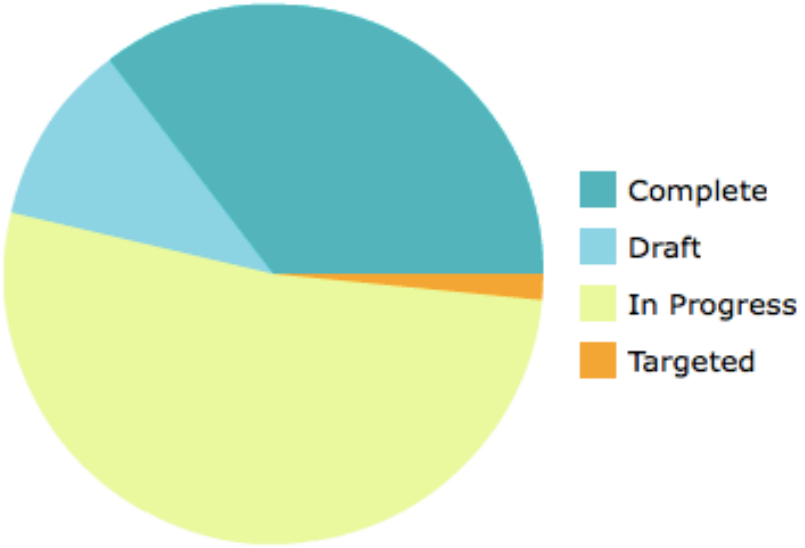
Command-Line BLAST

Agenda

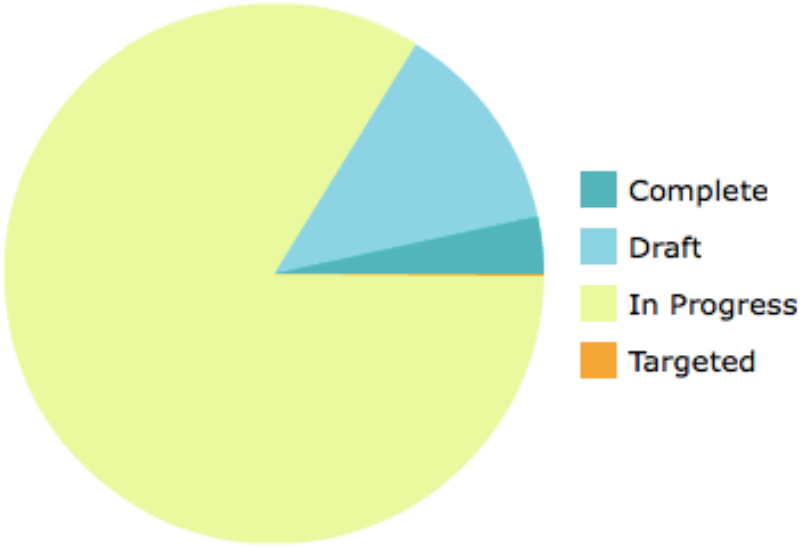
- Searching Databases for Similar Sequences
- Similarity and Differences in Sequence Data
- Introduction to the Command Line
- BLAST

There are databases *full* of sequence data to be searched!



Bacterial

32227 genomes



Eukaryotic

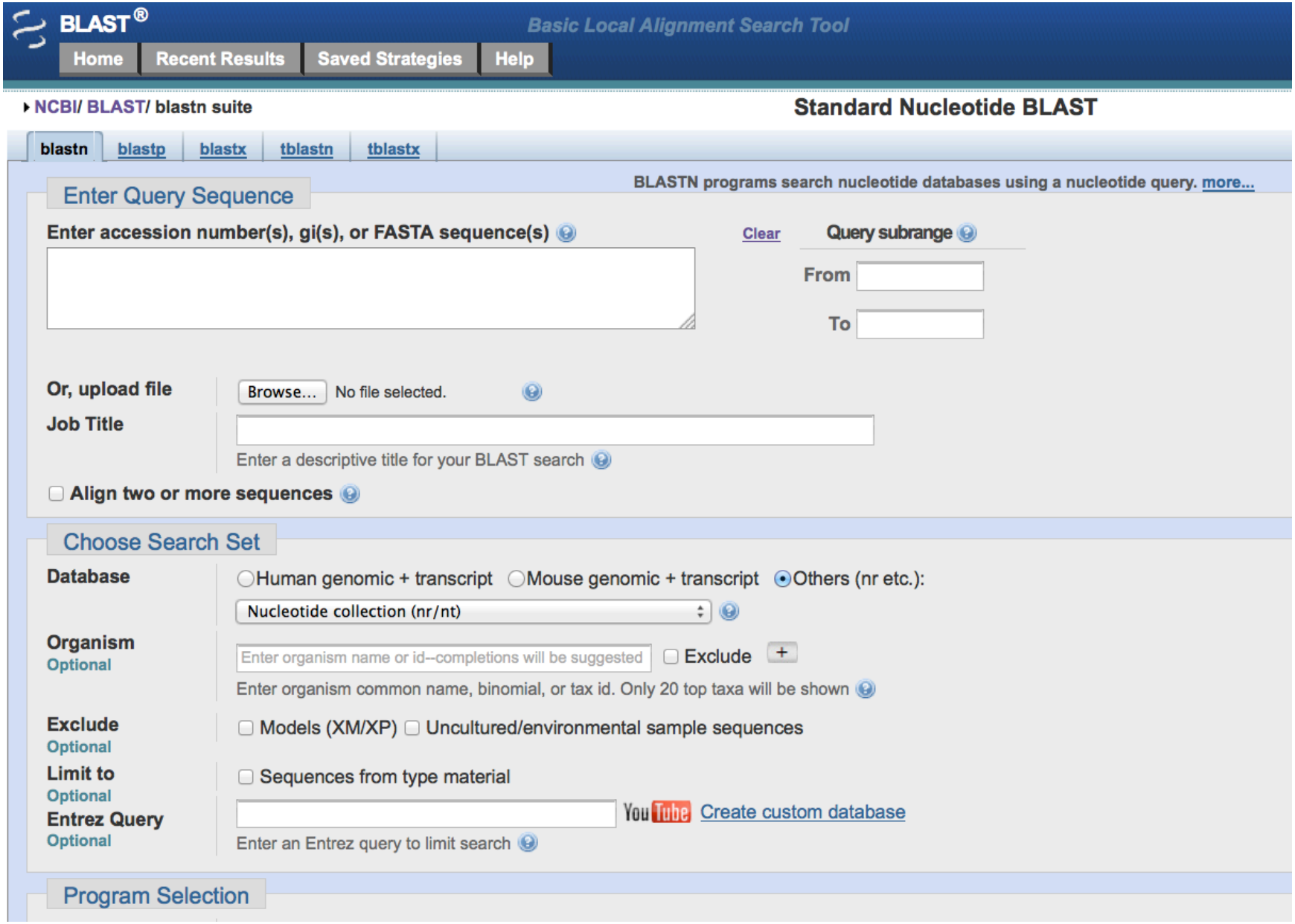
7236 genomes

from the **Genomes OnLine Database***s of Feb 2014*

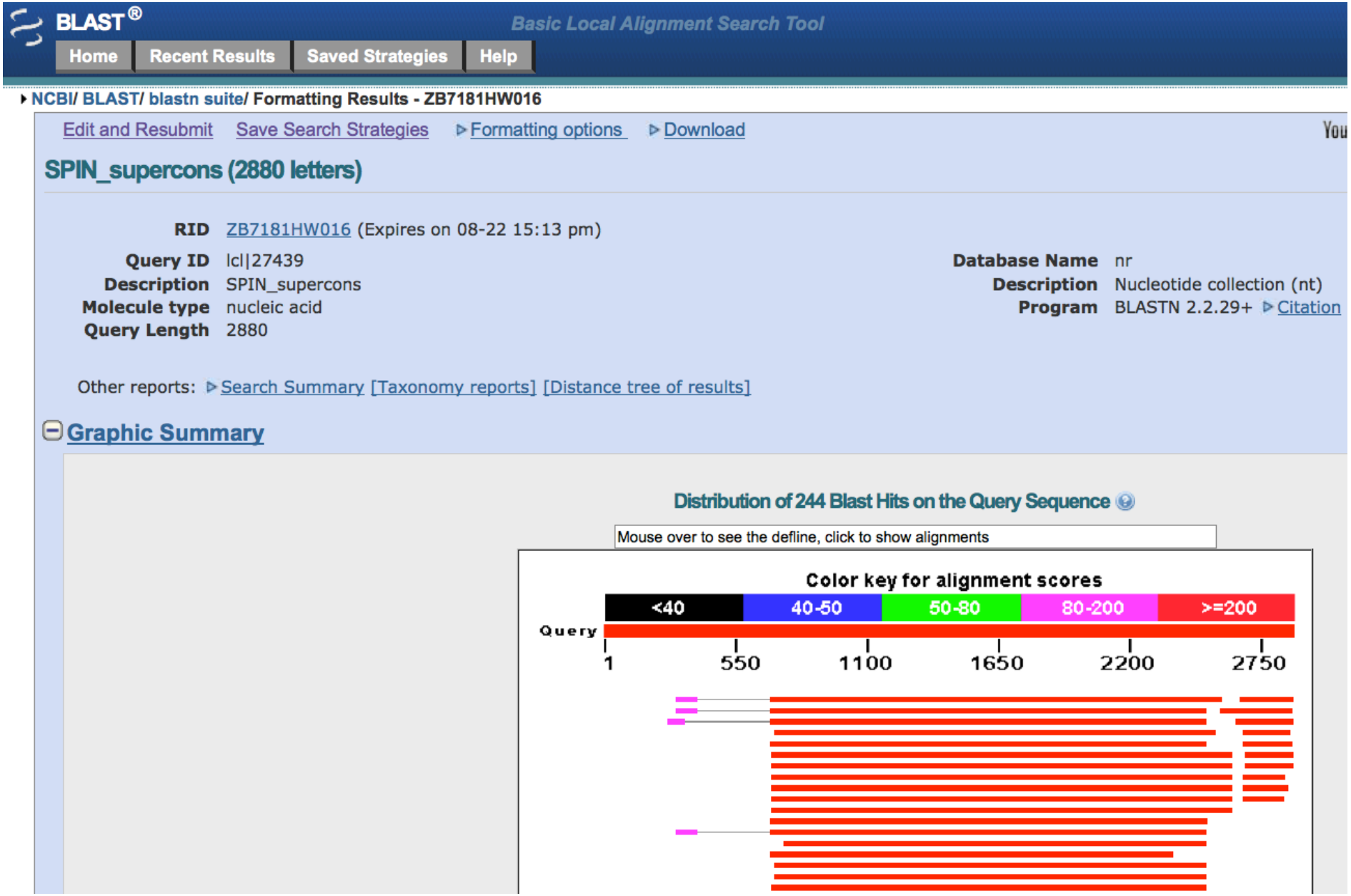
What are we looking for when we search sequences?

- Similar sequences
  - Find orthologous genes in other species
  - Find paralogous genes within a species
  - Learn about gene function by finidng genes with known function that contain similar domains
  - Identify species
  - Discover biologically meaningful genetic changes (e.g., new functions)
  - Date divergence times
  - Find repetitive elements
  - .
  - .
  - .

How Do We Typically Search? BLAST!



BLAST Returns Similar Sequences from the Selected Database



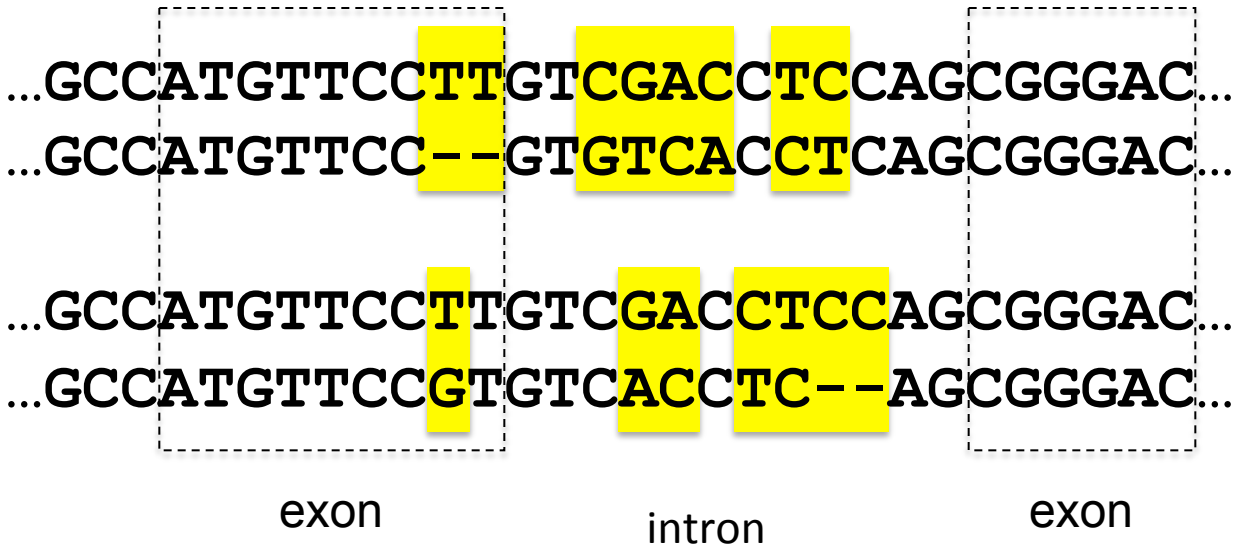
How? BLAST is an Alignment Program

What does it mean to "align" DNA sequences?

```
…GCCATGTTCCTTGTCGACCTCCAGCGGGAC…
…GCCATGTTCC--GTGTCACCTCAGCGGGAC…

…GCCATGTTCCTTGTCGACCTCCAGCGGGAC…
…GCCATGTTCCGTGTCACCTC--AGCGGGAC…
```

Which alignment is better?



Which alignment is *much* better?

Consider the task of aligning this pair of sequences:

species 1 **GCCTACGACCTCCAGAC**

species 2 **GCGTTGGCTCCAGAC**

Two Possibilities

species 1	**GCCTACGACCTCC**	**GCCTACGACCTCC**
species 2	**GCGTTGG--CTCC**	**GCGTT-GGC-TCC**

Two Possibilities

species 1	GCCTACGACCTCC	GCCTACGACCTCC
species 2	GCGTTGG--CTCC	GCGTT-GGC-TCC

Scoring an alignment

species 1	GCCTACGACCTCC	GCCTACGACCTCC
species 2	GCGTTGG--CTCC	GCGTT-GGC-TCC

- Considerations:
  - Measure percent identity
  - Scoring: match, mismatch, gap
  - Two different alignments may give same score

Is a gap worse than a mismatch? Why?

Is a longer gap worse, or are multiple gaps worse?

Scoring an alignment

species 1	GCCTACGACCTCC	GCCTACGACCTCC
species 2	GCGTTGG--CTCC	GCGTT-GGC-TCC

- Scoring metric:
  - Match score (match bonus)
  - Mismatch score
  - Gap score (gap penalty)

example:

1

0

-1

or

-1 gap open

-0.25 gap extension

Exhaustive alignment algorithm

- *Try all possible alignments.*
- *Choose alignment(s) with highest score.*

-1 GCCTAC GCT---	0 GCCTAC G-CT--	-2 GCCTAC G--CT-	-2 GCCTAC G---CT
-1 GCCTAC G-C-T-	-1 GCCTAC G-C--T	-3 GCCTAC -G-C-T	-3 GCCTAC --G-CT

Match = +1
Mismatch = 0
Gap = -1

Keep in mind, the lowest scoring alignment may still not be the "true" alignment! (That is, it may not reflect an alignment of sites that are truly homologous to each other.) *It is a best estimate!*

Algorithms

- Needleman-Wunsch -- Global alignment
  - Scores matches, mismatches and gaps
  - Provides optimal alignment and score
  - Efficient
- Smith-Waterman -- Local alignment
  - All similar areas between the two strings are returned
  - High-scoring partial alignments
  - Partial match between sequences
  - Allow for introns
  - Find shared domains within distinct proteins
  - BLAST

Local Alignment Algorithm Implemented in BLAST

– Heuristic
– Local alignment of "words" (*k*-tuple)
– Extend match from each end of matching word

query sequence: foreviltoflourishitonlyrequiresgoodmentodonothing

k-tuples ("words"):
35 ...
36 men
37 ent * search for initial match, then extend
38 nto
39 ...

threshold = 15

database sequence A: no match

```
allthatisnecessaryforthetriumphofevilisthatgoodmendonothing
```

database sequence B: initial match — score: 13

```
allthatisnecessaryforeviltosucceedisforgoodmentodonothing
                   ito---nl--yrequires---goodmentodonothing
                   <---------------------------extend ---------->
```

database sequence C: initial match — score: 17

```
allthatisnecessaryfortheforcesofeviltowinintheworldisfo---renough-goodmentodonothing
                                                   it-onlyrequiresgoodmentodonothing
                                                   <-------------------extend ----------->
```

When sequences do differ, why is that?

- Usually, Mutation!
  - Many types
- Consequences of mutations
  - No effect
  - Changes in structure→changes in function
- Mutations can be
  - Maintained or purged (Natural Selection)
  - Accidentally lost or kept (Genetic Drift)
  - Shared between populations by migration (Gene Flow)

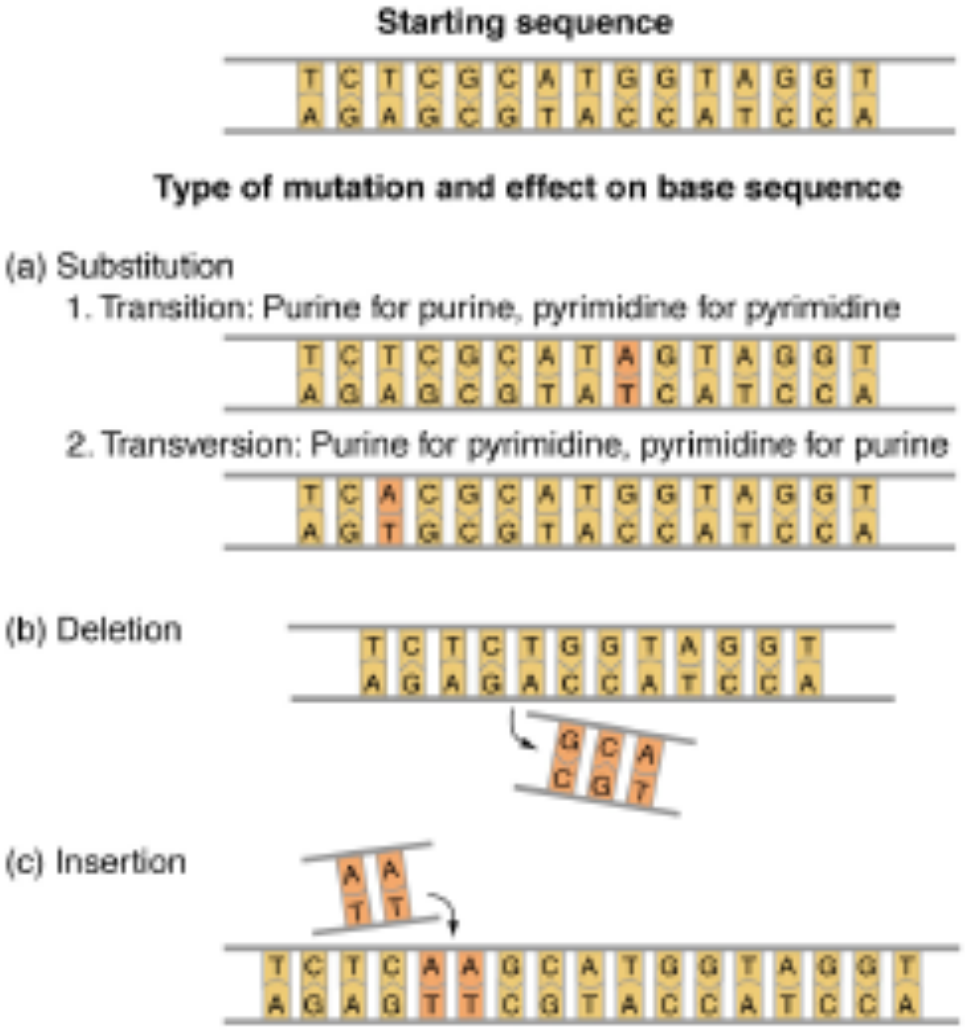
If there are genetic changes in population over time,
the population is said to be *evolving*.
Evolution occurs at the population level (individuals do not evolve).

Small-scale mutations

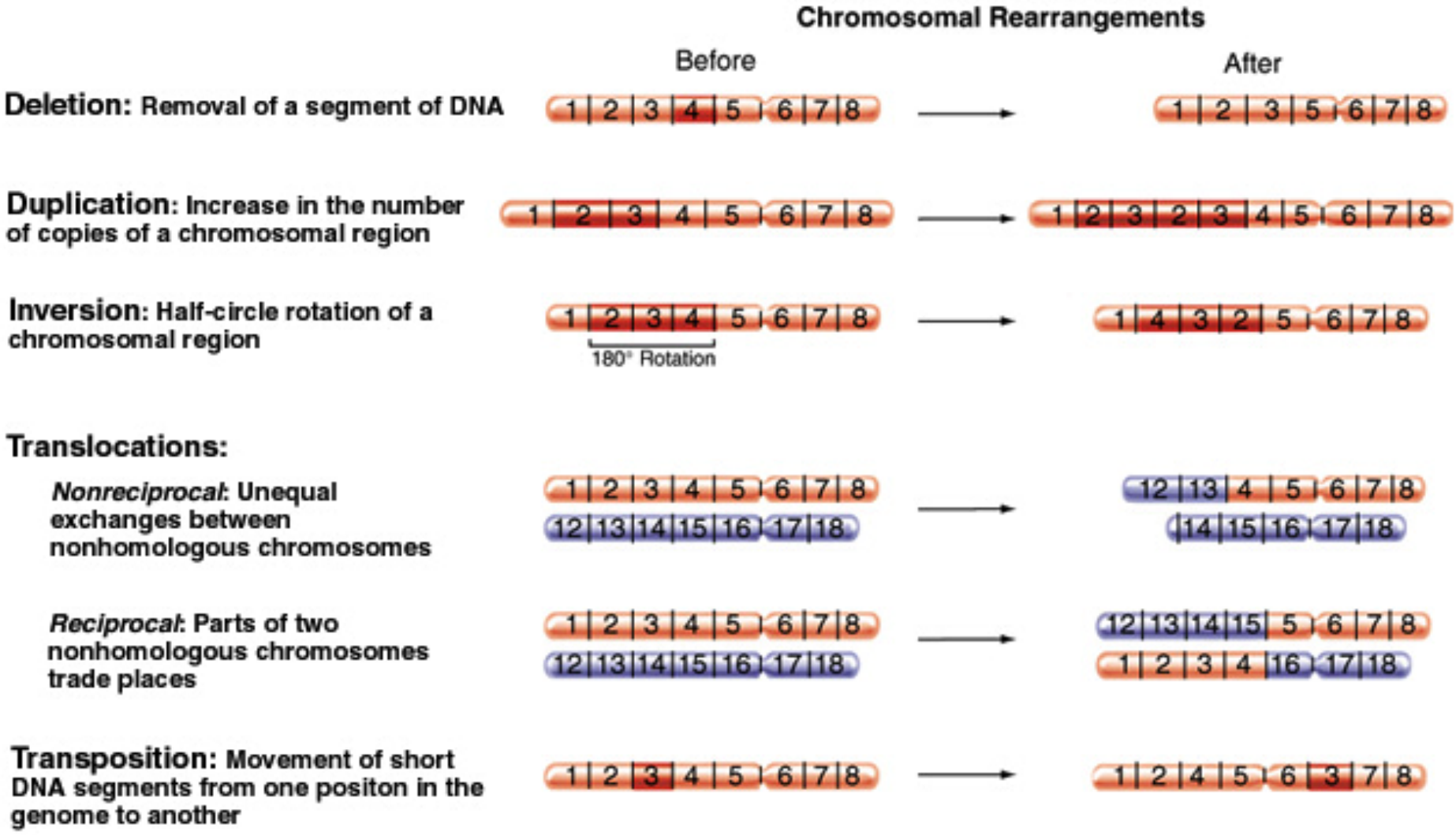
- Substitution
  - ***Transition***

  pur-pur or pyr-pyr change
  - ***Transversion***

  pyr-pur change
- Deletion
- Insertion



LARGE-Scale Mutations



2 major patterns when searching sequence data

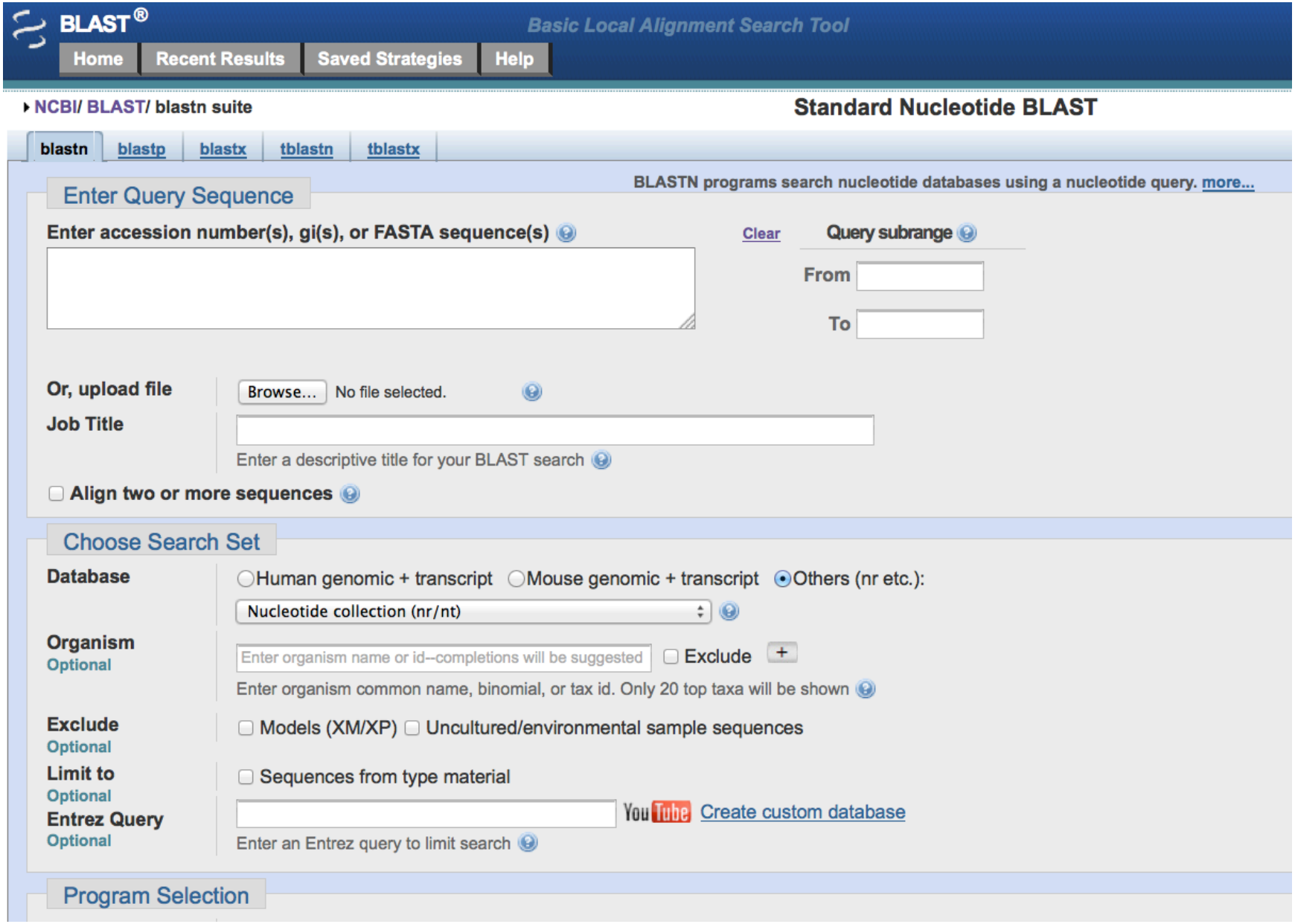
- Similarity
  - Conservation
  - Convergence
- Non-Similarity
  - Divergence
  - Neutral Polymorphism

We often rely on levels of
similarity
to make sense of sequence data.

To measure sequence similarity, we need fast algorithms for searching well organized, accurately annotated, up-to-date databases. We also need tools for visualizing and quantitative metrics for assessing our results.

And we need us! Well-trained, careful, rigorous investigators.

BLAST on The Web Is Convenient for Some Tasks

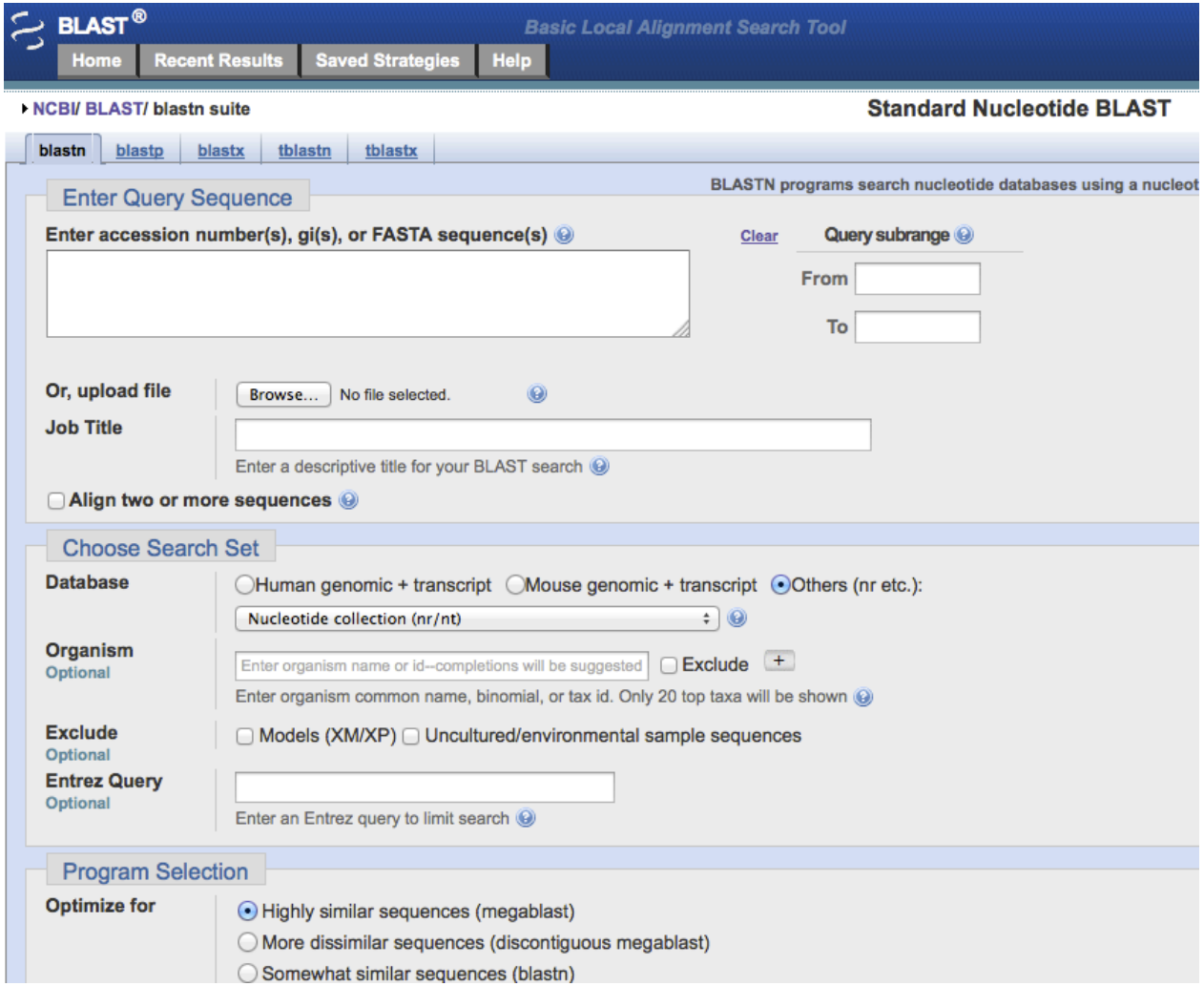


But a Graphical User Interface Is Both Good and Bad

Not good for long sequences of operations that need to be repeated on multiple datasets

No log of what you did and what commands you executed

Not conducive to executing jobs remotely on a cluster



Plus, you may want to sequence, search, and analyze a genome or transcriptome *before making it public*

As an aside: What determines if you want to sequence a genome or a transcriptome?

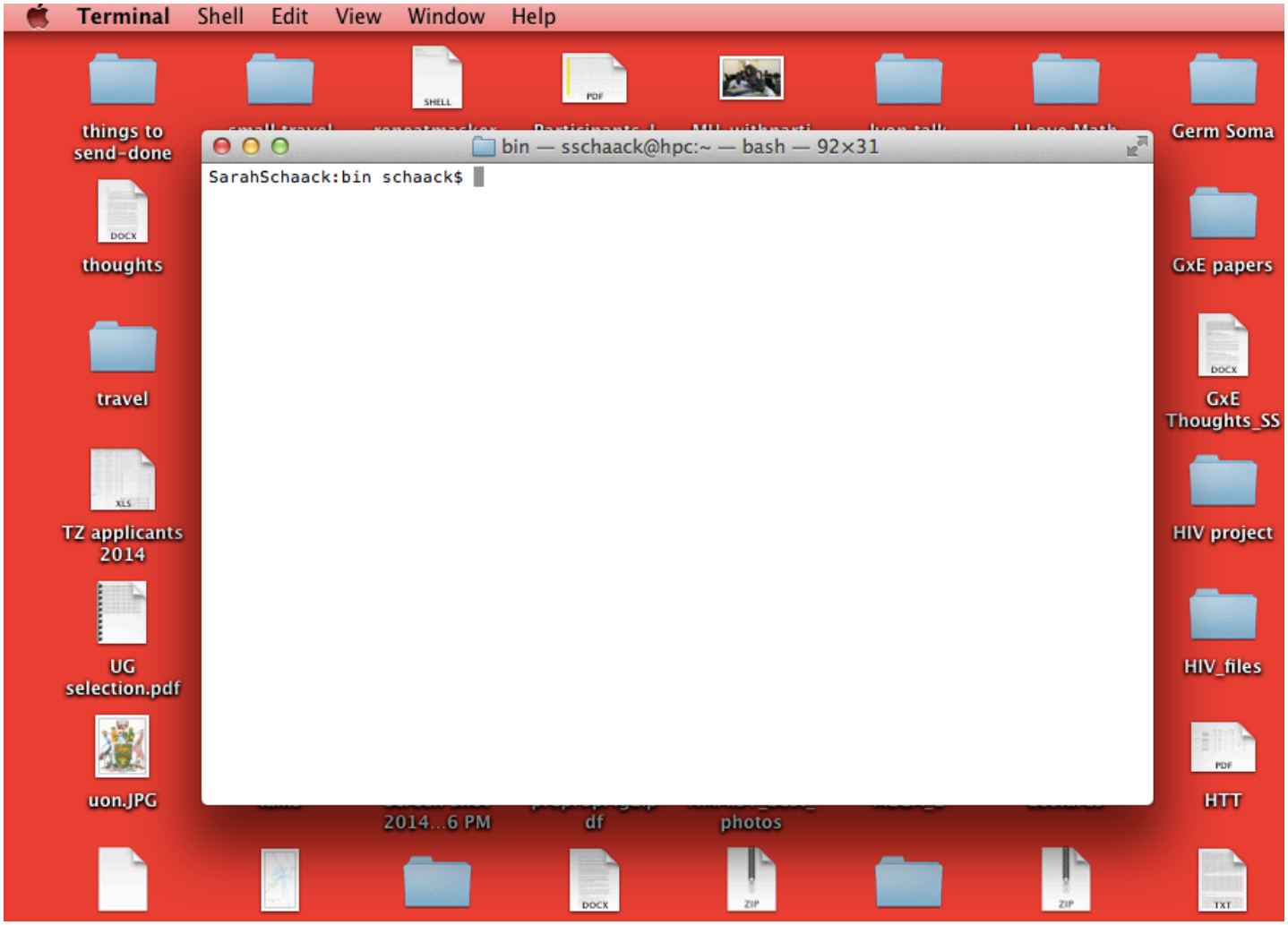
- Size
  - For example, among animals
    - smallest <20 Mb, (*Pratylenchus coffeae*, Plant-parasitic nematode)
    - biggest >130 Gb, (*Protopterus aethiopicus*, Marbled lungfish)
- Community interest in the data
  - Availability of a reference genome
  - Interest in genes versus other aspects of the genome
  - Note: cDNAs frequently sequenced in addition to WGS for assembly and gene prediction

Or search specific databases to shorten search time

- Public databases
  - Coding/Non-Coding DNA and mRNAs (nt/nr)
  - Whole Genome Shotgun (WGS)
  - Expressed sequence tags (EST)
  - Reference genomes
  - Your own!
- Private databases that you make

The most commonly used sequence searching algorithm is BLAST because it is a good combination of fast and accurate.

If you install BLAST locally you can do big jobs or to search your own sequence data!



But requires getting comfortable with the *command line*….

THE COMMAND LINE

- Advantages
  - Extremely, extremely powerful – NOT a primitive interface
  - Up-front investment with ENORMOUS long-term payoff
  - Absolute necessity for big datasets, which characterize modern biology
  - Automate tasks
  - Do and re-do and re-do and re-do analyses with minimal added effort
  - Easy to record what you did, minimizing mistakes while enhancing repeatability and troubleshooting!

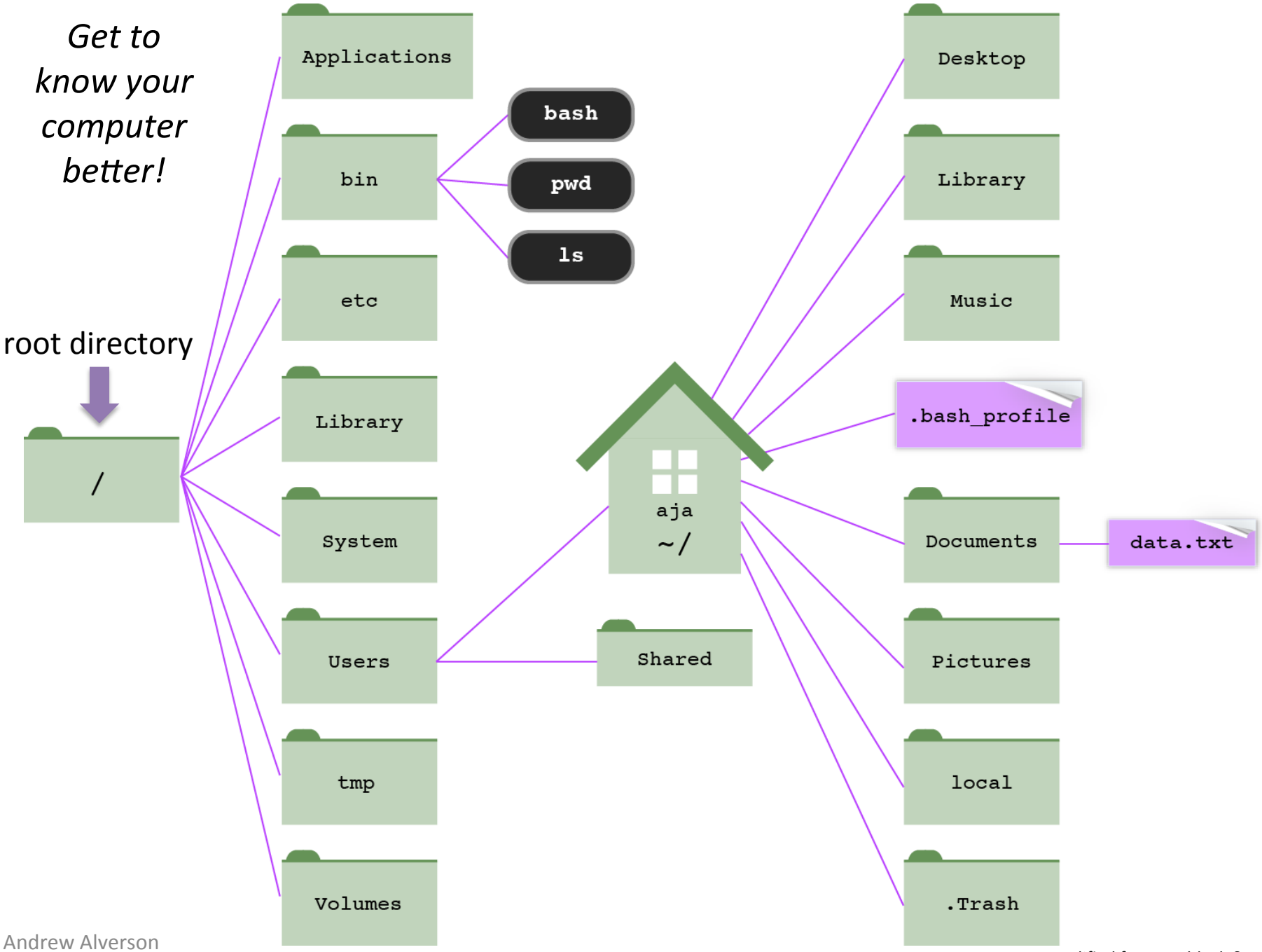
THE COMMAND LINE

- Requirements
  - Gather a small amount of courage
  - Get to know your computer a little better
  - Learn to use paths to locate folders (instead of Finder or clicking through Windows) to navigate around your computer
  - Learn some simple commands
  - Open a Terminal or Cmd Window

Finder > Go > Utilities > Terminal

Start> Run> Cmd

Get to know your computer better!



Andrew Alverson

modified from Haddock & Dunn

WRITING OUT PATHS

- A **path** is a written description of a location in the file system
- Both directories and files have paths
- Consists of directory names separated by slashes (fwd `/` or `rev\`)
- **Example of a complete path**

`/Users/schaack/Desktop/watermelon_files/watermelon.fsa`

- **Relative path** – describes where a file or folder is in relation to another folder, usually the working directory
- **Working directory** – where you're at right now; if you're in the Desktop, the relative path to `watermelon.fsa` is:

`watermelon_files/watermelon.fsa`

SOME SIMPLE COMMANDS

--navigating folders $ cd – change directory

--see contents of a file $ ls – list files

or

$ dir (in PC world)

Many more commands, plus some awesome short cuts (like tabbing!), are appended at the end of this presentation.

BEWARE OF TYPOS!

99% of command-line fails are due to typos.

Extra spaces, backwards slash marks, wrong commands (PC vs Mac), or misspellings will thwart your analysis! If you get an error message, double check that you didn't make a typo by pressing the up (↑) arrow and checking the last command you gave.

Using Command-line BLAST

- Basic Local Alignment Search Tool
- Single most important algorithm in the field of bioinformatics
- In essence, BLAST finds statistically significant similarities between sequences by evaluating pairwise alignments
- When you download command-line BLAST*, there won't be an icon on your desktop*, but you can search for where it is located on your computer if you want to see
- Two types of alignment
    1. Global – sequences aligned across their *entire* length and best alignment is found
    2. Local – the best *subsequence* alignment is found

BLAST is a LOCAL Search Algorithm…. See Module 4 for more on Global Alignments!

The BLAST algorithm

1. "Seeding" – Chop up the query sequence into short (generally 7–28 nt) subsequences (or "words")
2. Make a look-up table of the query words, and find similar "neighboring words" in the subject sequence ("word hits")
3. "Extension" – When there's a match, try to extend it beyond the word match using a set of rules and scoring schemes, including:
    - match rewards and mismatch penalties
    - the penalty for opening a new gap
    - the penalty for extending an existing gap
4. Compile the best alignments based on their scores

The Five BLAST Programs

Program	Query	Database
BLASTN	Nucleotide	Nucleotide
BLASTP	Protein	Protein
BLASTX	Translated nucleotide	Protein
TBLASTN	Protein	Translated nucleotide
TBLASTX	Translated nucleotide	Translated nucleotide

Running BLAST

- Input FASTA-formatted files

```
>Citrullus_nad4L
ACGGATCCTATCAAATATTTCACATTTTCTATGATCATC
TTGGGTTAGCCATTTTCGTTATTACTTTCCGAGTCCGAG
```

- Remote searches to GenBank's non-redundant (nr) database

```
$ blastn -query nad4L.fasta -remote -db nr -num_descriptions 10
```

- Local searches require a query and a subject (database)
    1. format a database
    2. query the database

Running BLAST Locally

Step 1: Format a BLAST database with `makeblastdb`

- input is a FASTA file with one or more sequences
- nucleotide OR amino acid data, not both

```
# see the program usage and options
  $ makeblastdb -help

# make a nucleotide database with indexed files
  $ makeblastdb –in watermelonmt.fsa -dbtype nucl
```

Querying a local BLAST database

Step 2: Query your database with any of the following programs

1. blastn
2. blastp
3. blastx
4. tblastn
5. tblastx

```
# see the program usage and options for blastn
$ blastn –help

# run blastn
$ blastn –query watermelon_nt/nad4L.fasta –db
    watermelonmt.fsa –word_size 11 –reward 2
    –penalty 3 –gapopen 5 –gapextend 2
```

BLAST* report: HEADER

```
BLASTX 2.2.24+
```

Program and version

```
Reference: Stephen F. Altschul, Thomas L. Madden, Alejandro A.
Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J.
Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of
protein database search programs", Nucleic Acids Res. 25:3389-3402.
```

Citation

```
Database: nad4L/nad4L
           20 sequences; 2,000 total letters
```

BLAST Database data

```
Query=  watermelon [organism=Citrullus lanatus] [common=watermelon]
[cultivar=Florida giant] [molecule=DNA] [location=mitochondrion]
[topology=circular] Citrullus lanatus mitochondrion, complete genome
Length=379236
```

Query data

BLAST* report: ONE-LINE SUMMARIES

List of "hits" in the database, ranked from best to worst

Sequences producing significant alignments:	Score (Bits)	E Value
Ricinus_communis	119	6e-29
Citrullus_lanatus	119	6e-29
Cucumis_sativus	119	1e-28
gi|114151577|ref|YP_740359.1| NADH dehydrogenase subunit 4L [Ze...	117	2e-28
gi|114151643|ref|YP_740396.1| NADH dehydrogenase subunit 4L [Ze...	117	2e-28
gi|94502695|ref|YP_588278.1| NADH dehydrogenase subunit 4L [Zea...	117	2e-28
gi|114151609|ref|YP_740431.1| NADH dehydrogenase subunit 4L [Ze...	117	2e-28
gi|115278615|ref|YP_762502.1| NADH dehydrogenase subunit 4L [Tr...	117	2e-28
gi|115278545|ref|YP_762346.1| NADH dehydrogenase subunit 4L [So...	117	2e-28
gi|89280726|ref|YP_514666.1| NADH dehydrogenase subunit 4L [Ory...	117	2e-28
Cucurbita_pepo	117	3e-28
Vigna_radiata	116	5e-28
gi|81176546|ref|YP_398428.1| nad4L [Triticum aestivum]	116	5e-28
gi|57013916|ref|YP_173389.1| NADH dehydrogenase subunit 4L [Nic...	116	5e-28
Carica_papaya	116	7e-28
gi|13449343|ref|NP_085525.1| NADH dehydrogenase subunit 4L [Ara...	113	4e-27
gi|9838383|ref|NP_063995.1| NADH dehydrogenase subunit 4L [Beta...	108	2e-25
gi|112253862|ref|YP_717118.1| NADH dehydrogenase subunit 4L [Br...	105	9e-25
Silene_latifolia	100	4e-23
Cycas_taitungensis	85.5	1e-18

BLAST* report: ALIGNMENTS

Database sequence

```
> gi|89280726|ref|YP_514666.1| NADH dehydrogenase subunit 4L [Oryza
sativa indica]
Length=100
```

```
 Score =  117 bits (294),  Expect = 2e-28
 Identities = 95/99 (95%), Positives = 95/99 (95%), Gaps = 0/99 (0%)
 Frame = -1

Query  366429  DPIKYFTFSMiisilgirgillnrrnipIMSMPIESMLLAvnsnflvfsvsSDDMMGQSF  366250
               DPIKYFTFSMIISILGIRGILLNRRNI IMSMPIESMLLAVN NFLVFSVS DDMMGQSF
Sbjct  2       DPIKYFTFSMIISILGIRGILLNRRNILIMSMPIESMLLAVNLNFLVFSVSLDDMMGQSF  61

Query  366249  ASLVPTVAAAESAIGLAIFVITFRVRGTIAVEFINSIQG  366133
               ASLVPTVAAAESAIGLAIFVITFRVRGTIAVEFIN IQG
Sbjct  62      ASLVPTVAAAESAIGLAIFVITFRVRGTIAVEFINCIQG  100
```

BLASTX report: ALIGNMENTS

```
> gi|89280726|ref|YP_514666.1| NADH dehydrogenase subunit 4L [Oryza
sativa indica]
Length=100
```

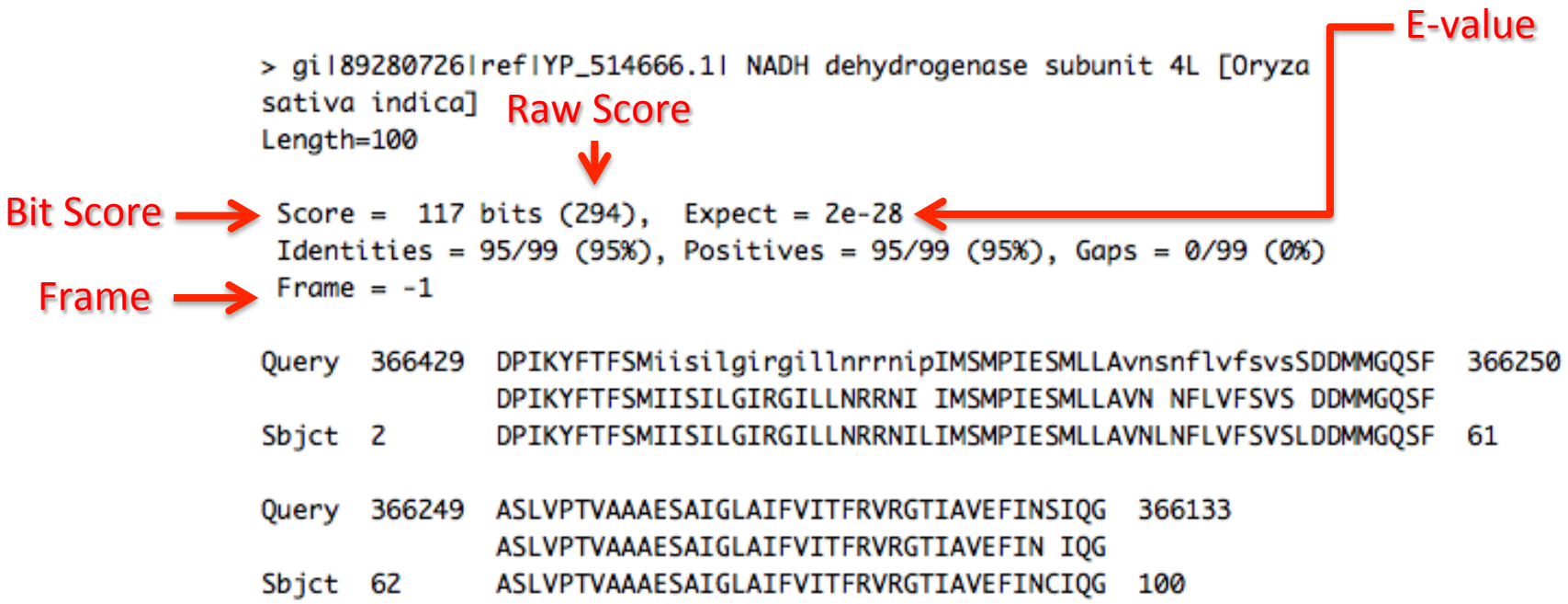
Statistics

```
 Score =  117 bits (294),  Expect = 2e-28
 Identities = 95/99 (95%), Positives = 95/99 (95%), Gaps = 0/99 (0%)
 Frame = -1

Query  366429  DPIKYFTFSMiisilgirgillnrrnipIMSMPIESMLLAvnsnflvfsvsSDDMMGQSF  366250
               DPIKYFTFSMIISILGIRGILLNRRNI IMSMPIESMLLAVN NFLVFSVS DDMMGQSF
Sbjct  2       DPIKYFTFSMIISILGIRGILLNRRNILIMSMPIESMLLAVNLNFLVFSVSLDDMMGQSF  61

Query  366249  ASLVPTVAAAESAIGLAIFVITFRVRGTIAVEFINSIQG  366133
               ASLVPTVAAAESAIGLAIFVITFRVRGTIAVEFIN IQG
Sbjct  62      ASLVPTVAAAESAIGLAIFVITFRVRGTIAVEFINCIQG  100
```

BLASTX report: ALIGNMENTS



BLAST scores

1. **Raw score** – based on match/mismatch or substitution scores
    - bigger is better
    - changes a lot with parameters, but not with database

2. **Bit score** – rescaled and normalized raw scores
    - bigger is better
    - normalized for the particulars of the scoring system
    - changes a little with parameters, but not with database

BLAST scores

3. **Expect (E) value** – the number of hits one can *expect* to find by chance, i.e., the random background noise

★ Depends on query length, database size and scoring matrix

$$E = mn2^{-S'}$$

m=query length, n=database length, S' = score

- analogous to the statistical probability of the hit
- decreases exponentially as the score of the match increases
- lower is better
- $E = 1e^{-6}$ *means* "in this database, I'd expect to find 1 in a million hits with a similar score simply by chance"
- E = 0.025, score would be found by chance 2.5 times in 100
- $E \leq 0.05$ technically statistically significant, BUT
  Short queries can't get a high score or low E
  Huge database! In practice $E \leq 10^{-5}$ is a common cutoff

Hands-on Exercise

1. Open watermelon_blast_statistics.xlsx
2. BLASTN searches of nad4L
    - Half use "Highly Similar" BLAST parameters
    - Half use "Somewhat Similar" BLAST parameters
3. Record the database sizes and scores
4. BLASTX the watermelon mt genome (nucleotide query) to a database of watermelon proteins (amino acids)

	A	B	C	D	E	F	G
1	BLASTN (Highly similar): word_size=28, reward=1, penalty=-2, gapopen=0, gapextend=2.5 [default]						
2							
3	Query	Query size (nt)	Database	Database size (nt)	Top hit		
4					Raw score	bitscore	E-value
5	Citrullus nad4L	303	Citrullus nad4L gene	303	303	560	1E-164
6	Citrullus nad4L	303	Citrullus nad genes	8970	303	560	4E-163
7	Citrullus nad4L	303	Citrullus mt genes	38,948	303	560	2E-162
8	Citrullus nad4L	303	Citrullus mt genome	379,236	303	560	2E-161
9	Citrullus nad4L	303	Plant mt genomes	9,356,449	303	560	4E-160
10	Citrullus nad4L	303	GenBank (nr)	**43,111,105,184**	303	560	2E-156
11				NEW?GenBank (nr) db #: 51561019060			
12							
13	BLASTN (Somewhat similar): word_size=11, reward=2, penalty=-3, gapopen=5, gapextend=2						
14							
15	Query	Query size (nt)	Database	Database size (nt)	Top hit		
16					Raw score	bitscore	E-value
17	Citrullus nad4L	303	Citrullus nad4L gene	303	606	547	1E-160
18	Citrullus nad4L	303	Citrullus nad genes	8970	606	547	3E-159
19	Citrullus nad4L	303	Citrullus mt genes	38,948	606	547	1E-158
20	Citrullus nad4L	303	Citrullus mt genome	379,236	606	547	1E-157
21	Citrullus nad4L	303	Plant mt genomes	9,356,449	606	547	3E-156
22	Citrullus nad4L	303	GenBank (nr)	43,111,105,184	606	547	2E-152

Note! Command-line BLAST reports “Raw” and “Bit” Scores, whereas online BLAST reports “Max” and “Total” Scores. Max and Bit scores are equivalent!

Using The Command Line: A Very Short Guide

To start:

MAC/Unix: Finder > Go > Utilities > Terminal

`PC: Start> Run> Cmd`

See the Cheat Sheet posted on the module website as well!

Navigating Around Your Computer with the Command Line

`$ cd` – change directory

- `$ cd ..` [move up one directory]
- `$ cd ../../` [move up two directories]
- `$ cd` [go home]
- `$ cd ~` [go home]
- `$ cd ~/Desktop/watermelon_files`
- `$ cd /Users/aja/Desktop/watermelon_files`

For Windows:

cd C:\Program Files\NCBI

AWESOME SHORTCUTS

1. Auto-complete your paths with `tab`
2. Re-run a previous command with Up [and down] arrow

Mac OS:

1. Find an old command with Ctrl+r
2. Find an old command with `history`
3. Open working directory in Finder with `open .`

Viewing directory contents with `ls`

You have options!

`-a` : list all directory contents, including hidden files **`(dir \aa)`**

`-l` : list in long format

`-h` : make file sizes human readable (only with `-l`)

`-t` : sort by time modified (most recently modified first)

`-s` : sort files by size

```
$ cd ~/Desktop/watermelon_files
$ ls -al
$ ls -alS
$ ls –alSh
```

For Windows:

ls = dir

`ls`/`dir` output

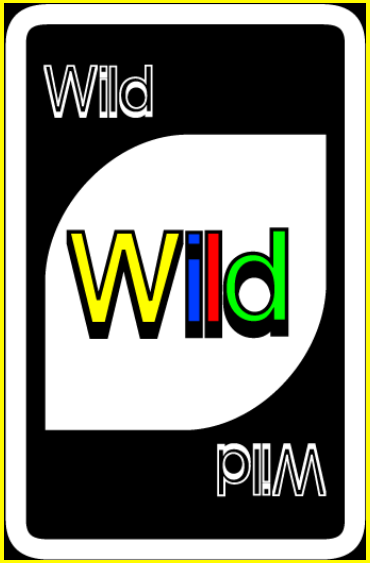
```
total 1208
-rw-r--r--@  1 aja  staff   371K Jan  8 20:14 watermelon.fsa
-rw-r--r--@  1 aja  staff   172K Jan  8 20:17 watermelon.gff.docx
-rw-r--r--   1 aja  staff    20K Jan  8 20:14 watermelon.gff
-rw-r--r--@  1 aja  staff    20K Jan  8 20:25 watermelon.mac.gff
drwxr-xr-x  62 aja  staff   2.1K Jan  8 20:14 watermelon_nt
drwxr-xr-x   9 aja  staff   306B Jan  8 20:29 .
drwxr-xr-x   8 aja  staff   272B Jan  8 20:15 ..
-rw-r--r--@  1 aja  staff   228B Jan  8 20:30 watermelon_genes.mac.txt
-rw-r--r--@  1 aja  staff   228B Jan  8 20:29 watermelon_genes.unix.txt
```

file name

* wildcards *

* the wildest of all wildcards, representing any number of any character (except a slash)

```
$ cd ~/Desktop/watermelon_files
$ ls *.txt
$ ls -al *.txt
$ ls -al *genes*
$ ls -al */sdh*
```



Common commands

`$ clear` – clear the Terminal screen

For Windows:
clear = CLS

`$ exit` – exit the session

- always exit your session before closing the Terminal window and quitting Terminal
- not doing so is like unplugging your computer

For Windows:
exit = exit

Copying and moving files

```
$ cp source file target file – copy files
   $ cp watermelon.fsa copy.fasta
   $ cp watermelon.fsa ~/Documents/copy.fasta

$ mv source file target file – move/rename files
   $ mv watermelon.fsa junk.dat
   $ mv watermelon.fsa /tmp/tmp.fas
   $ mv watermelon_files /tmp
   $ mv /tmp/watermelon_files/ .
```

For Windows:

cp = copy

mv = move

Making directories (folders)

`$ mkdir directory-name ...` – make directories

```
$ mkdir cowgirl
$ mkdir cowgirl in the sand
$ ls –al
```

For Windows:

mkdir = mkdir

Deleting stuff

- Removing files

  `$ rm filename ...` – remove files

  ```
  $ rm ~/Documents/copy.fasta
  $ rm junk.dat
  ```

- Removing files and directories recursively

  ```
  $ rm -r cowgirl
  ```

  - removes all the files in `cowgirl`, then removes `cowgirl` itself
  - dangerous command (that I use all the time)

For Windows:
rm = del

Really important warnings

- Unix/command line commands are permanent.
- There is no Recycle bin.
- There are usually no warnings.
- If you **overwrite** a file, it's gone forever.
- If you **delete** a file, it's gone forever.

Concatenate and Print files

If a colleague asked you to send her all the files you are working with (separately), how would you do it?

How long would it take for one species? 5 species? 100 species? 1,000 species?

`$ cat` – concatenate and print files

```
$ cat watermelon_aa/nad1.fasta
$ cat watermelon_aa/nad*
```

For Windows:

cat = type

Redirecting output

Use `>` to redirect output to a file instead of the screen.

Write to a file

```
$ cat watermelon_aa/nad*.fasta > nad_genes.fasta
```

Append to an existing file

```
$ cat watermelon_nt/nad*.fasta >> nad_genes.fasta
```

If you redirect (single '>') to an existing file, it will completely **overwrite** it **without** warning.

For Windows:

cat = type