

TAS-PF：一个利用概率场扩展的 TAS 图解软件。

TAS-PF: TAS Diagram extended with Probabilistic Field.

版本号: 1.0.0

1 简介

1. 概述

本软件采用概率场（PF）扩展了经典的 TAS 分类图解方法。TAS，全称为 Total Alkali Silica，是一种通过绘制碱与二氧化硅总量关系图来对未定名火成岩进行岩石类型分类的技术。

1.1 开发背景与创新

传统 TAS 方法基于火成岩中总碱与二氧化硅含量的关联性，在二维图解上直观划分岩石类别，其操作简便易行。然而，随着数据集的增长和多样性的增加，传统的边界区分方式可能变得模糊或不稳定。

在此基础上开发的 TAS-PF 版本引入了概率场概念，其中 PF 部分即代表这一改进。该概率场是依据 GEOROC 全球火成岩数据库构建而成，利用大数据分析技术生成的。

在 TAS-PF 系统中，每个待分类样本会根据从 GEOROC 数据库获取的概率分布逐一确定所属岩石类别。这种方法不仅保留了经典图解法中的分类边界，确保了新系统的前向兼容性，而且通过引入概率统计手段增强了对新增数据的适应能力，实现了后向兼容性。随着数据量不断积累，概率场模型将更加稳健可靠，从而提供更精确、灵活的火成岩分类解决方案。

1.2 开发环境

本软件的开发环境如下：

库名	版本号	功能描述
PySide6	6.6.1	Qt 绑定库，用于 Python 的 GUI 开发
beeware	0.3.0	用于原生跨平台移动和桌面应用开发工具
matplotlib	3.8.0	数据可视化库，绘制高质量的图表
numpy	1.26.3	科学计算库，提供大量数学函数和数组操作功能
pandas	2.1.4	数据分析库，提供 DataFrame 等数据结构及数据分析工具

库名	版本号	功能描述
scipy	1.11.4	科学计算库，包含数值积分、优化、插值等高级数学算法
toga	0.4.0	Python UI 工具包，构建跨平台本地 GUI 应用程序
toga-chart	0.2.0	Toga 库的一个扩展，用于创建图表和数据可视化组件
scikit-learn	1.2.0	机器学习库，包含多种监督和无监督学习算法

注意：不要盲目追求最新版本，比如 `scikit-learn`，如果更新成新版本，就可能无法运行了。

1.3 运行环境

本软件运行所需的最低硬件配置如下：

配置项	最低要求
处理器	2 GHz 或更快的处理器
运行内存（RAM）	4 GB 以上
空闲硬盘空间	1 GB 以上
显示分辨率	1920 x 1080 分辨率

本软件结合 PySide6 与 BeeWare，是一种编写跨平台 GUI 应用：利用 PySide6 构建界面，借力 BeeWare 的 Briefcase 实现一键打包和运行。此方案确保应用在 Windows、GNU/Linux 等几种主流操作系统上原生运行。

本软件运行所需的推荐操作系统如下：

操作系统	版本号	架构	最低版本要求	推荐版本要求
Windows	10	x64	10	11
Linux	Ubuntu 22.04	x64	Ubuntu 20.04	Ubuntu 22.04

2 安装

对于 Windows 用户，安装特别简单。

2.1 Windows 系统下载后运行

下载链接：<https://pan.baidu.com/s/1QprhPGUFd99K-4MgA9pFCA?pwd=iaiz>

提取码：iaiz

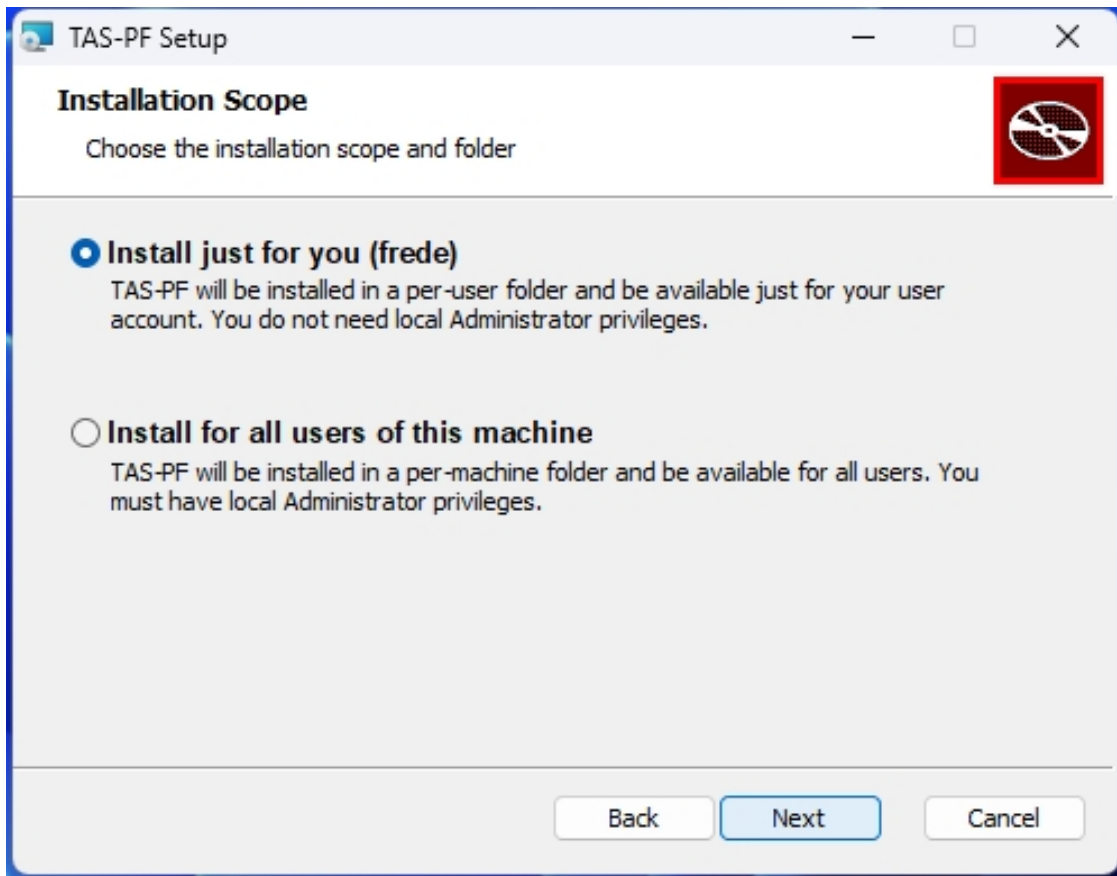
请注意，当前提供的安装包仅适用于最新的主流 Windows 操作系统版本，如 Windows 10 和 Windows 11。对于更早的操作系统版本，由于开发者不具备相应环境且无法进行有效测试，故软件及其依赖包可能无法在这些旧版系统上正常运行。

2.1.1 Windows 系统下使用 MSI 安装包安装

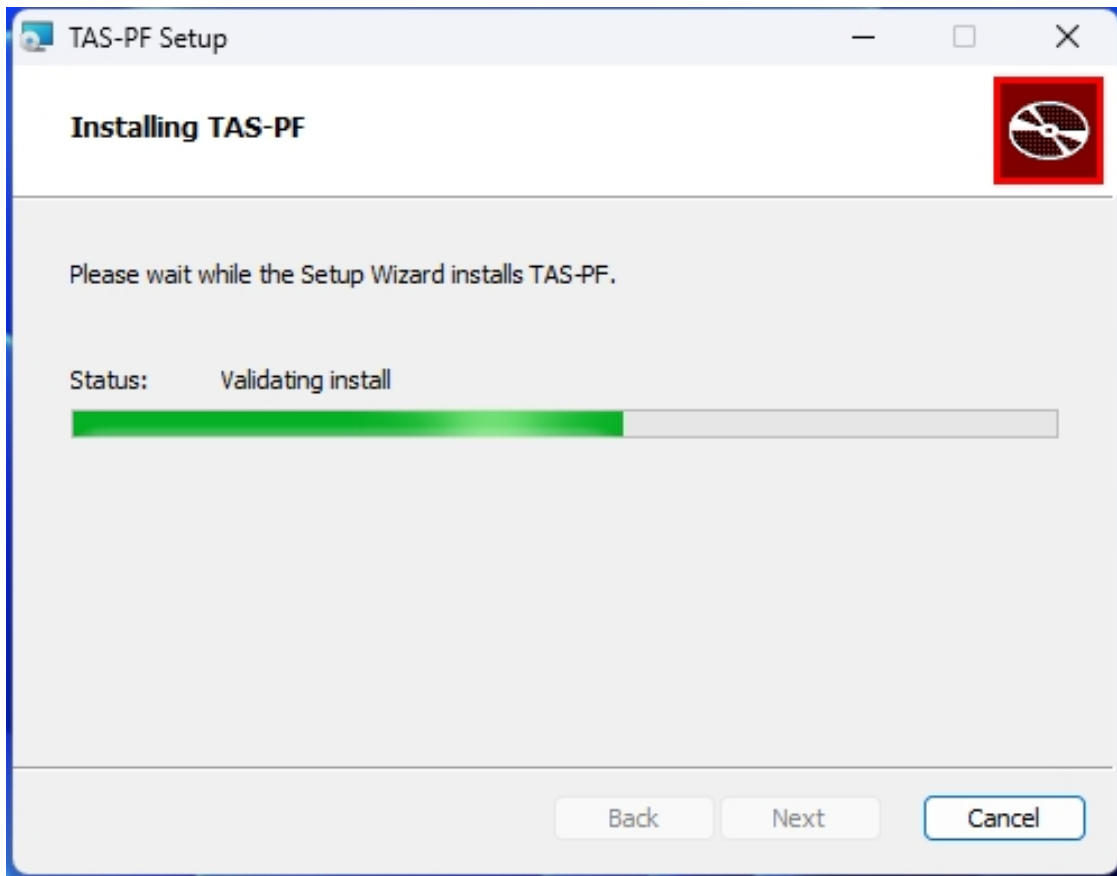
通过上述链接，您可以获取名为“TAS-PF-1.0.0.msi”的安装文件。只需简单下载并双击该文件，即可启动安装程序进行软件安装。



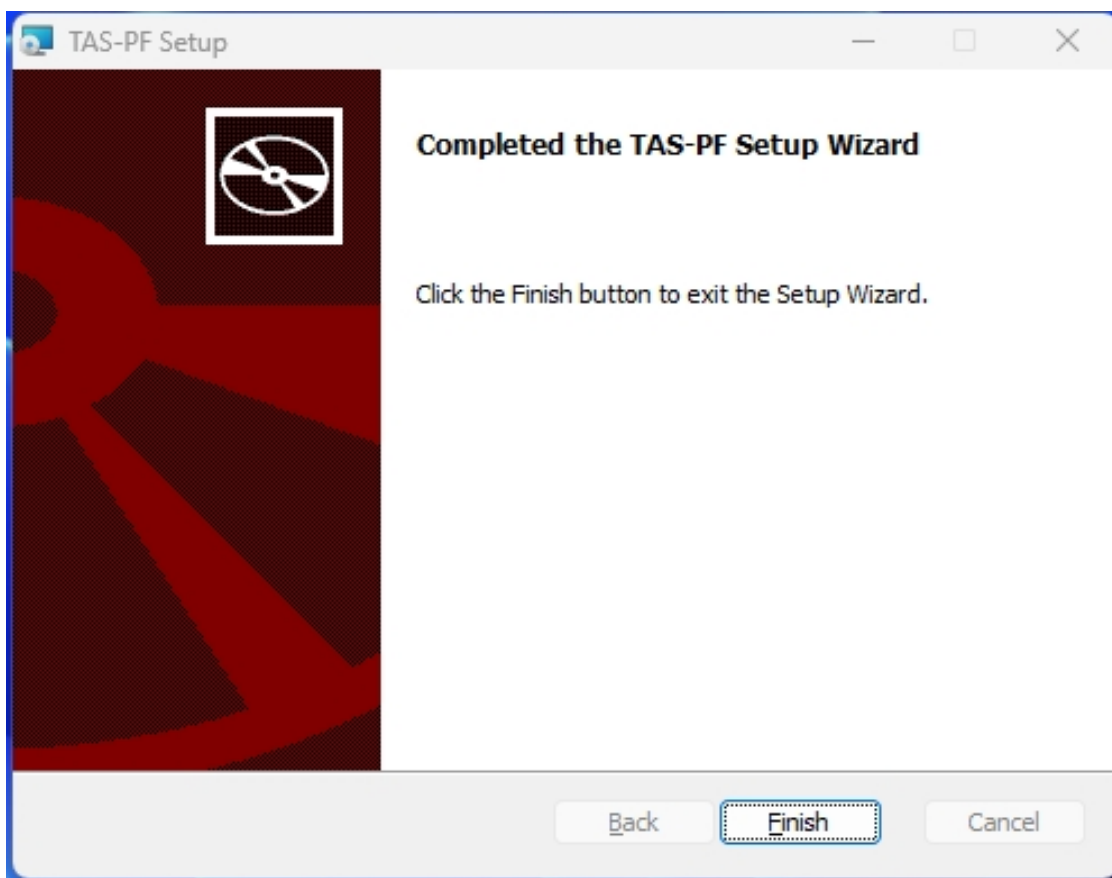
安装过程中，您将面临两种模式选择：一种是仅针对当前用户进行安装，另一种则是面向整个系统环境进行全局安装。



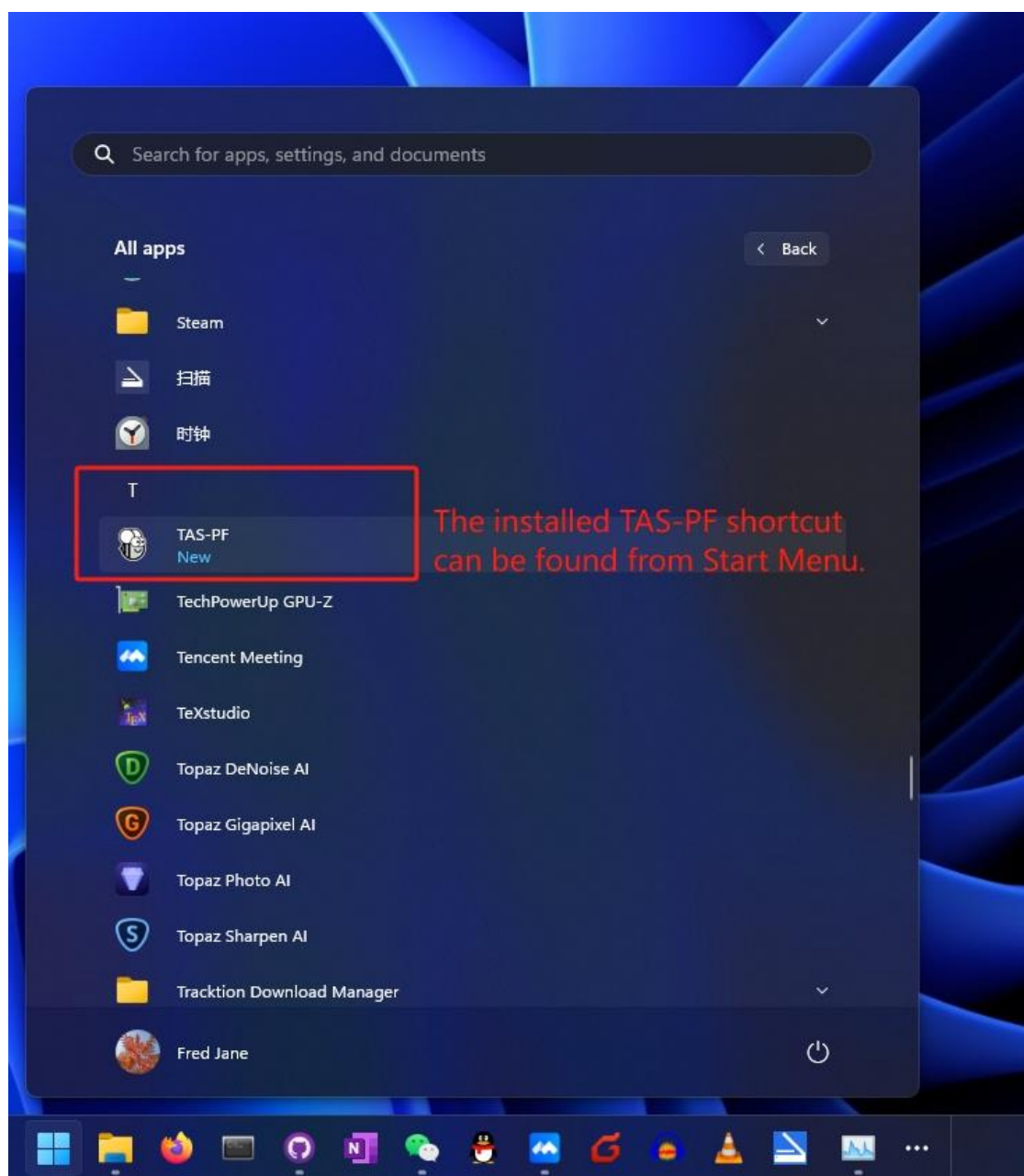
安装过程可能需要几分钟，需要一点点耐心等待。



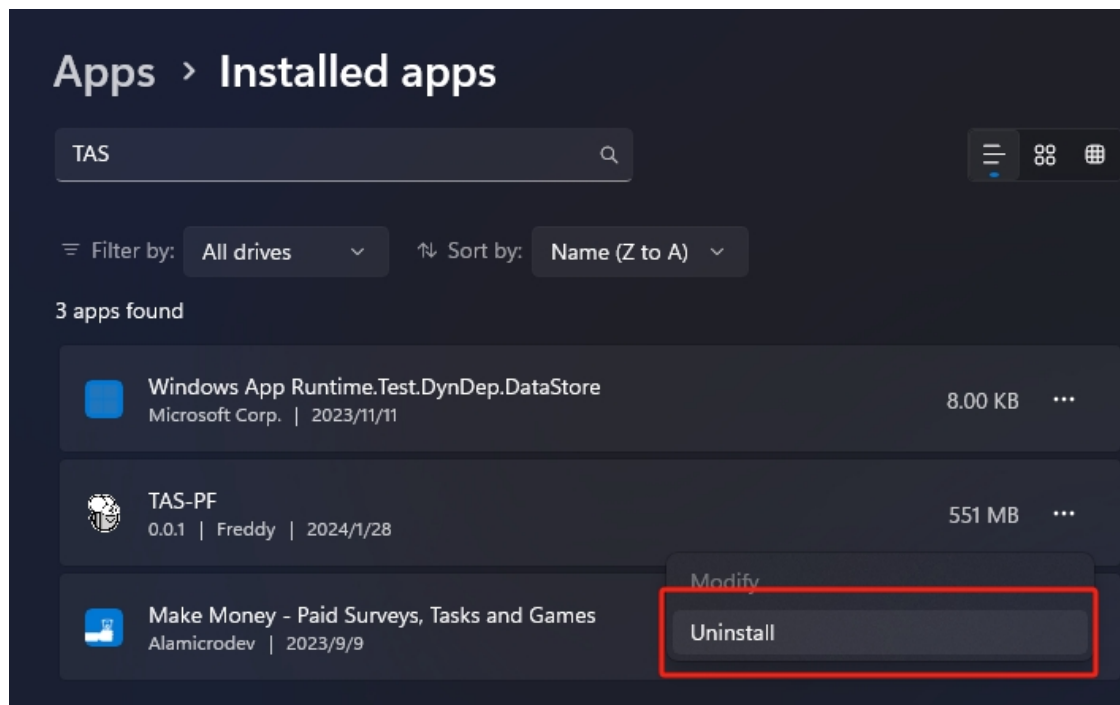
如果看到安装成功的提示，就是可以用了。



安装完成之后在开始菜单里面就能找到安装好的应用了。

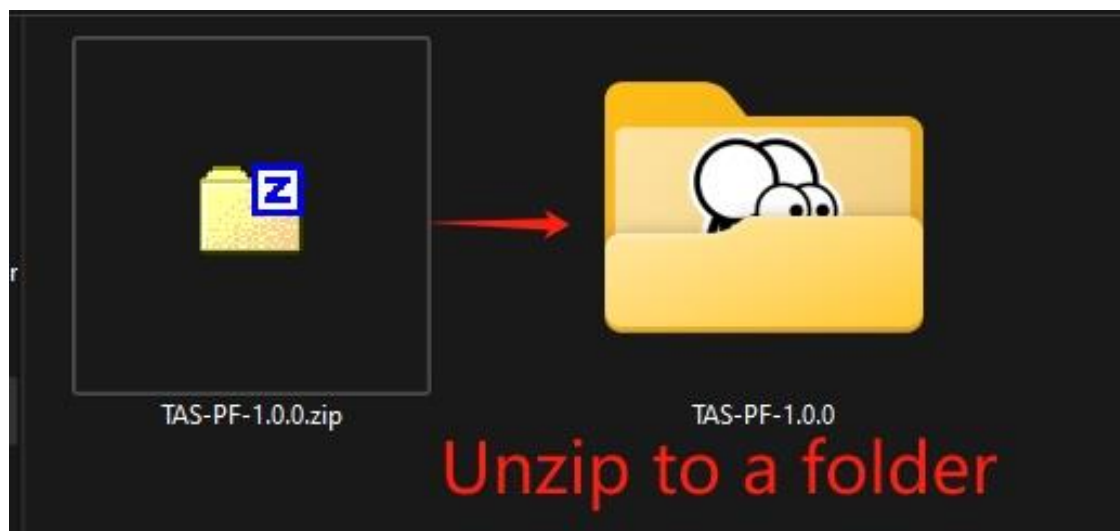


如果你需要移除掉已经安装的版本，去应用管理里面卸载即可。

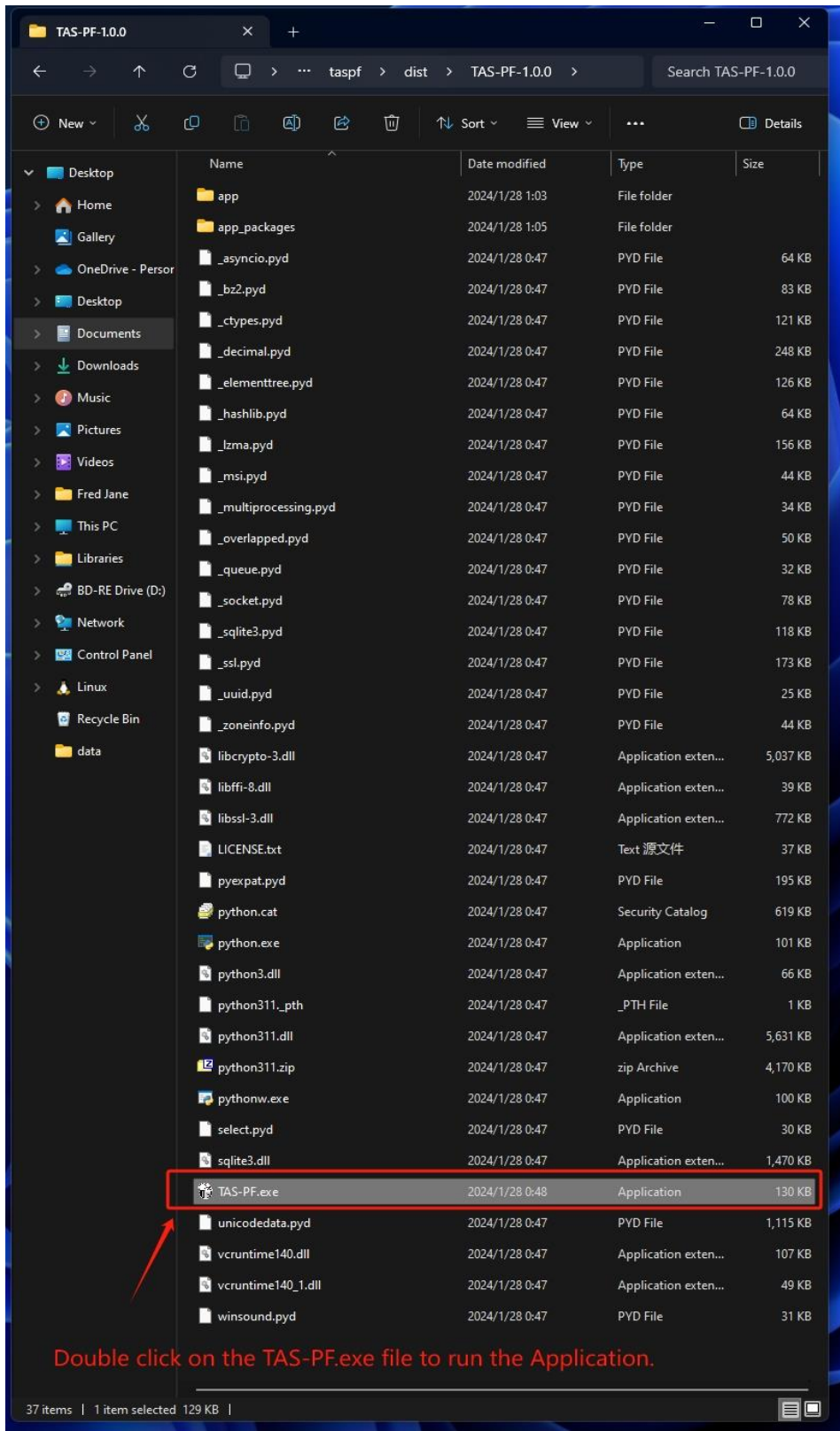


2.1.2 绿色版zip 文件解压缩运行

在上述链接中，您还会发现一个名为“TAS-PF-1.0.0.zip”的文件。下载并解压该文件后，将会生成一个独立的文件夹。为了确保内置解释器能够正常运行，请务必将其存放于一个全英文路径下，并避免路径中包含任何空格字符。



保证路径没有问题之后，到解压缩的文件夹内寻找一个叫做“TAS-PF.exe”的文件，双击这个“TAS-PF.exe”文件就可以运行程序了。



2.2 克隆代码然后运行

2.2.1 克隆代码

这个运行方法主要适合 GNU/Linux 和 macOS 用户，当然 Windows 用户也可以这么运行，就是有点麻烦了。

要运行本项目的应用软件，需要使用 `git` 来克隆，因此要实现安装，而且还需要能够有流畅的网络，访问 `github` 来获取代码。克隆好代码之后，还需要一个 Python 3.10 或者以上的版本来运行克隆好的文件。

```
(base) fred@Windows:/mnt/c/Users/frede$ python3 --version
Python 3.10.12
(base) fred@Windows:/mnt/c/Users/frede$ |
```

除了上述提及的文件之外，还可能需安装一些其他的系统层级依赖包。

在此，我们以开发者所采用的具体操作系统环境——Ubuntu 22.04.3 LTS 为例进行说明。

为了确保项目的完整运行，首先，请按照以下命令进行依赖包的安装：

```
sudo apt update
sudo apt install build-essential git pkg-config python3-dev python3-venv libgirepository1.0-dev libcairo2-dev gir1.2-webkit2-4.0 libcanberra-gtk3-module libxkbcommon0 qtwayland5 libegl1-mesa
```

如果不进行上面的步骤，后面在 `pip install pycairo` 就可能会出错，就无法完成依赖环境的完整安装。

然后克隆代码和 `pip` 安装依赖包的命令如下所示：

```
git clone https://github.com/GeoPyTool/TAS-PF.git
cd TAS-PF
pip install -r requirement.txt
```

如果完成了上述的依赖包安装，就能看到类似下面的信息，有时候可能有一些 warning，但这都不会有太大问题，只要不是 error 就基本都没事。

```
(base) fred@windows: /mnt/c/Users/frede/Desktop/TAS-PF/taspf$ briefcase run

[taspf] Finalizing application configuration...
Targeting ubuntu-jammy (Vendor base debian)
Determining glibc version... done
Targeting glibc 2.35
Targeting Python3.10

[taspf] Generating application template...
Using app template: https://github.com/beeware/briefcase-linux-system-template.git, branch v0.3.16

[taspf] Installing support package...
No support package required.

[taspf] Installing application code...
Installing src/taspf... done

[taspf] Installing requirements...
Collecting PySide6-Essentials==6.5
  Obtaining dependency information for PySide6-Essentials==6.5 from https://files.pythonhosted.org/packages/56/65/59/d8cd768b717f799a5ab724b6d48eb533a862ef9dcd67d19c7277280b0/PySide6-Essentials-6.6.1-cp38-ab13-manylinux\_2\_28\_x86\_64.whl.metadata
  Using cached PySide6-Essentials-6.6.1-cp38-ab13-manylinux_2_28_x86_64.whl.metadata (3.7 kB)
Collecting matplotlib
  Obtaining dependency information for matplotlib from https://files.pythonhosted.org/packages/19/e5/adea514515f27b274035c69239ab7a3d15ed3109ee2b8e63017461946/matplotlib-3.8.2-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached matplotlib-3.8.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.8 kB)
Collecting numpy
  Obtaining dependency information for numpy from https://files.pythonhosted.org/packages/a5/77/d1453c9ff9f7638e68ec936c6fb56ba8d7c769daa8a4883cb8ef8ee45993/numpy-1.26.3-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached numpy-1.26.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (61 kB)
Collecting pandas
  Obtaining dependency information for pandas from https://files.pythonhosted.org/packages/b3/b3/3102c3a4abca1093e50cfac721192a1c65c0b318a4031395d0121e6e699/pandas-2.2.0-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached pandas-2.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (19 kB)
Collecting scipy
  Obtaining dependency information for scipy from https://files.pythonhosted.org/packages/f5/aa/8a6071a5eddcaec6bb5b22e0991ee79c59c5d772112b9c236ec1aff57d/scipy-1.12.0-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached scipy-1.12.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (60 kB)
Collecting toga
  Obtaining dependency information for toga from https://files.pythonhosted.org/packages/ee/9f/7bca32d0545843748b43d2d8fddde981f98d66e2ab3a98331e8a88f1ab9/toga-0.4.1-py3-none-any.whl.metadata
  Using cached toga-0.4.1-py3-none-any.whl.metadata (4.0 kB)
Collecting toga-chart
  Obtaining dependency information for toga-chart from https://files.pythonhosted.org/packages/c4/73/7c15f978692aee4a7109e3e7709b5a6c0b6e0d8b18e63ba1ce9/toga-chart-0.2.0-py3-none-any.whl.metadata
  Using cached toga-chart-0.2.0-py3-none-any.whl.metadata (4.3 kB)
Collecting scikit-learn==1.2.0
  Using cached scikit-learn-1.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (9.5 MB)
Collecting joblib>=1.1.1 (from scikit-learn==1.2.0)
  Obtaining dependency information for joblib>=1.1.1 from https://files.pythonhosted.org/packages/10/40/d855119c85db202f1f38ba8b7c96aca2f32040a8a4f928ca8a08b6d02/joblib-1.3.2-py3-none-any.whl.metadata
  Using cached joblib-1.3.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=2.0.0 (from scikit-learn==1.2.0)
  Obtaining dependency information for threadpoolctl>=2.0.0 from https://files.pythonhosted.org/packages/81/12/f0adea011af9d69c1ca486c75f3f7262cdcbac9b712ee58ca779a72865/threadpoolctl-3.2.0-py3-none-any.whl.metadata
  Using cached threadpoolctl-3.2.0-py3-none-any.whl.metadata (10.0 kB)
Collecting shiboken6==6.6.1 (from PySide6-Essentials==6.5)
  Obtaining dependency information for shiboken6==6.6.1 from https://files.pythonhosted.org/packages/bb/72/e50f758e99e8da8dc9999086c41a110b856889c4ca85d60c0ba97bd/shiboken6-6.6.1-cp38-ab13-manylinux\_2\_28\_x86\_64.whl.metadata
  Using cached shiboken6-6.6.1-cp38-ab13-manylinux_2_28_x86_64.whl.metadata (2.5 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Obtaining dependency information for contourpy>=1.0.1 from https://files.pythonhosted.org/packages/58/56/e2c3d3dfa1f9c70d4d5e3d6f5134b20ed953f4e2133a4b12f8062108db58/contourpy-1.2.0-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached contourpy-1.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.8 kB)
Collecting cycler>=0.10 (from matplotlib)
  Obtaining dependency information for cycler>=0.10 from https://files.pythonhosted.org/packages/e7/85/c19819d5e309529a4cf27099629a6f031d090d0e31110e3304c5a6/cycler-0.12.1-py3-none-any.whl.metadata
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Obtaining dependency information for fonttools>=4.22.0 from https://files.pythonhosted.org/packages/68/68/af4b7cf435c7018e17b5a43b108be4b0f0e8feb485a88dec20b186f2a7/fonttools-4.47.2-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached fonttools-4.47.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (157 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Obtaining dependency information for kiwisolver>=1.3.1 from https://files.pythonhosted.org/packages/6f/09/4ab1fdb57fcd80ce5983f04ae1aed7c1d5930da4f0c8aa526c12fe28a/kiwisolver-1.4.5-cp310-cp310-manylinux\_2\_12\_x86\_64.manylinux2018\_x86\_64.whl.metadata
  Using cached kiwisolver-1.4.5-cp310-cp310-manylinux_2_12_x86_64.manylinux2018_x86_64.whl.metadata (6.4 kB)
Collecting packaging>=20.0 (from matplotlib)
  Obtaining dependency information for packaging>=20.0 from https://files.pythonhosted.org/packages/ec/1a/610693ac4ee14cf2d098f3c493378e4f2ef7ae2e1021707a27f92f307d/packaging-23.2-py3-none-any.whl.metadata
  Downloading packaging-23.2-py3-none-any.whl.metadata (3.2 kB)
Collecting pillow>=8 (from matplotlib)
  Obtaining dependency information for pillow>=8 from https://files.pythonhosted.org/packages/cb/c3/98faa1e92cf866b9446c4042f1fe887e672b2f50e000cb984157b8895797/pillow-10.2.0-cp310-cp310-manylinux\_2\_28\_x86\_64.whl.metadata
  Downloading pillow-10.2.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (9.7 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Obtaining dependency information for pyparsing>=2.3.1 from https://files.pythonhosted.org/packages/39/92/8486ede85fcc08e1b3db3ace92a0726126f090b0808a213167948a2075/pyparsing-3.1.3-py3-none-any.whl.metadata
  Using cached pyparsing-3.1.3-py3-none-any.whl.metadata (5.1 kB)
```

2.2.2 运行项目

还是保持在克隆的代码的路径，然后找到子文件夹‘taspf’，利用 cd 命令进入，然后运行‘briefcase run’就可以运行代码了。

```
cd taspf
briefcase run
```

第一次运行可能需要联网更新一些程序模板等等，所以可能还需要一点时间，耐心是美德，没办法，只能等待，而且还要有一个比较稳定的网络环境，以便于访问 GitHub。

```
(base) fred@Windows:/mnt/c/Users/frede/Desktop/TAS-PF/taspf$ briefcase run

[taspf] Finalizing application configuration...
Targeting ubuntu:jammy (Vendor base debian)
Determining glibc version... done
Targeting glibc 2.35
Targeting Python3.10

[taspf] Generating application template...
Using app template: https://github.com/beeware/briefcase-linux-system-template.git, branch v0.3.16

[taspf] Installing support package...
No support package required.

[taspf] Installing application code...
Installing src/taspf... done

[taspf] Installing requirements...
Collecting PySide6-Essentials~=6.5
  Obtaining dependency information for PySide6-Essentials~=6.5 from https://files.pythonhosted.org/packages/56/65/50/d0ced768b717f709a5ab224b648eb533a862efd0cde67d19c2727200b0/PySide6\_Essentials-6.6.1-cp38-abi3-manylinux\_2\_28\_x86\_64.whl.metadata
  Using cached PySide6_Essentials-6.6.1-cp38-abi3-manylinux_2_28_x86_64.whl.metadata (3.7 kB)
Collecting matplotlib
  Obtaining dependency information for matplotlib from https://files.pythonhosted.org/packages/19/e5/a4ea514515f270224435c69359abb7a3d152ed31b9ee3ba5e63017461945/matplotlib-3.8.2-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached matplotlib-3.8.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.8 kB)
Collecting numpy
  Obtaining dependency information for numpy from https://files.pythonhosted.org/packages/a5/37/d1453c9ff4f7630e68ec036c6fb56ba0d7c769daa8a4083cb4ef8ee45995/numpy-1.26.3-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached numpy-1.26.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (61 kB)
Collecting pandas
  Obtaining dependency information for pandas from https://files.pythonhosted.org/packages/b3/b3/3102c3a4abca1093e50cfec2213102a1c65c0b318a4431395d0121e6e690/pandas-2.2.0-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached pandas-2.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (19 kB)
Collecting scipy
  Obtaining dependency information for scipy from https://files.pythonhosted.org/packages/f5/aa/8e6071a5e4dca4ec68b5b22e4991ee74c59c5d372112b9c236ec1faff57d/scipy-1.12.0-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata
  Using cached scipy-1.12.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (60 kB)
Collecting toga
```

这些很耗费时间的步骤基本上只是出现在第一次运行的时候，后续的运行过程中就不会有太多类似的问题了，也会节省很多时间。

3 操作与使用

TAS-PF 的设计非常简单，所以用起来应该很容易上手

3.1 数据文件格式

TAS-PF 支持 CSV 和 Excel 文件，具体的设置可以参考下面的截图：

	A	B	C	D	E	F	G	H	I	J
1	Label	SiO2(wt%)	K2O(wt%)	Na2O(wt%)	Marker	Size	Width	Style	Alpha	
2	[3]	66.88	4.08	3.53	o	50	1	-	0.9	
3	[3]	67.52	4.19	3.47	o	50	1	-	0.9	
4	[3]	67.42	4.12	3.57	o	50	1	-	0.9	
5	[3]	66.57	4	3.74	o	50	1	-	0.9	
6	[3]	66.75	4.22	3.55	o	50	1	-	0.9	
7	[3]	66.87	4.04	3.63	o	50	1	-	0.9	
8	[3]	66.07	4.18	3.76	o	50	1	-	0.9	
9	[3]	66.79	3.92	3.61	o	50	1	-	0.9	
10	[4]	70.22	2.86	4.13	o	50	1	-	0.9	
11	[4]	67.99	4.56	3.07	o	50	1	-	0.9	
12	[4]	68.24	2.69	4.33	o	50	1	-	0.9	
13	[4]	71.42	4	3.67	o	50	1	-	0.9	
14	[4]	73.37	4.36	3.53	o	50	1	-	0.9	
15	[4]	73.64	4.7	3.29	o	50	1	-	0.9	
16	[4]	70.25	3.06	3.95	o	50	1	-	0.9	
17	[5]	69.48	3.27	3.91	o	50	1	-	0.9	
18	[5]	69.8	3.35	3.83	o	50	1	-	0.9	

必须有的列就是 Label, SiO2(wt%), K2O(wt%), Na2O(wt%)，Label 就是你的样品标签，SiO2(wt%) 和 K2O(wt%)+Na2O(wt%) 是你投图的数据。

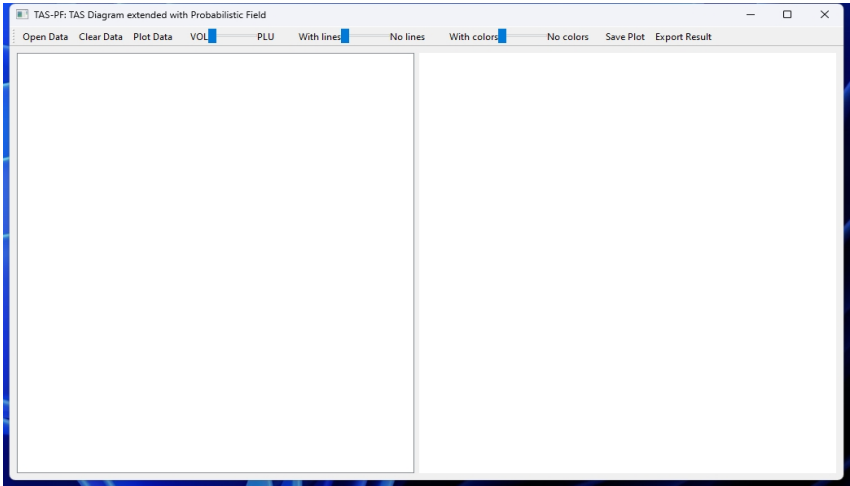
注意，这里你应该使用将所有氧化物相加然后再校准过的数据，才比较可靠。

为了简化运行流程和降低设计复杂度，TAS-PF 本身不再提供数据归一化的功能，这个大家自己在 Excel 或者 WPS 上就能完成，也不麻烦。

这次的 TAS-PF 不需要任何额外设置，程序可以根据数据的 label 分组自动赋给不同的颜色，参考数据例如 [data_samples/TAS.csv](#)。

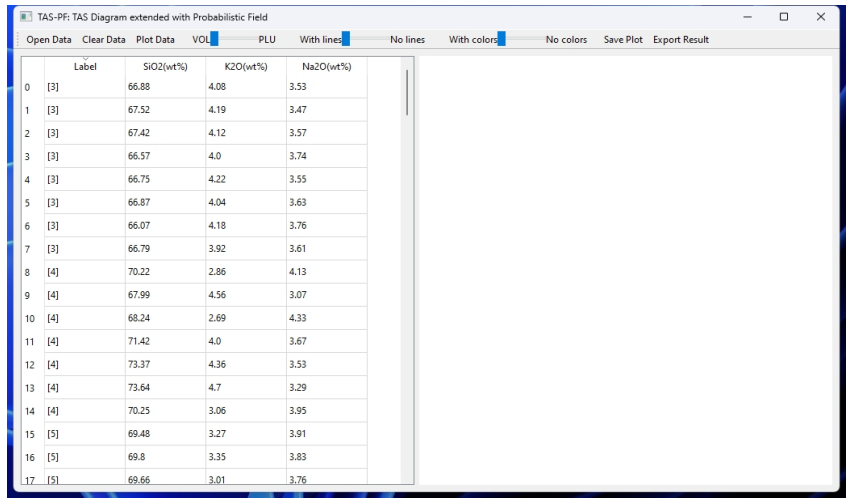
3.2 运行程序

无论是如何安装和运行，整体的图形界面都基本一样：



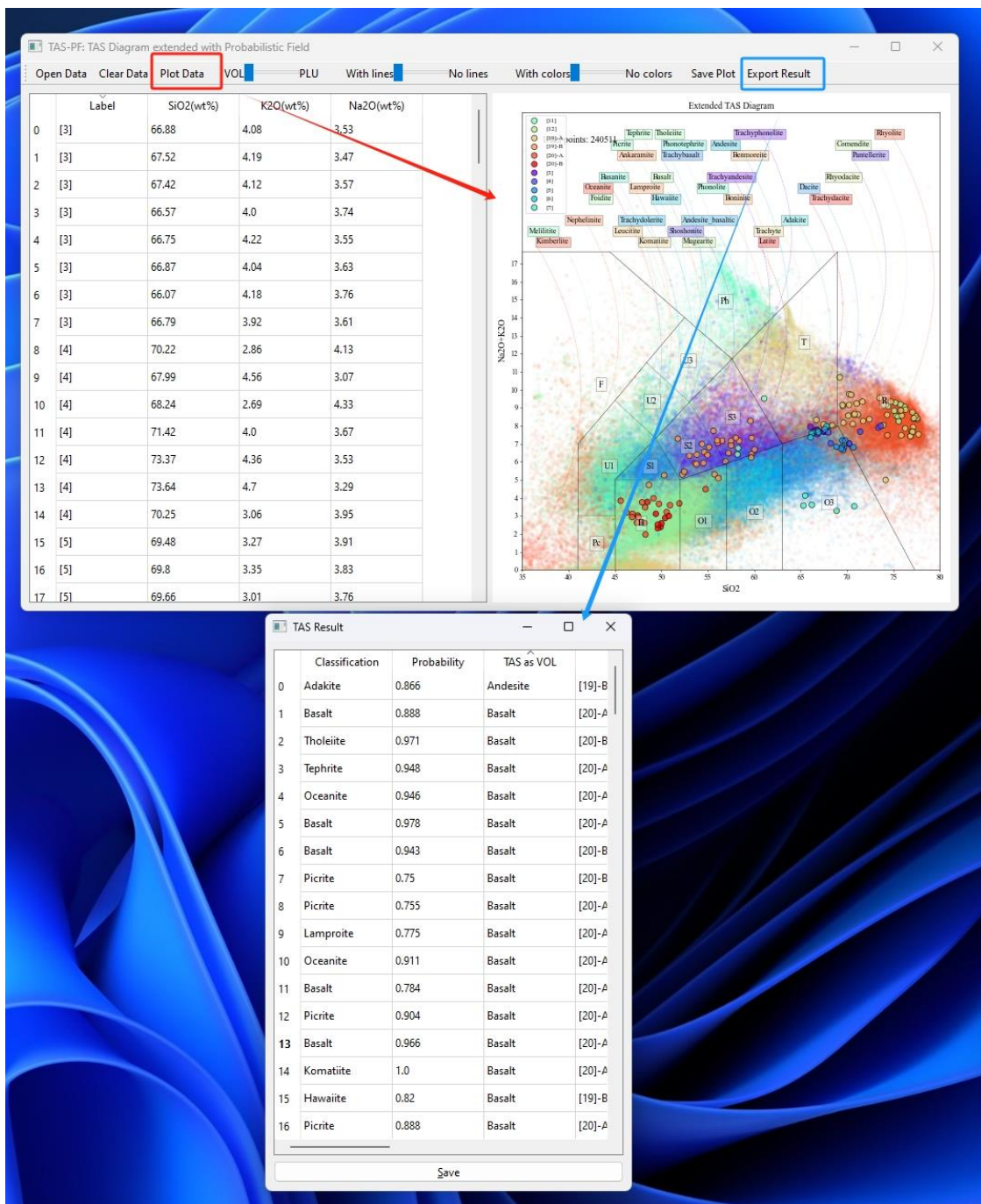
3.2.1 打开数据文件

就选择一个 CSV 或者 Excel 文件就可以加载到程序界面了。



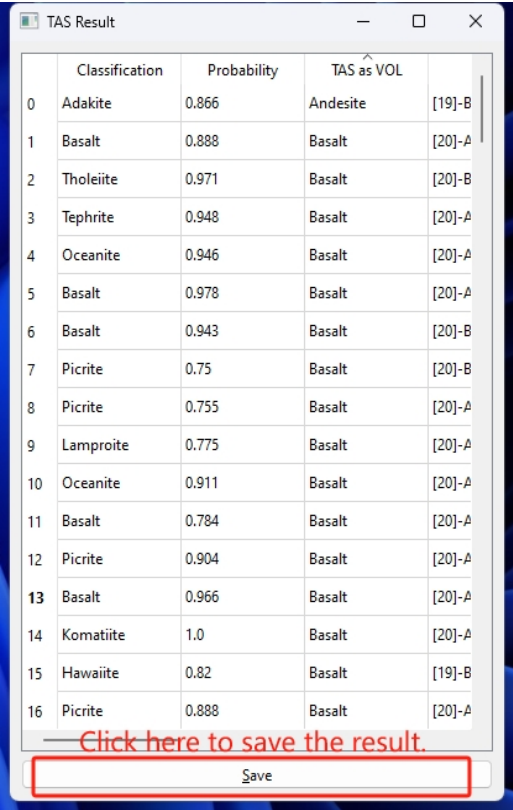
3.2.2 投图和判断

主界面上有 Plot Data 和 Export Result 两个按钮，按前者就会生成投图，按后者就会弹出分类结果。



这个操作流程非常直观，基本不需要什么学习成本，大家根据直觉来探索着很快就能上手。

3.2.3 保存结果



	Classification	Probability	TAS as VOL	
0	Adakite	0.866	Andesite	[19]-B
1	Basalt	0.888	Basalt	[20]-A
2	Tholeiite	0.971	Basalt	[20]-B
3	Tephrite	0.948	Basalt	[20]-A
4	Oceanite	0.946	Basalt	[20]-A
5	Basalt	0.978	Basalt	[20]-A
6	Basalt	0.943	Basalt	[20]-B
7	Picrite	0.75	Basalt	[20]-B
8	Picrite	0.755	Basalt	[20]-A
9	Lamproite	0.775	Basalt	[20]-A
10	Oceanite	0.911	Basalt	[20]-A
11	Basalt	0.784	Basalt	[20]-A
12	Picrite	0.904	Basalt	[20]-A
13	Basalt	0.966	Basalt	[20]-A
14	Komatiite	1.0	Basalt	[20]-A
15	Hawaiite	0.82	Basalt	[19]-B
16	Picrite	0.888	Basalt	[20]-A

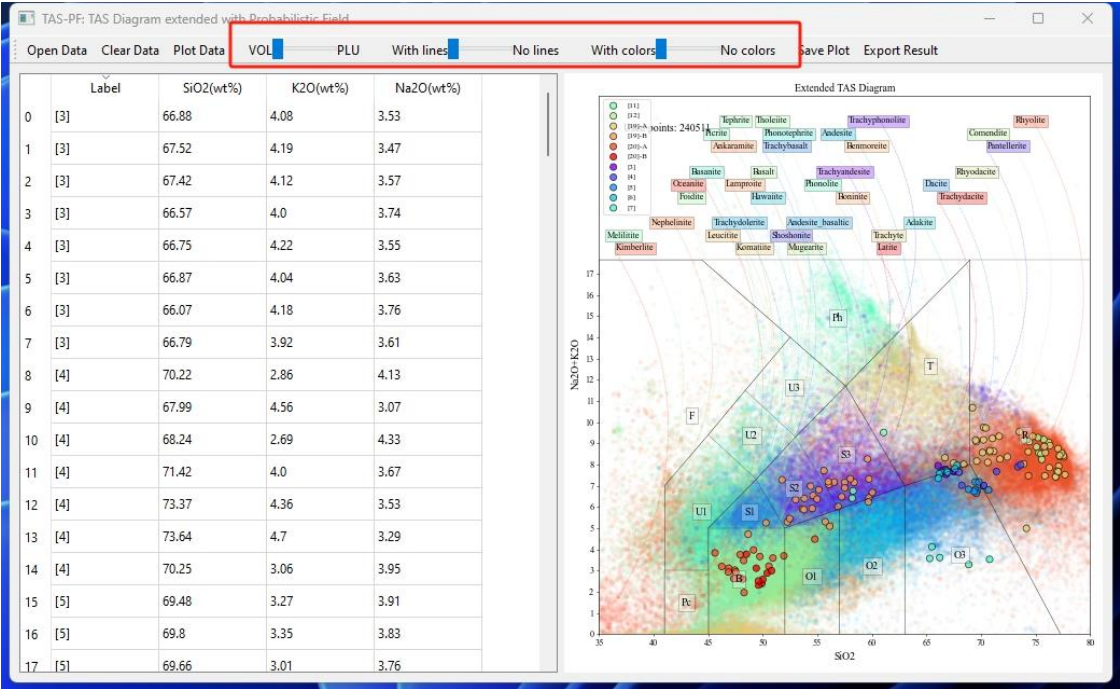
Click here to save the result.

Save

弹出的判断结果表格可以保存成一个文件，可以参考 [/data_samples/TAS_Result.csv](#)，实际就是在源数据文件的左侧加上概率场判断类别、判断概率、TAS 经典图解判断类别这三列。

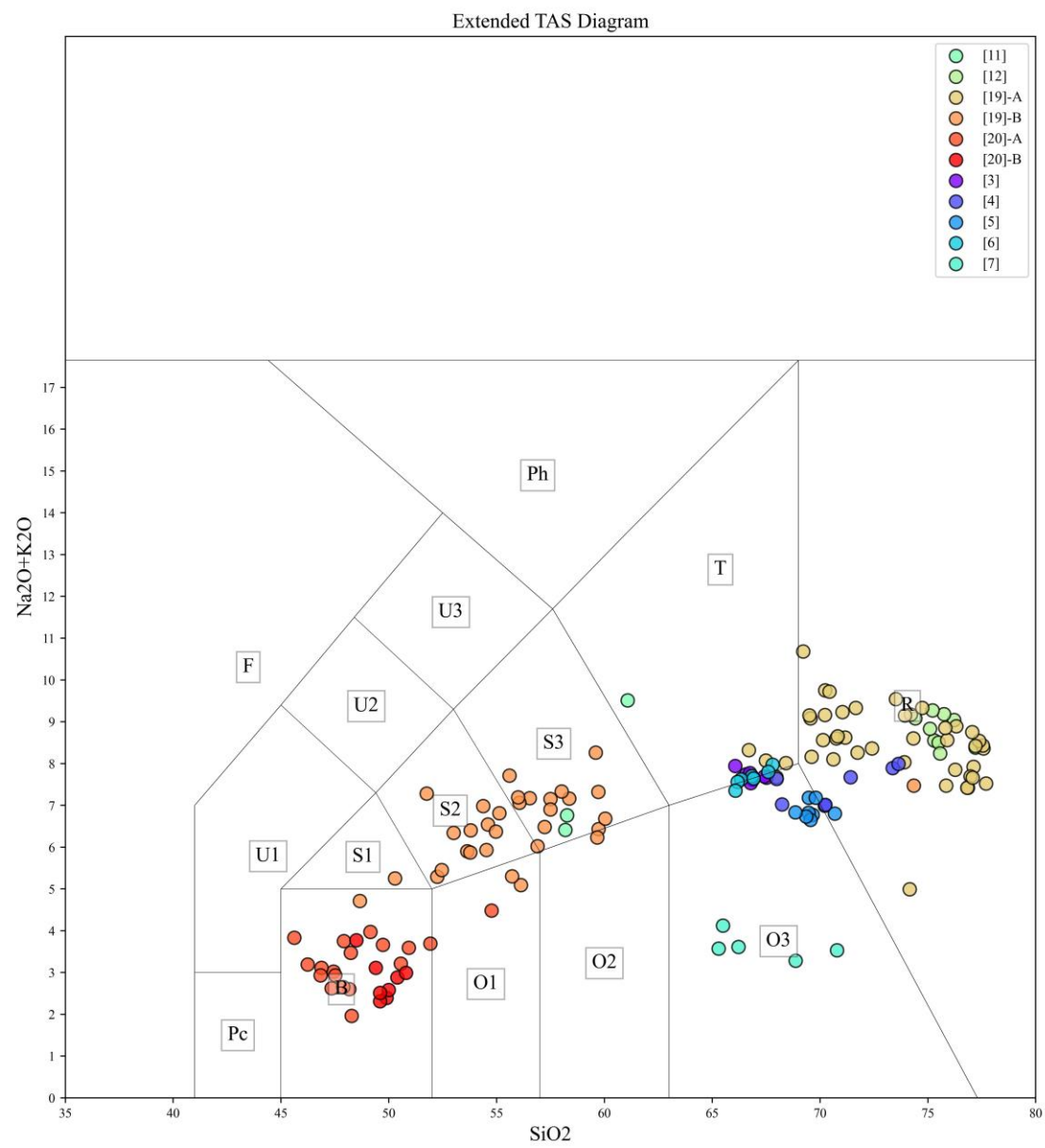
3.3 切换效果

在图形界面上，可以在 VOL 和 PLU 之间切换 GEOROC 底图数据，另外还可以切换是否显示线条，还有对底图显示的开关。

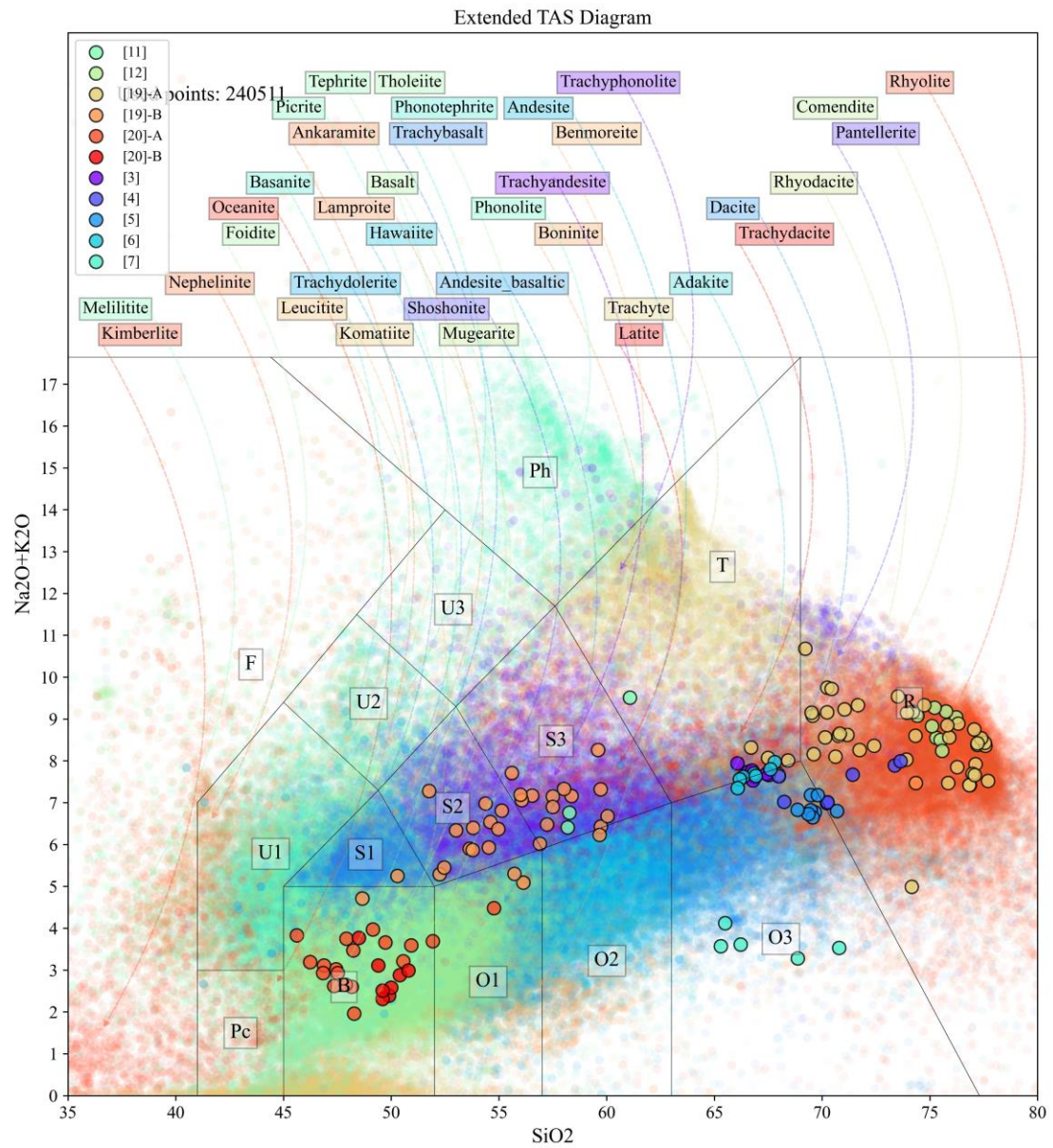


具体效果如后文附图所示。

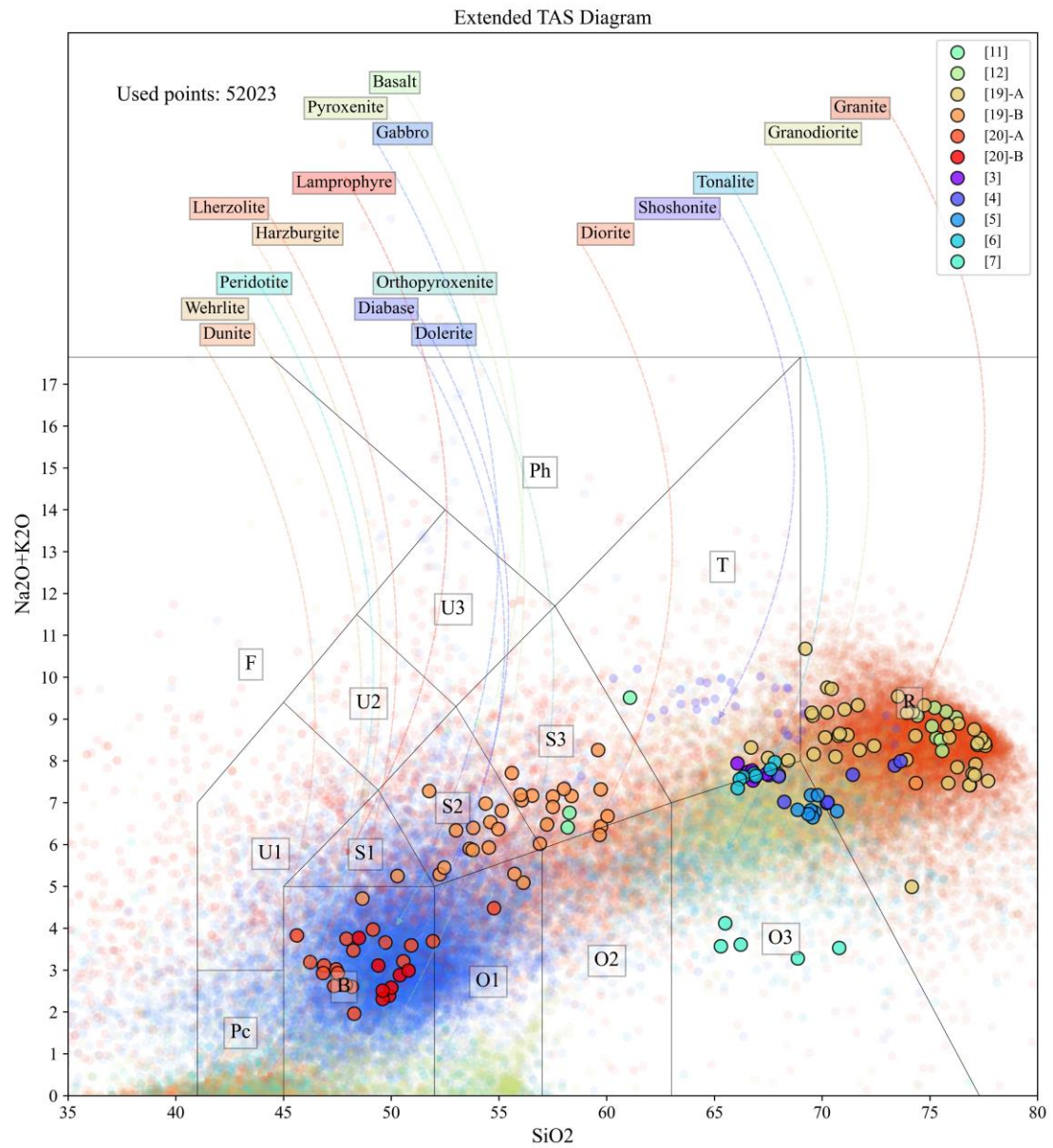
Traditional Diagram



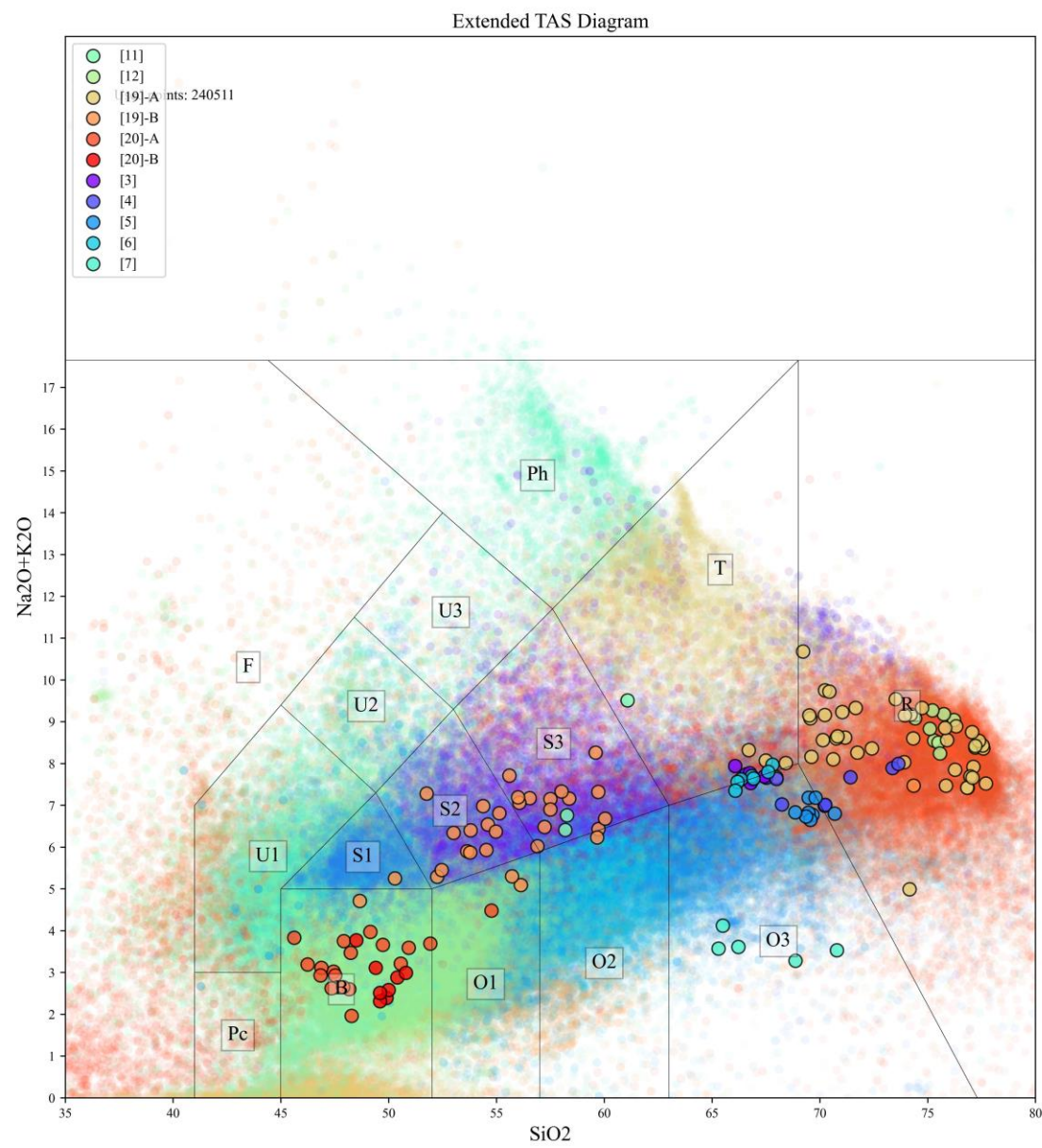
VOL with Probabilistic Field



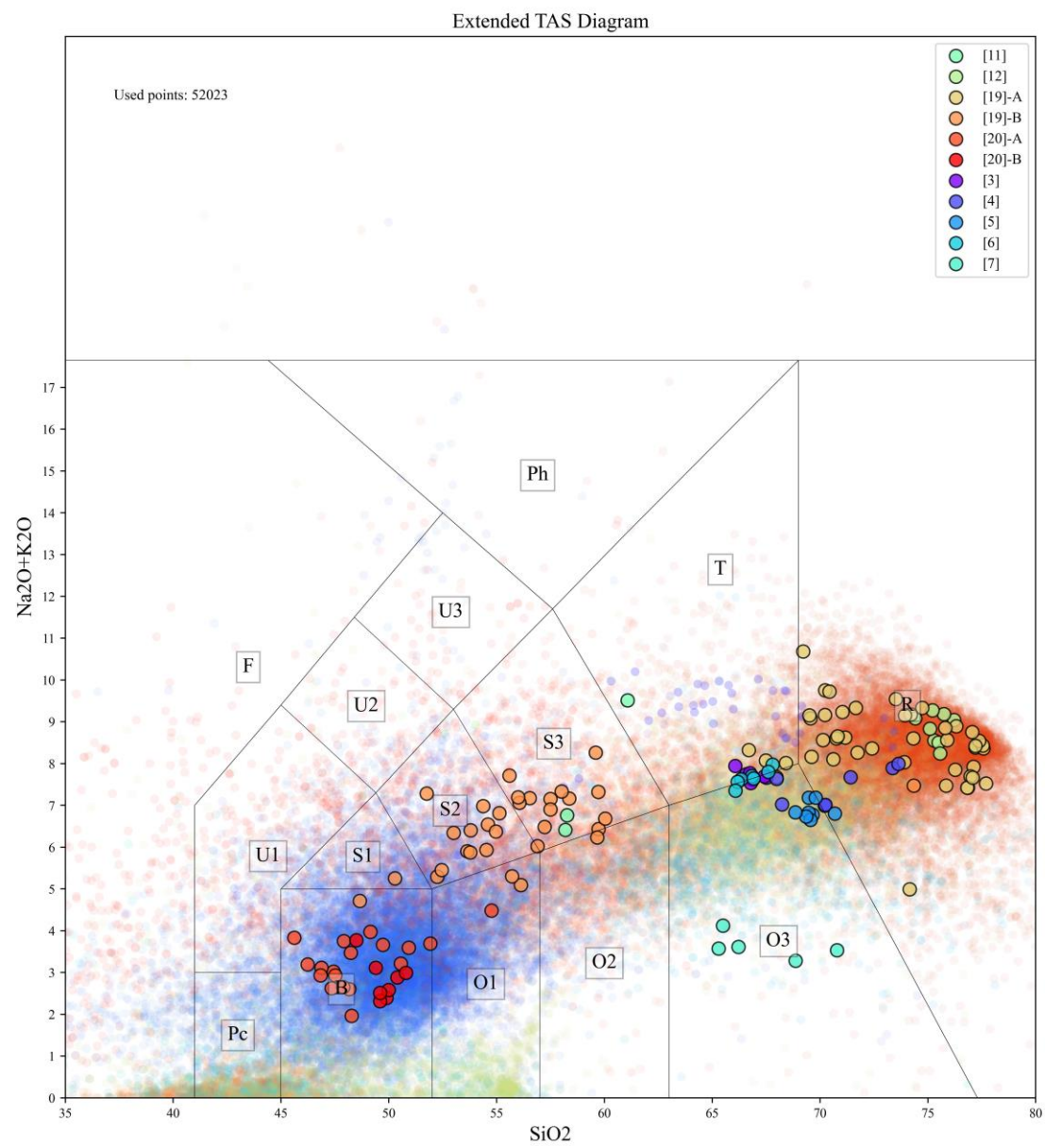
PLU with Probabilistic Field



VOL with No Lines

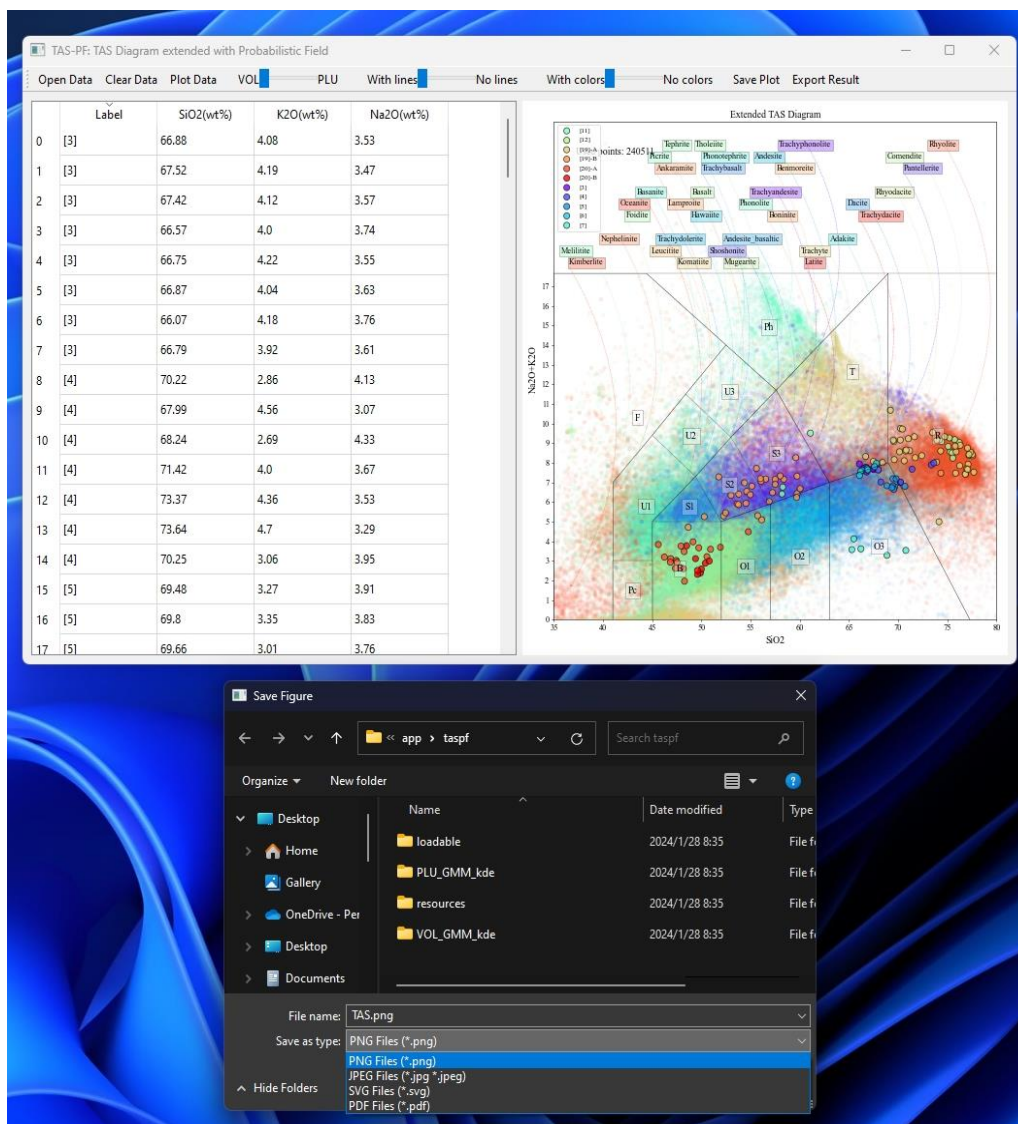


PLU with No Lines



3.4 保存投图

经过一番调整之后，如果你想要保存下来当前投图，就可以点击 **Save Plot**，如下图所示：



该程序提供了四种保存格式：JPG 和 PNG 两种是位图，PDF 和 SVG 两种是矢量图。大家如果用 CorelDraw 或者 Adobe Illustrator，推荐使用 PDF。如果使用 Ink Scape 或者 Affinity Designer，推荐使用 SVG。

4 致谢

该项目的灵感源于我已故的同学——李解博士。同窗逾十载，我们曾经一起组装电脑、学习编程、开发软件，也曾经讨论各种学术问题。不幸的是，在 2023 年，李解博士在毕业不久之后因病逝世。他曾对我早期项目中关于分类界限的过分应用提出过批判性的见解，而这个问题我一直未能寻得理想的解决方案。原本计划与他一同改进。奈何天妒英才，斯人已逝，如今只能独自尝试了。通过构建概率场的方式来弥补原有分类边界的不足，催生了本项目。这个项目的诞生是对李解博士深刻批评的回应，也承载着对他的怀念。