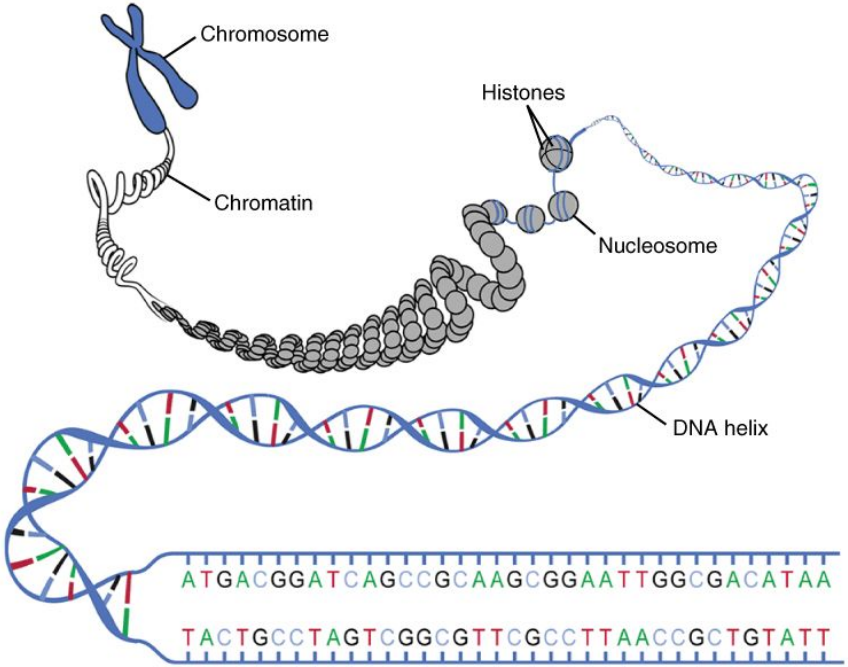
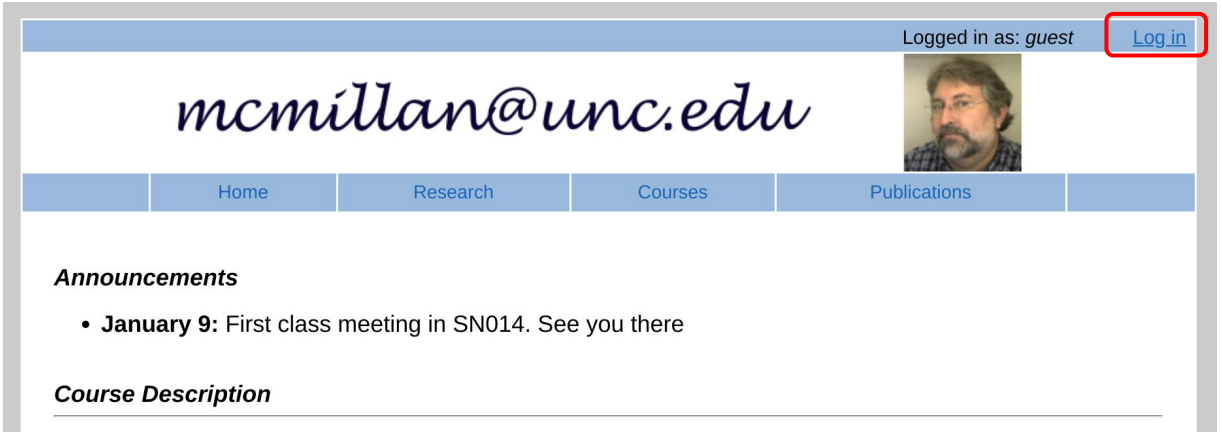
Comp 555 - BioAlgorithms - Spring 2021



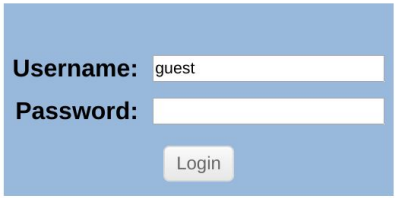
Finding Patterns in DNA

Login to Course Website

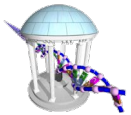
1) Login to your Comp555 account



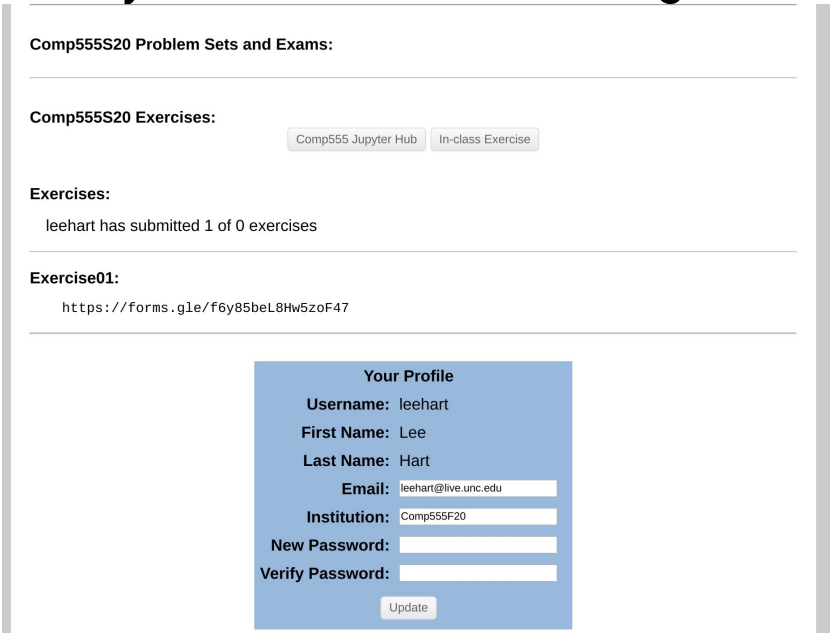
2) Your username is your UNC ONYEN and password is your PID



Next Steps



3) Once you are logged in, press "Course" and then a “Setup” button should appear. Press "Setup" and you should see something like:



4) (BTW, you can also change your password here if you want).

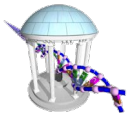
For those without a login...

- Go back to the login page, and click "registered"

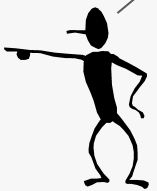
No password is required to logon as "guest"
You must be registered to have full access or modify content.

- Then enter the following information:
- Once registered a screen will indicate you've been verified; then click "Course" and "Setup" as before.

You've seen a genome… let's scale up

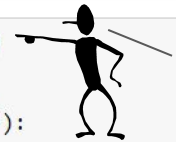


In [15]:
```
def loadFasta(filename):
    """ Parses a classically formatted and possibly
        compressed FASTA file into a list of headers
        and fragment sequences for each sequence contained"""
    if (filename.endswith(".gz")):
        fp = gzip.open(filename, 'r')
    else:
        fp = open(filename, 'r')
    # split at headers
    data = fp.read().split('>')
    fp.close()
    # ignore whatever appears before the 1st header
    data.pop(0)
    headers = []
    sequences = []
    for sequence in data:
        lines = sequence.split('\n')
        headers.append(lines.pop(0))
        # add an extra "+" to make string "1-referenced"
        sequences.append('+' + ''.join(lines))
    return (headers, sequences)
```



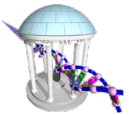
This is the same FASTA format file parser from last lecture

In [19]:
```
header, seq = loadFasta("GCA_000001405.15_GRCh38_genomic.fna")
print(len(header), "sequences")
for i in range(len(header)):
    if header[i].startswith("CM") or header[i].startswith("J0"):
        print(header[i])
        print(len(seq[i])-1, "bases", seq[i][:30], "...", seq[i][-30:])
```



This is a recent version of the human genome. But, we're only going to look at a subset of it.

What's inside



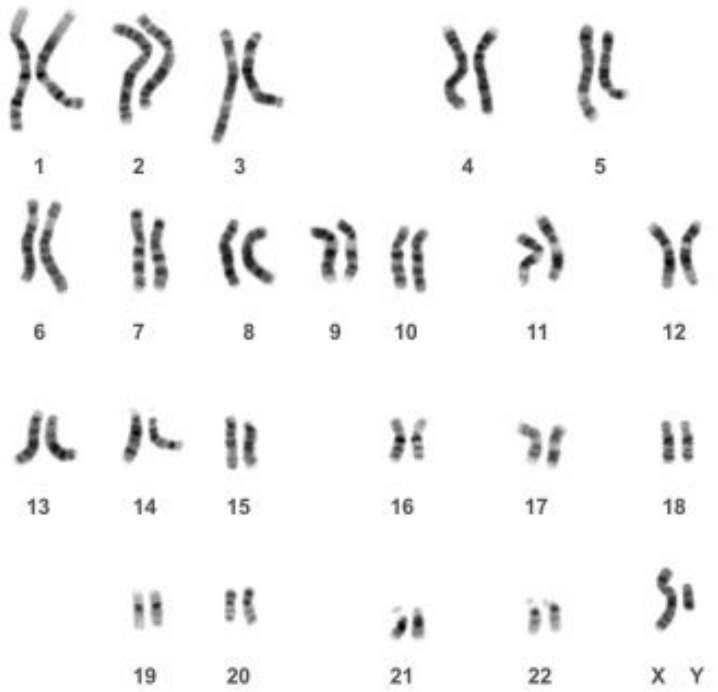
Human DNA

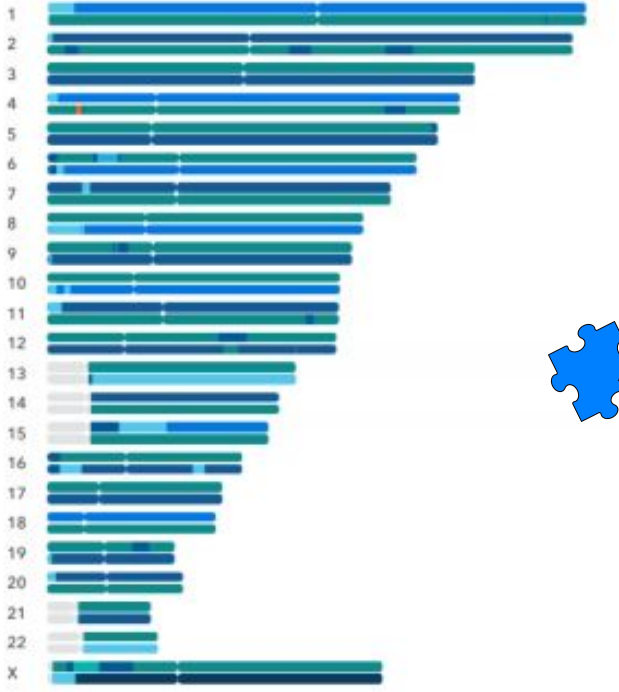
- Is distributed across 23 primary chromosomes named 1-22, an X.
- We actually have two copies of of the numbered chromosomes, and two copies of X if you are female or one X and one Y if you are male.
- There is also a short DNA sequence that appears in a cell organelles called mitochondria.
- There are also 430 other "unmapped" sequences

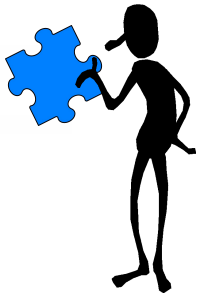
```
455 sequences
CM000663.2 Homo sapiens chromosome 1, GRCh38 reference primary assembly
248956422 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000664.2 Homo sapiens chromosome 2, GRCh38 reference primary assembly
242193529 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000665.2 Homo sapiens chromosome 3, GRCh38 reference primary assembly
198295559 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000666.2 Homo sapiens chromosome 4, GRCh38 reference primary assembly
190214555 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000667.2 Homo sapiens chromosome 5, GRCh38 reference primary assembly
181538259 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000668.2 Homo sapiens chromosome 6, GRCh38 reference primary assembly
170805979 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000669.2 Homo sapiens chromosome 7, GRCh38 reference primary assembly
159345973 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000670.2 Homo sapiens chromosome 8, GRCh38 reference primary assembly
145138636 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000671.2 Homo sapiens chromosome 9, GRCh38 reference primary assembly
138394717 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000672.2 Homo sapiens chromosome 10, GRCh38 reference primary assembly
133797422 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000673.2 Homo sapiens chromosome 11, GRCh38 reference primary assembly
135086622 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000674.2 Homo sapiens chromosome 12, GRCh38 reference primary assembly
133275309 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000675.2 Homo sapiens chromosome 13, GRCh38 reference primary assembly
114364328 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000676.2 Homo sapiens chromosome 14, GRCh38 reference primary assembly
107043718 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000677.2 Homo sapiens chromosome 15, GRCh38 reference primary assembly
101991189 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000678.2 Homo sapiens chromosome 16, GRCh38 reference primary assembly
90338345 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000679.2 Homo sapiens chromosome 17, GRCh38 reference primary assembly
83257441 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000680.2 Homo sapiens chromosome 18, GRCh38 reference primary assembly
80373285 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000681.2 Homo sapiens chromosome 19, GRCh38 reference primary assembly
58617616 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000682.2 Homo sapiens chromosome 20, GRCh38 reference primary assembly
64444167 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000683.2 Homo sapiens chromosome 21, GRCh38 reference primary assembly
46709983 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000684.2 Homo sapiens chromosome 22, GRCh38 reference primary assembly
50818468 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000685.2 Homo sapiens chromosome X, GRCh38 reference primary assembly
156040895 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
CM000686.2 Homo sapiens chromosome Y, GRCh38 reference primary assembly
57227415 bases +NNNNNNNNNNNNNNNNNNNNNNNNNNN ... NNNNNNNNNNNNNNNNNNNNNNNNNNN
J01415.2 Homo sapiens mitochondrion, complete genome
16569 bases +GATCACAGGTCTATCACCCTATTAACCAC ... CACGTTCCCCTTAAATAAGACATCACGATG
```

Missing Puzzle Pieces

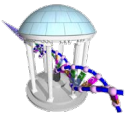
There are still missing, partially assembled, pieces, that we don't yet know where they are placed.







What it's sequence looks like



As with SARS-CoV-2, we can get some insights into a genome by examining its k-mer distributions. But before we start, let's look consider the genome's size?

- **In total, there are 3,272,116,950 base pairs in the primary (forward) sequence**
- **Many of these are unknown, and are indicated by 'N'**
- **There are also a small number of ambiguous bases indicated using a standard called UIPAC**

Chromosome lengths | Total lengths | Ungapped lengths | N50s | Gaps | Counts

Chromosome lengths are calculated by summing the length of the placed scaffolds and estimated gaps.

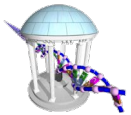
Chromosome	Total length (bp)	GenBank accession	RefSeq accession
1	248,956,422	CM000663.2	NC_000001.11
2	242,193,529	CM000664.2	NC_000002.12
3	198,295,559	CM000665.2	NC_000003.12
4	190,214,555	CM000666.2	NC_000004.12
5	181,538,259	CM000667.2	
6	170,805,979	CM000668.2	
7	159,345,973	CM000669.2	
8	145,138,636	CM000670.2	
9	138,394,717	CM000671.2	
10	133,797,422	CM000672.2	
11	135,086,622	CM000673.2	
12	133,275,309	CM000674.2	
13	114,364,328	CM000675.2	
14	107,043,718	CM000676.2	
15	101,991,189	CM000677.2	
16	90,338,345	CM000678.2	
17	83,257,441	CM000679.2	
18	80,373,285	CM000680.2	
19	58,617,616	CM000681.2	
20	64,444,167	CM000682.2	
21	46,709,983	CM000683.2	
22	50,818,468	CM000684.2	
X	156,040,895	CM000685.2	
Y	57,227,415	CM000686.2	NC_000024.10

IUPAC degenerate base symbols[2]

Description	Symbol	Bases represented No.	A	C	G	T	Complementary bases[a]
Adenine	A	1	A				T
Cytosine	C			C			G
Guanine	G				G		C
Thymine	T					T	A
Uracil	U					U	A
Weak	W	2	A			T	W
Strong	S			C	G		S
Amino	M		A	C			K
Keto	K				G	T	M
Purine	R		A		G		Y
Pyrimidine	Y			C		T	R
Not A[b]	B	3		C	G	T	V
Not C[b]	D		A		G	T	H
Not G[b]	H		A	C		T	D
Not T[b]	V		A	C	G		B
Any one base	N	4	A	C	G	T	N
Zero	Z	0					Z

a. ^ I.e., here, read the represented bases in reverse.
b. ^ [a] [b] [c] [d] Represented by the letter following (excluding U).

Let's reformat our sequences



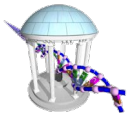
It's a little annoying to load a series of sequences from FASTA files over and over again. Especially when we will mostly deal with a subset, and of those we will consider them one at a time.

So I decided to write out each sequence as a single string to its own file.

In [ ]:
```
header, seq = loadFasta("data/GCA_000001405.15_GRCh38_genomic.fna")
print(len(header), "sequences")
for i in range(len(header)):
    if header[i].startswith("CM") or header[i].startswith("J0"):
        start = header[i].find('chromosome ')
        chromo = header[i][start+11:header[i].find(',')] if (start >= 0) else "MT"
        with open("data/Chr%s.seq" % chromo, 'w') as fp:
            fp.write(seq[i])
```

You might want to wait and do this later.

A quick helpful function



- DNA is actually two sequences, a primary and reverse-complement version
- Genomes report only one (the primary one), and the reverse complement version can be derived from it.
- When we consider k-mers, in most cases, we don't care which of the sequences they come from
- Here's a simple function that maps back and forth

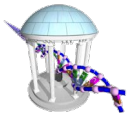
In [ ]:
```
def revComp(dnaSeq):
    return ''.join([{'A':'T','C':'G','G':'C','T':'A'}[base] for base in reversed(dnaSeq)])
```

- Here's an example:

In [4]:
```
print(revComp("GAGACAT"))
print(revComp("ATGTCTC"))
```

```
ATGTCTC
GAGACAT
```

Let's consider some k-mer statistics



For what value of k?

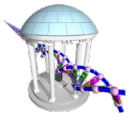
```
In [ ]: import time

chromo = [str(i) for i in xrange(1,23)] + ['X', 'Y', 'MT']

kmerCount = {}
K = 11
L = 0
for contig in chromo:
    tick = time.time()
    with open("Chr%s.seq" % contig, 'r') as fp:
        seq = fp.read()
    for i in xrange(1,len(seq)-K+1):
        kmer = seq[i:i+K]
        for base in "RYSWKMBDHVN":
            if (base in kmer):
                break
        else:
            kmerCount[kmer] = kmerCount.get(kmer,0) + 1
            kmer = revComp(kmer)
            kmerCount[kmer] = kmerCount.get(kmer,0) + 1
    tock = time.time()
    print(contig, len(seq)-1, len(kmerCount), "%6.2f secs" % (tock - tick))
    tick = tock
    L += len(seq) - 1
print(L, len(kmerCount))
```

A "for-else" statement. Have you seen one of those before?

This is similar to our kmer counter from last lecture, except every k-mer is consider twice. Once as it appears, and once as it's reverse complement.

We'll also skip over any k-mer with an 'N' or with one of those strange IUPAC bases.

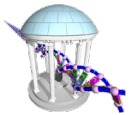
DON'T RUN IT!



```
'1', 248956422, 4133410, '1388.88 secs'
'2', 242193529, 4175438, '1364.77 secs'
'3', 198295559, 4184312, '1064.28 secs'
'4', 190214555, 4188228, '1155.32 secs'
'5', 181538259, 4190446, '1031.80 secs'
'6', 170805979, 4191700, '930.77 secs'
'7', 159345973, 4192490, '1025.24 secs'
'8', 145138636, 4192908, '931.29 secs'
'9', 138394717, 4193190, '751.69 secs'
'10', 133797422, 4193464, '827.02 secs'
'11', 135086622, 4193648, '780.05 secs'
'12', 133275309, 4193788, '724.03 secs'
'13', 114364328, 4193862, '635.01 secs'
'14', 107043718, 4193926, '534.36 secs'
'15', 101991189, 4193988, '530.65 secs'
'16', 90338345, 4194048, '478.42 secs'
'17', 83257441, 4194088, '447.81 secs'
'18', 80373285, 4194110, '448.31 secs'
'19', 58617616, 4194136, '345.26 secs'
'20', 64444167, 4194158, '363.14 secs'
'21', 46709983, 4194166, '251.95 secs'
'22', 50818468, 4194182, '253.91 secs'
'X', 156040895, 4194200, '866.71 secs'
'Y', 57227415, 4194200, '189.44 secs'
'MT', 16569, 4194200, '  0.10 secs'
3088286401, 4194200
```

- It takes a while to run (there are actually faster ways to do this!)
- 1000 secs is around 16 minutes
- And we still don't see every possible 11-mer

What does the distribution look like?

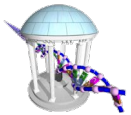


In [14]:
```
import matplotlib
import matplotlib.pyplot as plot
%matplotlib inline

# Compute a histogram of kmerCount (i.e. how many kmers appear 1 time, 2 times, 3 times ...)
maxcount = 1000
hist = [0 for i in range(maxcount)]
for kmer in kmerCount:
    count = kmerCount[kmer]
    if (count < maxcount):
        hist[count] += 1

fig = plot.figure(figsize=(16,4))
plot.plot([i for i in range(maxcount)], hist)
plot.show()
```

It looks like the sum of at least two normal distributions.

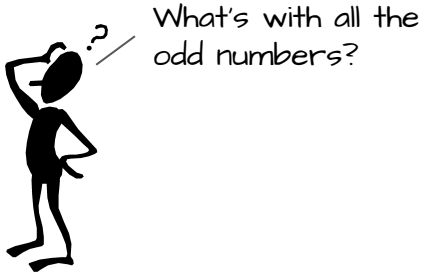
One with a mode around 20 and a second with a mode near 125.

What could we learn from other values of k

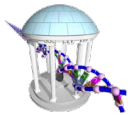


- Our genome includes every possible 11-mer
- How large should k be so that we'd expect most k-mers to be unique?
- Recall the genome has 3,272,116,950 bases

There are 4,194,304 11-mers
There are 67,108,864 13-mers
There are 1,073,741,824 15-mers
There are 17,179,869,184 17-mers
There are 274,877,906,944 19-mers
There are 4,398,046,511,104 21-mers
There are 70,368,744,177,664 23-mers
There are 1,125,899,906,842,624 25-mers
There are 18,014,398,509,481,984 27-mers
There are 288,230,376,151,711,744 29-mers
There are 4,611,686,018,427,387,904 31-mers
There are 73,786,976,294,838,206,464 33-mers
There are 1,180,591,620,717,411,303,424 35-mers
There are 18,889,465,931,478,580,854,784 37-mers
There are 302,231,454,903,657,293,676,544 39-mers
There are 4,835,703,278,458,516,698,824,704 41-mers
There are 77,371,252,455,336,267,181,195,264 43-mers
There are 1,237,940,039,285,380,274,899,124,224 45-mers



While I was bored last night...

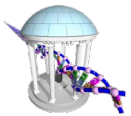


I broke the human genome into non-overlapping 45-mers, and counted how many time each appears in the genome...

TGGCCGAATAGGAACAGCTCCGGTCTACAGCTCCCAGCGTGAGCG|ACGCAGAAGACGGTGATTTCTGCATTTCCATCTGAGGTACCGGGT|TCATCTCACTAGGGAGTGCCAGACAGTGGGCGCAGGCCAGTGTGT|GTGCGCACCGTGCACGAGCCGAAGCAGGGCGAGGCATTGCCTCAC|

CTGGGAAGCGCAAGGGGTCAGGGAGTTCCCTTTCCGAGTCAAAGA|AAGGGGTGACGGTCGCACCTGGAAAATCGGGTCACTCCCACCCGA|ATATTGCGCTTTTCAGACCGGCTTAAGAAACGGCGCACCACGAGA|CTATATCCCACACCTGGCTCGGAGGGTCCTACGCCCACGGAATCT|...

Most places look like this...



```
Chromosome 4
 79935885     1 TCCAGCTGTTGCATAGCTTTGTTAAAGAGTGACACTTAGGCTAAT
 79935930     1 GTACTCTAAGGAAATGACTCCGCTCCCAGTGGAATCTCTCTTCTG
 79935975     1 AAACAATAAATGCCTGTTCCAACAAAAGAGCACCTTAAACTATGA
 79936020     1 TTCCATTCCAAAGTTGTAAAAAAATGGAATTAATTAGGATTTAGC
 79936065     1 AAGTGTACAACCTCTAGCCAGGAGTCATATATTCTAATTTTGAGA
 79936110     1 TATTATTCATAGTTCTCAATGCAGAGAGTTACTACACATTATTTT
 79936155     1 TACTATCAGTACAATACCACTTTTTAAAAGGGTTCAGATGTTTTA
 79936200     1 ATCACTATTACACAAGTACTACAAATGATATAATTAGTTGCATTC
 79936245     1 TTATTTGCAGAATATTTAATTGATCTCTATTCAGATAAATTTTTA
 79936290     1 AATGACAAAATGCTATTTAACTGTCTTATTTTCAGACCTCCTGTC
 79936335     1 ATCAGAGCTTAGCTACTCTCTTTCAAAACCATTACTATTCTCCTT
 79936380     1 CACAACCTAGCACTAAAACTGGTTGCACTGCATTAGAACTCTGCT
 79936425     1 TAGGTACTTCCTATGCAAGTTTTCTCCTTCATCTCACAACCAAAA
 79936470     1 CCTAAATGAACAATCAAAACATTCTCACATTTTTCTAATTCTCCA
 79936515     1 AAAGTTTCCTTTTCTTTCAAGGTAATTTATGTCCTCATAAAAGCT
 79936560     1 TAGGTATAATTCCTGATGGGAGAAATTAGTGAAATATTTCAACAG
 79936605     1 TATTCAAGTTGGTCTCAGGGACTGGGAAATAATGCAAAAGAAATA
 79936650     1 AAAAAAATCCTTTATTAAACGTAAAAGGGAAGAAGAAAAGACTAA
 79936695     1 CACATGAAATAATTAGAAAACAATTAATTTAAAAATTACTGCACA
 79936740     1 ATATTTAAACTAATTCACTTTAAGTTCTGATTATTAATTGCACTA
 79936785     1 TGGGTTATAATCACCTTTTTTGTGCTTATAATCATTCCCACCTAG
 79936830     1 TACAATGGTACTACACTAGTAGTACCACACTAAAGGTACTACACT
 79936875     1 CTACCACTGTACTAACCAATGGTAAAAACTACACTTATATTTTCT
 79936920     1 ATTTTTTTCTATCACTTGTATAATGCTAGGCAGAAAGTCAGCAAG
 79936965     1 CAATGGATAATTGATTATGTATTCATTCAATCATTTTAAAGCATT
 79937010     1 TTTCATTTTAATCTTTGTTGCAAAGAAGAAAAATGTATTAATAAT
 79937055     1 TTTTACTTTTAATAAATACATTAGCATTAACTTTATACAGTTTTA
 79937100     1 AAACACTCATAAACTTAATTAAGCTTTTAATTCAGTTATAAATAG
 79937145     1 GACTCATTCACTGTATTCTCAACAGTAGCATTAAAAAAACCAGGT
 79937190     1 GCCTATTTCATATTCTTAATGAAGCAATTGCTAGCAATAGGAAAA
 79937235     1 CCTCAAAAGATTCACATTTGGCTCAACTAAGTTCCTTGAAAATTA
 79937280     1 ATTACATATAATCATTTAAAACAGCACAAAATTGAGCAGAAGAAA
 79937325     1 AAAAATTTTTGGAAGAATGTTTGTAATATCCATAAATGTTTAGGC
 79937370     1 TAGTTTGGCTGGTTTCTGATTAACTGCATTTGGACATATCTTCAT
 79937415     1 TGAAAGTTTCACTGTAACATACTCACAGAAAGCTTTTTATCTGCA
 79937460     1 AGTGACTTTTTGTGCCACTTGCTTGGGCCACTTTTTCCCAACTCT
 79937505     1 AATTTGCAATTTGTATCTACCCTGAGAGAGGTACTGTCTATCAGG
 79937550     1 GTATATAGTACCATACTCAAACAGATTTGTTCCGTTATCTAAACT
 79937595     1 AGAAATAAATAAATCATAAAATGTTATGTGTCACTAACAAGGTAA
 79937640     1 CAACTTGAATGCTTATGTATATATTGAGCATCAATTATGTACCCA
```

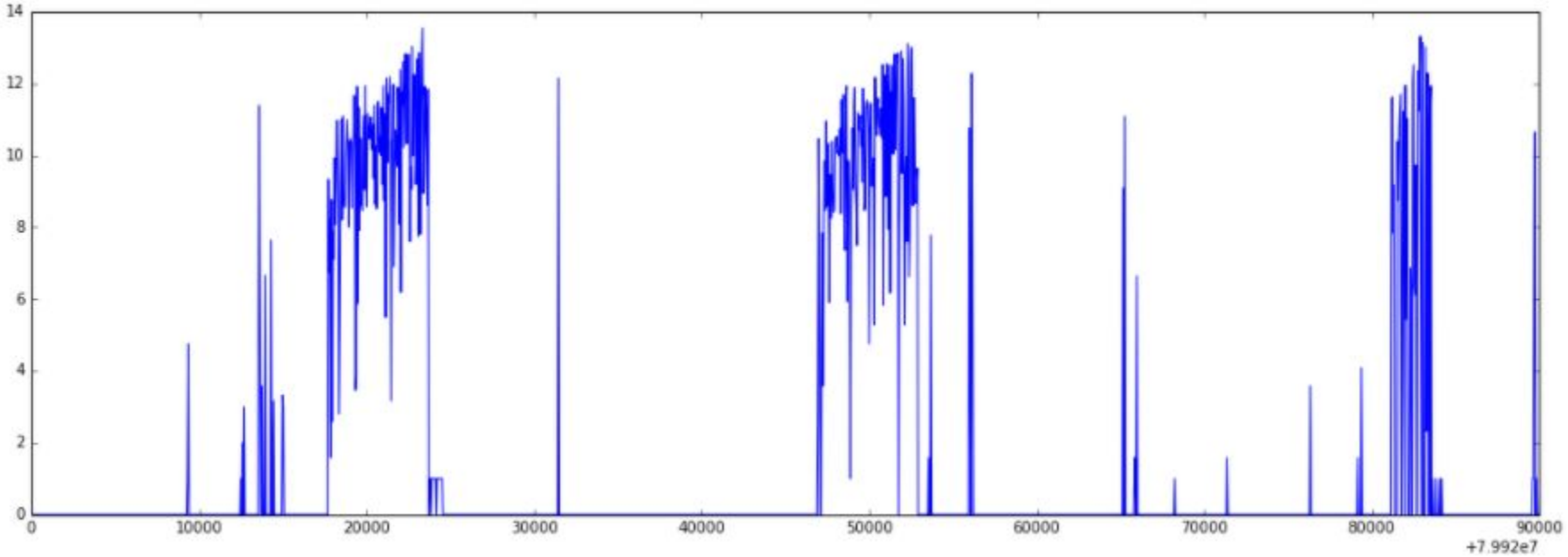
As you'd expect, most 45-mers are unique.

But, occasionally, we run into a series that are repeated all over the genome.

And, they aren't trivial repetitive sequences.

```
Chromosome 4
 79937415     1 TGAAAGTTTCACTGTAACATACTCACAGAAAGCTTTTTATCTGCA
 79937460     1 AGTGACTTTTTGTGCCACTTGCTTGGGCCACTTTTTCCCAACTCT
 79937505     1 AATTTGCAATTTGTATCTACCCTGAGAGAGGTACTGTCTATCAGG
 79937550     1 GTATATAGTACCATACTCAAACAGATTTGTTCCGTTATCTAAACT
 79937595     1 AGAAATAAATAAATCATAAAATGTTATGTGTCACTAACAAGGTAA
 79937640     1 CAACTTGAATGCTTATGTATATATTGAGCATCAATTATGTACCCA
 79937685     1 GCACTGTGATAGTGTTTTTAAAACCCCCTAAGAGAGGAGCCAAGA
 79937730   645 TGGCCGAATAGGAACAGCTCCGGTCTACAGCTCCCAGCGTGAGCG
 79937775   108 ACGCAGAAGACGGTGATTTCTGCATTTCCATCTGAGGTACCGGGT
 79937820   295 TCATCTCACTAGGGAGTGCCAGACAGTGGGCGCAGGCCAGTGTGT
 79937865     3 GTGCGCACCGTGCACGAGCCGAAGCAGGGCGAGGCATTGCCTCAC
 79937910   436 CTGGGAAGCGCAAGGGGTCAGGGAGTTCCCTTTCCGAGTCAAAGA
 79937955     6 AAGGGGTGACGGTCGCACCTGGAAAATCGGGTCACTCCCACCCGA
 79938000   233 ATATTGCGCTTTTCAGACCGGCTTAAGAAACGGCGCACCACGAGA
 79938045   138 CTATATCCCACACCTGGCTCGGAGGGTCCTACGCCCACGGAATCT
 79938090   973 CGCTGATTGCTAGCACAGCAGTCTGAGATCAAACTGCAAGGCGGC
 79938135   270 AACGAGGCTGGGGGAGGGGCGCCCGCCATTGCCCAGGCTTGCTTA
 79938180   546 GGTAAACAAAGCAGCCGGGAAGCTCGAACTGGGTGGAGCCCACCA
 79938225  2005 CAGCTCAAGGAGGCCTGCCTGCCTCTGTAGGCTCCACCTCTGGGG
 79938270  2010 GCAGGGCACAGACAAACAAAAAGACAGCAGTAACCTCTGCAGACT
 79938315   405 TAAGTGTCCCTGTCTGACAGCTTTGAAGAGAGCAGTGGTTCTCCC
 79938360     7 AGCACGCAGCTGGAGATCTGAGAACGGGCAGACAGACTGCCTCCT
 79938405    73 CAAGTGGGTCCCTGACTCCTGACCCCCGAGCAGCCTAACTGGGAG
 79938450   590 GCACCCCCCAGCAGGGGCACACTGACACCTCACACGGCAGGGTAT
 79938495  2051 TCCAACAGACCTGCAGCTGAGGGTCCTGTCTGTTAGAAGGAAAAC
 79938540   295 TAACAACCAGAAAGGACATCTACACCGAAAACCCATCTGTACATC
 79938585  2174 ACCATCATCAAAGACCAAAAGTAGATAAAACCACAAAGATGGGGA
 79938630   501 AAAAACAGAACAGAAAAACTGGAAACTCTAAAACGCAGAGCGCCT
 79938675   377 CTCCTCCTCCAAAGGAACGCAGTTCCTCACCAGCAACAGAACAAA
 79938720   382 GCTGGATGGAGAATGATTTTGACGAGCTGAGAGAAGAAGGCTTCA
 79938765   955 GACGATCAAATTACTCTGAGCTACGGGAGGACATTCAAACCAAAG
 79938810  1453 GCAAAGAAGTTGAAAACTTTGAAAAAAATTTAGAAGAATGTATAA
 79938855  2046 CTAGAATAACCAATACAGAGAAGTGCTTAAAGGAGCTGATGGAGC
 79938900  1271 TGAAAACCAAGGCTCGAGAACTACGTGAAGAATGCAGAAGCCTCA
 79938945   258 GGAGCCGATGCGATCAACTGGAAGAAAGGGTATCAGCGATGGAAG
 79938990  1313 ATGAAATGAATGAAATGAAGCGAGAAGGGAAGTTTAGAGAAAAAA
 79939035  1398 GAATAAAAAGAAATGAGCAAAGCCTCCAAGAAATATGGGACTATG
 79939080  1315 TGAAAAGACCAAATCTACGTCTGATTGGTGTACCTGAAAGTGATG
 79939125   371 TGGAGAATGGAACCAAGTTGGAAAACACTCTGCAGGATATTATCC
 79939170  1155 AGGAGAACTTCCCCAATCTAGCAAGGCAGGCCAACGTTCAGATTC
```

Zooming out


14
12
10
8
6
4
2
0
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
+7.992e7

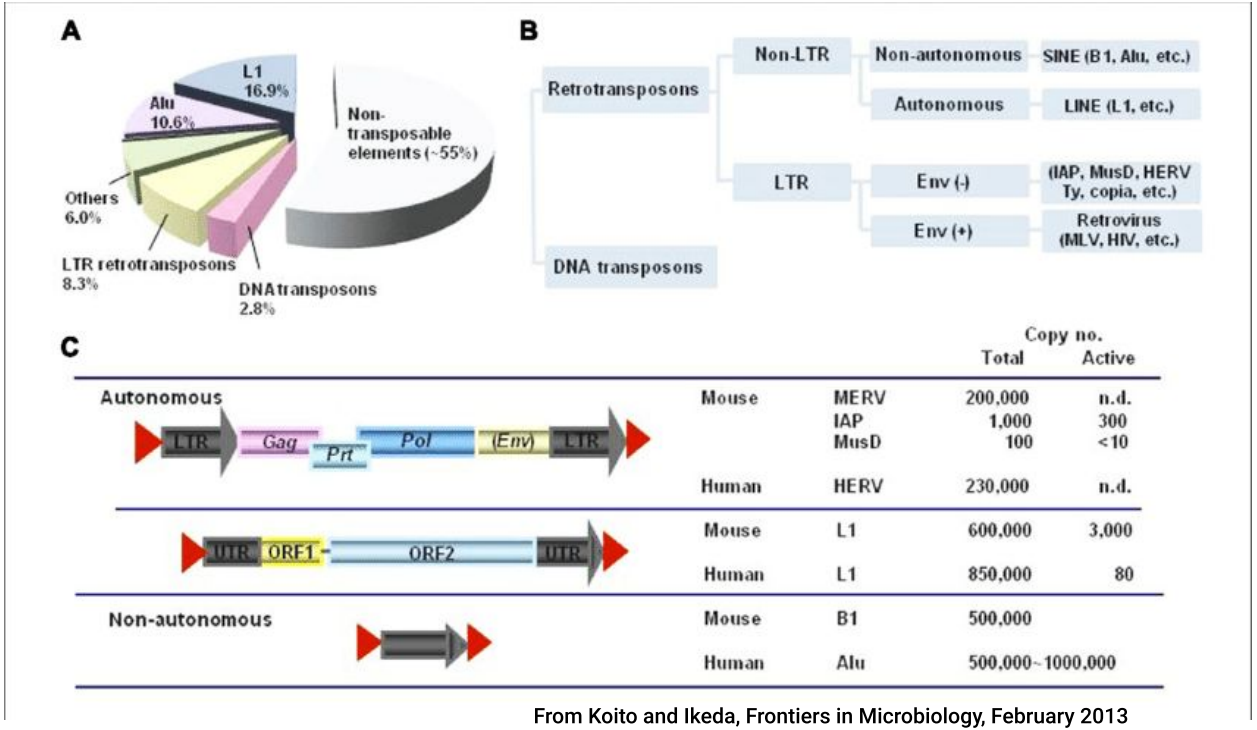
Repeated regions of our genome

Our genome is full of copies… most are due to Transposable Elements

About 45%

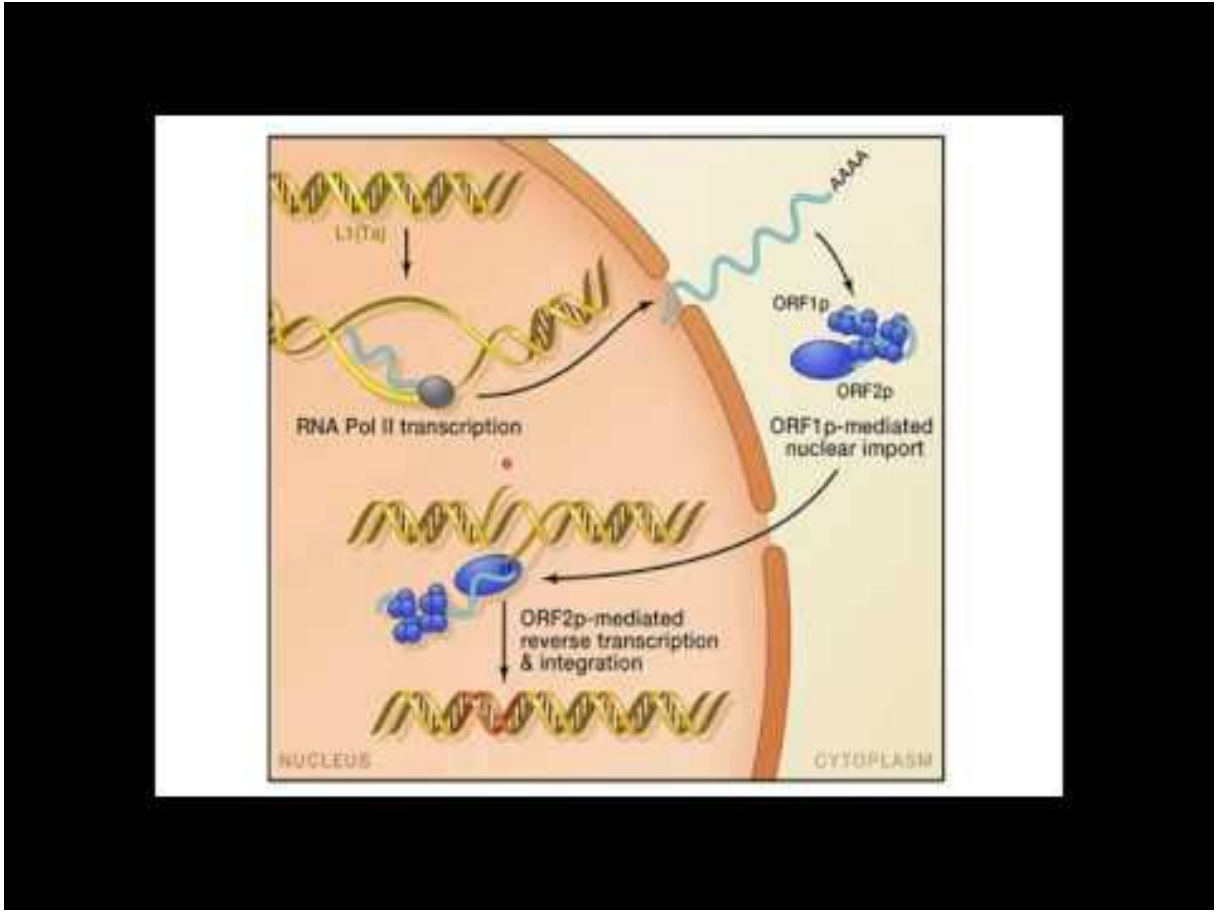
Cut-and-paste
DNA transposons

Copy-and-paste
Retrotransposons

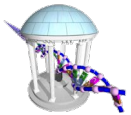


From Koito and Ikeda, Frontiers in Microbiology, February 2013

TEs are everywhere



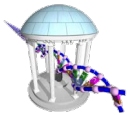
Let's find all copies of one of our repeats



In [5]:
```
chromo = [str(i) for i in xrange(1,23)] + ['X', 'Y', 'MT']

target = "AGCACGCAGCTGGAGATCTGAGAACGGGCAGACAGACTGCCTCCT"    # was 7 times
revtar = revComp(target)

for contig in chromo:
    with open("Chr%s.seq" % contig, 'r') as fp:
        seq = fp.read()
    start = 0
    while True:
        i = seq.find(target, start)
        if (i > 0):
            print(contig, i, "+")
            start = i + 1
        else:
            break
    start = 0
    while True:
        i = seq.find(revtar, start)
        if (i > 0):
            print(contig, i, "-")
            start = i + 1
        else:
            break
```

```
2 169249269 +
4 79938360 +
5 156067276 -
8 72880901 -
11 93137285 +
11 93421619 +
16 33957922 -
```

And look around where we found them



Here's one of the copies of "AGCACGCAGCTGGAGATCTGAGAACGGGCAGACAGACTGCCTCCT"

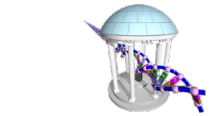
Not surprisingly, it is surrounded by repeated 45-mers that are similar to the ones on chromosome 4.

So, with what we now know let's go hunting for a particular type of Transposable Element.

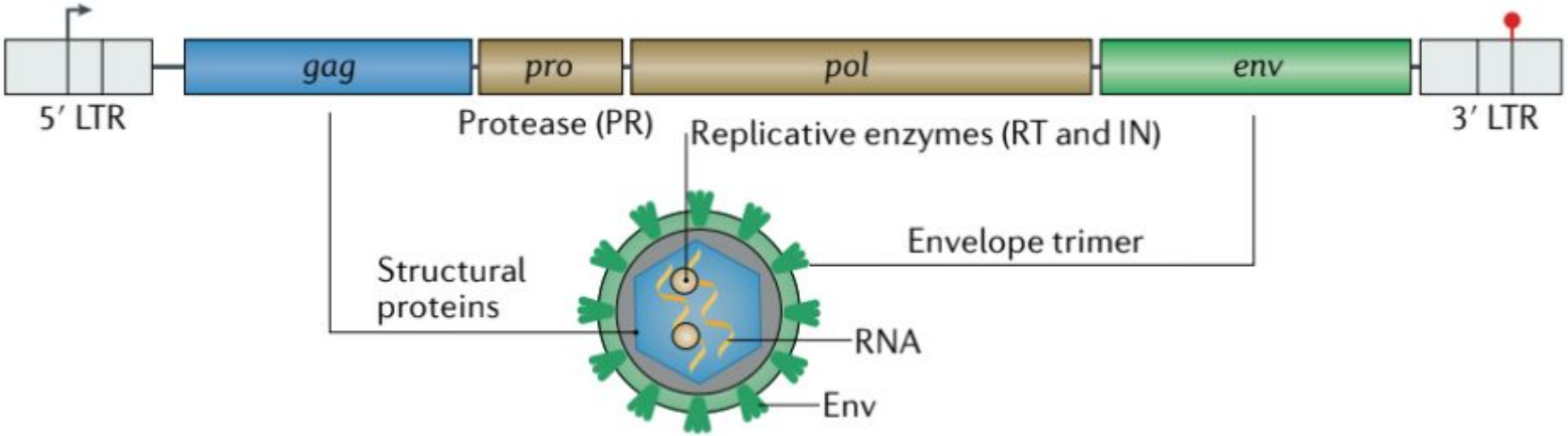
One of Viral origin! Yes, our genomes have parasites.

```
Chromosome 11
 93136365     1 TTGCAGAGAAGTAGGAATGCTTTTACACTGTCGGTGGGAATGTAA
 93136410     1 ATTAGGTCAACTATTGTGGAAGACAGTGTGCAATTCCTCAAAGAT
 93136455     5 CTAGAACCAGAAATACGATTTGACCCAGCAATCCCATTACTGGGT
 93136500     1 AAATACCCAAAAGAATATAAATCATTCTATTATAGAGATACATGC
 93136545     1 ATGAGTATGTTCAGTGCAGCACACTTCACAATAGCAAAGACATGG
 93136590     1 AATCAACACAACTGCCCATCAATGATAGACTAAAGAAAACGTGGT
 93136635     1 ACATGGGGGAGGAGCCAAGATGGCCGAATAGGAACAGCTCCGGTC
 93136680    93 TACAGCTCCCAGCGTGAGCGACGCAGAAGACGGTGATTTCTGCAT
 93136725  1334 TTCCATCTGAGGTACCGGGTTCATCTCACTAGGGAGTGCCAGACA
 93136770   169 GTGGGCGCAGGCCAGTGTGTGTGCGCACCGTGCGCGAGCCGAAGC
 93136815   449 AGGGCGAGGCATTGCCTCACCTGGGAAGCGCAAGGGGTCAGGGAG
 93136860     3 TTCCCTTTCTGAGTCAAAGAAAGGGGTGACGGTCGCACCTGGAAA
 93136905   437 ATCGGGTCACTCCCACCCGAATATTGCGCTTTTCAGACCGGCTTA
 93136950    55 AGAAACGGCGCACCACGAGACTATATCCCACACCTGGCTCGGAGG
 93136995   393 GTCCTACGCCCACGGAATCTCGCTGATTGCTAGCACAGCAGTCTG
 93137040   270 AGATCAAACTGCAAGGCGGCAACGAGGCTGGGGGAGGGGCGCCCG
 93137085   581 CCATTGCCCAGGCTTGCTTAGGTAAACAAAGCAGCCGGGAAGCTC
 93137130  1565 GAACTGGGTGGAGCCCACCACAGCTCAAGGAGGCCTGCCTGCCTC
 93137175  1908 TGTAGGCTCCACCTCTGGGGGCAGGGCACAGACAAACAAAAAGAC
 93137220   406 AGCAGTAACCTCTGCAGACTTAAGTGTCCCTGTCTGACAGCTTTG
 93137265  1104 AAGAGAGCAGTGGTTCTCCCAGCACGCAGCTGGAGATCTGAGAAC
 93137310     8 GGGCAGACAGACTGCCTCCTCAAGTGGGTCCCTGACTCCTGACCC
 93137355   817 CCGAGCAGCCTAACTGGGAGGCACCCCCCAGCAGGGGCACACTGA
 93137400    81 CACCTCACATGGCAGGGTATTCCAACAGACCTGCAGCTGAGGGTC
 93137445   329 CTGTCTGTTAGAAGGAAAACTAACAACCAGAAAGGACATCTACAC
 93137490   526 CGAAAACCCATCTGTACATCACCATCATCAAAGACCAAAAGTAGA
 93137535  1308 TAAAACCACAAAGATGGGGAAAAAACAGAACAGAAAAACTGGAAA
 93137580   446 CTCTAAAACGCAGAGCGCCTCTCCTCCTCCAAAGGAACGCAGTTC
 93137625   246 CTCACCAGCAACAGAACAAAGCTGGATGGAGAATGATTTTGACGA
 93137670   886 GCTGAGAGAAGAAGGCTTCAGACGATCAAATTACTCTGAGCTACG
 93137715  1553 GGAGGACATTCAAACCAAAGGCAAAGAAGTTGAAAACTTTGAAAA
 93137760  1512 AAATTTAGAAGAATGTATAACTAGAATAACCAATACAGAGAAGTG
 93137805  1272 CTTAAAGGAGCTGATGGAGCTGAAAACCAAGGCTCGAGAACTACG
 93137850  1265 TGAAGAATGCAGAAGCCTCAGGAGCCGATGCGATCAACTGGAAGA
 93137895   775 AAGGGTATCAGCAATGGAAGATGAAATGAATGAAATGAAGCGAGA
 93137940  1267 AGGGAAGTTTAGAGAAAAAAGAATAAAAAGAAATGAGCAAAGCCT
 93137985   115 CCAAGAAATATGGGACTATGTGAAAAGACCAAATCTACGTCTGAC
 93138030   367 TGGTGTACCTGAAAGTGATGTGGAGAATGGAACCAAGTTGGAAAA
 93138075  2615 CACTCTGCAGGATATTATCCAGGAGAACTTCCCCAATCTAGCAAG
 93138120   955 GCAGGCCAACGTTCAGATTCAGGAAATACAGAGAACGCCACAAAG
```

An Endogenous Retrovirus



These TEs include many genes related to those in viral sequences.



The LTRs are identical sequences that enclose the a virus-like genome.

Next Time

We will develop a strategy to find LTR-like sequences in a genome.