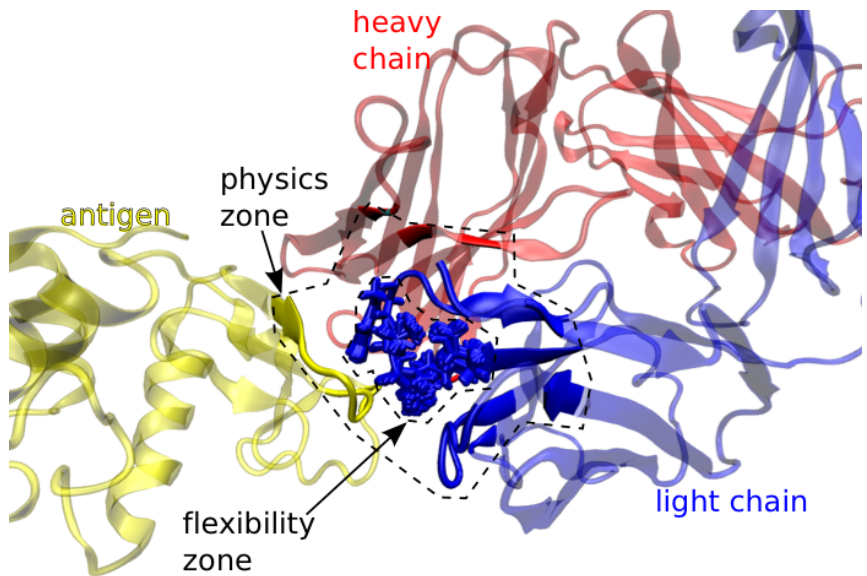




UPPSALA  
UNIVERSITET

# breeder 1.1



## User and developer tutorial

June 12, 2013

---



### **Copyright and Permission Notice**

Copyright (c) 2009 Samuel Flores, Uppsala University

For full copyright and permission notice, see the *Reference guide*.



# Acknowledgments

Ongoing development of breeder is funded by Uppsala University, Sweden.



# Table of Contents

<b>1</b>	<b>OVERVIEW .....</b>	<b>7</b>
<b>2</b>	<b>WORKING WITH SUBVERSION (SVN) .....</b>	<b>13</b>
<b>3</b>	<b>COMPILING .....</b>	<b>14</b>
<b>4</b>	<b>RUNNING .....</b>	<b>15</b>
4.1	Starting a new job .....	18
4.2	Useful MySQL commands: .....	18





# 1 Overview

breeder is a genetic algorithm based protein-protein interface optimizer.

## 2 Installing Simbody (SimTK)

MMB requires SimTK. You can get precompiled binaries from [SimTK.org](http://SimTK.org). However if you're in my lab you can just copy mine, since it doesn't get updated with all that much frequency. First, make a directory called:

```
~/SimTK
```

Then copy my SimTK to yours:

```
cp -Rf ~samuelf/SimTK/* ~/SimTK
```

## 3 Checking out MMB with subversion (svn)

Breeder requires MMB. You can get precompiled binaries from SimTK.org. However if you're in my lab you probably need to install from source so you can get updates easily. In that case, do the following. First, make a directory called:

```
~/svn/RNAToolbox/trunk
```

```
svn checkout https://simtk.org/svn/rnatoolbox/trunk ~/svn/rnatoolbox/trunk
```

If you are going to make changes to a file, it is good to make sure you have an up to date distribution:

```
cd ~/svn/rnatoolbox/trunk  
svn update .
```

Let's say you modify BiopolymerClass.cpp. The way to commit this to the repository is this:

```
svn commit ~/svn/ rnatoolboxtrunk/src/BiopolymerClass.cpp
```

If you get an error saying the editor has not been set, this means you have to issue

```
export SVN_EDITOR=vim
```

You should add this line to your `~/.bash_profile` so you don't do this every time.

Reference:

<http://www.techchorus.net/how-set-svneditor-environment-variable-vim>

## 4 Compiling MMB

Your LD\_LIBRARY\_PATH should point to wherever you have SimTK libraries, e.g. :  
`export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/SimTK/lib64`

You will then need to compile:

```
mkdir ~/svn/RNAToolbox/trunk/build
```

```
cd ~/svn/RNAToolbox/trunk/build
```

```
ccmake ..
```

At this point cmake may puke and tell you a newer version of cmake is needed. I have a 2.8 version here:

```
/bubo/home/h17/samuelf/cmake-2.8.7-Linux-i386
```

Make a similar directory in your home directory and copy the contents of the above into it. Uppmax is a bit difficult with user groups, so you may find it useful to copy it first to /scratch .

You will then find it useful to create an alias which overrides the default ccmake. Put something like this in your ~/.bash\_profile :

```
alias ccmake='/bubo/home/h17/samuelf/cmake-2.8.7-Linux-i386/bin/ccmake'
```

.. except substituting your own home directory of course.

Then check your cmake variables. The big pitfalls are the ones that locate your linked libraries ( SimTK). Also, breeder should for now be compiled in Debug mode. Here are values that work for me:

```

BUILD_MMB_GUI_TEST      OFF
BUILD_MMB_SHARED_EXEC    ON
BUILD_MMB_SHARED_LIB     ON
BUILD_MMB_STATIC        OFF
Boost_INCLUDE_DIR
CMAKE_AR                 /usr/bin/ar
CMAKE_BUILD_TYPE         Release
CMAKE_COLOR_MAKEFILE     ON
CMAKE_CXX_COMPILER       /usr/bin/c++
CMAKE_CXX_FLAGS
CMAKE_CXX_FLAGS_DEBUG    -g
CMAKE_CXX_FLAGS_MINSIZEREL -Os -DNDEBUG
CMAKE_CXX_FLAGS_RELEASE  -O3 -DNDEBUG
CMAKE_CXX_FLAGS_RELWITHDEBINFO -O2 -g
CMAKE_C_COMPILER         /usr/bin/gcc
CMAKE_C_FLAGS
CMAKE_C_FLAGS_DEBUG      -g
CMAKE_C_FLAGS_MINSIZEREL -Os -DNDEBUG
CMAKE_C_FLAGS_RELEASE    -O3 -DNDEBUG
CMAKE_C_FLAGS_RELWITHDEBINFO -O2 -g
CMAKE_EXE_LINKER_FLAGS
CMAKE_EXE_LINKER_FLAGS_DEBUG
CMAKE_EXE_LINKER_FLAGS_MINSIZE
CMAKE_EXE_LINKER_FLAGS_RELEASE
CMAKE_EXE_LINKER_FLAGS_RELWITH
CMAKE_EXPORT_COMPILE_COMMANDS OFF
CMAKE_INSTALL_PREFIX     /usr/local
CMAKE_LINKER             /usr/bin/ld

```

```

CMAKE_MAKE_PROGRAM      /usr/bin/gmake
CMAKE_MODULE_LINKER_FLAGS
CMAKE_MODULE_LINKER_FLAGS_DEBU
CMAKE_MODULE_LINKER_FLAGS_MINS
CMAKE_MODULE_LINKER_FLAGS_RELE
CMAKE_MODULE_LINKER_FLAGS_RELW
CMAKE_NM                /usr/bin/nm
CMAKE_OBJCOPY           /usr/bin/objcopy
CMAKE_OBJDUMP           /usr/bin/objdump
CMAKE_RANLIB            /usr/bin/ranlib
CMAKE_SHARED_LINKER_FLAGS
CMAKE_SHARED_LINKER_FLAGS_DEBU
CMAKE_SHARED_LINKER_FLAGS_MINS
CMAKE_SHARED_LINKER_FLAGS_RELE
CMAKE_SHARED_LINKER_FLAGS_RELW
CMAKE_SKIP_RPATH        OFF
CMAKE_STRIP             /usr/bin/strip
CMAKE_USE_RELATIVE_PATHS OFF
CMAKE_VERBOSE_MAKEFILE  OFF
SimTKmolmodel           /bubo/home/h17/samuelf/SimTK
SimTKsimbody            /bubo/home/h17/samuelf/SimTK

```

Now compile ..

```

[samuelf@tintin1 build]$ pwd
/bubo/home/h17/samuelf/svn/rnatoobox/trunk/build
[samuelf@tintin1 build]$ make -j8

```



## 5 Checking out breeder with subversion (svn)

To install , do this:

```
mkdir ~/svn/breeder  
svn checkout https://simtk.org/svn/breeder ~/svn/breeder
```

If you are going to make changes to a file, it is good to make sure you have an up to date distribution:

```
cd ~/svn/breeder  
svn update .
```

Let's say you modify Breed.cpp. The way to commit this to the repository is this:

```
svn commit ~/svn/breeder/src/Breed.cpp
```

If you get an error saying the editor has not been set, this means you have to issue

```
export SVN_EDITOR=vim
```

You should add this line to your ~/.bash\_profile so you don't do this every time.

Reference:

<http://www.techchorus.net/how-set-svneditor-environment-variable-vim>

## 6 Compiling breeder

Your LD\_LIBRARY\_PATH should point to wherever you have libMMBlib.so, e.g. :

```
export LD_LIBRARY_PATH=~/.svn/RNAToolbox/trunk/build
```

You will then need to compile:

```
cd ~/.svn/breeder/build
```

```
ccmake ..
```

Then check your cmake variables. The big pitfalls are the ones that locate your linked libraries (mysql, SimTK, MMB).

Now compile ..

```
[samuelf@tintin1 build]$ pwd
/bubo/home/h17/samuelf/.svn/breeder/build
[samuelf@tintin1 build]$ make -j8
Linking CXX executable breeder
[100%] Built target breeder
[samuelf@tintin1 build]$
```



## 7 Running

To run, use:

```
~/svn/breeder/build/breeder -ID 14MAY13 -MLF ./mutations-list.txt
```

Where the file specified by `-MLF` contains as many mutants as you wish, in the format : X-NNI-S.x-nni-s where X is the chain ID, NNI is the residue ID and insertion code (if any) and S is the substituted residue type. Separate the mutants with “.” For example:

```
C-128-V.C-147-H.C-163-N
```

```
C-128-R.C-147-H.C-163-N
```

Breeder will fetch all run parameters (except the actual mutations) from the mysql jobInfo table:

```
mysql> select * from jobInfo;
```

```
+-----+-----+
-----+-----+
-----+-----+
---+-----+
-----+
-----+
-----+
-----+
-----+
-----+
-----+
-----+
```

	runID					
batch_directory						
	maxMutants		generationSize		maxGenerations	
decayConstantsBetweenHighestAndLowest					wildTypeSequence	
	reportingInterval			numReportingIntervals		
wildTypeStructureFileName						
+-----+						
						+-----+
+-----			+-----			
---+						
						+-----
+-----			+-----			
-----+						
	14MAY13					
/bubo/home/h17/samuelf/glob/projects/protein-A/14MAY13						
4		2				1000
5						
C:FNKEQQNAFYEILHLPNLNEEQRNGFIQSLKDDPSQSANLLAEA.D:PSVFLFPPKPKDTLMISRT						
PEVTCVVVDVSHEDPQVKFNWYVDGVQVHNAKTKPREQQYNSTYRVVSVLTVLHQNWLDGKEYKCKVS						
NKALPAPIEKTISKAKGQPREPQVYTLPPSREEMTKNQVSLTCLVKGFYPSDIAVEWESNGQPENNYK						
TTPPVLDSDGSFFLYSKLTVDKSRWQQGNVFSCSVMHEALHNHYTQKSLSLS						
1						15
/bubo/home/h17/samuelf/glob/projects/protein-A/last.1.pdb						
+-----+						
						+-----+
+-----			+-----			
---+						
						+-----
+-----			+-----			
-----+						
7 rows in set (0.00 sec)						

```
mysql>
```

You can change `reportingInterval` and `numReportingIntervals` on `jobInfo`, but as yet not other MMB parameters. If you wish to modify MMB's other parameters, to go into `commands.dat`, modify `src/Chromosome.cpp` method `writeCommonPart`:

```
void Chromosome::writeCommonPart(std::ofstream & output, MySqlConnection &
mysqlConnection) {
```

```
    output <<"firstStage 2 "<<std::endl;
    output <<"lastStage 2"<<std::endl;
    output <<"loadSequencesFromPdb "<<std::endl;
    output <<"smallGroupInertiaMultiplier 11 "<<std::endl;
    output <<"#reportingInterval .000000000000000000000001"<<std::endl;
    output <<"reportingInterval ";
    output <<mysqlConnection.getReportingInterval();
    output <<std::endl;
    output <<"numReportingIntervals "; //150 "<<std::endl;
    output << mysqlConnection.getNumReportingIntervals();
    output <<std::endl;
    output <<"#numReportingIntervals 1 "<<std::endl;
    output <<"readAtStage 2"<<std::endl;
    output <<"globalBondTorsionScaleFactor 1"<<std::endl;
    output <<"globalBondBendScaleFactor 1"<<std::endl;
    output <<"globalCoulombScaleFactor 1"<<std::endl;
    output <<"globalVdwScaleFactor 1"<<std::endl;
    output <<"globalAmberImproperTorsionScaleFactor 1"<<std::endl;
    output <<"readBlockEnd"<<std::endl;
    output <<"randomizeInitialVelocities false"<<std::endl;
    output <<"mobilizer Rigid"<<std::endl;
    output <<"constrainChainRigidSegments"<<std::endl;
```

```
};
```

You will then need to recompile:

```
[samuelf@tintin1 build]$ pwd
/bubo/home/h17/samuelf/svn/breeder/build
[samuelf@tintin1 build]$ make -j8
Linking CXX executable breeder
[100%] Built target breeder
[samuelf@tintin1 build]$
```

## 7.1 Starting a new job

First, go to the jobInfo table and specify the parameters for the job, e.g.:

```
insert into jobInfo (runID , batch_directory , ,reportingInterval ,numReportingIntervals ,
wildTypeStructureFileName)
VALUES ("17MAY13" ,
"/bubo/home/h17/samuelf/glob/projects/recA/17MAY13" , 0.1, 150,
"/bubo/home/h17/samuelf/glob/projects/recA/last.1.pdb");
```

Now put all your mutants in a text file. Name whatever you want (e.g. myMutants.txt). Then run as:

```
./breeder -ID 17MAY13 -MLF ./myMutants.txt
```

## 7.2 Useful MySQL commands:

Include update and delete

```
update jobInfo set lastStage = 3 where runID = "14MAY13";
```

Be very careful with the delete command! Always run it with a carefully selected “where” restriction:

```
delete from jobInfo where runID = "14MAY13";
```

