RepeatModeler Download

RepeatModeler - 1.0.11

Arian Smit, Robert Hubley - Institute for Systems Biology

RepeatModeler is a de-novo repeat family identification and modeling package. At the heart of RepeatModeler are two de-novo repeat finding programs (RECON and RepeatScout) which employ complementary computational methods for identifying repeat element boundaries and family relationships from sequence data. RepeatModeler assists in automating the runs of RECON and RepeatScout given a genomic database and uses the output to build, refine and classify consensus models of putative interspersed repeats.

Prerequisites

1. **Perl**
   Available at http://www.perl.org/get.html. Developed and tested with version 5.8.8.
2. **RepeatMasker & Libraries**
   Developed and tested with open-4.0.7. The program is available at http://www.repeatmasker.org/RMDownload.html and the libraries are at http://www.girinst.org.
3. **RECON - De Novo Repeat Finder, Bao Z. and Eddy S.R.**
   Developed and tested with our patched version of RECON (1.08). The 1.08 version fixes problems with running RECON on 64 bit machines and supplies a workaround to a division by zero bug along with some buffer overrun fixes. The program is available at: http://www.repeatmasker.org/RECON-1.08.tar.gz. The original version is available at http://eddylab.org/software/recon/.
4. **RepeatScout - De Novo Repeat Finder, Price A.L., Jones N.C. and Pevzner P.A.**
   Developed and tested with our multiple sequence version of RepeatScout (1.0.5). This version is now available at http://repeatscout.bioprojects.org/.
   Or...if this link isn't working you may download it here: http://www.repeatmasker.org/RepeatScout-1.0.5.tar.gz
5. **TRF - Tandem Repeat Finder, G. Benson et al.**
   You can obtain a free copy at http://tandem.bu.edu/trf/trf.html. RepeatModeler was developed using 4.0.4.
6. **NSEG - Low complexity sequence identification.**
   This is required by the RepeatScout package and can be downloaded from: ftp://ftp.ncbi.nih.gov/pub/seg/nseg/.
7. And one or both of the following search engines:
   - **RMBlast - A modified version of NCBI Blast for use with RepeatMasker and RepeatModeler.**
     Precompiled binaries and source can be found at http://www.repeatmasker.org/RMBlast.html
   - **ABBlast/WUBlast - Sequence Search Engine, W. Gish et al.**
     The program can be downloaded at http://blast.advbiocomp.com/licensing/.

RepeatModeler Installation

1. **Download RepeatModeler**
   Latest Version Released 10/20/2017: **RepeatModeler-open-1.0.11.tar.gz**
   Previous Version Released 5/12/2017: **RepeatModeler-open-1.0.10.tar.gz**
2. **Unpack Distribution**
   Unpack the distribution in your home directory or in a location where it may be shared with other users of your system (ie. /usr/local/).
   - cp RepeatModeler-open-1-#-#.tar.gz /usr/local
   - cd /usr/local
   - gunzip RepeatModeler-open-1-#-#.tar.gz
   - tar xvf RepeatModeler-open-1-#-#.tar
3. **Configure for your site**
   Automatic:
   - Run the "configure" script contained in the RepeatModeler distribution as: perl ./configure

Example Run

In this example we first downloaded elephant () sequences from Genbank (approx 11MB) into a file called elephant.fa.

1. Create a Database for RepeatModeler RepeatModeler uses the ABBlast/RMBlast(ncbi) database as input to the repeat modeling pipeline. A utility is provided to assist the user in creating a single database from several types of input structures.

   [*RepeatModelerPath*]/BuildDatabase -name elephant -engine ncbi elephant.fa

   Run "BuildDatabase" without any options in order to see the full documentation on this utility. There are several options which make it easier to import multiple sequence files into one database.

   NOTE: It is a good idea to place your datafiles and run this program suite from a local disk rather than over NFS.

2. Run RepeatModeler RepeatModeler runs several compute intensive programs on the input sequence. For best results run this on a machine with a moderate amount of memory and several processors. Our typical setup was P4 - 4 cpus, 2.4Ghz, 3GB Memory, and Red Hat Linux.

   nohup [*RepeatModelerPath*]/RepeatModeler -engine ncbi -pa 3 -database elephant >& run.out &

   The nohup is used on our machines when running long (> 3-4 hour) jobs. The output is saved to a file and the process is backgrounded. The -pa parameter specifies how many searches should be run in parallel. For typical runtimes (can be > 2 days with this configuration) see the run statistics section of this file.

3. Interpret the results This development version of RepeatModeler produces a voluminous amount of output. The raw output is directed to a working directory named RM_. ie. "RM_5098.MonMar141305172005" and remains after each run for debugging purposes. At the completion of the run two files are generated:
   <database_name>-families.fa : Consensus sequences
   <database_name>-families.stk : Seed alignments
   The seed alignment file is in a Dfam compatible Stockholm format and may be uploaded to the new open Dfam_consensus database using the util/dfamConsensusTool.pl. See http://www.repeatmasker.org/RepeatModeler/dfamConsensusTool for details. The fasta format is useful for running quick custom library searches using RepeatMasker. Ie.:

   [*RepeatMaskerPath*]/RepeatMasker -lib consensi.fa.classified mySequence.fa

   Other files produced in the working directory include:

```
RM_./
   round-1/
        sampleDB-#.fa       : The genomic sample used in this round
        sampleDB-#.fa.lfreq : The RepeatScout lmer table
        sampleDB-#.fa.rscons: The RepeatScout generated consensi
        sampleDB-#.fa.rscons.filtered : The simple repeat/low
                                        complexity filtered
                                        version of *.rscons
        consensi.fa         : The final consensi db for this round
        family-#-cons.html  : A visualization of the model
                              refinement process.  This can be opened
                              in web browsers that support zooming.
                              ( such as firefox ).
                              This is used to track down problems
                              with the Refiner.pl
        index.html          : A HTML index to all the family-#-cons.html
                              files.
   round-2/
        sampleDB-#.fa       : The genomic sample used in this round
        msps.out            : The output of the sample all-vs-all
                              comparison
        summary/            : The RECON output directory
             eles           : The RECON family output
        consensi.fa         : Same as above
        family-#-cons.html  : Same as above
        index.html          : Same as above
   round-3/
        Same as round-2
    ..
   round-n/
```

Please see the RELEASE-NOTES file for more details.

4. Recover from a failure If for some reason RepeatModeler fails, you may restart an analysis starting from the last round it was working on. The -recoverDir [*ResultDir*] option allows you to specify a diretory (i.e RM_./) where a previous run of RepeatModeler was working and it will automatically determine how to continue the analysis.

Benchmarks

RepeatModeler 1.0.2 (RECON + RepeatScout):

Genome	Genome DB Size (bp)	Sample*** Size (bp)	Run Time* (hh:mm)	Models Built	Models Classified	% Sample Masked**
Human HG18	3.1 Bbp	238 Mbp	46:36	614	611	35.66
Zebrafinch	1.3 Bbp	220 Mbp	63:57	233	104	9.41
Sea Urchin	867 Mbp	220 Mbp	40:03	1830	360	33.85
diatom	32,930,227	32,930,227	4:41	128	35	2.86
Rabbit	11,770,949	11,770,949	3:14	83	72	31.30

* Analysis run on a 4 processor P4, 2.4Ghz, 3GB RAM, machine running Red Hat Linux.

** Includes simple repeats and low complexity DNA. Results obtained with RepeatMasker open-3.1.9, WUBlast and the -lib option.

*** Sample size does not include 40 Mbp used in the RepeatScout analysis. This 40 Mbp is randomly chosen and may overlap 0-100% of the sample used in the RECON analysis.

Caveats

- Genomes with numerous short contigs (Diatom for example) will take longer to BLAST than larger genomes with larger contigs. This is an optimization problem left for future releases.

Credits

Arnie Kas for the work done on the original MultAln.pm.

Andy Siegel for statistics consultations.

Thanks so much to Warren Gish for his invaluable assistance and consultation on his WUBlast program suite.

Alkes Price and Pavel Pevzner for assistance with RepeatScout and hosting my multi-sequence version of RepeatScout.

This work was supported by the NIH (R44 HG02244), (R01 HG002939) and the Institute for Systems Biology.

License

RepeatModeler "open-1.0" is licensed under the Open Source License v2.1.