Repeat Modeler

a program to annotate transposable repeat elements

Repeated elements are sometimes very difficult to deal with when assembling a genome. Once the genome is assembled, however, they become a feature of the genome as much as any other part is.

Repeat modeler website: <http://www.repeatmasker.org/RepeatModeler/>

Also the in-system help is triggered by loading the module and using the command with no arguments or conditions.

Repeat Modeler is a 2 part system that makes us of a database built by the first part of the program, aptly called BuildDatabase, and the second part which blasts Repeat sequences against this database.

This little tutorial is for use with Repeat Modeler version 1.0.11

Since the idea is a simple one, lets dive right into the code:

name=what\_you\_want\_the\_outputs\_named

fasta=your\_assembled\_gapfilled\_polished\_genome.fasta

BuildDatabase -name ${name} -engine ncbi ${fasta} && RepeatModeler -engine ncbi -pa 8 -database ${name}

Lets go through this one term at a time:

**For BuildDatabase:**

**-name**

This tells the program that the next term is the name that the outputs should have added to them. It need not be a variable but using a variable can help to make the code easier to read and parse for errors.

**-engine**

This tells the program what blast tool will be used. Other options besides ncbi are also blast searches, such as abblast, wublast, etc. but it appears that the ncbi is included with the program while the others listed are not.

**${fasta}**

This is the variable defined earlier in the script, but it is just the genome you want to use for this program. The input, if you will.

**For RepeatModeler:**

**-engine**

This tells the program what BLAST method you want to use. Same as before.

**-pa**

This is the number of threads allocated to the BLAST searches. This should always be less than the total number of threads requested in order for the program itself to continue running in parallel. So if you request 9 threads, you can put in **-pa 8** to assign 8 of them to the searches.

**-database**

This term tells the program that the next term is what you want the program to use at the database to blast against. That database was built in the BuildDatabase command.

Notes about runtime:

This program can take a while to run, but that time is related to two factors: 1) how big your genome is. Typically, the bigger the longer it takes. 2) how much repeated data there is. If you have a smaller genome but it is packed with repeats, expect it to take longer than a genome of the same size with fewer repeats. If you want some estimations, the website listed above has some time values that were from programs run on the developer’s system.

Final Note:

There is an option to use if during a long run there is a disruption of the program and it quits. Simply use the -recoverdir option. If you do use it the proper way is:

**-recoverdir <directory\_where\_results\_were\_being\_put>**