# **054 Penguin shotgun metagenomics - pozible project**

**illumina** account - email and Stimpy2036\*

budget = $9120

48 samples - shotgun metagenomics

2 HiSeq runs

kits = 2 x Nextera XT:

from illumina = $1314

For 250bp reads you are going to need access to a HiSeq 1500 or 2500 which both have the option to perform 250bp reads by utilising the 'rapid run' mode.

http://support.illumina.com/downloads/sequencing\_coverage\_calculator.html

Nextera XT Index Kit (24 indices, 96 samples) FC-131-1001 AND HiSeq Rapid Duo cBot loading kit CT-403-2001. With this option your 48 samples would be divided into two sets of 24 uniquely barcoded samples which need to be loaded onto separate lane with the help of the loading kit and a cBot

OR

Nextera XT Index Kit v2 Set A (96 indices, 384 samples) FC-131-2001 - index all 48 samples uniquely and run them across both lanes of the flowcell. All clustering done on board the HiSeq and no extra loading kit required.

**macrogen** tallnutt stimpy2036\*

quoted truseq nano kit 100bp pe hiseq2500 @$450usd per sample!!

Hiseq2500 100bpPE = USD2,400 (400-500M reads) or Hiseq4000 100bpPE = USD2,500 (500-600M reads);

Or Hiseq2500 250bpPE rapid mode = USD6,400 (8500 AUD)per flow cell (2 lanes) (250-350M reads per lane)

**AGRF** $8476 for 2x Hiseq 125 pe

**Ramaciotti**

Hi,

can you please give me a quote for NGS: 48 samples for shotgun metagenomics, already prepped and indexed by us. Preferably paired 125bp HiSeq (or longer).

Thanks,

Theo Allnutt

HiSeq 2500 Rapid Run:

A 2x250bp run costs $8950 and yields up to ~150Gb per run or 380M read pairs. Two of these runs would give you up to ~6.25Gb/sample on average.

A 2x150bp run costs $7125 and yields up to ~90Gb per run or 380M read pairs. Two of these runs would give you up to ~3.75Gb/sample on average.

NextSeq 500: high error rate?

A 2x150bp High Output run costs $6,955 and typically yields ~100-120Gb per run or 400M read pairs. Two of these runs would give you up to ~5Gb/sample on average.

**Garvan**

also emailed as above.

$2950 per lane HiSeq

Hi Theo,

Thanks for your enquiry. There's no simple way to determine the amount of sequencing that will be sufficient for your sample pool so I would recommend running your samples on a single lane of a high output HiSeq flow cell which will yield at least 400 million paired end reads at 2 x 125 bp (62.5 Gb). After preliminary analysis further sequencing might be required.

Sequencing the sample pool on the HiSeq 2500 in rapid run should also be considered as one rapid run flow cell has read lengths of up to 2 x 250 bp which should make de novo assembly much more robust. Expected yields in rapid run mode would be 125 - 150 Gb and up to 300 million reads per flow cell. We can provide a quote for whichever approach you choose to go with.

see pdf quote, $10,450 two lanes of Hiseq 2500 Rapid flowcell

**Samples with enough DNA:**

**LP1**

**LP2**

**LP4**

**LP6**

**LP12**

**LP13**

**LP18**

**MP8**

**MP10**

**MP11**

**MP12**

**GP1**

**GP2**

**GP3**

**GP4**

**GP5**

**GP6**

**GP7**

**GP9**

**GP11**

**GP12**

**KP1**

**KP2**

**KP3**

**KP4**

**KP5**

**KP6**

**KP8**

**KP9**

**KPC21**

**KPC24**

**KPC23**

**KPC22**

**KPC25**

**KPC27**

54.1 miniseq test

n50folder.py './\*.fastq' fastq

File Count Total\_bp Max Min N50 Mean

Theo\_1\_S18\_L001\_R1\_001.fastq 14559 2198409 151 151 151 151

Theo\_1\_S18\_L001\_R2\_001.fastq 14559 2198409 151 151 151 151

Theo\_2\_S19\_L001\_R1\_001.fastq 44567 6729617 151 151 151 151

Theo\_2\_S19\_L001\_R2\_001.fastq 44567 6729617 151 151 151 151

Theo\_3\_S20\_L001\_R1\_001.fastq 30993 4679943 151 151 151 151

Theo\_3\_S20\_L001\_R2\_001.fastq 30993 4679943 151 151 151 151

Theo\_4\_S21\_L001\_R1\_001.fastq 42021 6345171 151 151 151 151

Theo\_4\_S21\_L001\_R2\_001.fastq 42021 6345171 151 151 151 151

Theo\_5\_S22\_L001\_R1\_001.fastq 15313 2312263 151 151 151 151

Theo\_5\_S22\_L001\_R2\_001.fastq 15313 2312263 151 151 151 151

Theo\_6\_S23\_L001\_R1\_001.fastq 104710 15811210 151 151 151 151

Theo\_6\_S23\_L001\_R2\_001.fastq 104710 15811210 151 151 151 151

Theo\_7\_S24\_L001\_R1\_001.fastq 90987 13739037 151 151 151 151

Theo\_7\_S24\_L001\_R2\_001.fastq 90987 13739037 151 151 151 151

Theo\_8\_S25\_L001\_R1\_001.fastq 73684 11126284 151 151 151 151

Theo\_8\_S25\_L001\_R2\_001.fastq 73684 11126284 151 151 151 151

Theo\_9\_S26\_L001\_R1\_001.fastq 71237 10756787 151 151 151 151

Theo\_9\_S26\_L001\_R2\_001.fastq 71237 10756787 151 151 151 151

Theo\_10\_S27\_L001\_R1\_001.fastq 139898 21124598 151 151 151 151

Theo\_10\_S27\_L001\_R2\_001.fastq 139898 21124598 151 151 151 151

Theo\_11\_S28\_L001\_R1\_001.fastq 144946 21886846 151 151 151 151

Theo\_11\_S28\_L001\_R2\_001.fastq 144946 21886846 151 151 151 151

Theo\_12\_S29\_L001\_R1\_001.fastq 116995 17666245 151 151 151 151

Theo\_12\_S29\_L001\_R2\_001.fastq 116995 17666245 151 151 151 151

Theo\_13\_S30\_L001\_R1\_001.fastq 114979 17361829 151 151 151 151

Theo\_13\_S30\_L001\_R2\_001.fastq 114979 17361829 151 151 151 151

Theo\_14\_S31\_L001\_R1\_001.fastq 141765 21406515 151 151 151 151

Theo\_14\_S31\_L001\_R2\_001.fastq 141765 21406515 151 151 151 151

Theo\_15\_S32\_L001\_R1\_001.fastq 129623 19573073 151 151 151 151

Theo\_15\_S32\_L001\_R2\_001.fastq 129623 19573073 151 151 151 151

Theo\_16\_S33\_L001\_R1\_001.fastq 54444 8221044 151 151 151 151

Theo\_16\_S33\_L001\_R2\_001.fastq 54444 8221044 151 151 151 151

trimmo.py 'miniseq/\*.fastq' 24 ~/db/trimmoall.fa p y

clip/Theo\_1\_S18\_L001\_R1\_001.fastq:9906

clip/Theo\_1\_S18\_L001\_R2\_001.fastq:9906

clip/Theo\_2\_S19\_L001\_R1\_001.fastq:31724

clip/Theo\_2\_S19\_L001\_R2\_001.fastq:31724

clip/Theo\_3\_S20\_L001\_R1\_001.fastq:23361

clip/Theo\_3\_S20\_L001\_R2\_001.fastq:23361

clip/Theo\_4\_S21\_L001\_R1\_001.fastq:16614

clip/Theo\_4\_S21\_L001\_R2\_001.fastq:16614

clip/Theo\_5\_S22\_L001\_R1\_001.fastq:9541

clip/Theo\_5\_S22\_L001\_R2\_001.fastq:9541

clip/Theo\_6\_S23\_L001\_R1\_001.fastq:69133

clip/Theo\_6\_S23\_L001\_R2\_001.fastq:69133

clip/Theo\_7\_S24\_L001\_R1\_001.fastq:60414

clip/Theo\_7\_S24\_L001\_R2\_001.fastq:60414

clip/Theo\_8\_S25\_L001\_R1\_001.fastq:49648

clip/Theo\_8\_S25\_L001\_R2\_001.fastq:49648

clip/Theo\_9\_S26\_L001\_R1\_001.fastq:38243

clip/Theo\_9\_S26\_L001\_R2\_001.fastq:38243

clip/Theo\_10\_S27\_L001\_R1\_001.fastq:68666

clip/Theo\_10\_S27\_L001\_R2\_001.fastq:68666

clip/Theo\_11\_S28\_L001\_R1\_001.fastq:83128

clip/Theo\_11\_S28\_L001\_R2\_001.fastq:83128

clip/Theo\_12\_S29\_L001\_R1\_001.fastq:67904

clip/Theo\_12\_S29\_L001\_R2\_001.fastq:67904

clip/Theo\_13\_S30\_L001\_R1\_001.fastq:75668

clip/Theo\_13\_S30\_L001\_R2\_001.fastq:75668

clip/Theo\_14\_S31\_L001\_R1\_001.fastq:81225

clip/Theo\_14\_S31\_L001\_R2\_001.fastq:81225

clip/Theo\_15\_S32\_L001\_R1\_001.fastq:77519

clip/Theo\_15\_S32\_L001\_R2\_001.fastq:77519

clip/Theo\_16\_S33\_L001\_R1\_001.fastq:37926

clip/Theo\_16\_S33\_L001\_R2\_001.fastq:37926

commands.sh to interleave

metaphlan2folder.py 'int/\*' metaphlan2/ a

meta\_otu\_table3.py metaphlan2 meta\_ txt "|" ".fastq" 0

all ok.

**54.2 Novaseq data**

catenate\_files.py 'reads/\*' cat/ "\_S"

try using bbmap reformat.sh need interleaved fasta for metaphlan2.. nb does not do adaptor trimming, so use fastp

fastp -i reads/Theo\_17\_R1.fastq -I reads/Theo\_17\_R2.fastq -o clip/Theo\_17\_R1.fastq -O clip/Theo\_17\_R2.fastq -g -l 100 -w 8 &

fastp -i reads/Theo\_18\_R1.fastq -I reads/Theo\_18\_R2.fastq -o clip/Theo\_18\_R1.fastq -O clip/Theo\_18\_R2.fastq -g -l 100 -w 8

fastp.sh

reformat.sh in=clip/Theo\_17\_R1.fastq in2=clip/Theo\_17\_R2.fastq out=fasta/17.fasta

reform.sh

10G of rads per sample is overkill.. subsample 5M reads each. (~10%)

for i in {1..16}

do

usearch10 -fastx\_subsample fasta/$i.fasta -sample\_size 5000000 -fastaout subsample/$i.fasta &

done

metaphlan2folder.py 'subsample/\*' metaphlan2/ a multifasta

meta\_otu\_table3.py 'metaphlan2/\*' meta\_ "|" 0 count

**5M reads may not be enough.. very low diversity observed**

**54.3**

**try a metaphlan rarefaction on diversity to test effect of number of reads.**

use 2.fasta.. = 57 767 162 reads

for i in {1000000..55000000..5000000}

do

usearch10 -fastx\_subsample ../fasta/2.fasta -sample\_size $i -fastaout rare/$i.fasta &

done

metaphlan2folder.py 'rare/\*' metarare/ multifasta

meta\_otu\_table3.py 'metarare/\*' metarare\_ "|" 0 count

**did not see any change in diversity with more reads, but slight increase in No. species.**

**54.4 repeat metaphlan with ~~20M~~ reads.**

~~for i in \*.fasta~~

~~do~~

~~usearch10 -fastx\_subsample $i -sample\_size 20000000 -fastaout ../fasta20M/$i &~~

~~done~~

~~metaphlan2folder.py '../fasta20M/\*' ./ multifasta~~

~~meta\_otu\_table3.py './\*.txt' meta20m "|" 0 freq~~

n.b. used 5M reads after bbmap filtering of penguin dna

otu\_habitat.py genus.tab ~/d/habitat/species-100hab.txt genus\_habitat.tab genus

**anova**

biom convert -i meta5m\_count\_s.tab -o meta5m\_count\_s.biom --to-json

biom convert -i meta5m\_count\_g.tab -o meta5m\_count\_g.biom --to-json

biom convert -i meta5m\_count\_f.tab -o meta5m\_count\_f.biom --to-json

biom convert -i meta5m\_count\_o.tab -o meta5m\_count\_o.biom --to-json

biom convert -i meta5m\_count\_c.tab -o meta5m\_count\_c.biom --to-json

biom convert -i meta5m\_count\_p.tab -o meta5m\_count\_p.biom --to-json

group\_significance.py -i meta5m\_count\_s.biom -m mapping.txt -c Description -o s-anova.txt

group\_significance.py -i meta5m\_count\_g.biom -m mapping.txt -c Description -o g-anova.txt

group\_significance.py -i meta5m\_count\_f.biom -m mapping.txt -c Description -o f-anova.txt

group\_significance.py -i meta5m\_count\_o.biom -m mapping.txt -c Description -o o-anova.txt

group\_significance.py -i meta5m\_count\_c.biom -m mapping.txt -c Description -o c-anova.txt

group\_significance.py -i meta5m\_count\_p.biom -m mapping.txt -c Description -o p-anova.txt

repeat with two groups:

group\_significance.py -i meta5m\_count\_s.biom -m mapping.txt -c Type -o s-2anova.txt

group\_significance.py -i meta5m\_count\_g.biom -m mapping.txt -c Type -o g-2anova.txt

group\_significance.py -i meta5m\_count\_f.biom -m mapping.txt -c Type -o f-2anova.txt

group\_significance.py -i meta5m\_count\_o.biom -m mapping.txt -c Type -o o-2anova.txt

group\_significance.py -i meta5m\_count\_c.biom -m mapping.txt -c Type -o c-2anova.txt

group\_significance.py -i meta5m\_count\_p.biom -m mapping.txt -c Type -o p-2anova.txt

**Diversity**

-run on genus only

alpha\_rarefaction.py -i meta5m\_count\_g.biom -m mapping.txt -o rarefaction2 -f -p qiime\_parameters1.txt -n 100 --min\_rare\_depth 100 --max\_rare\_depth 10000

compare\_alpha\_diversity.py -i rarefaction2/alpha\_div\_collated/simpson.txt -m mapping.txt -c Type,Description -o compare\_simpson -t nonparametric -d 2971

compare\_alpha\_diversity.py -i rarefaction2/alpha\_div\_collated/shannon.txt -m mapping.txt -c Type,Description -o compare\_shannon -t nonparametric -d 2971

compare\_alpha\_diversity.py -i rarefaction2/alpha\_div\_collated/observed\_species.txt -m mapping.txt -c Type,Description -o compare\_obs\_species -t nonparametric -d 2971

**pco**

to make pco need to calc unifrac.. but to do that need 16S tree of genera.. use tax2\_16s.py

sed -i 's/\[//g' genus16s.txt

sed -i 's/\]//g' genus16s.txt

sed -i 's/ c\_\_;//g' genus16s.txt

sed -i 's/ o\_\_;//g' genus16s.txt

sed -i 's/ f\_\_;//g' genus16s.txt

sed -i 's/ g\_\_;//g' genus16s.txt

sed -i ':a;N;$!ba;s/ s\_\_\n/\n/g' genus16s.txt

sed -i ':a;N;$!ba;s/;\n/\n/g' genus16s.txt

sed -i 's/|/; /g' genus16s.txt

delete viruses and eukaryota

tax2\_16s.py genus16s.txt ~/db/gg\_13\_5.1.fasta ~/db/gg\_13\_5\_taxonomy.txt genus16s.fasta

rename genus16s.fasta with otu simple names - in excel

nb changed ; to | in output

remove non bacteria from otu table - rename otus with simple name, not taxonomy

meta\_otu\_table3.py 'metaphlan/\*.txt' ./meta5m\_prop "|" 0 freq

genus\_bact.tab

biom convert -i genus\_bact.tab -o genus\_bact.biom --to-json --table-type 'OTU table'

muscle -in genus16s.fasta -out genus16s.aln

FastTree -nt genus16s.aln > genus16s.tree

beta\_diversity.py -i genus\_bact.biom -m weighted\_unifrac -o ./ -t genus16s.tree

principal\_coordinates.py -i weighted\_unifrac\_genus\_bact.txt -o pco.out

**54.5 try assembly and maxbin**

idba.sh

idba\_ud -r ../fasta/1.fasta -o idba\_1 --num\_threads 44 --mink 31 --maxk 71 --step 20 --min\_contig 300 --min\_pairs 2

also try tadpole

tadpole.sh in=../fasta/2.fasta out=tad2.fa k=31 merge=t overwrite=t -Xmx100g

tad.sh

tadpole assemblies oddly uneven.. 1-3 ok then far too few contigs. --mostly fixed by lower kmer - even though read stats look similar - dues ot memory fail... set xmx

kmer too big use default 31

for i in {1..16}

do

usearch10 -fastx\_subsample subsample/$i.fasta -sample\_size 1000000 -fastaout subsample2/$i.fasta &

done

~/bin/MaxBin-2.2.4/run\_MaxBin.pl -thread 8 -contig idba/17.fa -out maxbin/17 -reads human\_\_unmapped/17.fasta -min\_contig\_length 150

maxbin.py '54.5maxbin/\*.fa' 'fasta/\*.fasta' 300

**maxbin is not working.. will not find bins due to not finding marker genes... try running on older files**

**Try harder to dig out marker genes from contigs.**

**Marker gene search reveals that the dataset cannot be binned (the medium of marker gene number <= 1). Program stop.**

~/bin/MaxBin-2.2.4/run\_MaxBin.pl -thread 20 -contig ~/d/056\_csiro\_metagenomics/idba/MC8.fa -out ./maxtest -reads ~/d/056\_csiro\_metagenomics/int/fasta/MC8.fasta -min\_contig\_length 1500

**...worked ok.**

**try GOBOND on contigs**

GOBOND -i gobond.txt -o gobond/

fails

**superfocus??**

canu?

Reads may have significant penguin contamination - manual blast shows some bird sequences. Therefore filter against penguin ref??

LP=Eudyptula minor

MP=Eudyptes chrysolophus

GP=Pygoscelis papua

KP=Aptenodytes patagonicus

-no assemblies

Aptenodytes\_forsteri - emporer in refseq

duk.sh

**bbduk.sh -Xmx128g in=fasta/1.fasta out=bbduk\_unmatch/1.fa outm=bbduk\_match/1.fa ref=emporer\_genomic.fna.gz k=31 hdist=1 stats=1stats.txt**

bbduk just seems very very slow and uses masses of memory.. try using bbmap instead.

bbmap\_general\_fast.py 'fasta/\*.fasta' emporer\_genomic.fna.gz ./ 1 emporer\_ f

bbmap\_general\_fast.py 'fasta/6.fasta' emporer\_genomic.fna.gz ./ 1 emporer6rpt\_ f

n.b. f = use disk for ref.

Up to 90% penguin... leaving minimum of 5M reads.. therfore rarefy to 5M

~~for i in {1..16}~~

~~do~~

~~usearch10 -fastx\_subsample emporer\_\_unmapped/$i.unmapped.fasta -sample\_size 5000000 -fastaout rare5m/$i.fasta &~~

~~done~~

**re-run idba and maxbin**

metaphlan2folder.py 'rare5m/\*.fasta' metaphlan/ multifasta

meta\_otu\_table3.py 'metaphlan/\*.txt' meta5m\_freq "|" 0 freq

meta\_otu\_table3.py 'metaphlan/\*.txt' meta5m\_count "|" 0 count

otu\_habitat.py genus.tab ~/d/habitat/species-100hab.txt genus\_habitat.tab genus

idba.sh

bioinf giving segmentation ram faults with idba.. try metabox

n.b. the rarefied data is not paired.. arghh

re-do the resampling

~~reformat.sh in=emporer\_\_unmapped/1.unmapped.fasta verifypairing~~

~~for i in {1..16}~~

~~do~~

~~reformat.sh in=emporer\_\_unmapped/$i.unmapped.fasta out1=paired\_unmapped/$i.unmapped\_R1\_.fasta out2=paired\_unmapped/$i.unmapped\_R2\_.fasta~~

~~done~~

~~for i in {1..16}~~

~~do~~

~~usearch10 -fastx\_subsample paired\_unmapped/$i.unmapped\_R1\_.fasta -reverse paired\_unmapped/$i.unmapped\_R2\_.fasta -sample\_size 5000000 -fastaout paired5m/$i.R1.fasta -output2 paired5m/$i.R2.fasta~~

~~done~~

~~for i in {1..16}~~

~~do~~

~~reformat.sh in1=paired5m/$i.R1.fasta in2=paired5m/$i.R2.fasta out1=rare5m/$i.fasta~~

~~done~~

for i in {1..16}

do

fastasample.py emporer\_\_unmapped/$i.unmapped.fasta rare5m/$i.fasta 5000000 &

done

idba.sh

test maxbin

~/bin/MaxBin-2.2.4/run\_MaxBin.pl -thread 12 -contig idba/1scaffold.fa -out maxbintest/1 -reads rare5m/1.fasta -min\_contig\_length 150

n.b. some of the reads are homopolymers, therefore future trimming, use ~30mers in adaptor file to clip them.

idba contigs good.. run maxbin

mkdir maxbin

maxbin.py 'idba/\*.fa' 'rare5m/\*.fasta' 150

chopfasta.py 'maxbin/\*.fasta' chop/ 300

metaphlan2folder.py 'chop/\*.fasta' max-metaphlan/ multifasta

meta\_otu\_table3.py 'max-metaphlan/\*.txt' max-metaphlan\_count "|" 0 count

blast unknown bins - 98 are unknown to metaphlan

get\_first\_contig.py "unk/\*.fasta" unk\_contig1/

blastfolderdb.py -i 'unk\_contig1/rpt/\*.fasta' -o blast/ -e 1e-10 -t remote --hsps 1 -m 1 --threads 10

lots of feckin blasts fail due to server overload.. rewrite sccript to repeat empty blast result files

also try all seqs in one file..

add\_filename\_to\_lines.py 'blast/\*.blast' blast\_rename txt

add\_filename\_to\_lines.py 'unk\_contig1/rpt/\*.fasta' contig\_rename fasta

cat contig\_rename/\*.fasta > tmp.fasta

blastn -task blastn -remote -query tmp.fasta -db nt -out ./tmp.blast -outfmt '6 qseqid sseqid pident length slen qstart qend sstart send evalue bitscore stitle' -evalue 1e-10 -max\_hsps 1 -max\_target\_seqs 1

cat blast\_rename/\*.blast >> unk.blast

add repeated blasts to unk.blast then annotate maxbin spreadsheet results with unknowns.

unknowns are a mixture of some penguin and some bacteria

**Attempt assembly of penguin**

KP 13-16 has highest number of penguin reads and 90% penguin..

Need to maxbin f and r reads then try to reconcile

pool 13-14 fasta

idba

maxbin

i.d. penguin bins with blast vs. ref

reassemble

cat fasta/13.fasta fasta/14.fasta fasta/15.fasta fasta/16.fasta > kp\_assembly/kp.fasta

idba\_ud -r kp\_assembly/kp.fasta -o kp\_assembly/idba\_kp --num\_threads 8 --mink 50 --maxk 150 --step 50 --min\_contig 1000 --min\_pairs 2

**contigs: 96968 n50: 24210 max: 275148 mean: 12374 total length: 1199921417 n80: 10152**

maxbin.py 'kp\_assembly/kp.fasta' kp\_assembly/scaffold.fa 1000

align emporer to assembly?

filter both to >5000bp

File Count Total\_bp Max Min N50 Mean

emp100k.fa 481 1240957849 28260285 100418 5075933 2579953

kp100k.fa 376 47550394 275148 100260 120699 126463

kp5k.fa 57175 1108683648 275148 5001 26491 19391

kp\_scaffold.fa 96968 1199921417 275148 150 24210 12374

**Attempt to filter non-penguin contigs from assembly using metphlan species list as DB**

cd ~/db/bacteria\_and\_viruses

https://www.ncbi.nlm.nih.gov/genome/doc/ftpfaq/#allcomplete

see virus\_download.xlsx

..or why not just use all bacteria and viruses?

cat ~/db/RefSeq/Bacteria/\*/\*.fna > bacteria.fna

cat ~/db/RefSeq/viruses/\*.fna > viruses.fna

cat ~/db/bacteria\_and\_viruses/viruses.fna ~/db/bacteria\_and\_viruses/bacteria.fna > ~/db/bacteria\_and\_viruses/v\_b.fna

~~makeblastdb -in bacteria.fna -out bacteria -parse\_seqids -dbtype nucl &~~

~~makeblastdb -in viruses.fna -out viruses -parse\_seqids -dbtype nucl~~ &

Filter reads..

bbmap\_general\_fast.py kp.fasta ~/db/bacteria\_and\_viruses/v\_b.fna bbfilt 1 bbfilt 0.75 t

#then assemble again. using unmapped reads

idba\_ud -r bbfilt/bbfilt\_unmapped/kp.unmapped.fasta -o idba\_kp --num\_threads 40 --mink 50 --maxk 150 --step 50 --min\_contig 1000 --min\_pairs 2

**contigs: 97007 n50: 24208 max: 275148 mean: 12361 total length: 1199159149 n80: 10152**

blast biggest contig.

get\_first\_contig.py './kpfilt.scaffold.fa' ./

**...try bbmap for taxonomy..**

to make more accurate need only one of each species in db...

see bact\_list.xlsx

also look at adding fungi

**cd ~/db/RefSeq**

make\_bbmeta\_db.sh

~~cat viruses/\* >> bbmeta.fna~~

bbmap\_general\_fast.py 'fasta/\*.fasta' ~/db/RefSeq/bbmeta.fna bbmeta 1 bbmeta 1.0 f

maps2tbl.py 'bbmeta\_mapped/\*.map' bbmeta.tab

works well, but virus phages etc. will interfere with best mapping to bacteria.. remove phage from viruses.

**fungi database**

size of refseq? -could do this no probs

**try diamond? no = protein aligner or translated dna**

~~make test files..~~

~~head -1000000 bbmeta.fna > reftest.fna~~

~~from Bio import SeqIO~~

~~f=open('bbmeta.fna','r')~~

~~g=open('test.fna','w')~~

~~c=0~~

~~for i in SeqIO.parse(f,'fasta'):~~

~~c=c+1~~

~~print c~~

~~if c<100:~~

~~SeqIO.write(i[0:100],g,'fasta')~~

~~else:~~

~~break~~

~~diamond makedb --in reftest.fna --db reftest~~

**MetaWatt binner and taxonomy**

java -jar ~/bin/MetaWatt-3.5.3/dist/MetaWatt-3.5.3.jar --run fasta/ --threads 12 --temp-folder ./temp

--massive pain in the arse to use!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

**Superfocus**

superfocus\_\_downloadDB.py diamond

superfocus.py -q fasta/ -dir superfocus/ -t 8 -a diamond

biom convert -i seed.tab -o seed.biom --to-json --table-type 'OTU table'

group\_significance.py -i seed.biom -m mapping.txt -c Type -o seed\_type.anova.txt

group\_significance.py -i seed.biom -m mapping.txt -c Description -o seed\_desc.anova.txt

biom convert -i level1.tab -o level1.biom --to-json --table-type 'OTU table'

group\_significance.py -i level1.biom -m mapping.txt -c Type -o level1\_type.anova.txt

group\_significance.py -i level1.biom -m mapping.txt -c Description -o level1\_desc.anova.txt

biom convert -i level2.tab -o level2.biom --to-json --table-type 'OTU table'

group\_significance.py -i level2.biom -m mapping.txt -c Type -o level2\_type.anova.txt

group\_significance.py -i level2.biom -m mapping.txt -c Description -o level2\_desc.anova.txt

biom convert -i level3.tab -o level3.biom --to-json --table-type 'OTU table'

group\_significance.py -i level3.biom -m mapping.txt -c Type -o level3\_type.anova.txt

group\_significance.py -i level3.biom -m mapping.txt -c Description -o level3\_desc.anova.txt

**54.4 mapping to viruses and assembly**

Murine\_osteosarcoma\_virus

Avian\_myelocytomatosis\_virus

Avian\_carcinoma\_virus

bbmap map\_a\_carc

bbmap\_general\_fast.py "/mnt/sdc1/054\_penguin\_metagenomics\_pozzible/fasta/\*.fasta" a\_carcinoma.fasta map\_a\_carc 1.5 0.8 t

bbmap\_general\_fast.py "/mnt/sdc1/054\_penguin\_metagenomics\_pozzible/fasta/\*.fasta" a\_myelo.fasta map\_a\_myelo 1.5 0.8 t

bbmap\_general\_fast.py "/mnt/sdc1/054\_penguin\_metagenomics\_pozzible/fasta/\*.fasta" m\_osteo.fasta map\_m\_osteo 1.5 0.8 t

delete all unmapped files

catenate all mapped reads

cat map\_a\_carc/mapped/\*.fasta > a\_carc\_mapped.fasta

cat map\_a\_myelo/mapped/\*.fasta > a\_myelo\_mapped.fasta

cat map\_m\_osteo/mapped/\*.fasta > m\_osteo\_mapped.fasta

**m. osteosarcoma, only two reads mapped - can't assemble or do much more with that virus**

idba\_ud -r a\_carc\_mapped.fasta -o idba\_acarc --num\_threads 4 --mink 51 --maxk 150 --step 10 --min\_contig 300 --min\_pairs 2

**contigs: 4 n50: 1157 max: 1477 mean: 913 total length: 3655 n80: 521**

idba\_ud -r a\_myelo\_mapped.fasta -o idba\_amyelo --num\_threads 4 --mink 51 --maxk 150 --step 10 --min\_contig 300 --min\_pairs 2

**contigs: 2 n50: 1539 max: 1539 mean: 1422 total length: 2845 n80: 1306**

**54.10 repeat the Metaphlan analysis – make sure counts are use on 5M reads.**

metaphlan2folder.py "fasta5m/\*" metaphlan/ fasta "" 12 rel\_ab\_w\_read\_stats

meta\_otu\_table3.py 'metaphlan/\*' metaphlan2\_ "|" 2 100 count