

## **Требования к системе и предустановленное программное обеспечение**

Для установки, запуска и стабильного функционирования программы требуется сервер со следующими минимальными системными требованиями (в случае разворачивания ПО на одном сервере):

- Количество физических ядер процессора (CPU) - 4;
- Оперативная память (RAM) - 32 Гб;
- Дисковое пространство (HDD) – 256 Гб.

Рекомендуемые требования (для случая разворачивания ПО на одном сервере):

- Количество физических ядер процессора (CPU) - 16;
- Оперативная память (RAM) - 128 Гб;
- Дисковое пространство (HDD) – 256 Гб.

Кроме того, сервер должен располагать GPU класса Nvidia, GTX-1080Ti и подключен к ИБП.

В случае использования различных серверов (таких как сервер приложений, сервер для хранения объектов на файловой системе) пользователем должны быть проведены мероприятия по обеспечению и поддержке сетевой связности между серверами.

Окружение для штатной работы веб-сервера Jupyter должно включать операционная система Ubuntu версии 22.XX или выше.

Программная совместимость отдельных частей и подпрограмм ПО обеспечивается за счет контейнеризации средствами Docker. Для запуска и работы программных модулей требуется предусмотреть наличие на сервере дистрибутива Docker (версия  $\geq 20.10$ ), драйверов для Nvidia GPU (версия  $\geq 450.80.02$ ), Nvidia Container Toolkit (версия  $\geq 2.12$ ).

Исходные коды ПО написаны на языках Python, R и C/C++.

## **Состав и структура дистрибутива**

Программа представлена как набор папок и файлов, формирующих единый дистрибутив:

- `docker`: папка, содержащая `Dockerfile` - файл с описанием программного окружения и зависимостей, необходимых для работы ПО
- `DL_template`: папка, обеспечивающая функционал основного модуля обучения нейросетевых моделей разных типов для предсказания ГФЭ.

Включает папки:

- `artifacts`, содержащую результаты работы модуля – `checkpoints` моделей и полногеномные предсказания моделей;
- `data`, в которую пользователь должен загрузить исходные данные для построения моделей;
- `assets`, содержащую два поясняющих изображения для шаблона `Template.ipynb`;
- `models`, содержащую примеры различных типов нейронных сетей;

и файлы

- `Template.ipynb`, представляющий пример работы с модулем и осуществляющий запуск работы модуля через редактирование шаблона.
- `README.mb` с документацией модуля для пользователя
- `OmicDC`: папка, обеспечивающая функционал модуля загрузки данных.

Включает папки:

- `OmicDC`, содержащую имплементацию функционала модуля загрузки данных
- `storage` для сохранения результатов по умолчанию

а также файлы:

- `README.mb` с документацией модуля для пользователя

- Example.ipynb, представляющий пример работы с модулем и осуществляющий запуск работы модуля через редактирование шаблона
- Association: папка, обеспечивающая функционал модуля исследования признаков на ассоциацию с ГФЭ. Включает папки:
  - Association , содержащую имплементацию функционала модуля исследования признаков на ассоциацию
  - storage для сохранения результатов по умолчанию
- DomainAdaptation: папка, обеспечивающая функционал модуля трансферного обучения для переноса аннотации ГФЭ. Включает папки и файлы:
  - DomainAdaptation, содержащую имплементацию функционала модуля трансферного обучения
  - storage для сохранения результатов по умолчанию
  - conda\_requirements.yml с необходимыми библиотеками и их версиями для установки с помощью conda
  - pip.requirements с необходимыми библиотеками и их версиями для установки с помощью pip
- Interpretation: папка, обеспечивающая функционал модуля интерпретации обученных моделей. Включает файл:
  - lrp.py, содержащий имплементацию функционала модуля интерпретации обученных моделей на языке программирования Python

## **Настройка программы**

Для настройки программы следует:

1. Зайти на сервер с помощью протокола SSH и установить SSH-соединение с переадресацией портов, используя следующую команду:

```
ssh -L localhost:8888:localhost:8888 -p <server-port>  
<username>@<server-address>
```

Перенаправить локальный порт 8888 на удаленный порт 8888.

Заменить адрес сервера для подключения и номер порта.

2. Скачать репозиторий на сервер:

```
git clone --recurse-submodules https://github.com/hse-bioinflab/framework
```

3. Создать контейнер Docker со всеми необходимыми зависимостями:

```
cd docker
```

```
docker build -t framework:1.0 .
```

```
cd ..
```

4. Запустить интерфейс Jupyter Lab:

```
docker run -it --rm --runtime=nvidia --gpus all \
```

```
--shm-size=4GB -p 8888:8888 \
```

```
-v $(pwd):/workspace/ \
```

```
framework:1.0
```

5. Проверить свою консоль на наличие URL-адреса, начинающегося с `http://127.0.0.1:8888/lab?token =`. Скопировать URL-адрес и открыть его в своем браузере.

6. Сохранить сеанс SSH запущенным во время работы.