

流式果蝇计数分选仪 用户手册及入门指南 V 0.0.1

Fluyd system document

乌肝达工作组



2019

地球上的一切工具和机器，不过是人肢体的知觉的延伸而已

——托马斯·爱迪生

您好，欢迎使用流式果蝇计数分选仪！希望我们的微小工作能为您的科研生活带来快捷与便利。以下是本文档的目录：

**用户手册**

1 概述

2 导言

3 硬件结构

4 可视化控制台

5 脚本与终端控制台

6 模块化

**入门指南**

1 导言：人的领域的扩展

2 安装与部署：硬件与软件

3 可视化控制台：在窗口化界面直观操作

4 终端命令及脚本：更高，更快，更强

5 模块化：连接您的上下游组件

**用户手册**

**1 概述**

果蝇流式计数分选系统（FluYd System）是一套自动化实现果蝇检测、鉴别、分选、计数的模块化系统，在原理上借鉴了流式细胞仪的思路。本系统的软件和硬件全部开源，遵循自由软件协议。我们维护了硬件系统模型和软件系统源码的代码仓库。您可以在gitlab和gitee上阅读下载完整的代码。

“自由”表示我们欢迎使用者对本系统及其源码的一切修改、研究与传播，这意味着您可以借助这套系统的框架实现您的更多想法与需求，如：计数分选其他可能的固体生物质材料、内嵌学习训练给气装置、使用更复杂的行为检测软件等等。为了方便使用者的扩展，我们提供了丰富的软件和硬件接口，这些内容会在后面的部分详述。

工作组始终欢迎更多有意愿的同志加入我们的开发之中。作为一个带有公益项目的团队，我们欢迎以个人名义进行的捐款。所有募得款项除用于必要的开发与项目维护外，其余都将通过透明途径捐献给中国贫困山区及乌干达的儿童。感谢您对我们的支持！

**2 导言**

果蝇是一种普遍用于神经生物学、遗传学乃至发育生物学研究的模式生物，其遗传背景明确、饲养方便、行为区分度高。在科学研究的过程中，果蝇的杂交、突变型筛选、行为检测甚至简单的分装都需要研究者高重复性的精细复杂劳动。使用肉眼识别和手动挑选耗时费力，且在长时间劳动后难免出现失误，而这些失误极有可能导致整个实验结果的假阳性或假阴性，进而对研究造成严重的后果。提高果蝇分析及筛选的效率与准确性能够显著提高科研效率，节约研究者精力。机器是人类手眼脑的扩展，利用机器使科研更快、更便利、更舒适，将更多的研究者从高强度重复性劳动中解放出来，从而将更多精力投入到更具启发性的工作中是我们的更级目的。

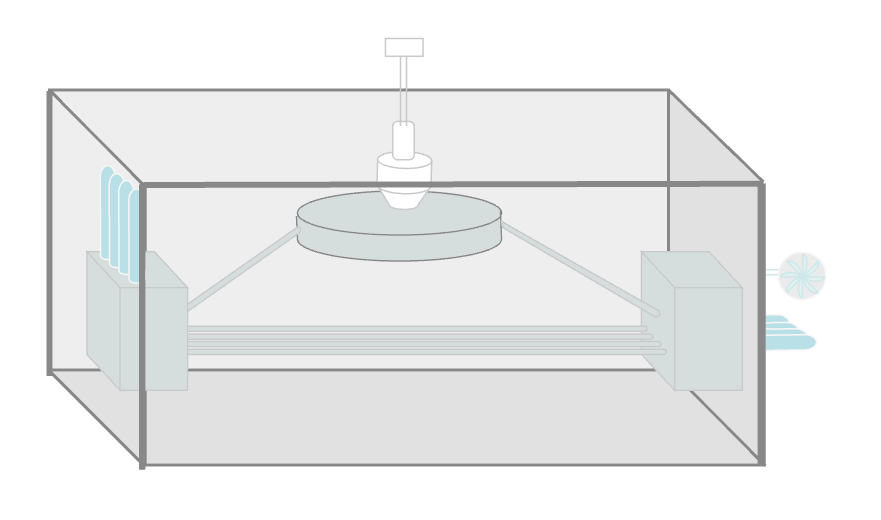
我们借鉴了流式细胞仪的思路，构建了一套高通量流式活体成蝇分选系统。本系统作为一个灵活的功能模块，能够拆解为分选系统与行为分析系统，也可与解剖体系自由拼接，兼具实用、简便、可塑性强的特点，具有良好的应用前景。

**3 硬件结构**

本仪器由适配接口、抽吸装置、影像装置、阀门管路结构、给气系统、分析控制平台六部分组成(图1)。

图 1 仪器概览

1. 适配接口 B) 抽吸装置 C) 影像装置 D)阀门管路系统



D

C

B

A

**3.1 适配接口**

适配接口的可复用性是仪器通用性的最直接体现，处于整个仪器流程的两端（图1 A）。分为进与出两类接口，根据实验室果蝇培养的习惯，各自留有四个独立接口。接口型号分培养管型和培养瓶型、清洗型三种，可自由更换。接口使用橡胶塞或螺纹盖密封，每个接口均有长短中三根玻璃管（图2 A），其中长管为进出气管，短管为进出果蝇管，中管为给气管。长管末端有细纱布遮蔽，防止果蝇顺长管飞出，短管末端位于锥筒结构内（图2B ），方便果蝇进出。中管末端有细纱布遮蔽，与给气系统连接。清洗型的入端只有长管，用于吸入清洗液（乙醇等），出端长短管相反，设计为安全瓶结构。

除了仪器自带的接口外，您还可以根据使用的容器换用更为多元的接口套件。这些接口既可以由您在已有容器上自行改装，也可以在我们的官网和开源代码库中免费下载完整的OpenSCAD模型源码并进行3D打印。我们也会根据您的反馈更新我们的接口模型，以保证实时有效的接口通用性，也欢迎您用您设计的新接口丰富我们的接口模型库。

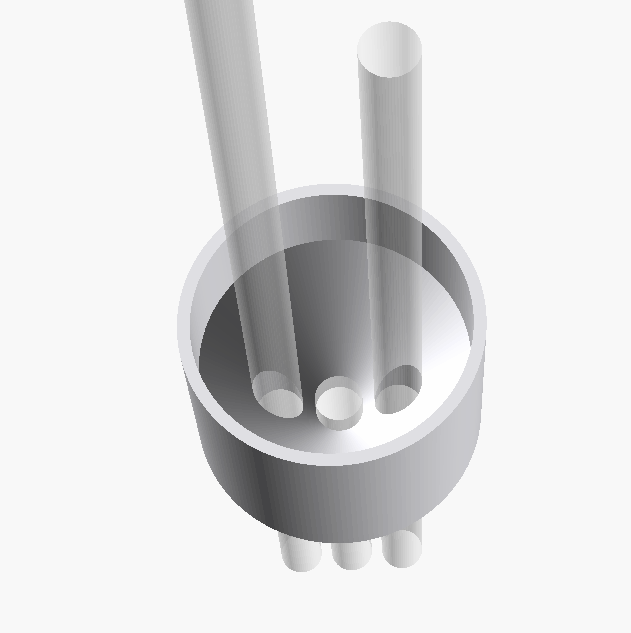
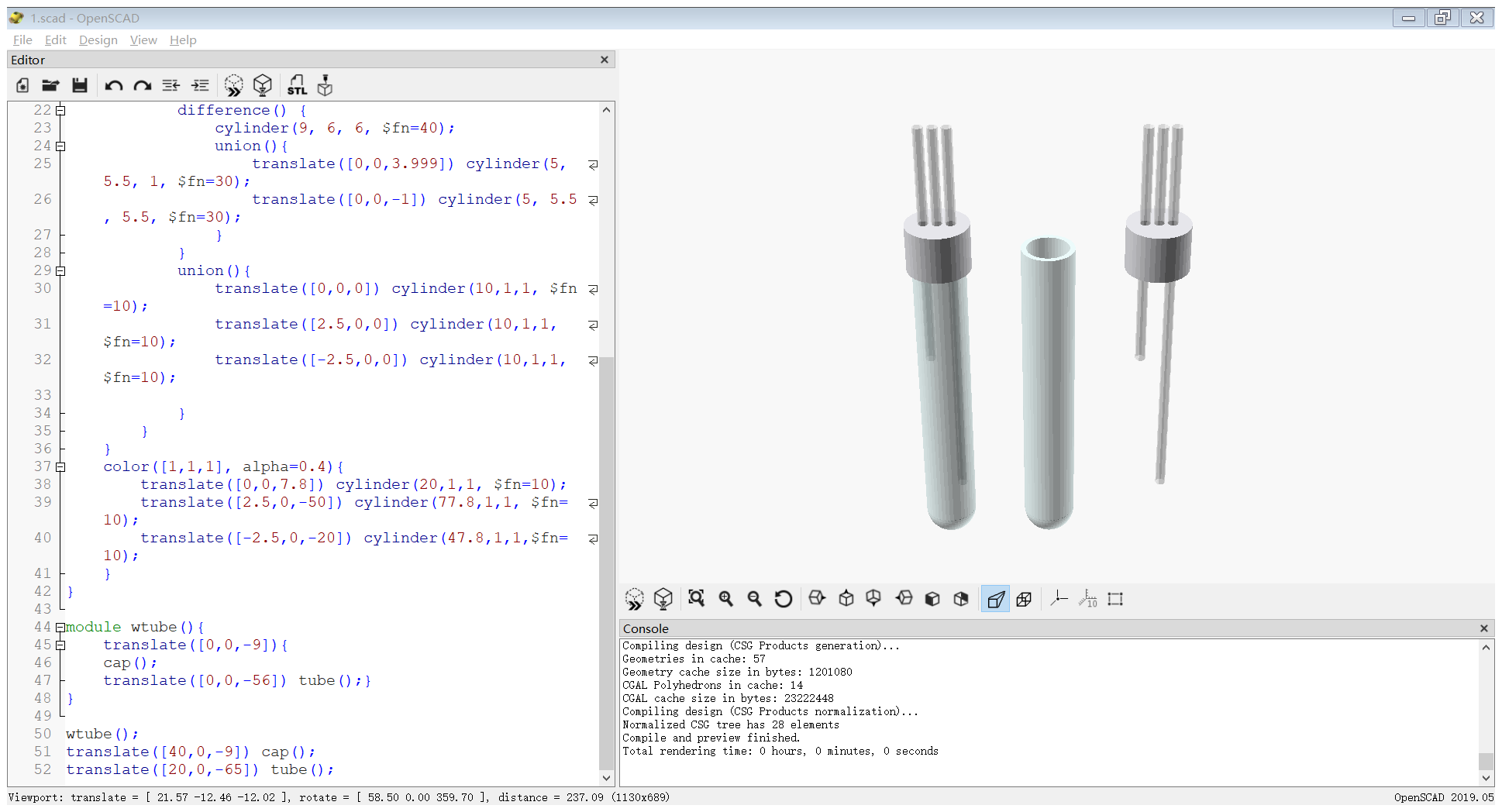


图 2 适配接口

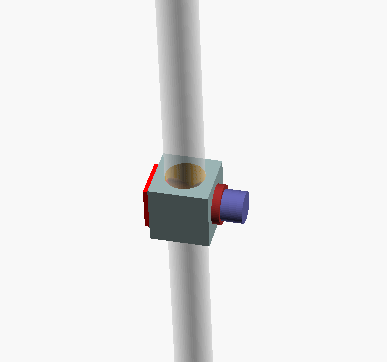
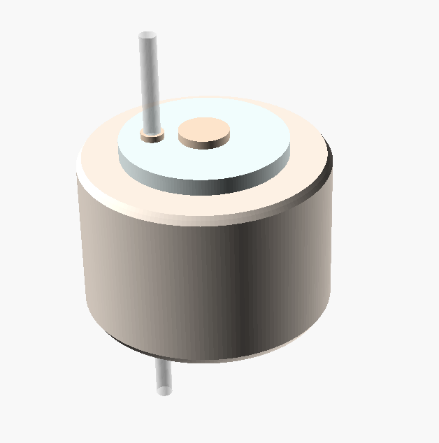
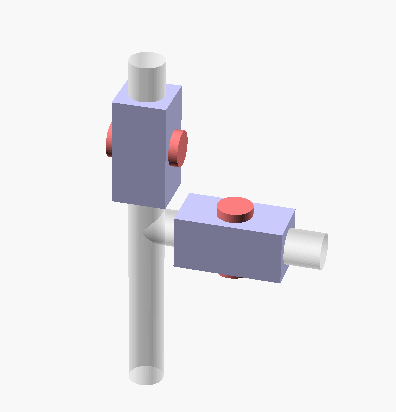
1. 接口的装卸 B）三管的位置构造

B

A

**3.2 抽吸装置**

抽吸接口为本仪器移动果蝇的动力来源（图3）。我们使用微型直流抽气泵实现果蝇的输送，如果您需要动力更大的气泵，可以使用常用于无人飞行器的微型电机进行改装，我们的模型库中提供适合常见尺寸微电机的气泵外壳（图3 B）。抽气泵经过一道电磁阀连接到对应的输出接口的长管。抽气泵的速度为匀速且可使用外接API进行调控。抽气泵可以造成输出容器内的均匀负压，进而产生由输入容器到输出容器的均匀气流。抽吸装置出气端接有一个T型管道（图3 A），在进行果蝇转移时，出气端接向大气。在果蝇转移结束后，T型转接到阀门系统中，进行一次高速气流反吹，以防止果蝇残留在管路中。在清洗时，则由抽吸装置产生负压，将输入端的清洗液经由整套管路系统吸入输出端的安全瓶中。



C

B

A

图 3 管路、抽吸及计数装置

A）T型管与电磁阀 B）直流微电机 C）激光检测计数装置

**3.3 影像装置**

本仪器使用两套影像装置，用于转移计数的八个激光检测计数器（图3 C）和一套大体显微摄像装置（图4）。均有API直接连接到分析控制平台。激光检测计数器设计在由输入瓶进入后的第一个电磁阀处和中央平台输出后第一个电磁阀处，用于快速转移果蝇时的计数及检测果蝇的入管状况。激光计数器使用一个微型激光发射器和一个光敏二极管组成，会实时将阻断信息传输到分析控制平台上。大体显微摄像装置采用市面上已经成型的设备（如ME10显微摄像头）结合照明装置完成，帧率保证在10fps以上，同时保证快门速度。大体显微摄像装置安置在中央平台处，管道在此处设计为一略微膨大扁平的玻璃管，以确保果蝇姿态为背面或腹面朝上。摄像头和照明均紧贴玻璃管，以防止反光影响观测结果。摄像头利用USB给电并将视频信号实时传输到分析控制平台。

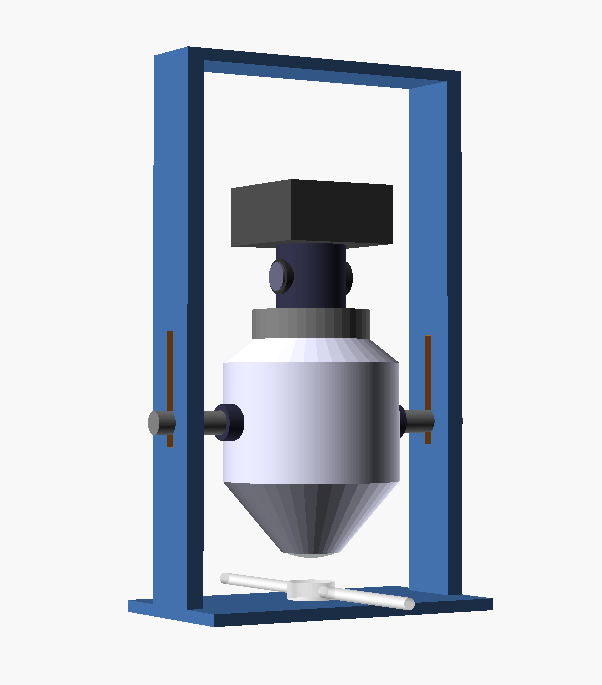


图 4 大体成像系统

分为框架、大体摄像头及分筛室中央平台三部分

**3.4 阀门管路系统**

本套设备具有复杂的阀门管路系统，所有阀门均为电磁橡胶阀（图3 A）或非接触式电磁珠阀门，由分析控制平台统一调控。根据不同的功能需求，管路结构可以进行一定的调整。当需要四对四快速转移时，中央平台输入输出管道全部关闭，直通电磁阀全部打开，激光检测计数器检测果蝇的进出数量。当需要合并时，中央平台入口阀门打开，直通输入阀门关闭，输出阀门仅打开一号，以使得四个输入端果蝇能够并入一个输出瓶中。当需要分瓶时，中央平台输入电磁阀仅打开一个，其余输入阀门全部关闭，四个电磁阀依次打开，输出计数器每检测到五只果蝇通过，就更换一个输出端，以保证分瓶后果蝇数目差别在5只以内（具体数目可以根据用户需求设置调整）。当需要进行筛选鉴别时，抽吸装置使用低功率低速工作，将果蝇吸入中央平台，获取影像信息，传输到分析控制平台，根据反馈选择打开不同的输出阀，以完成果蝇的分选。

**3.5 给气系统**

给气系统为模块化独立系统，可以根据实验室需求选择进行安装，使用模块自带的气泵进行送气。常用的有行为实验中训练使用的O、M芳香气、信息素，麻醉用的CO2、乙醚气及处死用的乙醇蒸汽。给气系统各自通向输入瓶和输出瓶的中管。根据是否配备给气系统，也可以对接口的管道数目进行选择（即是否配置中管）。

**3.6 分析控制平台**

拟设计为单片机与外接API结合的双分析控制系统，以减少接口压力，同时节约成本。对于计数及阀门控制具体操作，利用单片机完成。而视频图像处理、阀门控制指令发放则由外置API连接到PC端计算机。对于除了鉴别分析的简单操控，可以利用蓝牙模块连接到个人手机的安卓端进行控制，简单易行。对于复杂的鉴别分析图像处理，在计算机平台进行，首先利用计算机视觉的手段对果蝇图像的特征进行提取和仿射变换，以减少果蝇姿态对分析的影像。再使用CNN卷积网络对果蝇的特征进行分析，与预设的学习结果进行比对，完成果蝇的鉴别分类。我们已利用这套系统建立起不同突变株、性别的果蝇的数据集，用户也可以根据实验室已有的果蝇品系对系统进行再训练，以个性化满足不同的研究需求。这部分的具体内容会在之后部分详述。

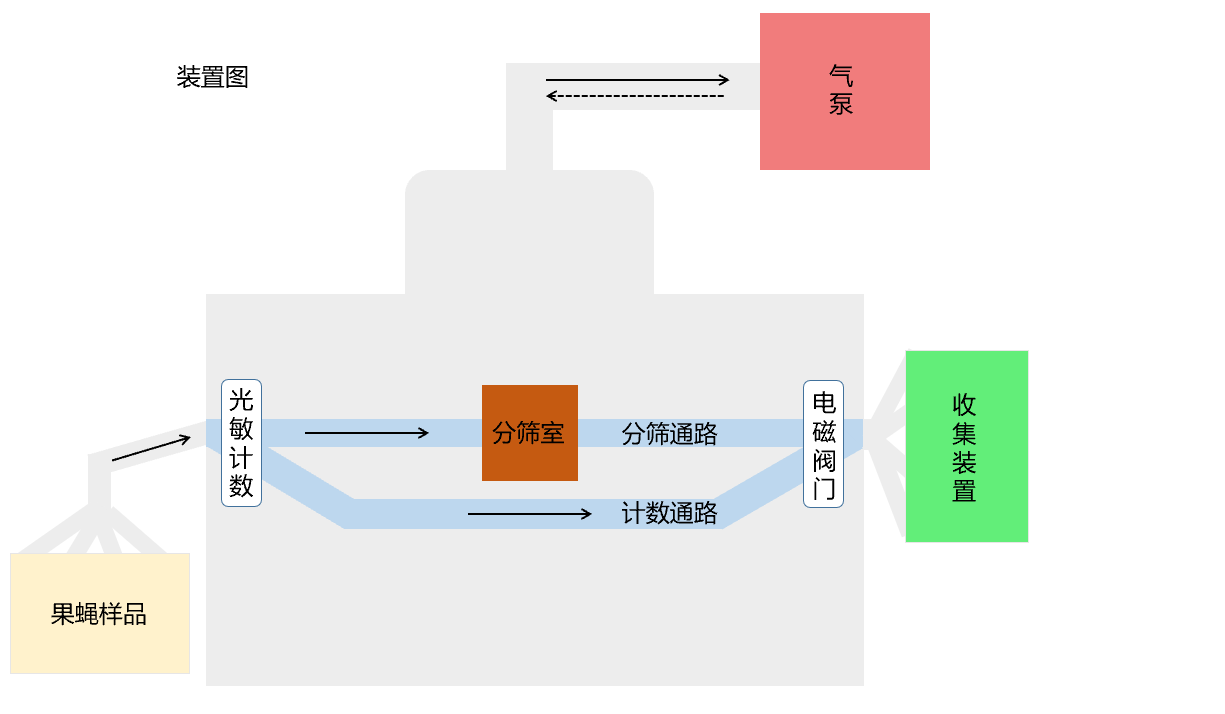


图 5 装置示意图

**4 可视化控制台**

我们的硬件系统信息首先汇总到单片机，再由USB端口发送到PC或安卓端。对于图像处理等工作，计算机平台显然更得心应手，且使得整套系统更易于维护、移动、移植和操作。

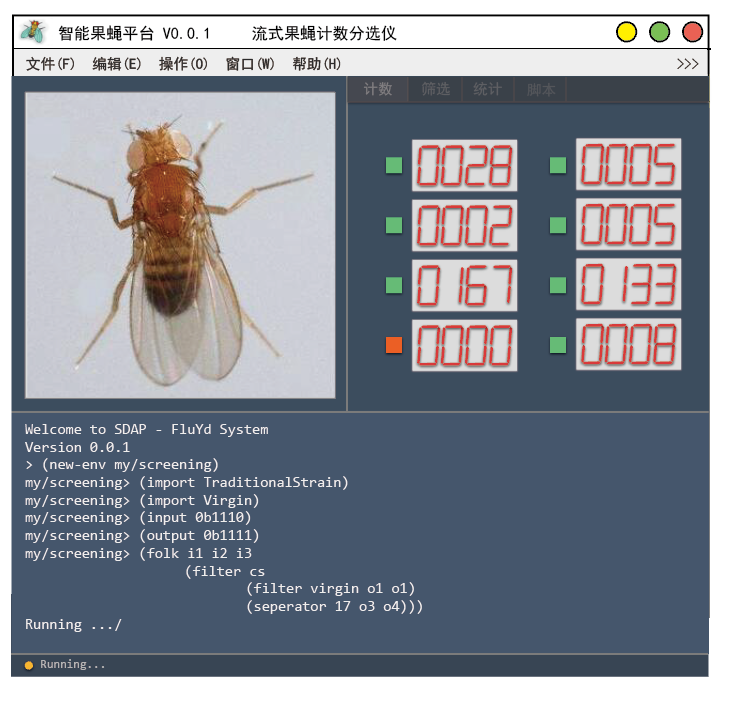


图 6 窗体概览

我们的软件系统全称为智能果蝇平台（Smart Drosophila Analyze Platform, SDAP）。使用Qt编写GUI，以实现跨平台功能。我们目前能够完美支持Windows 10、Linux、OSX等主流系统，后续还会继续开发相应的Android和iOS版本。与此同时，我们也在考虑移植到浏览器端，以提供更好的视觉体验和跨平台性。我们的程序框体和内部分析模块使用C++编写，内嵌的脚本语言解释器语法则基于Scheme实现，语法形式也有典型的Lisp风格，这是源于Lisp类语言在流处理方面天生的优越性与Scheme的易学易上手特征。

对于快捷键的描述，我们使用类Emacs的模式，前缀C代表PC端的Ctrl和Mac的Control键，前缀M代表PC的Alt和Mac的Option，前缀S或大写字母代表Shift。

**4.1 菜单栏**

本平台的大多数功能可以通过菜单栏进行实现。

**4.1.1 文件（F）**

文件菜单下的选项涉及到对文件的操作。其中主要是对环境、脚本、学习集、统计数据及影像资料的操作。

1. **新建**

* **新建环境（C-E）**

新建并转到一个新的环境（工作区）。本平台的工作范围由工作区界定，不同工作区间相互独立。新环境内只包含预置函数，不包含任何学习集和样本库。不同环境可以打开在多个窗体内。如果您用同台PC连接多个平台，也可以通过调用multithread库实现多线程处理。对于转移、分支与合并等高通量简单工作来说，多线程可以大大提升通量。不过对于需要运用计算机视觉和深度学习手段的分选筛查工作，还是建议您首先谨慎考虑计算机性能。新建的环境默认名为Untitled\_Env.sev。在可视化环境内部包含导入的模块与库、创建的函数、命名空间及变量、学习集甚至用户偏好和保存时的窗体状态。

* **新建脚本（C-N）**

脚本是本套平台的灵魂。新建脚本会自动打开脚本选项卡。脚本默认使用UTF-8编码的平文本书写。采用我们扩展的称为DroScheme的语言，其基于Chez Scheme实现，是R6RS的一个超集。新建的脚本默认名为Untitled.scm，支持编译。

* **新建学习集**

新建一个空的果蝇形态学习集，自动打开统计选项卡，并跳转到录制模式。在统计选项卡的第一行第一列输入新的果蝇形态的名称，选中第二列并点击“操作→录制学习集”，同时输入相应名称的果蝇样本。学习的参数可以在首选项中进行调整。之后再选中第二行第二列，录入新的果蝇品系。每个学习集都可以录入多种果蝇品系的特征。我们目前仅支持有监督的学习，之后拟尝试加入聚类等无监督学习的方式，以促进对新品系的发现，扩展本平台的用途。

1. **打开（C-x C-f）**

打开脚本或环境，脚本的后缀名建议为\*.scm，当然我们也支持一切UTF-8编码的平文本文件。环境的后缀名则建议使用\*.sev。

1. **保存（C-x C-s）**

将脚本或数据保存到硬盘位置。建议您在编写脚本的时候随时使用保存快捷键，相信我，这个好习惯会给您提供极大的帮助。

1. **另存为**

顾名思义，将训练集、脚本或统计数据另存为。需要焦点在相应选项卡内才能操作。

1. **导入（C-i）**

导入模块、库、统计数据或学习集。模块和库的后缀都是\*.so，统计数据支持csv格式，学习集也是csv格式的文件，但我们使用\*.dlc（意为果蝇学习CSV）来加以区分。

1. **导出**

* **导出工作环境**

将您新建好并完成配置的工作环境导出成\*.sev文件，以便下一次打开使用。好的工作环境能让您的工作更为舒心愉悦。

* **导出图像（C-M-A）**

体视摄像头抓拍并导出到环境所保存目录下。

* **导出统计结果**

将统计选项卡内数据或学习集整体导出，文件后缀为\*.csv或\*.dlv格式。

1. **退出**

直接退出窗体。

**4.1.2 编辑（E）**

编辑菜单下各种功能与常见文本编辑软件相似。

1. **撤销（C-z）**

撤回前一步操纵。

1. **重做（C-Z）**

重做被撤销的操作。

1. **剪切（C-x）**
2. **复制（C-c）**
3. **粘贴（C-v）**
4. **查找（C-s）**
5. **替换（C-i）**
6. **首选项**

用户的本地化自定义窗口。在首选项窗口内，您可以调整平台的主题、快捷键位、API硬件接口等诸多细节设置。

**4.1.3 操作（O）**

操作菜单主要涉及到对硬件接口的直接操控。具体需要根据进一步的硬件设计来确定。操作内含有大量表意模糊的子选项，建议将这些操作转移到脚本和命令行端进行。

1. **生成脚本并开始工作（F5）**

根据GUI端的选项、文本框、复选框等交互式信息，自动生成完整的控制脚本。并配合硬件端运行脚本。

1. **生成脚本**

根据GUI端的选项、文本框、复选框等交互式信息，自动生成完整的控制脚本。

1. **工作（Space）**

直接在硬件端运行脚本，开始工作。需要脚本窗口内有首个合法脚本。

1. **休止（Space）**

临时暂停，所有电磁阀关闭，抽吸系统关闭。

1. **终止（ESC）**

停止，进行一次高速喷气，将管路中残存的果蝇吹出。并将平台状态除统计窗口的数据外还原到运行前。

1. **语法检查**

简单检查脚本的语法，包括拼写、缩进、括号匹配、对象属性、流的完整性等。

1. **清洗**

配合清洗接口，使用乙醇溶液对仪器的管路系统进行自动化清理。

1. **录制**

将实时图像信号提取出的果蝇学习集录制到统计窗口中。

1. **过滤器生成**

将学习集内指定行转化为同名的过滤器函数回调。

1. **编译（F）**

将脚本编译成\*.so等运行较快的连接前程序库。

**4.1.4 窗口（W）**

这部分涉及到各个窗体的布局，不同窗体的具体内容将在后续部分详述。

1. **默认布局**

恢复默认窗口布局。

1. **布局设计器**

打开一个用户自定义的布局设计器。可用的控件有Tab选项卡，水平分割框架，垂直分割框架，悬浮窗及各种窗口对象。

1. **实时图像**

打开或关闭实时图像系统。

1. **终端**

打开或关闭终端。

1. **计数**

默认在选项卡中。

1. **筛选**

默认在选项卡中。

1. **统计**

默认在选项卡中，下还分下一级选项卡和工作簿。

1. **脚本**

默认在选项卡中，下还分下一级选项卡，为不同的脚本缓冲区。

**4.1.5 帮助（H）**

**a. 帮助文档**

打开完整的用户手册。

**b. 入门指南**

打开缩减版的入门指南，方便新用户迅速上手

**c. 关于**

包括我们的团队介绍、版本信息及捐赠途径。

**4.1.6 自定义（>>>）**

单击菜单栏最右侧三箭头可以打开一个下拉式菜单，初始状态是空白的，用户可以通过脚本、终端控制台或首选项偏好设置将自己常用的功能或自己编写的模块、函数挂在这个菜单中。

**4.2 状态栏**

状态栏位于窗体的最下方。显示平台目前所处的状态。左侧的指示灯表示了程序的资源占用状况。红色代表运行错误，黄色代表运行中资源占用，绿色代表内核空闲，处于准备状态。文字则具体描述了其状态，如黄光及Running代表正在常规运行中。黄光及红光配合Tube Blocked表示硬件系统出现阻塞。Runtime Error则很可能是软件平台内存溢出及调用非法对象等问题。绿光及Ready则是已准备运行。以下是状态对应表格。

|  |  |  |
| --- | --- | --- |
| 指示灯 | 文字 | 状态 |
| 红 | Runtime Error | 程序运行时错误 |
| 红 | Tube Blocked | 硬件阻塞 |
| 红 | Syntax Wrong | 有简单的语法错误 |
| 黄 | Compiling | 正在编译中 |
| 黄 | Running | 正在运行中 |
| 黄 | Checking | 语法检查中 |
| 黄 | Cleaning | 自动清洁中 |
| 绿 | Ready | 准备完毕 |
| 绿 | Successful | 成功完成 |

**4.3 实时图像**

显示体视摄像头拍摄的实时图像。实时图像窗口可以打开侧边栏，在侧边栏上具有调节亮度、调节焦距、调节放大倍数的按钮。同时也可以在首选项中调节实时图像的帧率。

实时图像也可以用于显示果蝇的统计流式散点图和其他检测图片。

**4.4 终端**

终端中及包含使用者的输入与交互，也包含平台的状态与输出。如果并非运行状态，则终端的最新行将显示为[env]>，[env]代表当前所处环境。具体将在下一章详述。

**4.5 计数**

计数窗口中有四行，代表左侧四列输入和右侧四列输出。每个输入都具有一个指示灯和一个模拟LCD计数屏。指示灯显示端口开启状态，绿色代表开启，红色代表关闭，单击指示灯即可控制端口的开关。每次停止运行后，计数结果会保存到统计窗口的新文件中。

**4.6 筛选**

筛选窗口能够实现最简单的分筛工作，需要与计数窗口配合使用。其中有四行内容，代表四个输出端口，每行有一个下拉列表，即从环境内选择所需的过滤器。

**4.7 统计**

统计窗口中为典型的表格编辑窗口，下方有录制按钮和选项卡。录制按钮可以在\*.dlc文件中录制学习集内容，选项卡则可以在打开的统计文件中切换。如果导入或录制学习集，则将会显示为一列名称和一列指向对应果蝇数据的指针。

**4.8 脚本**

脚本编辑器具有一定的的Emacs特色，支持大多数Emacs快捷键。上方有工具栏，其中包含复制、剪切、粘贴、撤回等常用操作。左侧有行号显示。有自动缩进4格功能。

**5 终端控制台与脚本**

SDAP的脚本遵循简单的语法规则，既可以在终端窗口交互式解释，也可以脚本窗口中书写，编译或直接送入解释器，还可以使用外部文本编辑器编写。我们设计了一套名为DroScheme的语法规则，直接实现在Chez Scheme实现上，是Scheme（R6RS）的一个超集。

SDAP中的所有脚本内容都与Scheme对象对应，所有的对象均为一阶对象，即都可以作为参数与返回值，因而直接支持Lambda表达式。对于流的处理和操作也可以基于Lambda表达式进行。

**5.1 DroScheme基本语法**

这里简单介绍DroScheme的基本语法，实际是Scheme语言的语法规则。更为详细的可以参看R6RS与tspl4（The Scheme Programming Language）。

Scheme语言使用括号定界，符合波兰表达式。括号表达式内的第一项为操作符（即过程），每个括号表达式在运算表达后会返回一个返回值。如(+ 6 4)的返回值为10。括号表达式可以嵌套，如(+ 6 (\* 3 5))返回21。

**5.1.1 对象**

DroScheme中的每个对象都是一阶对象，即一视同仁地可以用作参数输入值和函数返回值。这些对象包括数、过程、符号、字符、字符串、广义表、向量。

数包含实数、整数、分数、复数等。支持形式化的分数表达，如2/3、不限制尾数长度的实数、无范围限制的整数、常规复数，如1+5i，极坐标表示的复数，如5@90、科学计数法等。

过程是Scheme对象，具有过程与值的对偶性。过程有参数与返回值，如果不显式返回，则返回值为空。接受几个输入值，并返回一个输出值。过程可以作为参数，并在另一个过程中使用，过程支持递归调用。

符号为一个标识符对象，用单引号前缀。

字符为一个字符对象，用#前缀。

字符串为字符组成的线性结构，使用双引号包围，结尾默认为\0

广义表和Scheme对象是对等的，一般用单引号前缀与Scheme语句区分。如’(+ a b)被视为有含有一个+过程，两个变量名标识符a与b的广义表，使用eval过程可以把广义表作为Scheme语句表达。

向量则用#( )包裹。

对象可以用define绑定到一个标识符，如(define a 3)。绑定会限制在变量名作用域内。

**5.1.2 环境操作**

DroScheme具有很多作为基本过程的环境操作。

* (new-env [envname])： 新建并转到一个名为envname的环境，envname为标识符，不需要前后缀，如(new-env my/screening)后，命令行的前缀即变为my/screening>
* (open-env [envpath])： 打开路径envpath的环境，可以使用相对路径，可以包含，但不必要包含后缀。envpath为一个字符串。
* (export-env [envdoc])：将当前环境输出到envdoc的目录下，可以使用相对路径。envdoc为一个字符串。
* (import [library])：导入库，库名为library。库文件默认放在环境目录下。

**5.1.3 过程**

DroScheme中的过程包含初级过程、语义扩展及用户自定义过程。DroScheme中的初级过程在Scheme的基础上进行了扩展，语义扩展涉及到Scheme的高级用法，在本文档中不再赘述。仅就用户自定义过程进行介绍。

Scheme中的过程可以表示为lambda表达式的形式，是lambda表达式表达计算的结果。初级过程lambda可以返回一个新的过程,如：

(lambda (x) (+ x 2))

返回一个过程，可以接受输入参数x并返回x+2作为结果。用法则是：

> ((lambda (x) (+ x 2)) 4) => 6

可以使用define初级过程将自定义过程绑定到标识符上。

> (define add-two (lambda (x) (+ x 2))) => <#procedure add-two>

> (add-two 4) => 6

这样就可以实现函数式编程中代码的复用。

**5.1.4 递归调用与循环结构**

DroScheme支持完全的递归调用，不主动限制调用栈深度。循环结构则利用尾递归调用实现。在我们的工作中，可以使用尾递归调用完成重复性的工作。

**5.2 果蝇流**

有一系列处理果蝇流的初级过程。

**5.2.1 接口启用**

接口序列使用一串四位二进制序列表示，以0b作为前缀，如0b1110表示前三个接口，0b1111表示四个接口。

* (input [adaptor])：开启input接口
* (output [adaptor])：开启output接口

开启的接口将表示为i1, i2, i3, i4和 o1, o2, o3, o4。

**5.2.2 流操作**

本套系统具有复杂的流操作。参数内所有的stream既可以是接口，也可以是另一个流操作的返回值。

* (merge [inputstream...] [outputstream])：合并流，将复数的输入流汇总到输出流中（最后一个参数即为输出流）。并返回一个果蝇流。
* (filter [strain-type] [outputstream1] [outputstream2])：strain-type为一个返回布尔值的果蝇判断操作。返回一个输入流对象，接受外界的输入并进行后续的分支操作，若满足判断，则输出到第一个流，若不满足，则输出到第二个流。
* (separator [ratio] [outputstream1] [outputstream2])：分配操作，返回一个输入流对象，将输入的果蝇流按照ratio的比例分配到两个输出流中。
* (run)：开始运行。
* (end)：终止。
* (pause)：暂停。

**5.2.3 过滤操作**

果蝇品系判定为一个布尔操作。是根据学习的结果获得的。我们使用计算机视觉与CNN深度学习的方式进行果蝇学习。

* （learn-strain [strain-name] [adaptor]）：将adaptor标明的输入接口(使用二进制表示)输入的品系学习特征并绑定在strain-name上，如果strain-name已有绑定，则在已有模型上继续学习。
* (record)： 录制学习集，在learn-strain后使用。

**5.3 多线程**

我们支持多线程，正在考虑支持CUDA实现的GPU多线程操作。

* (multithread [procedure])：将过程加入多线程
* (run-multithread)：运行多线程

**5.4 解释器**

解释器既可以在窗口内嵌的终端中打开，也可以在外部终端命令行中打开。

**5.5 编译器**

可以在内部进行编译。

* (compile [src] [tar])：将源文件编译到目标文件。

**6 模块化**

作为一款现代化、高灵活性、高鲁棒性的系统，我们采取了模块化的设计思路，不仅仪器内的部件可以根据需求进行更替，还可以在上下游加装不同的模块，您可以自由发挥您的想象力，分析您的需求来设计您的模块架构，以下是两个经典案例。

**6.1 学习训练工作流程**

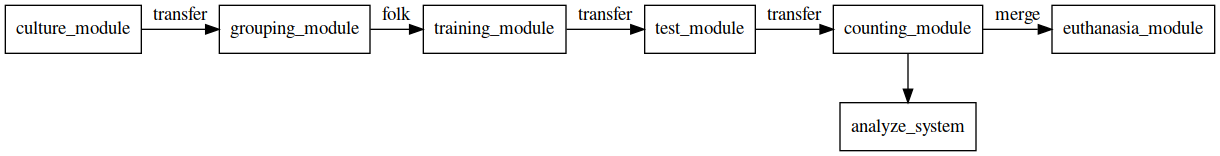
由培养模块转移到分组与挑选模块,分组模块分支到不同训练管,再转移到测试模块。测试模块转移到计数模块进行统计,计数模块发送信息到分析系统,计数模块中的果蝇转移到处死模块(动物福利安乐死)。

图 7 学习训练流程图

**6.2 性别、处女蝇和突变型筛选系统工作流程**

基于分组模块的功能独立性，引入影响影像模块对果蝇的相应表征进行鉴定。影响模块的图像信息被送往分析模块进行下一步分析。

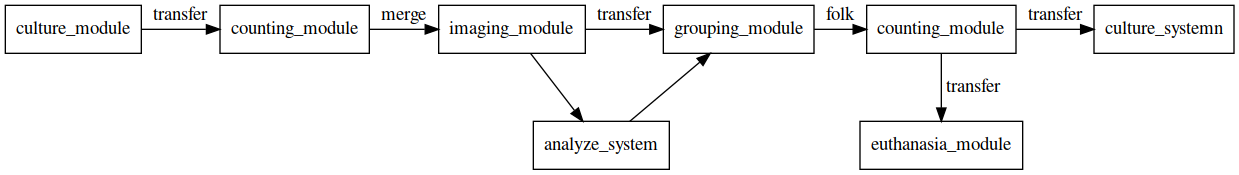


图8 性状分选流程图