

Prohemsyne

Reading, Reflection, Internalization, Improvisation

User's Guide

Chien-Hsin Lin

March 8, 2026

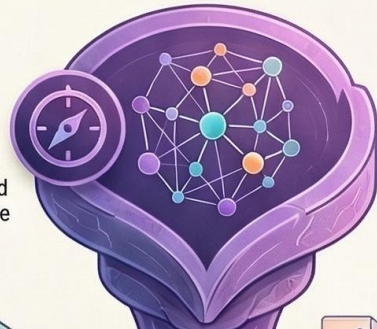
Prohemsyne:

Your Divine Workflow for Academic Research Intelligence

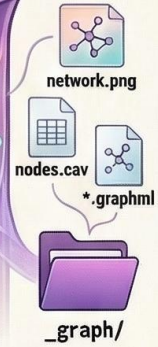
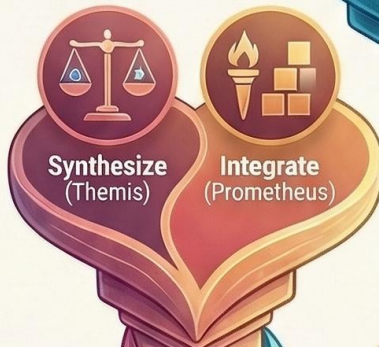


Map Mode (Atlas): The Topography Pillar

Maps the 'logical landscape' of a field by visualizing keyword co-occurrence and hidden connections.

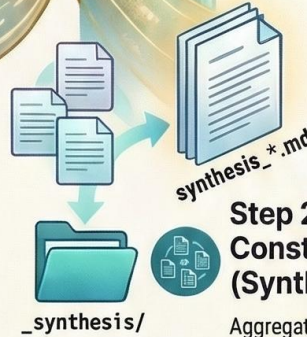


Synthesize & Integrate: The Order & Vision Pillars



Step 3: Framework Building (Integrate Mode)

Combines multiple synthesis reports into a complete thesis draft (Chapter 1 & 2) in the `_integrate` folder.



Step 2: Construct Synthesis (Synthesize Mode)

Aggregates notes by tag to produce `synthesis_*.md` reports summarizing single focal constructs.



The Research Evolution Workflow

Step 1: Extraction (Note Mode)

Processes single PDFs into structured `.md` notes and merged PDFs in the `_note` and `_merged` folders.



The Four Divine
Modules of Intelligence



Note (Mnemosyne): The Memory Pillar

Faithfully extracts paper essence into structured Markdown notes without bias or hallucinations.



license

Prohemsyne © License & Disclaimer

Content License

Content (Documentation / Instructions / Prompts / Text) © 2025–2026 Chien-Hsin Lin

This content is licensed under **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**:

- **Attribution Required:** You must give appropriate credit, provide a link to the license, and indicate if changes were made.
- **NonCommercial Use Only:** You may not use the material for commercial purposes.
- **ShareAlike:** If you remix, transform, or build upon the material, you must distribute your contributions under the same license.

Full License Text: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Software License

Software (Code / app.py) © 2025–2026 Chien-Hsin Lin

All Rights Reserved

Terms and Conditions

1. **Restrictions:** No copying, modification, distribution, or commercial use is permitted without the author's prior written permission.
 2. **No Redistribution / No Sublicense:** This license is personal, non-transferable, and non-sublicensable. You may not share, distribute, publish, or provide the Software to any third party without the author's prior written permission.
 3. **Permitted Use:** This software is provided for **research and teaching purposes only**.
 4. **Prohibited Use:** This software is **NOT** intended for:
 - Medical applications
 - Safety-critical systems
 - Other high-risk applications
 - Any commercial use without explicit written permission
 5. **License Requests:** For licensing inquiries, please contact the author.
-

Disclaimer

THIS SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

中文版本

內容授權

內容（文件 / 說明 / 提示詞 / 文字） © 2025–2026 Chien-Hsin Lin

本內容採用 創用 CC 姓名標示-非商業性-相同方式分享 4.0 國際授權條款 (CC BY-NC-SA 4.0)：

- **必須標示姓名**：您必須標示原作者姓名，提供授權條款連結，並註明是否變更內容。
- **禁止商業使用**：您不得將本內容用於商業目的。
- **相同方式分享**：若您改作、轉換本內容，或依本內容建立新作品，您必須依本授權條款或相同條款散布您的貢獻。

完整授權條款：https://creativecommons.org/licenses/by-nc-sa/4.0/deed.zh_TW

軟體授權

軟體（程式碼 / app.py） © 2025–2026 Chien-Hsin Lin

版權所有

條款與條件

1. **限制**：未經作者書面許可，不得複製、修改、散布或用於商業用途。
2. **禁止轉傳 / 禁止轉授權**：本授權為個人專屬、不得轉讓、不得轉授權。未經作者事前書面同意，不得將本軟體提供、分享、散布或交付任何第三人。
3. **允許用途**：本軟體僅供**研究與教學用途**。
4. **禁止用途**：本軟體**不得用於**：
 - 醫療應用
 - 安全關鍵系統

- 其他高風險應用
- 任何未經明確書面許可的商業用途

5. **授權申請**：如需授權，請聯繫作者。

免責聲明

本軟體以「現狀」提供，不提供任何明示或暗示的擔保，包括但不限於適售性、特定用途適用性及未侵權的擔保。

在任何情況下，作者均不對任何索賠、損害或其他責任負責，無論是基於合約、侵權或其他原因，因使用本軟體或與本軟體相關的使用或其他交易而產生。

Contact Information

Contact: Chien-Hsin Lin (prohemsyne@gmail.com)

License & Copyright Last Updated: March 3, 2026.

about

關於 Prohemsyne ©

Current version: Janus-MMXXVI-III-VII

本系統以古希臘泰坦神祇為名，將學術研究流程拆解為四層智慧的演進。取法雙面神 Janus 凝視往復之姿：於文獻回溯中固守研究根基，於邏輯轉譯間洞見學術裂隙，為您的研究推開通往理論前沿的智慧之門。

Prohemsyne 發音 /proʊ'ɛmzɪni/ (Pro-EM-zi-ni)，h 為優雅靜音，重音在中間的 EM，結尾承襲 Mnemosyne 之韻律。

Mnemosyne (PaperNote)

- **神格**：記憶女神，繆思之母。
 - **核心**：忠實記錄。承襲女神守護宇宙的記憶，本模組負責精確萃取論文精華，不帶偏見地轉化為結構化筆記。
 - **版本**：Mnemosyne-MMXXVI-III-VII
-

Themis (PaperSynthesize)

- **神格**：正義與秩序女神，手持天平。
 - **核心**：邏輯權衡，定義校準。本模組在多篇文獻間建立嚴謹的邏輯網路與變數對照，建立可追溯的對照框架，以嚴格判準維持論證秩序。
 - **版本**：Themis-MMXXVI-III-VII
-

Prometheus (PaperIntegrate)

- **神格**：先見者與火種帶來者。
 - **核心**：開創未來。透過高階抽象推理與流暢的學術敘事，本模組將零散資訊鍛造成全新的研究架構，為學術探索點燃火種。
 - **版本**：Prometheus-MMXXVI-III-VII
-

Atlas (PaperGraph)

- **神格**：擎天泰坦，結構與座標之神。
 - **核心**：空間測繪，俯瞰研究領域的邏輯地形。本模組將抽象的變數關係轉換為具象的空間圖譜，在龐雜的文獻叢林中撐起清晰的知識座標，為學術導航提供全景式的結構地圖。
 - **版本**：Atlas-MMXXVI-II-XXV
-

About Last Updated: March 7, 2026.

getting_started

Prohemsyne © Getting Started

Prohemsyne © 是一套以 Python 開發、由林建信老師設計的學術論文閱讀與教學輔助系統，專門用來支援「閱讀、教學、研究」的完整流程。系統強調學習歷程中的 **Reading** → **Reflection** → **Internalization** → **Improvisation**，協助使用者將論文知識從被動閱讀轉化為可內化、可再運用、可即興發揮的教學與研究資源。

PaperNote (structured extraction)

透過 Streamlit 網頁介面搭配 Google Gemini API 運作，使用者只要在瀏覽器中開啟系統並上傳 PDF 論文，即可依照預先設計的 Stage 1–8 流程，自動完成論文內容擷取並結構化整理重點、提出課堂討論題與參考解答設計，以及延伸閱讀與研究作業說明撰寫，最後彙整為一份附有封面的完整 TXT 報告。完成的筆記會依統一檔名規則儲存於本機 `_note` 資料夾，並自動同步到 Google Drive 專用目錄，方便備份與分享。

Papernote 在技術層面結合多種 Python 套件完成 PDF 文字擷取、OCR（掃描論文辨識）、Markdown 處理與 PDF 合併等工作。系統會將原始 PDF 複製並重新命名後歸檔至 `_paper`（手動上傳的原始檔案保留，Auto Flash/Random Flash 的檔案會從 `_temp` 刪除），再將 Stage 8 報告由 Markdown 轉成 Note PDF，並與原始論文合併為一份 Merged PDF 存放於 `_merged`，同樣自動上傳至雲端。

除了單篇分析外，Papernote 亦提供 Auto Flash（從 `_temp` 批次讀取最新數篇 PDF）與 Random Flash（隨機抽一篇 PDF）模式，協助使用者快速清空待讀論文清單，將「堆積如山的 PDF」轉化為「結構化、可教學、可再利用的知識資產」。

PaperSynthesize (intra-construct vertical synthesis)

對單一核心構念進行文獻綜合分析，整合多篇論文中關於該構念的研究發現，產出包含變數調和、逐篇摘要、跨研究綜整與研究架構的綜合報告。

PaperIntegrate (inter-construct lateral integration)

整合多個構念的 Synthesis 報告，構建完整的研究模型，並自動撰寫論文的 Chapter 1（緒論）與 Chapter 2（文獻回顧）。

PaperGraph (knowledge cartography)

分析所有論文筆記的關鍵字共現關係，自動產出網絡圖（PNG）與結構化資料（`nodes.csv`、`edges.csv`、`network.graphml`），協助使用者視覺化知識結構、探索文獻之間的隱藏連結。GraphML 檔案可直接在 Gephi Lite 開啟，CSV 檔案可匯入 Gephi Desktop 進行更專業的網絡分析與視覺化。

以下為 Prohemsyne 的安裝步驟。

第零階段：選擇執行方式

Prohemsyne 提供兩種執行方式，請依您的需求選擇：

方式 A：使用預編譯執行檔（Papernote.exe） - 推薦一般使用者

適合對象：不熟悉程式開發、希望快速開始使用的使用者

優點：

- 無需安裝 Python 環境
- 雙擊即可執行，操作簡單
- 所有相依套件已打包完成

操作說明：直接跳到「🔑 第一階段」繼續閱讀即可。

方式 B：使用 Python 執行原始碼（app.py） - 進階使用者適用


適合對象：熟悉 Python 環境、需要自訂功能或除錯的進階使用者

優點：

- 可以查看與修改原始碼
- 方便進行功能擴充與客製化
- 便於追蹤與回報問題

前置需求：需要先安裝 Python 3.9 以上版本及相關套件

Python 環境設定（僅限方式 B 使用者）

 **提示：**若您選擇「方式 A：使用 Papernote.exe」，請直接跳過本節，前往「🔑 第一階段」。

步驟 1：下載並安裝 Python

1) 前往 Python 官方網站下載

請前往 Python 官方網站下載最新的穩定版本：

- 下載連結：<https://www.python.org/downloads/>

- **建議版本**：Python 3.9 或以上版本（建議 Python 3.10 或 3.11）

2) 執行安裝程式

1. 下載完成後，執行安裝程式（例如：`python-3.11.x-amd64.exe`）
2. **重要**：在安裝畫面的最下方，務必勾選 **"Add Python to PATH"**（將 Python 加入系統路徑）
3. 點擊 **"Install Now"**（建議使用預設安裝路徑）
4. 等待安裝完成

3) 驗證安裝是否成功

在 **Command Prompt (命令提示字元)** 中驗證：

1. 按下 `Win + R`，輸入 `cmd`，按 `Enter` 開啟命令提示字元
2. 輸入以下指令並按 `Enter`：

```
python --version
```

3. 若顯示類似 `Python 3.11.x` 的版本資訊，表示安裝成功

在 **PowerShell** 中驗證：

1. 按下 `Win + X`，選擇「Windows PowerShell」
2. 輸入以下指令並按 `Enter`：

```
python --version
```

3. 若顯示版本資訊，表示安裝成功

⚠ 注意：若出現「'python' 不是內部或外部命令」的錯誤訊息，表示 Python 未正確加入系統路徑，請重新執行安裝程式並確認勾選「Add Python to PATH」。

步驟 2：取得 Prohemsyne 原始碼

請確保您已取得包含 `app.py` 的完整程式資料夾，並將其放置在適當位置（例如：`C:\papernote\`）。

資料夾內應包含：

- `app.py`（主程式檔案）
- `_doc/` 資料夾（內含 `requirements.txt` 相依套件清單及系統說明文件）

步驟 3：安裝相依套件

開啟命令提示字元 (Command Prompt) 或 PowerShell，並切換到程式所在的資料夾。

在 Command Prompt 中操作：

```
cd C:\papernote
python -m pip install --upgrade pip
pip install -r _doc\requirements.txt
```

在 PowerShell 中操作：

```
cd C:\papernote
python -m pip install --upgrade pip
pip install -r _doc\requirements.txt
```

說明：

- `cd C:\papernote`：切換到程式所在的資料夾 (請依您的實際路徑修改)
- `python -m pip install --upgrade pip`：更新 pip 套件管理工具到最新版本
- `pip install -r _doc\requirements.txt`：自動安裝 `_doc\requirements.txt` 中列出的所有相依套件

安裝過程

安裝過程可能需要數分鐘，系統會自動下載並安裝以下套件 (依 `_doc\requirements.txt` 內容而定)：

- `streamlit`：網頁介面框架
- `google-generativeai`：Google Gemini API 客戶端
- `PyPDF2`、`pdfplumber`、`pdf2image`：PDF 處理工具
- `pytesseract`：OCR 文字辨識
- `Pillow`：圖片處理
- `fpdf2`、`markdown-pdf`：PDF 生成工具
- `google-auth`、`google-auth-oauthlib`、`google-api-python-client`：Google Drive API
- 其他相依套件

安裝完成後，會顯示 `Successfully installed...` 訊息。

系統依賴組件安裝：Poppler (Windows 專用)

由於系統中的 `pdf2image` 套件僅是 Poppler 的 Python 封裝，Windows 使用者必須手動安裝 **Poppler 二進位程式集**，否則系統將無法將 PDF 轉換為圖片進行分析。

1) 下載 Poppler

請前往 [poppler-windows 釋出頁面](#) 下載最新的 `.zip` 壓縮檔（例如 `Release-25.xx.x-x.zip`）。

2) 解壓縮與放置

將下載的壓縮檔解壓縮到一個固定路徑。建議路徑為：`C:\poppler`。

3) 設定系統環境變數 (PATH)

這是最關鍵的一步，讓系統能找到 Poppler 指令：

1. 在 Windows 搜尋列輸入「**編輯系統環境變數**」並開啟。
2. 點擊「**環境變數**」按鈕。
3. 在下方的「系統變數」區域找到 **Path**，選取後點擊「**編輯**」。
4. 點擊右側的「**新增**」，輸入你解壓縮資料夾內的 **bin** 路徑。
 - 範例路徑：`C:\poppler\Library\bin` (請依實際資料夾名稱為準)。
5. 連續點擊「**確定**」關閉所有視窗。

4) 驗證安裝

1. 按下 `Win + R`，輸入 `cmd` 開啟命令提示字元。
2. 輸入指令：`pdftoppm -h` 並按 `Enter`。
3. 若出現一長串說明文字而非「不是內部或外部命令」，表示安裝成功！

⚠ 注意：完成設定後，請務必**重新啟動**您的命令提示字元 (CMD) 或 PowerShell，環境變數才會生效。

步驟 5：啟動程式

安裝完成後，即可使用以下指令啟動 Prohemsyne：

在 Command Prompt 中啟動：

```
cd C:\papernote
streamlit run app.py
```

在 PowerShell 中啟動：

```
cd C:\papernote
streamlit run app.py
```

說明：

- `streamlit run app.py`：使用 Streamlit 框架執行 `app.py` 主程式

啟動後的畫面

1. 執行指令後，終端機（Command Prompt 或 PowerShell）會顯示啟動訊息，類似：

```
You can now view your Streamlit app in your browser.
```

```
Local URL: http://localhost:8501
```

```
Network URL: http://192.168.x.x:8501
```

2. 系統會自動開啟瀏覽器，並導向 `http://localhost:8501`，顯示 Prohemsyne 的操作介面
3. **重要**：請保持終端機視窗開啟，不要關閉它，否則程式會停止運作

停止程式

當您想要關閉程式時，請在終端機視窗中按下 `Ctrl + C`，系統會詢問是否終止，輸入 `Y` 並按 Enter 即可停止程式。

Instructions 來源（僅限方式 B，進階）

執行 `app.py` 時，Note / Synthesize / Integrate 使用的 AI 指令（instructions）來源由環境變數 `PAPERNOTE_MODE` 決定：

- **未設定 `PAPERNOTE_MODE`，或設為 `prod`（**prod 模式**）**：程式使用已嵌入在 `app.py` 內的三份 instructions（`papernote_instructions`、`paperanalyze_instructions`、`paperintegrate_instructions`），無需外部 `.txt` 檔案即可運作。
- **設為 `PAPERNOTE_MODE=dev`（**dev 模式**，開發版）**：程式改為**優先讀取**程式目錄下的三個外部 `.txt` 檔案（`papernote_instructions.txt`、`paperanalyze_instructions.txt`、`paperintegrate_instructions.txt`）；若讀取失敗或檔案不存在，則自動退回使用 `app.py` 內嵌的 instructions。

何時會是 dev 還是 prod？

- **從 `start_papernote.bat` 進入**：bat 裡有設定 `set PAPERNOTE_MODE=dev`，所以一定是 **dev 模式**，會優先讀取外部三個 `.txt`，讀不到才用內嵌。
- **在終端機直接輸入 `streamlit run app.py`**：沒有設定環境變數，所以是 **prod 模式**，只會用 `app.py` 裡已內嵌的三份 instructions，不會去讀外部 `.txt`。

程式運行中如何確認是 dev 還是 prod？

執行時可從**命令視窗（黑視窗）**的輸出訊息判斷目前使用的模式與指令來源。當您切換到 Note / Synthesize / Integrate 並觸發對應流程時，視窗內會印出下列其中一種：

- **DEV 模式**（使用外部 .txt）：
[Note] 使用外部 `papernote_instructions.txt` (DEV 模式)
[Synthesize] 使用外部 `paperanalyze_instructions.txt` (DEV 模式)
[Integrate] 使用外部 `paperintegrate_instructions.txt` (DEV 模式)
- **PROD 模式**（使用內嵌 instructions）：
[Note] 使用內嵌 `instructions` (PROD 模式)
[Synthesize] 使用內嵌 `instructions` (PROD 模式)
[Integrate] 使用內嵌 `instructions` (PROD 模式)

給開發者的提醒：若您直接 `streamlit run app.py` (prod 模式)，但 `app.py` 內嵌的 `instructions` 不完整或過舊，程式可能執行到一半報錯或產出異常。發佈給他人使用時，請確保內嵌的 `instructions` 已更新完整；若自己開發、想用外部 .txt 修改指令，請改由 **bat 進入** (dev 模式)。

此設定僅影響「指令內容的來源」，不影響一般操作流程；一般使用者無需設定，維持預設即可。

完成 Python 環境設定

✅ 完成以上步驟後，您已成功設定 Python 環境並啟動 Prohemsyne。

接下來請繼續閱讀「🔑 第一階段：取得 Gemini API Key」，完成後續的設定步驟（API Key、Google Drive 同步、字型設定等），這些步驟對於 `Papernote.exe` 與 `app.py` 執行方式都是相同的。

🔑 第一階段：取得 Gemini API Key (AI 的大腦)

Papernote 需要 Google 的 Gemini 模型來閱讀論文。您需要申請一把免費的鑰匙 (API Key)。

步驟

1. 前往 Google AI Studio：<https://aistudio.google.com/>
2. 點擊左上角的登入 (Sign in)，使用您的 Google 帳號登入
3. 點擊左側選單的 "Get API key" (鑰匙圖示)
4. 點擊 "Create API key" 按鈕
5. 選擇 "Create API key in new project" (在建立新專案中產生金鑰)
6. 系統會產生一串以 AIza 開頭的長代碼，請點擊旁邊的 "Copy" (複製) 按鈕，並將其暫時貼在記事本上備用

如何在軟體中使用？

- 當您開啟 Papernote.exe 時，如果系統偵測不到金鑰，會在畫面最上方顯示一個輸入框：「請輸入你的 Gemini API key」
- 將剛剛複製的 AIza... 代碼貼上並按 Enter，系統就會啟動並開始運作
- **註：**這個金鑰只會暫存在您的瀏覽器中，關閉程式後就會清除，下次開啟需重新輸入。若想永久儲存，可參閱進階設定

📁 第二階段：設定 Google Drive 同步（備份報告）

💡 提示：此為選用功能 (Optional)

如果您不需要將筆記自動備份到 Google Drive，您可以完全忽略 `credentials.json` 和 `token.json` 這兩個檔案，直接跳過本階段。軟體會自動變成一個「純單機版」的 AI 論文助手，核心功能（AI 分析、報告生成、本地存檔）依然強大且完全不受影響。

若要啟用雲端同步功能，資料夾中需要有 `credentials.json` 檔案。請繼續閱讀以下步驟：

Papernote 支援將分析報告自動備份至您的 Google Drive。若要啟用此功能，資料夾中需要有 `credentials.json` 檔案。

1. 取得 `credentials.json`（應用程式身分證）

1. 前往 Google Cloud Console：<https://console.cloud.google.com/>
2. 建立一個新專案 (New Project)，名稱隨意 (如 MyPapernote)
3. 在左側選單找「API 和服務」>「已啟用的 API」，點擊「+ 啟用 API 和服務」，搜尋並啟用 **"Google Drive API"**
4. 設定「OAuth 同意畫面」：
 - User Type 選「外部 (External)」
 - 填寫 App 名稱 (Papernote) 和 Email
 - **重要：**最後務必點擊 **"發布應用程式 (Publish App)"**，否則會出現 403 錯誤
5. 下載憑證：
 - 到「憑證 (Credentials)」頁面 >「+ 建立憑證」>「OAuth 用戶端 ID」
 - 應用程式類型選「桌面應用程式」
 - 建立後下載 JSON 檔，並**重新命名為** `credentials.json`

2. 安裝檔案

請確保您的資料夾結構如下：

```
graph TD; Root[C:\Papernote\] --- File[Papernote.exe (或 app.py + 其他程式檔案)];
```

```
└─ credentials.json <-- 把剛剛下載的檔案放這裡 (若只需單機版則免放)
└─ token.json <-- 見以下說明 (首次登入自動產生, 若只需單機版則無)
```

3. 首次登入 (僅限有放入 credentials.json 者)

1. 執行 Papernote.exe
2. (若您有啟用雲端功能) 程式偵測到 credentials.json 但尚未登入時, 程式會自動彈出瀏覽器要求登入 Google Drive
(若您未放入該檔案, 則不會有任何彈窗, 程式將以單機模式運作)
3. 若出現「Google 尚未驗證應用程式」, 請點「進階」>「前往 Papernote (不安全)」, 並點擊「允許」
4. 登入成功後, 資料夾內會自動產生 token.json, 之後開啟軟體即會自動登入, 不再彈出視窗

第三階段：啟動與首次登入 (重要！)

1. 執行程式

重要：請將程式檔案放在可寫入的資料夾中再執行 (例如桌面、D 槽資料夾)。
請勿放在 C:\Program Files\... 等系統資料夾, 否則程式可能無法建立或寫入必需的檔案 (例如 _doc/、poppler_bin/、以及後續自動產生的 _note/、_paper/ 等資料夾)。

方式 A：使用 Papernote.exe

點擊兩下執行 Papernote.exe。

方式 B：使用 Python 執行 app.py

開啟 Command Prompt 或 PowerShell, 切換到程式所在資料夾, 執行以下指令：

```
cd C:\papernote
streamlit run app.py
```

系統會自動開啟瀏覽器並顯示操作介面。請保持終端機視窗開啟。

2. 關於黑色視窗 (Console) 的操作說明

程式啟動時, 會伴隨出現一個 **黑色背景的視窗**, 這是程式的核心控制台, **請勿關閉它**, 否則程式會停止運作。

【關鍵步驟】

若黑色視窗中出現文字詢問 (例如 Email:) 並停住不動：

👉 請點擊黑色視窗，並直接按下鍵盤上的【Enter】鍵。
程式即會繼續執行，並自動彈出網頁操作介面。

3. Google Drive 授權 (若有啟用)

(若您有放入 `credentials.json`)：程式啟動後，會自動彈出瀏覽器要求授權。

1. 若出現「Google 尚未驗證應用程式」，請點「進階」>「前往 Papernote (不安全)」，並點擊「允許」
2. 登入成功後，資料夾內會自動產生 `token.json`，之後開啟軟體即會自動登入，不再彈出視窗

⚡ 第四階段：自動化功能與資料夾結構說明

本軟體內建 **Auto Flash** (自動批次) 與 **Random Flash** (隨機抽樣) 功能，專為快速消化大量文獻設計。

📁 1. 準備工作：建立 `_temp` 資料夾 (重要)

程式會自動掃描「軟體所在位置」旁邊的 `_temp` 資料夾。請務必依照以下規則放置檔案：

1. 在存放程式檔案 (`Papernote.exe` 或 `app.py`) 的同一個資料夾內，建立一個名為 `_temp` 的子資料夾
(正確範例：若程式放在 `C:\papernote`，請建立 `C:\papernote_temp`)
2. 將您想要分析的 PDF 論文檔案，全部放入這個 `_temp` 資料夾中
3. 軟體啟動後點擊 **Auto Flash** 或 **Random Flash**，系統就會自動從這個資料夾抓取檔案進行分析

📁 2. 系統自動產出的資料夾說明

當分析完成後，Prohemsyne 會自動在程式目錄下建立以下資料夾，分別存放不同檔案：

Note 模式相關

- `_note` (筆記區)
 - 存放分析完成的 AI 筆記報告 (MD 格式)
- `_paper` (歸檔區)
 - 存放原始 PDF 論文的副本 (重新命名後)
 - 處理邏輯：
 - 手動上傳的 PDF：系統會複製一份到 `_paper/`，原始檔案保留在原位置
 - **Auto Flash / Random Flash** 的 PDF：系統會複製一份到 `_paper/`，並刪除 `_temp/` 中的原始檔案，避免重複分析
- `_merged` (整合區)
 - 存放整合後的 PDF 報告 (包含原始論文 + AI 筆記)

Synthesize 模式相關

- `_aggregate` (聚合區)
 - 存放依 tag 聚合的多篇論文筆記檔案 (作為 Synthesize 的輸入)
- `_synthesis` (綜合區)
 - 存放單一構念的文獻綜合報告 (作為 Integrate 的輸入)

Integrate 模式相關

- `_integrate` (整合區)
 - 存放完整的研究模型與論文章節檔案

Graph 模式相關

- `_graph` (知識網絡視覺區)
 - 存放完整的關鍵字網絡分析檔案 (網絡圖、GraphML、CSV、Mapping 調和關鍵字與 Embedding 再合併產出的 `keyword_mapping / keyword_embedding.json`、關鍵字網絡分析.md)

第五階段：中文 PDF 字型設定 (解決亂碼問題)

為了讓 Papernote 生成的 PDF 報告能完美顯示繁體中文，請務必依照以下步驟放入字型檔案。若未執行此步驟，PDF 中的中文可能會變成方框或亂碼。

1. 建立資料夾

在存放程式檔案 (Papernote.exe 或 app.py) 的同一個資料夾內，建立一個名為 `fonts` 的子資料夾。

2. 下載並放入字型

請準備 **Noto Sans TC (思源黑體)** 的以下 4 個檔案，並放入 `fonts` 資料夾中：

- `NotoSansTC-Regular.ttf`
- `NotoSansTC-Bold.ttf`
- `NotoSansTC-Medium.ttf`
- `NotoSansTC-Black.ttf`

字型可至 *Google Fonts* 免費下載：<https://fonts.google.com/download?family=Noto%20Sans%20TC>

總結：完整運作的資料夾結構圖

```
└─ Papernote/ (使用者自己建立)
```

—	—	—	—	Papernote.exe	<-- 方式A：預編譯執行檔
—	—	—	—	app.py	<-- 方式B：Python 原始碼主程式
—	—	—	—	credentials.json	<-- 使用者看說明書自己去 Google 申請（選用：若需雲端備份才放）
—	—	—	—	token.json	<-- 在有credentials.json情況下，首次成功登入後，自動產生
—	—	—	—	fonts/	<-- 使用者看說明書自己去下載放進去的字型檔案（若無此資料夾，中文 PDF 可能會有亂碼）
—	—	—	—	NotoSansTC-Regular.ttf	
—	—	—	—	NotoSansTC-Bold.ttf	
—	—	—	—	NotoSansTC-Medium.ttf	
—	—	—	—	NotoSansTC-Black.ttf	
—	—	—	—	_doc/	<-- 系統相關的說明檔案，點選主程式側邊欄下方的按鈕開啟這些檔案。
—	—	—	—	about.md	<-- 說明系統每個模組的開發理念
—	—	—	—	faq.md	<-- 常見問題
—	—	—	—	getting_started.md	<-- 安裝步驟與操作說明
—	—	—	—	license.md	<-- 授權與免責聲明
—	—	—	—	requirements.txt	<-- 方式B：Python 相依套件清單（程式內無法開啟）
—	—	—	—	_merged/	<-- 程式自動產生（Note 模式：放原始論文+筆記的合併 PDF 檔案）
—	—	—	—	_note/	<-- 程式自動產生（Note 模式：放筆記 .md 檔案）
—	—	—	—	_paper/	<-- 程式自動產生（Note 模式：原始在 _temp 的 PDF 檔案，會重新命名放到這裡）
—	—	—	—	_temp/	<-- 使用者自己建立，將要分析的 PDF 檔案放在這裡
—	—	—	—	_aggregate/	<-- 程式自動產生（Synthesize 模式：聚合檔案）
—	—	—	—	_synthesis/	<-- 程式自動產生（Synthesize 模式：綜合報告）
—	—	—	—	_integrate/	<-- 程式自動產生（Integrate 模式：整合報告）
—	—	—	—	_graph/	<-- 程式自動產生（Graph 模式：網絡圖、GraphML、CSV、mapping.json、embedding.json、關鍵字網絡分析.md）

側邊欄：筆記語言（EN / TC）、Gemini 模型選擇與 Global Temperature（全域創意度設定）

啟動 Prohemsyne 後，畫面左側的側邊欄會出現三組切換選項：

筆記語言切換：EN / TC

這一組是控制 Papernote 產出內容的語言，不會改變原始 PDF 的內容語言。

選項說明：

- **EN (English)**：預設，所有筆記內容全部使用英文輸出

- **TC (Traditional Chinese)**：所有筆記內容使用繁體中文

建議：

- 在開始執行 Stage 1 前先選好 EN 或 TC，整篇論文的筆記語言就會保持一致，較方便後續教學與整理。
- Prohemsyne 很多流程均以 note.md 筆記內容為判斷邏輯，程式對英文筆記內容之擷取及判斷比較穩定。

Gemini 模型選擇：Gemini 2.5 / Gemini 3 Preview

這一組是控制本次論文分析所使用的 Gemini 模型版本。

選項說明：

- **Gemini 3 Flash Preview (預設)**：以穩定性與速度為優先的版本，適合日常的論文閱讀、教學筆記整理、摘要與段落改寫等工作。多數情境下建議維持此設定，能在輸出品質、成本與等待時間之間取得較好的平衡。
- **Gemini 3.1 Pro Preview**：以推理深度與文字品質為優先的版本，適合在你想比較新一代模型的寫作風格、論證嚴謹度與回應細緻度時使用。建議用於較高難度的整合任務（例如跨文獻綜整、理論推演、框架建構、假說推導與章節撰寫），但通常會需要較長的生成時間與較高的資源成本。

實際可用情況依 Google 帳號與 API 權限而定。

Papernote 會將側邊欄的選擇套用到整個運行流程中：你在 Sidebar 選擇的模型，會用於 Stage 1–8 的所有 AI 呼叫。

Stage 7 封面的「Generated By」欄位也會顯示對應的模型名稱（例如：Generated By: Gemini 2.5 Pro）。

建議：每一篇論文「從 Stage 1 開始之前」就先選定模型版本，同一篇論文過程中不要中途切換，這樣 Stage 1–8 的內容風格與封面標示會保持一致，較方便比較不同論文、不同模型的輸出差異。

Global Temperature (全域創意度設定)

這一組是控制 AI 回應的創意度與隨機性，影響生成內容的變化性與精確度。

設定範圍與預設值

- **範圍**：0.2 ~ 1.2
- **預設值**：0.6
- **步進**：0.1

設定原則與數值差異性

Temperature 數值越高，AI 的回應越具創意與變化性，但也可能降低精確度與一致性；數值越低，回應越保守、精確、可預測，但可能較為單調。

- **低溫區 (0.2-0.4)**：適合需要高精確度、一致性的任務
 - 特點：回應穩定、可預測、重複性高
 - 適用：變數調和、結構化資料提取、需要嚴格遵循格式的任務
- **中溫區 (0.5-0.7)**：平衡創意與精確度
 - 特點：在創意與精確度間取得平衡
 - 適用：一般論文分析、摘要生成、多數日常任務
- **高溫區 (0.8-1.2)**：適合需要創意與流暢性的任務
 - 特點：回應多樣、文筆流暢、具創意性
 - 適用：文獻回顧撰寫、研究假說推導、需要流暢學術寫作的任務

各模組程序的 Temperature 設定

Note 模式

- **Stage 1-8**：使用 Sidebar 的 Global Temperature 設定（預設 0.6）
 - 使用者可透過 Sidebar slider 自行調整
 - 若未調整，所有 Stage 均使用預設值 0.6

Synthesize 模式

- **Phase 0 (Construct Harmonization)**：固定 0.2
 - 原因：需要高精確度的變數名稱統一，確保調和結果穩定一致
 - 不受 Sidebar 設定影響
- **Phase 1 (Paper-level Table)**：使用 Sidebar 的 Global Temperature 設定（預設 0.6）
 - 原因：逐篇摘要表需要平衡精確度與結構化輸出
 - 使用者可透過 Sidebar slider 自行調整
- **Phase 2 (Cross-study Synthesis)**：使用 Sidebar 的 Global Temperature 設定（預設 0.6）
 - 原因：跨研究綜整需要平衡精確度與整合性分析
 - 使用者可透過 Sidebar slider 自行調整
- **Phase 3 (Research Framework And Hypotheses)**：使用 Sidebar 的 Global Temperature 設定（預設 0.6）
 - 原因：研究架構與假說推導需要平衡邏輯嚴謹性與創意性
 - 使用者可透過 Sidebar slider 自行調整

Integrate 模式

- **智能探索 (Smart Explore)**：固定 0.8
 - 原因：需要創意性來探索構念組合的可能性與新穎性
 - 不受 Sidebar 設定影響

- **Phase 1 (Construct Harmonization)：**固定 0.2
 - 原因：構念整合需要精確的變數名稱統一與對應，避免過度創意導致整合不一致
 - 不受 Sidebar 設定影響
- **Phase 2 (Framework Construction)：**使用 Sidebar 的 Global Temperature 設定 (預設 0.6)
 - 原因：研究架構建構需要平衡邏輯嚴謹性與理論推導的彈性
 - 使用者可透過 Sidebar slider 自行調整
- **Phase 3 (Hypotheses Development)：**
 - **Plan 階段：**固定 0.3
 - 原因：假說推導的邏輯規劃需要嚴謹思考
 - **Write 階段：**固定 0.8
 - 原因：假說陳述需要流暢的學術寫作風格
 - 不受 Sidebar 設定影響
- **Phase 4 (Chapter 2 Literature Review)：**固定 0.8
 - 原因：文獻回顧需要流暢的學術敘事與整合性寫作
 - 不受 Sidebar 設定影響
- **Phase 5 (Chapter 1 Introduction)：**
 - **正文撰寫：**固定 0.8
 - 原因：緒論需要流暢的研究背景與動機敘述
 - **標題生成：**固定 1.0
 - 原因：標題需要高度創意性來吸引讀者注意
 - 不受 Sidebar 設定影響

Graph 模式

- **Mapping 調和關鍵字：**固定 0.0
 - 原因：需要高精確度的 Mapping 調和關鍵字，並強制輸出完整的 JSON 格式，確保調和結果穩定一致
 - 不受 Sidebar 設定影響

重要說明

- 若函數內部有指定 Temperature (如 Synthesize Phase 0、Integrate 各 Phase)，則以該固定值為準，不受 Sidebar 設定影響
- 若函數內部未指定 Temperature (如 Note 模式各 Stage)，則使用 Sidebar 的 Global Temperature 設定
- 建議在開始分析前先設定好 Global Temperature，確保 Note 模式的各 Stage 使用一致的創意度

第六階段：Tesseract OCR 設定 (掃描檔辨識必備)

Papernote 內建 OCR (光學字元辨識) 功能，當您上傳「圖片掃描檔」類型的 PDF 時，系統需要呼叫 Tesseract 引擎來抓取文字。若您的電腦未安裝此軟體，掃描檔 PDF 將無法被分析。

1. 下載安裝檔

請前往 UB-Mannheim (Tesseract Windows 安裝檔維護者) 的頁面下載：

- 下載連結：<https://github.com/UB-Mannheim/tesseract/wiki>
- 下載目標：請尋找最新版本的 `tesseract-ocr-w64-setup-v5.x.x.xxxx.exe` (64-bit) 進行下載。

2. 執行安裝與設定 (關鍵步驟)

執行下載的 `.exe` 檔案，在安裝過程中請務必注意以下兩點：

1. 選擇語言包 (重要)：
 - 到了「Choose Components」步驟時，請展開 **"Additional Script Data"** 或 **"Additional Language Data"**。
 - 勾選 **"Han Traditional"** (繁體中文)。
 - 註：預設已包含 *English*，若您常閱讀其他語言論文，也可在此一併勾選。
2. 安裝路徑 (建議維持預設)：
 - 建議直接安裝在預設路徑：`C:\Program Files\Tesseract-OCR`。
 - 若您更改了安裝路徑，請確保您的系統環境變數 (PATH) 有包含該路徑，否則 Papernote 可能找不到它。

3. 完成

安裝完成後無需重啟電腦，直接開啟 Papernote 即可。當系統遇到掃描版 PDF 時，就會自動呼叫 Tesseract 進行辨識。

第七階段：詳細操作說明

Prohemsyne 系統提供四種主要模式，以下為各模式的詳細操作說明：

模式一：Note (PaperNote - 單篇論文結構化筆記提取)

程序目的

對單篇學術論文進行結構化筆記提取，產出包含 A-Q 各區塊的完整筆記檔案。

輸入資料

- PDF 檔案：學術論文的 PDF 檔案
- 來源方式：

- 手動上傳 PDF
- Auto Flash 1/5：自動從 _temp/ 資料夾讀取最新 1 或 5 篇 PDF
- Random Flash：隨機從 _temp/ 資料夾讀取 1 篇 PDF

執行階段 (8 個 Stage)

Stage 1：Aggregate A–E Notes (結構性與脈絡映射)

- 提取論文的結構性與脈絡資訊 (A. 論文標題、B. 論文結構、C. 研究背景、D. 研究缺口、E. 研究目的)

Stage 2：Aggregate F–I Notes (理論架構與邏輯建構)

- 提取理論基礎與文獻回顧 (F. 統觀理論、G. 文獻回顧、H. 研究命題或研究假說、I. 研究架構)

Stage 3：Aggregate J–N Notes (方法與證據審計)

- 提取研究方法與實證結果 (J. 研究方法、K. 資料分析與結果、L. 研究發現、M. 理論與實務意涵、N. 研究限制與未來研究建議)

Stage 4：Aggregate O–Q Notes (批判性評價)

- 進行批判性評價與編輯級評論 (O. 未證實/未獲支持的研究假說、P. 非預期的研究發現、Q. 模擬期刊編輯評論)

Stage 5：Aggregate Class Discussion Notes (課堂討論筆記)





- 生成課堂討論問題與參考答案

Stage 6:Aggregate Further Reading & Assignment Brief(延伸閱讀與作業簡報)

- 生成延伸閱讀清單與作業建議
- **Google Search Grounding (自動觸發)**：系統在執行此階段時，會自動啟用 Google Search Grounding 功能，強制 AI 聯網搜尋以驗證「Recent Papers」的真實性與引用資訊。
- **⚠️ 人工檢查提醒**：雖然經由 Google Search 驗證可大幅降低幻覺，但 **DOI 與連結仍需人工確認**。請務必檢查每個連結是否有效，避免引用到錯誤或失效的網址。

Stage 7：Cover Page (封面頁)

- 生成筆記檔案的封面資訊 (檔名、期刊、DOI、引用、標籤、日期、版本、Scite.ai 指標)

Scite_ (指標)：系統會用 DOI 向 Scite.ai 查詢 Smart Citations，顯示為  Total /  Mentioning /  Supporting /  Contrasting；詳細解讀請見 FAQ「Scite_ 指標怎麼看？」。

Stage 8：Compile Full Report with Cover Page (編譯完整報告)

- 組合所有 Stage 結果，產出完整筆記檔案，並自動產生 PDF 副本與 Merged PDF

操作流程

方式 A：手動上傳

1. 點選側邊欄「Note」按鈕（確保在 main 模式）
2. 上傳 PDF 檔案
3. 系統自動讀取 PDF 內容
4. 依序執行 Stage 1 → Stage 2 → ... → Stage 8
5. 系統自動儲存並上傳至 Google Drive

方式 B：Flash Report（自動執行）

1. 上傳 PDF 檔案
2. 點選「 Flash Report 
」按鈕3. 系統自動依序執行 Stage 1-7
4. 完成後自動執行 Stage 8 編譯完整報告

方式 C：Auto Flash 1/5（批量處理）

1. 將 PDF 檔案放入 `_temp/` 資料夾
2. 點選「 Auto Flash 1 
」或「 Auto Flash 5 」3. 系統自動讀取最新 1 或 5 篇 PDF
4. 自動執行 Flash Report 流程
5. 批次完成後顯示統計資訊

方式 D：Random Flash（隨機處理）

1. 將 PDF 檔案放入 `_temp/` 資料夾
2. 點選「 Random Flash 
」3. 系統隨機選擇 1 篇 PDF
4. 自動執行 Flash Report 流程

輸出檔案

- **筆記檔案**：YYYY-Author-PaperTitle-Note_YYYYMMDD_HHMMSS.md（儲存於 `_note/`）
- **PDF 副本**：YYYY-Author-PaperTitle-Paper.pdf（儲存於 `_paper/`）
- **Merged PDF**：YYYY-Author-PaperTitle-Merged_YYYYMMDD_HHMMSS.pdf（儲存於 `_merged/`）

PDF 檔案處理邏輯（重要）

系統在 Stage 8 完成後，會自動處理原始 PDF 檔案，處理方式依來源而異：

1. 手動上傳的 PDF 檔案

- **處理方式**：系統會複製 (Copy) 一份 PDF 檔案到 `_paper/` 資料夾
- **檔案重新命名**：原始檔名會根據論文資訊重新命名為 `YYYY-Author-PaperTitle-Paper.pdf` 格式
- **原始檔案**：**不會被刪除**，仍保留在使用者上傳的原始位置
- **目的**：保留使用者原始檔案，同時建立標準化命名的歸檔副本

2. Auto Flash / Random Flash 的 PDF 檔案

- **處理方式**：
 - 步驟 1：系統會複製 (Copy) 一份 PDF 檔案到 `_paper/` 資料夾
 - 步驟 2：檔案重新命名為 `YYYY-Author-PaperTitle-Paper.pdf` 格式
 - 步驟 3：**刪除 (Delete)** `_temp/` 資料夾中的原始 PDF 檔案
- **目的**：避免重複分析，自動清理已處理的檔案，保持 `_temp/` 資料夾整潔
- **注意事項**：
 - 若 PDF 檔案處理失敗或 Stage 8 未完成，原始檔案**不會被刪除**
 - 只有成功完成 Stage 8 且檔案來源確認為 `_temp/` 資料夾時，才會執行刪除動作

3. PDF 副本的儲存位置

- **本機**：`_paper/` 資料夾
- **雲端**：Google Drive `/papernote/_paper/` (若已啟用 Google Drive 同步)

4. 檔案命名規則

- **PDF 副本**：`YYYY-Author-PaperTitle-Paper.pdf`
 - `YYYY`：論文發表年份 (從 Stage 1 提取)
 - `Author`：第一作者姓氏 (從 Stage 1 提取)
 - `PaperTitle`：論文標題 (去除空格，以連字號 - 連接)
 - 若檔名重複，系統會自動在檔名後加上 (1)、(2) 等編號

5. 錯誤處理

- 若複製 PDF 副本失敗，系統會顯示警告訊息，但不影響筆記檔案的產生
- 若刪除 `_temp/` 中的檔案失敗，系統會顯示警告訊息，使用者可手動刪除

模式二：Aggregate (論文筆記聚合 - 依關鍵字標籤分類)

程序目的

從 `_note` 資料夾中，根據使用者指定的關鍵字標籤 (Tag)，自動找出所有包含該標籤的論文筆記檔案，並彙整成一個 aggregate 檔案，作為 Synthesize 模式的輸入資料。

輸入資料

- **Tag 關鍵字**：使用者輸入的關鍵字標籤（例如：PV、#PV、Customer、#Value）
 - 支援格式：
 - 純關鍵字：PV、Customer
 - 帶井號：#PV、#Customer（系統會自動去除 # 進行匹配）
 - 自動分析模式：PV (perceived value)（聚合後自動執行 Synthesize）
 - 來源資料夾：_note/ 資料夾中的所有 .md 筆記檔案

選擇邏輯與匹配規則

1. 檔案掃描範圍

- 系統會掃描 _note 資料夾中的所有 .md 檔案
- 檔案命名格式：YYYY-Author-PaperTitle-Note_YYYYMMDD_HHMMSS.md

2. 關鍵字匹配機制

- 系統會在每個筆記檔案中搜尋包含 My Keywords: 的行
- 注意事項：My Keywords欄位預設是空白，使用者需根據自行需要輸入可以辨識但關鍵字。
- 匹配規則：
 - **不區分大小寫**：輸入 PV 可匹配 #PV、#pv、PV、pv 等
 - **支援帶或不帶井號**：輸入 PV 可匹配 #PV 或 PV；輸入 #PV 也會匹配 PV 或 #PV
 - **部分匹配**：只要 My Keywords: 行中包含該關鍵字即視為匹配
 - **範例**：
 - 輸入 PV → 可匹配 My Keywords: #PV, #Value, #Customer
 - 輸入 Customer → 可匹配 My Keywords: #PV, #Customer, #Engagement
 - 輸入 #Value → 可匹配 My Keywords: PV, Value, Customer

3. 檔案排序

- 匹配到的檔案會依檔名排序（通常依年份由小到大）

4. 輸出檔案

- **檔名格式**：aggregate_tag_{tag}_{數量}_{時間戳記}.md
 - {tag}：使用者輸入的關鍵字（去除 # 後）
 - {數量}：匹配到的論文筆記數量
 - {時間戳記}：格式為 YYYYMMDD_HHMMSS
 - **範例**：aggregate_tag_PV_5_20251210_091530.md
- **儲存位置**：
 - 本機：_aggregate/ 資料夾
 - 雲端：Google Drive /papernote/_aggregate/（若已啟用 Google Drive 同步）

5. 輸出檔案內容結構

- 檔案開頭：# Aggregate notes for tag: {使用者輸入的 tag}
- 每篇論文之間以分隔線區隔：

```

---
PAPERNOTE_PAPER_BREAK
---

## {檔名}

{完整筆記內容}

```

操作流程

方式 A：基本聚合

1. 在側邊欄或主介面的「Tag for Aggregation」輸入框中輸入關鍵字（例如：PV 或 #PV）
2. 點擊「Aggregate」按鈕
3. 系統自動掃描 _note 資料夾，找出所有 My Keywords: 行包含該關鍵字的筆記檔案
4. 系統自動產生 aggregate 檔案並儲存至 _aggregate/ 資料夾
5. 若已啟用 Google Drive 同步，檔案會自動上傳至雲端

方式 B：自動聚合 + Synthesize 分析（一鍵完成）

1. 在輸入框中輸入 {關鍵字} ({構念名稱}) 格式（例如：PV (perceived value)）
2. 點擊「Aggregate」按鈕
3. 系統會自動執行：
 - 步驟 1：聚合筆記檔案（產生 aggregate 檔案）
 - 步驟 2：自動切換到 Synthesize 模式
 - 步驟 3：自動執行 Phase 0 → Phase 1 → Phase 2 → Phase 3 → Export 完整流程

輸出檔案範例

檔名：aggregate_tag_PV_3_20251210_091530.md

內容結構：

```

# Aggregate notes for tag: PV

---
PAPERNOTE_PAPER_BREAK
---

## 2020-Lin-Industrial-tourism-Note_20251210_080000.md

[第一篇論文的完整筆記內容]

---

```

PAPERNOTE_PAPER_BREAK

2021-Smith-Customer-value-Note_20251210_081500.md

[第二篇論文的完整筆記內容]

PAPERNOTE_PAPER_BREAK

2022-Johnson-Perceived-value-Note_20251210_083000.md

[第三篇論文的完整筆記內容]

注意事項

- **My Keywords 欄位設定**：在 Note 模式產生的筆記檔案中，My Keywords: 欄位預設為空白，需由使用者手動填入關鍵字標籤
- **關鍵字格式建議**：建議使用 #關鍵字 格式（例如：#PV、#Customer），但系統也支援不帶 # 的格式
- **匹配靈活性**：系統會自動處理大小寫與井號的差異，使用者無需擔心格式不一致的問題
- **檔案不存在處理**：若 _note 資料夾不存在或找不到匹配的檔案，系統會顯示錯誤訊息

模式三：Synthesize (PaperSynthesize - 單一構念文獻綜合)

程序目的

對單一核心構念 (Focal Construct) 進行文獻綜合分析，產出該構念的綜合報告。

輸入資料

- **Aggregate 檔案**：包含多篇論文筆記的 .md 檔案（檔名需含 "aggregate"）
 - 通常由 Aggregate 模式產生，或手動上傳符合格式的 aggregate 檔案
 - 檔案格式：每篇論文之間以 ---\n PAPERNOTE_PAPER_BREAK\n--- 分隔
- **Focal Construct**：核心構念名稱（例如：Perceived Value、Customer Engagement）

執行階段

Phase 0：Construct Harmonization (構念調和)

- 統一變數名稱，建立對照表

Phase 1：Paper-level Table (逐篇摘要表)

- 逐篇分析論文，提取關鍵資訊（CSV 表格，包含 12 個欄位）

Phase 2 : Cross-study Synthesis (跨研究綜整)

- 跨研究整合分析結果，找出共通模式與分歧點

Phase 3 : Research Framework And Hypotheses (研究架構與假說)

- 提出研究架構與研究假說（含 APA 引用與 LOCK 指針）

Export : 匯出綜合報告

- 組合所有階段結果，產出完整報告：`synthesis_tag_{構念名稱}_{論文數量}_{時間戳記}.md`

操作流程

1. 點選側邊欄「Synthesize」按鈕
2. 上傳 aggregate.md 檔案
3. 輸入 Focal Construct 名稱
4. 依序執行 Phase 0 → Phase 1 → Phase 2 → Phase 3 → Export
5. 系統會自動儲存結果並上傳至 Google Drive

特殊功能

- **自動模式**：在 Aggregate 輸入框輸入 PV (perceived value) 格式，系統會自動執行 Phase 0 → Export 的完整流程
- **進度追蹤**：系統會記錄每篇論文的處理狀態，避免重複處理

模式四：Integrate (PaperIntegrate - 多構念整合與論文章節撰寫)

程序目的

整合多個 synthesis.md 檔案，構建完整的研究模型，並撰寫論文的 Chapter 1（緒論）與 Chapter 2（文獻回顧）。

輸入資料

- **多個 Synthesis 檔案**：檔名需含 "synthesis" 的 .md 檔案（可複選上傳）
- **或使用智能探索推薦**：系統會自動推薦可整合的構念組合

執行階段

Phase 1 : Construct Harmonization (構念整合)

- 整合多個 synthesis 檔案中的構念名稱

Phase 2 : Framework Construction (研究架構建構)

- 基於整合後的構念，建構 Hayes PROCESS 模型 (Model 4/6/7/14)

Phase 3 : Hypotheses Development (研究假說發展)

- 推導研究假說 (H1, H2, H3...)，每個假說包含假說陳述、類型與推論邏輯

Phase 4 : Chapter 2 Literature Review (第二章文獻回顧)

- 撰寫完整的第二章文獻回顧 (依變數順序：Y → X → M1 → M2 → W/V → 研究假說與架構)

Phase 5 : Chapter 1 Introduction (第一章緒論)

- 撰寫第一章緒論 (研究背景與動機、研究目的與問題、研究重要性與貢獻)

Export : 匯出整合報告

- 組合所有階段結果，產出完整報告：`integrate_{檔名標題}_{時間戳記}.md`

操作流程

方式 A : 手動上傳 Synthesis 檔案

1. 點選側邊欄「Integrate」按鈕
2. 手動上傳多個 synthesis.md 檔案 (可複選)
3. 依序執行 Phase 1 → Phase 2 → Phase 3 → Phase 4 → Phase 5 → Export
4. 系統會自動儲存結果並上傳至 Google Drive

方式 B : 智能探索Top5 (輔助決策)

當您不確定哪些構念適合整合時，可使用「智能探索Top5」功能讓 AI 協助推薦。

1. 點選側邊欄「Integrate」按鈕
2. 展開「智能探索」區塊
3. 系統會自動掃描 `_synthesis/` 資料夾，顯示候選構念清單
4. 點擊「智能探索Top5」按鈕
5. AI 會分析所有候選構念組合，並推薦 Top 5 (依評分由高到低排序)
6. 查看推薦結果，選擇任一組合點擊「 套用此組合進行完整的 Integrate 分析」
7. 系統會自動讀取該組合的 synthesis.md 檔案，並儲存該推薦的 model、roles、架構描述等規格
8. 依序執行 Phase 1 → Phase 2 → Phase 3 → Phase 4 → Phase 5 → Export (此流程會強制依所套用的推薦規格產出，與手動上傳時的自由推導不同)

方式 C : 智能整合#1 (一鍵完成)

當您想快速獲得最佳推薦並立即執行完整分析時，使用「智能整合#1」功能。

1. 點選側邊欄「Integrate」按鈕
2. 展開「智能探索」區塊
3. 點擊「智能整合#1」按鈕
4. 系統會自動執行以下流程：
 - 步驟 1：執行智能探索，分析所有候選構念組合
 - 步驟 2：自動套用 Top 1 推薦（評分最高的組合），並儲存該推薦的 model、roles、架構描述等規格
 - 步驟 3：自動依序執行 Phase 1 → Phase 2 → Phase 3 → Phase 4 → Phase 5 → Export（強制依 Top 1 的規格產出）
 - 步驟 4：自動儲存結果並上傳至 Google Drive（若已啟用）
5. 完成後即可在 `_integrate/` 資料夾中查看完整報告

智能探索功能詳細說明

功能目的

在使用者尚未確定哪些構念適合整合時，「智能探索」功能會自動分析 `_synthesis/` 資料夾中的所有 `synthesis.md` 檔案，運用 AI 推薦最有潛力形成研究架構的構念組合，協助使用者快速找到適合的研究方向。

評分機制（5 個維度，各 0-2 分，總分 0-10）

系統會針對每個構念組合進行多維度評估，並提供詳細的評分理由：

1. **模型可行性（0-2 分）**：能否清楚對應到 Hayes PROCESS Model 4/6/7/14（X/M1/M2/Y/W/V 角色合理，不硬湊）
2. **因果鏈合理性（0-2 分）**：能否講出合理的「為什麼 X 會影響 Y」機制（中介/序列中介/調節位置合理）
3. **互補性與非重複（0-2 分）**：構念彼此互補（前因/機制/結果/邊界），不是同義或高度重疊
4. **研究新穎/潛在貢獻（0-2 分）**：看起來有理論或實務貢獻，不是太教科書或太直覺
5. **可操作化/可測量性（0-2 分）**：從內容看起來，有機會量化或操作（問卷/實驗/二手資料不會太牽強）

輸出資訊

每個推薦組合會包含以下資訊：

- **排名與總分**：例如「#1 | 分數：9/10」
- **模型類型**：對應的 Hayes PROCESS Model（例如：Model 6 + 7）
- **評分細目**：顯示 5 個維度的各別分數與理由
- **矛盾旗標**：若構念之間存在矛盾/互斥情況（例如因果方向相反），會明確標示並說明研究機會（不影響評分）
- **角色對應**：X、M1、M2、Y、W、V 等變數角色分配
- **架構描述**：該組合若進入正式 Integrate 後，可能形成怎樣的研究架構

- **檔案清單**：使用到的完整 synthesis.md 檔名

候選構念篩選規則

- 系統會自動掃描 `_synthesis/` 資料夾中檔名含有 "synthesis" 的 .md 檔案
- 相同構念（依檔名判斷）只會保留最新時間戳記的檔案
- 若候選構念超過 40 個（預設值），系統會優先選擇 paper_count 較高的前 40 個構念送入 AI 分析

Temperature 設定

- 智能探索使用固定 Temperature = 0.8（較高創意度，適合探索構念組合的可能性與新穎性）
- 不受 Sidebar 的 Global Temperature 設定影響

輸入來源優先次序

當同時存在手動上傳檔案與智能探索推薦時，系統會依照以下優先順序決定使用哪個來源：

1. **手動上傳檔案**（最高優先）：若使用者手動上傳了 synthesis.md 檔案，永遠優先使用手動上傳的檔案
2. **智能探索推薦**（次優先）：若未手動上傳檔案，則使用「最後一次套用」的智能探索推薦

套用時強制使用規格（Spec）的說明

- **智能探索 Top 5 套用 或 智能整合#1**：當您「套用」某一個推薦組合後，系統會儲存該推薦的 **模型類型（Model）、變數角色對應（Roles：X/M1/M2/Y/W/V）、架構描述（framework description）與 矛盾旗標（contradiction flags）**。執行 Phase 1~Phase 5 時，AI 會**強制依此規格**產出：構念調和、研究架構建構、假說發展與章節撰寫皆與您所選的推薦架構一致，確保最終整合報告與智能探索套用的架構一致。
- **手動上傳檔案**：若您改為**手動上傳** synthesis.md 檔案並執行 Integrate，系統**不會**套用上述強制規格。此時 AI 會依上傳內容自由推導研究架構與假說，不受智能探索推薦影響。切換回手動上傳時，系統也會自動清空先前套用的推薦規格，避免沿用舊的 model/架構。

注意事項

- 智能探索的推薦結果會儲存在介面上，使用者可重複套用不同推薦組合
- 每次套用新推薦會覆蓋前一次的選擇（以「最後一次套用」為準）
- 若候選構念不足或檔名格式不符，系統會顯示提示訊息
- 智能探索階段不要求 LOCK 指針與引文嚴格性，正式 Integrate 階段會補充完整

模式五：Graph（知識網絡視覺化分析）

程序目的


分析 `_note`（或自訂來源）資料夾中所有論文筆記的關鍵字共現關係，產出網絡圖與結構化資料，協助使用者探索文獻之間的隱藏連結與知識結構。

輸入資料

- **來源資料夾：** `_note/`（或自訂來源）資料夾中的所有 `.md` 筆記檔案
- **關鍵字來源：** 從每篇筆記的 Author Keywords: 欄位提取標籤（例如：`#PV`、`#Customer`、`#Engagement`）

Mapping 調和關鍵字（Keyword Harmonization，選用）

不同論文對同一概念的 Author Keywords 寫法常不一致（例如 SEM / Structural Equation Modeling、behavior / behaviour、單複數差異），若直接建圖會產生多個節點、邊的權重分散。**Mapping 調和關鍵字**會在分析網絡前，將「原始關鍵字」對應到「標準名稱」，再以標準名稱建立節點與邊，使網絡分析更集中、指標更穩定。

- **功能說明：**系統掃描 `_note` 中所有筆記的 Author Keywords，產生「原始關鍵字清單（含出現次數）」；若啟用 **Mapping 調和關鍵字**，AI 會產出一份「原始詞 → 標準詞」的對照表（同義詞、縮寫、單複數、拼字差異合併為單一標準名）。建圖、匯出 CSV/GraphML、Modularity 與後續指標（P、z、角色、路徑、對等性）皆以調和後的標準名稱計算；**原始 `_note` 內的 `.md` 檔案完全不會被修改。**
- **對照表檔案：**對照表存於 `_graph/` 資料夾，檔名為 `keyword_mapping_YYYYMMDD_HHMMSS.json`。每次由 **Mapping 調和關鍵字**（AI）重新產生調和結果時會寫入新檔（帶時間戳），不會覆蓋舊檔。系統依「**檔案最後修改時間**」判斷「最新」的對照表；因此若您手動編輯或覆蓋某一份 `keyword_mapping_*.json` 並存檔，下次執行分析時會自動採用該份。**排序與分批：**原始關鍵字會依「關鍵字字串」排序後送給 LLM，使同前綴／相近詞相鄰；若關鍵字數超過單批上限（約 200 筆），會自動分批呼叫 LLM，第二批起會傳入前幾批已產生的標準詞，要求同義詞對齊到既有標準詞。
- **介面操作：**在 Graph 模式的參數區可勾選「**Mapping 調和關鍵字**」或點擊「**Mapping 調和關鍵字**」按鈕（預設開啟）。若已有對照表，會顯示「 已載入對照表：檔名」；可勾選「**重新生成對照表**」強制產生新對照表。點擊「 執行分析」後，若有啟用 Mapping 調和且需要新對照表，會先進行關鍵字掃描與 AI 調和、寫入 `keyword_mapping_*.json`，再執行網絡分析。
- **手動編修對照表：**若不滿意 AI 的合併結果，可至 `_graph/` 資料夾開啟**目前修改時間最新**的 `keyword_mapping_*.json`，直接編輯「原始詞 → 標準詞」對應；存檔後，下次執行分析時（勿勾選「重新生成對照表」）會採用您修改後的對照表，節點與邊會依新對照表重新計算。

Embedding 語義再合併（選用）

在「**Mapping 調和關鍵字**」之後，可進一步啟用 **Embedding 語義再合併**，將 LLM 產出的標準詞依「語義相似度」再收斂，減少節點數、合併同家族概念（例如不同寫法的 `engagement`、`experience`、`value`、`food/culinary` 等）。

- **概念與邏輯**：系統先以 **Mapping 調和關鍵字** (LLM) 產出 `keyword_mapping_*.json` (原始關鍵字 → 標準詞)；若啟用 **Embedding 再合併**，會使用 **Gemini Embedding 模型 (gemini-embedding-001)** 對所有「標準詞」計算語義向量，以**餘弦相似度**判斷哪些標準詞可視為同一概念並合併為單一代表詞。合併結果存成 `keyword_embedding_*.json` (三欄位：原始關鍵字、`llm_standard`、`embedding_merged`)；**做圖時**若有啟用 **Embedding 再合併**，則使用「原始關鍵字 → `embedding_merged`」建立節點，其餘流程 (共現、Modularity、CSV、GraphML、PNG) 不變。**排序與分批**：標準詞依字串排序後分批處理 (每批約 250 筆)；第二批起會先與前幾批的「代表詞」比對相似度，達門檻即合併到該代表詞，未達門檻者再於本批內聚類，使同義詞盡量對齊到同一代表詞。
- **UI**：勾選「**Mapping 調和關鍵字**」後會顯示「**Embedding 再合併**」核取方塊 (與「重新生成對照表」同一區塊右側)；勾選後會出現「**Embedding 語義再合併門檻**」滑桿，範圍 **0.75~0.90**，預設 **0.82**。滑桿代表「餘弦相似度 ≥ 此值才合併」：門檻過低會過度合併 (例如 `experience` 被併入 `engagement`)，過高則幾乎不再合併；論文或關鍵字變多時可依結果在介面微調，無須改程式。
- **檔案**：`keyword_embedding_{時間戳記}.json` 存於 `_graph/`，時間戳記與對應的 `keyword_mapping_*.json` 一致。格式為**物件**：每個 key 為原始關鍵字，value 為 `{"llm_standard": "標準詞", "embedding_merged": "再合併後的詞"}`。做圖僅使用「原始關鍵字 → `embedding_merged`」。
- **何時用 Mapping 產出的 mapping、何時用 embedding**：未勾選「Embedding 再合併」時，建圖使用 **Mapping 調和關鍵字**產出的 `keyword_mapping_*.json` (原始 → 標準詞)；勾選後則使用 `keyword_embedding_*.json` (原始 → `embedding_merged`)。若已有 Mapping 產出的 mapping 檔但尚無對應的 embedding 檔，勾選 **Embedding 再合併**時系統會依目前 mapping 當場計算並存檔。

執行流程

步驟 1：掃描與提取

- 系統自動掃描 `_note` (或自訂來源) 資料夾中的所有筆記檔案
- 從每篇筆記的 `Author Keywords` 欄位提取關鍵字標籤
- 統計每個關鍵字出現的論文數量

步驟 2：共現分析

- 計算關鍵字之間的共現關係 (co-occurrence)
- 若兩個關鍵字同時出現在同一篇論文中，則視為一次共現
- 統計每對關鍵字的共現次數，作為連結權重 (edge weight)

步驟 3：網絡圖繪製

- 系統自動繪製關鍵字網絡圖 (PNG 格式)
- **節點 (Node)**：代表關鍵字，大小反映該關鍵字出現的論文數量
- **邊 (Edge)**：代表關鍵字之間的共現關係，粗細反映共現次數

步驟 4：關鍵字分群與角色分類

- 系統使用 **Louvain 演算法** 自動進行關鍵字分群
- 計算 **Modularity (模組化程度)**：衡量網絡社群結構的明顯程度
- 計算每個節點的 **Participation Coefficient (P, 參與係數)** 和 **Within-module z-score (z)**
- 根據 Guimerà & Amaral (2005) 的理論，將節點分類為 **R1-R7 七種角色**

步驟 5：資料匯出

- 系統自動產出 GraphML、CSV、Modularity CSV 與 PNG：
 - **network.graphml**：GraphML 格式，可直接在 [Gephi Lite](#) 開啟
 - **nodes.csv**：節點清單（關鍵字與出現次數）
 - **edges.csv**：邊清單（關鍵字對與共現次數）
 - **modularity.csv**：關鍵字分群結果，含 P、z-score、角色分類（R1-R7）
 - **network.png**：網絡圖視覺化

操作流程

1. 點選側邊欄「Graph」按鈕
2. 系統自動掃描 `_note` 資料夾
3. 選擇分析參數（可選）：
 - **Mapping 調和關鍵字**：是否啟用 AI 調和同義詞／縮寫／單複數（預設：是）；可勾選「重新生成對照表」強制產生新對照表
 - **Embedding 再合併**：是否在 LLM 標準詞基礎上，依語義相似度再合併標準詞以減少節點（預設：否）；勾選後會顯示「Embedding 語義再合併門檻」滑桿（0.75~0.90，預設 0.82）
 - **Min Weight**：最小連結權重（預設：1，只顯示共現次數 ≥ 1 的連結）
 - **Top N Edges**：顯示前 N 條最強連結（預設：2000）
 - **Top N Nodes**：顯示前 N 個最重要節點（預設：200）
 - **K-Clique 社群偵測 K 值**：重疊社群偵測參數（3~6，預設 4），見下方「重疊社群偵測」說明
4. 點擊「 執行分析」執行分析
5. 系統自動產出：
 - GraphML、CSV（nodes/edges）、網絡圖 PNG
 - **Modularity CSV**：含關鍵字分群、Participation Coefficient、z-score、角色分類
 - **K-Clique CSV**：重疊社群偵測結果（`kclique_k{K}_{nodes}_{edges}_{社群數}_{timestamp}.csv`），含重疊節點與未覆蓋節點
 - 所有檔案儲存至 `_graph/` 資料夾
6. 畫面顯示 **模組化程度**、**關鍵字分群數量**、**Connected components**（連通分量）與 **K-Clique 重疊社群分析結果**（若有）

產出【完整網絡分析】報告

分析完成後（或當 `_graph/` 中已有同一 timestamp 的完整一組檔案時），可點擊「 產出【完整網絡分析】報告」按鈕，由 AI 根據當前網絡資料自動撰寫一份場域層級分析報告與閱讀策略建議。

邏輯與資料來源

- 系統會掃描 `_graph/`，找出 **timestamp 最新且齊全** 的一組檔案（GraphML、Nodes CSV、Edges CSV、Modularity CSV、NodeIndices CSV；若有 K-Clique CSV 則報告會納入重疊社群分析）。**不依賴本次是否剛執行分析**，只要該組檔案存在即可產出。
- 產出時會從磁碟讀取上述檔案，例如從 GraphML 計算連通分量（Connected components、各塊節點數與占比），從 NodeIndices 與 Modularity 取得節點指標與分群，組合成一份給 AI 的資料後呼叫 Gemini 撰寫報告。
- 報告開頭會由系統自動附加產出時間、本報告依據之檔案清單、完整節點指標可參閱之 CSV 檔名，以及連通塊（Connected components = N、First component: Size = ...、% of nodes = ...）供對照。

報告內容

- **Modularity 摘要**：模組化程度、關鍵字分群數與各 cluster 關鍵字列表。
- **Network Overview**：節點數、邊數、密度、連通分量（connected components）與白話說明、平均群聚係數等整體指標。
- **Core Keywords**：依 **Degree Centrality 與 Frequency** 排序固定取前 30 個節點之指標與綜合解讀（頻率、PageRank、度中心性、中介中心性、接近中心性、參與係數、z、角色、結構洞限制、群聚係數等）。
- **Bridging Keywords**：高參與係數／低結構洞限制之節點與角色說明。
- **Overlapping Communities**：若有 K-Clique 檔案，報告會撰寫重疊社群節點逐一分析（與 Modularity 互補，見 FAQ「重疊社群偵測」）；節點順序依**所屬 K-Clique Cluster 數與 Frequency**，與 Core Keywords 順序不同。
- **Peripheral Patterns**：低中心性節點之整體型態。
- **Reading Strategy Recommendations**：依網絡結構建議之閱讀優先順序與操作建議。

輸出檔案

- 檔名：AuthorKey_Network_Analysis_{nodes}_{edges}_{報告產出時間戳}.md
- 儲存位置：`_graph/` 資料夾。


注意

- 產出報告會呼叫 Gemini API，需消耗 token；報告長度與節點數有關，若節點很多，生成時間可能較長。
- 連通分量在報告中為**產出當下從 GraphML 重算**，與執行分析完成時畫面上顯示的連通分量計算方式一致，結果相同。

重疊社群偵測（K-Clique / CPM）

執行分析時，系統會一併進行 **K-Clique 重疊社群偵測**，與 Modularity (Louvain) 分群互補使用。


- **用途**：找出同時參與多個緊密子群的**橋接關鍵字**，揭示跨領域整合結構。
- **參數**：在參數區可調整「**K-Clique 社群偵測 K 值**」（3~6，預設 4）。K=3 以三角形為基本單元；K 愈大社群愈緊密。
- **輸出**：kclique_k{K}_{nodes}_{edges}_{社群數}_{timestamp}.csv，欄位含 K-Clique 群組、關鍵字、所屬群組數、所屬全部群組。
- **解讀**：所屬群組數 ≥ 2 為**重疊節點**（橋接多個 K-Clique Cluster）；所屬群組數 = 1 為非重疊；不參與任何 K-clique 者為**未覆蓋節點**。
- **Modularity vs. K-Clique**：Louvain 為非重疊分群，每個關鍵字只屬一個 Cluster；K-Clique 為重疊分群，可揭露跨主題整合概念。兩者併看可更完整理解場域。

 **提示**：重疊社群偵測的函數、演算法、輸出解讀與 Modularity 互補關係，詳見 FAQ 「重疊社群偵測 (K-Clique / CPM)」 章節。

進階分析：關鍵字路徑探索

分析完成後，系統會在右側顯示「進階分析：關鍵字路徑探索」區塊，協助使用者探索兩個關鍵字之間的連結關係。

操作方式


1. 從 Source（起點）下拉選單選擇一個關鍵字
2. 從 Target（終點）下拉選單選擇另一個關鍵字
3. 點擊「 分析路徑」按鈕執行分析

輸出內容

- **最短路徑**：顯示從起點到終點的最短連結路徑與步數
- **邊的權重**：路徑上每條邊的共現論文數
- **節點指標表格**：路徑上每個關鍵字的頻率、PageRank、度中心性、中介中心性、接近中心性、**參與係數/角色 (P/z/Role)**、結構洞限制、群聚係數
- **路徑視覺化**：以線性圖呈現路徑結構，節點大小反映頻率，邊粗細反映共現次數
- **整體關鍵字網絡模組化**：Modularity 數值與關鍵字分群數量
- **Source 與 Target 對等性分析**：比較起點與終點在網絡中的結構關係與角色相似性（見下方說明）

P/z/Role 欄位說明：

- **P** (Participation Coefficient)：衡量節點連結的跨社群分布程度 (0~1)
- **z** (Within-module z-score)：衡量節點在所屬社群內的重要性
- **Role** (角色分類)：根據 P 和 z 的組合，分類為 R1-R7 七種角色
 - R1-R4：Non-hub 節點 ($z < 2.5$)
 - R5-R7：Hub 節點 ($z \geq 2.5$)

 **提示：**節點指標、對等性分析與 Modularity 的詳細計算公式、數值範圍與解釋說明，請參閱 FAQ「Graph 模式節點指標說明」章節。

若兩個關鍵字之間沒有連結路徑，系統會顯示提示訊息。

Source 與 Target 對等性分析

在路徑視覺化下方，系統會自動計算並顯示 Source 與 Target 兩個節點的三種對等性指標，協助使用者理解這兩個關鍵字在網絡中的結構關係與角色相似性。

三種對等性指標

對等性類型	定義	主要指標
結構對等性 (Structural Equivalence)	兩節點若連接到完全相同的鄰居，則具有結構對等性	Jaccard 相似度、Pearson 相關係數
自同構對等性 (Automorphic Equivalence)	若存在圖的自同構映射能將節點 A 映射到節點 B，則兩者自同構對等。換句話說，它們在網絡中的「結構位置」相同，但不一定連接到相同的鄰居	Ego Network 同構判斷、結構特徵相似度
規則對等性 (Regular Equivalence)	兩節點是規則對等的，若它們與「對等的其他節點」有相同類型的關係。這是最寬鬆的對等性概念，不要求連接到相同鄰居，只要在網絡中扮演類似的「角色」	REGE 分數、REGE 百分位、簡化版分數

輸出內容

系統會以三個 Tab 分別顯示各對等性的詳細資訊：

1. **結構對等性 Tab**：Jaccard 相似度、Pearson 相關係數、共同鄰居列表、解釋說明
2. **自同構對等性 Tab**：Ego Network 同構判斷、結構特徵相似度、兩節點的結構特徵比較表 (Degree、Clustering Coefficient、K-Core Number、Ego Network 節點數、Ego Network 密度)、解釋說明
3. **規則對等性 Tab**：簡化版分數、REGE 分數、REGE 百分位、鄰居 Degree 分布統計比較表、解釋說明

對等性分析總結表

系統會在最後顯示一個總結表格，彙整三種對等性的主要指標與輔助指標。


CSV 自動匯出

分析完成後，系統會自動將對等性分析結果儲存為 CSV 檔案至 `_graph/` 資料夾，檔名格式為：`equivalence_{source}_{target}_{時間戳記}.csv`。

此外，系統也會自動執行「整體關鍵字網絡模組化分析」，並將結果儲存為：
`modularity_{nodes數量}_{edges數量}_{群組數}_{時間戳記}.csv`。

使用時機

- **探索關鍵字關係**：了解兩個關鍵字在知識網絡中是否扮演相似的角色
- **發現研究主題**：找出結構位置相似但尚未被連結的關鍵字組合
- **文獻回顧分析**：評估不同研究主題之間的結構相似性與互補性
- **教學應用**：向學生說明網絡分析中的對等性概念與實際應用

 **提示**：對等性指標的詳細計算公式與解釋說明，請參閱 FAQ「Graph 模式對等性分析」章節。

輸出檔案與位置

- 本機：`_graph/` 資料夾
- 檔名格式：
 - `AuthorKey_Network_<全量nodes>_<全量edges>_時間戳記.graphml`（100% 全資料輸出，Gephi Lite 直接可用）
 - `AuthorKey_Nodes_<全量nodes>_時間戳記.csv`（100% 全資料輸出，不受分析參數篩選機制影響）
 - `AuthorKey_Edges_<全量edges>_時間戳記.csv`（100% 全資料輸出，不受分析參數篩選機制影響）
 - `AuthorKey_Network_<filter後nodes>_<filter後edges>_時間戳記.png`（分析參數篩選後之輸出）
 - `modularity_{nodes數量}_{edges數量}_{群組數}_{時間戳記}.csv`（關鍵字分群 + P/z/Role）
 - `kclique_k{k}_{nodes數量}_{edges數量}_{社群數}_{時間戳記}.csv`（K-Clique 重疊社群偵測）
 - `equivalence_{source}_{target}_{時間戳記}.csv`（對等性分析結果）

modularity.csv 檔案內容：

- 第一列：Modularity 值與群組總數摘要
- 欄位：群組，關鍵字，Participation (P)，z-score (z)，角色 (Role)，角色說明
- 範例：

```
群組,關鍵字,Participation (P),z-score (z),角色 (Role),角色說明
Modularity: 0.8099,共 41 個群組,,,,
群組 1,PV,0.3214,2.85,R6,Connector Hub
群組 1,Customer,0.1523,1.92,R2,Peripheral
群組 2,Engagement,0.4521,3.12,R6,Connector Hub
```

進階應用：Gephi Lite / Gephi Desktop 網絡分析

Gephi Lite (推薦, 免安裝)

- 輸出的 `.graphml` 檔案可直接在瀏覽器中開啟, 無需安裝軟體
- **Gephi Lite 官網**: <https://gephi.org/lite/>
- **使用方式**: 開啟網站 → File → Open → 選擇 `.graphml` 檔案

Gephi Desktop 整合

- 輸出的 `nodes.csv` 與 `edges.csv` 為 100% 全資料輸出, 不受篩選機制影響, 可匯入 Gephi 軟體, 進行更專業的視覺化分析。
- **Gephi 官網**: <https://gephi.org/>

CSV 匯入步驟 (Gephi Desktop)

1. 開啟 Gephi
2. 點選「File」>「Import Spreadsheet」
3. 先匯入 `edges.csv` (選擇「Edges table」)
4. 再匯入 `nodes.csv` (選擇「Nodes table」)
5. 自訂節點與邊的視覺樣式 (顏色、大小、標籤)
6. 套用 filter (e.g., Giant Component, Degree Range, Edge Weight Range)
7. 應用 layout 演算法 (e.g., OpenOrd, Yifan Hu, ForceAtlas2, Prevent Overlap, Label Adjust)
8. 匯出高解析度網絡圖或互動式網頁版本

Gephi 進階分析功能

- **關鍵字分群**: 找出關鍵字群組 (modularity clustering) — 本系統已內建 Louvain 演算法
- **中心性分析**: 識別最重要的關鍵字節點 (degree centrality、betweenness centrality)
- **路徑分析**: 探索關鍵字之間的連結路徑 — 本系統已內建「關鍵字路徑探索」功能
- **時間序列**: 若資料包含時間資訊, 可製作動態網絡圖

內建進階分析功能 (無需 Gephi)

- **關鍵字分群**: Louvain 演算法自動分群, 輸出 Modularity 值與群組分配
- **重疊社群偵測**: K-Clique (CPM) 偵測同時參與多個緊密子群的橋接關鍵字, 輸出重疊節點與未覆蓋節點
- **角色分類**: 基於 Guimerà & Amaral (2005), 計算 Participation Coefficient (P) 與 z-score, 將節點分類為 R1-R7 七種角色
- **路徑探索**: 選擇起點與終點關鍵字, 分析最短路徑上各節點的網絡指標 (degC, betC, cloC, P/z/Role, Constraint, Clustering)
- **對等性分析**: 比較路徑上節點在不同指標的相對位置

使用時機

- **知識地圖繪製**: 快速了解自己的研究領域涵蓋哪些主題

- **文獻缺口探索**：找出尚未被充分連結的關鍵字組合
- **研究主題發現**：透過網絡結構發現潛在的研究方向
- **教學簡報**：製作視覺化的文獻回顧簡報
- **論文寫作**：作為文獻回顧的視覺化輔助

注意事項

- **關鍵字標籤**：Graph 依賴 Author Keywords:，AI 於 Note 分析自動填入，若該欄位為空白或格式不一致，會影響分析結果
- **標籤格式建議**：若需要人工編修關鍵字，建議統一使用 #關鍵字 格式（例如：#PV、#Customer），並保持命名一致性
- **大小寫處理**：一律忽略大小寫（PV 與 pv 視為相同）
- **資料量建議**：建議至少有 10 篇以上的論文筆記，才能產出有意義的網絡結構

模式六：Small World（關鍵字距離與中心／邊陲）

Small World 以「關鍵字距離」與「誰是中心／誰是邊陲」呈現閱讀網絡，概念類似 The Oracle of Bacon (<https://oracleofbacon.org/>)；全部本機計算、不呼叫 AI。使用 `_graph/` 中 **timestamp** 最新的 GraphML，因此需先在 **Graph 模式** 執行分析並產出 GraphML。

進入方式

- 點選側邊欄「**Small World**」按鈕（位於 Graph 下方）。
- 若 `_graph/` 內尚無 GraphML，畫面會提示「目前沒有可用的關鍵字網絡資料，請先到 Graph 執行分析」。

資料來源

- 畫面會標示目前使用的 GraphML 檔名與節點數（例如：
AuthorKey_Network_431_1725_20260219_195339.graphml（431 個節點））。

三大功能（上方三顆按鈕）

1. **關鍵字 A → B 最短路徑**：選擇起點與終點關鍵字，顯示兩者間的最短路徑與步數。
2. **中心指標（分布表 + 平均距離）**：選擇一個關鍵字為「中心」，顯示該中心到所有可連通節點的**距離分布表**（距離 0, 1, 2, ... 各有幾節點）、可連通節點數、Weighted total、平均距離；右欄並有計算過程與名詞說明（可連通節點數、Weighted total、平均距離）。
3. **距離 N 有誰**：選擇中心關鍵字與距離 N，列出與該中心**恰為 N 步**的所有關鍵字。

Which Keywords are Central and Peripheral to Your Reading?

- 一進入 Small World 即自動計算（同一份 GraphML 只算一次），結果持續顯示於下方。
- 僅針對**最大連通塊**內關鍵字，依「平均距離」由小到大排序；**同一表格列前 25 名**（最佳中心）與**最後 25 名**（最邊陲）。僅取最大連通塊可避免小塊內關鍵字因可連通節點少、平

均距離易偏小而被誤判為「閱讀核心」的盲點。右欄說明含意與排序規則（僅計最大連通塊、平均距離越小越像核心；平均相同時依關鍵字名稱穩定排序）。

使用時機

- 想快速看「誰是閱讀網絡的核心關鍵字、誰最邊陲」「兩關鍵字幾步可連」「距離某關鍵字 N 步有誰」時使用；不需 AI、即時本機計算。

六種模式的關係

資料流向

```
PDF 論文
  ↓ Note (單篇論文筆記提取)
Note 檔案 (_note/)
  └─ → Graph (知識網絡視覺化) → 網絡圖、GraphML、CSV、mapping 或
embedding.json、關鍵字網絡分析.md (_graph/)
  └─ → Small World (關鍵字距離與中心／邊陲) → 使用 _graph/ 最新 GraphML
  ↓ Aggregate (聚合，依 tag 分類)
Aggregate 檔案 (_aggregate/)
  ↓ Synthesize (單一構念綜合)
Synthesis 檔案 (_synthesis/)
  ↓ Integrate (多構念整合)
Integrate 檔案 (_integrate/)
```

使用時機

- **Note**：當您需要對單篇論文進行結構化筆記提取時使用
- **Aggregate**：當您已完成多篇論文的 Note，想要根據關鍵字標籤聚合相關論文筆記時使用（作為 Synthesize 的準備步驟）
- **Synthesize**：當您想針對單一構念進行深入文獻分析時使用（需要先有 Aggregate 檔案）
- **Integrate**：當您已完成多個構念的 Synthesis，想要整合成完整研究模型並撰寫論文章節時使用
- **Graph**：當您想視覺化分析所有論文筆記的關鍵字網絡關係，探索知識結構與潛在連結時使用（可在任何時候執行，只要 _note/ 資料夾中有足夠的筆記檔案）
- **Small World**：當您已有 Graph 產出的 GraphML，想快速查看關鍵字距離、誰是中心／邊陲、或「距離 N 有誰」時使用（本機計算、不呼叫 AI）

第八階段：Obsidian 關鍵字共現網路圖（選用功能）

 提示：此為選用功能 (Optional)

本階段說明如何使用 **Obsidian + Dataview 外掛**，在本機離線環境中視覺化呈現「關鍵字

共現網路圖」(Co-occurrence Network)，協助您快速檢視論文筆記中的關鍵字關聯性與共現模式。若您不需要此功能，可直接跳過本階段。

功能簡介

本功能可用於快速檢視：

- **Node (節點)**：關鍵字 (Tag)
- **Node size (圓圈大小)**：paper count (含該 tag 的論文篇數)
- **Edge (邊)**：兩個 tag 在同一篇論文中共同出現的次數
- **Edge width (線粗)**：weight (共現次數)

不需要網路：DataviewJS 讀取的是 vault 內的本機 .md 檔案，再用 canvas 在 Obsidian 內繪圖 (離線可用)。

1. 資料夾與檔案放置 (重要)

1) 您的筆記檔案位置 (來源資料)

請確認所有 Papernote 產生的筆記都在：

```
C:\papernote\_note\
```


2) 網路圖檔案 (Network.md)

關鍵字來源網路圖檔案：

- `MyKey_Network.md`：使用 **My Keywords** (您手動維護的關鍵字)

請將這檔案放在同一個資料夾：

```
C:\papernote\_note\MyKey_Network.md
```

 **注意**：Network 檔案本身會在程式中被排除 (避免被當成 paper 筆記掃描)。

2. 安裝 Obsidian 與 Dataview 外掛

1) 安裝 Obsidian (桌面版)

請先安裝 Obsidian Desktop (Windows 版本)。

- **下載連結**：<https://obsidian.md/>

2) 啟用 Community plugins (社群外掛)

1. 打開 Obsidian
2. 左下角  Settings
3. 進入 **Community plugins**
4. 關閉 **Safe mode** (允許社群外掛)

3) 安裝 Dataview

1. Settings → Community plugins → **Browse**
2. 搜尋：Dataview
3. Install → Enable (安裝後立刻啟用)

3. Dataview 必要設定 (一定要做)

1. Settings → Dataview
2. 找到並開啟：
 - **Enable JavaScript queries** (允許 DataviewJS)
3. 建議 (可選)：
 - 若您的 vault 很大，Dataview 重新索引會較慢；這是正常現象
 - 您只要確保 `_note` 在同一個 vault 內即可

4. 用正確方式開啟 Vault (最常見錯誤)


您的 `_note` 路徑是：C:\papernote_note

所以您的 vault 建議直接開在：

```
C:\papernote
```

正確開法 (Windows Obsidian)

1. Obsidian 左下角 (或啟動畫面) 選：**Open folder as vault**
2. 指到：C:\papernote (不是 `_note`)
3. 開啟後在左側檔案總管中，應該看得到 `_note` 資料夾

 如果您遇到「Files in `_note`: 0」，幾乎都是 vault 開錯層級 (例如只開到 `_note` 或其他資料夾)，導致 Dataview 找不到 `_note`。請確保開啟的是包含 `_note` 的上層資料夾。

5. 如何操作互動網路圖 (MyKey)

1) 開啟網路圖筆記

到 `_note` 資料夾，開啟 `MyKey_Network.md`

2) 切換到 Reading view (閱讀模式)

右上角切換：**Reading view** (閱讀模式)

DataviewJS 在 Source mode (編輯模式) 通常不會完整渲染圖表。

3) 正常情況會看到

- 上方：互動網路圖 (canvas)
- Total papers analyzed: N (總共分析幾篇)
- Top co-occurrence pairs 表格
- Tag frequency (papers containing tag) 表格

4) 互動方式 (滑鼠)

- **拖曳節點**：移動單一 node
- **縮放**：滾輪 or Zoom+/Zoom- 按鈕
- **Fit**：自動縮放到畫面剛好符合畫布
- **Freeze layout**：停止力導向排版 (圖不再自動跑)
- **Hover**：滑鼠移到 node 會顯示 label / count / degree 資訊
- **Resume**：恢復力導向排版
- **力導向排版 (force-directed layout)**：是一組數學上的推拉規則，把網路圖排得比較好看不重疊：

斥力 (repulsion)：所有節點彼此「推開」，避免擠成一坨

引力／彈簧力 (spring/attraction)：有連線的節點彼此「拉近」，讓相關的靠在一起

中心拉回 (centering force)：把整團往畫面中心拉，避免越跑越遠

阻尼 (damping)：讓它慢慢穩定下來，不要一直震盪

恢復力導向排版 = 讓這些推拉規則重新開始運算，節點會再次自動調整位置

6. 關鍵字來源說明 (My Keywords)

My Keywords (您手動維護)

在每篇 note 裡找這一行 (您自己填 tag)：

- `**My Keywords:**
 #PV #CE ...`

您只要在這一行後面維護 `#tag` 即可 (大小寫視為同一個，`#PV`、`#pv`、`#Pv` 全部視為同一個 tag，最後在圖上以小寫 `#pv` 形式出現)。

💡 **提醒**：My Keywords 欄位預設是空白，需由您根據需要手動輸入可辨識的關鍵字。建議在完成 Note 模式後，依照論文內容手動標註關鍵字，以便後續使用 Aggregate 與網路圖功能。

📁 7. 常見問題排除（最常見的 4 個）

1) DataviewJS 內容顯示成「純文字」沒有執行

- 檢查是否真的放在：
 - 三個反引號開頭： ````dataviewjs`
 - 三個反引號結尾： `````
- 檢查 Dataview 設定： Enable JavaScript queries 是否已開

2) 圖完全不出現 / 空白

- 先確認 vault 開在 `C:\papernote`（能看到 `_note`）
- 再確認 `_note` 裡確實有 paper notes（不是只有 Network 檔）
- 再確認 Network 檔案名稱有被排除（避免自掃描）

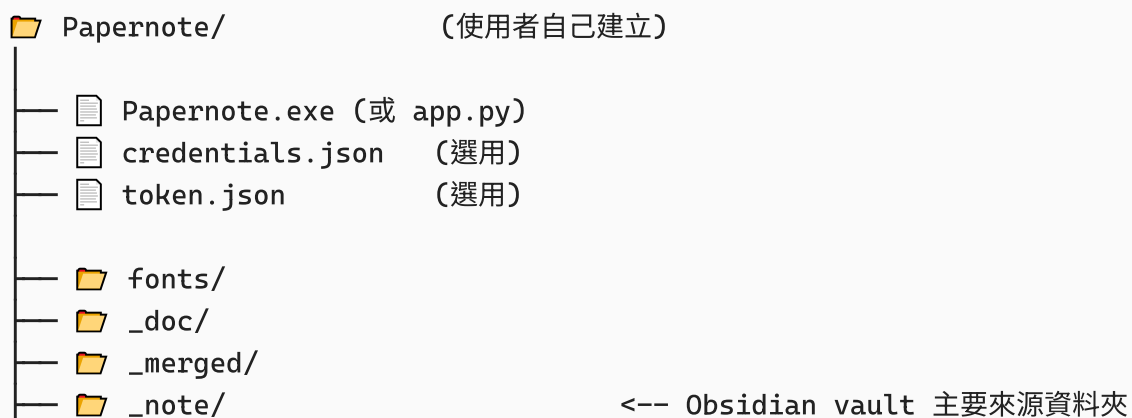
3) 一直轉圈很久、layout 跑不完、全部擠一起

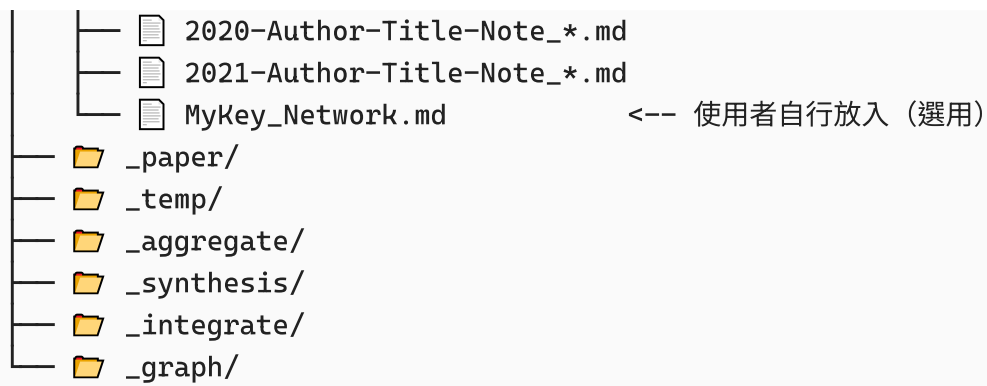
- 先按 **Freeze**，再按 **Fit**
- 或調小 `topEdges`（只顯示前 N 條最重的邊）
- 或提高 `minWeight`（例如只畫共現 ≥ 2 的邊）

4) isolate（孤立節點）不見

- isolate 沒邊時很容易被力場推走；您目前版本建議 isolate 初始「固定（pin）」在右側，之後可手動拖動並固定。

📁 總結：完整運作的資料夾結構圖（含 Obsidian）





建議工作流程：

1. 使用 Papernote 完成論文的 Note 模式分析
2. 手動在每篇筆記的 My Keywords: 欄位填入關鍵字標籤 (例如：#PV、#CE)
3. 將 MyKey_Network.md 放入 _note 資料夾
4. 使用 Obsidian 開啟 C:\papernote 作為 vault
5. 在 Obsidian 中開啟 MyKey_Network.md，切換到 Reading view 查看互動網路圖

Getting Started Last Updated: March 1, 2026.

faq

Prohemsyne © FAQ

基本設定與 API

Q: Gemini API 是免費的嗎？

A: 是的，Google 目前提供免費層級 (Free Tier)，對於個人閱讀論文的用量來說非常足夠。但請注意不要將您的 API Key 分享給他人，以免額度被用光。

Q: 為什麼每次打開程式都要輸入 API Key ？

A: 為了安全起見，我們預設不儲存您的金鑰。

(進階使用者)：如果您希望自動載入，可以在 Windows 中設定環境變數，名稱設為 GEMINI_API_KEY，值設為您的金鑰。程式啟動時會自動讀取。

Q: 出現 Error 403: access_denied ？

A: 這表示您的 Google Cloud 專案還在「測試模式」。請回到 Google Cloud Console 的「OAuth 同意畫面」，點擊「發布應用程式 (Publish App)」即可解決。

Q: 電腦裡出現很多 _MEI 開頭的資料夾？

A: 這是軟體執行時產生的暫存資料夾，通常位於 C:\Users\[使用者名稱]\AppData\Local\Temp_MEIxxxx。

這些是程式運作時的必要暫存檔，關閉程式後若未自動消失，您可以定期手動刪除這些資料夾以釋放硬碟空間，這不會影響軟體的正常運作。

Q: 本系統串接了哪些外部 API ？各自用途為何？

A: 以下是 app 內使用的外部 API 彙總表，供進階使用者與維運參考。

API	正確名稱／端點	在 app 內用途
GenAI	Google GenAI SDK (google.genai)	Note / Analyze / Integrate / Graph 的生成與串流 (含關鍵字調和)。
Grounding	Gemini Tool : GoogleSearch()	僅 Note Stage 6 的 Recent Papers 即時搜尋。
Embedding	Gemini embed_content，模型 gemini-embedding-001	僅 Graph mode 關鍵字語義再合併。
Scite	POST https://api.scite.ai/tallies	Note 模式：依 DOI 取 Smart Citations，填 Scite_ 與 Stage 7。

API	正確名稱／端點	在 app 內用途
Google Drive	Drive API v3 + OAuth 2.0 (drive.file)	筆記 TXT、PDF 副本及各流程存檔到 papernote/_note、papernote/_paper 等。

檔案與資料夾

Q: 如何根據某一篇已知 paper 找到相似 paper 給 Prohemsyne 分析？

A: 可以使用 Citation Mapping Tool，例如 ResearchRabbit, Inciteful 或 Google Scholar 找到符合關鍵字的 paper。

在有安裝 Zotero (及 Zotero Connector Plugin、Inciteful Zotero Plugin、ZotMoov Plugin) 時，自動擷取文獻到 Zotero，ZotMoov 設定 Copy Files to C:\papernote_temp，有文獻進到 Zotero 時，會自動複製副本到 _temp，即可使用 PaperNote 開始分析。

Q: 如何將筆記MD檔案同步至 Obsidian？

A: 若要連結 Obsidian，請使用 Obsidian 的『開啟資料夾為儲存庫 (Open folder as vault)』功能，並直接選擇 C:\papernote_note 資料夾即可。

_synthesize, _integrate 路徑也可同樣方式設定。

Q: 檔案命名規則為何？

A:

- **Note** : YYYY-Author-PaperTitle-Note_YYYYMMDD_HHMMSS.md
- **PDF 副本** : YYYY-Author-PaperTitle-Paper.pdf
- **Merged PDF** : YYYY-Author-PaperTitle-Merged_YYYYMMDD_HHMMSS.pdf
- **Aggregate** : aggregate_tag_{tag}_{數量}_{時間戳記}.md
- **Synthesis** : synthesis_tag_{構念名稱}_{論文數量}_{時間戳記}.md
- **Integrate** : integrate_{檔名標題}_{時間戳記}.md

Q: Synthesize 和 Integrate 模式有什麼檔案名稱限制？

A:

- Synthesize 只接受檔名含 "aggregate" 的檔案
- Integrate 只接受檔名含 "synthesis" 的檔案

Q: 手動上傳與 Auto Flash/Random Flash 的 PDF 檔案處理方式有何不同？

A:

- **手動上傳的 PDF：**
 - 系統會複製一份到 `_paper/` 資料夾（重新命名）
 - 原始上傳的檔案**不會被刪除**，仍保留在原位置
- **Auto Flash / Random Flash 的 PDF：**
 - 系統會複製一份到 `_paper/` 資料夾（重新命名）
 - `_temp/` 資料夾中的原始檔案會**自動被刪除**，避免重複分析
- 所有 PDF 副本都會依統一格式重新命名：`YYYY-Author-PaperTitle-Paper.pdf`

Note 模式

Q: Note 模式的 Flash Report 會執行哪些 Stage ？

A: Flash Report 會自動依序執行 Stage 1-8 編譯完整報告。

Q: Note 模式的 Scite 數據如何取得？

A: 系統會自動從 Stage 7 封面頁的 DOI 欄位查詢 Scite API，取得引用統計數據。

Q: 封面頁的 Scite_：📖 / 💬 / ✅ / ⚠️ 分別代表什麼？怎麼解讀？

A: Scite_ 是系統依據論文 DOI，向 Scite.ai 的 Smart Citations 統計查詢後，回填在封面頁的快速指標。

顯示格式：

- 📖 **Total**：Scite 收錄的「總 Smart Citations 數」（總被引用的量級概覽）
- 💬 **Mentioning**：引用文獻「提及/引用但不明確支持或反駁」的次數（中性提及）
- ✅ **Supporting**：引用文獻「支持」該研究主張/發現的次數
- ⚠️ **Contrasting**：引用文獻「對比/反駁/呈現相反證據」的次數

解讀建議（教學/閱讀情境）：

- 📖 高但 ✅/⚠️ 低：多數是一般性引用或背景提及，較少直接對主張表態。
- ✅ 明顯高：該研究常被後續工作當作支持證據引用（但仍建議回查引用脈絡）。
- ⚠️ 有值：代表存在「相反/對比」引用，值得優先讀那些反向證據來建立邊界條件（boundary condition）。

注意事項：

- 若 DOI 未能被系統偵測、或 Scite 沒有該 DOI 的統計，Scite_ 可能顯示 N/A 或留空。
- 這些數字是「引用句子的分類統計」而非論文品質分數；仍需回到引用脈絡與研究設計本身判讀。

Aggregate 模式

Q: Aggregate 模式有哪些操作方式？

A: Aggregate 模式提供兩種操作方式，適合不同使用情境：

方式 A：基本聚合（僅建立 aggregate 檔案）

1. 在 PaperSynthesize 主介面上方的「Tag for Aggregation」輸入框中輸入關鍵字（例如：PV 或 #PV）
2. 點擊「Aggregate」按鈕
3. 系統自動掃描 _note 資料夾，找出所有 My Keywords: 行包含該關鍵字的筆記檔案
4. 系統自動產生 aggregate 檔案並儲存至 _aggregate/ 資料夾
5. 若已啟用 Google Drive 同步，檔案會自動上傳至雲端

適合情境：想先檢查聚合結果，再手動決定是否進行 Synthesize 分析

方式 B：自動聚合 + Synthesize 分析（一鍵完成）

1. 在輸入框中輸入 {關鍵字} ({構念名稱}) 格式（例如：PV (perceived value)）
2. 點擊「Aggregate」按鈕
3. 系統會自動執行：
 - 步驟 1：聚合筆記檔案（產生 aggregate 檔案）
 - 步驟 2：自動切換到 Synthesize 模式
 - 步驟 3：自動執行 Phase 0 → Phase 1 → Phase 2 → Phase 3 → Export 完整流程

適合情境：已確定要對某構念進行完整文獻綜合，想一次完成所有步驟

Q: 如何使用「自動聚合 + Synthesize 分析」功能？

A:

關鍵是使用特殊的輸入格式：{關鍵字} ({構念名稱})

範例：

- 輸入：PV (perceived value)
- 輸入：Customer (customer engagement)
- 輸入：#Trust (brand trust)

重要提醒：

- 關鍵字部分（括號前）用於匹配 My Keywords: （例如：PV 或 #PV）
- 構念名稱部分（括號內）會自動填入 Synthesize 的 Focal Construct 欄位
- 必須使用英文括號 ()，不可使用全形括號 ()

- 若只輸入關鍵字（不含括號），則只執行基本聚合，不會自動進入 Synthesize

Q: Aggregate 模式如何匹配關鍵字？

A:

- 系統會在 `_note` 資料夾中掃描所有 `.md` 檔案
- 查找每個檔案中包含 `My Keywords:` 的行
- 匹配時不區分大小寫，且支援帶或不帶 `#` 的格式
- 只要 `My Keywords:` 行中包含輸入的關鍵字即視為匹配
- 範例：輸入 `PV` 可匹配 `My Keywords: #PV, #Value` 或 `My Keywords: pv, Customer`

Q: Aggregate 模式產生的檔案格式為何？

A:

- 檔名格式：`aggregate_tag_{tag}_{數量}_{時間戳記}.md`
- 檔案內容：每篇論文之間以 `---\nPAPERNOTE_PAPER_BREAK\n---` 分隔
- 儲存位置：本機 `_aggregate/` 資料夾，並自動上傳至 Google Drive（若已啟用）

Q: 如何設定 My Keywords？

A:

- 在 Note 模式產生的筆記檔案中，`My Keywords:` 欄位預設為空白
- 使用者需手動開啟筆記檔案（`.md`），在 `My Keywords:` 後填入關鍵字標籤
- 建議格式：`My Keywords: #PV, #Customer, #Value`（使用 `#` 前綴，以逗號分隔）

Q: 如果找不到任何匹配的筆記檔案怎麼辦？

A:

系統會顯示錯誤訊息：「找不到任何 `My Keywords:` 含有「{關鍵字}」的筆記。」

可能原因與解決方法：

- **未設定 My Keywords**：請確認 `_note/` 資料夾中的筆記檔案已手動填入 `My Keywords:` 欄位
- **關鍵字拼寫錯誤**：檢查輸入的關鍵字是否與筆記檔案中的 `My Keywords:` 一致
- **`_note` 資料夾不存在**：確認程式目錄下是否有 `_note/` 資料夾
- **關鍵字格式問題**：系統不區分大小寫和 `#` 符號，但關鍵字本身必須完全匹配（例如：輸入 `PV` 無法匹配 `Perceived`）

Q: 為什麼 My Keywords 的維護很重要？它有哪些用途？

A: `My Keywords` 是 Prohemysne 系統中連接「單篇論文筆記」與「跨文獻分析功能」的關鍵樞紐，扮演著「知識標籤索引」的角色。雖然 Note 模式產生的筆記檔案中，`My Keywords:`

欄位預設為空白，但建議您在閱讀論文的 Note 後，立即手動填入關鍵字標籤，以便後續使用多種進階功能。

My Keywords 的三大用途

1. Aggregate 模式（論文筆記聚合）

- **功能：**根據 My Keywords 自動找出相關論文筆記，並彙整成 aggregate 檔案
- **運作方式：**系統會掃描 `_note/` 資料夾中所有 `.md` 檔案，匹配 My Keywords: 行中包含的關鍵字
- **範例：**
 - 您在 5 篇論文筆記中都標註了 My Keywords: #PV, #EXP, #SAT
 - 當您在 Aggregate 輸入框輸入 PV 並點擊「Aggregate」
 - 系統會自動找出這 5 篇筆記，產生 `aggregate_tag_PV_5_*.md` 檔案
- **重要性：**若未維護 My Keywords，Aggregate 功能將無法找到任何匹配的筆記，後續的 Synthesize 分析也無法進行

2. Synthesize 模式（單一構念文獻綜合）

- **功能：**透過 Aggregate 產生的檔案，對單一構念進行深入文獻分析
- **連結關係：**Synthesize 的輸入來源（aggregate 檔案）完全依賴 My Keywords 的標註品質
- **影響：**My Keywords 標註得越精確、一致，Synthesize 分析的文獻範圍就越完整，綜合報告的品質也越高

3. Obsidian 關鍵字共現網路圖（選用功能）

- **功能：**視覺化呈現關鍵字之間的共現關係與網路結構
- **運作方式：**
 - **MyKey_Network.md：**讀取所有筆記檔案中的 My Keywords: 欄位
 - 計算關鍵字節點（node）：含該 tag 的論文篇數決定圓圈大小
 - 計算共現邊（edge）：兩個 tag 在同一篇論文中共同出現的次數決定線粗
- **範例：**
 - 若 3 篇論文都標註了 My Keywords: #PV, #SAT
 - 另 2 篇論文標註了 My Keywords: #PV, #EXP
 - 網路圖會顯示 PV 節點較大（5 篇）、PV-SAT 的邊較粗（3 次共現）
- **重要性：**若未維護 My Keywords，網路圖將無法正確呈現研究主題的關聯性與熱點分布

建議的 My Keywords 維護流程

1. 標註時機

- **最佳時機：**完整閱讀 Note 對論文有特定的想法後，立即開啟筆記檔案填入 My Keywords
- **原因：**此時您對論文內容的理解最清晰，能準確判斷核心構念與關鍵主題

2. 標註原則

- **精確性**：選擇論文的核心構念與主要變數（例如：#PV、#Trust、#Engagement）
- **一致性**：相同概念在不同論文中使用相同標籤（例如：統一使用 #PV 而非混用 #PerceivedValue、#Value）
- **適量性**：建議每篇論文標註 3-5 個關鍵字，避免過多或過少
- **格式**：建議使用 # 前綴並以逗號分隔（例如：My Keywords: #PV, #Trust, #Customer）

3. 標註格式範例

在 Note 檔案的封面頁找到這一行：

```
**My Keywords:**
```

手動填入關鍵字

```
**My Keywords:** #PV, #SAT, #EXP
```

4. 維護建議

- **定期回顧**：當累積一定數量的論文筆記後（例如：每 10 篇），檢查 My Keywords 的一致性
- **標籤統一**：建立自己的「關鍵字對照表」，確保相同概念使用相同標籤
- **AI 或 Author keywords 參考**：可參考系統自動產生的 AI 或 Author Keywords: 欄位，但建議根據自己的研究需求調整

兩種關鍵字來源的差異

Prohemsyne 系統提供兩種關鍵字來源，各有不同用途：

- **AI 或 Author Keywords**（系統自動產生）：
 - 來源：由 AI 從論文中提取作者關鍵字或核心概念
 - 用途：供 AuthorKey_Network.md 使用，呈現原始論文的關鍵字分布
 - 特性：自動產生，無需手動維護
- **My Keywords**（使用者手動維護）：
 - 來源：由使用者根據研究需求手動標註
 - 用途：供 Aggregate 模式、Synthesize 模式、MyKey_Network.md 使用
 - 特性：需要手動維護，但更符合個人研究脈絡與知識管理需求

建議：優先維護 My Keywords，因為它直接影響 Aggregate、Synthesize 與個人化網路圖的使用效果。

未維護 My Keywords 的後果

- **Aggregate 模式無法運作**：系統找不到匹配的筆記檔案，無法產生 aggregate 檔案
- **Synthesize 模式無法進行**：缺少 aggregate 檔案作為輸入，無法執行文獻綜合分析
- **網路圖無內容**：MyKey_Network.md 無法讀取關鍵字，網路圖將顯示空白或僅有零星節點
- **知識管理效率低**：無法快速找出相關論文筆記，需手動逐一檢查檔案內容

總結：My Keywords 的維護是 Prohemsyne 系統發揮完整功能的基礎，建議將其納入日常的論文閱讀與筆記整理流程中，養成「讀完即標註」的習慣，以建立個人化的學術知識網路。

Synthesize 模式

Q: Synthesize 的 Phase 0 是否必須執行？

A: 建議執行，但非強制。Phase 0 可確保變數名稱統一，提高後續階段的一致性。

Synthesize 與 Integrate 模式比較

Q: PaperSynthesize 和 PaperIntegrate 有什麼差異？

A: 兩者在分析流程中處於不同階段，分析深度與輸出目標也不同：

PaperSynthesize (Intra-construct Vertical Synthesis)

- **中文名稱**：單一構念文獻綜合
- **分析層次**：單一構念的「垂直深入」分析 (Intra-construct = 構念內部)
- **輸入資料**：Aggregate 檔案 (包含多篇論文筆記)
- **分析重點**：
 - 針對**單一核心構念** (Focal Construct) 進行深入文獻分析
 - 統一變數名稱 (Phase 0 : Construct Harmonization)
 - 逐篇提取關鍵資訊 (Phase 1 : Paper-level Table)
 - 跨研究綜整共通模式與分歧點 (Phase 2 : Cross-study Synthesis)
 - 提出該構念的研究架構與假說 (Phase 3 : Research Framework And Hypotheses)
- **輸出檔案**：synthesis_tag_{構念名稱}_{論文數量}_{時間戳記}.md
- **輸出內容**：單一構念的完整文獻綜合報告
- **範例**：分析「Perceived Value」這個構念在多篇論文中如何被定義、測量、與其他變數的關係

PaperIntegrate (Inter-construct Lateral Integration)

- **中文名稱**：多構念整合與論文章節撰寫
- **分析層次**：多個構念的「橫向整合」分析 (Inter-construct = 構念之間)
- **輸入資料**：多個 Synthesis 檔案 (每個檔案代表一個構念的綜合報告)

- **分析重點：**
 - 整合**多個構念**，構建完整的研究模型
 - 整合多個 synthesis 檔案中的構念 (Phase 1：Construct Harmonization)
 - 建構 Hayes PROCESS 模型 (Phase 2：Framework Construction)
 - 推導跨構念的研究假說 (Phase 3：Hypotheses Development)
 - 撰寫完整的論文章節 (Phase 4：Chapter 2 Literature Review、Phase 5：Chapter 1 Introduction)
- **輸出檔案：** `integrate_{檔名標題}_{時間戳記}.md`
- **輸出內容：** 完整的研究模型與論文章節 (Chapter 1 + Chapter 2)
- **範例：** 整合「Perceived Value」、「Customer Engagement」、「Brand Trust」三個構念，建構 X→M→Y 的研究模型，並撰寫緒論與文獻回顧

Q: Synthesize 和 Integrate 在分析流程中的前後關係是什麼？

A: 兩者是連續的分析階段，Synthesize 的輸出是 Integrate 的輸入：

完整分析流程：

1. Note 模式
 - ↓ 產出：單篇論文筆記 (`_note/`)
2. Aggregate 模式
 - ↓ 產出：依關鍵字聚合的論文筆記 (`_aggregate/`)
3. Synthesize 模式 (Intra-construct Vertical Synthesis)
 - ↓ 輸入：aggregate 檔案
 - ↓ 分析：針對單一構念進行垂直深入分析
 - ↓ 產出：單一構念的 synthesis 報告 (`_synthesis/`)
4. Integrate 模式 (Inter-construct Lateral Integration)
 - ↓ 輸入：多個 synthesis 檔案
 - ↓ 分析：跨構念的橫向整合，建構完整研究模型
 - ↓ 產出：完整的研究架構與論文章節 (`_integrate/`)

關鍵要點：

- 必須先完成至少 2-5 個構念的 Synthesize，才能進行 Integrate
- Synthesize 是「深度」分析 (單一構念的各種面向)
- Integrate 是「廣度」整合 (多個構念之間的關係與模型)
- Synthesize 產出的 `synthesis_*.md` 檔案，會成為 Integrate 的輸入檔案

Q: 何時該用 Synthesize，何時該用 Integrate？

A:

使用 Synthesize (PaperSynthesize) 的時機：

- 當您想深入了解**單一構念**在文獻中的定義、測量、前因後果時
- 當您需要整理某個構念的研究發現與共通模式時
- 當您想建立某個構念的研究架構與假說時
- **範例情境**：「我想徹底了解『Perceived Value』這個構念」

使用 Integrate (PaperIntegrate) 的時機：

- 當您已完成多個構念的 Synthesize，想建構完整研究模型時
- 當您需要整合多個構念之間的因果關係 (X→M→Y) 時
- 當您想撰寫完整的論文章節 (Chapter 1、Chapter 2) 時
- **範例情境**：「我已經分析完『Perceived Value』、『Customer Engagement』、『Brand Trust』，現在想建構一個完整的研究模型，並撰寫論文」

建議順序：

1. 先對每個感興趣的構念分別執行 Synthesize (例如：先分析 PV，再分析 CE，再分析 Trust)
2. 確認已有 2-5 個構念的 synthesis 報告
3. 再使用 Integrate 整合這些構念，建構完整研究模型

Integrate 模式

Q: 智能探索Top5 和智能整合#1 有什麼差異？

A:

- **智能探索Top5 (輔助決策模式)：**
 - AI 會分析所有候選構念組合，推薦 Top 5 (依評分排序)
 - 使用者可查看每個推薦的詳細資訊 (評分細目、角色對應、架構描述等)
 - 使用者可手動選擇任一推薦組合套用，再執行 Integrate 分析
 - 適合：想比較多個選項、需要深入了解推薦理由的使用者
- **智能整合#1 (一鍵完成模式)：**
 - AI 會自動套用 Top 1 推薦 (評分最高的組合)
 - 立即自動執行完整 Integrate 流程 (Phase 1 → Phase 2 → Phase 3 → Phase 4 → Phase 5 → Export)
 - 無需人工介入，直接產出完整研究架構報告
 - 適合：想快速獲得最佳推薦、信任 AI 判斷的使用者

Q: 智能探索的評分標準是什麼？

A: 智能探索使用 5 個維度來評估構念組合，各維度 0-2 分，總分 0-10：

1. **模型可行性 (0-2 分)：**能否清楚對應到 Hayes PROCESS Model 4/6/7/14

2. **因果鏈合理性 (0-2 分)**：能否講出合理的「為什麼 X 會影響 Y」機制
3. **互補性與非重複 (0-2 分)**：構念彼此互補，不是同義或高度重疊
4. **研究新穎/潛在貢獻 (0-2 分)**：有理論或實務貢獻，不是太教科書或太直覺
5. **可操作化/可測量性 (0-2 分)**：有機會量化或操作（問卷/實驗/二手資料）

每個推薦組合都會顯示各維度的分數與理由，方便使用者評估。

Q: 智能探索會考慮所有 synthesis 檔案嗎？

A: 不一定。智能探索有以下篩選機制：

- **自動去重**：相同構念（依檔名判斷）只會保留最新時間戳記的檔案
- **數量限制**：若候選構念超過 40 個（預設值），系統會優先選擇 paper_count 較高的前 40 個構念送入 AI 分析
- **檔名要求**：只掃描 _synthesis/ 資料夾中檔名含有 "synthesis" 的 .md 檔案

若候選過多，系統會顯示警告訊息，使用者可自行調整 MAX_CONSTRUCTS_FOR_LLM 參數（需修改程式碼）。

Q: 如果手動上傳檔案和智能探索推薦同時存在，系統會使用哪個？

A: 系統會依照以下優先順序決定使用哪個來源：

1. **手動上傳檔案**（最高優先）：若使用者手動上傳了 synthesis.md 檔案，永遠優先使用手動上傳的檔案
2. **智能探索推薦**（次優先）：若未手動上傳檔案，則使用「最後一次套用」的智能探索推薦

這個設計確保使用者的手動選擇永遠優先於 AI 推薦。

Q: 智能探索的矛盾旗標是什麼？會影響評分嗎？

A:

- **矛盾旗標**：當 AI 偵測到構念之間存在「矛盾/互斥」情況時（例如：因果方向相反、核心定義相反、同一角色定位互換等），會在推薦結果中明確標示
- **顯示內容**：包含矛盾點描述（issue）與研究機會（research_opportunity）
- **影響評分**：**不會影響評分**。矛盾旗標是提醒使用者注意潛在的研究議題，但不會因此降低組合的總分
- **研究價值**：有些矛盾點反而能成為有價值的研究機會，例如探討邊界條件（boundary condition）或調節效果（moderation）

Q: 為什麼智能探索使用 Temperature 0.8？

A:

- **Temperature 0.8**：屬於「高溫區」設定，適合需要創意與流暢性的任務

- **選擇理由**：智能探索需要 AI 具有較高的創意性來探索構念組合的「可能性與新穎性」，而非僅產出保守或教科書式的組合
- **不受影響**：此設定為固定值，不受 Sidebar 的 Global Temperature 設定影響
- **對比**：正式 Integrate 的 Phase 1 (Construct Harmonization) 使用 Temperature 0.3 (需要精確性)，Phase 4/5 (章節撰寫) 使用 Temperature 0.8 (需要流暢寫作)

Q: 智能探索的推薦可以修改嗎？

A: 可以。您有以下選項：

- 在 Top 5 推薦中選擇任一組合套用 (可重複套用不同推薦)
- 手動上傳 synthesis.md 檔案覆蓋推薦 (手動上傳優先)
- 每次套用新推薦會覆蓋前一次的選擇 (以「最後一次套用」為準)

Q: 為什麼手動上傳「和智能探索 Top 5 第一名同一批」synthesis 檔案，多數情況下會得到不同的架構？

A: 概念上，輸入的檔案相同，但系統給 AI 的**任務**不同，所以產出的架構可能不同。

差異的細微之處與原因：

1. 智能探索 (第一輪)

AI 讀的是候選構念的摘要／摘錄，回答的是：「這些構念裡，哪一個組合最適合做成研究架構？」

輸出是一個**建議**，例如：「#1：Model 6+7，X=A、M1=B、M2=C、Y=D、W=E」，以及架構描述等。

2. 套用 Top 5 第一名 (或智能整合#1) 時

系統除了帶入那幾份 synthesis 的**完整內容**，還會把智能探索給出的 **model、roles、架構描述** 一併存下來，並在 Phase 1~5 的指示中**強制寫明**：「請依照這個規格產出。」

所以 Phase 1~5 的 AI 是在**實作**你選定的架構，產出會與 Top 5 第一名的建議一致。

3. 手動上傳同一批檔案時

系統**不會**帶入任何「預先選定的架構」；Phase 1~5 的指示只有：「請根據這些 synthesis 內容做構念調和、建構研究架構、發展假說、撰寫章節。」

也就是說，AI 會**再從同一批文件重新推論**一次：「該用哪個 Hayes model、哪些變數該當 X/M1/M2/Y/W/V？」

這等於做了**第二次、且沒有被約束要對齊第一次**的推論。

4. 為何第二次推論常會不一樣？

- 兩次推理彼此**獨立**：第一次的結論 (例如 Model 6+7) 沒有被當成第二次的**必遵規格**，模型有空間選成 Model 4、或把某個構念放到不同角色。
- 模型有 **Temperature** (非 0)，同一段文字多推幾次本來就可能得到不同結論。
- 智能探索讀的是**摘錄**，Phase 1~5 讀的是**完整內文**；脈絡與資訊量不同，也可能導出不同的因果與變數角色。

實務建議：

- 若希望最終架構與智能探索 Top 5 第一名一致：請使用「套用此組合」或「智能整合 #1」，不要改用手動上傳同一批檔案。
- 若希望 AI 自由從這批檔案再推一次架構（可能得到與 #1 不同的 model／角色）：請用手動上傳同一批 synthesis 檔案即可。

一般操作

Q: 可以跳過某些 Phase/Stage 嗎？

A: 不建議。每個 Phase/Stage 都依賴前一個的結果，跳過可能導致錯誤。

Q: 如何切換不同模式？

A: 使用側邊欄的按鈕：

- 「Note」：切換到 PaperNote 模式
- 「Synthesize」：切換到 PaperSynthesize 模式
- 「Integrate」：切換到 PaperIntegrate 模式

Q: References 如何處理？

A:

- Note：References 會自動從 PDF 中提取
- Synthesize：從 aggregate.md 中提取 APA 引用，自動附加到 Phase 3 結果後
- Integrate：系統會自動從 Phase 2-5 中提取被引用的文獻，僅包含實際被引用的條目

Citation + LOCK Double Reference 學術證據鏈

Q: 什麼是可追溯、可稽核、可回查的 Citation + LOCK Double Reference 的學術證據鏈？

A: Prohemsyne 的所有「文獻式主張」（例如：研究發現、理論論證、假說推論、量表來源）一律採用 Citation + LOCK 雙重引用格式，確保每一句關鍵敘述都能「從文字 → 回到證據層 → 精準定位」。

Citation 用於學術寫作格式；LOCK 用於回查證據座標（區塊代碼/序號/頁碼微段），讓每個主張都能從輸出一路追溯回原始 Paper 的對應位置。原始證據層以 Papernote 產出的單篇筆記檔（-Note.md）為準，必要時再回到原始 PDF / Merged PDF 核對（若資訊不足則以 p=N/A 標示）。

1. 什麼是 Double Reference？

- **Citation (APA in-text citation)**：用於學術寫作的標準引用格式（例如：Lin et al., 2007），讓讀者知道「這句話引用哪篇研究」
- **LOCK (Evidence Pointer)**：用於「稽核與回查」的指針，提供更細的定位資訊，讓你能回到原始內容所在位置（例如：筆記區段/頁碼/微段落）
你可以把 LOCK 理解為：每一個引用點的「證據座標」

2. LOCK 標註的結構（格式）

Full LOCK 固定格式：

```
[LOCK:<Paper_Citation_Narrative>|<Section>|<n>|p=<micro-page-range>]
```

範例：

```
[LOCK:Lin et al. (2007)|C|1|p=642.1]
```

各欄位意義：

- **<Paper_Citation_Narrative>**（例：Lin et al. (2007)）：文獻識別字串（通常與 APA citation 一致，用於鎖定是哪一篇 paper）
- （例：C）：來源區塊代碼（對應系統輸出筆記的區段，例如該篇 -Note.md 中的「C. Research Background」）
- （例：1）：同一 Section 內的第 n 個「證據段落塊」索引（evidence/text block index），用於在該 Section 快速定位
計數規則：空行分隔的段落視為 1 塊；條列清單每一個 bullet item 視為 1 塊；同一 Section 自 1 起算
- **p=**（例：p=642.1）：頁碼 + 微段索引（micro-segment index）定位標記；其中 642 為頁碼，.1 為該頁的 micro-segment index（系統用於回查的頁內分段序號）
可為單點（如 642.1）或範圍（如 642.1–642.3）。若來源不足無法判定頁碼則標示 p=N/A

備註：micro-segment index 用於回查證據，通常接近原文的段落塊，但因 PDF 抽取與版面結構差異，不保證與 paper 的 paragraph 完全一致。最終仍以回到 PDF 的原文位置核對為準。

3. 如何用 LOCK 追溯引用來源（實際回查流程）

當你在 Synthesize / Integrate 的輸出中看到一個主張後面帶有 LOCK：

1. **先看 Citation**：確認這句話引用哪篇研究（作者—年份）
2. **再看 LOCK**：記下「Section + n + p=」，它會告訴你：這句話在該篇 -Note.md 的哪個 Section、哪個證據段落塊索引，以及對應原始 paper 的頁碼與微段索引位置
3. **回到對應單篇筆記檔 (-Note.md)**：用「|Section|n|」快速找回原先摘錄或推論的位置
4. **需要稽核時**：再依 p=... 回到原始 PDF / Merged PDF 定位段落，完成「文字—證據」的閉環驗證

4. Double Reference 的重要性

- **教學用途**：你可以把每個討論題、每個假說推導，都快速連回原文證據，避免「憑印象講文獻」
- **研究用途**：寫論文時可降低引用錯置、來源混淆、或跨篇整合時的歸因錯誤
- **可稽核性**：任何一句關鍵主張都能被回查與追驗，符合高標準的學術可靠性要求

Q: Note、Synthesize、Integrate 的 instructions 對「禁止挪用思想」與「A/B 雙引」做了哪些限制？

A: Prohemsyne 在三份外部 instructions (papernote_instructions.txt、paperanalyze_instructions.txt、paperintegrate_instructions.txt) 中，依模式職責分別規範如下：

- **Note 模式 (papernote_instructions.txt)**
Note 僅對單一上傳的原始論文做忠實摘要，不涉及「綜合多篇文獻」或「改寫成新論點」。instructions 要求：若原文中作者引用他人 (如 Author B)，摘要時須保留該 in-text citation，勿將該想法寫成焦點論文作者原創。Note 不負擔「判斷是否需同時引 A 與 B」的責任，只負責如實反映該篇論文的敘述。
- **Synthesize 模式 (paperanalyze_instructions.txt)**
輸入為多篇筆記 (aggregate.md)，AI 會解讀與改寫多篇來源。置頂規則明訂：**【禁止挪用思想 (No Appropriation) 與 A/B 雙引】**。當來源顯示某概念或論點為「作者 A 引用自作者 B」或「源自 B」時，嚴禁僅歸屬給 A；須同時引用 A 與 B，或明確標示「概念源自 B (見 A 之引用)」，並可建議使用者補上 B 的完整引用。
- **Integrate 模式 (paperintegrate_instructions.txt)**
輸入為 synthesis.md，產出 Chapter 1/2 與模型。同樣在置頂規則中訂定：**【禁止挪用思想與 A/B 雙引】**。當來源顯示「A 引用 B」或「源自 B」時，嚴禁僅歸屬給 A；須同時引用 A 與 B，或標示源自 B。對僅有 in-text、無 full reference 的作者 (即可能列於 Missing References 者)，若概念實為 B 原創而經 A 引用，敘述時仍須同時標示 A 與 B，不得僅引 A。

Q: 如何使用 LOCK 溯源到可追溯的最早來源？

A: 使用者可依 **【LOCK:...】** 依序回查，直到可追溯的最早來源為止：

1. 在 **Synthesize 或 Integrate 輸出中**：找到該段敘述的 APA in-text citation 與對應的 **【LOCK:...】**。
2. **解讀 LOCK**：從 LOCK 中的 <Paper_Citation_Narrative>、<Section>、<n>、p= 得知該句對應哪一篇 paper、哪一區塊、哪一段落、以及原始頁碼 (若有)。
3. **回到該篇筆記 (-Note.md)**：依 Section 與段落索引，在對應的 -Note.md 中找到摘錄或推論的原文位置。
4. **回到原始 PDF**：依 p= 的頁碼與微段，在原始 paper.pdf 或 Merged PDF 中定位段落，比對參考文獻與 in-text citation (如誠信電子報所建議的「回到原文、比對參考文獻與 in-text citation、判斷是 A 原創還是 A 引用 B」)。

因此，**可追溯的最早來源**在多數情況下是「原始論文的某一頁／某一段」；若該段本身又引用其他文獻 (B)，則使用者可依該論文之 References 與 in-text 對照，進一步判斷是否需補引 B。

Q: LOCK 機制如何幫助降低 AI 幻覺 (hallucination) 與溯源成本？

A: LOCK 機制在以下兩方面有明確幫助，但有其限制：

降低「假文獻」幻覺

instructions 要求所有文獻式主張必須對應到**已提供的來源** (note → synthesis → integrate)，且僅能引用/LOCK 至已提供之文獻。輸出時須同時附上 APA citation 與 LOCK 指針，較難憑空造出「不存在的作者／論文」或假 DOI，有助於降低假引用、假文獻的風險。

提高使用者判斷可行性

使用者可依【LOCK:...】回查 note.md → 原始 PDF，進行「回到原文、比對參考文獻與 in-text citation、判斷是 A 原創還是 A 引用 B」的步驟。若沒有 LOCK，使用者很難知道某句話該對應哪一段原文，溯源成本高；有 LOCK 後，可快速定位到筆記區段與原始頁碼，大幅降低溯源成本。

限制

LOCK 只保證「這句話對應這份來源」，**不自動解決**「概念其實是 A 引用 B，應同時引 A 與 B」的判斷。真正的「誰是原創、要不要補引 B」仍須使用者依 LOCK 回查後自行判斷。

小結

LOCK 能**降低 AI 幻覺**、並**提高使用者自行判斷的可行性**，但**無法取代**使用者做「引文追溯」與「補充 B」的判斷。

Q: Integrate 輸出的 Missing References 是否代表「AI 沒有挪用思想」？實務上應如何解讀與處理？

A: 需要分開理解，並搭配使用者檢查：

Missing References 的成因

Note 僅忠實摘要單一論文，筆記中會自然出現該論文中已有的 in-text citation (其他作者，如 Lin, 2005)；但 note.md 通常**不包含**這些「其他論文」的 full reference (full reference 僅存在於原始 paper.pdf 的 References 區)。Synthesize 與 Integrate 讀取 note/synthesis 時，會沿用這些 in-text citation 撰寫內容，但**無法**為僅在 note/synthesis 以 in-text 出現的作者造出 full reference，因此程式會將這類引用列為 **Missing References**，提醒使用者補上。

是否代表「AI 沒有挪用思想」？

「這些 missing 來自 AI 正確沿用 note/synthesis 的 in-text 引用、且未捏造 full reference」，可視為在**這些項目上沒有造假**的證據；但**並不自動等於**「AI 在敘述上沒有挪用思想」(例如僅歸屬給 A 而未標示 B)。要完整降低挪用思想的風險，仍須依賴：(1) instructions 中對「A 引用 B 時須同時引 A 與 B」的嚴格要求，以及 (2) 使用者對 Missing References 的逐一溯源與補引時的判斷。

實務建議

Missing References 清單正是使用者應優先檢查的對象：在補上完整條目時，一併確認該概念是否源自「被引用的作者 B」，若然，則在正文與 References 中同時補上 B，避免僅引 A 而造成的挪用疑慮。

Graph 模式（關鍵字網絡分析）

本節導讀：想了解關鍵字從哪來、何時用 mapping/embedding → 見下方「前置：Mapping 調和關鍵字與 Embedding 再合併」；想了解怎麼跑分析、會產出哪些檔、邊的 weight 怎麼算 → 見「執行分析與輸出檔案」；想了解報告內容與何時可產出 → 見「完整網絡分析報告」；想了解重疊社群偵測、K-Clique 與 Modularity 的互補 → 見「重疊社群偵測 (K-Clique / CPM)」；想了解各項數字的意義與應用 → 見「節點與整體指標說明」與「各項網絡指標的對照與應用」；想了解兩兩節點是否結構相似、角色是否相同 → 見「對等性分析」。

前置：Mapping 調和關鍵字與 Embedding 再合併

Q: 什麼是 Graph 模式的「Embedding 再合併」？

A: 「Embedding 再合併」是在 **Mapping 調和關鍵字**（即介面上的「Mapping 調和關鍵字」按鈕所執行的 LLM 調和）之後的選用步驟。系統會使用 **Gemini Embedding 模型 (gemini-embedding-001)** 對「標準詞」（即 keyword_mapping_*.json 的 value）計算語義向量，再以**餘弦相似度**判斷哪些標準詞語義接近、可合併為同一個代表詞。合併結果存成 keyword_embedding_*.json；做圖時若啟用 Embedding 再合併，節點名稱會使用「embedding_merged」而非僅「LLM 標準詞」，可進一步減少節點數、收斂同家族概念（例如不同寫法的 engagement、experience、value）。

Q: Embedding 語義再合併門檻（滑桿）如何設定？

A: 滑桿範圍為 **0.75~0.90**，預設 **0.82**。代表「餘弦相似度 \geq 此值才合併」：

- **門檻較低（例如 0.75）：**合併較多，節點更少，但可能出現過度合併（例如 experience 被併入 engagement、food 相關詞被併入不相關概念）。
- **門檻較高（例如 0.85~0.90）：**只合併非常接近的寫法，節點數減少較少，但較不易錯合。
- **建議：**首次可維持預設 0.82；若發現合併過多或過少，在介面直接調整滑桿即可，無須改程式。論文或關鍵字變多時可依當次結果微調。

Q: Mapping 調和關鍵字與 Embedding 的排序、分批邏輯是什麼？

A: 為使同義／相近關鍵字盡量在同一批被處理、並與前批結果對齊，系統採用以下設計：

- **排序：**原始關鍵字依「關鍵字字串」排序（再依頻率為次要鍵）後再送給 **Mapping**（LLM 調和）或做 Embedding。拼寫相近、同前綴的詞（例如 consumer-、customer-）會相鄰，較容易在同一次 **Mapping 調和**／合併中對齊成同一標準詞。

- **Mapping 分批**：若關鍵字數超過單批上限（約 200 筆），會自動分成多批呼叫 LLM。**第二批起**會把「前幾批已產生的標準詞」一併傳給 LLM，要求同義詞對齊到既有標準詞，避免出現「第一批用 customer engagement、第二批用 consumer engagement」的平行標準。
- **Embedding 分批**：若標準詞數超過單批上限（約 250 筆），會分批計算 embedding 並做相似度合併。**第二批起**會先與「前幾批的代表詞」比對相似度，達門檻即合併到該代表詞；未達門檻者再於本批內聚類，新產生的代表詞會加入清單供下一批對齊。因此同義詞會盡量收斂到同一代表詞。

Q: keywordembedding.json 與 keywordmapping.json 有什麼關係？

A:

- **keywordmapping*.json**：由 **Mapping 調和關鍵字**（LLM）產出，格式為「原始關鍵字 → 標準詞」的單一對照。
- **keywordembedding*.json**：在 **Mapping** 產出的 mapping 檔基礎上，對「標準詞」做 embedding 相似度合併後產出；格式為物件，每個 key 為原始關鍵字，value 為 {"llm_standard": "標準詞", "embedding_merged": "再合併後的詞"}。時間戳記與對應的 mapping 檔一致（例如 keyword_mapping_20260209_235256.json 與 keyword_embedding_20260209_235256.json）。
- 做圖時：**未勾選** Embedding 再合併 → 使用 Mapping 產出的 mapping 檔（原始 → 標準詞）；**勾選** Embedding 再合併 → 使用 embedding 檔（原始 → embedding_merged）。

Q: 做圖時什麼時候用 Mapping 產出的 mapping、什麼時候用 embedding？

A: 完全由介面勾選決定：

- **未勾選「Embedding 再合併」**：建圖一律使用 **Mapping 調和關鍵字**產出的 keyword_mapping_*.json（原始關鍵字 → LLM 標準詞）。
- **勾選「Embedding 再合併」**：建圖使用 keyword_embedding_*.json（原始關鍵字 → embedding_merged）。若目前已有 Mapping 產出的 mapping 檔但尚無對應時間戳的 embedding 檔，系統會依現有 mapping 當場計算 embedding 合併並存檔，再以該結果做圖。

Q: Embedding 再合併的選用時機？何時勾選才值得？

A: Embedding 會對所有標準詞呼叫 API 計算語義向量，會消耗 token；若 **Mapping 調和關鍵字**已將同義詞收斂得很一致，門檻 0.82 下往往只多合併少數幾筆，邊際效益有限。建議以下選用時機：

- **建議勾選**：當您覺得節點還是太多、或懷疑還有同義詞沒併好（例如不同批次的 Mapping 留下寫法略異的標準詞）時，再啟用 Embedding 再合併，可進一步收斂節點。
- **建議先不勾選**：可先用「**Mapping 調和關鍵字**」跑一次，看節點數與節點名稱；若結果已滿意，則無須多花 token 跑 Embedding。若發現節點過多或仍有明顯同義詞分散，再勾選

「Embedding 再合併」重跑即可。

如此可依實際需求決定是否啟用，避免不必要的 API 用量。

執行分析與輸出檔案

Q: 執行分析會產出哪些檔案？同一對關鍵字在多篇論文共現時，邊的 weight 怎麼算？

A: 執行分析會掃描 `_note/` 中的筆記、從 **Author Keywords** 建圖，並產出以下檔案（同一 timestamp，存於 `_graph/`）：

- `AuthorKey_Network_{nodes}_{edges}_{timestamp}.graphml`（圖結構，Gephi Lite 可直接開啟）
- `AuthorKey_Nodes_{nodes}_{timestamp}.csv`
- `AuthorKey_Edges_{edges}_{timestamp}.csv`
- `modularity_{nodes}_{edges}_{群組數}_{timestamp}.csv`
- `kclique_k{K}_{nodes}_{edges}_{社群數}_{timestamp}.csv`（K-Clique 重疊社群偵測結果，見「重疊社群偵測」說明）
- `AuthorKey_NodeIndices_{nodes}_{edges}_{timestamp}.csv`
- `AuthorKey_Network_{nodes}_{edges}_{timestamp}.png`（可依 Min Weight、Top N 篩選繪圖，不影響上述檔案）

同一對關鍵字在多篇論文共現時：圖上只會有一條無向邊，**weight = 共現的論文篇數**（系統會累加，不會產生多條重複的邊）。例如關鍵字 A 與 B 在三篇論文中都曾同時出現，則邊 (A, B) 的 **weight = 3**。

完整網絡分析報告

Q: 什麼是「完整網絡分析報告」？

A: 完整網絡分析報告是一份由 AI（Gemini）根據當前關鍵字網絡的結構化資料自動產出的**場域層級分析報告與閱讀策略建議**，以繁體中文撰寫，存成 `.md` 檔案於 `_graph/` 資料夾。報告內容包含：Modularity 摘要、網絡總覽（Network Overview）、核心節點構念分析（Core Constructs）、橋接節點（Bridging Keywords）、**重疊社群（Overlapping Communities，若有 K-Clique 檔案）**、關鍵橋接共現對、邊陲型態（Peripheral Patterns）與閱讀策略建議（Reading Strategy Recommendations），協助使用者快速掌握關鍵字網絡的整體結構與重點節點。報告中凡列出節點或共現對的區塊，均依指定方式排序，並會在該節內具體說明排序方式。**兩節排序依據不同：Core Keywords 依 Degree Centrality（度中心性）與 Frequency（頻率）排序（與 Node-Level Indices 表一致）；Overlapping Communities 依所屬 K-Clique Cluster 數與 Frequency 排序，兩者順序可能不同。關鍵橋接共現對依邊中介中心性（edge betweenness）降序。**

Q: 產出報告需要先「執行分析」嗎？可以使用現有檔案嗎？

A: 不需要在本次工作階段重新執行分析。系統會依 `_graph/` 資料夾中「同一 timestamp」且齊全的一組檔案來產出報告，只要該組檔案存在即可。也就是說，您大多數時候可以直接用現有的 **GraphML**、**Nodes/Edges CSV**、**Modularity CSV**、**NodeIndices CSV** 點擊「產出【完整網絡分析】報告」，無須從 Mapping 調和關鍵字、Embedding 再合併或執行分析從頭跑一遍。產出時程式會從磁碟讀取這些檔案（例如讀取 GraphML 計算連通分量、讀取 NodeIndices 與 Modularity 摘要），組好資料後送給 AI 撰寫報告。

Q: 報告依據哪些檔案？缺檔會怎樣？

A: 報告依賴以下檔案檔名時間戳記一致（同一組分析產出）：

- AuthorKey_Network_{nodes}_{edges}_{timestamp}.graphml
- AuthorKey_Nodes_{nodes}_{timestamp}.csv
- AuthorKey_Edges_{edges}_{timestamp}.csv
- modularity_{nodes}_{edges}_{群組數}_{timestamp}.csv
- AuthorKey_NodeIndices_{nodes}_{edges}_{timestamp}.csv
- kclique_k{K}_{nodes}_{edges}_{社群數}_{timestamp}.csv（若有，報告會納入 Overlapping Communities 節）

前五個為必備；若缺少任一個，按鈕會無法使用或顯示錯誤。若 `_graph/` 內有多組（多個 timestamp）檔案，系統只會選 timestamp 最新的一組；若該組有缺檔，只會報錯並列出缺檔，不會自動改用次新或其他組。K-Clique CSV 若存在則報告會額外撰寫重疊社群分析。請先完成一次「執行分析」產出完整一組檔案後，再產出報告。

Q: 報告中的節點層級分析涵蓋哪些節點？其餘節點如何呈現？

A: 完整網絡分析報告中的節點層級逐筆分析（核心節點分析、橋接節點），依 **Degree Centrality**（度中心性）與 **Frequency**（頻率）排序，固定取前 30 個節點的關鍵字；此篩選範圍與 `AuthorKey_NodeIndices_*.csv` 中依 `degree Centrality` 與 `frequency` 排序後的順序一致。**Overlapping Communities** 節則依所屬 **K-Clique Cluster** 數與 **Frequency** 排序，與 **Core Keywords** 順序不同。其餘節點不會在報告中逐一點名，而以**整體描述**（如邊陲型態、分布與邊陲傾向）呈現。報告中每個節點的解讀須綜合表列之各項指標（如頻率、PageRank、度中心性、特徵向量中心性、參與係數、角色、結構洞限制、群聚係數、K-核數、有效規模、群內排名等），依多指標提供足夠 insight，避免解讀過於精簡。若需查閱所有節點的完整指標，請開啟報告開頭所列之 `NodeIndices CSV` 檔案。

節點與整體指標說明

Q: 路徑上各節點的指標分別代表什麼意思？

A: 在 Graph 模式的「進階分析：關鍵字路徑探索」功能中，系統會計算路徑上每個關鍵字的多項網絡指標。以下是各指標的詳細說明：

1. 頻率 (Frequency)

項目	說明
定義	該關鍵字出現的論文篇數
數值範圍	1 ~ ∞ (正整數)
計算方式	統計 _note/ 資料夾中， Author Keywords: 欄位包含該關鍵字的論文數量
高值代表	常見/熱門的研究關鍵字，在您的文獻庫中頻繁出現
低值代表	較少被使用的關鍵字，可能是新興主題或特定領域專用

2. PageRank

項目	說明
定義	由 Google 創始人 Larry Page 與 Sergey Brin 發明的節點重要性演算法
數值範圍	0 ~ 1
計算方式	迭代計算：節點的重要性 = 連接到它的節點的重要性加權總和
核心概念	「被重要節點連接的節點，本身也重要」
高值代表	該關鍵字被其他重要關鍵字頻繁連結，在知識網絡中具有高影響力
低值代表	該關鍵字較少被重要關鍵字連結，或位於網絡邊緣

公式：

$$PR(A) = (1-d)/N + d \times \sum [PR(T_i) / C(T_i)]$$

- d：阻尼因子 (damping factor)，通常設為 0.85
- N：網絡中的總節點數
- T_i ：連接到節點 A 的節點
- $C(T_i)$ ：節點 T_i 的出度 (連結數)

3. 度中心性 (Degree Centrality, degC)

項目	說明
定義	節點的連結數量占最大可能連結數的比例
數值範圍	0 ~ 1
計算方式	$degC(v) = degree(v) / (n - 1)$ ，其中 n 為網絡總節點數
高值代表	該關鍵字與許多其他關鍵字有共現關係，是網絡中的「活躍節點」
低值代表	該關鍵字的共現關係較少，與其他主題的連結有限

解讀參考：

degC 數值	解釋
≥ 0.50	高度活躍：與超過半數的關鍵字有共現關係
0.25 ~ 0.49	中度活躍：與約 1/4 ~ 1/2 的關鍵字有共現關係
0.10 ~ 0.24	低度活躍：共現關係有限
< 0.10	邊緣節點：僅與少數關鍵字有共現關係

4. 中介中心性 (Betweenness Centrality, betC)

項目	說明
定義	節點位於其他節點對之間「最短路徑」上的比例
數值範圍	0 ~ 1
計算方式	$betC(v) = \sum[\sigma(s,t$
核心概念	衡量節點作為「橋樑」或「中介者」的重要性
高值代表	該關鍵字是連接不同主題群組的「橋樑概念」，資訊流通必須經過它
低值代表	該關鍵字不是主要的資訊傳遞路徑，移除它不會顯著影響網絡連通性

公式符號說明：

- $\sigma(s, t)$ ：節點 s 到 t 的最短路徑總數
- $\sigma(s, t|v)$ ：經過節點 v 的最短路徑數

解讀參考：

betC 數值	解釋
≥ 0.20	高度中介：是連接不同主題群組的關鍵橋樑
0.10 ~ 0.19	中度中介：在部分路徑中扮演橋樑角色
0.05 ~ 0.09	低度中介：偶爾作為中介節點
< 0.05	非中介：不是主要的資訊傳遞路徑

5. 接近中心性 (Closeness Centrality, cloC)

項目	說明
定義	節點到網絡中所有其他節點的平均最短路徑長度的倒數
數值範圍	0 ~ 1
計算方式	$cloC(v) = (n - 1) / \sum d(v, u)$ ，其中 $d(v,u)$ 為 v 到 u 的最短路徑長度
核心概念	衡量節點「接近」網絡中心的程度

項目	說明
高值代表	該關鍵字能快速「到達」網絡中的其他關鍵字，資訊傳播效率高
低值代表	該關鍵字距離其他節點較遠，位於網絡邊緣

解讀參考：

cloC 數值	解釋
≥ 0.60	高度接近：位於網絡核心，能快速連接到大多數關鍵字
0.40 ~ 0.59	中度接近：與網絡中心有一定距離
0.25 ~ 0.39	低度接近：位於網絡外圍
< 0.25	邊緣位置：距離網絡核心較遠

6. 參與係數 (Participation Coefficient, P)

項目	說明
定義	衡量節點連結「跨群組分布」的程度
數值範圍	0 ~ 1
計算方式	$P = 1 - \sum (k_{is} / k_i)^2$ ，其中 k_{is} 為連到群組 s 的連結數， k_i 為總連結數
理論基礎	Guimerà & Amaral (2005), Nature
高值代表	該關鍵字連結均勻分布在多個群組，是「跨領域連接者」
低值代表	該關鍵字連結集中在自己所屬群組，是「領域專家」

解讀參考：

P 數值	解釋
≥ 0.80	極高跨群：連結幾乎均勻分布在所有群組
0.62 ~ 0.79	高跨群：擔任連接不同群組的角色
0.30 ~ 0.61	中跨群：部分連結跨越群組邊界
< 0.30	低跨群：連結主要集中在自己所屬群組

💡 參與係數需搭配 **z-score (模組內分支度)** 一起解讀，可將節點分類為 R1-R7 七種角色，詳見後續 FAQ。

7. 結構洞限制 (Structural Holes Constraint)

項目	說明
定義	由社會學家 Ronald Burt 提出，衡量節點受限於其網絡位置的程度
數值範圍	0 ~ 2 (理論上限，實際通常 < 1)
計算方式	$C(i) = \frac{\sum (p_{ij} + \sum p_{iq} \times p_{qj})^2}{\sum p_{ij}}$ ，對所有鄰居 j 求和
核心概念	「結構洞」是指網絡中不相連的群組之間的空隙；佔據結構洞的節點具有資訊優勢
低值代表	該關鍵字佔據「結構洞」位置，連接到彼此不相連的群組，具有資訊中介優勢
高值代表	該關鍵字的鄰居彼此緊密相連，處於「封閉」的網絡結構中，資訊冗餘度高

公式符號說明：

- p_{ij} ：節點 i 投入在鄰居 j 的關係比例 (依連結權重計算)

解讀參考 (注意：此指標低值較佳)：

Constraint 數值	解釋
< 0.30	低限制：佔據結構洞，具有資訊中介優勢，是連接不同主題的「經紀...
0.30 ~ 0.50	中度限制：部分佔據結構洞位置
0.50 ~ 0.70	高度限制：鄰居之間有較多連結，資訊優勢較低
> 0.70	極高限制：處於封閉網絡結構，鄰居彼此高度連結

8. 群聚係數 (Clustering Coefficient)

項目	說明
定義	節點的鄰居之間實際連結數占最大可能連結數的比例
數值範圍	0 ~ 1
計算方式	$CC(v) = \frac{2 \times E_v}{k_v \times (k_v - 1)}$ ，其中 E_v 為鄰居間的連結數， k_v 為鄰居數
核心概念	衡量節點的鄰居是否形成緊密的「小圈圈」
高值代表	該關鍵字的鄰居彼此也經常共現，形成一個緊密的主題群組

項目	說明
低值代表	該關鍵字的鄰居彼此較少共現，該關鍵字可能是連接不同主題的橋樑

解讀參考：

CC 數值	解釋
≥ 0.70	高度群聚：鄰居形成緊密的主題群組
0.40 ~ 0.69	中度群聚：鄰居之間有一定程度的連結
0.20 ~ 0.39	低度群聚：鄰居之間連結較少
< 0.20	極低群聚：鄰居彼此幾乎不相連，該關鍵字可能是橋樑節點

參與係數、結構洞限制與群聚係數的關係：

這三個指標從不同角度衡量節點的網絡結構特性：

網絡結構特性	參與係數 (Participation)	結構洞限制 (Constraint)	群聚係數 (Clustering)	解釋
跨域橋樑	高	低	低	連結跨多群組，鄰居彼此不相連
群內核心	低	高	高	連結集中自己群組，鄰居緊密連結
跨域樞紐	中~高	低~中	低~中	兼具群內與跨群連結；其「群聚係數」與「結構洞限制」的高低，取決於鄰居彼此是否緊密互連：若鄰居互連多→群聚係數較高、結構洞限制較高；若鄰居互連少→群聚係數較低、結構洞限制較低

9. 特徵向量中心性 (Eigenvector Centrality)

項目	說明
定義	節點的重要性不只來自連結數量，還來自「與之連結的節點本身有多重要」
數值範圍	0 ~ 1
計算方式	迭代計算特徵向量：節點的分數與其鄰居分數的加權和成正比
核心概念	「與重要節點連結的節點，本身也重要」
高值代表	該關鍵字與其他結構上重要的關鍵字頻繁共現，屬網絡中的核心語彙

項目	說明
低值代表	較少與核心節點共現，或位於邊緣

解讀：與度中心性（只看連結數）、PageRank（隨機游走）互補；特徵向量中心性強調「與誰連結」，可區分「連很多次要節點」與「連少數但很核心的節點」。

10. K-核數 (K-Core Number / 核心度)

項目	說明
定義	節點所屬的「最大 k-核」的 k 值；k-核為刪除所有度數 < k 的節點後仍連通的子圖
數值範圍	0 ~ 最大度數（正整數）
計算方式	依節點度數迭代剝離，每個節點得到一個 core number
核心概念	衡量節點在「稠密核心」中的深度
高值代表	該關鍵字位於網絡的稠密核心區，即使移除許多低連結節點仍與其他高連結節點相連
低值代表	位於網絡外圍或較不稠密的區域

解讀：可與角色（R1–R7）、群聚係數併看。高 k-核且低參與係數 → 典型群內核心；高 k-核且高參與係數 → 跨群但仍在結構核心。

11. 邊的中介中心性 (Edge Betweenness) 與關鍵橋接共現對

項目	說明
定義	每條邊有多少「最短路徑」經過它；衡量該邊作為橋樑的程度
數值範圍	0 ~ 1（標準化後）
計算方式	對所有節點對計算最短路徑，統計經過該邊的路徑比例
核心概念	高邊中介中心性的邊一但移除，會使許多節點對之間的最短路徑變長或斷開
高值代表	該共現對（關鍵字 A–關鍵字 B）是網絡中的「關鍵橋」，連接不同主題或群組
低值代表	該共現對多數時候不在最短路徑上，橋接意義較低

在報告中的應用：完整關鍵字網絡分析報告會列出「關鍵橋接共現對」區塊，**依邊的中介中心性 (edge betweenness) 由高至低排序**。邊的中介中心性衡量「有多少對節點之間的最短路徑會經過這條邊」；**數值越高**表示該共現對越常擔任網絡中不同主題之間的**必經紐帶**，若該邊不存在或很弱，部分主題間的知識連結會變長、繞道或變弱，故**高值邊對維持網絡連通性具有**

關鍵意義。 解讀各共現對時可說明：這些關鍵字對若在文獻中較少同時出現，主題間的連結會明顯變弱。

12. 群內排名 (Within-Community Rank)

項目	說明
定義	該節點在所屬 Louvain 群組內，依連結數 (degree) 降序排列後的排名
數值範圍	1 ~ 該群組節點數 (正整數)
計算方式	每個群組內將節點依 degree 由高至低排序，排名 1 表示該群組內連結數第一
高意義 (排名數字小)	在該主題群組內是「數一數二」的關鍵字
低意義 (排名數字大)	在該群組內連結較少，屬群組內較邊緣的節點

解讀：與模組內 z-score (z) 互補；z 為標準化、較抽象，群內排名為直觀的「第幾名」。報告中解讀時可寫「在所屬群組內排名第 N」。

13. 富 club 係數 (Rich-Club Coefficient)

項目	說明
定義	整體網絡性質：度數 (degree) $\geq k$ 的節點之間，實際連結數占最大可能連結數的比例 (與隨機期望比較)
數值範圍	0 ~ 1 (或相對隨機網絡的比值)
計算方式	選定 連結數門檻 (degree threshold) k ，取度數 (degree) $\geq k$ 的節點誘導子圖，計算其邊密度並與隨機基準比較。此處 k 為 連結數 (degree) ，不是 0~1 的 度中心性 (degree centrality) 。
高值代表	「核心關鍵字」彼此之間連結特別密，形成「富者愈富」的小圈 (rich-club 現...
低值代表	高度數節點之間連結較分散，核心較不凝聚

k 的用詞說明：報告撰寫時，Rich-club 的 k 應寫成「**連結數門檻 (degree threshold)**」，不得寫成「度中心性門檻」或「度數門檻」。因 k 的數值為整數 (例如 22)，代表「至少與幾個其他關鍵字有共現」的連結數 (degree)，而度中心性 (degree centrality) 為 0~1 的標準化值，兩者不同，寫錯易造成讀者誤解。

解讀：用於 Network Overview 描述整體網絡是否具明顯「富 club」性質，即核心關鍵字是否彼此高度共現。

14. 有效規模 (Effective Size)

項目	說明
定義	由 Ronald Burt 結構洞理論提出，衡量節點的「非冗餘」接觸規模
數值範圍	0 ~ 節點的度數 (無向未加權時： $e = n - 2t/n$ ，n 為鄰居數，t 為鄰居間連結...
計算方式	考慮鄰居之間的冗餘：若鄰居彼此緊密相連，有效規模會小於名義上的連結數
高值代表	該關鍵字擁有較多「非冗餘」的連結，資訊來源較多元
低值代表	鄰居之間重疊度高，有效接觸較少

解讀：與結構洞限制 (Constraint) 互補；Constraint 為「低值較佳」(低=佔據結構洞)，有效規模為「高值較佳」(高=有效連結多)。兩者併看可區分「低限制但有效接觸少」與「低限制且有效接觸多」的節點。

Q: 整體關鍵字網絡模組化 (Modularity) 代表什麼意思？

A: Modularity 是衡量整個關鍵字網絡中「群組結構」明確程度的指標，用於評估網絡是否能被清楚地劃分為不同的主題群組。

基本資訊

項目	說明
定義	網絡中實際群組內連結數與隨機網絡期望值的差異
數值範圍	-0.5 ~ 1
計算方式	先用 Louvain 演算法進行關鍵字分群，再計算 Modularity 值
高值代表	關鍵字群組結構明顯，不同研究主題的關鍵字能被清楚劃分
低值/負值代表	網絡缺乏明確的群組結構，關鍵字之間的連結較為分散或隨機

公式

$$Q = (1/2m) \times \sum [A_{ij} - (k_i \times k_j / 2m)] \times \delta(c_i, c_j)$$

- m：網絡中的總邊數
- A_{ij} ：節點 i 與 j 之間的連結權重 (有連結為 1，無連結為 0)
- k_i 、 k_j ：節點 i 與 j 的度數 (連結數)

- $\delta(c_i, c_j)$ ：若節點 i 與 j 屬於同一群組則為 1，否則為 0
- c_i ：節點 i 所屬的群組

解讀參考

Modularity 數值	解釋
≥ 0.70	強烈群組結構：關鍵字能被非常清楚地劃分為不同主題群組
0.50 ~ 0.69	明顯群組結構：存在可辨識的主題群組
0.30 ~ 0.49	中度群組結構：有一定的主題分群傾向
0.10 ~ 0.29	弱群組結構：主題分群不明顯
< 0.10	無群組結構：關鍵字連結較為隨機，無法清楚劃分主題

Louvain 演算法簡介

Louvain 是一種高效的關鍵字分群演算法，用於將網絡節點劃分為不同群組：

1. **初始化**：每個節點獨自成為一個群組
2. **局部優化**：將每個節點移動到能最大化 Modularity 增益的鄰近群組
3. **群組聚合**：將同一群組的節點合併為「超級節點」
4. **迭代重複**：重複步驟 2-3 直到 Modularity 不再提升


輸出內容

系統會顯示：

1. **模組化程度**：Modularity 數值（如：0.45）
2. **關鍵字分群**：每個群組包含哪些關鍵字（各群組內關鍵字依 **within-community rank** 排序：主要依連結數 degree 降序、次要依頻率 frequency 降序；報告中會說明此排序方式）

範例輸出：

```

 整體關鍵字網絡模組化 (Modularity)
├── 模組化程度：0.45
└── 關鍵字分群：
    ├── 群組 1: machine learning, deep learning, neural network
    ├── 群組 2: data mining, classification, clustering
    └── 群組 3: natural language, text mining, sentiment
  
```

實務應用

- **高 Modularity**：您的文獻涵蓋多個明確的研究主題，各主題之間相對獨立
- **低 Modularity**：您的文獻主題較為聚焦，或不同主題之間有較多交叉連結
- **群組識別**：透過分群結果，可快速了解您的文獻庫涵蓋哪些主要研究主題

進階說明：分群排他性與重疊分群偵測

在使用 Modularity 進行分群解讀時，請留意以下特性：

- **Louvain 演算法的排他性：**

本系統目前採用的 Louvain 演算法屬於「非重疊分群 (Hard Clustering)」。這意味著在計算結果中，**每一個關鍵字只會被分配到一個特定的群組 (Cluster)**，無法同時隸屬於多個群組。這有助於明確劃分主要結構，但可能忽略了跨領域概念的模糊性。

- **重疊社群 (Overlapping Communities) 的偵測：**

若您的研究需要偵測「一個關鍵字同時跨越多個主題群組」的複雜結構（例如：某個核心概念同時屬於「行銷」與「心理學」群組），本系統於執行分析時會一併進行 K-Clique 重疊社群偵測。詳見下方「**Q: 重疊社群偵測 (K-Clique / CPM) 是什麼？如何使用與解讀？**」。


Q: 重疊社群偵測 (K-Clique / CPM) 是什麼？如何使用與解讀？

A: 重疊社群偵測與 Louvain Modularity 分群互補，用來找出同時參與多個緊密子群的橋接關鍵字，揭示場域內跨領域整合的結構。

使用的函數與演算法

- **函數：** NetworkX 套件中的 `networkx.algorithms.community.k_clique_communities`
- **演算法：** CPM (Clique Percolation Method, clique 滲透法)
- **原理：** 以 K-clique (K 個節點兩兩相連的完全子圖) 為基本單元；兩 K-clique 若共享 K-1 個節點則屬於同一社群。關鍵字可同時出現在多個社群中 (重疊)。

使用方法

- 在 Graph 模式執行分析前，可調整參數區的「**K-Clique 社群偵測 K 值**」滑桿 (3~6，預設 4)。
- K=3 表示以「三角形」為基本單元；K 越大，偵測到的社群越緊密但數量可能減少。
- 點擊「 執行分析」後，系統會自動執行 Louvain 分群與 K-Clique 偵測，兩者產出會一併顯示與寫入檔案。

輸出檔案

檔名格式	說明
<code>kclique_k{K}_{nodes}_{edges}_{社群數}_{timestamp}.csv</code>	K-Clique 重疊社群結果 (K 值、各群組關鍵字、重疊節點、未覆蓋節點)

與 Modularity CSV、NodeIndices CSV 等共用同一 timestamp，可與其他檔案對照。

輸出內容解讀

欄位	說明
K-Clique 群組	該列關鍵字所屬的 K-Clique Cluster (CSV 欄位名, 報告與文中一律稱「K-Clique Cluster 1、2、...」)
關鍵字	節點 (關鍵字) 名稱
所屬群組數	該關鍵字屬於幾個 K-Clique Cluster
所屬全部群組	所屬 K-Clique Cluster 的編號 (如「1, 2」表示同時屬於 K-Clique Cluster 1 與 2)

重疊節點：所屬群組數 ≥ 2 ，同時橋接多個 K-Clique Cluster，是跨領域整合的核心概念。

非重疊節點：所屬群組數 = 1，僅屬於單一 K-Clique Cluster。

未覆蓋節點：不參與任何 K-clique 結構，位於網絡稀疏邊陲。

Modularity vs. K-Clique：異、同與互補使用

項目	Modularity (Louvain)	K-Clique (CPM)
分群類型	非重疊 (Hard Clustering)	重疊 (Overlapping)
每個關鍵字	只屬於一個 Cluster	可屬於多個 K-Clique Cluster
結構層次	全域模組化，整體主題切分	局部 clique 滲透，微型緊密子群
輸出焦點	Cluster 1, 2, 3... 的主題邊界	K-Clique Cluster 1, 2, 3...、重疊節點、橋接關係、未覆蓋節點

互補使用：Louvain 提供「主題群組怎麼切」的全局視角；K-Clique 則揭示「哪些關鍵字跨越多個主題、扮演整合角色」。報告中會合併兩者：Core Keywords 與 Bridging Keywords 會納入各節點的重疊狀態；Overlapping Communities 節則逐一說明重疊節點的橋接涵義。

Q: 連通分量 (Connected Components) 是什麼？公式與解讀

A: 連通分量是圖論中的概念，指圖中「彼此可透過邊相連、但與圖其餘部分不相連」的**最大連通子圖**。在關鍵字網絡中，一個連通分量代表一組關鍵字，這組關鍵字之間可以透過共現關係 (邊) 一路相連，但與其他分量內的關鍵字沒有任何共現連結。

定義與計算方式

項目	說明
定義	若從分量內任一點出發，沿著邊可到達該分量內所有節點，且無法到達分量外任何節點，則這些節點構成一個連通分量。

項目	說明
計算方式	系統從圖 (GraphML) 以廣度優先搜尋 (BFS) 或深度優先搜尋 (DFS) 找出所有連通分量；每個節點只會屬於一個分量。

概念式說明 (非單一封閉公式)：連通分量數 = 圖中「極大連通子圖」的個數。極大連通子圖意指：子圖內任意兩節點間存在路徑，且無法再加入任一其他節點仍保持連通。

畫面上與報告中的數字解讀

顯示項目	說明
Connected components = N	整張關鍵字網絡圖被分成 N 塊互不相連的區域。
First component: Size = k_1 , % of nodes = $p_1\%$	最大的一塊 有 k_1 個節點，占全部節點的 $p_1\%$ 。
Second component: ...	第二大的塊的節點數與占比，依此類推。

解讀：若只有 1 個連通分量，表示所有關鍵字透過共現關係可串成一片；若 ≥ 2 ，表示存在至少兩群關鍵字在目前資料中彼此沒有任何共現連結（例如不同脈絡或冷門主題自成一塊）。

執行分析時算的，還是產報告時算的？

產報告時會重算。執行分析時會算一份連通分量用於畫面顯示（模組化區塊的「Connected components = N、First component: ...」），但該結果只存在記憶體 (session)，**沒有寫入任何檔案**。產出報告時，程式會從磁碟讀取 GraphML，**重新計算**連通分量與各塊節點數、占比，再寫入報告開頭與送給 AI 的資料。兩邊數字會一致（因為圖相同），但邏輯上是「報告端獨立從 GraphML 重算」，並非讀取執行分析時儲存的數字。計算為本機執行，資源消耗極低。


報告開頭的「連通塊」與執行分析畫面上的連通分量一樣嗎？

是。報告開頭與執行分析完成後畫面上顯示的連通分量 (Connected components = N、First component: Size = ...、% of nodes = ...) **格式與計算方式一致**；報告端是產出當下從 GraphML 重算，執行分析端是分析完成時從記憶體中的圖計算，兩者結果相同。

Q: 執行分析後畫面上的「整體關鍵字網絡模組化」區塊，各項數字是什麼意思？

A:

正式解釋

執行 Graph 分析後，畫面上「 整體關鍵字網絡模組化 (Modularity)」區塊會顯示以下項目，其定義如下：

畫面上看到的	正式定義
模組化程度 (如 0.47)	Modularity 係數，數值範圍 -0.5 ~ 1，衡量網絡群組結構的明確程度；越高表示關鍵字能被清楚劃分為不同主題群組。
Modularity 係數介於 -0.5 到 1，越高表示社群結構越明顯	係數的範圍與解讀說明。
關鍵字分群：N 個關鍵字群組	經 Louvain 演算法將關鍵字劃分為 N 個群組；同一群組內關鍵字共現較多，群組間連結較少。
Connected components = K	連通分量數為 K，即圖中有 K 塊彼此無邊相連的最大連通子圖。
First component: Size = k_1 ， % of nodes = $p_1\%$	節點數最多的一塊（第一連通分量）包含 k_1 個節點，占全網絡節點的 $p_1\%$ 。
Second component: Size = k_2 ， % of nodes = $p_2\%$	第二大的連通分量包含 k_2 個節點，占 $p_2\%$ ，其餘分量依此類推。

白話解釋

以下用簡單白話說明每一行在講什麼（數值僅為舉例）：

畫面上看到的	白話意思
模組化程度：0.47	一個介於 -0.5 到 1 之間的數字，用來表示「關鍵字能不能被清楚分成好幾團主題」。數字越高，代表「這一團跟那一團」分得越清楚；數字越低，代表關鍵字之間的連結比較像一盤散沙、沒那麼好分群。例如 0.47 就是「有一點分群結構，但不算非常極端」。
Modularity 係數介於 -0.5 到 1，越高表示社群結構越明顯	同上：這個係數的範圍與解讀。不用背公式，只要記得「越高 = 主題分群越明顯」即可。
關鍵字分群：13 個關鍵字群組	系統用演算法把關鍵字自動切成 13 團。每一團裡的關鍵字彼此共現較多、比較像同一主題；團與團之間相對沒那麼緊密。所以「13 個群組」就是在說：目前你的文獻關鍵字被分成了 13 個主題塊。
Connected components = 2	「連通分量」有 2 個。白話說就是：整張關鍵字網絡圖裡，有 2 塊彼此沒有連結的區域。一塊裡的關鍵字可以透過共現關係一路連到同一塊的其它關鍵字，但跨不到另一塊。例如一塊可能是「行銷類關鍵字」，另一塊是「方法論類關鍵字」，兩邊在目前資料裡沒有一起出現過。
First component: Size = 327， % of nodes = 98.49%	第一塊裡面有 327 個關鍵字，佔全部節點的 98.49%。也就是說，絕大多數關鍵字都落在這一塊，是整張圖的「主體」。
Second component: Size = 5， % of nodes = 1.51%	第二塊裡面只有 5 個關鍵字，佔 1.51%。代表有一小撮關鍵字自成一團，和主體那 327 個沒有共現連結，可能是較冷門或不同脈絡的主題。

一句話總結：這塊畫面在告訴你——（1）主題分群有多明顯（模組化程度）、（2）被切成幾團（關鍵字分群數）、（3）整張圖有幾塊互不相連的區域（Connected components）、以及（4）每一塊有多大（Size 與 % of nodes）。數值會隨你的筆記與關鍵字而變，以上僅為舉例說明。

Q: K-Clique 重疊社群分析結果，如何解讀？

A: 執行 Graph 分析後，「重疊社群分析 (K-Clique / CPM)」區塊會顯示 K 值、重疊社群數、重疊節點數、未覆蓋節點數，以及重疊節點列表。以下逐一說明。

範例畫面

K 值：3 | 重疊社群數：38 | 重疊節點數：46 | 未覆蓋節點數：0

重疊節點（橋接多個社群）：

perception：K-Clique Cluster 2, K-Clique Cluster 6, K-Clique Cluster 9

brand：K-Clique Cluster 2, K-Clique Cluster 7, K-Clique Cluster 14

各項數字的白話說明

項目	白話解釋
K 值：3	以「三角形」（3 個節點彼此全相連）為基本單元。K=3 表示每個 k-clique 至少包含 3 個關鍵字，這 3 個關鍵字兩兩都有共現連結。
重疊社群數：38	CPM 偵測到 38 個 K-Clique Cluster （重疊社群）。即圖中有 38 團「至少 3 個節點彼此全相連」的緊密小團體；每一團可透過共享節點與團內其他 clique 相連，但不與其他團相連。
重疊節點數：46	有 46 個 關鍵字同時出現在 2 個以上 K-Clique Cluster 中。這些節點是 橋接節點 ，同時參與多個緊密子群，通常代表跨主題、跨領域整合的關鍵概念。
未覆蓋節點數：0	有 0 個 關鍵字完全不參與任何 k-clique。若為 0，表示每個節點都至少參與一個 k-clique，沒有完全孤立的節點。若大於 0，表示有若干節點位於網絡稀疏邊陲，不屬於任何緊密小團體。

重疊節點列表的解讀

「重疊節點（橋接多個社群）」會逐一列出每個橋接節點及其所屬的 K-Clique Cluster。例如：

- **perception：K-Clique Cluster 2, K-Clique Cluster 6, K-Clique Cluster 9** → 「perception」這個關鍵字同時出現在 3 個不同的緊密子群中，扮演連接這 3 團的橋梁角色。
- **brand：K-Clique Cluster 2, K-Clique Cluster 7, K-Clique Cluster 14** → 「brand」同時出現在 3 個 K-Clique Cluster，是跨領域整合的核心概念。

橋接的 K-Clique Cluster 越多，表示該節點跨越多個主題或子領域，整合程度越高。

與整體節點數的關係

若網絡共有 431 個節點：46 個為重疊節點（橋接 ≥ 2 個 K-Clique Cluster），0 個為未覆蓋節點，其餘 385 個為**非重疊節點**（只屬於 1 個 K-Clique Cluster）。非重疊節點仍參與緊密結構，只是未同時橋接多個 K-Clique Cluster。

Q: 為什麼每次計算 Modularity 的分群結果（群組數）會不太一樣？

A: 這在網絡分析中是**正常的現象**。主要原因在於系統採用的 **Louvain 演算法** 具有「非決定性 (Non-deterministic)」的特性。

以下是導致分群結果些微變動的三個主要原因：

1. 演算法的隨機性：

Louvain 演算法在優化過程中，會**隨機決定**節點的處理順序。在複雜的網絡中，不同的處理順序可能會導致節點被合併到不同的鄰近社群，進而影響最終的群組總數。

2. 多重局部最佳解 (Local Optima)：

複雜的關鍵字網絡通常存在多種「切法」都能達到不錯的模組化分數 (Modularity Score)。演算法可能會在不同的執行次數中，收斂到不同的「局部最佳解」，導致分群邊界出現細微差異。

3. 資料的動態變化：

若您在兩次分析之間新增或修改了筆記（改變了關鍵字共現關係），網絡結構即已改變，分群結果自然會隨之調整。

如何應對？

- **關注核心結構**：儘管邊緣節點的歸屬可能變動，但主要的核心群組（大群）通常是穩定的。
- **多次執行**：若發現某次分群結果特別不合理，可以重新整理或再次執行分析，通常會回歸到最穩定的結構。

Q: K-Clique (CPM) 分析結果是穩定的嗎？

A: 是。K-Clique (Clique Percolation Method, CPM) 為**確定性 (Deterministic) 演算法**，與 Louvain 相反。只要原始資料（網路結構）沒有變動，且設定的參數（如 k 值）相同，兩次分析得到的社群劃分在**結構上**理論上應完全一致。

CPM 的運作邏輯非常固定，主要分為四個標準步驟，這四個步驟都沒有涉及隨機性：

1. **尋找所有最大派系 (Maximal Cliques)**：在圖中找出所有大小至少為 k 的全連接子圖（即 k-clique）。

2. **建立派系圖 (Clique Graph)**：將每一個 k-clique 視為一個新的節點。
3. **連結相鄰派系**：如果兩個 k-clique 共享了 k-1 個共同節點，就在派系圖中將它們連起來。
4. **劃分連通分量**：派系圖中的每一個連通分量 (Connected Component) 就定義為一個社群 (Community)。

使用同一個軟體、同樣的數據、同樣的 k 值進行兩次分析時：

項目	結果
社群內的節點組成	應該是一模一樣的
社群的編號／順序	可能會改變 (取決於程式實作，例如依連通分量大小或迭代順序編號)，但不影響結構本身

Q: 連通分量 (Connected Components) 在關鍵字網絡與 CPM 中分別指什麼？

A: 「連通分量」在圖論中指的是**同一個概念**：圖中「彼此可透過邊相連、但與圖其餘部分不連」的**最大連通子圖**。但系統會在**兩種不同的圖**上使用這概念，因此涵義不同。

核心定義

從圖上任一節點出發，沿著邊可到達該分量內所有節點，且無法到達分量外任何節點，則這些節點構成一個連通分量。亦即：分量內任意兩節點間存在路徑，且無法再加入任一其他節點仍保持連通。

兩種使用情境

情境	圖的結構	連通分量的涵義
報告中的 Connected components	原始關鍵字網絡：節點 = 關鍵字，邊 = 共現關係	一群關鍵字可透過共現關係一路相連，但與其他分量內關鍵字沒有任何共現連結。一塊內關鍵字彼此互通，跨不到另一塊。
CPM 的 Step 4	派系圖：節點 = 每個 k-clique，邊 = 兩 clique 共享 k-1 節點	一群彼此相鄰的 k-clique (可透過共享節點串起來)；每一個連通分量對應一個 K-Clique 重疊社群。

為何 CPM 要用連通分量？

CPM 先建出派系圖 (以 k-clique 為節點、共享 k-1 節點為邊)，再在這個**新圖**上找連通分量。每個連通分量 = 一組「透過共享節點而相鄰」的 clique，這組 clique 的節點聯集就形成一個重疊社群。因此，連通分量概念相同，只是應用的圖不同：前者在**關鍵字圖**上回答「整張網絡分成幾塊」，後者在**派系圖**上定義「重疊社群怎麼切」。

簡例 (k=3)

假設有五個 3-clique：A{1,2,3}、B{1,2,4}、C{1,3,4}、D{5,6,7}、E{5,6,8}。A–B–C 共享節點相連，D–E 相連，但 A–B–C 與 D–E 不相連。派系圖上有兩個連通分量：{A,B,C} 與 {D,E}，分別對應兩個重疊社群。

與既有 FAQ 的對應

報告開頭與畫面上的「Connected components = N、First component: ...」指的是**關鍵字網絡**的連通分量（情境 A）。CPM 步驟 4 的「劃分連通分量」指的是**派系圖**的連通分量（情境 B）。兩者計算邏輯一致，只是作用的圖不同。

Q: Participation Coefficient (參與係數) 是什麼？

A: Participation Coefficient (P, 參與係數) 衡量一個節點的連結「跨社群分布」的程度。

理論基礎： Guimerà, R., & Amaral, L. A. N. (2005). Cartography of complex networks: modules and universal roles. *Nature*, 433(7028), 895-900.

公式：

$$P_i = 1 - \sum (k_{is} / k_i)^2$$

其中：

- k_{is} = 節點 i 連接到群組 s 的連結數
- k_i = 節點 i 的總度數
- N_M = 群組總數
- \sum 對所有群組 $s = 1$ 到 N_M 求和

解讀：

- $P \approx 0$ ：該節點的連結幾乎全部集中在自己所屬的社群內（「地頭蛇」型）
- $P \approx 1$ ：該節點的連結均勻分布在所有社群（「跨界連接者」型）

Q: Within-module Degree Z-score (模組內分支度 z-score) 是什麼？

A: Z-score (z) 衡量一個節點在其所屬社群內的連結程度，相對於同社群其他節點而言。

公式：

$$z_i = (k_i - \bar{k}_s) / \sigma_s$$

其中：

- k_i = 節點 i 與同群組其他節點的連結數
- \bar{k}_s = 該群組內所有節點的平均連結數
- σ_s = 該群組內連結數的標準差

解讀：

- $z \geq 2.5$ ：該節點是其所屬社群的「樞紐 (Hub)」，在社群內連結特別多
- $z < 2.5$ ：該節點是社群內的一般成員 (Non-hub)

Q: 節點角色分類 R1-R7 代表什麼意思？

A: 根據 Guimerà & Amaral (2005) 的理論，結合 **Z-score** 和 **Participation Coefficient** 可將網絡中的節點分為 **7 種角色**：

Non-hub 節點 ($z < 2.5$)：社群內連結不多的一般節點

角色代號	英文名稱	中文名稱	P 範圍	在關鍵字網絡中的意義
R1	Ultra-peripheral	超邊緣節點	$P \leq 0.05$	「孤島概念」幾乎只與自己社群內的少數節點連結，是高度專業化或冷門的關鍵字
R2	Peripheral	邊緣節點	$0.05 < P \leq 0.62$	「領域內概念」主要在自己社群內活動，偶爾與其他社群有連結
R3	Non-hub Connector	非樞紐連接者	$0.62 < P \leq 0.80$	「跨界橋樑」雖然不是社群核心，但擔任連接不同社群的角色
R4	Non-hub Kinless	非樞紐無親緣節點	$P > 0.80$	「全域游離者」連結均勻分布在各社群，不屬於任何特定領域

Hub 節點 ($z \geq 2.5$)：社群內連結特別多的樞紐節點

角色代號	英文名稱	中文名稱	P 範圍	在關鍵字網絡中的意義
R5	Provincial Hub	在地樞紐	$P \leq 0.30$	「領域權威／土皇帝」在自己社群內連結極多，但幾乎不跟外界打交道。這是該特定領域最經典、最排他的核心概念
R6	Connector Hub	連接樞紐	$0.30 < P \leq$	「跨界領袖」既是自己領域的核心，又承擔著連接外部領域的重任。這是最具學術影響力的

角色代號	英文名稱	中文名稱	P 範圍	在關鍵字網絡中的意義
			0.75	概念，能帶動跨領域整合
R7	Kinless Hub	無親緣樞紐	P > 0.75	「全球通／超級中樞」連結極多且分佈極廣，幾乎支撐著整個網絡的運作。在關鍵字網絡中較少見，通常出現在像 Google 這種巨型網絡的首頁節點

Q: Louvain 分群與節點結構指標的組合策略比較？

A: 透過將「關鍵字分群 (Louvain)」與不同的「節點結構指標」結合，可以從不同理論視角解析關鍵字的角色。請參考下表選擇適合您的研究策略：

組合策略	分析焦點	尋找「橋樑/中介」能力	尋找「核心/凝聚」能力	學術慣用領域	理論基礎
Louvain + Participation Coefficient	跨群連結性 (Global Inter-connectivity)	★★★★★ (最直接)	★★★★	複雜網絡、物理、生物資訊	Guimerà & Amaral (2005) Cartographic Roles (Nature 期刊)
Louvain + Structural Holes Constraint	資訊優勢/結構洞 (Meso-level Brokerage)	★★★★★ (最經典)	★★★★	管理學、社會學、組織行為	Ronald Burt (1992) Structural Holes Theory
Louvain + Clustering Coefficient	局部凝聚力 (Local Cohesion)	★★★ (較弱)	★★★★★ (最強)	社會科學一般分析	Watts & Strogatz (1998) Small-World Networks

Q: 各項網絡指標的對照與應用？

A: 以下是 Graph 模式中各項指標的比較與應用場景：

節點層級指標：

指標	英文名稱	衡量什麼	高值意義	低值意義
頻率	Frequency	該關鍵字出現的論文篇數	常見／熱門的研究關鍵字	較少被使用的關鍵字
PageRank	PageRank	被其他重要節點連結的程度（迭代權重）	在知識網絡中具有高影響力	較少被重要節點連結或位於邊緣
度數	Degree	節點直接連結的數量（未標準化）	連結數多	連結數少
度中心性	Degree Centrality (degC)	節點連結數占最大可能連結數的比例	連結廣泛的熱門概念	連結較少的專門概念
特徵向量中心性	Eigenvector Centrality	與「重要節點」連結的程度	結構上的核心語彙	較少與核心節點共現
中介中心性	Betweenness Centrality (betC)	節點作為「橋樑」的重要性	連接不同群組的關鍵概念	非橋樑位置的概念
接近中心性	Closeness Centrality (cloC)	節點到所有其他節點的平均距離	位於網絡中心的概念	位於網絡邊緣的概念
參與係數	Participation Coefficient (P)	連結的跨社群分布程度	跨領域連接者	專注單一領域的概念
模組內 z-score	Within-module z-score (z)	在所屬社群內的重要性	社群內的核心／樞紐	社群內的一般成員
角色	Role (R1-R7)	由 P 與 z 分類的節點角色 (Guimerà & Amaral)	見 R1-R7 各角色定義	—
結構洞限制	Structural Holes Constraint	節點周圍的結構洞程度	鄰居彼此緊密連結（冗餘）	鄰居彼此不連結（佔據結構洞）
群聚係數	Clustering Coefficient	節點鄰居之間的連結密度	鄰居形成緊密小圈子	鄰居之間較少直接連結
K-核數	K-Core Number	節點在稠密核心中的深度	位於網絡稠密核心區	位於外圍或較不稠密區
有效規模	Effective Size	非冗餘的接觸規模 (Burt)	有效連結多、資訊來源多元	鄰居重疊度高、有效接觸少

指標	英文名稱	衡量什麼	高值意義	低值意義
群內排名	Within-Community Rank	在所屬群組內依連結數（主要）與頻率（次要）的排名	排名 1 = 該群組內連結數第一	該群組內較邊緣
社群	community_id	所屬 Louvain 群組編號（ 1-based ，與報告中「Cluster 1、Cluster 2、...」一一對應）	例如 社群 3 = Cluster 3	報告解讀時提及所屬群組一律用「Cluster N」與前文一致
邊中介中心性	Edge Betweenness	該邊位於最短路徑上的程度（邊層級）	關鍵橋接共現對	橋接意義較低

整體網絡指標：

以下為描述**整張關鍵字網絡**的指標（與節點層級指標不同），報告中的 Global Indices 與執行分析畫面上的「整體關鍵字網絡模組化」區塊會呈現這些數值。

指標	英文／說明	高值意義	低值意義
節點數	Nodes	—	整體網絡規模（關鍵字種類數）
邊數	Edges	—	整體連結數量（共現對...）
模組化程度	Modularity (Louvain)	主題群組結構明顯，關鍵字能清楚劃分為不同主題群組	群組結構不明顯或較隨機
關鍵字分群數	Louvain 分群數（N 個關鍵字群組）	—	主題被切成幾團（分群...）
密度	Density	節點間連結較密（實際邊數占最大可能邊數的比例高）	連結較稀疏
連通分量數	Connected components	圖較破碎、多塊互不相連	圖較連通（1 表示所有關鍵字可透過共現串成一...）
第一連通塊占比	First component (% of nodes)	—	最大一塊的節點數占全體比例
平均群聚係數	Average clustering	整體上節點的鄰居之間連結較密、小圈較多	整體上較少局部小圈
富 club 係數	Rich-Club Coefficient	核心關鍵字彼此高度共現（富者愈富）	核心較分散
Rich-club k	富 club 係數所對應的連結數門檻（degree threshold）；k 為連結數，非度中心性	—	係數是在「度數（degree）≥ k」的節點子圖下計算

指標	英文／說明	高值意義	低值意義
K-Clique (CPM)	Clique Percolation Method；重疊社群數、重疊節點數、未覆蓋節點數	重疊社群多表示網絡具局部 K-clique 緊密子群；重疊節點多表示跨領域橋接明顯	未覆蓋節點多表示邊陲較多、網絡較破碎；未偵測到重疊社群表示缺乏 K-clique 結構

應用建議：

- **找核心概念**：看 degC、PageRank、特徵向量中心性高的節點
- **找跨領域橋樑**：看 betC 高、P 高的節點；**關鍵橋接共現對**看邊中介中心性高的邊
- **找領域權威**：看 z 高但 P 低的節點 (R5)；群內排名第 1 且 k-核數高
- **找跨界領袖**：看 z 高且 P 中等的節點 (R6)
- **找結構洞機會**：看 Constraint 低、有效規模高的節點
- **整體網絡性質**：Rich-club 係數高表示核心關鍵字形成小圈；低表示核心較分散

對等性分析

Q: 什麼是對等性分析 (Equivalence Analysis) ?

A: 對等性分析是社會網絡分析 (Social Network Analysis) 中的核心概念，用於判斷網絡中的兩個節點是否在結構上「相似」或「扮演相同角色」。Prohemsyne 的 Graph 模式提供三種對等性指標，協助使用者理解關鍵字在知識網絡中的結構關係。

三種對等性的嚴格程度 (由嚴到寬)：

結構對等性 > 自同構對等性 > 規則對等性
(最嚴格) (最寬鬆)

應用場景：

- **結構對等性**：找出「連接到相同鄰居」的節點，適合分析緊密相關的概念
- **自同構對等性**：找出「結構位置相同」的節點，適合分析在網絡中扮演相似角色的概念
- **規則對等性**：找出「角色功能相似」的節點，適合發現潛在的概念替代或互補關係

Q: 結構對等性 (Structural Equivalence) 的計算公式與含意是什麼？

A: 結構對等性是最嚴格的對等性定義：若兩個節點連接到**完全相同的鄰居集合**，則具有結構對等性。

計算公式

1. Jaccard 相似度 (Jaccard Similarity)

$$\text{Jaccard} = |N(A) \cap N(B)| / |N(A) \cup N(B)|$$

- $N(A)$ ：節點 A 的鄰居集合
- $N(B)$ ：節點 B 的鄰居集合
- n ：交集（共同鄰居）
- U ：聯集（所有鄰居）
- **數值範圍**：0 ~ 1
- **解釋**：共同鄰居數量占所有鄰居數量的比例

範例：

- 節點 A 的鄰居： $\{X, Y, Z\}$
- 節點 B 的鄰居： $\{X, Y, W\}$
- 共同鄰居： $\{X, Y\}$ （2 個）
- 聯集鄰居： $\{X, Y, Z, W\}$ （4 個）
- $\text{Jaccard} = 2 / 4 = 0.50$

2. Pearson 相關係數 (Pearson Correlation)

$$\text{Pearson}(A, B) = \frac{\sum[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{[\sum(x_i - \bar{x})^2 \times \sum(y_i - \bar{y})^2]}}$$

- x_i ：節點 A 的鄰接向量（若 A 與節點 i 相連則為 1，否則為 0）
- y_i ：節點 B 的鄰接向量
- **數值範圍**：-1 ~ 1
- **解釋**：兩節點鄰接向量的線性相關程度

指標解釋

Jaccard 相似度解釋：

Jaccard 數值	解釋
≥ 0.70	高度結構對等 ：兩節點連接到幾乎相同的鄰居群
0.40 ~ 0.69	中度結構對等 ：兩節點有顯著的共同鄰居
0.10 ~ 0.39	低度結構對等 ：兩節點有少數共同鄰居
< 0.10	無結構對等 ：兩節點幾乎沒有共同鄰居

Pearson 相關係數解釋：

Pearson 數值	解釋
≥ 0.70	高度正相關 ：兩節點的鄰接模式高度一致，連接到相似的鄰居群

Pearson 數值	解釋
0.40 ~ 0.69	中度正相關 ：兩節點的鄰接模式有顯著相似性
0.10 ~ 0.39	低度正相關 ：兩節點的鄰接模式有些許相似
-0.09 ~ 0.09	無相關 ：兩節點的鄰接模式無明顯關聯
-0.39 ~ -0.10	低度負相關 ：兩節點的鄰接模式略呈互補（A 連接的，B 傾向不連接）
-0.69 ~ -0.40	中度負相關 ：兩節點的鄰接模式呈現互補傾向
≤ -0.70	高度負相關 ：兩節點的鄰接模式高度互補，連接到不同的鄰居群

💡 提示：在共現網絡中，Pearson 負值較少見，因為關鍵字通常不會呈現「互斥」的共現模式。若出現負值，可能表示這兩個關鍵字屬於不同的研究領域或主題群。

實務含意

- **高結構對等**：兩個關鍵字經常與相同的其他關鍵字一起出現，可能屬於同一研究主題或概念群
- **低結構對等**：兩個關鍵字的關聯脈絡不同，可能屬於不同研究領域或主題

Q: 自同構對等性 (Automorphic Equivalence) 的計算公式與含意是什麼？

A: 自同構對等性比結構對等性寬鬆：若存在圖的**自同構映射** (automorphism) 能將節點 A 映射到節點 B，則兩者自同構對等。換句話說，它們在網絡中的「結構位置」相同，但不一定連接到相同的鄰居。

計算方式

1. Ego Network 同構判斷

```
is_isomorphic(Ego(A), Ego(B))
```

- **Ego(A)**：以節點 A 為中心的自我中心網絡（包含 A 及其所有直接鄰居與它們之間的連結）
- **輸出**：True（同構） / False（不同構）
- **解釋**：兩節點的局部網絡結構是否完全相同（可透過節點重新標記使兩個圖完全一致）

2. 結構特徵相似度 (Structural Feature Similarity)

使用餘弦相似度比較兩節點的結構特徵向量：

$$\text{Cosine}(A, B) = (F_A \cdot F_B) / (||F_A|| \times ||F_B||)$$

特徵向量 F 包含：

特徵	說明
Degree	節點的連結數（鄰居數量）
Clustering Coefficient	節點的群聚係數（鄰居之間的連結密度）
K-Core Number	節點所屬的最大 k-core（衡量節點在核心結構中的深度）
Ego Network 節點數	自我中心網絡的節點數量
Ego Network 密度	自我中心網絡的連結密度

- 數值範圍：0 ~ 1
- 解釋：兩節點在多個結構特徵上的綜合相似程度

指標解釋

情況	解釋
Ego 同構 = 是	完全自同構對等 ：兩節點的自我中心網絡結構完全相同
特徵相似度 ≥ 0.90	高度自同構對等 ：兩節點在網絡中扮演非常相似的結構角色
特徵相似度 0.70 ~ 0.89	中度自同構對等 ：兩節點的結構位置有一定相似性
特徵相似度 0.50 ~ 0.69	低度自同構對等 ：兩節點的結構角色有些許相似
特徵相似度 < 0.50	無自同構對等 ：兩節點在網絡中扮演不同的結構角色

實務含意

- **高自同構對等**：兩個關鍵字在知識網絡中處於相似的「結構位置」，可能扮演相似的角色（如：都是連接不同主題的橋樑）
- **低自同構對等**：兩個關鍵字的結構角色不同（如：一個是核心概念，另一個是邊緣概念）

Q: 規則對等性（Regular Equivalence）的計算公式與含意是什麼？

A: 規則對等性是最寬鬆的對等性定義：兩節點是規則對等的，若它們與「對等的其他節點」有相同類型的關係。這是一種遞迴定義，不要求連接到相同鄰居，只要求在網絡中扮演類似的「角色」。

計算方式

1. 簡化版分數（Simplified Score）

基於鄰居 Degree 分布的相似度：

簡化版分數 = $1 - (\text{歐氏距離} / \text{最大可能距離})$

比較的統計量：

- 鄰居 Degree 的平均值 (Mean)
- 鄰居 Degree 的標準差 (Std)
- 鄰居 Degree 的最小值 (Min)
- 鄰居 Degree 的最大值 (Max)
- 數值範圍：0 ~ 1
- 解釋：兩節點的鄰居在「連結能力」上的分布是否相似

2. REGE 分數 (REGE Algorithm)

REGE (Regular Equivalence) 是社會網絡分析中的經典迭代演算法：

初始化： $R^0(i, j) = 1$ (所有節點對的對等性初始為 1)

迭代更新 (直到收斂)：

$$R^{(t+1)}(i, j) = [\text{match}(i \rightarrow j) + \text{match}(j \rightarrow i)] / 2$$

其中：

$$\text{match}(i \rightarrow j) = (1/|N(i)|) \times \sum \max\{R^t(n_i, n_j) : n_j \in N(j)\}$$

對於每個 $n_i \in N(i)$

- $N(i)$ ：節點 i 的鄰居集合
- $R^t(i, j)$ ：第 t 次迭代時，節點 i 與 j 的規則對等性分數
- 數值範圍：0 ~ 1
- 解釋：若 i 的每個鄰居都能在 j 的鄰居中找到「對等的」對應，則 i 與 j 具有高規則對等性

3. REGE 百分位 (REGE Percentile)

百分位 = $(\text{REGE 分數在所有節點對中的排名} / \text{總節點對數}) \times 100\%$

- 數值範圍：0% ~ 100%
- 解釋：此分數在整個網絡的所有節點對中，排名在前百分之幾

指標解釋

REGE 分數	解釋
≥ 0.90	高度規則對等 ：兩節點與「對等的其他節點」有相同類型的關係，扮演相似的網絡角色

REGE 分數	解釋
0.70 ~ 0.89	中度規則對等 ：兩節點在網絡中扮演部分相似的角色
0.50 ~ 0.69	低度規則對等 ：兩節點的網絡角色有些許相似性
< 0.50	無規則對等 ：兩節點在網絡中扮演不同的角色

實務含意

- **高規則對等**：兩個關鍵字在知識網絡中扮演相似的「功能角色」，即使它們不連接到相同的鄰居。例如：兩個關鍵字可能都是「連接理論與實務的橋樑概念」
- **低規則對等**：兩個關鍵字的功能角色不同。例如：一個是「核心理論概念」，另一個是「應用領域標籤」

Q: 三種對等性有什麼差異？如何選擇使用？

A: 三種對等性從不同角度評估節點關係，適合不同的分析目的：

對等性類型	核心問題	判斷標準	適合分析
結構對等性	「它們連接到相同的人嗎？」	共同鄰居的重疊程度	緊密相關的概念、同一主題群
自同構對等性	「它們在網絡中的位置相同嗎？」	局部網絡結構的同構性	結構位置相似的概念、角色替代
規則對等性	「它們扮演相同的角色嗎？」	與「對等節點」的關係模式	功能角色相似的概念、角色類別

選擇建議：

- 想找「屬於同一主題」的關鍵字：優先看結構對等性（Jaccard）
- 想找「結構位置相似」的關鍵字：優先看自同構對等性（Ego 同構 + 特徵相似度）
- 想找「扮演相似角色」的關鍵字：優先看規則對等性（REGE 分數）

Q: 對等性分析總結表格的「解釋」欄位是如何判斷的？

A: 對等性分析總結表格中的「解釋」欄位，主要根據**主要指標**進行判斷，以下是各對等性類型的詳細判斷準則：

結構對等性（根據主要指標：Jaccard 相似度）

Jaccard 數值	解釋
≥ 0.70	高度結構對等
0.40 ~ 0.69	中度結構對等
0.10 ~ 0.39	低度結構對等
< 0.10	無結構對等

自同構對等性（優先主要指標，再看輔助指標）

自同構對等性的判斷邏輯較特殊，會先檢查主要指標（Ego 同構），若為 True 則直接判定為「完全自同構對等」；若為 False，則再根據輔助指標（特徵相似度）進行判斷：

判斷順序	條件	解釋
1（優先）	Ego 同構 = 是	完全自同構對等
2	Ego 同構 = 否，特徵相似度 ≥ 0.90	高度自同構對等
3	Ego 同構 = 否，特徵相似度 0.70 ~ 0.89	中度自同構對等
4	Ego 同構 = 否，特徵相似度 0.50 ~ 0.69	低度自同構對等
5	Ego 同構 = 否，特徵相似度 < 0.50	無自同構對等

為什麼自同構對等性有特殊判斷邏輯？

因為「Ego Network 同構」是一個布林值（是/否），它代表兩節點的局部網絡結構是否完全相同。這是一個非常嚴格的條件，一旦滿足，即可直接判定為「完全自同構對等」，無需再看其他指標。若不滿足，才需要透過「特徵相似度」這個連續數值來評估相似程度。

規則對等性（根據主要指標：REGE 分數）

REGE 分數	解釋
≥ 0.90	高度規則對等
0.70 ~ 0.89	中度規則對等
0.50 ~ 0.69	低度規則對等
< 0.50	無規則對等

判斷邏輯總結

結構對等性 → 100% 根據主要指標（Jaccard）

自同構對等性 → 優先主要指標（Ego 同構），若為 False 再根據輔助指標（特徵相似度）

💡 提示：總結表格中的「解釋」可能過於簡化，建議研究者參考「主要指標」及「輔助指標」綜合判斷。

Q: 對等性分析的輸出檔案包含哪些內容？

A: 系統會自動將對等性分析結果儲存為 CSV 檔案至 `_graph/` 資料夾。

檔名格式： `equivalence_{source}_{target}_{時間戳記}.csv`

檔案欄位：

欄位	說明
分析項目	基本資訊、結構對等性、自同構對等性、規則對等性
指標名稱	各項指標的名稱 (如：Jaccard 相似度、REGE 分數)
數值	計算結果
說明	指標的解釋與定義

包含內容：

- **基本資訊：** Source 節點、Target 節點
- **結構對等性：** Jaccard 相似度、Pearson 相關係數、共同鄰居數量、解釋
- **自同構對等性：** Ego Network 同構、結構特徵相似度、兩節點的 Degree 和 Clustering Coefficient、解釋
- **規則對等性：** 簡化版分數、REGE 分數、REGE 百分位、解釋

Small World 模式

Q: Small World 的「中心」是接近中心性 (closeness centrality) 嗎？

A: 是。Small World 的「誰是中心、誰是邊陲」排名，依據的是每個關鍵字到最大連通塊內其餘節點的**平均最短路徑距離** (步數)：平均距離**越小**越中心、**越大**越邊陲。這與學術上常用的**接近中心性 (closeness centrality)** 是同一概念。差異僅在呈現方式：Small World 直接顯示**平均距離** (數值越小越中心)，而許多文獻或套件會用其**倒數**表示接近中心性 (數值越大越中心)；排序結果一致。

Q: 什麼時候該用 Small World ? 和 Graph 有什麼不同？

A: Graph 負責產出網絡圖、GraphML、報告與各類指標；Small World 只讀取 `_graph/` 中時間戳最新的一份 GraphML，做本機計算（最短路徑、距離分布、平均距離等），不呼叫 AI。當您已有 Graph 產出的 GraphML，想快速查「兩關鍵字幾步可連」「以某關鍵字為中心時的距離分布與平均距離」「距離某關鍵字 N 步有誰」或「最大連通塊內誰較像核心、誰較邊陲」時使用。操作與三大功能說明請見 **Getting Started** 的「模式六：Small World」。

Q: 為什麼進入 Small World 會顯示「目前沒有可用的關鍵字網絡資料」？

A: Small World 僅使用 `_graph/` 內時間戳最新的 GraphML。若資料夾尚無 GraphML，或尚未在 **Graph 模式** 執行分析並成功產出，即會出現此提示。請先點選側邊欄 **Graph** 完成分析後再使用。

參考文獻

以下為 Prohemsyne 各模式所參考之學術文獻，依作者姓氏排序。**備註**欄標示各篇的使用位置與對應功能。

編號	引用 (APA 7)	備註
1	Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. <i>Journal of Statistical Mechanics: Theory and Experiment</i> , 2008(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008	Graph mode，Louvain 社群偵測。
2	Borgatti, S. P. (1997). Structural holes: Unpacking Burt's density measure. <i>Connections</i> , 20(1), 35–42.	Graph mode，structural holes 指標解析 (constraint、effective size)。
3	Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. <i>Computer Networks and ISDN Systems</i> , 30(1–7), 107–117. https://doi.org/10.1016/S0169-7552(98)00110-X	Graph mode，PageRank。
4	Burt, R. S. (1992). <i>Structural holes: The social structure of competition</i> . Harvard University Press.	Graph mode，constraint、effective size (結構洞)。
5	Colizza, V., Flammini, A., Serrano, M. A., & Vespignani, A. (2006). Detecting rich-club ordering in complex networks. <i>Nature Physics</i> , 2(2), 110–115. https://doi.org/10.1038/nphys209	Graph mode，rich-club coefficient。
6	Daft, R. L. (1995). Why I recommended that your manuscript be rejected and what you can do about it. In L.	Note mode，Q Section。

編號	引用 (APA 7)	備註
	L. Cummings & P. J. Frost (Eds.), <i>Publishing in the organizational sciences</i> (2nd ed., pp. 164–182). Sage.	
7	Freeman, L. C. (1977). A set of measures of centrality based on betweenness. <i>Sociometry</i> , <i>40</i> (1), 35–41. https://doi.org/10.2307/3033543	Graph mode , betweenness centrality (節點、邊中介中心性)。
8	Guimerà, R., & Amaral, L. A. N. (2005). Cartography of complex networks: modules and universal roles. <i>Nature</i> , <i>433</i> (7028), 895–900. https://doi.org/10.1038/nature03288	Graph mode , 參與係數 (P) 、 within-module z-score 、 R1–R7 角色。
9	Hayes, A. F. (2022). <i>Introduction to mediation, moderation, and conditional process analysis: A regression-based approach</i> (3rd ed.). Guilford Press.	Integrate mode , Phase 2 、 Phase 3 、 Phase 4 。
10	Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. <i>Physical Review E</i> , <i>69</i> (2), 026113. https://doi.org/10.1103/PhysRevE.69.026113	Graph mode , Modularity 。
11	Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. <i>Nature</i> , <i>435</i> (7043), 814–818. https://doi.org/10.1038/nature03607	Graph mode , K-Clique / CPM 重疊社群偵測。
12	Poole, M. S., & Van de Ven, A. H. (1989). Using paradox to build management and organization theories. <i>Academy of Management Review</i> , <i>14</i> (4), 562–578. https://doi.org/10.5465/amr.1989.4308389	Integrate mode , Phase 2 、 Phase 3 、 Phase 4 、 Phase 5 。
13	Straub, D. W. (2009). Editor's comments: Why top journals accept your paper. <i>MIS Quarterly</i> , <i>33</i> (3), iii–x. https://www.jstor.org/stable/20650302	Note mode , Q Section 。
14	Sutton, R. I., & Staw, B. M. (1995). What theory is not. <i>Administrative Science Quarterly</i> , <i>40</i> (3), 371–384. https://doi.org/10.2307/2393788	Note mode , Q Section ; Integrate mode , Phase 3 、 Phase 4 。
15	Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. <i>Nature</i> , <i>393</i> (6684), 440–442. https://doi.org/10.1038/30918	Graph mode , clustering coefficient (群聚係數) 。
16	Weick, K. E. (1995). What theory is not, theorizing is. <i>Administrative Science Quarterly</i> , <i>40</i> (3), 385–390. https://doi.org/10.2307/2393789	Note mode , Q Section (Sutton & Staw 支柱內) 。
17	Whetten, D. A. (1989). What constitutes a theoretical contribution? <i>Academy of Management Review</i> , <i>14</i> (4), 490–495. https://doi.org/10.2307/258554	Note mode , Stage 4 、 Q Section 。
18	國家科學及技術委員會 (National Science and Technology Council). (2026, March). 研究誠信電子報 (第	Citation + LOCK 溯源實務、回到原文比

編號	引用 (APA 7)	備註
	64期) [Research integrity newsletter, No. 64]. https://www.nstc.gov.tw/ori/ch/list/ceded792-f5ee-4bbd-b4b1-3799ee818c9f	對參考文獻與 in-text citation、判斷 A 原創或 A 引用 B 之參考。

FAQ Last Updated: March 7, 2026.