

КОГНИТИВНЫЕ И СОЦИОГУМАНИТАРНЫЕ ИССЛЕДОВАНИЯ

УДК 612.821

ПРОТОТИП ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗИРОВАННОЙ ПРЕДОБРАБОТКИ ФУНКЦИОНАЛЬНЫХ МРТ ДАННЫХ

© 2022 г. В. А. Орлов^{1,*}, М. С. Орлов^{1,**}

¹ Национальный исследовательский центр “Курчатовский институт”, Москва, Россия

*E-mail: orlov_va@nrcki.ru

**E-mail: orlov.mse@yandex.ru

Поступила в редакцию 15.03.2022 г.

После доработки 20.03.2022 г.

Принята к публикации 20.03.2022 г.

Обработка данных МРТ (магнитно-резонансной томографии) концентрируется в первую очередь на борьбе с артефактами сканирования. Для этого были созданы программные и аппаратные средства, однако существующие решения не предлагают простых в использовании методов для автоматизированной обработки функциональных МРТ (фМРТ) данных. Написана программа, удовлетворяющая условиям простоты и полной автоматизации обработки фМРТ-данных.

DOI: 10.56304/S2782375X2204009X

ВВЕДЕНИЕ

В магнитно-резонансных томографах используются аппаратные средства для борьбы с артефактами сканирования, но они не удаляют последние целиком. Поэтому для дальнейшей очистки объемов применяется обработка в специализированном программном обеспечении (ПО). Вот одни из наиболее популярных программных комплексов: FMRIB Software Library (FSL) [1], Analysis of Functional NeuroImages (AFNI) [2], Statistical Parametric Mapping (SPM12) [3], FreeSurfer [4], Advanced Normalization Tools (ANTs) (<https://github.com/ANTsX/ANTs>).

Однако при работе с перечисленными пакетами возникает ряд трудностей, например в FSL, FreeSurfer, AFNI, ANTs отсутствует возможность установки и работы в операционной системе (ОС) Windows без использования вспомогательных средств, в то время как эта ОС является наиболее популярной ОС для персональных компьютеров; для работы с SPM12 требуется установка дорогостоящей программной среды Matlab, за которой следует нетривиальная работа с самим ПО, не адаптированным к русскому языку; каждая из этих программ требует длительного процесса освоения и практики. Этим списком ряд трудностей при работе с указанными выше программами не исчерпывается, однако уже критически сильно осложняет их применение.

Учитывая все перечисленное, была разработана программа, сочетающая в себе самые необхо-

димые для предобработки функциональных МРТ-данных компоненты существующих программных решений, в то же время управляемая через минималистический графический интерфейс.

В результате, чтобы перейти к обработке данных, необходимо только выбрать один из заранее составленных алгоритмов, четыре директории с соответствующими данными и запустить обработчик нажатием кнопки “Ok”. После описанных манипуляций программа в автоматическом режиме проведет обработку согласно выбранному алгоритму и в результате создаст директорию со скорректированными объемами.

МЕТОДЫ

Язык программирования, использованный для написания программы: Python 3.7. Для развертывания вычислительной части программы использовали модуль Nipype [5]. В нем обеспечена совместимость методов, принадлежащих более чем 30 популярным программным пакетам для обработки и анализа данных МРТ.

Для создания системы обработки фМРТ-данных при помощи модуля Nipype необходимо организовать систему классов “Workflow” – потоков – последовательностей, представляемых ориентированными графами, узлами которых являются классы “Node”, содержащие в себе инструкции – те самые методы обработки, заимствованные из стороннего ПО – для обработки данных, прибы-

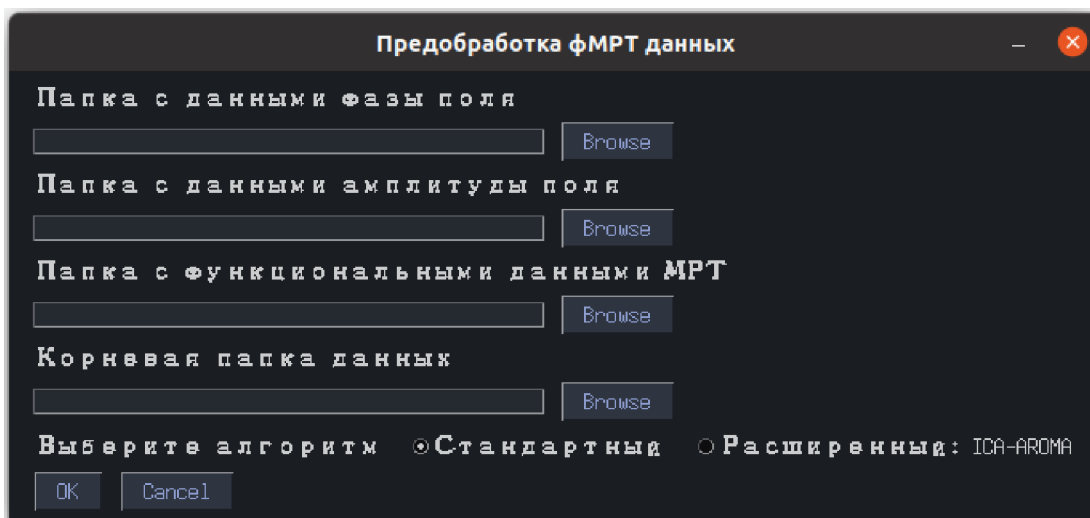


Рис. 1. Интерфейс программы. “Папка с данными фазы поля” – выбор директории с объемами распределения фазы поля в пространстве томографа; “Папка с данными амплитуды поля” – выбор директории с объемами распределения амплитуды поля в пространстве МРТ; “Папка с функциональными данными МРТ” – выбор директории с объемами функциональной активности мозга; “Корневая папка данных” – выбор директории, содержащей предыдущие директории; “Выберите алгоритм” – выбор алгоритма обработки: “стандартный” либо “с коррекцией артефактов модулем ICA-AROMA”.

вающих от предыдущего узла. В результате код программы будет представлять собой описание потоков, инструкций, соответствующих им узлов, а также связей между созданными объектами указанных выше классов.

Для повышения удобства использования программы был создан графический интерфейс (рис. 1). Реализацию последнего проводили с помощью модуля PySimpleGUI 4.60 (<https://github.com/PySimpleGUI/PySimpleGUI>).

Вычислительная часть программы представляет собой серию запрограммированных пайплайнов – последовательностей обработки данных. В актуальной версии программы представлены две версии алгоритма обработки (рис. 2).

1. Стандартный алгоритм

“Сбор данных поля” – на этом шаге проводится выбор данных, описывающих неоднородности магнитного поля, вызванные присутствием участника исследования в рабочем пространстве томографа.

“Картирование магнитного поля” – создание карты искажений магнитного поля при помощи инструмента FieldMap (источник – SPM12). В работах Джеспера [6], Дженкинсона [7], Джебзарда [8] и Хаттона [9] описываются соответственно возможность решения обратной задачи вычисления неоднородностей поля на основании эхопланарных изображений объекта, метод коррекции искажений, вносимых статическими неоднородностями магнитного поля, на основании карты

магнитного поля, метод автоматического для N-мерного развертывания карт фазы, методы для получения данных и расчета карт поля, а также последующие эффекты применения карт к временным рядам изображений. Указанные подходы используются в работе инструмента FieldMap. Алгоритм, используемый в программе, проводит преобразование карты поля в карту смещения пикселей:

$$\Delta r_{pe} = \gamma \Delta B_0(x, y, z) \cdot N (2\tau_{ramp} + N \cdot DW), \quad (1)$$

где γ – гиромагнитное соотношение, B_0 – магнитная индукция постоянного магнитного поля в томографе, N – число строк матрицы К-пространства, τ_{ramp} – время переключения градиентов, DW – время регистрации среза.

“Расчет и коррекция артефактов движения” – проводится выравнивание изображений друг относительно друга методом наименьших квадратов и шестипараметрическими преобразованиями твердого тела. Метод основан на работах [10, 11], в которых рассмотрено и описано содержание подхода по обработке данных, искаженных неизбежными перемещениями субъекта внутри томографа. Статистический анализ сигнала проводится на основании обобщенной линейной модели, заключающейся в представлении сигнала в виде линейной комбинации двух компонент: полезного сигнала, а также ошибки – искажения, вызванного движением. В результате полезная компонента выражается через исходный сигнал, а также матрицу параметров смещения, рассчитываемых для каждого изображения. В [11] для нахождения

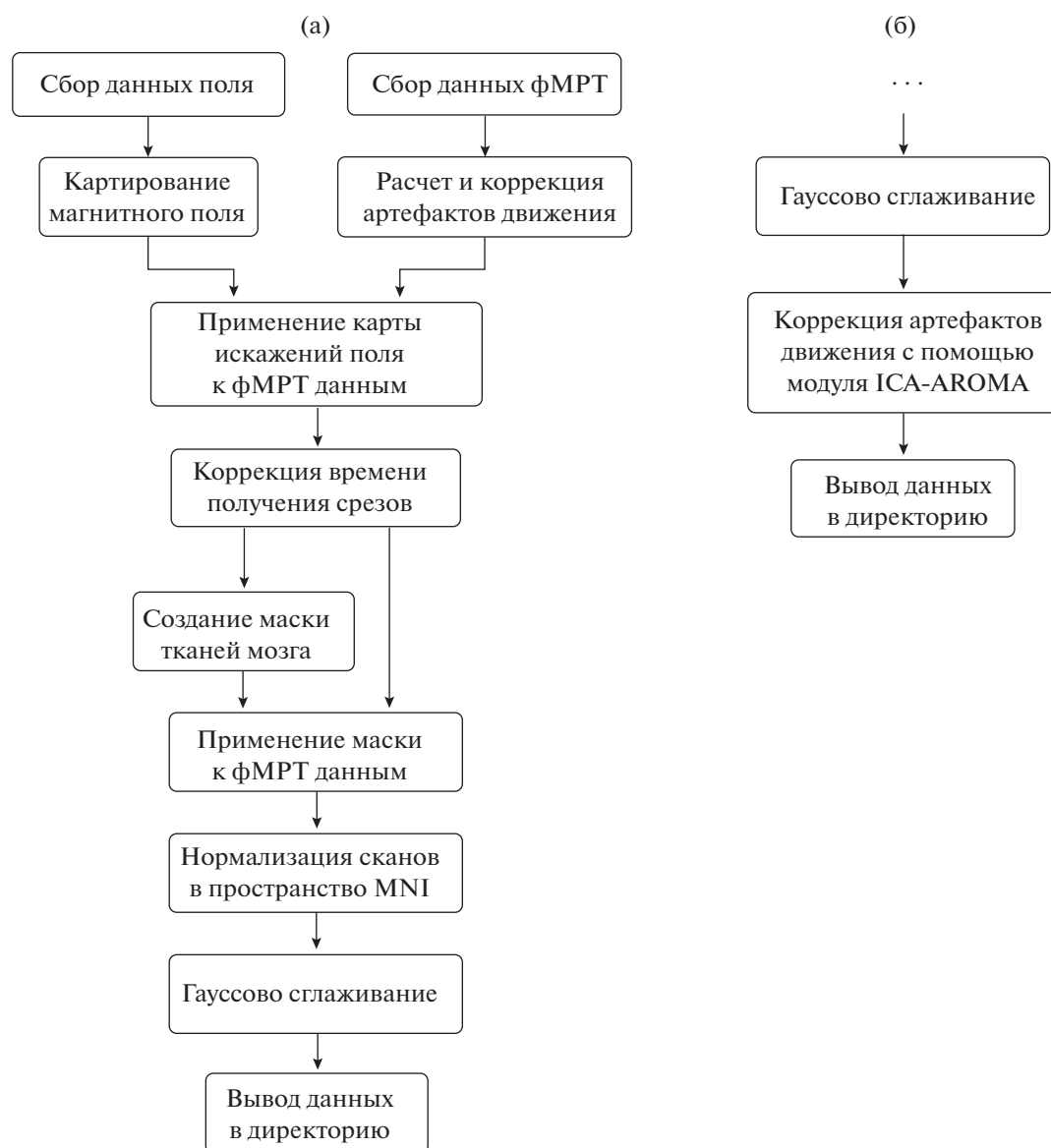


Рис. 2. Блок-схемы алгоритмов, представленных в текущей версии программы: а – стандартный алгоритм обработки; б – расширенный алгоритм обработки.

метода устранения артефактов движения сигнала фМРТ решена задача выравнивания двух изображений друг относительно друга с точностью до скалярного множителя β скалярных функций вектора расположения в пространстве $\tau(\mathbf{x})$ и $\Omega(\mathbf{x})$. Функция Ω разлагается в ряд Тейлора как

$$\Omega \approx G \cdot [b \cdot \gamma]^T, \quad (2)$$

где $G \approx \left[\tau - \frac{\partial \Omega}{\partial \gamma} \right]$, γ – вектор параметров преобразования твердого тела, а b – приближение β .

Способ вычисления γ :

$$[b \cdot \gamma]^T = (G^T \cdot G)^{-1} \cdot G^T \cdot \Omega, \quad (3)$$

чем можно перейти к коррекции артефактов движения. Далее сигнал можно описать в виде суперпозиции полезного сигнала X^* - и Q -матрицы параметров смещения γ_i , где γ – это вектор столбец параметров γ_i , вычисляющийся из уравнения (3). Можно прийти к следующему выражению для функции полезного сигнала:

$$X^* = X - Q \cdot (Q^T \cdot Q)^{-1} \cdot Q^T \cdot X. \quad (4)$$

“Применение карты искажений поля к фМРТ-данным” – выполняется перераспределение значений вокселей в данных фМРТ на основании карты стационарных искажений магнитного поля, представляющей собой матрицу про-

странственного распределения плотности потока сигнала фМРТ.

“Коррекция времени получения срезов” – сдвиг фазы в фурье-представлении сигналов соседствующих срезов данных МРТ для получения изображения в “единый момент времени” (как будто срезы были получены одномоментно).

“Создание маски тканей мозга” – применяется автоматический инструмент для сегментации скана головы субъекта на ткани мозга. Это достигается путем выполнения вычисления грубых пороговых значений интенсивности, бинаризация на их основе для определения “центра масс” мозга, грубое вычисление радиуса мозга на изображении для инициализации процесса генерации модели мозга и тесселяции модели до тех пор, пока не будет получен удовлетворительный результат [12].

“Применение маски к фМРТ данным” – выделение сегмента, соответствующего тканям мозга, применяя ранее полученную маску к обрабатываемым данным фМРТ. Этот шаг необходим для корректной нормализации объема в пространство Montreal Neurological Institute (MNI) и дальнейшего статистического анализа изображений.

“Нормализация сканов в пространство MNI” – пространственная нормализация путем сегментации и выравнивания афинными преобразованиями объемов фМРТ относительно шаблонного MNI152, полученного в Монреальском неврологическом институте.

“Гауссово сглаживание” – сглаживание полученных изображений по Гауссову ядру с шириной на полувысоте, равной 6 мм.

2. Расширенный алгоритм с использованием ICA-AROMA

В дополнение ко всем операциям из стандартного алгоритма расширенный алгоритм содержит следующий узел.

“Коррекция артефактов движения с помощью ICA-AROMA” – метод коррекции изображений, минимально влияющий на автокорреляцию данных, не требующий дополнительных настроек классификаторов коррекции, а также сохраняющий много больше степеней свободы в данных по сравнению с существующими методиками, получая тем самым преимущество перед последними. Ключевыми шагами работы ICA-AROMA [13] являются: автоматическое формирование набора независимых компонент, сверка их с критериями присутствия эффектов перемещения, регрессия выделенных компонент из общего массива данных.

В конце обработки выводятся итоговые файлы в заранее определенную директорию.

Для объективного сравнения качества данных, полученных в результате обработки программой исходных объемов, с обработанными существующими программными комплексами будем использовать методику, предложенную в [14]. Анализ при помощи этого метода позволяет выявить, какой из инструментов обработки показывает более высокую чувствительность к полезному сигналу и точность его локализации.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Испытания программы проводили на данных фМРТ, полученных с МРТ-сканера Siemens Magnetom Verio 3T (Siemens Healthcare GmbH, Германия) в НИЦ “Курчатовский институт”.

Время обработки данных эксперимента (30 срезов, размер воксела $3 \times 3 \times 4$ мм³, 120 временных динамик) 4.5 мин. Отсчет времени проводили начиная с момента запуска вычислительной части программы. Для измерений применяли метод `perf_counter()` встроенного в Python 3.7 модуля `time`. Обработку проводили на ЭВМ ASUS TUF GAMING A15, процессор – AMD Ryzen 7 4800H, ОЗУ – 16 Гб, видеокарта – NVIDIA GeForce GTX 1660 Ti, операционная система – Ubuntu 20.04 Focal Fossa.

В дальнейшем будет увеличен функционал, расширен набор доступных линий обработки данных, а также добавлена функция сохранения пользовательских конфигураций алгоритмов обработки, проведена замена блока первичной коррекции артефактов на более производительный, включена возможность выбора источника (ПО) используемых узлов для улучшения производительности на разных аппаратных платформах.

ВЫВОДЫ

Создана программа, обладающая графическим интерфейсом и выполняющая автоматическую обработку функциональных МРТ-данных. Модульность и взаимная независимость компонент проекта предоставляют широкие возможности для дальнейшей модификации и оптимизации.

В настоящее время проводятся пилотные испытания и доработка, направленные на оптимизацию ПО и обеспечение совместимости со сканирующим оборудованием более широкого круга производителей.

Внедрение универсального, компактного и в то же время простого в использовании ПО для автоматизированной предобработки фМРТ-объемов в круги клиницистов значительно упростит решение задачи обработки соответствующих данных.

СПИСОК ЛИТЕРАТУРЫ

1. <https://fsl.fmrib.ox.ac.uk>
2. <https://afni.nimh.nih.gov>
3. <https://www.fil.ion.ucl.ac.uk/spm/software/spm12>
4. <https://surfer.nmr.mgh.harvard.edu>
5. *Esteban O., Markiewicz C.J., Burns C. et al.* (2022). nipy/nipype: 1.8.3 (1.8.3). Zenodo. <https://doi.org/10.5281/zenodo.6834519>
6. *Andersson J.L., Hutton C., Ashburner J. et al.* // *NeuroImage*. 2001. V. 13. № 5. P. 903. <https://doi.org/10.1006/nimg.2001.0746>
7. *Jenkinson Mark* // *Magn. Reson. Med.* 2003. V. 49. P. 193. <https://doi.org/10.1002/mrm.10354>
8. *Jezzard P., Balaban R.S.* // *Magn. Reson. Med.* 1995. V. 34. P. 65. <https://doi.org/10.1002/mrm.1910340111>
9. *Hutton C., Bork A., Josephs O. et al.* // *NeuroImage*. 2002. V. 16. P. 217. <https://doi.org/10.1006/nimg.2001.1054>
10. *Friston K.J., Harrison L., Penny W. et al.* // *NeuroImage*. 2003. V. 19. P. 1273. [https://doi.org/10.1016/s1053-8119\(03\)00202-7](https://doi.org/10.1016/s1053-8119(03)00202-7)
11. *Friston K.J., Williams S., Howard R. et al.* // *Mag. Reson. Med.* 1996. V. 35. P. 346. <https://doi.org/10.1002/mrm.1910350312>
12. *Smith S.M.* // *Hum. Brain Mapp.* 2002. V. 17. P. 143–155. <https://doi.org/10.1002/hbm.10062>
13. *Pruim R.H.R., Mennes M., van Rooij D. et al.* // *NeuroImage*. 2015. V. 112. P. 267. <https://doi.org/10.1016/j.neuroimage.2015.02.064>
14. *Pedoia V., Colli V., Strocchi S. et al.* // *Proc. SPIE 7965, Medical Imaging 2011: Biomedical Applications in Molecular, Structural, and Functional Imaging, 796528*. 2011. <https://doi.org/10.1117/12.877067>