

# Perl – For Bioinformatics

BIOPERL

Amarnath Raj

# P E R L

- Practical Extraction and Reporting Language
- Created by Larry Wall and thousand of others
- Evolved from C and Unix shell syntax
- The preferred language of Unix sysadmins
- Heavy use of regular expressions
- Extensive libraries, object orientation
- Bio::Perl is a library of PERL

# Perl - “Hello World” Example

```
print ("Hello World\n");
```

*There's More Than One Way To Do It*

**3rd Edition**  
Revised & Updated



*Programming*

**Perl**

**O'REILLY®**

*Larry Wall, Tom Christiansen & Jon Orwant*

Language Defining Book

There's More than one way to do it.

# Perl Syntax

- Comments start with #
- Variables
  - \$abc – Scalar
  - @abc – Arrays
  - %abc – Hashes
- Statements
  - Simple – XXXXX;
  - Compound – { Simple; Simple; }
- First line to be #!/usr/bin/perl to run program as a shell script

# Perl – Language constructs

- `if (EXPR) { ... }`
- `if (EXPR) { ... } else { ... }`
- `if (EXPR) { ... } elsif (EXPR) { ... } else { ... }`
- `while (EXPR) { ... }`
- `for (EXPR; EXPR; EXPR) { ... }`
- `foreach VAR (LIST) { ... }`

# Perl – Interesting constructs

- Reverse usage
  - unless (reverses if)
  - until (reverses while)
- Statement modifiers
  - `$a = 20 if $b == 30;`
  - `print $a++ until $a == 200;`
  - `die ('Open Failed') unless(open(FD,"test.txt"));`

# Files

- Opening
  - open (<HANDLE>, “Name”);
  - open (<HANDLE>, “>name”);
  - open (<HANDLE>, “l name”);
  - open (<HANDLE>, “name l”);
- Reading
  - \$line = <HANDLE>
  - \$line = <STDIN>

# Perl - Regular Expressions

- Perl's regular expressions make it the most useful language for Bio-Informatics
  - Perl has an extensive array of regular expressions.
  - Regular expressions can become very complex
  - Very fast sub-string search
  - Comparison and replace

# Perl Resources

- <http://perl.org>
  - The perl site
- <http://www.activestate.com>
  - For Windows users
- <http://cpan.perl.org>
  - The Comprehensive Perl Archive Network

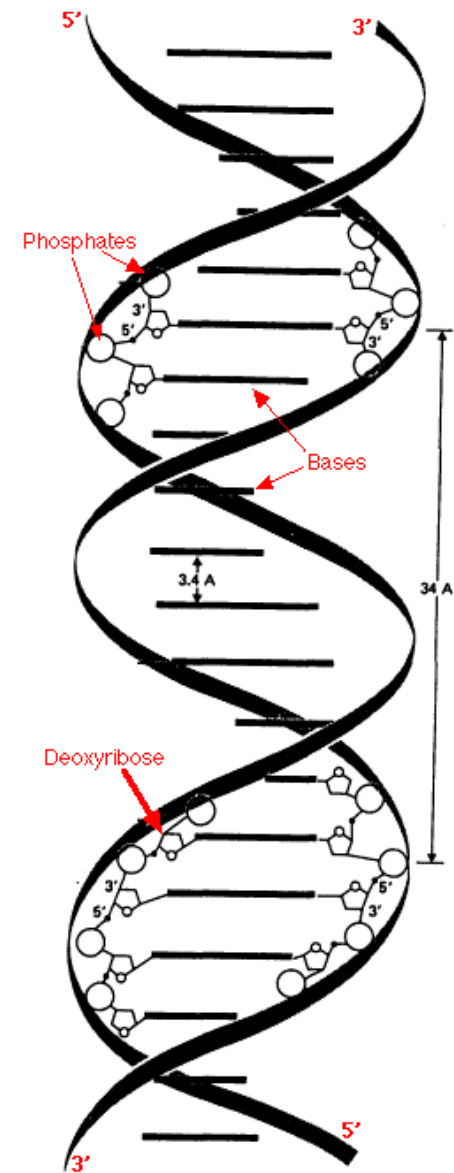
# BioInformatics

**“This structure  
has novel features  
which are of considerable biological  
interest”**

In 1953, James Watson and Francis Crick wrote these words as part of the opening paragraph of a letter to Nature Magazine.

# DNA

- DeoxyriboNucleic acid
- DNA is the Molecule of Inheritance in an organism
- DNA is a polymer of four Nucleotides  
Adenine ( A ),Guanine ( G ),Thymine ( T ),Cytosine ( C )
- DNA in a cell consists of two strands of the nucleotide chains held together in a double helix structure



# DNA representation on a Computer

- DNA is a “STRING” of characters comprised of a four letter subset of the English alphabet
- “agtctgatcatgagatcatagcatgatgaca.....”
- Human Genome is about 3GB long if one alphabet occupies one byte.
- Ecoli is about 4KB
- These “Strings” are available on the Internet and downloadable into files. There are some popular “File Formats” to store these FASTA, GenBank

# Amino Acids & Proteins

- Proteins form a large part of our bodies
- Proteins are composed of “Amino Acids”
- There are only 20 Amino Acids that form all the proteins in living organisms.
- Hence Proteins can be described for computational purposes as:
- A string of characters comprised of a twenty letter subset of the English alphabet

# The Central Dogma of BioInformatics

## RNA, Transcription and Translation

- RNA is similar to the DNA in composition
- It has a Nucleotide Uracil instead of Thymine
- RNA is manufactured using DNA as a template
- DNA used to manufacture an RNA is a “gene”
- This process is called “**Transcription**”
- The RNA is then “**Translated**” into Protein
- This process is called the “Dogma of Molecular Biology” and now of BioInformatics

# RNA Translation into Protein

- There are 20 Amino Acids in proteins
- There are 4 Nucleotides in RNA
- Hence a combinations of three Nucleotides (4x4x4) are required to represent Amino Acids
- A set of three Nucleotides is called a “Codon”
- The Nucleotides in an RNA is read in sets of “codons”, and translated into amino Acids.
- This chain of amino acids is a “Protein”

Second nucleotide

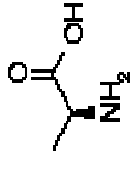
	U	C	A	G	
U	UUU <b>Phenylalanine</b> (Phe)	UCU <b>Serine</b> (Ser)	UAU <b>Tyrosine</b> (Tyr)	UGU <b>Cysteine</b> (Cys)	U
	UUC Phe	UCC Ser	UAC Tyr	UGC Cys	C
	UUA <b>Leucine</b> (Leu)	UCA Ser	UAA <b>STOP</b>	UGA <b>STOP</b>	A
	UUG Leu	UCG Ser	UAG <b>STOP</b>	UGG <b>Tryptophan</b> (Trp)	G
C	CUU <b>Leucine</b> (Leu)	CCU <b>Proline</b> (Pro)	CAU <b>Histidine</b> (His)	CGU <b>Arginine</b> (Arg)	U
	CUC Leu	CCC Pro	CAC His	CGC Arg	C
	CUA Leu	CCA Pro	CAA <b>Glutamine</b> (Gln)	CGA Arg	A
	CUG Leu	CCG Pro	CAG Gln	CGG Arg	G
A	AUU <b>Isoleucine</b> (Ile)	ACU <b>Threonine</b> (Thr)	AAU <b>Asparagine</b> (Asn)	AGU <b>Serine</b> (Ser)	U
	AUC Ile	ACC Thr	AAC Asn	AGC Ser	C
	AUA Ile	ACA Thr	AAA <b>Lysine</b> (Lys)	AGA <b>Arginine</b> (Arg)	A
	AUG <b>Methionine</b> (Met) or <b>START</b>	ACG Thr	AAG Lys	AGG Arg	G
G	GUU <b>Valine</b> Val	GCU <b>Alanine</b> (Ala)	GAU <b>Aspartic acid</b> (Asp)	GGU <b>Glycine</b> (Gly)	U
	GUC (Val)	GCC Ala	GAC Asp	GGC Gly	C
	GUA Val	GCA Ala	GAA <b>Glutamic acid</b> (Glu)	GGA Gly	A
	GUG Val	GCG Ala	GAG Glu	GGG Gly	G

# Amino Acids

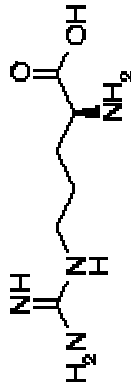
A	Alanine	N	Asparagine
B	Aspartic Acid, Asparagine	P	Proline
C	Cystine	Q	Glutamine
D	Aspartic Acid	R	Arginine
E	Glutamic Acid	S	Serine
F	Phenylalanine	T	Threonine
G	Glycine	V	Valine
H	Histidine	W	Tryptophan
I	Isoleucine	X	Unknown
K	Lysine	Y	Tyrosine
L	Leucine	Z	Glutamic Acid, Glutamine
M	Methionine	*	Terminator



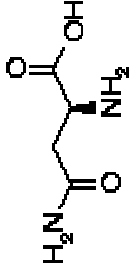
gly g Glycin



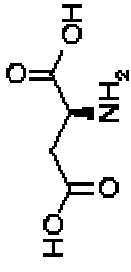
ala a Alanin



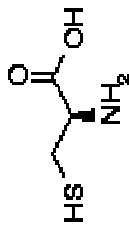
arg r Arginin



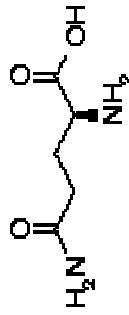
asn n Asparagin



asp d Asparaginsäure



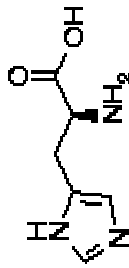
cys c Cystein



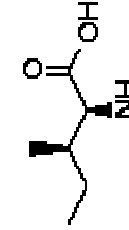
gln q Glutamin



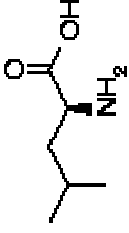
glu e Glutaminsäure



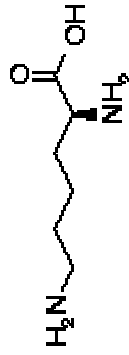
his h Histidin



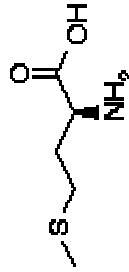
ile i Isoleucin



leu l Leucin



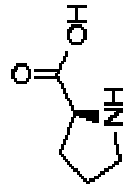
lys k Lysin



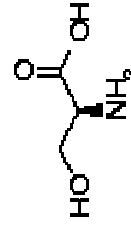
met m Methionin



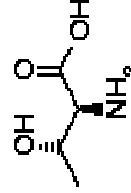
phe f Phenylalanin



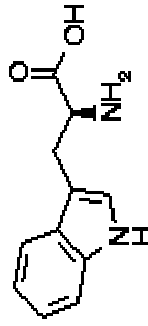
pro p Prolin



ser s Serin



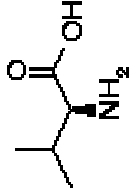
thrt Threonin



trp w Tryptophan



tyr y Tyrosin



val v Valin

# BioPerl

# Bio Perl

- Collection of PERL modules for bioinformatics
  - sequence manipulation
  - accessing web databases
  - parsing of the results
- Open source software
  - Source code available (Contributed by many)
  - Need for open source and content

# Installing Bio-perl

- On-line and off-line installation

- On-line

```
$ perl -MCPAN -e shell
```

```
cpan>install Bundle::BioPerl
```

- Off-line

```
Download files "Bundle-BioPerl-2.1.5.tar.gz"
```

```
tar xvzf Bundle-BioPerl-2.1.5.tar.gz
```

```
perl Makefile.pl
```

```
make
```

```
make test
```

```
make install
```

# What can BioPerl Do?

- **Accessing sequence data from local and remote databases**
- **Transforming formats of database/ file records**
- **Getting information from sequences**
- **Searching for similar sequences**
- **Creating and manipulating sequence alignments**
- **Searching for genes and other structures on genomic DNA**
- **Developing machine readable sequence annotations**

# 1. Accessing Sequence Data

- BioPerl currently supports the following on-line databases
  - genbank – National Centre for Biotech Information
  - RefSeq, - NCBI Reference sequence
  - Swissprot – Swiss Bioinformatics Institute
  - EMBL – European Molecular Biology Lab

# The first program

```
use Bio::Perl;
```

```
# this script will only work if you have an internet connection on the  
# computer you're using, the databases you can get sequences from  
# are 'swiss', 'genbank', 'genpept', 'embl', and 'refseq'
```

```
$seq_object = get_sequence('swiss',"ROA1_HUMAN");
```

```
write_sequence(">roa1.fasta",'fasta',$seq_object);
```

# A Quick Bio::perl

- `get_sequence` - gets a sequence from standard, internet accessible databases
- `read_sequence` - reads a sequence from a file
- `read_all_sequences` - reads all sequences from a file
- `new_sequence` - makes a bioperl sequence just from a string
- `write_sequence` - writes a single or an array of sequence to a file
- `translate` - provides a translation of a sequence
- `translate_as_string` - provides a translation of a sequence, returning back just the sequence as a string
- `blast_sequence` - BLASTs a sequence against standard databases at NCBI
- `write_blast` - writes a blast report out to a file

## 2. Transforming formats

- Changing popular formats
  - FASTA
  - EMBL
  - SwissProt
  - etc

### 3. Information from sequences

- `$seqobj->display_id();`
- `$seqobj->seq();`
- `$seqobj->subseq(5,10);`
- `$seqobj->accession_number();`
- `$seqobj->alphabet();`
- `$seqobj->primary_id();`
- `$seqobj->desc();`

# Translating DNA -> Protein

- Translating from DNA to Protein
- Central Dogma of Molecular Biology
- Maps the DNA to the mRNA
- Each sequence of three Nucleotide maps to one Amino acid.

## 4. Searching for similar Sequences

- Basic Local Alignment Search Tool
- The blast program may be run locally or on supercomputers available on the web. The results are cached hence faster.
- A lot of data is generated
- Programs to interpret the data

# BIO-PERL

- In Brief
  - Perl is a popular programming language written by Larry Wall, with an extensive Regular Expressions
  - BioPerl is a language extension of Perl
  - The Language is extended by including a Module (Library) into the standard package.
  - BioPerl can be used by scientists to manipulate data relating to Molecular Biology.

Thanks