

GlyHunter

安装

移动到 glyhunter 目录（包含pyproject.toml的目录）。

```
cd path/to/glyhunter
```

通过 pipx 安装（请先安装 [pipx](#)）。

```
pipx install .
```

如果之前已经安装过，需要覆盖安装新版本，请使用 --force 选项。

```
pipx install . --force
```

初次使用

初次使用请初始化。

```
glyhunter init
```

该命令会在用户目录下创建一个 .glyhunter 文件夹，其中包含配置文件 config.yaml 和默认的糖库。

重新安装后一般不需要再次初始化。如果想要重新初始化，可以使用 `glyhunter init --force` 命令。

使用

概述

用法: glyhunter COMMAND [ARGS, OPTIONS]...

Commands:

- `config`: View and update GlyHunter configuration.
- `db`: View and update GlyHunter database.
- `init`: Initialize GlyHunter.
- `run`: Run the GlyHunter workflow.

Options:

- `config`:

- `--copy`: Copy the configuration file to a directory.
- `--update`: Update the configuration file.
- `-h, --help`: Show help.
- `db`:
 - `--copy`: Copy the database file to a directory.
 - `--update`: Update the database file.
 - `-h, --help`: Show help.
- `init`:
 - `-f, --force`: Force re-initialization.
 - `-h, --help`: Show help.
- `run`:
 - `-c, --config`: Specify the configuration file.
 - `-d, --database`: Specify the database file.
 - `-n, --denovo`: Run in De-Novo mode.
 - `-o, --output`: Specify the output directory.
 - `-a, --all-candidates`: Output all candidates.
 - `-h, --help`: Show help.

运行 GlyHunter

GlyHunter 可以直接处理 flexAnalysis 导出的 mass list (XLSX文件)。

```
glyhunter run data.xlsx
```

此处, data.xlsx 是 flexAnalysis 导出的 mass list 文件。GlyHunter 会创建一个文件夹, 名为 XLSX 文件的文件名 (不包含后缀), 加上 "_glyhunter_results"。例如, 在上面的例子中, GlyHunter 会创建一个名为 data_glyhunter_results 的文件夹。

文件夹中包括:

- 每个谱图的结果 (原 XLSX 文件中的不同 sheet) 会单独保存到一个 CSV 文件中, 文件名为原 sheet 的名字。
- 三个 summary 表格, 分别汇总了 intensity, area 和 sn。

修改全局配置文件

GlyHunter 的配置文件是一个 YAML 文件, 名为 config.yaml, 默认位于用户目录下的 .glyhunter 文件夹中。如果要修改配置文件, 请先使用 `glyhunter config --copy` 将配置文件复制到某一目录, 然后再修改。

例如:

```
glyhunter config --copy /Users/username/Desktop
```

此时, /Users/username/Desktop 会出现一个名为 config.yaml 的配置文件副本。请修改该文件并保存, 具体的修改方法在配置文件中有详细说明。修改完成后, 运行 `glyhunter config --update` 命令更新 GlyHunter 的配置。

```
glyhunter config --update /Users/username/Desktop/config.yaml
```

修改全局糖库

GlyHunter 的糖库是 BYONIC 文件，位于用户目录下的 .glyhunter 文件夹中。默认糖库为人血清N-糖库。如果要修改，请使用 `glyhunter db --update` 更新配置文件。同样，可以用 `glyhunter db --copy` 命令保存当前糖库的副本。

BYONIC 文件的格式为：糖组成 % 糖质量，例如：

```
Hex(5)HexNAc(2) % 2039.742
```

注意，% 后面的质量会被 GlyHunter 忽略，因为 GlyHunter 会自己计算质量。

在运行时指定临时配置或糖库

可使用 `-c` 或 `-d` 命令在运行时临时指定配置文件或糖库。例如：

```
glyhunter run data.xlsx -c /Users/username/Desktop/config.yaml -d  
/Users/username/Desktop/db.byonic
```

使用这种方法指定的配置文件或糖库只会在本次运行中生效，不会影响全局配置。

指定输出文件夹

GlyHunter 默认输出的文件夹是输入的 XLSX 的文件名加上 "_glyhunter_results"。如果要修改输出的文件夹名称，使用 `-o` 或 `--output` 选项。

例如：

```
glyhunter run data.xlsx -o results
```

输出所有候选结果

GlyHunter 默认输出和每个峰最匹配的糖组成结果。如果一个有多个糖组成都在某个峰的 tol 内，ppm 最小的那个糖组成会被当做最匹配的糖组成。如果要输出所有候选结果，使用 `-a` 或 `--all-candidates` 选项。

在 "all candidates" 模式下，在每张谱图的结果中，同一峰可能会有多个糖组成结果，即一个峰的搜索结果可能有多行。例如，对于一个分子量为 1663.265 的峰，可能有以下两个结果：

| glycan | raw_mz | calibrated_mz | theoretical_mz | ... |
|-----------------|----------|---------------|----------------|-----|
| Hex(5)HexNAc(4) | 1663.265 | 1663.580 | 1663.582 | ... |

| glycan | raw_mz | calibrated_mz | theoretical_mz | ... |
|------------------------|----------|---------------|----------------|-----|
| Hex(3)HexNAc(2)dHex(5) | 1663.265 | 1663.580 | 1663.607 | ... |

可以以 raw_mz 或 calibrated_mz 为每个峰的标识。此外，在此模式下，GlyHunter 不会生成 summary 表格，因为特定样本中特定糖的含量无法确定。

De-Novo 模式

GlyHunter 默认使用搜库模式，即使用糖库进行糖注释。GlyHunter 还支持 De-Novo 模式，即不使用糖库，而是根据分子量计算所有可能的糖组成。使用 `-n` 或 `--denovo` 选项运行 De-Novo 模式。

```
glyhunter run data.xlsx --denovo
```

使用 De-Novo 模式时，GlyHunter 会忽略糖库，即使在配置文件（默认配置或使用 `-c` 指定的配置）中指定了糖库也不会使用。此外，`--denovo` 选项不能与 `-d` (`--database`) 选项同时使用。

De-novo 模式的计算非常耗时，且可能产生大量假阳性结果。为了减少假阳性结果和加快计算速度，请在配置文件中编辑 `constraints` 参数，限制每个单糖的数量。