ggplot2 for microbiome analysis

Presenter: Huyha

Table of contents

Why we need to visualise data?

	mathscore	englishscore	sciencescore	state_of_living_in_usa	age	grade
1	64.4	97.8	94.5	California	18	D
2	89.4	72.7	84.6	New York	16	F
3	70.4	83.9	82.0	Florida	14	B
4	94.2	78.6	99.7	California	16	A
5	97.0	55.1	82.8	New York	14	A
6	52.3	95.0	85.4	Illinois	15	C
7	76.4	62.3	77.2	Florida	18	A
8	94.6	52.1	79.7	Texas	11	F
9	77.6	66.4	64.5	Illinois	14	A
10	72.8	97.7	57.4	California	17	B

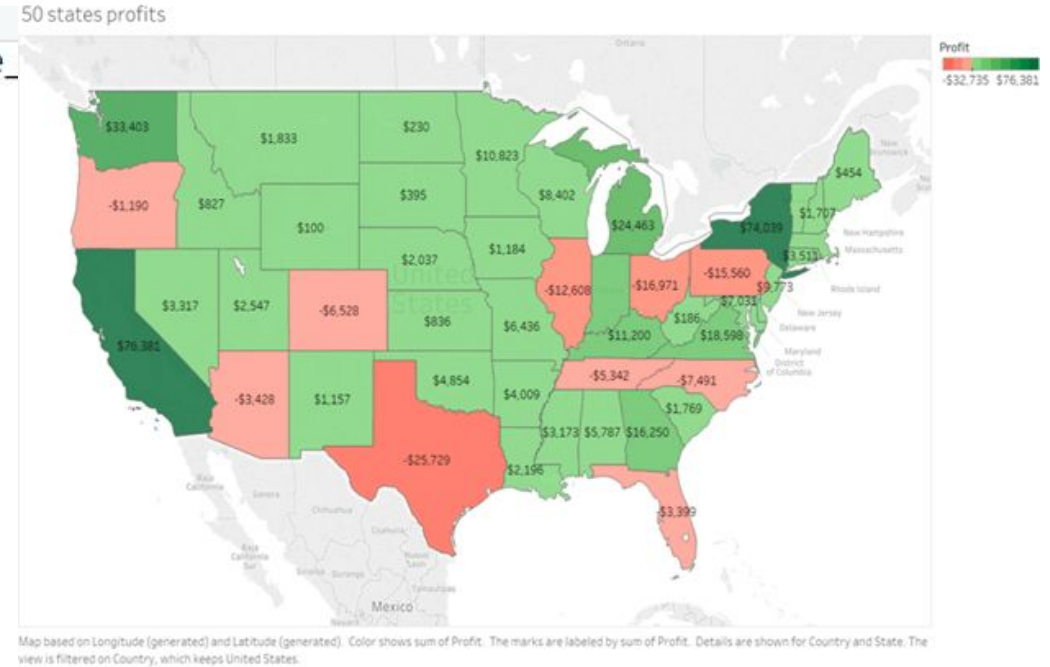
```
>
```

Why we need to visualise data?

R 4.3.3 · ~/

	mathscore	englishscore	sciencescore	state_
1	64.4	97.8	94.5	
2	89.4	72.7	84.6	
3	70.4	83.9	82.0	
4	94.2	78.6	99.7	
5	97.0	55.1	82.8	
6	52.3	95.0	85.4	
7	76.4	62.3	77.2	
8	94.6	52.1	79.7	
9	77.6	66.4	64.5	
10	72.8	97.7	57.4	

```
>
```



Data [edit]

For all four datasets:

Property	Value	Accuracy
Mean of x	9	exact
Sample variance of x: s_x^2		
Mean of y		
Sample variance of y: s_y^2		
Correlation between x and y		
Linear regression line		
Coefficient of determination of th		

	x1	x2	x3	x4	y1	y2	y3	y4
1	10	10	10	8	8.04	9.14	7.46	6.58
2	8	8	8	8	6.95	8.14	6.77	5.76
3	13	13	13	8	7.58	8.74	12.74	7.71
4	9	9	9	8	8.81	8.77	7.11	8.84
5	11	11	11	8	8.33	9.26	7.81	8.47
6	14	14	14	8	9.96	8.10	8.84	7.04
7	6	6	6	8	7.24	6.13	6.08	5.25
8	4	4	4	19	4.26	3.10	5.39	12.50
9	12	12	12	8	10.84	9.13	8.15	5.56
10	7	7	7	8	4.82	7.26	6.42	7.91
11	5	5	5	8	5.68	4.74	5.73	6.89

set <chr>	mean_x <dbl>	mean_y <dbl>	sd_x <dbl>	sd_y <dbl>	cor <dbl>
1	9	7.500909	3.316625	2.031568	0.8164205
2	9	7.500909	3.316625	2.031657	0.8162365
3	9	7.500000	3.316625	2.030424	0.8162867
4	9	7.500909	3.316625	2.030579	0.8165214

4 rows

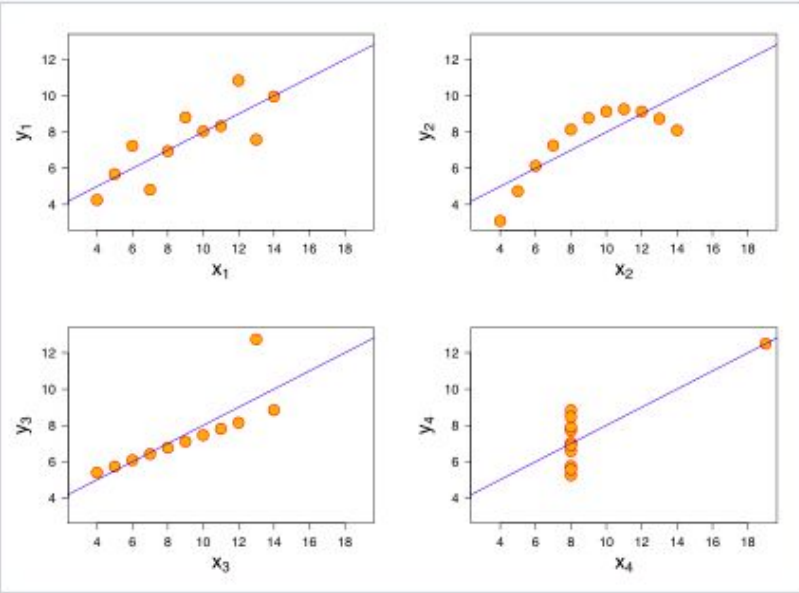
Anscombe's quartet

20 languages

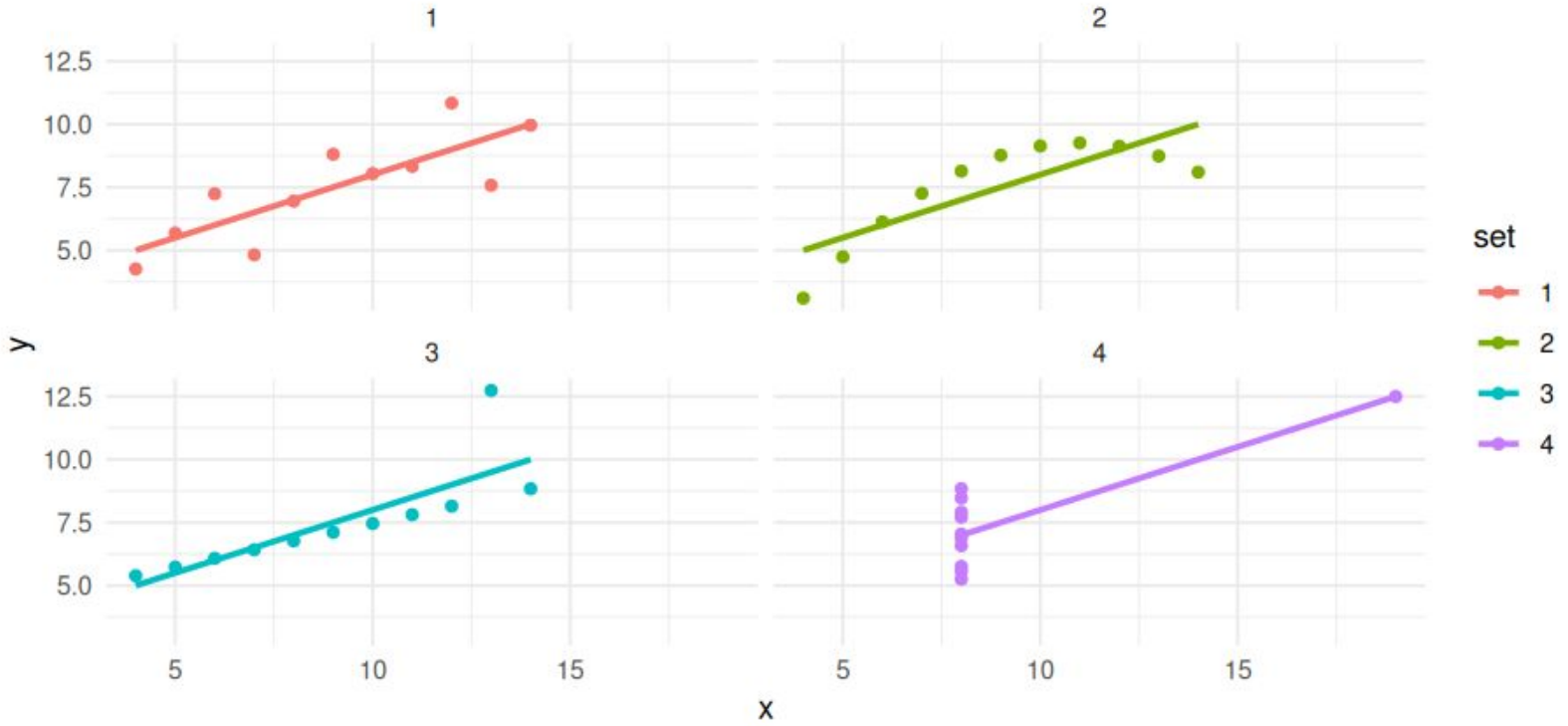
Article Talk Read Edit View history Tools

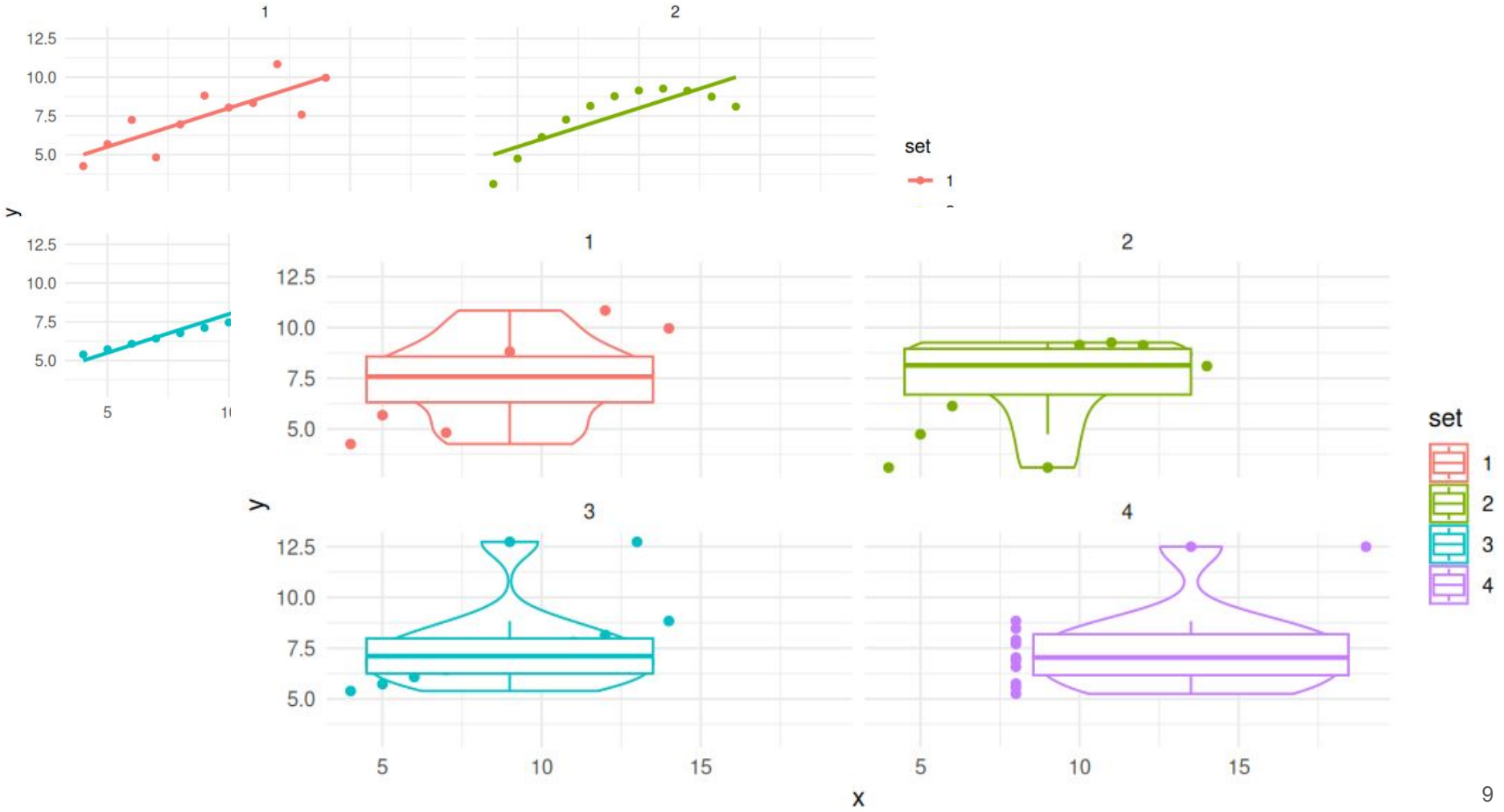
From Wikipedia, the free encyclopedia

Anscombe's quartet comprises four datasets that have nearly identical simple descriptive statistics, yet have very different distributions and appear very different when graphed. Each dataset consists of eleven (*x*, *y*) points. They were constructed in 1973 by the statistician Francis Anscombe to demonstrate both the importance of graphing data when analyzing it, and the effect of outliers and other influential observations on statistical properties. He described the article as being intended to counter the impression among statisticians that "numerical calculations are exact, but graphs are rough".[1]



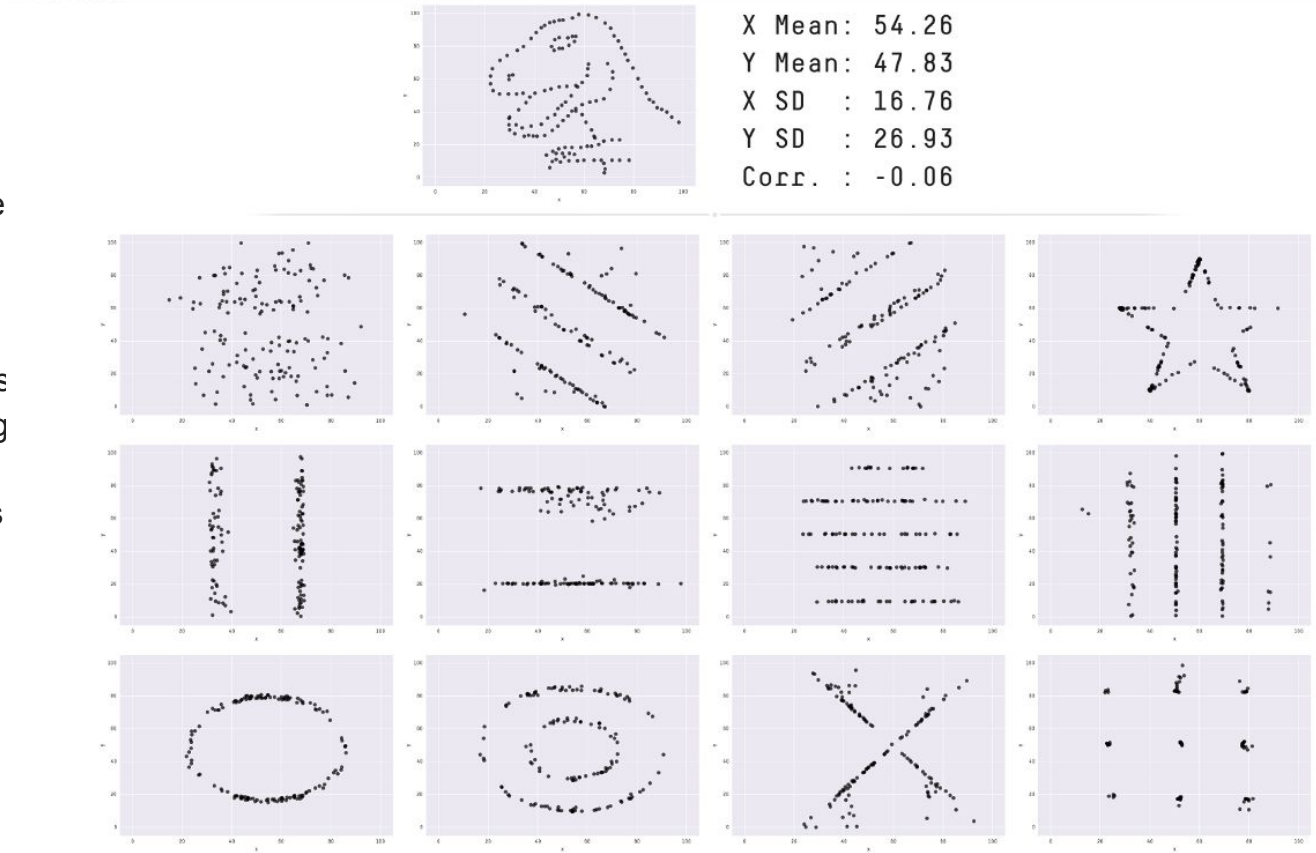
The four datasets composing Anscombe's quartet. All four sets have identical statistical parameters, but the graphs show them to be considerably different


1
2
3
4
12.5
10.0
7.5
5.0
5
10
15
y
x
set
1
2
3
4


1
2
3
4
y
x
set
1
2
3
4
12.5
10.0
7.5
5.0
5
10
15

DATASAURUS

Recently, Albert Cairo created the Datasaurus dataset which urges people to “never trust summary statistics alone; always visualize your data.” While the data exhibits normal-seeming statistics, plotting the data reveals a picture of a dinosaur. Inspired by Anscombe’s Quartet and Datasaurus, we present, The Datasaurus Dozon (download.csv).



https://www.research.autodesk.com/publications/same-stats-different-graphs/

2. How to draw in ggplot

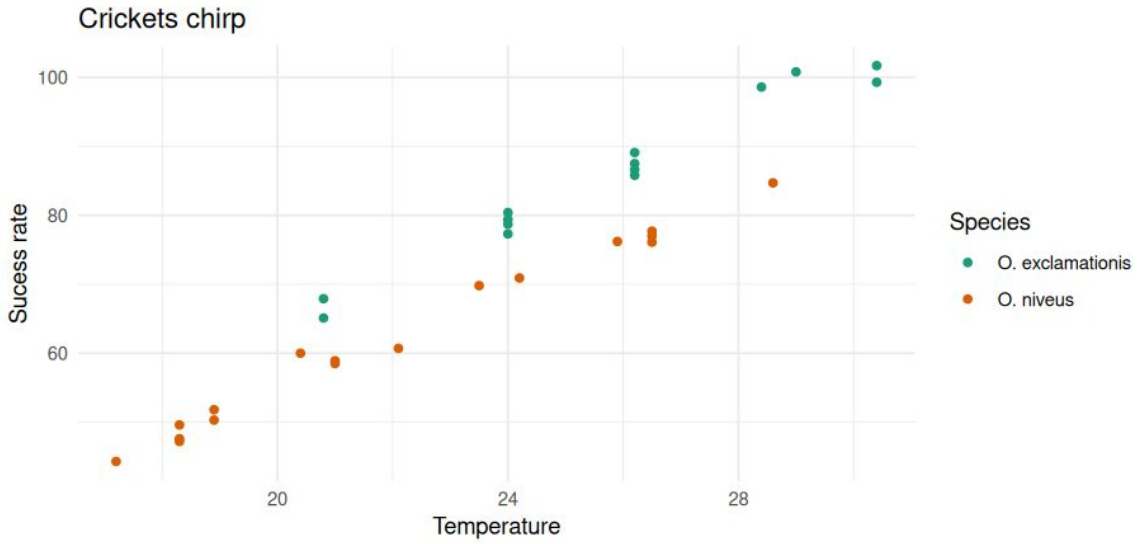
1. Import-data
2. Data manipulation
3. Basic plot
4. Extend more detail
5. Save file

Crickets chirp relate to temp

	species	temp	rate
	<fct>	<dbl>	<dbl>
1	O. exclamationis	20.8	67.9
2	O. exclamationis	20.8	65.1
3	O. exclamationis	24	77.3
4	O. exclamationis	24	78.7
5	O. exclamationis	24	79.4
6	O. exclamationis	24	80.4
7	O. exclamationis	26.2	85.8
8	O. exclamationis	26.2	86.6
9	O. exclamationis	26.2	87.5
10	O. exclamationis	26.2	89.1

i 21 more rows

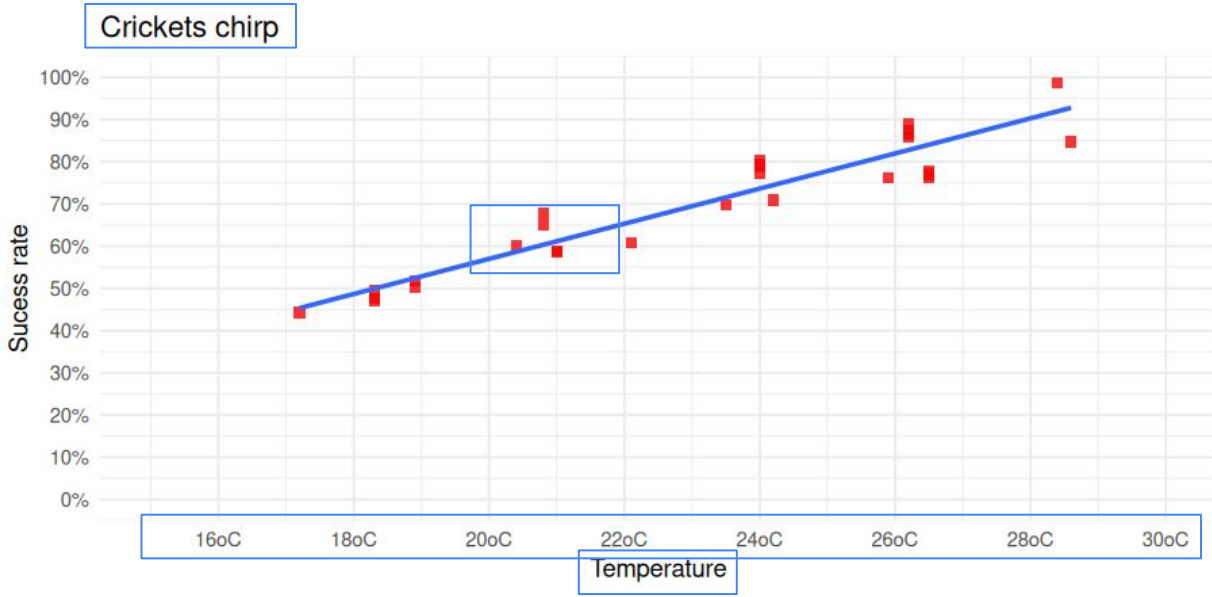
2. Basic plot




```
> ggplot(crickets, aes(x=temp,y=rate,
color=species))+
+  geom_point() +
+  labs(x="Temperature", y="Success rate",
+       color="Species",
+       title="Crickets chirp")+
+  scale_color_brewer(palette = "Dark2")
```

2. Extend more detail

```
ggplot(crickets, aes(x=temp,y=rate,))+
  geom_point(color="red",
             size=2,
             shape="square",
             alpha=0.8) +
  labs(x="Temperature", y="Success rate",
       color="Species",
       title="Crickets chirp") +
  scale_x_continuous(
    limits = c(15,30),
    breaks = seq(10,30, by =2),
    labels = function(x) paste0(x,"oC")
    )+
  scale_y_continuous(
    limits = c(0,100),
    breaks = seq(0,100, by=10),
    labels = function(x) paste0(x,"%")
  )+
  geom_smooth(method="lm",se=FALSE)
```

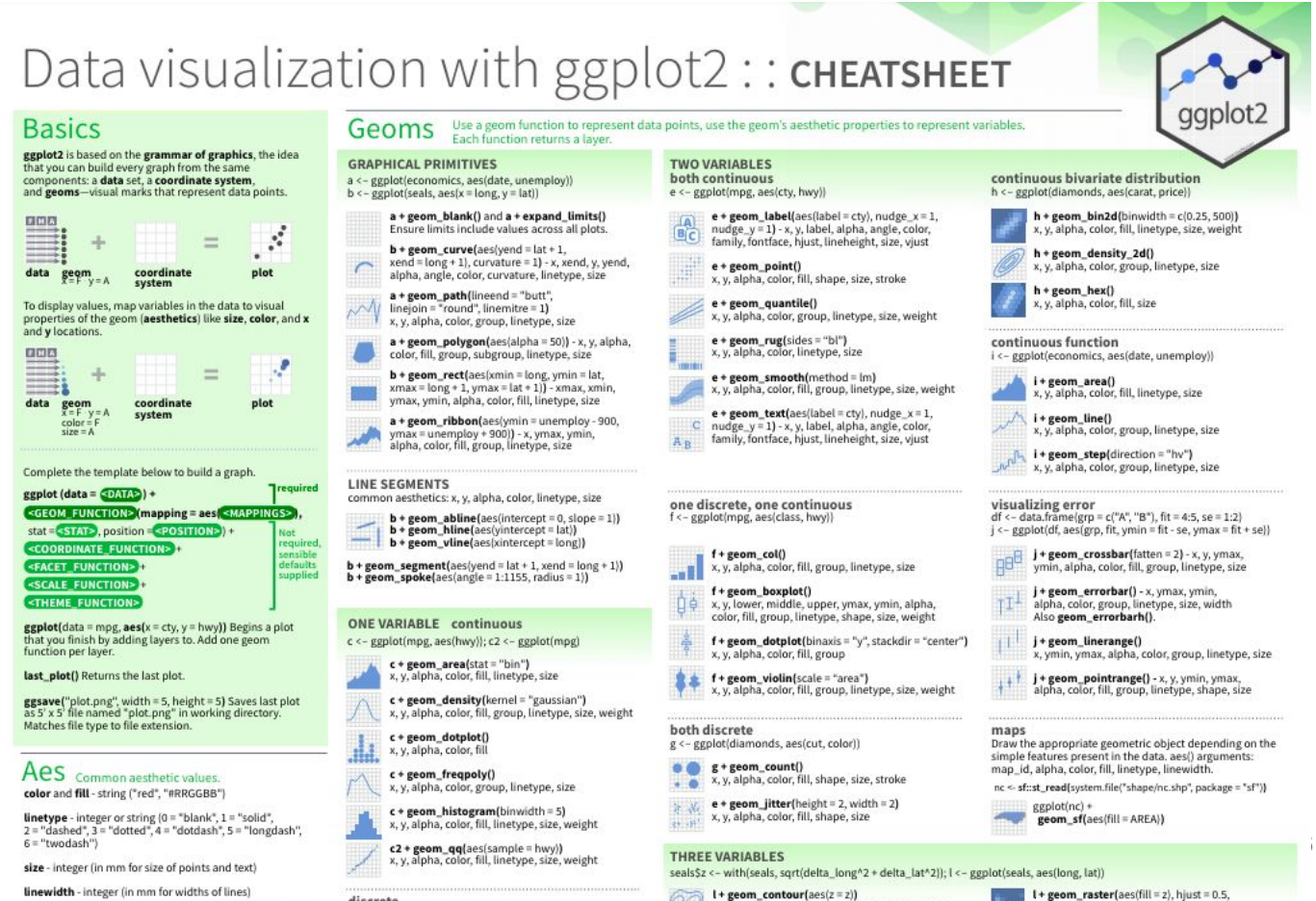


How to learn ggplot2 by yourself?

1. Cheatsheet
2. Help function in Rstudio
3. Use AI to help
4. Book - (recommended)

Introduction to ggplot(recommend)

1. Cheatsheet



Data visualization with ggplot2 : : CHEATSHEET

ggplot2

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.

data | geom (x = F, y = A) | coordinate system | plot

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

data | geom (x = F, y = A, color = F, size = A) | coordinate system | plot

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +                                   required
<GEOM_FUNCTION> (mapping = aes(<MAPPINGS>),
  stat = <STAT>, position = <POSITION>) +                  Not required, sensible defaults supplied
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

ggplot(data = mpg, **aes(**x = cty, y = hwy**))** Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5**)** Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

size - integer (in mm for size of points and text)

linewidth - integer (in mm for widths of lines)

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))
```

a + geom_blank() and **a + expand_limits()** Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1**)** - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1**)** x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)**)** - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)**)** - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)**)** - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1)**)**
b + geom_hline(aes(yintercept = lat)**)**
b + geom_vline(aes(xintercept = long)**)**

b + geom_segment(aes(yend = lat + 1, xend = long + 1)**)**
b + geom_spoke(aes(angle = 1:1155, radius = 1)**)**

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
```

c + geom_area(stat = "bin"**)** x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian"**)** x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot() x, y, alpha, color, fill

c + geom_freqpoly() x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5**)** x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)**)** x, y, alpha, color, fill, linetype, size, weight

TWO VARIABLES

both continuous

```
e <- ggplot(mpg, aes(cty, hwy))
```

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point() x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile() x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)** x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)** x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous

```
f <- ggplot(mpg, aes(class, hwy))
```

f + geom_col() x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center"**)** x, y, alpha, color, fill, group

f + geom_violin(scale = "area"**)** x, y, alpha, color, fill, group, linetype, size, weight

both discrete

```
g <- ggplot(diamonds, aes(cut, color))
```

g + geom_count() x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2**)** x, y, alpha, color, fill, shape, size

THREE VARIABLES

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))
```

l + geom_contour(aes(z = z)**)**

l + geom_raster(aes(fill = z), hjust = 0.5,

continuous bivariate distribution

```
h <- ggplot(diamonds, aes(carat, price))
```

h + geom_bin2d(binwidth = c(0.25, 500)**)** x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d() x, y, alpha, color, group, linetype, size

h + geom_hex() x, y, alpha, color, fill, size

continuous function

```
i <- ggplot(economics, aes(date, unemploy))
```

i + geom_area() x, y, alpha, color, fill, linetype, size

i + geom_line() x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv"**)** x, y, alpha, color, group, linetype, size

visualizing error

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))
```

j + geom_crossbar(fatten = 2**)** - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width Also **geom_errorbarh()**.

j + geom_linerange() x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

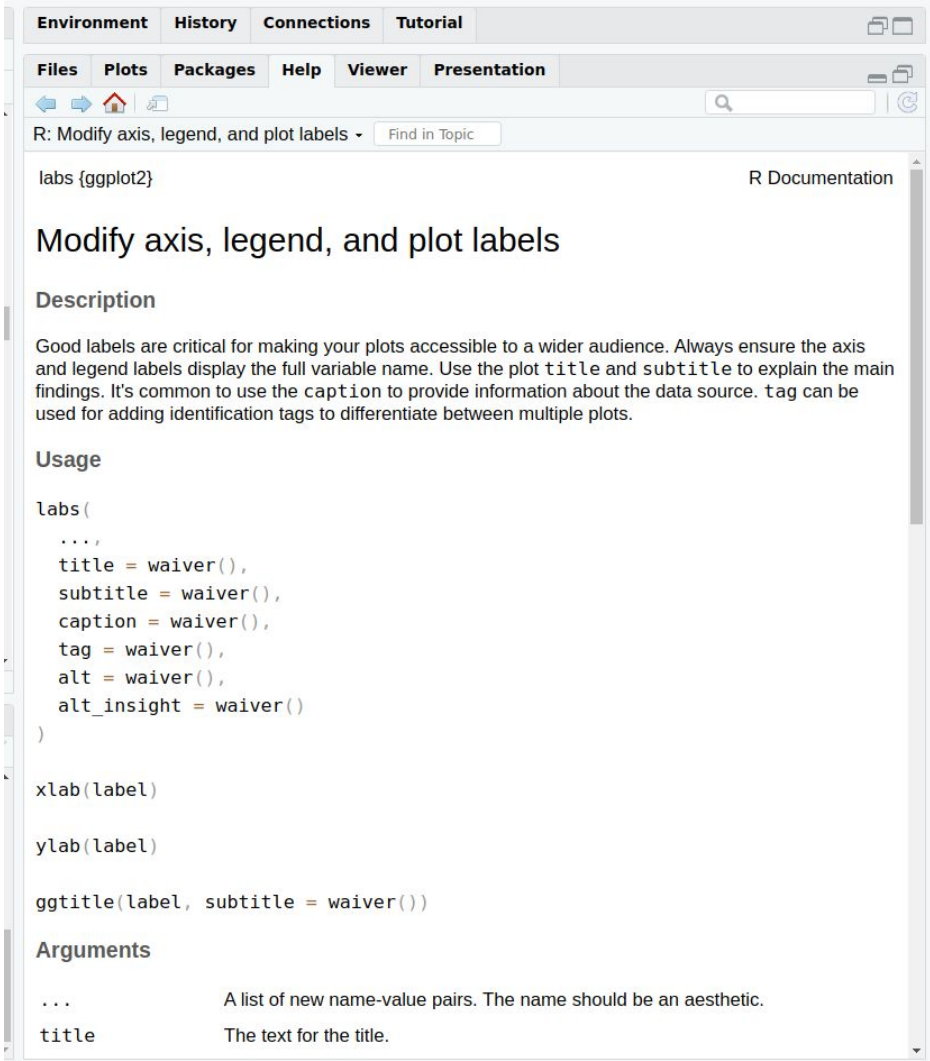
Draw the appropriate geometric object depending on the simple features present in the data. aes() arguments: map_id, alpha, color, fill, linetype, linewidth.

```
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"))
ggplot(nc) +
  geom_sf(aes(fill = AREA))
```

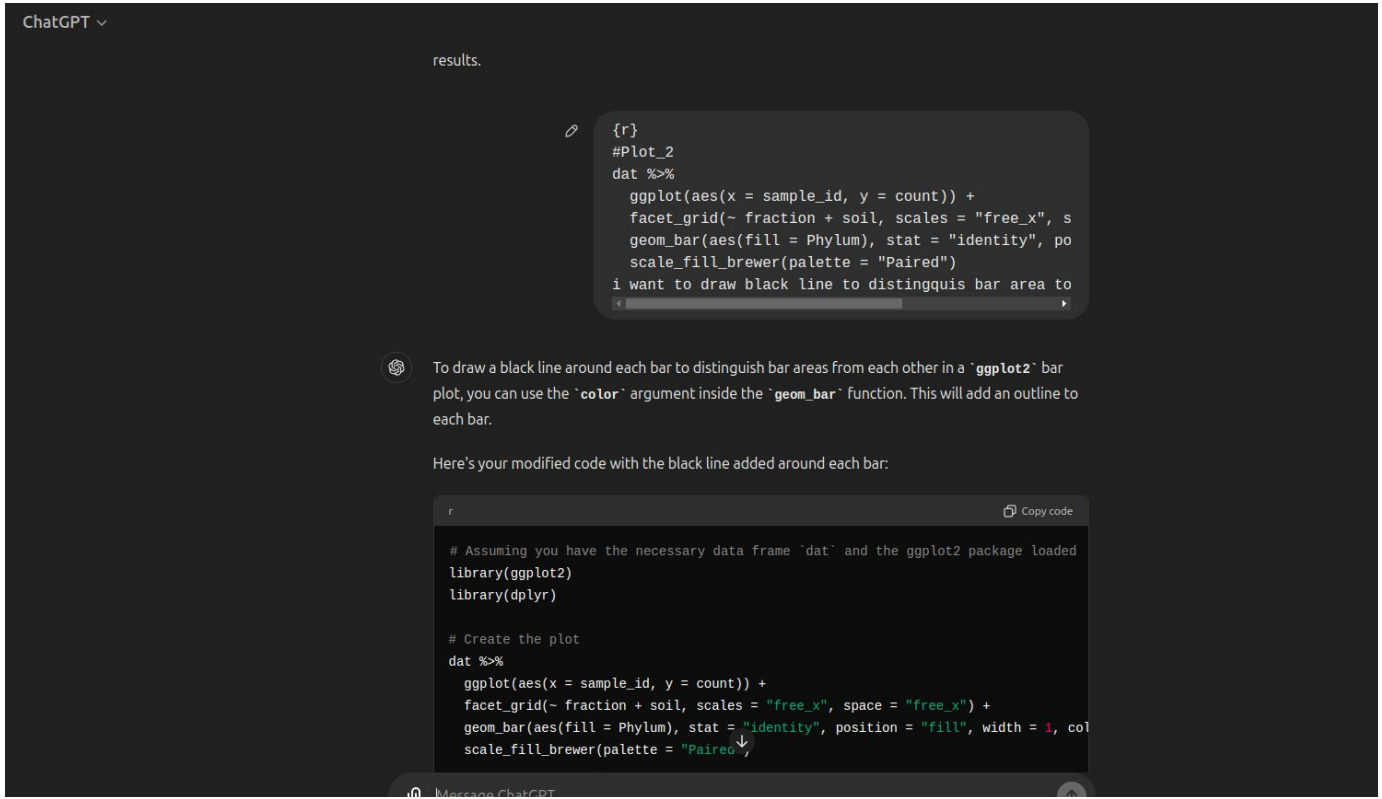
2. Helps in Rstudio

?<command>

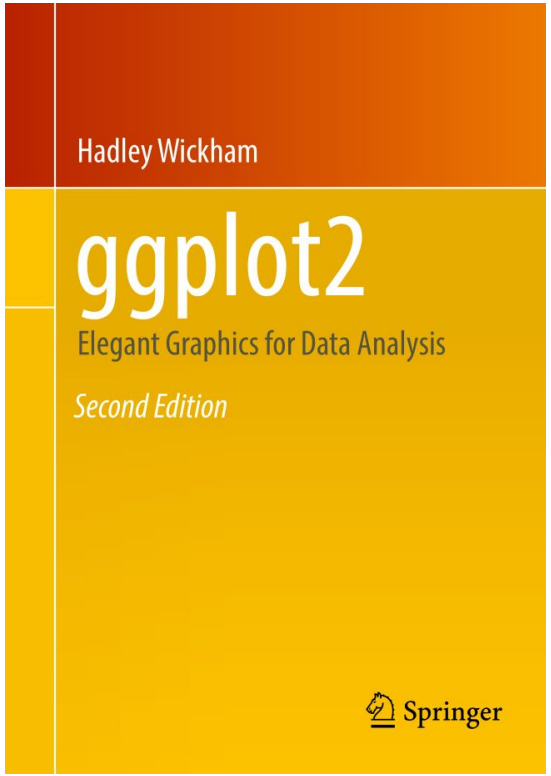
Example: ?labs

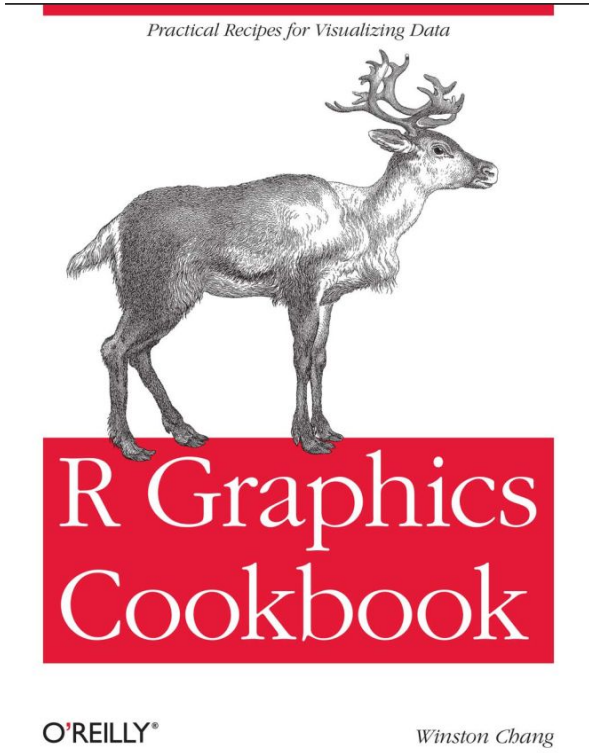


3. USE AI to learn



4. Book to learn





Practice is the key to success

II. Some addition function

Reshape from Long

to Wide and back

country	year	metric
x	1960	10
x	1970	13
x	2010	15
y	1960	20
y	1970	23
y	2010	25
z	1960	30
z	1970	33
z	2010	35

```
pivot_wider(names_from = “year”,
            names_prefix = “yr”,
            values_from = “metric”)
```

country	yr1960	yr1970	yr2010
x	10	13	15
y	20	23	25
z	30	33	35

```
pivot_longer(cols = yr1960:yr2010,
             names_to = “year”,
             names_prefix = “yr”
             values_to = “metric”)
```

```
  subject measurement1 measurement2 measurement3
1       A           10           15           20
2       B           20           25           30
3       C           30           35           40
>
```

```
  subject  measurement value
1       A measurement1    10
2       B measurement1    20
3       C measurement1    30
4       A measurement2    15
5       B measurement2    25
6       C measurement2    35
7       A measurement3    20
8       B measurement3    30
9       C measurement3    40
```

Reshape from wide to long format form **tidyr**

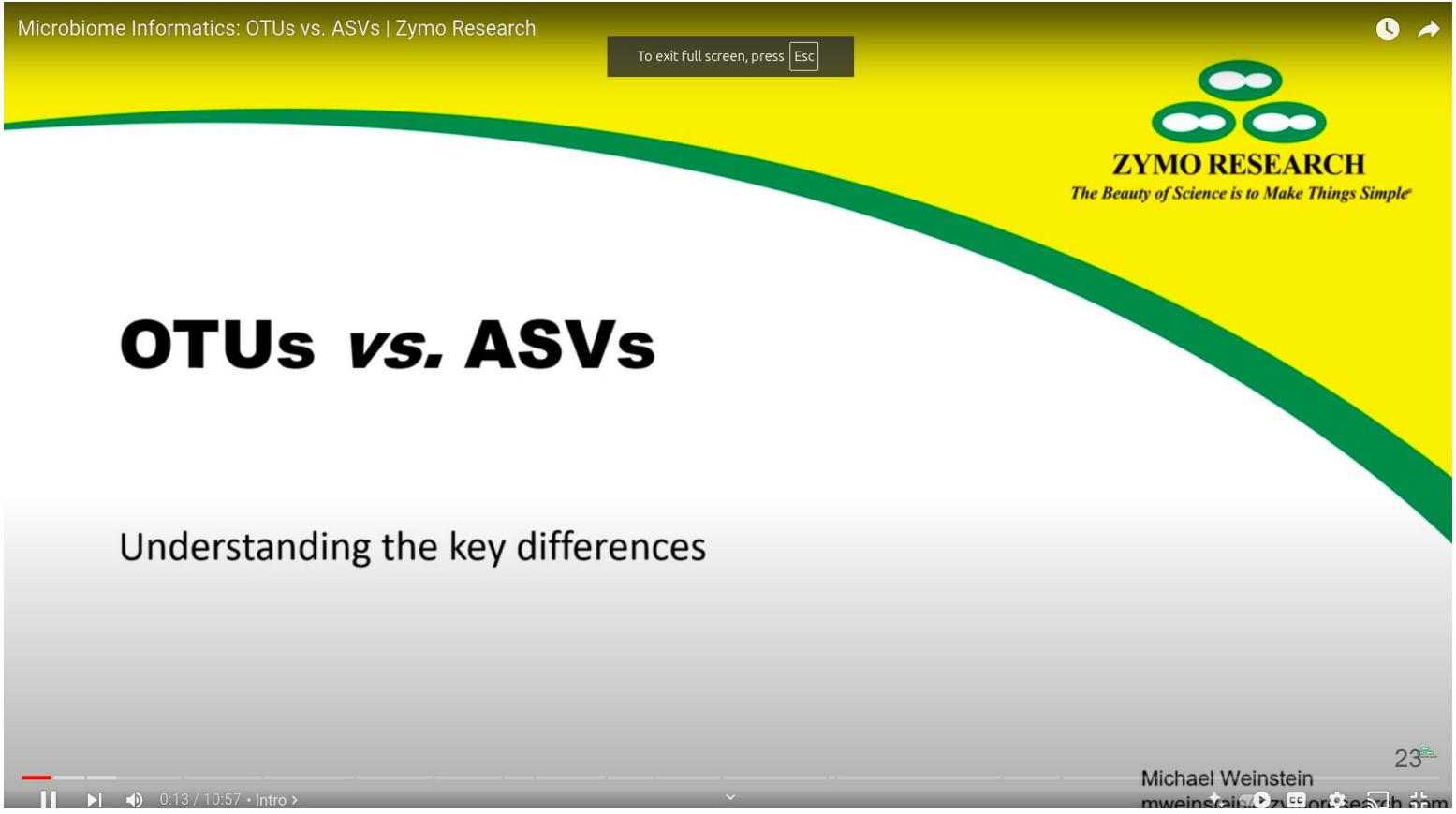
```
df_long1 <- df %>%
  pivot_longer(
    cols = starts_with("measurement"),
    names_to = "measurement",
    values_to = "value"
  )
```

Using melt from package **reshape2**

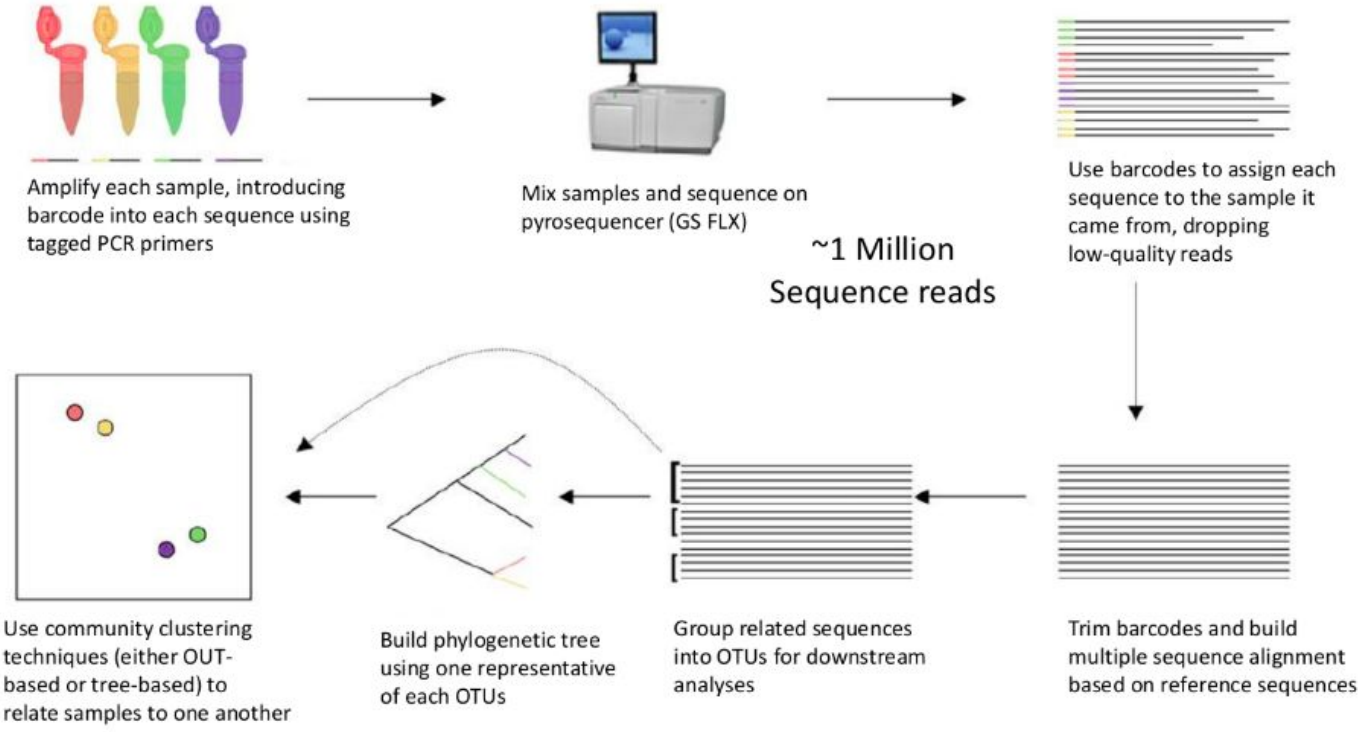
```
df_long2 <- melt(df, id.vars = "subject",
                 variable.name = "measurement",
                 value.name = "value")
print(df_long2)
```

III. OTU vs AVS

Sourse: https://www.youtube.com/watch?v=azI9taClDhQ

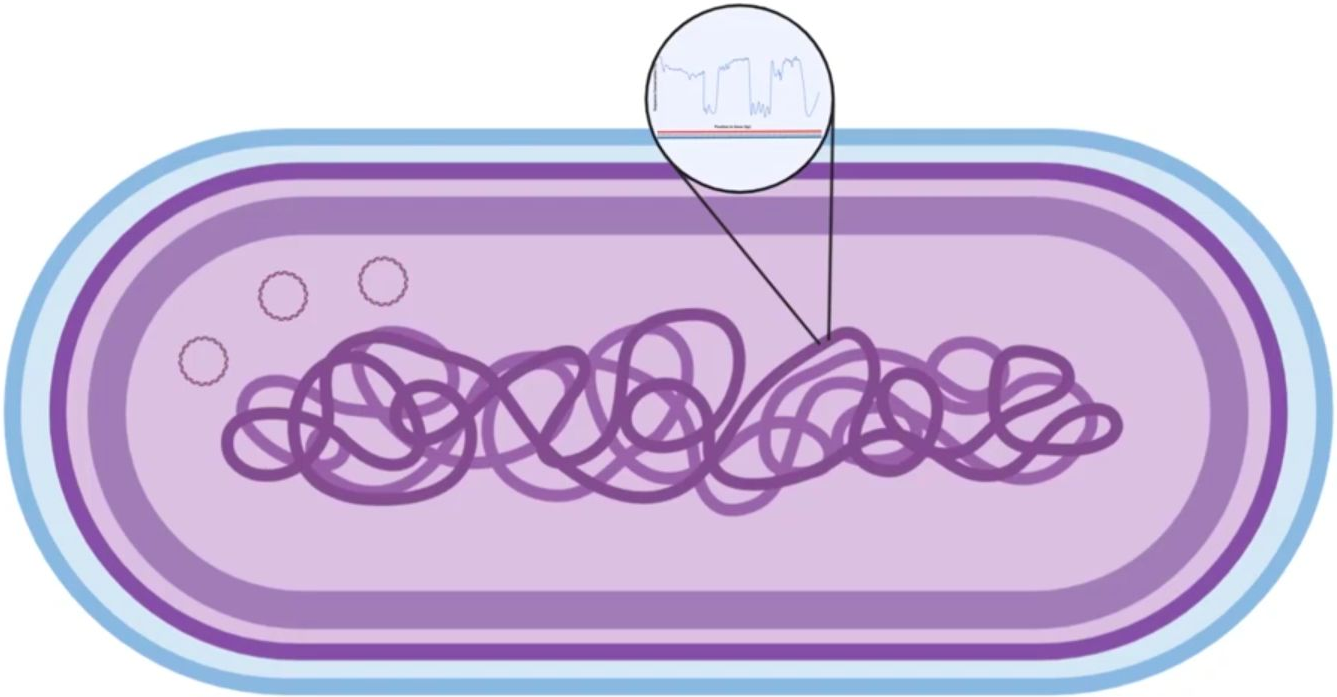


Overview of workflow with amplicon 16S metagenomics



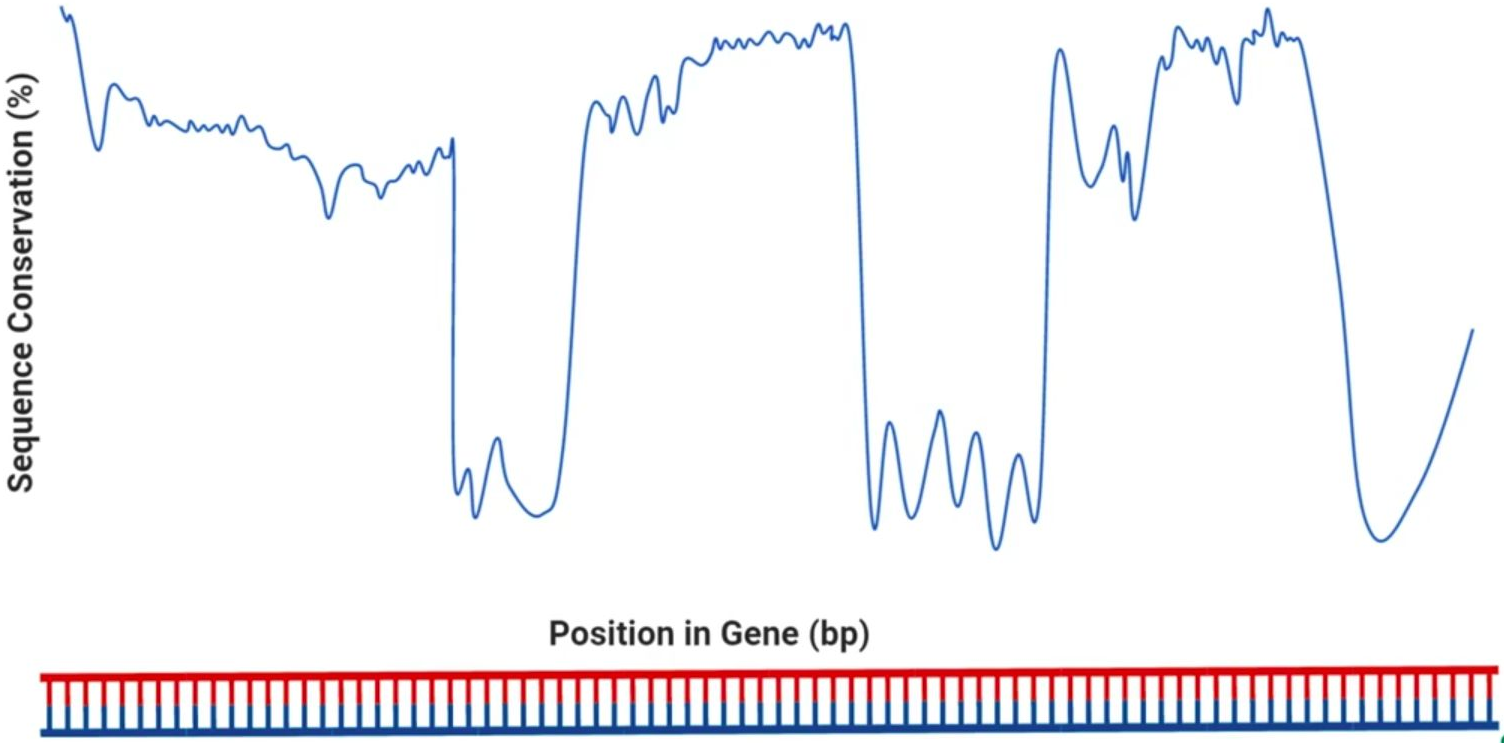
Hamady & Knight 2009 Genome Res. 19: 1141

Targeted Sequencing



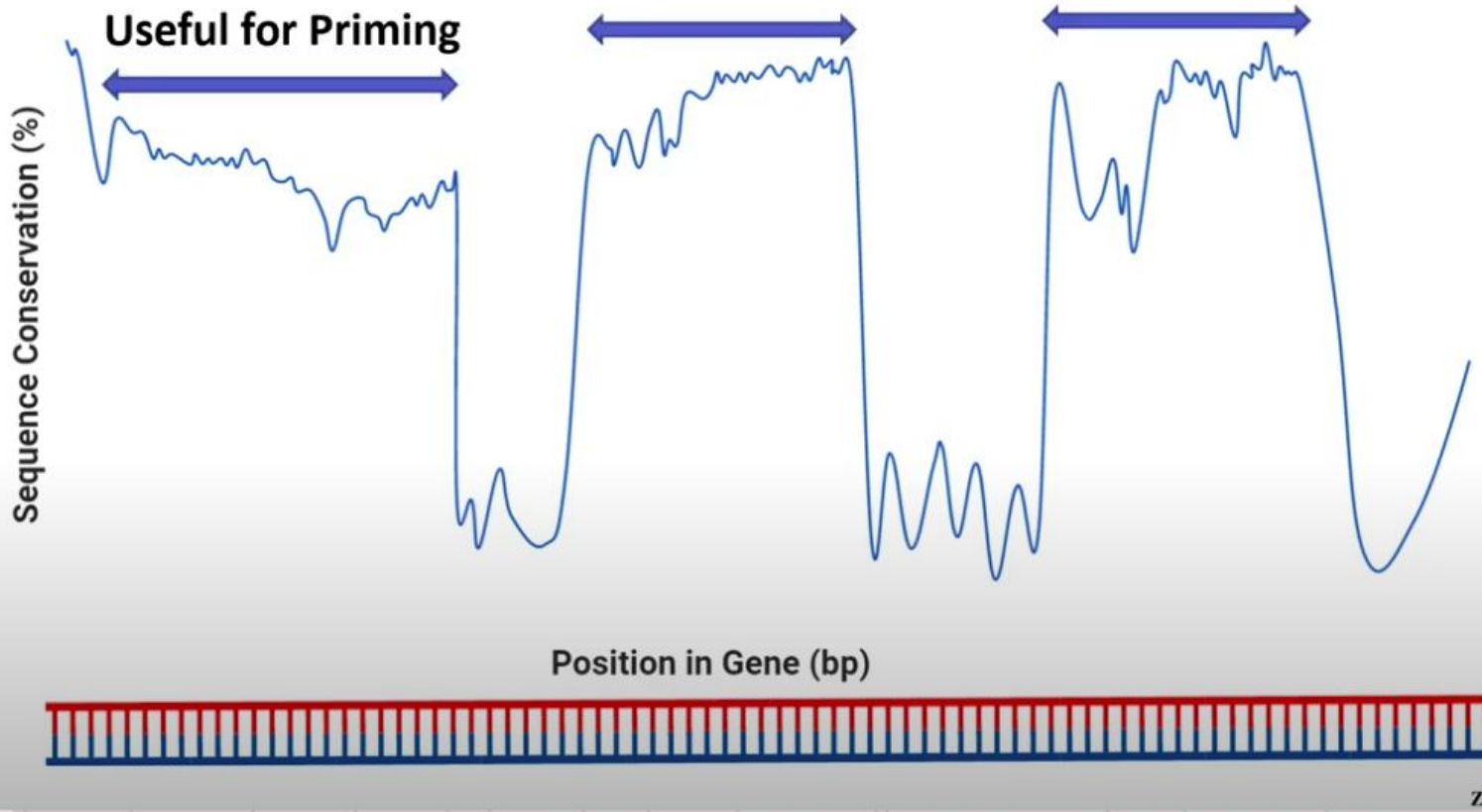
ZYMO RESEARCH

A Potential Target for Sequencing

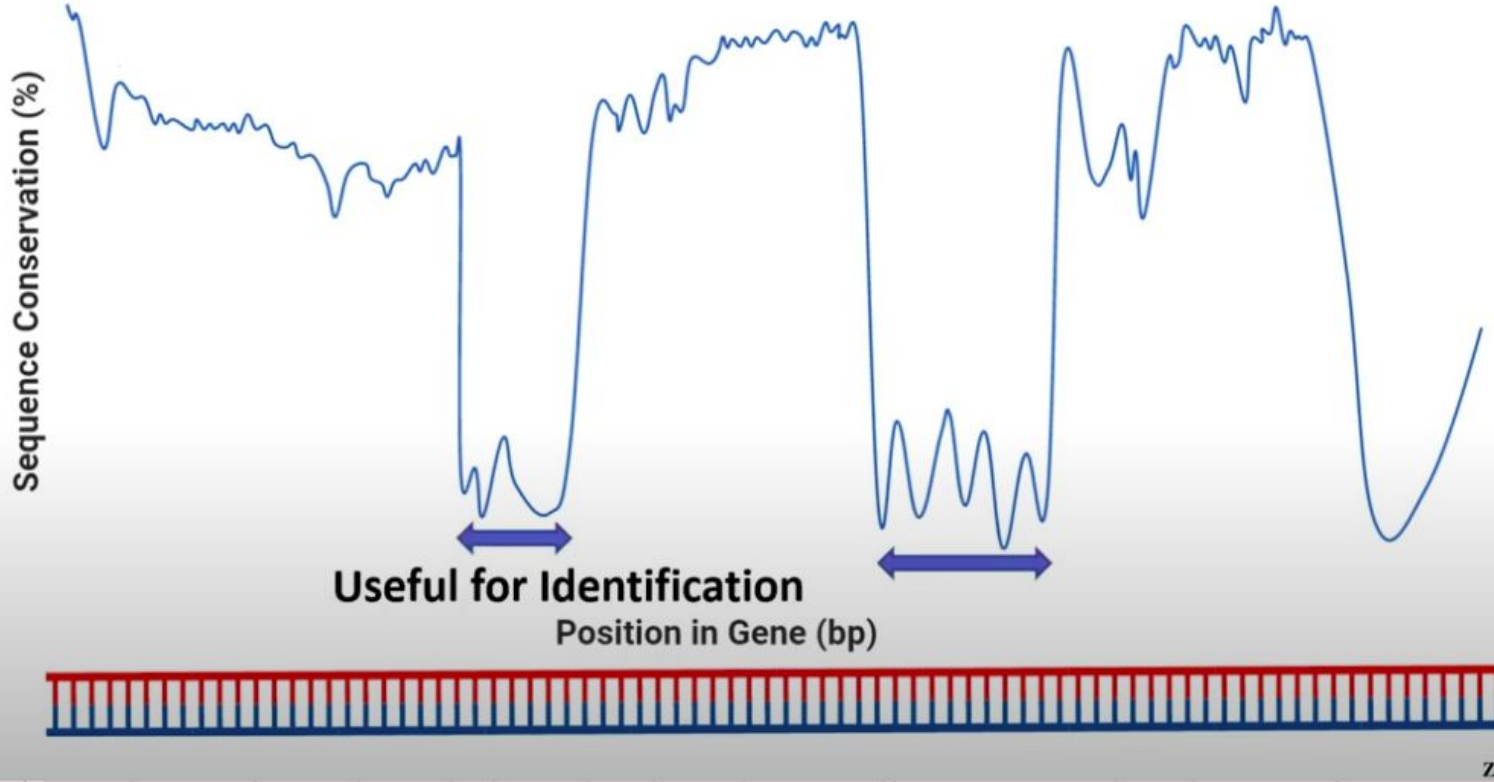


ZYMO RESEARCH

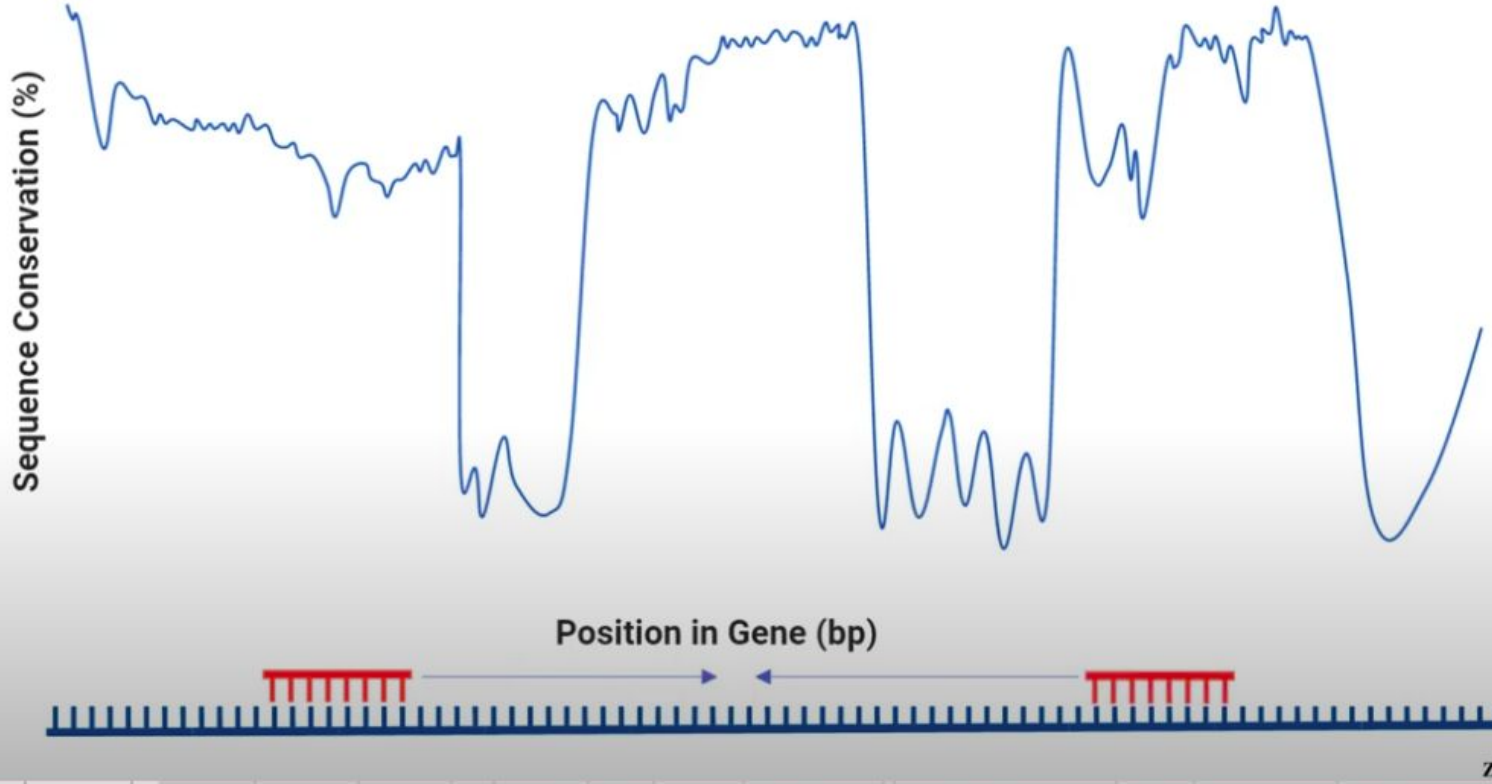
A Potential Target for Sequencing



A Potential Target for Sequencing



A Potential Target for Sequencing



Finding an Ideal Target Sequence

What to look for in a targeted sequencing gene

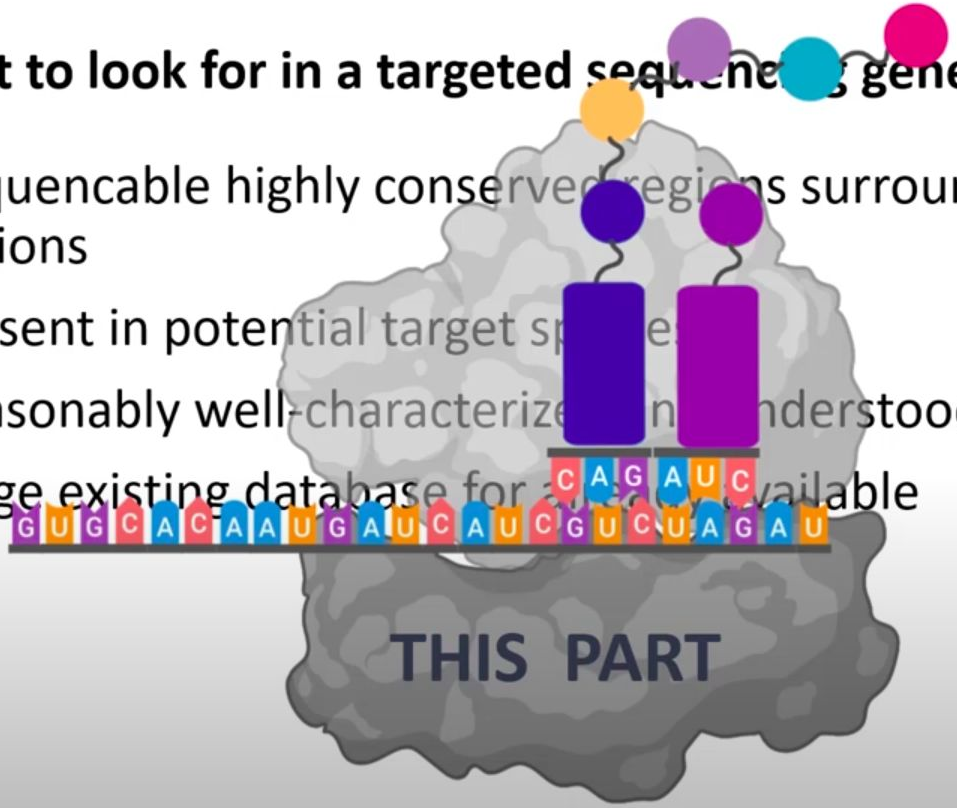
- Sequencable highly conserved regions surrounding variable regions
- Present in potential target species
- Reasonably well-characterized and understood
- Large existing database of reference sequences already available

ZYMO RESEARCH

Finding an Ideal Target Sequence

What to look for in a targeted sequencing gene

- Sequencable highly conserved regions surrounding variable regions
- Present in potential target species
- Reasonably well-characterized and understood
- Large existing database for analysis available



ZYMO RESEARCH

16S Sequencing Challenges

Targeted sequence with a few bases differentiating species

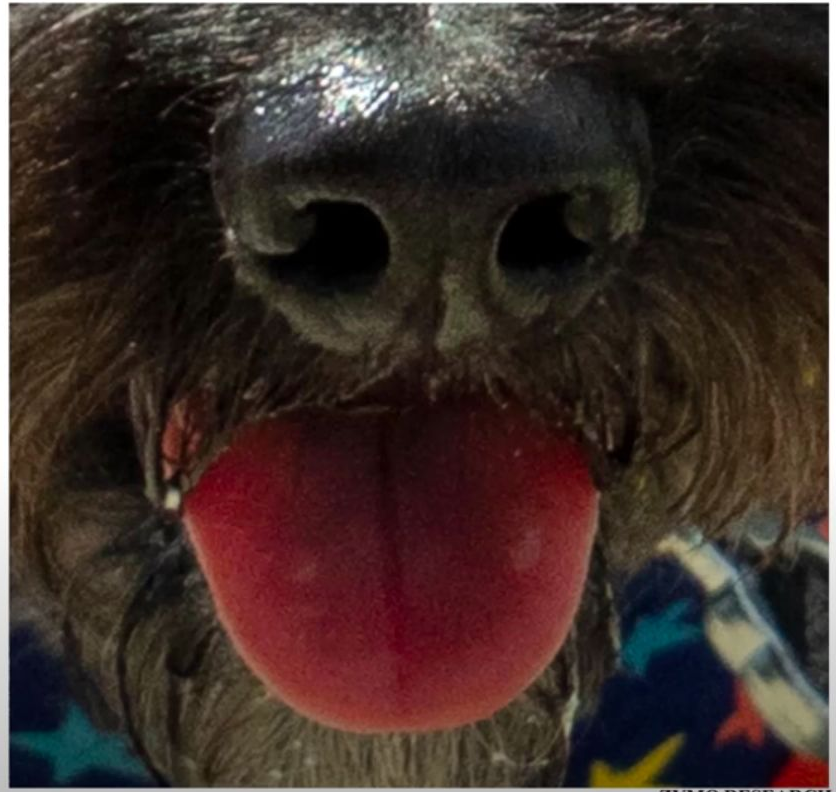
- Sequencing is imperfect
  - Illumina usually makes some base call errors
  - Nanopore makes more
  - Errors are not necessarily evenly-distributed
- We do not want errors to be confused with real diversity/new species

ZYMO RESEARCH

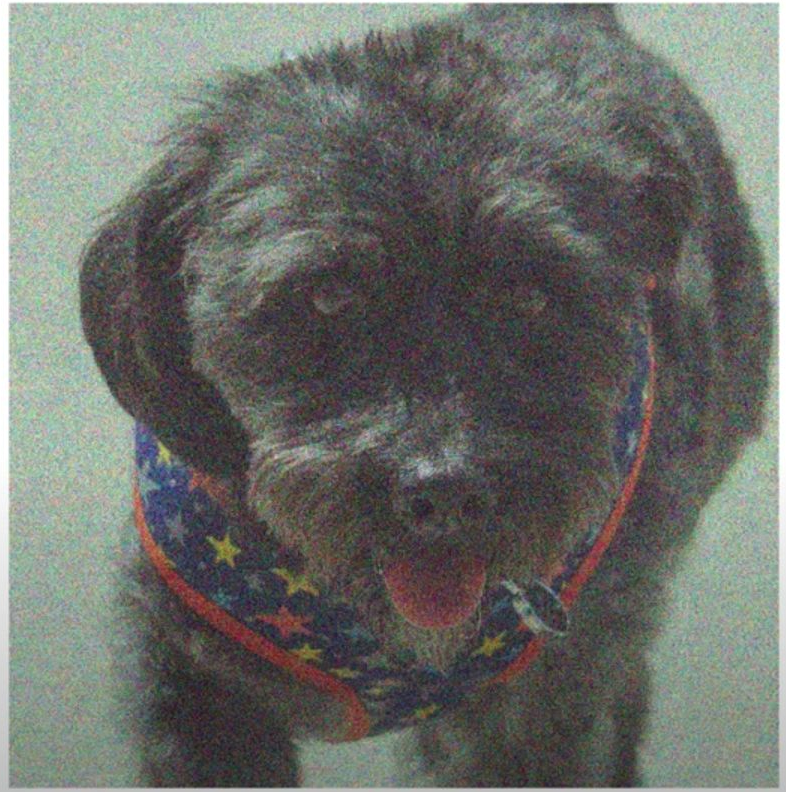
The True Image

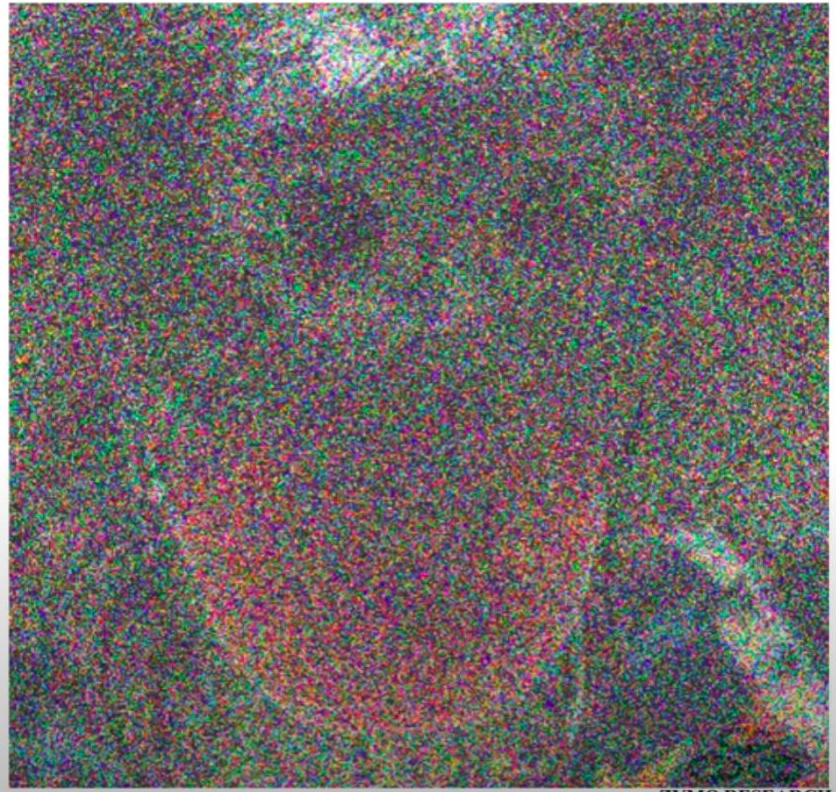
ZYMO RESEARCH

The True Image

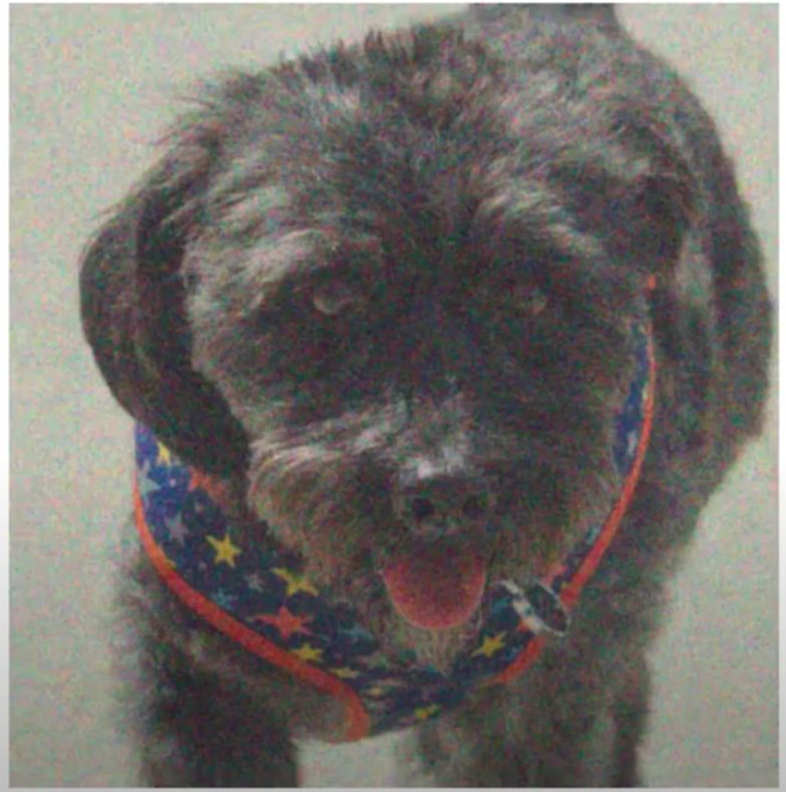


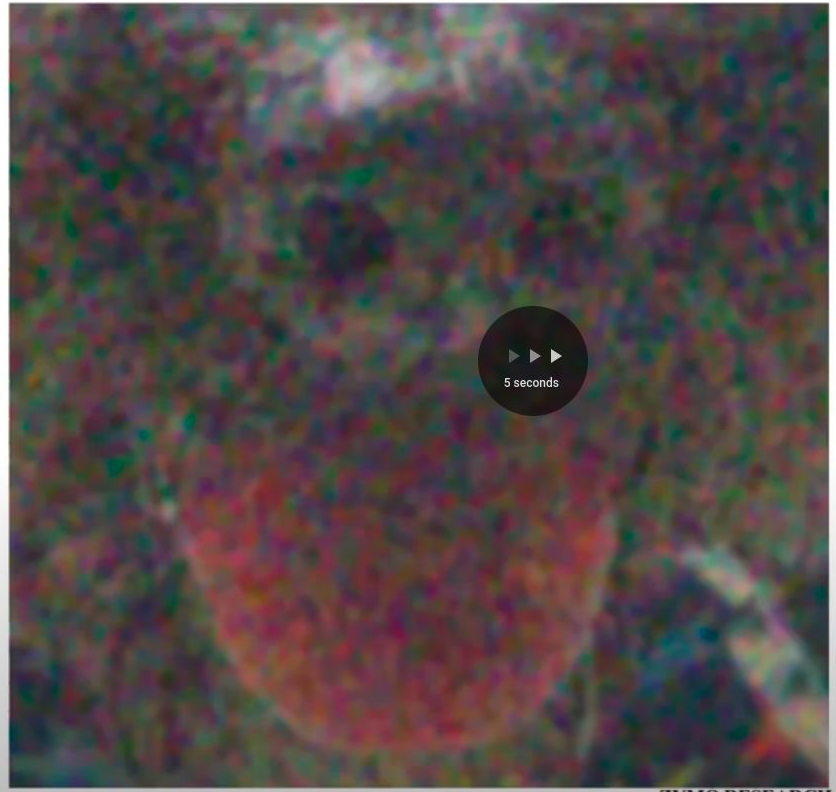
The Addition of Noise





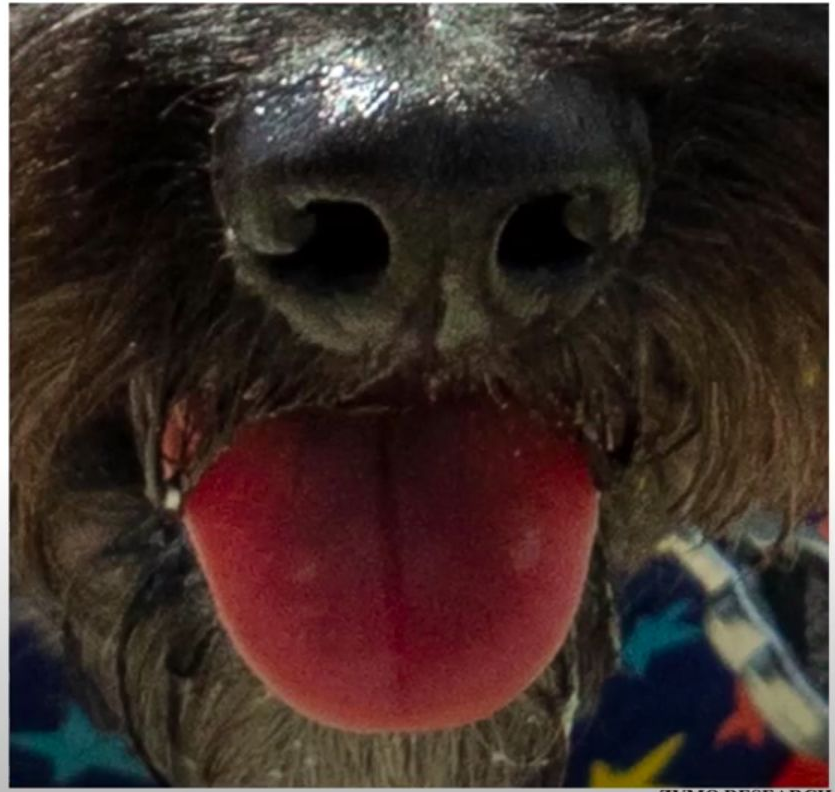
Blur: Loss of Detail and Noise



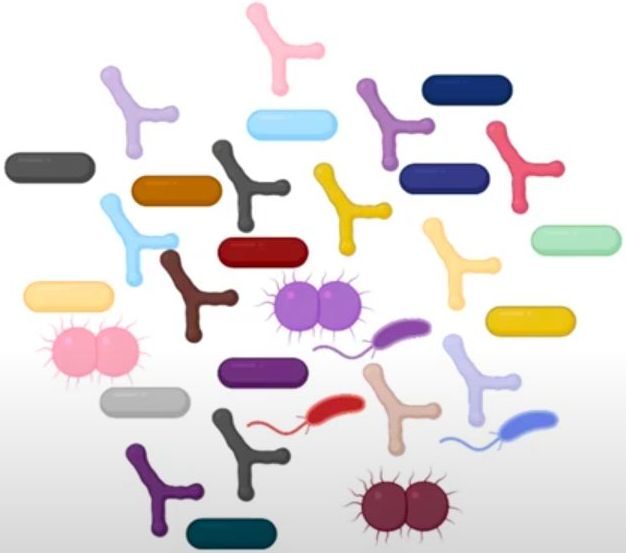


ZYMO RESEARCH

The True Image



Microbes Need to Be Organized/grouped



ZYMO RESEARCH

A Quick Note

FOR OUR PURPOSES:

 **= 1 sequence and one species**

A Quick Note

FOR OUR PURPOSES:

= 1 sequence and one species

REAL BACTERIA

One species can have multiple, different copies of a gene

A Quick Note

FOR OUR PURPOSES:

= 1 sequence and one species

REAL BACTERIA

One species can have multiple, different copies of a gene

Two similar species may share an identical sequence

ZYMO RESEARCH

A Quick Note

FOR OUR PURPOSES:

= 1 sequence* and one species
***May contain errors**

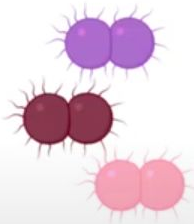
REAL BACTERIA

One species can have multiple, different copies of a gene

Two similar species may share an identical sequence

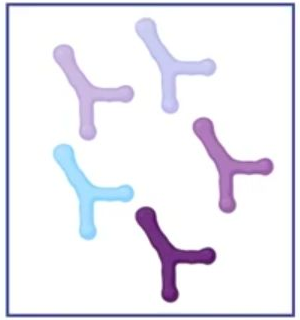
ZYMO RESEARCH

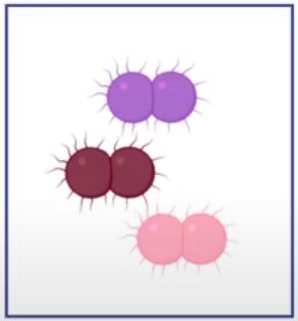
Clustering (Blurring)

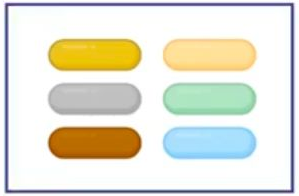


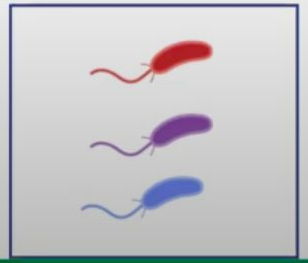
ZYMO RESEARCH

Clustering



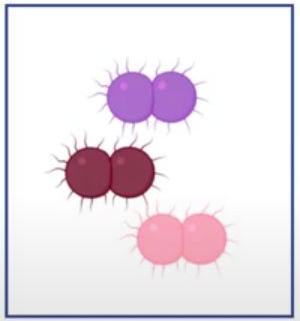


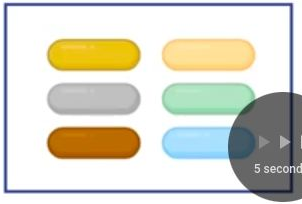


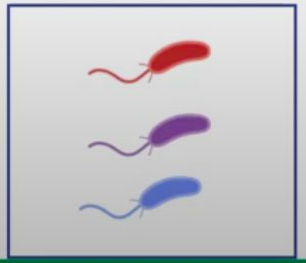


ZYMO RESEARCH

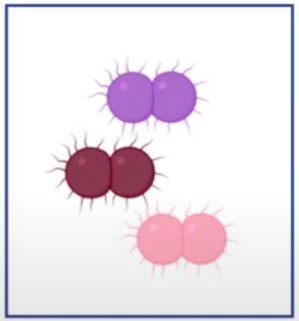
Clustering (*De Novo*)

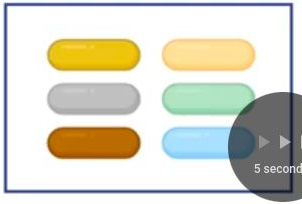


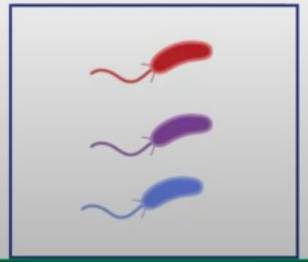




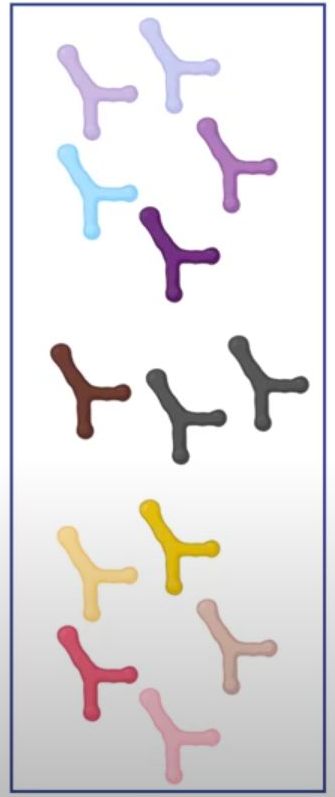
Clustering (*De Novo*)

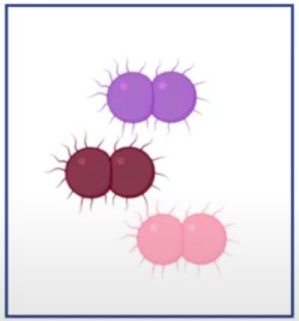


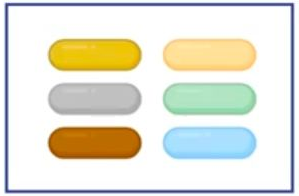


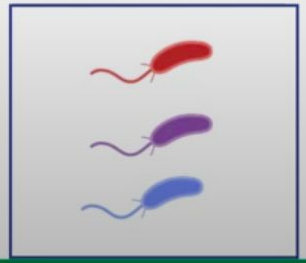


Clustering (*De Novo*)

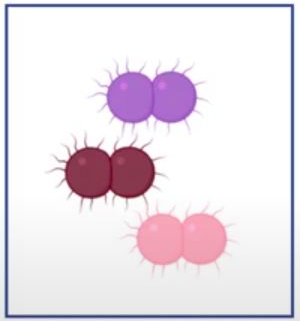


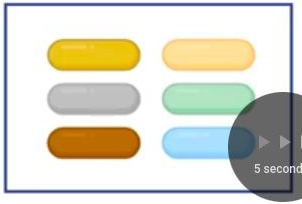


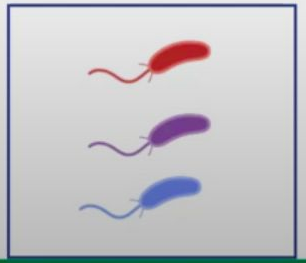




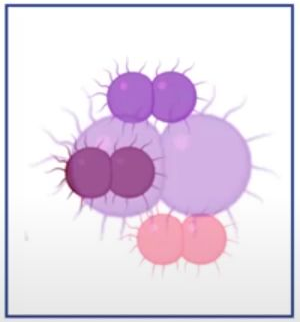
Clustering (Open Reference)

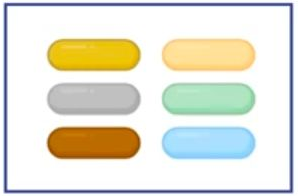


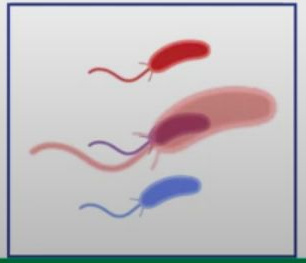




Clustering (Open Reference)

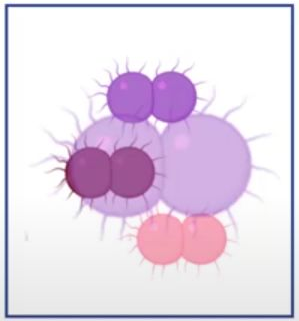


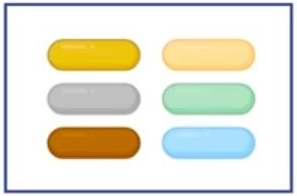


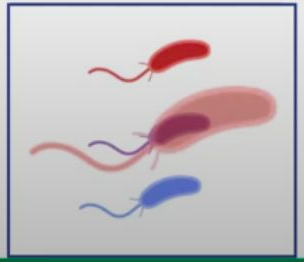


ZYMO RESEARCH

Clustering (Closed Reference)

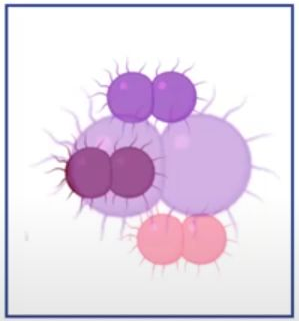


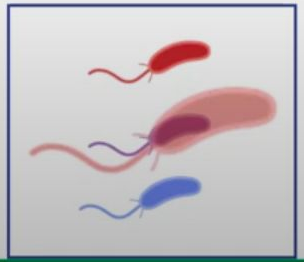




ZYMO RESEARCH

Clustering (Closed Reference)





ZYMO RESEARCH

Operational Taxonomic Unit (OTU) Approach

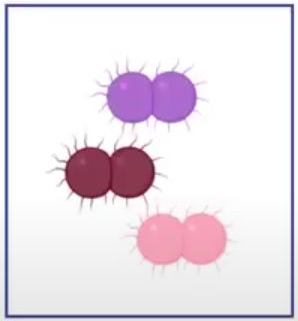
We know some of these sequences arose from error/artifact.

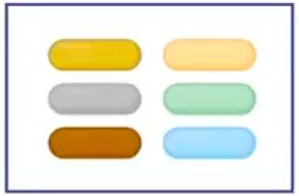
Combine extremely similar sequences (usually 97% or more identity) to minimize the effects of observed errors. Then treat each OTU as a representative sequence.

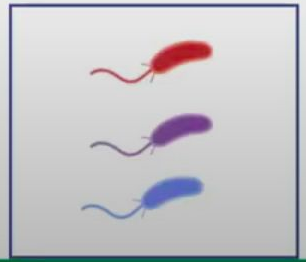
- OTUs can be represented by a representative consensus sequence
- Closed reference OTUs are fast to create, but are subject to reference bias
- *De novo* OTUs are free of reference bias, but computationally expensive and can change with changed samples
- Open reference OTUs are in between, with sequence similar to reference behaving like closed reference and more novel sequences behaving like *de novo*

ZYMO RESEARCH

But...

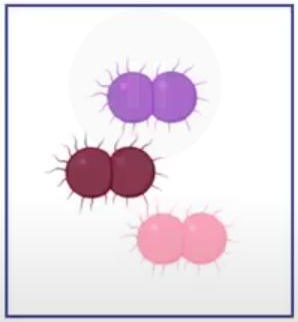




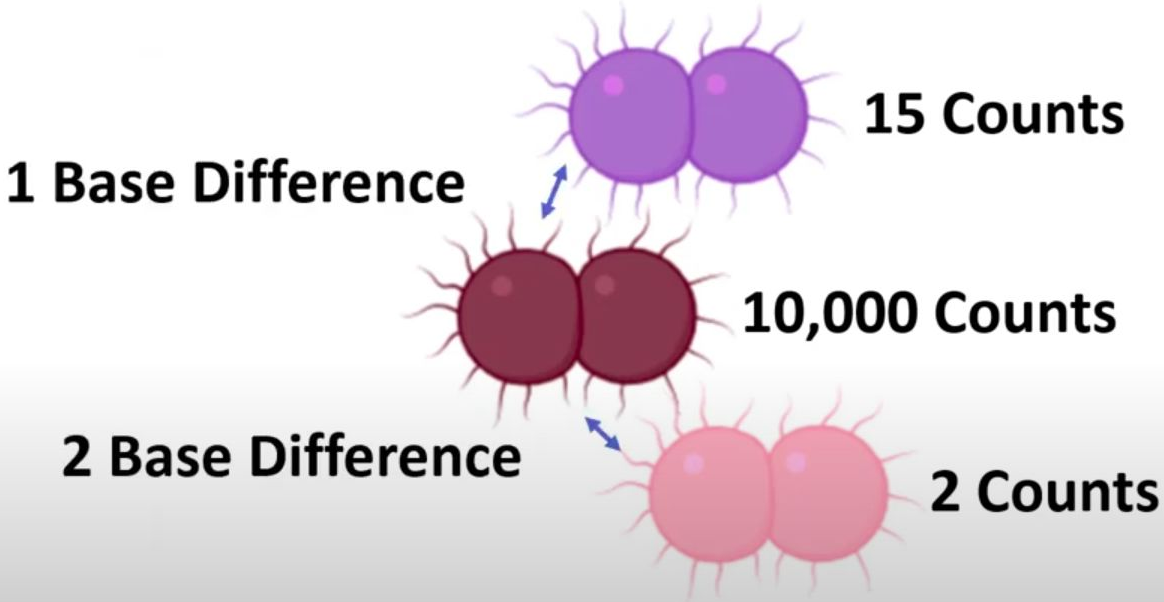


ZYMO RESEARCH

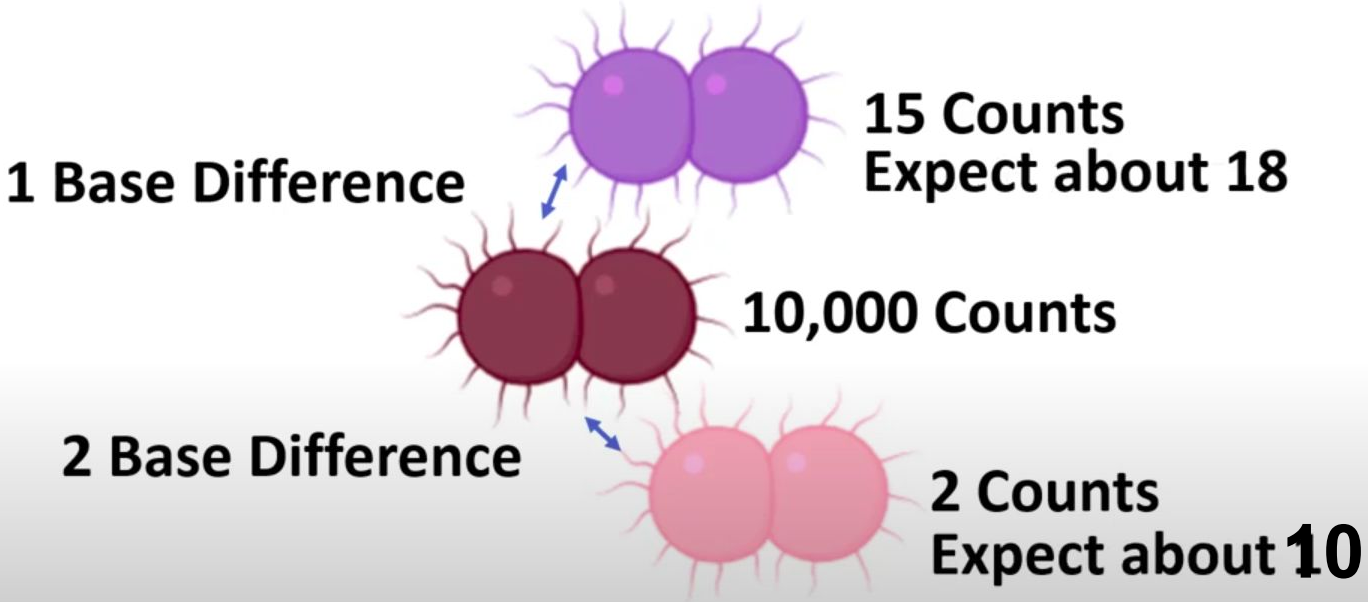
What if…



And Differences...

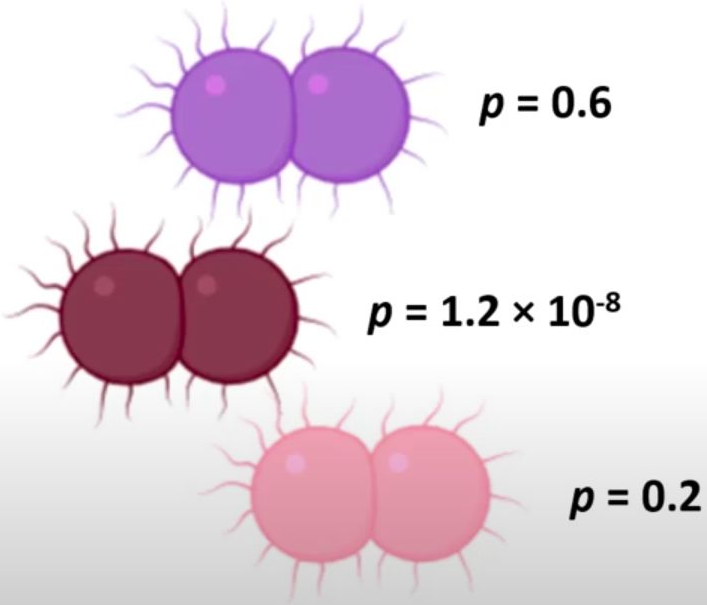


And an Error Model for the Run…



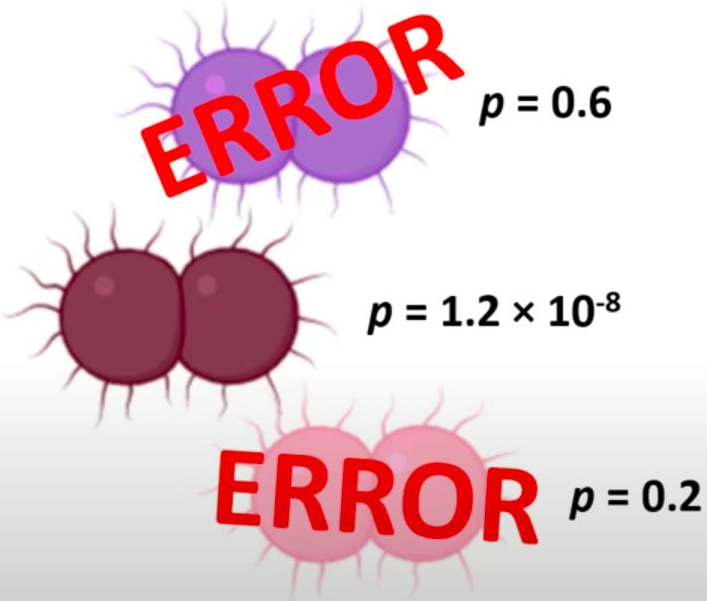
ZYMO RESEARCH

We Could Determine Our Confidence In a Sequence



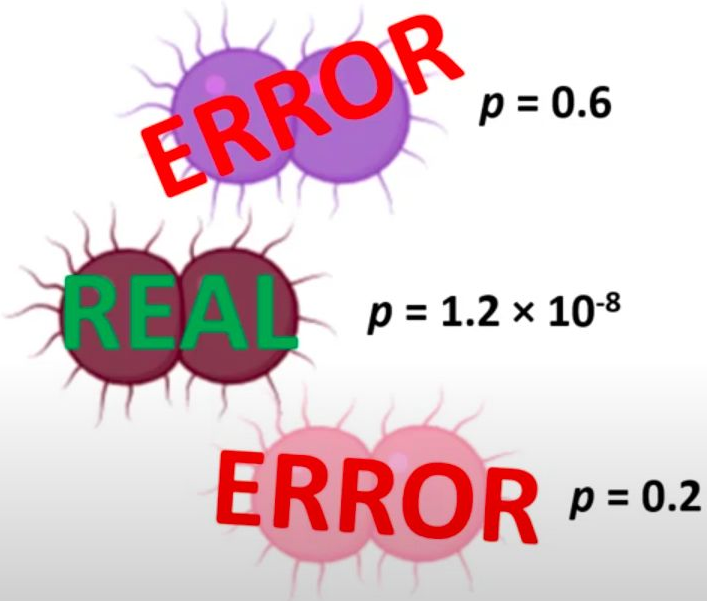
ZYMO RESEARCH

And Drop Sequences that Are Explained by Error



ZYMO RESEARCH

While Retaining Those In Which We Are Confident



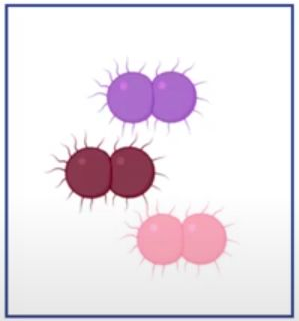
ZYMO RESEARCH

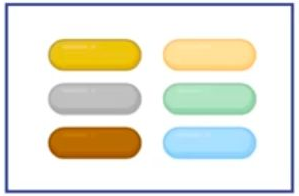
The Outcome Looks Like Our Clustering…

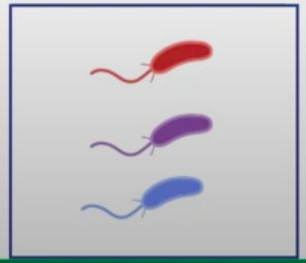
Operating

Taxonomic

Units







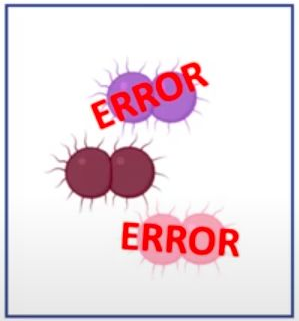
ZYMO RESEARCH

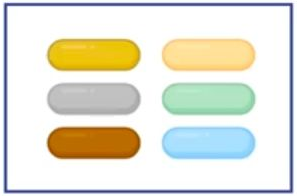
...What Changed?

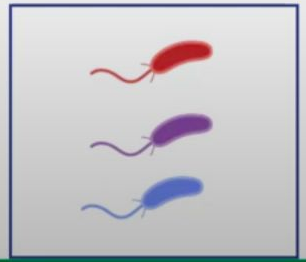
Operating

Taxonomic

Units







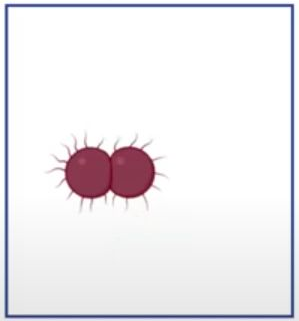
ZYMO RESEARCH

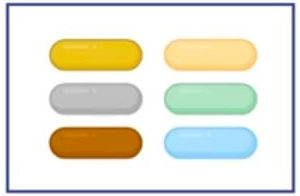
...What Changed?

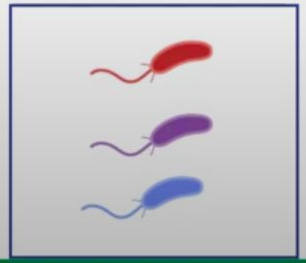
Operating

Taxonomic

Units







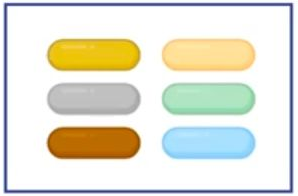
ZYMO RESEARCH

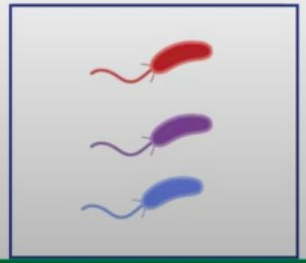
...What Changed?

Amplicon

Sequence

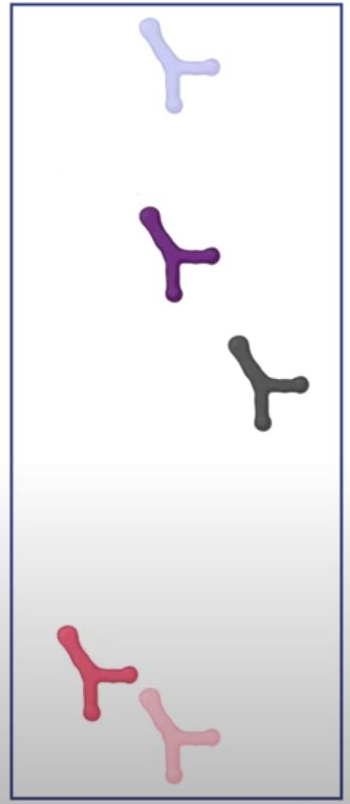
Variant





ZYMO RESEARCH

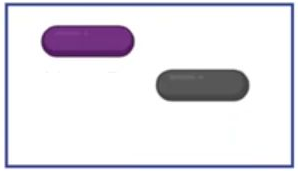
...What Changed?

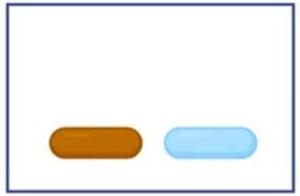


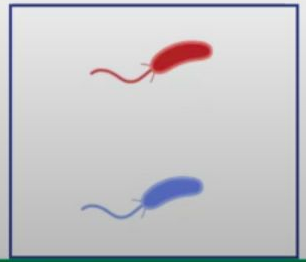
Amplicon

Sequence

Variant

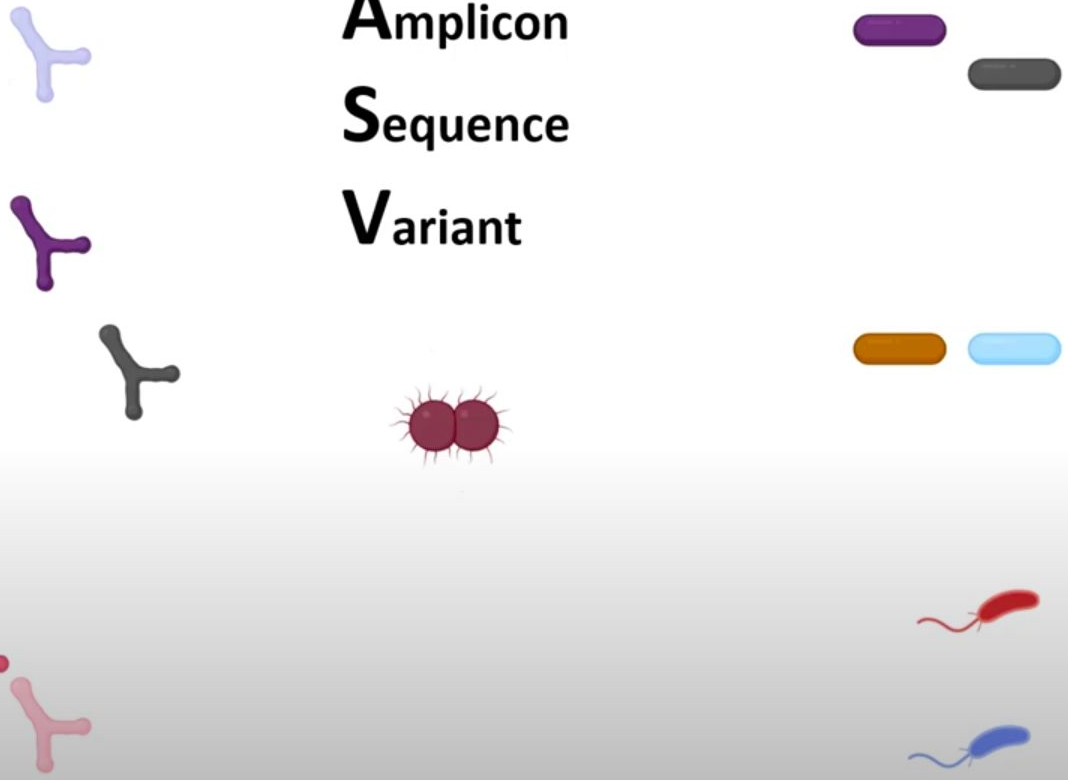




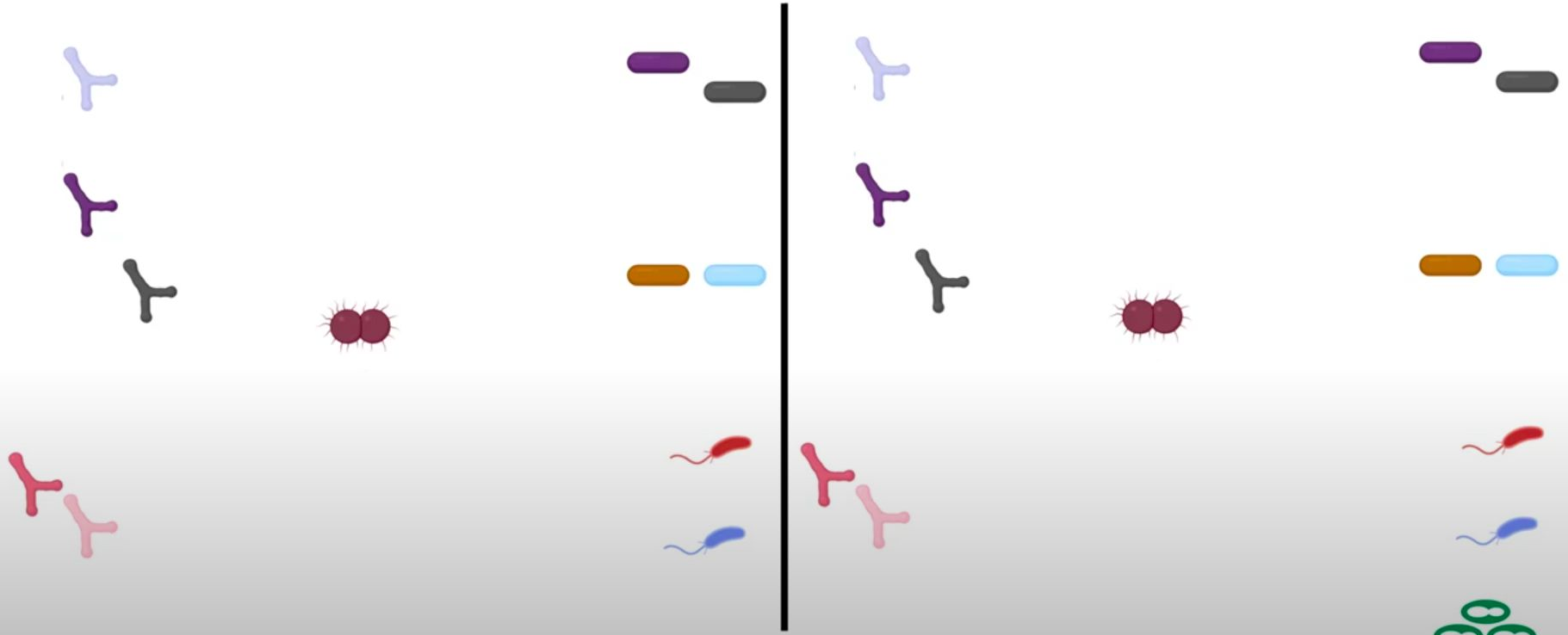


ZYMO RESEARCH

...What Changed?

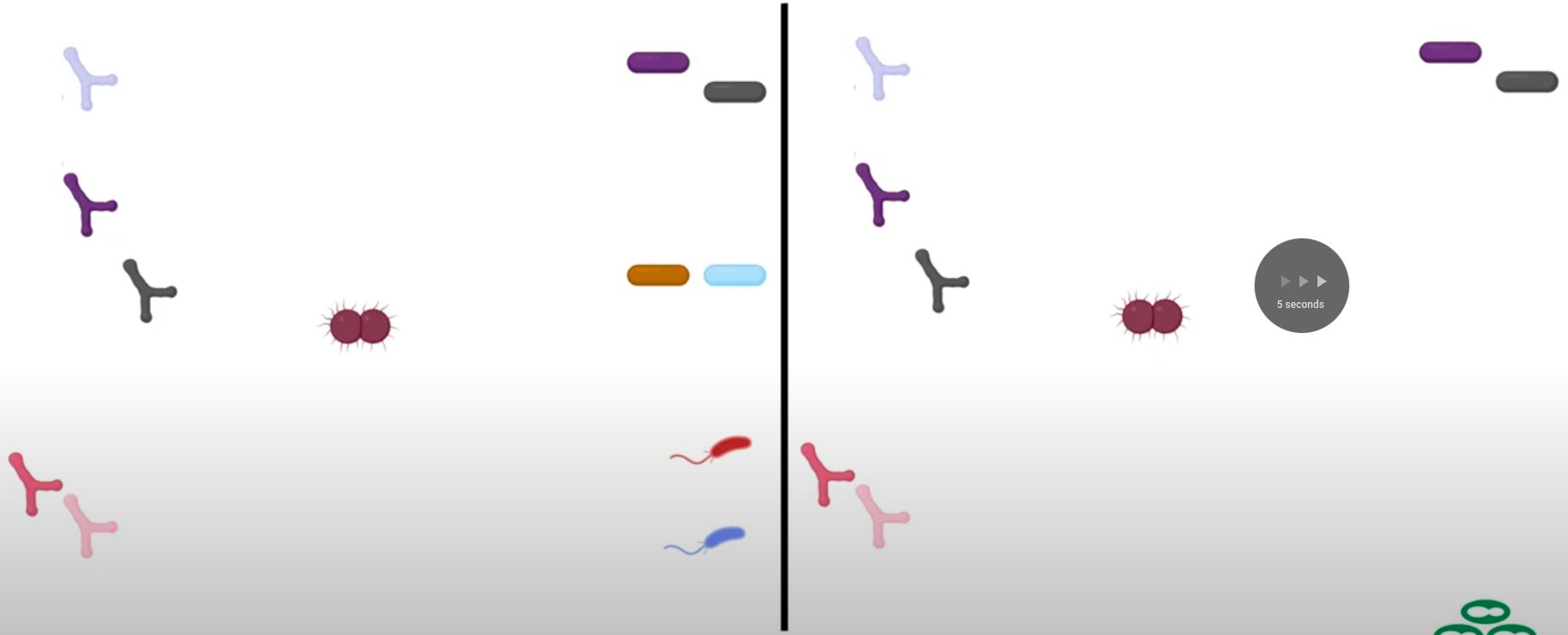


Adding More Samples



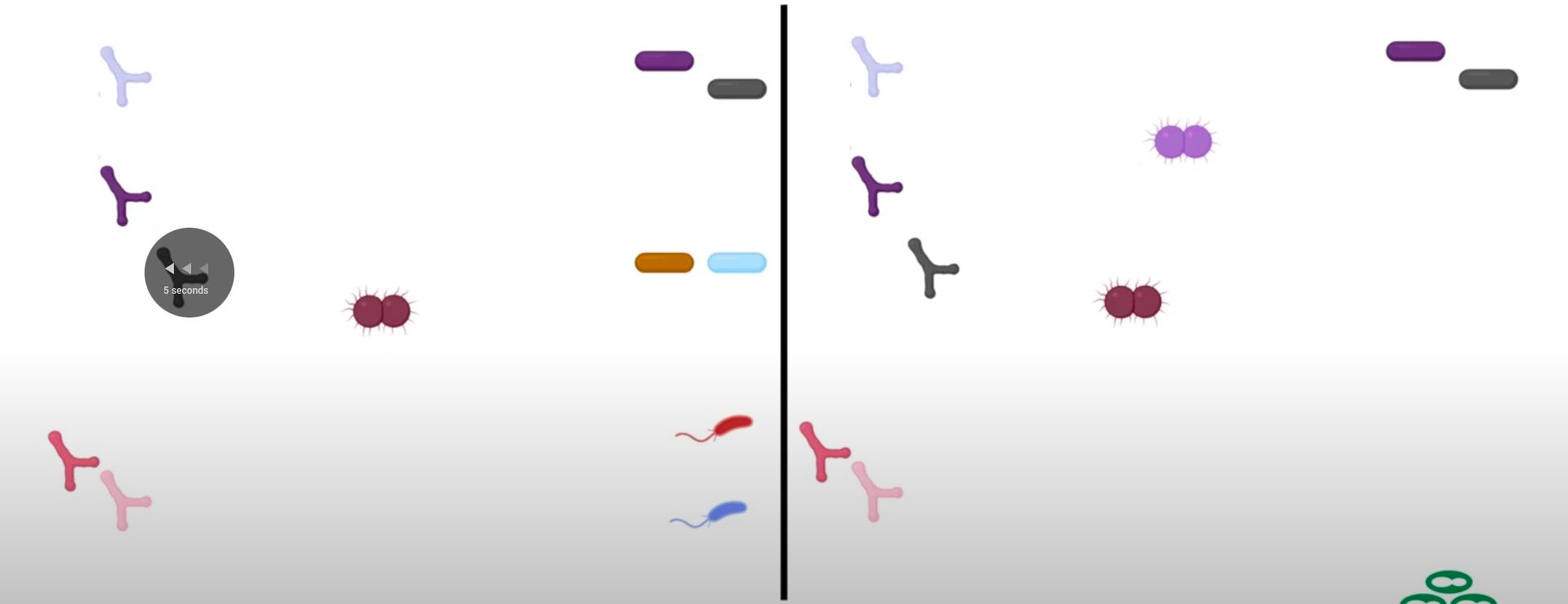
ZYMO RESEARCH

Adding More Samples

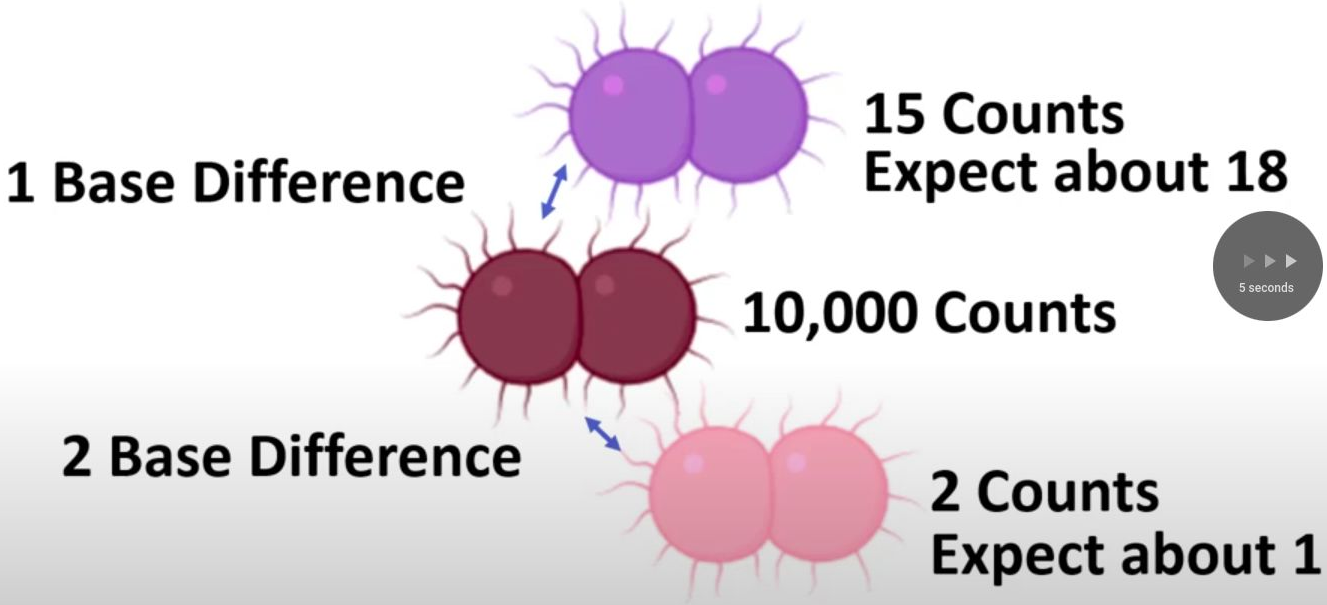


ZYMO RESEARCH

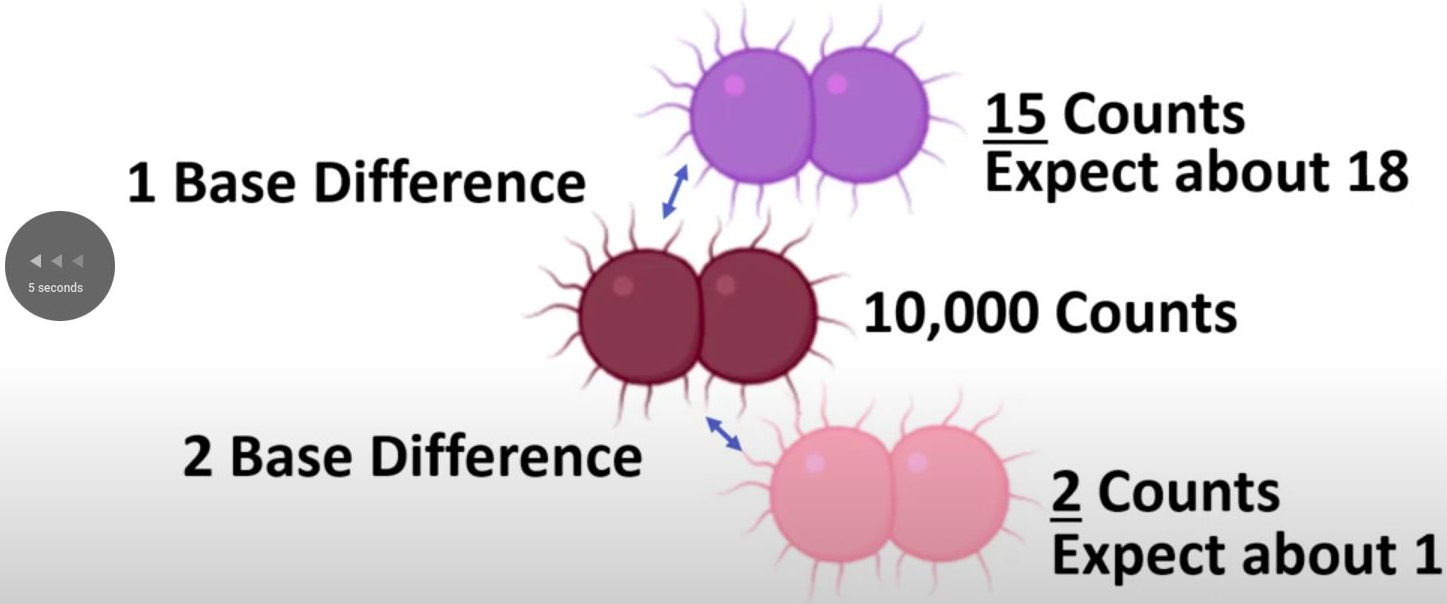
Adding More Samples



What Else Can Change?



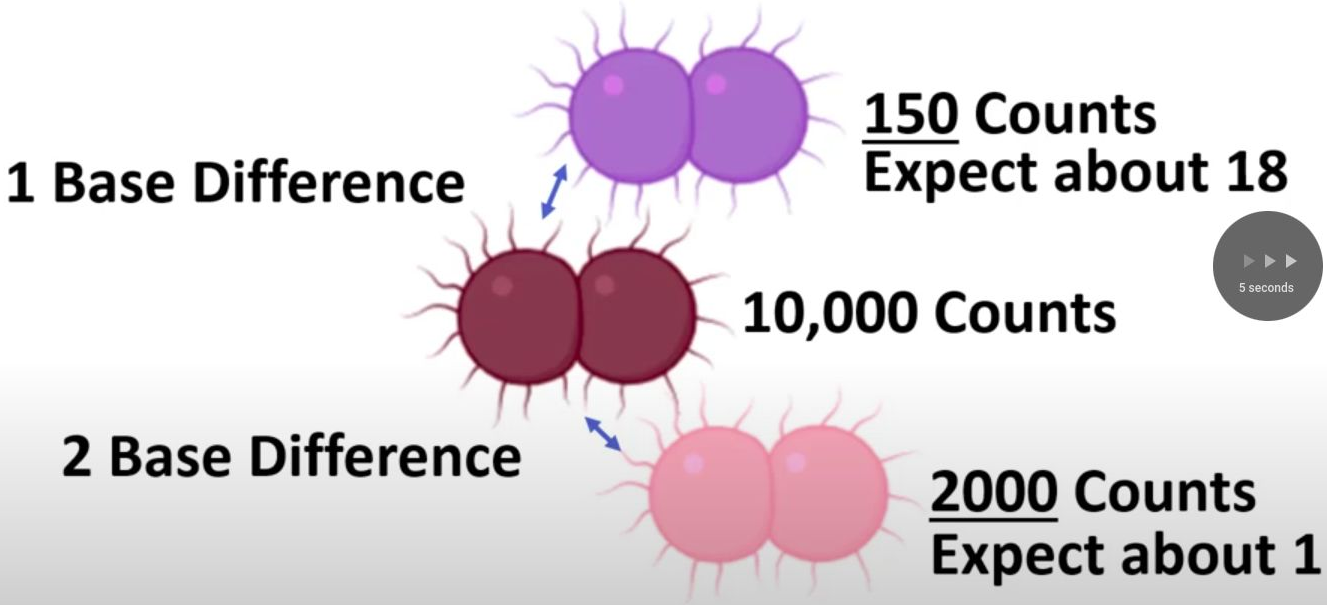
What Else Can Change?



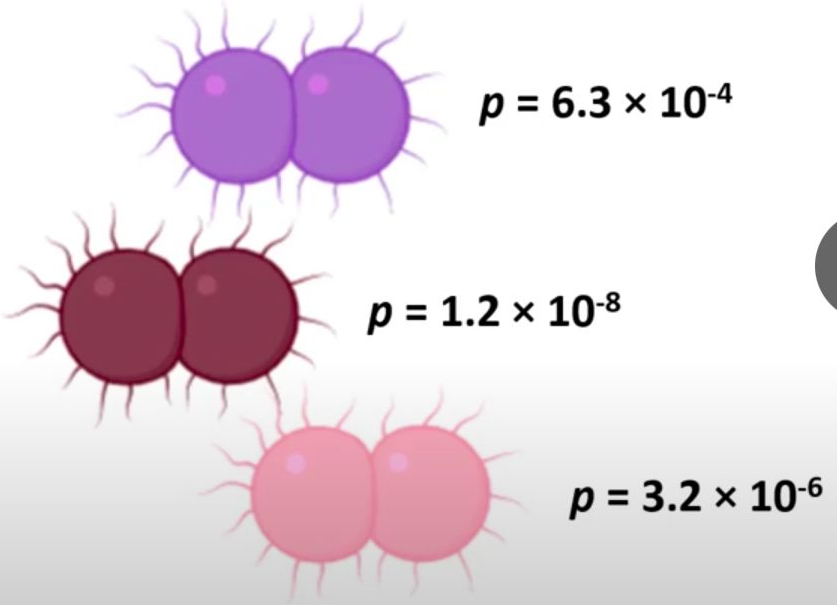
5 seconds

ZYMO RESEARCH

What If The Counts Were Far Higher?



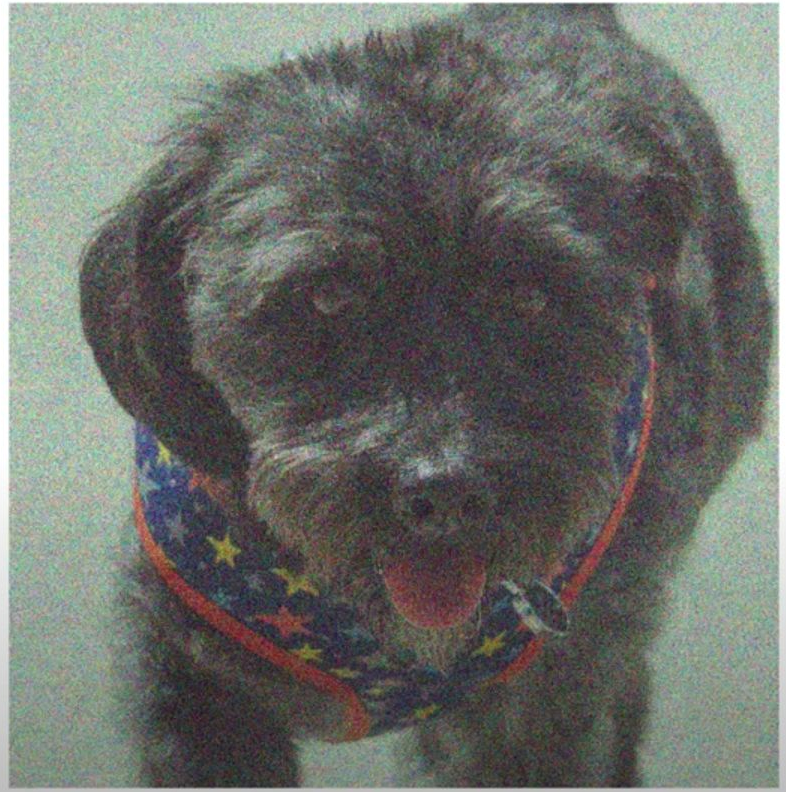
What If The Counts Were Far Higher?

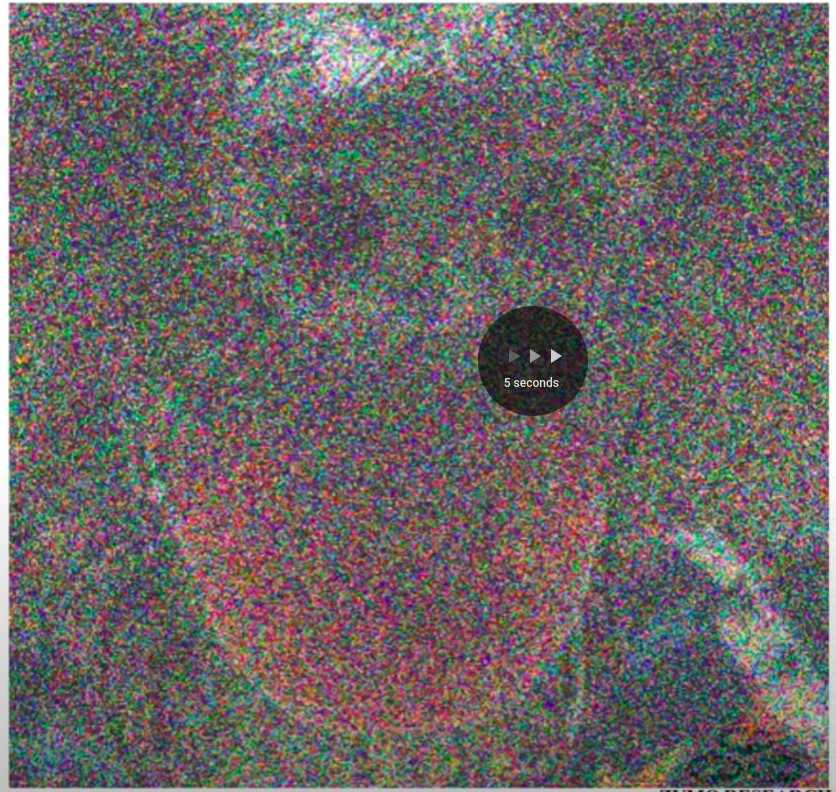


5 seconds

ZYMO RESEARCH

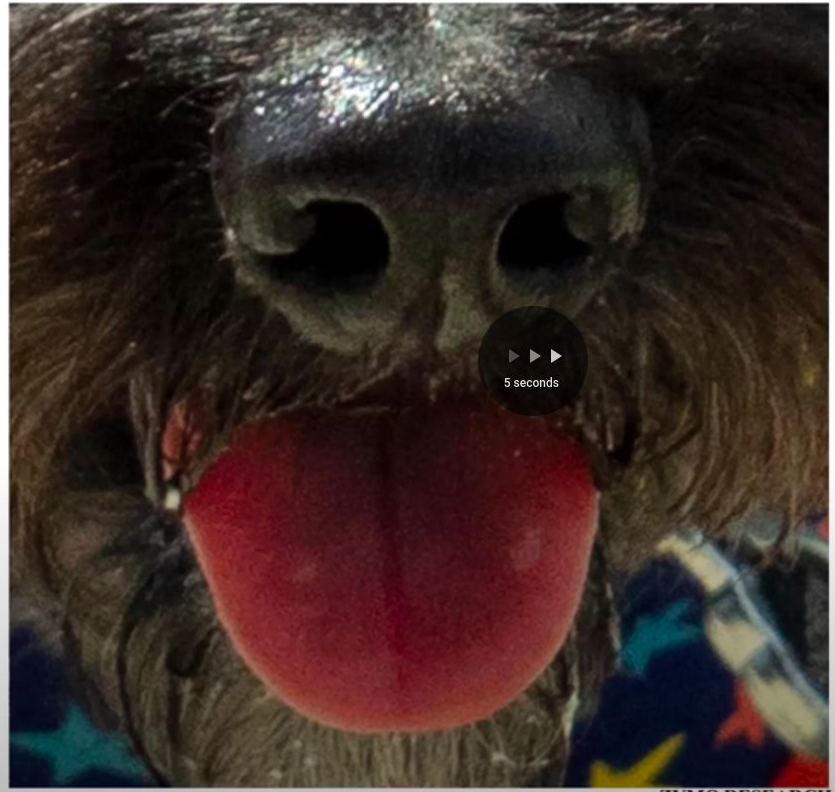
Can This Noise Be Predicted?





ZYMO RESEARCH

The True Image



Amplicon Sequence Variant (ASV) Approach

What is the statistical support for each sequence's existence?

Throw out amplicon sequences that lack strong statistical support for not being artifacts of sequencing. Cost: potential loss of real sequence that was present at very low levels.

- No representative consensus sequence
  - Each sequence is supported as being present in the sample
- Potentially higher resolution, no potential to combine multiple "real" sequences into an abstract
- Generated without the use of a reference, no risk of reference bias.
- May also be called an ESV (exact sequence variant) or zOTU (zero-radius OTU)

OTU *vs.* ASV

OTU	ASV
Can be subject to reference bias	Reference is not used until taxonomy assignment
OTU tables cannot be combined between studies	ASV tables can be compared across studies
Represented by a consensus sequence	Represented by an exact sequence
Can represent multiple species with different sequences	If it represents multiple species, it is because they share the sequence
Subject to chimeric sequences	Subject to chimeric sequences
Chimera detection can be complex and may require reference bias	Chimera detection is simple and reference-free

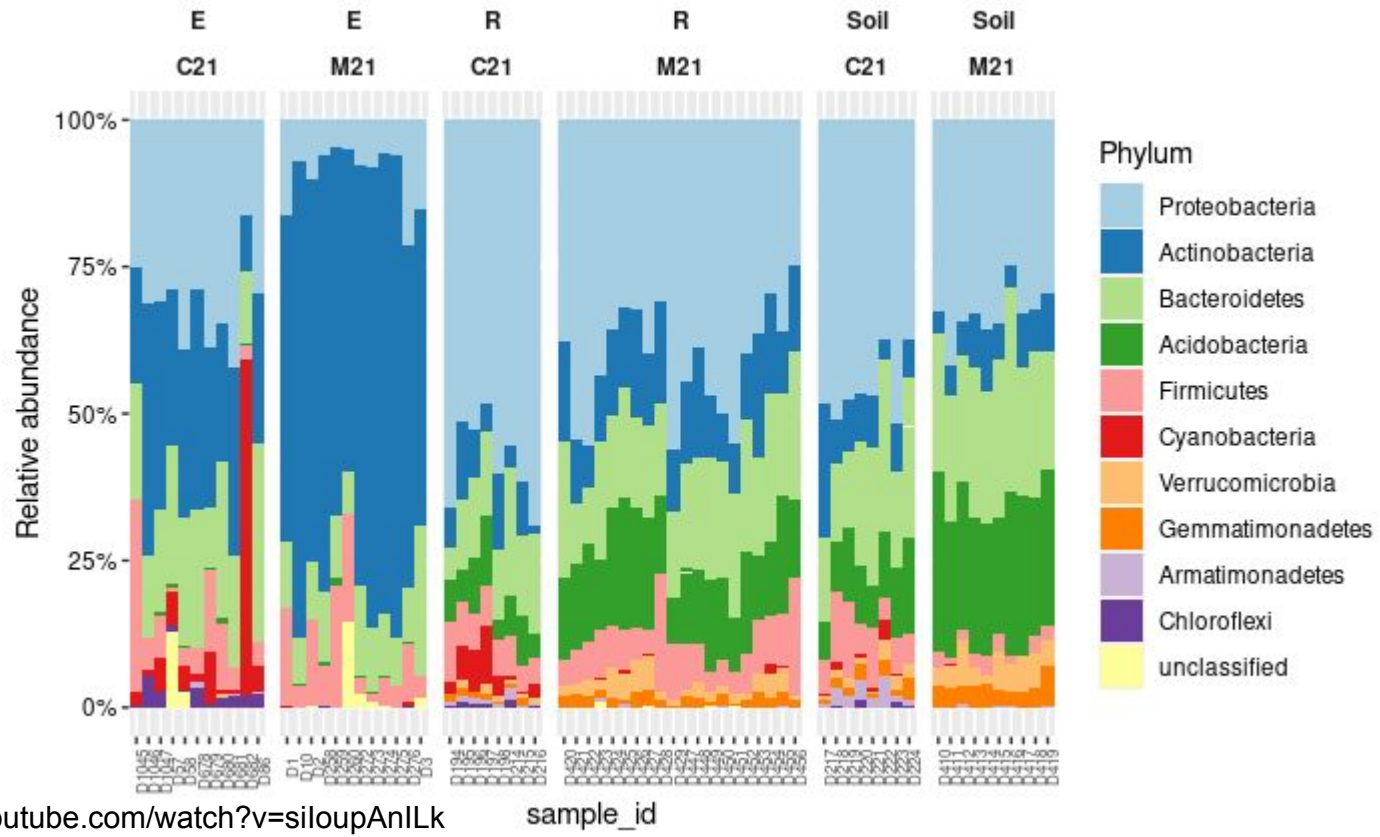
ZYMO RESEARCH

OTU *vs.* ASV

OTU	ASV
Can be subject to reference bias	Reference is not used until taxonomy assignment
OTU tables cannot be combined between studies	ASV tables can be compared across studies
Represented by a consensus sequence	Represented by an exact sequence
Can represent multiple species with different sequences	If it represents multiple species, it is because they share the sequence
Subject to chimeric sequences	Subject to chimeric sequences
Chimera detection can be complex and may require reference bias	Chimera detection is simple and reference-free
Constantly changed to “OUT” by autocorrect	Generally left alone

ZYMO RESEARCH

III. Microbiome composition abundance visualize tutorial



https://www.youtube.com/watch?v=siloupAnILk

1 - Otu table

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	otu	X10n	X10p	X11n	X11p	X120n	X120p	X121n	X121p	X122n	X122p	X125n	X125p	X126n	X126p	X127n	X13n	X13p	X140n	X140p	X141n
2	Otu001	13679	6292	42	2500	18850	5	43	7138	9432	10541	9	9772	1388	7	31538	38	2338	23	9	1358
3	Otu002	18	7134	38	9830	45	61420	182	23751	36	11	4535	3502	11018	5473	26	14411	38	19018	12	3080
4	Otu003	9939	8983	31	13	24620	19	19	16	12502	3831	4621	2240	9924	4052	9292	18	0	37	7	3680
5	Otu004	3675	4234	24	22	11	16	32967	35	6	18	6908	5	16	8702	24	11	37717	0	25	4196
6	Otu005	0	5	0	7	0	8	0	16	20166	0	0	2	5	8	2	16	0	13	0	0
7	Otu006	0	8	0	0	0	8	0	0	5	3	3	0	0	9	0	5	4	0	0	3
8	Otu007	4587	518	4	386	8775	5	6	1102	14336	0	0	3626	51	0	6	12	0	10	0	395
9	Otu008	1	8	2	4408	3	29	6	12355	0	0	0	0	0	9	3	1588	0	6	3	3
10	Otu009	115	914	3	325	0	629	1	834	5	0	1354	2108	1117	67	0	2010	1897	11227	1	3
11	Otu010	780	8	23810	12	3279	0	12	7	3027	0	2	4156	0	0	18	0	0	0	0	0
12	Otu011	0	3	2	2	0	13	5	5	4	7	3081	11	4	6804	0	3	11	0	5	0
13	Otu012	0	0	0	6	0	0	0	16	3	0	0	0	0	0	0	17	0	6	0	0
14	Otu013	6321	2471	2	0	12	3	0	0	4	20272	0	15	9	0	5	0	11	0	14	0
15	Otu014	0	82	4	3304	1	1667	4	9233	13	3	0	2707	0	0	3	4806	9	3	5	0
16	Otu015	0	12	0	3	7	25	1	6	10	0	4	2772	1	3	0	2	0	10	13	8052
17	Otu016	1	0	0	9	5	0	0	14	0	0	0	0	2654	0	0	6	1	1	0	0
18	Otu017	0	0	0	0	0	0	0	0	17	8	0	0	0	0	0	17	24	48	35210	4
19	Otu018	1	0	9	911	0	0	15	2702	6	4	342	2217	606	0	13	3846	4	6	8513	1
20	Otu019	0	0	13	0	0	0	29	0	0	0	0	0	0	0	11	0	0	5	4	0
21	Otu020	425	0	1	0	1706	0	8447	1	0	0	0	0	0	26	0	0	3490	0	2620	0
22	Otu021	0	4	0	0	0	10	0	0	0	0	2	0	0	4	0	0	0	0	0	4
23	Otu022	0	0	0	4987	0	0	0	6	90	1	1	524	0	467	0	4	8	6198	0	1
24	Otu023	4	0	1	0	0	3	0	0	0	0	0	0	3351	3	0	3910	1	2	3	0
25	Otu024	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	2	1	0
26	Otu025	69	0	0	0	290	0	0	0	21	0	118	2	9	513	2	0	0	2	0	0
27	Otu026	0	2	0	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0
28	Otu027	6	2304	0	0	5	0	0	0	57	4	0	14529	9597	2	6	0	0	0	0	0

2 - Taxonomy Table

	A	B	C	D	E	F	G	H	I
1	otu	Domain	Supergroup	Division	Class	Order	Family	Genus	
2	Otu001	Eukaryota	Archaeplastida	Chlorophyta	Mamiellophyceae	Mamiellales	Bathycoccaceae	Ostreococcus	
3	Otu002	Eukaryota	Hacrobia	Haptophyta	Prymnesiophyceae	Prymnesiophyceae_X	Braarudosphaeraceae	UCYN_A1_host	
4	Otu003	Eukaryota	Archaeplastida	Chlorophyta	Mamiellophyceae	Mamiellales	Bathycoccaceae	Bathycoccus	
5	Otu004	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Dinophyceae_X	Dinophyceae_X	Prorocentrum	
6	Otu005	Eukaryota	Stramenopiles	Ochrophyta	Bacillariophyta	Mediophyceae	Mediophyceae_X	Thalassiosira	
7	Otu006	Eukaryota	Stramenopiles	Ochrophyta	Bacillariophyta	Bacillariophyceae	Bacillariophyceae_X	Pseudo_nitzschia	
8	Otu007	Eukaryota	Stramenopiles	Ochrophyta	Pelagophyceae	Pelagophyceae_X	Pelagophyceae_X	Pelagomonas	
9	Otu008	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Dinophyceae_X	Dinophyceae_X	Dinophyceae_X	
10	Otu009	Eukaryota	Hacrobia	Haptophyta	Prymnesiophyceae	Prymnesiales	Chrysochromulinaceae	Chrysochromulina	
11	Otu010	Eukaryota	Opisthokonta	Metazoa	Craniata	Craniata_X	Craniata_XX	Craniata_XX_unclassified	
12	Otu011	Eukaryota	Stramenopiles	Ochrophyta	Chrysophyceae	Chrysophyceae_X	Chrysophyceae_Clade_C	Chrysophyceae_Clade_C_X	
13	Otu012	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Dinophyceae_X	Dinophyceae_X	Gonyaulax	
14	Otu013	Eukaryota	Alveolata	Dinophyta	Syndiniales	Syndiniales_Group_III	Syndiniales_Group_III_X	Syndiniales_Group_III_X	
15	Otu014	Eukaryota	Stramenopiles	Ochrophyta	Chrysophyceae	Chrysophyceae_X	Chrysophyceae_Clade_G	Chrysophyceae_Clade_G_X	
16	Otu015	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Dinophyceae_X	Dinophyceae_X	Dinophyceae_X	
17	Otu016	Eukaryota	Hacrobia	Centroheliozoa	Centroheliozoa_X	Pterocystida	Pterocystida_X	Pterocystida_X	
18	Otu017	Eukaryota	Opisthokonta	Fungi	Basidiomycota	Agaricomycotina	Agaricomycetes	Hyphodontia	
19	Otu018	Eukaryota	Stramenopiles	Ochrophyta	Dictyochophyceae	Dictyochophyceae_X	Pedinellales	Pedinellales_X	
20	Otu019	Eukaryota	Opisthokonta	Fungi	Basidiomycota	Agaricomycotina	Agaricomycetes	Itersonilia	
21	Otu020	Eukaryota	Hacrobia	Haptophyta	Prymnesiophyceae	Prymnesiophyceae_X	Braarudosphaeraceae	Braarudosphaera	
22	Otu021	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Dinophyceae_X	Dinophyceae_X	Dinophyceae_X	
23	Otu022	Eukaryota	Hacrobia	Haptophyta	Prymnesiophyceae	Prymnesiophyceae_X	Prymnesiophyceae_X	Syracosphaera	
24	Otu023	Eukaryota	Stramenopiles	Ochrophyta	Bacillariophyta	Bacillariophyceae	Bacillariophyceae_X	Bacillariophyceae_X	
25	Otu024	Eukaryota	Archaeplastida	Streptophyta	Klebsormidiophyceae	Klebsormidiophyceae_X	Klebsormidiophyceae_XX	Klebsormidium	
26	Otu025	Eukaryota	Archaeplastida	Chlorophyta	Mamiellophyceae	Mamiellales	Mamiellaceae	Micromonas	
27	Otu026	Eukaryota	Stramenopiles	Ochrophyta	Bacillariophyta	Bacillariophyceae	Bacillariophyceae_X	Cylindrotheca	
28	Otu027	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Suessiales	Suessiales_X	Karlodinium	
29	Otu028	Eukaryota	Hacrobia	Haptophyta	Prymnesiophyceae	Isochrysidales	Noelaerhabdaceae	Emiliania	
30	Otu029	Eukaryota	Opisthokonta	Fungi	Ascomycota	Saccharomycotina	Saccharomycetales	Debaryomyces	
31	Otu030	Eukaryota	Hacrobia	Cryptophyta	Cryptophyceae	Cryptophyceae_X	Cryptomonadales	Teleaulax	
32	Otu031	Eukaryota	Alveolata	Dinophyta	Syndiniales	Syndiniales_Group_I	Syndiniales_Group_I_Clade_1	Syndiniales_Group_I_Clade_1_X	
33	Otu032	Eukaryota	Archaeplastida	Chlorophyta	Prasino_Clade_VII	Prasino_Clade_VII_X	Prasino_Clade_VII_A	Prasino_Clade_VII_A_4_X	

3 - Sample table (metadata)

	A	B	C	D	E	F	G	H	I	J	K	L
1	sample	fraction	Select_18S_nifH	total_18S	total_16S	total_nifH	sample_number	transect	station	depth	latitude	longitude
2	X10n	Nano	Yes	53230	8772	36	10	1	81	140	-27.42	-44.72
3	X10p	Pico	Yes	47390	4448	6241	10	1	81	140	-27.42	-44.72
4	X11n	Nano	No	24007	6193	3772	11	1	85	110	-26.8	-45.3
5	X11p	Pico	Yes	31899	14	10201	11	1	85	110	-26.8	-45.3
6	X120n	Nano	Yes	70455	5292	93	120	2	96	5	-27.39	-47.82
7	X120p	Pico	Yes	76182	53272	23147	120	2	96	5	-27.39	-47.82
8	X121n	Nano	Yes	52401	5958	26838	121	2	96	30	-27.39	-47.82
9	X121p	Pico	Yes	71785	10993	23706	121	2	96	30	-27.39	-47.82
10	X122n	Nano	Yes	78740	11730	15543	122	2	96	50	-27.39	-47.82
11	X122p	Pico	Yes	37364	11817	11045	122	2	96	50	-27.39	-47.82
12	X125n	Nano	Yes	27381	9	14331	125	2	98	5	-27.59	-47.39
13	X125p	Pico	Yes	55179	10419	21461	125	2	98	5	-27.59	-47.39
14	X126n	Nano	Yes	65714	15	16929	126	2	98	50	-27.59	-47.39
15	X126p	Pico	Yes	30406	3	10140	126	2	98	50	-27.59	-47.39
16	X127n	Nano	Yes	60610	9	11493	127	2	98	85	-27.59	-47.39
17	X13n	Nano	Yes	46001	33	21316	13	1	86	105	-26.33	-45.41
18	X13p	Pico	Yes	59626	7217	11954	13	1	86	105	-26.33	-45.41
19	X140n	Nano	Yes	48126	10428	25286	140	2	101	5	-27.79	-46.96
20	X140p	Pico	Yes	46569	10448	12301	140	2	101	5	-27.79	-46.96
21	X141n	Nano	Yes	30081	6394	21302	141	2	101	60	-27.79	-46.96
22	X141p	Pico	Yes	64221	11318	10428	141	2	101	60	-27.79	-46.96
23	X142n	Nano	Yes	85219	23243	11753	142	2	101	110	-27.79	-46.96
24	X142p	Pico	Yes	89797	9553	17156	142	2	101	110	-27.79	-46.96
25	X155n	Nano	Yes	54162	8237	20674	155	2	106	5	-28.12	-46.17
26	X155p	Pico	Yes	50782	7384	66172	155	2	106	5	-28.12	-46.17
27	X156n	Nano	Yes	55065	11371	14447	156	2	106	60	-28.12	-46.17
28	X156p	Pico	Yes	43917	9665	16093	156	2	106	60	-28.12	-46.17
29	X157n	Nano	Yes	29078	4978	15532	157	2	106	100	-28.12	-46.17
30	X157p	Pico	Yes	51848	9139	15204	157	2	106	100	-28.12	-46.17
31	X15n	Nano	Yes	22468	2887	2678	15	1	87	105	-26.22	-45.48
32	X15p	Pico	Yes	78390	13813	1033	15	1	87	105	-26.22	-45.48
33	X165n	Nano	Yes	50732	15337	14706	165	2	114	5	-28.65	-44.99
34	X165p	Pico	Yes	48514	10902	39918	165	2	114	5	-28.65	-44.99

Work in R_notebook

1.-Import data

2.-Analyse-manipulate data

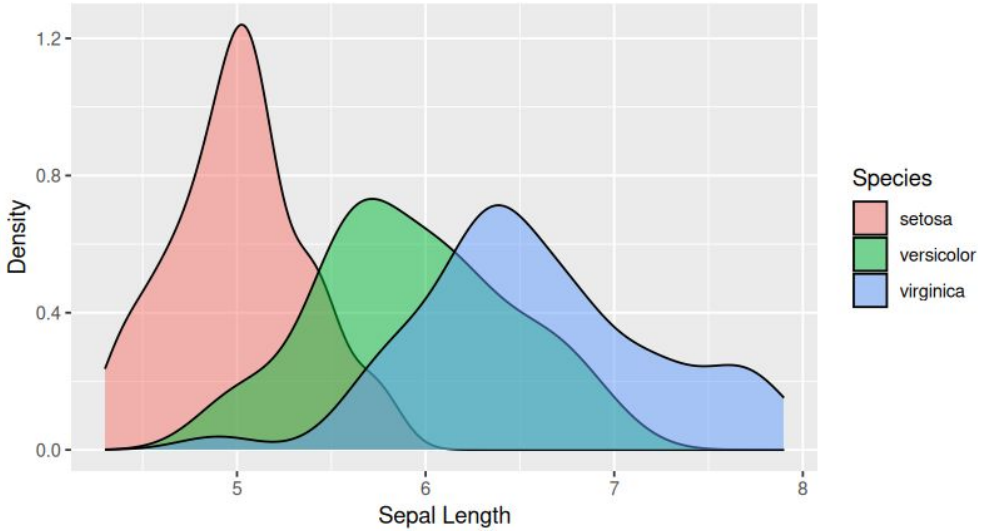
3-Draw basic_plot

4-Extend more detail

Summary

1. Visualise data is IMPORTANT!-quick note about install R packages in Linux
2. Reshape2
3. Learn by yourself
4. OTU and ASV
5. Microbiome barplot tutorial
6. Homework

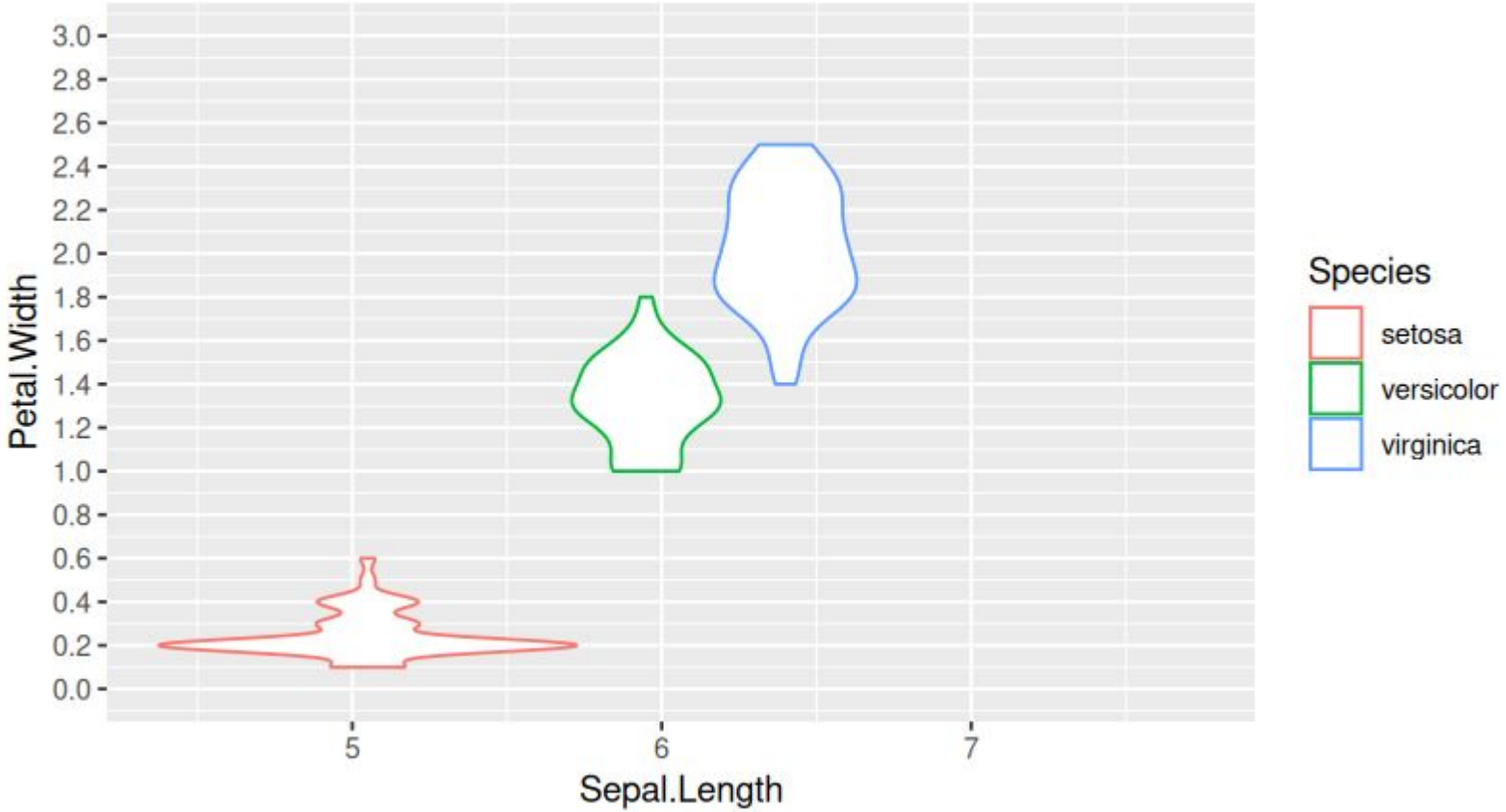
Homework

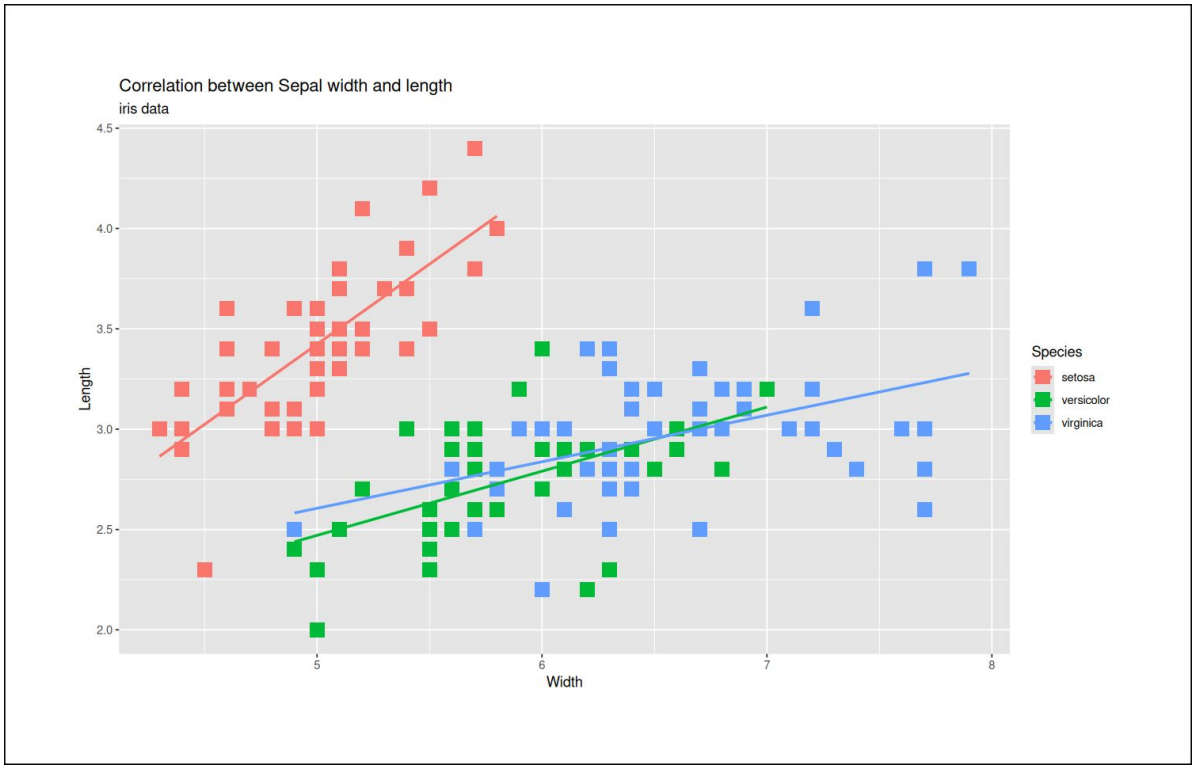



```
# Generate the sample data frame

data("iris")

print(data)
```


Petal.Width
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
5
6
7
Sepal.Length
Species
setosa
versicolor
virginica


Correlation between Sepal width and length
iris data
Length
Width
4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8
Species
setosa
versicolor
virginica