MolTK



User's Manual

Release 0.4.0 November 4, 2011

Website: rotatingpenguin.com/moltk

Copyright and Permission Notice

Copyright (c) 2011 Christopher M. Bruns Contributors: Cami K. Bruns

Permission is hereby granted, free of charge, to any person obtaining a copy of this document (the "Document"), to deal in the Document without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Document, and to permit persons to whom the Document is furnished to do so, subject to the following conditions:

This copyright and permission notice shall be included in all copies or substantial portions of the Document.

THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS, CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT.

Acknowledgments

Special thanks to Joy Ku who created the beautiful template for this manual. Thanks also to Christopher D. Putnam and Robert Edgar for helpful discussions about sequence alignment.

Table of Contents

1	IN	ГROD	OUCTION TO MOLTK	1
	1.1	Overv	riew	1
		1.1.1	What is MolTK?	1
		1.1.2	Design Goals of MolTK	1
	1.2	Instal	ling MolTK	2
		1.2.1	Installing the MolTK Application	2
		1.2.2	Installing the MolTK Python Module	2
		1.2.3	Installing the MolTK C++ SDK	
		1.2.4	Building MolTK from Source Code	2
	1.3	Softw	are License	2
	1.4	Resou	ırces	3
2	TH	Е МО	OLTK APPLICATION	5
3			AMMING MOLTK	
	3.1	Pytho	n	···· 7
		3.1.1	Tutorial	7
		3.1.2	MolTK for SEQUOIA Users	9
	3.2	C++		12
		3.2.1	MolTK Coding Style Guidelines	
4	BI	BLIO	GRAPHY	. 15

List of Tables

No table of figures entries found.

List of Figures

No table of figures entries found.

List of Examples

1 Introduction to MolTK

1.1 Overview

1.1.1 What is MolTK?

There are three faces of MolTK; in order from easiest-to-use to most-powerful:

- 1. MolTK is a computer application that allows the user to view and align molecular sequences and structures.
- 2. Moltk is a Python programming language module that allows the user to align and otherwise compute on molecular sequences and structures.
- 3. Moltk is a C++ API that allow the programmer to compute with molecular sequences and structures with high efficiency.

1.1.2 Design Goals of MolTK

- Python programming language interactive environment for sequence/structure alignment that is just as easy to use as our (aging) dedicated alignment tool SEQUOIA.
- Flexible architecture that makes it easy to experiment with custom alignment methods and scoring systems.
- Consistent, well documented API for both Python and C++ programmers.
- Units-aware quantity type system. Thus alignment scores are not just numbers, they are information quantities with units of "bits". Atomic coordinates are not just numbers x, y, z, but are vector quantities with units of nanometers. Units-aware quantity types are an important part of scientific computing hygiene:
 - o Converting a "quantity" to a raw number requires a "unit" to express the quantity in. This requires the user to pay attention to units at precisely the

- moment when knowing the unit is most important. The rest of the time is "just works", even if you are wrong about what the current units are!
- This sort of type safety in scientific computing might help prevent errors such as, say, crashing \$100 million orbiters into planets.
- Type safety: Adding a length to a volume makes no sense, and will result in an error. Dividing a length by a time results in a velocity.
- Next-generation molecular sequence/structure viewer that adheres to our user interface principles.

1.2 Installing MolTK

MolTK download site: http://code.google.com/p/moltk/downloads/list

- 1.2.1 Installing the MolTK Application
- 1.2.2 Installing the MolTK Python Module
- **1.2.3** Installing the MolTK C++ SDK
- 1.2.4 Building MolTK from Source Code
- 1.3 Software License

Copyright (C) 2011 Christopher M. Bruns

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Commercial users should ask about our dual licensing model.

For questions contact: cmbruns@rotatingpenguin.com

1.4 Resources

Browse the MolTK home page at:

http://rotatingpenguin.com/moltk/

Download MolTK at:

http://code.google.com/p/moltk/downloads/list

Find questions and answers in the moltk-users forum at:

http://groups.google.com/group/moltk-users

Report issues and get the source code at:

http://code.google.com/p/moltk/

Study the MolTK Python API at:

http://www.rotatingpenguin.com/moltk/api_python/python_api.html

Study the MolTK C++ API at:

http://www.rotatingpenguin.com/moltk/api_cxx/cxx_api.html

2 The MolTK Application

3 Programming MolTK

3.1 Python

3.1.1 Tutorial

Download and install MolTK from the MolTK download site: http://code.google.com/p/moltk/downloads/list

In the following examples text following the >>> prompt represents commands typed by the user. Other lines represent the output of the program.

3.1.1.1 Scenario 1: Aligning 3 Sequences

In this example, the insulin_shark_v1.fasta, insulin_pig_v1.fasta, and insulin_human_v1.fasta are sequences in the FASTA format. Information about the FASTA format can be found at http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml. A file called test.fasta can be saved with the alignment. The alignment can be also saved as test.pretty which is a nicely formatted but not computer parsable version of the alignment. What follows is an example of performing an alignment of three sequences.

3.1.1.1.1 Step 1. Download FASTA Test Sequences

Go to download page http://code.google.com/p/moltk/downloads/list and download the three FASTA sequences: insulin_shark_v1.fasta, insulin_pig_v1.fasta, and insulin_human_v1.fasta.

3.1.1.1.2 Step 2. Start an Alignment Session

```
% python -i
>>> import moltk
```

```
3.1.1.1.3 Step 3. Load test1.fasta Sequence into Seq1
>>> seq1 = moltk.load_fasta("insulin_human_v1.fasta")
3.1.1.1.4 Step 4. Load test2.fasta Sequence into Seq2
>>> seq2 = moltk.load fasta("insulin_pig_v1.fasta")
3.1.1.1.5 Step 5. Align Seq1 and Seq2
>>> align1 = moltk.align(seq1, seq2)
3.1.1.1.6 Step 6. Display the Alignment on the Screen
>>> print align1
3.1.1.1.7 Step 7. Load test3.fasta Sequence into Seq3
>>> seq3 = moltk.load fasta("insulin_shark_v1.fasta")
3.1.1.1.8 Step 8. Align Previous Alignment of test1.fasta and test2.fasta with
         test3.fasta
>>> align2 = moltk.align(align1, seq3)
3.1.1.1.9 Step 9. Save the Alignment to a File
>>> align2.write_fasta("test.fasta")
You can also save as test.pretty for a nicely formatted but not computer parseable version.
>>> align2.write_pretty("test.pretty")
3.1.1.1.10 Step 10. Display a Table of Pairwise Sequence Identities
>>> print align2.id_table()
```

3.1.1.1.11 Step 11. End Your Alignment Session >>> quit()

3.1.1.2 Scenario 2: Writing a Python Script to Help Automate the Alignment Process

3.1.1.3 Scenario 3: Aligning 2 Sequences and Assessing Significance

3.1.1.4 Scenario 4: Overlaying 2 tertiary structures

3.1.2 MolTK for SEQUOIA Users

Sequoia Command	Corresponding MolTK Command(s)	Description
% sequoia	% python -i	Start an alignment
% sequota	import moltk	session
SEQUOIA> read SEQ1	>>> seq1 =	Load a sequence or
test1.fasta	<pre>moltk.load_fasta("test1.fasta")</pre>	alignment into seq1
SEQUOIA> read SEQ2	>>> seq2 =	Load a sequence or
test2.fasta	<pre>moltk.load_fasta("test2.fasta")</pre>	alignment into seq2
SEQUOIA> align	<pre>>>> align1 = moltk.align(seq1,</pre>	Align two
SLQUOIA/ align	seq2)	sequences/alignments
SEQUOIA> print ALIGN	>>> print align1	Display alignment on
Seguotas print actum	777 princ arigni	screen
SEQUOIA> set SEQ1	>>> seq1 = align1	Copy alignment to seq1
ALIGN	777 Seq1 - diigni	copy anginnent to sequ
SEQUOIA> write ALIGN	<pre>>>> align1.write_fasta("test.fasta")</pre>	Save alignment to a file
test.fasta	/// diigni.write_rasta(test.rasta)	Save angument to a me

SEQUOIA> print ALIGN >>> but not computer parsable, version of alignment to a file SEQUOIA> print id ALIGN >>> print align1.id_table() SEQUOIA> quit >>> quit() SEQUOIA> @test.inp >>> execfile("test.py") Save nicely formath but not computer parsable, version of alignment to a file Display a table or pairwise sequence identities SEQUOIA> Quit >>> quit() End alignment sess Run a script file you wrote Run a script you wrote	
test.pretty align1.write_pretty("test.pretty") parsable, version of alignment to a file SEQUOIA> print id ALIGN >>> print align1.id_table() pairwise sequence identities SEQUOIA> quit >>> quit() End alignment sess SEQUOIA> @test.inp >>> execfile("test.py") Run a script file you wrote	
SEQUOIA> print id ALIGN >>> print align1.id_table() SEQUOIA> quit >>> quit() SEQUOIA> @test.inp >>> execfile("test.py") alignment to a file Display a table or pairwise sequence identities Run a script file you wrote	
SEQUOIA> print id ALIGN >>> print align1.id_table() SEQUOIA> quit >>> quit() SEQUOIA> @test.inp >>> execfile("test.py") Display a table or pairwise sequence identities Run a script file you wrote	on
SEQUOIA> print id ALIGN >>> print align1.id_table() pairwise sequence identities SEQUOIA> quit >>> quit() SEQUOIA> @test.inp >>> execfile("test.py") Run a script file you wrote	on
ALIGN >>> print align1.id_table() pairwise sequence identities SEQUOIA> quit >>> quit() SEQUOIA> @test.inp >>> execfile("test.py") Run a script file you wrote	ion
SEQUOIA> quit >>> quit() End alignment sess SEQUOIA> @test.inp >>> execfile("test.py") Run a script file you wrote	ion
SEQUOIA> @test.inp >>> execfile("test.py") Run a script file you wrote	ion
SEQUOIA> @test.inp	
wrote	l
Run a script you wr	
·	ote
% sequoia < testp.inp % python test.py directly from a com	mand
prompt	
SEQUOIA> COMMENT This	
is a comment Statement is ignore	1
SEQUOIA> help Access documentat	on
<pre></pre>	
SEQUOIA> system >>> import os Issue an operating	
<pre><command/></pre>	
Create a consensus	
SEQUOIA> consensus sequences with X's	at
<pre> <sigma></sigma></pre>	tions
Improve an alignme	ent by
SEQUOIA> optimize successively removi	ng
and realigning sequ	ences
Display the current	
SEQUOIA> print matrix scoring matrix	
SEQUOIA> read matrix	
test.mat Load a scoring matr	1X
Calculate 5 alignme	nt
SEQUOIA> random 5 scores with SEQ2	
randomly shuffled	
Remove sequence 5	from
SEQUOIA> split 5 alignment and place	e it in
seq1	

Sequoia Command	Corresponding MolTK Command(s)	Description
SEQUOIA> weight		Apply sequence weights
		to alignment
SECULOTAL set enen 0 5		Set gap extension
SEQUOIA> set epen 0.5		penalty
SEQUOIA> set gpen 10		Set gap opening penalty
SEQUOIA> set		Set sequence width of
pretty_length 50		formatted alignments
SEQUOIA> set		Change seed for
random_seed 1		RANDOM command
SEQUOIA> set		Parameter for allowing
suboptimal 0.1		suboptimal alignments
SEQUOIA> read STRUCT1		Load a structure into
test1.pdb		struct1
SEQUOIA> salign		Create a structure based
SEQUUIA> Salign		alignment
SEQUOIA> overlay		Superpose two
SEQUOTAS OVERTAY		structures
		Assign equivalent
SEQUOIA> equivalence		residues between 2
		structures
SEQUOIA>set acutoff		Max rotation difference
		for structure alignment
45		in degrees
SEQUOIA>set dcutoff		Max distance difference
4.5		for structure alignment
4.5		in angstroms
SEQUOIA> set		Min length required for a
_		run of structurally
runlength 4.5		equivalent residues
SEQUOIA> set useangle		Whether to use rotation
		similarity in structure
		alignment
SEQUOIA> tabulate		Populate pairwise
	There is no tabulate command in MolTK	residue scores before
		aligning (seldom used)
L	1	I

Sequoia Command	Corresponding MolTK Command(s)	Description	
SEQUOIA> stabulate	There is no stabulate command in MolTK	Populate pairwise	
		residue structure scores	
		before aligning (seldom	
		used)	
SEQUOIA> set echo 1	Don't know how to do this in python	Output a copy of user's	
		commands	

3.2 C++

3.2.1 MolTK Coding Style Guidelines

Purpose: To provide a consistent look for both python and C++ MolTK code.

3.2.1.1 Indentation

Use four space characters per indent. No tabs.

3.2.1.2 *Class names*

Use CapitalizedWords

Use all caps for abbreviations, e.g. PDBStructure.

Class names should be nouns.

3.2.1.3 Macro, enum, and constant names

CAPITALIZED_WITH_UNDERSCORES

3.2.1.4 Method and function names

Use lower_case_with_underscores.

Begin method and function names with a verb.

3.2.1.5 Attribute and member names

lower_case_with_underscores

adjective_noun or noun

b_variable_name for boolean values

3.2.1.6 Namespace, package, and module names

lower case, single word, short names

3.2.1.7 Method parameters

lower_case_with_underscores

Remember that method parameter names are more important in python than in C++, because python allows named parameter use. In all C++ header files, every parameter to every exposed method should have an understandable name.

4 Bibliography

Dayhoff, M. O., 1969. *Atlas of Protein Sequence and Structure*. Silver Spring, MD: National Biomedical Research Foundation.

Edgar, R. C., 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5), pp. 1792-1797.

Gusfield, D., 1997. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press.

Henikoff, S. & Henikoff, J. G., 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, Volume 89, pp. 10915-10919.

Needleman, S. & Wunsch, C. D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, Volume 48, pp. 443-453.