WeGene 轻应用开发文档

版本：0.2

1. 说明

轻应用是 WeGene 为生物信息开发者提供的轻量级基因应用的开发引擎，帮助生物信息开发者在不使用 WeGene 开放平台 API 的情况下利用脚本语言来直接开发各种基因数据分析应用并相应部署到 WeGene 平台上。在测试阶段，所有的轻应用均可被 WeGene 用户免费使用，但未来我们会逐步开放轻应用的收费和开发者分成的功能。目前，轻应用支持 `Python 2.7` 及 `R` 语言。

2. 名词定义

- 轻应用：部署在 WeGene 服务端、入口在 WeGene 用户网站上的一个应用
- 平台：WeGene，可狭义理解为 WeGene 的用户网站
- 开发者：一个轻应用的开发者
- 用户：一个轻应用的使用者，同时该使用者也是 WeGene 自身的用户

3. 架构

一个典型的轻应用共分为 3 个部分：1）输入；2）计算；3）输出。输入数据由平台直接根据开发者在申请轻应用时要求的数据，得到用户授权同意后利用其已经存在于平台上的基因数据生成并输入。输入数据经由开发者自定义的计算脚本进行计算，并从脚本输出结果。该输出结果被平台读取后，将会被渲染成用户可见的结果页面。

简而言之，开发者在整个轻应用的开发中，只需专注一件事：**开发核心的基因数据分析算法和脚本**。该脚本只需按一定的规则处理输入、输出即可。需要注意的是，计算过程中不可使用任何网络资源。

4. 工程架构

下面介绍如何开发一个轻应用。附件 `weapp_example_python.zip` 与 `weapp_example_r.zip` 为相应的工程文件。

一个可以被在平台上被部署的轻应用必须是打包为 `.zip` 的代码包，其中包含如下的内容：

- （必选）主计算脚本 `main.py` 或 `main.R` - 轻应用执行的主体，需要读取输入、进行计算、输出结果。**整个轻应用的计算过程目前需要在 120 秒内完成。**
- （可选）依赖代码 - 如果主计算脚本依赖其他开发者自定义的依赖代码，这些代码可以被存储在其他代码文件中。开发者只需要在主计算脚本中通过相应的**相对路径**进行引入。
- （可选）第三方依赖包定义文件 `requirements.txt`（Python） 或 `pacman.R`（R）- 如果轻应用依赖其他的第三方包，开发者只需要将所需的依赖按各自语言的依赖定义文件上传即可（无需上传实际依赖文件）。目前，Python 开发的轻应用使用 pip 进行包管理，R 开发的轻应用使用 pacman 进行包管理。
- （可选）参考文件 - 如果轻应用计算需要任何参考文件，同样只需要在代码中通过相应的**相对路径**引入即可。

请注意，上述的文件在打包时，不要包含上一级的目录，即直接将所有的文件全选打包

下面是两个轻应用工程的典型构成：

`Python`

```
weapp_example_python.zip
├── indexes （参考文件）
|   └── index_wegene_affy_2.idx
├── main.py （主计算脚本）
├── requirements.txt （第三方依赖包定义文件）
└── wegene_utils.py （依赖代码）
```

R

```
weapp_example_r.zip
├── indexes
|   └── index_wegene_affy_2.idx  (参考文件)
├── main.R  (主计算脚本)
├── pacman.R  (第三方依赖包定义文件)
└── wegene_utils.R  (依赖代码)
```

4.1 主计算脚本

主计算脚本需要获取输入、计算、输入输出。其中，输入必须通过 `stdin` 获取而输出需要通过 `stdout` 输出。输入会以 `json` 格式输入（见后文），而输出则建议为轻应用针对基因数据分析的一句话结论。输出不可以含任何代码（html、markdown 等）如果在计算的过程中遇到任何错误，需要在 `stderr` 中抛出。在获取输入和打出输出之间是轻应用核心的计算逻辑。下面是主计算脚本在 Python 和 R 中的例子：

`main.py`

```
# 永远从 stdin 读取输入
body = sys.stdin.read()

try:
    # 解析输入的 json 数据
    inputs = json.loads(body)['inputs']

   # 开始进行数据分析和计算
    result = do_something(inputs)

   # 输出结果
    print result
except Exception as e:
    # 输出计算过程中的异常
    sys.stderr.write(e.message)
```

`main.R`

```
# 永远从 stdin 读取输入
body <- readLines(file('stdin', 'r'), warn = F, n = 1)

tryCatch({
    # 解析输入的 json 数据
    inputs <- fromJSON(json_str = body)

   # 开始进行数据分析和计算
    result <- do_something(inputs)

   # 输出结果，使用 cat 而不是 print，这样结果中不会有行数被打印
    cat(result)
}, error = function(e) {
    # 输出计算过程中的异常
    write(conditionMessage(e), stderr())
})
```

请注意，计算过程中打印出的结论或异常信息有可能对用户可见，请以对用户友好的形式输出。例如，如果输入的数据不全，打印 `很抱歉，您的部分数据缺失，暂时无法计算` ，而非 `data missing xxxx`

4.2 第三方依赖

第三方依赖通过定义文件定义，凡是在定义文件中定义过的依赖包，均可以被轻应用使用。轻应用代码中不可引入没有定义的第三方依赖。

`requirements.txt`

```
numpy==1.11.2
scipy==0.18.1
```

`pacman.R`

```
pacman::p_load(base64enc, R.utils, rjson)
```

然后可以在计算脚本中引入这些依赖——

`main.py`

```
import numpy
import scipy
```

`main.R`

```
# 引入依赖时需要禁用警告，否则会生成额外的错误日志
suppressMessages(library(base64enc))
suppressMessages(library(R.utils))
suppressMessages(library(rjson))
```

4.3 输入参数

在 ***4.1*** 中我们提到，通过 `stdin` 传入给轻应用的参数永远是 `json` 格式的数据，并且数据存储在`inputs` field当中。`inputs` 中传入的数据由创建应用时选择的入参决定，可能的值有——

字段名	字段说明	举例	值选项
format	基因数据的格式	`"format": "wegene_affy_2"`	`wegene_affy_2`，`23andme`，`ancestry`，`ancestry_2`
data	`gzip` 压缩并经过了 `base64` 编码的全部位点数据，需要配合index文件解析	`"data": "xfgakljdflkja..."`	
rsxxxxx	某个rs位点上的基因型数据	`"rs671": "AA"`	
sex	性别	`"sex": 1`	`0`（缺失），`1`（男），`2`（女）
age	年龄数据	`"age": 173007321`	以 Linux Timestamp 存储的生日
haplogroup	单倍群	`"haplogroup": {"mt": "A", "y": "O1"}`	有 `mt`、`y` 两个单倍群数据
ancestry	祖源	`"ancestry": {"block": {"xxx": "0.25"}, "area": {"xxx": "0.2"}}`	`block` 为大区域祖源数据，`area` 为小区域祖源数据，值存储为字符串格式

- 祖源数据中，各大区域的可能值如下表——

字段名	字段说明
chinese_nation	中华民族
ne_asian	东北亚
se_asian	东南亚
south_asian	南亚
central_asian	中亚
middle_eastern	中东
african	非洲
european	欧洲
american	美洲
oceanian	大洋洲

- 祖源数据中，各小区域的可能值如下表——

字段名	字段说明
han_southern	南方汉族
han_northern	北方汉族
dai	傣族
tungus	通古斯族群
lahu	拉祜族
she	畲族
miao_yao	苗瑶语族群
mongolian	蒙古语族群
uygur	维吾尔族
tibetan	藏族
gaoshan	高山族群
ny	纳西/彝族
korean	韩国人
japanese	日本人
yakut	雅库特人
thai	泰国人
cambodian	柬埔寨人
kinh	越南京族
mala	印度人
bengali	孟加拉人
sindhi	信德人
kyrgyz	柯尔克孜族
uzbek	乌孜别克族
iranian	伊朗人
saudi	沙特阿拉伯人
egyptian	埃及人
mbuti	姆布蒂人
yoruba	约鲁巴人
somali	索马里人
bantusa	南非班图人
french	法国人

sardinian	意大利撒丁岛人
finnish_russian	芬兰/俄罗斯人
hungarian	匈牙利人
balkan	巴尔干半岛
spanish	西班牙人
ashkenazi	德系犹太人
english	英国人
eskimo	因纽特人
pima	美洲土著
mayan	墨西哥玛雅人
papuan	巴布亚人

依据上面介绍的各个字段，根据申请应用时要求的入参，可以分为两种情况：

- 轻应用申请了用户的全部基因位点数据 - 此时，传入的数据是通过 `gzip` 压缩并经过了 `base64` 编码的基因序列数据。需要配合相应的注释文件进行解析。如何处理编码后的基因序列并解析可以参考示例代码中的 `wegene_utils.py` 或 `wegene_utils.R`，过程中需要的参考注释文件（即 `indexes` 文件夹）会自动存在于轻应用的运行环境下，无需打包在代码包中上传。

```
{
  "inputs" : {
    "age" : 173007321,
    "data" : "xfgakljdflkja...",
    "sex" : 1,
    "haplogroup" : {
      "mt" : "A",
      "y" : "O1"
    },
    "ancestry" : {
      "block" : {
        "southeast_asia" : "0.000020",
        "else_asia" : "0.000050",
        "...": "0.000050"
      },
      "area" : {
        "uygur" : "0.000010",
        "russian" : "0.000010",
        "..." : "0.000010"
      },
    "format" : "wegene_affy_2"
    }
  }
}
```

- 轻应用申请了用户的部分基因位点数据 - 此时，传入的数据时可以直接使用的位点数据。

```
{
  "inputs" : {
    "sex" : 1,
    "haplogroup" : {
      "mt" : "A",
      "y" : "O1"
    },
    "rs12203592" : "CA",
    "rs671" : "--",
    "rs..." : "AA",
    "ancestry" : {
      "block" : {
        "southeast_asia" : "0.000020",
        "else_asia" : "0.000050",
        "...": "0.000050"
      },
      "area" : {
        "uygur" : "0.000010",
        "russian" : "0.000010",
        "..." : "0.000010"
      }
    }
  }
}
```

5. 创建轻应用

轻应用的创建可以在 WeGene 平台上实现。开发者在创建轻应用过程中可以申请需要的用户数据并上传轻应用的工程包。上传完成后，轻应用将在后台经过一段时间的自动构建。如果构建成功，开发者可以测试轻应用并发布。发布完成后通过管理员审核即可被用户使用。如果在构建、测试过程中轻应用出现了错误，则不可发布。开发者可以通过重新上传工程包进行修改直到该轻应用可用。

轻应用申请界面

Q & A

- 什么样的应用适合轻应用?

  对单用户基因数据进行轻量级分析的任务，例如根据基因数据计算每个用户每天需要摄入的碘盐含量、某种疾病的风险或 K12 祖源分析等。由于轻应用执行时的两大限制（120秒运行时间、不可请求网络资源），轻应用不适合进行计算复杂的离线分析。

- 在轻应用的计算过程中，可以调用第三方程序进行计算吗?

  可以，但不建议。如果要使用的第三方程序较小，可以在代码工程包中直接打包，并且以相对路径进行调用。由于目前我们限制上传的工程包不能超过 10 mb，如果第三方程序体积较大，则无法支持。同时，由于操作系统的差异，在本地测试中可以运行的第三方程序并不一定能够在轻应用的运行环境中使用。建议您联系我们在轻应用的运行环境中直接安装需要的第三方程序以供调用。

- 开发者可以获得用户的原始基因数据吗?

  不能，整个轻应用的数据授权、输入完全在 WeGene 平台上完成，并且在轻应用被执行完成后就会销毁。开发者在这个过程中无法获取到用户授权的数据。

- 轻应用的输出只能是一句话吗?

  技术上目前我们没有做限制，但出于用户体验和前端页面展示的考虑，一句话的结论是最合适的。未来我们的平台会增加对部分 Markdown 语法的支持。目前输出中的 `\n` 将会被解析为换行。