

# Автоматическое обнаружение и классификация апноэ/гипопноэ

Автор и название статьи: B. H. Taha, J. A. Dempsey: *Automated Detection and Classification of Sleep-Disordered Breathing From Conventional Polysomnography Data*

Репозиторий с решением:

<https://github.com/yolog59/incart>

## План:

Срок	Дата	Промежуточный результат
1 неделя	2021-10-31	Прочитаны файлы сигналов и разметки, отображены графики] (+)
2 неделя	2021-11-07	Работа с сигналом сатурации, обнаружение периодов десатурации (+)
3 неделя	2021-11-14	Работа с сигналом реопневмограммы, обнаружение минимумов вдоха
4 неделя	2021-11-21	Сопоставление сигналов и выявление событий апноэ/гипопноэ
5 неделя	2021-11-28	Сравнение с референтной разметкой, совершенствование итоговой программы

## Требования к решению (ТЗ)

Пункты проверки	Требования
<b>Используемые инструменты и внешние требования</b>	
Структура кода	Код разбит на основные модули / функции / подсистемы, их описание соответствует реализации
Оформление кода	Соблюдены правила написания кода PEP 257
[...]	[...]
<b>Входные данные и доступные ресурсы</b>	
Список баз	breath_base.
Состав записей и реф. разметок	Носовой поток, реопневмограмма, сатурация, разметка апноэ/гипопноэ
<b>Спецификация выходных данных</b>	
метки событий [...]	метки апноэ/гипопноэ
интервалы событий [...]	список полей данных
Список статистик - по файлам,	TP, FP, FN, TN, Se, Sp

- по базам	
<b>Ограничение по времени</b>	
Время обработки по одной записи	25 секунд
Время обработки по всем базам	3 минуты
<b>Ограничение точности</b>	
Описание процедуры проверки	[...]
Заявленная точность в статье:	Апноэ: Se = 73.6, Sp = 90.8 Гипопноэ: Se = 84.1, Sp = 86.1
Точность по дополнительным базам:	-
<b>Вероятные изменения</b>	
Изменения самих алгоритмов	[...]
Добавление новых алгоритмов	[...]
<b>Обработка нежелательных событий</b>	
Падение обработки на файле во время расчета статистики	
Падение на участке сигнала	[...]
Отсутствует файл для обработки	[...]
[...]	[...]

## Структура проекта

### Структура модулей

Определяем повторяющиеся элементы в сценариях выше.

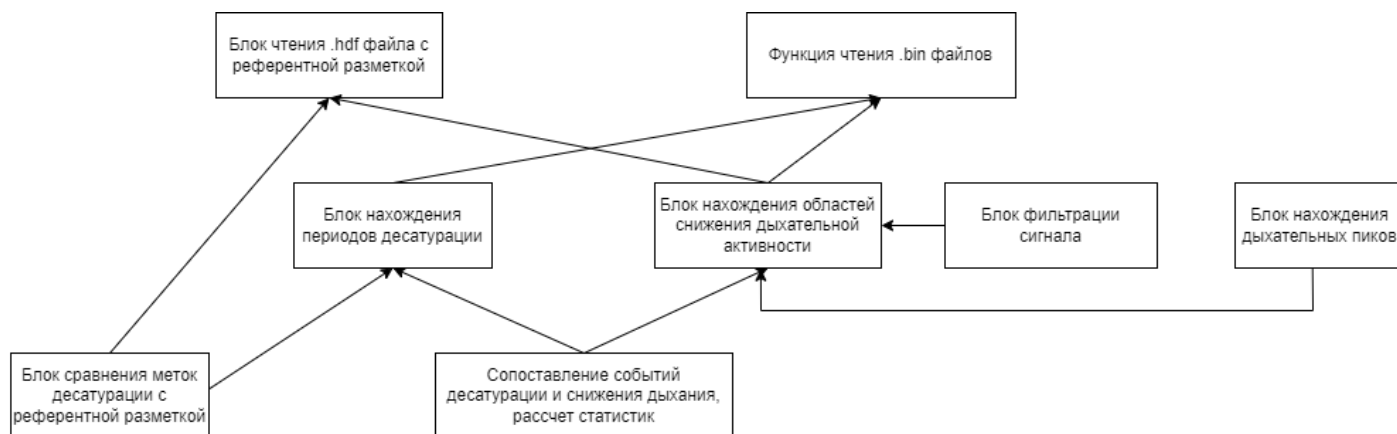
Определяем, какие данные передаются между разными элементами.

Делим эти элементы на модули.

Составляем иерархическую схему: какие модули используются в других модулях.

Желательно выделять уровни в зависимости от того, что где используется:

1. Базовые модули
2. Внутренние модули
3. Пользовательские модули (конечные)



## Описание модулей

<b>ИМЯ БЛОКА</b>	Функция чтения .bin файлов (BinFileUtils)
<b>ВХОД + НАСТРОЙКИ</b>	.bin файл с трехканальным сигналом
<b>ВЫХОД</b>	<ol style="list-style-type: none"> <li>1. Значения отсчетов сигнала</li> <li>2. Информация о сигнале (начало, конец записи, частота дискретизации)</li> </ol>
<b>ОПИСАНИЕ</b>	Блок представляет из себя функцию для чтения трехканального .bin файла и вывода всей информации.

<b>ИМЯ БЛОКА</b>	Блок чтения .hdf файла с референтной разметкой (hdfdata)
<b>ВХОД + НАСТРОЙКИ</b>	Файлы .hdf с референтной разметкой
<b>ВЫХОД</b>	<ol style="list-style-type: none"> <li>1. Частота дискретизации сигналов реопневмограммы и сатурации</li> <li>2. Время начала и конца сна</li> <li>3. Таблицы с референтной разметкой для обоих сигналов</li> </ol>
<b>ОПИСАНИЕ</b>	Блок представляет из себя функцию, которая читает файл .hdf и считывает с него все данные, а также приводит их к нужному виду (приводит отсчеты референтной разметки сатурации к секундам, а также удаляет все лишние разметки)

<b>ИМЯ БЛОКА</b>	Блок нахождения периодов десатурации (spo2)
<b>ВХОД + НАСТРОЙКИ</b>	<ol style="list-style-type: none"> <li>1. Значения отсчетов сигнала сатурации из .bin файла с помощью функции из 1 блока</li> <li>2. Частота дискретизации из .hdf файла, время начала и конца сна</li> </ol>
<b>ВЫХОД</b>	<ol style="list-style-type: none"> <li>1. Таблица с прореженными отсчетами</li> <li>2. Полученные метки начала и конца десатурации</li> </ol>
<b>ОПИСАНИЕ</b>	Сигнал, прочитанный из .bin файла, прореживается до 2 Гц. После этого все отрицательные отсчеты заменяются на "NaN". Далее идёт цикл нахождения периодов десатурации согласно статье: когда сигнал начинает уменьшаться со скоростью

	<p>более 0.1% в секунду, ставится метка 'ibeg' - начала десатурации. Уровень падения должен быть не менее 2% относительно начала. Цикл продолжается до тех пор, пока уровень сигнала не восстановится до уровня с: уровня, на котором значение сигнала будет либо не менее чем на 3% больше, чем уровень минимума (b), либо минимум на 1% меньше, чем уровень начала падения. Период десатурации должен длиться не менее 10 и не более 60 секунд - в отсчетах 2 Гц эти пределы принимают значение 20 и 120 соответственно. В таблицу 'result' записываются значения начала, минимума и конца десатурации, а также значения отсчетов в этот момент.</p>
--	--

<b>ИМЯ БЛОКА</b>	Блок фильтрации сигнала реопневмограммы (butterworth_filter)
<b>ВХОД + НАСТРОЙКИ</b>	<ol style="list-style-type: none"> <li>1. Отсчеты сигнала</li> <li>2. Верхняя и нижняя частоты среза</li> <li>3. Частота дискретизации сигнала, порядок.</li> </ol>
<b>ВЫХОД</b>	Отфильтрованные отсчеты сигнала
<b>ОПИСАНИЕ</b>	Блок представляет из себя функцию полосового фильтра Баттерворта для фильтрации сигнала реопневмограммы. Реализован с помощью библиотеки scipy, функции butter.

<b>ИМЯ БЛОКА</b>	Блок нахождения дыхательных пиков (peak_detection)
<b>ВХОД + НАСТРОЙКИ</b>	<ol style="list-style-type: none"> <li>1. Таблица с отсчетами сигнала</li> <li>2. Радиус поиска</li> </ol>
<b>ВЫХОД</b>	Функция нахождения пиков с возвращаемыми значениями минимумов и максимумов
<b>ОПИСАНИЕ</b>	Блок представляет из себя функцию для нахождения пиков в сигнале реопневмограммы. На вход подается таблица с номерами и значениями отсчетов, а также радиус поиска, на выход поступают две строки с максимумами и минимумами.

<b>ИМЯ БЛОКА</b>	Блок нахождения областей снижения дыхательной активности (geo)
<b>ВХОД + НАСТРОЙКИ</b>	<ol style="list-style-type: none"> <li>1. Отсчеты сигнала реопневмограммы</li> <li>2. Частота дискретизации из .hdf файла</li> <li>3. Время начала и конца сна.</li> </ol>
<b>ВЫХОД</b>	<ol style="list-style-type: none"> <li>1. Таблица с отсчетами начала и конца периодов снижения объема вдыхаемого воздуха, метками "apn"/"hupor"</li> <li>2. Таблица со значениями и номерами отсчетов сигнала.</li> </ol>
<b>ОПИСАНИЕ</b>	Прочитанный сигнал прореживается до частоты из .hdf файла, после чего к нему применяется функция фильтрации и функция нахождения пиков. Далее идет вычисление размахов каждого дыхательного цикла: из максимума вычитается среднее из двух минимумов. Далее идет сравнение размахов согласно статье: если размах цикла составляет менее 80% от предыдущего, ставится метка 'ibeg'. Событие должно длиться не менее 10 и не более 180 секунд и может закончиться, когда размах восстановится до уровня не менее 90% от начального размаха. Для апноэ: если длительность между двумя пиками больше, чем 10 секунд, событию ставится

	метка апноэ (apn). В 1 случае ставится метка гипапноэ (hypor). Поиск апноэ также проводится внутри цикла с поиском гипапноэ. Если период длится более 180 секунд, он не учитывается, цикл продолжается со следующего дыхательного цикла.
--	--

<b>ИМЯ БЛОКА</b>	Блок сравнения меток десатурации с референтной разметкой (stats_spo2)
<b>ВХОД + НАСТРОЙКИ</b>	<ol style="list-style-type: none"> <li>1. Референтная разметка из .hdf файла с частотой и началом/концом событий по сигналу сатурации</li> <li>2. Полученные таблицы отсчетов сигнала и значений начала/конца событий десатурации</li> <li>3. Путь к файлам</li> <li>4. Файл с именами файлов</li> </ol>
<b>ВЫХОД</b>	<ol style="list-style-type: none"> <li>1. Статистики точности, чувствительности и специфичности по каждой записи</li> <li>2. Время обработки всех записей</li> </ol>
<b>ОПИСАНИЕ</b>	<p>Программа через цикл перебирает все записи и считает статистики следующим образом:</p> <ol style="list-style-type: none"> <li>1) Номерам отсчетов сигнала присваиваются 0 и 1 в зависимости от наличия события. Присвоение идет по тестовой и референтной разметкам</li> <li>2) Далее идёт сравнение: если в референтном отсчете 1 и в тестовом тоже - ставится метка TP, если 1 и 0 соответственно - FP, 0 и 1 - FN, 0 и 0 - TN.</li> <li>3) Считается количество всех меток и рассчитываются статистики точности, чувствительности и специфичности по формулам.</li> </ol>

<b>ИМЯ БЛОКА</b>	Сопоставление событий десатурации и снижения дыхания, расчет статистик (results)
<b>ВХОД + НАСТРОЙКИ</b>	<ol style="list-style-type: none"> <li>1. Референтная разметка из .hdf файла с частотой и началом/концом событий по сигналу реопневмограммы</li> <li>2. Полученные таблицы отсчетов сигнала и значений начала/конца событий десатурации в 1 сигнале и начала/конца событий снижения дыхательной активности во 2 сигнале</li> <li>3. Путь к файлам</li> <li>4. Файл с именами файлов</li> </ol>
<b>ВЫХОД</b>	<ol style="list-style-type: none"> <li>1. Статистики точности, чувствительности и специфичности по каждой записи</li> <li>2. Время обработки всех записей</li> </ol>
<b>ОПИСАНИЕ</b>	<p>В данном блоке все записи также перебираются через цикл, идёт сопоставление событий снижения дыхательной активности с десатурацией: если от начала события снижения дыхательной активности наблюдается десатурация не более, чем через 40 секунд, событию присваивается "+". Далее метки всех событий без "+" удаляются и оставшиеся метки сравниваются с референтной разметкой описанным в предыдущем блоке способом. Разделения на апноэ/гипапноэ пока нет, сравнение идет по наличию события в принципе. На выходе представляется таблица с результатом обработки по всем записям, а также время работы программы.</p>

# Инструкция

Код всех блоков реализован на языке Python. Для запуска понадобится установка следующих библиотек: numpy, pandas, matplotlib, scipy, h5py.

Для запуска необходимо, чтобы у всех трех файлов (.bin, .hdr, .hdf) по каждой записи было одно название. Также у файла с именами всех записей должно быть имя "files.txt". Для удобства я переименовал все файлы с референтной разметкой и выгрузил всё на ЯД, включая файл со списком записей: [https://disk.yandex.ru/d/IEksBsNea\\_eP1w](https://disk.yandex.ru/d/IEksBsNea_eP1w)

Файл stats\_spo2.py рассчитывает статистики по сигналам сатурации, после его запуска в переменную foldpath нужно ввести путь к папке (с двойными бекслешами), в которой лежат все файлы, включая файл со списком записей. Далее программу можно сразу запускать, её работа будет длиться примерно 120 секунд для 7 записей, после чего будет выведена таблица со статистиками по каждой записи, которая также будет сохранена в формате .csv в исходную папку. В терминале также будет выведено время работы программы.

Файл results.py рассчитывает статистики по сигналам реопневмограммы и работает аналогично предыдущей, все результаты также будут выведены в терминале и сохранены в исходной папке в формате .csv.