Always Evolving:

A wild foray into the world of genomics and visualization

By Blake Joyce & Asher

And there’ll be Plotly coding later on

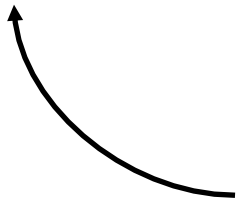
Let’s Talk Biology

I promise it’s not as boring as your Intro Bio class probably made it

Besides you’ll be human all your life (hopefully): it’s useful to know how you work

- We propose:

Genetics has always been a big data & visualization problem



Did he just say “job security”?

The 3 Challenges of Genetics: in brief

Need to visualization three major things in particular:

1) **Initial visualization of genetic sequence (a.k.a. ‘Sequencing’)**

    1.1) How to “see” chemical sequences at A, T, G, C

    1.2) Crazy sizes of data (hard to get all that information)

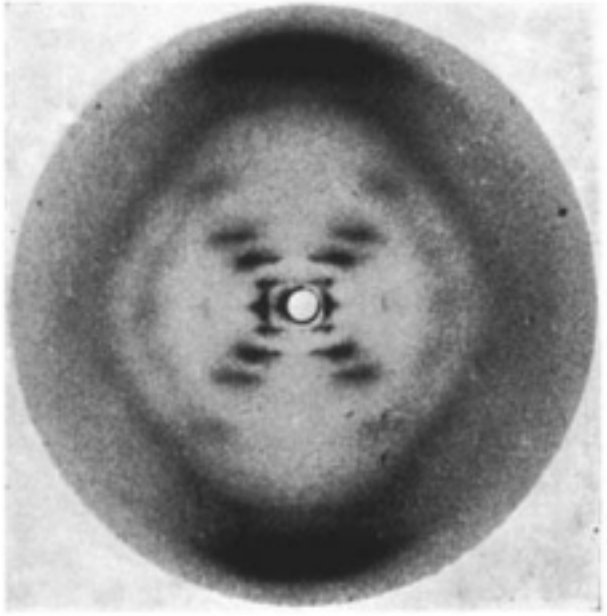
2) **Visualization to understand the sequence**

    2.1) Hard to ‘comprehend’ that amount of data

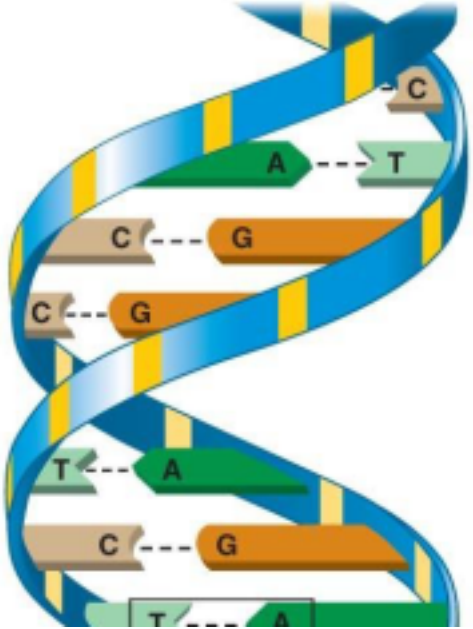
    2.2) Compare to other genetics to understand evolution, gene function, etc

3) Assess and visualize to teach/conceptualize genetics (another conversation)

Challenge 1.1: How Can We See the Sequence?

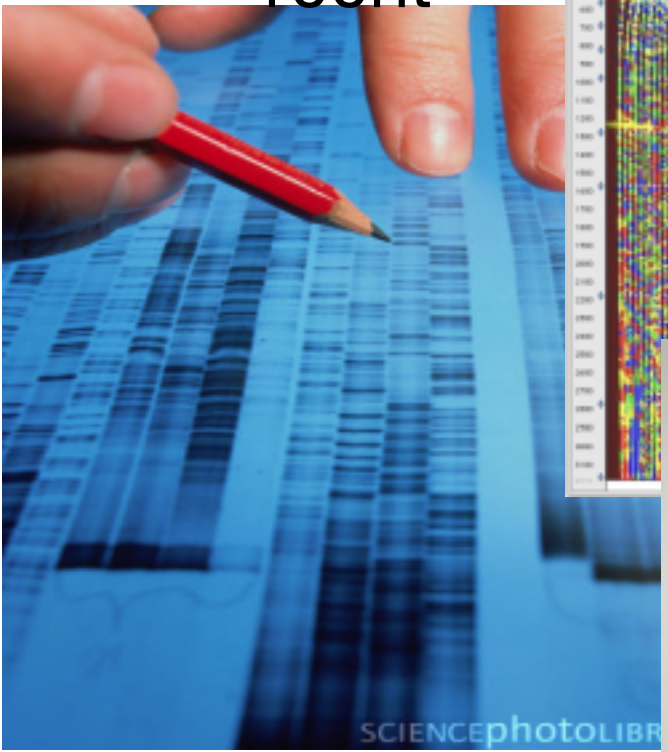


=

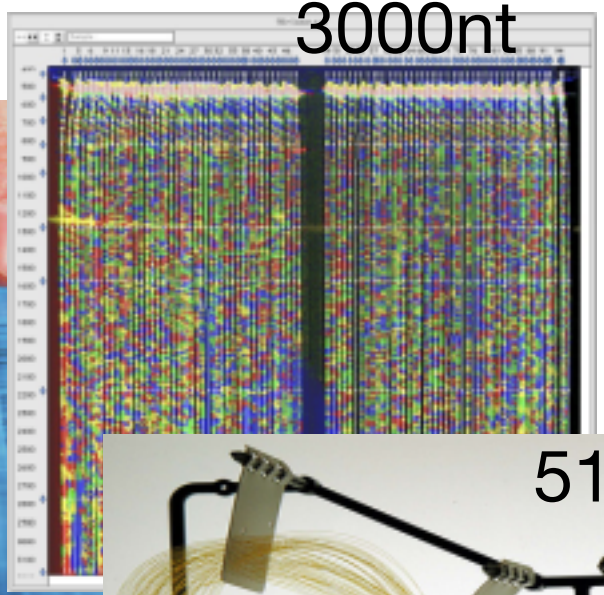


DNA sequencing

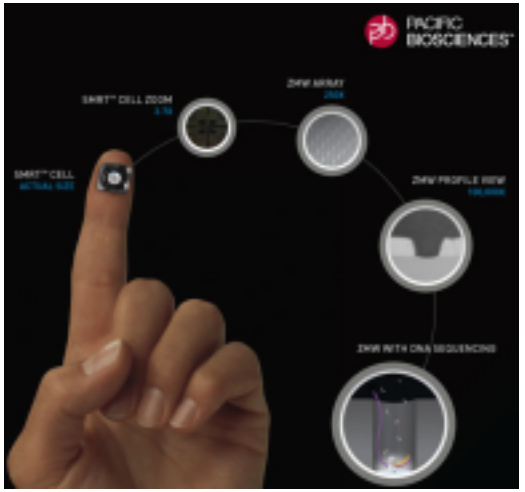
100nt



3000nt



10-20 kb reads



512,000,000nt/2 weeks

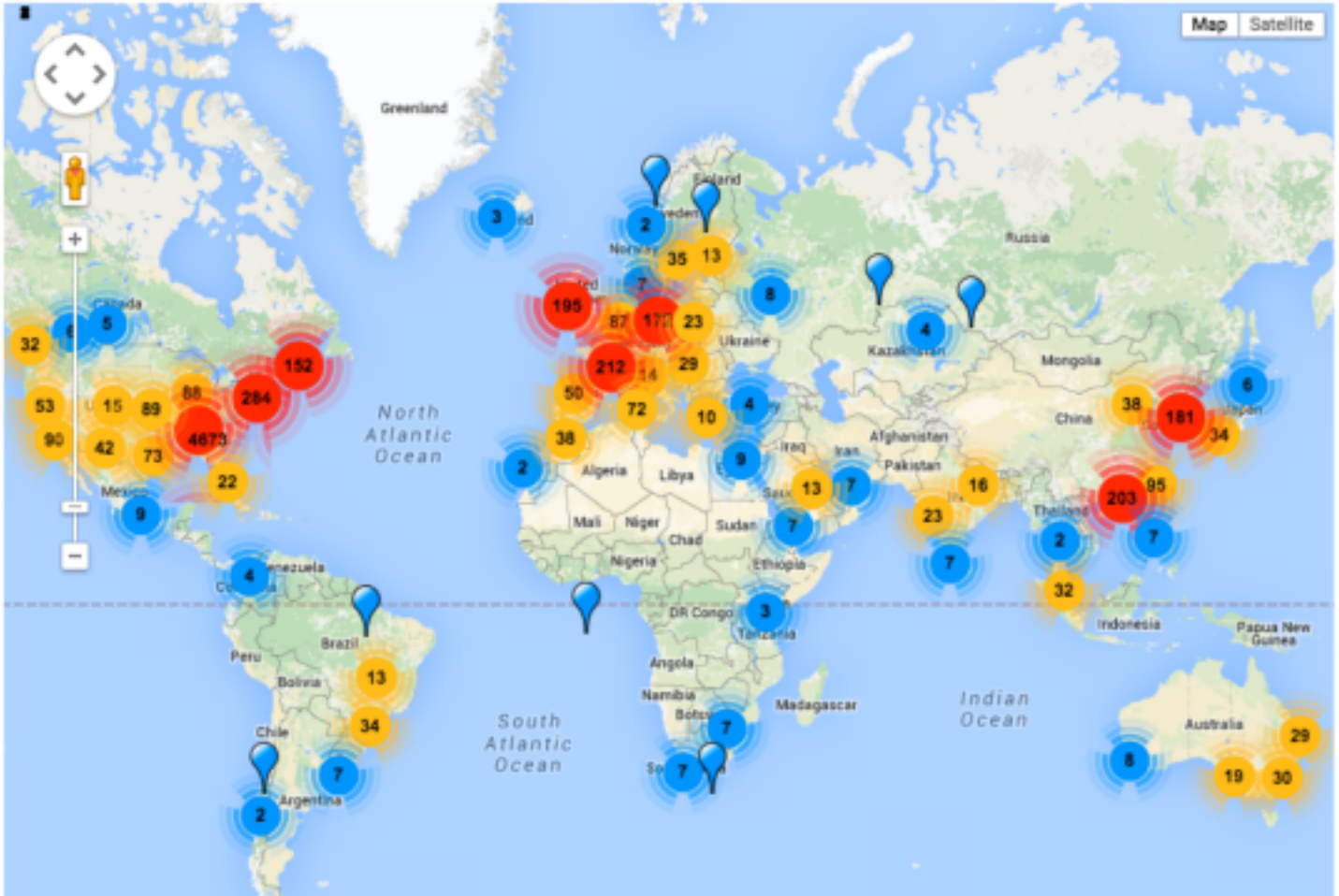
100,000nt

Illumina HiSeq X Ten

In < 3 Days:

- Generates 1800 GB of raw data
- 1,600,000,000,000 bases of DNA sequence (500 human genomes)
- +18,000 humans/year @ ~$1000/each

And there are sequencers across the globe



Cool. So We Can See It Pretty Well.

Challenge 1.1 solved?

The short answer is kind of….

Long answer: Challenge 1.1 isn’t a bottleneck any more

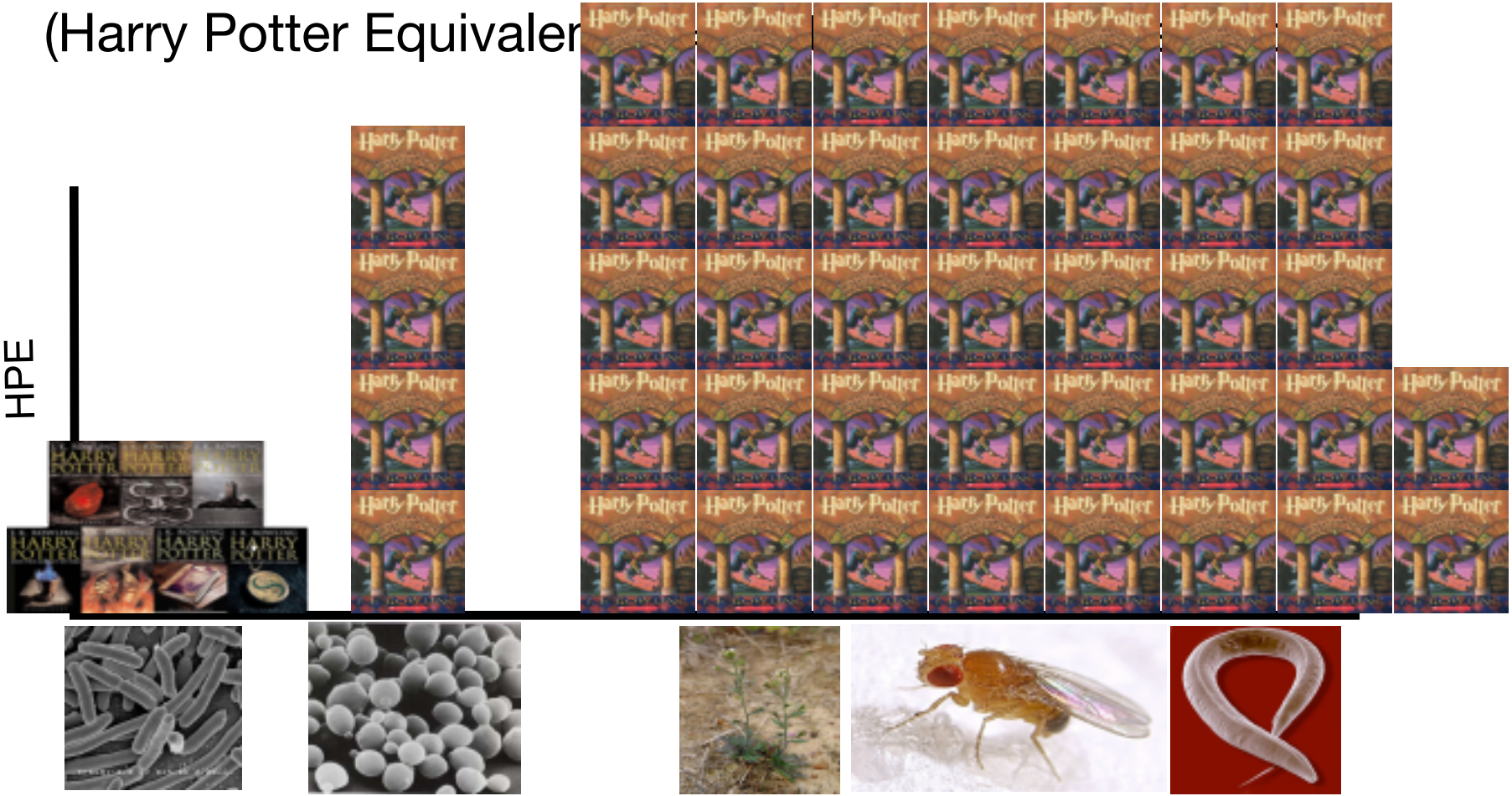
Challenge 1.2: Size Does Matter ←(scientifically proven)

- “Here be bottlenecks”
- Data science and cyberinfrastructure to the rescue!
- Got to move, analyze, assess, and keep forever

And there’s a lot of data. (feel free to say: “How much is there?” in unison)

Genome size as HPE

(Harry Potter Equivaler



Plant Genome Sizes

Arabidopsis:	37 HPES
Rice:	100 HPES
Maize:	650 HPES
Wheat:	6000 HPES
Paris japonica:	50,000 HPES

One half of one human genome.

HPE

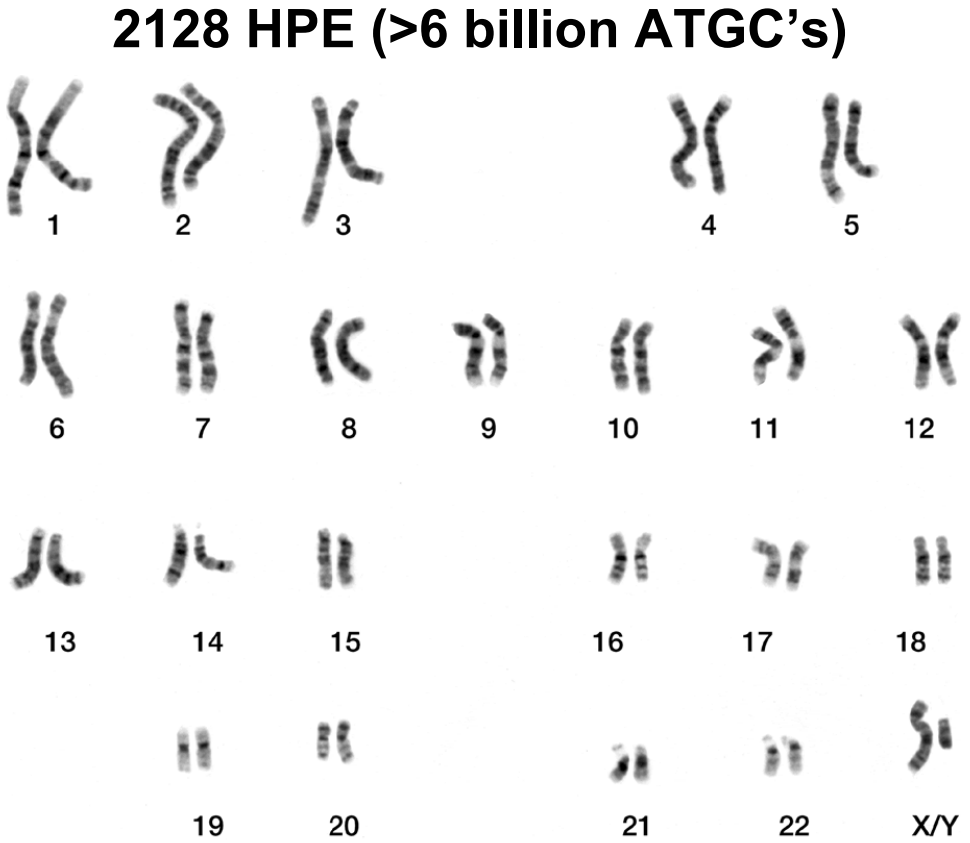
Homo sapien

And that's just half…

You get two halves…

One from "Mom"

One from "Dad"



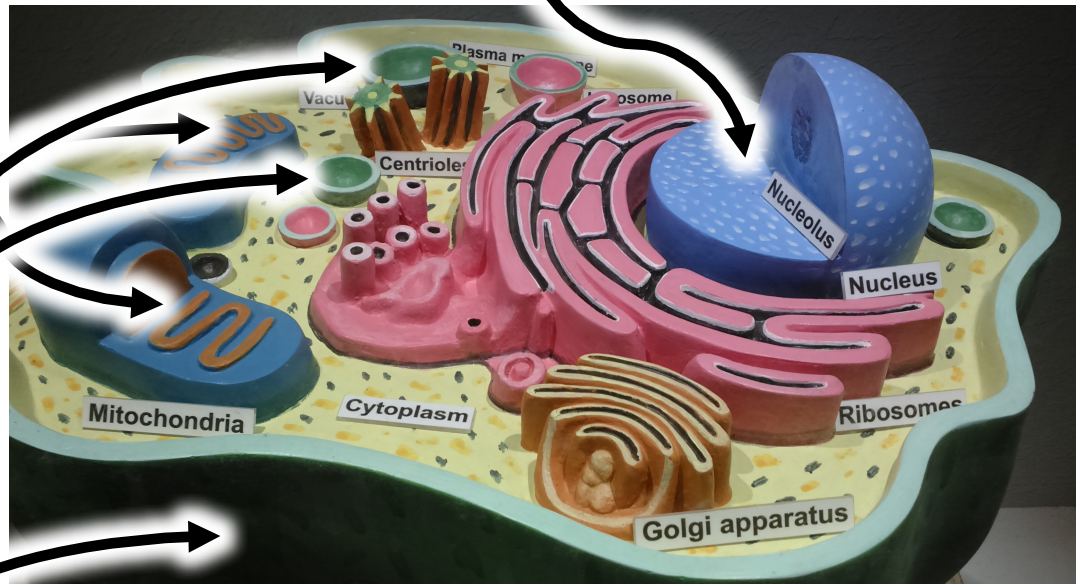
And that's just the nucleus…

There's DNA in other places in the cell…

Here Be the "Two Halves"

But DNA Also Be Here

And chloroplasts (plants)



Speaking of visualization problems: this is NOT how a cell looks

And that’s just one cell…

You’ve got trillions of cells working together in a cacophony of concerted harmony to maintain your life (biology) every moment of every day…

I’m not sure how to visualize this

¯_(ツ)_/¯

Trillions of Your Cells + Trillions of Microbes

Here's the only way I can think to visualize that:

One human

We've got ~7.2 billion of these to feed, water, clothe, house…

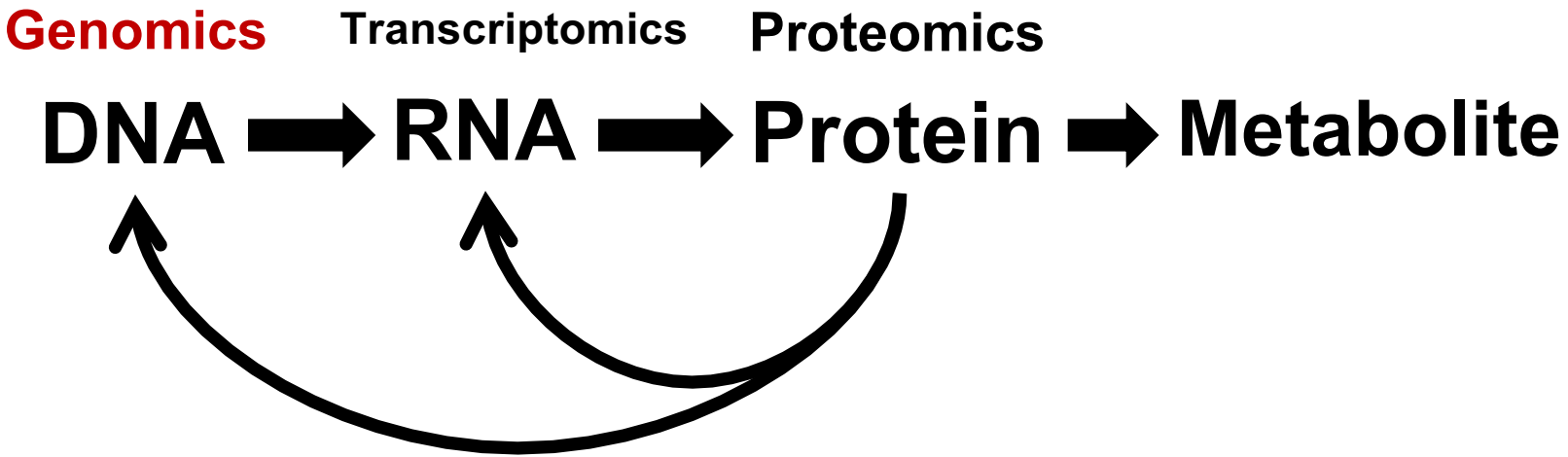
Each cell has a slightly different genome from mutations

Oh look! Cancer

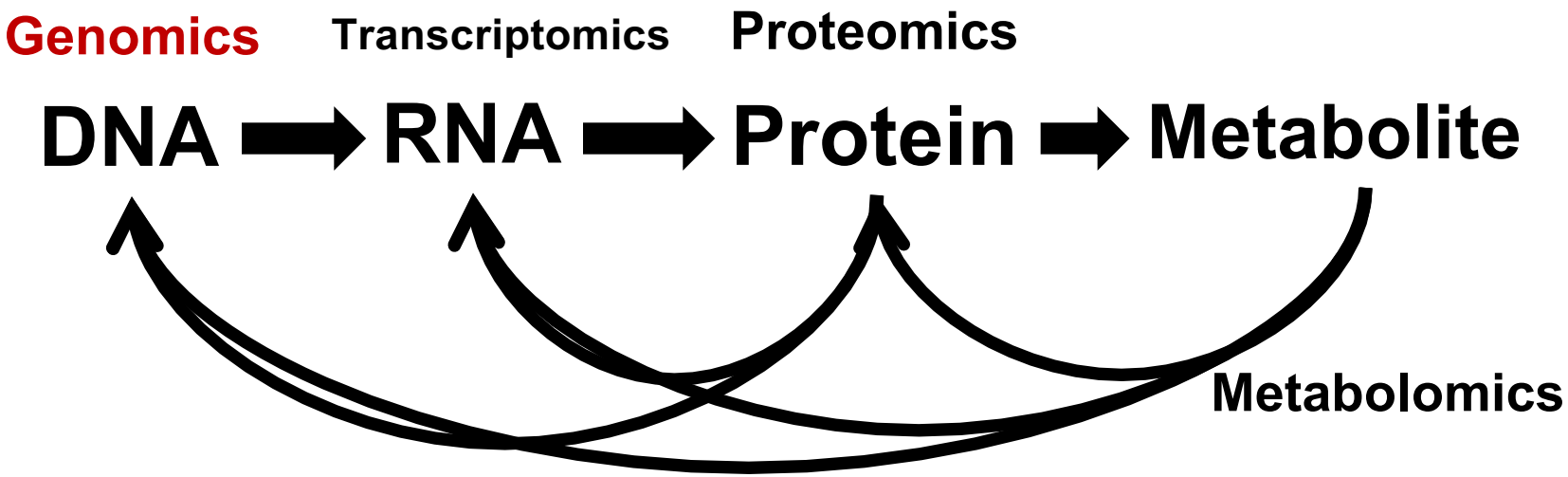
And that’s just genomes…

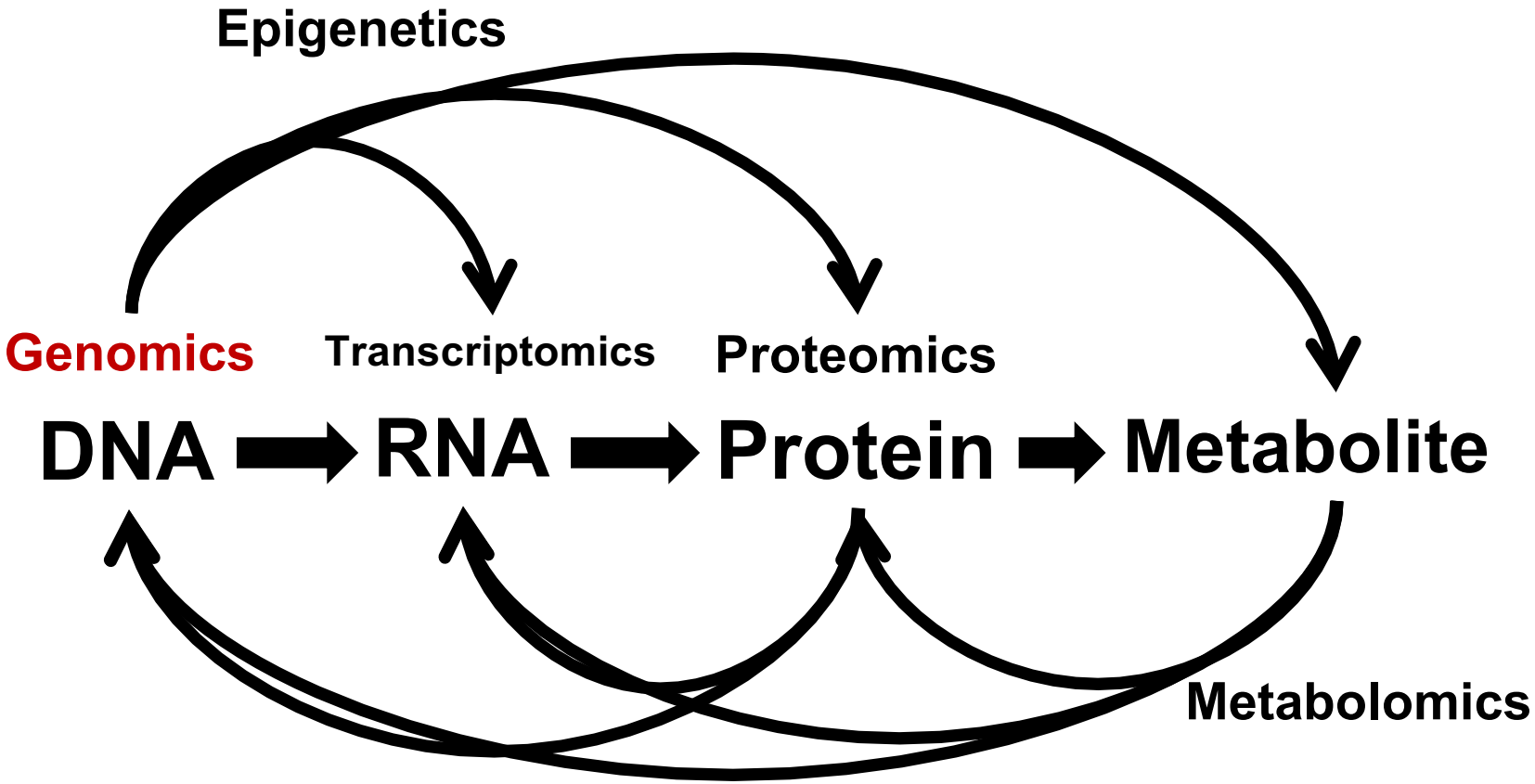
“Textbook Biology”

Challenge 1.2 Mark II: integrate all the genome data with other data types

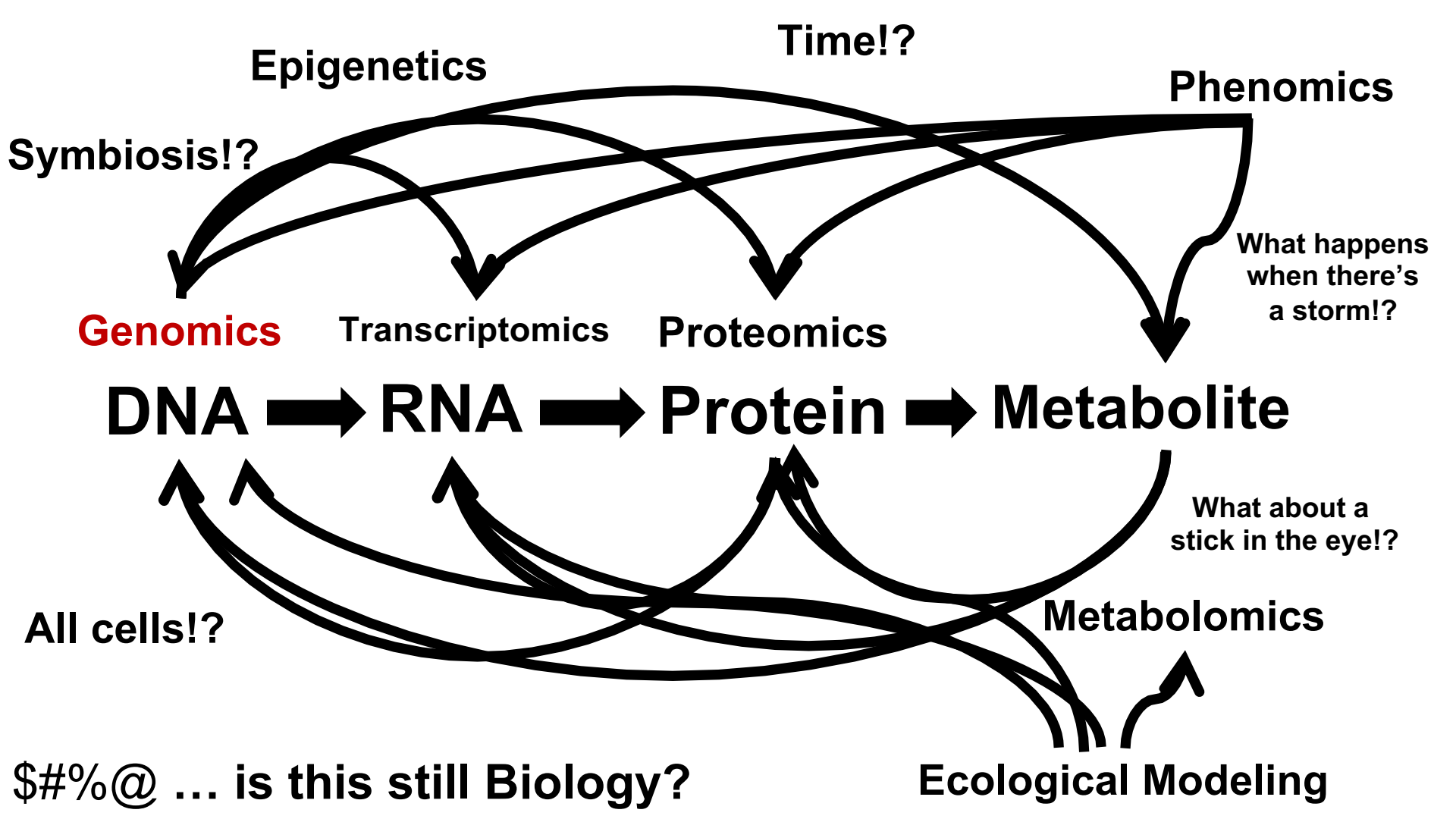


Truer “Textbook Biology”





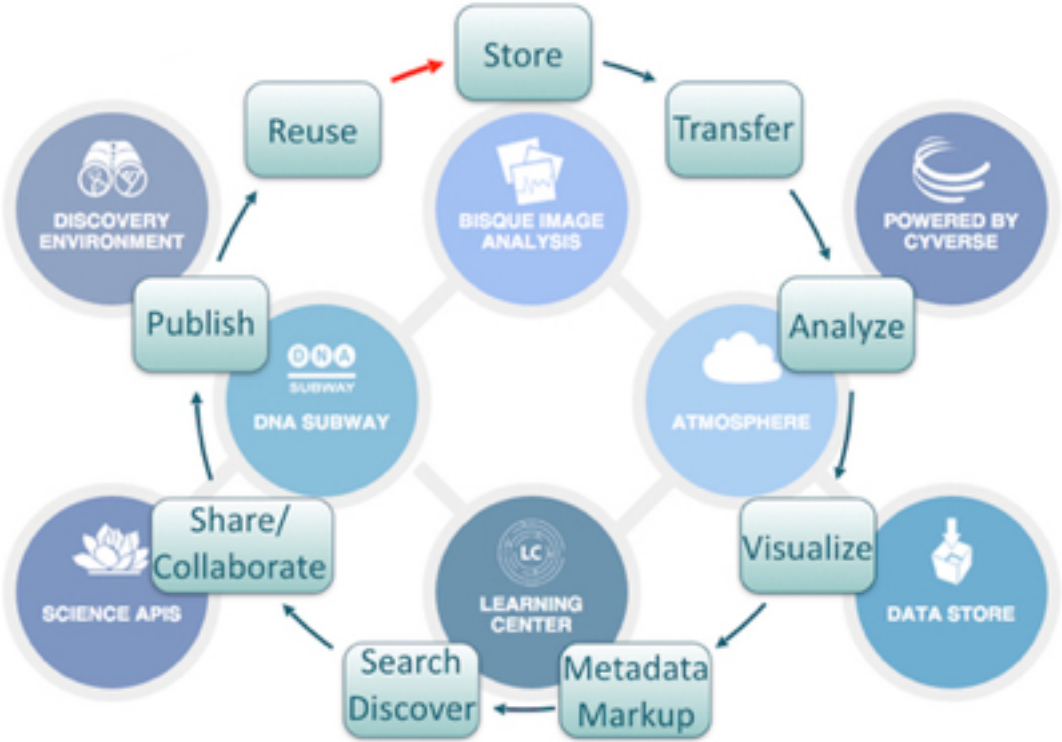
Even Truer Biology. We're still working on the textbooks.


Time!?
Epigenetics
Phenomics
Symbiosis!?
What happens when there's a storm!?
Genomics
Transcriptomics
Proteomics
DNA
RNA
Protein
Metabolite
What about a stick in the eye!?
All cells!?
Metabolomics
$#%@ … is this still Biology?
Ecological Modeling

DON'T PANIC

Embrace the chaos!
The effort is worth it…

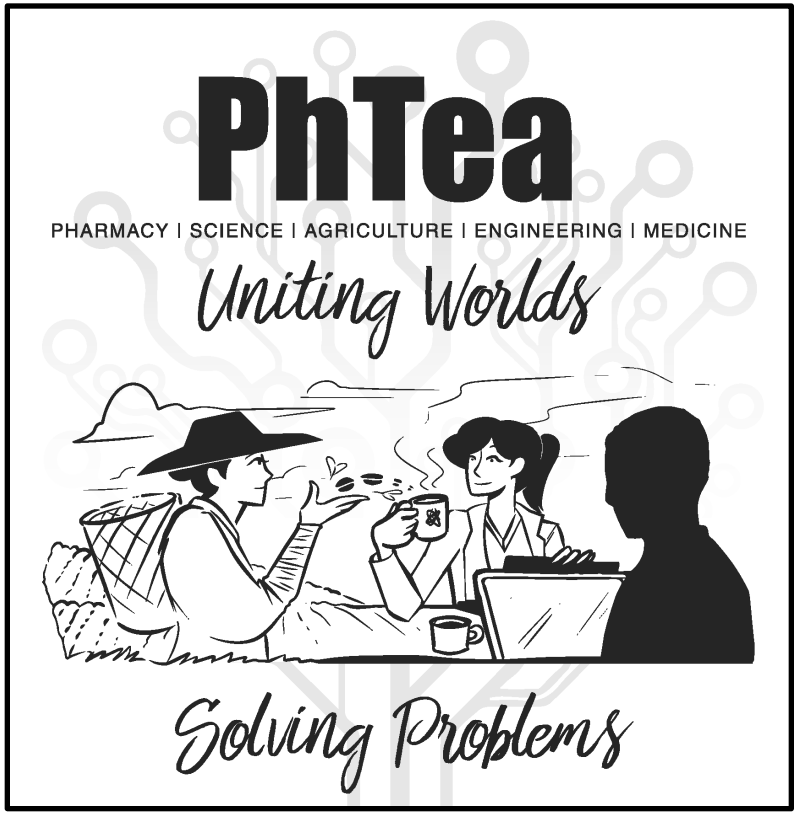
Challenge 1.2 Solved: Sign up for CYVERSE™



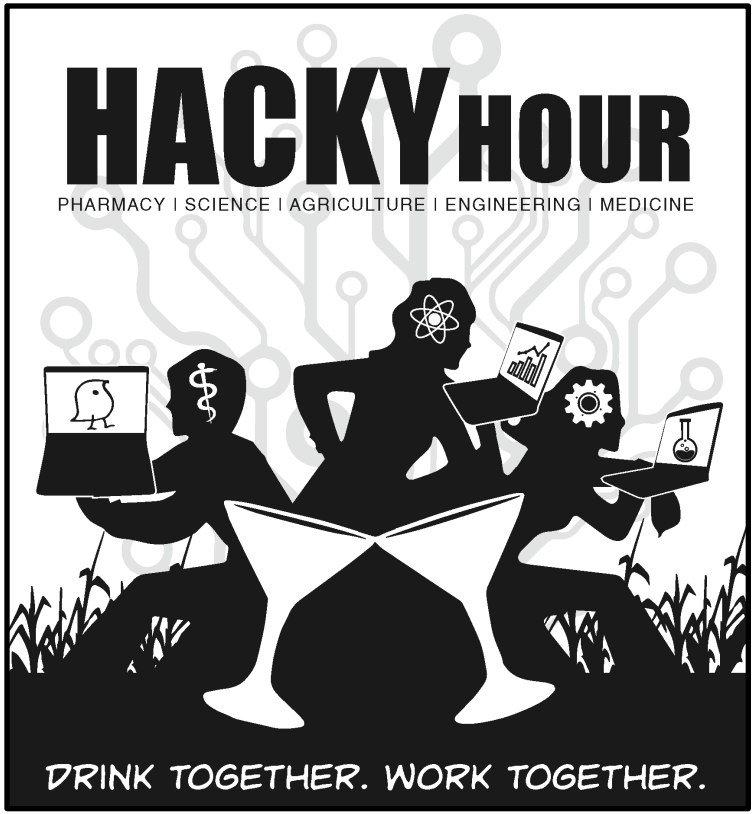
I am a CyVerse SI Team member: think of me as your chaos sherpa (Beaker is my spirit animal)

Shameless Plug:

ResB(az)^2
RESEARCH BAZAAR ARIZONA



Weekly on Tuesdays 8-10 AM BIO5 Institute



Weekly on Thursdays 4-7 PM Gentle Ben's


So We've Got Data…
And We Can Move It and
Fun Things Like That

TACCCTTCAGGAACAAATAGGATGGATGACAAATAATCCACCTATCCCAGTAGGAGAA

ATTTATAAAAGATGGATAATCCTGGGATTAAATAAAATAGTAAGAATGTATAGCCCTA

CCAGCATTCTGGACATAAGACAAGGACCAAAGGAACCCTTTAGAGACTATGTAGACCG

GT

GA

GA

CC

AT

GC

AT

GG

But what does the genome look like? (to us)

AGCCAACAGCCCCACCAGAAGAGAGCTTCAGGTCTGGGGTAGAGACAACAACTCCCCC

TCAGAAGCAGGAGCCGATAGACAAGGAACTGTATCCTTTAACTTCCCTCAGGTCACTC

TTTGGCAACGACCCCTCGTCACAATAAAGATAGGGGGGCAACTAAAGGAAGCTCTAT

TAGATACAGGAGCAGATGATACAGTATTAGAAGAAATGAGTTTGCCAGGAAGATGGAA

TACCCTTCAGGAACAAATAGGATGGATGACAAATAATCCACCTATCCCAGTAGGAGAA
ATTTATAAAAGATGGATAATCCTGGGATTAAATAAAATAGTAAGAATGTATAGCCCTA
CCAGCATTCTGGACATAAGACAAGGACCAAAGGAACCCTTTAGAGACTATGTAGACCG
GTTCTATAAAACTCTAAGAGCCGAGCAAGCTTCACAGGAGGTAAAAAATTGGATGACA
GAAACCTTGTTGGTCCAAAATGCGAACCCAGATTGTAAGACTATTTTAAAAGCATTGG
GACCAGCGGCTACACTAGAAGAAATGATGACAGCATGTCAGGGAGTAGGAGGACCCGG
CCATAAGGCAAGAGT[illegible]ATTCAGCTACCATA
ATGATGCAGAGAGGC[illegible]GTGTTTCAATTGTG
GCAAAGAAGGGCACA[illegible]AAGGGCTGTTGGAA
ATGTGGAAAGGAAGGACACCAAATGAAAGATTGTACTGAGAGACAGGCTAATTTTTTA
GGGAAGATCTGGCCTTCCTACAAGGGAAGGCCAGGGAATTTTCTTCAGAGCAGACCAG
AGCCAACAGCCCCACCAGAAGAGAGCTTCAGGTCTGGGGTAGAGACAACAACTCCCCC
TCAGAAGCAGGAGCCGATAGACAAGGAACTGTATCCTTTAACTTCCCTCAGGTCACTC
TTTGGCAACGACCCCTCGTCACAATAAAGATAGGGGGGCAACTAAAGGAAGCTCTAT
TAGATACAGGAGCAGATGATACAGTATTAGAAGAAATGAGTTTGCCAGGAAGATGGAA

A lot of this.

TACCCTTCAGGAACAAATAGGATGGATGACAAATAATCCACCTATCCCAGTAGGAGAA

ATTTATAAAAGATGGATAATCCTGGGATTAAATAAAATAGTAAGAATGTATAGCCCTA

CCAGCATTCTGGACATAAGACAAGGACCAAAGGAACCCTTTAGAGACTATGTAGACCG

GTTCTATAAAACTCTAAGAGCCGAGCAAGCTTCACAGGAGGTAAAAAATTGGATGACA

GAAACCTTGTTGGTCCAAAATGCGAACCCAGATTGTAAGACTATTTTAAAAGCATTGG

GACCAG[illegible]CCCGG

CCATAA[illegible]CCATA

ATGATG[illegible]TTGTG

GCAAAG[illegible]TGGAA

ATGTGG[illegible]TTTTA

But this turns into cool micromachines (life)

GGGAAGATCTGGCCTTCCTACAAGGGAAGGCCAGGGAATTTTCTTCAGAGCAGACCAG

AGCCAACAGCCCCACCAGAAGAGAGCTTCAGGTCTGGGGTAGAGACAACAACTCCCCC

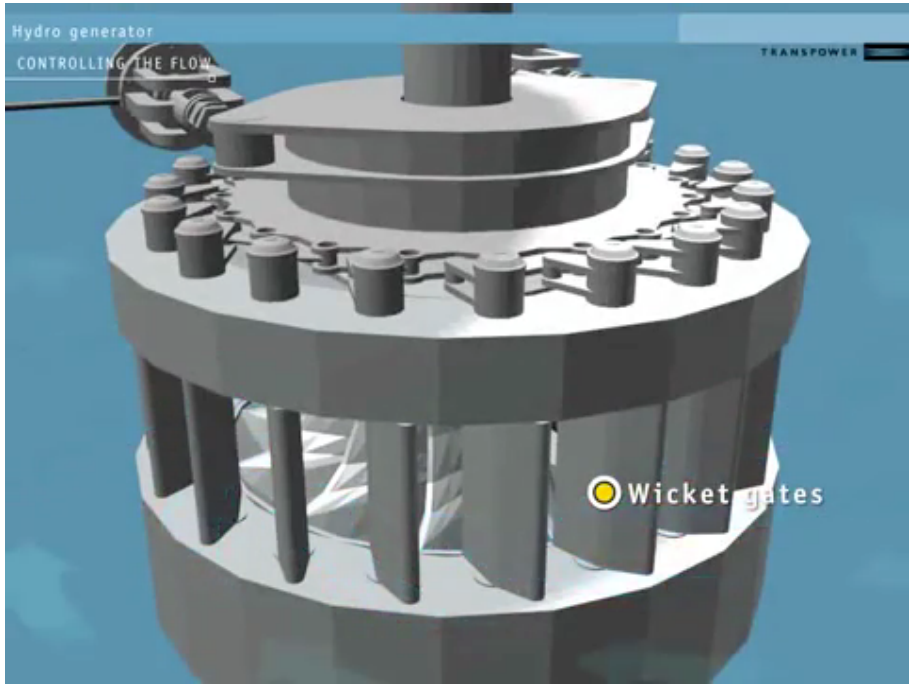
TCAGAAGCAGGAGCCGATAGACAAGGAACTGTATCCTTTAACTTCCCTCAGGTCACTC

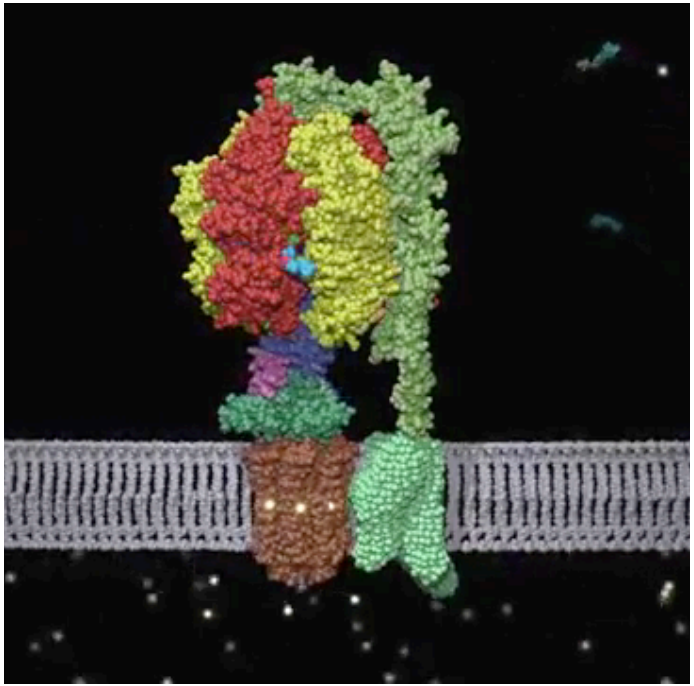
TTTGGCAACGACCCCTCGTCACAATAAAGATAGGGGGGCAACTAAAGGAAGCTCTAT

TAGATACAGGAGCAGATGATACAGTATTAGAAGAAATGAGTTTGCCAGGAAGATGGAA

And Bio-micromachines (Proteins) Do All Kinds of Things

(and biology did it first)

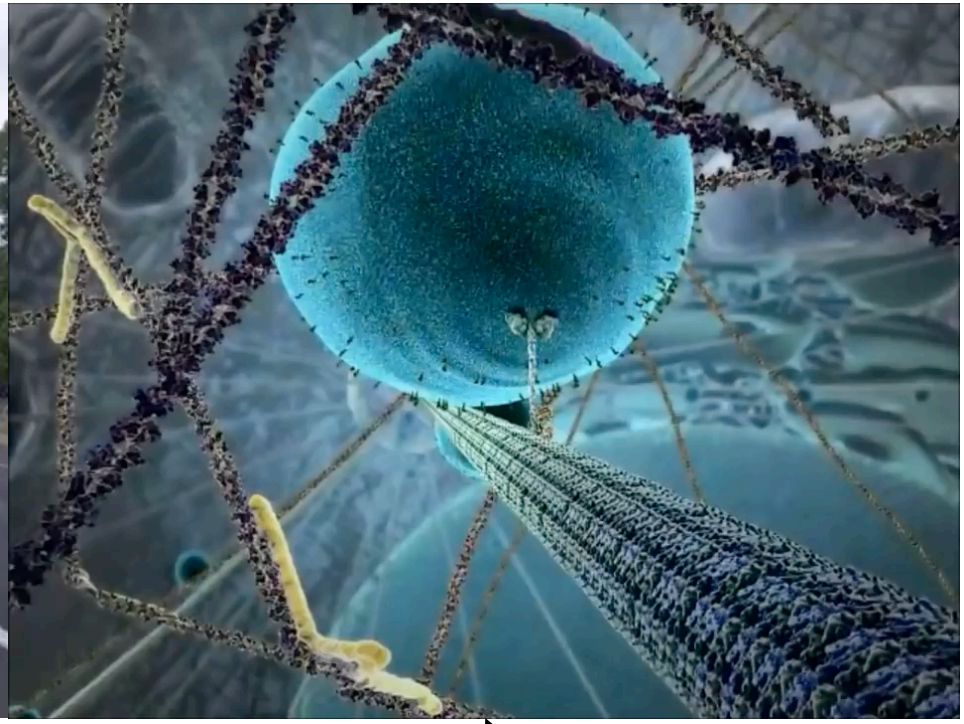




Uses rotation and gradients to convert kinetic energy into other forms

And Bio-micromachines (Proteins) Do All Kinds of Things

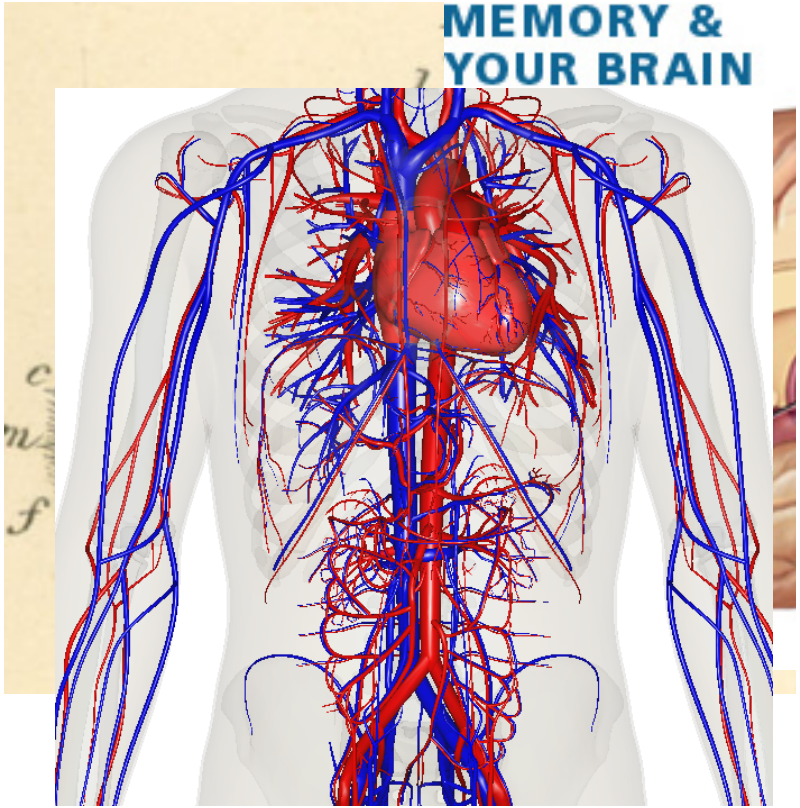
(and biology did it first)

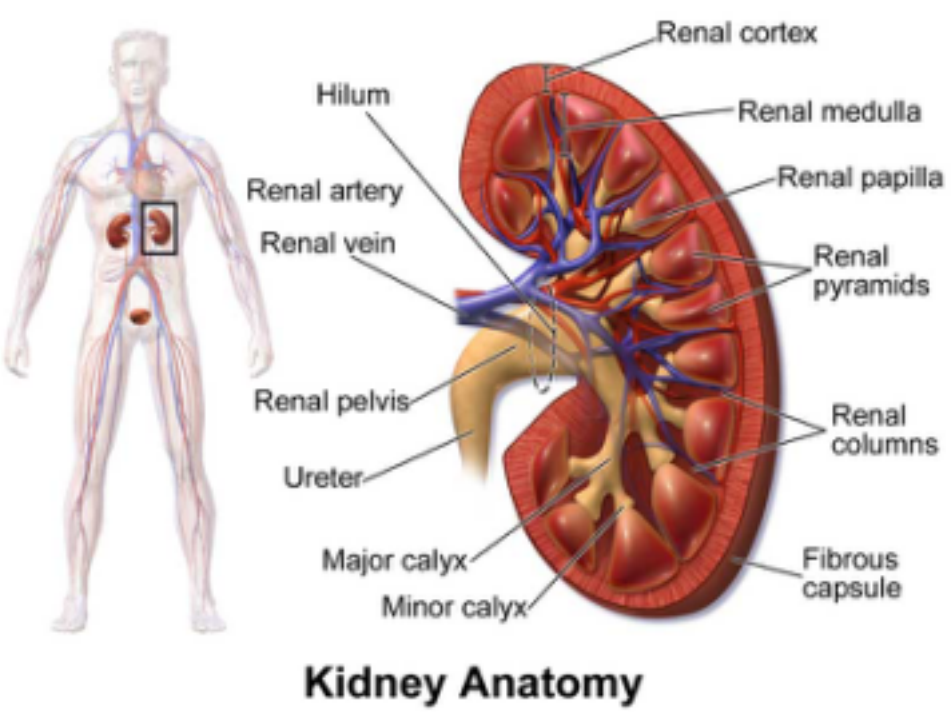


Converts chemical energy into kinetic energy (movement)

And That's Just One Cell

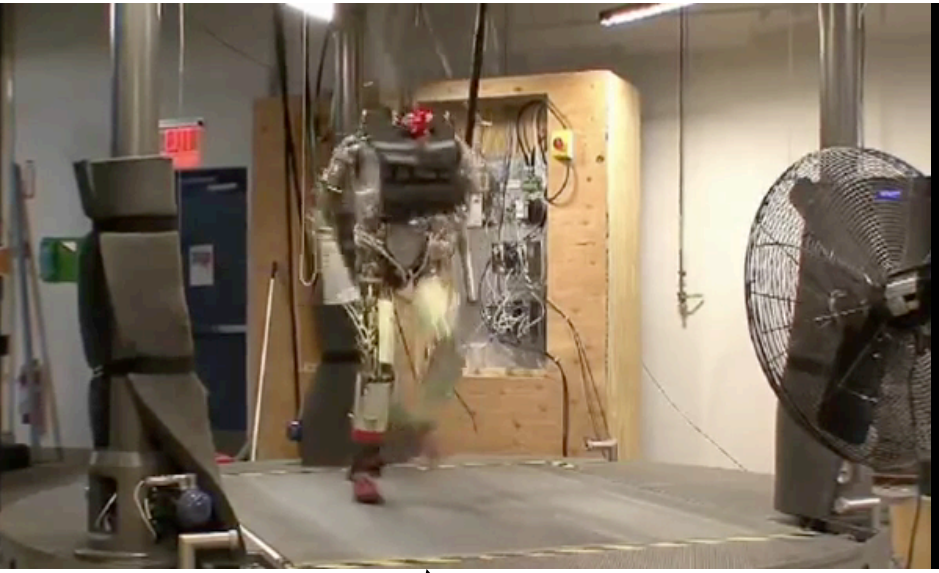
Combinations of cells yield crazy things (that we mostly take for granted):





And That's Just One Cell

Combinations of cells yield crazy things (that we mostly take for granted):

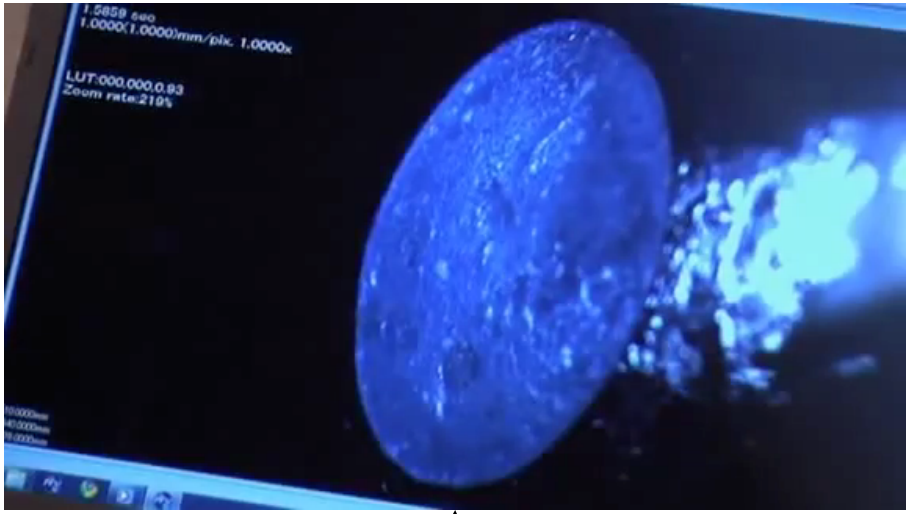


DARPA Project ($$$Millions)

And That's Just Humans (who frankly are boring)

Zombified snails

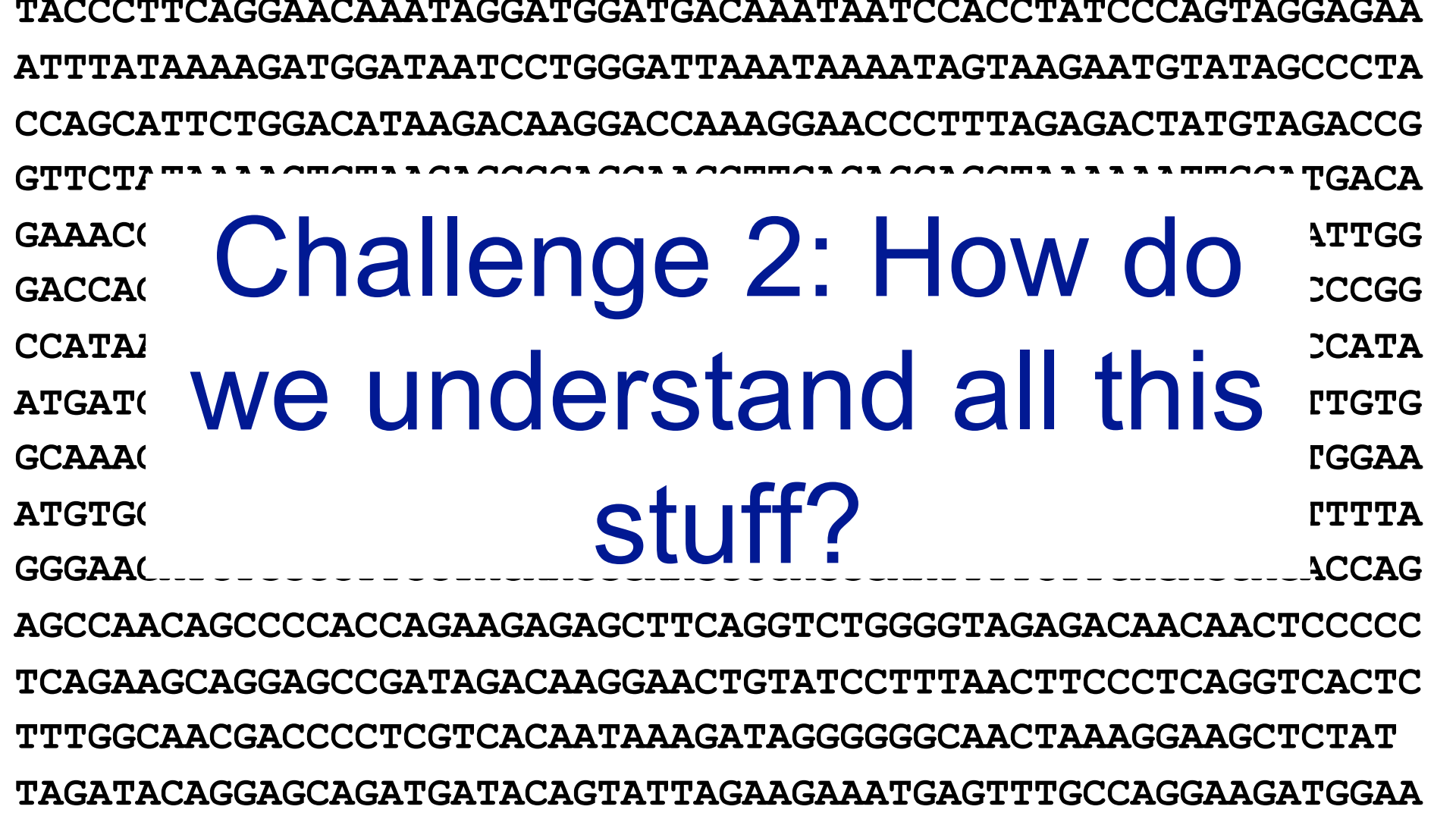
And There's Some Interesting Applications (like bulletproof skin)



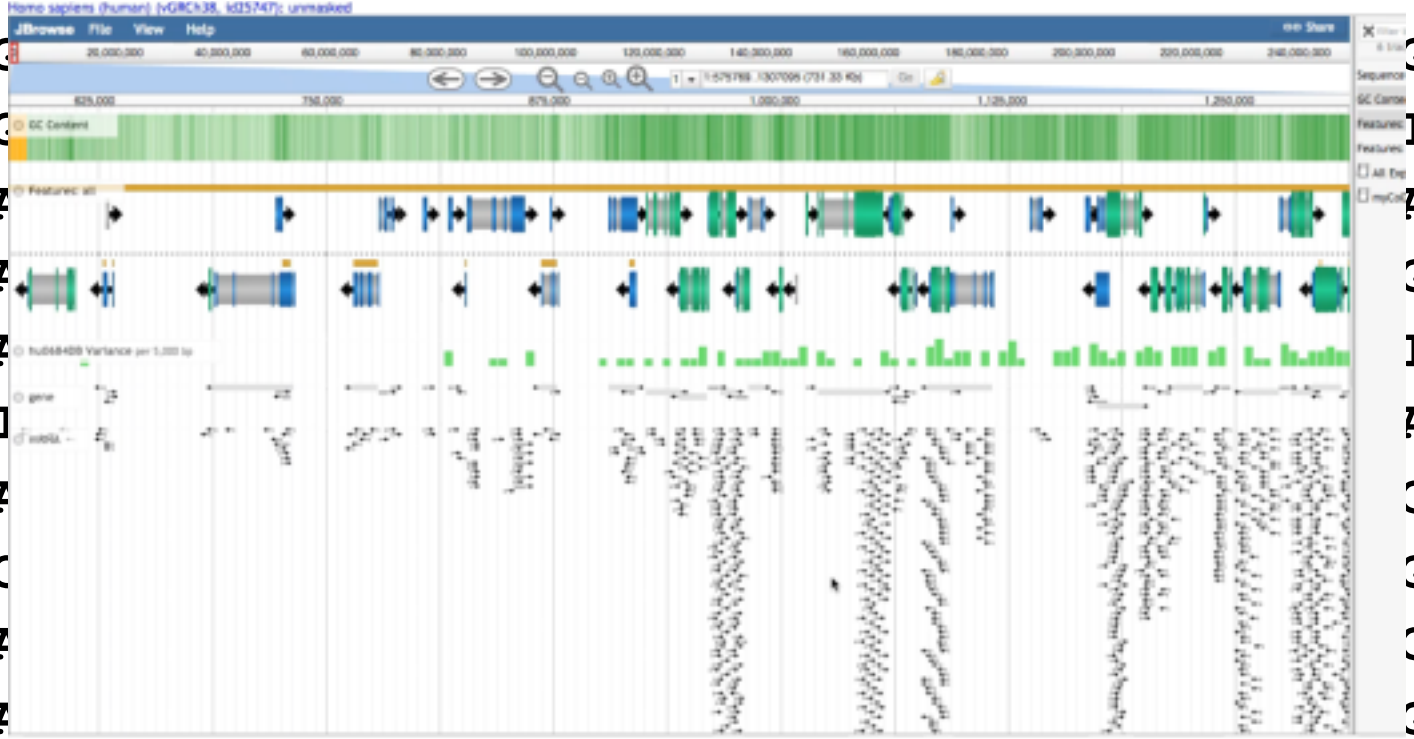
Spider goats

Human skin cells with spider silk


TACCCTTCAGGAACAAATAGGATGGATGACAAATAATCCACCTATCCCAGTAGGAGAA
ATTTATAAAAGATGGATAATCCTGGGATTAAATAAAATAGTAAGAATGTATAGCCCTA
CCAGCATTCTGGACATAAGACAAGGACCAAAGGAACCCTTTAGAGACTATGTAGACCG
GTTCTATAAAACTCTAAGAGCCGAGCAAGCTTCACAGGAGGTAAAAAATTGGATGACA
GAAACCTT
GACCAGCG
CCATAAGG
ATGATGCA
GCAAAGAA
GCATTGG
GACCCGG
TACCATA
AATTGTG
GTTGGAA
And it's all right here
(somewhere)
ATGTGGAAAGGAAGGACACCAAATGAAAGATTGTACTGAGAGACAGGCTAATTTTTTA
GGGAAGATCTGGCCTTCCTACAAGGGAAGGCCAGGGAATTTTCTTCAGAGCAGACCAG
AGCCAACAGCCCCACCAGAAGAGAGCTTCAGGTCTGGGGTAGAGACAACAACTCCCCC
TCAGAAGCAGGAGCCGATAGACAAGGAACTGTATCCTTTAACTTCCCTCAGGTCACTC
TTTGGCAACGACCCCTCGTCACAATAAAGATAGGGGGGCAACTAAAGGAAGCTCTAT
TAGATACAGGAGCAGATGATACAGTATTAGAAGAAATGAGTTTGCCAGGAAGATGGAA


Challenge 2: How do we understand all this stuff?

Need to transform the raw data



We've Come a Long Way

Hand comparisons!

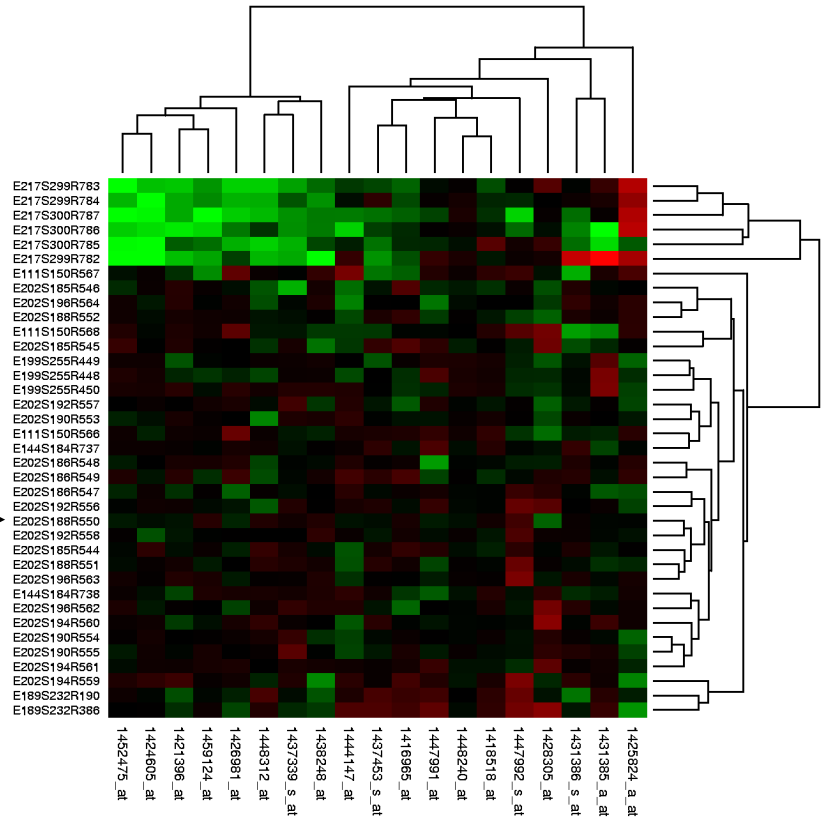
(insane, but feasible with a big enough blackboard)

Basic Local Alignment and Search Tool

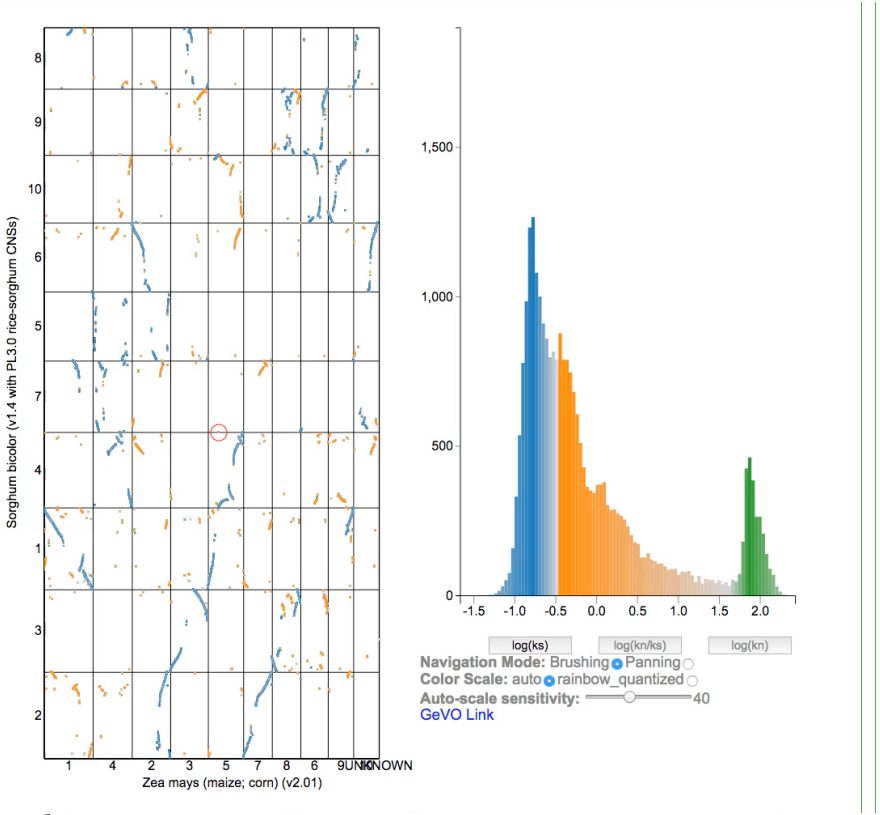
(I bet it was a BLAST after trying to do things by hand)

"Heat" maps

(Red and green? What the $#%@!?)



Example #1: Synteny Plots

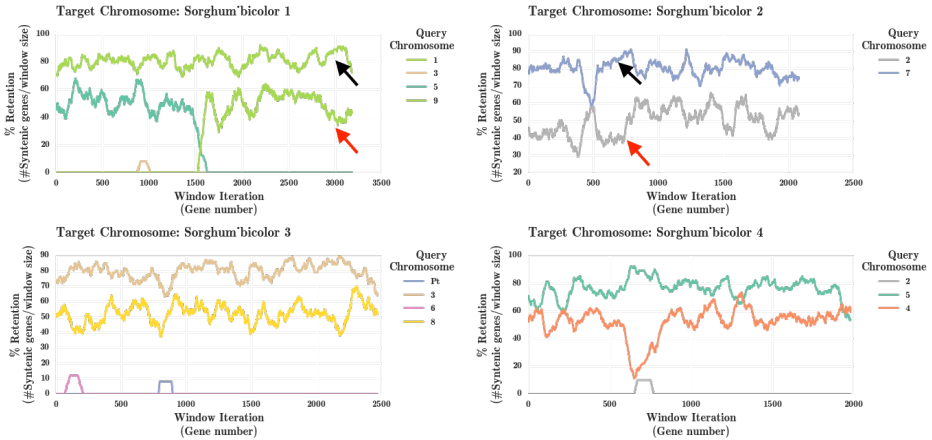


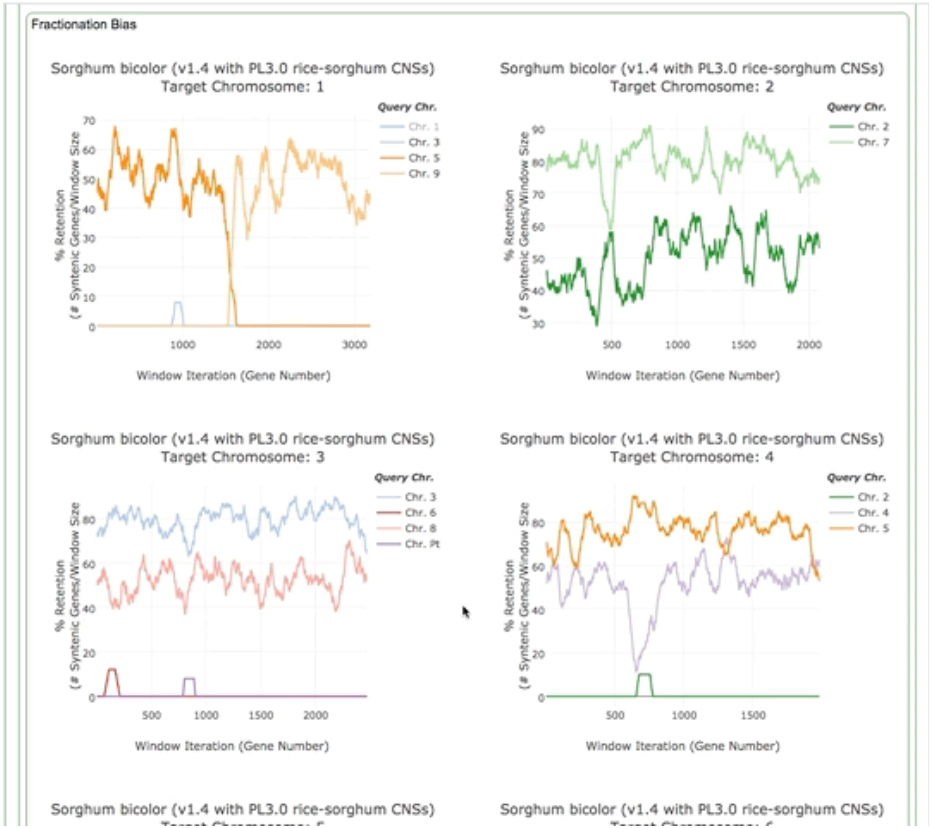
* **NOT** Plotly!

@seanastephens

https://github.com/hdc-arizona/synteny-vis

Example #2: Fractionation Bias





Thanks, Plotly!

Visions of the Future (aka Asher and Blake's wish list)

Don't forget all that chaos!

1) **Multidimensional analysis (N-genome, N-dataset comparisons)**

2) Parallelization of everything for more computational power. Literally everything

3) **Build systems that move, analyze, and visualize data + interoperate**

4) Death to single-purpose analysis tools

5) Online manuscripts, including data hosting & new (interactive) visualizations

6) Convince Microsoft to remove visualizations from Excel

7) Rule the world with an iron fist

Case Study #1

ssessing the Age of Duplicated Gene

Synonymous -vs- Non-Synonymous Mutation

…a simplified example using my favorite gene…

RNA!

"transcript"

AUGGCUUCUCAUGAACGUUAG

start!

A - S - H - E - R

stop!

Protein!

		Second Position								
		U		C		A		G		
		code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	
First Position	U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U
		UUC		UCC		UAC		UGC		C
		UUA	leu	UCA		UAA	STOP	UGA	STOP	A
		UUG		UCG		UAG	STOP	UGG	trp	G
	C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U
		CUC		CCC		CAC		CGC		C
		CUA		CCA		CAA	gln	CGA		A
		CUG		CCG		CAG		CGG		G
	A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U
		AUC		ACC		AAC		AGC		C
		AUA		ACA		AAA	lys	AGA	arg	A
		AUG	met	ACG		AAG		AGG		G
	G	GUU	val	GCU	ala	GAU	asp	GGU	gly	U
		GUC		GCC		GAC		GGC		C
		GUA		GCA		GAA	glu	GGA		A
		GUG		GCG		GAG		GGG		G

Third Position

Synonymous Mutation (Ks)

…protein product **doesn't change**...

AUGGCUUCUCAUGAACGCUAG

start! **A S H E R** stop!

First Position	Second Position: U		Second Position: C		Second Position: A		Second Position: G		Third Position
	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	
U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U
	UUC		UCC		UAC		UGC		C
	UUA	leu	UCA		UAA	STOP	UGA	STOP	A
	UUG		UCG		UAG	STOP	UGG	trp	G
C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	lys	AGA	arg	A
	AUG	met	ACG		AAG		AGG		G
G	GUU	val	GCU	ala	GAU	asp	GGU	gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	glu	GGA		A
	GUG		GCG		GAG		GGG		G

Non-Synonymous Mutation (Kn)

*…protein product **changes**...*

AUGGCUUCUCAUGAAAGUUAG

start! **A S H E S** stop!

First Position	Second Position: U		Second Position: C		Second Position: A		Second Position: G		Third Position
	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	
U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U
	UUC		UCC		UAC		UGC		C
	UUA	leu	UCA		UAA	STOP	UGA	STOP	A
	UUG		UCG		UAG	STOP	UGG	trp	G
C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	lys	AGA	arg	A
	AUG	met	ACG		AAG		AGG		G
G	GUU	val	GCU	ala	GAU	asp	GGU	gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	glu	GGA		A
	GUG		GCG		GAG		GGG		G

+ time

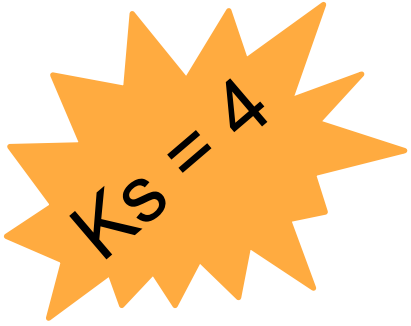
AUGGCUUCUCACGAACGCUAG

start! A S H E R stop!

Ks = 2

First Position	Second Position: U				Second Position: C				Second Position: A				Second Position: G				Third Position
	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	code	Amino Acid	
U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U
	UUC		UCC		UAC		UGC		C
	UUA	leu	UCA		UAA	STOP	UGA	STOP	A
	UUG		UCG		UAG	STOP	UGG	trp	G
C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	lys	AGA	arg	A
	AUG	met	ACG		AAG		AGG		G
G	GUU	val	GCU	ala	GAU	asp	GGU	gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	glu	GGA		A
	GUG		GCG		GAG		GGG		G

+ *more* time



First Position	Second Position: U code	U Amino Acid	C code	C Amino Acid	A code	A Amino Acid	G code	G Amino Acid	Third Position
U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U
	UUC		UCC		UAC		UGC		C
	UUA	leu	UCA		UAA	STOP	UGA	STOP	A
	UUG		UCG		UAG	STOP	UGG	trp	G
C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	lys	AGA	arg	A
	AUG	met	ACG		AAG		AGG		G
G	GUU	val	GCU	ala	GAU	asp	GGU	gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	glu	GGA		A
	GUG		GCG		GAG		GGG		G

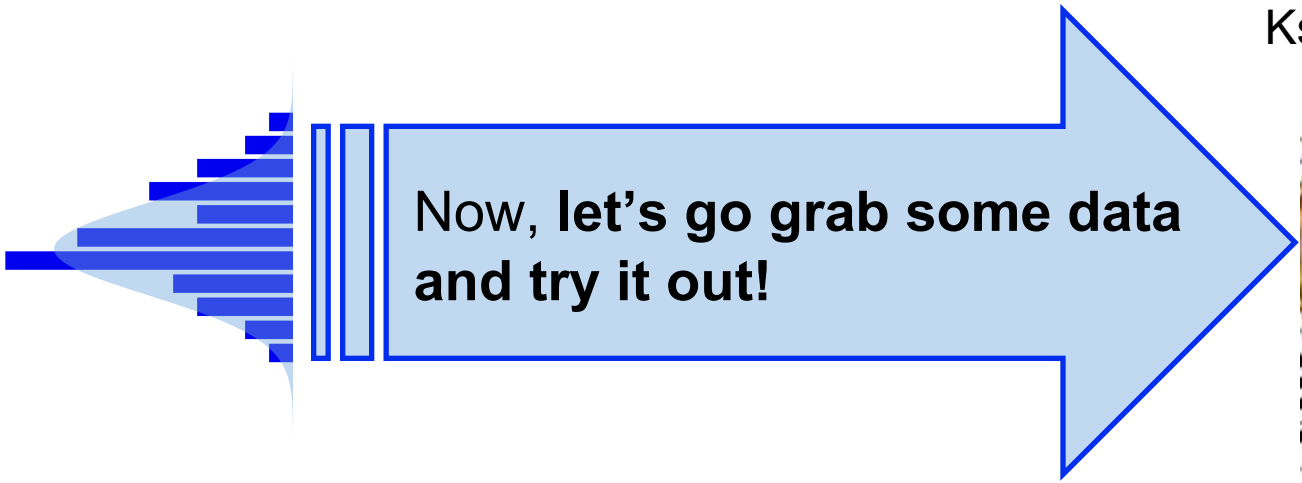
WHAT DOES IT MEAN THOUGH?!?!?!

Synonymous mutations do not affect fitness (no functional changes)

…thus, synonymous mutations are invisible to selection.

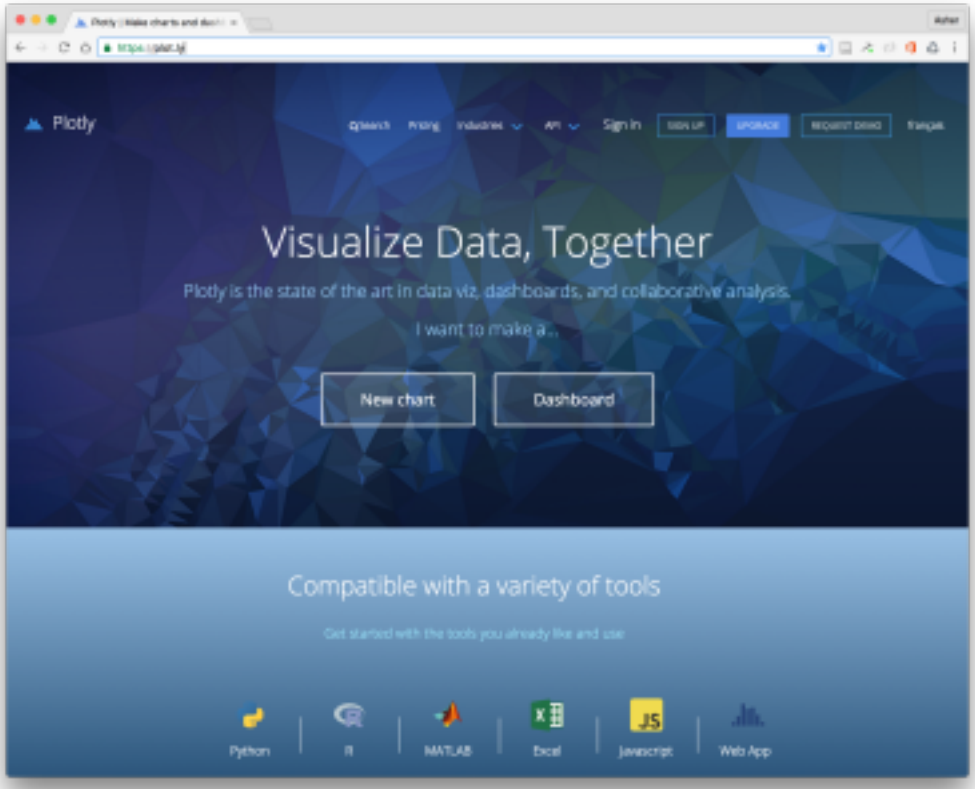
…thus, synonymous mutations can accumulate at a roughly linear rate.

…thus, **more synonymous mutations = more time since gene duplication**



Ks ~= Molecular Clock

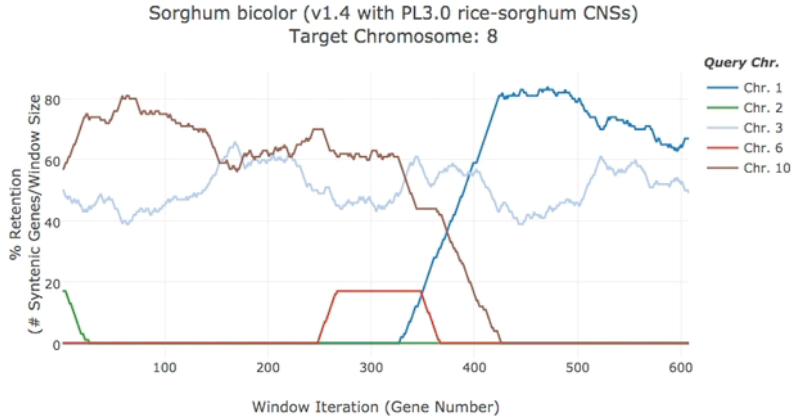
Getting Started with Plot.ly



What IS Plot.ly?

Plot.ly is an open-source* data visualization tool.

… was built using **Python** (Django) & JavaScript.
… offers a web application for **visualization** & **analysis**.
… provides plotting **APIs** for many popular languages.
… plots are **fully interactive**, and rendered with D3.js or WebGL (for 3D).
… free, paid, and on-site offerings.



* well, *mostly* open-source.

Step 1: *Install Plotly API Package*

Install with PIP!

```
(sudo) pip install plotly
```

Update with PIP!

```
(sudo) pip install plotly --upgrade
```

Install with CRAN!

```
install.packages("plotly")
```

Or, with **DevTools & GitHub**!

```
devtools::install_github("ropensci/plotly")
```

Step 2: *Setup your API keys*

...these are necessary for communicating with Plot.ly's servers.

Question: *What is an API?*
Answer: *An "Application Programming Interface"*

Question: *…?*
Slightly More Useful Answer: *A way to interact with someone else's program, using programming languages rather than a graphical user interface*

```
import plotly
plotly.tools.set_credentials_file( \
        username='YourAccount', \
        api_key='YourKey')
```

NOTE: *You only need to do this once on your computer account, even if you use different virtual environments.*

```
Sys.setenv("plotly_username"="user")
Sys.setenv("plotly_api_key"="pass")
```

NOTE: *I don't know if you need to do this every session or only once.*

Step 3: *Compose a Plot*

Figure() - The Plotly "Plot" Object

Composed of two parts: 'Data' and 'Layout'

Data()

Contains the information to be plotted.
Composed of 'trace's

Layout()

Contains information about the plot
i.e. title, labels, fonts, annotations, etc.

Figure()

Combines Data() and Layout()

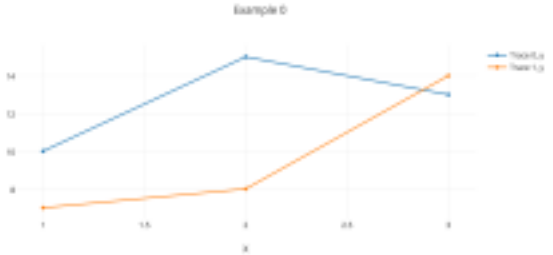
```
import plotly.plotly as py
from plotly.graph_objs import *

trace0 = Scatter(x=[1, 2, 3],y=[10, 15, 13])
trace1 = Scatter(x=[1, 2, 3],y=[7, 8, 14])
my_data = Data([trace0, trace1])

my_layout = Layout(
    title="Example 0",
    xaxis=dict(title="X"))

my_figure = Figure(
    data=my_data, layout=my_layout)

py.plot(my_figure, filename='example0')
```



https://plot.ly/~asherkhb/462/example-0/

Step 3: *Compose a Plot*

plotly charts are described *declaratively*…

1. plotly::plot_ly
2. plotly::add_trace
3. plotly::layout

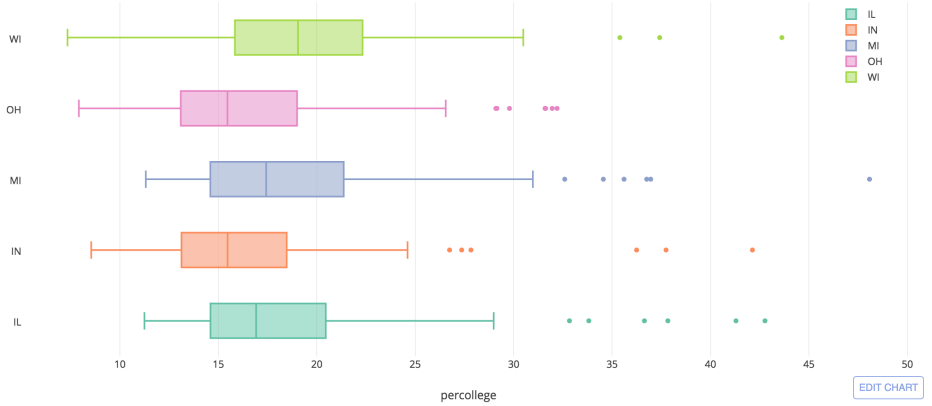
Every aspect of a chart (the colors, the grid-lines, the data, and so on) has a corresponding key in these call signatures.

https://plot.ly/r/reference/

```
library(plotly)
p <- plot_ly(midwest, x = ~percollege, color = ~state, type = "box")
p
```

Copy to clipboard!

Simply printing the plotly object will render the chart locally in your web browser or in the R Studio viewer.



Step 4: *Plot!!*

```
# Import Online Module
import plotly.plotly as py

# Import Offline Module
Import plotly.offline as pyo
```

Regular Python

```
py.plot()    # For ONLINE
pyo.plot()   # For OFFLINE
```

iPython

```
py.iplot()    # ONLINE (still inline)
pyo.iplot()   # OFFLINE (inline)
```

Default is offline, simply print!
Interactive plot will open either in web browser or in Rstudio viewer

```
>>> library(plotly)
>>> p <- plot_ly(midwest,
                 x=~percollege,
                 color = ~state,
                 type = "box")
>>> p
```

To save your plots online to Plotly's servers…

```
>>> plotly_POST(p, filename = "r-
docs/midwest-boxplots")
```

Online vs **Offline** Plotting

Online Plotting

- Data is sent to Plotly's servers, plots are generated and links are returned.
- Embedded plots can be returned to iPython.

Pros

- Ability to adjust look of plots using GUI.
- Easy to share plots with other users.

Cons

- Internet connection required, need to set up API keys.
- Data is public (unless Pro account).

Offline Plotting

- Data is bundled with Plotly's library locally, made available as local .html files or divs.
- Embedded plots are available when using iPython.

Pros

- No internet connection or keys required.
- Data remains private without paying.

Cons

- Must programmatically adjust all visual parameters to achieve desired looks
- Some of Plotly's functionality not available.

Other Features & Benefits of Plotly

➔ Matplotlib converter!

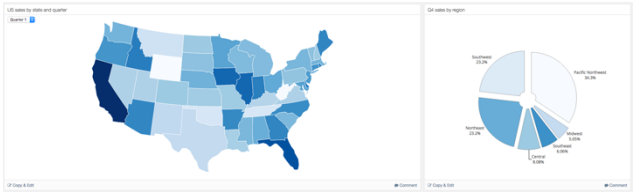
https://plot.ly/matplotlib/

`py.plot_mpl(<mpl_fig>)`

plotly

➔ Live & static dashboards!!

https://plot.ly/dashboards/

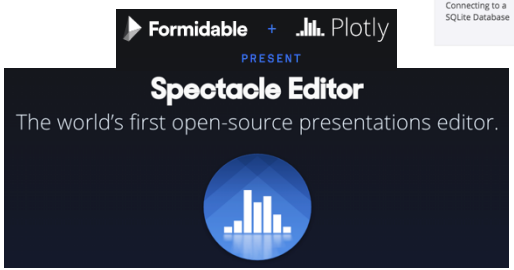


➔ Database connections!*

http://help.plot.ly/database-connectors/

➔ Presentations?!*

https://formidable.com/open-source/spectacle-editor/



➔ Mobile Ready!

* untested by me personally.

Getting more **help**...

Check out the docs for basics & examples...

Python: https://plot.ly/python/

R: https://plot.ly/r/

If you can't figure it out, post to the forums…

http://community.plot.ly/