

# Evaluating alignment quality and stress-testing the aligner

## Purpose

When constructing a sequence alignment system various design decisions can have an impact on sequence alignment quality. To assess the effects each of these decisions can have on alignment quality, we have created a method to score alignment accuracy across numerous test cases using genome sequences that have diverged to varying degrees. We refer to such test results as "accuracy profiles" because they demonstrate the aligner's average accuracy over many evolutionary scenarios.

## Overview of the methodology

- \* simulate evolution
- \* align simulated genomes
- \* score the alignment
- \* do this many times using different parameter combinations to characterize the aligner's behavior in the presence of different types of evolution
- \* Use a high-throughput computing environment such as [Condor](#) to quickly process many simulations

## Using the evolver software

A pre-compiled Windows executable is available here: <http://gel.ahabs.wisc.edu/mauve/downloads/sgEvolver.exe>

The following components are necessary to compile the evolver software under Windows, Linux, or Mac OS X:

libGenome <http://www.libgenome.org> libMems <http://gel.ahabs.wisc.edu/mauve/source> sgEvolver  
<http://gel.ahabs.wisc.edu/mauve/source> mauveAligner <http://gel.ahabs.wisc.edu/mauve/source> pkg-config  
<http://www.freedesktop.org/software/pkgconfig/releases/wxWidgets> <http://www.wxwidgets.org>

The following additional software supports automated aligner testing and generation of postscript accuracy plots:

seq-gen <http://evolve.zoo.ox.ac.uk/software.html?id=seqgen> R <http://r-project.org> perl <http://perl.org>

All of these packages must be installed in order to perform aligner accuracy profiling. Some of these packages may already be installed on your system.

## Necessary hardware

Genome alignment software is notoriously CPU intensive. Performing all but the most rudimentary set of alignment experiments will require significant computation resources. This software is designed to work on large compute clusters with the condor job scheduler, however submission scripts for other scheduling systems should be straightforward.

## Compiling sgEvolver

- \* Follow instructions to [compile mauveAligner from source](#)
- \* download sgEvolver source
- \* configure --with-libGenome=/prefix/to/libGenome
- \* make
- \* make install

## Compiling from a source snapshot on unix

The nightly subversion source code snapshots do not include a distributable build system. In order to build from the snapshots you will need the GNU autotools (e.g. automake, autoconf, etc.) installed on your system. After unpacking the tarball, execute the following sequence of commands to generate the build system: libtoolize && aclocal && autoheader && automake -a && autoconf. The binaries can then be built using the usual ./configure, make, make install procedure.

## How to simulate evolution and test an aligner's accuracy:

- 1) Install the software listed above
- 2) Create a new directory for the simulation experiment and make a copy of `simujobparams.pm` in this directory.
- 3) Create a raw ancestral sequence file. The ancestral sequence must be at least twice as long as the size of genomes to be evolved, preferably longer. It must also be a "raw" sequence file, meaning that it has no Multi-FastA formatting and no deflines: just the

characters { A,C,G,T,a,c,g,t } all on a single line. A utility application called toRawSequence is included with mauveAligner that can convert many major sequence formats to raw sequence.

- 4) Edit simujobparams.pm appropriately. At the very least be sure to set the name of the ancestral sequence file and the locations where software is installed. The simujobparams.pm file has detailed instructions describing each variable.
- 5) Run simujobgen.pl Running the simujobgen script will create a number of alignjob directories, each of which corresponds to a simulation and alignment job with a different combination of mutation parameters.
- 6) change into one of the alignjob directories
- 7) Run simujobrun.pl <aligner> where <aligner> is the aligner to use and can be one of none, mauve, mavid, mlagan, or slagan, assuming these aligners have been installed correctly. When none is selected, the procedure generates the evolved genomes without trying to align and score them.
- 8) In case something goes wrong, check the files command\_lines.txt and all the files ending in .err. The .err files correspond to each program's standard error output.
- 9) If none was selected for your aligner, then the evolved sequences will reside in the Multi-FastA file evolved\_seqs.fas, and an alignment of orthologous regions will reside in evolved.dat. If an aligner was tested then the alignment scores are reported in the file scores.txt and the scores on regions conserved among all genomes (backbone) are in the file bb\_scores.txt

## Using the Condor High Throughput Computing environment to process many simulations rapidly

- 1) Follow steps 1 through 5 above. When editing simujobparams.pm be sure to set the paths to the aligners and the scoring tools. These paths must be accessible from the compute node running the job, thus they should be on shared storage. The simujobrun.pl script executes programs in order to carry out simulated evolution, alignment, and scoring of alignments. In particular, evolution requires dd, seq-gen, and sgEvolver. Alignment requires an aligner, and scoring requires scoreAlignment and extractBackbone.
- 2) Step 5 will create a condor DagMan submission script called jobs.dag and a job submission script called mauveAlign.condor. Edit the mauveAlign.condor submission script to select the aligner you would like to test. Options are mauve, mavid, mlagan, slagan, and none. Optionally, the debug parameter may also be given to simujobrun. When debug is used none of the data files generated during the simulated evolution and alignment process are deleted and all get sent back to the job submission host. It may be necessary to set other condor-specific parameters as well.
- 3) Submit the condor jobs with condor\_submit\_dag -maxjobs ## jobs.dag Here ## is the maximum number of condor jobs that will run simultaneously.
- 4) When all jobs have completed successfully, use scoregen.pl to extract scores from the alignjob directories and generate a heat plot in postscript format. scoregen.pl depends on rgradientplot.R in your tools directory. When running multiple replicates of the simulation, scoregen.pl will automatically average together the scores for each replicate and plot a single average score for that combination of mutation rates.