embed3GL User's Manual

General

The program embed3GL is intended for phylogenetic studies of joint genes and species evolution. The program allows the user to solve three phylogenetic tasks using an original algorithm of polynomial (cubic) complexity [1]. All the three tasks uses common input data including:

a) *the rooted species tree*. Initially it is a binary one, but during the input data preparation some additional nodes are inserted on the tree edges (also referred to as *tubes*) to split the tree into temporal slices so that all leaves (extant species) are in the same, deepest, slice. The number of additional nodes on a tube is specified as the length of that tube: no nodes are added if the length equals 1 (or not specified), the length of 2 means that one node is added on this tube, and so on.

b) *the set of rooted gene trees*. Those trees are mostly binary ones, but may contain a number of polytomous nodes. The label of each gene (i.e. leaf of a gene tree) shall enable unambiguous identification of the species (i.e. leaf of the species tree) this gene pertain to. In addition, genes can be specially labeled to calculate the Task 3 statistics separately for such labeled genes and their common ancestors.

The program allows the user to solve the following three tasks.

Given a gene tree, Task 1 involves the calculation of the cost of that tree embedding into the species tree. The task is solved independently for each gene tree with the results being costs of embedding for each tree and the total cost of embedding over whole input set of trees. The side effect of the Task 1 solution is a binarization (binary resolution) of gene trees containing polytomous nodes.

Task 2 is solved on the basis of working data obtained from Task 1. As a result, for each gene tree, the optimal (in the sense of minimum cost) scenario of its embedding into the species tree is built. The scenario has the form of a tree of evolutionary events that contains both unary and binary edges.

Задача 3 состоит в вычислении ряда статистик для всего исходного набора деревьев генов и решается после бинаризации всех политомических деревьев. При этом используются задаваемые пользователем дополнительные данные двух видов:

> I-тип - фиксированное множество типов эволюционных событий (например, потери, возникновения, дупликации, переносы);
>
> T-тип - множество вершин деревьев генов, у которых все листья-потомки помечены определенным образом в одном или нескольких деревьях генов (например, "множество предков рибосомных генов").

Результатом решения Задачи 3 являются представленные в табличной форме две функции: f(I,x) – мат. ожидание числа событий типа I в трубе (ребре) x дерева видов, и g(I,T) – мат. ожидание числа событий типа I, происшедших с ребрами типа T.

Программа embed3GL написана на C/C++ и имеет интерфейс командной строки. Программа автоматически использует распараллеливание, если при запуске обнаружена среда MPI версии 1.2 и выше. Текст программы переносим, и после соответствующей перекомпиляции может

использоваться в среде ОС Windows 32/64-bit, Linux, Unix, MacOS. Исходный код программы предоставляется бесплатно на условиях лицензии GNU General Public License (GPL) версии 3. Кроме того, предлагаются двоичные исполняемые модули для работы в однопроцессорном режиме в среде Windows (32- и 64-битная версии). Все варианты для загрузки содержат контрольные примеры. Сообщения об ошибках направляйте Рубанову Л.И. (rubanov@iitp.ru).

Установка и запуск embed3GL в среде Windows

Исполняемые модули программы не требуют установки. Загруженный с сайта архив ZIP необходимо просто распаковать в выбранную пользователем папку. В среде Windows 32-bit может работать только 32-битный вариант исполняемого модуля; в среде Windows 64-bit работоспособны оба варианта программы – 32- и 64-битный.

Примечание. Быстродействие обоих вариантов приблизительно одинаковое, однако 64-битный исполняемый модуль расходует больше оперативной памяти. С другой стороны, 32-битный вариант программы способен использовать не более 2 Гб оперативной памяти, что ограничивает возможности решения задач большой размерности. 64-битный вариант программы не имеет такого ограничения, и способен задействовать всю имеющуюся оперативную память компьютера. Поэтому в среде Windows 64-bit при имеющейся памяти 3 Гб и выше рекомендуется использовать 64-битный исполняемый модуль, а в среде Windows 32-bit или при объеме оперативной памяти 2 Гб и менее – 32-битный.

После распаковки загруженного архива в выбранную папку, например, D:\embed3GL, выполните следующие действия:

1. Запустите командный процессор Windows (Пуск > Выполнить > cmd).

2. Перейдите в папку с программой, для чего выполните команды:
   d:
   cd d:\embed3GL

3. Введите одну из команд:
   embed3GL -h (если используется 32-битный вариант программы), или
   embed3GLx64 -h (если используется 64-битный вариант программы).

4. При успешном запуске программа выдаст на экран краткую справку о параметрах командной строки. Если при запуске операционная система выдает сообщение, «Программа не может быть выполнена» или аналогичного содержания, то обычно это означает, что в данной системе не установлена свободно распространяемая библиотека Microsoft для Visual C++ 2008 SP1. В этом случае необходимо загрузить с сайта Microsoft соответствующую (32 или 64 бит, а также языку системы) версию библиотеки и установить ее.
   Примечание: На момент написания данного руководства библиотеки располагались на сайте по адресам:
   32 bit: http://www.microsoft.com/en-us/download/details.aspx?id=5582
   64 bit: http://www.microsoft.com/en-us/download/details.aspx?id=2092
   Если адреса изменились, ищите “Microsoft Visual C++ SP1 Redistributable Package (x86/x64)”.

5. После успешного выполнения п. 4 запустите программу на контрольном примере командой **embed3GL** (32-битная версия) или **embed3GLx64** (64-битная версия) без параметров. Время

счета обычно не превышает минуты; по окончании в подпапке Test появится файл протокола **embed3GL.log**. Программа готова к использованию.

Компиляция и запуск embed3GL в среде Linux

Исходные тексты программы и примеры для проверки ее работоспособности доступны для загрузки с сайта в виде единого архива **embed3GL-sc-X.Y.Z.tgz**, где X.Y.Z – номер версии. После загрузки и распаковки необходимо выполнить следующие команды из каталога программы:

1. **make**
   Будет выполнена компиляция и сборка программы, в результате в каталоге программы будет создан исполняемый файл embed3GL.
   Примечание: Имеющийся в составе дистрибутива файл **makefile** разработан в предположении, что компиляция программ на C++ производится компилятором **gcc** или **icc**, и для совместной компиляции C++ с библиотеками MPI используется альтернативное имя компилятора **mpiCC**. При необходимости следует внести изменения в **makefile**. Перед повторным запуском **make** рекомендуется выполнить команду **make clean**.

2. **make test**
   Выполняется обработка первого контрольного примера в однопроцессорном режиме. Пример использует единственное дерево генов, и время счета не превышает минуты. В результате выполнения будет создан файл протокола **./Test/embed3GL.log** размером свыше 12 Мб, который демонстрирует возможные виды выдаваемой информации.

3. **make mpitest**
   Выполняется обработка второго контрольного примера в мультипроцессорном режиме на 16 процессорах. В этом примере исходный набор состоит из 1000 деревьев генов; время счета обычно не превышает 5 минут. В результате выполнения будет создан файл протокола **./Test/artif.log**, содержащий только итоговую статистику Задачи 3.

Параметры конфигурации программы

Основная часть параметров программы задается в *файле конфигурации*; кроме того, некоторые параметры могут задаваться или модифицироваться в командной строке запуска embed3GL. Указанные в командной строке значения параметров всегда имеют приоритет перед значениями в файле конфигурации.

По умолчанию файл конфигурации с именем **embed3GL.ini** ищется в каталоге с программой; с помощью опций **-q** (или **-i**) также можно указать любой другой каталог и/или имя.

Файл конфигурации является обычным текстовым файлом и содержит набор предложений вида

параметр = значение

Каждое предложение занимает одну строку. Пустые строки и строки, начинающиеся любым из символов ';' '/' '#', рассматриваются как комментарии. Также комментариями считается часть строки, начинающаяся с символов '//'. Пробелы и знаки табуляции игнорируются (если они не являются частью значения, заключенного в двойные кавычки). Регистр букв в имени параметра не важен, но в значениях – учитывается.

Ниже следует перечень параметров и их допустимых значений, относящихся к текущей версии программы. Порядок указания параметров в файле конфигурации не имеет значения.

Настройки режима работы

OptimumCost = <логическое значение>

Если указано значение *true*, то будет решаться Задача 1, т.е. находиться цена оптимального вложения каждого дерева генов в данное дерево видов. Решение Задачи 1 необходимо для решения двух других задач, поэтому не рекомендуется указывать другое значение этого параметра.

Примечание: Здесь и далее вместо логического значения *true* может указываться *1, on, yes, y, enable*; вместо логического значения *false* – 0, off, no, n, disable.

Embedding = <логическое значение>

Если указано значение *true*, то будет решаться Задача 2, т.е. строиться сценарий оптимального вложения каждого дерева генов в данное дерево видов; в противном случае оптимальный сценарий не строится.

Statistics = <логическое значение>

Если указано значение *true*, то будет решаться Задача 3, т.е. вычисляться заданные статистические характеристики эволюционных событий для всего исходного набора деревьев генов. В противном случае Задача 3 не решается.

MaxEvents = <число>

Задает максимальную степень ветвления эволюционного сценария, т.е. число наиболее вероятных событий (с наименьшими ценами) в каждой точке. Указанное значение сильно влияет на скорость работы программы и потребление памяти.

MemoryCount = <логическое значение>

Если указано значение *true*, то будет подсчитываться объем потребляемой памяти при обработке каждого дерева генов (опционально) и всего исходного набора. Эта информация выводится в протокол работы и помогает планировать расчеты.

Цены эволюционных событий

C_loss = <число> Цена эволюционного события потери гена.

Примечание: Здесь и далее значения цен могут быть как целыми, так и дробными, однако учитываются только первые два знака после запятой.

C_dupl = <число> Цена события дупликации.

C_gain = <число> Цена события приобретения.

C_gain_big = <число> Цена события большого приобретения.

C_sleep = <число> Цена события перехода гена в спящее состояние.

C_tr_with = <число> Цена события горизонтального переноса гена с сохранением.

C_tr_without = <число> Цена события горизонтального переноса гена без сохранения.

K_correction = <число>

Параметр коррекции цен горизонтального переноса в зависимости от степени

ветвления *K* (значение параметра **MaxEvents** выше). Если указано значение 0, коррекция не используется; в противном случае к значениям параметров **C_tr_with** и **C_tr_without** прибавляется величина K_correction * $\log_2 K$.

InitialCost = <число>

Начальное значение цены, используемое при вычислении вероятности событий вместе с показателем степени, указанным параметром **C_exponent** (см.)

C_exponent = <число>

Вместе со значением начальной цены c_0 (параметр **InitialCost**) используется для вычисления относительных вероятностей *K* событий с ценами $c_1, c_2, ..., c_K$ так, чтобы $\Sigma p_k = 1$ и $p_i / p_j = [(c_j - c^* + c_0)/ (c_i - c^* + c_0)]^{C\_exponent}$, где $c^* = \min\{c_k\}$.

Параметры I-типа

Эти параметры задают набор эволюционных событий, для которых должны вычисляться статистики f(I,x) при решении Задачи 3 (I-тип).

I_loss = <логическое значение>

Если указано значение *true*, то I-тип включает событие потери гена.

I_dupl = <логическое значение>

Если указано значение *true*, то I-тип включает событие дупликации.

I_gain = <логическое значение>

Если указано значение *true*, то I-тип включает событие приобретения.

I_gain_big = <логическое значение>

Если указано значение *true*, то I-тип включает событие большого приобретения.

I_sleep = <логическое значение>

Если указано значение *true*, то I-тип включает переход гена в спящее состояние.

I_tr_o = <логическое значение>

Если указано значение *true*, то I-тип включает событие переноса гена из трубы.

I_tr_i = <логическое значение>

Если указано значение *true*, то I-тип включает событие переноса гена в трубу.

Параметры исходных данных

DataDirectory = <имя_каталога>

Определяет каталог, в котором находятся файлы исходных данных и куда будут записываться результаты. Если параметр не указан, используется текущий каталог, откуда запущена программа. Каталог записывается по правилам операционной системы. Если в имени каталога содержатся пробелы или специальные символы, значение параметра необходимо заключить в двойные кавычки.

SpeciesTree = <имя_файла>

Указывает имя файла, содержащего дерево видов в скобочной записи (формат Newick). Имя может включать путь к каталогу. Следует учитывать, что если указан параметр **DataDirectory**, его значение будет добавлено *перед* значением данного

параметра. Если параметр содержит пробелы или специальные знаки, значение необходимо заключить в двойные кавычки.

LeafMinWords = <число>

Параметр определяет минимально необходимое число «слов» в названии вида. Предполагается, что в дереве видов имя каждого листа начинается с названия вида, которое состоит из одного или более слов, разделенных символом подчеркивания (формат Newick запрещает использовать пробел в метке вершины). Если имя листа содержит меньше слов, чем указано данным параметром, это будет воспринято как ошибка. Значение 0 выключает данную проверку.

LeafMaxWords = <число>

Параметр задает максимальное число слов в названии вида. Слова сверх указанного количества не будут считаться входящими в название вида и игнорируются.

Примечание: Значения параметров **LeafMinWords**, **LeafMaxWords** должны быть указаны такими, чтобы обеспечить однозначную идентификацию вида, к которому относится каждый ген в исходном наборе деревьев генов. Предполагается, что во всех деревьях генов имя каждого листа также начинается с названия вида, которое состоит из одного или более слов, разделенных символом подчеркивания. Хотя имена листьев в деревьях генов и видов могут содержать *разную* дополнительную информацию (например, штамм, номенклатурный код, названия таксонов верхних уровней и т.п.), их начальные части в пределах от минимального до максимального числа слов для правильной работы программы должны быть согласованы.

OutgroupName = <символьная_строка>

Параметр указывает имя аутгруппы, которое присутствует в дереве видов как отдельный лист (т.е. «вид»). Наличие аутгруппы в дереве видов обязательно.

GeneTree = <имя_файла>

Указывает имя файла, содержащего одно или более дерево генов в скобочной записи (формат Newick). Каждое дерево (если их более одного) должно начинаться в файле с новой строки. Имя может включать путь к каталогу. Следует учитывать, что если указан параметр **DataDirectory**, его значение будет добавлено *перед* значением данного параметра. Если параметр содержит пробелы или специальные знаки, значение необходимо заключить в двойные кавычки.

Примечание: Данный параметр может встречаться в конфигурации несколько раз. Исходный набор деревьев генов есть объединение деревьев, записанных во всех указанных файлах. Это позволяет разбивать набор исходных деревьев генов на смысловые группы любого размера, вплоть до одиночных деревьев. Очередность обработки деревьев определяется порядком предложений **GeneTree** в файле конфигурации.

TmarkChar = <символ>

Данный параметр определяет, каким символом в деревьях генов будет указываться принадлежность гена к Т-типу. Этот символ добавляется к имени гена (т.е. метке листа в дереве гена) в соответствии с параметром **TmarkAnywhere**, но не считается частью имени.

Примечание: В качестве символа не могут использоваться элементы скобочной записи формата Newick (квадратные и круглые скобки, запятая, двоеточие, точка с запятой, точка, подчеркивание, кавычки, пробелы и символы табуляции). Что касается прочих символов, помимо букв и цифр, следует иметь в виду, что стандартный формат Newick не допускает их использование в метках вершин, поэтому могут возникать трудности при использовании других программ. Например, если использовать символ «звездочка», то дерево не может быть отображено программой TreeView, но будет отображаться программами TreeViewX и Dendroscope.

TmarkAnywhere = <логическое значение>

Если указано значение *true*, то указанный параметром **TmarkChar** символ может стоять в произвольном месте имени гена. В противном случае этот символ должен быть *последним* символом имени гена.

TmarkShow = <логическое значение>

Если указано значение *true*, то в протоколе работы программы имена всех относящихся к Т-типу вершин в каждом дереве видов будут указываться с символом **TmarkChar** (как будто он является частью имени вершины). В противном случае этот символ будет опускаться.

Общие настройки выходного протокола

LogFilename = <имя_файла>

Указывает имя основного файла протокола, куда записываются результаты работы программы embed3GL. Имя может включать путь к каталогу. Следует учитывать, что если указан параметр **DataDirectory**, его значение будет добавлено *перед* значением данного параметра. Если параметр содержит пробелы или специальные знаки, значение необходимо заключить в двойные кавычки.

LogAppend = <логическое значение>

Если указано значение *true*, то существующий файл протокола с указанным именем будет открыт в режиме дозаписи в конец (старое содержимое сохранится). В противном случае ранее записанная информация будет потеряна.

Console = <логическое значение>

Если указано значение *true*, то по ходу работы программы будет проводиться выдача информации о работе на служебную консоль (файл stdout). При работе в пакетном режиме и/или на параллельном кластере это может быть нежелательно, и параметр используется для запрета такой выдачи (даже в корневой ветви).

Console2Log = <логическое значение>

Если указано значение *true*, то информация, предназначенная к выдаче на консоль, будет дополнительно дублироваться в основной файл протокола. В противном случае в протокол будут выданы только затребованные результаты работы программы.

SecondaryCon = <логическое значение>

Данный параметр учитывается только при параллельной работе программы в среде MPI. Он аналогичен параметру **Console**, но относится к системной консоли

вторичных ветвей программы (а не корневой). Включать эту выдачу (т.е. указывать значение *true*) имеет смысл, если доступны системные консоли всех ветвей (например, на кластерах с Windows) и требуется следить за ходом обработки во всех ветвях.

SecondaryLog = <логическое значение>

Данный параметр управляет выдачей протокола работы вторичных ветвей при параллельной работе программы в среде MPI. Если указано значение *false* (обычная ситуация), то только корневая ветвь формирует основной протокол, указанный параметром **LogFilename**, содержащий сводные результаты работы всех ветвей программы. Если указать значение *true*, дополнительно формируются протоколы работы вторичных ветвей с именами вида **_n* (где *n* – номер ветви).

MeanWidth = <число>

Указывает ширину поля (число позиций в строке протокола) для значений мат. ожиданий (т.е. функций f(I,x) и g(I,T)), включая целую и дробную часть и десятичную точку.

MeanPrecision = <число>

Указывает число десятичных знаков после запятой при выдаче значений мат. ожидания.

Параметры выдачи дерева видов

Эта группа параметров управляет выдачей исходного дерева видов в протокол работы embed3GL. В частности, такая выдача позволяет идентифицировать внутренние вершины деревьев, имена которым автоматически присвоены программой.

LogSpeciesTree = <логическое значение>

Если указано значение *true*, то дерево видов будет включено в протокол работы программы; в противном случае дерево видов не выдается.

StreeHeading = <логическое значение>

Если указано значение *true*, то непосредственно перед деревом видов в протокол будет выдана строка заголовка, помогающая его идентифицировать.

StreeLengths = <логическое значение>,<логическое значение>,<логическое значение>

В данном параметре должно быть указано в точности три логические значения:
– если 1-е значение *true*, будет выдана длина корневой ветви дерева;
– если 2-е значение *true*, будут выданы длины внутренних ветвей дерева;
– если 3-е значение *true*, будут выданы длины листовых ветвей дерева видов.
Длины выдаются вслед за символом двоеточия, согласно формату Newick.

StreeLabels = <логическое значение>,<логическое значение>,<логическое значение>

В данном параметре должно быть указано в точности три логические значения:
– если 1-е значение *true*, будет выдано имя корня дерева;
– если 2-е значение *true*, будут выданы имена внутренних вершин дерева;
– если 3-е значение *true*, будут выданы имена листьев дерева видов.

StreeNumbers = <логическое значение>,<логическое значение>,<логическое значение>

В данном параметре должно быть указано в точности три логические значения:

– если 1-е значение *true*, будет выдан номер корневой вершины дерева;
– если 2-е значение *true*, будут выданы номера внутренних вершин дерева;
– если 3-е значение *true*, будут выданы номера листьев дерева видов.
Примечание: Из-за ограничений формата Newick в именах вершин не допускаются никакие разделители, так что если одновременно запросить выдачу имени и номера вершины, то они будут приведены слитно.

StreeFullLabel = <логическое значение>
Если указано значение *true*, то выдается полное имя листа дерева, как оно было указано в исходном файле. В противном случае выдается только та часть имени, которая считается названием вида (т.е. первые **LeafMaxWords** слов).

Настройки протокола для всех деревьев генов

Параметры данного раздела применяются к каждому файлу исходного набора деревьев генов; это следует учитывать при обработке больших исходных наборов, состоящих из тысяч деревьев.

LogGeneTree = <логическое значение>
Если указано значение *true*, то каждое дерево генов будет включено в протокол работы программы; в противном случае дерево генов не выдается.

OptimumCostJ = <логическое значение>
Если указано значение *true*, то для каждого дерева генов в протокол будет включено информационное сообщение, которое содержит: порядковый номер дерева генов, число ребер в нем, число листьев (в том числе отмеченных символом **TmarkChar**), число уровней и цена оптимального вложения.

GtreeHeading = <логическое значение>
Если указано значение *true*, то непосредственно перед деревом генов в протокол будет выдана строка заголовка с порядковым номером дерева в исходном наборе.

GtreeLengths = <логическое значение>,<логическое значение>,<логическое значение>
Параметр аналогичен **StreeLengths**, но относится к дереву генов.

GtreeLabels = <логическое значение>,<логическое значение>,<логическое значение>
Параметр аналогичен **StreeLabels**, но относится к дереву генов.

GtreeNumbers = <логическое значение>,<логическое значение>,<логическое значение>
Параметр аналогичен **StreeNumbers**, но относится к дереву генов.

GtreeFullLabel = <логическое значение>
Параметр аналогичен **StreeFullLabel**, но относится к дереву генов.

LogTable1 = <логическое значение>
Если указано значение *true*, то в протокол будет включена таблица наиболее вероятных эволюционных событий (первые **MaxEvents** вариантов в порядке возрастания цены) для каждой пары <ребро дерева генов, труба дерева видов>. Таблица строится в процессе решения Задачи 1. Строки таблицы соответствуют ребрам, столбцы – трубам; события в каждой клетке перечислены сверху вниз, каждое занимает в протоколе четыре строки, которые содержат: цену (или значение –0.0, если ячейка не используется), идентификатор типа события (или

прочерк), два номера труб, дополнительно связанные с событием (0 означает неиспользуемое значение, т.к. трубы и вершины дерева видов нумеруются, начиная с 1). Распечатка таблицы даже для одного дерева может быть объемной.

T1TextID = <логическое значение>

Если указано значение *true*, то идентификатор типа события в таблице будет текстовый (т.е. осмысленный, например, dupl), в противном случае – числовой согласно нумерации в Техническом задании.

T1EdgeLabel = <число>

Ширина поля имени ребра дерева генов (используется для идентификации строк таблицы). Если указано значение 0, имена ребер не печатаются.

T1EdgeNumber = <число>

Ширина поля номера ребра дерева генов (используется для идентификации строк таблицы). Если указано значение 0, номера ребер не печатаются.

T1TubeLabel = <число>

Ширина поля имени трубы дерева видов (используется для идентификации столбцов таблицы). Если указано значение 0, имена труб не печатаются.

T1TubeNumber = <число>

Ширина поля номера трубы дерева видов (используется для идентификации столбцов таблицы). Если указано значение 0, номера труб не печатаются.

T1Precision = <число>

Число десятичных знаков после запятой при выводе значений цены (если указано 0, значения округляются до целых).

LogScenario = <логическое значение>

Если указано значение *true*, то в протокол будет включен оптимальный сценарий вложения каждого дерева генов в дерево видов, построенный в процессе решения Задачи 2.

LogS2Message = <логическое значение>

Если указано значение *true*, то в протокол будет включено информационное сообщение с ценой оптимального сценария (если запрошена выдача сценария, сообщение непосредственно предшествует ему).

ConS2Message = <логическое значение>

Если указано значение *true*, то на консоль будет выдаваться информационное сообщение с ценой оптимального сценария вложения каждого дерева генов.

S2Condense = <логическое значение>

Если указано значение *true*, то оптимальный сценарий выдается в сжатом формате (исходя из актуальных длин имен и номеров ребер и труб); в противном случае используются значения соответствующих параметров ниже.

S2Indent = <число>

Параметр указывает величину абзацного отступа при печати сценария в протоколе.

Формально сценарий – это эволюционное дерево, которое выдается вершина за вершиной, начиная с корня, так что каждая вершина занимает одну строку. Порядок обхода дерева – сначала вглубь. При переходе на более глубокий уровень абзацный отступ увеличивается на указанную величину, при подъеме вверх – уменьшается, что помогает правильно интерпретировать дерево. Если указано значение 0, абзацного отступа нет и данные всех вершин выводятся строго друг под другом.

S2EdgeLabel = <число>

Ширина поля имени ребра дерева генов при выдаче сценария. Если указано значение 0, имена ребер не печатаются.

S2EdgeNumber = <число>

Ширина поля номера ребра дерева генов при выдаче сценария (номер отделяется от имени знаком =). Если указано значение 0, номера ребер не печатаются.

S2TubeLabel = <число>

Ширина поля имени трубы дерева видов при выдаче сценария. Если указано значение 0, имена труб не печатаются.

S2TubeNumber = <число>

Ширина поля номера трубы дерева видов при выдаче сценария (номер отделяется от имени знаком =). Если указано значение 0, номера труб не печатаются.

S2EventLabel = <число>

Ширина поля текстового идентификатора типа события. Если указано значение 0, текстовые идентификаторы событий не выдаются.

S2EventID = <число>

Ширина поля номера типа события (согласно нумерации в ТЗ). Если указано значение 0, номера типов событий не выдаются.

S2EventCost = <число>

Ширина поля значения цены при выдаче сценария. Если указано значение 0, цены не включаются в сценарий.

S2Precision = <число>

Число десятичных знаков после запятой при выводе значений цены (если указано 0, значения округляются до целых).

LogOrbigraph0 = <логическое значение>

Если указано значение *true*, то в протокол будет включено состояние орбиграфа после прямого хода при решении Задачи 3.

LogOrbigraph = <логическое значение>

Если указано значение *true*, то в протокол будет включено состояние орбиграфа после обратного хода при решении Задачи 3.

LogOgMessage = <логическое значение>

Если указано значение *true*, то в протокол будет включено информационное

сообщение с параметрами орбиграфа (если запрошена выдача сценария, сообщение непосредственно предшествует ему). Сообщение содержит: порядковый номер дерева генов в исходном наборе, число вершин орбиграфа, число унарных ребер, число бинарных ребер (биребер), число троек на ребрах.

ConOgMessage = <логическое значение>

Если указано значение *true*, то на консоль будет выдаваться информационное сообщение с параметрами орбиграфа (такое же, как описано в связи с параметром **LogOgMessage**).

OgEdgeLabel = <число>

Ширина поля имени ребра дерева генов при выдаче орбиграфа. Если указано значение 0, имена ребер не печатаются.

OgEdgeNumber = <число>

Ширина поля номера ребра дерева генов при выдаче орбиграфа (номер отделяется от имени знаком равенства). Если указано значение 0, номера ребер не печатаются.

OgTubeLabel = <число>

Ширина поля имени трубы дерева видов при выдаче орбиграфа. Если указано значение 0, имена труб не печатаются.

OgTubeNumber = <число>

Ширина поля номера трубы дерева видов при выдаче орбиграфа (номер отделяется от имени знаком равенства). Если указано значение 0, номера труб не печатаются.

OgPrecision = <число>

Точность (число десятичных знаков после запятой) выдачи значений вероятностей вершин и ребер орбиграфа в протокол.

LogTmarkJ = <логическое значение>

Если указано значение *true*, то при решении Задачи 3 в протокол включаются значения функции g(I,T) отдельно для каждого дерева генов (а не только сумма по всем деревьям).

LogTubeJ = <логическое значение>

Если указано значение *true*, то при решении Задачи 3 в протокол включаются значения функции f(I,x) отдельно для каждого дерева генов (а не только сумма по всем деревьям).

LogOTubeJ = <логическое значение>

Параметр аналогичен **LogTubeJ**, но таблица значений функции f(I,x) строится для «старых» труб, т.е. до разбиения исходного дерева видов на слои.

LogMeanJ = <логическое значение>

Если указано значение *true*, то выдается информационное сообщение с величиной мат. ожидания цены сценариев для каждого дерева генов в отдельности (а не суммарное по всему набору).

LogMemoryJ = <логическое значение>

Если указано значение *true*, то в протоколе приводятся данные о расходе оперативной памяти при обработке каждого дерева генов (максимальное пиковое значение в килобайтах указано после метки total=).

Настройки итогового протокола

Итоговый протокол содержит результаты решения Задачи 3 и выдается по окончании обработки всего исходного набора деревьев генов. Предусмотрены следующие управляющие параметры.

LogTmarkStats = <логическое значение>

Если указано значение true, то в протокол включаются итоговые значения функции g(I,T) суммарно по всем деревьям генов.

LogTubeStats = <логическое значение>

Если указано значение *true*, то в протокол включаются итоговые значения функции f(I,x) суммарно по всем деревьям генов.

LogOTubeStats = <логическое значение>

Параметр аналогичен **LogTubeStats**, но таблица значений функции f(I,x) строится для «старых» труб, т.е. до разбиения исходного дерева видов на слои.

LogTotalMean = <логическое значение>

Если указано значение *true*, то выдается информационное сообщение с величиной мат. ожидания суммарной цены сценариев для всего набора деревьев генов.

Параметры командной строки программы

В дополнение к конфигурационному файлу, предусмотрен ряд опций командной строки запуска программы, позволяющих оперативно изменять некоторые из параметров, перечисленных в предыдущем разделе. Вообще говоря, использовать эти опции не обязательно, и если файл конфигурации находится в каталоге программы и имеет стандартное имя **embed3GL.ini**, программу можно запускать вообще без параметров. Если имя и/или расположение файла конфигурации иные, достаточно использовать опцию **-g** (или, что то же самое, **-i**) для указания нужного файла конфигурации. Значения параметров в командной строке имеют приоритет перед указанными в файле конфигурации.

Каждая опция задается ключом (знак дефиса и 1-5 символов сразу за ним), некоторые опции требуют значения в следующем параметре, отделяемом от ключа одним или более пробелов. Регистр символов ключа не имеет значения. Текущая версия программы допускает следующие опции командной строки (для получения подсказки со списком служит параметр **-h**).

-q <имя_файла> (синоним: **-i** <имя_файла>)

Указывает имя файла конфигурации. Имя может содержать путь к каталогу (абсолютный или из каталога программы). Предпочтительным является использование ключа **-q**, поскольку опция **-i** неправильно воспринимается на некоторых кластерах.

-a <имя_файла> (параметры конфигурации **LogFilename**, **LogAppend**)

Задает имя файла протокола, который будет дозаписываться (старое содержимое

сохраняется). Имя может содержать путь к каталогу. Если ключ указан в форме **--a**, то наоборот, протокол будет перезаписан, независимо от значения **LogAppend**. Следует учитывать, что если файл конфигурации содержит параметр **DataDirectory**, его значение будет добавлено перед указанным именем файла.

-l <имя_файла> (параметры конфигурации **LogFilename**, **LogAppend**)
Задает имя файла протокола, который будет дозаписываться (старое содержимое сохраняется). Имя может содержать путь к каталогу. Если ключ указан в форме **--a**, то наоборот, протокол будет перезаписан, независимо от значения **LogAppend**. Следует учитывать, что если файл конфигурации содержит параметр **DataDirectory**, его значение будет добавлено перед указанным именем файла.

-cbiga <число> (параметр конфигурации **C_gain_big)**
Цена события большого приобретения (здесь и далее указанное значение цены округляется до двух десятичных знаков после запятой).

-cdupl <число> (параметр конфигурации **C_dupl)**
Цена события дупликации гена.

-cgain <число> (параметр конфигурации **C_gain)**
Цена события приобретения гена.

-closs <число> (параметр конфигурации **C_loss)**
Цена события потери гена.

-cslep <число> (параметр конфигурации **C_sleep)**
Цена события перехода гена в спящее состояние. (Допускается также синонимичная форма ключа, **-csleep**).

-ctrwi <число> (параметр конфигурации **C_tr_with)**
Цена события горизонтального переноса гена с сохранением.

-ctrwo <число> (параметр конфигурации **C_tr_without)**
Цена события горизонтального переноса гена без сохранения.

-cinit <число> (параметр конфигурации **InitialCost**)
Начальная цена (см. описание параметра **C_exponent**).

-cexpo <число> (параметр конфигурации **C_exponent**)
Показатель степени (см. описание параметра **C_exponent**).

-corrk <число> (параметр конфигурации **K_correction**)
Коэффициент коррекции (см. описание параметра **K_correction**).

-g <имя_файла> (параметр конфигурации **GeneTree**)
Задает имя файла, содержащего одно или более деревьев генов. Имя может содержать путь к каталогу, однако следует учитывать, что если файл конфигурации содержит параметр **DataDirectory**, его значение будет добавлено перед указанным именем файла. Деревья в указанном файле будут добавлены к исходному набору деревьев генов, заданному параметрами **GeneTree** (их может быть несколько) в файле конфигурации. Напротив, если ключ указан в форме **--g**, то деревья генов в

файле конфигурации игнорируются, и исходный набор состоит только из деревьев в предлагаемом файле.

-h Выдает краткую подсказку о допустимых параметрах командной строки.

-k <число> (параметр конфигурации **MaxEvents**)
Максимальная степень ветвления эволюционного сценария.

-nompi (синоним: **--mpi**)
Если указана эта опция, то программа будет работать в однопроцессорном режиме независимо от наличия подходящей параллельной среды.

-s <имя_файла> (параметр конфигурации **SpeciesTree**)
Задает имя файла, содержащего дерево видов. Имя может содержать путь к каталогу, однако следует учитывать, что если файл конфигурации содержит параметр **DataDirectory**, его значение будет добавлено перед указанным именем файла. Дерево видов в указанном файле будет использовано вместо заданного параметром **SpeciesTree** в файле конфигурации.

Литература

1. Lyubetsky V.A., Rubanov L.I., Rusin L.Yu., Gorbunov K.Yu. "Cubic time algorithms of amalgamating gene trees and building evolutionary scenarios", *Biology Direct*, 2012 (submitted).

2. К.Ю. Горбунов, В.А. Любецкий «Реконструкция эволюции генов вдоль дерева видов», *Молекулярная биология*, 2009, том 43, № 5, стр. 946–958.

3. К.Ю. Горбунов, В.А. Любецкий «Об одном алгоритме согласования деревьев генов и видов с учетом дупликаций, потерь и горизонтальных переносов генов», *Информационные процессы*, 2010, том 10, № 2, стр. 140–144.

4. К.Ю. Горбунов, В.А. Любецкий «Дерево, ближайшее в среднем к данному набору деревьев», *Проблемы передачи информации*, 2011, том 47, вып. 3, стр. 64–79.