

INTRODUCTION

CANB7640

Aik Choon Tan, Ph.D.

Associate Professor of Bioinformatics

Division of Medical Oncology

Department of Medicine

aikchoon.tan@ucdenver.edu

9/4/2018

<http://tanlab.ucdenver.edu/labHomePage/teaching/CANB7640/>

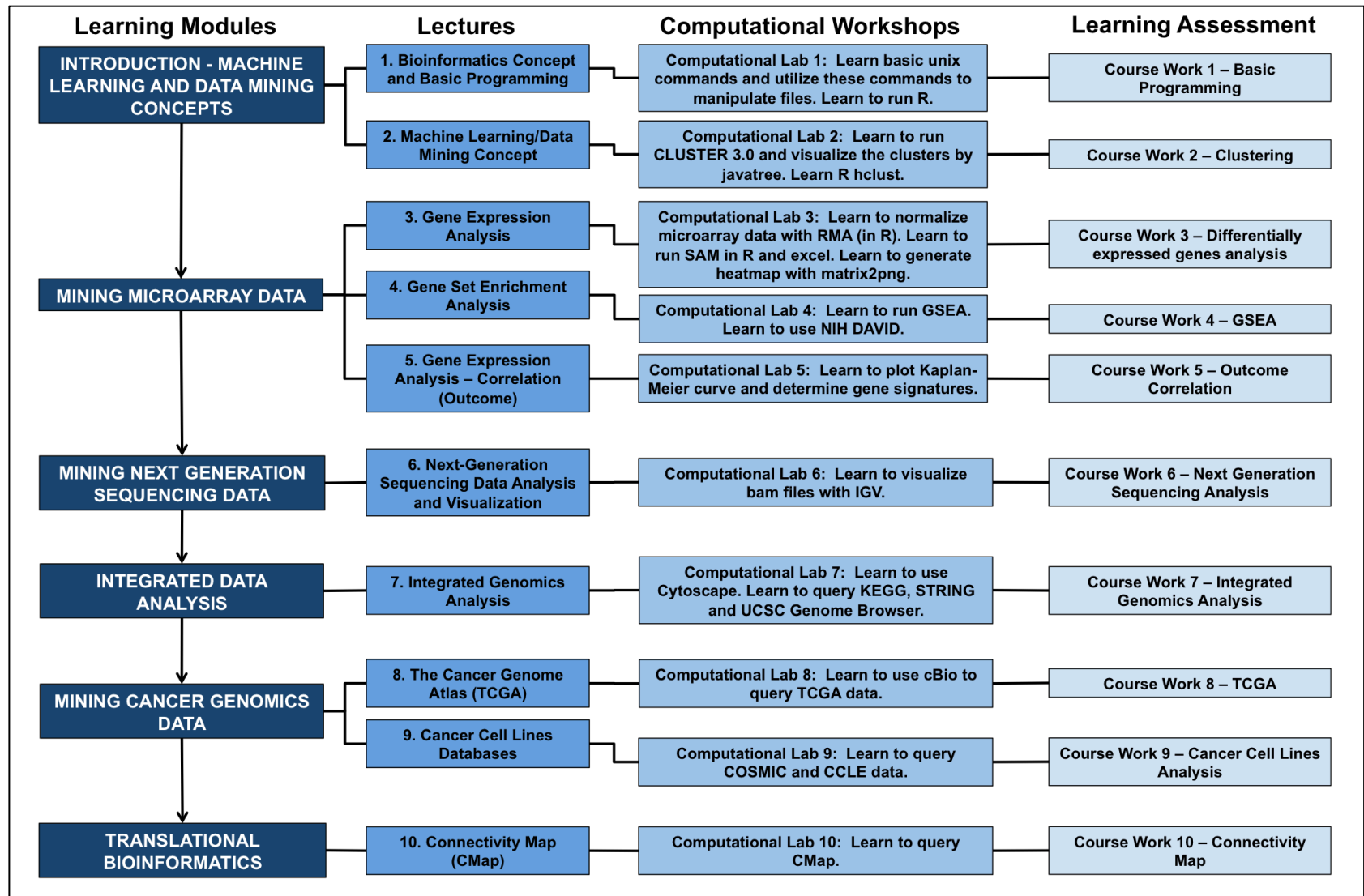
Faculty/Instructors

- Aik Choon Tan, Ph.D. (aikchoon.tan @ucdenver.edu)
- Jihye Kim, Ph.D. (jihye.kim @ucdenver.edu)

Class Format

- ~1 hour lecture on background of the tools / methods (1pm – 2pm)
- ~5 mins break
- ~ 2 hours of computational workshop (2:05pm – 4pm)
- Use your own laptop
 - Windows – install Cygwin (<https://www.cygwin.com/>) and install Perl and X11 packages
 - Mac – need to find your terminal
- Class materials will be available at:
<http://tanlab.ucdenver.edu/labHomePage/teaching/CANB7640/>

CANB 7640 Syllabus



Grading

| Item | Points |
|---|--------|
| Course work 1 – Basic Programming | 5 |
| Course work 2 – Clustering | 5 |
| Course work 3 – Differentially Expressed Genes Analysis | 5 |
| Course work 4 – GSEA | 5 |
| Course work 5 – Outcome Correlation | 5 |
| Course work 6 – Next Generation Sequencing | 5 |
| Course work 7 – Integrated Genomics Analysis | 5 |
| Course work 8 – TCGA | 5 |
| Course work 9 – Cancer Cell Lines Analysis | 5 |
| Course work 10 – Connectivity Map | 5 |
| Attendance and Participation | 10 |
| Final Project | 30 |
| Final Presentation | 10 |
| Total | 100 |

Course Work / Assignment

- 2-3 pages word documents
- If programming works, need pseudocode in word and the program (executable)
- Put everything in a folder and compressed
- Send to aikchoon.tan @ ucdenver.edu by next Monday 5pm

Final Project

The final project is based on a real problem and the students have to use the skills and knowledge that they have learned in the course to solve the problem, e.g. “*Identify the resistant pathways for metformin in triple negative breast cancer*”. Students have to submit a written report (listing out the tools, datasets that they used, and rationale for using these tools/datasets) and a presentation to describe their analysis and interpretation to the problem.

Before next class, set up a meeting with me and we can discuss about your final project.

CANB 7640 Learning Objectives

- Understand the basic concepts of current bioinformatics in mining microarray and next generation sequencing data.
- Apply relevant bioinformatics for the analysis of microarray and next generation sequencing data in research projects.
- Locate, download and correlate genes of interest with published data sets.
- Generate and visualize processed genomics data.

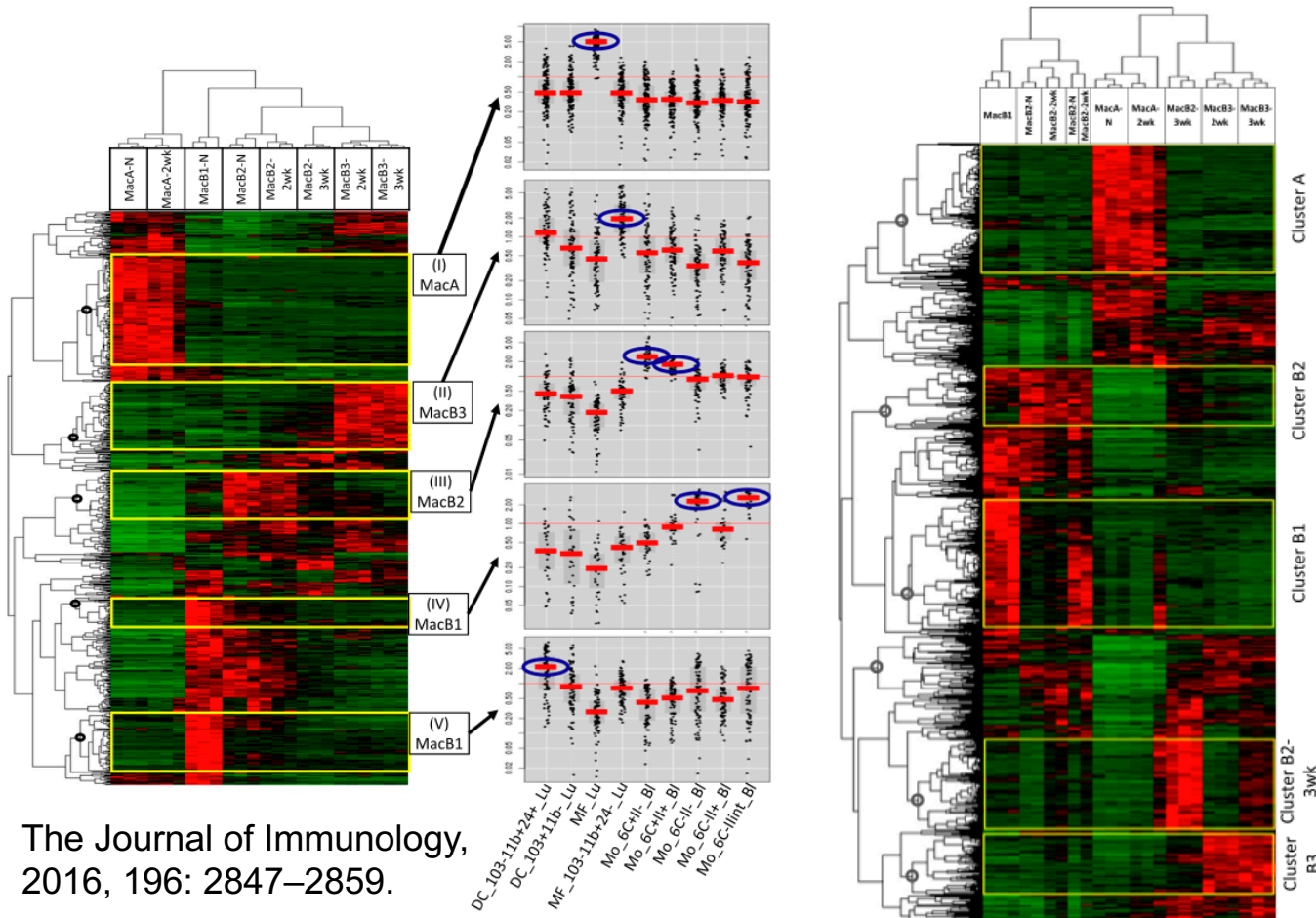
Bonus goals ...

- Not afraid to read a bioinformatics paper and understand the concept / algorithm.
- Be able to download, install and execute program.
- Be able to download published data sets from public repositories and perform analysis using bioinformatics tools.
- Maybe, come out with new ideas and implement new bioinformatics tools 😊

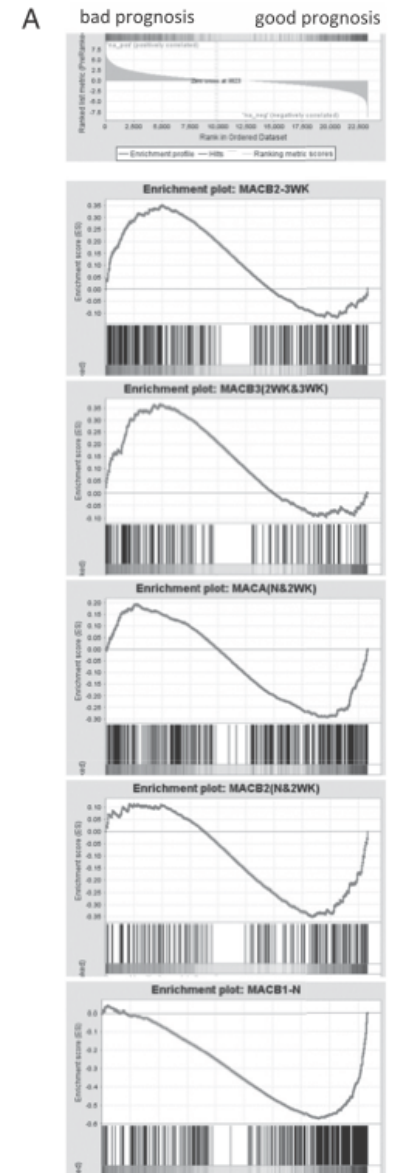
Alumni (No pressure ...)

Expression Profiling of Macrophages Reveals Multiple Populations with Distinct Biological Roles in an Immunocompetent Orthotopic Model of Lung Cancer

Joanna M. Poczobutt,* Subhajyoti De,* Vinod K. Yadav,* Teresa T. Nguyen,*
Howard Li,*[†] Trisha R. Sippel,* Mary C. M. Weiser-Evans,*[‡] and Raphael A. Nemenoff*[‡]



The Journal of Immunology,
2016, 196: 2847–2859.



Alumni (No pressure ...)

www.impactjournals.com/oncotarget/

Oncotarget, Vol. 7, No. 32

Research Paper

Integrating heterogeneous drug sensitivity data from cancer pharmacogenomic studies

Nikita Pozdeyev¹, Minjae Yoo¹, Ryan Mackie¹, Rebecca E. Schweppe¹, Aik Choon Tan^{1,*}, Bryan R. Haugen^{1,*}

¹Department of Medicine, University of Colorado Cancer Center, University of Colorado School of Medicine, Aurora, CO

*Two senior authors contributed equally to this work

Correspondence to: Nikita Pozdeyev, email: nikita.pozdeyev@ucdenver.edu

Keywords: database integration, drug sensitivity, pharmacogenomics, cancer, cell line

Received: April 08, 2016

Accepted: May 29, 2016

Published: June 14, 2016

Quantitative Analysis of Pharmacogenomics in Cancer

Choose Drug
Erlotinib

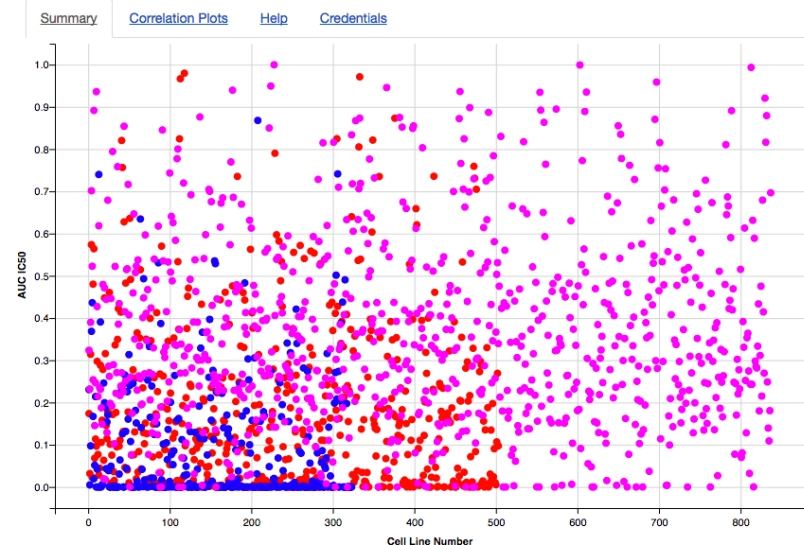
Choose Metric
AUC IC50

Select Database
☒ CCLE
☒ GDSC
☒ CTRP

Data Reconciliation
☐ Reconcile Data

Plot Controls
☐ Sort Data

[Download](#)



<http://tanlab.ucdenver.edu/QAPC/>

Alumni (No pressure ...)



Database, 2016, 1–13
doi: 10.1093/database/baw043
Original article



Database, Vol. 2016, Article ID baw043

Original article

BRONCO: Biomedical entity Relation ONcology COrpus for extracting gene-variant-disease-drug relations

Kyubum Lee¹, Sunwon Lee¹, Sungjoon Park¹, Sunkyu Kim¹,
Suhkyung Kim¹, Kwanghun Choi¹, Aik Choon Tan^{2,*} and
Jaewoo Kang^{1,*}

¹Department of Computer Science and Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul, 02841 Korea and ²Translational Bioinformatics and Cancer Systems Biology Laboratory, Division of Medical Oncology, Department of Medicine, University of Colorado Anschutz Medical Campus, 12801 East 17th Avenue Aurora, CO 80045, USA

*Corresponding author: Email: kangj@korea.ac.kr, Correspondence may also be addressed to Aik Choon Tan.
Email: aikchoon.tan@ucdenver.edu

Citation details: Lee,K., Lee,S., Park,S. *et al.* BRONCO: Biomedical entity Relation ONcology COrpus for extracting gene-variant-disease-drug relations. *Database* (2016) Vol. 2016: article ID baw043; doi:10.1093/database/baw043

Received 3 October 2015; Revised 9 March 2016; Accepted 9 March 2016

Abstract

Comprehensive knowledge of genomic variants in a biological context is key for precision medicine. As next-generation sequencing technologies improve, the amount of literature containing genomic variant data, such as new functions or related phenotypes, rapidly increases. Because numerous articles are published every day, it is almost impossible to manually curate all the variant information from the literature. Many researchers focus on creating an improved automated biomedical natural language processing (BioNLP) method that extracts useful variants and their functional information from the literature. However, there is no gold-standard data set that contains texts annotated with variants and their related functions. To overcome these limitations, we introduce a Biomedical entity Relation ONcology COrpus (BRONCO) that contains more than 400 variants and their relations with genes, diseases, drugs and cell lines in the context of cancer and anti-tumor drug screening research. The variants and their relations were manually extracted from 108 full-text articles. BRONCO can be utilized to evaluate and train new methods used for extracting biomedical entity relations from full-text publications, and thus be a valuable resource to the biomedical text mining research community. Using BRONCO, we quantitatively and qualitatively evaluated the performance of three state-of-the-art BioNLP methods. We also identified their shortcomings, and suggested remedies for each method. We implemented post-processing modules for the three BioNLP methods, which improved their performance.

Database URL: <http://infos.korea.ac.kr/bronco>

Acknowledgements

We thank Susan Kim for suggestions and editing of the manuscript. We would also like to thank University of Colorado Anschutz Medical graduate students/fellows of the CANB7640 class and Heewon Lee of Korea University—Division of Biotechnology for assisting in the curation task for BRONCO.

Thank you for helping out!

Why do you want to learn Bioinformatics?

- Lots of lots of data
 - Heterogeneous (expression, structure, sequence, phenotype, clinical etc)
 - Big data (high dimensional and high data storage)
- Not possible to go through in excel (manually)
- Need faster algorithm(s) to find patterns
- Correlation with published data (other studies)

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|--------------|
| GGCACGAGCC | ACCGTCCAGG | GAGCAGGTAG | CTGCTGGGCT | CCGGGGACAC | TTTGCGTTTCG |
| GGCTGGGAGC | GTGCTTTTCCA | CGACGGTGAC | ACGCTTCCCT | GGATTGGCAG | CCAGACTGCC |
| TTCCGGGTCA | CTGCCATGGA | GGAGCCGCAG | TCAGATCCTA | GCGTCGAGCC | CCCTCTGAGT |
| CAGGAAACAT | TTTCAGACCT | ATGGAAACTA | CTTCCTGAAA | ACAACGTTCT | GTCCCCCTTG |
| CCGTCCCAAG | CAATGGATGA | TTTGATGCTG | TCCCCGGACG | ATATTGAACA | ATGGTTTCACT |
| GAAGACCCAG | GTCCAGATGA | AGCTCCAGA | ATGCCAGAGG | CTGCTCCCCG | CGTGGCCCCCT |
| GCACCAGCAG | CTCCTACACC | GGCGGCCCCCT | GCACCAGCCC | CCTCCTGGCC | CCTGTCTATCT |
| TCTGTCCCTT | CCCAGAAAAC | CTACCAGGGC | AGCTACGGTT | TCCGTCTGGG | CTTCTTTGCAT |
| TCTGGGACAG | CCAAGTCTGT | GACTTGACAG | TACTCCCCTG | CCCTCAACAA | GATGTTTTTGC |
| CAACTGGCCA | AGACCTGCCC | TGTGCAGCTG | TGGGTTGATT | CCACACCCCC | GCCCGGCACC |
| CGCGTCCGCG | CCATGGCCAT | CTACAAGCAG | TCACAGCACA | TGACGGAGGT | TGTGAGGCGC |
| TGCCCCCACC | ATGAGCGCTG | CTCAGATAGC | GATGGTCTGG | CCCCTCCTCA | GCATCTTATC |
| CGAGTGGAAG | GAAATTTGCG | TGTGGAGTAT | TTGGATGACA | GAAACACTTT | TCGACATAGT |
| GTGGTGGTGC | CCTATGAGCC | GCCTGAGGTT | GGCTCTGACT | GTACCACCAT | CCACTACAAC |
| TACATGTGTA | ACAGTTCCTG | CATGGGCGGC | ATGAACCGGA | GGCCCATCCT | CACCATCATC |
| ACACTGGAAG | ACTCCAGTGG | TAATCTACTG | GGACGGAACA | GCTTTGAGGT | GCGTGTTTTGT |
| GCCTGTCTCTG | GGAGAGACCG | GCGCACAGAG | GAAGAGAATC | TCCGCAAGAA | AGGGGAGCCT |
| CACCACGAGC | TGCCCCCAGG | GAGCACTAAG | CGAGCACTGC | CCAACAACAC | CAGCTCCTCT |
| CCCCAGCCAA | AGAAGAAACC | ACTGGATGGA | GAATATTTCA | CCCTTCAGAT | CCGTGGGCGT |
| GAGCGCTTCG | AGATGTTCCG | AGAGCTGAAT | GAGGCCTTGG | AACTCAAGGA | TGCCCAGGCT |
| GGGAAGGAGC | CAGGGGGGAG | CAGGGCTCAC | TCCAGCCACC | TGAAGTCCAA | AAAGGGTCAG |
| TCTACCTCCC | GCCATAAAAA | ACTCATGTTC | AAGACAGAAG | GGCCTGACTC | AGACTGACAT |
| TCTCCACTTC | TTGTTCCCCA | CTGACAGCCT | CCCACCCCCA | TCTCTCCCTC | CCCTGCCATT |
| TTGGGTTTTTG | GGTCTTTTGAA | CCCTTGCTTG | CAATAGGTGT | GCGTCAGAAG | CACCCAGGAC |
| TTCCATTTTGC | TTTGTCCCGG | GGCTCCACTG | AACAAGTTGG | CCTGCACTGG | TGTTTTTGTG |
| TGGGGAGGAG | GATGGGGAGT | AGGACATACC | AGCTTAGATT | TTAAGGTTTT | TACTGTGAGG |
| GATGTTTGGG | AGATGTAAGA | AATGTTCTTG | CAGTTAAGGG | TTAGTTTACA | ATCAGCCACA |
| TTCTAGGTAG | GGGCCCCACTT | CACCGTACTA | ACCAGGGAAG | CTGTCCCTCA | CTGTTGAATT |
| TTCTCTAACT | TCAAGGCCCA | TATCTGTGAA | ATGCTGGCAT | TTGCACCTAC | CTCACAGAGT |
| GCATTGTGAG | GGTTAATGAA | ATAATGTACA | TCTGGCCTTG | AAACCACCTT | TTATTACATG |
| GGGTCTAGAA | CTTGACCCCC | TTGAGGGTGC | TTGTTCCCTC | TCCCTGTTGG | TCGGTGGGTT |
| GGTAGTTTCT | ACAGTTGGGC | AGCTGGTTAG | GTAGAGGGAG | TTGTCAAGTC | TCTGCTGGCC |
| CAGCCAAACC | CTGTCTGACC | ACCTCTTGGT | GAACCTTAGT | ACCTAAAAGG | AAATCTCACC |
| CCATCCCACA | CCCTGGAGGA | TTTCATCTCT | TGTATATGAT | GATCTGGATC | CACCAAGACT |
| TGTTTTTATGC | TCAGGGTCAA | TTTCTTTTTTT | CTTTTTTTTTT | TTTTTTTTTCT | TTTTCTTTGA |
| GACTGGGTCT | CGCTTTGTG | CCCAGGCTGG | AGTGGAGTGG | CGTGATCTTG | GCTTACTGCA |
| GCCTTTGCCT | CCCCGGCTCG | AGCAGTCCTG | CCTCAGCCTC | CGGAGTAGCT | GGGACCACAG |
| GTTTCATGCCA | CCATGGCCAG | CCAACTTTTG | CATGTTTTTGT | AGAGATGGGG | TCTCACAGTG |
| TTGCCCAGGC | TGGTCTCAAA | CTCCTGGGCT | CAGGCGATCC | ACCTGTCTCA | GCCTCCCAGA |
| GTGCTGGGAT | TACAATTGTG | AGCCACCACG | TCCAGCTGGA | AGGGTCAACA | TCTTTTACAT |
| TCTGCAAGCA | CATCTGCATT | TTCACCCCAC | CCTTCCCCTC | CTTCTCCCTT | TTTATATCCC |
| ATTTTTTATAT | CGATCTCTTA | TTTTTACAATA | AAACTTTGCT | GCCAAAAAAA | AAAAAAAAAAAA |

MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDL

LSPDDIEQWFTEDPGPDEAPRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKT

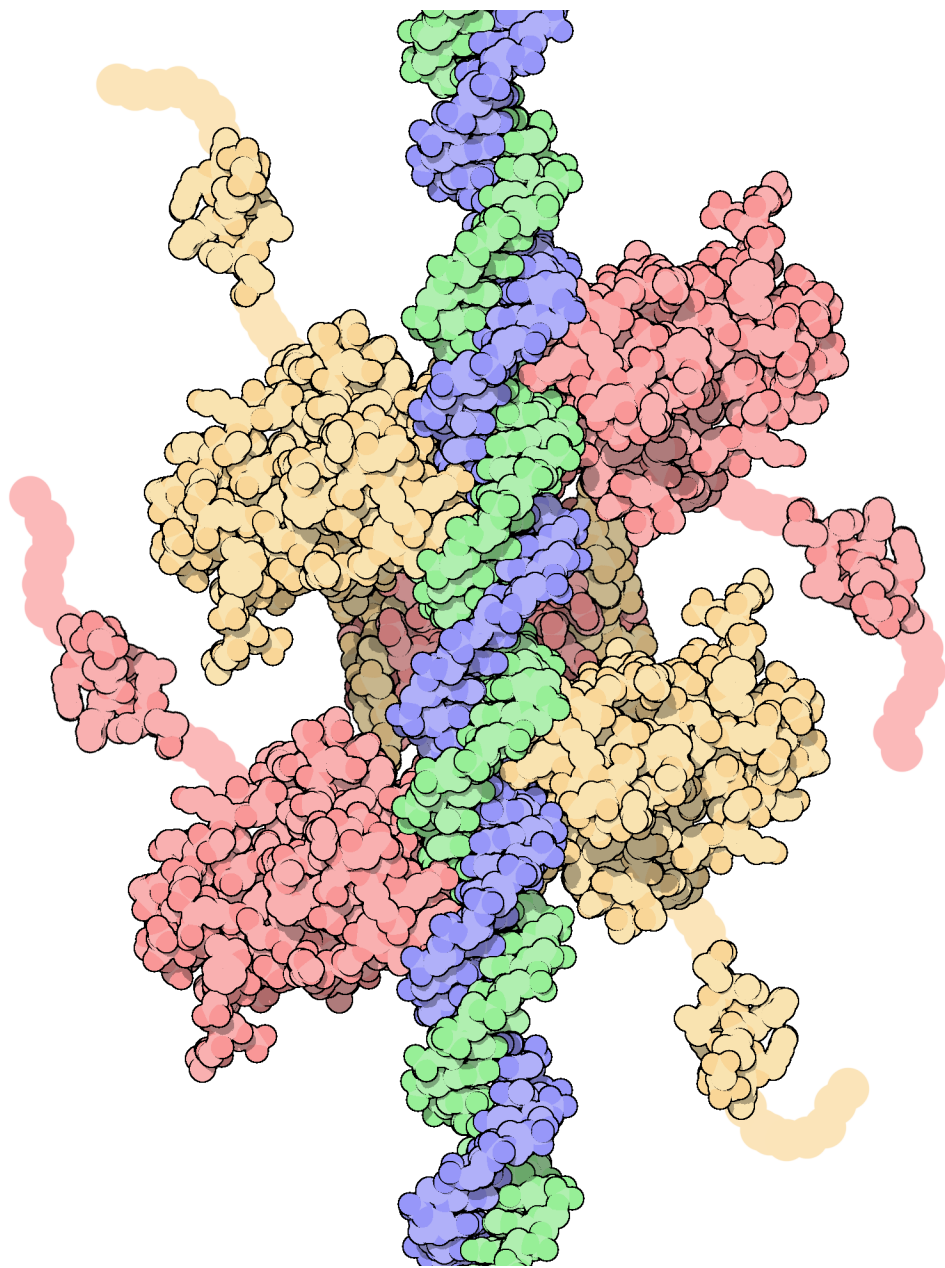
YQGSYGFRLGFLHSGTAKSVTCTYSPALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAM

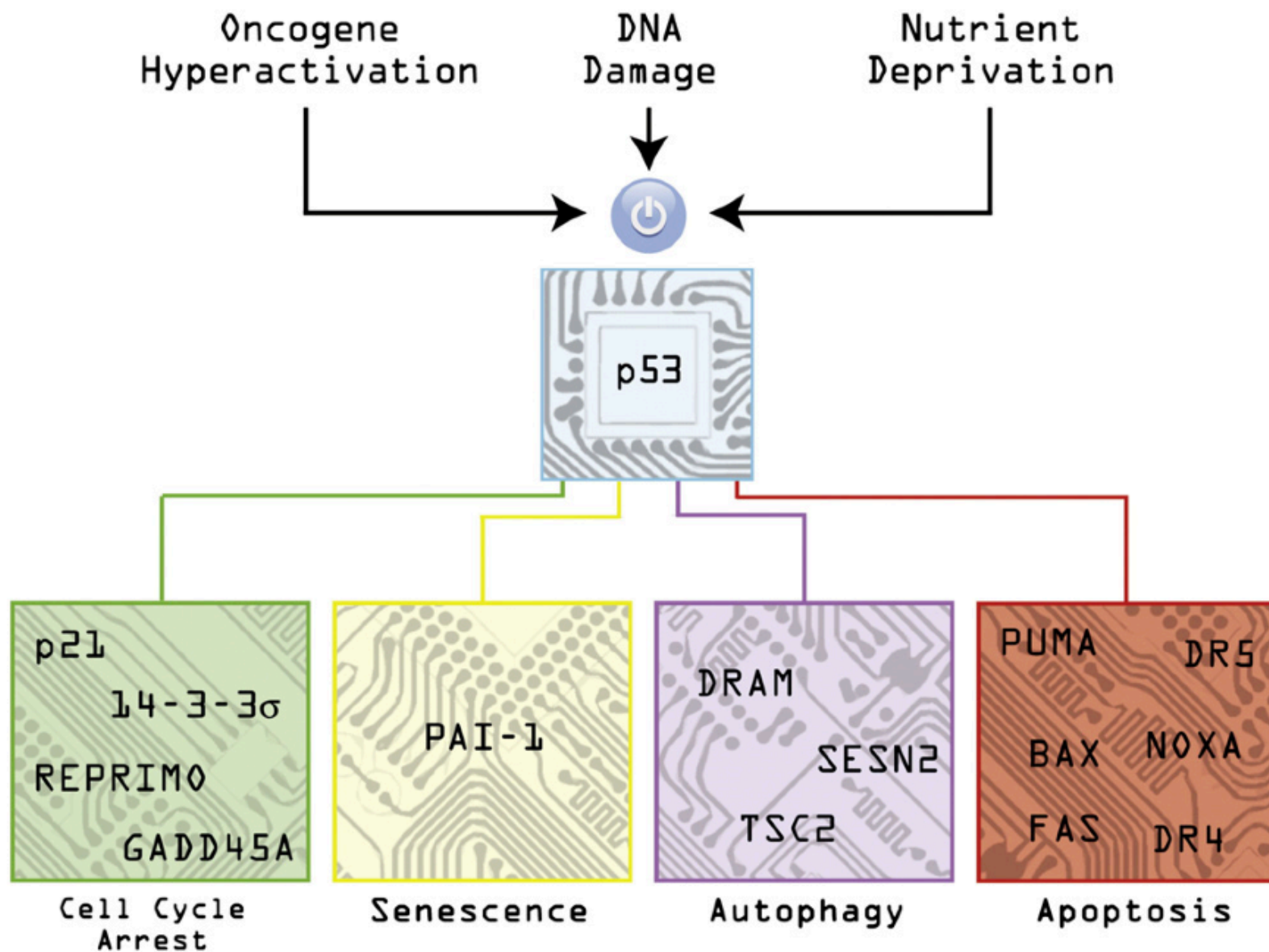
AIYKQSQHMTEVVRRCPPHHERCSDSDGLAPPQHLIRVEGNLRVEYLDDRNTFRHSVVV

PYEPPEVGSDCTTIHYNMCMNSSCMGMNRRPILTIITLEDSSGNLLGRNSFEVRVCA

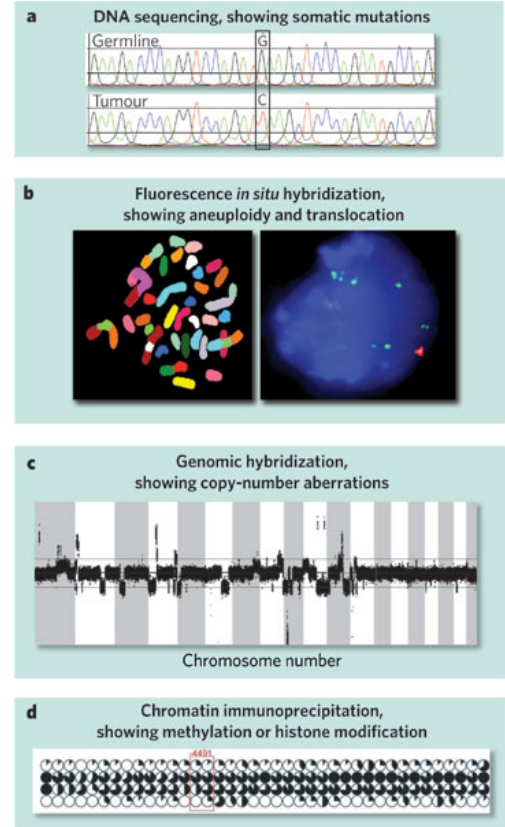
CPGRDRRTEENLRKKGEPHHELPPGSTKRALPNNTSSSPQPKKKPLDGEYFTLQIRG

RERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD





Cancer is the phenotypic end point of numerous genomic and/or epigenomic alterations that have accumulated within cells, and of the interactions of such altered cells with the stromal components in a unique host microenvironment.



- Change in level of expression
- Shift in pattern of alternatively spliced variants
- Presence of aberrant transcripts (for example, fusion transcripts)

(Chin & Gray, Nature 2008)

Bioinformatics

- Bio - Molecular Biology
- Informatics - Computer Science
- Bioinformatics - the study of the application of
 - molecular biology, computer science, artificial intelligence, statistics and mathematics
 - *to analyze, model, organize, integrate, visualize, understand and discover interesting knowledge (generate new hypothesis)* associated with the large scale molecular biology databases,
 - *to guide* assays for biological experiments.
 - *to translate* biological insights into clinical applications
 - “Computational Biology”

National Center for Biotechnology Information (NCBI)

<http://www.ncbi.nlm.nih.gov/>

The screenshot displays the NCBI Entrez search engine interface. At the top, the NCBI logo and the text "Entrez, The Life Sciences Search Engine" are visible. Below the header, a navigation bar includes links for HOME, SEARCH, SITE MAP, PubMed, All Databases, Human Genome, GenBank, Map Viewer, and BLAST. The main search area features a search bar with the query "BRCA1" and buttons for GO, Clear, and Help. Below the search bar, a message states: "Result counts displayed in gray indicate one or more terms not found". The search results are organized into two columns, each containing a list of database entries with their respective counts and descriptions. The results are as follows:

| Count | Database | Description |
|--------|--------------------|--|
| 7358 | PubMed | biomedical literature citations and abstracts |
| 4667 | PubMed Central | free, full text journal articles |
| 14 | Site Search | NCBI web and FTP sites |
| 468 | Books | online books |
| 144 | OMIM | online Mendelian Inheritance in Man |
| none | OMIA | online Mendelian Inheritance in Animals |
| 6275 | Nucleotide | Core subset of nucleotide sequence records |
| 377 | EST | Expressed Sequence Tag records |
| 192 | GSS | Genome Survey Sequence records |
| 12598 | Protein | sequence database |
| 181 | Genome | whole genome sequences |
| 80 | Structure | three-dimensional macromolecular structures |
| none | Taxonomy | organisms in GenBank |
| 730 | SNP | single nucleotide polymorphism |
| none | dbVar | Genomic structural variation |
| 3459 | Gene | gene-centered information |
| none | SRA | Sequence Read Archive |
| 54 | BioSystems | Pathways and systems of interacting molecules |
| 60 | HomoloGene | eukaryotic homology groups |
| 16 | GENSAT | gene expression atlas of mouse central nervous system |
| 1025 | Probe | sequence-specific reagents |
| none | Genome Project | genome project information |
| 2 | dbGaP | genotype and phenotype |
| 230 | UniGene | gene-oriented clusters of transcript sequences |
| 18 | CDD | conserved protein domain database |
| 149 | 3D Domains | domains from Entrez Structure |
| 159 | UniSTS | markers and mapping data |
| 59 | PopSet | population study data sets |
| 124676 | GEO Profiles | expression and molecular abundance profiles |
| 42 | GEO DataSets | experimental sets of GEO data |
| 15 | Cancer Chromosomes | cytogenetic databases |
| 76 | PubChem BioAssay | bioactivity screens of chemical substances |
| none | PubChem Compound | unique small molecule chemical structures |
| 35 | PubChem Substance | deposited chemical substance records |
| 9 | Protein Clusters | a collection of related protein sequences |
| none | Peptidome | MS/MS proteomic experiments |
| none | Journals | detailed information about the journals indexed in PubMed and other Entrez databases |
| 36 | NLM Catalog | catalog of books, journals, and audiovisuals in the NLM collections |
| 19 | MeSH | detailed information about NLM's controlled vocabulary |

Principles of Bioinformatics (Algorithms)

- **Pattern discovery** (learning)
- **Pattern matching**

Principles of Bioinformatics (Algorithms)

- **Pattern discovery** (learning)
 - Group together biological features (sequences, structures, expressions) thought to have common biological (structural, functional) properties -> families (biological - semantic level)
 - Study the purely syntactic properties common to these biological features ignoring their biological (semantic) properties -> patterns, clusters (mathematical - syntactic level)
 - Test whether the discovered patterns make sense (back to semantic level)

Biological Patterns

[illegible]

Principles of Bioinformatics (Algorithms)

- **Pattern matching**

- Deterministic: is a *boolean* function which either matches a given object (i.e. sequence, structure) or not
- R-x-Y-[ST]** (e.g. regular expression for sequence pattern)
- Probabilistic: Assigns each sequence with a probability that generated by the model. The higher the probability, the better is the match between a sequence and a pattern (e.g. Profile for sequence pattern)

BLAST

Basic Local Alignment Search Tool

J. Mol. Biol. (1990) 215, 403–410

Basic Local Alignment Search Tool

Stephen F. Altschul¹, Warren Gish¹, Webb Miller²
Eugene W. Myers³ and David J. Lipman¹

¹National Center for Biotechnology Information
National Library of Medicine, National Institutes of Health
Bethesda, MD 20894, U.S.A.

²Department of Computer Science
The Pennsylvania State University, University Park, PA 16802, U.S.A.

³Department of Computer Science
University of Arizona, Tucson, AZ 85721, U.S.A.

(Received 26 February 1990; accepted 15 May 1990)

Cited > 132,886 times

The screenshot shows the NCBI BLAST Basic Local Alignment Search Tool interface. At the top, there's a navigation bar with links for Home, Recent Results, Saved Strategies, and Help. Below this, the main section is titled "Enter Query Sequence". It includes a text input field for the accession number, GI, or FASTA sequence, a "Clear" button, and a "Query subrange" section with "From" and "To" input fields. There's also an "Or, upload file" section with a "Choose File" button and a "Job Title" input field. Below this, there's a checkbox for "Align two or more sequences". The "Choose Search Set" section includes a "Database" dropdown menu (set to "Human genomic + transcript"), a "Human genomic plus transcript (Human G+T)" dropdown, and checkboxes for "Exclude Models (XM/XP)" and "Uncultured/environmental sample sequences". There's also an "Entrez Query" input field. The "Program Selection" section has radio buttons for "Optimize for" (set to "Highly similar sequences (megablast)"), "More dissimilar sequences (discontiguous megablast)", and "Somewhat similar sequences (blastn)". At the bottom, there's a "BLAST" button and a checkbox for "Show results in a new window".

BLASTing ...

1

blastn **blastp** blastx tblastn tblastx

BLASTN programs search nucleotide databases using a nucleotide query. [more...](#)

Enter Query Sequence

Enter accession number, gi, or FASTA sequence [?](#) [Clear](#)

Query subrange [?](#)

From

To

>query
ACTGGACTGACCTGACCTAGAAGATCGAGATC

Or, upload file no file selected [?](#)

Job Title

Enter a descriptive title for your BLAST search [?](#)

☐ Align two or more sequences [?](#)

Choose Search Set

Database ☒ Human genomic + transcript ☐ Mouse genomic + transcript ☐ Others (nr etc.):

Human genomic plus transcript (Human G+T) [?](#)

Exclude ☐ Models (XM/XP) ☐ Uncultured/environmental sample sequences

Entrez Query

Enter an Entrez query to limit search [?](#)

Program Selection

Optimize for ☒ Highly similar sequences (megablast)

☐ More dissimilar sequences (discontiguous megablast)

☐ Somewhat similar sequences (blastn)

Choose a BLAST algorithm [?](#)

BLAST Search **database Human G+T** using **Megablast (Optimize for highly similar sequences)**

☐ Show results in a new window


► [Algorithm parameters](#)

2

3

On the fly BLASTing

Job Title: query

 Your search parameters were adjusted to search for a short input sequence.

| | |
|-----------------------|--------------------------|
| Request ID | 1N1PARGH01S |
| Status | Searching |
| Submitted at | Thu Jun 17 16:34:18 2010 |
| Current time | Thu Jun 17 16:34:36 2010 |
| Time since submission | 00:00:18 |

This page will be automatically updated in 1 seconds

BLAST results

► NCBI/ BLAST/ blastn suite/ Formatting Results - 1N1PARGH01S

① Your search parameters were adjusted to search for a short input sequence.

[Edit and Resubmit](#) [Save Search Strategies](#) [► Formatting options](#) [► Download](#)

query

Query ID |cl|56749
Description query
Molecule type nucleic acid
Query Length 32

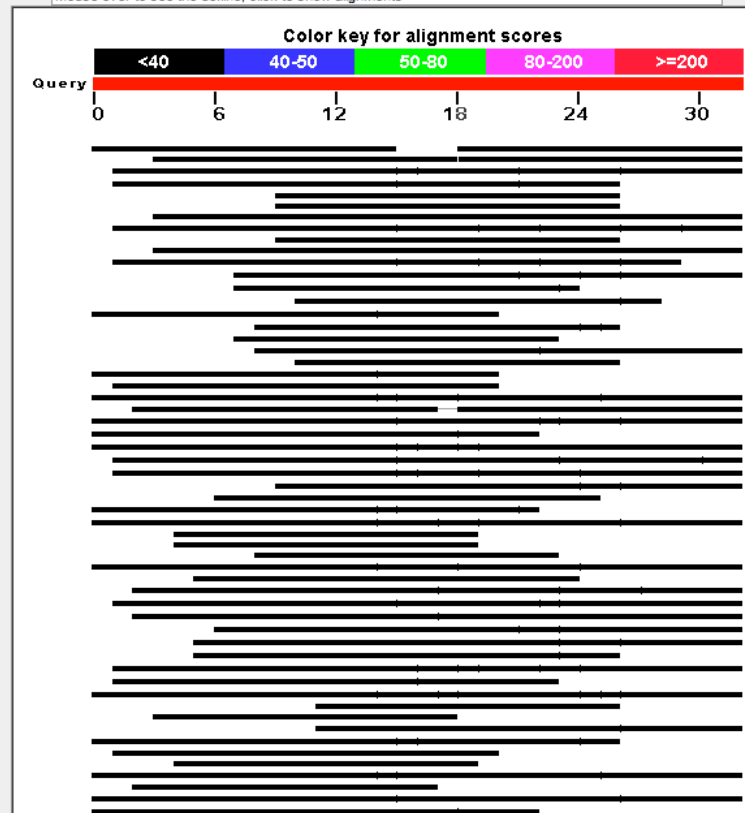
Database Name 3 databases
Description [► See details](#)
Program BLASTN 2.2.23+ [► Citation](#)

Other reports: [► Search Summary](#) [\[Taxonomy reports\]](#) [\[Distance tree of results\]](#) [\[genome view\]](#)

▼ Graphic Summary

Distribution of 340 Blast Hits on the Query Sequence

Mouse over to see the define, click to show alignments



BLAST Results

▼ Descriptions

Legend for links to other resources: [U](#) UniGene [E](#) GEO [G](#) Gene [S](#) Structure [M](#) Map Viewer

Sequences producing significant alignments:

| Accession | Description | Max score | Total score | Query coverage | E value | Max ident | Links |
|---|---|----------------------|-------------|----------------|---------|-----------|-------------------------------------|
| Transcripts | | | | | | | |
| NM_001014342.2 | Homo sapiens filaggrin family member 2 (FLG2), mRNA | 30.2 | 30.2 | 46% | 65 | 100% | G M |
| NM_005632.2 | Homo sapiens small optic lobes homolog (Drosophila) (SOL) | 30.2 | 30.2 | 46% | 65 | 100% | G M |
| Genomic sequences [show first] | | | | | | | |
| NT_010498.15 | Homo sapiens chromosome 16 genomic contig, GRCh37 re | 36.2 | 318 | 96% | 1.1 | 100% | |
| NW_001838290.1 | Homo sapiens chromosome 16 genomic contig, alternate a | 36.2 | 120 | 78% | 1.1 | 100% | |
| NT_011896.9 | Homo sapiens chromosome Y genomic contig, GRCh37 refe | 34.2 | 34.2 | 53% | 4.2 | 100% | |
| NT_011651.17 | Homo sapiens chromosome X genomic contig, GRCh37 refe | 34.2 | 62.4 | 53% | 4.2 | 100% | |
| NT_010783.15 | Homo sapiens chromosome 17 genomic contig, GRCh37 re | 34.2 | 62.4 | 90% | 4.2 | 100% | |
| NT_008413.18 | Homo sapiens chromosome 9 genomic contig, GRCh37 refe | 34.2 | 235 | 96% | 4.2 | 100% | |
| NW_001842386.2 | Homo sapiens chromosome X genomic contig, alternate as | 34.2 | 34.2 | 53% | 4.2 | 100% | |
| NW_001838448.1 | Homo sapiens chromosome 17 genomic contig, alternate a | 34.2 | 62.4 | 90% | 4.2 | 100% | |
| NW_001839149.2 | Homo sapiens chromosome 9 genomic contig, alternate as | 34.2 | 177 | 87% | 4.2 | 100% | |

▼ Alignments

☐ Select All [Get selected sequences](#) [Distance tree of results](#)

```
>☐ref|NM_001014342.2| G M Homo sapiens filaggrin family member 2 (FLG2), mRNA
Length=9170
```

```
GENE ID: 388698 FLG2 | filaggrin family member 2 [Homo sapiens]
(10 or fewer PubMed links)
```

```
Score = 30.2 bits (15), Expect = 65
Identities = 15/15 (100%), Gaps = 0/15 (0%)
Strand=Plus/Minus
```

```
Query 1      ACTGGACTGACCTGA 15
             |||||
Sbjct 2758    ACTGGACTGACCTGA 2744
```

What is Algorithm

- An ***algorithm*** is a sequence of instructions that one must perform in order to solve a well-formulated ***problem***.
- Specify ***problems*** in terms of their ***inputs*** and ***outputs***.
- ***Algorithm*** is the method of translating the ***inputs*** into the ***outputs***.

Algorithm Design Techniques

- Exhaustive Search (Brute Force)
- Branch-and-Bound Algorithm (Pruning)
Greedy Algorithm
- Dynamic Programming
- Divide-and-Conquer Algorithm
- Machine Learning
- Randomized Algorithms

Code and Pseudocode

- Code
 - Algorithm implemented in programming language (such as **C** or **Java** or **Perl** or **Python**)
 - Syntax
 - Details and specifics
- Pseudocode
 - Algorithm described in human language
 - Ignore details and specifics
 - Preserved the key operations (instructions)

Pseudocode - Assignment

Format: $a \leftarrow b$

Effect: Sets the variable a to the value b .

Pseudocode:

Example:

$b \leftarrow 2$

$a \leftarrow b$

Result: The value of a is 2

Pseudocode - Arithmetic

Format: $a+b$, $a-b$, $a \bullet b$, a/b , a^b

Effect: Addition, subtraction, multiplication, division, and exponentiation of numbers.

Pseudocode:

EUCLIDEANDIST(x_1 , y_1 , x_2 , y_2)

1. $dx \leftarrow (x_2 - x_1)^2$

2. $dy \leftarrow (y_2 - y_1)^2$

3. **return** SQRT($dx + dy$)

Result: EUCLIDEANDIST(x_1, y_1, x_2, y_2) computes the Euclidean distance between points with coordinates (x_1, y_1) and (x_2, y_2).

EUCLIDEANDIST(0, 0, 3, 4) returns 5.

Pseudocode – Conditional (if)

Format: if A is true

B

else

C

Effect: If statement A is true, executes instructions B , otherwise executes instructions C . Sometimes we will omit “else C ”, in which case, this will either execute B or not, depending on whether A is true.

Pseudocode:

MAX(a, b)

1. **if** $a < b$
2. **return** b
3. **else**
4. **return** a

Result: Max(a, b) computes the maximum of the numbers a and b .

Pseudocode – for loops

Format: for $i \leftarrow a$ to b

B

Effect: Set i to a and executes instructions B . Sets i to $a+1$ and executes instructions B again. Repeats for $i = a+2, a+3, \dots, b-1, b$.

Pseudocode:

SUMINTEGERS(n)

1. $sum \leftarrow 0$
2. **for** $i \leftarrow 1$ to n
3. $sum \leftarrow sum + i$
4. **return** sum

Result: SUMINTEGERS(n) computes the sum of integers from 1 to n .

SUMINTEGERS(10) returns 55. ($1+2+3+\dots+10$)

Pseudocode – while loops

Format: while A is true

B

Effect: Check the condition A . If it is true, then executes instructions B . Checks A again; if it is true, it executes B again. Repeats until A is not true.

Pseudocode:

ADDUNTIL(b)

1. $i \leftarrow 1$
2. $total \leftarrow i$
3. **while** $total \leq b$
4. $i \leftarrow i + 1$
5. $total \leftarrow total + 1$
6. **return** i

Result: ADDUNTIL(b) computes the smallest integer i such that $1 + 2 + \dots + i$ is larger than b .

ADDUNTIL(25) returns 7. ($1+2+3+\dots+7 = 28$; therefore $1+2+\dots+6 = 21$)

Pseudocode – Array access

Format: a_i

Effect: The i th number of array $\mathbf{a} = (a_1, \dots, a_i, \dots, a_n)$.

Pseudocode:

$\mathbf{F} = (1, 1, 2, 3, 5, 8, 13)$ [\mathbf{F} is the array containing 7 elements]

FIBONACCI(n)

1. $F_1 \leftarrow 1$
2. $F_2 \leftarrow 1$
3. **for** $i \leftarrow 3$ **to** n
4. $F_i \leftarrow F_{i-1} + F_{i-2}$
5. **return** F_i

Result: FIBONACCI(n) computes the n th Fibonacci number.

FIBONACCI(8) returns 21.

Common tasks in a data analysis pipeline

Data Preprocessing (~50% of the time)

- Converting one format to another format
- Matching patterns from certain inputs
- Parsing from one data source to another
- Comparing lists between files
- Extracting certain lines from a file

Running the Program (~15% of the time)

Data Analysis and Interpretation and Visualization (~35% of the time)

- Converting one format to another format
- Matching patterns from certain inputs
- Graph plotting and visualization techniques

Perl vs Java vs C

| Perl, Python | Java | C, C++ |
|--|---|--|
| Pros: Reformatting data files Reading, writing and parsing files Building workflow pipelines Connecting to web pages and accessing databases Writing scripts quickly, mix and match with command lines | Pros: Very fast calculations Professional grade programming language Good graphical user interfaces (GUI) | Pros: Very fast calculations Efficient memory usage |
| Cons: Higher memory usage Slower calculation performance | Cons: Requires more time to write (and skills) | Cons: Requires more time to write (and skills) |

UNIX (-like) TERMINAL

- Lots of command lines useful for file manipulation
- Coding and executing program
- Communicating with your computer (via command lines)

UNIX (-like) TERMINAL

- Cygwin – virtual machine of unix running on windows
 - Open source ware and can be downloaded from:
<http://www.cygwin.com/>
 - Make sure to install X11 and Perl packages
- Mac OSX – Terminal
 - Built in Mac OS X
 - Has most of the X11 codes and Perl

Data Structures

- Scalar Data
- Lists and Arrays

Data Structures

- Scalar Data
 - Singular
 - Single variable
 - Number
 - String
 - Can act on scalar value with operators (like addition or concatenation) to generate a scalar result
 - Can store a scalar value in a scalar variable
 - Read scalars from files and devices
 - Write scalars to files and devices

Scalar - Numbers

- Floating-point literals

1.25

255.0001

7.36e10

- Integer literals

0

100

123456789

- Numeric operators

+ # addition

- # subtraction

* #multiplication

/ #division

Scalar - Strings

- Strings are sequences of characters

hello

- Strings may contain any combination of any characters
- Single-quoted string literals
 - 'hello' #those five characters: h,e,l,l,o
 - ' ' #null/empty string
- Double-quoted string literals
 - "hello" #this is the word hello
 - "hello\n" #hello and a new line

Scalar Variables

- A variable is a name for a container that holds one or more values.
- Scalar variable holds exactly one value.
- The name of the variable stays the same throughout your program, however, the value of the variable can change over and over again throughout the execution of the program.
- In perl, a scalar variable name begins with a dollar sign (\$) followed by a letter or underscore and more letters, digits etc.

`$name`

`$Name`

`$NAME`

Scalar Assignment

- Assign a value to a variable

```
$gene = 17; #give $gene the value of 17
```

```
$mygene = 'KRAS'; #give $mygene the four-character  
string 'KRAS'
```

```
$anothergene = "BRAF"; #give $anothergene the  
string "BRAF"
```

```
$newgene = $gene + 3; # give $newgene the value of  
17+3 = 20
```


Numeric and String Comparison Operators

| Comparison | Numeric | String |
|--------------------------|---------|--------|
| Equal | == | eq |
| Not equal | != | ne |
| Less than | < | lt |
| Greater than | > | gt |
| Less than or equal to | <= | le |
| Greater than or equal to | >= | ge |

if Operator

```
if ($value > 10)
{
    print "The value is greater than 10\n";
}
else
{
    print "The value is smaller than 10\n";
}
```

for Operator

```
for ($value = 0; $value < 10; $value++)  
{  
    print "The value is $value\n";  
}
```

#This will print

```
The value is 0  
The value is 1  
The value is 2  
The value is 3  
The value is 4  
The value is 5  
The value is 6  
The value is 7  
The value is 8  
The value is 9
```

while Operator

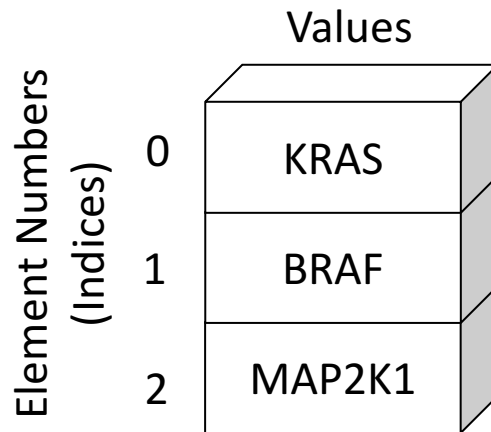
```
$count = 0;
while ($count < 10)
{
    print "The value is $count\n";
    $count = $count + 1;
}
```

#This will print

```
The value is 0
The value is 1
The value is 2
The value is 3
The value is 4
The value is 5
The value is 6
The value is 7
The value is 8
The value is 9
```

Data Structures

- Lists and Arrays
 - Plural
 - Many variables (can be numbers or strings)
 - List is an ordered collection of scalars (data)
 - Array is a variable that contains a list (variable)
 - Same operations as scalar



A list with three elements.

Arrays in Perl

- Arrays elements are numbered using sequential integers
- Beginning at **zero** and increasing by one for each element (***VERY IMPORTANT**)

```
$gene[0] = "KRAS";
```

```
$gene[1] = "BRAF";
```

```
$gene[2] = "MAP2K1";
```

- The array name itself (in this case, gene) is from a completely separate namespace than scalars use. E.g. you can have a scalar value \$gene in the same program, and Perl will treat them differently and will not be confused (but you may be confused).

- Here, the array gene has three elements:

```
@gene = ("KRAS", "BRAF", "MAP2K1")
```

```
print "$gene[1]\n";
```

BRAF

Arrays in Perl

- Find how many arrays elements in an array

```
$number_of_genes = $#gene; (REMEMBER, array index start at 0)
```

```
$number_of_genes_real = $#gene + 1;
```

```
$gene[$#gene] = "ERK"; (last element in the array)
```

```
$gene[-1] = "ERK"; (short cut to assessing the last element in the array)
```

- Copying a list from one array to another

```
@my_new_genes = @genes;
```

pop and push Operators

- pop operator takes the last element off of an array and returns it:

```
@gene = ("KRAS", "BRAF", "MAP2K1");
```

```
$last_gene = pop(@gene); ##now @gene has 2 elements, KRAS and BRAF
```

```
print $last_gene; ##$last_gene = MAP2K1
```

- push operator adds a new element to the last index in an array :

```
$new_gene = "ERK";
```

```
push(@gene, $new_gene); ##Now @gene has 3 elements, KRAS, BRAF and ERK
```


shift and unshift Operators

- `shift` operator takes the first element off of an array and returns it:

```
@gene = ("KRAS", "BRAF", "MAP2K1");
```

```
$first_gene = shift(@gene); ##now @gene has 2 elements, BRAF and MAP2K1
```

```
print $first_gene; ##$first_gene = KRAS
```

- `unshift` operator adds a new element to the first index in an array :

```
$new_gene = "ERK";
```

```
unshift(@gene, $new_gene); ##Now @gene has 3 elements, ERK, BRAF and  
MAP2K1
```

foreach Operators

- `foreach` operator loops through a list of values, executing one iteration (time through the loop) for each value in an array:

```
foreach $gene (@gene)
{
    print "$gene\n";
}
```

I/O for Perl

- <STDIN> simplest way to get input from Users (or files)
 - this is a command line input

```
$line = <STDIN>
if ($line eq "\n")
{
    print "That was just a blank line!\n";
}
else
{
    print "That line of input was: $line\n";
}
```

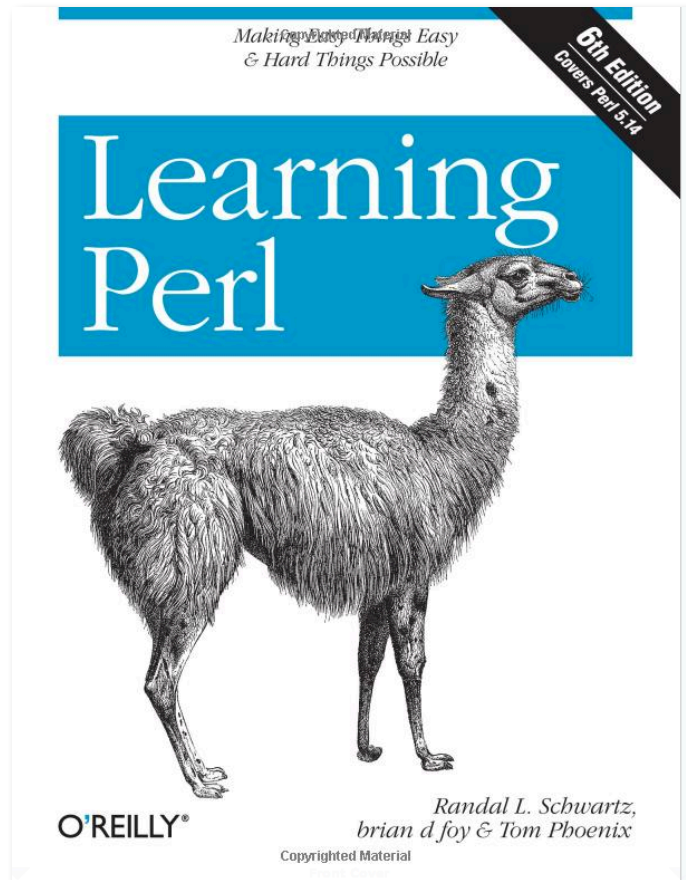
Output for Perl

- `<STDIN>` simplest way to get input from Users (or files)
 - this is a command line input

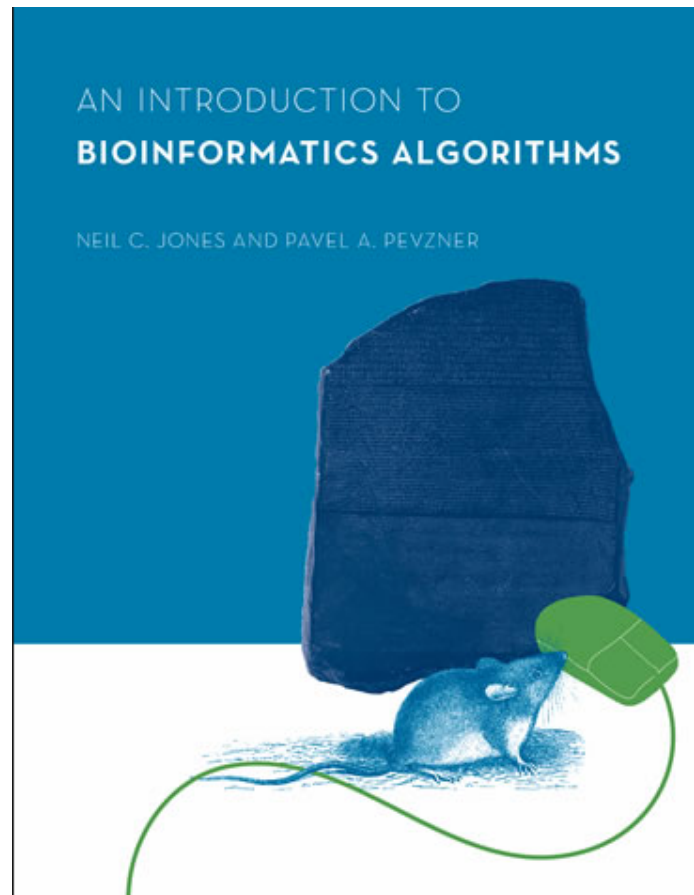
```
@line = <STDIN> #reading standard input in list context
```

More information on Perl

- <http://learn.perl.org/>

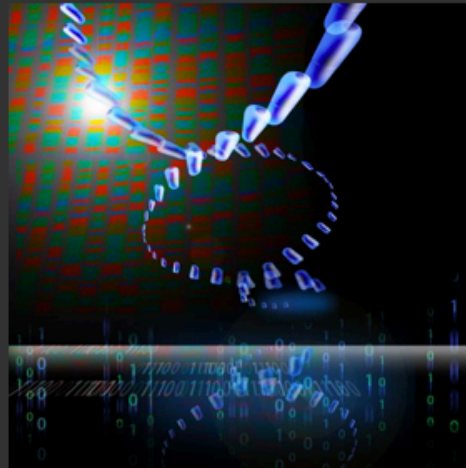


More information on Algorithms for bioinformatics



PLOS Computational Biology : Collection of Translational Bioinformatics

www.ploscollections.org/translationalbioinformatics



COVER

Image Credit: PLOS

'Translational Bioinformatics' is a collection of *PLOS Computational Biology* Education articles which reads as a "book" to be used as a reference or tutorial for a graduate level introductory course on the science of translational bioinformatics.

Translational bioinformatics is an emerging field that addresses the current challenges of integrating increasingly voluminous amounts of molecular and clinical data. Its aim is to provide a better understanding of the molecular basis of disease, which in turn will inform clinical practice and ultimately improve human health.

The concept of a translational bioinformatics introductory book was originally conceived in 2009 by Jake Chen and Maricel Kann. Each chapter was crafted by leading experts who provide a solid

introduction to the topics covered, complete with training exercises and answers. The rapid evolution of this field is expected to lead to updates and new chapters that will be incorporated into this collection.

Collection editors: Maricel Kann, Guest Editor, and Fran Lewitter, *PLOS Computational Biology* Education Editor.

Download the full Translational Bioinformatics collection here: [PDF](#) | [EPUB](#) | [MOBI](#)

Read *PLOS Computational Biology* Founding Editor-in-Chief Phil Bourne's blog post: '[Let's make those book chapters open too!](#)'

Collection URL: www.ploscollections.org/translationalbioinformatics