

BioAcknowledgements

- Passionate researchers of the world.
- Scientists who shaped the world as we know by being scientific, rational, honest, and free-thinking.
- Tax payers who support researchers.
- My former and present colleagues in MRC, Harvard Med., EBI, KAIST, and KOBIC from whom I learned so much.
- Korean Perl Conference organizers

BioDisclaimer

- The contents of these slides are produced by Jong after shamelessly stealing other people's copyrighted ideas and knowledge.
- Everyone is welcome to take the slides in part or in entirety without any permission whatsoever.
- Everything synthesized and presented here is under (♥) BioLicense.

(<http://biolicense.org>)

BioPerl: philosophy, history, & future

박종화

(Jong Bhak or Jong Park)

KOBIC

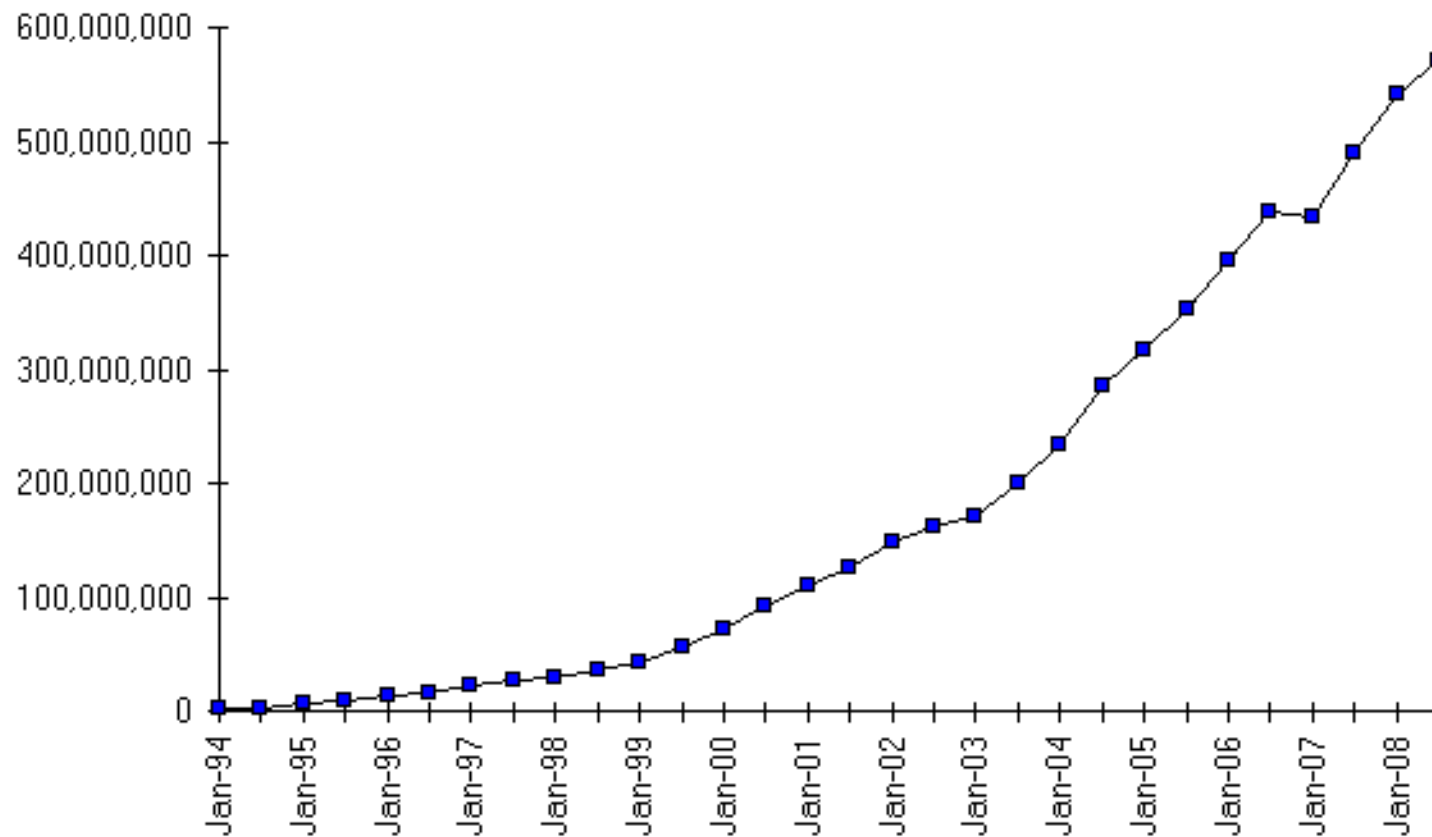
(Korean Bioinformation Center)

Korea

jongbhak@yahoo.com, <http://jongbhak.com>

Internet host count

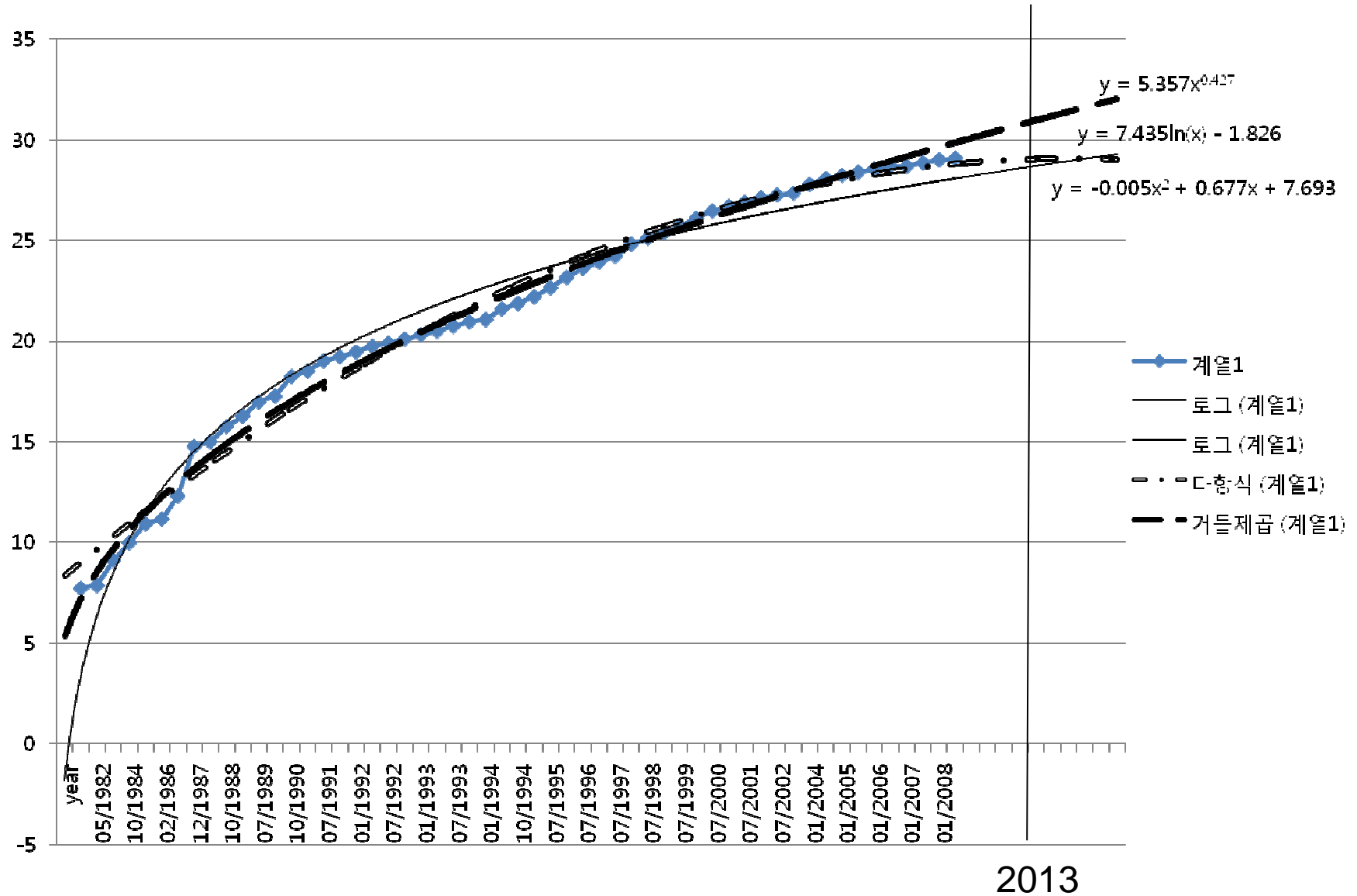
Internet Domain Survey Host Count



Source: Internet Systems Consortium (www.isc.org)

Internet Host Count (1982 ~)

2013: 16 Billion Hosts



Brief History of Perl

- 1987: Larry Wall released version 1.0
 - Unhappy by Sed, C, awk, and Bourne Shell
- The first edition of “Programming Perl”
- 1994: Perl version 5.0, complete re-write
- 1995: First official public release (0.6.2) of the [Apache](#) web server. (April)
- 1995: CPAN officially introduced
- 2009: Perl 6?

**Darwin (evolution)
Mendel (genetic principle)**

DNA modelling (Watson & Crick, 1953)

Hemoglobin/Myoglobin structure (Max Perutz, John Kendrew)

Computational Methodology (Chris Sander & Arther Lesk)

Structure

Sequence

Dynamic programming
Sequence comparison
App module
(Niedleman & Wunsch)

**DNA → codon → anticodon
→ peptide → Protein**

**X-174 Genome (F.sanger)
First full genome sequencing**

Southern blot
Hybridization methology

DB (Gen Bank, PIR, ...)

**DNA chip &
Microarray technology**

Bioinformatics Established

Expression

- Structural genomics
 - Comparative genomics
 - Sequencing
 - Functional genomics (DNA chip)
 - SNP
 - Proteomics (Mass spec. protein chip)
 - DataBases
 - Computational analysis Methodology
- 1995**

Omics World

Computer

INTERNET

Long Definition of Bioinformatics

- Bioinformatics is a discipline of science that analyses, seeks understanding and models the whole life as an information processing phenomenon utilizing energy with methods from philosophy, mathematics and computer science using biological experimental data.

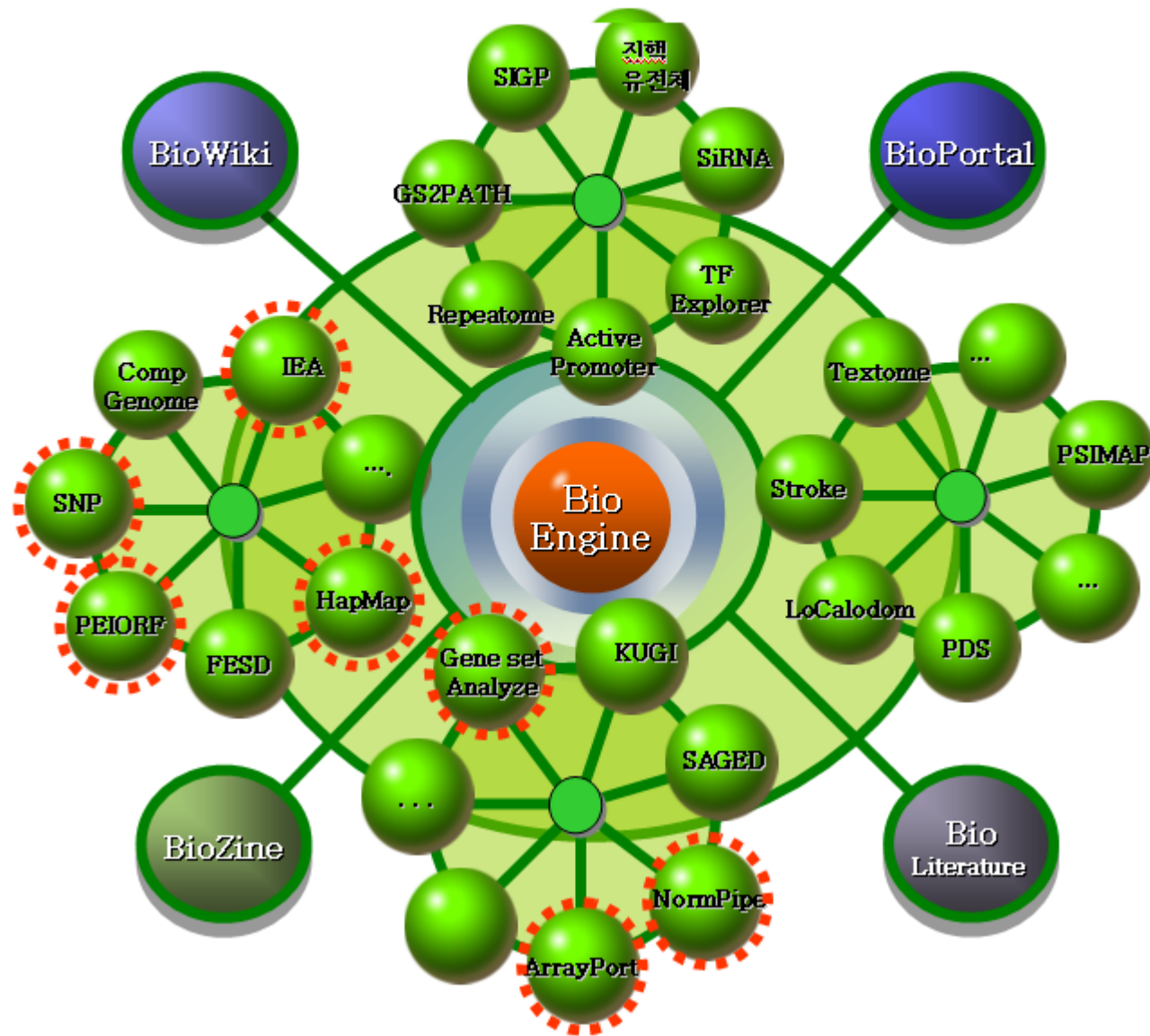
-- Jong Bhak

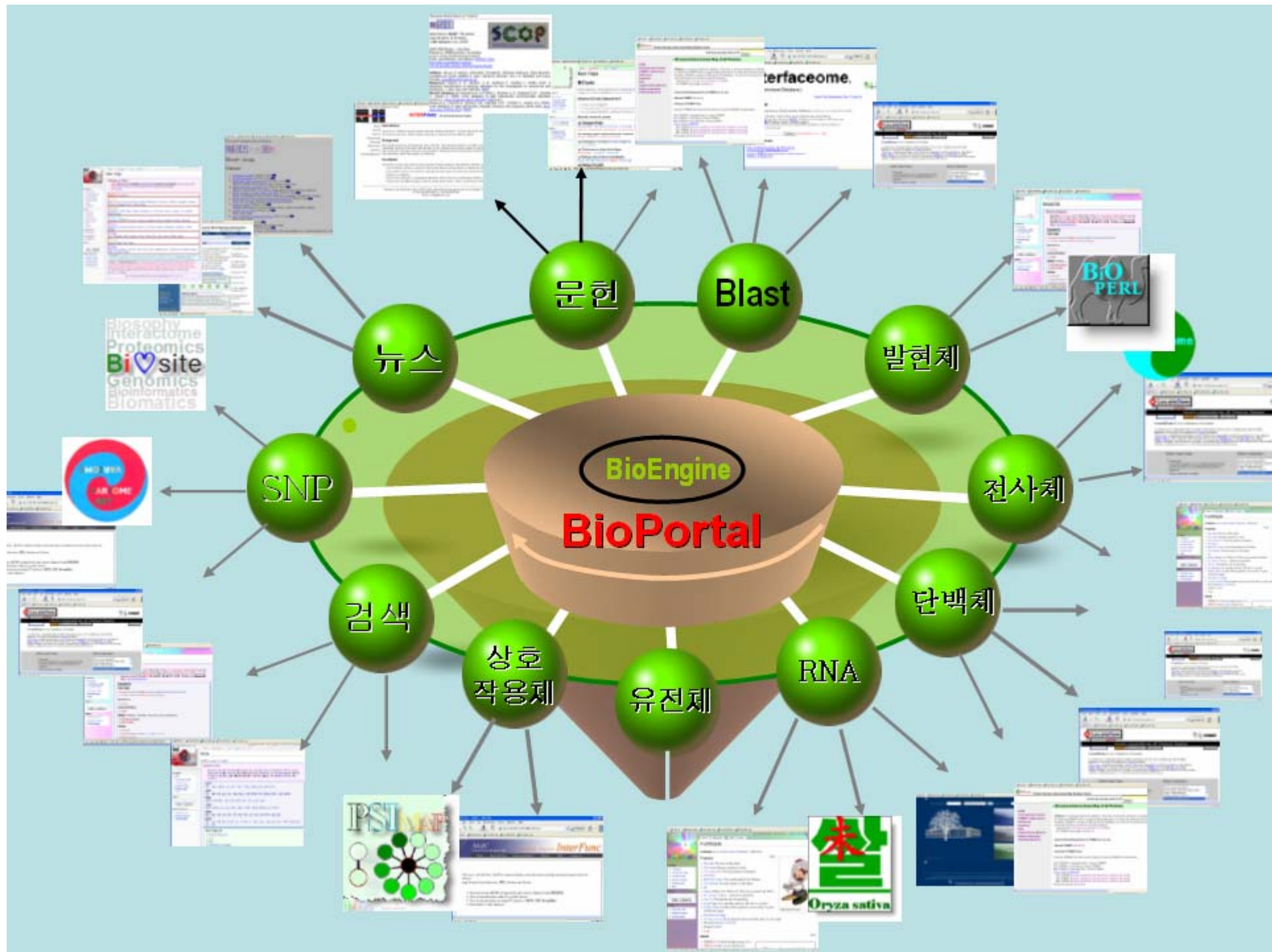
A short definition of Bioinformatics

- Biology is bioinformatics and bioinformatics, biology.

-- Jong Bhak

BioEngine: Automatic BioInformation Pipelines Processing System





Genome sequence (Phi X 174)

- gagttttatc gcttccatga cgcagaagtt
aacactttcg gatatttctg atgagtcgaa
aaattatctt gataaagcag gaattactac
tgcttgttta cgaattaaat cgaagtggac
tgctggcgga aaatgagaaa attcgacctt
tccttgcgca gctcgagaag ctcttacttt
gcgacctttc gccatcaact aacgattctg
tcaaaaactg acgcgttgga tgaggagaag
tggttaata tgcttggcac gttcgtcaag
gactggttta gatatgagtc acattttgtt

..... .

Perl Parsers

- Converting DNA to Protein sequence

```

• #!/usr/bin/perl
• # Made by make_standalone_subroutines.pl at: Sat Sep 27 15:51:22 BST 1997
• #
• # Title   : convert_dna_to_protein
• # Usage   :
• # Function : translate DNA or RNA seq to protein seq.
• # Example :
• # Warning : Copyright (C) 1993-1994 by James Tisdall
• #         : stolen from Tisdall
• # Keywords : dna2protein, dna_2_protein, DNA2protein, translate_dna
• #         : dna2protein, convert_DNA_to_protein, translate_nucleic_acid
• #         : rna2protein, rna_2_protein, RNA2protein, translate_rna
• #         : dna2protein, convert_RNA_to_protein
• # Options :
• # Returns : a ref. of an array for protein translation
• # Argument : a scalar for DNA sequence data
• # Version  : 1.2
• #-----
• sub convert_dna_to_protein{      ##### RnaToProtein
•   my(%seq) = %{$_[0]};
•   my($seq, $i,$len, $j, $PY, %out, $codon, @keys);
•   @keys=keys %seq;
•   $PY = 'W[UTW]';
•   for($i=0; $i < @keys : $i++){
•     my ($output);
•     $seq=$seq{$keys[$i]};
•     $len=length($seq);
•
•     for($j=0; $j < ($len-2) : $j+=3){
•       $codon = substr($seq, $j, 3);
•       if ($codon =~ /^[UTC]/i)      {$output .= 'S'; } # Serine
•       elsif($codon =~ /^[UT][UT][UTC]/i) {$output .= 'F'; } # Phenylalanine
•       elsif($codon =~ /^[UT][UT][AG]/i) {$output .= 'L'; } # Leucine
•       elsif($codon =~ /^[UT]A[UTC]/i)  {$output .= 'Y'; } # Tyrosine
•       elsif($codon =~ /^[UT]A[AG]/i)   {$output .= '_'; } # Stop
•       elsif($codon =~ /^[UT]G[TUC]/i)   {$output .= 'C'; } # Cysteine
•       elsif($codon =~ /^[UT]GA/i)      {$output .= '_'; } # Stop
•       elsif($codon =~ /^[UT]GG/i)      {$output .= 'W'; } # Tryptophan
•       elsif($codon =~ /^C[UT]/i)       {$output .= 'L'; } # Leucine
•       elsif($codon =~ /^CC/i)          {$output .= 'P'; } # Proline
•       elsif($codon =~ /^CA[UTC]/i)     {$output .= 'H'; } # Histidine
•       elsif($codon =~ /^CA[AG]/i)     {$output .= 'Q'; } # Glutamine
•       elsif($codon =~ /^CG/i)          {$output .= 'R'; } # Arginine
•       elsif($codon =~ /^A[UT][UTCA]/i) {$output .= 'I'; } # Isoleucine
•       elsif($codon =~ /^A[UT]G/i)     {$output .= 'M'; } # Methionine
•       elsif($codon =~ /^AC/i)         {$output .= 'T'; } # Threonine
•       elsif($codon =~ /^AA[TUC]/i)    {$output .= 'N'; } # Asparagine
•       elsif($codon =~ /^AA[AG]/i)     {$output .= 'K'; } # Lysine
•       elsif($codon =~ /^AG[TUC]/i)    {$output .= 'S'; } # Serine
•       elsif($codon =~ /^AG[AG]/i)     {$output .= 'R'; } # Arginine
•       elsif($codon =~ /^G[UT]/i)      {$output .= 'V'; } # Valine
•       elsif($codon =~ /^GC/i)         {$output .= 'A'; } # Alanine
•       elsif($codon =~ /^GA[TUC]/i)    {$output .= 'D'; } # Aspartic Acid
•       elsif($codon =~ /^GA[AG]/i)     {$output .= 'E'; } # Glutamic Acid
•       elsif($codon =~ /^GG/i)        {$output .= 'G'; } # Glycine
•       else {print "Wn# convert_dna_to_protein: unrecognized codon $codon Wn";}
•     }
•   }
• }

```

History of Bioperl

- 1994: Tim Hubbard: th_lib.pl
- 1994: Jong Bhak: jong_lib.pl, Bio.pl
- 1995: Jong and Steve Brenner: Bioperl
- 1995: Bio.pm
- 1996: Georg Fuellen and bioinformatics course in Bielefeld Germany
- 1997->: Steve Chervitz, Ewan Birney
- 2000~ a world wide project

Bioperl: 1995

- BioPerl: is to enhance the adoption of an easy and efficient programming language for Biology community.

Perl is a very easy but powerful programming language which can be used for Biology. It has built in functions for various types of text handling which can be used for sequence handling. As there is no easy and wide spread common language in Biological computing field, it would be very nice to develop such a language to enhance the exchange of programs between biologists. Perl has been already used for many genome research works and it is extremely easy to learn and fun to use. Rather than wasting time and resource to develop independent programs which do basically same things, why do we borrow other people's perl codes and give them ours. Perl codes can be easily copied and modified but if we set up a standard way of programming (which is not strict or needed to be followed) with basic guideline it will even enhance the exchange of completely developed codes.

Perl is the largest and very extendable general purpose language comprising of numerous 'modules'. **By registering bioperl as a module, we can extend perl language to be used in Biology community with great efficiency.**

- –Cambridge, 1995.

Bioperl and founders

- Original Bioperl did not have the concept of founders.
- Everyone from past, present, and future are perpetual founders.
 - In other words, we are still waiting for the founders/synthesizers of Bioperl
- BioOriginality: <http://biosophy.org/index.php/BioOriginality>

Philosophy of Bioperl

- ...Bioperl's authors not only the people who are in the past, present but also the future. There is no one who created 'vision' :-) or anything like that. Everybody who agrees and like the idea of Perl and more open attitude in exchanging idea has the real 'vision'. ...

http://backpan.cpan.org/authors/

	SAT_v2.0.tar	16-Jun-1999 01:40	24.6M	tar archive
	Th_lib.1.0.pm.gz	05-Dec-1995 00:08	90k	GZIP compressed docume>
	bio_lib.1.6.readme	03-Oct-1997 10:17	3k	
	bio_lib.1.6.tar.gz	21-Sep-1997 13:45	226k	GZIP compressed docume>
	bio_lib.1.7.readme	06-Oct-1997 15:56	3k	
	bio_lib.1.7.tar.gz	06-Oct-1997 15:59	239k	GZIP compressed docume>
	bio_lib.1.8.readme	10-Jul-1998 16:19	3k	
	bio_lib.1.8.tar.gz	10-Jul-1998 16:36	389k	GZIP compressed docume>
	bio_lib.pl.1.3.tar.gz	16-Oct-1996 20:48	153k	GZIP compressed docume>
	bio_lib.pl.v.1.2.readme	19-Mar-1996 22:57	1k	
	bio_lib.pl.v.1.2.tar.gz	19-Mar-1996 23:16	128k	GZIP compressed docume>
	divclus.2.1.readme	21-Sep-1997 13:54	2k	
	divclus.2.1.tar.gz	21-Sep-1997 13:57	25k	GZIP compressed docume>
	divclus_2.2.readme	06-Oct-1997 15:47	3k	
	divclus_2.2.tar.gz	06-Oct-1997 15:51	264k	GZIP compressed docume>
	evalign.1.1.readme	21-Sep-1997 14:20	3k	

Georg Fuellen's Bioperl page

Welcome to the Bioperl Project !

What's New (December 9, 1998):

New release of the Bioperl modules --

Please visit our new [homepage](#).

This site is now relevant only as the distribution site for any upcoming standalone versions of the UnivAln.pm module, and for people interested in history.

What's New (August 18, 1998): Our new Homepage, [bio.perl.org](#).

- [Introduction](#)
- [Current core modules](#)
 - [Bio::PreSeq](#)
 - [Bio::UnivAln](#)
 - [Bio::Object, etc](#)
 - [Blast modules \(currently collecting pointers!\)](#)
 - [Bio::Struct \(conceptual\)](#)
- [Mailing lists](#)
 - [Announcement List](#)
 - [Discussion List](#)
 - [Technical List](#)
- [Related sites](#)
- [Contributors and sponsors](#)

What's New (May 7, 1996): We present a poster at ISMB'98, [see the mailing list archive](#).

What's New (Dec 4, 1997): Started collecting [Links to Blast modules](#).

What's New (Nov 7, 1997): A [response](#) to a Request For Information of the *Object Management Group*. -re- Bio-objects/CORBA.

Perl Conference Aug 19-21: A [talk by Steve Brenner on Bioperl](#).

What's New (July 2, 1997): We've presented a [Poster](#) at the [Objects in Bioinformatics, June 1997](#) EBI Meeting.

Introduction

We develop free software, in particular for handling biosequence and alignment data. Perl is an [excellent](#) "Glue Language" for interfacing C/Fortran code, WWW/CGI, graphics, numerical analysis and much more. With the current development of a true Perl Compiler, it is a serious competitor to C: it's got high-level features, and relieves you from manual memory management, segmentation faults, bus errors, most portability problems, etc, etc.

Current modules

N.B. This is beta-code; in particular the interface of the Bio::PreSeq object may be revised until the first non-beta release is made. It will then be named "Bio::Seq".

Bio::PreSeq, a Perl module for reading, accessing, manipulating, and analyzing single biological sequences.


- **Status: beta code.** Version 0.035 from 26 MAY 1997.
- [gzipped tar file](#), [Docu](#), [Directory](#).
- Goal: Provide a standard representation for biological sequences, so that the same sequence may be submitted to a large number of different manipulation and analysis functions.
- Planned Enhancements: Increased consistency, error-checking, and more analysis functions. Bio::PreSeq is a tentative version of Bio::Seq, and will be renamed once the interfaces are fully fixed. Users may build code on Bio::PreSeq with the knowledge that its interface will remain reasonably constant, as major changes will be reserved for Bio::Seq.
- Chief Developer ("Pumpkin Holder"): Chris Dagdigan, cdagdigan@genetics.com.
- [Developer's site](#) (for the newest stuff)

May 25, 1998: Bio::UnivAln Version 1.009 is out -- PLEASE, DO MAIL ME (georg.fuellen@uni-bielefeld.de) IF YOU HAVE PROBLEMS !

Bio::UnivAln, a Perl module for reading, parsing, writing, slicing, and manipulating multiple biosequences (sequence multisets and alignments).

- **Status: beta code.** Version 1.009 from 25 May 98.
- [gzipped tar file](#), [tar file](#), [Docu](#), [Code](#), [Tests \(univaln.t\)](#) and Test output ([univaln.o](#)). (Old version: [UnivAln1.006.tar](#)).
- Goal: Manage and manipulate multiple sequences and their alignments.
- Very flexible slicing, mapping, consensus, (n)variable sites, gap-free sites, reverse, complement, and reverse complement.
- Of special interest: You can specify functions for the slicing operation, e.g. use a function that returns true if column characters are "different" according to some definition of yours, and slice all the other columns off... Voila - you have the variable sites only !
- Also, you can apply a function to some columns and/or rows, and receive the list of function values. E.g. use a function that returns the majority residue based on your own criteria, apply it to the columns, and there you go: Your consensus sequence !
- Planned enhancements: More CGI and Graphics: improved format handling: better validity checking.
- Chief Developer ("Pumpkin Holder"): Georg Fuellen, fuellen@Techfak.Uni-Bielefeld.DE, fuellen@dali.mathematik.uni-bielefeld.de.
- [Developer's site](#) (for the newest stuff)

Bioperl.org



page discussion view source history

Main Page

Introduction

This is the [BioPerl](#) project's community documentation site. You can read up on [Getting BioPerl](#), [Installing BioPerl](#), and [Getting Started](#). Also use the [Frequently Asked Questions](#), [HOWTOs](#), and the [BioPerl Tutorials](#) as starting place for learning about the toolkit's compo

More specific information for developers can be found [here](#) and on the sidebar.

There is a short [History of BioPerl](#) with background on the project and [Lincoln Stein's](#) article on [How Perl Saved the Human Genome I](#)

See what [BioPerl Users](#) are using the toolkit for and what [publications](#) cite the toolkit.

The toolkit is divided into several packages, most people will only want to deal with the [Core package](#).

- [Core package](#) provides the main parsers, this is the basic package and it's required by all the other packages (bioperl-live [SVN](#) dir)
- [Run package](#) provides [wrappers](#) for executing some 60 common [bioinformatics](#) applications (bioperl-run in [SVN](#))
- [BioPerl db package](#) is a subproject to store sequence and annotation data in a [BioSQL relational database](#) (bioperl-db in [SVN](#)).
- [Network package](#) parses and analyzes protein-protein interaction data (bioperl-network in [SVN](#)).
- See all the available packages via [Downloads page](#)

API documentation for each module at [doc.bioperl.org](#) and each module page on this site is linked to the doc site and the [CPAN](#) modules, and [Ext modules](#) available on this site and the external [API documentation](#).

main links

- [Main Page](#)
- [Getting Started](#)
- [Downloads](#)
- [Installation](#)
- [Recent changes](#)
- [Random page](#)

documentation

- [Quick Start](#)
- [FAQ](#)
- [HOWTOs](#)
- [BioPerl Tutorial](#)
- [Tutorials](#)
- [Deobfuscator](#)
- [Browse Modules](#)

Oo Bioperl usage

```
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);
```

Running Blast

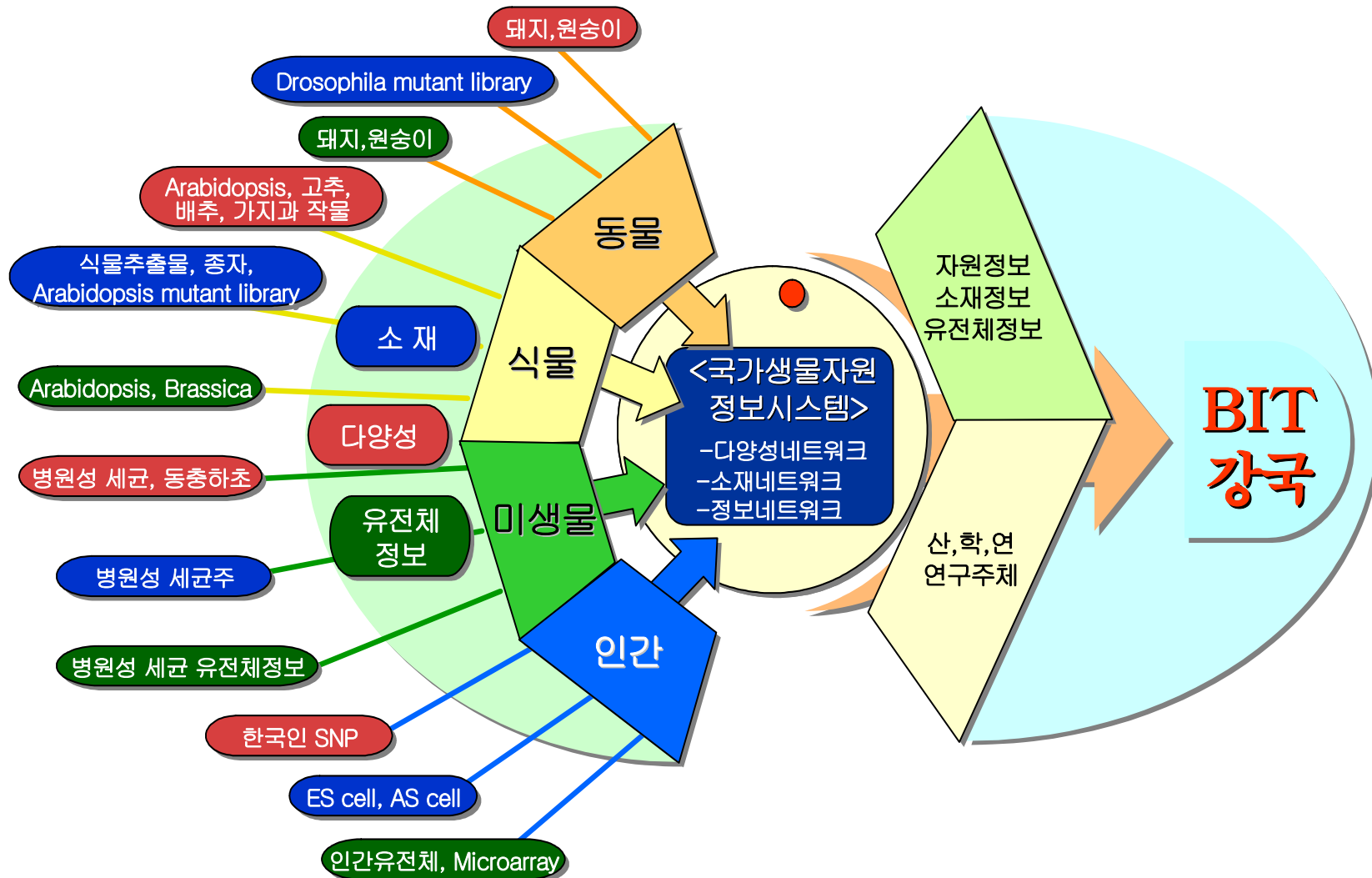
```
use Bio::Perl;  
  
$seq = get_sequence('swiss', "ROA1_HUMAN");  
  
# uses the default database - nr in this case  
$blast_result = blast_sequence($seq);  
  
write_blast(">roa1.blast", $blast_result);
```

Bioperl Future

- Developing standalone compiler for Biology utilizing present Bioperl modules, subroutines, algorithms, and objects
- Construction of Bioperl shell, Bioperl OS
- Commercializing Bioperl

추진 체계

범 자원적 연계시스템 구축



추진 전략

- 국내 생물정보 BioMatrix 연구/개발/분석 시스템 구축
 - 생명공학 주요 연구주제별 매트릭스 정보 시스템 구축 및 지원

