**例2.5.4(APT恶意软件家族相似度分析)** APT(Advanced Persistent Threat)高级持续性威胁)是一种周期较长、隐蔽性极强的攻击模式，一个攻击组织通常会复用其恶意软件，导致其持续攻击软件代码具有一定的相似性。本例尝试用杰卡德指数来对其相似度进行分析。

为了分析其相似度，通常的步骤如下

Jaccard框图

图2.5.1 分析相似度框架图

图2.5.2是APT家族“STARSYPOUND”的某个恶意攻击软件样本的名称“F6655E39465C2FF5B016980D918EA028”，通过逆向攻击还原出其二进制代码片段如下

|  |
| --- |
| 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 |
| B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |
| 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| 00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00 |
| 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 |
| 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F |
| 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 |
| 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 |

图2.5.2 样本“F6655E39465C2FF5B016980D918EA028”的源码片段

在第二步从源码中提取特征有多种方法，本例采用提取可打印字符串的经典方法。下面对图2.5.1的源码进行分析如下

1、4D 5A: 代表 "MZ"，是 Microsoft Executable (MZ) 文件的标识符。这是历史遗留问题，因为早期的 DOS 可执行文件使用这个标识符。

2、90 00 03 00 00 00 04 00 00 00: 这是 PE 文件头的偏移量，表示 PE 文件头位于 MZ 头部之后的第 0x0000000390 个字节处。

2、FF FF 00 00: 这是 MZ 头部的保留字段，通常是空的，可以忽略。

3、B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00: 这是 MOV EAX, 0x40 这条汇编指令，它将十六进制数 0x40 写入 EAX 寄存器。在 PE 文件头部中，这条指令的作用是设置 PE 文件头部的地址，通常它会指向 PE 文件头的开始位置。

4、E8 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21: 这一部分包含了 CALL 指令，MOV 指令，MOV AH, 0x09 指令，以及 INT 21H 指令。 这些指令可能用于加载 PE 文件头的地址和执行一些简单的操作，例如打印错误信息。

5、54 68 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00: 这是一个 ASCII 字符串，它代表着 "This program cannot be run in DOS mode." 这句话。

其中"This program cannot be run in DOS mode."就是我们需要提取的可打印字符串。注意前面的CD 21 指令中的 21 对应着 ASCII 码表中的 !(感叹号)，这里21并不直接代表 !，而是代表着 INT 21H 指令的立即数，用来指定中断号。但是INT 21H 指令可能会调用一些 DOS 系统服务，这些服务可能会最终将 ! 符号输出到屏幕上。所以我们提取出来的可打印字符串变成了"！This program cannot be run in DOS mode."，我们将其放入特征袋中。将所有源码中的可打印字符串提取出来，放入集合中，我们就得到了这个样本的特征袋。

使用 Python 中的 os 模块从恶意软件样本中提取字符串的代码如下：

strings = os.popen("strings '{0}'".format(fullpath)).read()

代码解析：

os.popen：这是一个 Python 内建函数，用于执行系统命令并返回一个类似文件的对象，可以读取命令的输出。

"strings '{0}'".format(fullpath)：这是一个格式化字符串，其中{0}被fullpath的值替换。strings 是一个命令行工具，用于从样本文件中提取字符串。

fullpath：这是一个变量，包含要处理的样本文件的完整路径。

因此，整个语句执行 strings 命令，并传递 fullpath 所代表的文件路径作为参数。

read() 方法从 os.popen 返回的类似文件对象中读取所有可打印字符串，并将结果作为字符串返回。

以下分别是提取的三个恶意软件样本的特征集合构成的特征袋：

样本1“F6655E39465C2FF5B016980D918EA028”：

{\_\_p\_\_fmode，%X @，WaitForSingleObject，\*(SY)#，=H @，D$ RP，SSPS，\_except\_handler3，DisconnectNamedPipe，GetCurrentProcess，\_XcptFilter，USER32.dll，h !@，D$dSV，=@ @，sprintf，WriteFile，L$(h，=, @，ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-/，%p @，t2Ht，PeekNamedPipe，%d @，T$\h@0@，D$(Qj，.rsrc，MSVCRT.dll，PADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADD，Rich，T$4hT0@，D$DP，%` @，D$ h，D$\QPU，\_beginthread，%\ @，=( @，exit，T$(Q，D$8D，\_acmdln，cmd.exe，LoadStringA，CreateThread，XPVSS，free，\_\_getmainargs，ExitThread，D$\h，\_^][，TerminateProcess，\$Hf，L$(hT0@，@.data，\_exit，t$p3，TerminateThread，SetEvent，KERNEL32.dll，T$4hp0@，%h @，[t V，\_itoa，GetStartupInfoA，L$,h，T$ QRP，WaitForMultipleObjects，SUVWj:h，T$0f，send = %d，\*(SY)# ，atol，GetComputerNameA，DuplicateHandle，\_\_setusermatherr，QRSSSj，UVWj，!This program cannot be run in DOS mode.，CreateEventA，\_strnicmp，\_controlfp，WS2\_32.dll，%l @，T$,h，malloc，hSVW，D$$V，\$lf，.text，L$$VQR，%x @，|$XS，`.rdata，\_\_p\_\_commode，strchr，\_adjust\_fdiv，GetModuleHandleA，ReadFile，L$\hL0@，=8 @，CloseHandle，\*(SY)# cmd，\_initterm，5< @，SVWj，CreatePipe，>"u:F，CreateProcessA，Sleep，L$pj，\_\_set\_app\_type}

样本2“STARSYPOUND\_sample\_9EA3C16194CE354C244C1B74C46CD92E”：

{\_\_p\_\_fmode，%X @，WaitForSingleObject，\*(SY)#，=H @，D$ RP，SSPS，\_except\_handler3，DisconnectNamedPipe，GetCurrentProcess，\_XcptFilter，USER32.dll，h !@，D$dSV，=@ @，sprintf，WriteFile，L$(h，=, @，ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-/，%p @，t2Ht，PeekNamedPipe，%d @，T$\h@0@，D$(Qj，.rsrc，MSVCRT.dll，Rich，T$4hT0@，D$DP，%` @，D$ h，D$\QPU，\_beginthread，%\ @，=( @，exit，T$(Q，D$8D，\_acmdln，PAPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDING，cmd.exe，LoadStringA，CreateThread，XPVSS，free，\_\_getmainargs，ExitThread，D$\h，\_^][，TerminateProcess，\$Hf，L$(hT0@，@.data，\_exit，t$p3，TerminateThread，SetEvent，KERNEL32.dll，T$4hp0@，%h @，[t V，\_itoa，GetStartupInfoA，L$,h，T$ QRP，WaitForMultipleObjects，SUVWj:h，T$0f，send = %d，\*(SY)# ，atol，GetComputerNameA，DuplicateHandle，\_\_setusermatherr，QRSSSj，UVWj，!This program cannot be run in DOS mode.，CreateEventA，\_strnicmp，\_controlfp，WS2\_32.dll，%l @，T$,h，malloc，hSVW，D$$V，\$lf，.text，L$$VQR，%x @，|$XS，`.rdata，\_\_p\_\_commode，strchr，\_adjust\_fdiv，GetModuleHandleA，ReadFile，L$\hL0@，=8 @，CloseHandle，\*(SY)# cmd，\_initterm，5< @，SVWj，CreatePipe，>"u:F，CreateProcessA，Sleep，L$pj，\_\_set\_app\_type}

样本3 “WEBC2-AUSOV\_sample\_097B5ABB53A3D84FA9EABDA02FEF9E91”：

{\_\_p\_\_fmode，vX |iR ，h0A@，\_except\_handler3，%P0@，ExpandEnvironmentStringsA，\_XcptFilter，%X0@，GetTempPathA，h8A@，%temp%，h0@@，-$81# 6%2+:，%x0@，strstr，PhxA@，strncmp，VWh|@@，LZ32.dll，RegCreateKeyExA，GetModuleFileNameA，strchr，2??,，exit，=<0@，strrchr，\_acmdln，GetModuleHandleA，XPVSS，h$@@，\_\_getmainargs，hD@@，t \_^，RegSetValueExA，@.data，\_exit，%\0@，RegCloseKey，vX Rich，LZOpenFileA，hP$@，KERNEL32.dll，MSVCRT.dll，LoadLibraryA，%l0@，strncpy，h A@，GetStartupInfoA，urlmon.dll，InternetCloseHandle，DeleteFileA，\_\_setusermatherr，InternetReadFile，\_\_p\_\_commode，LZCopy，atoi，hh@@，GetProcAddress，hX@@，URLDownloadToFileA，InternetOpenA，!This program cannot be run in DOS mode.，CopyFileA，ADVAPI32.dll，\_controlfp，%L0@，vX |i\ ，hSVW，i]\@GZ.，.text，U^CD，hD$@，`.rdata，\_adjust\_fdiv，%H0@，wininet.dll，\svchost.exe，CloseHandle， -s ，\_initterm，InternetOpenUrlA，>"u:F，CreateProcessA，GetLongPathNameA，Sleep，LZClose，vY ;vX %PS ，\_\_set\_app\_type}

各样本提取的特征数如表1所示。

表1 各样本特征数

|  |  |
| --- | --- |
| 样本名 | 特征数量 |
| F6655E39465C2FF5B016980D918EA028 | 113 |
| STARSYPOUND\_sample\_  9EA3C16194CE354C244C1B74C46CD92E | 113 |
| WEBC2-AUSOV\_sample\_  097B5ABB53A3D84FA9EABDA02FEF9E91 | 89 |

根据Jaccard系数计算公式：

Jaccard系数=共享特征数/总特征数

我们分别计算这三个样本之间的Jaccard系数，如表2所示。

表2 各样本对的Jaccard系数

|  |  |  |  |
| --- | --- | --- | --- |
| 样本对 | 共享特征数 | 总特征数 | Jaccard系数 |
| 样本1和样本2 | 112 | 114 | 0.98 |
| 样本1和样本3 | 29 | 173 | 0.17 |
| 样本2和样本3 | 29 | 173 | 0.17 |

以下是计算三个样本之间Jaccard系数的韦恩图示例：

样本1和2

图2.5.3 三个样本之间的Jaccard系数计算韦恩图示例

各软件样本之间的Jaccard系数代表它们的代码共享程度，我们通过样本间Jaccard系数表格来直观地查看这种代码共享关系，如表3所示。

表3 样本间Jaccard系数

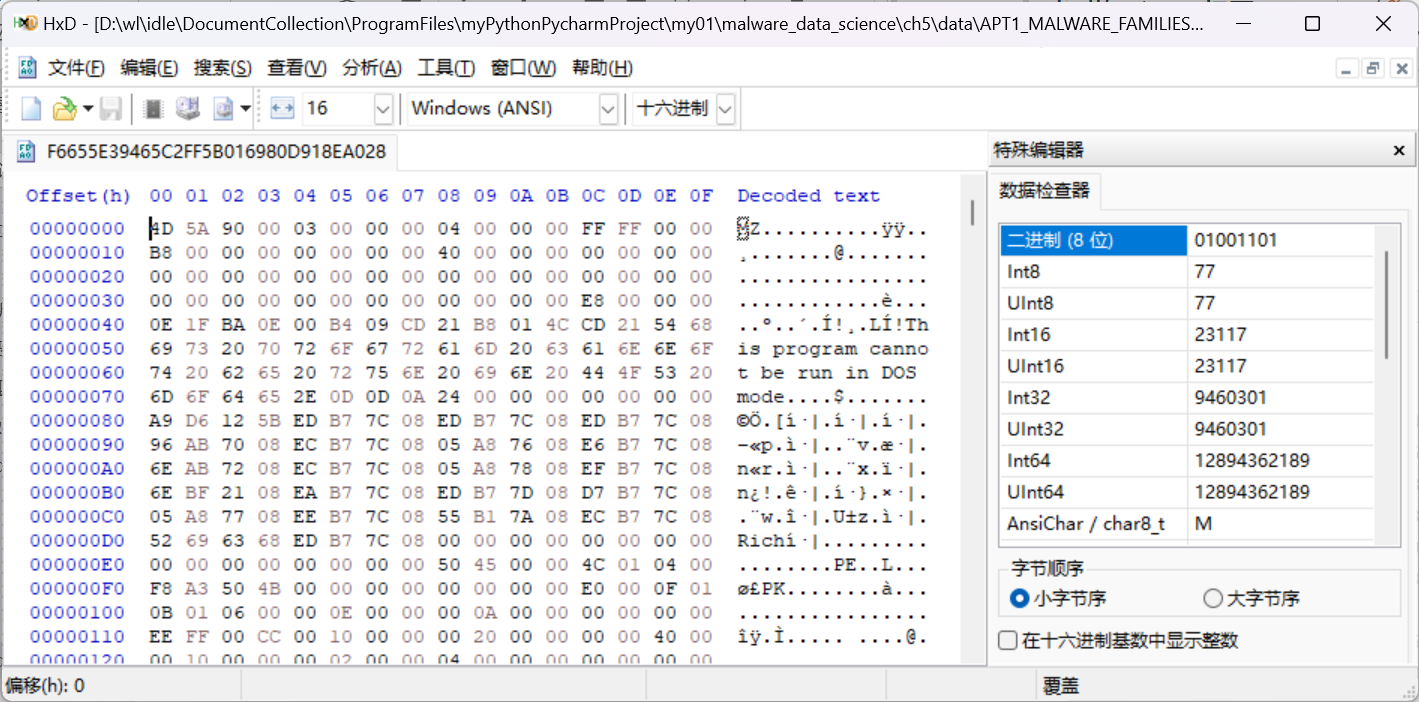
|  |  |  |  |
| --- | --- | --- | --- |
| Jaccard系数 | 样本1 | 样本2 | 样本3 |
| 样本1 | 1 | 0.98 | 0.17 |
| 样本2 | 0.98 | 1 | 0.17 |
| 样本3 | 0.17 | 0.17 | 1 |

由表\*可知，样本1和样本2大概率是属于同一个恶意软件家族。

利用Jaccard系数分析恶意软件家族相似性的实验步骤：

#### 一、逆向

利用常用的逆向工具，比如HxD打开恶意软件样本即可获取恶意软件样本的二进制源码，如图\*所示。



图\* HxD获取恶意软件样本源码

#### 二、特征提取

本例选取样本中的可打印字符串作为特征，使用 Python 中的 os 模块从恶意软件样本中提取字符串的代码如下：

|  |
| --- |
| strings = os.popen("strings '{0}'".format(fullpath)).read() |
| strings = set(strings.split("\n")) |

代码解析：

os.popen：这是一个 Python 内建函数，用于执行系统命令并返回一个类似文件的对象，可以读取命令的输出。

"strings '{0}'".format(fullpath)：这是一个格式化字符串，其中{0}被fullpath的值替换。strings 是一个命令行工具，用于从样本文件中提取字符串。

fullpath：这是一个变量，包含要处理的样本文件的完整路径。

read() ：从 os.popen 返回的类似文件对象中读取所有可打印字符串，并将结果作为字符串返回。

strings.split("\n")：将 strings 变量中的字符串按照换行符（\n）进行分割，并生成一个包含所有字符串的列表。

set(...)： 将分割后的字符串列表转换为一个集合（set）。

利用代码提取的三个恶意软件样本的特征集合构成的特征袋如下：

**样本1“F6655E39465C2FF5B016980D918EA028”：**

{\_\_p\_\_fmode，%X @，WaitForSingleObject，\*(SY)#，=H @，D$ RP，SSPS，\_except\_handler3，DisconnectNamedPipe，GetCurrentProcess，\_XcptFilter，USER32.dll，h !@，D$dSV，=@ @，sprintf，WriteFile，L$(h，=, @，ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-/，%p @，t2Ht，PeekNamedPipe，%d @，T$\h@0@，D$(Qj，.rsrc，MSVCRT.dll，PADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADD，Rich，T$4hT0@，D$DP，%` @，D$ h，D$\QPU，\_beginthread，%\ @，=( @，exit，T$(Q，D$8D，\_acmdln，cmd.exe，LoadStringA，CreateThread，XPVSS，free，\_\_getmainargs，ExitThread，D$\h，\_^][，TerminateProcess，\$Hf，L$(hT0@，@.data，\_exit，t$p3，TerminateThread，SetEvent，KERNEL32.dll，T$4hp0@，%h @，[t V，\_itoa，GetStartupInfoA，L$,h，T$ QRP，WaitForMultipleObjects，SUVWj:h，T$0f，send = %d，\*(SY)# ，atol，GetComputerNameA，DuplicateHandle，\_\_setusermatherr，QRSSSj，UVWj，!This program cannot be run in DOS mode.，CreateEventA，\_strnicmp，\_controlfp，WS2\_32.dll，%l @，T$,h，malloc，hSVW，D$$V，\$lf，.text，L$$VQR，%x @，|$XS，`.rdata，\_\_p\_\_commode，strchr，\_adjust\_fdiv，GetModuleHandleA，ReadFile，L$\hL0@，=8 @，CloseHandle，\*(SY)# cmd，\_initterm，5< @，SVWj，CreatePipe，>"u:F，CreateProcessA，Sleep，L$pj，\_\_set\_app\_type}

**样本2“STARSYPOUND\_sample\_9EA3C16194CE354C244C1B74C46CD92E”：**

{\_\_p\_\_fmode，%X @，WaitForSingleObject，\*(SY)#，=H @，D$ RP，SSPS，\_except\_handler3，DisconnectNamedPipe，GetCurrentProcess，\_XcptFilter，USER32.dll，h !@，D$dSV，=@ @，sprintf，WriteFile，L$(h，=, @，ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-/，%p @，t2Ht，PeekNamedPipe，%d @，T$\h@0@，D$(Qj，.rsrc，MSVCRT.dll，Rich，T$4hT0@，D$DP，%` @，D$ h，D$\QPU，\_beginthread，%\ @，=( @，exit，T$(Q，D$8D，\_acmdln，PAPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDING，cmd.exe，LoadStringA，CreateThread，XPVSS，free，\_\_getmainargs，ExitThread，D$\h，\_^][，TerminateProcess，\$Hf，L$(hT0@，@.data，\_exit，t$p3，TerminateThread，SetEvent，KERNEL32.dll，T$4hp0@，%h @，[t V，\_itoa，GetStartupInfoA，L$,h，T$ QRP，WaitForMultipleObjects，SUVWj:h，T$0f，send = %d，\*(SY)# ，atol，GetComputerNameA，DuplicateHandle，\_\_setusermatherr，QRSSSj，UVWj，!This program cannot be run in DOS mode.，CreateEventA，\_strnicmp，\_controlfp，WS2\_32.dll，%l @，T$,h，malloc，hSVW，D$$V，\$lf，.text，L$$VQR，%x @，|$XS，`.rdata，\_\_p\_\_commode，strchr，\_adjust\_fdiv，GetModuleHandleA，ReadFile，L$\hL0@，=8 @，CloseHandle，\*(SY)# cmd，\_initterm，5< @，SVWj，CreatePipe，>"u:F，CreateProcessA，Sleep，L$pj，\_\_set\_app\_type}

**样本3“WEBC2-AUSOV\_sample\_097B5ABB53A3D84FA9EABDA02FEF9E91”：**

{\_\_p\_\_fmode，vX |iR ，h0A@，\_except\_handler3，%P0@，ExpandEnvironmentStringsA，\_XcptFilter，%X0@，GetTempPathA，h8A@，%temp%，h0@@，-$81# 6%2+:，%x0@，strstr，PhxA@，strncmp，VWh|@@，LZ32.dll，RegCreateKeyExA，GetModuleFileNameA，strchr，2??,，exit，=<0@，strrchr，\_acmdln，GetModuleHandleA，XPVSS，h$@@，\_\_getmainargs，hD@@，t \_^，RegSetValueExA，@.data，\_exit，%\0@，RegCloseKey，vX Rich，LZOpenFileA，hP$@，KERNEL32.dll，MSVCRT.dll，LoadLibraryA，%l0@，strncpy，h A@，GetStartupInfoA，urlmon.dll，InternetCloseHandle，DeleteFileA，\_\_setusermatherr，InternetReadFile，\_\_p\_\_commode，LZCopy，atoi，hh@@，GetProcAddress，hX@@，URLDownloadToFileA，InternetOpenA，!This program cannot be run in DOS mode.，CopyFileA，ADVAPI32.dll，\_controlfp，%L0@，vX |i\ ，hSVW，i]\@GZ.，.text，U^CD，hD$@，`.rdata，\_adjust\_fdiv，%H0@，wininet.dll，\svchost.exe，CloseHandle， -s ，\_initterm，InternetOpenUrlA，>"u:F，CreateProcessA，GetLongPathNameA，Sleep，LZClose，vY ;vX %PS ，\_\_set\_app\_type}

#### 三、Jaccard系数计算

计算Jaccard系数的代码如下：

|  |
| --- |
| def jaccard(set1,set2):  intersection = set1.intersection(set2)  intersection\_length = float(len(intersection))  union = set1.union(set2)  union\_length = float(len(union))  return intersection\_length / union\_length |

各样本提取的特征数如表1所示。

表1 各样本特征数

|  |  |
| --- | --- |
| 样本名 | 特征数量 |
| F6655E39465C2FF5B016980D918EA028 | 113 |
| STARSYPOUND\_sample\_  9EA3C16194CE354C244C1B74C46CD92E | 113 |
| WEBC2-AUSOV\_sample\_  097B5ABB53A3D84FA9EABDA02FEF9E91 | 89 |

根据Jaccard系数计算公式：

Jaccard系数=共享特征数/总特征数

我们分别计算这三个样本之间的Jaccard系数，如表2所示。

表2 各样本对的Jaccard系数

|  |  |  |  |
| --- | --- | --- | --- |
| 样本对 | 共享特征数 | 总特征数 | Jaccard系数 |
| 样本1和样本2 | 112 | 114 | 0.98 |
| 样本1和样本3 | 29 | 173 | 0.17 |
| 样本2和样本3 | 29 | 173 | 0.17 |

以下是计算三个样本之间Jaccard系数的韦恩图示例：

样本1和2

图4 三个样本之间的Jaccard系数计算韦恩图示例

各软件样本之间的Jaccard系数代表它们的代码共享程度，我们通过样本间Jaccard系数表格来直观地查看这种代码共享关系，如表3所示。

表3 样本间Jaccard系数

|  |  |  |  |
| --- | --- | --- | --- |
| Jaccard系数 | 样本1 | 样本2 | 样本3 |
| 样本1 | 1 | 0.98 | 0.17 |
| 样本2 | 0.98 | 1 | 0.17 |
| 样本3 | 0.17 | 0.17 | 1 |

在图11中，色块颜色的深浅反映了样本间的代码共享程度，颜色越深代表共享程度越高。我们通常把共享程度高于某一阈值的样本归类为同一家族。

为了制作jaccard系数表格，修改如下代码以创建jaccard系数矩阵，如图7所示：

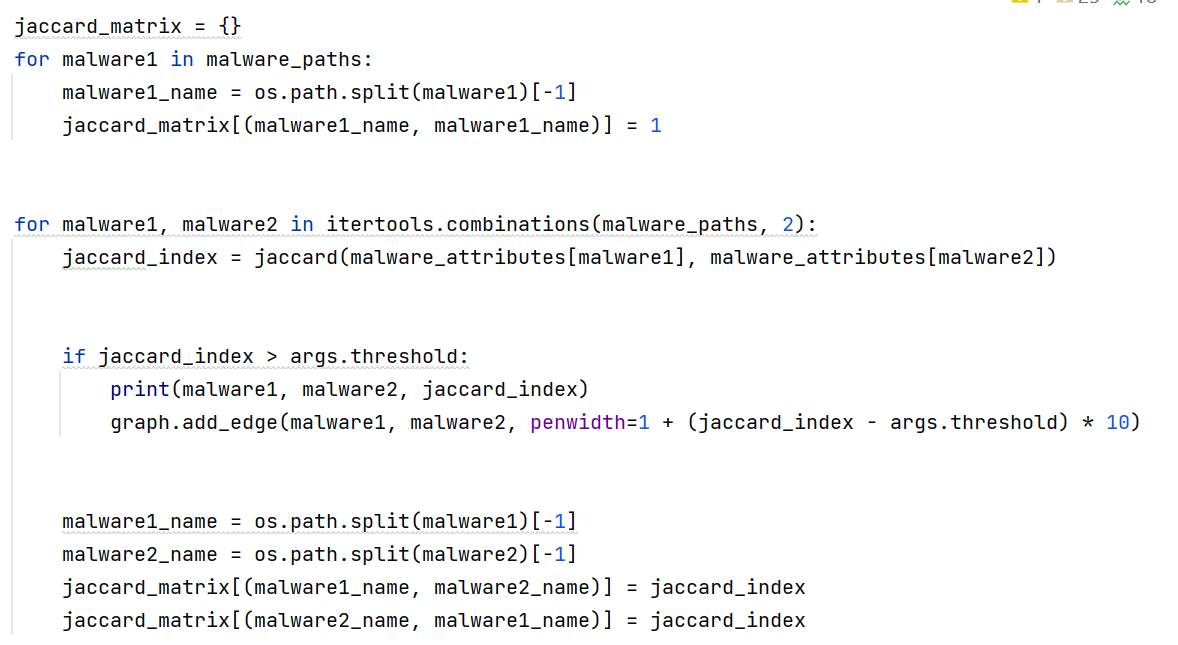


图7 修改的jaccard系数矩阵代码

修改后的代码新增了以下功能：

1、创建字典存储 Jaccard 系数: 在主程序中，创建了一个名为 jaccard\_matrix 的字典，用于存储 Jaccard 系数。

2、存储 Jaccard 系数: 在计算 Jaccard 系数后，将 (malware1\_name, malware2\_name) 作为键，Jaccard 系数作为值存储到 jaccard\_matrix 中。

然后我们添加以下代码制作jaccard系数表格，表格的横纵坐标代表恶意样本，表格内容则代表样本之间的jaccard系数。代码如图8所示。

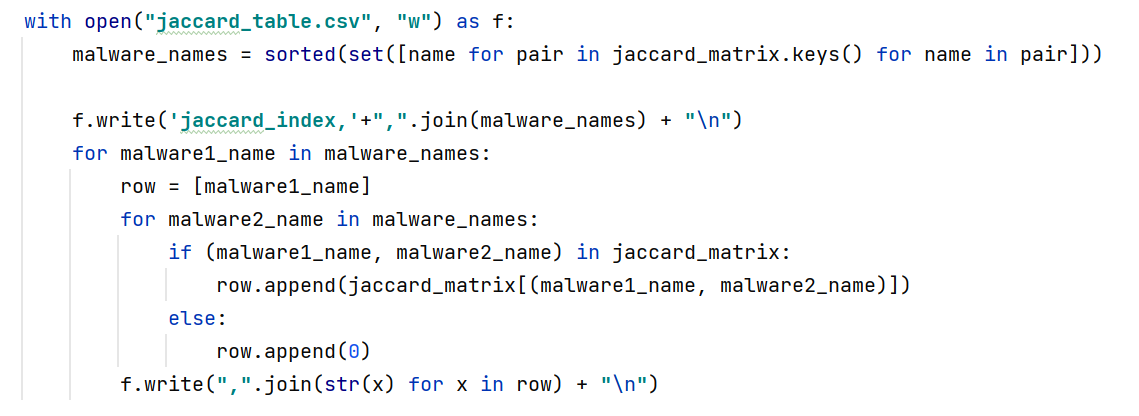


图8 制作jaccard系数表格的代码

代码解释：

1、打开文件: 使用 open("jaccard\_table.csv", "w") 打开文件 jaccard\_table.csv，并以写入模式 "w" 打开。

2、写入表头: 使用 f.write('jaccard,'+",".join(malware\_names) + "\n") 写入表头，表头包含所有恶意软件名称。

3、写入数据: 遍历所有恶意软件名称，对于每个恶意软件，将它的名称和与其他恶意软件的 Jaccard 系数写入文件。

4、关闭文件: 使用 f.close() 关闭文件。

此时在终端运行代码后，我们会得到一个名为jaccard\_table.csv的csv表格，表格内容如图9所示。

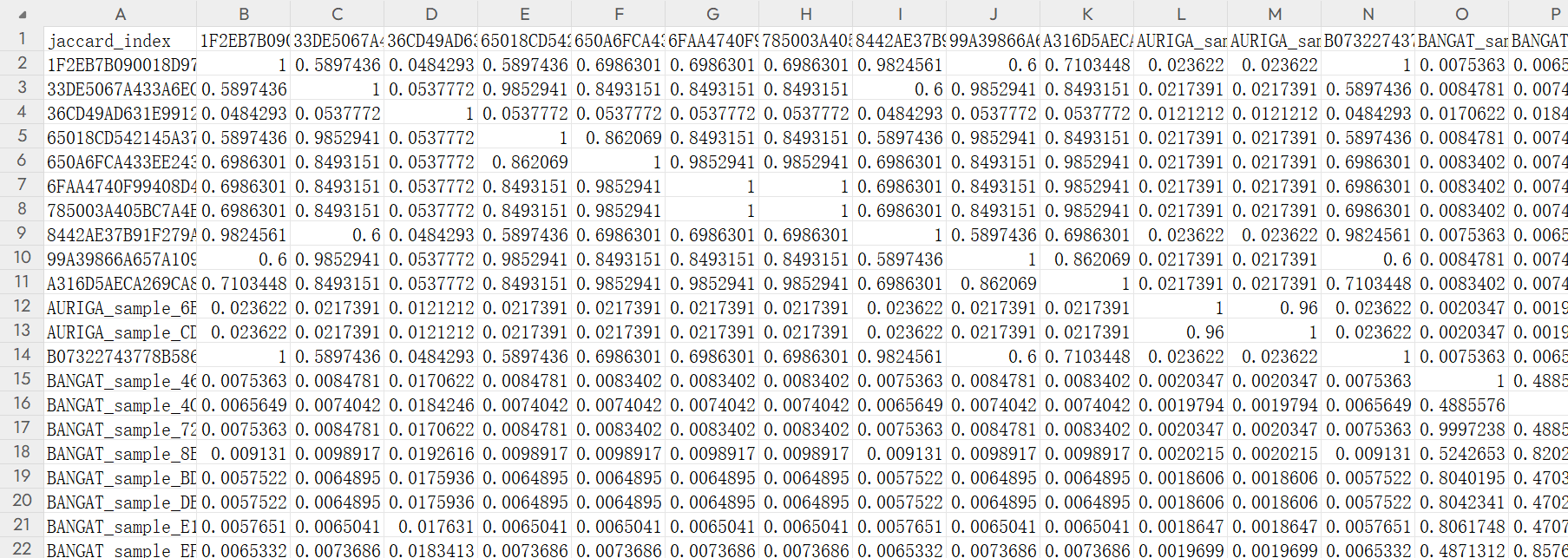


图9 jaccard\_table.csv

#### 四、可视化

首先绘制出恶意软件样本之间的相似图。代码在上一节已经提供。在终端运行代码：python code/listing\_5\_1.py data output.dot -j 0.8，如图5所示。

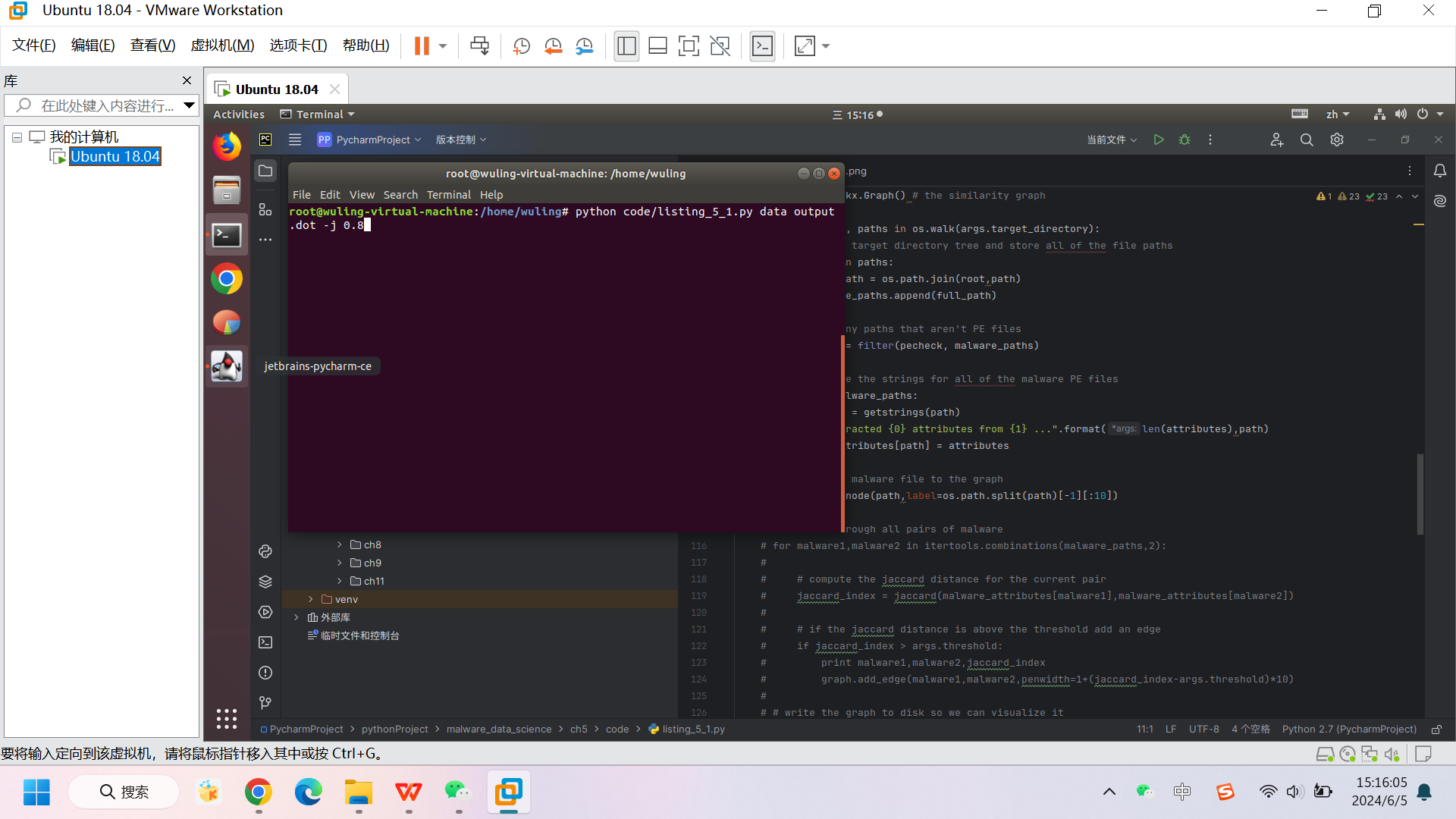


图5 终端运行代码

如果此时报错，提示ImportError: No module named pydot，是因为你的 Python 环境中缺少 pydot 库。pydot 是一个用于与 Graphviz 图形库交互的 Python 库，用于将 NetworkX 图转换为 DOT 文件，以便用 Graphviz 可视化。

我们首先在终端输入sudo apt-get install graphviz，这会安装 Graphviz 图形库，它是 pydot 的依赖。然后使用pip install pydot命令安装 pydot 库。

运行代码之后我们会得到一个dot文件：output.dot。想要查看此dot文件，我们可以使用 dot 命令将 DOT 文件转换为图像文件，例如 PNG 或 SVG，然后使用图像查看器打开。例如，将 output.dot 文件转换为 PNG 图像文件，可以在终端中输入以下命令：dot output.dot -Tpng -o output.png。这将在当前目录生成一个名为 output.png 的 PNG 图像文件，你可以使用图像查看器打开它，如图6所示。

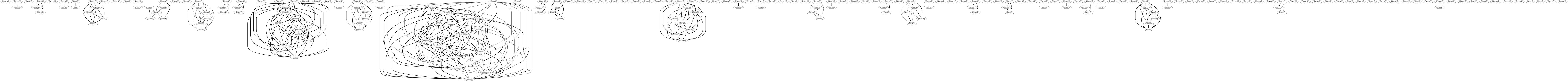


图6 output.png

图6展示了恶意软件代码共享可视化效果。

我们再添加以下代码以绘制恶意样本的相似性灰度热图。添加的代码如图10所示。

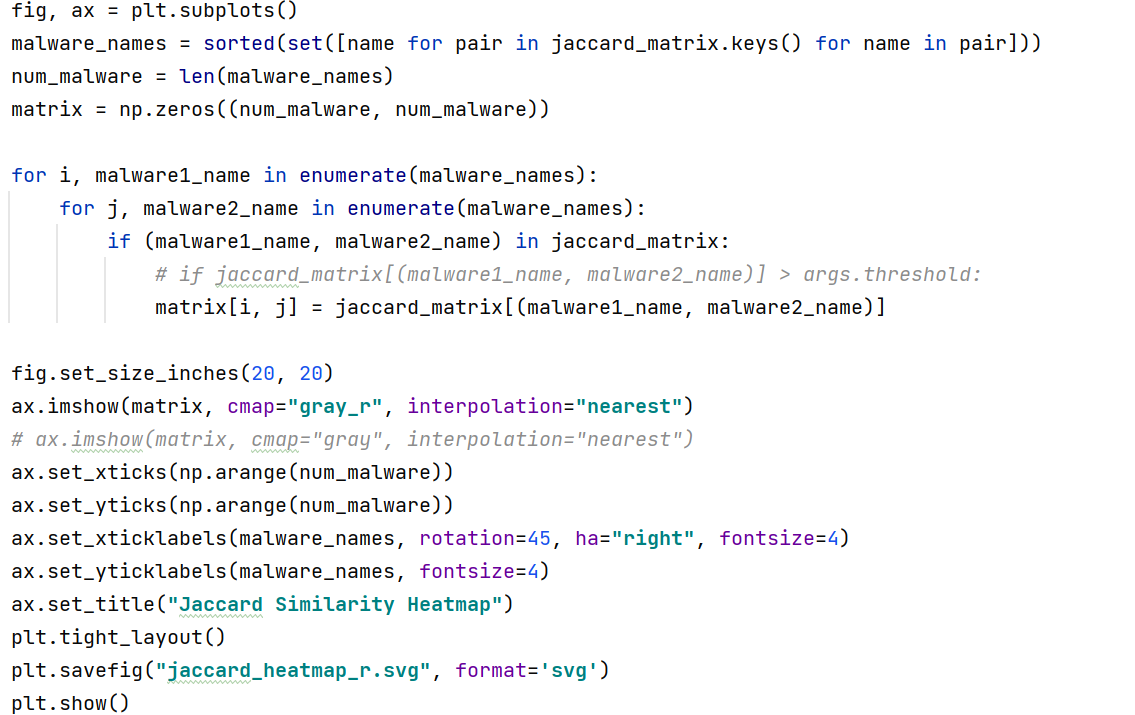


图10 绘制相似性灰度热图的代码

这段代码会绘制一个灰度热图，其中 Jaccard 系数越大的位置将显示为越接近黑色，而 Jaccard 系数越小的位置将显示为越接近白色，从而直观地展示出恶意软件样本之间的相似度关系。

运行代码后如果报错，提示ImportError: No module named \_tkinter, please install the python-tk package，这个错误信息 ImportError: No module named \_tkinter, please install the python-tk package 表明你的 Python 环境中缺少 tkinter 库。tkinter 是 Python 的标准 GUI 库，它依赖于 \_tkinter 模块，该模块是 Python 与 Tcl/Tk GUI 工具包的接口。

解决方法：在 Ubuntu 中，使用 apt-get 命令安装 python-tk 包：sudo apt-get install python-tk。

运行之后得到jaccard\_heatmap\_r.svg，打开之后如图11所示。

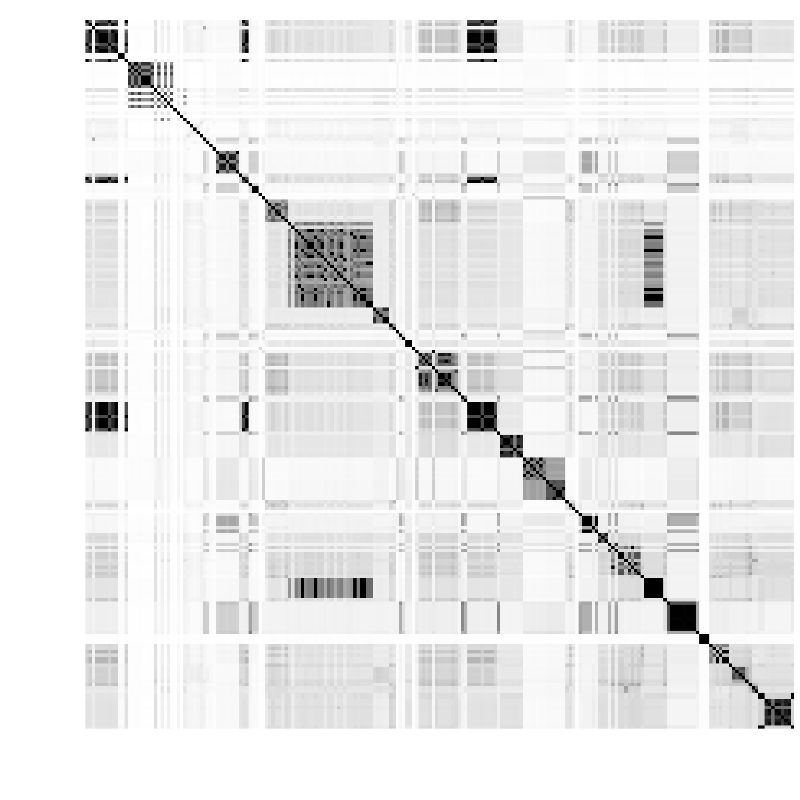


图11 jaccard\_heatmap\_r.svg