ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(НИУ ВШЭ)

УТВЕРЖДАЮ

Руководитель
Исследовательского
центра в сфере
искусственного интеллекта
(должность)

Масютин А.А. _____
(ФИО)

«28» июня 2023 г.

М.П.

Программное обеспечение

«Библиотека программных и аналитических средств (фреймворка), направленных на предсказание расположения геномных функциональных элементов методами глубинного обучения на основе омиксных данных молекулярной биологии»

Руководство системного программиста

Лист утверждения

НИУ ВШЭ. 00003-01 32-ЛУ

СОГЛАСОВАНО

Заведующий
международной
лабораторией
биоинформатики
(должность)

Попцова М.С. _____
(ФИО)

«28» июня 2023 г.

Подпись и дата | Инв. № дубл. | Взам. инв. № | Подпись и дата | Инв. № подл.

Утвержден

НИУ ВШЭ.00003-01 32-ЛУ

Программное обеспечение

«Библиотека программных и аналитических средств (фреймворка), направленных на предсказание расположения геномных функциональных элементов методами глубинного обучения на основе омиксных данных молекулярной биологии»

Руководство системного программиста

НИУ ВШЭ.00003-01 32

Листов 16

2023 г.

Аннотация

В данном документе приводятся общие сведения о программе, описание структуры, настройки и проверки программы, описание дополнительных возможностей программы, а также сообщения системному программисту программного обеспечения «Библиотека программных и аналитических средств (фреймворка), направленных на предсказание расположения геномных функциональных элементов методами глубинного обучения на основе омиксных данных молекулярной биологии».

Оглавление

1.1 Общие сведения о программе

1.2 Назначение и функции программы

ПО предназначено для применения биоинформатиками, биологами, медиками и другими аналитиками геномных данных с целью получения поддержки в принятии решений по предсказанию геномной разметки функциональных элементов с использованием экспериментальных данных по расположению исследуемых ГФЭ и омиксных данных на основе нейронных сетей разных типов. Программа решает следующие прикладные задачи:

1. Выбор, скачивание и переформатирование общедоступных омиксных данных из открытых источников.
2. Обучение и применение сверточных, рекуррентных, гибридных сверточно-рекуррентных, генеративно-состязательных, графовых нейронных сетей, а также нейронных сетей типа трансформер для предсказания расположения ГФЭ.
3. Анализ геномных признаков на ассоциацию с ГФЭ.
4. Трансферное обучение для переноса аннотации ГФЭ на геномы без доступных экспериментальных данных.
5. Интерпретация нейросетевых моделей разных типов архитектур, обученных для предсказания расположения ГФЭ.

1.3 Сведения о технических и программных средствах

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.

Окружение для штатной работы веб-сервера Jupyter должно включать:

- Операционная система Ubuntu версии 22.XX и выше;
- Предустановленные библиотеки и драйверы (см. раздел «Требования к информационной и программной совместимости» Технического задания).

2. Структура программы

2.1 Сведения о структуре программы

Программное обеспечение реализовано как библиотека программных и аналитических средств, направленных на решение биоинформатических прикладных задач Центра ИИ.

Архитектура ПО подразумевает наличие следующих логических уровней:

- Фронтенд;
- Бэкенд;
- Файловая система сервера.

Программа состоит из следующих логических уровней, указанных в таблице 2.1:

Таблица 2.1. — Составные части программы

Сервисы	Описание	Технологии
Jupyter Lab	Фронтенд, позволяющий пользователю производить загрузку данных на сервер, загрузку общедоступных омиксных данных, последовательностей полных геномов с удаленных серверов, а также выбор, конфигурацию и обучение моделей на пользовательских данных	JupyterLab
Jupyter server	Веб-сервер, взаимодействующий с файловой системой сервера и передающий команды на выполнение Jupyter Kernel	Jupyter Server

Jupyter Kernel	Исполняет Python, R или bash код, и возвращает результат в Jupyter Server	JupyterLab
Файловая система сервера	Заранее заданная структура директорий, в которых содержатся необходимые скрипты и объекты для запуска ПО	

Файловая система должна включать следующие директории:

- docker
- DL-template
- OmicsDC
- Association
- DomainAdaptation
- Interpretation

Директория «docker». Содержит Dockerfile, позволяющий контейнеризовать программное обеспечение.

Директория «DL-template» обеспечивает функционал основного модуля обучения нейронных сетей разных типов для предсказания ГФЭ.

Директория «OmicsDC» обеспечивает функционал модуля загрузки омиксных данных.

Директория «Association» обеспечивает функционал модуля исследования признаков на ассоциацию с ГФЭ.

Директория «DomainAdaptation» обеспечивает функционал модуля трансферного обучения для переноса аннотации ГФЭ.

Директория «Interpretation» обеспечивает функционал модуля интерпретации обученных моделей.

2.2 Связи между составными частями программы

Структура взаимодействия уровней программного обеспечения показана на рисунке 2.2.

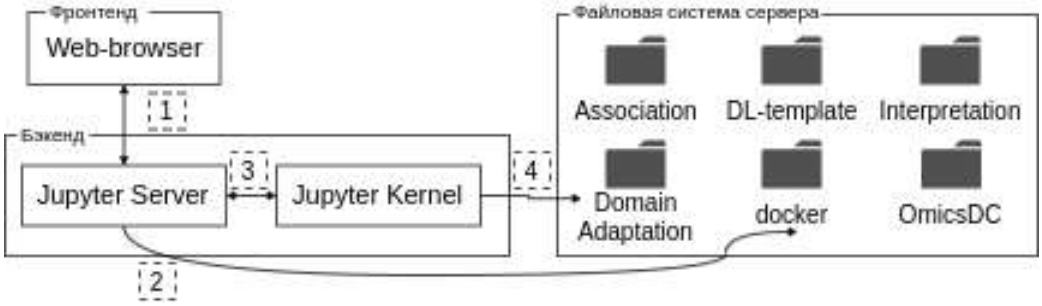

Рис.2.2 Архитектура ПО

Взаимосвязь элементов программы осуществляется за счет взаимодействия по REST-протоколу.

2.3 Основные характеристики программы

ПО предоставляет пользователю возможность редактировать запускаемый файл-шаблон.

Время восстановления после отказа, вызванного сбоем электроснабжения, сетевого оборудования, программных средств, нефатальным сбоем централизованного хранилища данных, не превышает трех суток.

Время восстановления после отказа, вызванного фатальным сбоем централизованного хранилища данных, который привел к логическому или физическому разрушению указанного хранилища, не превышает времени, необходимого на восстановление централизованного хранилища данных из резервной копии.

Время восстановления после отказа, вызванного неисправностью технических средств, не превышает времени, необходимого на устранение неисправностей или замену технических средств.

Определена следующая частота осуществления резервного копирования сервера: не менее 1 раза в неделю.

3. Настройка программы

3.1 Подготовка настройки программы

Необходимо предусмотреть наличие на сервере дистрибутива Docker (версия >= 20.10), драйверов для Nvidia GPU (версия >= 450.80.02), Nvidia Container Toolkit (версия >= 2.12).

3.2 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

- OmicsDC: папка, обеспечивающая функционал модуля загрузки данных. Включает папки:
    - OmicsDC, 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

3.3 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.

4. Проверка программы

4.1 Описание способов проверки

Проверка соответствия программы предъявляемым требования проводится путем проведения испытаний отдельных показателей с использованием документа «Программа и методика испытаний» НИУ ВШЭ.00003-01 51 на программу.

5. Дополнительные возможности

5.1 Стандарт PEP-8

Программа соответствует стандарту PEP 8 (Python Enhancement Proposal - «Предложение по усовершенствованию Python»). PEP 8 представляет собой документ, содержащий рекомендации по написанию кода на Python.

5.2 Добавление новых методов в программные модули

ПО предполагает возможность добавления новых методов в основной модуль, таких как, например, обучение нейронных сетей других типов или моделей других архитектур. Добавление осуществляется соответствующим редактированием шаблона основного модуля.

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					

Подп. и дата | Инв. N дубл. | Взам. инв. N | Подп. и дата | Инв. N подл.