GO、HPO 与植物性状本体

Python 实现的 GO 富集分析

Jingbo Xia

华中农业大学（Huazhong Agricultural University）

xiajingbo.math@gmail.com

2026-10-22

课程结构

1. 基因本体（GO） 4
2. 富集分析的数学基础 12
3. 人类表型本体（HPO） 16
4. 植物性状本体（PTO/RTO） 21
5. 富集分析 Python 实验 29

什么是本体?

本体（Ontology）是什么？[a]

- 它收集某个领域中的概念，并描述概念的属性及相互关系。
- 它不只是词表或分类目录，还对概念的定义和关系作形式化描述。
- 它把领域知识表达为机器能够处理的语义结构。

[a] Harrow et al. Ontology mapping for semantically enabled applications. Drug Discovery Today, 2019. https://www.sciencedirect.com/science/article/pii/S1359644618304215

[Jingbo] 如遇专有名词，请勤记、辨识。

1 基因本体（GO） 4

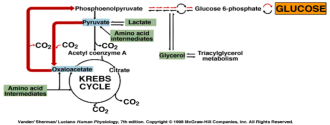
名称不同，可能指同一概念；名称相近，也可能指不同概念。
基因本体（Gene Ontology，GO）用稳定编号、定义和条目间的关系组织基因产物的功能知识。它使文献中的术语、数据库注释和跨物种比较有共同的参照。[1]

id: GO:0006094	unique GO ID
name: gluconeogenesis	term name
namespace: process	ontology
def: The formation of glucose from noncarbohydrate precursors, such as pyruvate, amino acids and glycerol. [http://cancerweb.ncl.ac.uk/omd/index.html]	definition
exact_synonym: glucose biosynthesis	synonym
xref_analog: MetaCyc:GLUCONEO-PWY	database ref
is_a: GO:0006006 is_a: GO:0006092	parentage

读条目时至少看 `id`、`name`、`def`、`synonym` 和 `is_a`：编号用于定位，定义用于判断含义；上下位关系不等于同义关系。

[1] https://geneontology.org/docs/ontology-documentation/

先看文中所述过程，再查本体定义。
糖异生（GO:0006094）指由非糖前体形成葡萄糖。`glucose biosynthesis` 是该条目收录的同义词；`glucose synthesis` 等较宽泛的表述，还要结合句子判断是否符合这一定义。[2]



三羧酸循环（GO:0006099）虽也属于碳代谢，但反应任务不同，不能和糖异生合并。糖异生又是葡萄糖代谢过程（GO:0006006）的下位概念，不能把父条目当作精确同义词。

[2] https://amigo.geneontology.org/amigo/term/GO:0006094

同一蛋白可以分别获得 **MF**、**BP** 和 **CC** 注释。

类别	回答的问题	条目例子
分子功能（MF）	基因产物能执行什么分子活动？	DNA 结合（GO:0003677）
生物过程（BP）	它参与什么生物学过程？	信号转导（GO:0007165）
细胞组分（CC）	它位于何处或构成什么结构？	细胞核（GO:0005634）

“在细胞核中”只说明位置，不能自动推出该蛋白结合 DNA；“参与信号转导”也不表示单个蛋白完成整个过程。

1. 基因本体（GO）

同一事件，三个不同条目

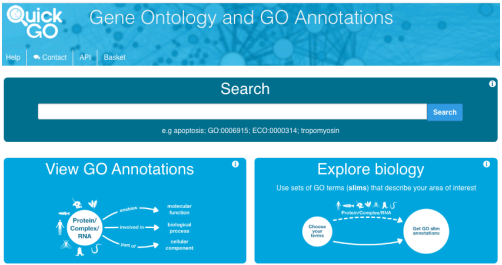
以蛋白质合成为例：

- 核糖体（GO:0005840）是执行合成活动的结构，属于 **CC**。
- 翻译（GO:0006412）描述合成蛋白质的过程，属于 **BP**。
- 核糖体结构组分（GO:0003735）描述基因产物的分子功能，属于 **MF**。

这些条目彼此相关，却不是同义词。文献标注应先确定句子所说的是活动、过程还是结构，再给出 GO 编号。

[Jingbo] 课堂判断：句子只写“核糖体”，能否据此标成“翻译过程”或“核糖体结构组分”？

查询条目，不等于找到某个基因的功能证据。



在 QuickGO[3] 中可按编号查名称、定义、上下位条目及注释记录。例如 GO:0006094 是 GO:0006006 的下位过程；反向推断不成立。

一条 GO 功能注释还需给出基因产物、关系、**GO** 编号、证据类别与来源。仅在句子里识别到“DNA binding”，尚未证明文中某个蛋白具有该功能。

[3] https://www.ebi.ac.uk/QuickGO/annotations

“注释”一词在不同任务中指不同输出。

- 文本标注：在句子中找到术语或实体的边界，并归一到合适的编号。
- 基因组结构注释：在序列中定位基因、外显子等区域。
- GO 功能注释：建立“基因产物–关系–GO 条目”的有证据陈述。

EXP 类证据来自实验，IEA 是电子自动注释；阅读时要核对证据来源。数据库中没有某条注释，不等于已证明该功能不存在。[4]

[4] https://geneontology.org/docs/go-annotations/

目标基因集中的某个 **GO** 条目，是否比背景中更常见？
先确定目标基因集与可检测的背景基因集，再用 GO 注释表统计命中数。富集检验给出观察到的重叠相对于随机抽样的程度，多个条目同时检验时还须校正 p 值。

$$\text{文献术语映射} \quad \neq \quad \text{基因产物的 GO 注释} \quad \neq \quad \text{基因集的 GO 富集结果}$$

即使一个条目富集，也不能仅凭它断言过程被激活、某基因具有因果作用，或相关疾病表型已被证实。后面的 Python 实验会逐步读入基因集、计算并解释这些结果。



2 富集分析的数学基础 12

先说清四个计数，再谈 p 值。



抽球问题	GO 富集分析
N：袋中球的总数	背景中可检测、且可用于检验的基因数
m：黑球数	背景中注释到指定 GO 条目的基因数
k：抽出的球数	目标基因集的有效基因数
x：抽中的黑球数	目标集中命中该条目的基因数

零假设是：目标基因相对于该条目没有额外富集。不放回随机抽样是统计参照，并不是说真实实验中的基因真的由球袋随机抽出。

富集检验取“至少命中 x 个”的上尾概率。

$$p = \Pr(X \geq x \mid H_0) = \sum_{i=x}^{\min(k,m)} \frac{\binom{m}{i}\binom{N-m}{k-i}}{\binom{N}{k}}.$$

若 $N = 15,\ m = 7,\ k = 6,\ x = 4$，则 $p = P(4) + P(5) + P(6) = 0.230769$。这与“恰好命中 4 个”的概率不同。
在 Python 中可用 SciPy 核对同一结果：

```
from scipy.stats import hypergeom  hypergeom.sf(3, M=15, n=7, N=6)
```

函数计算的是 $P(X > 3) = P(X \geq 4)$。在真正的 GO 分析中，同一数据要检验许多条目，不能只看未经校正的 p 值。

背景、命中数和多重校正必须一起报告。

- 目标集命中比例为 x/k，背景比例为 m/N；比例较高不必然意味着统计显著。
- 一次检验的原始 `P-value` 与多条目检验后的 `Adjusted P-value` 不同。后者将在 Python 实验中由 GSEApy 采用 BH 方法计算。
- 若使用了不同的背景基因或 GO 注释版本，同一目标基因集的结果也可能改变。

GO 富集说明目标集对某个注释条目的命中是否偏多；它不直接检验通路是否被激活，也不证明某个基因导致疾病。Python 实验会用具体输出表检查这些边界。

3 人类表型本体（HPO） 16
- HPO 基础 17
- HPO-Shuffle 基因排序案例 19

人类表型本体（**Human Phenotype Ontology**，**HPO**）提供标准术语，描述疾病相关的异常表现，并组织疾病与表型之间的关联。[5]

[Term]
id: HP:0000006
name: **Autosomal dominant inheritance**
alt_id: HP:0001415
alt_id: HP:0001447
alt_id: HP:0001448
alt_id: HP:0001451
alt_id: HP:0001455
alt_id: HP:0001456
alt_id: HP:0001463
def: "A mode of inheritance that is observed for traits related to a gene encoded on one of the autosomes (i.e., the human chromosomes 1-22) in which a trait manifests in heterozygotes. In the context of medical genetics, an autosomal dominant disorder is caused when a single copy of the mutant allele is present. Males and females are affected equally, and can both transmit the disorder with a risk of 50% for each child of inheriting the mutant allele." [HPO:curators]
synonym: "Autosomal dominant" EXACT []
synonym: "Autosomal dominant form" RELATED [HPO:skoehler]
synonym: "Autosomal dominant type" RELATED [HPO:skoehler]
xref: SNOMEDCT_US:263681008
xref: UMLS:C0443147
is_a: HP:0000005 ! Mode of inheritance

图中的 HP:0000006 是常染色体显性遗传，描述遗传方式，不是患者的异常体征。HPO 还包含起病时间等类别；编号、定义和上下位关系要连同条目类别一起读。

[5] https://hpo.jax.org/app/help/introduction

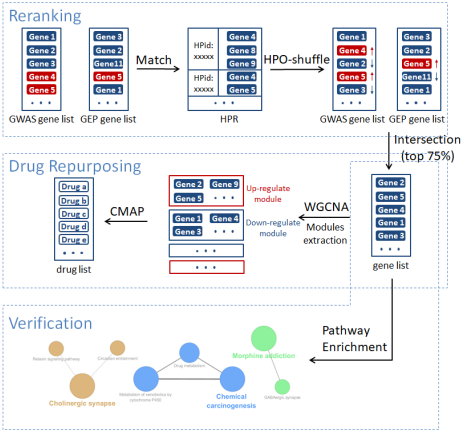
一个疾病可以关联多个表型；一个表型也可见于不同疾病。

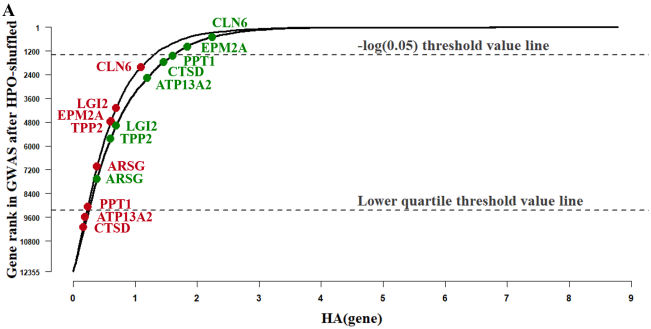
疾病编号	基因	HPO 编号	记录内容
OMIM:608647	HYDIN	HP:0011108	反复鼻窦炎
OMIM:608647	HYDIN	HP:0002837	反复支气管炎
OMIM:608647	HYDIN	HP:0003593	婴儿期起病

前两项分别涉及上、下呼吸道；第三项表示起病时间。HYDIN 相关的纤毛运动障碍可以为反复呼吸道感染提供生物学解释，但数据库中的关联记录并不是某位患者的诊断结论。[6]
文本中的“反复鼻窦炎”先要归一到 HP 编号，之后才能与疾病–基因–表型关联数据比较。使用时保存疾病、基因、条目编号和来源。

[6] https://www.ncbi.nlm.nih.gov/books/NBK1122/

案例：**HPO-Shuffle** 重新排列犬癫痫候选基因。
研究先从 GWAS 等数据得到候选基因，再用 HPO 疾病表型关联调整名次，随后寻找候选药物；具体算法不在本课展开。药物只是研究线索，不是疗效结论。





图中红、绿点分别表示八个已报道犬癫痫相关基因重排前后的位置，不是 $-\log p$ 的数值。五个基因名次上升，三个下降。

这一例子展示了本体信息如何帮助候选基因排序；不能说所有基因都得到改善，也不能凭名次变化证明某个候选药物有效。跨物种使用人类 HPO 信息时，还需核对犬基因、疾病和表型之间的对应证据。

论文：https://pmc.ncbi.nlm.nih.gov/articles/PMC6732366/

4 植物性状本体（PTO/RTO） 21
- PTO 基础 22
- 水稻性状本体（RTO） 24
- 植物性状文献标注案例 27

植物性状本体（**Plant Trait Ontology**，**PTO**）把不同论文和作物数据库中的性状描述对应到共同条目。[7]

一条记录通常包含 **TO** 编号、首选名称、同义词、定义和上位条目，还可能标记为已废弃。BioPortal 页面使用 PTO 作为本体简称，条目编号写作 `TO:...`。

条目	需要辨明的对象
花色	花器官在指定时期的颜色
叶片衰老	叶随生长过程发生的变化，失绿只是可能的观察指标
可溶性糖含量	植物或指定器官中可溶性糖的测量值

即使名称里都有“颜色”或“含量”，也不能忽略器官、发育阶段和测量对象。

[7] https://bioportal.bioontology.org/ontologies/PTO

耐冷性（**TO:0000303**）描述植物在低温下的表现。[8]

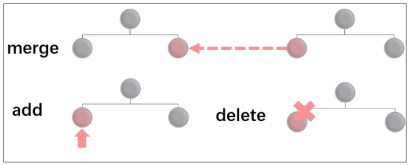
- “幼苗低温处理 72 小时”给出的是处理条件，尚未报告耐冷表现。
- “处理后甲品种成活率高于乙品种”才有相对耐冷表现的证据，还要核对温度、阶段、重复和对照。
- “某基因在低温下表达升高”是分子响应，不足以证明该基因使植株更耐冷。

同一个 TO 编号能归一化不同论文的性状名称；比较性状结论时，还须保留物种、器官、处理和测量值。只抽到性状词，不能直接抽出基因–性状因果关系。

[8] http://purl.obolibrary.org/obo/TO_0000303

水稻性状本体（**Rice Trait Ontology**，**RTO**）的建设案例从已有植物性状资源出发，整理适用于水稻的条目，再处理原有本体未覆盖或含义不合适的概念。

Rice Trait Ontology Construction



As the figure shows, three strategies for ontology integration:

- **Merge**
- **Add**
- **Delete**

图示把工作概括为合并、增补和删去。三种操作都要保留原编号、定义和决策理由；文字相似并不足以证明两个条目等价。

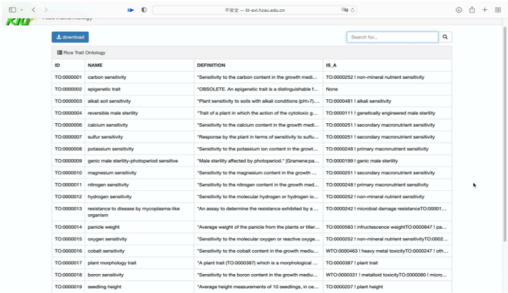
“营养缺乏”与“营养敏感性”应分开。

原图条目	实际描述
WTO:0000095 营养缺乏	培养介质中的营养供给不足，是环境或处理条件
TO:0000480 营养敏感性	植株对营养变化的反应，是要测量的性状

原图把前者并入后者，适合作为课堂反例：在缺素条件下生长，不代表植株高度敏感。可建立“条件影响表现”的关联，但不能将二者标为精确同义词。

[Jingbo] 本体对齐前，先判断原文说的是环境条件、植物性状，还是测量结果。

Rice Trait Ontology Web: http://lit-evi.hzau.edu.cn/RiceTraitOntology



网页查询应能返回条目编号、名称、定义和上位条目。做文献术语归一时还应保存原文表达与来源；若只有相近词而定义不符，应保留“未匹配”或较宽泛的候选关系，而不是强行给出编号。

植物性状文献标注试验：三项任务的输出不同。[9]

- 文献筛选：这篇文献是否提及植物性状？
- 条目标注：提及的性状应对应哪个 TO 编号？
- 基因–性状关联：文献是否给出某基因与该性状相关的证据？

第二项用到了本体的名称、同义词和定义；第三项还需基因标识、关系方向和证据句。前两项完成，不表示第三项也成立。

[9]项目资料：https://github.com/bionlp-hzau/RiceTraitOntology

教学示例：`Rice seedlings showed improved cold tolerance.`

- 文献提及耐冷性，任务一可以判为“是”。
- `cold tolerance` 可作为 TO:0000303 的候选映射；核对定义并保存原词、编号和物种。
- 句子没有给出基因名称，不能凭此建立基因–耐冷性关联。

句中的 `improved` 也没有说明对照、温度和测量指标。若原文是 `Cold tolerance was not measured.`，性状仍被提及，却没有肯定的观测结果。标注时须分开记录术语提及和事实判断。

5 富集分析 Python 实验 29
- 输入数据与运行环境 30
- 背景基因与 GO 基因集 36
- 执行富集并检查结果 44
- 结果图与可重复运行 52

问题：老师选定的 **5** 个基因在 **GO** 的哪些条目中比背景数据更显著集中于某些 **GO** 条目??

操作顺序

获取 `GeneNames` 中的 gene symbol

↓

读入 Ensembl BioMart 人类背景基因

↓

使用 GSEApy 进行 GO 富集分析（注: 与传统 r 代码进行对照）

↓

解读：结果表、气泡图、柱状图与 GO 层级图

Jingboi 注：
输入是一组基因符号，没有表达量或上、下调方向。
Enrichr 在本实验中提供 GO 基因集；富集统计由 GSEApy 在本地完成。

实验目录中的文件各有用途。

```
experiments/
|-- GeneNames
|-- requirements.txt
|-- go_enrichment_gseapy.py
|-- plot_go_enrichment.py
|-- results/
```

GeneNames：每行一个人类基因符号

```
AASDH
ABCB11
ADAM12
ADAMTS16
ADAMTS18
```

输入不含表达量。本实验把这 5 个符号作为一个目标基因集，先检查符号能否在背景中找到，再进行 GO 统计。

```
cd experiments
py -3.12 -m venv .venv312
.\.venv312\Scripts\python.exe --version
.\.venv312\Scripts\python.exe -m pip install -r requirements.txt
.\.venv312\Scripts\python.exe -m pip show gseapy pandas
```

`requirements.txt` 列出 GSEApy 和 pandas；数值计算、绘图及网络读取所需依赖会一并安装。

[Jingbo] 注：本次实际运行记录：Python 3.12.14、GSEApy 1.3.1、pandas 3.1.0。(Kimi3 协助编程)

核心文件：**experiments/go_enrichment_gseapy.py**

关键语句	在本实验中的作用
`import gseapy as gp`、`import pandas as pd`	分别调用富集分析功能和表格处理功能；`requests` 负责请求 BioMart。
`HERE = Path(__file__).resolve().parent`	取得脚本所在目录，使输入和输出路径不依赖从哪个目录启动命令。
`GENE_FILE`、`RESULTS_DIR`	集中定义输入文件和结果目录，后续函数复用这些路径。
`read_gene_symbols()`、`fetch_human_background()`	分别准备目标基因列表和人类参考背景。
`newest_library()`、`save_gmt()`	选择 GO 库版本并保存本次使用的注释快照。
`main()` 与文件末尾的主程序判断	按顺序调用上述步骤；直接运行脚本时启动完整流程。

先把文本文件转换成干净、无重复的 **Python** 列表。

```python
def read_gene_symbols(path):
    if not path.is_file():
        raise FileNotFoundError(path)

    column = pd.read_csv(
        path, header=None, sep=r"\s+", dtype=str
    ).iloc[:, 0]
    symbols = column.dropna().str.strip()
    symbols = symbols[symbols.ne("")].drop_duplicates().tolist()

    if not symbols:
        raise ValueError("No gene symbols found")
    return symbols
```

`header=None` 防止把第一行误当列名；`dropna()` 与空字符串检查清除空行；`drop_duplicates()` 防止重复符号被重复计数。读入后本例有 5 个不同符号。

关键语句	解释
`path.is_file()`	先确认文件存在；路径错误时立即报错，不让空输入继续流入统计。
`pd.read_csv(..., header=None, sep=r"\s+", dtype=str)`	文件没有表头；按空格或制表符分隔；全部按文本读取，保留基因符号原样。`.iloc[:, 0]` 取第一列。
`dropna().str.strip()` 与 `symbols.ne("")`	去掉缺失值，并清理首尾空格及空字符串。
`drop_duplicates().tolist()` 与空列表检查	去重后转成 Python 列表；若没有有效基因，明确报错而不是继续分析。

背景决定“比什么更集中”。

- 本例用 Ensembl BioMart 的人类数据集 `hsapiens_gene_ensembl`。
- 查询字段为 `external_gene_name`，即 BioMart 返回的基因名称。
- 本次下载并去重后得到 41,859 个符号，5 个输入符号都在其中。[10]
- 这是通用人类参考背景，不代表某个 RNA-seq 实验实际检测到的基因。

[10] Ensembl BioMart RESTful access: https://www.ensembl.org/info/data/biomart/biomart_restful.html

BioMart 根据 **XML** 查询返回制表符分隔的文本。

```
BIOMART_ENDPOINT = "https://www.ensembl.org/biomart/martservice"

query = (
    '<?xml version="1.0" encoding="UTF-8"?>'
    '<!DOCTYPE Query>'
    '<Query virtualSchemaName="default" formatter="TSV" '
    'header="0" uniqueRows="1">'
    '<Dataset name="hsapiens_gene_ensembl" interface="default">'
    '<Attribute name="external_gene_name"/>'
    '</Dataset></Query>'
)
response = requests.get(
    BIOMART_ENDPOINT, params={"query": query}, timeout=180
)
response.raise_for_status()
```

`params` 负责对 XML 查询进行 URL 编码；`raise_for_status()` 遇到 HTTP 错误时立即停止，避免把错误页面当成背景文件。

背景下载后先检查规模，再写入本地缓存。

```
symbols = sorted({
    line.strip()
    for line in response.text.splitlines()
    if line.strip()
})
if len(symbols) < 10_000:
    raise RuntimeError("Incomplete BioMart background")

BACKGROUND_CACHE.write_text(
    "\n".join(symbols) + "\n", encoding="utf-8"
)
```

本次得到 41,859 个不同符号，保存为 `results/human_biomart_gene_symbols.txt`。后续运行会读取同一份缓存，避免每次重新下载；背景来源和下载时间记录在 JSON 元数据中。

关键语句	解释
`if BACKGROUND_CACHE.is_file()`	已有缓存时直接读取，保持后续运行使用同一背景，不重复请求网络。
`requests.get(..., params=..., timeout=180)`	把 BioMart XML 查询作为参数发送；`timeout` 限定等待时间。
`response.raise_for_status()` 与响应内容检查	HTTP 请求失败或返回错误页面时停止，避免把错误文本当成基因列表。
`sorted({...})` 与 `len(symbols) < 10_000`	去重、排序，并用规模下限拦截明显不完整的响应；这个下限是完整性保护，不是生物学筛选标准。
`write_text()` 与 JSON 元数据	保存可复用的背景文件，并记录来源、数据集、下载时间和符号数量。

只有标识符能对上，目标集与注释库才能比较。

```
background_set = set(background_genes)
mapped_input_genes = sorted(
    set(genes) & background_set
)
print(
    f"Symbols found in background "
    f"({len(mapped_input_genes)}): "
    f"{', '.join(mapped_input_genes)}"
)
```

本次 5 个符号全部匹配。若有符号未匹配，先检查物种、大小写、旧别名和输入格式；不要把未映射的符号当成“没有 GO 功能”。本例气泡图中的基因比例分母为 5。

脚本从 **Enrichr** 查询人类基因集名称，并选取最新年份版本。

```
library_names = gp.get_library_name(organism="Human")

go_libraries = [
    newest_library(library_names, prefix)
    for prefix in GO_LIBRARY_PREFIXES
]
```

三个前缀分别对应 BP（生物过程）、MF（分子功能）和 CC（细胞组分）。本次查询选择了 `GO_Biological_Process_2026`、`GO_Molecular_Function_2026` 和 `GO_Cellular_Component_2026`。脚本按年份匹配，不把某个固定年份写死。

每个 **GO** 库会被读取为“条目到基因列表”的字典。

```
gene_sets = gp.get_library(
    name=library_name,
    organism="Human",
    min_size=0,
    max_size=1_000_000,
)
snapshot_path = save_gmt(library_name, gene_sets)
```

字典中每个 GO 条目对应一组基因符号。脚本把三个下载版本保存为 GMT 文件，便于追溯本次结果使用的注释内容。Enrichr 负责提供基因集名称和数据；之后的检验使用本地保存的基因集。[11]

[11] GSEApy 关于基因集读取和富集分析的说明：https://gseapy.readthedocs.io/en/latest/run.html。

关键语句	解释
`re.compile(...)` 与 `fullmatch()`	检查库名是否以四位年份结尾，并且整个名称符合预期格式。
`candidates.append((year, name))`	把年份和库名放在一起，便于按年份比较。
`return max(candidates)[1]`	取年份最大的库名；这样脚本不必把某个年份固定写死。
`gp.get_library(...)`	下载所选 GO 库，得到“GO 条目名称到基因符号列表”的字典。
`save_gmt(...)`	把每个条目和对应基因写入 GMT 文件；保存快照后，结果可追溯到本次使用的注释版本。

目标基因、**GO** 字典和背景集一起传给 **GSEApy**。

```
analysis = gp.enrichr(
    gene_list=genes,
    gene_sets=gene_sets,
    organism="human",
    background=background_genes,
    outdir=None,
    cutoff=1.0,
    no_plot=True,
)
one_library = analysis.results.copy()
one_library["Gene_set"] = library_name
```

这里的 gene_sets 是已下载的 Python 字典，因此 GSEApy 用给定背景在本地进行超几何检验。若只传 Enrichr 库名称，background 不会以同样方式用于本地检验。使用字典或 GMT 文件，才能明确指定自定义背景。[12]

[12]GSEApy 文档说明：自定义 dict/GMT 基因集支持显式背景；Enrichr 在线库名称使用另一条分析路径。参见 https://gseapy.readthedocs.io/en/latest/run.html。

参数或返回值	本次分析中的含义
`gene_list=genes`	5 个输入基因构成的目标集合。
`gene_sets=gene_sets`	当前 GO 库的本地字典；每个条目对应一组基因。
`background=background_genes`	显式传入人类参考背景，定义“随机抽取”的基因范围。
`cutoff=1.0`	不在函数内部按显著性截掉结果；之后另存全部结果并按校正后 p 值筛选。
`no_plot=True`	不让 GSEApy 另画默认图；本课的气泡图由单独脚本生成。
`analysis.results` 与 `Gene_set`	读取本次库的结果表，并标明每行来自 BP、MF 或 CC 的哪个库。

每个 **GO** 条目都对应一次“抽球”检验。

符号	本实验中的含义
N	背景集中的基因数，本次为 41,859
m	背景集中注释到当前 GO 条目的基因数
k	能映射到背景集的目标基因数，本次为 5
x	目标集中命中当前 GO 条目的基因数

检验计算在背景中随机抽取 k 个基因时，至少观察到 x 个命中的上尾概率。GSEApy 对每个 GO 库中的条目计算原始 p 值，并用 Benjamini–Hochberg 方法校正多重检验。

先保留所有返回条目，再单独筛选校正后 $p < 0.05$。

```
results.to_csv(
    all_results_path, index=False, encoding="utf-8-sig"
)

adjusted_p = pd.to_numeric(
    results["Adjusted P-value"], errors="coerce"
)
significant = results.loc[adjusted_p < 0.05].copy()
significant.to_csv(
    significant_path, index=False, encoding="utf-8-sig"
)
```

`cutoff=1.0` 保留本次返回的全部条目；显著性由后面的 $q < 0.05$ 筛选决定。校正值在 BP、MF、CC 三个库中分别计算。

关键语句	解释
`pd.concat(result_frames, ignore_index=True)`	把 BP、MF、CC 三个库的结果表纵向合并。
`results.to_csv(...)`	保存所有返回条目；完整结果保留，便于复查和重新作图。
`pd.to_numeric(...)` 与 `results.loc[adjusted_p < 0.05]`	将校正后 p 值转成数值并筛选；检验与校正在 GSEApy 中已完成。
`run_metadata` 与 `json.dumps(...)`	记录输入基因、背景、GO 库快照、软件版本、统计方法和显著性标准。
`if __name__ == "__main__": main()`	只有直接运行该文件时才启动完整流程；导入其中函数时不会自动下载或分析。

输入的 **5** 个基因全部匹配，脚本返回 **51** 个 **GO** 条目。

GO 分支	返回条目数	调整后 $p < 0.05$
BP 生物过程	38	38
MF 分子功能	7	7
CC 细胞组分	6	3
合计	51	48

结果写入 `GO_enrichment_all_terms.csv` 与 `GO_enrichment_adjusted_p_lt_0.05.csv`。输入、背景、GO 库和软件版本记录在 `run_metadata.json`。

同一组基因会命中多个相关 **GO** 条目。

GO 条目	命中数/条目基因数	调整后 p 值	输入中的命中基因
金属内肽酶活性 *metalloendopeptidase activity*（GO:0004222）	3/104	9.50×10^{-7}	ADAM12、ADAMTS16、ADAMTS18
蛋白水解 *proteolysis*（GO:0006508）	3/358	2.33×10^{-4}	ADAM12、ADAMTS16、ADAMTS18
脂肪酸代谢过程 *fatty acid metabolic process*（GO:0006631）	2/135	6.69×10^{-4}	ABCB11、AASDH
细胞外基质组织 *extracellular matrix organization*（GO:0030198）	2/192	1.13×10^{-3}	ADAMTS16、ADAMTS18
胆汁酸跨膜转运蛋白活性 *bile acid transmembrane transporter activity*（GO:0015125）	1/18	3.01×10^{-3}	ABCB11

3/104 的分母是该 GO 条目的基因集大小；Gene ratio 的分母是本次 5 个输入基因。

显著条目主要由两组输入基因贡献。

- **ADAM12**、**ADAMTS16**、**ADAMTS18** 共同命中蛋白水解和金属内肽酶活性；ADAMTS16、ADAMTS18 还命中细胞外基质组织相关条目。
- **ABCB11**、**AASDH** 共同命中脂肪酸代谢相关条目；ABCB11 单独命中胆汁酸跨膜转运蛋白活性。

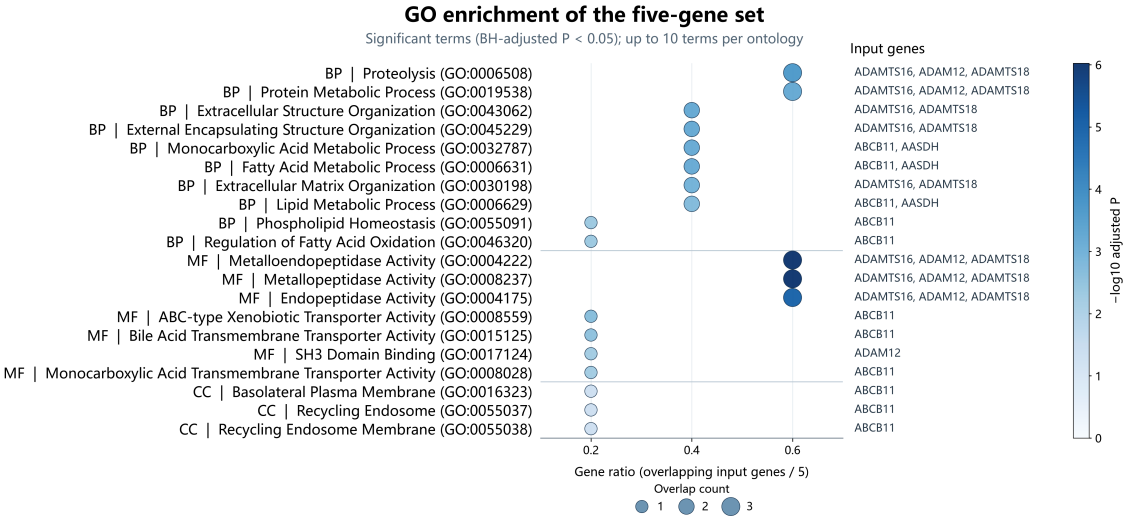
这些结果概括的是基因注释重叠。可进一步查阅 GO 条目定义和相关基因文献，判断它们是否支持同一生物学主题。

横轴、点大小和颜色编码三类结果信息。

```
overlap = data["Overlap"].str.extract(r"^(\d+)/(\d+)$")
data["Count"] = overlap[0].astype(int)
data["GeneRatio"] = data["Count"] / input_gene_count
data["MinusLog10AdjustedP"] = -np.log10(
    data["Adjusted P-value"].clip(lower=1e-300)
)

points = ax.scatter(
    data["GeneRatio"], data["Y"],
    s=65 + 85 * data["Count"],
    c=data["MinusLog10AdjustedP"], cmap="Blues",
)
```

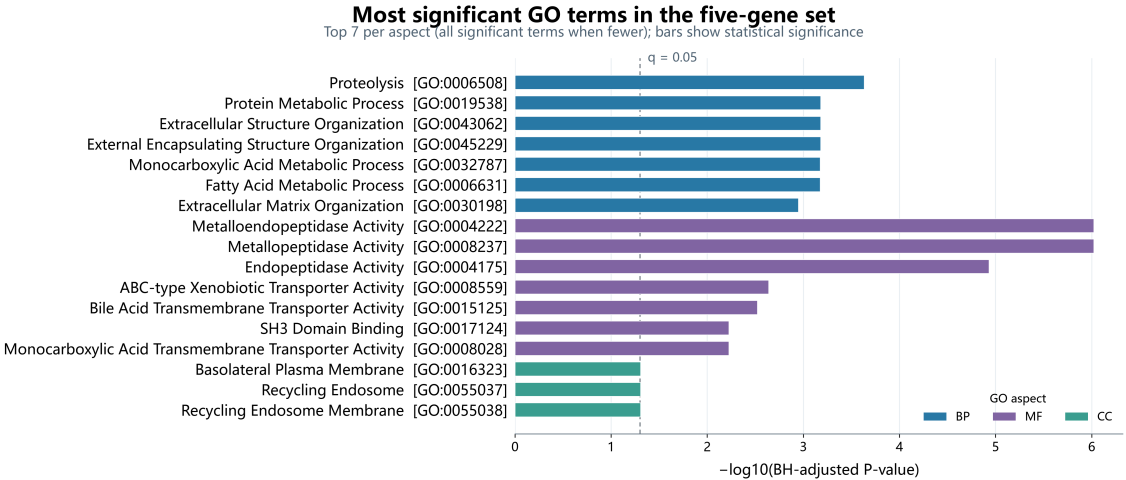
横轴是命中基因数除以 5；点越大，命中基因越多；颜色越深，调整后 p 值越小。每个本体最多展示 10 个显著条目，按调整后 p 值排序。右列列出命中的输入基因。

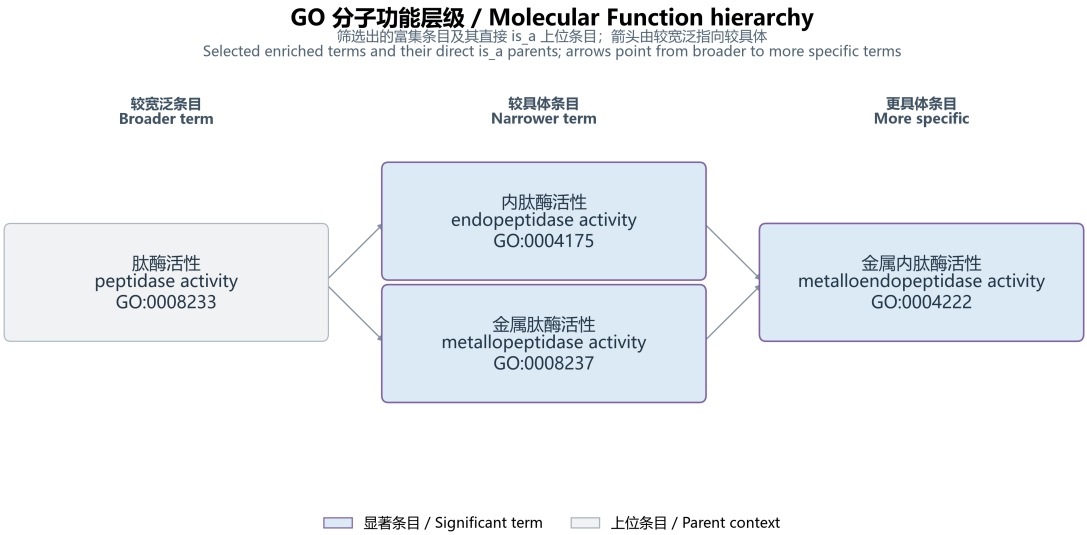

GO enrichment of the five-gene set
Significant terms (BH-adjusted P < 0.05); up to 10 terms per ontology
BP | Proteolysis (GO:0006508)
BP | Protein Metabolic Process (GO:0019538)
BP | Extracellular Structure Organization (GO:0043062)
BP | External Encapsulating Structure Organization (GO:0045229)
BP | Monocarboxylic Acid Metabolic Process (GO:0032787)
BP | Fatty Acid Metabolic Process (GO:0006631)
BP | Extracellular Matrix Organization (GO:0030198)
BP | Lipid Metabolic Process (GO:0006629)
BP | Phospholipid Homeostasis (GO:0055091)
BP | Regulation of Fatty Acid Oxidation (GO:0046320)
MF | Metalloendopeptidase Activity (GO:0004222)
MF | Metallopeptidase Activity (GO:0008237)
MF | Endopeptidase Activity (GO:0004175)
MF | ABC-type Xenobiotic Transporter Activity (GO:0008559)
MF | Bile Acid Transmembrane Transporter Activity (GO:0015125)
MF | SH3 Domain Binding (GO:0017124)
MF | Monocarboxylic Acid Transmembrane Transporter Activity (GO:0008028)
CC | Basolateral Plasma Membrane (GO:0016323)
CC | Recycling Endosome (GO:0055037)
CC | Recycling Endosome Membrane (GO:0055038)
Input genes
ADAMTS16, ADAM12, ADAMTS18
ADAMTS16, ADAM12, ADAMTS18
ADAMTS16, ADAMTS18
ADAMTS16, ADAMTS18
ABCB11, AASDH
ABCB11, AASDH
ADAMTS16, ADAMTS18
ABCB11, AASDH
ABCB11
ABCB11
ADAMTS16, ADAM12, ADAMTS18
ADAMTS16, ADAM12, ADAMTS18
ADAMTS16, ADAM12, ADAMTS18
ABCB11
ABCB11
ADAM12
ABCB11
ABCB11
ABCB11
ABCB11
0.2
0.4
0.6
Gene ratio (overlapping input genes / 5)
Overlap count
1
2
3
−log10 adjusted P
6
5
4
3
2
1
0

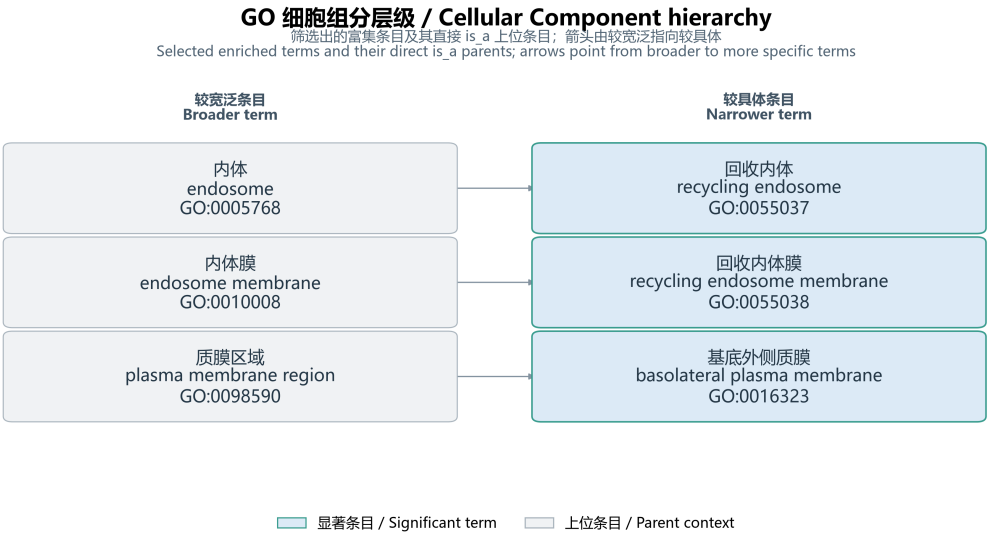
柱长按 **Benjamini–Hochberg** 校正后的 p 值计算。

```
score = -np.log10(
    plot_data["Adjusted P-value"].clip(lower=1e-300)
)
ax.barh(
    y, score,
    color=[ASPECT_COLORS[c] for c in plot_data["AspectCode"]]
)
ax.axvline(-np.log10(0.05), linestyle="--")
```

虚线标出 $q = 0.05$。每个本体按校正后 p 值选取前 7 项；条目不足 7 项时全部显示。本例 MF 和 CC 分别显示 7 项与 3 项。


Most significant GO terms in the five-gene set
Top 7 per aspect (all significant terms when fewer); bars show statistical significance
q = 0.05
Proteolysis [GO:0006508]
Protein Metabolic Process [GO:0019538]
Extracellular Structure Organization [GO:0043062]
External Encapsulating Structure Organization [GO:0045229]
Monocarboxylic Acid Metabolic Process [GO:0032787]
Fatty Acid Metabolic Process [GO:0006631]
Extracellular Matrix Organization [GO:0030198]
Metalloendopeptidase Activity [GO:0004222]
Metallopeptidase Activity [GO:0008237]
Endopeptidase Activity [GO:0004175]
ABC-type Xenobiotic Transporter Activity [GO:0008559]
Bile Acid Transmembrane Transporter Activity [GO:0015125]
SH3 Domain Binding [GO:0017124]
Monocarboxylic Acid Transmembrane Transporter Activity [GO:0008028]
Basolateral Plasma Membrane [GO:0016323]
Recycling Endosome [GO:0055037]
Recycling Endosome Membrane [GO:0055038]
GO aspect
BP
MF
CC
0
1
2
3
4
5
6
−log10(BH-adjusted P-value)


GO 分子功能层级 / Molecular Function hierarchy
筛选出的富集条目及其直接 is_a 上位条目；箭头由较宽泛指向较具体
Selected enriched terms and their direct is_a parents; arrows point from broader to more specific terms
较宽泛条目
Broader term
较具体条目
Narrower term
更具体条目
More specific
肽酶活性
peptidase activity
GO:0008233
内肽酶活性
endopeptidase activity
GO:0004175
金属肽酶活性
metallopeptidase activity
GO:0008237
金属内肽酶活性
metalloendopeptidase activity
GO:0004222
显著条目 / Significant term
上位条目 / Parent context


GO 细胞组分层级 / Cellular Component hierarchy
筛选出的富集条目及其直接 is_a 上位条目；箭头由较宽泛指向较具体
Selected enriched terms and their direct is_a parents; arrows point from broader to more specific terms
较宽泛条目
Broader term
较具体条目
Narrower term
内体
endosome
GO:0005768
回收内体
recycling endosome
GO:0055037
内体膜
endosome membrane
GO:0010008
回收内体膜
recycling endosome membrane
GO:0055038
质膜区域
plasma membrane region
GO:0098590
基底外侧质膜
basolateral plasma membrane
GO:0016323
显著条目 / Significant term
上位条目 / Parent context

主脚本完成背景下载与富集；作图脚本读取 **CSV** 后生成图。

```
.\.venv312\Scripts\python.exe go_enrichment_gseapy.py
.\.venv312\Scripts\python.exe plot_go_enrichment.py
```

文件	内容
GO_enrichment_all_terms.csv	三类 GO 库返回的全部条目
GO_enrichment_adjusted p_lt_0.05.csv	调整后 $p < 0.05$ 的条目
GO_enrichment_bubble_plot.png	气泡图与柱状图
GO_enrichment_bar_plot.png	
GO_enrichment_hierarchy_ BP/MF/CC.png	三类 GO 层级图
.svg / .pdf 与绘图 CSV	图形矢量版本及作图数据
GO_*.gmt、go-basic.obo.gz	富集基因集与层级关系快照
run_metadata.json go_basic_ontology_metadata.json	输入、背景、版本与来源记录

背景基因集应与研究设计一致。

- 本例使用 Ensembl BioMart 返回的 41,859 个基因符号作为通用背景。
- 如何考虑特定需要？

[Jingbo] 如果更换一个场景，不是 GO 富集，而是 HPO 富集，或者 RTO 富集，你会如何进行呢？

