Materials

For each plate of 96 samples, you will need the following (multiply by 2 for replicates):

blue indicates on original list from Elledge lab

Amount	Item	Supplier	Catalog #	cost per unit
2 plates	Two 96-well deep (1.1 mL), round well plates	Cole-Palmer	EW-07904-04	$156/24 pack
2ml	Protein A Dynabeads	Invitrogen	10008D	$4955/50ml
2ml	Protein G Dynabeads	Invitrogen	10009D	$4955/50ml
128 wells	Human IgG ELISA Quantitation Set	Bethyl	E80-104	$360/1000 wells
5 sheets	MicroAmp optical adhesive film	Invitrogen	4311971	$184.80/100
400	Gel loading tips	Westnet	13810	$92.59/10 trays of 96*
	bovine serum albumin	Fisher	9048-46-8	$122/100g bottle
40ml	TSBT 10x	Genesee	18-235B	$60.25/1L bottle
2	PCR plate for serum dilutions	Fisher	05408210	$92.23/25
	aluminum sealing film	Corning Axygen	14222343	$272.65/500

- Pipet tip price is for Thermo Scientific ART barrier tips, 100ul

Day 0: Quantify IgG, Block plates

Quantify IgG in serum using Human IgG ELISA Quantitation Set (Bethyl, E80-104). Dilute serum 1:100,000 (1:50, 1:200, and 1:10 dilution) before quantifying.

To simplify later processing, make aliquots of serum diluted to approximately 0.2ug/uL in PBS in a 96 well plate PCR plate. This is the 1:50 dilution of serum used for the ELISA. Seal with aluminum sealing tape and store at -80dC.

For each set of 96 samples, add 1mL of 3% BSA in TBST to each well of four 96 deep well plate. Cover with an adhesive sealing tape and using packing tape to secure the plates on a rotator. Rotate overnight at 4dC. If necessary, tap the rotator on the table as the plates are rotating to ensure that the blocking solution also coats the top of the well.

Day 1: Complex formation

Start thawing the library and samples on ice. 50mL aliquots of the library can take up to 2 hours to thaw.

Make 100mL stock of the T7 library diluted to 2e10pfu/mL (~2e5 representation) in phage extraction buffer (20 mM Tris-HCl, pH 8.0, 100mM NaCl, 6mM $MgSO_4$). Add 100uL of 50mg/mL chloramphenicol and 100uL of 50mg/mL kanamycin to inhibit bacterial growth and T7 genome replication. Mix very well.

Dump out the blocking solution from two plates. These two plates will be technical replicates. Distribute 1mL of T7 library to each well. Add sera containing 2ug of antibody to each well (4uL each from the 0.5ug/uL plate previously prepared). Blot the surface of the plates to remove any liquid. Seal the plates with MicroAmp optical adhesive film (Invitrogen, 4311971). **Make sure to seal the wells EXTREMELY well.** Press down firmly on all the spaces between wells to remove bubbles and prevent cross-contamination. Then repeatedly slide a rounded object (like a plate sealer or the cap of a VWR pen) across the tape to seal.

Secure the plates on a rotator using packing tape. Rotate at 4dC for 20h.

Day 2: Immunoprecipitation

Centrifuge sample/phage plate (250*g* for 2 min) to collect volume from seal (this is important to avoid cross contamination during removal of seal). Tightly hold down plate while removing seal to avoid any splashing.

Resuspend Protein A and Protein G Dynabeads (Invitrogen, 10008D and 10009D) by vortexing gently for 30s. Invert several times and check that no beads remain settled at the bottom. Add 4mL each of Protein A and Protein G Dynabeads to a 15mL Falcon tube and mix well. Transfer to a reagent reservoir for multichannel pipetting. Add 40uL of Protein A/Protein G Dynabeads to each well.

Secure the plates on a rotator using packing tape. Rotate at 4dC for 4h.

Place the plate on a magnetic separation rack (NEB, S1511S). Let it sit for 2 minutes to allow beads to collect, you should be able to see the solution clear up. Aspirate the liquid using gel loading tips (Westnet, 13810). Switch tips after each well to avoid cross-contamination. When aspirating, make sure to hold the plate flush against the magnetic rods to ensure the strongest magnetic pull to avoid aspirating the beads. Adjust the direction depending on where the magnetic rod sits relative to the wells (i.e., push from the top when aspirating Row A, from the bottom for Row B, etc.). After aspirating each row, add 400uL of PhIP-Seq wash buffer (50mM Tris-HCl, pH 7.5, 150mM NaCl, 0.1% NP-40) to the empty wells to prevent drying out. Remove the plate from the magnetic separation rack and use a multichannel pipettor to resuspend the beads in all the wells by pipetting up and down 10 times. Repeat the steps in this paragraph for a total of 3 washes. After resuspending after the second wash, dump out the blocking solution from the remaining plates and transfer the beads to a fresh plate.

After the third wash, aspirate the remaining liquid. Cover the plate with an adhesive seal and spin down the beads for 1m at 250*g*. Resuspend each well in 40uL of dH_2O and transfer to a PCR plate. Heat the plate to 95dC for 10m to lyse the phage. Replace seal with an aluminum seal and store at -80dC until next step.

Day 3: Library DNA preparation

PCR 1

Primers

>IS7_HsORF5_2 *Green is IS7_short_amp.P5 with Tm=64.7dC*
ACACTCTTTCCCTACACGACTCCAGTCAGGTGTGATGCTC

>IS8_HsORF3_2 *Red is IS8_short_amp.P7 with Tm = 67.6dC*
GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCCGAGCTTATCGTCGTCATCC

Protocol

Resuspend the beads and use directly as template for the PCR 1 reaction.

NEB Q5	Stock	Final	1rxns	100rxns
dH2O			2.80uL	280.00uL
Reaction Buffer	5X	1X	6.00uL	600.00uL
dNTPs	10mM	0.2mM	0.60uL	60.00uL
IS7_HsORF5_2	100uM	0.5uM	0.15uL	15.00uL
IS8_HsORF3_2	100uM	0.5uM	0.15uL	15.00uL
Q5	2U/uL	0.02U/uL	0.30uL	30.00uL
Template	2X	1X	20.00uL	
			30.00uL	10.00uL

1	98dC	0:30
2	98dC	0:05
3	66dC	0:10
4	72dC	0:30
5	Goto 2	29X
6	72dC	2:00
7	4dC	forever

Mix by inverting 10x and spinning down 3000xg for 5 min. Pop bubbles by flicking wells and then spin down again.

PCR 2

Primers

Index oligo aliquot plates are 2.5uM each. Thaw index oligo plate and spin down.

>IS4_HsORF5_2 *Overlap is green + underlined with Tm=61.2dC (69.6)*
AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACTCCAGT

>Index1 *Bold is index and red is overlap with IS8_HsORF3_2*
CAAGCAGAAGACGGCATACGAGAT**tcgcagg**GTGACTGGAGTTCAGACGTGT

Protocol

NEB Q5	Stock	Final	1rxns	200rxns				
dH2O			4.65uL	930.00uL		1	98dC	0:30
Reaction Buffer	5X	1X	2.00uL	400.00uL		2	98dC	0:05
dNTPs	10mM	0.2mM	0.20uL	40.00uL		3	68dC	0:10
IS4_HsORF5_2	100uM	0.5uM	0.05uL	10.00uL		4	72dC	0:30
INDEX_PRIMER_R	2.5uM	0.5uM	2.00uL			5	Goto 2	7X
Q5	2U/uL	0.02U/uL	0.10uL	20.00uL		6	72dC	2:00
Template	2X	1X	1.00uL			7	4dC	forever
			10.00uL	7.00uL				

Mix by inverting 10x and spinning down 3000xg for 5 min. Pop bubbles by flicking wells and then spin down again.

PCR product quantification

Serially dilute PCR2 product 1 ul into 199 ul twice for a final dilution of 40,000.

Primers

>5'NEST-qPCR
TCGGGGATCCAGGAATTC

>3'NEST-qPCR
CGTCGTCATCCTTGTAATCG

>Taqman probe_3 (5' FAM – Zen/Iowa Black)
TAATCGCGGCCGCAAGCTTGTC

TaqMan	Stock	Final	1rxns	100rxns				
dH2O			8.75uL	875uL		1x	50dC	2:00
Universal Mix	2X	1X	10uL	1000uL			95dC	10:00
3' NEST qPCR primer	100uM	0.5uM	0.1uL	10uL		40x	95dC	0:15
5' NEST qPCR primer	100uM	0.5uM	0.1uL	10uL			60dC	2:00
NEST qPCR probe	100uM	0.25uM	0.05uL	5uL				
PCR2, diluted 1:40k			1uL					
			20uL	19uL				

Mix by inverting 10x and spinning down 3000xg for 5 min. Pop bubbles by flicking wells and then spin down again.

Use FAST 7500 mode, data collection on Stage 2 Step 2.

To calculate the relative concentration:

Concentration = (0.5226/4)*EXP(-0.597*Ct).

Pool equimolar amounts of the sequencing libraries. We've found that the concentrations rarely differ by more than a factor of 2, so we just pool equal volumes. Using a multichannel pipetteman, pool 2uL of

each of the samples in a reagent reservoir, then transfer to 1.5 mL eppendorf tube. Run 40uL on a 2% TAE gel and purify the correct size band:

Library	Amplicon Size
T7-PEP2	478 bp
T7-VIR	376 bp

Submit for Sequencing

Log into the Biopolymers website (http://genome.med.harvard.edu). Click on “Next-Gen Sequencing” under the “Services” drop-down. Click on “Place and Order”

Click “New Library Pools for QC & HiSeq” under “Illumina HiSeq2000 Sequencing”.

Unless you need a full flowcell (i.e., if you are using a custom Read 2 primer), click “Independent Lanes”.

Enter the number of library pools you are sequencing. A library pool is one multiplexed pool of indexed DNA.

For each library pool, enter the requisite information. Remember to select “Yes” for “Requires Custom Sequencing Primer,” choose the right custom sequencing primer from the drop-down menus, and click “Confirm”.

Then click on “Click here to add library info” and enter in the information. “Library Type” is “Amplicon/PCR product”, “Library Category” is “DNA-Seq”, and “Adaptor Type” is “Custom Single Read Adaptor”. Remember to note if you are expecting a bias. For “Index Set Used”, select “Ben Larman 96 -7 bp” if you are using Ben’s 96 indexing oligos. If you are pooling 96 indexes, instead of manually submitting information for the 96 indexes, just choose 100% for index 1 and then make a note on the printed form and in an email to Biopolymers saying you are pooling equal volumes of all the indexes.

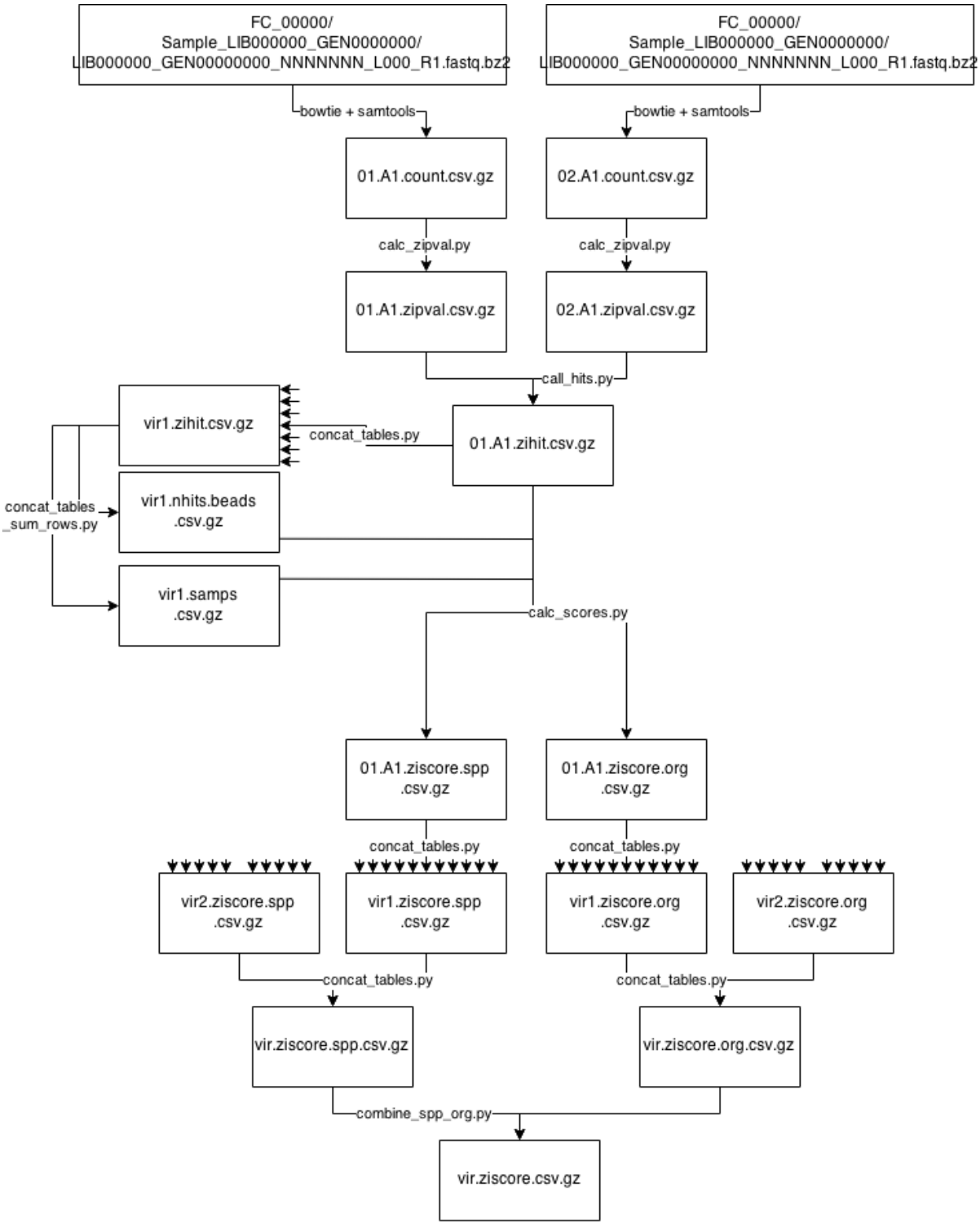
Submit the samples according to the instructions. Also submit 10uL of 100uM of the custom sequencing primer.

>T7-Pep2.2_Illumina_SP This lands 8bp upstream of library insert. (primer for previous assay – do not use - TSA)
GGTGTGATGCTCGGGGATCCAGGAATTC

>T7-Illumina-READ1-A (use this one – TSA)
TGCTCGGGGATCCAGGAATTCCGCTGCGT

Data Analysis Pipeline

Overview Flowchart



Get on Orchestra

You need an account on the HMS Orchestra cluster. Access the cluster using SSH from the Terminal on Mac OS or PuTTY on Windows.

I recommend doing everything in an interactive session

```
bsub -q interactive -W 12:00 -n 12 bash
```

If you run long, computationally intensive commands on one of the logins shells (mezannine, balcony), it will likely be terminated and you will get a warning. The alternative is to submit jobs, but I think it's easier to run interactively and unless you need to process 1000s of samples of data, it doesn't save that much time to parallelize by submitting separate jobs.

Download data to Orchestra

Follow the instructions from the Biopolymers email to download the data. Since there is limited space on personal accounts, I store my data in the shared Elledge lab drive at `/n/data2/hms/genetics/elledge/`. You may need to ask Eric to give you access.

Using the automated pipeline I set up

Fill out metadata.xslx

Add rows to the "screens" worksheet in the "metadata.xslx" file for each sample you screened.

The "bpf_fc" and "bpf_lib" are the flowcell (fc) and library (lib) identifiers from the Biopolymers Facility (bpf). The "idx" is the sequence of the barcode used for that sample. You can find the index sequences in the "96_idx" and "IDX001..096" and "IDX097..192" worksheets.

The "library" and "amp_date" are the library used and the date the library was amplified. This is necessary because each amplification generates a new input count distribution. Make sure each ("library", "amp_date") has a single row labeled "input" for the "plate".

"plate", "row", and "col" together uniquely identify a screen. Note that the column "screen" is automatically calculated based on these values.

"beads" identifies what kind of beads were used for the screen. For example, the same sample could be screened on both Protein A+G and anti-IgM beads.

"provider" and "label" together uniquely identify a sample. "Provider" refers to whom we got the sample from and "label" generally refers to the label on the tube.

Generate dependency makefiles

After updating "metadata.xlsx", upload it to `/n/data2/hms/genetics/elledge/gjx1/screens/results` and replace the old version already there.

Then run the following command to automatically generate the dependencies in the `/n/data2/hms/genetics/elledge/gjx1/screens/results/mk` folder:

```
python generate_dependencies.py
```

Run make

Now you can use the makefile (`/n/data2/hms/genetics/elledge/gjx1/screens/results/makefile`) to automatically create files.

For example, to create the count file for the 01.A1 screen (all individual files are stored in the directory `/n/data2/hms/genetics/elledge/gjx1/screens/results/parts`)

```
make parts/01.A1.count.csv.gz
```

Or to create the combined zipval file for all the vir2 screens (all combined files are stored directly in `/n/data2/hms/genetics/elledge/gjx1/screens/results/`)

```
make vir2.zipval.csv.gz
```

The makefile should automatically figure out what steps need to be run. You can also parallelize using six cores using `make -j 6`.

Documentation for each step of data analysis pipeline

Align sequencing data to reference

Use Bowtie to align the reads to a reference sequence. Bowtie should automatically be installed on Orchestra at `/opt/bowtie/`.

Upload reference sequences FASTA file

Before you can align, you need to build an index for the reference sequences. To do this, create a FASTA file containing all the sequences you want to align to. Upload this file to Orchestra using SCP from the Terminal on Mac OSX or WinSCP on Windows.

Note: if you upload from Windows, you will have to convert the line endings using the command

```
sed -i 's/\r$//' REFERENCE.FASTA
```

where REFERENCE.fasta is the name of your reference files. If you don't do this, nothing will align, I think because bowtie-build doesn't read the sequence correctly when the line endings are wrong.

Build Bowtie index

To build the bowtie index use the bowtie-build command. For example:

```
/opt/bowtie/bowtie-build REFERENCE.fasta REFERENCE_OUTPUT_NAME
```

This will create a bowtie index from the file *REFERENCE.fasta*. It will save the output as *REFERENCE_OUTPUT_NAME.1.ebwt*, *REFERENCE_OUTPUT_NAME.2.ebwt*, etc.

Align

I keep the full alignment results as BAM files, generated using samtools. Samtools should automatically be installed. I use `/opt/samtools-1.1/bin/samtools`.

Use this pipeline to decompress the bzipped sequence data, uses bowtie to align, and finally outputs a sorted compressed BAM alignment file.

```
bzip2 -dc path/to/FASTQ_SEQUENCES.fastq.bz2 | /opt/bowtie/bowtie -n 3
-l 30 -e 1000 --tryhard --nomaqround --norc --best --sam --quiet
path/to/REFERENCE_OUTPUT_NAME - | /opt/samtools-1.1/bin/samtools view
-u - | /opt/samtools-1.1/bin/samtools sort -T BAM_OUTPUT_NAME.bam - -o
$@
```

where

path/to/FASTQ_SEQUENCES.fastq.bz2 is the path to the FASTQ file containing the sequencing reads

path/to/REFERENCE_OUTPUT_NAME is the path to bowtie index (without the .1.ebwt suffix)

BAM_OUTPUT_NAME.bam is the name of the bam file to which you will save the alignment results

For additional information on the parameters, see the Bowtie manual and the Samtools manual. In particular, you can use the -5 and -3 parameters on bowtie to trim adapter sequences from the 5' or 3' ends. However, I prefer to include the adapter sequences in the reference sequence to avoid such trimming.

Count reads

To count the number of reads for each sequence, I use the samtools idxstats command. Before using idxstats, it is necessary to index the bam file by running the command:

/opt/samtools-1.1/bin/samtools index BAM_OUTPUT_NAME.bam

This will create a file *BAM_OUTPUT_NAME.bai*, which is the index of *BAM_OUTPUT_NAME.bam*.

Now use the following pipeline to count the number of reads for each reference sequence and convert it into a csv file. The first column of the csv file will be the oligo id's, taken from the sequence names in the *REFERENCE.fasta* file. The second column is the number of times that oligo was read.

```
/opt/samtools-1.1/bin/samtool idxstats BAM_OUTPUT_NAME.bam | cut -f
1,3 | sed -e '/^\*\t/d' -e '1 i id\tSAMPLE_ID' | tr "\\t" ","
>COUNT_FILE.csv
```

Where SAMPLE_ID is the id of the sample and COUNT_FILE.csv is the name of the count file.

I compress the csv files with gzip to save space. To do this, run the command:

```
gzip COUNT_FILE.csv
```

which will compress the file and rename is as *COUNT_FILE.csv.gz*

Install Python packages

The rest of the analysis requires custom Python scripts that depend on various packages. The easiest way to install them is to use the Anaconda package manager.

Download the appropriate Anaconda installer:

```
wget https://repo.continuum.io/archive/Anaconda3-2.2.0-Linux-x86_64.sh
```

Then, install Anaconda by calling

```
bash Anaconda3-2.2.0-Linux-x86_64.sh
```

Anaconda should automatically install the necessary dependencies (numpy, scipy, matplotlib) by default.

Calculate zero-inflated p-values from counts

The python script *calc_zipval.py* that will calculate zero-inflated p-values for a set of output counts based on a set of input counts. Note that the directory *calc_zipval.py* sits in must also contain the file *zigp.py*, which contains the definition of the zero-inflated p-value model.

Run the script using this command

```
python calc_zipval.py OUTPUT.count.csv.gz INPUT.count.csv.gz
log_directory >OUTPUT.zipval.csv
```

OUTPUT.count.csv is the output read count

INTPUT.count.csv is the input read count

log_directory is a directory where the script will save several plots showing model fits

OUTPUT.zipval.csv is the resulting zero-inflated p-values. First column is oligo id, second is zipvalue

Again, I gzip these csv files to save space.

Call hits from replicate zero-inflated p-values

The python script *call_hits.py* will call hits based on replicate zero-inflated p-values.

```
python call_hits.py REPLICATE1.zipval.csv.gz REPLICATE2.zipval.csv.gz
THRESHOLD log_directory >OUTPUT.zihit.csv
```

REPLICATE1.zipval.csv.gz and *REPLICATE2.zipval.csv.gz* are the two replicate zero-inflated pvalue files

THRESHOLD is the threshold zero-inflated p-value for calling hits (2.3 for VirScan)

log_directory is a directory where the script will save plots showing correlation between the replicates

OUTPUT.zihit.csv is the resulting hits. First column is oligo id, second is True/False

Again, I gzip these csv files to save space.

Calculate virus scores from hits

The python script *calc_scores.py* calculates virus scores using the maximum parsimony approach. In addition, it will filter out any hits found in at least 3 of the beads samples or only one of the serum samples.

```
python calc_scores.py ZIHITS.zihit.csv.gz METADATA.csv.gz
NHITS.BEADS.csv.gz NHITS.SAMPS.csv.gz GROUPING_LEVEL EPITOPE_LEN
>OUTPUT.ziscore.spp.csv
```

ZIHITS.zihit.csv.gz is the gzipped results of call_hits.py

METADATA.csv.gz is the file containing the metadata for the virus library

NHITS.BEADS.csv.gz is a two column gzipped csv file, column 1 is oligo id, column 2 is the number of beads samples in which that oligo was a hit

NHITS.SAMPS.csv.gz is a two column gzipped csv file, column 1 is oligo id, column 2 is the number of non-beads samples in which that oligo was a hit

GROUPING_LEVEL can be *Species* or *Organism*, depending on

EPITOPE_LEN should be 7, the length of a linear epitope

OUTPUT.ziscore.spp.csv is a two column csv file. First column is either Species or Organism, depending on *GROUPING_LEVEL*, and second column is the score.

Again, I gzip these csv files to save space.

Combine multiple csvs into one table

It's easier to work with one file containing data for all the samples rather than with 100s of files for each sample. The *concat_tables.py* script combines multiple csv files into one.

```
python concat_tables.py TABLE1.csv.gz TABLE2.csv.gz TABLE3.csv.gz …
>COMBINED.csv.gz
```

You can list as many tables as you want. The script will combine then using the first column as the index.