

Описание request-файлов openModeller

[Обсудить в форуме](#) Комментариев — 0

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/openmodeller-request.html>

Чудесной девушке Ане, которая вдохновила меня на написание статьи.

Изучаем структуру request-файлов openModeller.

В статье, посвященной [приложениям openModeller](#), неоднократно упоминался request-файл. При использовании openModeller в командной строке сталкиваться с этими файлами приходится очень часто, с их помощью можно выполнить практически все необходимые действия и пройти путь от исходных данных до результатов моделирования.

Содержание

- [1 Общие сведения](#)
- [2 Описание точек встречи](#)
- [3 Описание слоёв \(создание модели\)](#)
- [4 Итоговая модель](#)
- [5 Настройки процесса применения модели](#)
- [6 Описание слоёв \(применение модели\)](#)
- [7 Результат моделирования](#)
- [8 Параметры алгоритма](#)
- [9 Заключение](#)
- [10 Ссылки по теме](#)

Общие сведения

Request-файл является обычным текстовым файлом, в котором содержатся пары КЛЮЧ = ЗНАЧЕНИЕ, разделенные одним или несколькими переводами строки. Request-файлы часто выступают в роли исходных данных для разных консольных приложений openModeller. Например, руководствуясь указаниями request-файла `om_console` может: создать модель; применить существующую модель к набору слоёв; создать модель и сразу же применить её к набору слоёв. С request-файлами также работают `om_viewer` и `om_sampler`.

Хотя строгой структуры request-файл не имеет, логически его можно разделить на несколько секций:

- описание точек встречи
- описание исходных слоёв, используемых при создании модели
- описание итоговой модели
- настройки процесса применения модели
- описание слоёв, к которым будет применяться модель
- описание результатов применения модели
- параметры алгоритма

Строки начинающиеся с символа «#» (решетка) являются комментариями и игнорируются. Также игнорируются пустые строки.

В зависимости от задач, файл может содержать как все из перечисленных секций, так и только некоторые из них. Рассмотрим каждую секцию подробнее.

Описание точек встречи

Как явствует из названия, это параметры тем или иным образом связанные с точками встречи вида. В группу входят следующие параметры:

- **WKT Coord System** — система координат точек встречи в формате WKT (Well-Known Text). Если точки находятся в разных системах координат, необходимо преобразовать их в одну общую
- **Occurrences source** — указывает на источник данных по точкам встречи. В качестве источника может использоваться абсолютный или относительный путь к текстовому файлу; абсолютный или относительный путь к файлу XML; адрес сервиса GBIF; адрес сервиса TAPIR. **Примечание:** необходимо помнить, что `om_console` может работать только с одним источником точек одновременно
- **Occurrences group** — метка, по которой будет выполняться отбор точек из источника (чаще всего это название вида). По умолчанию последнее найденное в списке точек значение
- **Spatially unique** — необязательный параметр при помощи которого можно отсеять точки с одинаковыми координатами. По умолчанию `false`
- **Environmentally unique** — необязательный параметр при помощи которого можно отсеять точки с одинаковыми экологическими и/или климатическими условиями. По умолчанию `false`

Все эти параметры используются только при создании новой модели. Вот пример:

```
WKT Coord System = GEOGCS"WGS84", DATUM"WGS84", SPHEROID"WGS84", 6378137.0,
298.257223563, PRIMEM"Greenwich", 0.0, UNIT"degree", 0.017453292519943295,
AXIS"Longitude",EAST, AXIS"Latitude",NORTH
Occurrences source = /home/alex/data/gbif/physalis_peruviana.txt
Occurrences group = Physalis peruviana
Spatially unique = true
Environmentally unique = false
```

Описание слоёв (создание модели)

Здесь описываются слои, используемые при создании модели.

- **Map** — растровый слой с непрерывными значениями, который будет использоваться для создания модели. Допускается использование нескольких параметров. В качестве значения можно использовать абсолютный или относительный путь к GDAL-совместимому растру или, если `openModeller` собран с поддержкой `TerraLib`, ссылку на растр `TerraLib` в виде

```
terralib>yourusername>yourpassword@PostgreSQL>localhost>terralib>5432>rain_coollest
```

- **Categorical map** — растровый слой с дискретными значениями, который будет использоваться для создания модели. Допускается использование нескольких параметров. В качестве значения можно использовать абсолютный или относительный путь к GDAL-совместимому растру или, если `openModeller` собран с поддержкой `TerraLib`, ссылку на растр `TerraLib` в виде

```
terralib>yourusername>yourpassword@PostgreSQL>localhost>terralib>5432>rain_coollest
```

- **Mask** — растровый слой маски, задающий область интереса (не обязательный параметр). На практике это значит, что все точки, попавшие за границы этого слоя, будут исключены из анализа. Если параметр не задан, в качестве слоя маски используется первый слой, заданный параметром `Map`. **Примечание:** слои, заданные параметрами `Map` и `Categorical map` тоже могут исключать точки из анализа, если эти точки попадают в места со значением «`nodata`».

Слой маски должен поддерживать установку значений «`nodata`», попадание точки в область со значением 0 (ноль) не приводит к ее исключению из анализа.

В качестве значения можно использовать абсолютный или относительный путь к GDAL-совместимому растру или, если `openModeller` собран с поддержкой `TerraLib`, ссылку на растр `TerraLib` в виде

```
terralib>yourusername>yourpassword@PostgreSQL>localhost>terralib>5432>south_america
```

Пример

```
Map = /data/cru_bioclim/meanDiurnal.asc
Map = /data/cru_bioclim/meanDiurnalOverCoolestM.asc
Map = /data/cru_bioclim/meanDiurnalOverWarmestM.asc
Map = /data/cru_bioclim/meanFrostDays.asc
Map = /data/cru_bioclim/meanPrecip.asc
Map = /data/cru_bioclim/meanPrecipOverCoolestQ.asc
Map = /data/cru_bioclim/meanPrecipOverDriestM.asc
Map = /data/cru_bioclim/meanPrecipOverDriestQ.asc
Map = /data/cru_bioclim/meanPrecipOverFrostFreeM.asc
```

```
Mask = /data/cru_bioclim/stdevMeanTemp.asc
```

Итоговая модель

Результатом анализа является модель, которая может быть сохранена на диск в формате XML (так называемая «сохраненная модель»). Если файл не планируется использовать для создания модели, все параметры этой группы должны быть опущены.

- **Output model** — абсолютный или относительный путь к файлу, в который будет записана полученная в результате анализа модель

```
Output model = /home/alex/models/my_model.xml
```

Настройки процесса применения модели

Ранее созданная при помощи `om_console` или `om_model` модель может использоваться в качестве исходной и применяться к некоторому набору слоёв. В этом случае все параметры, относящиеся к точкам встречи и исходным слоям будут проигнорированы.

- **Input model** — абсолютный или относительный путь к файлу сохраненной модели

```
Input model = /home/alex/models/test_model.xml
```

Описание слоёв (применение модели)

Эта группа параметров необходима только если выполняется применение модели к другому набору слоёв.

Примечание: необходимо помнить, что число слоёв, их порядок и величины, которые они представляют, должны соответствовать слоям, использованным при создании модели. Например, если модель строилась с использованием трех слоев (данные о температуре, влажности и кислотности почвы), то и второй набор должен состоять из трех слоёв, описывающих те же величины (температуру, влажность и кислотность почвы).

- **Output map** — растровый слой с непрерывными значениями, который будет использоваться при применении модели. Допускается использование нескольких параметров. В качестве значения можно использовать абсолютный или относительный путь к GDAL-совместимому растру или, если openModeller собран с поддержкой TerraLib, ссылку на растр TerraLib в виде

```
terralib>yourusername>yourpassword@PostgreSQL>localhost>terralib>5432>rain_coollest
```

- **Output categorical map** — растровый слой с дискретными значениями, который будет использоваться при применении модели. Допускается использование нескольких параметров. В качестве значения можно использовать абсолютный или относительный путь к GDAL-совместимому растру или, если openModeller собран с поддержкой TerraLib, ссылку на растр TerraLib в виде

```
terralib>yourusername>yourpassword@PostgreSQL>localhost>terralib>5432>rain_coollest
```

- **Output mask** — растровый слой маски, задающий область интереса (не обязательный параметр). На практике это значит, что все точки, попавшие за границы этого слоя, будут исключены из анализа. Если параметр не задан, в качестве слоя маски используется первый слой, заданный параметром Output map. **Примечание:** слои, заданные параметрами Output map и Output categorical map тоже могут исключать точки из анализа, если эти точки попадают в места со значением «nodata».

Слой маски должен поддерживать установку значений «nodata», попадание точки в область со значением 0 (ноль) не приводит к ее исключению из анализа.

В качестве значения можно использовать абсолютный или относительный путь к GDAL-совместимому растру или, если openModeller собран с поддержкой TerraLib, ссылку на растр TerraLib в виде

```
terralib>yourusername>yourpassword@PostgreSQL>localhost>terralib>5432>south_america
```

Пример:

```
Output map = /data/clim_2050/meanDiurnal.asc
Output map = /data/clim_2050/meanDiurnalOverCoolestM.asc
Output map = /data/clim_2050/meanDiurnalOverWarmestM.asc
Output map = /data/clim_2050/meanFrostDays.asc
Output map = /data/clim_2050/meanPrecip.asc
Output map = /data/clim_2050/meanPrecipOverCoolestQ.asc
Output map = /data/clim_2050/meanPrecipOverDriestM.asc
Output map = /data/clim_2050/meanPrecipOverDriestQ.asc
Output map = /data/clim_2050/meanPrecipOverFrostFreeM.asc

Output mask = /data/clim_2050/stddevMeanTemp.asc
```

Результат моделирования

Результатом моделирования (применения модели) является растровый файл, показывающий распределение вероятности в заданной области. Охват этой области определяется пересечением охватов всех слоёв, заданных как Output map и слоя маски. Система координат результирующего растра а также размер пикселя будут такими же, как и у растра-«шаблона». «Шаблон» задается параметром Output format.

- **Output file type** — задает тип итогового растра. На момент написания статьи поддерживались следующие типы:
 - GreyTiff — GeoTiff, оттенки серого, диапазон допустимых значений 0 - 254, «nodata» — 255
 - GreyTiff100 — GeoTiff, оттенки серого, диапазон допустимых значений 0 - 100, «nodata» — 127
 - FloatingTiff — GeoTiff, тип данных — десятичное число, диапазон допустимых значений 0.0 - 1.0, «nodata» — -1
 - GreyBMP — BMP, оттенки серого, диапазон допустимых значений 0 - 255, «nodata» — 0
 - FloatingHFA — Erdas Imagine, тип данных — десятичное число, диапазон допустимых значений 0.0 - 1.0, «nodata» — -1
 - ByteHFA — Erdas Imagine, диапазон допустимых значений 0 - 100, «nodata» — 101
 - ByteASC — ARC/Info ASCII grid, диапазон допустимых значений 0 - 100, «nodata» — 101
 - FloatingASC — ARC/Info ASCII grid, тип данных — десятичное число, диапазон допустимых значений 0.0 - 1.0, «nodata» — -9999
- **Output format** — растр-«шаблон», задающий систему координат и размер пикселя для итогового файла
- **Output file** — абсолютный или относительный путь к файлу, в который будет записан результат применения модели

```
Output format = /data/clim_2050/stddevMeanTemp.asc
Output file type = ByteHFA
Output file = /data/output/acacia-2050.img
```

Параметры алгоритма

Если параметры этой группы не заданы, om_console переходит в интерактивный режим, выводит список всех имеющихся алгоритмов и просит указать нужный. Аналогично будет выполняться и установка параметров алгоритма. Если же алгоритм и его параметры заданы в request-файле, то om_console будет работать в неинтерактивном режиме и обработает файл без вмешательства пользователя.

Параметры этой группы необходимы только при создании модели.

- **Algorithm** — идентификатор алгоритма. Если не задан, обработка файла приостанавливается и идентификатор запрашивается у пользователя. Получить список доступных алгоритмов и их идентификаторы можно при помощи [om_algorithm](#)
- **Parameter** — настройки алгоритма, заданного ключом Algorithm. Допускается использование нескольких ключей. Формат записи: «идентификатор_параметра значение_параметра».

Если все или часть параметров не заданы, om_console приостанавливает работу и запрашивает значение

отсутствующих параметров у пользователя

Например

```
Algorithm = CSMBS
Parameter = Randomisations 8
Parameter = StandardDeviations 2
Parameter = MinComponents 1
Parameter = VerboseDebugging 1
```

Полный request-файл будет иметь примерно такой вид

```
# occurrences
WKT Coord System = GEOGCS"WGS84", DATUM"WGS84", SPHEROID"WGS84", 6378137.0,
298.257223563, PRIMEM"Greenwich", 0.0, UNIT"degree", 0.017453292519943295,
AXIS"Longitude",EAST, AXIS"Latitude",NORTH
Occurrences source = /home/alex/data/gbif/physalis_peruviana.txt
Occurrences group = Physalis peruviana
Spatially unique = true
Environmentally unique = false

# input layers
Map = /data/cru_bioclim/meanDiurnal.asc
Map = /data/cru_bioclim/meanDiurnalOverCoolestM.asc
Map = /data/cru_bioclim/meanDiurnalOverWarmestM.asc
Map = /data/cru_bioclim/meanFrostDays.asc
Map = /data/cru_bioclim/meanPrecip.asc
Map = /data/cru_bioclim/meanPrecipOverCoolestQ.asc
Map = /data/cru_bioclim/meanPrecipOverDriestM.asc
Map = /data/cru_bioclim/meanPrecipOverDriestQ.asc
Map = /data/cru_bioclim/meanPrecipOverFrostFreeM.asc

Mask = /data/cru_bioclim/stdevMeanTemp.asc

# save model as
Output model = /home/alex/models/my_model.xml

# apply model using this layers
Output map = /data/clim_2050/meanDiurnal.asc
Output map = /data/clim_2050/meanDiurnalOverCoolestM.asc
Output map = /data/clim_2050/meanDiurnalOverWarmestM.asc
Output map = /data/clim_2050/meanFrostDays.asc
Output map = /data/clim_2050/meanPrecip.asc
Output map = /data/clim_2050/meanPrecipOverCoolestQ.asc
Output map = /data/clim_2050/meanPrecipOverDriestM.asc
Output map = /data/clim_2050/meanPrecipOverDriestQ.asc
Output map = /data/clim_2050/meanPrecipOverFrostFreeM.asc

Output mask = /data/clim_2050/stdevMeanTemp.asc

# result
Output format = /data/clim_2050/stdevMeanTemp.asc
Output file type = ByteHFA
Output file = /data/output/acacia-2050.img

# algorithm to use
Algorithm = SVM
Parameter = SvmType 0
Parameter = KernelType 2
Parameter = Degree 3
Parameter = Gamma 0
Parameter = C 1
Parameter = Coef0 0
Parameter = Nu 0.5
```

```
Parameter = ProbabilisticOutput 0  
Parameter = NumberOfPseudoAbsences 500
```

Заключение

Как видим, ничего сложного в request-файлах нет. Простой текстовый формат позволяет легко формировать такие файлы программно, а затем использовать их для обработки данных.

Ссылки по теме

1. [openModeller Home](#)
2. [Начало работы с openModeller](#)
3. [Консольные приложения openModeller](#)
4. [Пример использования openModeller Desktop для новых условий](#)
5. [Глобальная база данных по биоразнообразию — GBIF](#)

[Обсудить в форуме](#) Комментариев — 0

Последнее обновление: 2014-05-15 00:07

Дата создания: 26.02.2012

Автор(ы): [Александр Бруй](#)