

# Lecture 07

# molecular sequences



Course: Practical Bioinformatics (BIOL 4220)  
Instructor: Michael Landis  
Email: [michael.landis@wustl.edu](mailto:michael.landis@wustl.edu)



# Lecture 07 outline

Last time: shell scripts

This time: molecular sequences

## Topics

- sequence data
- GenBank
- BLAST

A ***molecular sequence*** is a string of characters from a molecular alphabet.

Examples: *DNA, RNA, amino acid sequences*

Sequences are key to understanding:

- disease mechanism
- gene expression
- developmental biology
- heredity and ancestry
- protein and cell function
- biodiversity patterns

sequence length is  
18 base pairs (bp)



...ATGCGACGATGGATACCATAG...



DNA alphabet:  
A, C, G, T

fourth position  
is in state C

...ATGCGACGATGGATACCATAG...



transcription

...AUGCGACGAUGGAAUACCAUAG...

RNA alphabet:  
A, C, G, U

...ATGCGACGATGGATACCATAG...

start  
codon



transcription

stop  
codon

...[AUG]CGACGAUGGAUACCA[UAG]...

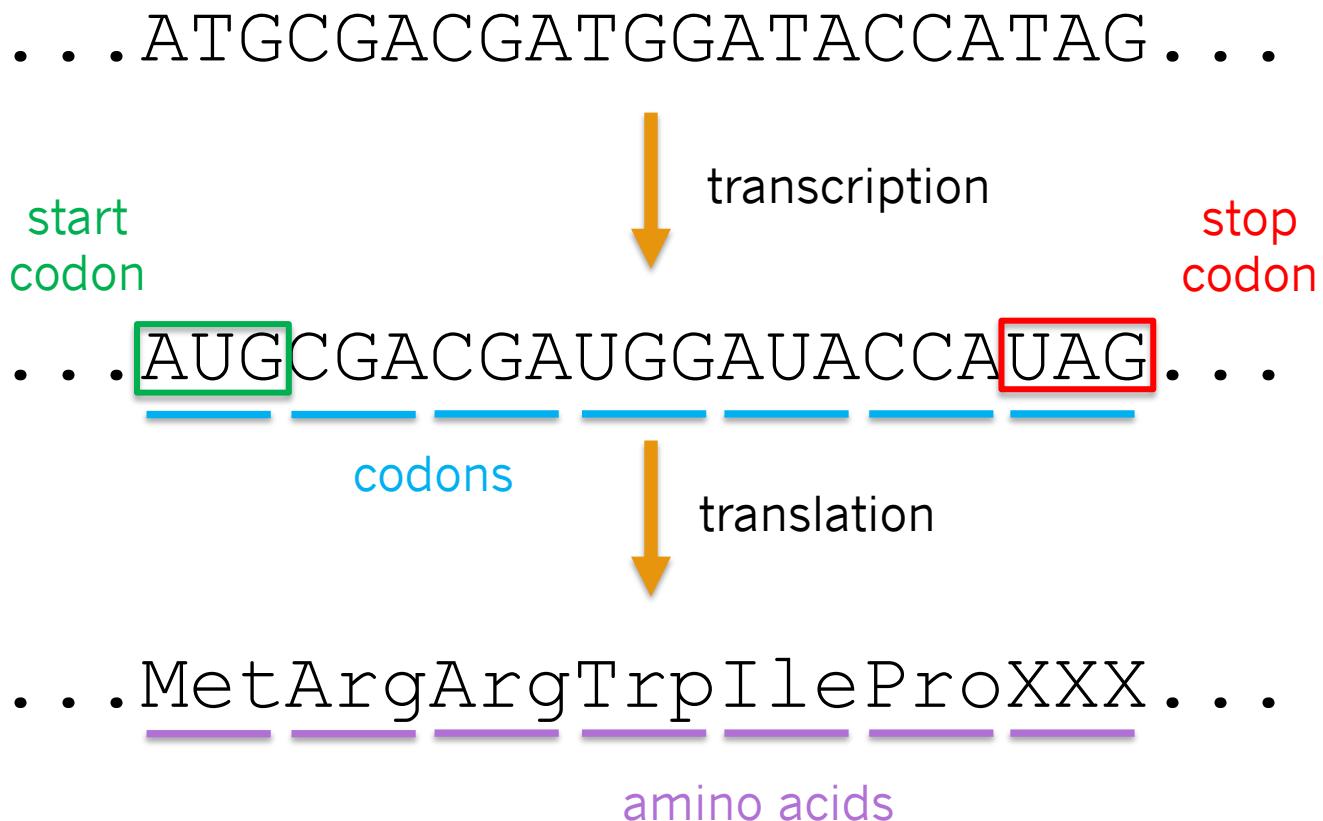
codons

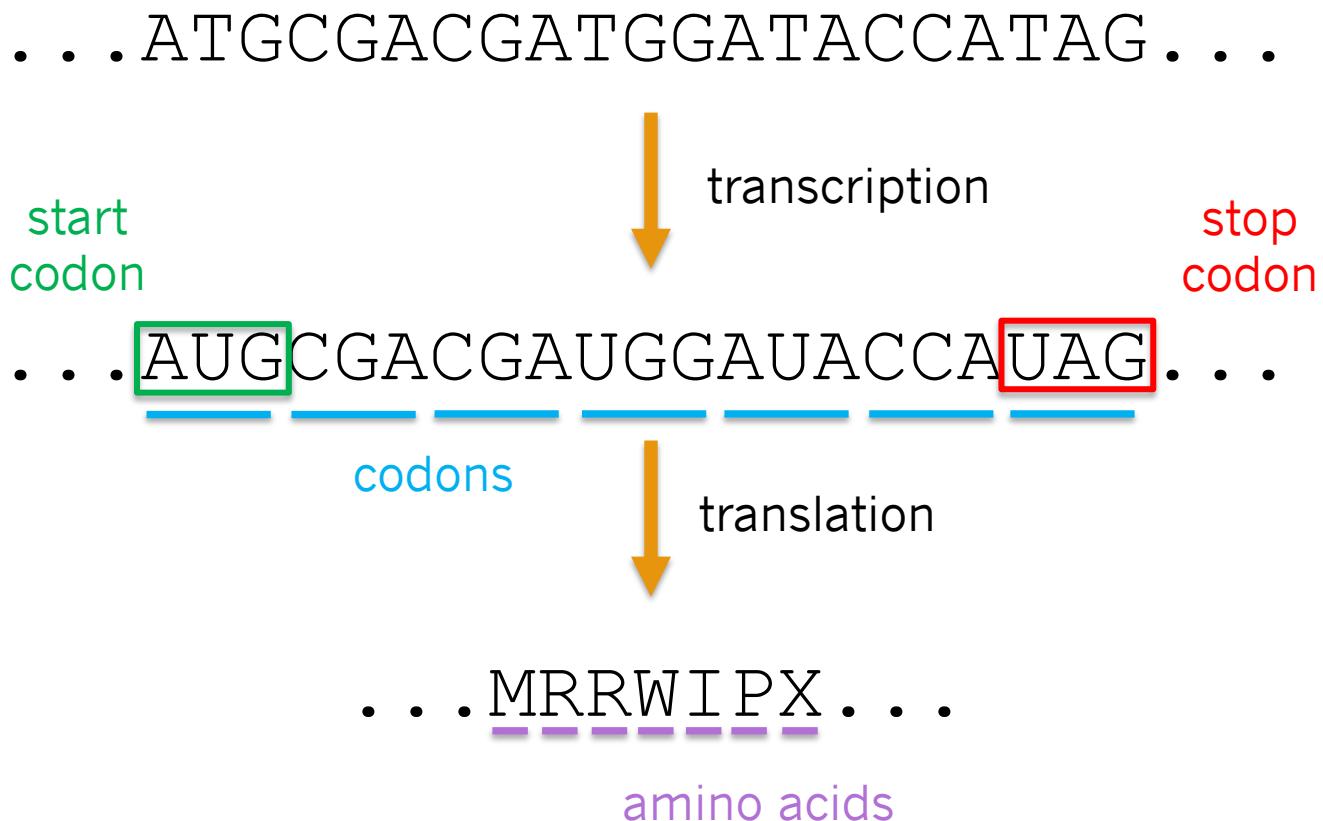
# Standard genetic code

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

# Standard genetic code

		Second letter											
		U	C	A	G								
First letter	U	UUU UUC UUA UUG	UCU UCC UCA UCG	UAU UAC	Tyrosine (Tyr)	UGU UGC	Cysteine (Cys)						
	C	CUU CUC CUA CUG	CCU CCC CCA CCG	CAU CAC CAA CAG	Histidine (His)	CGU CGC CGA CGG	Stop						
	A	AUU AUC AUU AUG	Isoleucine (Ile)	ACU ACC ACA ACG	Threonine (Thr)	AAU AAC	Glutamine (Gln)						
						AAA AAG	Lysine (Lys)						
	G	GUU GUC GUA GUG	Valine (Val)	GCU GCC GCA GCG	Alanine (Ala)	GAU GAC GAA GAG	Aspartic acid (Asp) Glutamic acid (Glu)						
start codon		GUU, GUC, GUA, GUG all encode Valine				AAA and AAG encode Lysine							
stop codons													
U C A G													





# NCBI GenBank

The screenshot shows the NCBI GenBank homepage. At the top, there's a navigation bar with links for NCBI, Resources, How To, and Sign in to NCBI. Below the navigation is a search bar with "GenBank" selected and a dropdown menu set to "Nucleotide". A search button is visible next to the search bar. Underneath the search bar is a horizontal menu with categories: GenBank, Submit, Genomes, WGS, Metagenomes, TPA, TSA, INSDC, Other. A prominent orange banner overlays this menu. The banner contains a black exclamation mark icon, the text "COVID-19 Information", and a close button (an "X"). Below this, there are two rows of links: "Public health information (CDC) | Research information (NIH)" and "SARS-CoV-2 data (NCBI) | Prevention and treatment information (HHS) | Español".

## GenBank Overview

### What is GenBank?

GenBank® is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences ([Nucleic Acids Research, 2013 Jan;41\(D1\):D36-42](#)). GenBank is part of the [International Nucleotide Sequence Database Collaboration](#), which comprises the DNA DataBank of Japan (DDBJ), the European Nucleotide Archive (ENA), and GenBank at NCBI. These three organizations exchange data on a daily basis.

A GenBank release occurs every two months and is available from the [ftp site](#). The [release notes](#) for the current version of GenBank provide detailed information about the release and notifications of upcoming changes to GenBank. Release notes for [previous GenBank releases](#) are also available. GenBank growth [statistics](#) for both the traditional GenBank divisions and the WGS division are available from each release.

An [annotated sample GenBank record](#) for a *Saccharomyces cerevisiae* gene demonstrates many of the features of the GenBank flat file format.

### GenBank Resources

[GenBank Home](#)

[Submission Types](#)

[Submission Tools](#)

[Search GenBank](#)

[Update GenBank Records](#)

<https://www.ncbi.nlm.nih.gov/genbank/>

# GenBank sequences

GenBank publicly hosts (Aug '24)

- 251,998,350 sequences
- 3,675,462,701,077 bases

NCBI sequences are used extensively

- identifying anonymous sequences
- inferring gene function
- searching for drug targets
- expanding datasets
- met-analyses

# Homo sapiens isolate C15542T cytochrome b (cytb) gene, partial cds; mitochondrial

GenBank: JN034136.1

[FASTA](#) [Graphics](#) [PopSet](#)

---

Go to:

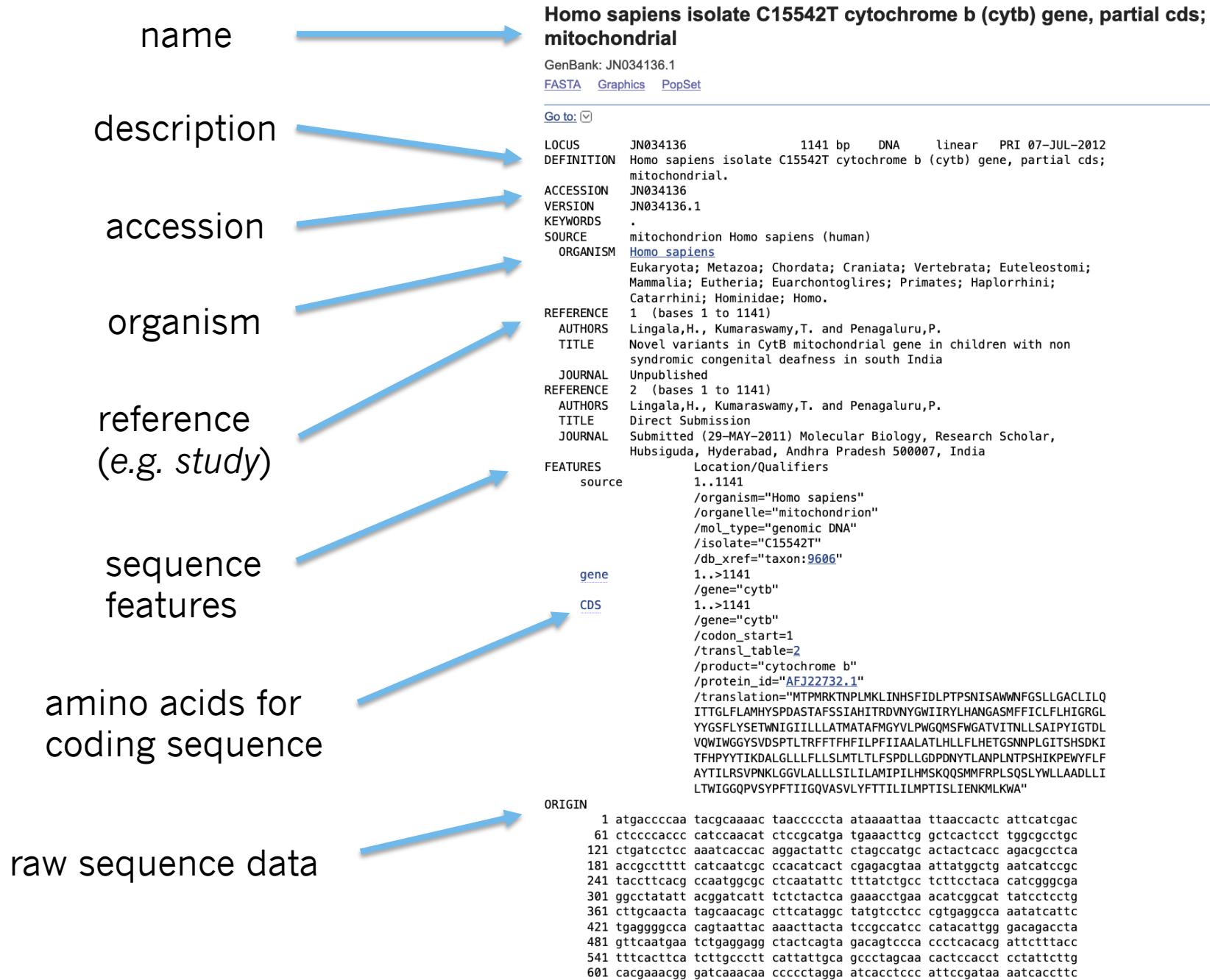
LOCUS JN034136 1141 bp DNA linear PRI 07-JUL-2012  
DEFINITION Homo sapiens isolate C15542T cytochrome b (cytb) gene, partial cds; mitochondrial.  
ACCESSION JN034136  
VERSION JN034136.1  
KEYWORDS .  
SOURCE mitochondrion Homo sapiens (human)  
ORGANISM [Homo sapiens](#)  
Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;  
Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;  
Catarrhini; Hominidae; Homo.  
REFERENCE 1 (bases 1 to 1141)  
AUTHORS Lingala,H., Kumaraswamy,T. and Penagaluru,P.  
TITLE Novel variants in CytB mitochondrial gene in children with non syndromic congenital deafness in south India  
JOURNAL Unpublished  
REFERENCE 2 (bases 1 to 1141)  
AUTHORS Lingala,H., Kumaraswamy,T. and Penagaluru,P.  
TITLE Direct Submission  
JOURNAL Submitted (29-MAY-2011) Molecular Biology, Research Scholar,  
Hubsiguda, Hyderabad, Andhra Pradesh 500007, India

FEATURES	Location/Qualifiers
source	1..1141 /organism="Homo sapiens" /organelle="mitochondrion" /mol_type="genomic DNA" /isolate="C15542T" /db_xref="taxon: <a href="#">9606</a> "
gene	1..>1141 /gene="cytb"
CDS	1..>1141 /gene="cytb" /codon_start=1 /transl_table= <a href="#">2</a> /product="cytochrome b" /protein_id=" <a href="#">AFJ22732.1</a> " /translation="MTPMRKTNPLMKLINHSFIDLPTPSNISAWWNFGSLLGACLILQ ITTGLFLAMHYSPDASTAFSSIAHITRDVNYGWIIRYLHANGASMFFICLFLHIGRGL YYGSFLYSETWNIGIILLLATMATAFMGYVLPWGQMSFWGATVITNLLSAIPYIGTDL VQWIWGGYSVDSPTLTRFFTfhFILPFIIAALATLHLLFLHETGSNNPLGITSHSDKI TFHPYYTIKDALGLLFLLSLMTLTLFSPDLDGDPDNYTLANPLNTPSHIKPEWYFLF AYTILRSVPNKLGGVLALLSILILAMIPILHMSKQQSMMFRPLSQSLYWLLAADLLI LTWIGGQPVSYPFTIIGQVASVLYFTTILILMPTISLIENKMLKWA"

ORIGIN

1 atgaccccaa tacgcaaaac taacccccta ataaaattaa ttaaccactc attcatcgac  
61 ctccccaccc catccaacat ctccgcata tgaaacttcg gctcactcct tggcgccctgc  
121 ctgatcctcc aaatcaccac aggactattc ctagccatgc actactcacc agacgcctca  
181 accgcctttt catcaatcgc ccacatcact cgagacgtaa attatggctg aatcatccgc  
241 taccttcacg ccaatggcgc ctcaatattc tttatctgcc tcttcctaca catcggcga  
301 ggcctatatt acggatcatt tctctactca gaaacctgaa acatcggcat tatcctcctg  
361 cttgcaacta tagcaacagc cttcataggc tatgtcctcc cgtgaggcca aatatcattc  
421 tgaggggcca cagtaattac aaacttacta tccgccatcc catacattgg gacagaccta  
481 gttcaatgaa tctgaggagg ctactcagta gacagtccca ccctcacacg attcttacc  
541 tttcacttca tcttgccctt cattattgca gccctagcaa cactccaccc cctattcttg  
601 cacgaaacgg gatcaaacaa ccccctagga atcacccccc attccgataaa aatcaccc  
661 cacccttact acacaatcaa agacgcccctc ggcttacttc tcttccttct ctccttaatg  
721 acattaacac tattctcacc agacccctta ggcgacccag acaattatac cctagccaac  
781 cccttaaaca ccccttccca catcaagccc gaatgatatt tcctattcgc ctacacaatt  
841 ctccgatccg tccctaacaa actaggaggc gtccttgccc tattactatc catcctcattc  
901 ctagcaataa tccccatcct ccatatatcc aaacaacaaa gcataatatt tcgcccacta  
961 agccaatcac tttattgact cctagccgca gacccctca ttctaacctg aatcggagga  
1021 caaccagtaa gctacccttt taccatcatt ggacaagtag catccgtact atacttcaca  
1081 acaatcctaa tcctaataacc aactatctcc ctaattgaaa acaaaaactact caaatgggcc  
1141 t

//



# Genome collections

[Viruses](#) > [Riboviria](#) > [Orthornavirae](#) > [Kitrinoviricota](#) > [Alsuviricetes](#) > [Martellivirales](#) > [Togaviridae](#) >

## Alphavirus - 34 complete genomes

Retrieve sequences: -- Select data set from the list --

\* The list view for each taxonomy node shows only the next level of sublineages.  
 \* Unclassified/unassigned names are written in copper

Species [33] unclassified Alphavirus [1]

Genome	Accession	RefSeq type	Source information	Segm	Length	Protein	Neighbors	Host	Created	Updated
<input checked="" type="checkbox"/> Show / hide all segment lists										>Download
Aura virus	<a href="#">NC_003900</a>	complete		-	11824 nt	3	<a href="#">1</a>		02/09/1999	08/13/2018
Barmah Forest virus	<a href="#">NC_001786</a>	complete	strain:BH2193	-	11488 nt	4	<a href="#">35</a>	human, invertebrates, vertebrates	01/14/1997	08/13/2018
Bebaru virus	<a href="#">NC_016962</a>	complete		-	11877 nt	3	-		03/09/2012	08/13/2018
Caaingua virus	<a href="#">NC_055569</a>	complete	isolate:MS681	-	12096 nt	2	-	invertebrates	06/01/2021	06/07/2021
Cabassou virus	<a href="#">NC_038670</a>	complete	strain:CaAr 508	-	11385 nt	2	-		08/24/2018	08/24/2018
Chikungunya virus	<a href="#">NC_004162</a>	complete	strain:S27-African prototype	-	11826 nt	2	<a href="#">922</a>	human, invertebrates, vertebrates	09/06/2002	10/29/2018
Eastern equine encephalitis virus	<a href="#">NC_003899</a>	complete	strain:ssp. North American variant	-	11675 nt	4	<a href="#">453</a>	human, invertebrates, vertebrates	12/16/1991	08/13/2018
Elat virus	<a href="#">NC_018615</a>	complete	isolate:EO329	-	11634 nt	3	<a href="#">1</a>	invertebrates	09/20/2012	08/13/2018
Everglades virus	<a href="#">NC_038671</a>	complete	strain:Everglades Fe3-7c	-	11395 nt	2	-		08/24/2018	08/24/2018
Fort Morgan virus	<a href="#">NC_013528</a>	complete	isolate:CM4-146	-	11381 nt	3	<a href="#">1</a>	invertebrates, vertebrates	11/25/2009	08/13/2018
Getah virus	<a href="#">NC_006558</a>	complete	isolate:swine	-	11597 nt	3	<a href="#">46</a>	invertebrates, vertebrates	12/17/2004	08/13/2018
Highlands J virus	<a href="#">NC_012561</a>	complete	isolate:585-01	-	11526 nt	3	<a href="#">9</a>	invertebrates, vertebrates	04/15/2009	08/13/2018
Madariaga virus	<a href="#">NC_023812</a>	complete	strain:MADV/Cebus apella/BRA/BEAN5122/1956	-	11624 nt	2	<a href="#">29</a>	human, invertebrates, vertebrates	03/20/2014	08/13/2018
Mayaro virus	<a href="#">NC_003417</a>	complete		-	11411 nt	4	<a href="#">41</a>	invertebrates, vertebrates	02/22/2002	08/13/2018
Middelburg virus	<a href="#">NC_024887</a>	complete	isolate:ArB-8422	-	11550 nt	2	<a href="#">11</a>	invertebrates, vertebrates	09/17/2014	08/13/2018
Mosso das Pedras virus	<a href="#">NC_038857</a>	complete	strain:78V-3531	-	11465 nt	2	-		08/24/2018	08/24/2018
Mucambo virus	<a href="#">NC_038672</a>	complete	strain:Mucambo BeAn 8	-	11391 nt	2	<a href="#">1</a>	invertebrates	08/24/2018	08/24/2018
Ndumu virus	<a href="#">NC_016959</a>	complete		-	11688 nt	4	<a href="#">1</a>	invertebrates, vertebrates	03/09/2012	08/13/2018
Onyong-nyong virus	<a href="#">NC_001512</a>	complete		-	11835 nt	3	<a href="#">3</a>	human, invertebrates	08/02/1993	08/13/2018
Pixuna virus	<a href="#">NC_038673</a>	complete	strain:Pixuna BeAr 35645	-	11344 nt	2	-	vertebrates	08/24/2018	08/24/2018

# BLAST: Basic Local Alignment Search Tool

What if the identity of your sequence is unknown?

Matches a ***query sequence*** against any number of ***target*** sequences in a database (e.g. GenBank)

Simple, versatile, and fast

Used to

- identify species
- identify gene function
- locate domains in gene
- locate gene in genome
- expand datasets
- filter out noisy data
- perform meta-analysis

# BLAST steps

1. Generate adjacent “words” from query sequence
2. Locate those words among subject sequences in database
3. Extend match between query and subject until score drops
4. Generate statistics for quality of match

# Generate adjacent “words” from query sequence

## 1. Generate words from sequence above threshold (e.g. T=11)

Query Sequence:

>gi|16329320 (residues 412 to 594)

SGANFARQLRTHKRQRIARQATTETQADRTQQAVGRIIGSIGVVTQTTG  
RHQGILTSWVSQASFTPPGIMLAIPGEFDAYGLAGQNKAFLVNLLQEGRS  
VRRHFDHQPLPKDGDNPFSRLEHYSTQNGCLILAEALAYLECLVQSWSNI  
GDHVLVYATVQAGQVLQPNGITAIRHRKSGGQY

Fragmentation into words:



Selection of words scoring above threshold (for word **SWV**):

Substitution Matrix\*

	R	G	I	K	F	S	T	W	V
R	5	0	-1	-1	-2	1	0	-3	0
G	6	-4	-2	-3	0	-2	-2	-3	
I		4	-3	0	-2	-1	-3	3	
K			5	-3	0	-1	-3	-2	
F				6	-2	-2	1	-1	
S					4	1	-3	-2	
T						5	-2	0	
W							11	-3	
V								4	

\*A portion of the BLOSUM 62 matrix

SWV (4+11+4 = 19)

SWI (4+11+3 = 18)

TWV (1+11+4 = 16)

GWV (0+11+4 = 15)

KWV (0+11+4 = 15)

SWS (4+11-2 = 13)

SFV (4+1+4 = 9)

SRV (4-3+4 = 5)

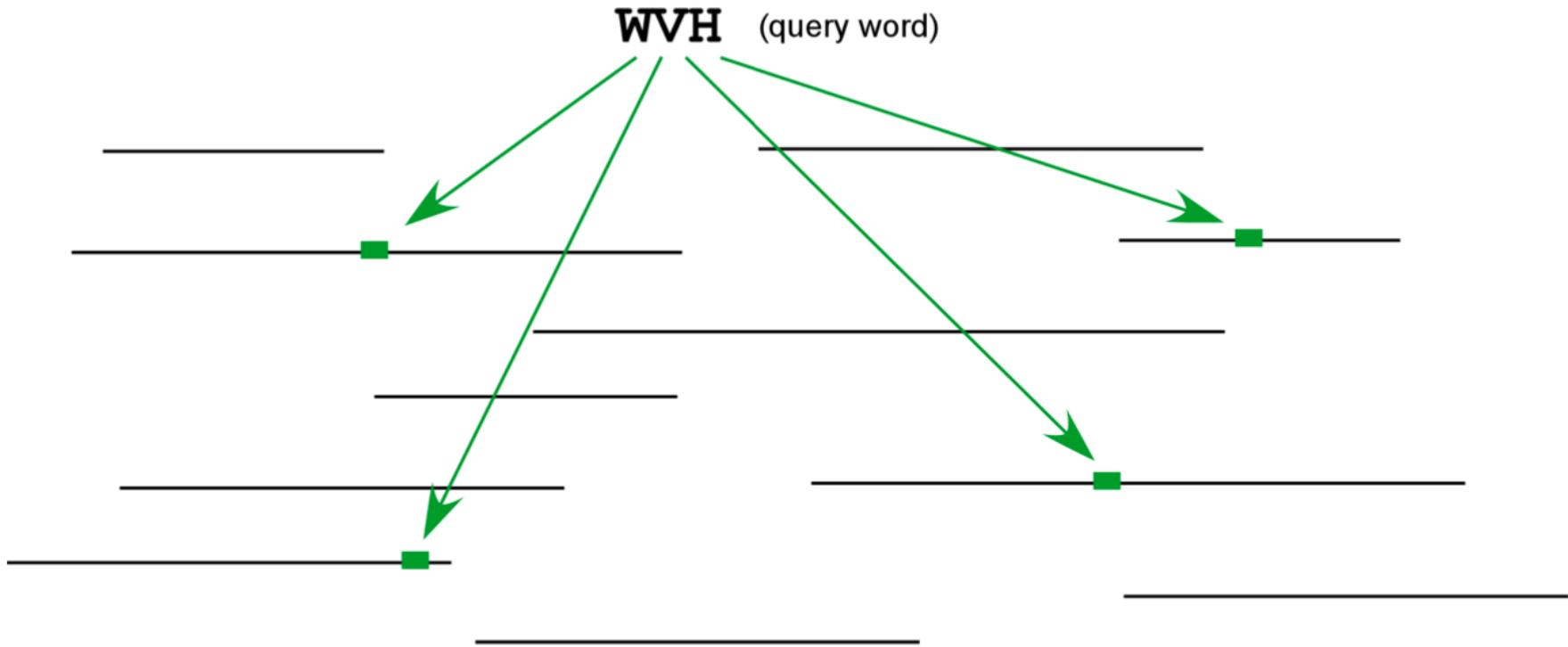
SWV

Synonyms above  
threshold 11...  
(others not shown)

Synonyms below  
threshold 11...  
(others not shown)

*expected costs of AA substitution, based  
on many sequence comparisons*

# Schematic of indexed matches



Result - instead of aligning these 3 amino acids to everything, they are aligned only with the tiny fraction of sequence regions that are good candidates for a valid alignment.

# Extension and scoring

	Match Score:	Total Score:
... QSVFEWVHLLPGA ... .. WIY ..	16	16
... QSVFEWVHLLPGA ... .. WIYQ ..	-3	13
... QSVFEWVHLLPGA ... .. WIYQK ..	-2	11
... QSVFEWVHLLPGA ... .. WIYQKA ..	-1	10

cost of match, mismatch,  
and gap is assigned by  
program and user



## Generate statistics for quality of match

### 4. Generate alignment and calculate statistics

```
>ref|YP_002482587.1| flavin reductase domain protein FMN-binding [Cyanothece sp.  
PCC 7425]  
gb|ACL44226.1| flavin reductase domain protein FMN-binding [Cyanothece sp. PCC  
7425]  
Length=585  
  
Score = 176 bits (446), Expect = 1e-42 Method: Compositional matrix adjust.  
Identities = 95/196 (48%), Positives = 125/196 (63%), Gaps = 16/196 (8%)  
  
Query 1      SGANFARQLRTHKRQRIARQATTETQADRTQQAVGRIIGSIGVVTTQTTGRH----- 52  
        +G++FA+ L+ K+QR RQ+ E Q+DRT+QAVGRIIGS+ V+T + H  
Sbjct 393     AGSDFAQVLKKAKKQRSPRQSILEVQSDRTEQAVGRIIGSLCVLTAQQQTHPHPEVEEP 452  
  
Query 53      -----QGILTSWVSQASFTPPIGIMLAIPGEFDAYGLAGQNKA  
        +L SWVSQASF PPG+ +A+ E A GL AFVLN+L+EG ++RRHF 107  
Sbjct 453     QLEVPTAMILVSWVSQASFNPPLTIALAKE-RAEGLDHSGDAFVLNVLEG  
        MNLRRHFSK 511  
  
Query 108     QPLPKDGDNPFSRLEHYSTQNGCLILAELAYLECLVQWSNSI  
        GDHVLYATVQAGQVLQ 167  
        P G++ F+ L +NGC +L + LAYLEC VQS GDH L+YATV G+VLQ  
Sbjct 512     SFAP--GEDRFAGLNIQWAENGCPVLQDCLAYLECTVQSRMECGDH  
        WLHYATVNNGKVLQ 569  
  
Query 168     PNGITAIRHRKSGGQY 183  
        P G TA++HRKSG QY  
Sbjct 570     PTGTTAVQHRKSGNQY 585
```

E-value: the number of expected random sequences in database with better match than score

Higher score, lower E-value  
Larger database, higher E-value

# NCBI BLAST interface

Input your  
**query sequence**  
as text



BLAST® » blastn suite

Standard Nucleotide BLAST

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

Query subrange

Or, upload file  No file selected.

Job Title

Enter a descriptive title for your BLAST search

Align two or more sequences

New columns added to the Description Table  
Click 'Select Columns' or 'Manage Columns'.

Choose Search Set

Database  Standard databases (nr etc.)  rRNA/ITS databases  Genomic + transcript databases  Betacoronavirus

Nucleotide collection (nr/nt)

Organism Optional Enter organism name or id—completions will be suggested  exclude

Exclude Optional Enter organism common name, binomial, or tax id. Only top 20 taxa will be shown  Models (XM/XP)  Uncultured/environmental sample sequences

Limit to Optional  Sequences from type material

Entrez Query Optional 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**   Show results in a new window

Perform BLAST  
against  
**subject sequences**



[https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE\\_TYPE=BlastSearch](https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch)  
gene from Kerfeld and Scott (PLoS Biology 2011)

# “BLAST hits”

Sequence name  
e.g. species/gene ID

Download  
sequences

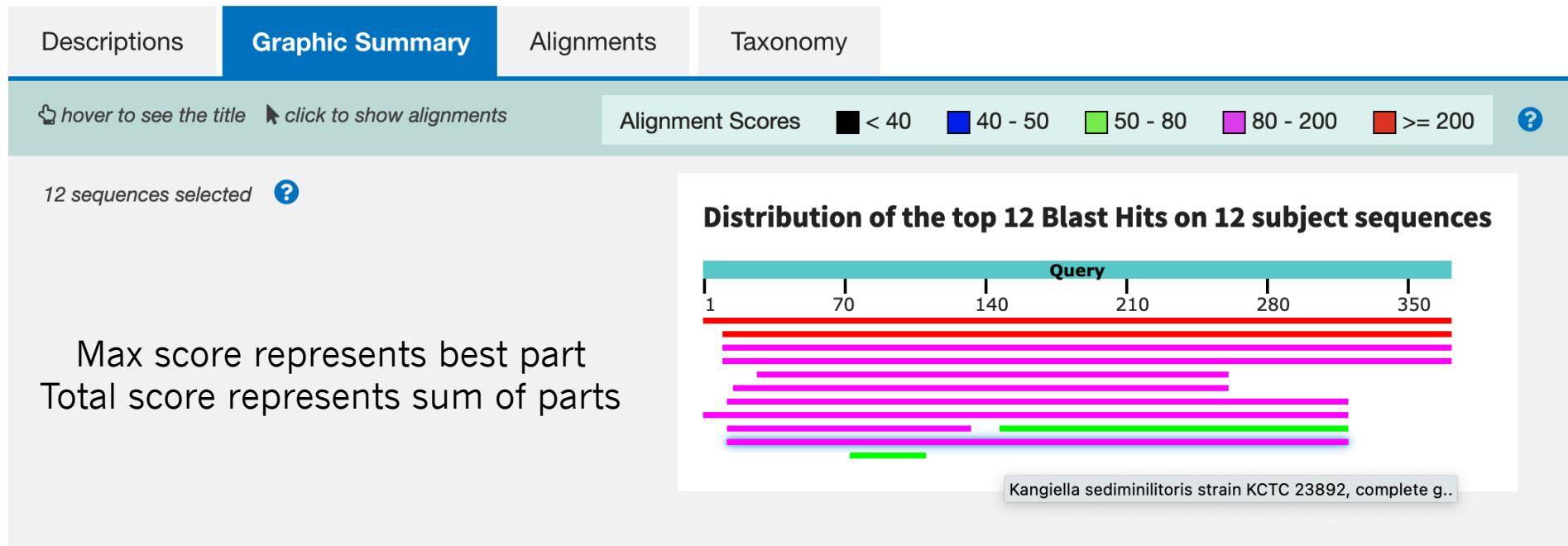
Expect value  
*lower = more significant*

% identical  
characters  
in alignment

Descriptions		Graphic Summary	Alignments	Taxonomy	Sequences producing significant alignments								
		Download									New Select columns	Show 100	?
		GenBank Graphics Distance tree of results									New MSA Viewer		
		Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession			
<input checked="" type="checkbox"/>		<a href="#">Thiomicrospira crunogena XCL-2, complete genome</a>	<a href="#">Hydrogenovibrio crunogen...</a>	688	688	100%	0.0	100.00%	2427734	<a href="#">CP000109.2</a>			
<input checked="" type="checkbox"/>		<a href="#">Hydrogenovibrio crunogenus strain SP-41 chromosome, complete genome</a>	<a href="#">Hydrogenovibrio crunogen...</a>	632	632	100%	6e-177	97.31%	2453259	<a href="#">CP032096.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Hydrogenovibrio thermophilus strain JR-2 chromosome, complete genome</a>	<a href="#">Hydrogenovibrio thermophil...</a>	427	427	100%	3e-115	87.40%	2612894	<a href="#">CP035033.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Thiomicrospira sp. S5 chromosome, complete genome</a>	<a href="#">Thiomicrospira sp. S5</a>	427	427	100%	3e-115	87.40%	2770466	<a href="#">CP014470.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Hydrogenovibrio marinus MH-110 DNA, complete genome</a>	<a href="#">Hydrogenovibrio marinus</a>	374	374	100%	4e-99	84.96%	2491293	<a href="#">AP020335.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Thiosulfatimonas sediminis aks77 DNA, complete genome</a>	<a href="#">Thiosulfatimonas sediminis</a>	272	272	97%	2e-68	80.49%	2722826	<a href="#">AP021889.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Thiomicrobacter aquaedulcis DNA, complete genome</a>	<a href="#">Thiomicrobacter aquaedul...</a>	250	250	97%	7e-62	79.45%	2440205	<a href="#">AP018722.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Shewanella dokdonensis strain DSM 23626 chromosome, complete genome</a>	<a href="#">Shewanella dokdonensis</a>	198	198	82%	3e-46	78.53%	4127406	<a href="#">CP074572.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Flocculibacter collagenilyticus strain SM1988 chromosome</a>	<a href="#">Flocculibacter collagenilyticus</a>	196	196	91%	9e-46	77.33%	3973578	<a href="#">CP059888.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Shewanella sp. FJAT-54481 chromosome, complete genome</a>	<a href="#">Shewanella sp. FJAT-54481</a>	187	187	87%	6e-43	77.20%	3812587	<a href="#">CP073587.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Amphritea japonica ATCC BAA-1530 DNA, complete genome</a>	<a href="#">Amphritea japonica ATCC ...</a>	174	174	83%	4e-39	77.29%	3833046	<a href="#">AP014545.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Legionella pneumophila subsp. fraseri strain F-4198 chromosome, complete genome</a>	<a href="#">Legionella pneumophila su...</a>	171	171	97%	6e-38	75.73%	3461540	<a href="#">CP021279.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Legionella pneumophila subsp. fraseri strain D-4058 chromosome, complete genome</a>	<a href="#">Legionella pneumophila su...</a>	171	171	97%	6e-38	75.73%	3548205	<a href="#">CP021277.1</a>			
<input checked="" type="checkbox"/>		<a href="#">Legionella pneumophila subsp. franciscana strain D-5027 chromosome, complete genome</a>	<a href="#">Legionella pneumophila su...</a>	171	171	97%	6e-38	75.73%	3471650	<a href="#">CP021264.1</a>			

BLAST alignment score  
higher = better match

# Graphic summary of BLAST scores



Scores determined by alignment costs  
for match/mismatch/indel reconciliation  
(covered in a later lab)

# BLAST generates a summary for each subject sequence

## Sulfurovum indicum strain ST-419 chromosome, complete genome

Sequence ID: [CP063164.1](#) Length: 2209694 Number of Matches: 1

Range 1: 2058656 to 2058979 [GenBank](#) [Graphics](#)

▼ Next Match ▲ Previous Match

Score	Expect	Identities	Gaps	Strand	
102 bits(55)	2e-17	241/330(73%)	15/330(4%)	Plus/Minus	
Query 1		ATGGCAATTACAAAAGACGATATTTAGAAGCAGTTGCTAACATGTCAGTAATGGAAGTT		60	
Sbjct 2058979		ATGGCAACAACAAAAGAAGATGTATTAGAATTATTCTAACCTTCAGTACTTGAGCTT		2058920	
Query 61		GTTGAACCTGTTGAAGCAATGGAAGAGAAGTTGGTGTCTCA---GCAGCAGTTGCG		117	
Sbjct 2058919		TCTGAGCTTGTAAAAGAATTGAGAAAGAAAAATTGGTGTAACTGCACAAGCTACAGTAGTT		2058860	
Query 118		GTTGCAGGTCTGCAGGTGATGCTGGCGCTGCTGGTGAAGAACAAACAGAGTTGACGTT		177	
Sbjct 2058859		GCAGCTGGTGCTGCCGGTGGTGCCTGCTGAAGCTGCTGAAGAGCAGACAGATTCAACGTT		2058800	
Query 178		GT-CTTGACTGGTGCTGGTACAACAAAGT-TGCAGCAATCAAAGCCGTTGTGGCGCA-		234	
Sbjct 2058799		GTTCTT-ACAGACGCTGGTGCAGAAAGATCAACCA-CAATTAAAGTTGTAAGAGCAGTC		2058743	
Query 235		ACTGGTCTTGGGCTTAAAGAAGCGAAAAGTGAGTTGAAAGTG-CACCAT-TACGCTT		291	
Sbjct 2058742		ACAGGTCTGGACTTAAAGAAGCGAAAGCTGCTGTTG-AAGAGACTCCATCTTA-CTT		2058686	
Query 292		AAAGAGGGTGTCTAAAGAAGAAGCAGAA 321			
Sbjct 2058685		AAAGAGGGTGTCTAAAGAAGAAGCTGAA 2058656			

Query sequence position

Subject sequence position

Poor alignment  
(gaps, mismatches = lower score)

Perfect alignment  
(more matches = higher score)

# arrays

an **array variable** contains multiple values  
that are individually accessed by index

```
#!/bin/sh
# populate array
LETTERS[0]="A"
LETTERS[1]="B"
LETTERS[2]="C"
# INCORRECT: access array, show element 0
echo $LETTERS
# INCORRECT: access array, add "[1]" text
echo ${LETTERS[1]}
# CORRECT: access element 1
echo ${LETTERS[1]}
```

*array1.sh*

```
$ ./array1.sh
A
A[1]
B
```

*running array1.sh*

```
#!/bin/sh
# create array
NUMBERS=(1 2 3 4)
echo ${NUMBERS[2]} # 3
echo ${NUMBERS[9]} # empty (no entry)
# assign new array values
NUMBERS[2]="three"
NUMBERS[9]="ten"
echo ${NUMBERS[2]} # "three"
echo ${NUMBERS[9]} # "ten" (new entry)
```

*array2.sh*

```
$ ./array2.sh
3
three
ten
```

*running array2.sh*

# more with arrays

```
#!/bin/sh
# arg1 is directory
DIR=$1
# make array with `ls` command subst.
FILES=( $(ls ${DIR}) )
# print individual array elements
echo "Individual elements"
echo "${FILES[0]}"
echo "${FILES[1]}"
# print entire array
echo "Entire array"
echo "${FILES[@]}
```

*array3.sh*

```
$ ls tmp
file1.txt file2.txt file3.txt
$ ./array3.sh tmp
Individual elements
file1.txt
file2.txt
Entire array
file1.txt file2.txt file3.txt
```

*running array3.sh*

```
#!/bin/sh
# all args are stored in $@ array
for i in "$@"; do
  if [[ -f ${i} ]]; then
    echo "Copying ${i} to ${i}.bak"
    cp ${i} ${i}.bak
  fi
done
```

```
$ ls tmp
file1.txt file2.txt file3.txt
$ ./array4.sh tmp/*.txt
Copying tmp/file1.txt to tmp/file1.txt.bak
Copying tmp/file2.txt to tmp/file2.txt.bak
Copying tmp/file3.txt to tmp/file3.txt.bak
$ ls tmp
file1.txt  file1.txt.bak  file2.txt
file2.txt.bak  file3.txt  file3.txt.bak
```

*running array4.sh*

# for-loops with files

two approaches to loop over lines in a file;  
set the ***input field separator*** (IFS) to split by line

```
#!/bin/bash
FILE=$1          # arg1: input file
SAVE_IFS=${IFS}    # restore IFS when done
IFS=$'\n'        # set '\n' for IFS
LINES=$(cat ${FILE}) # extract text from FILE
for LINE in ${LINES} # for each line (split by IFS)
do
    echo ${LINE}      # print LINE to stdout
done
IFS=${SAVE_IFS}      # done! restore IFS
```

*forloop1.sh* is simpler, but less portable

```
$ ./forloop1.sh file.txt
Knock, knock!
Who's there?
```

running *forloop1.sh*

```
#!/bin/bash
FILE=$1          # arg1: input file
# while : loop until stop condition (EOF)
# IFS= : IFS set to null ('\n')
# read : split input by IFS + advance file position
# -r : do not treat '\' as special char
# LINE : store each token in LINE
while IFS= read -r LINE
do
    echo ${LINE}      # begin code block
done < ${FILE}      # input redirect to while/done
```

*forloop2.sh* is more complex, but more portable

```
$ ./forloop2.sh file.txt
Knock, knock!
Who's there?
```

running *forloop2.sh*

# xargs

xargs (extended arguments) converts input into arguments for another command

```
$ # find all .txt files
$ find data -name "*.txt"
data/file1.txt
data/docs/file2.txt
data/secrets/password.txt
$ # word count of `find` output
$ find data -name "*.txt" | wc
    3 3 61
$ # word count against each found file
$ find data -name "*.txt" | xargs wc
0 0 0 data/file1.txt
0 0 0 data/docs/file2.txt
0 0 0 data/secrets/password.txt
0 0 0 total
```

```
$ # argument file
$ cat files.txt
example/file1.txt
example/file2.txt
example/file3.txt
$ # target files to remove
$ ls example/
file1.txt file2.txt file3.txt
$ # use argument file lines for rm
$ xargs -a files.txt rm
$ # files deleted
$ ls example/
```

```
$ ls
a.txt b.txt
$ echo "a.txt a.txt.tmp b.txt b.txt.tmp" | xargs cp
cp: target 'b.txt.tmp' is not a directory
$ echo "a.txt a.txt.tmp b.txt b.txt.tmp" | xargs -n2
a.txt a.txt.tmp
b.txt b.txt.tmp
$ echo "a.txt a.txt.tmp b.txt b.txt.tmp" | xargs -n2 cp
$ ls
a.txt a.txt.tmp b.txt b.txt.tmp
```

# Overview for Lab 07