Краткое введение в MaxEnt

Обсудить в форуме Комментариев — 4

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

Автор: Стивен Филлипс (Steven Phillips), AT&T Research.
Оригинал: http://www.cs.princeton.edu/~schapire/maxent/
Перевод: Максим Дубинин, Юлия Калашникова (с изменениями редакторов).
Translation: Maxim Dubinin, Yulia Kalashnikova (with edits).

Это руководство представляет из себя краткое введение в использование программы MaxEnt, предназначенной для моделирования географического распространения биологических видов методом максимальной энтропии. Авторы руководства — Стивен Филлипс (Steven Phillips), Миро Дудик (Miro Dudik) и Роб Шапир (Rob Schapire), благодаря поддержке AT&T Labs-Research, Университета Принстона и Центра биоразнообразия и охраны природы Американского музея естественной истории (Center for Biodiversity and Conservation, American Museum of Natural History). Более подробное описание теории энтропийного моделирования и описание использованных наборов данных и типов статистического анализ можно найти в следующих статьях:

Steven J. Phillips, Robert P. Anderson and Robert E. Schapire, Maximum entropy modeling of species geographic distributions. Ecological Modeling, Vol 190/3-4 pp 231-259, 2006.

Вторая статья, описывающая относительно новую функциональность Maxent:

Steven J. Phillips and Miroslav Dudik, Modeling of species distributions with Maxent: new extensions and a comprehensive evaluation. Ecography, Vol 31, pp 161-175, 2008.

Используемые данные о параметрах окружающей среды включают климатические и высотные данные по Южной Америке, а также слой потенциальной растительности. Моделируемый вид – Бурогорлый ленивец (*Bradypus variegates*). Это руководство подразумевает, что все учебные данные находятся в той же папке, что и сам Maxent; если это не так – добавляйте свой путь (например, `c:\data\maxent\tutorial`) к используемым здесь названиям файлов.

Содержание

Словарь

- feature — числовой признак, являющийся функцией входных данных (градиентов среды) или, другими словами, функция градиента среды
- product feature — произведение числовых признаков
- hinge feature — нелинейный числовой признак
- threshold feature - пороговый числовой признак
- presence — присутствие
- absence — отсутствие
- response curve — кривая зависимости (кривая отклика)
- predicted suitability — пригодность местообитаний
- threshold features — пороговые числовые характеристики
- step function — ступенчатая функция
- piece-wise linear function — кусочно-линейная функция
- sigmoid function — сигмоидная функция
- additive function — аддитивная функция
- Maxent exponent — экспонента Maxent
- overfitting — переобучение (излишнее обучение) модели
- clamping — слияние
- boostrapping - методы рандомизации
- gain - прирост

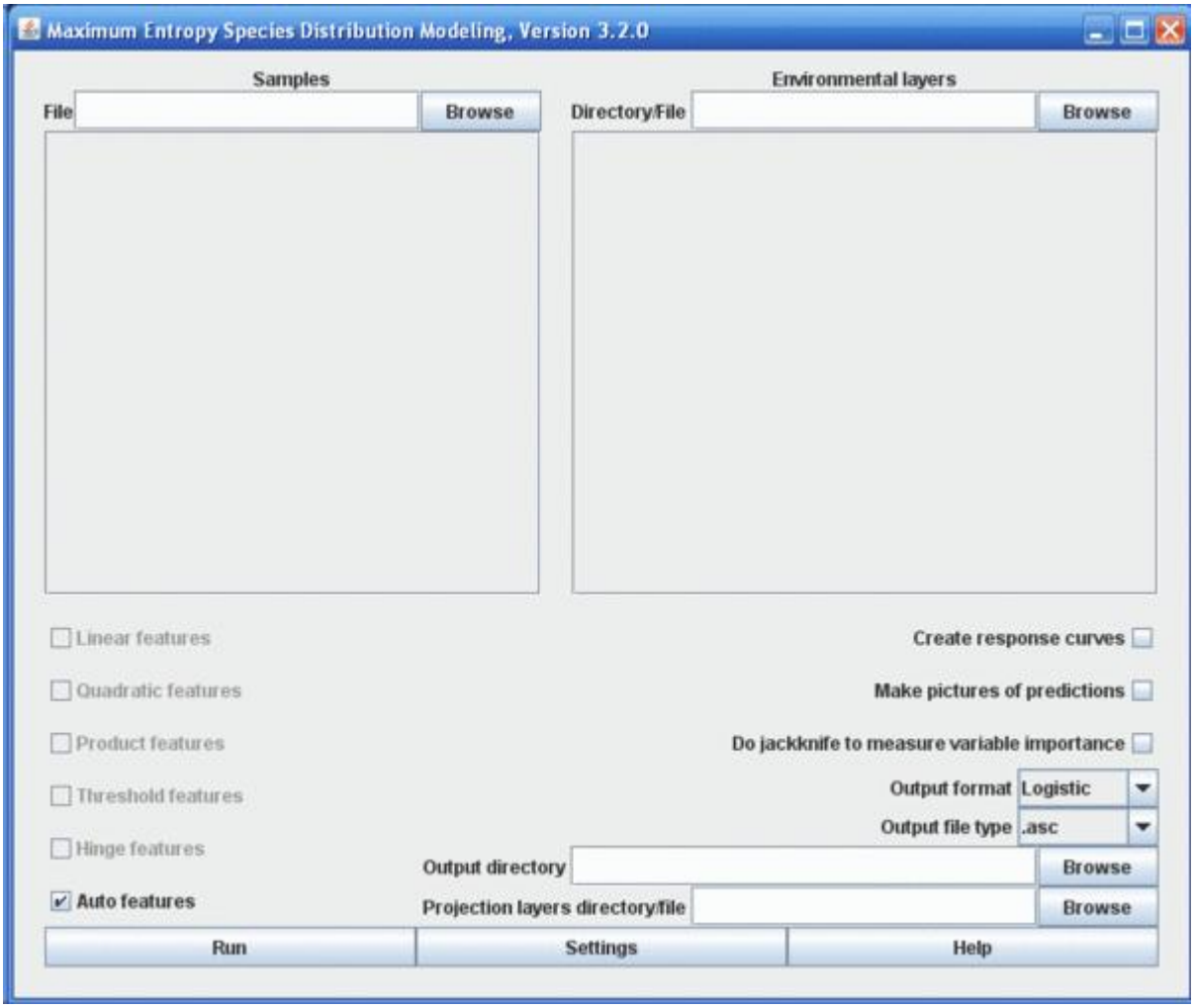
Начало

Загрузка

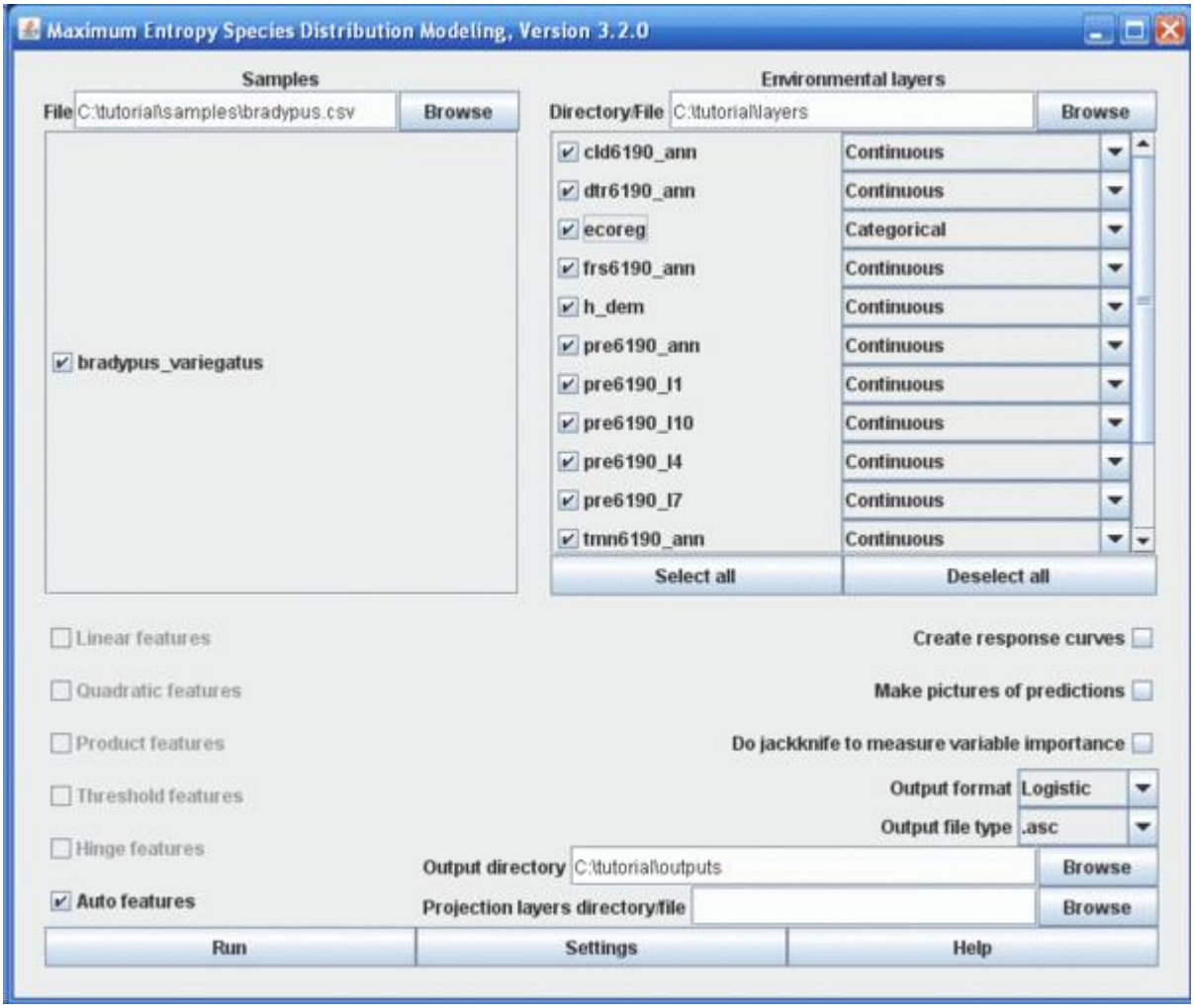
Программное обеспечение включает файл **maxent.jar**, который может быть запущен на любом компьютере, на котором есть Java версии 1.4 или выше. Сам Maxent и документацию можно загрузить по адресу http://www.cs.princeton.edu/~schapire/maxent. Среду выполнения Java можно получить по адресу http://www.oracle.com/technetwork/java/javase/downloads/index.html. Если вы используете Microsoft Windows (как здесь подразумевается), то нужно также загрузить файл **maxent.bat** и сохранить его в той же папке, где хранится **maxent.jar**. На веб-сайте есть файл “readme.txt”, где содержатся инструкции по установке программы.

Запуск

Если вы используете Microsoft Windows, просто щёлкните по файлу **maxent.bat**. Если у вас другая операционная система, введите в командной строке "java -mx512m -jar maxent.jar" (где "512" можно заменить на количество мегабайт памяти, которое вы хотите выделить для программы). Появится такое окно:



Чтобы запустить процесс, нужно предоставить файл содержащий точки находок биологического вида (образцы,“samples”), папку, содержащую слои с параметрами окружающей среды, и выходную папку. В нашем случае точки встреч находятся в файле “samples\bradypus.csv”, слои параметров среды в папке “layers” и выходные результаты будут сохраняться в папке “outputs”. Вы можете вводить эти значения вручную или использовать менеджер файлов. Когда вы ищете переменные среды, помните, что нужна папка, которая их содержит, а не сами файлы. После ввода необходимых параметров окно программы должно выглядеть следующим образом:



Файл “samples\bradypus.csv” содержит находки в формате CSV. Первые строки файла выглядят следующим образом:

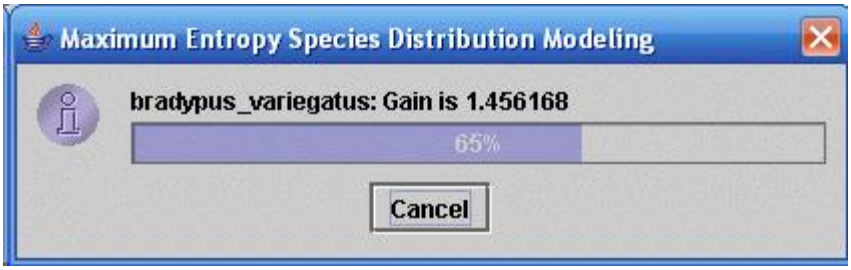
```
species,longitude,latitude
bradypus_variegatus,-65.4,-10.3833
bradypus_variegatus,-65.3833,-10.3833
bradypus_variegatus,-65.1333,-16.8
bradypus_variegatus,-63.6667,-17.45
bradypus_variegatus,-63.85,-17.4
```

В файле находок может быть несколько видов, в этом случае будет сгенерировано больше результатов, кроме *Bradypus*. Координаты находок могут находиться в системах координат, отличных от географической ("широта/долгота"), но в них должны быть и находки, и слои данных. В файле находок координата X (в нашем случае долгота) должна располагаться перед координатой Y (широта). Если в данных есть дубликаты (несколько записей для одного и того же вида в одной и той же ячейке), то по умолчанию они удаляются; это поведение можно отключить, нажав, на кнопку “Settings” и отключив “Remove duplicate presence records”.

Папка “layers” содержит растровые данные в формате Arc/Info ASCII Grid, каждый из которых описывает определенный параметр окружающей среды. Все растры должны иметь одинаковый географический охват и размер ячейки (т.е. заголовки файлов должны быть строго одинаковыми). Одна из наших переменных “ecoreg” — категорийная, она указывает класс потенциальной растительности. Категории должны быть указаны цифрами, а не буквами или словами. Необходимо указать программе, какие из переменных являются категорийными, так, как показано на иллюстрации выше.

Запуск модели

Просто нажмите кнопку “Run”. Появится индикатор прогресса, описывающий текущие шаги, выполняемые программой. После загрузки слоёв и инициализации прогресс тренировки модели максимальной энтропии показывается так:

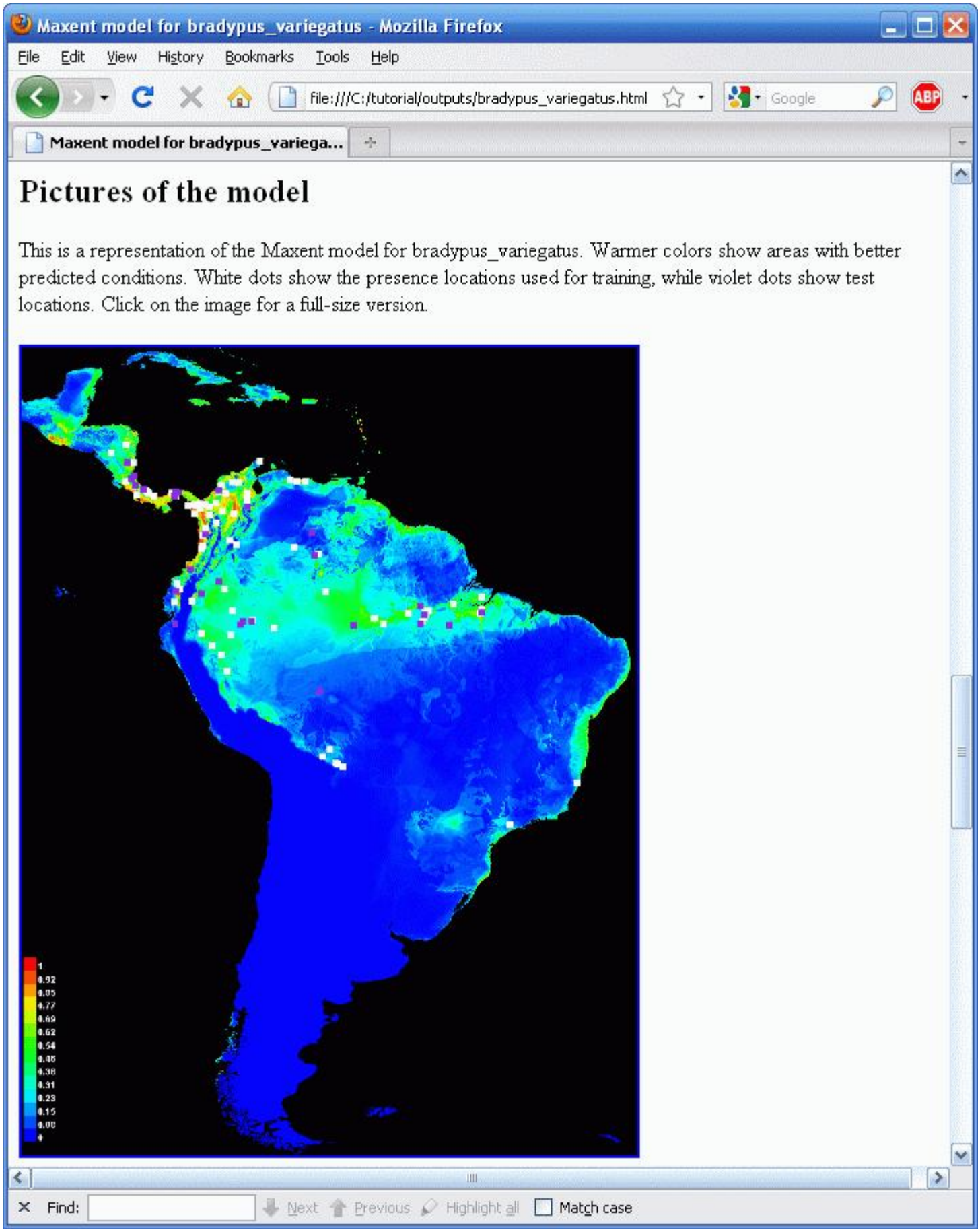


Прирост тесно связан с Deviance, которая является мерой качества модели (статистическим критерием), используемой в обобщённых аддитивных и линейных моделях. Прирост начинается с нуля и асимптотически растёт в процессе расчёта. При расчёте Maxent генерирует распределение вероятности ячеек растра, начиная с равномерного распределения, и постепенно улучшает соответствие модели данным. Прирост определяется как средняя логарифмическая вероятность образцов присутствия минус константа, которая делает равным нулю прирост равномерного распределения. В конце прохода прирост показывает, насколько сильно модель сконцентрирована вокруг образцов присутствия. Например, если прирост равен 2, это означает, что среднее правдоподобие образцов присутствия в exp(2) (≈ 7.4) раз выше, чем у случайных ячеек фона. Отметьте, что Maxent не рассчитывает “вероятность присутствия” напрямую. Вероятность, которую Maxent назначает каждой ячейке, обычно очень мала, так как значения всех ячеек растра должны в сумме быть равны единице (мы вернёмся к этому моменту, когда будем сравнивать выходные форматы).

После запуска модели будет создано несколько выходных файлов, основным из которых является “bradypus.html”. В конце этого файла также содержатся ссылки на другие результаты:

Просмотр предсказания

По умолчанию выходные результаты в виде HTML-страницы содержат графический результат модели, применённой к заданным параметрам окружающей среды:



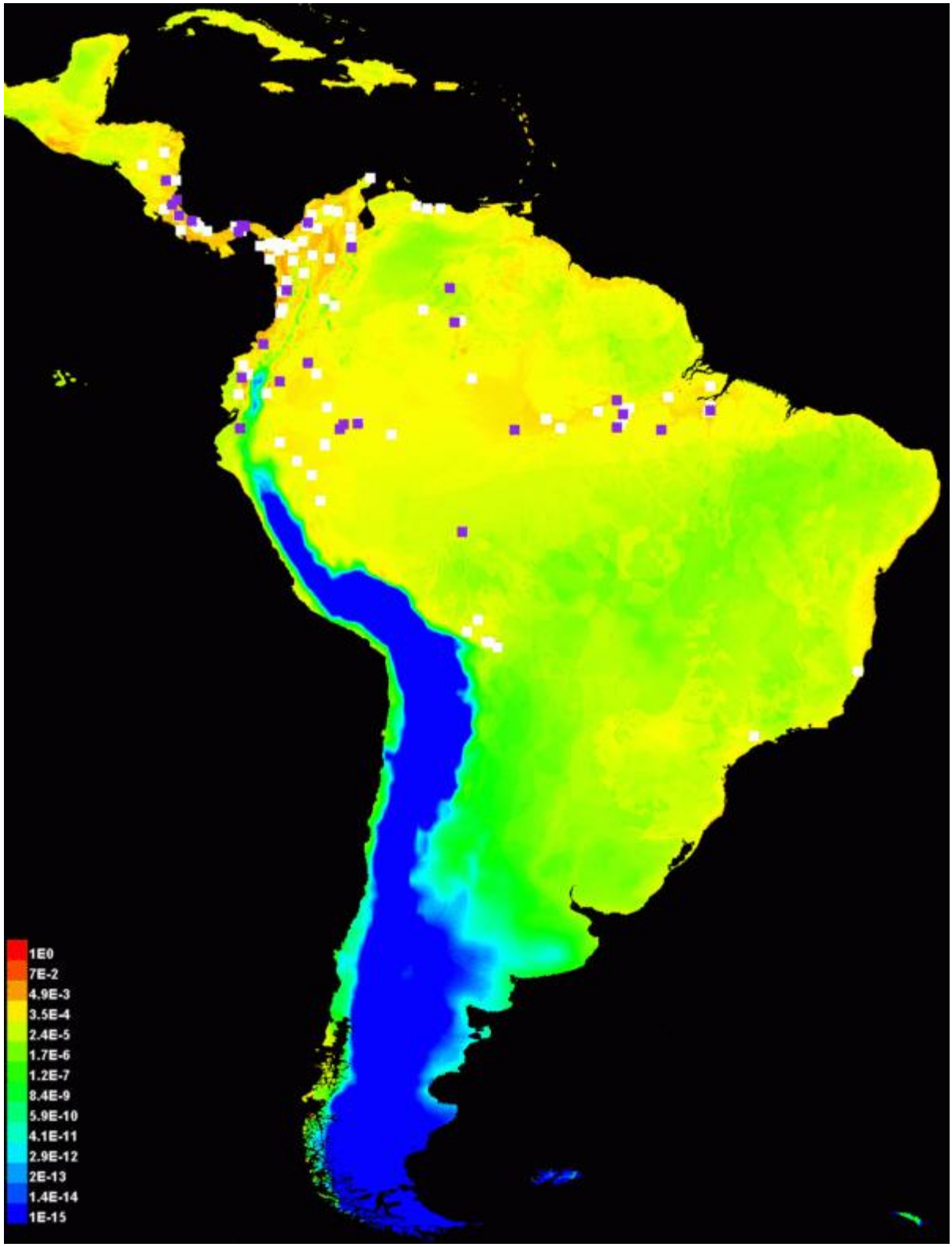
Результирующее изображение цветами показывает расчётную вероятность того, что условия для находки подходящие: красным показывается высокая вероятность подходящих условий для вида; зелёным — условия, похожие на те, в которых находится вид; оттенки синего – маловероятные условия. Для нашего вида, *Bradypus*, можно отметить, что для подходящих условий предсказывается высокая вероятность на территории большей части равнин Центральной Америки, влажных низменностей северо-запада Южной Америки, в бассейне Амазонки, на Карибских островах, и большей части Атлантических лесов юго-восточной Бразилии.

Графическое представление модели — простой графический файл (в формате PNG), по нему можно щёлкнуть для просмотра или открыть в любом графическом редакторе. Если вам нужно скопировать эти данные или открыть их с помощью другого ПО, вам нужно найти файлы *.png в каталоге “plots”, там же, где и результаты работы.

Тестовые точки представляют из себя случайную выборку из набора находок. Эта же случайная выборка используется каждый раз при запуске Maxent на одном и том же наборе данных, если в настройках программы не выбрана опция генерации случайной выборки “random seed”. В качестве альтернативы, тестовые данные могут находиться в отдельных файлах и указаны в опции “Test sample file” в настройках.

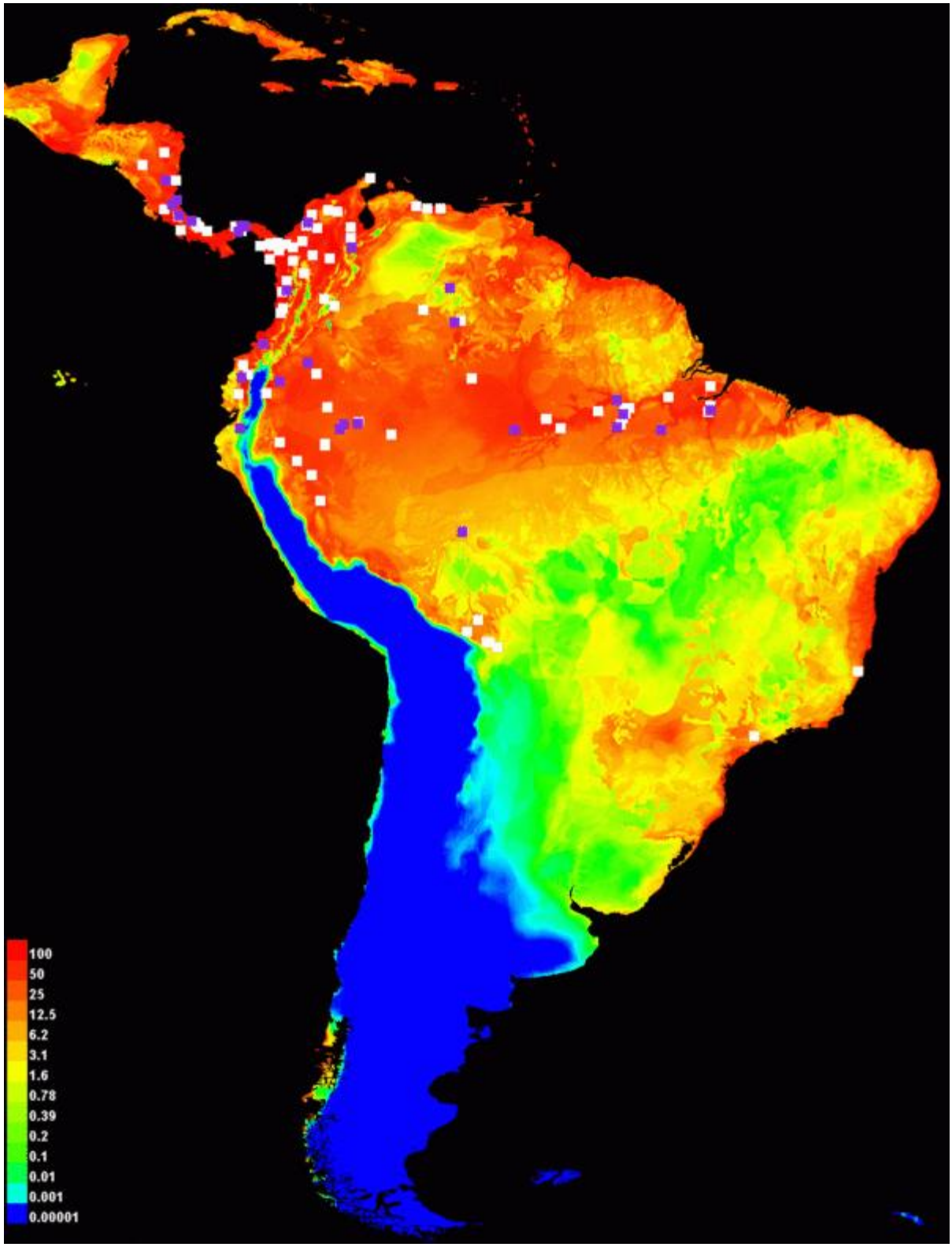
Выходные форматы

Maxent поддерживает три выходных формата значений модели: исходный, кумулятивный и логистический. Первый исходный формат представляет из себя саму экспоненциальную модель Maxent. Второй — кумулятивное значение, соответствующее исходному значению *r* – процент распределения Maxent с исходным значением в большинстве *r*. Кумулятивный выходной формат лучше интерпретировать как предсказываемый уровень омиссии (пропуска). Если мы установим кумулятивный порог *c*, результирующее бинарное предсказание будет иметь уровень омиссии *c*% при выборке из самого распределения Maxent, и мы можем предсказать такой же уровень омиссии для выборки из распределения вида. Третий формат — *c* является экспоненциалом энтропии распределения Maxent, то логистическое значение, соответствующее исходному значению *r*, рассчитывается как $c \cdot r/(1+c \cdot r)$. Это логистическая функция, так как исходное значение есть экспоненциальная функция переменных среды. Три выходных формата монотонно связаны, но разным образом масштабированы и имеют разную интерпретацию. Выходной формат по умолчанию – логистический, его интерпретировать проще всего: он даёт оценку вероятности находки в интервале между 0 и 1. Отметьте, что вероятность находки зависит от того, как собирались данные, например, от размера участка (для мобильных видов), от времени наблюдения. Логистическое значение оценивает вероятность находки, подразумевая, что сэмплинг таков, что типичные локации имеют вероятность находки, равную 0.5. Иллюстрация модели *Bradypus* выше использует логистический формат. Для сравнения, исходный формат даст такое изображение:



Обратите внимание, что используется логарифмическая шкала цветов. Линейная шкала была бы представлена в основном синими цветами с несколькими красными пикселами (убедиться в этом можно, отключив пункт “Logscale pictures” в панели Settings), так как исходные данные обычно имеют небольшое число пунктов с относительно большими значениями. Это можно рассматривать как следствие того, что сырые выходные данные имеют экспоненциальное распространение.

Выбор кумулятивного выходного формата даст следующую картину:

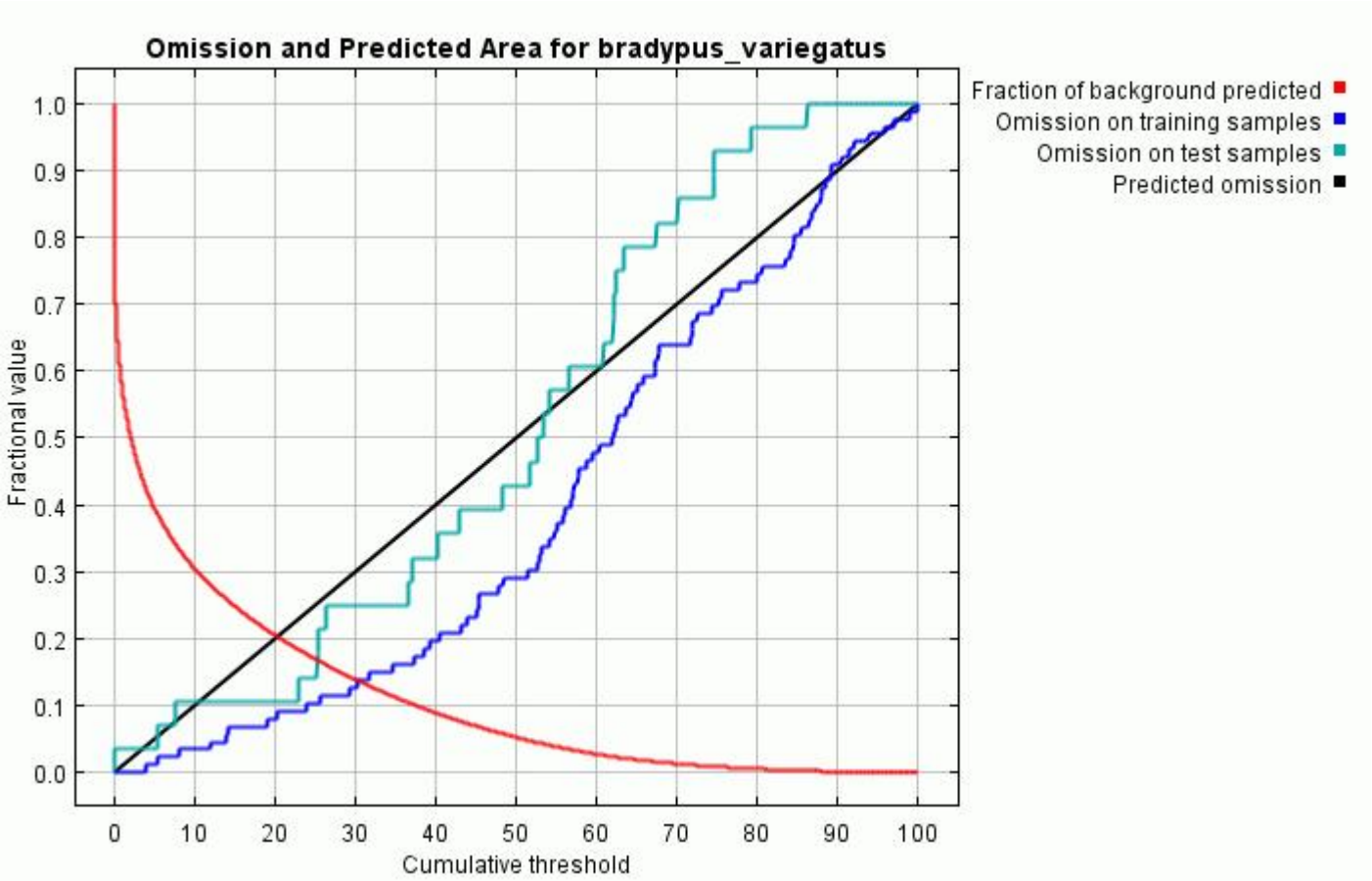


Так же, как и с сырыми выходными данными, мы использовали логарифмическую шкалу для расцветки, чтобы подчеркнуть различия между небольшими значениями. Кумулятивные выходные данные могут быть интерпретированы как предсказание подходящих условий для вида выше порога с примерным диапазоном 1-20 (цветовой градиент от желтого до оранжевого на иллюстрации), и в зависимости от подходящего уровня предсказанной омиссии.

Статистический анализ

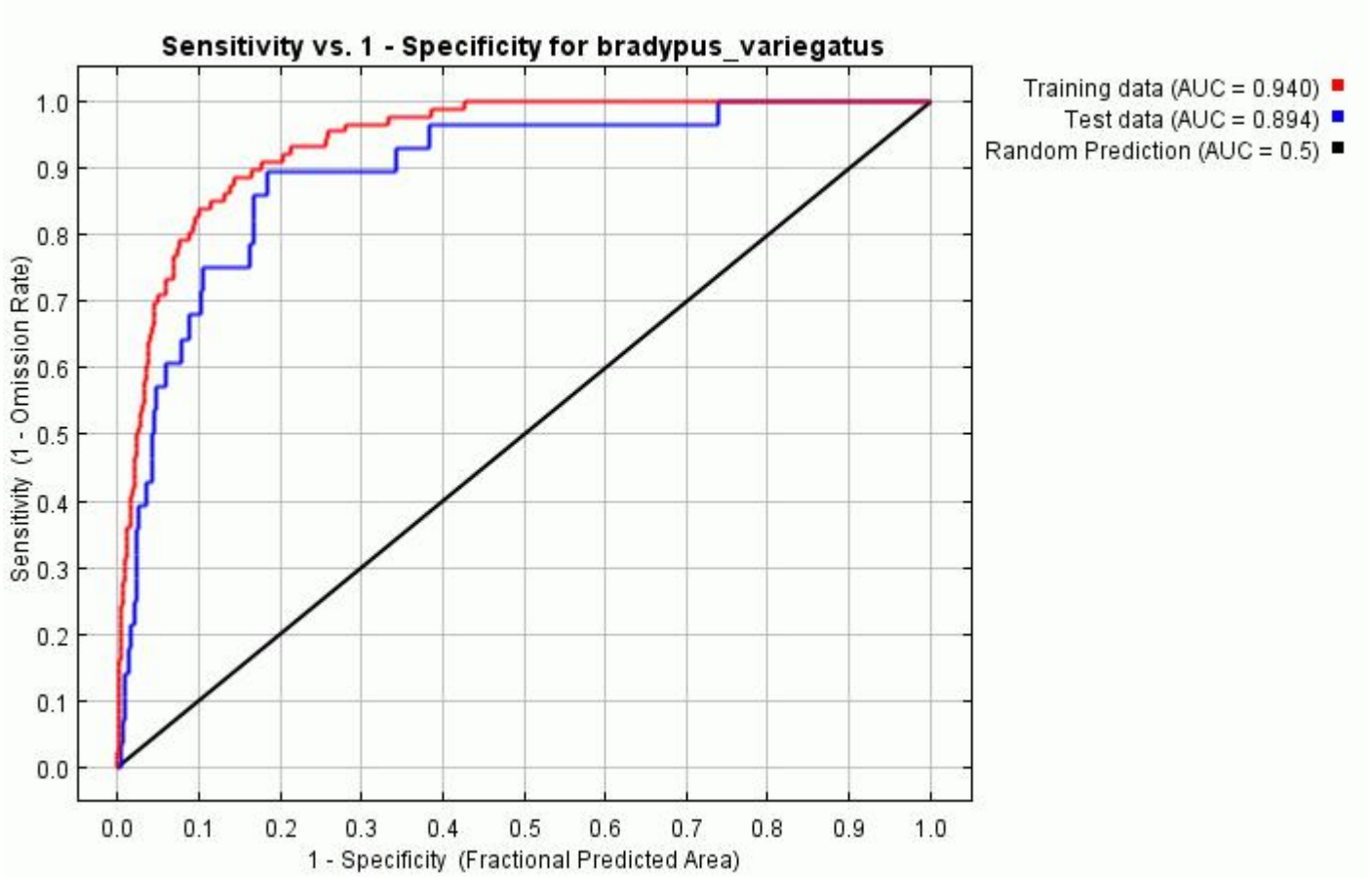
Цифра “25”, которую мы ввели как процент случайных тестовых данных (“random test percentage”), говорит программе, что она должна случайным образом отобрать 25% находок и отложить их в сторону для тестирования. Это позволяет произвести статистический анализ точности. Часто для анализа используется порог для бинаризации предсказания, условия считаются пригодными, если предсказание выше порога, и непригодными, если ниже. Первый график показывает, как меняется омиссия и предсказанная территория по

тестовым и тренировочным точкам в зависимости от кумулятивного порога:



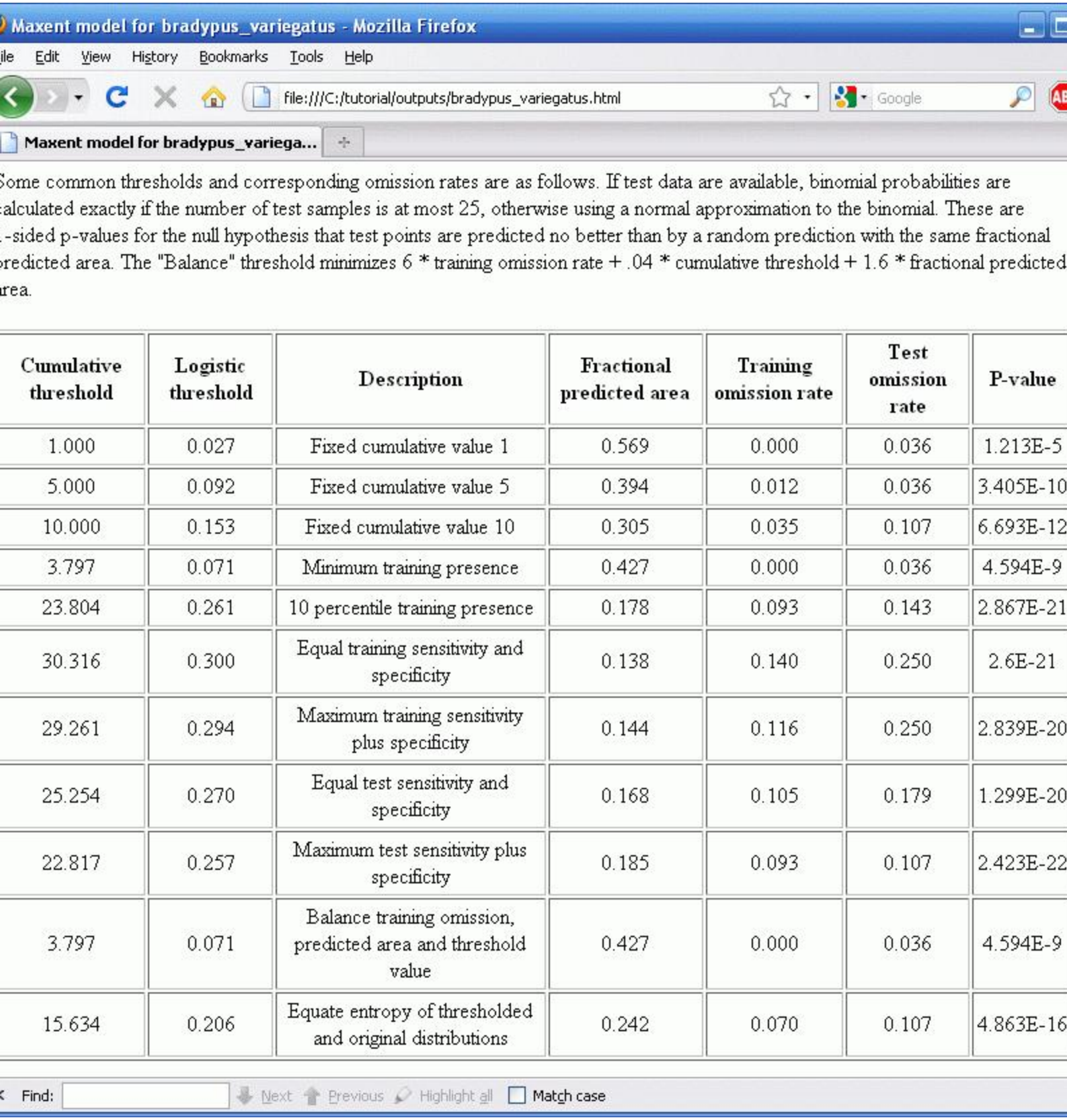
По этому графику можно сказать, что омиссия по тестовым точкам довольно хорошо совпадает с предсказанной динамикой омиссии, рассчитанной для тестовых данных, полученных из самого распределения Maxent. Предсказанная омиссия является прямой линией по определению кумулятивного формата. В некоторых ситуациях линия омиссии по тестовым точкам может лежать ниже предсказанной линии. Обычно это объясняется тем, что тестовые и тренировочные данные независимы, если они получены, например, из общего автокоррелированного набора данных о находках.

Следующий график (см. ниже) показывает receiver operating curve для тренировочных и тестовых данных. Приведена площадь под кривой ROC (AUC); если есть тестовые данные, ниже на странице отчёта можно увидеть стандартную ошибку AUC на тестовых данных.



Красный и синий график будут совпадать, если для тренировки и тестирования используются одни и те же данные. Если данные разделены на две части, одна из которых для тренировки, а вторая для тестирования, то красная кривая (тренировка) показывает значение AUC выше, чем синяя (тестирование). Красная кривая показывает, насколько хорошо модель описывает тренировочные данные ("fit"). Синяя линия показывает, насколько хорошо модель описывает тестовые данные и является реальным тестом предсказательной способности модели. Черная линия показывает ситуацию, которую можно было бы ожидать, если бы надежность предсказаний модели была на случайном уровне. Если синяя или красная линии находятся ниже чёрной, это означает, что уровень достоверных предсказаний модели даже ниже, чем случайный. Чем ближе к верхнему левому углу находится синяя линия, тем лучше модель предсказывает находки, содержащиеся в тестовой выборке. Подробную начальную информацию об AUC можно найти в: Fielding, A.H. & Bell, J.F. (1997) A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24(1): 38-49. Поскольку у нас есть только данные о находках, но нет данных об отсутствии, вместо ошибки омиссии (доля отсутствующих, предсказанная как встречи) используется "fractional predicted area" (доля от площади территории исследования, занимаемая встречами). С обсуждением выбора этого показателя можно ознакомиться в статье в журнале Ecological Modelling, упоминаемой на первой странице этого руководства. Важно отметить, что значения AUC имеют тенденцию быть выше для видов с небольшими участками относительно территории исследования, описываемой слоями данных. Это не обязательно означает, что модель лучше, это всего лишь артефакт показателя AUC.

Если есть данные для тестирования, программа автоматически подсчитает статистическую значимость предсказания, используя биномиальный тест на омиссию. Для *Bradypus* получится:


Maxent model for bradypus_variegatus - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/tutorial/outputs/bradypus_variegatus.html

Maxent model for bradypus_variega...

Some common thresholds and corresponding omission rates are as follows. If test data are available, binomial probabilities are calculated exactly if the number of test samples is at most 25, otherwise using a normal approximation to the binomial. These are 1-sided p-values for the null hypothesis that test points are predicted no better than by a random prediction with the same fractional predicted area. The "Balance" threshold minimizes 6 * training omission rate + .04 * cumulative threshold + 1.6 * fractional predicted area.

Cumulative threshold	Logistic threshold	Description	Fractional predicted area	Training omission rate	Test omission rate	P-value
1.000	0.027	Fixed cumulative value 1	0.569	0.000	0.036	1.213E-5
5.000	0.092	Fixed cumulative value 5	0.394	0.012	0.036	3.405E-10
10.000	0.153	Fixed cumulative value 10	0.305	0.035	0.107	6.693E-12
3.797	0.071	Minimum training presence	0.427	0.000	0.036	4.594E-9
23.804	0.261	10 percentile training presence	0.178	0.093	0.143	2.867E-21
30.316	0.300	Equal training sensitivity and specificity	0.138	0.140	0.250	2.6E-21
29.261	0.294	Maximum training sensitivity plus specificity	0.144	0.116	0.250	2.839E-20
25.254	0.270	Equal test sensitivity and specificity	0.168	0.105	0.179	1.299E-20
22.817	0.257	Maximum test sensitivity plus specificity	0.185	0.093	0.107	2.423E-22
3.797	0.071	Balance training omission, predicted area and threshold value	0.427	0.000	0.036	4.594E-9
15.634	0.206	Equate entropy of thresholded and original distributions	0.242	0.070	0.107	4.863E-16

Find: Next Previous Highlight all Match case

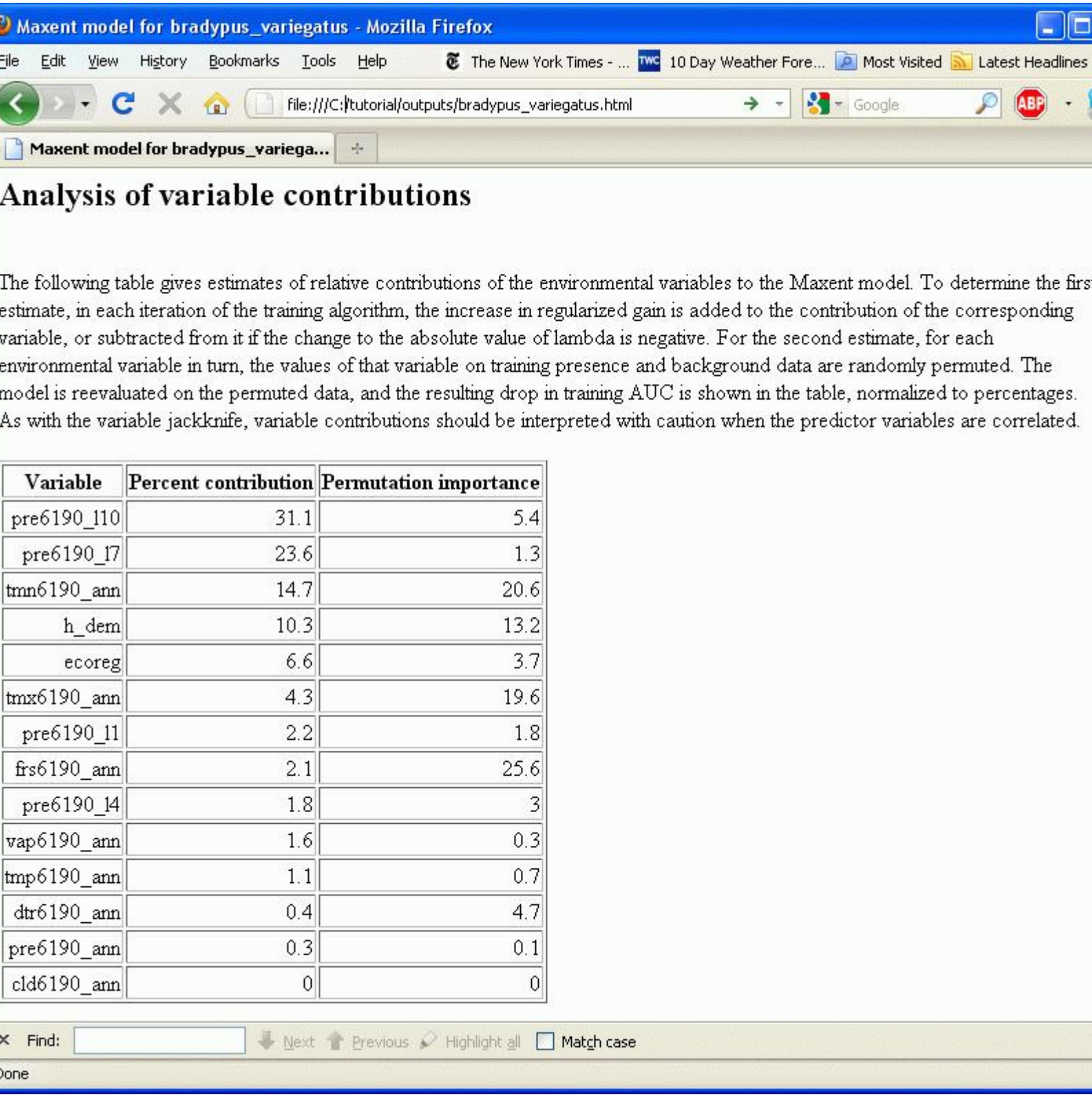
Подробную информацию по биномиальной статистике можно найти в статье в Ecological Modelling.

Какие переменные имеют больший вес?

Естественное применение моделирования – попытка ответить на вопрос, какая из переменных более важна для вида? Существует несколько способов дать ответ на этот вопрос с помощью Maxent.

В процессе тренировки модели Maxent она отслеживает, какие переменные среды вносят вклад в построение модели. Каждый шаг алгоритма Maxent увеличивает прирост модели, изменяя коэффициент для одной из функций градиента среды; программа назначает увеличение прироста той переменной или переменным среды, от которой зависит эта функция. В конце процесса тренировки происходит пересчёт приростов в

проценты и получается средняя колонка в таблице:

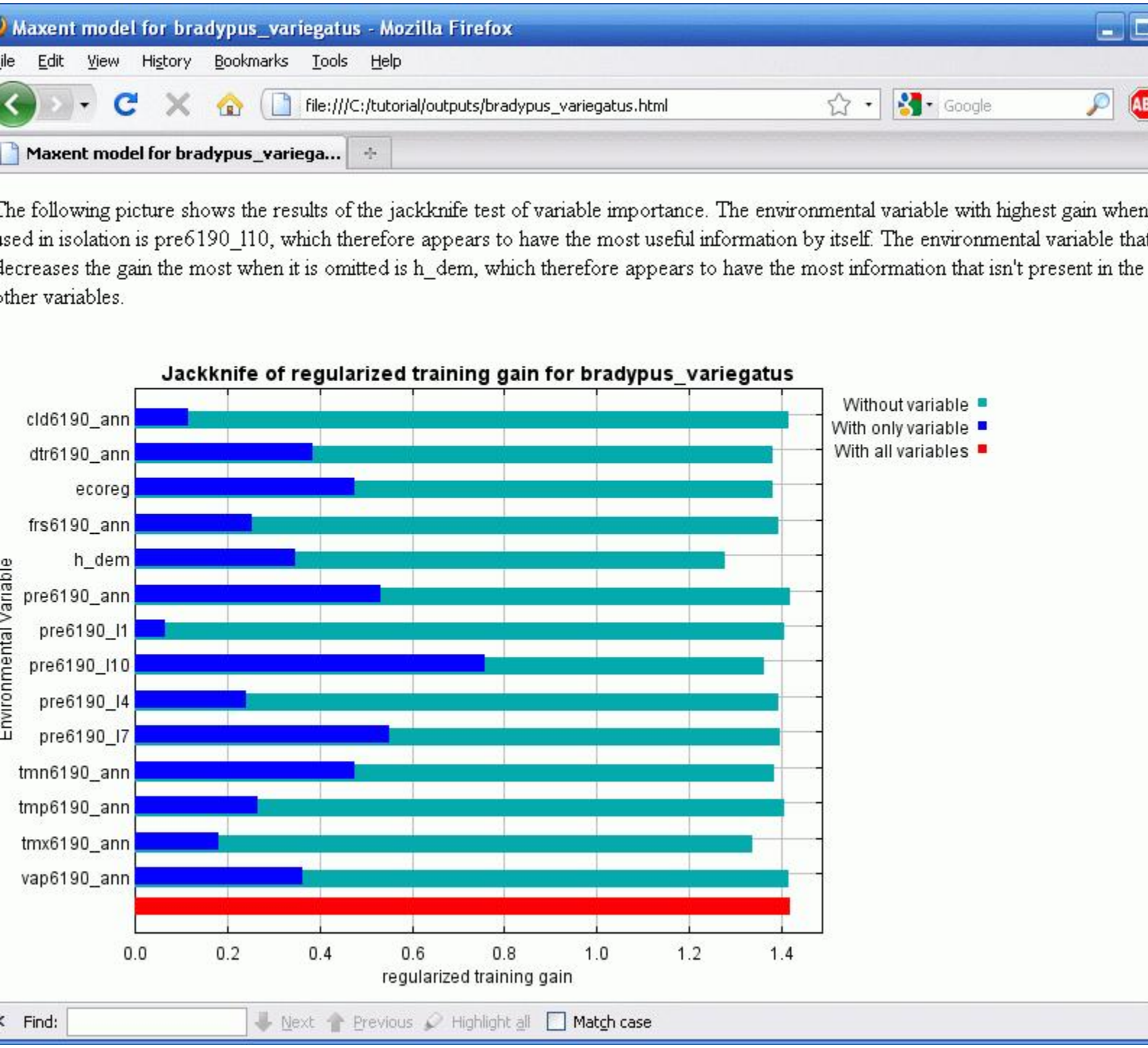


Эти процентные вклады определяются только эвристически и зависят от конкретного пути, по которому пошёл код Maxent, чтобы достичь оптимального решения. Другой алгоритм мог бы прийти к тому же решению, используя другой путь, который бы закончился другими процентами вклада. Дополнительно, если присутствуют сильно скоррелированные переменные, вклад должен интерпретироваться с осторожностью. В нашем примере с *Bradypus* годовые осадки сильно скоррелированы с осадками в октябре и июле. Хотя таблица выше показывает, что Maxent использовал количество осадков за октябрь больше, чем любую другую переменную, и вообще не использовал годовые осадки, это не обязательно означает, что осадки в октябре гораздо более важны для вида, чем годовые осадки.

Правая колонка в таблице показывает второй показатель вклада переменной – важность при пермутации (permutation importance). Этот показатель зависит только от окончательной модели, а не пути, который был

пройден, чтобы её получить. Вклад каждой переменной определяется посредством случайного изменения значений этой переменной у тренировочных точек (и находок, и фона), а также измерения уменьшения тренировочной AUC. Значительное уменьшение свидетельствует о том, что модель сильно зависит от этой переменной. Значения нормализованы и показываются в процентах.

Альтернативной оценкой важности переменной может быть "jackknife"-тест, который можно провести, включив переключатель “Do jackknife to measure variable important” (производить "jackknife"-тест для измерения важности переменной). После нажатия кнопки “Run” (Запуск) создаётся набор моделей. Каждая переменная исключается по очереди очередь и модель создаётся с остальными переменными. Затем модель создаётся только с этой переменной. Дополнительно создаётся модель со всеми переменными, как раньше. Результаты "jackknife"-тестов будут показываться в файле “bradypus.html” в виде трёх столбчатых диаграмм, первая из них показана ниже.

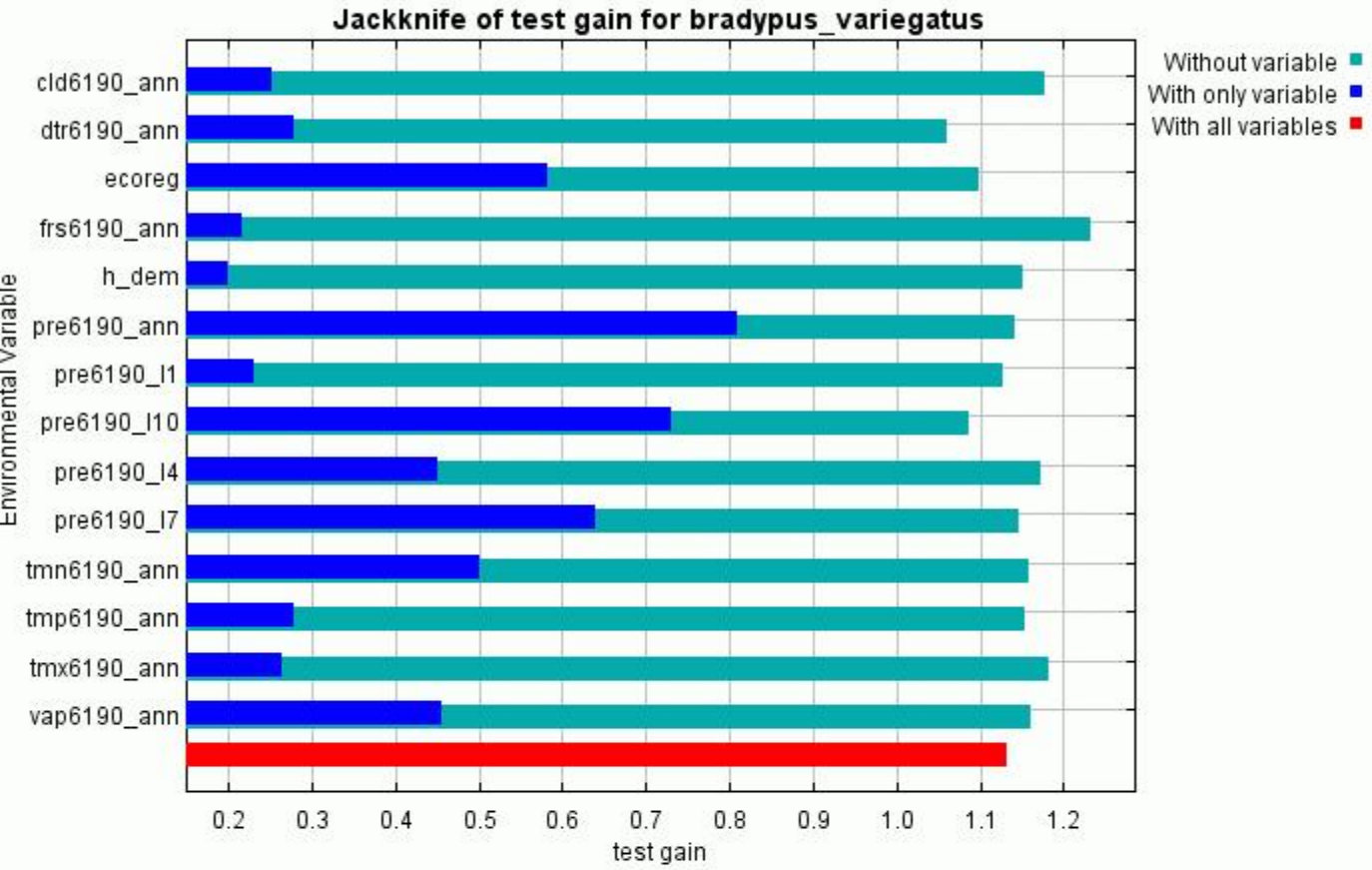


На примере видно, что если Maxent использует только pre6190_l1 (среднее количество осадков в январе), то прироста почти нет, так что, сама по себе эта переменная бесполезна для моделирования распространения *Bradypus*. С другой стороны, осадки в октябре (pre6190_l10) достаточно хорошо описывают данные. Переключась на голубые столбцы, можно отметить, что ни одна из переменных не содержит значительного

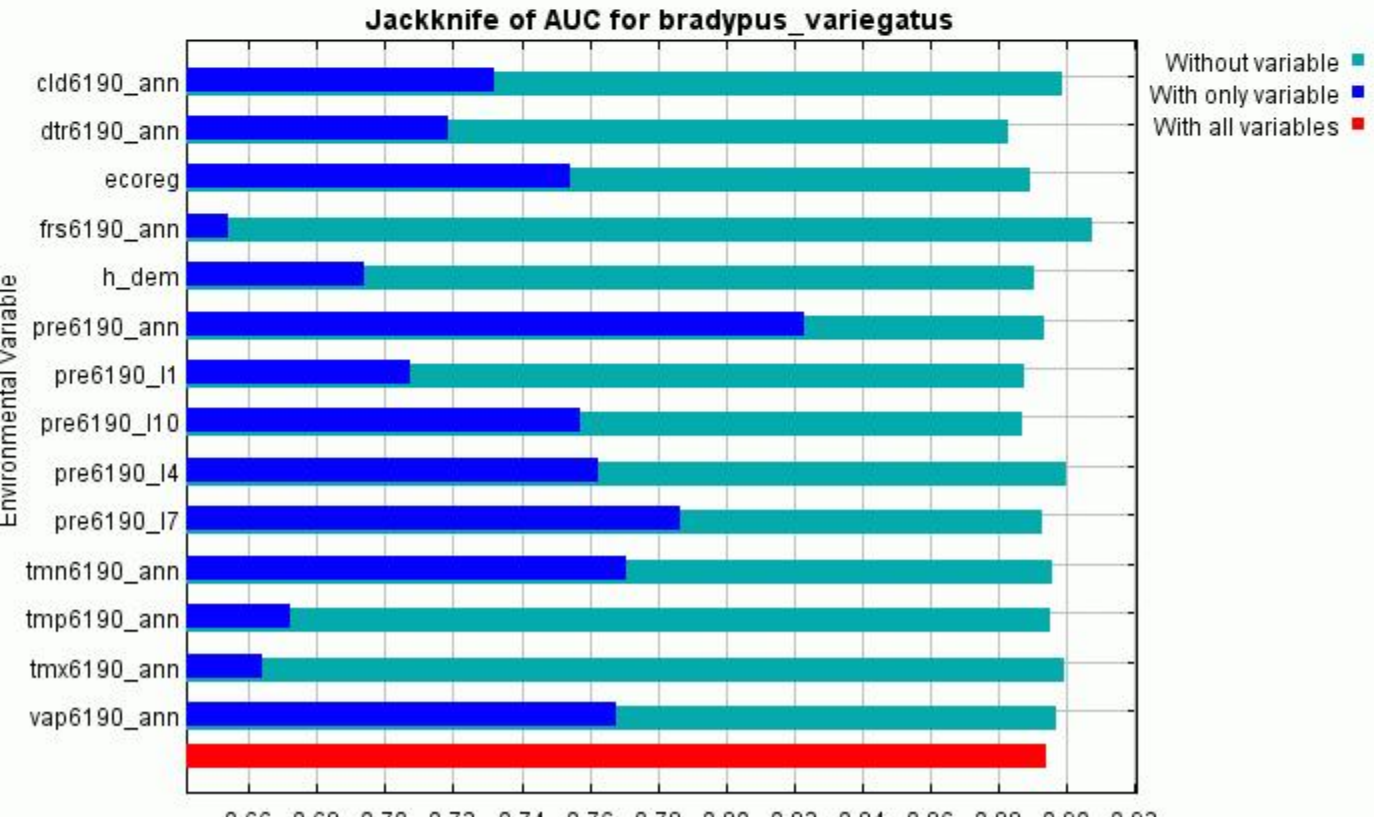
количества уникальной информации, которая бы не содержалась в других, поэтому выключение каждой переменной не привело к значительному уменьшению прироста.

В файле "bradypus.html" также находятся две диаграммы результатов "jackknife"-теста, показывающих прирост для тестовых данных или AUC, см. ниже:

The next picture shows the same jackknife test, using test gain instead of training gain. Note that conclusions about which variables are most important can change, now that we're looking at test data.



Lastly, we have the same jackknife test, using AUC on test data.



Сравнение диаграмм может быть весьма полезным. Диаграмма AUC показывает, что годовые осадки - это переменная, которая в одиночку наиболее эффективно предсказывает распределение точек находок, которые были выделены в тестовый набор данных. Эта эффективность предсказания измерена с помощью AUC, однако, эта переменная практически не используется для построения модели когда используются все переменные. Относительная важность годовых осадков также достаточно велика судя по диаграмме прироста по тестовым данным, по сравнению с тренировочными. Дополнительно, эти две диаграммы показывают, что некоторые из светло-голубых столбцов (особенно для месячных осадков) длиннее чем красный столбец, что говорит о том, что предсказательная сила модели увеличивается если эти переменные не используются.

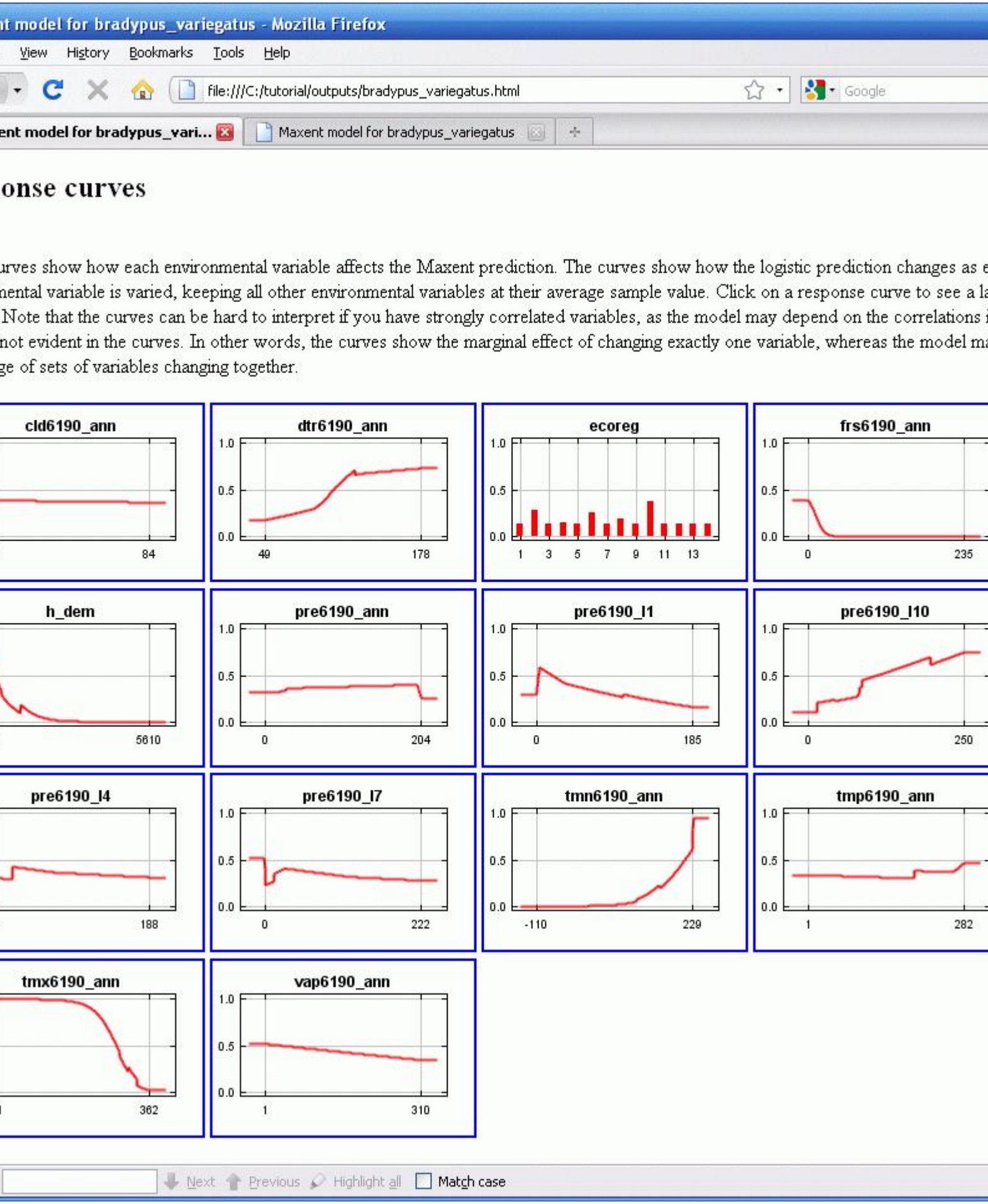
Это говорит нам о том, что переменные месячных осадков помогают Maxent хорошо описать тренировочные данные, но годовые осадки лучше генерализуют, показывая сравнительно лучший результат для отложенных тестовых данных. Другими словами, модели созданные с месячными осадками являются менее переносимыми. Это важно, если нашей целью является перенос модели, например применение ее к будущим климатическим переменным с тем, чтобы оценить будущее распространение вида при условии изменения климата. То, что месячные значения осадков являются менее переносимыми – логично: вероятно, что подходящие местообитания для *Bradypus* будут зависеть не от точных значений осадков в конкретный месяц, а от общих значений и, возможно, периодичности дождей и отсутствия сухих периодов. При моделировании на континентальном уровне весьма вероятны сдвиги в точном времени сезонных осадков, которые будут влиять на месячные осадки, но не на подходящие условия для *Bradypus*.

В целом, лучше использовать переменные которые будут вероятнее напрямую связаны с моделируемым видом. Например, вебсайт Worldclim (www.worldclim.org) предоставляет переменные “BIOCLIM”, включающие прозводные, такие как “осадки в четверть с наибольшим их количеством ”, а не просто месячные значения.

Последнее замечание о результатах jackknife-теста: диаграмма прироста с тестовыми данными показывает, что модель созданная только с осадками в январе (pre6190_l1) имеет отрицательный прирост. Это означает, что модель хуже, чем нулевая (т.е., a uniform distribution) дял предсказания распространения находок отложенных для тестирования. Это предоставляет дальнейшие данные о том, что значения месячных осадков не самый лучший выбор для предсказания.

Как предсказание зависит от переменных?

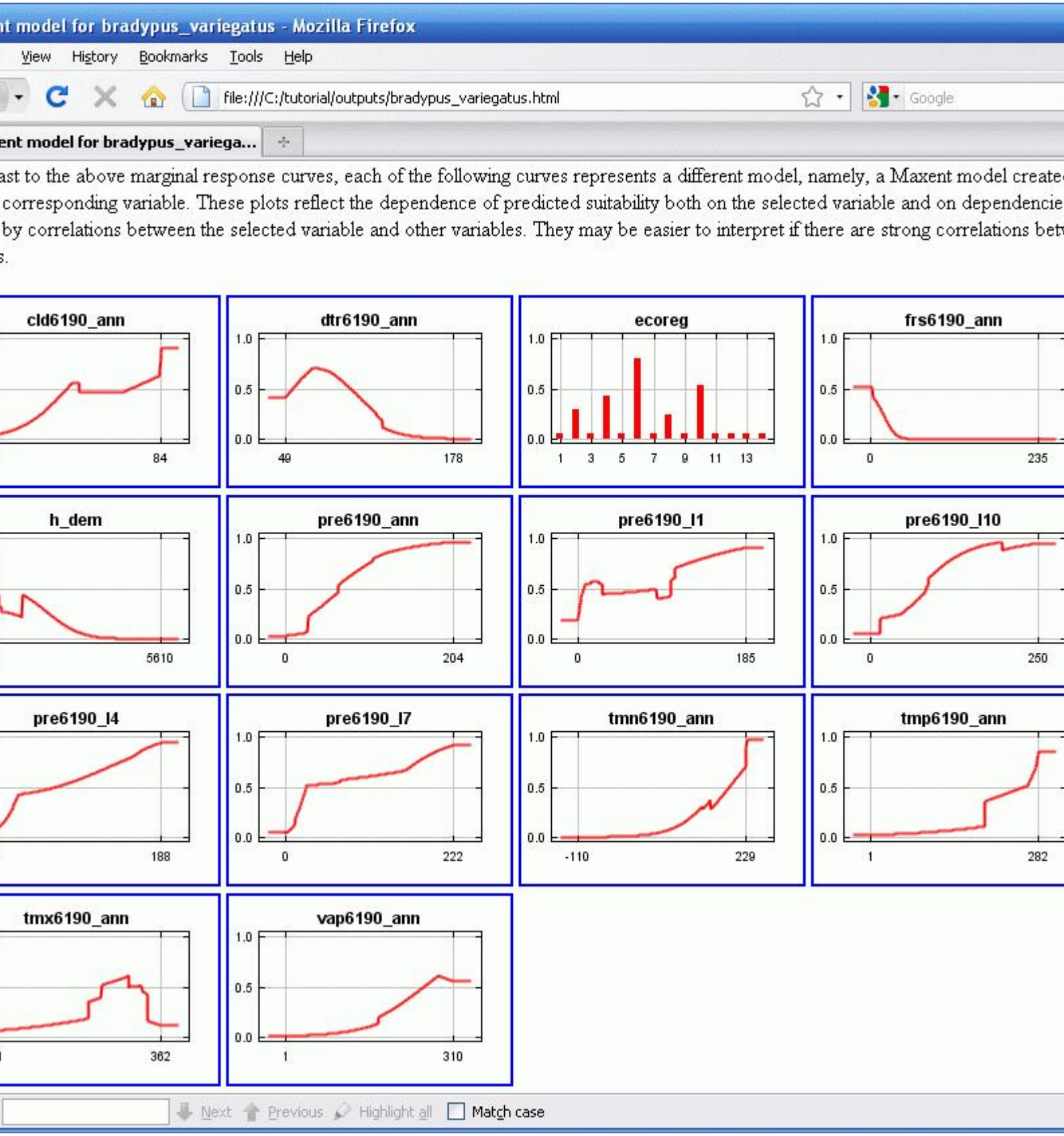
Теперь нажмите на “Create response curves”, отключите jackknife, и перезапустите расчеты. В результате к “bradypus.html” добавится еще одна секция:



Каждая из картинок является ссылкой, при нажатии на которую откроется полная версия графика, сами файлы в формате .png находятся в папке “plots”. Изучение vap6190_ann показывает, что отклик имеет небольшие значения при значениях vap6190_ann в диапазоне 1-200, и высокие при значениях в диапазоне 200-300.

Значение на оси Y это предсказанная вероятность подходящих условий, в логистическом формате вывода, при том что все остальные переменные становлены в средние значения расчитанные по всему набору находок (presence localities).

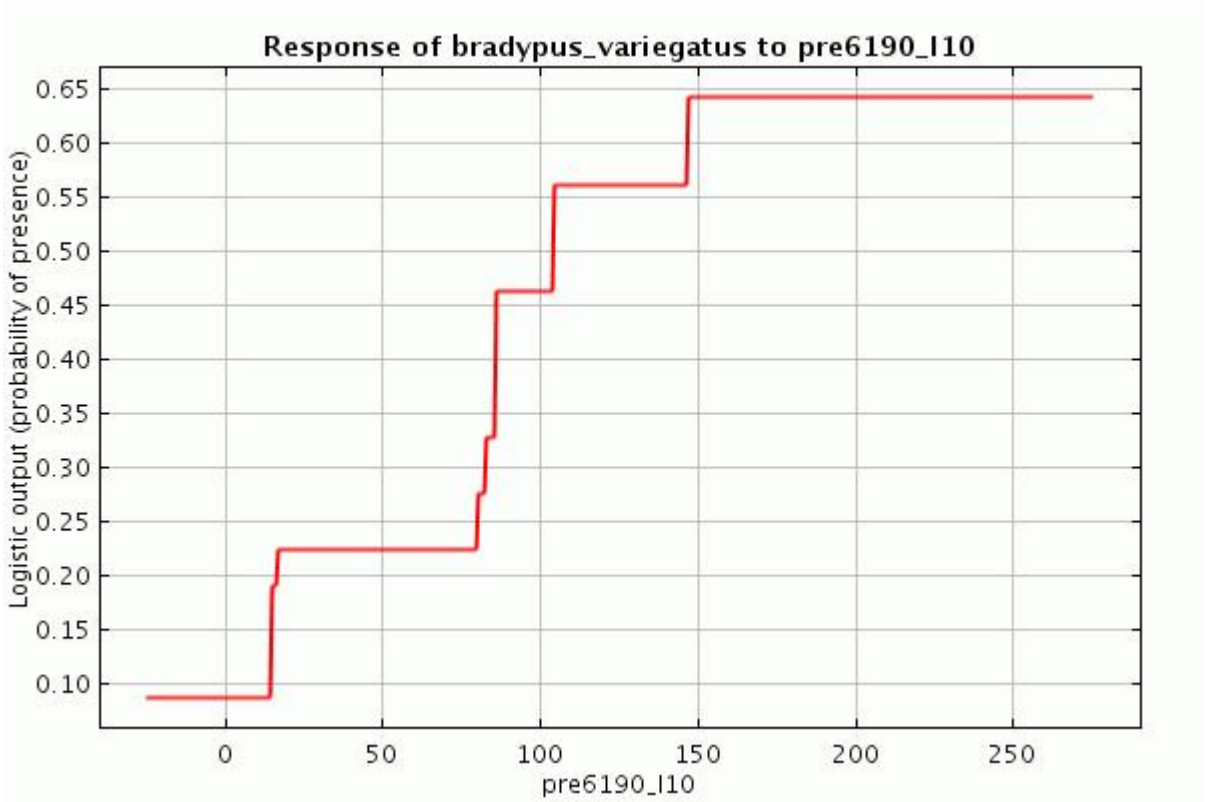
Отметьте, что если переменные среды скоррелированы, как в нашем примере, кривые отклика могут быть противоречивы. Например, если две сильно коррелирующие переменные имеют кривые отклика которые ведут себя совершенно по разному, то совокупный эффект двух переменных может быть очень мал для большинства пикселей. Другой пример, из примера видно, что предсказанная пригодность местообитания (predicted suitability) отрицательно коррелирует с годовыми осадками (pre6190_ann), если все другие переменные неизменны. Другими словами, как только эффект всех других переменных учтен, эффект увеличения годовых осадков будет выражаться в уменьшении предсказанной пригодности местообитания. Однако, годовые осадки сильно коррелируют с ежемесячными осадками, так что на самом деле мы вряд ли сможем зафиксировать ежемесячные значения и менять годовые. Поэтому программа создает два набора кривых отклика, во втором наборе каждая кривая приводится для случая, когда модель построена используя только саму переменную и другие переменные в нее не вводятся:



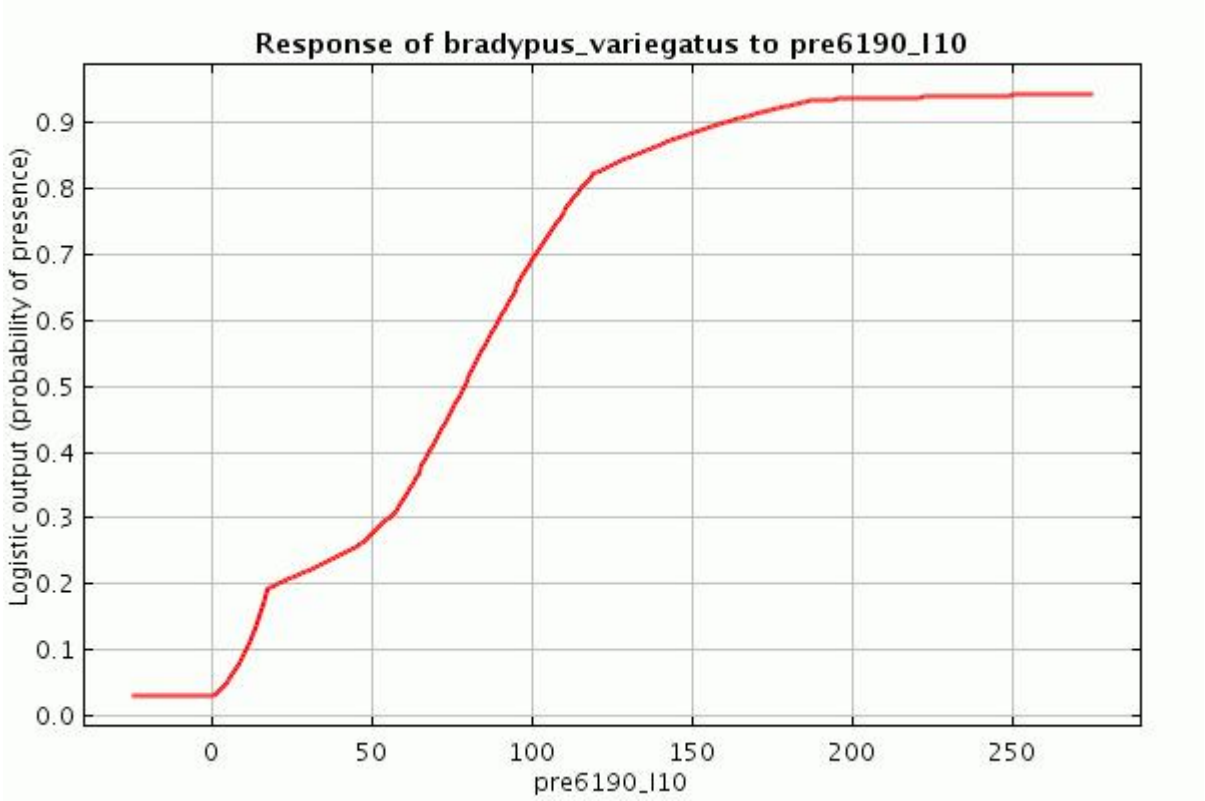
По сравнению с откликом на годовые осадки в первом наборе, во втором мы видим, что предсказанная пригодность местообитания в целом растет с ростом годовых осадков.

Типы функций и кривые зависимостей

Кривые зависимостей позволяют нам увидеть разницу между различными типами функций градиентов среды. Отключите авто-функции(“Auto features”), оставьте выбранными только пороговые числовые характеристики(“Threshold features”), и снова нажмите кнопку запуска (“Run”). Посмотрите на результирующий профиль функции – она выглядит как набор ступенек, например так для переменной 6190_l10:



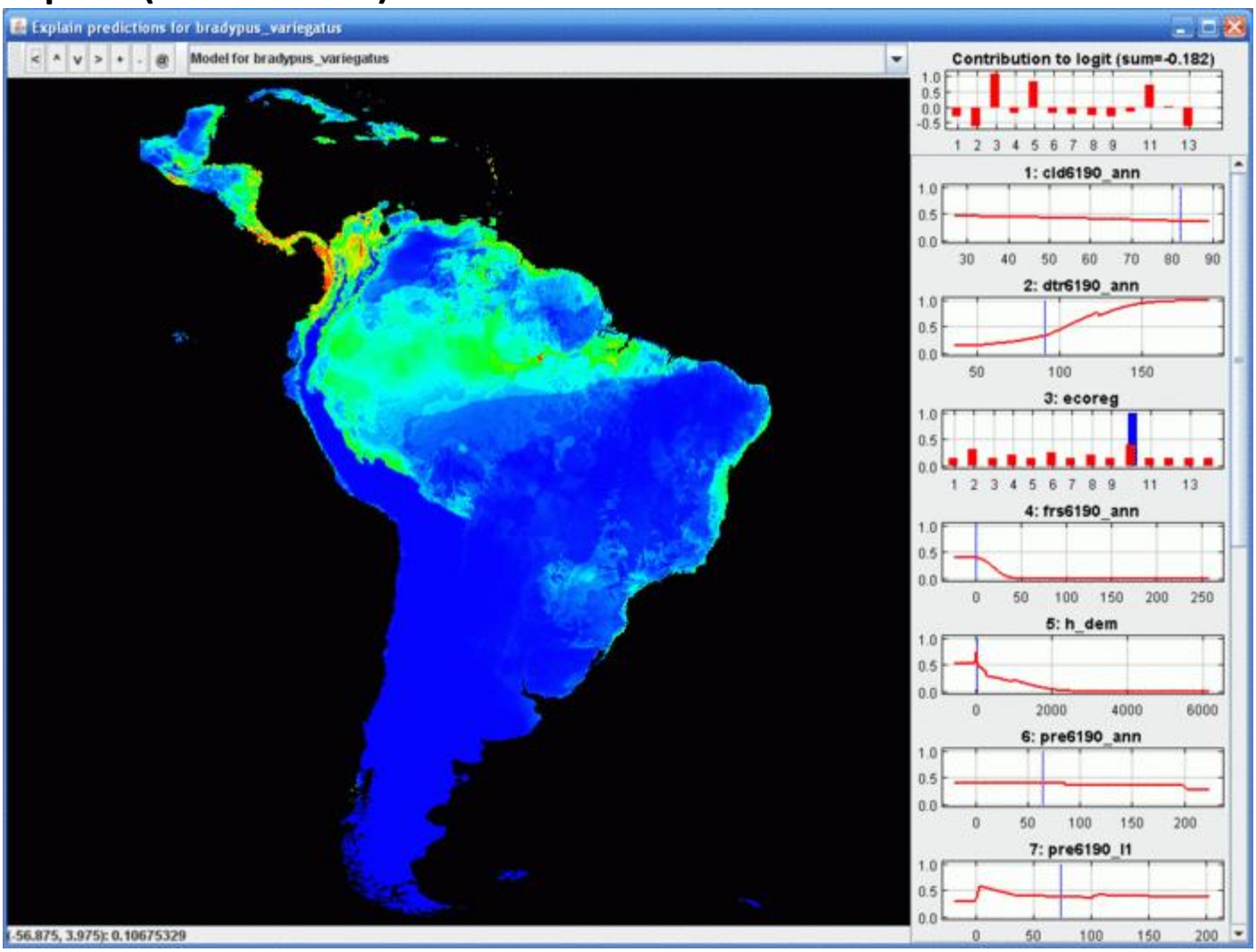
Теперь попробуйте тоже самое используя только нелинейные числовые признаки («hinge features»), результат будет выглядеть примерно так:



Общий контур двух профилей похож, но в деталях они разные, так как разные типы функций позволяют разные возможные формы кривых зависимостей. Экспонента в модели Maxent является суммой числовых характеристик функций среды, а сумма пороговых числовых характеристик (threshold features) это всегда ступенчатая функция, так что логистический результат - тоже ступенчатая функция (также как сырой и кумулятивный результат). Для сравнения, сумма нелинейных числовых характеристик (hinge features) - всегда кусочно-линейная функция, так что если используются только нелинейные признаки, экспонента Maxent является кусочно линейной. Это объясняет последовательность соединенных линейных сегментов во второй кривой зависимости приведенной выше. (Отметьте, что линии слегка кривые, особенно там где значения переменной близки к экстремальным; это происходит потому что логистический результат применяет

сигмоидную функцию к экспоненте Maxent.) Использование всех видов объектов (по умолчанию, если достаточно данных) позволяет точно моделировать даже сложные зависимости. Более подробное объяснение различных видов функций доступно в справке программы (кнопка help).

Интерактивное изучение результатов предсказания: инструмент Explain (объяснение)



Этот интерактивный инструмент позволяет изучить, как предсказание Maxent связано со значениями предикторов в любом месте территории исследования. Щелкнув на точку на карте можно посмотреть на ее положение на кривой отклика. Верхний правый график показывает каков вклад каждой переменной в логит предсказания (logit of the prediction) (наведя курсор на одну из колонок можно увидеть имя переменной и значение в цифрах). Изучив вклад в логит, можно сказать, как разные переменные влияют на предсказание в разных частях территории исследования.

Инструменту нужно, чтобы модель была аддитивная (без взаимодействий между переменными) поэтому использовать инструмент можно только на результатах полученных без произведения числовых признаков. Так же, компьютеру может понадобится больше памяти, чтобы держать в ней одновременно все предикторы. Если вы прогоните модель без product features, после главной иллюстрации работы модели появится ссылка, нажав на которую можно будет запустить инструмент.

Формат SWD

Еще один формат который может оказаться полезным, особенно если вы имеете дело с предикторами очень большими по объему. Для простоты этот формат называется образцы с данными (“samples with data”) или SWD. SWD-версия файла *Bradypus* называется “bradypus_swd.csv” и его содержание начинается так:

```
species,longitude,latitude,cld6190_ann,dtr6190_ann,ecoreg,frs6190_ann,h_dem,pre6190_ann
,pre6190_l10,pre6190_l1,pre6190_l4,pre6190_l7,tmn6190_ann,tmp6190_ann,tmx6190_ann,vap61
```

```
90_ann
bradypus_variegatus,-65.4,-
10.3833,76.0,104.0,10.0,2.0,121.0,46.0,41.0,84.0,54.0,3.0,192.0,266.0,337.0,279.0
bradypus_variegatus,-65.3833,-
10.3833,76.0,104.0,10.0,2.0,121.0,46.0,40.0,84.0,54.0,3.0,192.0,266.0,337.0,279.0
bradypus_variegatus,-65.1333,-
16.8,57.0,114.0,10.0,1.0,211.0,65.0,56.0,129.0,58.0,34.0,140.0,244.0,321.0,221.0
bradypus_variegatus,-63.6667,-
17.45,57.0,112.0,10.0,3.0,363.0,36.0,33.0,71.0,27.0,13.0,135.0,229.0,307.0,202.0
bradypus_variegatus,-63.85,-
17.4,57.0,113.0,10.0,3.0,303.0,39.0,35.0,77.0,29.0,15.0,134.0,229.0,306.0,202.0
```

Такой файл может использоваться вместо обычного файла образцов. Разница только в том, что при использовании SWD программе не нужно смотреть в слои предикторов (ASCII файлы) чтобы получить значения переменных в точках, вместо этого она считывает эти значения прямо из таблицы. Слои предикторов в этом случае используются только для того, чтобы считать данные о “фоновых” пикселях - т.е. пикселях, где вид (явление) не было обязательно детектировано. На самом деле, фоновые пиксели тоже могут быть заданы как файл в формате SWD. Файл “background.csv” содержит 10,000 фоновых точек. Несколько первых записей выглядят так:

```
background,-
61.775,6.175,60.0,100.0,10.0,0.0,747.0,55.0,24.0,57.0,45.0,81.0,182.0,239.0,300.0,232.0
background,-
66.075,5.325,67.0,116.0,10.0,3.0,1038.0,75.0,16.0,68.0,64.0,145.0,181.0,246.0,331.0,234
.0
background,-59.875,-
26.325,47.0,129.0,9.0,1.0,73.0,31.0,43.0,32.0,43.0,10.0,97.0,218.0,339.0,189.0
background,-68.375,-
15.375,58.0,112.0,10.0,44.0,2039.0,33.0,67.0,31.0,30.0,6.0,101.0,181.0,251.0,133.0
background,-
68.525,4.775,72.0,95.0,10.0,0.0,65.0,72.0,16.0,65.0,69.0,133.0,218.0,271.0,346.0,289.0
```

Мы можем запустить Maxent с “bradypus_swd.csv” в качестве файла образцов и “background.csv” в качестве файла слоёв предикторов (оба файла находятся в папке “swd”). Попробуйте запустить процесс - вы увидете, что он идет гораздо быстрее, потому что не нужно загружать массивы предикторов целиком. Еще одно преимущество состоит в том, что вы можете связывать образцы с условиями среды за разные промежутся времени. Например, у вас может быть две точки встречи из одной и той же ячейки, но разделенных во времени промежутком в 100 лет, вполне вероятно, что условия в эти два момента времени сильно отличались друг от друга. Если вы не используете формат SWD, обе записи получат абсолютно одинаковые значения переменных. Недостаток этого подхода состоит в том, что вы не сможете создать карты или выходные гриды, потому что вы не используете все данные. Обойти это можно используя “проектирование”, описанное ниже.

Запуск из командной строки

Иногда необходимо создать несколько моделей, с разными параметрами или набором предиктором. Создание моделей может быть автоматизировано посредством запуска из командной строки, это исключает необходимость многократно повторяемых действий мышью в графическом интерфейсе. Параметры командной строки могут использоваться в шелл и в bat-файлах. Пример такого файла - файл “batchExample.bat” (щелкните по нему правой кнопкой мыши и выберите открыть с помощью Notepad). Он содержит следующую строку:

```
java -mx512m -jar maxent.jar environmentallayers=layers togglelayertype=ecoreg
samplesfile=samples\bradypus.csv outputdirectory=outputs redoifexists autorun
```

Эта строка говорит программе где найти слои предикторов и файл с образцами и куда положить результаты, она также указывает, что переменная ecoreg является категорийной. Флаг “autorun” говорит программе немедленно начинать выполнение, не ожидая нажатия кнопки Run. Попробуйте дважды щелкнуть мышью по файлу и посмотреть что произойдет.

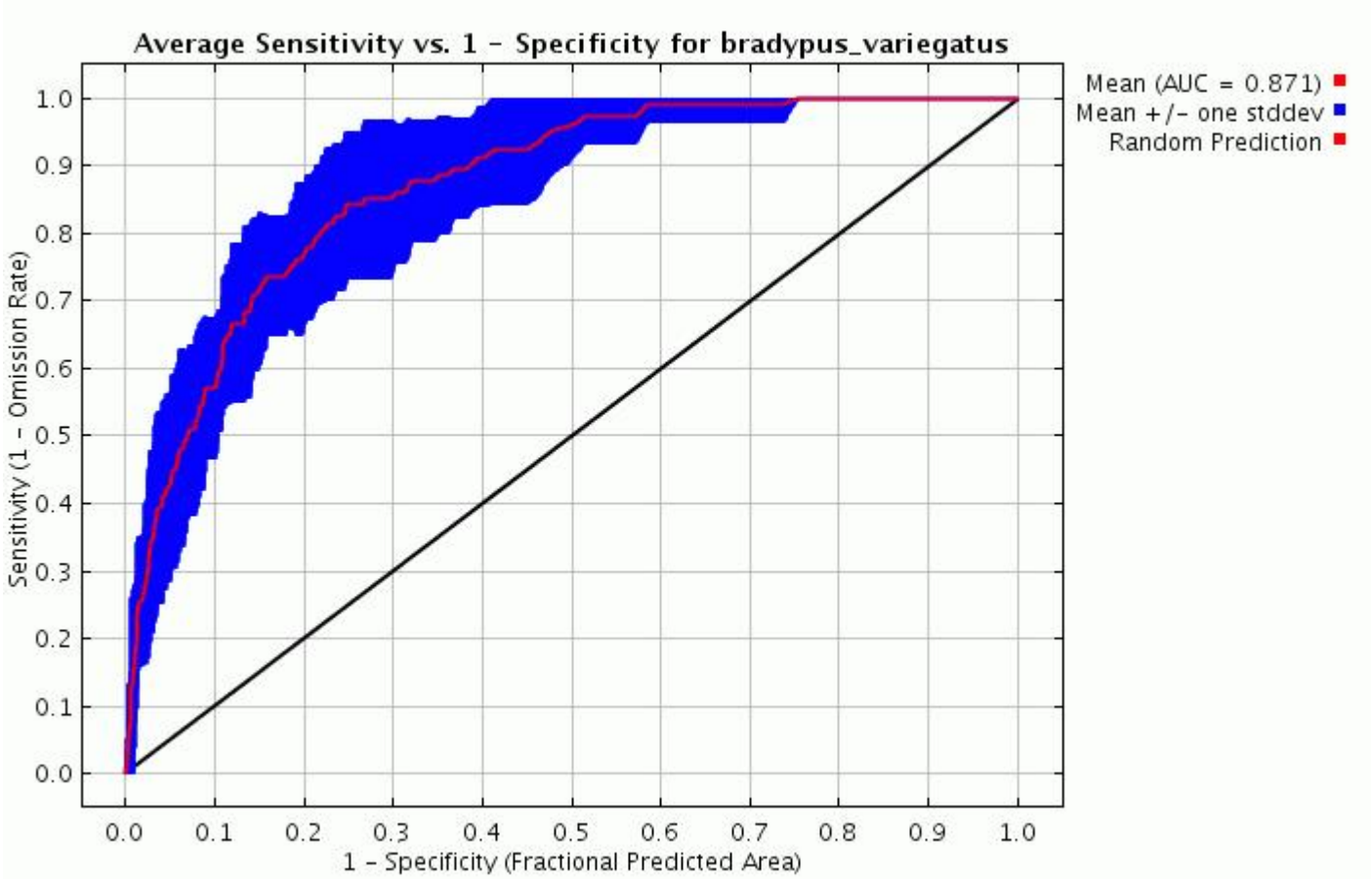
Большинство параметров Maxent можно настраивать из командной строки - нажмите кнопку “Help” чтобы увидеть все возможности. В одном командном файле может быть несколько запусков программы, они будут запущены друг за другом. Можно изменить значения по умолчанию параметров изменяя их в файле

“maxent.bat”. Многие из параметров также могут записываться в сокращенной форме, так, запуск модели в batchExample.bat может также выглядеть вот так:

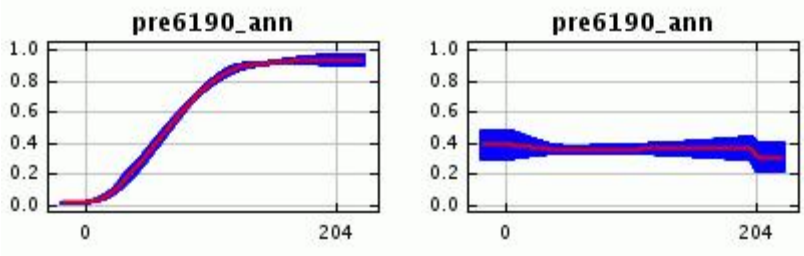
```
java -mx512m -jar maxent.jar -e layers -t eco -s samples\bradypus.csv -o outputs -r -a
```

Репликация

Опция репликации ("replicates") может быть использована для того, чтобы прогнать модель для одного и того же вида несколько раз. Наиболее распространенное применение репликации – сэмплирование и кросс-валидация. Управлять репликацией можно либо из панели Настройки (Settings), либо через параметры командной строки. По умолчанию, при репликации используется кросс-валидация, во время которой образцы разбиваются на группы равного размера, так называемые “folds”, и модель создается по каждому из них отдельно. Оставшиеся группы потом используются для оценки точности. Кросс-валидация имеет одно большое преимущество перед единичным разбиением на тренировочный-тестовый набор: она использует все данные для валидации, что хорошо для небольших наборов данных. Таким образом, если количество репликаций равно 10, то будет создано 10 html-страниц отчетов, плюс еще одна страница суммирующая всю информацию кросс-валидации. Так же, мы получаем ROC-кривые с столбцами ошибок и средние AUC по всем моделям, а также кривые отклика со столбцами ошибок в одно стандартное отклонение. Для *Bradypus*, кросс-валидированная ROC-кривая показывает некоторую изменчивость моделей:



Отклик *Bradypus* при использовании только одной переменной годовых осадков достаточно стабилен (внизу слева), в то время как краевой отклик (marginal response) на годовые осадки меняется большее значительно (внизу справа).

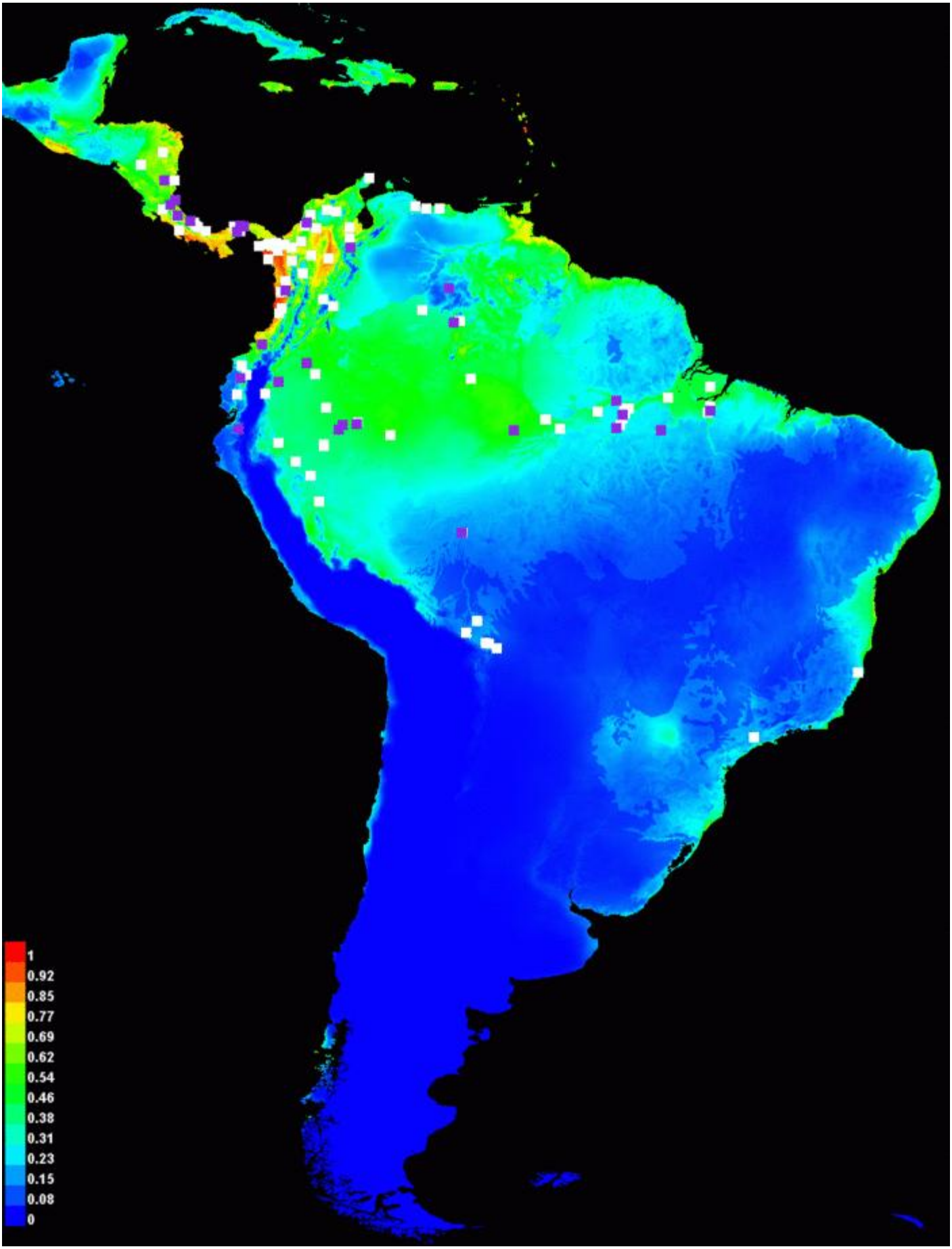


Поддерживаются две формы репликации: повторяющаяся выборка, в процессе которой образцы несколько раз случайно разбиваются на тренировочный и тестовый набор, и метод рэндомизации, когда тренировочный набор создается выборкой с заменой из образцов, количество выборок равно количеству образцов. В случае использования метода рэндомизации, число образцов в каждом наборе будет равно общему количеству образцов (**ШТО?!**), так что тренировочный набор будет содержать повторяющиеся записи.

В обоих случаях, может оказаться полезным отключение опции записи выходных матриц (“write output grids”), что не даст создавать выходные матрицы для каждой репликации и сэкономит дисковое пространство. Созданы будут только общие статистики, среднее, стандартное отклонение и т.д.

Регуляризация

Параметр “множитель регуляризации” (regularization multiplier) доступный через панель настроек управляет тем, насколько жестко выходное распределение "притягивается" к исходным данным образцов - значение меньше 1.0 (значение по умолчанию) приведет к сильнее локализованному выходному распределению, которое более точно соотвествует образцам, но это также может привести к излишнему переобучению модели (overfitting) (подгонка к данным так тесно, что модель становится слабо генерализованной и очень плохо предсказывает независимый тестовый набор). Больший параметр регуляризации даст более широкое, менее локализованное распределение. Попробуйте изменить множитель и посмотрите на результаты и изменения в AUC. Например, установив значение множителя равным 3 мы получим такую карту, где распределение имеет более размытые границы, чем раньше:

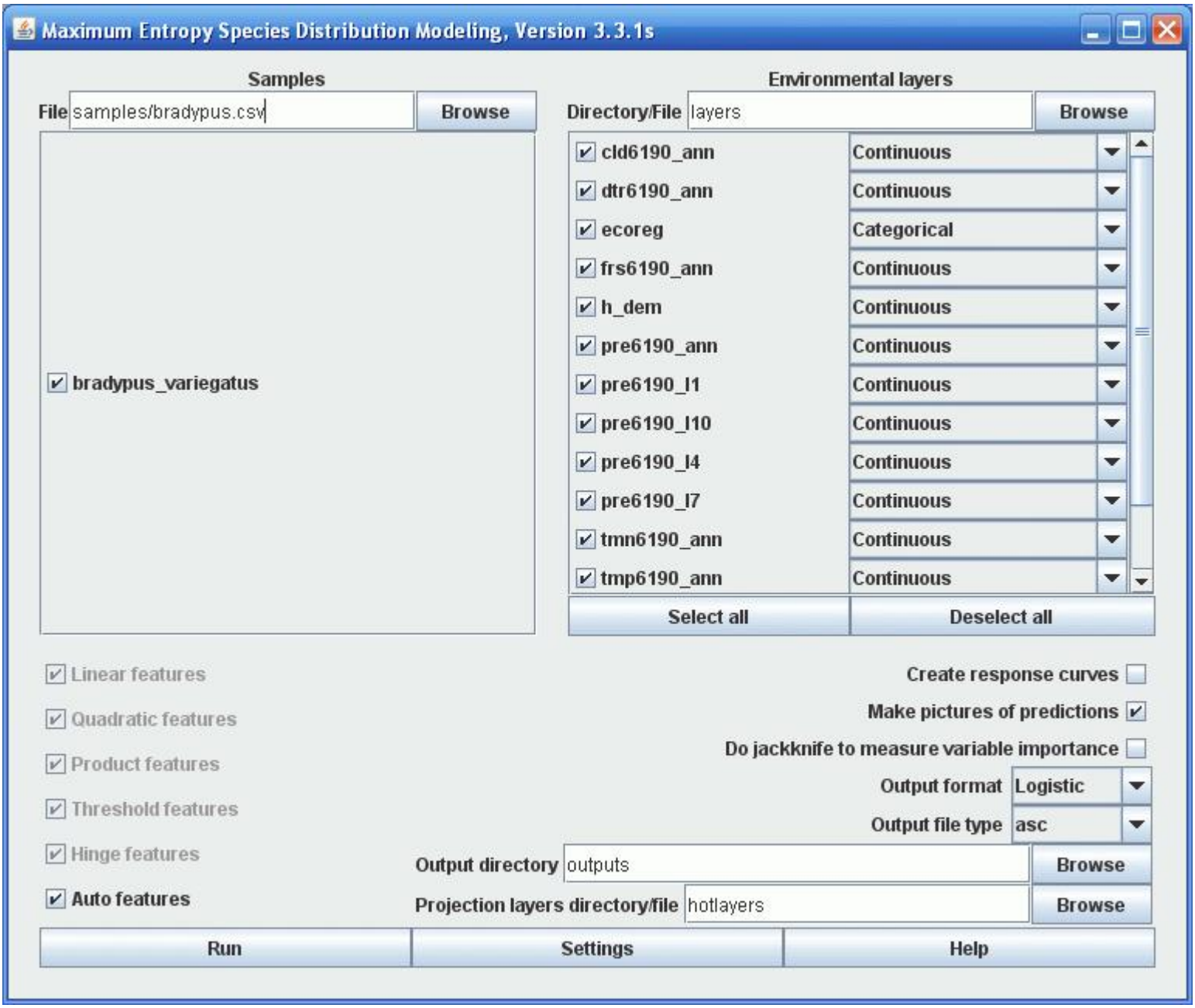


Склонность к переобучению увеличивается с ростом сложности модели. Что бы увидеть сильно "переученную" модель, попробуйте сначала установить множитель равным очень небольшой величине (например 0.01) со стандартным набором функций. Потом попробуйте тоже значение множителя с линейными и квадратичными.

Предсказание

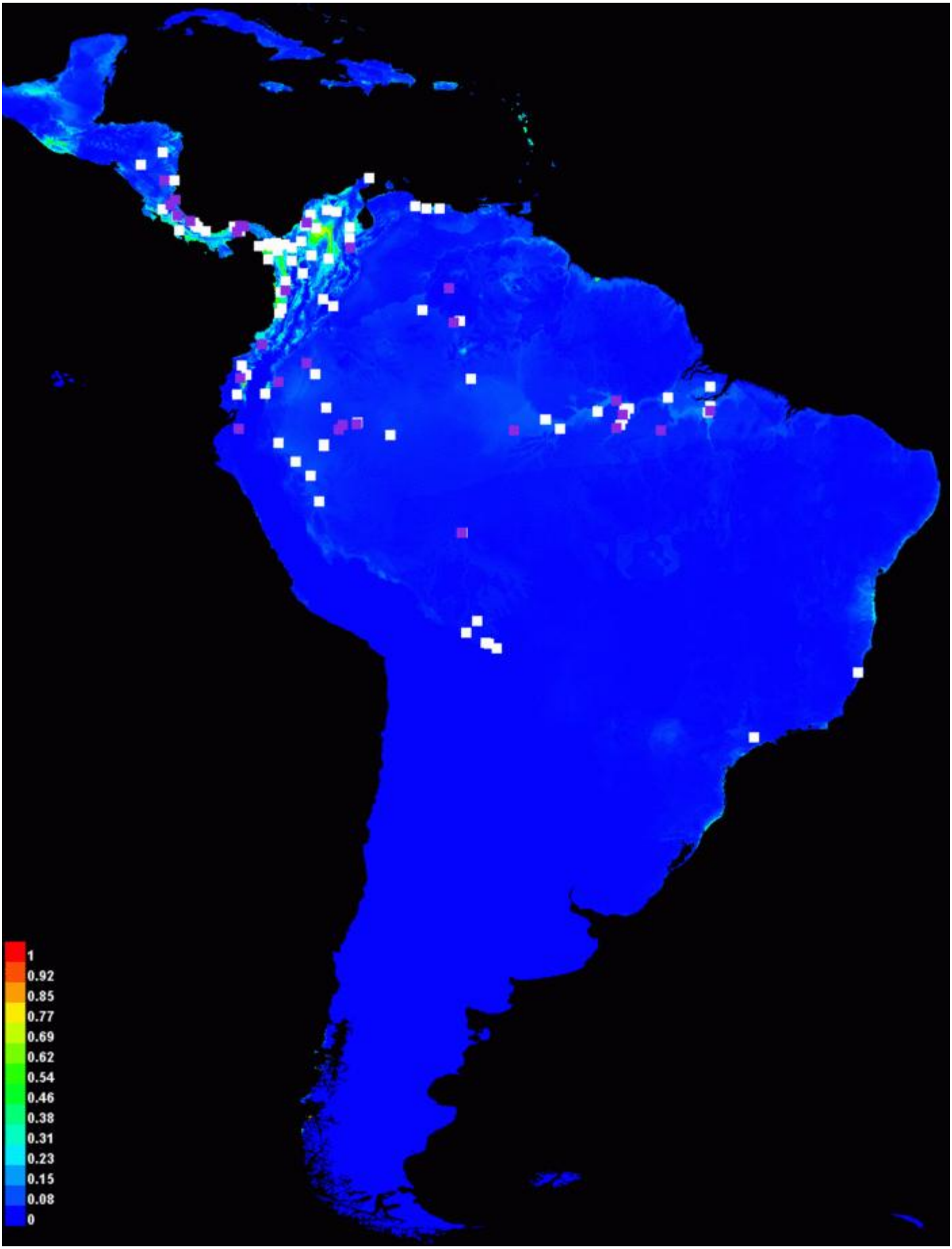
Модель, созданная на одном наборе слоёв (или файле SWD) может быть “спроецирована” путем ее применения к другому набору слоёв (или файлу SWD). Проецирование может понадобиться в ситуациях когда, например, моделируется распределение видов в меняющихся климатических условиях или для оценки инвазивного риска, когда модель нативного распределения инвазивного вида применяется для новой территории, или просто для оценки модели в наборе тестовых локаций для дальнейшего статистического анализа. Здесь мы применим проецирование для простого предсказания изменения климата, чтобы оценить трудности с которыми придется столкнуться при создании надежных моделей распределений в рамках подобных процессов.

Папка “hotlayers” содержит те же самые переменные среды, что и папка “layers” с двумя отличиями: значения переменной среднегодовой температуры (tmp6190_ann.asc) увеличены на 30, что значит равномерное (по всей поверхности) повышение температуры на 3 градуса Цельсия, в то время как значения переменной максимальных температур (tmx6190_ann.asc) увеличены на 40, т.е. повышение температуры на 4 градуса Цельсия. Эти отличия передают очень упрощенную оценку будущего климата с увеличенной средней температурой и с большой температурной изменчивостью, но без изменения в количестве осадков. Чтобы испытать модель "Bradypus" на этом новом климате, укажите программе файл выборки и современные переменные среды, используя либо грид либо формат SWD, а так же укажите путь к папке “hotlayers” в "Папка с прогнозными данными" (Projection Layers Directory) как показано ниже.

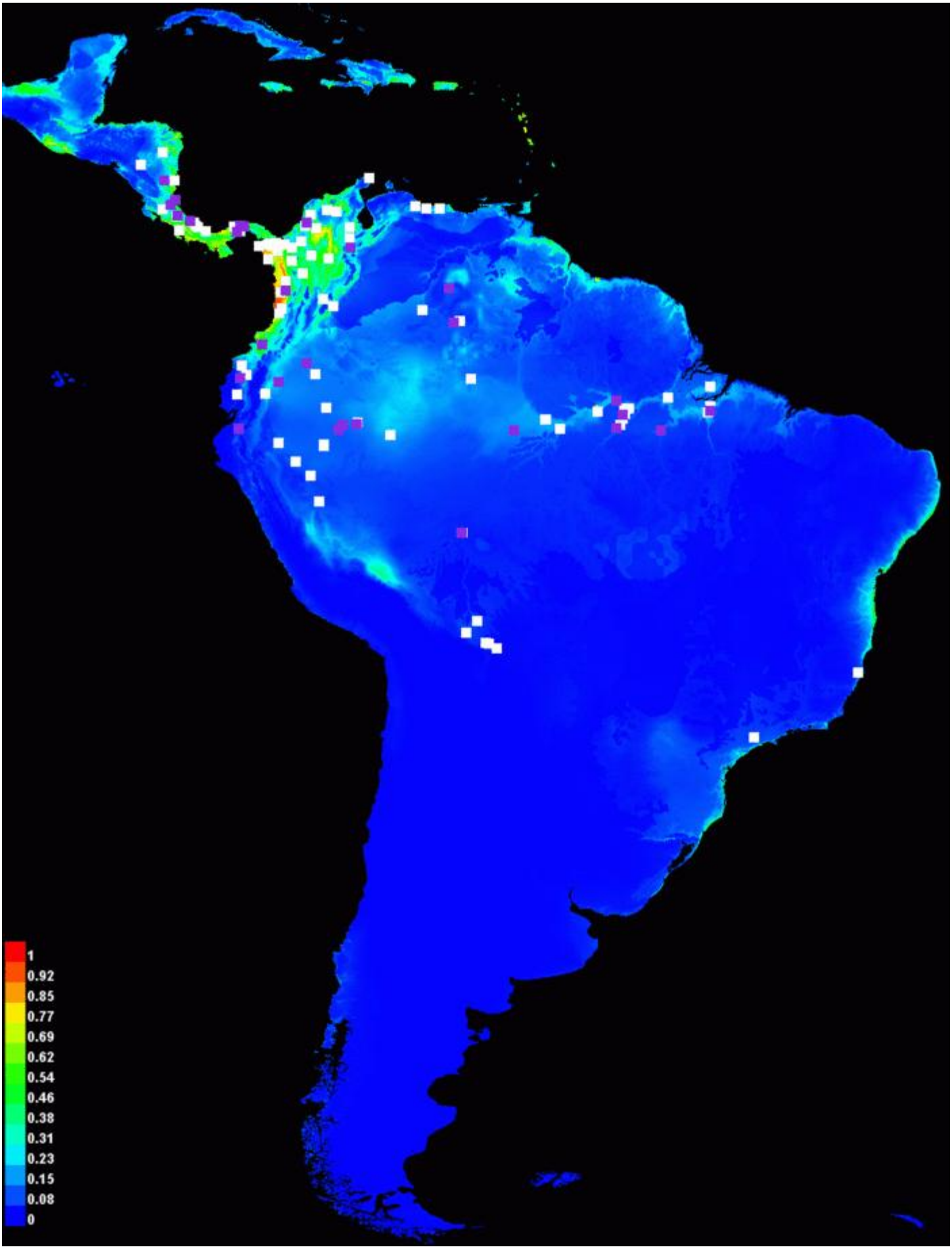


Папка с слоями прогнозов (или SWD) должна содержать переменные под теми же именами, что и переменные использованные для тренировки модели, но описывающие иные условия (например, другой географический район или иные климатические модели). И для тренировки модели и для прогнозирования имя каждой переменной указано в заголовке табличной колонки (при использовании SWD) или имя файла без расширения .asc (если указывается папка с гридами).

После того, как вы нажмете на кнопку “Run”, модель проведет тренировку на переменных среды соответствующих современным климатическим условиям, в затем проецирует результат на ascii-гриды из папки “hotlayers”. Результирующий ascii-грид будет назван “bradypus_variegatus_hotlayers.asc”, и, как правило, имя папки добавляется к названию вида для того, чтобы отличить его от стандартной, не прогнозной, версии. Если будет отмечено “make pictures of predictions”(создать прогнозное изображение), картинка прогнозной модели появится в “bradypus.html”. В нашем случае, был создано такое изображение:

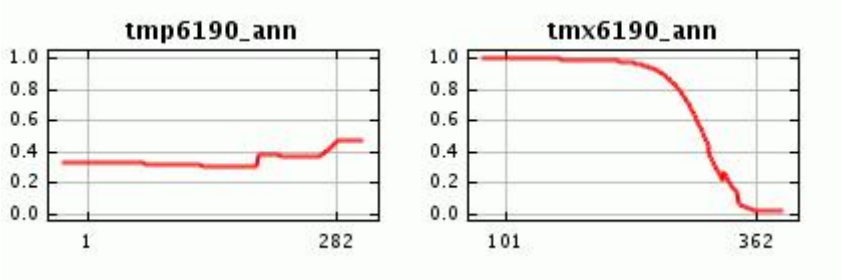


Хорошо видно, что прогнозная вероятность присутствия значительно ниже в условиях более теплого климата. Предсказание, безусловно, зависит от параметров модели, которую мы проецируем. Если мы используем только петлевые и категорийные объекты вместо отмеченных по умолчанию, прогнозное распространение гораздо существенней:

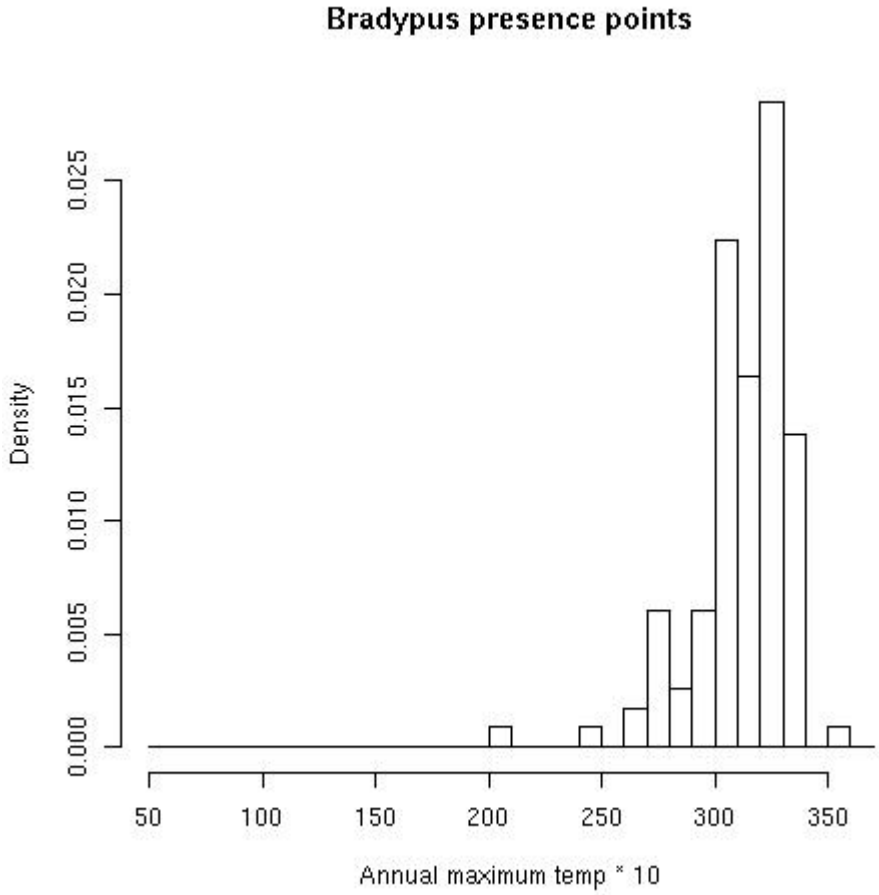


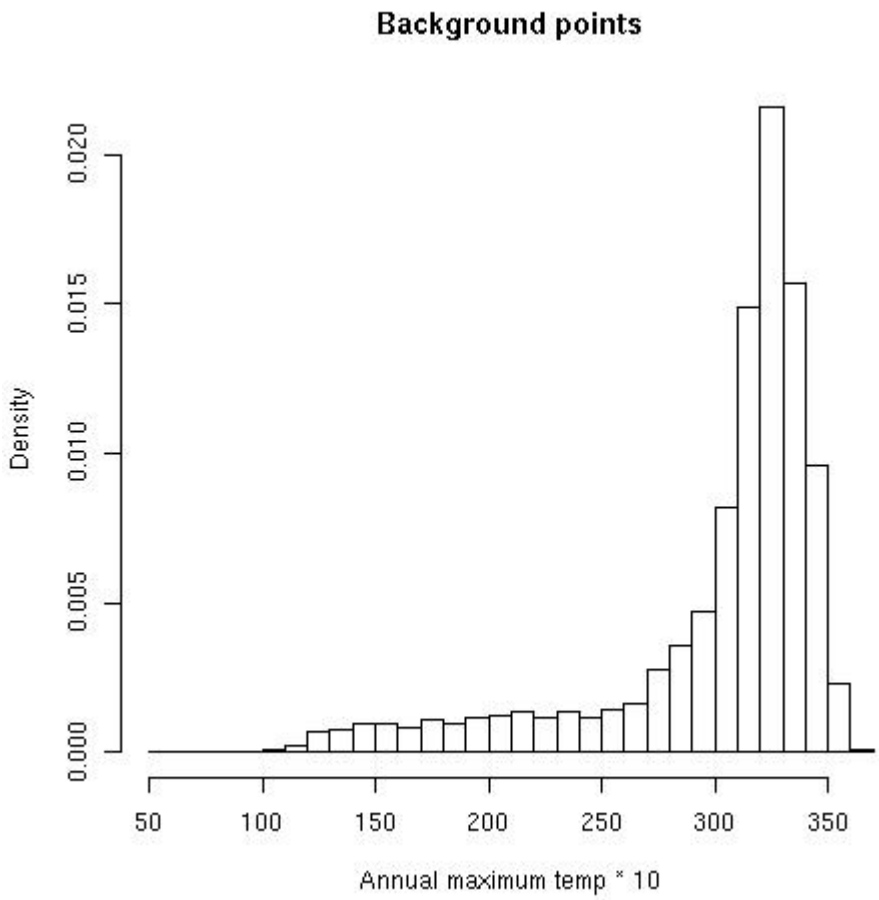
Две различные модели, которые очень похожи в области, использованной для тренировки, могут сильно отличаться, если их спроецировать в новую географическую область или в новые климатические условия. Это особенно хорошо заметно, если присутствуют скореллированные переменные, которые могут различными способами применить сходные по виду модели, поскольку корреляция между переменными может меняться в области на которую вы производите проекцию.

Адекватно ли полученное прогнозное уменьшение *Bradypus* в измененных климатических условиях? Если мы посмотрим на краевой отклик для модели с функциями отмеченными по умолчанию, мы увидим, что максимум температур оказывает гораздо более сильное влияние на предсказание:



Глядя на гистограмму максимальных температур в местах известных встреч *Bradypus*, мы увидим, что большинство находок (около 80%) расположено в областях с максимумами температур между 30 и 34 градусами Цельсия. Только один из них был найден в более теплой области, в то время как значительная часть предпочитала значения между 34 и 35 градусами.





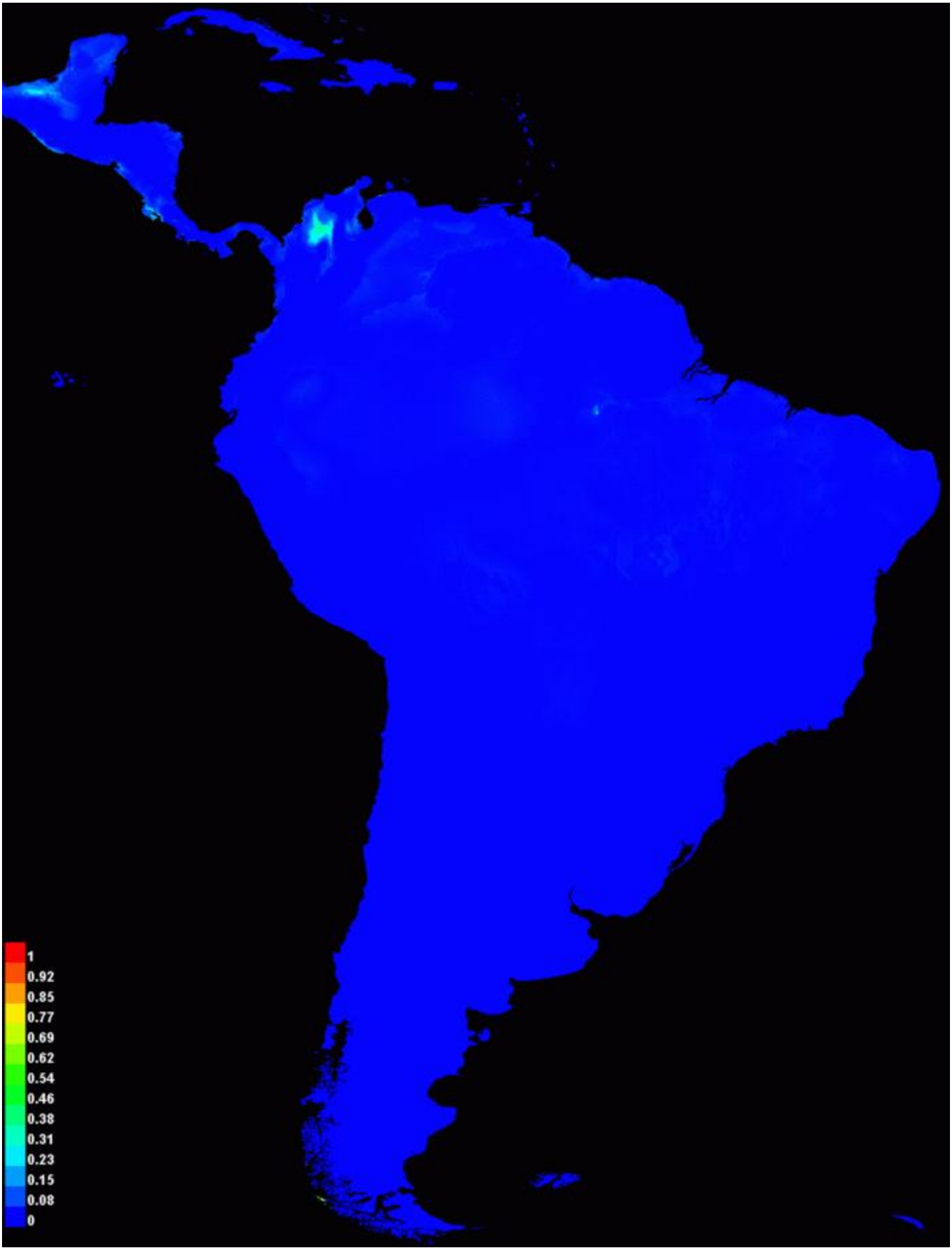
Согласно нашему климатическому прогнозу, все 80% мест находок *Bradypus*, которые в настоящих условиях имеют температуру выше 30 градусов, потеплеют примерно до максимума температур в 34 градуса. Поэтому логично предположить, что такие места больше не будут подходить для обитания *Bradypus*, и он не выживет на большей части территории своего нынешнего расселения. Отметим, что трудно делать какие-то выводы о том, почему такие условия ему не подходят: может быть *Bradypus* нетолерантен к жаре или может быть повышенный максимум температур спровоцирует пирогенную замену тропического леса пожароустойчивыми древесными видами, уничтожив, таким образом, оптимальное для *Bradypus* местообитание. Для дальнейшего выяснения будущего *Bradypus* в новых климатических условиях, мы можем провести физиологическое изучение толернатности вида к жаре или изучить экологию пожаров на границах тропического леса в районе исследований.

Примечание: две приведенные выше гистограммы - полезный инструмент для получения новых сведений о ваших данных. Они сделаны в R с использованием следующих команд:

```
swdPresence <- read.csv("swd/bradypus_swd.csv")
hist(swdPresence$tmx6190_ann, probability=TRUE, breaks=c(5:37*10), xlab="Annual maximum
temp * 10", main="Bradypus presence points")
swdBackground <- read.csv("swd/background.csv")
hist(swdBackground$tmx6190_ann, probability=TRUE, breaks=c(5:37*10), xlab="Annual
maximum temp * 10", main="Background points")
```

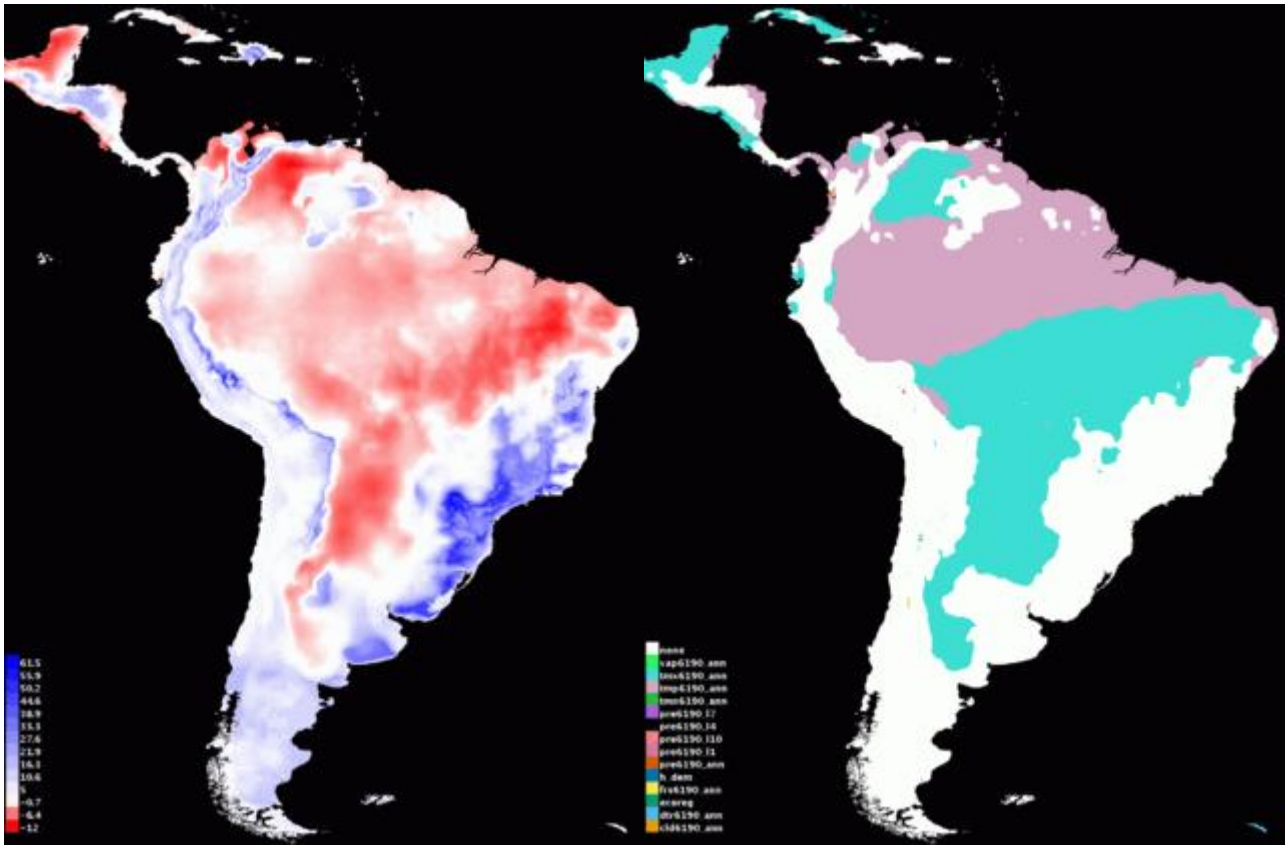
Из этих гистограмм видно, что *Bradypus* иногда может переносить высокие температуры, о чем свидетельствует единственная находка в зоне с температурным максимумом в 35 градусов. С другой стороны, крайне мало находок в зонах 36 и выше градусов, так что у нас нет доказательств или опровержений того, что *Bradypus* может выдерживать даже более высокие температуры, которые будут характерны для предсказанного климата. Это называется проблемой условий нового климата: во время прогнозирования переменные предиктора могут заимствовать значения вне диапазона, отраженного во время тренировки модели. Главным способом устранить эту проблему может “clamping” (слияние), благодаря которому переменные вне тренировочного диапазона будут условно в него введены. Этот эффект может быть виден на кривых зависимости, описанных выше, так как зависимость сохраняется неизменной вне тренировочного диапазона. После создания прогноза, Максент создаст изображение, которое покажет, где слияние имело

наибольший эффект. Прогнозная модель *Bradypus* созданная со всеми видами функций дает такую картину слияния, где переменные представляют абсолютную разность между предсказаниями с и без слияния.



Очевидно, что сведение имело небольшой эффект в этом случае - в частности, кривая зависимости для максимума температур выше показывает, что предсказания уже были выровнены в диапазоне около нуля и в "горячем диапазоне" шкалы, так что сведение мало на них повлияло.

Мы так же сравнили переменные среды для прогноза с теми, что были использованы для тренировки модели. После сведения, мы можем увидеть два таких изображения:



Картинка слева - это мультивариантная поверхность подобия (MESS), описанная в Elith *et al.*, Methods in Ecology and Evolution (Методы в Экологии и Эволюции), 2010. Она демонстрирует насколько сходна каждая точка в *hotlayers* с условиями представленными во время тренировки модели. Отрицательные значения (показанные красным) указывают на новый климат, т.е. значения *hotlayers* не совпадают со значениями *layers*. Указанное значение - минимум по предикторам, показывает насколько отлично значение точки (в сравнении с настоящим временем) выраженное как доля значения прогнозной точки по сравнению со значением этой же точки в *layers*. Положительные значения (голубые) похожи на значения BIOCLIM, а индекс равный 100 означает, что точка отнюдь не новая, в том смысле, что все ее значения в *hotlayers* равны срединным значениям *layers*. Картинка справа показывает несхожие переменные (MoD), и как мы и предполагали, условия нового климата в *hotlayers* из-за средних значений температур (сиреневый, в основном к северу от Амазонки) или максимуму температур (сине-зеленый, протянувшийся почти до самого юга Амазонки) находятся вне тренировочного температурного диапазона.

Маска

Переменная "Маска" может быть полезной, если вы хотите тренировать модель в пределах подмножества конкретного района. Например, мы можем тренировать модель *Bradypus* на данных по встречам в Центральной Америке, а затем экстраполировать модель на всю Южную Америку. Чтобы это сделать нужно создать новую переменную "предиктор" (назовем ее, к примеру, mask.asc) с такими же линейными параметрами, размерами пикселя и проекцией как и переменные среды, но содержащую константное значение (скажем, 1) в области Центральной Америки и no-data (отсутствие значений) в остальных зонах. Эта перменная-маска помещается в ту же папку, что и переменные среды и используется таким же образом как и они. Поскольку значения ее постоянны, она никогда не используется как модель, но участки no-data ограничат тренировку модели областью Центральной Америки.

Чтобы экстраполировать результат на Южную Америку мы создадим новую папку, содержащую копии переменных среды вместе с новой маской (так же названной mask.asc), которая равняется единице по всей территории Южной Америки и не имеет пустых значений. Эта новая папка для Maxent'a указывается в качестве параметра “projection layers”.

Ошибка предвзятости выборки

По умолчанию, при использовании Maxent, мы предполагаем, что частоты встречаемости нашего вида это непредвзятые, случайные примеры из распределения вида. Это правило легко нарушается, если, к примеру, выборка собрана в более легкодоступных местах (более близких к дорогам или населенным пунктам). Если вам кажется, что распределение ваших данных предвзято и у вас есть ясное представление того, как распределяется сложность пространственного сбора ваших данных, то вы можете предоставить Maxent "bias grid" - грид предвзятости, который в дальнейшем будет использован для исправления модели. Этот грид предвзятости должен иметь те же линейные параметры, размер пикселя и проекцию, что и переменные среды и значения в нем должны быть больше нуля или пустыми. Значения указывают на относительную сложность сбора данных, т.е. если два пиксела имеют значения 1 и 2, это означает, что вероятность посещения клетки 2 в два раза выше, чем клетки 1. Заметьте, что этот грид показывает а приорную вероятность выборки, а не то где она была сделана в действительности.

Дополнительные инструменты для командной строки

Файл Maxent jar содержит инструментарий, к которому можно получить доступ из командной строки. Для пользователей Microsoft: описанные здесь инструменты могут быть задействованы через командный файл, например, maxent.bat. В качестве альтернативы, Start->run->cmd запускает командную оболочку для использования команд в интерактивном режиме; cygwin (есть в свободном доступе в сети) - это хорошая альтернатива с гораздо более мощной оболочкой и встроенными сервисами unix.

Быстрое отображение файла грид

Файлы грида представленные в форматах .asc, .grd и .mxe и реже в формате .bil, могут быть просмотрены при помощи следующих команд:

```
java -mx512m -cp maxent.jar density.Show имя файла
```

Для всех команд описанных ниже, вам может понадобиться добавить путь в файл maxent.jar и/или имя файла, который вы хотите увидеть. Например, вы можете ввести:

```
java -mx1000m -cp C:\maxentfiles\maxent.jar density.Show C:\mydata\var1.asc
```

Есть опционные варианты просмотра, требующие введения дополнительных параметров (сразу после density.Show):

-s sampleFile вывод файла находок в виде белых точек

-S speciesname указывает какой вид отмечен в файле находок, отображенном точками

-r radius регулирует размер белых и красных точек в зависимости от частоты встреч

-L убирает легенду

-o пишет изображение в файл .png

С помощью пары фокусов вы можете запустить просмотр под Windows, просто кликнув на .asc, .grd или .mxe файл. Для этого нужно создать командный файл, который будет называться, к примеру, showFile.bat, с единственной строкой в нем:

```
java -mx512m -cp "c:\maxentfiles\maxent.jar" density.Show %1
```

а затем связать файлы форматов .asc, .grd или .mxe с ним через Windows Explorer (ака "Мой Компьютер"), Tools->Folder Options->File Types... Вам может понадобиться сделать командный файл исполняемым: кликните на него правой клавишей и следуйте указаниям.

Создание файла SWD

Чтобы создать SWD-файл из не-SWD:

```
java -cp maxent.jar density.Getval samplesfile grid1 grid2 ...
```

где samplesfile это .csv файл встреч, а grid1, grid2, etc. это гриды в .asc, .mxe, .grd или .bil. Результирующий файл пишется в папку "standard output", что значит, он появится в командном окне. Чтобы записать результат в файл используйте команду "redirect":

```
java -cp maxent.jar density.Getval samplesfile grid1 grid2 ... > outfile
```

Если все гриды в одной папке, вы можете избежать необходимости вносить их списком, используя команду "wildcard":

```
java -cp maxent.jar density.Getval samplesfile directory/*.asc ... > outfile
```

так как wildcard (*) разворачивается в список всех файлов по адресу.

Создание фонового SWD файла

Чтобы получить равномерную выборку фоновых случайных точек в зоне исследований:

```
java -cp maxent.jar density.tools.RandomSample num grid1 grid2 ...
```

где "num" это количество фоновых точек.

Расчет AUC

Следующей командой:

```
java -cp maxent.jar density.AUC testpointfile predictionfile
```

вычисляется AUC фонового присутствия, где точки находок это testpointfile, а фоновые точки случайным образом отбираются из predictionfile. Testpointfile это файл .csv (который может быть при желании переведен в SWD), а predictionfile это грид, обычно представленный результатом модели распространения вида.

Прогноз

Этот инструмент позволит вам применять ранее вычисленные модели Maxent к новым наборам данных:

```
java -cp maxent.jar density.Project lambdaFile gridDir outFile args
```

Где lambdaFile это .lambdas файл описывающий модель Maxent, а gridDir это папка с гридами для всех переменных предикторов, описанных в файле .lambdas. В качестве альтернативы, gridDir может быть файлом .swd. При желании, args может содержать любые ключи, распознаваемые Maxent - например, ключ "grd" создаст грид density.Project в формате .grd.

Преобразование файлов

Чтобы преобразовать все гриды в папке в другой формат:

```
java -cp maxent.jar density.Convert indir insuffix outdir outsuffix
```

где indir и outdir это папка, а insuffix и outsuffix в форматах .asc, .mxe, .grd или .bil.

Анализ результатов MaxEnt в R

Maxent производит несколько результирующих файлов при каждом запуске. Некоторые из этих файлов могут быть импортированы в другие программы, если вы хотите провести свой собственный анализ. Ниже мы покажем, как использовать эти результаты в свободной программе статистического анализа R: эта секция для

тех, кто имеет опыт ее использования. Мы возьмем следующие два файла, произведенные Maxent:

bradypus_variegatus_backgroundPredictions.csv

bradypus_variegatus_samplePredictions.csv

Первый файл создается, когда включена опция “writebackgroundpredictions” (писать промежуточные результаты) либо с введением ключа в командную строку, либо при выборе на панели настроек Maxent’а. Всегда создается второй файл. Убедитесь, что у вас есть тестовые данные (например, установив случайный показатель выборки в 25 процентов); мы будем оценивать результирующие файлы Maxent, используя те же самые тестовые данные, что использовал сам Maxent. Сначала мы запустим R, а затем установим несколько дополнительных пакетов (если мы впервые их задействуем), а затем загрузим их, напечатав (или вставив):

```
install.packages("ROCR", dependencies=TRUE)
install.packages("vcd", dependencies=TRUE)
library(ROCR)
library(vcd)
library(boot)
```

В этой секции мы будем использовать голубой текст, чтобы показать код и команды R и зеленый текст, чтобы показать результаты R (*в переводе цвета не соблюдаются, примечание переводчика*). Далее мы меняем папку на ту, где находятся результаты Maxent, например:

```
setwd("c:/maxent/tutorial/outputs")
```

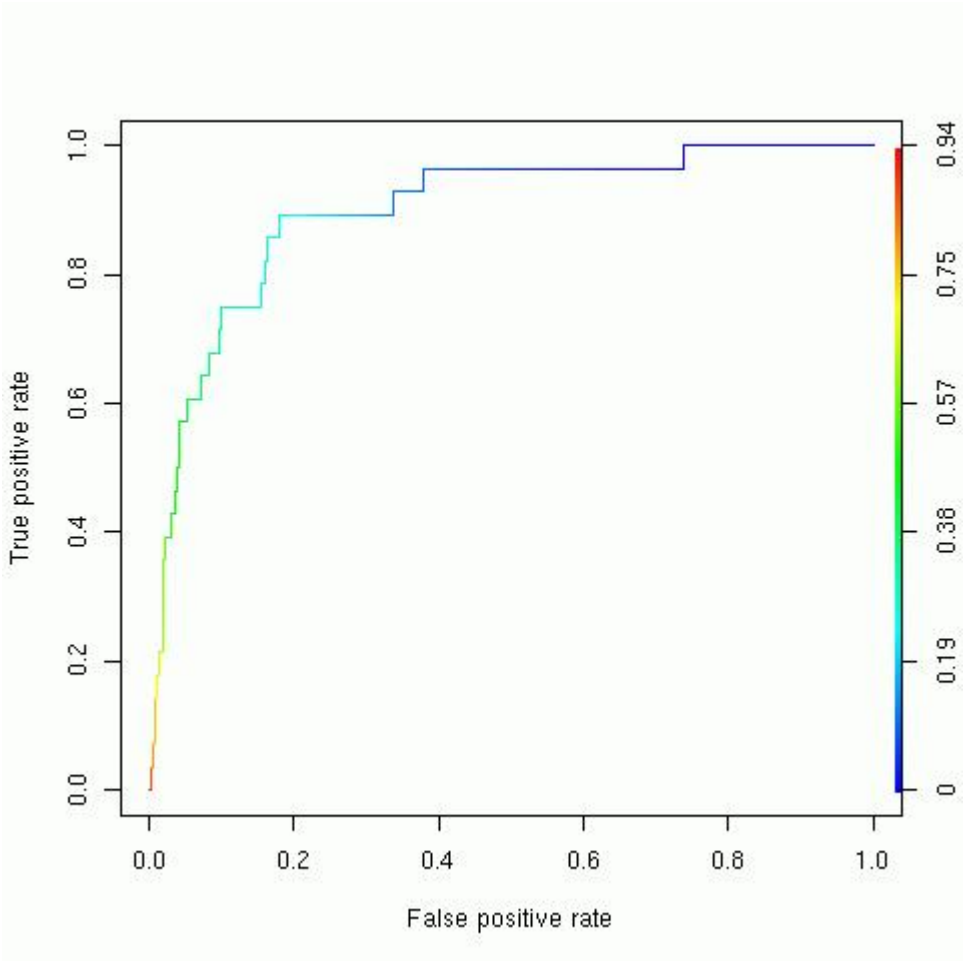
а затем мы вводим предсказания Maxent в местах находок и в фоновых точках и извлекаем нужные нам колонки:

```
presence <- read.csv("bradypus_variegatus_samplePredictions.csv")
background <- read.csv("bradypus_variegatus_backgroundPredictions.csv")
pp <- presence$Logistic.prediction </nowiki># взять колонку с предсказаниями
testpp <- pppresence$Test.or.train=="test" # выбрать только тестовые точки
trainpp <- pppresence$Test.or.train=="train" <nowiki># выбрать только тренировочные
точки
bb <- background$logistic
```

Теперь мы можем перевести значения предсказаний в формат для ROCR, пакет использующийся для ROC анализа и генерации ROC-кривой.

```
combined <- c(testpp, bb) </nowiki># собрать в единый вектор
label <- c(rep(1,length(testpp)),rep(0,length(bb))) # подписи: 1=присутствие,
0=случайная
pred <- prediction(combined, label) <nowiki># подпись предсказания
perf <- performance(pred, "tpr", "fpr") # верно / ложно положительные, для ROC кривых
plot(perf, colorize=TRUE) <nowiki># показать ROC кривую
performance(pred, "auc")@y.values1 # вычислить AUC
```

Команда для построения графика дает следующий результат:



команда “performance” показывает AUC значение 0.8677759, что совпадает с AUC отчетом у Maxent. Далее, в качестве примера теста доступного в R, но не в Maxent, мы запустим рэндом-анализ стандартного отклонения AUC.

```
AUC <- function(p,ind) {
pres <- pind
combined <- c(pres, bb)
label <- c(rep(1,length(pres)),rep(0,length(bb)))
predic <- prediction(combined, label)
return(performance(predic, "auc")@y.values1)
}

b1 <- boot(testpp, AUC, 100) # сделать 100 AUC вычислений методом рэндомизации
```

Получаем результат:

```
ORDINARY NONPARAMETRIC BOOTSTRAP

Call :
boot(data = testpp, statistic = AUC, R = 100)

Bootstrap Statistics :

original bias std. error
t1* 0.8677759 -0.0003724138 0.02972513
```

и мы видим, что вычисление стандартной ошибки методом рэндомизации (0.02972513) близок к стандартной ошибке вычисленной Maxent (0.028). Метод рэндомизации так же может быть использован для оценки надежности интервалов для AUC:

```
boot.ci(b1)
```

получаем следующую оценку - см. секцию R ссылки в конце этого руководства с ресурсами, которые помогут с определениями и сравнениями для этих результатов.

```
 Intervals :
Level Normal Basic
95% ( 0.8099, 0.9264 ) ( 0.8104, 0.9291 )

Level Percentile BCa
95% ( 0.8064, 0.9252 ) ( 0.7786, 0.9191 )
```

Тот, кто знаком с использованием метода рэндомизации заметит, что мы используем в нем только значения находок. Мы можем так же использовать и фоновые показатели, но эти результаты не изменят картины, учитывая очень большие значения фоновых показателей (10000).

В качестве последнего примера мы рассмотрим вычисление биноминальной статистики и Cohen's Kappa на примере правил для пороговых значений. Для начала, следующий код вычисляет Kappa для порога заданного минимальным значением предсказаний:

```
confusion <- function(thresh) {
    return(cbind(c(length(testpptestpp>=thresh),
length(testpptestpp<thresh)),c(length(bbbb>=thresh), length(bbbb<thresh))))
}

mykappa <- function(thresh) {
    return(Kappa(confusion(thresh)))
}

mykappa(min(trainpp))
```

что возвращает нам значение 0.0072. Если мы хотим использовать порог, который минимизирует сумму чувствительности и специфичности тестовых данных, мы можем сделать следующее, используя верно положительные и ложно положительные значения из объекта "performance", который был использован выше для построения кривой ROC.

```
fpr = perf@x.values1
tpr = perf@y.values1
sum = tpr + (1-fpr)
index = which.max(sum)
cutoff = perf@alpha.values1index
mykappa(cutoff)
```

Это дает нам значение kappa в 0.0144. Для определения биноминального распределения для этих двух пороговых значений, мы можем запустить:

```
mybinomial <- function(thresh) {
conf <- confusion(thresh)
trials <- length(testpp)
return(binom.test(conf11, trials, conf1,2 / length(bb), "greater"))
}

mybinomial(min(trainpp))
mybinomial(cutoff)
```

Это возвращает значения p=5.979e-09 и 2.397e-11 соответственно, что несколько больше значений для p от Maxent. Причина этой разницы - количество тестовых образцов превышающее 25, порог, после которого Maxent использует нормальную аппроксимацию для вычисления биномиальных значений p.

R ссылки

Некоторые хорошие вводные материалы по R могут быть найдены в:

http://spider.stat.umn.edu/R/doc/manual/R-intro.html, и некоторые другие страницы на этом сайте.

http://www.math.ilstu.edu/dhkim/Rstuff/Rtutor.html

Обсудить в форуме Комментариев — 4

Последнее обновление: 2014-05-14 23:54

Дата создания: 17.03.2013
Автор(ы): Максим Дубинин , Юлия Калашникова