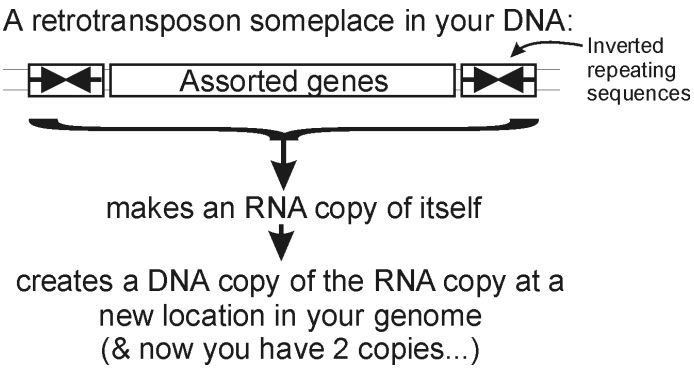
You and your (DNA) parasites



Events like these, happening over and over again, have led to…

You and your (DNA) parasites

Major types of repeats in the human genome

			Length	Copies	Fraction of genome
LINEs	Autonomous	ORF1 ORF2 (pol) AAA	6-8 kb	850,000	21%
SINEs	Non-autonomous	A B AAA	100-300 bp	1,500,000	13%
Retrovirus-like elements	Autonomous	gag pol (env)	6-11 kb	450,000	8%
	Non-autonomous	(gag)	1.5-3 kb		
DNA transposon fossils	Autonomous	transposase	2-3 kb	300,000	3%
	Non-autonomous		80-3,000 bp		
					~45%

Bottom line: Roughly half of your (and my) genome is the fossil wreckage of genomic parasites.

We know this (in part) from sequence alignments.

So far, we've talked about

- DNA, RNA (or rather, not RNA), and protein sequences
- How to compare sequences to decide if they are related
- Having databases full of sequences and comparing them rapidly (BLAST)

In fact, <u>many</u> such databases exist, so today we'll start with a brief tour of <u>some</u> of the biological data on the web.

Just some of the resources available for bioinformatics

Think of these as the raw data for new discoveries

Database	Records	Address
BioGRID	>2 M protein interactions	https://thebiogrid.org
EcoCyc/MetaCyc	>2,700 pathways from >3,000 organisms	http://www.ecocyc.org, http://www.metacyc.org
Ensembl (+ BioMart for easy sequence queries)	Major repository of DNA sequences, genomes, genes, proteins, and transcripts	http://useast.ensembl.org/index.html
Entrez Genome	Thousands of genome sequences	http://www.ncbi.nlm.nih.gov/genome?db=genome
Expression Atlas	139K mRNA expression expts in 65 species	https://www.ebi.ac.uk/gxa/home/
Genbank	>1 trillion bases sequenced; > 14 trillion bases as whole genome shotgun data	https://www.ncbi.nlm.nih.gov/genbank/
Gene Expression Omnibus (GEO)	>4 M mRNA or protein expression expts	http://www.ncbi.nlm.nih.gov/geo/
Genomes Online Database (GOLD)	>180K genome sequences, many in progress	https://gold.jgi.doe.gov/index
Human Protein Atlas	millions of high-res images of ~17K human proteins across tissues, cancers, & cell lines	http://www.proteinatlas.org/
KEGG	Most known pathways, in 548 graphical diagrams and >7K organisms (*via* homology)	http://www.genome.ad.jp/kegg/
Medline / PubMed	>30 million references	https://www.ncbi.nlm.nih.gov/PubMed/
Mouse Genome Informatics	~20,000 mouse genes, diverse associated data & annotations	http://www.informatics.jax.org/
Online Mendelian Inheritance in Man (OMIM)	Compendium of human genes and genetic phenotypes, data for >16,000 human genes	https://www.ncbi.nlm.nih.gov/omim/
Pride	Hundreds of millions of peptide mass spectra from 10's of thousands of experiments	https://www.ebi.ac.uk/pride/archive/
Reactome	>2K pathways involving >10K human proteins, also other organisms	https://www.reactome.org/
SGD	~6,000 yeast genes, diverse associated data & annotations	https://www.yeastgenome.org/
UniProtKB/SWISS-PROT	>550K hand-curated sequence entries from >14K organisms	https://www.uniprot.org/

Just some of the resources available for bioinformatics

Think of these as the raw data for new discoveries

Database	Record	
BioGRID	>2 M protein interactions	
EcoCyc/MetaCyc	>2,700 pathways from >3,00	
Ensembl (+ BioMart for easy sequence queries)	Major repository of DNA se genes, proteins, and transcripts	
Entrez Genome	Thousands of genome sequences	http://www.ncbi.nlm.nih.gov/genome?db=genome
Expression Atlas	121K mRNA expression expts in 62 species	
Genbank	>386 billion bases sequenced; > 5 trillion bases as whole genome shotgun data	
Gene Expression Omnibus (GEO)	>4 M mRNA or protein expressi expts	
Genomes Online Database (GOLD)	>150K genome sequences, many in progress	
Human Protein Atlas		http://www.proteinatlas.org/
KEGG		htt
Medline / PubMed		htt
Mouse Genome Informatics	~20,000 mouse genes, diverse associated data & annotations	htt
Online Mendelian Inheritance in Man (OMIM)	Compendium of human genes and genetic phenotypes, data for >16,000 human genes	htt
Pride	Hundred from 10's	
Reactome	>2K path also othe	
SGD	~6,000 ye annotati	
UniProtKB/SWISS-PROT	>550K ha organisms	

Biogrid has >2 M protein-protein interactions (https://thebiogrid.org/)

GEO has millions of experiments, each measuring 1000's of mRNA or protein abundances

Medline has >30 million research articles, many with complete text online

OMIM = the most important resource for human genetic disease

Uniprot = a frequent first step to learn about genes. **Also <u>amazingly useful</u> for interconverting IDs and linking to other resources**

Live demo Ensembl->BioMart->filter for [IPR031588], OMIM, Reactome, Human Protein Atlas

It's nice to know that all of this exists, but ideally, you'd like to be able to so something constructive with the data.

That means getting the data inside your own programs.

All of these databases let you download data in big batches, but this isn't always the case, so….

Let's empower your Python scripts to grab data from the web.

We'll use Python library/module = an optional, specialized set of Python methods

This particular Python module is called ***urllib*** (Py3) or ***urllib2*** (Py2)

urllib/urllib2 is:

- A collection of programs/tools to let you to surf the web from inside your programs.
- Much more powerful than the simple tasks we'll do with it.
- More details:
  https://docs.python.org/3.8/library/urllib.request.html
  or http://docs.python.org/2/library/urllib2.html

The basic idea:

We first set up a “request” by opening a connection to the URL.

We then save the response in a variable and print it.

If it can’t connect to the site, it’ll print out a helpful error message instead of the page.

You can more or less use the commands in a cookbook fashion….

For example:

```
import urllib.request                    # include the urllib.request module

url = "https://www.utexas.edu/"

x = urllib.request.urlopen(url)          # setup a request
print(x.read())                          # read page and show the result to the user
```

Python 3 version

We can be slightly fancier in order to handle different formats and the inevitable internet connection errors

```
import urllib.request                    # include the urllib.request module

url = "https://www.utexas.edu/"

try:                    # this 'try' statement tells Python that we might expect an error.
    request = urllib.request.urlopen(url)   # setup a request
    page = request.read().decode('utf-8')   # save the response
    print(page)                             # show the result to the user

except urllib.error.URLError:               # handle a page not found error
    print("Could not find page.")
```

→ **Run this…**

Python 3 version

(Heres' the Python 2 version in case you need it)

```
import urllib2                           # include the urllib2 module

url = "https://www.utexas.edu/"

try:                    # this 'try' statement tells Python that we might expect an error.
    request = urllib2.urlopen(url)          # setup a request
    page = request.read()                   # save the response
    print(page)                             # show the result to the user

except urllib2.URLError:                    # handle a page not found error
    print("Could not find page.")
```

→ **Run this…**

Python 2 version

→ **We just captured the UT web page and printed it out…**

```
>>>
<!doctype html>
<html lang="en" dir="ltr">

<head>
….
<meta name="apple-mobile-web-app-title" content="UT Austin" />
<meta name="description" content="The University of Texas at Austin is a bold, ambitious leader, providing a first-class education and the tools of discovery to more than 51,000 students." />….
```

…and so on, and on, and on…

That was (more or less) a static web page.

Let's try one that requires some sort of action, for example by entering a document id or an id code for a sequence.

Many web pages pass this information along in the web URL itself…

Here's a complete Python program to retrieve a single entry from Medline:

```
import urllib.request
pmid = 11237011

# Insert the pmid where the {} are in the following URL:
url = "https://pubmed.ncbi.nlm.nih.gov/?term={0}[uid]&format=pubmed".format(pmid)

try:                                # there might be an error!
    request = urllib.request.urlopen(url)
    page = request.read().decode('utf-8')
    print(page)

except urllib.error.URLError:       # handle page not found error
    print("Could not connect to Medline!")
```

Python 3 version

Here's a complete Python program to retrieve a single entry from Medline:

```
import urllib2
pmid = 11237011

# Insert the pmid where the {} are in the following URL:
url = "https://pubmed.ncbi.nlm.nih.gov/?term={0}[uid]&format=pubmed".format(pmid)

try:                                # there might be an error!
    request = urllib2.urlopen(url)
    page = request.read()
    print(page)

except urllib2.URLError:            # handle page not found error
    print("Could not connect to Medline!")
```

Python 2 version

If you run that program, you should get back…

```
>>>
<!DOCTYPE html>

.....lots of metadata.....

OWN - NLM
STAT- MEDLINE
DCOM- 20010322
LR  - 20210108
IS  - 0028-0836 (Print)
IS  - 0028-0836 (Linking)
VI  - 409
IP  - 6822
DP  - 2001 Feb 15
TI  - Initial sequencing and analysis of the human genome.
PG  - 860-921
AB  - The human genome holds an extraordinary trove of information about human
      development, physiology, medicine and evolution. Here we report the results of an
      international collaboration to produce and make freely available a draft sequence of
      the human genome. We also present an initial analysis of the data, describing some
      of the insights that can be gleaned from the sequence.
FAU - Lander, E S
AU  - Lander ES
AD  - Whitehead Institute for Biomedical Research, Center for Genome Research, Cambridge,
      MA 02142, USA. lander@genome.wi.mit.edu

      [and so on]
```

the Medline entry for the human genome sequence paper

If you run that program, you should get back…

```
>>>
<!DOCTYPE html>

.....lots of metadata.....

OWN - NLM
STAT- MEDLINE
DCOM- 20010322
LR  - 20210108
IS  - 0028-0836 (Print)
IS  - 0028-0836 (Linking)
VI  - 409
IP  - 6822
DP  - 2001 Feb 15
TI  - Initial sequencing and analysis of the human genome.
PG  - 860-921
AB  - The human genome holds an extraordinary trove of information about human
      development, physiology, medicine and evolution. Here we report the results of an
      international collaboration to produce and make freely available a draft sequence of
      the human genome. We also present an initial analysis of the data, describing some
      of the insights that can be gleaned from the sequence.
FAU - Lander, E S
AU  - Lander ES
AD  - Whitehead Institute for Biomedical Research, Center for Genome Research, Cambridge,
      MA 02142, USA. lander@genome.wi.mit.edu

      [and so on]
```

We just printed it. We could have saved it or extracted data from it. For example…

Here's our Python program again to retrieve a single entry from Medline. How would we modify this to count the authors?

```
import urllib.request
pmid = 11237011

# Insert the pmid where the {} are in the following URL:
url = "https://pubmed.ncbi.nlm.nih.gov/?term={0}[uid]&format=pubmed".format(pmid)

try:                                # there might be an error!
    request = urllib.request.urlopen(url)
    page = request.read().decode('utf-8')
    print(page)

except urllib.error.URLError:       # handle page not found error
    print("Could not connect to Medline!")
```

Python 3 version

Here's our Python program again to retrieve a single entry from Medline. How would we modify this to count the authors?

```
import urllib.request
pmid = 11237011

# Insert the pmid where the {} are in the following URL:
url = "https://pubmed.ncbi.nlm.nih.gov/?term={0}[uid]&format=pubmed".format(pmid)

try:                                # there might be an error!
    request = urllib.request.urlopen(url)
    page = request.read().decode('utf-8')
    print(page.count("AU  - "))

except urllib.error.URLError:       # handle page not found error
    print("Could not connect to Medline!")
```

Medline begins author lines with "AU - " , so…

→ **Run this, & get …**

```
>>>
256
```

So, there were 256 authors on one (of the two) human genome papers

Python 3 version

(& the Python 2 version, just for the sake of completeness)

```
import urllib2
pmid = 11237011

# Insert the pmid where the {} are in the following URL:
url = "https://pubmed.ncbi.nlm.nih.gov/?term={0}[uid]&format=pubmed".format(pmid)

try:                                # there might be an error!
    request = urllib2.urlopen(url)
    page = request.read()
    print(page.count("AU  - "))

except urllib2. URLError:       # handle page not found error
    print("Could not connect to Medline!")
```

Python 2 version

- Queries to Medline or any other NCBI database, including GenBank, are described at:
  http://www.ncbi.nlm.nih.gov/books/NBK3862/
  (& for that matter, all of medline is downloadable)

- You can often figure out the form of the URL just by looking something up in a database, then noting the address of the web page with the data.

- This very simple approach could easily be the basis for:
  - a home-made web browser
  - a program to consult biological databases in real time
  - a program to map the internet, etc.

- Of course, with this kind of power available to you, the imagination reels...

A note about the Rosalind homework & BioPython

- URLLIB works with many web pages, but for bio databases, it's often easier to use **BioPython**
- BioPython lets you access sequence & structure databases, read fasta/genome files, do simple sequence analyses, BLAST, etc, right from your Python code
- If you need to install it, just open an Anaconda prompt (on a PC) or launch a console window from Anaconda Navigator & type "pip install biopython"

e.g.

```
from Bio import Entrez
Entrez.email = "your_email@gmail.com"    # Always tell NCBI who you are
handle = Entrez.efetch(db="nucleotide", id="EU490707", rettype="gb", retmode="text")
print(handle.read())
```

```
LOCUS       EU490707               1302 bp    DNA     linear   PLN 26-JUL-2016
DEFINITION  Selenipedium aequinoctiale maturase K (matK) gene, partial cds;
            chloroplast.
ACCESSION   EU490707
VERSION     EU490707.1
KEYWORDS    .
SOURCE      chloroplast Selenipedium aequinoctiale
  ORGANISM  Selenipedium aequinoctiale
.....
ORIGIN
        1 atttttacg aacctgtgga aatttttggt tatgacaata aatctagttt agtacttgtg
       61 aaacgtttaa ttactcgaat gtatcaacag aatttttga tttcttcggt taatgattct  ....
```

There's a complete pdf tutorial @ http://biopython.org/DIST/docs/tutorial/Tutorial.pdf