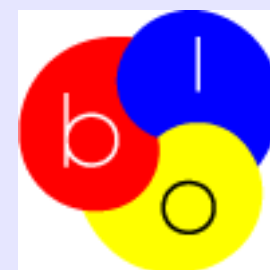


```
AGTGTGCACGATATCG BioInt : Biobhasha interpreter (www.biobhasha.org)
TGGAGAGAGGAGCGCG Copyright(c) : 1998-2013 Dr. Prasad Burra VLS (admin@biobhasha.org)
CTTGCCCATTTGCTGG Authors : Dr. Prasad Burra VLS, Sanket Desai
TGGTCGGCGGGAGAAC Version : Biobhasha 2.0
TAGACAATGTCGAGG Masaharu Goto is duly acknowledged for design and development of CINT.
GCCAAGCAAAGGAAC
CTGCCGGTGCGAATGC No main() function found in given source file. Interactive interface started.
AAAGATATCTTCAGAT 'h':help, 'q':quit, '{statements;}' or 'p [expr]' to evaluate
ATCATTACCACAATTC BioInt> { BioPoint x(2.5,3.7,8.4),y(7.4,9.4,2.7);
AAAGTAATGGCCAGGG end with '}', '@':abort > cout<<"Distance between point X & Y : "<<
CTCGGCAACTTGGCCG end with '}', '@':abort > BioDistance(x,y) << endl;}
AATCTGAGAGATGTGA Distance between point X & Y : 9.43345
GAAGACGGCTCCTGGG (class ostream)155495328
GGAGCAGGTCAATGTG BioInt> _
CGCCTCATGACACAGG
```

Introduction to BioInt

- by Sanket Desai.



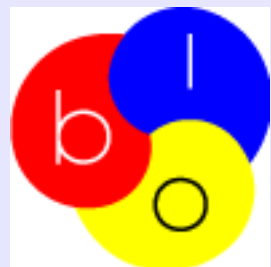
Outline

- ✓ Object Oriented Paradigm
- ✓ Biobhasha
- ✓ CINT
- ✓ BioInt
- ✓ Getting BioInt
- ✓ Using BioInt



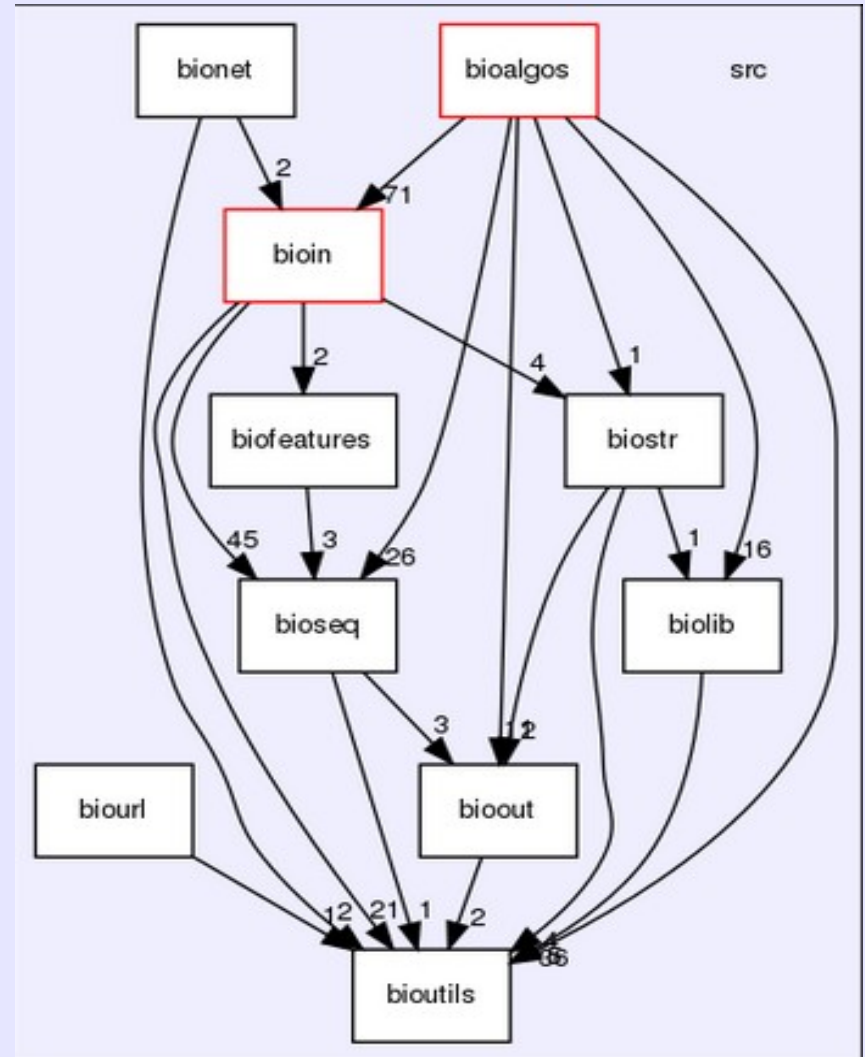
Object Oriented Paradigm

- ◆ Object oriented design (OOD) makes the application modular, re-usable and extensible.
- ◆ Object oriented application framework (OOAF) is a domain specific, 'semi-complete' application that can be specialized to produce custom applications
- ◆ **ROOT** , **R**, **MatlabTM** are some of the highly successful programming environments in respective domains taking advantage of OOP designs / concepts.
- ◆ Biology is inherently object oriented. Biological entities holds certain data/information and a unique associated behavior – a motivation to design an OO framework for biology
- ◆ Biobhasha/ BioInt is an attempt to design & develop Biology specific OOAF and a bio-programming environment.

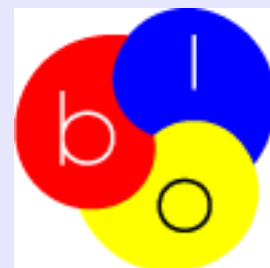


Biobhasha

- ◆Collection of large number of Biological Abstract Data Types (BioADTs) implemented as *classes*.
- ◆Written in C++, with more the 450 biological objects (truly object oriented).
- ◆Framework consists of 10 modules (as shown in figures), based on the biological data type, algorithms and functionality.



Modules in Biobhasha



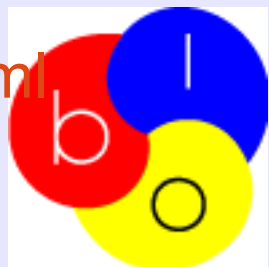
Features of Biobhasha

Current version of Biobhasha has;

- 14 Algorithm implementations
- 17 Database input formats
- 3 Output formats
- 5 Biological Libraries
- Several sequence, structure and network (interaction) specific objects
- Statistics and matrix functions and other miscellaneous classes and functions.

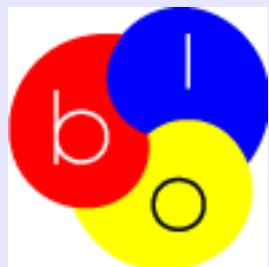
API documentation at

<http://www.biobhasha.org/docs/biointdocs/html/classes.html>



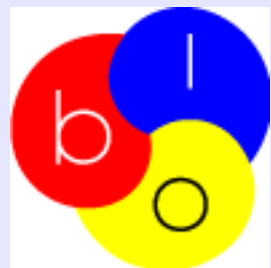
CINT

- ◆ Developed by Masaharu Goto (available at <http://root.cern.ch/drupal/content/cint>).
- ◆ BioInt uses a C++ interpreter CINT to create a command line/ script interpreter.
- ◆ Allows embedding compiled C/C++ library code and dynamic loading/ unloading of libraries.
- ◆ It allows use of special commands:
 - 'q' quit
 - 'x' script.C execute script 'script.C'
 - 'L' script.C load script 'script.C' (if it contains class definition)
 - ... and more.



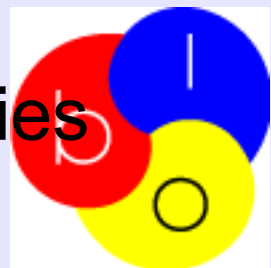
BioInt

- ♦Linux based application for biological data mining and analysis from various biological databases and algorithms, be it genomics, structure analysis, literature mining, systems biology approaches or all together.
- ♦Interpreter for object-oriented application framework.
(**Biobhasha**)
- ♦An interactive bio-programming environment to implement custom biological workflows.



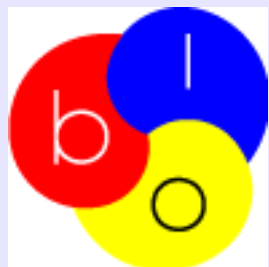
Features of BioInt

- ◆ Bio-scripting using C++ syntax.
- ◆ 'Write' & 'Run' the instructions interactively with an interpreter.
- ◆ Gives flexibility in designing custom bio-analytical workflows.
- ◆ Access to various preloaded BioADTs and utility functions as well as allows creation of ADTs of your own.
- ◆ Seamless extensibility by reuse of old community contributed bio-scripts.
- ◆ Dynamic loading/ unloading external source libraries



Getting BioInt

- ◆ Current version of BioInt is Version 1.02.
- ◆ Visit **Download** page at Biobhasha website: <http://www.biobhasha.org/#downloads> to download current version.
- ◆ Ones downloaded, refer '*README.txt*' in the <bioint-version> directory for installation instructions.
- ◆ Example scripts can be downloaded at www.biobhasha.org/sampleScripts/bioint_demo_scripts.tar.gz
- ◆ Comments/queries/suggestions/bug reports can be sent to admin@biobhasha.org



Using BioInt

BioInt can be used in two modes;

1. Interactive session mode

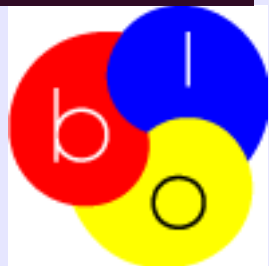
```
neo@neo:~$ BioInt

BioInt : Biobhasha interpreter (www.biobhasha.org)
  Copyright(c) : 1998~2013 Dr. Prasad Burra VLS (admin@biobhasha.org)
  Authors      : Dr. Prasad Burra VLS, Sanket Desai
  Version      : Biobhasha 2.0
  Masaharu Goto is duly acknowledged for design and development of CINT.

No main() function found in given source file. Interactive interface started.
'h':help, 'q':quit, '{statements;}' or 'p [expr]' to evaluate

BioInt> {
end with '}', '@':abort > BioPoint x(2.5,5.7,1.7),y(6.3,9.4,3.6);
end with '}', '@':abort > cout << "Distance between point X & Y : " << BioDistance(x,y)<< endl;
end with '}', '@':abort > }
Distance between point X & Y : 5.63383
(class ostream)155462560
BioInt> _
```

Figure demonstrating use of interactive session to run a script



2. Command line mode

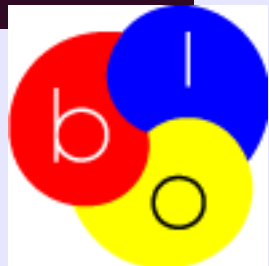
```
Points.C
class Points
{
    BioPoint x,y;
public:
    Points(){BioPoint m(2.5, 5.7, 1.7),n(6.3, 9.4, 3.6); x = m; y = n;}
    printDistance(){ cout << "\tDistance between point X & Y : " << BioDistance(x,y) << endl; }
};

demo.C
#include "Points.C"
void demo()
{
    Points p;
    p.printDistance();
}

main.C
#include "demo.C"
void main()
{
    demo();
}

sanket@matrix:~$ BioInt main.C
Distance between point X & Y : 5.63383
sanket@matrix:~$
```

Figure demonstrating use of command line session to run a user defined script



Authors

Sanket S. Desai

Dr. Prasad Burra VLS

Contact:

admin@biobhasha.org

