**Repeat Masker**

Masking repeat sequences so that they are skipped in future analysis, or annotated and masked.

The idea of this program is also simple enough. Using a pre-determined library, the program uses BLAST searches to identify repeated sequence and masks it. The default is to replace those repeated bases with ‘N’.

The website for this program is: <http://www.repeatmasker.org/>

Here is the sample code we can look at:

lib=path/to/the/library/used/in\_the/blast\_searches

fasta=your\_assembled\_gap-filled\_polished\_genome.fasta

output=output/path/for/results

Making these terms into variables has several uses, not the least of which is a cleaner, easier to read line of code. But it is as always largely optional.

RepeatMasker -pa 8 -lib ${lib} -xsmall -gff -dir ${output} ${fasta} && buildSummary.pl ${output}/${fasta}.out > ${output}/${fasta}.detailed.tbl

This program has several options, and we will go through many of them. It would still be wise to familiarize yourself with the in program help, which can be looked at by loading the module and typing RepeatMasker without any arguments or options.

Lets go in order as they are written in the script above:

**For RepeatMasker:**

**-pa** This option tells the program how may threads are devoted to the BLAST searches. It should always be at least 1 less than the number of total threads requested to run the job.

**-lib ${lib}** This tells the program the pathway to the library files it needs to use to blast against the supplied Fasta file.

**-xsmall** This optional term tells the program to output the repeats not as ‘N’s but as lower-case versions of the regular ATCG bases. This is sometimes referred to as soft masking.

**-gff** This is the term telling Repeat Masker what format you want the output files put into.

**-dir** This option tells the program where you want the output files saved to instead of stdout. In this case it is a variable ${output}.

The last term in the line before the next section is the ${fasta} variable, which is telling the program what you want used as the input.

The next section of the code deals with **buildSummary.pl**

**buildSummary.pl**

This command takes the output from RepeatMasker and builds a table detailing the types of sequences found and masked, as well as what percentage of the genome was masked with each one, and also the total masked percentage when all types of masked sequence are put together.

To make it run you simply need to first give it the files you want summarized, here written as

**${output}/${fasta}.out**

directly after the command, then in this case the output was told to write to a specific place,

**${output}/${fasta}.detailed.tbl**

otherwise it will print to stdout.

Note:

This program will likely take some time to run if given a low amount of memory or threads. This is due to the sheer amount of BLAST searches that need to be done. It will of course be shorter if the genome is smaller, and likewise if the repeated portion is also small.

Final Note:

There are a number of other options not discussed here, some of which have to do with which types of repeated or low complexity bases are masked and how they are masked. There are also some options that deal with how the summary tables are built. You are encouraged to use this small tutorial as a jump off point when learning about this program prior to using it.