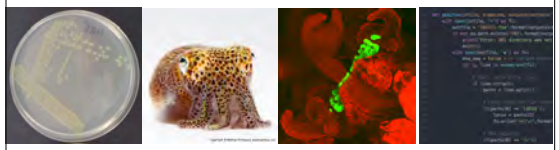
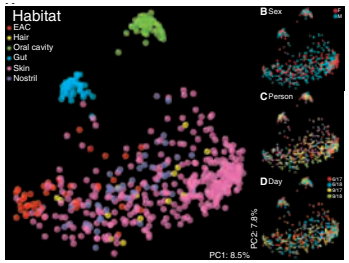
Fancy genetics and simple scripts: Manipulating DNA data and becoming more proficient with Python



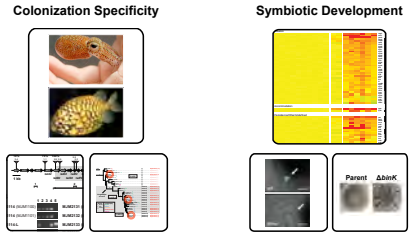
Mark Mandel
@markjmandel
Northwestern University Feinberg School of Medicine
Department of Microbiology-Immunology

Host specialization in humans



Costello 2009 Science

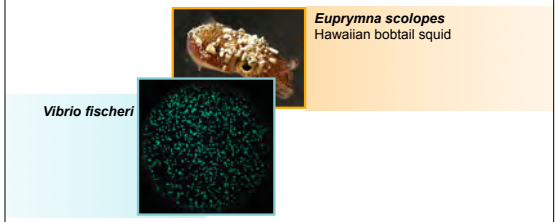
Lab Interests

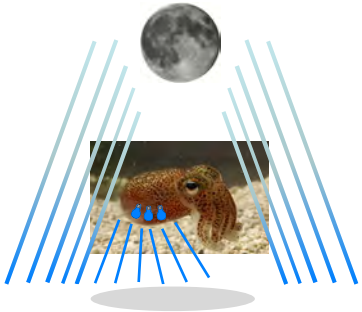


Comparative Genomics on Natural Isolates

Functional Genomics on a Model Strain

Study system





Euprymna scolopes: Hawaiian bobtail squid

Adult

Hatchlings

light organ / ink sac

Adult image: Chris Frazee

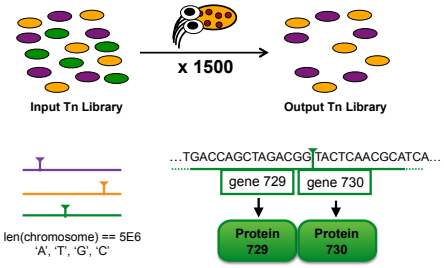
Stages of colonization

What genes (and proteins) are required for bacterial colonization?

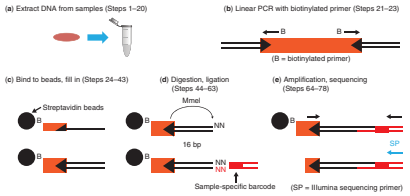
Objectives

- Establish better laboratory practices for reproducibility of data analyses and for harnessing of large data sets.
- Expand the analyses that we can perform… Stop leaving (so much) unanalyzed data on the table…
- Example: Insertion Sequencing (INSeq)

Screen for novel factors

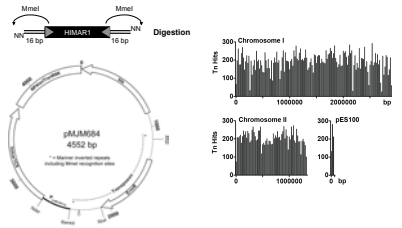


Insertion Sequencing (INSeq)



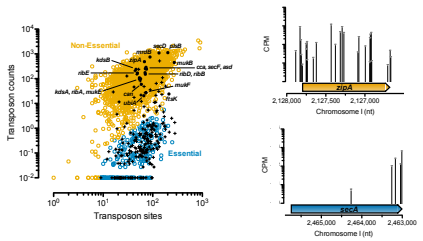
Goodman et al. 2011 Nature Protocols

Tools for INSeq in *V. fischeri*



mBio 2014, PNAS 2014

Essential genes in *V. fischeri*



PNAS 2014

What problems do I hope to address?

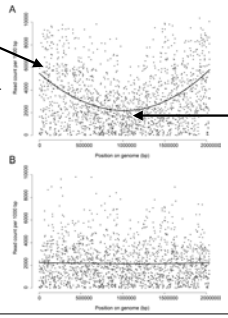
- Extend the current pipeline:
  - New transposon
  - Transposon with new functionality i.e., L != R
- Provide more robust data normalization
- Automate file format conversions
- Automate analyses
- Compare datasets robustly
- Examine down to individual genes

LOESS Normalization

DNA replication starts here (origin)

bacteria end up with multiple copies of chromosomes near the origin

DNA replication ends here (terminus)



Each spot = count of reads in 1 kb. For normalization, the authors calculated the local read density within a 100-kb window and normalized the number of reads at that position relative to the average local read density for that window.

Zomer et al. 2012 PLOS One

The pyinseq pipeline

- input files:
  - genome file = chromosomal reference sequences
  - reads file = experimental DNA sequences (190M reads)
  - sample file = DNA barcode key

Genome file (.gbk)

```
LOCUS       CP000020            2897536 bp    DNA     circular BCT 02-APR-2008
DEFINITION  Vibrio fischeri ES114 chromosome I, complete sequence.
ACCESSION   CP000020
[41 lines]
...
     gene            19150..20790
                     /gene="malS"
                     /locus_tag="VF_0017"
     CDS             19150..20790
                     /gene="malS"
                     /locus_tag="VF_0017"
[50443 lines]
...
BASE COUNT   882812 a 565193 c 563483 g 886048 t
ORIGIN
        1 aagatcactt aatatatata agatctttta aagagatctt ttattagatc tattatatag
       61 atcgtcgatc tctgtggata agtgataaat gatcaatagg atcatatact ttagatggat
      121 ccaaagttgt tatctttctt tgatcttcga tcggacagct tgaggacaaa agagttagtt
      181 atccacaagg ggggagggcg ttagatcttt ctattgcaat ggataactat aacttgatca
      241 atctttctat agttatccac atagtaggta tcatctattt aataactttt atagatcgga
[48294 lines]
```

genome.fna

```
>CP000020
aagatcacttaatatatataagatcttttaaagagatcttttattagatctattatatag
atcgtcgatctctgtggataagtgataaatgatcaatggatcatatactttagatggat
ccaaagttgttatctttctttgatcttcgatcggacagcttgaggacaaaagagttagtt
atccacaaggggggagggcgttagatctttattcaatggataactataacttgatcactgg
atctttctatagttatccacatagtaggtatcatctatttaataacttttatagatcgga
caacacttttttattaacaaatgtgttgtttttagccacaattctgctgttctttcagggat
actattttgagttacatctatctctaatagagggtacttttcgatagcgccatgctctttt
taaggtattttgaagtgattttcctgctgcgcagaaagtgtcataacttgaatcaccgat
agcaacaacagcaaattcaacgctcgataatggctgagtcacgcttttctaactgttgaat
aaatggtttaatgttgtcagggtaatcaccagcaccgtgagttgatacaacgagtaacca
taagctatcaatatcaatatcatctaaatttggttgattatgaatatcggtggaaaaatc
catttcttctaataaatcagcaagatggtcaccaacatactcagcaccacctagagtgct
tcctgtaataatagatacttttttcatgaatttatcctatataaaaatataaaaaatgggcc
tacataggcccattattaatcttattaatattggttttatttaccaatacagaatgaagt
aaatatacgtccaagtaaatcatcagaggtaaattcaccggtgatttcacttaaatgttg
```

genome.ftt

```
LOCUS       CP000020
Location    Strand      Length      PID         Gene        Synonym     Code        COG         Product
313..747    -           435         AAW84496.1  mioC        VF_0001     -           -           FMN-
binding protein MioC
817..2184   -           1368        AAW84497.1  trmE        VF_0002     -           -           tRNA
modification GTPase TrmE
2281..3906  -           1626        AAW84498.1  yidC        VF_0003     -           -
cytoplasmic insertase into membrane protein, Se
4133..4486  -           354         AAW84499.1  rnpA        VF_0004     -           -
ribonuclease P protein component
4502..4636  -           135         AAW84500.1  rpmH        VF_0005     -           -           50S
ribosomal protein L34
4822..5559  -           738         AAW84501.1  yecC        VF_0006     -           -           cystine
transport ATP-binding protein
5556..6227  -           672         AAW84502.1  yecS        VF_0007     -           -           cystine
transport system permease protein
6352..7104  -           753         AAW84503.1  fliY        VF_0008     -           -           cystine-
binding protein
7401..8810  +           1410        AAW84504.1  dnaA        VF_0009     -           -
chromosomal replication initiator protein DnaA
8843..9943  +           1101        AAW84505.1  dnaN        VF_0010     -           -           DNA
polymerase III, beta subunit
9957..11036 +           1080        AAW84506.1  recF        VF_0011     -           -
gap repair protein
```

Reads file (.fastq)

```
@DGL9ZZQ1:720:C6YD0ACXX:2:1101:1246:2185 1:N:0:
GAAGCACATTAACCTCATTTAACAGGTTGGATGATAAGTCCCCGGTCTTCG
+
CCCFFFDDHHHHHCGHHHIJDHIIJJFHHJIJJJJIJJDHIJJHIAGIJJJ
@DGL9ZZQ1:720:C6YD0ACXX:2:1101:1246:2185 1:N:0:
CTTTCGACACCGACGACGGTAACAGGTTGGATGATAAGTCCCCGGTCTTCG
+
CCCFFFDDHHHHHCGHHHIJDHIIJJFHHJIJJJJIJJDHIJJHIAGIJJJ
```

4 lines x 190,000,000 reads (760,000,000 lines in file)

```
@identifier
barcodechromosometransposon
+
quality scores
```

Sample file (.txt)

E001_01 GAAG
E001_02 CTTT

The Results…

```
$ python2.7 pyinseq.py -i reads.fastq -s samples.txt
-g genome.gb -e exp01

$ python2.7 pyinseq.py -i _exampleData/
example01.fastq -s _exampleData/example01.txt -g
_exampleData/ES114v2.gb -e example01
```

Contig	Start	End	Strand	Length	PID	Gene	Synonym	Code	COG	Product	example01:CTTT	example01:GAAG
CP000020	11054	13471	+	2418	AAW84507.1	gyrB	VF_0012	-	-	DNA gyrase, subu	100000	100000
CP000020	39622	40281	+	660	AAW84528.1	thiE	VF_0033	-	-	thiamin phosphat	100000	50000
CP000021	566145	567248	-	1104	AAW87562.1	amnK	VF_A0492	-	-	anhydro-N-acetyl	250000	300000
CP000022	3771	4268	-	498	AAW88249.1	-	VF_B0007	-	-	DNA repair protei	300000	300000

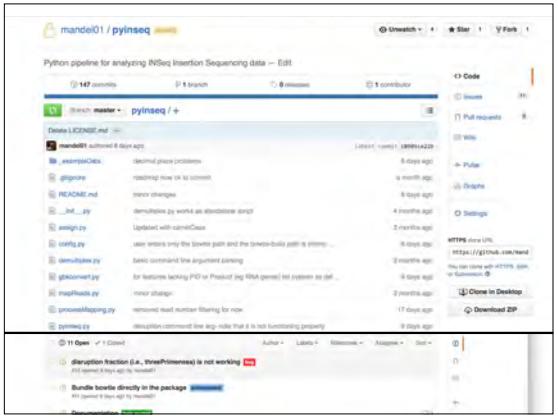
Real results would have ~ 4000 rows; one for each gene in the genome.

What have I learned in the past 6 months?

- Python :)
- Test data set (and data sets < 50,000 reads)
- generator to read in large data set
- argparse
- basic logging into the terminal

```
with open(infile, 'r') as fi:
    for line in fi:
        …
```

- combining results into an informative table
  - Previous Perl script would output 3 files/ sample x 16 samples, each of which would be run with a separate shell command, then output separate lists.
- Git/Github



Next Steps

- scaling up to work for larger files (don't pass through data so many times!)
- string formatting so it works in Python3
- modularize… then optimize and expand.

1	2	3
Prepare Reads: Demultiplex by barcode; separate into folders	**Basic INSeq mapping:** Map to genome, normalization(s), map to gene, output tables.	**Analysis:** Statistical comparisons, plotting, quality checking *Connect to other DBs*

Some of the many things I don't know (but will soon!)

- buffered reading & writing
- numpy
- pandas and dataframes
- plotting with matplotlib
- unit testing
- classes (oh, the shame…)
- And much, much more

Acknowledgments