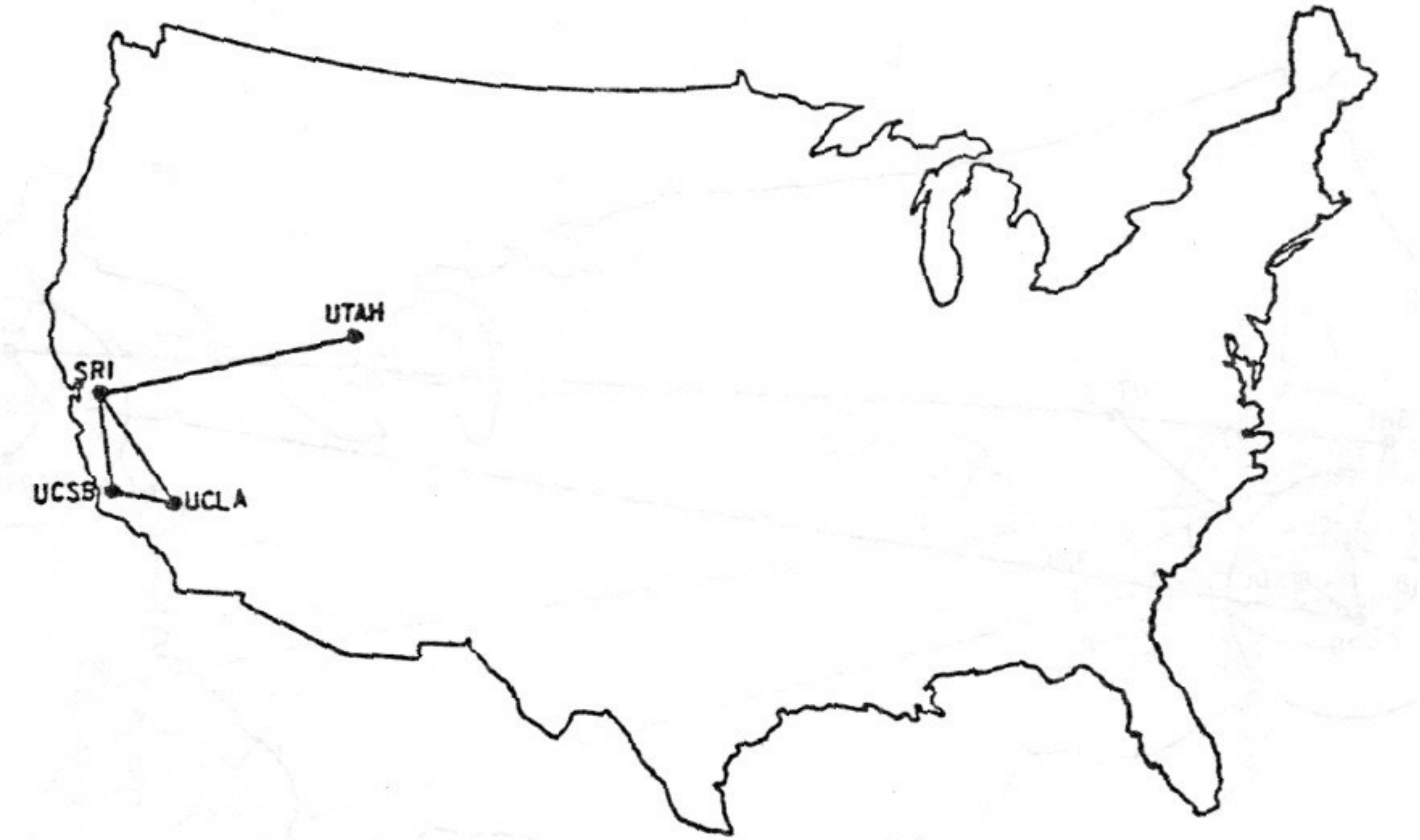
ICQB

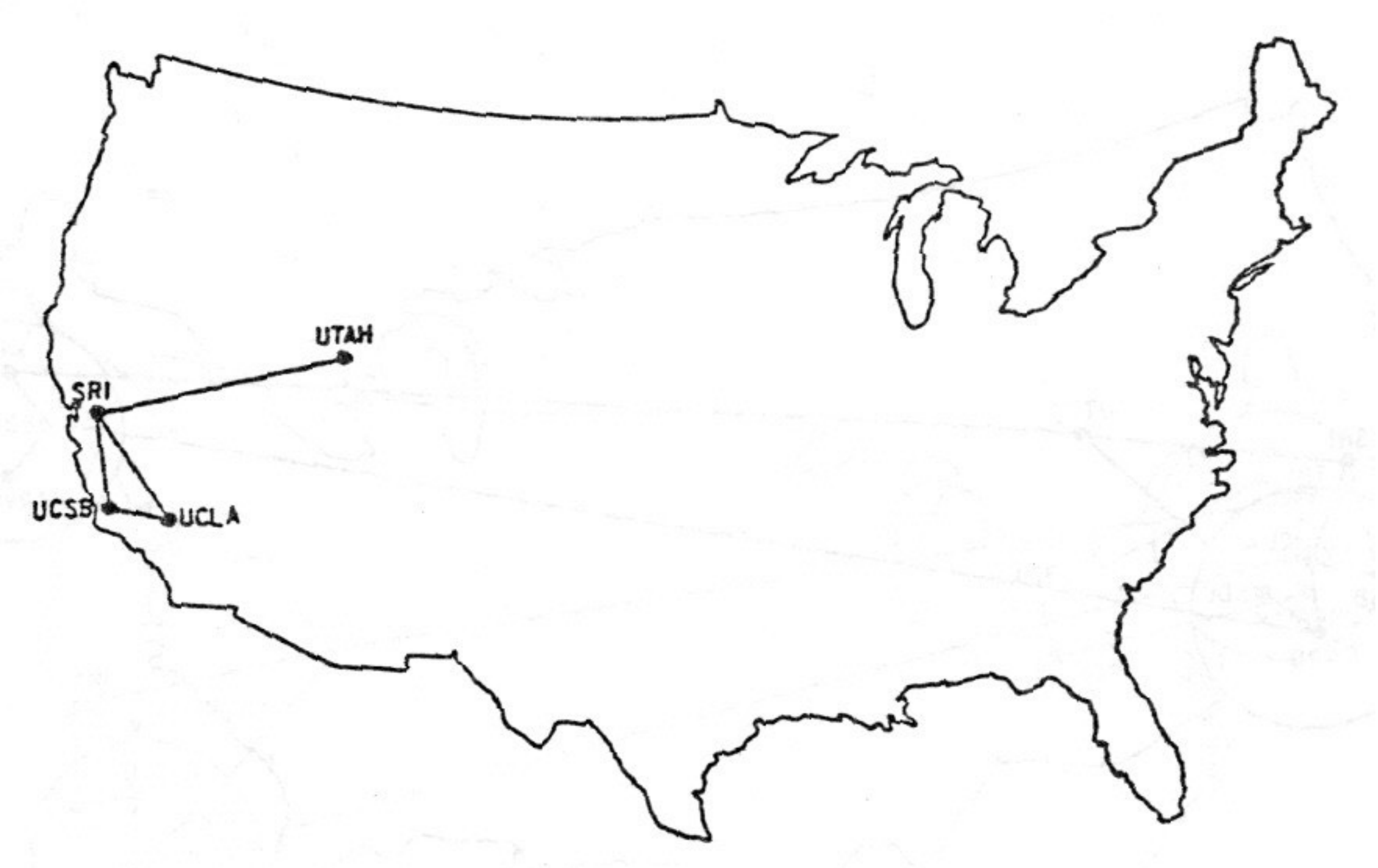
Introduction to Computational & Quantitative Biology (G4120)
Fall 2022
Oliver Jovanovic, Ph.D.
Columbia University
Department of Microbiology & Immunology

Lecture 2: Databases and Internet Resources
September 20, 2022



Lecture 2: Databases and Internet Resources
September 20, 2022

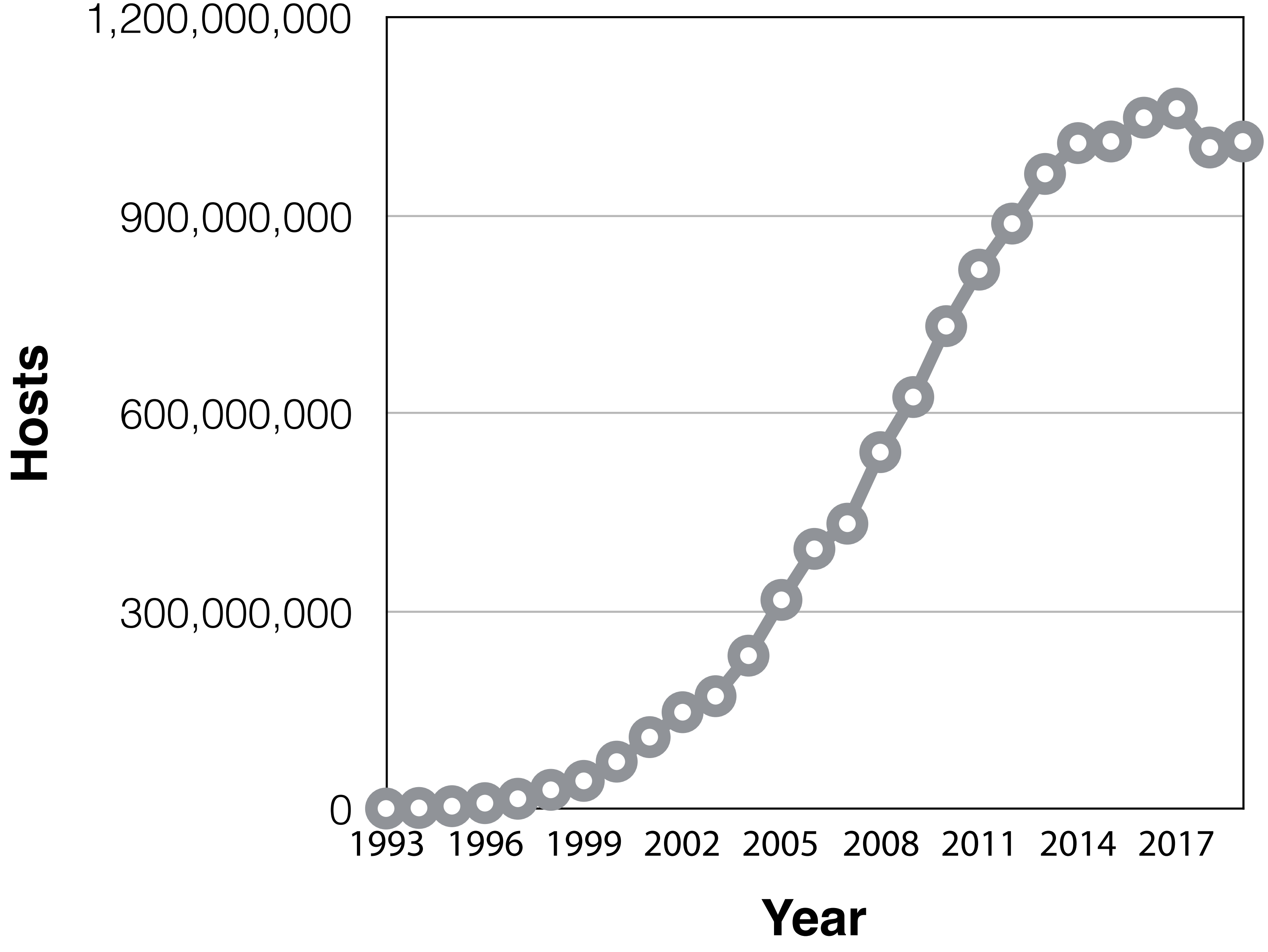
The Internet in 1969



The ARPANET in December 1969

Source: Heart, F., McKenzie, A., McQuillian, J. and Walden, D., ARPANET Completion Report, 1978

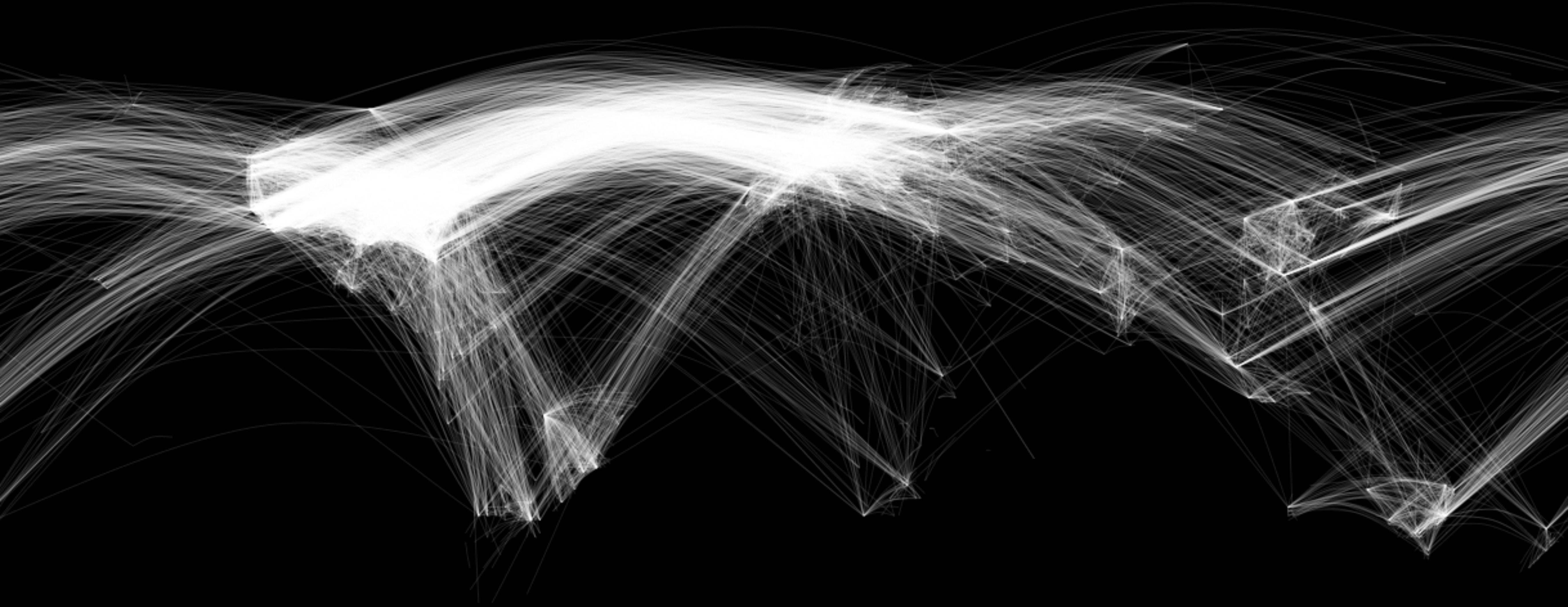
Growth of the Internet



From 1.3 million IPv4 Internet hosts in 1993 to over 1 billion in 2019
(Excludes IPv6 hosts, which are now a substantial minority.)

Source: Data from January 2019
Internet Domain Survey, **www.isc.org**

The Modern Internet



Source: Chris Harrison,
Carnegie Mellon University

Lecture 2: Databases and Internet Resources
September 20, 2022

Internet Protocols

TCP/IP (Transmission Control Protocol/Internet Protocol)

A multilayered protocol architecture that allows for the reliable transmission of data over networks. TCP provides connection oriented streaming with error detection and correction between two computers. IP governs the delivery of datagrams, packets of data containing the IP addressing information needed to switch them from one network to another until they arrive at their final destination. The older version is IPv4, slowly being succeeded by the new version, IPv6.

UDP (User Datagram Protocol)

Provides low overhead connectionless datagram delivery.

DHCP (Dynamic Host Configuration Protocol)

Automates the configuration of computers that use TCP/IP by automatically assigning Internet addressing information from a network DHCP server.

HTTP (HyperText Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure)

The protocol behind the World Wide Web (WWW). The secure form is **HTTPS**, which provides an encrypted **SSL/TLS** connection that depends on a trusted certificate authority and public key infrastructure and is now commonly used.

DNS (Domain Name Server)

A hierarchical naming system used to locate servers on the WWW by name instead of numerical IP address.

URL (Uniform Resource Locator)

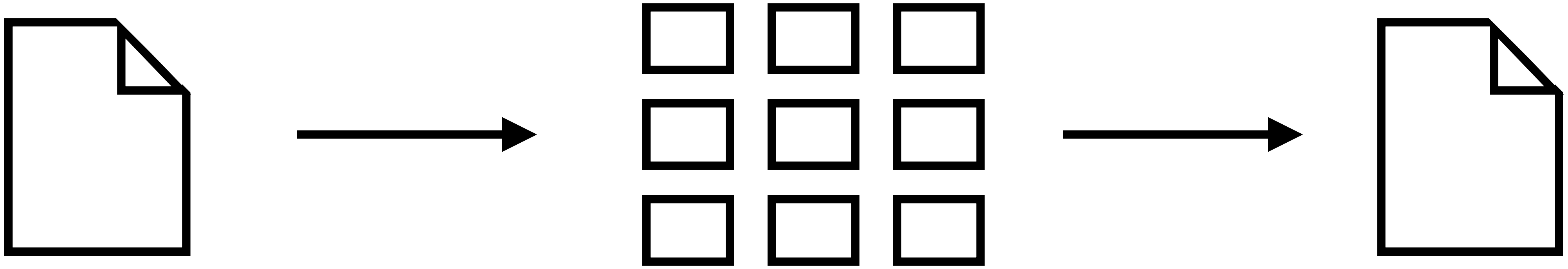
Used to locate resources on the WWW. Various prefixes, usually HTTPS, so **https://www.columbia.edu** but could be **ftp://ftp.ncbi.nlm.nih.gov** or **file://Macintosh%20HD/Documents/Read%20Me.pdf** (note, a blank space in a URL “ ” is replaced by a “**%20**”). The DNS part is not case sensitive, but the rest of the URL can be.

TCP/IP Details

Step 1
Data file broken into TCP packets and sent to destination

Step 2
Packets travel to destination from router to router via IP

Step 3
Packets reassembled at destination into original file



Transmission Control Protocol (TCP) Header
Each TCP packet has a header with 10 required fields totaling 20 bytes, which include information such as the source port number, destination number, sequence number, acknowledgement number and a checksum. An optional data field of up to 40 bytes can be appended. The receiver will acknowledge each packet received, and if a packet does not arrive at its destination, the protocol will automatically resend it, which makes the protocol reliable.

File Transfer Protocols

FTP (File Transfer Protocol)

Allows you to transfer files to or from a remote machine running FTP. Usually anonymous, but insecure. Use Fetch (for Mac, free to students at **fetchsoftworks.com/fetch/free**) or FileZilla (Mac or Win, free at **filezilla-project.org**).

SFTP (Secure File Transfer Protocol)

Allows you to securely transfer files with the data encrypted. Use Fugu (for Mac, free at: **sourceforge.net/projects/fugussh/files**) or WinSCP (for Win, free at **winscp.net**) or FileZilla.

AFP (Apple File Protocol)

Allows you to transfer files to or from a Macintosh.

SMB (Server Message Block)

Allows you to transfer files to or from a PC.

SMTP (Simple Mail Transfer Protocol)

Used by email systems and clients.

SSH (Secure Shell)

Can use to securely login remotely or forward outgoing email. Can transfer files securely using secure copy (SCP) or FTP over an SSH tunnel.

Internet Addressing

IP Address (IPv4)

An IP address is a 32 bit number, written in the form of four decimal numbers in the range 0-255 separated by dots (e.g. **128.59.48.24**). Columbia University Irving Medical Center (CUIMC) public IP addresses will always have the format **156.111.x.x** or **156.145.x.x.** A private address, such as a printer on a local VLAN, will have the format **10.x.x.x**.

Subnet Mask

A subnet mask allows for defining a local network, called a subnet, within a larger network. CUIMC subnet masks are always in the format **255.255.255.0.**

Router

A device that routes packets of data between networks. A router sits between your computer and local area network and the networks beyond it. CUIMC router IP addresses generally match the public address but end in a **.1**, so **156.111.x.1** or **156.145.x.1.**

DNS (Domain Name Server)

These specialized servers automatically translate an easy to remember domain name (e.g. **microbiology.columbia.edu**) into the appropriate IP address (e.g. **156.111.98.150**). CUIMC DNS IP addresses are **10.70.235.40** and **10.168.1.20.**

Ethernet Address

A unique 48 bit number, usually written in the form of 12 hexadecimal digits in six groups of two digits each separated by colons (e.g. **00:03:93:bc:3c:18**), which is assigned to every piece of network hardware, including Ethernet cards and AirPort cards. It is also called a MAC (media access control) address. When registering a wired device, use the Ethernet MAC, not the wireless MAC.

IPv6

The latest version of the Internet Protocol will eventually replace 32 bit IPv4 addresses with 128 bit IPv6 addresses, to address the growing problem of IPv4 address exhaustion. An IPv6 address is represented by 32 hexadecimal digits in eight groups of four digits each separated by colons. Leading zeros can be omitted, all zero groups can be represented by a single zero, and the leftmost longest run of all zeros can be replaced by a double colon (e.g. **::ffff:803b:3018** instead of **0000:0000:0000:0000:0000:ffff:803b:3018**).

Hexadecimal Notation

Hexadecimal notation is a 16 digit notation often used in programming. It allows you to use powers of two easily without resorting to binary notation.

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

CUIMC Internet

The CUIMC campus network has redundant core routers linked to a router in each building. Each floor of a building also has its own router. The campus has its own 40 Gbps WAN with redundant 10 Gbps links to the main campus and other sites, as well as partial wireless network coverage.

The CUIMC network is walled off from the rest of the Internet by a firewall and is centrally administered by CUIMC IT. See **https://cumcprod.service-now.com/kb?id=kb_services** for details. All network traffic is monitored for threats and suspicious activity, including, but not limited to potential HIPPA, HITECH, PHI or PII violations. Third parties can track copyright violations to campus computers, resulting in fines and potentially loss of network access.

Computers personally owned by students must be registered through the CUIMC IT Service Desk on the second floor of the Hammer Building after they are scanned to ensure they are malware free and have up to date operating systems. To register a lab owned computer for access, see **https://cumcprod.service-now.com/kb?id=kb_article_view&sys_kb_id=f66fac611b47eb009ff2bbfccd4bcb38** for details, and call **5-HELP** to register a device. For wired access, you will need to provide your computer's Ethernet adapter hardware address, also known as Media Access Control (MAC) address. This is a 12 digit hexadecimal number (e.g. **00:00:af:a0:b1:89**). If you have problems getting connected, try calling the CUIMC computer help line at **212-305-HELP** or see the help desk on the second floor of the Hammer Building.

CUIMC Network Settings

IP Address: 156.111.x.x or 156.145.x.x
Subnet Mask: 255.255.255.0
Router: 156.111.x.1 or 156.145.x.1

DNS Servers
10.70.235.40
10.168.1.20

Internet Resources

National Center for Biotechnology Information (NCBI)

- PubMed, PubMed Central, Books and other reference material
- GenBank, RefSeq, CDD, MMDB and other sequence and structure databases
- Prokaryotic genome data and browsers (complete and in progress)
- Eukaryotic genome data and browsers (complete genomes, also in progress, maps, partial sequences)
- BLAST, PSI-BLAST, Primer-BLAST, DELTA-BLAST and VAST search tools, Cn3D visualization tool

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

Ensembl, EMBL-EBI and Biocatalogue

Ensembl provides access to genome data and browsers, including vertebrates (human, chimp, mouse, rat, etc.), metazoa (C. elegans, D. melanoganster, A. gambia, etc.), plants, fungi, protists and bacteria. EMBL-EBI provides database search and sequence analysis tools. Biocatalogue provides a biological web services directory.

http://www.ensembl.org/ and **http://www.ebi.ac.uk/** and **http://biocatalogue.org/**

UCSC Genome Bioinformatics and ENCODE

Genome data and browsers including mammals (human, chimp, gorilla, rhesus, mouse, rat, dog, cat, horse, cow and platypus among others), vertebrates (chicken, X. tropicalis, zebrafish, elephant shark and fugu among others), deutrosomes (lancelet, C. intestinalis and S. purpuratus) insects (D. melanogaster, D. simulans and A. gambiae among others), nematodes (C. elegans and C. briggsae among others), virus (Ebola) and other (S. cerevisiae). The Encyclopedia of DNA Elements (ENCODE) provides a database of functional elements in the human and mouse genomes.

http://genome.ucsc.edu/ and **http://www.encodeproject.org**

Protein Data Bank (PDB) and Expert Protein Analysis System (ExPASy)

PDB is a worldwide repository for 3D protein structure data and biochemical information. ExPASy is a bioinformatics resource portal with links to useful tools, software and references for genomics, proteomics and systems biology.

http://www.rcsb.org/ and **http://www.expasy.org/**

National Center for Biotechnology Information (NCBI)

NCBI

https://www.ncbi.nlm.nih.gov

- PubMed (over 34 million citations), PubMed Central (over 7 million full text articles from over 10,000 journals), textbooks, other reference material
- GenBank, RefSeq, CDD, MMDB and other sequence and structure databases
- Nearly 19 trillion base pairs (19 terabases) from over 532,000 species
- Prokaryotic genome data and browsers (over 437,000 microbial genomes (over 31,000 complete assemblies), over 51,000 viruses, over 41,000 plasmids, and additional sequences)
- Eukaryotic genome data and browsers (nearly 25,000 eukaryotic genomes and over 25,000 organelles)
- BLAST, Primer-BLAST, Smart-BLAST, Ig-BLAST, MOLE-BLAST, CD Search, CDART and VecScreen search tools, global and multiple alignment tools, HMM annotation tools, command line tools.

NCBI BLAST and BLAST Tutorials

https://blast.ncbi.nlm.nih.gov/

https://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf

https://www.ncbi.nlm.nih.gov/Class/BLAST/blast_course.short.html

https://www.ncbi.nlm.nih.gov/Class/BLAST/blast_course.html

Software Resources

European Molecular Biology Open Software Suite (EMBOSS)
http://emboss.sourceforge.net/

J. Craig Venter Institute Open Source Software Tools
https://www.jcvi.org/research/software-tools#projects

SourceForge
http://sourceforge.net

GitHub
https://github.com

The Bio Projects
www.bioconductor.org (R), www.biojava.com (Java), www.bioperl.org (Perl), www.biopython.org (Python), www.bioruby.org (Ruby)

DNASTAR Lasergene Departmental Site License
https://microbiology.columbia.edu/dnastar

Information Security

CUIMC is subject to HIPPA and HITECH regulations, so is particularly sensitive to Internet and computer security issues.

PHI and PII

Any Protected Health Information (PHI) or Personally Identifiable Information (PII) must be stored on fully disk encrypted computers, mobile devices or memory sticks. This information must also be protected in transit. Encrypting laptops, mobile devices and memory sticks is generally a good idea, since they can easily be stolen or lost, putting your personal data at risk. Using a reliable backup system allows for easy recovery of data from such a loss.

Malwarebytes

Anti-virus and anti-malware software (Mac or Win, available free to faculty, staff or students for personal use at **https://sec-downloads.cuit.columbia.edu/malwarebytes-students**)

Encryption

Disk Encryption

Full disk encryption with pre-boot authentication is required for CUIMC computers storing PHI or PII. This is automatically provided on Macs by FileVault using AES-XTS block cipher encryption (**System Preferences > Security & Privacy > FileVault**) and on Windows by commercial solutions such as Symantec Endpoint Encryption (formerly known as GuardianEdge). Disk encryption typically uses symmetric key algorithms, which use the same key to encrypt and decrypt the data. Care must be taken to use a memorable password and store the password (or a recovery key) in a secure manner. When backing up a fully encrypted disk, make sure the backups are encrypted as well.

File Encryption

Individual files or folders can be encrypted or stored on encrypted virtual volumes. This can be useful for securing or transporting sensitive personal data when full disk encryption is not available. Disk Utility on any Mac can create an encrypted disk image (.dmg) using up to 256 bit AES encryption. GnuPG is free software available for Mac, Windows and Unix at **www.gnupg.org** that supports file encryption with a variety of algorithms, including RSA, up to 256 bit AES, Blowfish and Twofish.

Public Key Encryption

Public key cryptography is often used to secure communication across open networks such as the Internet. It is based on the use of two keys, one private, which remains secret, the other public. The keys are related in a mathematically intractable manner. The public key is used to encrypt or verify a signature, the private key is used to decrypt or sign. An Enveloped Public Key Encryption (EPKE) method adds additional security. The RSA algorithm is the best known public key encryption system, and is based on the mathematical difficulty of factoring the product of two large prime numbers.

Backup

Time Machine

Macintosh computers come with a built in backup system called Time Machine that creates a complete incremental backup that allows for the restoration of accidentally deleted files or earlier version of files as well as the entire computer. It automatically backs up hourly to an external drive or Time Capsule and keeps hourly backups for 24 hours, daily backups for a month, and weekly backups for as long as there is space. Encryption is a built-in option.

Windows Backup

Windows has built in backup systems which vary slightly from version to version. Vista and Windows 7 let you use Backup and Restore (from Control Panel) to backup files or the entire computer, allowing you to select the backup drive and set up an automatic backup schedule. Windows 8, 10 and 11 have a new backup system called File History (from Control Panel) which allows for partial Time Machine style incremental backups of user account files.

Cloud Backup

A number of commercial cloud backup services exist, including Carbonite and CrashPlan. Personal cloud storage through iCloud, Google Drive, or Dropbox can serve to backup data as well. Cloud backup offers the advantage of automatic offsite backup, but is not approved for use at CUIMC, so is only appropriate for personal use with data that is not sensitive.

Cloud Computing

Cloud computing uses networks of remote servers to provide data storage and computing services across the Internet. Amazon and Google are two major cloud service providers.

Amazon Web Services (AWS)

Using Amazon's Elastic Compute Cloud (EC2), virtual servers with storage can be launched as needed (even thousands at once), billed in hourly increments, depending on server performance. Persistent Elastic Block Store (EBS) storage is included. Auto Scaling to adjust capacity is free. Elastic Load Balancing and detailed monitoring is available at additional cost. EC2 server instances can run either Linux or Windows, and you can either create your own Amazon Machine Images (AMIs) or use a variety of premade AMIs. Elastic MapReduce (EMR) costs extra, but allows easy launching of a server as a node in a Hapdoop cluster (MapR or Amazon Hadoop). Cognito provides authentication services.

Google Compute Engine

Virtual Linux machines can be launched on demand as needed, billed by the minute (10 minute minimum), depending on performance. Persistent Google Cloud Storage is extra. BigQuery allows for interactive analysis of massive datasets at a cost that varies with the amount of data processed in an analysis. Firebase allows for rapid database and app development.

Cloud Storage

Amazon S3

Simple Storage Service. Cloud storage with low monthly costs, transfer in is free, transfer out from S3 to the Internet is charged at a low cost. Can be used to host static web content or entire websites. CloudFront allows delivery from a global network of edge locations for rapid downloads in other areas of the world at an additional cost.

Amazon Glacier

Archival cloud storage with very low long term monthly storage costs, transfer in free, retrieval or transfer out has a low cost varying with the speed of retrieval.

Google Cloud Storage

Cloud storage with low monthly costs, free transfer in, and low cost transfer out. Can be used to host static web content or static web sites. Data can be stored in the U.S. or Europe.

GitHub and Version Control

GitHub

GitHub is a source code hosting platform for software development and version control that uses the open source Git distributed version control system to track changes in sets of files called repositories, allowing for collaborative source code and software development.

GitHub is currently used by over 83 million developers and contains over 200 million software repositories, including over 28 million public repositories. It is the largest such platform, but alternatives such as GitLab exist.

Distributed Version Control

A type of version control in which the complete set of files and history of file changes are kept on each developer’s local computer. Changes made locally are “committed”, then “pushed” up to the distributed version control system, from which they can be “pulled” by other developers. New repositories are created by “initializing” them. An existing repository on the distributed version control system can be “cloned” or independently “forked”.

Algorithms

An algorithm is simply a series of steps used to solve a problem. One of a computer's great strengths is its ability to rapidly and accurately repeat recursive steps in an algorithm. This ability is essential to the data processing functions of modern computers. Many algorithms of specific interest to biologists, such as the BLAST algorithm, could not be practically applied without computers.

The first bioinformatics algorithms for comparing sequences to each other attempted to find the best possible (optimal) alignment. With the tremendous growth of sequence data, many modern search algorithms use heuristic (rule of thumb) approaches which may not find the absolutely best solution, but will quickly find a good solution.

Algorithms can improve as the underlying problem is better understood. Early algorithms for searching sequence data for significant matches depended on consensus sequences. It rapidly became clear that biologically significant sequences rarely perfectly matched a consensus, and more sophisticated approaches were adopted, including the use of matrices, Markov chains and hidden Markov models.

Optimal vs. Heuristic Algorithms

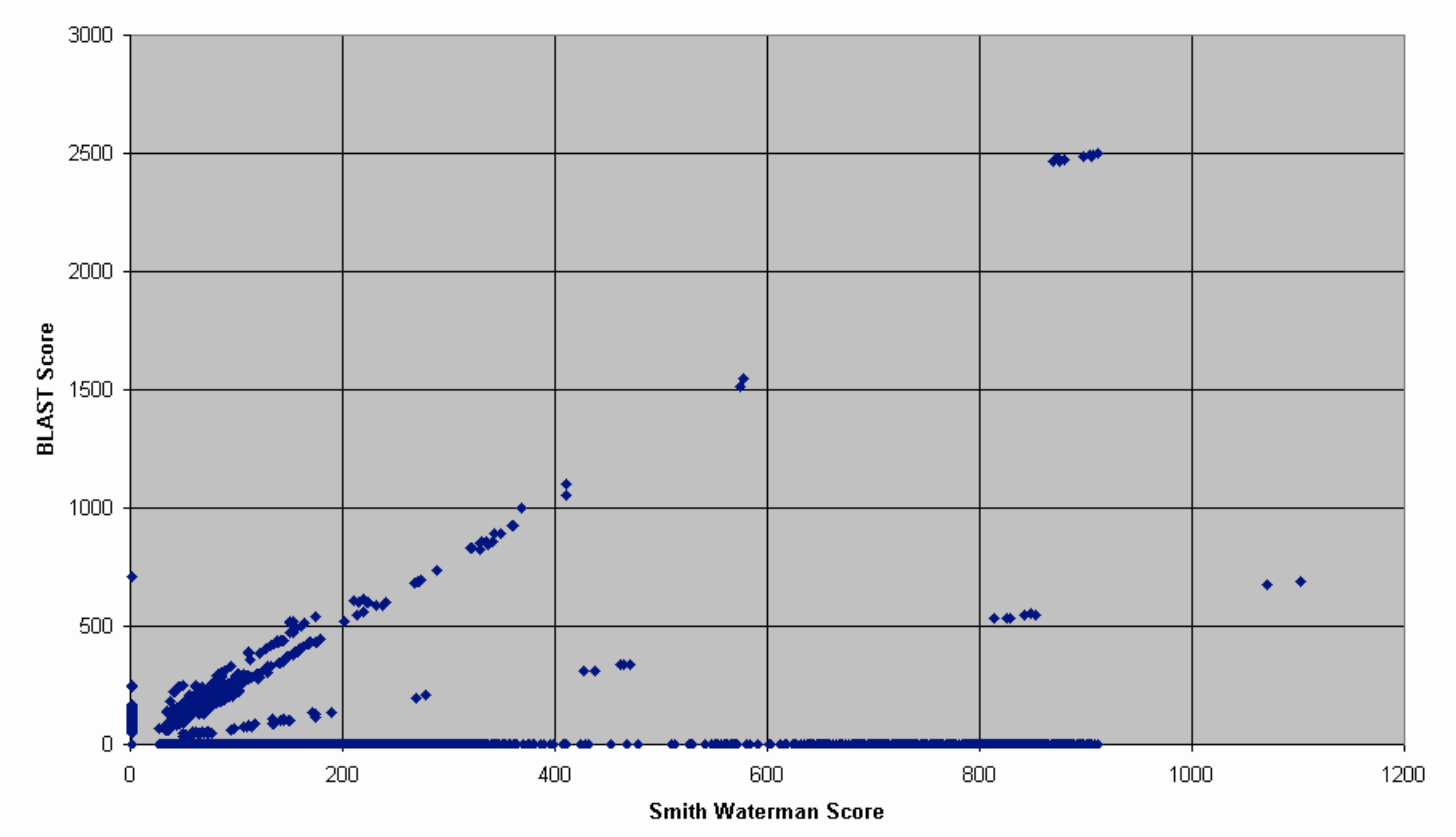
Optimal Algorithm

- Finds the optimal solution to a problem.
- Often uses an approach called dynamic programming, which solves the problem by breaking it into smaller subproblems, which are separately solved, then sequentially reassembled to solve the entire problem.
- This approach was first applied to solving biological sequence comparison problems by Saul Needleman and Christian Wunsch in 1970, and can be used to solve either a global comparison problem (Needleman-Wunsch) or a local comparison problem (Smith-Waterman).

Heuristic Algorithm

- Solves a problem by using rules of thumb to reach a solution. The solution is not guaranteed to be an optimal solution, but is generally arrived at far faster than using an optimal solution approach such as dynamic programming.
- In sequence comparison, well known heuristic approaches include the BLAST algorithm, developed by Stephen Altschul in 1990, and the FASTA algorithm, developed by William Pearson in 1988.
- Although a heuristic algorithm such as BLAST may be 100 to 1,000 times faster than an optimal algorithm such as Smith-Waterman, it may miss matches found by the optimal algorithm.

Smith-Waterman (Optimal) vs. BLAST (Heuristic)



Source: Dwan, C., University of Minnesota Center for Computational Genomics and Bioinformatics, 2001.

Search Algorithms in Bioinformatics

Global Alignment Search

- **Needleman-Wunsch** algorithm, Needleman & Wunsch, 1970
- Finds the best complete alignment of two sequences that maximizes the number of matches and minimizes the number of gaps.

Local Alignment Search

- **Smith-Waterman** algorithm, Smith & Waterman, 1981
- Makes an optimal alignment of the best segment of similarity between two sequences.
- Often better for comparing sequences of different lengths, or when looking at a particular region of interest.

Heuristic Approximations to Smith-Waterman

- **FASTA**, Pearson, 1988
- **BLAST**, Altschul, 1990
- **Gapped BLAST and PSI-BLAST**, Altschul, 1997
- **BLAT**, Kent, 2002
- **DELTA-BLAST**, Boratyn, 2012

Global Alignment (Needleman-Wunsch)

Gap, from the GCG Wisconsin Package, uses the algorithm of Needleman and Wunsch to find the alignment of two complete sequences that maximizes the number of matches and minimizes the number of gaps.

```
GAP RK2_ssb x Ecoli_ssb       January 29, 2003 00:07


              .         .         .         .         .
   1 ..MSHNQFQFIGNLTRDTEVRHGNSNKPQAIFDIAVNEEWRNDA.GDKQE 47
          |.   :||| .| |||:  .    |   :| .| ||. | |: .|
   1 ASRGVNKVILVGNLGQDPEVRYMPNGGAVANITLATSESWRDKATGEMKE 50
              .         .         .         .         .
  48 RTDFFRIKCFGSQAEAHGKYLGKGSLVFVQGKIRNTKY.EKDGQTVYGTD 96
     .|:. |:  ||  ||   .|| ||| |:::|.:|  |: :. ||  | |:
  51 QTEWHRVVLFGKLAEVASEYLRKGSQVYIEGQLRTRKWTDQSGQDRYTTE 100
              .         .         .         .         .
  97 FIAD...KVDYLDTKAPGGSNQE........................... 116
      : .    .  |  :  ||.
 101 VVVNVGGTMQMLGGRQGGGAPAGGNIGGGQPQGGWGQPQQPQGGNQFSGG 150
              .         .
     ...........................

 151 AQSRPQQSAPAAPSNEPPMDFDDDIPF 177

Matrix:  blosum62
Gap Penalties:  default
Length:  177
Percent Similarity:  45.690
Percent Identity:  32.759
```

Local Alignment (Smith-Waterman)

BestFit, from the GCG Wisconsin Package, makes an optimal alignment of the best segment of similarity between two sequences. Optimal alignments are found by inserting gaps to maximize the number of matches using the local homology algorithm of Smith and Waterman.

```
BESTFIT RK2_ssb x Ecoli_ssb       January 29, 2003 00:08


                  .         .         .         .         .
       4 NQFQFIGNLTRDTEVRHGNSNKPQAIFDIAVNEEWRNDA.GDKQERTDFF 52
         |.   :||| .| |||:  .    |   :| .| ||. | |: .|.|:.
       6 NKVILVGNLGQDPEVRYMPNGGAVANITLATSESWRDKATGEMKEQTEWH 55
                  .         .         .         .
      53 RIKCFGSQAEAHGKYLGKGSLVFVQGKIRNTKY.EKDGQTVYGTDFIAD 100
         |:  ||  ||   .|| ||| |:::|.:|  |: :. ||  | |: : .
      56 RVVLFGKLAEVASEYLRKGSQVYIEGQLRTRKWTDQSGQDRYTTEVVVN 104


Matrix:  blosum62
Gap Penalties:  default
Length:  99
Percent Similarity:  50.515
Percent Identity:  36.082
```

Global (Needleman-Wunch) vs. Local (Smith-Waterman) Alignment

```
  1 ..MSHNQFQFIGNLTRDTEVRHGNSNKPQAIFDIAVNEEWRNDA.GDKQE 47
         |.   :||| .| |||:  .    |   :| .| ||. | |: .|
  1 ASRGVNKVILVGNLGQDPEVRYMPNGGAVANITLATSESWRDKATGEMKE 50

             .         .         .         .         .
 48 RTDFFRIKCFGSQAEAHGKYLGKGSLVFVQGKIRNTKY.EKDGQTVYGTD 96
    .|:. |:  ||  ||   .|| ||| |:::|.:|  |: :. ||  | |:
 51 QTEWHRVVLFGKLAEVASEYLRKGSQVYIEGQLRTRKWTDQSGQDRYTTE 100

             .         .         .         .         .
 97 FIAD...KVDYLDTKAPGGSNQE........................... 116
     : .    .  |  :  ||.
101 VVVNVGGTMQMLGGRQGGGAPAGGNIGGGQPQGGWGQPQQPQGGNQFSGG 150

             .         .
    ...........................

151 AQSRPQQSAPAAPSNEPPMDFDDDIPF 177
```

Dynamic Programming and Optimal Alignment

Dynamic Programming
Solves a problem by breaking the problem into smaller subproblems, which are separately solved, then sequentially reassembled to solve the entire problem. The solution to each subproblem is stored in a table along with a score, and the final answer is arrived at by choosing the sequence of solutions that yields the highest score.

Dynamic programming was invented in 1950 by Richard Bellman at Princeton University, and works well when many solutions are possible but an optimal solution must be found.

Dynamic Programming in Sequence Alignment
This approach was first applied to solving biological sequence alignment problems by Saul Needleman and Christian Wunsch in 1970. It can be used to optimally solve either a global alignment problem (Needleman-Wunsch) or a local alignment problem (Smith-Waterman).

Dynamic Programming Steps

How Dynamic Programming Compares Sequences

1. Create matrix and fill with best scores
There are only three possible choices at each position of the matrix: (a) match the residues present; (b) insert a gap in the top sequence; or (c) insert a gap in the side sequence. The exact score depends on the substitution, gap creation and gap insertion values chosen. The best score of each choice is selected, and a pointer to the preceding position used to arrive at the score is stored with the score.

2. Find highest score
For global alignment (Needleman-Wunsch), the highest score in the final row and final column is used. For local alignment (Smith-Waterman), the highest score anywhere in the matrix is used.

3. Trace pointers back to start and generate alignment
The sequence alignment is created in reverse order, by tracing the pointer back from the highest score to the previous highest score, until either the very start (global) or when it reaches a starting score of 0 (local).

Dynamic Programming Illustrated

		G	C	T	G	G	A	A	G	G	C	A	T
	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	0	0	5	5	0	0	5	5	0	0	0
C	0	0	10	3	0	1	1	0	0	1	10	3	0
A	0	0	3	6	0	0	0	6	.	0	3	15	8
G	0	5	0	0	11	5	0	2	11	5	0	8	11
A	0	0	1	0	4	7	10	5	4	7	1	5	4
G	0	5	0	0	5	9	3	6	10	9	3	0	1
C	0	0	10	3	0	2	5	0	3	6	14	7	0
A	0	0	3	6	0	0	7	10	3	0	7	19	12
C	0	0	5	0	2	0	0	3	6	0	5	12	15
T	0	0	0	10	3	0	0	0	0	2	0	5	17

DYNAMIC PROGRAMMING STEPS

1. Create matrix and fill with best scores (keeping pointers)
2. Find highest score (global in final row and column; local anywhere)
3. Trace pointers back to start and generate sequence alignment

Scores are from a local dynamic programming example in Gibas & Jambeck

```
G C T G G A A G G C A . T
| |   |     |   | | |   |
G C A G . . A . G C A C T
```

GLOBAL

Needleman-Wunsch (GCG Gap)

Starts at final row and column in lower right (17), retraces path

```
G A A G . G C A
|   | |   | | |
G C A G A G C A
```

LOCAL

Smith-Waterman (GCG BestFit)

Starts at highest score in matrix (19), retraces path

Heuristic Sequence Comparison

Heuristic Algorithms
Solve a problem by using rules of thumb to reach a solution. The solution is not guaranteed to be an optimal solution, but is generally arrived at far faster than using an optimal solution approach such as dynamic programming.

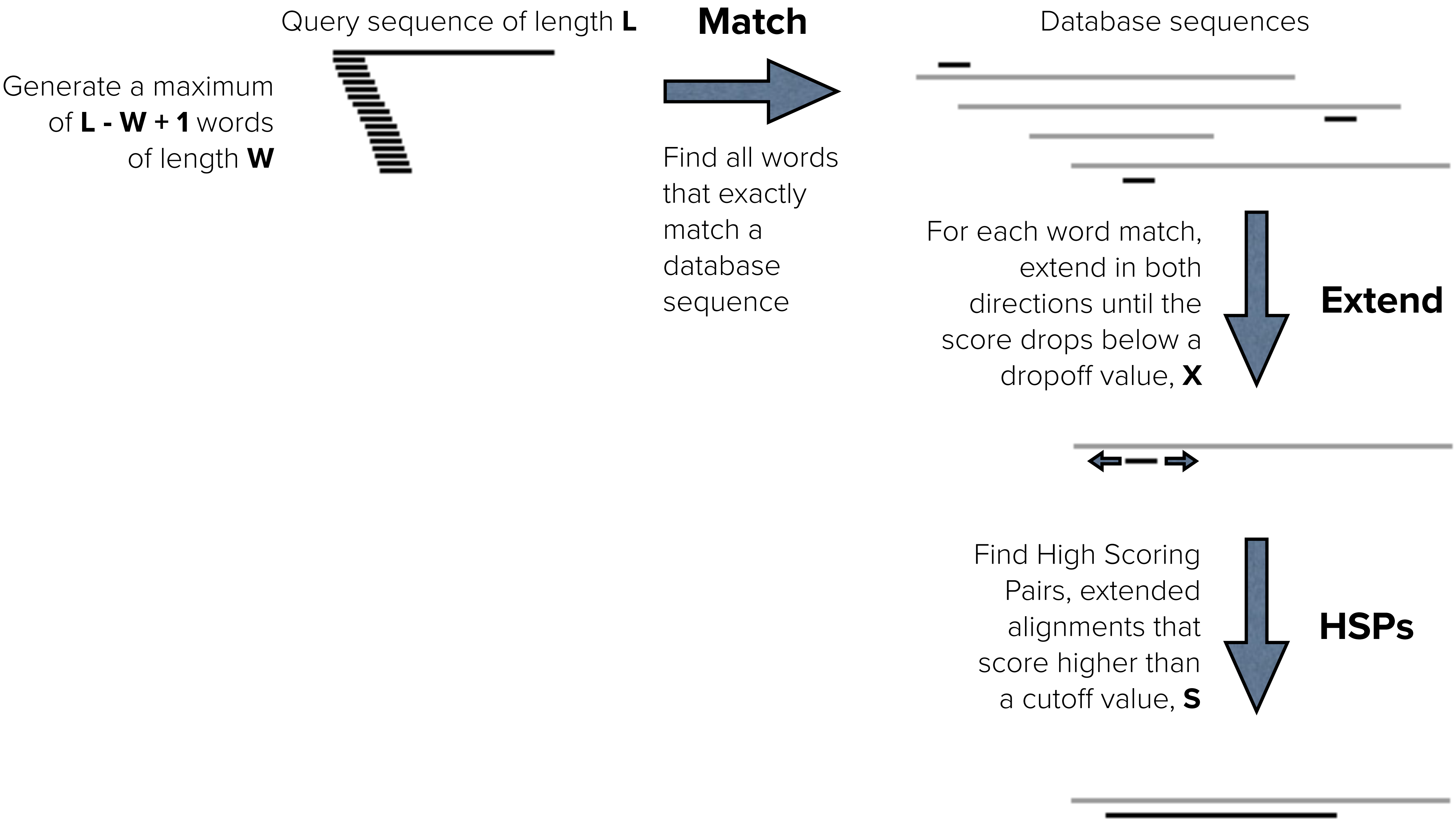
Heuristic Sequence Comparison Algorithms
In sequence comparison, commonly used heuristic approaches include the BLAT algorithm, developed by Jim Kent in 2002, the BLAST algorithm, developed by Stephen Altschul in 1990, and the FASTA algorithm, developed by William Pearson in 1988. The best known of these is the BLAST algorithm.

BLAST Heuristic Algorithm

How BLAST Works in Nucleotide Sequence Comparison

1. Make a list of 11 letter words (default DNA word length (W) = 11) in the DNA sequence. In a query sequence of length L, the maximum number of words will be L - W + 1 (L - 10 for the default word length of 11).
2. Search with each word for exact matches to words in a pre-indexed database of sequences.
3. For each match found, keep extending the alignment in each direction along the sequence, allowing small gaps, until the score drops below a dropoff value (X), to create high scoring segment pairs (HSPs).
4. Select HSPs that score above the HSP cutoff score (S).
5. Determine the statistical significance of each selected HSP score.
6. Use the Smith-Waterman algorithm to generate a local sequence alignment for each selected HSP.

Nucleotide BLAST Illustrated



Query sequence of length **L**

Generate a maximum of **L - W + 1** words of length **W**

Match

Find all words that exactly match a database sequence

Database sequences

For each word match, extend in both directions until the score drops below a dropoff value, **X**

Extend

Find High Scoring Pairs, extended alignments that score higher than a cutoff value, **S**

HSPs

Nucleotide BLAST

Choose Search Set
You will generally be running searches against the nr (nonredundant) database. You may often need to choose another database, limit a search by Organism or Entrez query, or exclude Uncultured/environmental samples.

blastn
Compares a nucleotide query sequence against a nucleotide sequence database. It is best used to find closely related DNA sequences (such as those with over 80% identity) to analyze non-coding DNA or highly conserved DNA regions. It is not well suited to doing other kinds of comparisons.

megablast and discontiguous megablast
This uses a variant of the BLAST algorithm called Mega BLAST which is optimized for quickly comparing nearly identical nucleotide sequences (over 90% identity). It is fastest with larger word sizes (16 and up). The discontiguous version of Mega BLAST is optimized to quickly compare more divergent nucleotide sequences (80% and below identity) and should give better results than Mega BLAST or BLAST for sequence comparisons involving distantly related organisms.

blastx
Compares the six-frame translation of a nucleotide query sequence against a protein sequence database. It is best used to find potential translation products of an unknown nucleotide sequence. It works well if you are unsure of the quality of your sequence data, as it can identify a coding region, despite sequencing induced errors such as frameshifts.

tblastx
Compares the six-frame translation of a nucleotide query sequence against the six-frame translation of a nucleotide sequence database. It is useful for discovering new proteins and ESTs, but is far more computationally intensive.

BLAST Parameters

Algorithm parameters

General Parameters

Decreasing the default word size (e.g. to **7** from the default of **11** for blastn) may result in a more sensitive search and find shorter regions of similarity, but will increase the search time. Decreasing the word size is particularly helpful with shorter query sequences. Increasing the word size will find longer regions of similarity and decrease the search time.

The default Expect threshold (E value) is **0.05**, which means that matches with only a less than 5% chance of appearing by chance alone are displayed. You can decrease the value to get only more significant results, or increase it to **1 or more** when searching with short queries, which are more likely to be found by chance in a given database.

Scoring Parameters

The Match/Mismatch Scores can be adjusted: a 1/-3 setting is best for 99% conserved sequences, a 1/-2 setting is best for 95% conserved sequences, and a 1/-1 setting is best for for 75% conserved sequences. The scoring parameters, including gap costs, are automatically adjusted to appropriate defaults for the algorithm used (blastn, megablast, etc.) or for short query sequences (no filter, word size 7, expect 1000), which can also be used when looking for short, nearly exact matches.

Filters and Masking

Unless you are working with a query sequence that contains repetitive residues or has a biased composition, it is best to turn filtering off, particularly when working with shorter query sequences. Masking of segments in the query sequence is indicated by X's, resulting in "**XXXXX...**" marked regions.

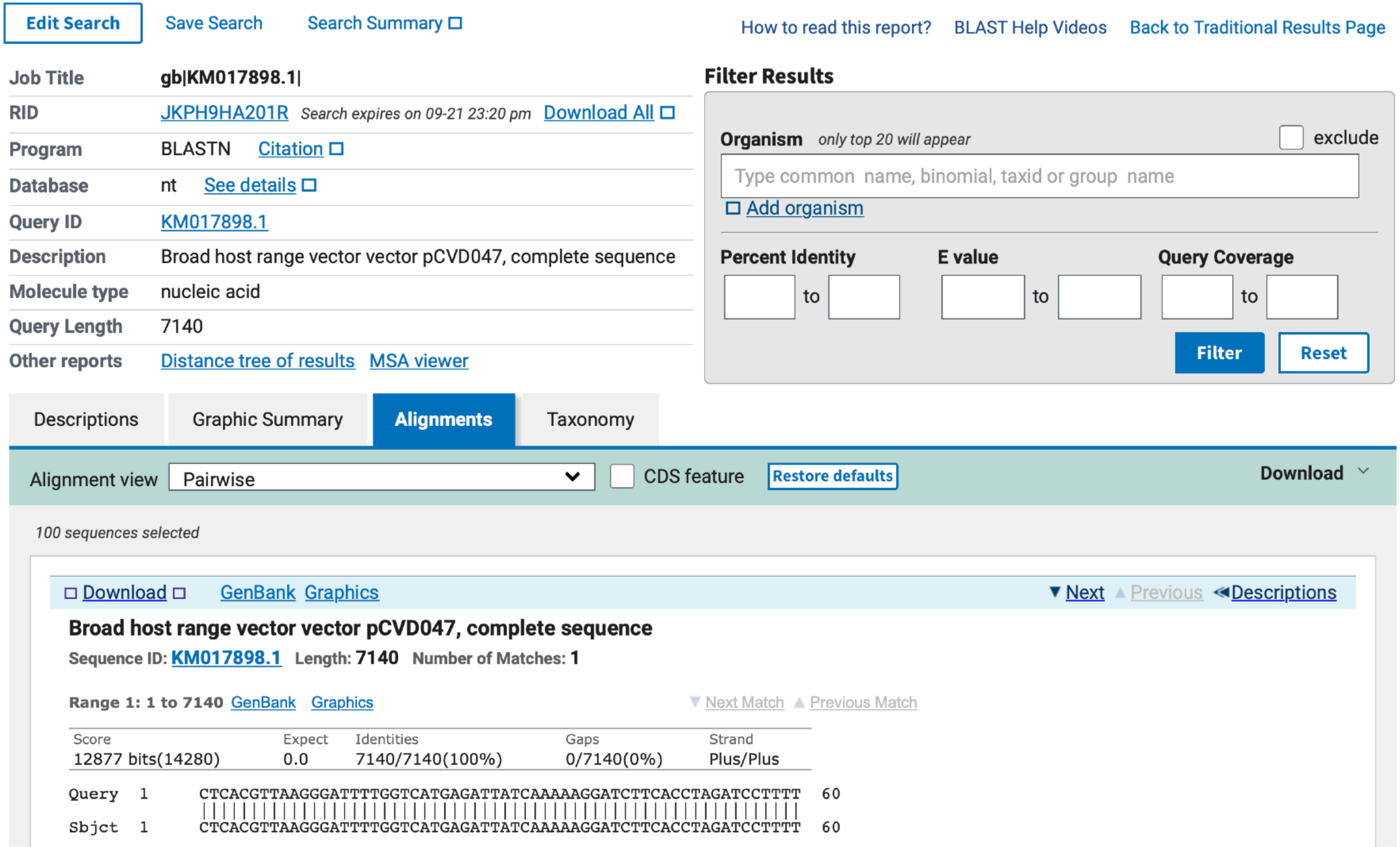
The **Low complexity regions** filter uses the DUST (blastn) or SEG (blastp) programs to filter low complexity regions.

The **Species-specifics repeats** filter in blastn masks repeat sequences such as LINE's, SINE's, and retroviral repeats.

The **Mask for lookup table only** option uses selected filters only for the initial match, but not the subsequent extension.

The **Mask lower case letters** option allows you to manually select which residues to mask by making them lower case, e.g. transmembrane or coiled-coil regions.

BLAST Alignment Example



Job Title	**gb|KM017898.1|**
RID	JKPH9HA201R *Search expires on 09-21 23:20 pm* Download All □
Program	BLASTN Citation □
Database	nt See details □
Query ID	KM017898.1
Description	Broad host range vector vector pCVD047, complete sequence
Molecule type	nucleic acid
Query Length	7140
Other reports	Distance tree of results MSA viewer

Filter Results

Organism *only top 20 will appear* □ exclude

Type common name, binomial, taxid or group name

□ Add organism

Percent Identity to **E value** to **Query Coverage** to

100 sequences selected

Broad host range vector vector pCVD047, complete sequence

Sequence ID: KM017898.1 **Length:** 7140 **Number of Matches:** 1

Range 1: 1 to 7140 GenBank Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Identities	Gaps	Strand
12877 bits(14280)	0.0	7140/7140(100%)	0/7140(0%)	Plus/Plus

```
Query  1      CTCACGTTAAGGGATTTTGGTCATGAGATTATCAAAAAGGATCTTCACCTAGATCCTTTT  60
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  1      CTCACGTTAAGGGATTTTGGTCATGAGATTATCAAAAAGGATCTTCACCTAGATCCTTTT  60
```

BLAST Results

Formatting options
You can view descriptions, a graphical summary, alignments or taxonomy. You can filter your results by Organism, a Percent identity range, or an E value range. The Search Summary gives a useful overview of the search parameters.

Score (bits)
A statistical measure of the significance of the alignment, measured in bits. The higher the score, the more likely for the alignment to be significant. **A score of 50 bits or higher is likely to be significant.**

Expect value
The expectation or E value. An estimate of the number of times the match may have occurred by chance. The lower the value, the more likely for the alignment to be significant. **An E value below 0.0001 (1e-4) is likely significant.**

Identities
The fraction of identical residues in the final alignment. The higher the identity, the more likely for the alignment to be significant. **A 70% identity for nucleotide sequences (with over 100 nucleotides of alignment) is likely to be significant, a 30% identity for protein sequences along their entire length is likely to be significant.**

Length
The length of the final alignment. The longer the length, the more likely for the alignment to be significant. Very short alignments (e.g. 10 nucleotides) may have a very high percent identity or very low E value, yet not be significant. **Long alignments (i.e. 100 nucleotides and above) with high scores, high identities or low E values are likely significant.**

- Remember that BLAST hits are not transitive, unless the alignments overlap, since BLAST alignment is fundamentally a local alignment. Thus, if query **A** results in significant hits to **B** and **C**, **B** and **C** are not necessarily similar to each other (**A** might contain domain **1** and **2**, **B** might contain only domain **1**, and **C** might contain only domain **2**). It is thus wise to carefully check the length and extent of any alignment.

Protein BLAST

BLAST Protein Sequence Comparison

- A variety of BLAST programs are featured by NCBI for protein-protein comparison, including blastp (protein vs. protein), PSI-BLAST (position specific iterated), PHI-BLAST (pattern hit initiated), DELTA-BLAST (incorporates a Conserved Domain Database search) and tblastn (protein vs. translated database).
- The default word size for protein BLAST searches is 3, this can be changed to 2 for more stringent, but slower searches.
- The choice of Dayhoff substitution matrix can be important. The default matrix for NCBI BLAST protein comparison is BLOSUM62, which is optimized for long query sequences (over 85 aa) and known close homologies. When searching with short query sequences or distant homologies, be sure to try other matrices.

Protein BLAST Results Rules of Thumb

- Proteins that are more than 30% identical throughout their entire lengths are likely homologous.
- Proteins that are 20 to 30% identical throughout their entire lengths may or may not be homologous (the “gray zone”).
- Proteins that are less than 20% identical throughout their entire lengths are not likely homologous.
- Matches that are more than 50% identical in a 20 to 40 amino acid region occur frequently by chance.

PubMed and EndNote

PubMed

https://www.ncbi.nlm.nih.gov/pubmed
https://www.nlm.nih.gov/bsd/disted/pubmedtutorial/
https://www.ncbi.nlm.nih.gov/books

EndNote and PubMed

EndNote can act as a database client to directly connect to the PubMed database and search and retrieve references from it: **Tools > Online Search... > Favorites > PubMed (NLM)**
Alternatively, the PubMed Clipboard can be used to collect references, export them as a text file in MEDLINE format, then import them into EndNote using the **PubMed (NLM)** import filter.

EndNote and PDF Files

EndNote can also be used to organize PDF files, which are otherwise easy to lose track of. With a reference selected or open, clicking on the paperclip, or **References > File Attachments > Attach File...** inserts a link to the PDF file you select. Use **References > File Attachments > Open with Preview** to open the linked PDF in Preview (on a Mac) or double click the reference to view the PDF in EndNote. EndNote also offers the ability to import folders of PDFs, share libraries with other researchers, and back them up to the web. It is available free to students at **library.columbia.edu/services/citation-management.html**

Other Options

Zotero and Mendelay are free options for reference management available at the above library link. Papers, available at a discount to students at **https://www.papersapp.com** is a commercial option for importing PubMed references and managing references and PDFs.

Databases

Flat File Database (FFDB)

A collection of similar files made useful by ordering and indexing. All the information about one sequence would be stored in one structured text file, and you generally examine one file at a time.

Examples: GenBank, Excel, older versions of FileMaker or Access

Relational Database Management System (RDBMS)

All data is stored inside one or more tables of rows and column, with all operations done on the tables themselves or producing other tables as the result. All the information about one sequence would be stored in a collection of tables with other data, so you can easily look at just the information relating to that sequence, or how it relates to the database as a whole. Structured Query Language (SQL) is used to access data in a relational database.

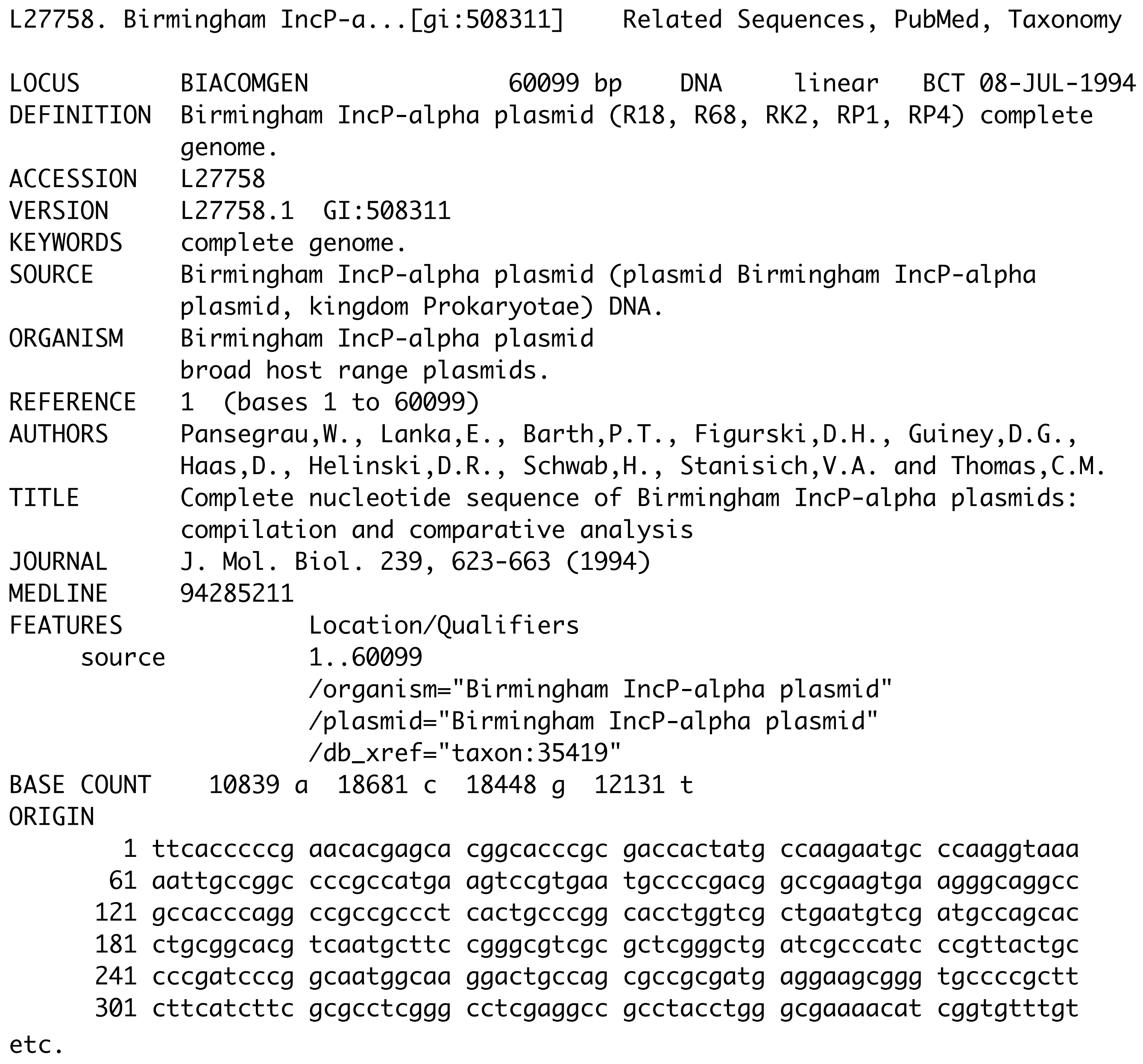
Examples: MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Oracle

Not Only SQL (NoSQL)

Data is stored and retrieved in a different manner than tabular relations, the details varying from one to the other. They are often used for big data or real time applications, and their data structures give them certain advantages and disadvantages handling particular data compared to an RDBMS.

Examples: Amazon DynamoDB (key-value) and SimpleDB (eventual consistency), Apache CouchDB (document) and HBase (column), Dynamo (key-value), MongoDB (document), OrientDB (graph)

Flat File



```
L27758. Birmingham IncP-a...[gi:508311]    Related Sequences, PubMed, Taxonomy

LOCUS       BIACOMGEN              60099 bp    DNA     linear   BCT 08-JUL-1994
DEFINITION  Birmingham IncP-alpha plasmid (R18, R68, RK2, RP1, RP4) complete
            genome.
ACCESSION   L27758
VERSION     L27758.1  GI:508311
KEYWORDS    complete genome.
SOURCE      Birmingham IncP-alpha plasmid (plasmid Birmingham IncP-alpha
            plasmid, kingdom Prokaryotae) DNA.
ORGANISM    Birmingham IncP-alpha plasmid
            broad host range plasmids.
REFERENCE   1  (bases 1 to 60099)
AUTHORS     Pansegrau,W., Lanka,E., Barth,P.T., Figurski,D.H., Guiney,D.G.,
            Haas,D., Helinski,D.R., Schwab,H., Stanisich,V.A. and Thomas,C.M.
TITLE       Complete nucleotide sequence of Birmingham IncP-alpha plasmids:
            compilation and comparative analysis
JOURNAL     J. Mol. Biol. 239, 623-663 (1994)
MEDLINE     94285211
FEATURES             Location/Qualifiers
     source          1..60099
                     /organism="Birmingham IncP-alpha plasmid"
                     /plasmid="Birmingham IncP-alpha plasmid"
                     /db_xref="taxon:35419"
BASE COUNT    10839 a  18681 c  18448 g  12131 t
ORIGIN
        1 ttcacccccg aacacgagca cggcacccgc gaccactatg ccaagaatgc ccaaggtaaa
       61 aattgccggc cccgccatga agtccgtgaa tgccccgacg gccgaagtga agggcaggcc
      121 gccacccagg ccgccgccct cactgcccgg cacctggtcg ctgaatgtcg atgccagcac
      181 ctgcggcacg tcaatgcttc cgggcgtcgc gctcgggctg atcgcccatc ccgttactgc
      241 cccgatcccg gcaatggcaa ggactgccag cgccgcgatg aggaagcggg tgccccgctt
      301 cttcatcttc gcgcctcggg cctcgaggcc gcctacctgg gcgaaaacat cggtgtttgt
etc.
```

- Data type not distinct from record.
- Tremendous duplication of data.
- Mixed hierarchical and non-hierarchical data.
- Difficult to query in a sophisticated manner.
- Difficult to link to other data.

Relational Database

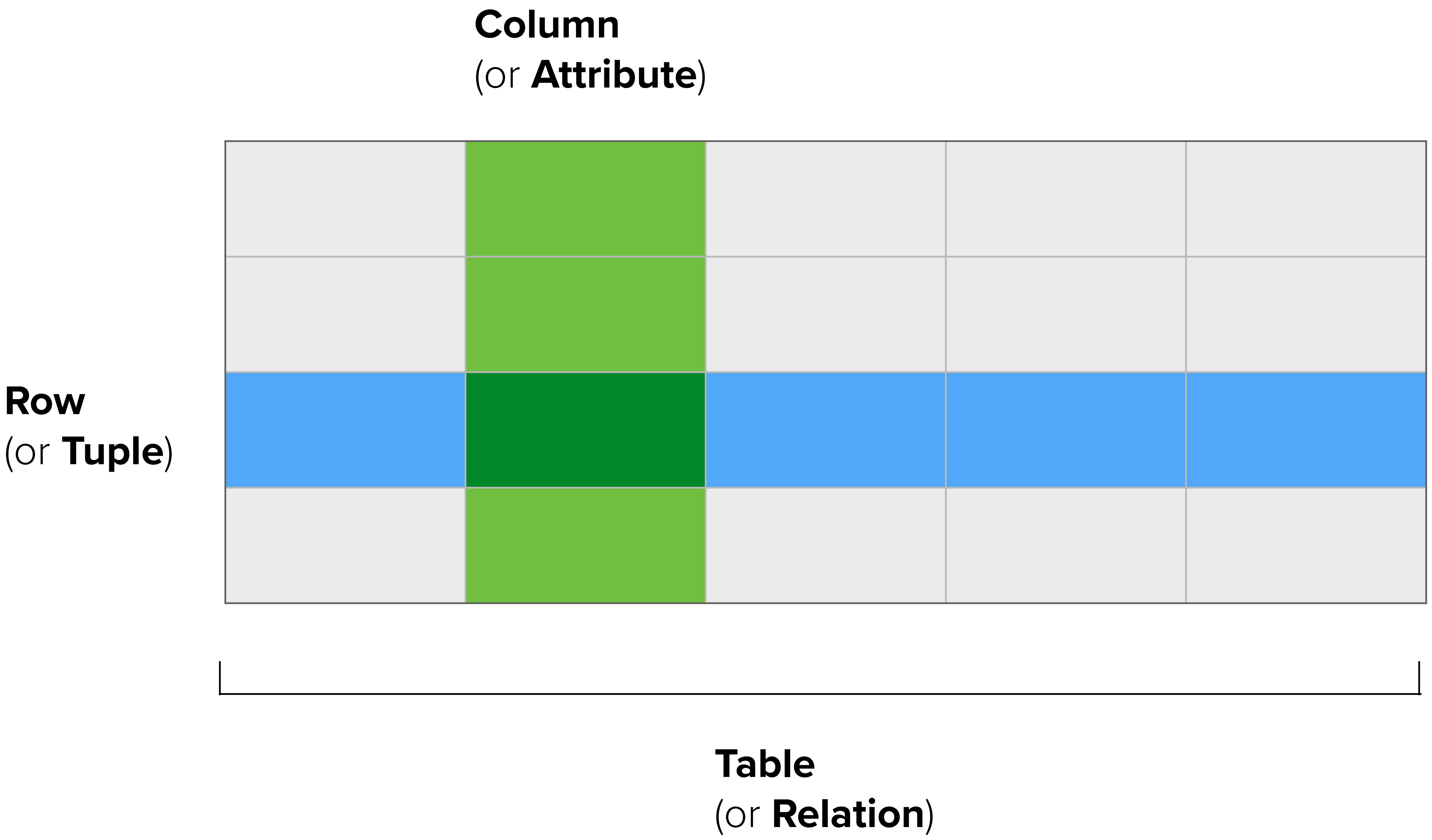
What distinguishes a relational database from a flat file or spreadsheet is the ability to use relational algebra to create sophisticated queries that provide many alternative views of the data or subsets of the data. The language used to do this is called Structured Query Language, or SQL, a declarative language for querying data in which you describe what you want to see.

- A relational database contains one or more tables.
- Tables are known as relations, and are identified by a unique name.
- A row is also known as a tuple, or record. No rows are identical. They are in no particular order.
- A column is also known as an attribute, or field. Each column is identified by a name and contains only one type of data, e.g. an integer or text.

Relational Databases in Biology

Using a relational database with biological data allows you to ask questions that would be difficult or impossible with a flat file format, add functional annotation and easily work with subsets of data to improve search strategy and sensitivity.

Relational Database Terminology



Column
(or **Attribute**)

Row
(or **Tuple**)

Table
(or **Relation**)

Structured Query Language

Structured Query Language, or SQL was developed in the 1970s at IBM by Donald Chamberlin and Raymond Boyce, and is now the standard for interacting with relational databases. It is a specialized programming language for managing data in a relational database management system, or RDBMS, with declarative and procedural features. It handles both data definition and data manipulation tasks, and supports relational algebra and calculus. SQL became an ANSI (American National Standards Institute) standard in 1978.

Note that SQL is not case sensitive, but SQL commands are often written in all caps. Some implementations require a semicolon at the end of any SQL statement, others only when chaining multiple statements together. Whitespace is generally ignored. An asterisk (*) is used to specify all columns in a table.

Undefined data is considered to have a value of null, and two null values cannot be logically compared.

Example:

```
SELECT * FROM data
```

SQL Data Manipulation Language

`CREATE DATABASE` creates a new database
`ALTER DATABASE` modifies an existing database
`CREATE TABLE` creates a new table
`ALTER TABLE` modifies an existing table
`DROP TABLE` deletes a table
`CREATE INDEX` creates an index
`DROP INDEX` deletes an index
`UPDATE` updates existing data using `SET` and `WHERE`
`DELETE` deletes data from a database
`INSERT INTO` inserts new data into a database

Each column in a table is expected to have a name and a specified data type. Common data types include `TEXT, CHAR(n), VARCHAR(max), INTEGER, FLOAT, BOOLEAN, DATE, TIME` and `BLOB.`

Example:

```
CREATE TABLE protein (
protein_id INTEGER PRIMARY KEY AUTOINCREMENT
sequence TEXT
length INTEGER)
```

SQL Queries

`SELECT` extracts a column from a database
`SELECT DISTINCT` extracts distinct values in a column from a database

Clauses

`FROM` specifies one or more tables to retrive data from
`WHERE` filters data that does not meet the specified criteria
`ORDER BY` specifies the order in which rows appear in the results
`INNER JOIN` returns all rows from two or more tables where there is at least one match in each table (other forms exist)

`UNION` combines the results of two or more `SELECT` statements
`RESTRICT` remove rows that do not meet the specified criteria
`PRODUCT` combines rows from two or more tables in all possible ways

Example:

```
SELECT protein_id FROM protein
```

SQL Functions and Operators

Functions

```
AVG()
COUNT()
COUNT (DISTINCT )
FIRST()
FORMAT()
GROUP BY ()
ORDER BY ()
LAST()
MAX()
MIN()
MIN()
SUM()
```

Operators

SQL also supports many standard mathematical operators, such as `=. >, >=. <, <=, BETWEEN, LIKE, IN, IS, IS NOT,` etc. `%` is a wildcard that matches one or more characters and `_` is a wildcard that matches any one character.

Example:

```
SELECT * FROM protein WHERE length BETWEEN 51 AND 100
```

SQL Optimization

Normalization

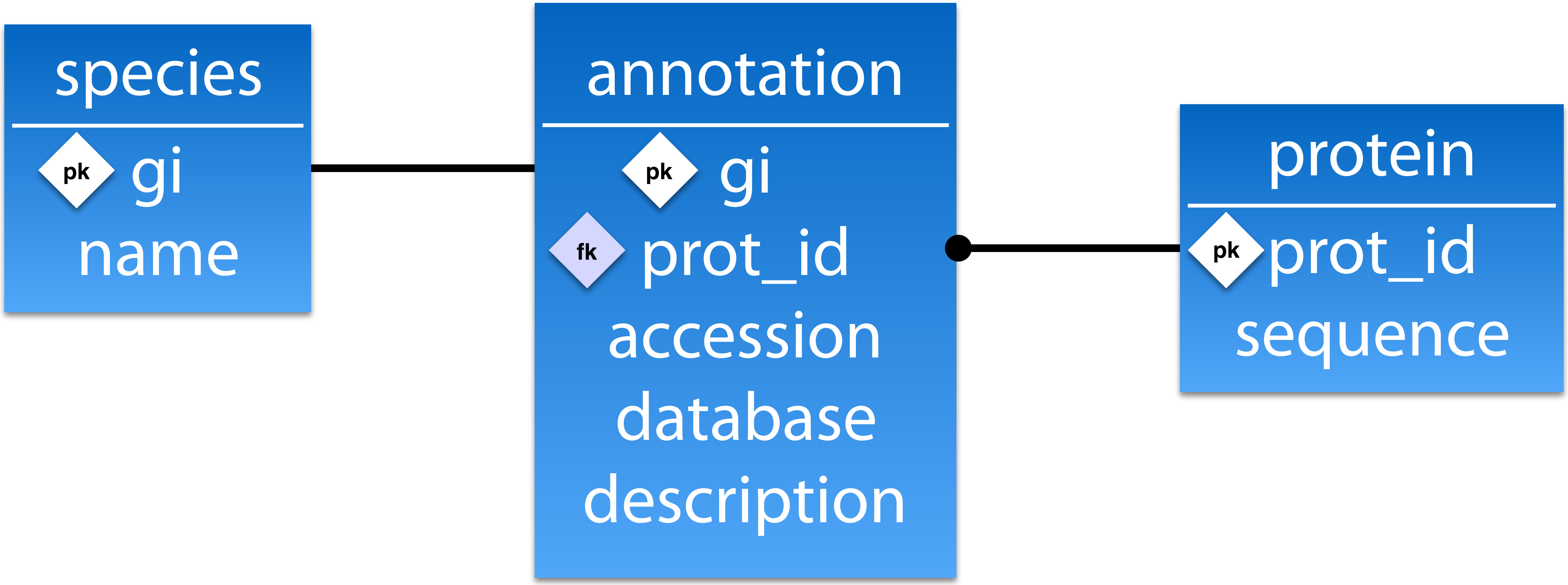
Normalization seeks to reduce redundancy, which increases storage efficiency, data consistency and scalability. This involves splitting large tables into smaller, less redundant tables, and defining their relationships such that a change of a field can be made in a single table and then propagated through the database.

For example, a unique primary key in a table can be helpful, but although a name that appears unique to an organism might appear to make a good primary key, it might later need to be changed or modified, which would require it to be changed throughout the entire database. A better primary key would be a unique number that never changes, stored in a table that also contains the organism name. The name would only have to be changed in that one table. Primary keys from one table used to provide a link to another table are called foreign keys.

Indexing

Indexing is a technique used to speed searching in indexed columns, it can greatly increase the speed of certain queries, but indexes will slow writing data to those columns and will occupy additional space. They are particularly useful when dealing with large tables and frequently queried columns.

Normalized Tables



species
pk gi
name

annotation
pk gi
fk prot_id
accession
database
description

protein
pk prot_id
sequence

Cloud Databases

Amazon SimpleDB

Simple non relational data store. Limited to 10 GB of data per table and 25 writes per second. The first 25 SimpleDB Machine Hours and 1 GB of storage are free each month.

Amazon RDS

Relational Database Service. On demand database instances with MySQL, PostgreSQL, MariaDB, Oracle, Microsoft SQL Server or Amazon Aurora preinstalled. Cost depends on the database and server speed.

Amazon DynamoDB

NoSQL database service. Cost based on reserved performance capacity plus a fee for data storage.

Amazon Redshift

Petabyte scale data warehousing capable of massive parallel processing of very large datasets. Cost depends on server speed and type of node (Dense Compute or Dense Storage). Backups and data transfer are free.

Google Cloud SQL

MySQL database service residing in Google's cloud. Cost based on server performance.

Database References

Practical Computing for Biologists
by Steven Haddock and Casey Dunn

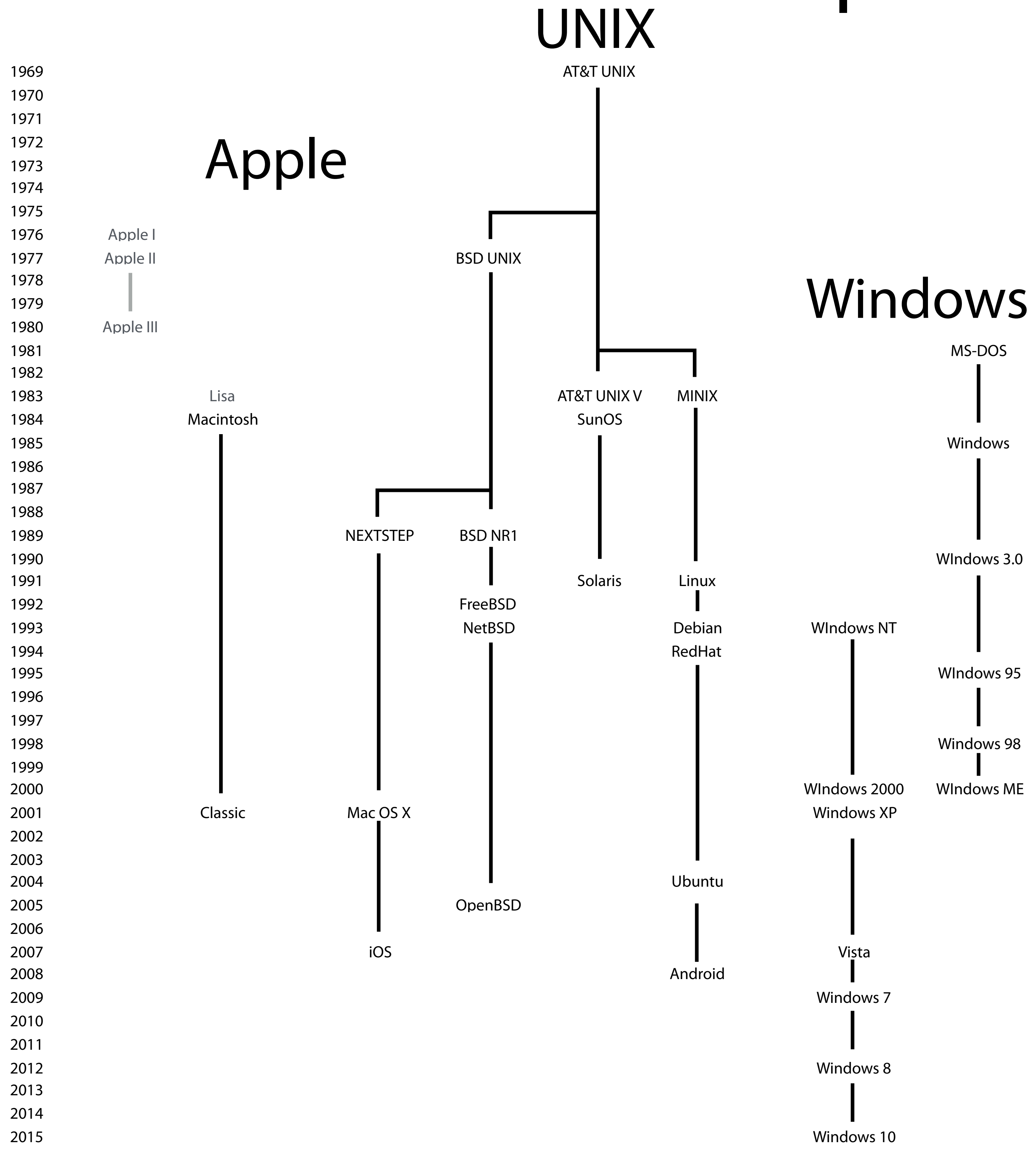
Databases Demystified, Second Edition
by Andrew Oppel

MySQL, Fifth Edition
by Paul Dubois

Databases in Depth: Relational Theory for Practitioners
by C.J. Date

Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design, Third Edition
by Michael Hernandez

Evolution of Modern Operating Systems



UNIX

Apple

Windows

1969 AT&T UNIX
1970
1971
1972
1973
1974
1975
1976 Apple I
1977 Apple II BSD UNIX
1978
1979
1980 Apple III
1981 MS-DOS
1982
1983 Lisa AT&T UNIX V MINIX
1984 Macintosh SunOS
1985 Windows
1986
1987
1988
1989 NEXTSTEP BSD NR1
1990 WIndows 3.0
1991 Solaris Linux
1992 FreeBSD
1993 NetBSD Debian WIndows NT
1994 RedHat
1995 WIndows 95
1996
1997
1998 Windows 98
1999
2000 WIndows 2000 WIndows ME
2001 Classic Mac OS X Windows XP
2002
2003
2004 Ubuntu
2005 OpenBSD
2006
2007 iOS Vista
2008 Android
2009 Windows 7
2010
2011
2012 Windows 8
2013
2014
2015 Windows 10

Unix and Computational Biology

- Historically, Unix has been used as a free academic and research operating system (BSD, FreeBSD, etc.), and many forms of Unix and programs that run on Unix are Open Source, available for free with source code.
- Unix is stable, efficient and easy to program, with many powerful scripting, automation and programming tools built in.
- New algorithms in computational biology are generally first implemented in Unix.
- All the bioinformatics tools needed to do complex analysis are available for free on Unix (BLAST, FASTA, CLUSTAL, PHYLIP, PHRED, PHRAP, CONSED, EMBOSS, HISAT, Lighter, Bowtie, etc.).
- Unix operating systems generally use a command line environment known as a shell to interact with them. A shell interprets and executes the commands you give it, and can also run text files called shell scripts that allow for commands to be strung together, manipulated, and automated. The default shell for OS X was **bash**, it is now **zsh**.
- Some Unix operating systems, particularly OS X, Linux and Ubuntu, have good GUIs as well (notably, Linux Mint/Cinnamon and elementaryOS/Pantheon).

Terminal and the Unix Command Line

Terminal

Terminal, located in **/Applications/Utilities**, is the application which gives an OS X user command line shell access to the underlying Unix operating system. One can drag a folder or application to the Terminal window to get its pathname, which is often required when issuing Unix commands.

`ls` (list)

Lists the current directory's contents. Adding the `-a` option (`ls -a`) lists all contents, including what is normally invisible (file or directory names starting with a period, e.g. **.bash_profile**, are normally invisible). Adding the `-l` option (`ls -l`) lists long information about files: type, permissions, links, owner, group, size, modification date & time and name. The wild card character (`*`) is often useful in arguments for this command, e.g. `ls *.doc` will list all Word files with that extension in a directory.

`cd` (change directory)

By itself, `cd` takes you to your home directory. Using an argument of two periods, i.e. `cd ..`, moves you to the directory directly above the current directory, while `cd /` moves you to the root directory. If you get lost, type `pwd` to print your working directory, that is, list your current directory as a pathname.

`exit`

Type `exit` to logout of a Terminal session.

Regular Expressions

Regular Expressions

A regular expression, also known as a regex, is a sequence of characters that defines a pattern. These patterns can then be used to perform a search, or search and replace. The concept of regular expressions is based on the work of the mathematician Stephan Cole Keene on regular languages and events.

`a`	matches the lowercase letter 'a'
`[a-z]`	matches any lowercase letter from 'a' to 'z'
`[1-9]`	matches any non-zero digit
`[0-9]`	matches any digit
`.`	matches any single character except a line break
`\r`	matches a line break
`\t`	matches a tab

Regular Expressions and grep

grep (Globally search for Regular Expression and Print)

Grep is a powerful Unix text searching utility that is available at both the command line in OS X and in certain applications such as BBEdit. Grep can search the input for lines containing a match to a given pattern, using regular expressions if necessary, then output the lines that matched. Grep can thus greatly automate searching for information in text files. The Unix **grep** utility can be invoked from the Terminal in OS X using the following syntax:

```
grep -[options] 'pattern' filename(s)
```

Useful grep Options

`-c` print count of matching lines, rather than the matching lines themselves
`-i` ignore case distinctions in pattern and file(s)
`-l` print filenames containing matching lines, but not the matching lines

grep Results

grep normally prints a list of every line within the file(s) searched containing a match.

In Terminal, typing `grep 'RNA' sars.txt` will find all lines containing RNA within the **sars.txt** file. To automatically output those lines to a file called **sarsrna.txt**, one would type `grep 'RNA' sars.txt > sarsrna.txt` (in Unix, **>** redirects output).

`grep '>' sequences.fasta` will return the name of every sequence in the **sequences.fasta** file (in a fasta file, the sequence name is on a line beginning with >).

`grep -c '>' sequences.fasta` will only return the number of sequences in the file.