`regevscape` Manual

Kevin Bullaughey
kbullaughey@gmail.com

February 8, 2011

Overview

This manual provides documentation of the `regevscape` program, which is short for **reg**ulatory **ev**olution land**scape** simulator. `regevscape` simulates sequence evolution of an enhancer using a computational model of regulatory function based on the Segal model [1], a fitness penalization for misexpression, and forward population simulations. It was originally used by Bullaughey [2] to investigate nucleotide substitution processes in a regime of stabilizing selection for a particular regulatory output of simple toy enhancers. For an overview of the model and details on the implementation, please refer to the original paper [2]. The purpose of this manual is to document usage of the program and aid interpretation of the output.

Terminology

Throughout this manual I invoke specific meanings of certain biological terms as they are relevant to the simulation.

A transcription factor's **activity** is the degree to which it contributes to activation of the basal transcription apparatus (for a repressor, this is a negative quantity). An **enhancer** is a specified length of DNA that is being modeled and is assumed to have regulatory function, and be in whatever chromatin configuration is necessary to be expressed. The enhancer is assumed to function in one or more discrete ***trans*-backgrounds**, which can be thought of as a particular cell type, developmental time point, or spatial position. Each *trans*-background is defined by the expression levels of the transcription factors (TFs) present. Each *trans*-background has an optimal expression level of the gene in that *trans*-background. The *trans*-backgrounds, together with the optima, a specification of the TF binding functions and activity levels, defines a **regulatory problem**, i.e., the particular task faced by the gene to express properly and what it has to work with.

Some terminology relates specifically to the Segal model and I follow my earlier conventions [2]. Occupancy of a nucleotide in the enhancer is the probability a TF is observed to be bound in a way overlapping the nucleotide. A configuration is a particular arrangement of TFs on the enhancer. The only valid configurations include the empty configuration (no TFs) or configurations with non-overlapping TFs.

Modes of operation

`regevscape` has two basic modes of operation. One is to do forward population simulations.

The other mode is to compute various statistics related to the sequence fitness landscape. Possibilities here include sampling independent and identically distributed (*iid*) sequences and computing fitness, computing all sequences within some number of mutational steps, computing expression profiles, computing various occupancy statistics, or sampling *iid* from the configurations of TFs.

Inputs

There are several components that need specification in order to parameterize the regulatory landscape and optionally, simulate regulatory evolution of an enhancer. Parameters required for specification of the fitness landscape include the binding functions of the TFs, the TF activity levels, the expression levels of these TFs in each *trans*-background, the optimal expression levels in each *trans*-background, and the shape of the fitness function. For simulating regulatory evolution, additional parameters need specification including population size, number of generations, mutation rates, insertion and deletion rates and distributions, and the initial sequence representing the initially monomorphic population. Finally, parameters dictating what output is to be provided are also needed. All these parameters are specified on the command line in an order-independent way.

Invoking `regevscape`: an example

As an example, the following is one invocation of the `regevscape` command used in the original paper:

```
regevscape --model=segal --steps=200000 --length=100 --seed=129993921 \
   --pwm=pwm_pair.pwms --tfs=2 --lambda=-1,2,-3 --gamma=2,-0.5,-0.5,2 \
   --condition=0.02,0.02 --condition=0.15,0.3 --condition=0.02,0.02 \
   --ffunc=f4 --coef=0.6,0.881,0.046,0.881\
   --popsim -N 1000 -g 2000000 -u 5e-07 \
   --pstat=most_freq_seq,100 --pstat=mutational_effects --pstat=allele_loss \
   --seq=ttatcacagagtaaggggc...
```

In the above, the initial sequence, specified with –seq, has been truncated but was originally 100 bp and the file, pwm_pair.pwms, is assumed to exist.

To get a condensed help message detailing the possible command-line options to `regevscape`, simply invoke the program with no options, or the –help or -h options.

Parameters

Specification of the model is rather verbose at the moment, which allows for some consistency checking so that copy-paste errors when running many variations among simulations can somewhat be avoided.

Currently the only model that is implemented in its entirety is the Segal model. Originally, I experimented with several other models of fitness landscapes (completely neutral, Orr's block model), but these are not complete. So all invocations of the program should contain the parameter: `--model=segal`.

And when invoking the Segal model, one needs to specify how many samples, S, will be drawn (*iid*) from the distribution of configurations. This is to compute the following summation, which is estimated using Monte Carlo sampling:

$$P(E) = \sum_{c_k \in C} P(E|c_k)P(c_k) \approx \frac{1}{S}\sum_{i=1}^{S} P(E|c_i)$$

How many samples to draw is given with the `--steps=S` parameter. Naturally, the more samples, the more accurate the estimate of expression. This is particularly important when running evolutionary simulations, because misestimation of expression will lead to misestimation of fitness, and potentially allow mutations to fix that would not ordinarily fix, and potentially, create artificial local optima, due to an overestimation of fitness, making subsequent mutations look artificially unfit. I found that 200,000 samples was appropriate for a population size of $N = 1000$ and a selection tolerance parameter of $\sigma^2 = 0.6$ (see below). In this setting, estimation error was generally an order of magnitude smaller than $\frac{1}{N}$ which is the approximately the magnitude when selection is overwhelmed by drift.

The `--length=N` parameter gives the length of the enhancer sequence. This is not required if a sequence is specified, but must be provided if computing statistics of sequences sampled uniformly from sequence space (so the program knows to generate sequences of the desired length).

Finally, the seed of the random number generator can be set with `--seed=N`, where `N` is an non-negative integer. Starting with the same seed is guaranteed to run deterministically, and identically each time.

Regulatory problem

Specification of the regulatory problem requires providing the parameters: `--pwm`, `--tfs`, `--lambda`, `--gamma`, and one or more `--condition` specifications. In what follows, I reproduce equations from the paper using this implementation [2], which are very similar to the formulation given in the original formulation of the Segal model [1]. For a more complete description of the model, please refer to these papers.

`--pwm=<filename>` gives the file name of the position weight matrix file which specifies the binding affinity of the TFs. Here is an example of a valid pwm file:

```
pwm 6
0.3369 0.2508 0.9626 0.8526 0.6829 0.8748
0.0513 0.3186 0.0181 0.0449 0.2433 0.0072
0.5338 0.2232 0.0077 0.1021 0.0123 0.0902
0.0779 0.2074 0.0116 0.0004 0.0615 0.0278
pwm 6
0.0834 0.1921 0.4285 0.1172 0.9270 0.2437
0.1072 0.0138 0.1041 0.0154 0.0022 0.1425
0.3878 0.1982 0.3906 0.8614 0.0650 0.0247
0.4216 0.5958 0.0768 0.0059 0.0058 0.5891
```

For this file to be formatted properly, it must consist of one or more PWM records. Each record has a header line with the word `pwm` followed by a space followed by an integer, say r, indicating the width of the PWM in nucleotide positions. The next four lines must contain a matrix of site-specific affinities such that there are r columns and each column sums to one. These are the relative affinity of each position to each of the four possible nucleotides. Rows correspond to the affinities of each site to A, G, C, and T respectively.

The number of TFs that are being modeled is specified with the `--tfs=T` parameter, and this must match the number of PWM records provided. `--lambda=`$\lambda_0, \lambda_1, \ldots, \lambda_T$ gives the activity levels of the T TFs preceded by the basal activity level of the promoter, λ_0. These (floating-point) parameters come into play in the formulation of $P(E|c_k)$, which follows a logistic sigmoid:

$$P(E|c_k) = \frac{1}{1 + \exp(-\sum_{i=0}^{M} \lambda_{TF_i})}$$

Where there are M TFs in configuration c_k. The length of the `--lambda` vector must be one more than the number of TFs given in `--tfs`. A positive λ codes for an activator and negative λ codes for a repressor. Similarly, a negative λ_0 codes for a generally repressive basal promoter and a positive one codes for a generally activated one. A present limitation of this implementation is that the λ parameters are the same across all *trans*-backgrounds, not allowing for a TF to sometimes be an activator and sometimes a repressor, a situation known to be important in some true biological settings.

Interaction properties of the TFs are given by `--gamma=`$\gamma_{11}, \gamma_{21}, \ldots, \gamma_{TT}$ gives a matrix (iterating over columns before rows, Fortran notation) of pair-wise TF-TF cooperativity parameters. Omitting this parameter means there are no interactions among TFs. $\gamma jk \in [-1, \infty)$, with $g_{jk} = 0$ indicating no interaction. These γ_{jk} parameters have the effect of altering the probability of a configuration in a way that is a function of the distance, d, between adjacently bound TF_{i-1} and TF_i, such that the probability of a configuration, c_k is:

$$P(c_k) \propto w(c_k) = \prod_{i=1}^{M} \tau_{\mathrm{TF}_i} \frac{\mathrm{PWM}_{\mathrm{TF}_i}(s_x, s_{x+1}, \cdots, s_{x+r-1})}{\mathrm{PWM}_b(s_x, s_{x+1}, \cdots, s_{x+r-1})} \prod_{i=2}^{M} \gamma(\mathrm{TF}_i, \mathrm{TF}_{i-1}, d)$$

where γ_{jk} is given by this function of the distance, d, between the adjacent TFs, $j = \mathrm{TF}_{i-1}$ and $k = \mathrm{TF}_i$:

$$\gamma(\mathrm{TF}_i = j, \mathrm{TF}_{i-1} = k, d) = 1 + g_{jk} e^{\frac{-d^2}{v}}$$

Currently, v is hard-coded to 80, which means that interactions only extend maybe 20 bp or so for modest g_{jk} (e.g.,. $-0.9 < g_{jk} < 9$, a range covering up to a 10-fold decrease or increase in $P(c_k)$ due to the interaction).

Finally, the last requirement for a regulatory problem is specifying the expression levels of the TFs in each of the *trans*-backgrounds. These expression profiles are given by the `--condition=`$\tau_1, \ldots, \tau_T$ parameters, where $\tau_i > 0$. So, if a regulatory problem has three distinct *trans*-backgrounds, then three `--condition` vectors must be given.

Fitness function

The original purpose of the Segal model was to provide a predictive model of expression and was essentially an optimization/inference problem. In order to use it to study evolution, I consider several parameterizations of expression-to-fitness mappings. Presently there are six fitness functions coded. The desired fitness function is given by the `--ffunc=<choice>` parameter where `<choice>` is a character string selecting one of the following:

Choice	Functional form	Notes
`linear`	$f = \mathbf{x}^T\mathbf{c}$	linear combination
`nop`	$f = 1$	neutral
`f1`	$f = c_1(x_1 - c_2)^2 + c_3(x_2 - c_4)^2 + c_5$	
`f2`	$f = c_1^{(-c_2*(x_1-c_3)^2)} c_4^{(-c_5*(x_2-c_6)^2)}$	
`f3`	$f = \prod_{i=1}^{B}(c_1^{(} - c_2 * (x_i - c_{i+2})^2))^{\frac{1}{C}}$	B conditions
`f4`	$f = \exp(-\sum_{i=1}^{B} \frac{(x_i - c_{i+1})^2}{c_1})$	Gaussian kernel, B conditions

Each fitness function (with the exception of `nop`) requires a vector of coefficients, `--coef=c1,c2,...`, parametrizing the selected function.

Evolution simulation

Opportunities for extension

Where possible, I have tried to code `regevscape` in a way that can be easily extended. Here I outline a few easy extensions that would be straightforward to integrate.

References

[1] Segal E, Raveh-Sadka T, Schroeder M, Unnerstall U, Gaul U (2008) Predicting expression patterns from regulatory sequence in drosophila segmentation. Nature 451: 535-40.

[2] Bullaughey KL (2010) Changes in selective effects over time facilitate turnover of enhancer sequences. Genetics : genetics.110.121590.