Refactoring Riboviz Analysis Code:

A Personal Journey

22 July 2020

Flic Anderson
The Wallace Lab, University of Edinburgh

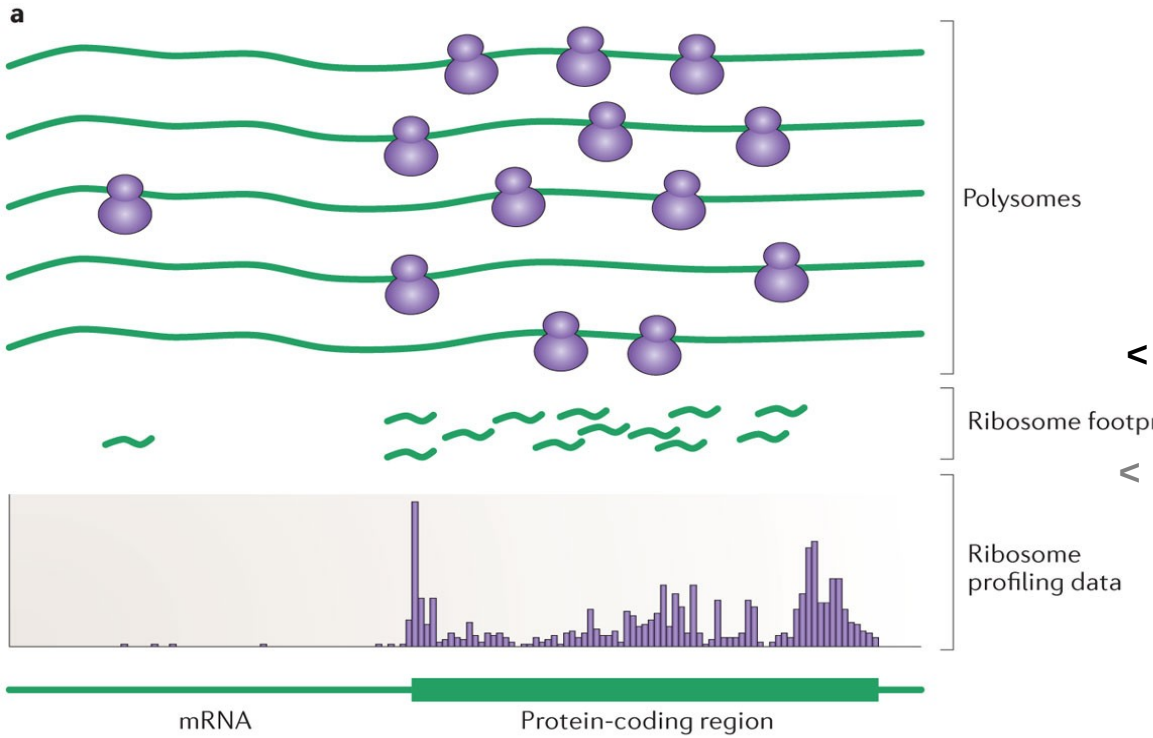
Outline

- Riboviz Aims
- Riboviz Workflow
- Analyis Code Refactoring
  - Example
  - Tips
  - Fails
- My Future Priorities

Riboviz 101

- **Riboviz processes & analyses ribosome profiling data**
- Ribosome profiling data helps unlock details of **active translation**: *mechanics of translation, regulation methods, translational efficiency*
- Developing/improving riboviz = **more researcher time** for biological questions rather than tinkering with pipelines & bespoke analysis code...

Polysomes to Footprints

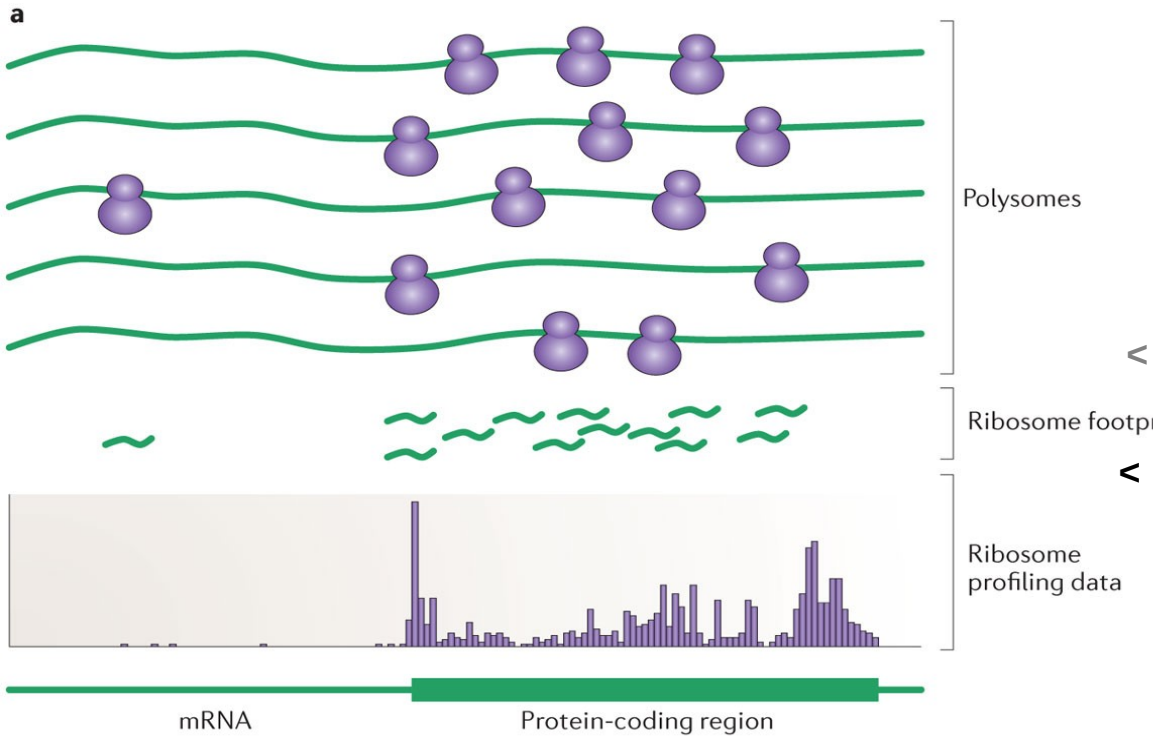


< Experimental Stages

< Processing & Analysis

Ingolia (2014). "Ribosome profiling: new views of translation, from single codons to genome scale". Nature Reviews. Genetics. 15 (3): 205–13. doi:10.1038/nrg3645

Footprints to Profiling Data



< Experimental Stages

< Processing & Analysis

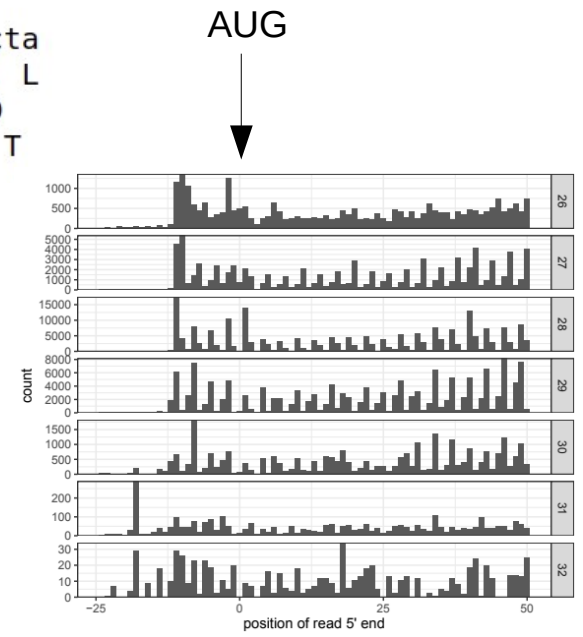
Ingolia (2014). "Ribosome profiling: new views of translation, from single codons to genome scale". Nature Reviews. Genetics. 15 (3): 205–13. doi:10.1038/nrg3645

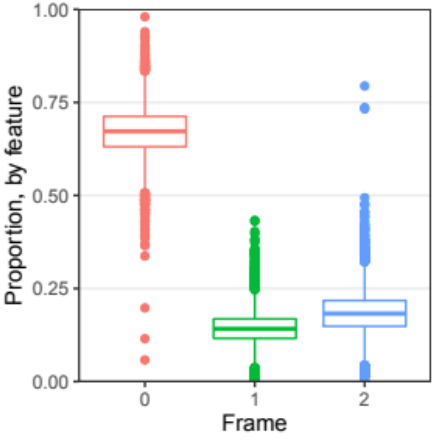
Analysis of Ribosome Profiling Data

- Looking for **3-NT periodicity**: ribosomes moving along transcript 1 codon at a time
- Most reads map to **coding regions** (98.8% in Ingolia et al 2009)
- Reasonable read-lengths (e.g. know should look for appx **~28-30NT**)
- Looking for most reads to be in **one frame**: this is a key **Quality Control** measure

Ribosome Data & Reading Frame

```
reading frame:          123
                        |||
                        acttacccgggacta
first reading frame       T  Y  P  G  L
second reading frame       L  T  R  D
third reading frame         L  P  G  T
```





G-Sc_2014: WTnone

Reading Frame: https://www.ncbi.nlm.nih.gov/Class/MLACourse/Original8Hour/Genetics/readingframe.html

Data: Guydosh & Green (2014). “Dom34 rescues ribosomes in 3' untranslated regions.” Cell. 156 (5): 950-62. doi: 10.1016/j.cell.2014.02.006.

Riboviz Workflow: Inputs

Organism Specific

Transcript Sequences
.fasta

Genome / Transcriptome Features
.gff

Contaminant Sequences (rRNA)
.fasta

(Additional Organism-Specific Data)

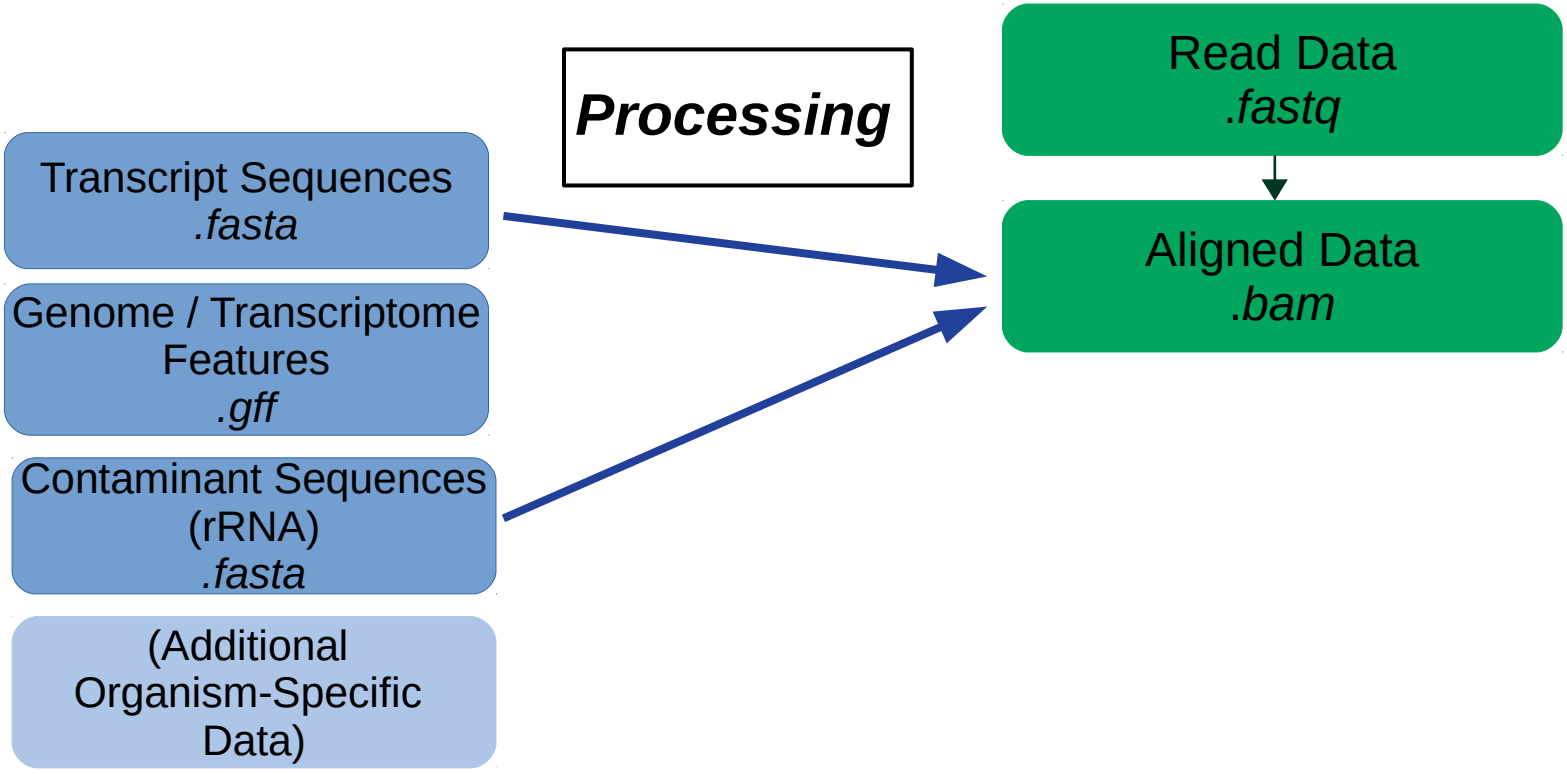
Sample Specific

Read Data
.fastq

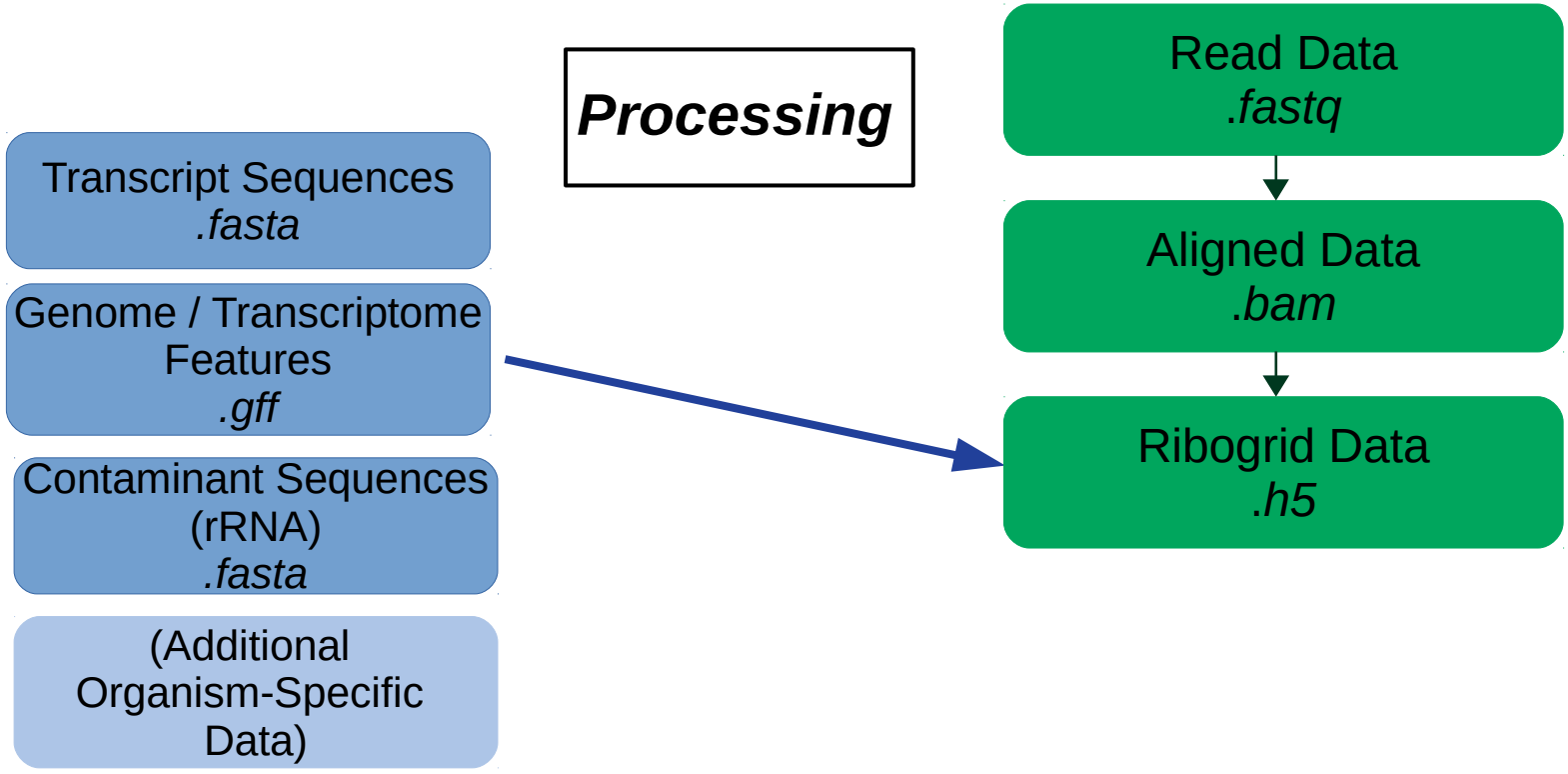
Configuration File
.yaml

Configuration File lists all files & parameters needed to run RiboViz

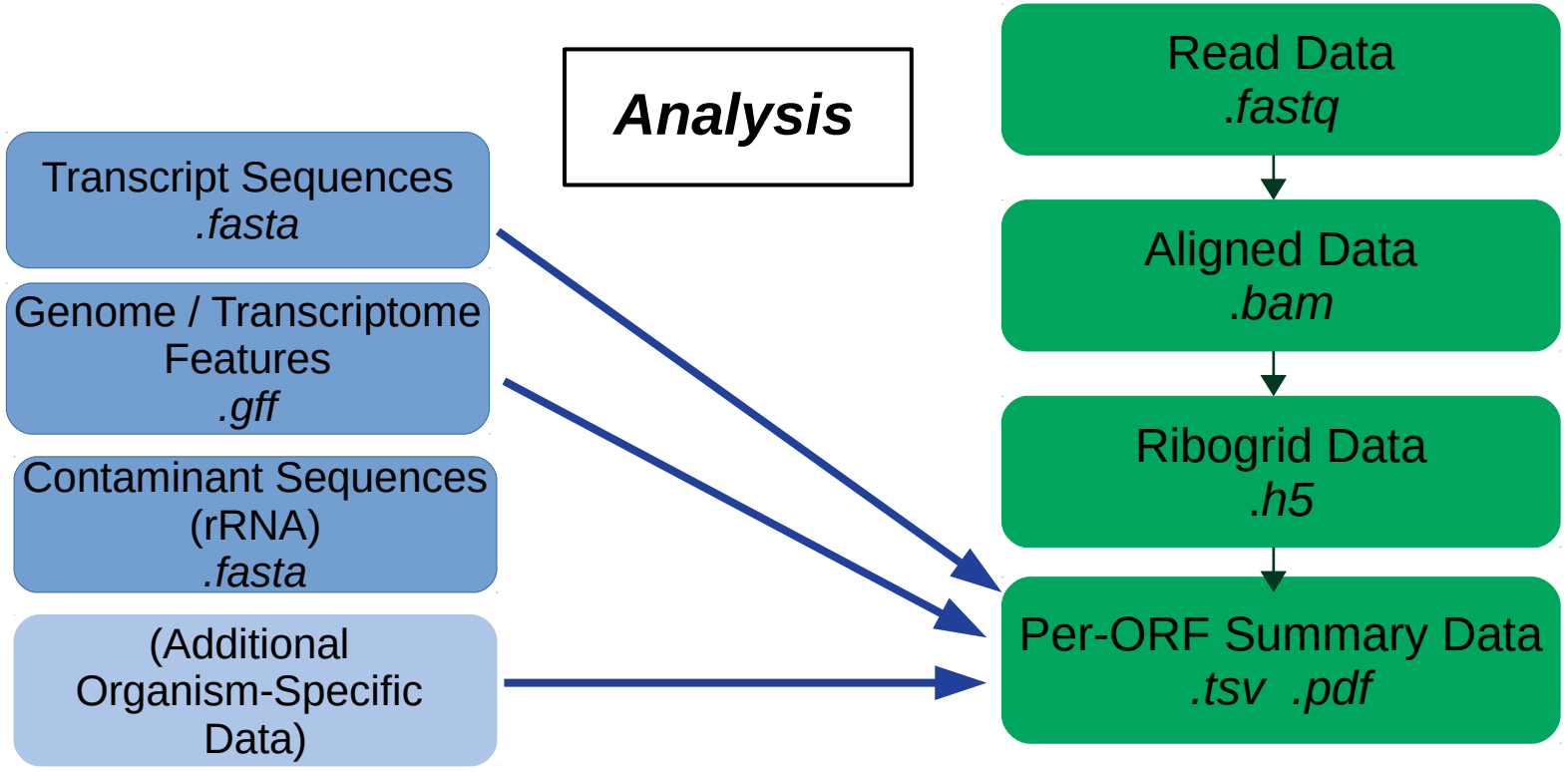
Riboviz Workflow



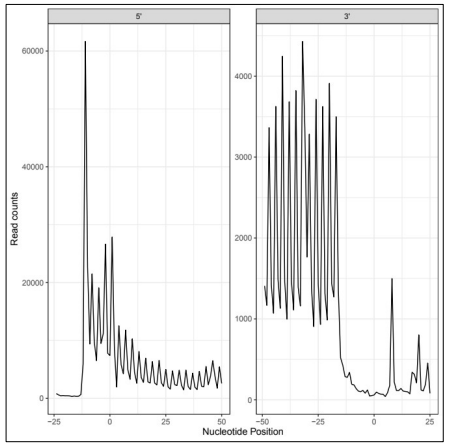
Riboviz Workflow

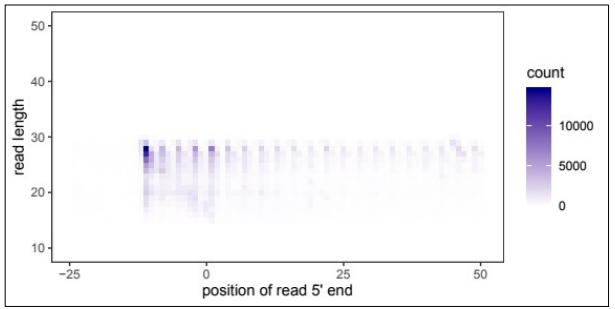


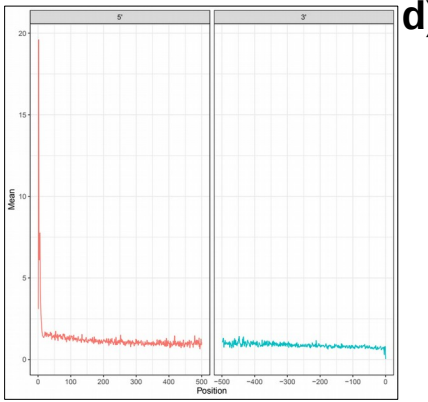
Riboviz Workflow

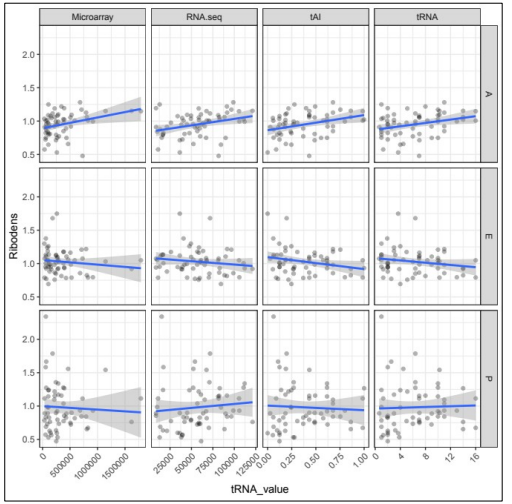


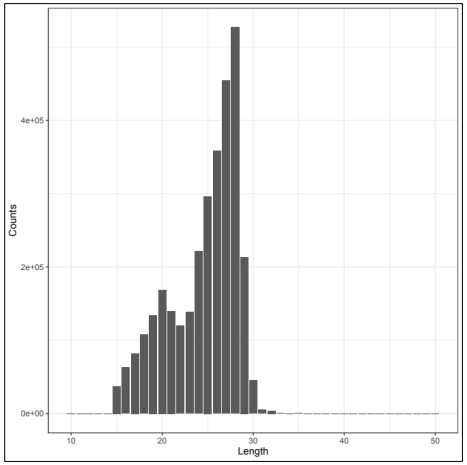
B-Sc_2012: ANAPH_1

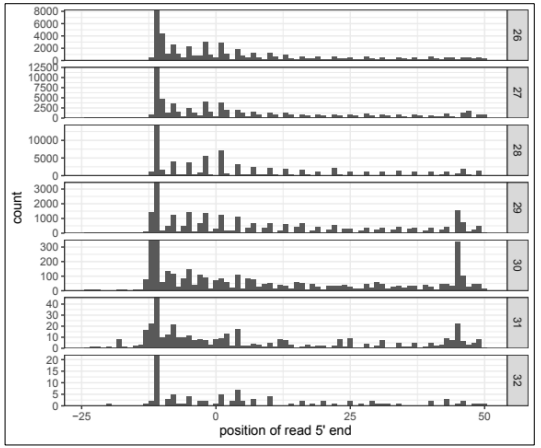


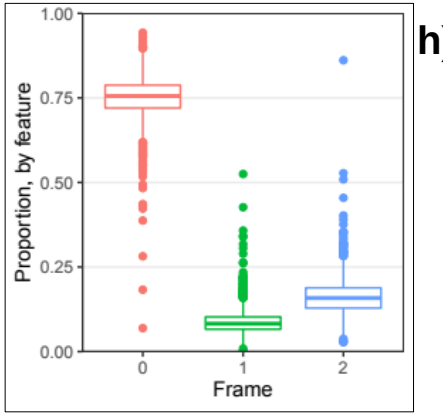


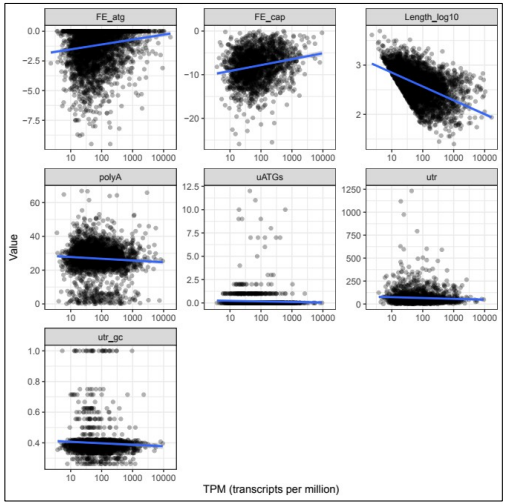












a) 3nt_periodicity.pdf; **b)** startcodon_ribogrid.pdf; **c)** pos_sp_rpf_norm_reads.pdf;
d) codon_ribodens.pdf; **e)** read_lengths.pdf; **f)** 3ntframe_propbygene.pdf; **g)** startcodon_ribogridbar.pdf;
h) features.pdf

B-Sc_2012: ANAPH_1

a)

Pos	Counts	End
-24	832	5'
-23	599	5'
-22	457	5'
-21	497	5'
-20	457	5'
-19	462	5'
-18	429	5'
-17	325	5'

b)

late 2020-03-04 13:45:40

Length	Counts
10	0
11	0
12	0
13	0
14	0
15	37534
16	62944
17	82132
18	107977
19	134409
20	168903
21	139873
22	120034

c)

03-04 13:45:40

Position	Mean	SD	End
1	3.09468381483036	0.461891291073541	5'
2	19.5972312330649	0.923539289394344	5'
3	6.6199958020074	0.406177250696485	5'
4	6.10299870243865	0.360194920428958	5'
5	6.89248368507423	0.358005285654314	5'
6	7.75982234858604	0.428073598442926	5'
7	4.31479696981262	0.228330248444835	5'
8	3.49490039308476	0.222734515131855	5'
9	2.77353738121589	0.181618039919093	5'
10	2.39400068694424	0.168236938518495	5'
11	1.92566213792314	0.115393752623745	5'

d)

4 13:45:40

Length	Position	Frame	A	C	G	T
10	1	0	0	0	0	0
10	2	0	0	0	0	0
...	...	...	...	...	...	...
30	14	0	0.275	0.196	0.334	0.195
30	15	0	0.167	0.0999	0.159	0.574
30	16	0	0.195	0.549	0.0751	0.181
30	17	0	0.143	0.352	0.207	0.298
30	18	0	0.185	0.324	0.161	0.33
30	19	0	0.2	0.222	0.292	0.286
30	20	0	0.165	0.404	0.106	0.325
30	21	0	0.501	0.2	0.162	0.137
30	22	0	0.181	0.24	0.415	0.164
30	23	0	0.143	0.253	0.227	0.377
30	24	0	0.287	0.0952	0.257	0.36
30	25	0	0.285	0.157	0.394	0.164
30	26	0	0.235	0.249	0.11	0.407
30	27	0	0.177	0.426	0.266	0.13
30	28	0	0.39	0.163	0.289	0.158
30	29	0	0.16	0.174	0.125	0.54
30	30	0	0.192	0.305	0.222	0.282
30	1	1	0.145	0.249	0.0958	0.51

e)

AA	Codon	tRNA	tAI	Microarray	RNA.seq	A	P	E
K	AAA	7	0.431034	222273	82386	1.05790819002401	1.15291058609784	0.86528716593696
N	AAC	10	0.615764	378101	110849	1.0281152292804	1.27708638433922	1.0050442206497
K	AAG	14	1	397111	83036	1.0224822832824	1.26675365537118	0.932261297646333
N	AAT	6.4	0.27032	241984.64	70943.36	0.99399477627165	1.35673169342604	0.848902424015964
T	ACA	4	0.246373	105862	47598	0.819260573895541	1.23679352095319	1.25916809965654
T	ACC	7.04	0.487685	244374.4	36396.16	0.712747801072039	0.896873450468225	1.11953584037134
T	ACG	1	0.140394	113104	16861	0.797757582450966	1.24033130452294	1.13475360386371
T	ACT	11	0.67734	381835	56869	0.849232525038719	1.03763261086384	1.17820012062022
R	AGA	11	0.67734	683001	98864	1.28096241997618	0.932921234833255	0.856577205337026
Z	AGC	2	0.123153	249948	61334	0.950093444227615	0.47288787628337	1.01587991491282
R	AGG	1	0.278325	106890	12911	1.24963311379243	0.529507496627423	0.953871071715205
Z	AGT	1.28	0.054064	159966.72	39253.76	1.04600979170965	0.563371259730159	0.914016268311146
I	ATA	2	0.123233	54352	39556	0.724039234288364	0.569831032158084	0.951443079270135
I	ATC	8.32	0.576355	586391.04	67769.6	1.00775448514063	0.817362847783829	1.04955306675208

g)

04 13:45:40

ORF	readcount	rpb	tpm
Q0045	277	0.167675544794189	54.1492075552714
Q0050	5	0.001959247648903	0.632720220010838
Q0275	94	0.109684947491249	35.4217008489988
...	...	...	...
YAL001C	103	0.029178470254958	9.42290686408889
YAL002W	37	0.009555785123967	3.08594907305286
YAL003W	848	1.26946107784431	409.9602686361
YAL005C	10367	5.24645748987854	1694.29308190438
YAL007C	87	0.125179856115108	40.4256328424997
YAL008W	140	0.217391304347826	70.20443484642
YAL009W	33	0.039903264812576	12.886376311592
YAL010C	24	0.015696533682145	5.06904487053288
YAL011W	37	0.019220779220779	6.20716613551204

h)*

-04 13:45:40

ORF	VEG_1	VEG_2	ANAPH_1	ANAPH_2	SPORE_1	SPORE_2
Q0045	0.5	0.2	54.1	88.4	33.1	44.5
Q0050	0.7	0.8	0.6	0.9	2	2.6
Q0055	0.1	0.2	0.4	0.6	0.3	1
Q0275	0.9	0.7	35.4	58.3	29.1	42.1
...	...	...	...	...	...	...
YAL001C	3.4	4.7	9.4	10	97.3	46.9
YAL002W	1.8	2.7	3.1	2.2	2.1	10.9
YAL003W	4043.5	3676.6	410	300.3	398.7	1623.6
YAL005C	1345.1	1386.2	1694.3	1327.5	1173.1	637.1
YAL007C	105.6	87.8	40.4	38.7	11.6	47.6
YAL008W	11.8	14.7	70.2	84.7	103.6	125.8
YAL009W	4.2	5.7	12.9	7.4	7.3	4.9
YAL010C	4.8	5.2	5.1	5.5	1.8	9.2
YAL011W	4.9	5.7	6.2	7.6	6.9	13.1
YAL012W	316	353.5	38.3	23.7	31	60.3
YAL013W	15.1	16.4	35.2	25.5	47.6	85.8
YAL014C	10.7	12	38.4	32.1	47.8	82.5
YAL015C	8.6	12.5	21.1	12.9	16.1	30.4

f)

gene	Ct_fr0	Ct_fr1	Ct_fr2	pval_fr0vs1	pval_fr0vs2	pval_fr0vsboth
YAL068C	0	0	0	1	1	1
YAL067W-A	0	0	0	1	1	1
YAL067C	8	3	0	0.09154837080959	0.007354384721559	0.003834285390189
YAL065C	1	1	0	0.681324055883031	0.5	0.630558659818236
YAL064W-B	1	0	0	0.5	0.5	0.5
YAL064C-A	0	0	0	1	1	1
YAL064W	0	0	0	1	1	1
YAL063C-A	0	0	0	1	1	1
YAL063C	12	2	8	0.018148283043708	0.091757084963638	0.007742087124372
YAL062W	108	10	15	2.35529750560817E-10	9.00694786412824E-09	2.53109099063683E-17
YAL061W	90	18	24	3.58565108086954E-08	1.6306211759895E-06	6.25221450296923E-13
YAL060W	17	2	3	0.000954008674701	0.002457242914611	1.43343558100775E-05
YAL059W	2	1	0	0.386414996342224	0.17288929307558	0.191934191495911

i)*

SampleName	Program	File	NumReads	Description
VEG_1	input	B-Sc_2012/input/SRR387871_GSM843747_	21155927	input
VEG_2	input	B-Sc_2012/input/SRR387872_GSM843748_	20915375	input
ANAPH_1	input	B-Sc_2012/input/SRR387890_GSM843766_	10210064	input
ANAPH_2	input	B-Sc_2012/input/SRR387891_GSM843767_	11620421	input
SPORE_1	input	B-Sc_2012/input/SRR387898_GSM843774_	11782015	input
SPORE_2	input	B-Sc_2012/input/SRR387899_GSM843775_	10793988	input
ANAPH_1	cutadapt	B-Sc_2012/tmp/ANAPH_1/trim.fq	9458608	Reads after removal of sequencing library
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/nonrRNA.fq	5999936	rRNA or other contaminating reads remov
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/rRNA_map.sam	9458608	Reads with rRNA and other contaminating
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/unaligned.fq	2919159	Unaligned reads removed by alignment o
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/orf_map.sam	3348087	Reads aligned to ORFs index files
ANAPH_1	riboviz.tools.trim	B-Sc_2012/tmp/ANAPH_1/orf_map_clean.sa	3348087	Reads after trimming of 5' mismatches a
ANAPH_2	cutadapt	B-Sc_2012/tmp/ANAPH_2/trim.fq	11231649	Reads after removal of sequencing library
ANAPH_2	hisat2	B-Sc_2012/tmp/ANAPH_2/nonrRNA.fq	9365904	rRNA or other contaminating reads remov
ANAPH_2	hisat2	B-Sc_2012/tmp/ANAPH_2/rRNA_map.sam	11231649	Reads with rRNA and other contaminating

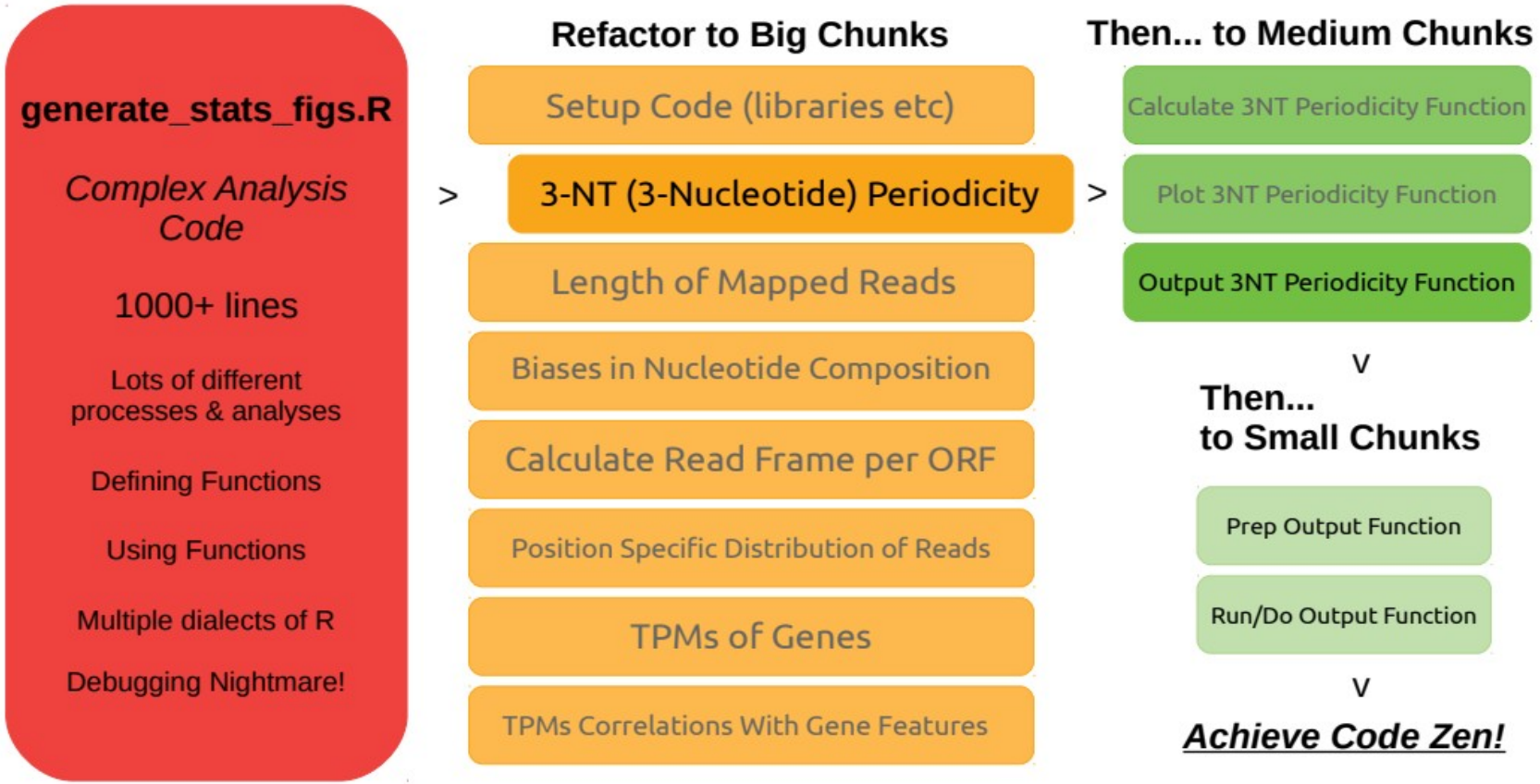
a) 3nt_periodicity.tsv; **b)** read_lengths.tsv; **c)** pos_sp_rpf_norm_reads.tsv;
d) pos_sp_nt_freq.tsv; **e)** codon_ribodens.tsv; **f)** 3ntframe_bygene.tsv; **g)** tpms.tsv;
h) TPMs_collated.tsv*; i**)** read_counts.tsv* * collates data from all 6 samples run

Analysis

generate_stats_figs.R:

Generates summary statistics, analysis plots & quality control plots

Refactoring Analysis Code



generate_stats_figs.R

- Setup Code (libraries, source etc)
- 3-NT (3-Nucleotide) Periodicity
- Length of Mapped Reads
- Biases in Nucleotide Composition
- Calculate Read Frame per ORF
- Position Specific Distribution of Reads
- TPMs of Genes
- TPMs Correlations With Gene Features

Big Chunks

Medium Chunks

stats_figs_block_functions.R

- Calculate 3NT Periodicity Function
- Plot 3NT Periodicity Function
- Output 3NT Periodicity Function

read_count_functions.R

Additional Libraries Loaded

- Prep Output Function
- Run/Do Output Function

Small Chunks

Anatomy of a Code Chunk...

- 3NT Periodicity Big Code Chunk:
  - Calculate! Get start positions & read counts at each position for each gene from the .h5 file, calculate periodicity from this matrix of positions & counts
  - Plots! {ggplot2}
  - Save plots as .pdf
  - Write information out as .tsv (includes provenance info)

```r
ThreeNucleotidePeriodicity <- function(gene_names, dataset, hd_file, gff_df) {

  # check for 3nt periodicity
  print("Starting: Check for 3nt periodicity globally")

  # CalculateThreeNucleotidePeriodicity():
  three_nucleotide_periodicity_data <- CalculateThreeNucleotidePeriodicity(gene_names = gene_names, dataset = dataset, hd_file =

  # PlotThreeNucleotidePeriodicity()
  three_nucleotide_periodicity_plot <- PlotThreeNucleotidePeriodicity(three_nucleotide_periodicity_data)

  # NOTE: repeated from inside CalculateThreeNucleotidePeriodicity() as preferred not to return multiple objects in list (hassle
  gene_poslen_counts_5start_df <- AllGenes5StartPositionLengthCountsTibble(gene_names = gene_names, dataset= dataset, hd_file =

  # run PlotStartCodonRiboGrid()
  start_codon_ribogrid_plot <- PlotStartCodonRiboGrid(gene_poslen_counts_5start_df)
  # creates plot object

  # run SaveStartCodonRiboGrid():
  SaveStartCodonRiboGrid(start_codon_ribogrid_plot)

  # run PlotStartCodonRiboGridBar():
  start_codon_ribogrid_bar_plot <- PlotStartCodonRiboGridBar(gene_poslen_counts_5start_df)
  # creates plot object

  # run SaveStartCodonRiboGridBar():
  SaveStartCodonRiboGridBar(start_codon_ribogrid_bar_plot)

  # run SavePlotThreeNucleotidePeriodicity():
  SavePlotThreeNucleotidePeriodicity(three_nucleotide_periodicity_plot)

  # run WriteThreeNucleotidePeriodicity():
  WriteThreeNucleotidePeriodicity(three_nucleotide_periodicity_data)

  print("Completed: Check for 3nt periodicity globally")

} # end ThreeNucleotidePeriodicity() function definition
# run ThreeNucleotidePeriodicity():
ThreeNucleotidePeriodicity(gene_names, dataset, hd_file, gff_df)
```

```r

CalculateThreeNucleotidePeriodicity <- function(gene_names, dataset, hd_file, gff_df){

  # get gene and position specific total counts for all read lengths
  gene_poslen_counts_5start_df <- AllGenes5StartPositionLengthCountsTibble(gene_names = gene_names, dataset=

  gene_poslen_counts_3end_df <- AllGenes3EndPositionLengthCountsTibble(gene_names = gene_names, dataset= da

  # summarize by adding different read lengths
  gene_pos_counts_5start <- gene_poslen_counts_5start_df %>%
    group_by(Pos) %>%
    summarize(Counts = sum(Counts))
  # gives:
  # > str(gene_pos_counts_5start)
  # Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   75 obs. of  2 variables:
  #   $ Pos   : int  -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 ...
  #   $ Counts: int  285 318 307 386 291 347 840 330 475 355 ...

  gene_pos_counts_3end <- gene_poslen_counts_3end_df  %>%
    group_by(Pos) %>%
    summarize(Counts = sum(Counts))
  # gives:
  # > str(gene_pos_counts_3end)
  # Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   75 obs. of  2 variables:
  #   $ Pos   : int  -49 -48 -47 -46 -45 -44 -43 -42 -41 -40 ...
  #   $ Counts: int  19030 13023 50280 19458 12573 46012 19043 13282 36968 20053 ...

  three_nucleotide_periodicity_data <- bind_rows(
    gene_pos_counts_5start %>% mutate(End = "5'"),
    gene_pos_counts_3end %>% mutate(End = "3'")
  ) %>%
    mutate(End = factor(End, levels = c("5'", "3'")))
  # gives:
  # > str(three_nucleotide_periodicity_data)
  # Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   150 obs. of  3 variables:
  #   $ Pos   : int  -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 ...
  #   $ Counts: int  285 318 307 386 291 347 840 330 475 355 ...
  #   $ End   : Factor w/ 2 levels "5'","3'": 1 1 1 1 1 1 1 1 1 1 ...

  return(three_nucleotide_periodicity_data)

} # end CalculateThreeNucleotidePeriodicity() definition
# gives:
#   CalculateThreeNucleotidePeriodicity(gene_names = gene_names, dataset = dataset, hd_file = hd_file, gff_
#   # A tibble: 150 x 3
#   Pos Counts End
#   <int>  <int> <fct>
#     1   -24    285 5'
#     2   -23    318 5'
#     3   -22    307 5'
#     4   -21    386 5'
#     5   -20    291 5'
#     6   -19    347 5'
#     7   -18    840 5'
#     8   -17    330 5'
#     9   -16    475 5'
#    10   -15    355 5'
#   # … with 140 more rows

```

```
238
239 # define PlotThreeNucleotidePeriodicity() function with reasonable arguments
240 PlotThreeNucleotidePeriodicity <- function(three_nucleotide_periodicity_data){
241
242   # Plot
243   three_nucleotide_periodicity_plot <- ggplot(
244     three_nucleotide_periodicity_data,
245     aes(x = Pos, y = Counts)) +
246     geom_line() +
247     facet_wrap(~End, scales = "free") +
248     labs(x = "Nucleotide Position", y = "Read counts")
249
250   return(three_nucleotide_periodicity_plot)
251
252 } # end PlotThreeNucleotidePeriodicity() definition
253
254 # potentially replace/tweak plot_ribogrid() to follow StyleGuide
255 PlotStartCodonRiboGrid <- function(gene_poslen_counts_5start_df){
256   # function to do the ribogrid & ribogridbar plots?
257   # ribogrid_5start
258   start_codon_ribogrid_plot <- plot_ribogrid(gene_poslen_counts_5start_df)
259   return(start_codon_ribogrid_plot)
260 } # end PlotStartCodonRiboGrid() definition
261
262 SaveStartCodonRiboGrid <- function(start_codon_ribogrid_plot){
263   # function to do the ribogrid & ribogridbar plots?
264   # ribogrid_5start
265   start_codon_ribogrid_plot %>%
266     ggsave(
267       filename = file.path(output_dir, paste0(output_prefix, "startcodon_ribogrid.pdf")),
268       width = 6, height = 3
269     )
270   #return() # no return as writing-out
271 } # end SaveStartCodonRiboGrid() definition
272
273 PlotStartCodonRiboGridBar <- function(gene_poslen_counts_5start_df){
274   start_codon_ribogrid_bar_plot <- barplot_ribogrid(gene_poslen_counts_5start_df)
275   return(start_codon_ribogrid_bar_plot)
276 } # end PlotStartCodonRiboGridBar() definition
277
278 SaveStartCodonRiboGridBar <- function(start_codon_ribogrid_bar_plot){
279   start_codon_ribogrid_bar_plot %>%
280     ggsave(
281       filename = file.path(output_dir, paste0(output_prefix, "startcodon_ribogridbar.pdf")),
282       width = 6, height = 5
283     )
284   #return() # no return as writing-out
285 } # end SaveStartCodonRiboGridBar() definition
286
287 SavePlotThreeNucleotidePeriodicity <- function(three_nucleotide_periodicity_plot) {
288   # Save plot and file
289   ggsave(
290     three_nucleotide_periodicity_plot,
291     filename = file.path(output_dir, paste0(output_prefix, "3nt_periodicity.pdf"))
292   )
293   # return() # NO RETURN as writing out
294 } # end of function definition SavePlotThreeNucleotidePeriodicity()
295
```

```
296 WriteThreeNucleotidePeriodicity <- function(three_nucleotide_periodicity_data) {
297   tsv_file_path <- file.path(output_dir, paste0(output_prefix, "3nt_periodicity.tsv"))
298   write_provenance_header(path_to_this_script, tsv_file_path)
299   write.table(
300     three_nucleotide_periodicity_data,
301     file = tsv_file_path,
302     append = T,
303     sep = "\t",
304     row = F,
305     col = T,
306     quote = F)
307   # return()? NO RETURN
308 }  # end of function definition WriteThreeNucleotidePeriodicity()
309
```

Refactoring Tips

- Break it down into **chunks**!
- **Regression tests** are your friend, but fails are not always a failure...
- Develop functions in your main script & then move into **separate functions script**
- Keep it specific: **issue-focussed working**

  … Decide how you'll know you're finished...

Refactoring LFMFs

- **Issue proliferation**… Getting lost amongst different issues & losing sight of the main goals
- Schroedinger's **debugging** issues!
- **Package::function()** is very useful!
- … Remember to **tell collaborators** about & document the new packages you add to the code to solve problems...

“Putting the YOU into USER”

- **User Testing** & User Interviews
- Existing **Documentation Feedback** – *Siyin &* others already helping with this :)
- Leave the **Code Docs** to the Robots? – {roxygen2}

“Testable, Reliable CompYOUtation”

- **Code Testing**:
  - Helping integrate functions from *Ania* & *Siyin’s* projects: want to have good testable code…
  - **{testthat} R package** – automated testing
- **More datasets**!

Priorities

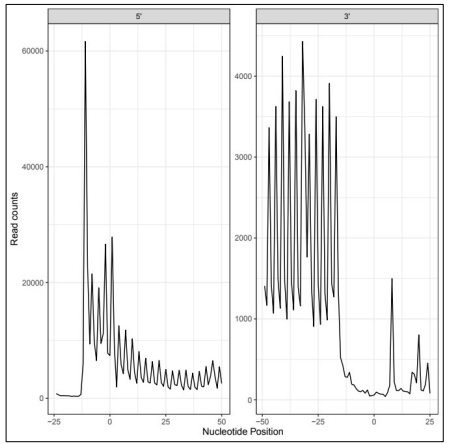
- **Tests** for the analysis code
- generate_stats_figs.R **finishing touches:** Documentation / better output format / styling
- **New datasets** run & added to example datasets repository
- Q: How does riboviz compare with **other tools**
- Q: Do we have the right **statistics & outputs** for diagnostics?

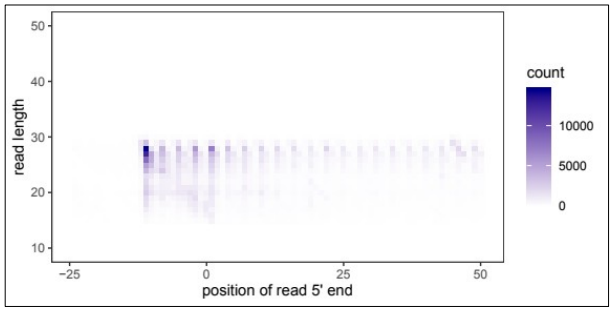
Thanks / Acknowledgements

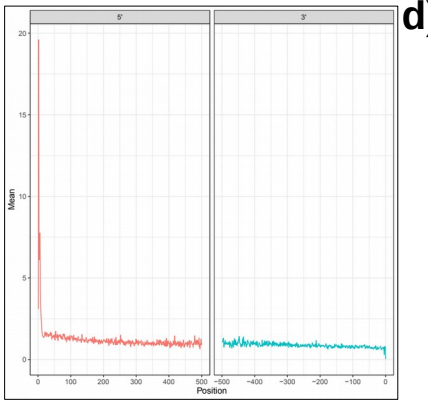
- BBSRC-NSF funded project
- Collaborative project:
  - Edward Wallace: *The Wallace Lab,* The University of Edinburgh.

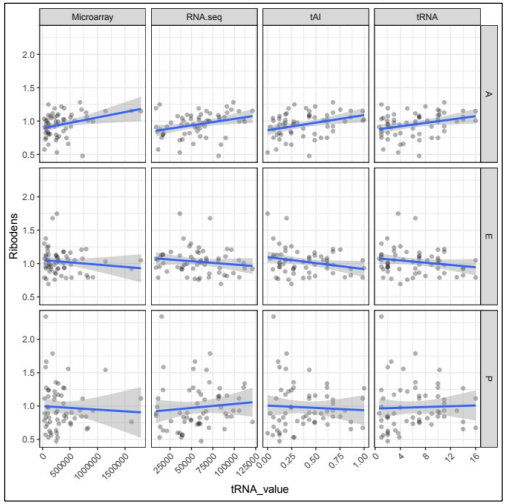
    \+ Siyin Xue, Ania Kurowska
  - Premal Shah, John Favate, Tongji Xing: *The Shah Lab,* Rutgers University.
  - Liana Lareau, Amanda Mok: *The Lareau Lab,* University of California, Berkeley.
  - Kostas Kavousannakis, Mike Jackson: *EPCC,* The University of Edinburgh.
  - Oana Carja, Joshua Plotkin: The University of Pennsylvania

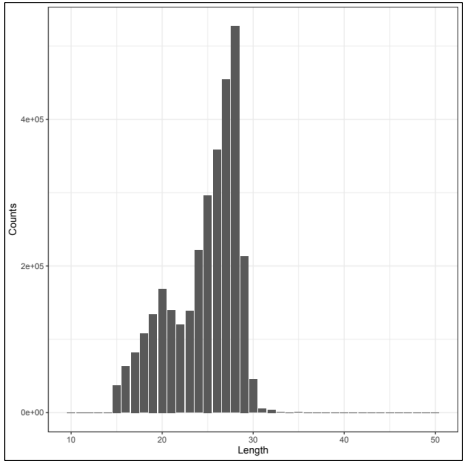
B-Sc_2012: ANAPH_1

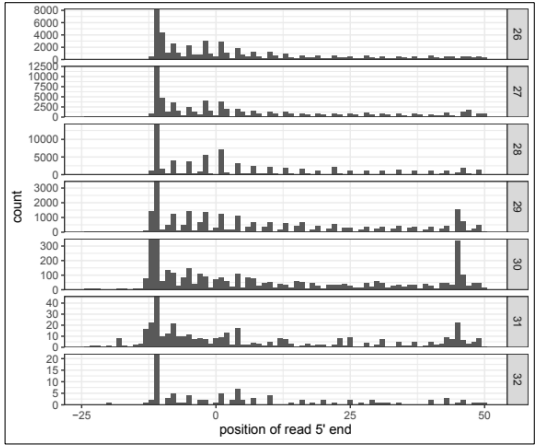
a) 

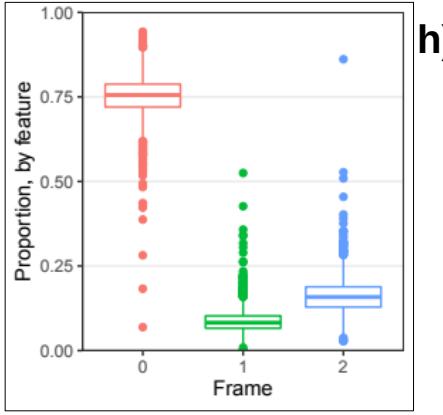
b) 

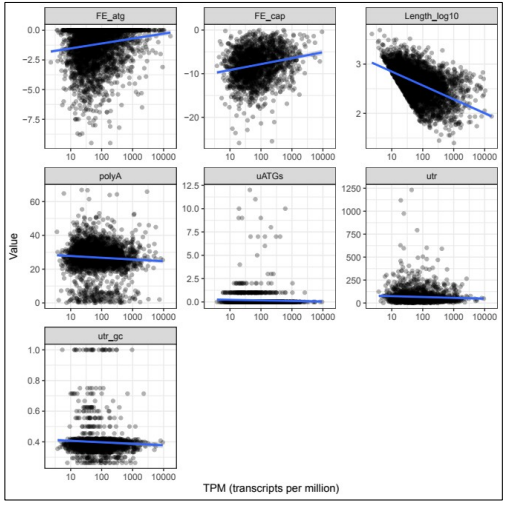
c) 

d) 

e) 

f) 

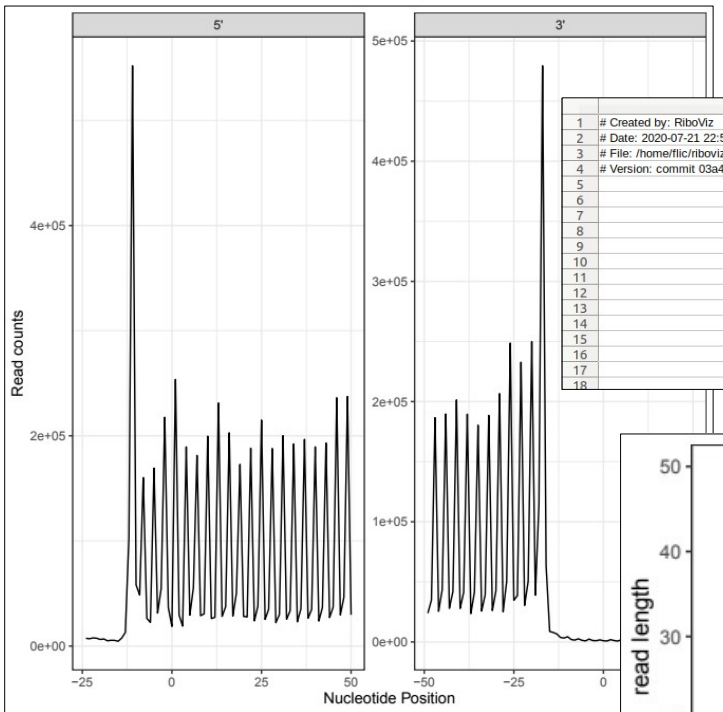
g) 

h) 

a) 3nt_periodicity.pdf; **b)** startcodon_ribogrid.pdf; **c)** pos_sp_rpf_norm_reads.pdf;
d) codon_ribodens.pdf; **e)** read_lengths.pdf; **f)** 3ntframe_propbygene.pdf; **g)** startcodon_ribogridbar.pdf;
h) features.pdf

High-resolution View of the Yeast Meiotic Program Revealed by Ribosome Profiling, *Brar et al.* 2012

Outputs: Three Nucleotide Periodicity

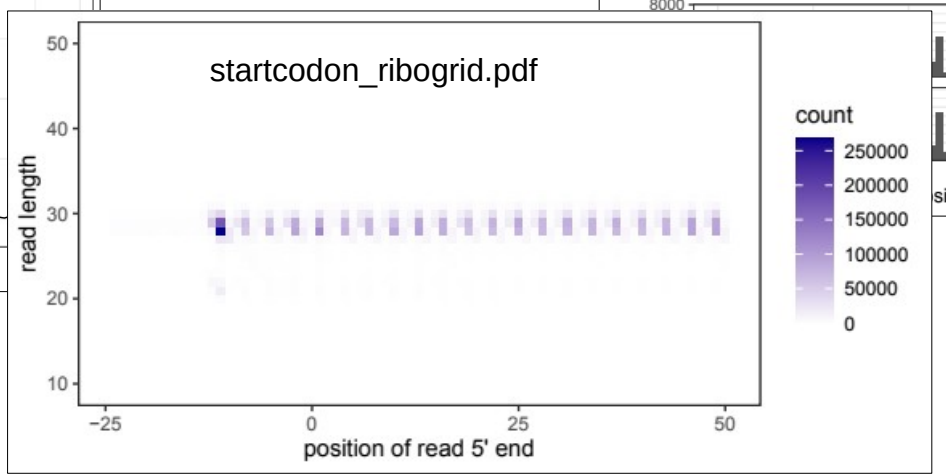


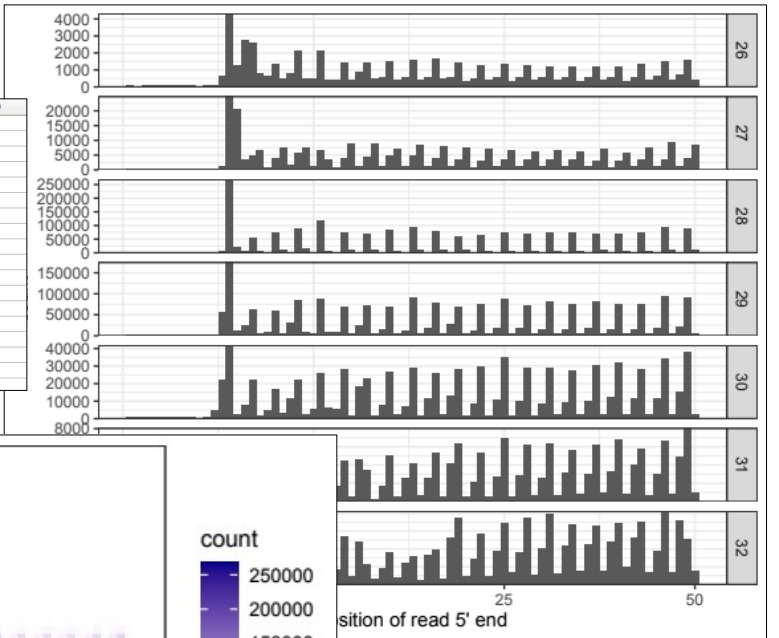
3nt_periodicity.pdf

W-Cn-H99_2020: H99r1

	A	B	C	D
1	# Created by: RiboViz			
2	# Date: 2020-07-21 22:50:30			
3	# File: /home/flic/riboviz/rscripts/generate_stats_figs.R			
4	# Version: commit 03a4f13b2d833b8650ebf33bdce81fe2639eb9cf date 2020-07-08 15:42:55			
5		Pos	Counts	End
6		-24	7355	5'
7		-23	6949	5'
8		-22	7728	5'
9		-21	7435	5'
10		-20	6284	5'
11		-19	6731	5'
12		-18	5038	5'
13		-17	5474	5'
14		-16	5424	5'
15		-15	4560	5'
16		-14	7614	5'
17		-13	12959	5'
18		-12	102718	5'

3nt_periodicity.tsv

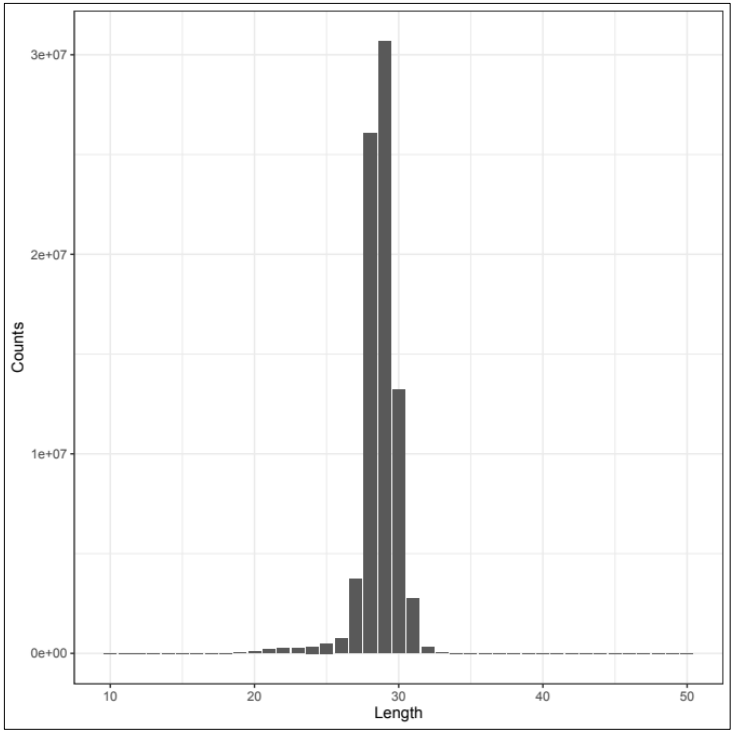




startcodon_ribogridbar.pdf

Quantitative global studies reveal differential translational control by start codon context across the fungal kingdom. *Wallace et al.* 2020

Outputs: Lengths of Mapped Reads



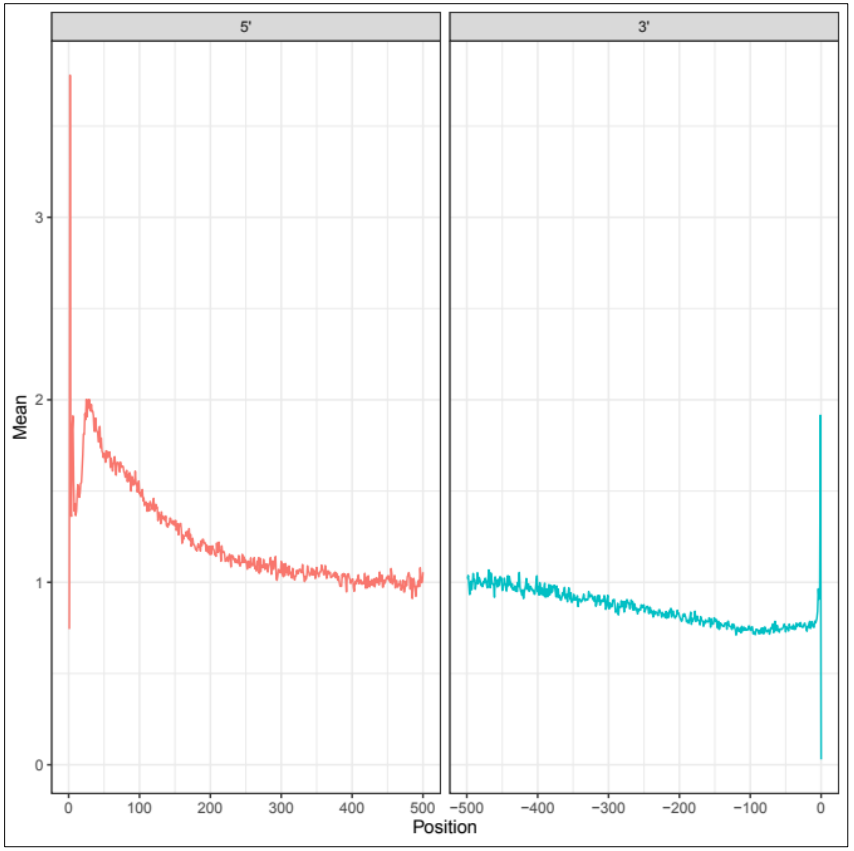
read_lengths.pdf

	A	B	C
1	# Created by: RiboViz		
2	# Date: 2020-07-21 22:50:42		
3	# File: /home/flic/riboviz/rscripts/generate_stats_figs.R		
4	# Version: commit 03a4f13b2d833b8650ebf33bdce81fe2639eb9cf date 2020-07-08 15:42:55		
5	Length	Counts	
6	10	0	
7	11	0	
8	12	0	
9	13	0	
10	14	0	
11	15	970	
12	16	2476	
13	17	4118	
14	18	11192	
15	19	28319	
16	20	76530	
17	21	214894	
18	22	266016	
19	23	243897	
20	24	330478	
21	25	489754	
22	26	748587	
23	27	3718527	
24	28	26058264	
25	29	30665395	
26	30	13218447	

read_lengths.tsv

W-Cn-H99_2020: H99r1

Outputs: Position Specific Distribution of Reads



pos_sp_rpf_norm_reads.pdf

	A	B	C	D	E
1	# Created by: RiboViz				
2	# Date: 2020-07-21 22:50:59				
3	# File: /home/flic/riboviz/rscripts/generate_stats_figs.R				
4	# Version: commit 03a4f13b2d833b8650ebf33bdce81fe2639eb9cf date 2020-07-08 15:42:55				
5	Position	Mean	SD	End	
6	1	0.743715220949264	0.031229600187047	5'	
7	2	3.77879354687865	0.072439794248305	5'	
8	3	1.37367313537526	0.046337619826982	5'	
9	4	1.36174888940846	0.043273088613514	5'	
10	5	1.83991115267711	0.054171849427169	5'	
11	6	1.91145662847791	0.055662380173019	5'	
12	7	1.55253682487725	0.045586392331073	5'	
13	8	1.39036707972878	0.035570025718962	5'	
14	9	1.42971709141922	0.038014496142156	5'	
15	10	1.3653261631985	0.035975450081833	5'	
16	11	1.39036707972878	0.0390399812953	5'	
17	12	1.4786064998831	0.041663315407996	5'	
18	13	1.53465045592705	0.04410778583119	5'	
19	14	1.51914893617021	0.038586859948562	5'	
20	15	1.46548982931962	0.035605798456862	5'	
21	16	1.48814589665653	0.035760813654431	5'	
22	17	1.54180500350713	0.037191723170447	5'	
23	18	1.54538227729717	0.03870610240823	5'	
24	19	1.62050502688801	0.039244797755444	5'	
25	20	1.69920505026888	0.037370586859949	5'	
26	21	1.81248538695347	0.047363104980126	5'	

pos_sp_rpf_norm_reads.tsv

W-Cn-H99_2020: H99r1

Outputs: TPMs of Genes

	A	B	C	D	E
1	# Created by: RiboViz				
2	# Date: 2020-07-21 22:51:47				
3	# File: /home/flic/riboviz/rscripts/generate_stats_figs.R				
4	# Version: commit 03a4f13b2d833b8650ebf33bdce81fe2639eb9cf date 2020-07-08 15:42:55				
5	ORF	readcount	rpb	tpm	
6	CNAG_00002	3479	6.08216783216783	67.2888954230649	
7	CNAG_00003	33	0.019903498190591	0.220198528757548	
8	CNAG_00004	1471	0.62649063032368	6.93105873935655	
9	CNAG_00005	1390	0.77914798206278	8.61995402475237	
10	CNAG_00006	15122	15.0617529880478	166.632810811397	
11	CNAG_00007	3508	1.9532293986637	21.6091782355655	
12	CNAG_00008	3848	1.08181051447849	11.9684028104778	
13	CNAG_00009	6559	5.72838427947598	63.3748790498934	
14	CNAG_00010	901	0.298641034139874	3.30395771208298	
15	CNAG_00011	1214	1.2019801980198	13.2978770196681	
16	CNAG_00012	7059	5.84354304635762	64.6489159452195	
17	CNAG_00013	4	0.006042296072508	0.066847781869453	
18	CNAG_00014	844	0.446797247220752	4.94305551460502	
19	CNAG_00015	129	0.188046647230321	2.08041795775784	
20	CNAG_00016	8997	29.7913907284768	329.591328380303	
21	CNAG_00017	22	0.009136212624585	0.101076733300115	
22	CNAG_00018	4658	1.91058244462674	21.1373618520833	
23	CNAG_00019	20962	66.1261829652997	731.5743223568	
24	CNAG_00020	4419	1.45314041433739	16.0765398248681	
25	CNAG_00021	10177	7.01378359751895	77.5956474790743	
26	CNAG_00022	8424	9.82963827304551	108.748314754374	
27	CNAG_00023	19	0.016769638128861	0.185527669981018	
28	CNAG_00024	4089	1.85274127775261	20.4974472136945	

tpms.tsv

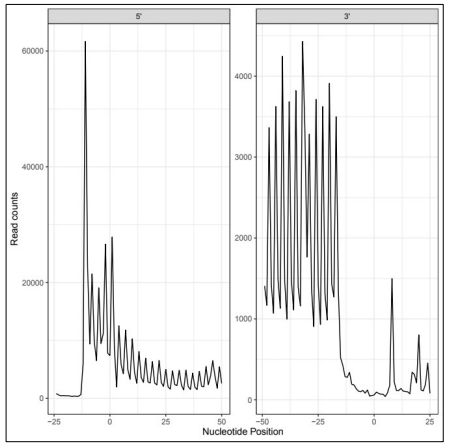
W-Cn-H99_2020: H99r1

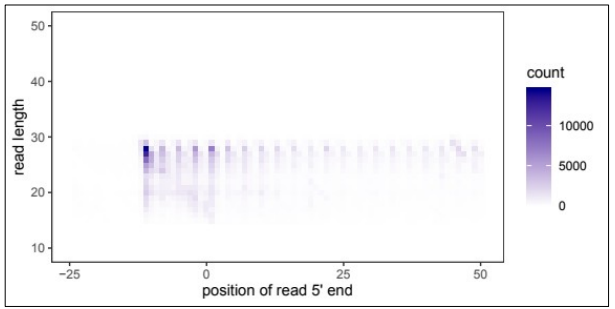
	A	B	C	D	E	F
1	# Created by: RiboViz					
2	# Date: 2020-07-22 01:08:37					
3	# File: /home/flic/riboviz/rscripts/collate_tpms.R					
4	# Version: commit 03a4f13b2d833b8650ebf33bdce81fe2639eb9cf date 2020-07-08 15:42:55					
5	ORF	H99r1	HdGWO1	H99r2	HdAGO1	
6	CNAG_00002	67.3	62.2	53	59.9	
7	CNAG_00003	0.2	0.1	0.1	0.1	
8	CNAG_00004	6.9	6.1	6.4	7	
9	CNAG_00005	8.6	7.5	8.6	9.7	
10	CNAG_00006	166.6	172.3	184.8	187.5	
11	CNAG_00007	21.6	22.6	29.6	29.4	
12	CNAG_00008	12	11.3	13.9	14.4	
13	CNAG_00009	63.4	58.8	56.1	59.7	
14	CNAG_00010	3.3	3.3	3.8	4	
15	CNAG_00011	13.3	11.6	11.7	13.4	
16	CNAG_00012	64.6	64.4	64.6	68.3	
17	CNAG_00013	0.1	0	0.1	0.1	
18	CNAG_00014	4.9	4.8	4.9	4.7	
19	CNAG_00015	2.1	1.7	2.3	2.1	
20	CNAG_00016	329.6	301.9	297.9	292.5	
21	CNAG_00017	0.1	0.1	0.1	0.1	
22	CNAG_00018	21.1	21.1	21	22.4	
23	CNAG_00019	731.6	712.9	726.4	712.2	
24	CNAG_00020	16.1	16.8	19.4	20.5	
25	CNAG_00021	77.6	79.2	84.9	82.1	
26	CNAG_00022	108.7	106.2	107.5	108	
27	CNAG_00023	0.2	0.2	0.1	0.1	
28	CNAG_00024	20.5	22.2	34.8	38.1	
29	CNAG_00025	8.5	9.1	13	13	
30	CNAG_00026	213.2	196.3	225.3	243.5	
31	CNAG_00027	4.4	3.8	3.9	4.2	
32	CNAG_00028	1.1	1.2	1	0.9	
33	CNAG_00029	1.4	1.6	1.2	1.4	
34	CNAG_00030	4.1	2.8	2.7	3	

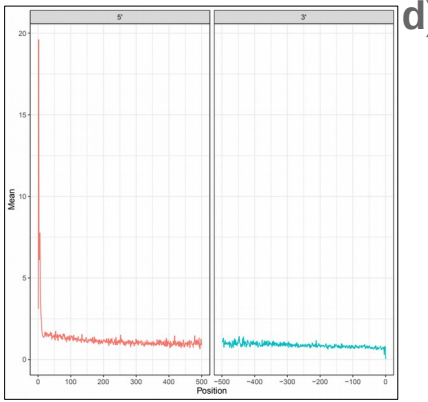
TPMs_collated.tsv

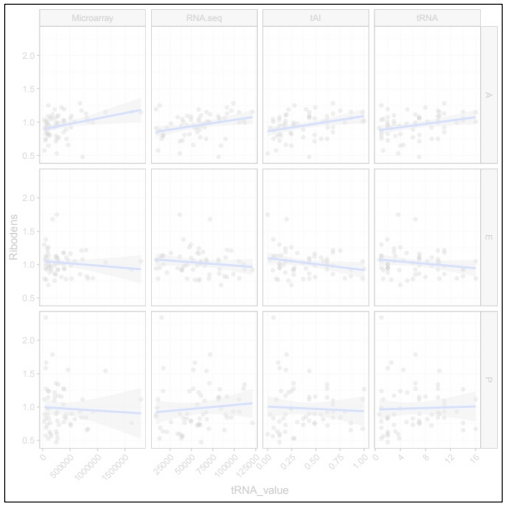
W-Cn-H99_2020: H99r1, HdGWO1, H99r2, HdAGO1

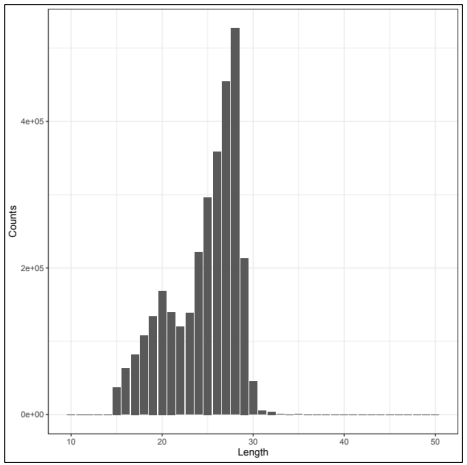
B-??_????: ANAPH_1

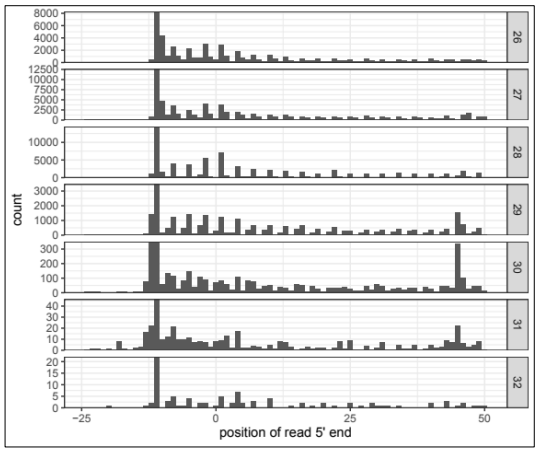


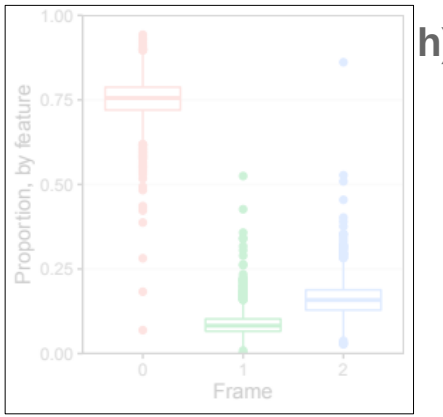


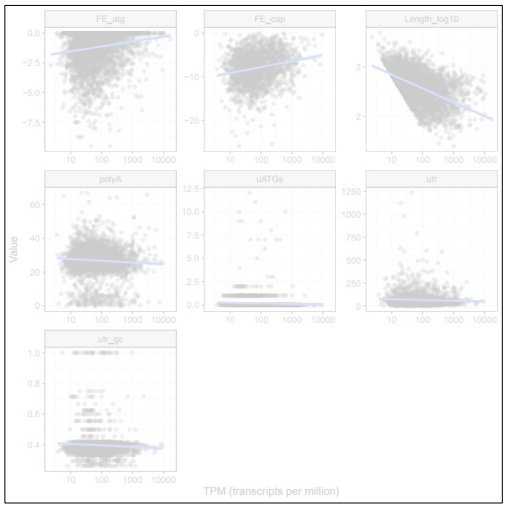












a) 3nt_periodicity.pdf; **b)** startcodon_ribogrid.pdf; **c)** pos_sp_rpf_norm_reads.pdf;
d) codon_ribodens.pdf; **e)** read_lengths.pdf; **f)** 3ntframe_propbygene.pdf; **g)** startcodon_ribogridbar.pdf;
h) features.pdf

High-resolution View of the Yeast Meiotic Program Revealed by Ribosome Profiling, *Brar et al.* 2012

B-??_????: ANAPH_1

a)

Pos	Counts	End
-24	832	5'
-23	599	5'
-22	457	5'
-21	497	5'
-20	457	5'
-19	462	5'
-18	429	5'
-17	325	5'

b)

late 2020-03-04 13:45:40

Length	Counts
10	0
11	0
12	0
13	0
14	0
15	37534
16	62944
17	82132
18	107977
19	134409
20	168903
21	139873
22	120034

c)

03-04 13:45:40

Position	Mean	SD	End
1	3.09468381483036	0.461891291073541	5'
2	19.5972312330649	0.923539289394344	5'
3	6.6199958020074	0.406177250696485	5'
4	6.10299870243865	0.360194920428958	5'
5	6.89248368507423	0.358005285654314	5'
6	7.75982234858604	0.428073598442926	5'
7	4.31479696981262	0.228330248444835	5'
8	3.49490039308476	0.222734515131855	5'
9	2.77353738121589	0.181618039919093	5'
10	2.39400068694424	0.16823693851849	5'
11	1.92566213792314	0.115393752623745	5'

d)

Length	Position	Frame	A	C	G	T
10	1	0	0	0	0	0
10	2	0	0	0	0	0
...	...	...	...	...	...	...
30	14	0	0.275	0.196	0.334	0.195
30	15	0	0.167	0.0999	0.159	0.574
30	16	0	0.195	0.549	0.0751	0.181
30	17	0	0.143	0.352	0.207	0.298
30	18	0	0.185	0.324	0.161	0.33
30	19	0	0.2	0.222	0.292	0.286
30	20	0	0.165	0.404	0.106	0.325
30	21	0	0.501	0.2	0.162	0.137
30	22	0	0.181	0.24	0.415	0.164
30	23	0	0.143	0.253	0.227	0.377
30	24	0	0.287	0.0952	0.257	0.36
30	25	0	0.285	0.157	0.394	0.164
30	26	0	0.235	0.249	0.11	0.407
30	27	0	0.177	0.426	0.266	0.13
30	28	0	0.39	0.163	0.289	0.158
30	29	0	0.16	0.174	0.125	0.54
30	30	0	0.192	0.305	0.222	0.282
30	1	1	0.145	0.249	0.0958	0.51

e)

AA	Codon	tRNA	tAI	Microarray	RNA-seq	A	P	E
K	AAA	7	0.431034	222273	82386	1.05790819002401	1.15291058609784	0.86528716593696
N	AAC	10	0.615764	378101	110849	1.02811522928041	1.27708638433922	1.0050442206497
K	AAG	14	1	397111	83036	1.0224822832824	1.26675365537118	0.932261297646333
N	AAT	6.4	0.27032	241984.64	70943.36	0.99399477627165	1.35673169342604	0.848902424015964
T	ACA	4	0.246373	105862	47598	0.819260573895541	1.23679352095319	1.25916809965654
T	ACC	7.04	0.487685	244374.4	36396.16	0.712747801072039	0.896873450468225	1.11953584037134
T	ACG	1	0.140394	113104	16861	0.797757582450966	1.24033130452294	1.13475360386371
T	ACT	11	0.67734	381835	56869	0.849232525038719	1.03763261086384	1.17820012062022
R	AGA	11	0.67734	683001	98864	1.28096241997618	0.932921234833255	0.856577205337026
Z	AGC	2	0.123153	249948	61334	0.950093444227615	0.472887876283237	1.01587991491282
R	AGG	1	0.278325	106890	12911	1.24963311379243	0.529507496627423	0.953871071715205
Z	AGT	1.28	0.054064	159966.72	39253.76	1.04600979170965	0.563371259730159	0.914016283111146
I	ATA	2	0.123233	54352	39556	0.724039234288364	0.569831032158084	0.951443079270135
I	ATC	8.32	0.576355	586391.04	67769.6	1.00775448514063	0.817362847783829	1.04955306675208

f)

gene	Ct_fr0	Ct_fr1	Ct_fr2	pval_fr0vs1	pval_fr0vs2	pval_fr0vsboth
YAL068C	0	0	0	1	1	1
YAL067W-A	0	0	0	1	1	1
YAL067C	8	3	0	0.09154837080959	0.007354384721559	0.003834285390189
YAL065C	1	1	0	0.681324055883031	0.5	0.630558659818236
YAL064W-B	1	0	0	0.5	0.5	0.5
YAL064C-A	0	0	0	1	1	1
YAL064W	0	0	0	1	1	1
YAL063C-A	0	0	0	1	1	1
YAL063C	12	2	8	0.018148283043708	0.091757084963638	0.007742087124372
YAL062W	108	10	15	2.35529750560817E-10	9.00694786412824E-09	2.53109099063683E-17
YAL061W	90	18	24	3.58565108086954E-08	1.6306211759895E-06	6.25221450296923E-13
YAL060W	17	2	3	0.000954008674701	0.002457242914611	1.43343558100775E-05
YAL059W	2	1	0	0.386414996342224	0.17288929307558	0.191934191495911

g)

04 13:45:40

ORF	readcount	rpb	tpm
Q0045	277	0.167675544794189	54.1492075552714
Q0050	5	0.001959247648903	0.632720220010838
Q0275	94	0.109684947491249	35.4217008489988
...	...	...	...
YAL001C	103	0.029178470254958	9.42290686408889
YAL002W	37	0.009555785123967	3.08594907305286
YAL003W	848	1.26946107784431	409.9602686361
YAL005C	10367	5.24645748987854	1694.29308190438
YAL007C	87	0.125179856115108	40.4256328424997
YAL008W	140	0.217391304347826	70.20443484642
YAL009W	33	0.039903264812576	12.886376311592
YAL010C	24	0.015696533682145	5.06904487053288
YAL011W	37	0.019220779220779	6.20716613551204

h)*

-04 13:45:40

ORF	VEG_1	VEG_2	ANAPH_1	ANAPH_2	SPORE_1	SPORE_2
Q0045	0.5	0.2	54.1	88.4	33.1	44.5
Q0050	0.7	0.8	0.6	0.9	2	2.6
Q0055	0.1	0.2	0.4	0.6	0.3	1
Q0275	0.9	0.7	35.4	58.3	29.1	42.1
...	...	...	...	...	...	...
YAL001C	3.4	4.7	9.4	10	97.3	46.9
YAL002W	1.8	2.7	3.1	2.2	2.1	10.9
YAL003W	4043.5	3676.6	410	300.3	398.7	1623.6
YAL005C	1345.1	1386.2	1694.3	1327.5	1173.1	637.1
YAL007C	105.6	87.8	40.4	38.7	11.6	47.6
YAL008W	11.8	14.7	70.2	84.7	103.6	125.8
YAL009W	4.2	5.7	12.9	7.4	7.3	4.9
YAL010C	4.8	5.2	5.1	5.5	1.8	9.2
YAL011W	4.9	5.7	6.2	7.6	6.9	13.1
YAL012W	316	353.5	38.3	23.7	31	60.3
YAL013W	15.1	16.4	35.2	25.5	47.6	85.8
YAL014C	10.7	12	38.4	32.1	47.8	82.5
YAL015C	8.6	12.5	21.1	12.9	16.1	30.4

i)*

SampleName	Program	File	NumReads	Description
VEG_1	input	B-Sc_2012/input/SRR387871_GSM843747_	21155927	input
VEG_2	input	B-Sc_2012/input/SRR387872_GSM843748_	20915375	input
ANAPH_1	input	B-Sc_2012/input/SRR387890_GSM843766_	10210064	input
ANAPH_2	input	B-Sc_2012/input/SRR387891_GSM843767_	11620421	input
SPORE_1	input	B-Sc_2012/input/SRR387898_GSM843774_	11782015	input
SPORE_2	input	B-Sc_2012/input/SRR387899_GSM843775_	10793988	input
ANAPH_1	cutadapt	B-Sc_2012/tmp/ANAPH_1/trim.fq	9458608	Reads after removal of sequencing library
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/nonrRNA.fq	5999936	rRNA or other contaminating reads remov
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/rRNA_map.sam	9458608	Reads with rRNA and other contaminating
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/unaligned.fq	2919159	Unaligned reads removed by alignment o
ANAPH_1	hisat2	B-Sc_2012/tmp/ANAPH_1/orf_map.sam	3348087	Reads aligned to ORFs index files
ANAPH_1	riboviz.tools.trim	B-Sc_2012/tmp/ANAPH_1/orf_map_clean.sa	3348087	Reads after trimming of 5' mismatches a
ANAPH_2	cutadapt	B-Sc_2012/tmp/ANAPH_2/trim.fq	11231649	Reads after removal of sequencing library
ANAPH_2	hisat2	B-Sc_2012/tmp/ANAPH_2/nonrRNA.fq	9365904	rRNA or other contaminating reads remov
ANAPH_2	hisat2	B-Sc_2012/tmp/ANAPH_2/rRNA_map.sam	11231649	Reads with rRNA and other contaminating

a) 3nt_periodicity.tsv; **b)** read_lengths.tsv; **c)** pos_sp_rpf_norm_reads.tsv;
d) pos_sp_nt_freq.tsv; **e)** codon_ribodens.tsv; **f)** 3ntframe_bygene.tsv; **g)** tpms.tsv;
h) TPMs_collated.tsv*; i**)** read_counts.tsv* * collates data from all 6 samples run

EXTRA SLIDES

Footprints to Ribosome Profiling Data

- **Processing**: *lots of steps*
  - Removing **adapter** sequences
  - Remove **UMIs** (Unique Molecular Identifiers) & **barcodes** if present
  - **Demultiplex** / **Deduplicate** reads if required
  - Need to filter out **contaminant** reads
  - **Align** reads to transcriptome

- **Analysis**: *more steps*
  - **Analyse** & quantify data:
  - Create outputs (including for quality-control, further analysis)

Process ribosome profiling sample data

If sample files (`fq_files`) are specified, then the workflow processes the sample files as follows:

1. Read configuration information from YAML configuration file.
2. Build hisat2 indices if requested (if `build_indices: TRUE`) using `hisat2 build` and save these into the index directory (`dir_index`).
3. Process each sample ID-sample file pair (`fq_files`) in turn:
    i. Cut out sequencing library adapters (`adapters`) using `cutadapt` .
    ii. Extract UMIs using `umi_tools extract` , if requested (if `extract_umis: TRUE`), using a UMI-tools-compliant regular expression pattern (`umi_regexp`). The extracted UMIs are inserted into the read headers of the FASTQ records.
    iii. Remove rRNA or other contaminating reads by alignment to rRNA index files (`rrna_index_prefix`) using `hisat2` .
    iv. Align remaining reads to ORFs index files (`orf_index_prefix`). using `hisat2` .
    v. Trim 5' mismatches from reads and remove reads with more than 2 mismatches using `trim_5p_mismatch` .
    vi. Output UMI groups pre-deduplication using `umi_tools group` if requested (if `dedup_umis: TRUE` and `group_umis: TRUE`)
    vii. Deduplicate reads using `umi_tools dedup` , if requested (if `dedup_umis: TRUE`)
    viii. Output UMI groups post-deduplication using `umi_tools group` if requested (if `dedup_umis: TRUE` and `group_umis: TRUE`)
    ix. Export bedgraph files for plus and minus strands, if requested (if `make_bedgraph: TRUE`) using `bedtools genomecov` .
    x. Write intermediate files produced above into a sample-specific directory, named using the sample ID, within the temporary directory (`dir_tmp`).
    xi. Make length-sensitive alignments in compressed h5 format using `bam_to_h5.R` .
    xii. Generate summary statistics, and analyses and QC plots for both RPF and mRNA datasets using `generate_stats_figs.R` . This includes estimated read counts, reads per base, and transcripts per million for each ORF in each sample.
    xiii. Write output files produced above into an sample-specific directory, named using the sample ID, within the output directory (`dir_out`).
4. Collate TPMs across results, using `collate_tpms.R` and write into output directory (`dir_out`). Only the results from successfully-processed samples are collated.
5. Count the number of reads (sequences) processed by specific stages if requested (if `count_reads: TRUE`).

https://github.com/riboviz/riboviz/blob/develop/docs/user/prep-riboviz-operation.md

*

```
11
12  # Handle interactive session behaviours or use get_Rscript_filename():
13  if (interactive()) {
14    # Use hard-coded script name and assume script is in "rscripts"
15    # directory. This assumes that interactive R is being run within
16    # the parent of rscripts/ but imposes no other constraints on
17    # where rscripts/ or its parents are located.
18    this_script <- "generate_stats_figs.R"
19    path_to_this_script <- here("rscripts", this_script)
20    source(here::here("rscripts", "provenance.R"))
21    source(here::here("rscripts", "read_count_functions.R"))
22  } else {
23    # Deduce file name and path using reflection as before.
24    this_script <- getopt::get_Rscript_filename()
25    path_to_this_script <- this_script
26    source(here::here("rscripts", "provenance.R"))
27    source(here::here("rscripts", "read_count_functions.R"))
28  }
29
```

Fails are not ALWAYS a failure!

- <3 Regression tests!
- Which output is the **correct** output? How do we know?
- How to decide when to make a **new issue** or keep working on a problem?
- **Rollback**...
- Importance of **code testing**… & understandable code!

Updates! UX

- Users:
  - Building user base?
  - Needs / wants?
  - Oven-ready datasets*?

* https://github.com/riboviz/example-datasets

Updates! UX

- Support & Documentation:
  - Existing docs suitable?
  - Document outputs better?
  - Translation UK workshop?
  - 2.0.beta → 2.0!

Updates! Testing

- General Testing:
  - New feature development ongoing
  - Bug fixes
  - Code reviews happening
  - New datasets

Updates! Testing

- Methods Testing:
  - Regression tests
  - Expected outputs: simulated data.

First Things First: Setup

- Initial Setup Code:
  - load libraries
  - handle interactive session behaviours if required*
  - source provenance & functions scripts
  - load parameters passed in from main riboviz workflow
  - read in key files (.gff, .fa, .h5)