIR}/motif_logos"
N_REP_LOGOS     = 5                                             # 代表 motif 前 N
REP_LOGO_DIR    = f"{DATA_DIR}/reps_motif_logos"
FIGSIZE         = (4, 1.5)                                      # 單圖尺寸 (inch)
DPI             = 200                                           # 解析度
COLOR_SCHEME    = "classic"                                     # Logomaker 色盤
```

畫Top N motif

畫Top N repeat motif

Cross species motif summary

整合在不同物種皆出現的motifs

Cross species motif summary (two species):

```
python cross_species_motif_cre_summary.py
```

```
# Parameter and threshold
SUMMARY_TYPE = "deg_summary"                    # 或 "cre_summary"

TOMTOM_THRESH   = 0.05                          # Tomtom q-value 上限
TEMP_DIR       = "tomtom_cross_temp"            # Tomtom 暫存資料夾
OUT_DIR        = "cross_species_motif"

if SUMMARY_TYPE == "deg_summary":
   SPECIES1_FILE   = "./motif_out/arabidopsis_1kb_sig_count_6/streme.txt"
   SPECIES2_FILE   = "./motif_out/tomato_1kb_sig_count_1/streme.txt"
   CRE_DIR = os.path.join(OUT_DIR, "deg_summary")
elif SUMMARY_TYPE == "cre_summary":
   SPECIES1_FILE   = "./multi_exp_arabidopsis/filtered.meme"
   SPECIES2_FILE   = "./multi_exp_tomato/filtered.meme"
   CRE_DIR = os.path.join(OUT_DIR, "cre_summary")

OUT_TSV        = os.path.join(CRE_DIR, "repeat_motif_cross_species.tsv")
```