UCDAVIS GENOME CENTER

MICROBIAL COMMUNITY PROFILING

MATT SETTLES, PHD
UNIVERSITY OF CALIFORNIA, DAVIS
SETTLES@UCDAVIS.EDU

BIOINFORMATICS.UCDAVIS.EDU

Bioinformatics Core

Genome Center

UC Davis

DISCLAIMER

- This talk/workshop is full of opinion, there are as many different way to perform analysis as there are Bioinformaticians.
- My opinion is based on over a decade of experience and spending a considerable amount of time to understand the data and how it relates to the biological question.
- Each experiment is unique, this workshop is a starting place and should be adapted to the specific characteristics of your experiment.

OUTLINE

1. Data Science
2. Introduction to Sequencing
3. Experimental Design: community analysis
4. Workshop example data
5. The command line
6. dbcAmplicons: Generating Amplicons
7. dbcAmplicons: Python Applicatoin

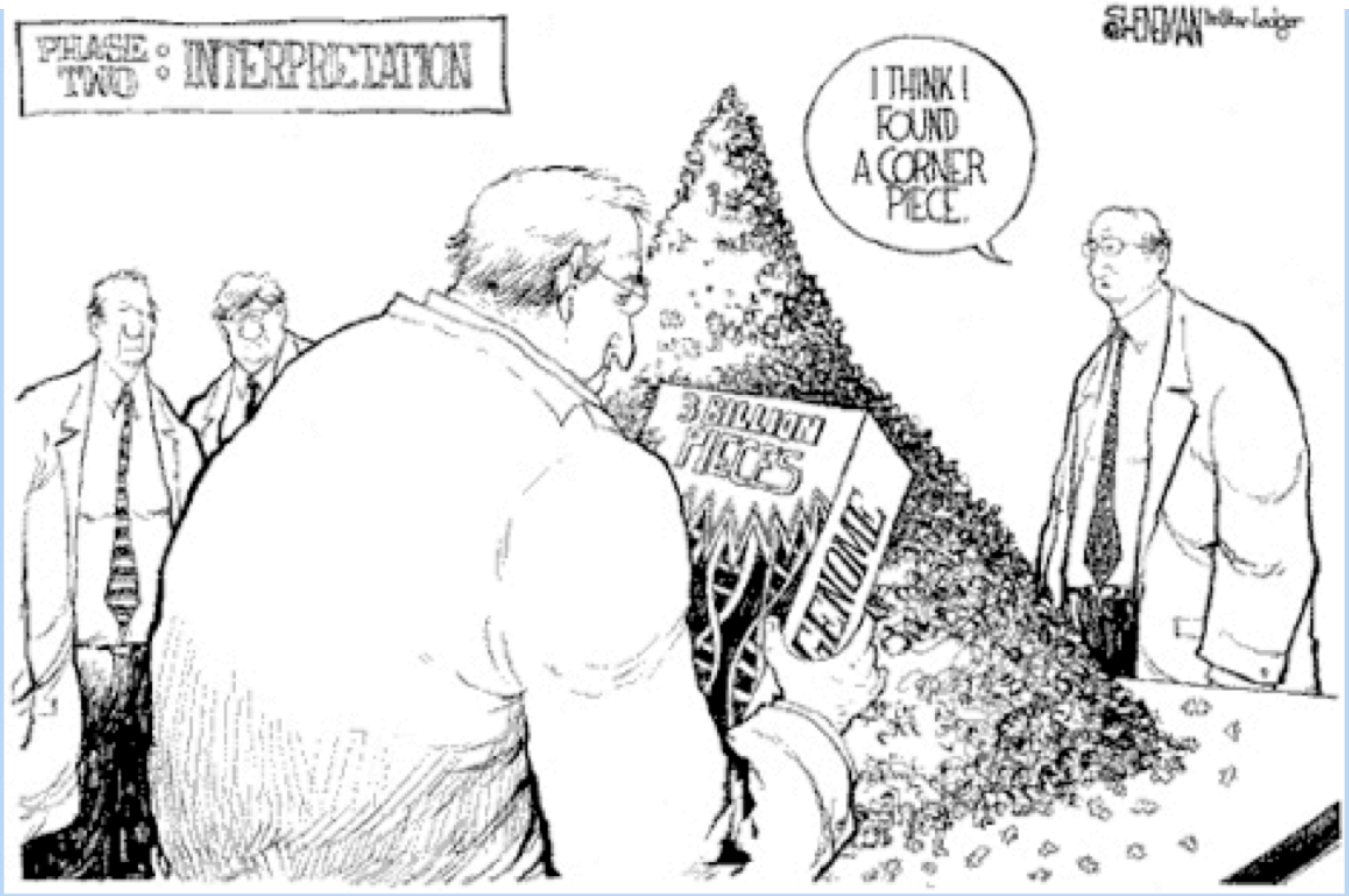
BIOINFORMATICS IS A DATA SCIENCE

Section

THE DATA DELUGE

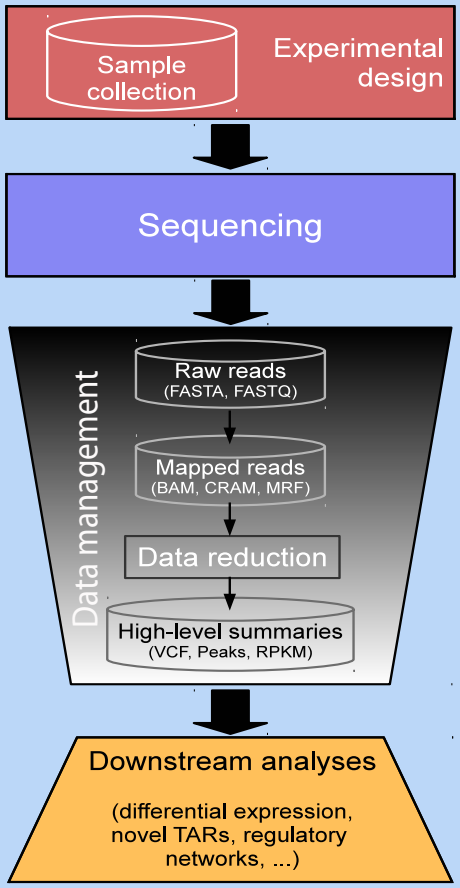
- Plucking the biology from the Noise

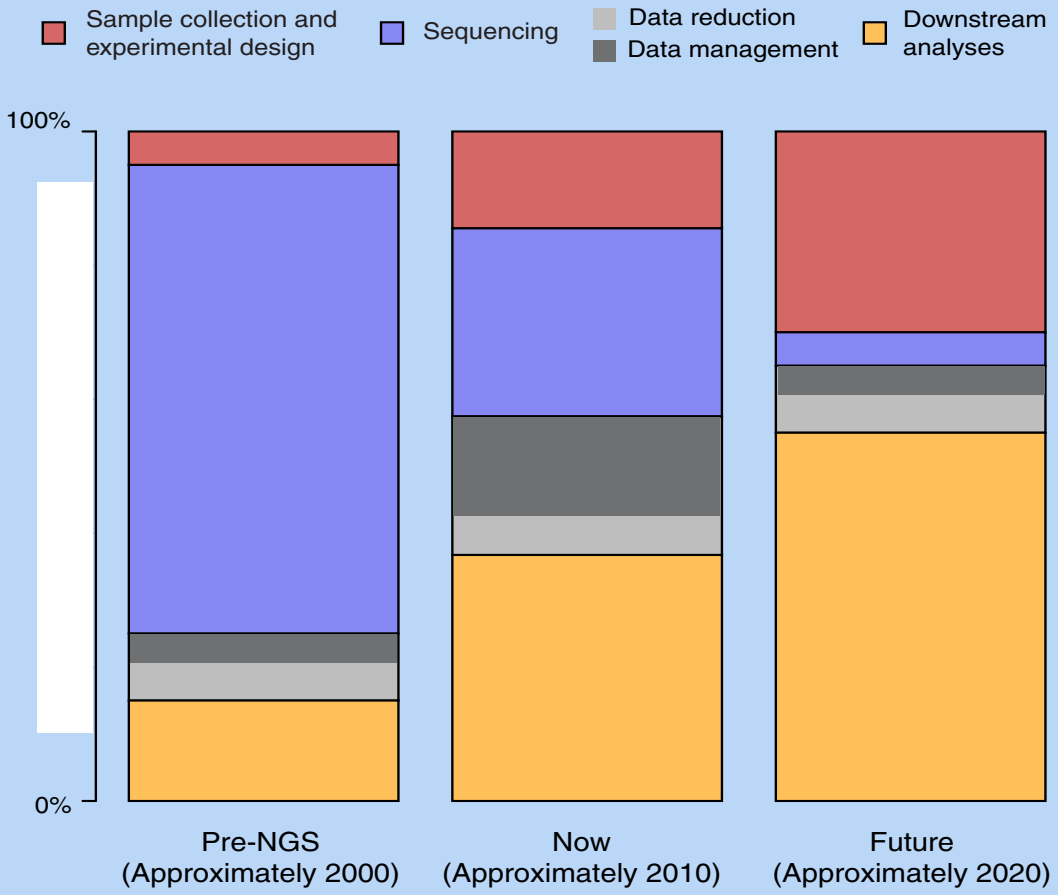
REALITY



- Its much more difficult than we may first think

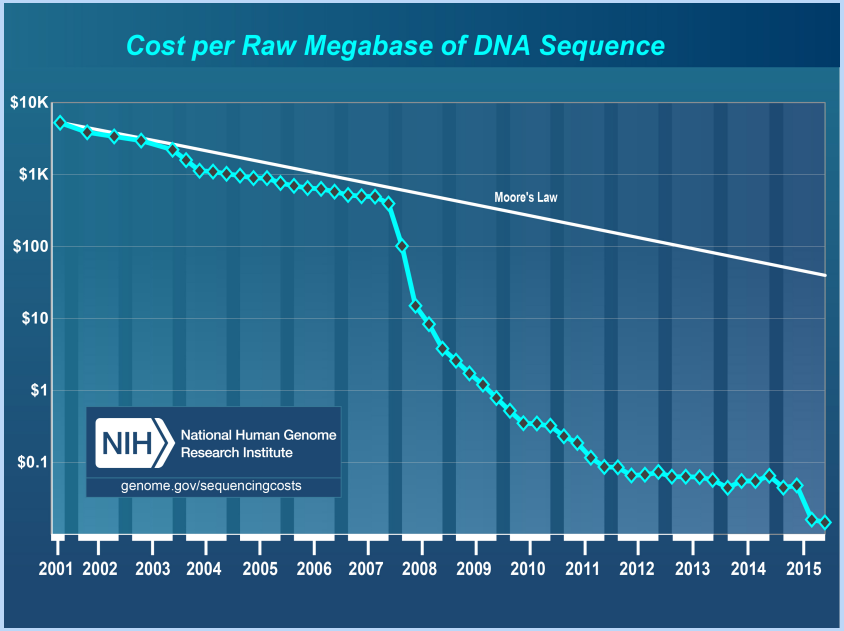
THE REAL COST OF SEQUENCING



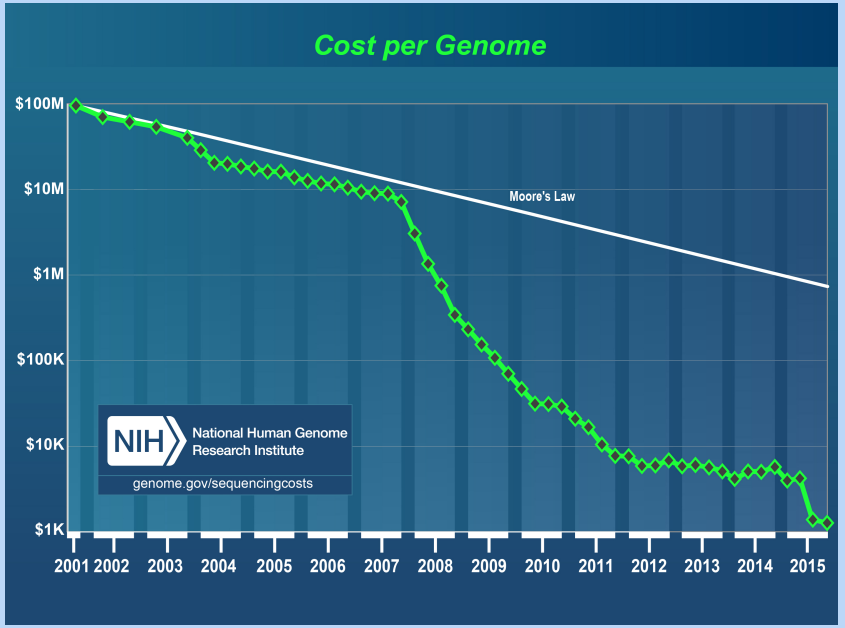


The real cost of sequencing: higher than you think!
Sboner et al. Genome Biology 2011 12:125 doi:10.1186/gb-2011-12-8-125 (2011)

SEQUENCING COSTS



GROWTH IN SEQUENCING

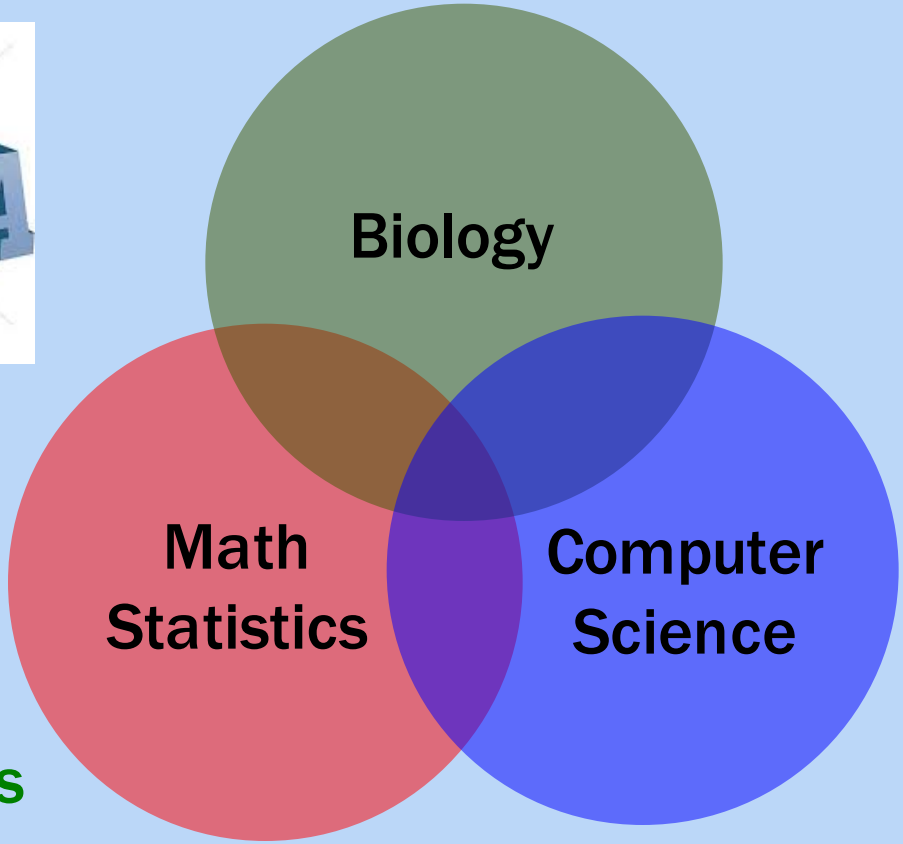


- October – 2015
  - $0.014 per Megabase
  - $1,245 per Human Sized Genome (30x coverage)

Data Only!!
Does not include any analysis or bioinformatics

BIOINFORMATICS IS DATA SCIENCE

Computational Biology



Biostatistics

Bioinformatics

'The data scientist role has been described as "part analyst, part artist."'
Anjul Bhambhri, vice president of big data products at IBM

DATA SCIENCE

Data science is the process of formulating a quantitative question that can be answered with data, collecting and cleaning the data, analyzing the data, and communicating the answer to the question to a relevant audience.

Five Fundamental Concepts of Data Science
statisticsviews.com November 11, 2013 by Kirk Borne

EXPERIMENT PHILOSOPHY

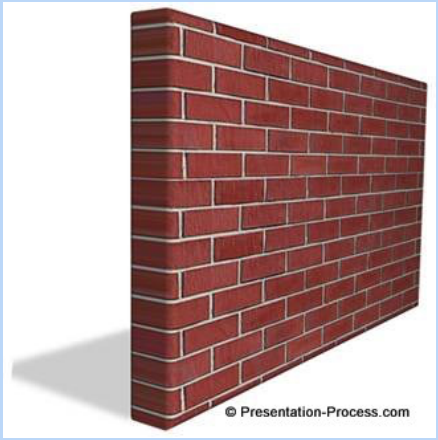
Design Experiment

Perform Experiment

Sample and Extract RNA/DNA

Prepare Libraries

Sequence



Analyze

Interpret

7 STAGES TO DATA SCIENCE

1. Define the question of interest

2. Get the data
3. Clean the data
4. Explore the data

5. Fit statistical models

6. Communicate the results
7. Make your analysis reproducible

STAGE 1: DEFINE THE QUESTION

1. Define the question of interest

Begin with the end in mind!

what is the question
how are we to know we are successful
what are our expectations

dictates

the data that should be collected
the features being analyzed
which algorithms should be use

STAGES 2-4: COLLECTING AND CLEANING

2. Get the data
3. Clean the data
4. Explore the data

Know your data!

know what the source was
technical processing in producing data (bias, artifacts, etc.)
"Data Profiling"

Data are never perfect but love your data anyway!

the collection of massive data sets often leads to unusual , surprising, unexpected and even outrageous.

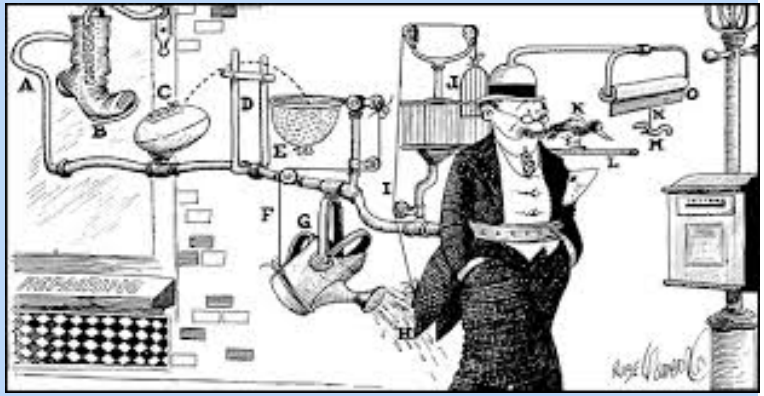
STAGE 5: ANALYZING THE DATA

5. Fit statistical models

Over fitting is a sin against data science!

Model's should not be over-complicated

- If the data scientist has done their job correctly the statistical models don't need to be incredibly complicated to identify important relationships
- In fact, if a complicated statistical model seems necessary, it often means that you don't have the right data to answer the question you really want to answer.



STAGES 6-7: COMMUNICATING YOUR RESULTS

6. Communicate the results
7. Make your analysis reproducible

Remember that this is 'science'!

We are experimenting with data selections, processing, algorithms, ensembles of algorithms, measurements, models. At some point these ***must all be tested for validity and applicability*** to the problem you are trying to solve.

BIOINFORMATICS DONE WELL

Data science done well looks easy – and that's a big problem for data scientists

simplystatistics.org
March 3, 2015 by Jeff Leek

SUBSTRATE

Cloud Computing

Google

amazon web services™ | EC2

Cluster Computing

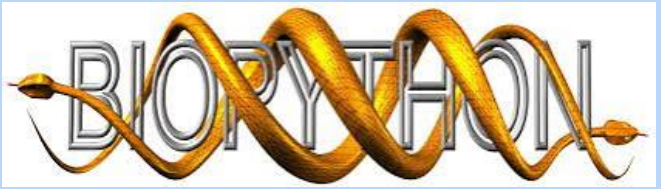
BAS™

LINUX Laptop & Desktop

ENVIRONMENT

“Command Line” and “Programming Languages”

Open Source



VS

Bioinformatics Software Suite

PREREQUISITES

- Access to a multi-core (24 cpu or greater), ‘high’ memory 64Gb or greater Linux server [Metagenomics].
- Familiarity with the ’command line’.
- Basic knowledge of how to install software
- Basic knowledge of R and statistical programming
- Basic knowledge of Statistics and model building

BIOINFORMATICS

- Know and Understand the experiment
  - "The Question of Interest"
- Develop a deep familiarity and understanding of the techniques/tools/methods
- Build a set of assumptions/expectations
  - Mix of technical and biological
  - Spend your time testing your assumptions/expectations
  - Don't spend your time finding the "best" software
- Don't under-estimate the time Bioinformatics may take
- Be prepared to accept 'failed' experiments

THE BOTTOM LINE

The Bottom Line:

- Spend the time (and money) planning and producing **good quality, accurate and sufficient data** for your experiment.
- Get to know to your data, develop and test expectations
- Result, you'll **spend much less time** (and less money) extracting biological significance and results during analysis.

INTRODUCTION TO SEQUENCING

Section

HISTORY

- It would take a few more decades after the discovery of the double helix in 1953 before we could readily analyze fragments of DNA.
- RNA sequencing actually preceded DNA sequencing when Walter Friers from the University of Ghent published the first complete gene and genome of Bacteriophage MS2 in 1972 and 1976 respectively.
- Location specific primer extension: Raw Wu (1970), using DNA polymerase catalysis and specific nucleotide labeling.
- Chain-terminating inhibitors: Frederick Sanger (1977), aided in speeding up the process

HISTORY

- Leroy E. Hood's laboratory at the California Institute of Technology announced the first semi-automated DNA sequencing machine in 1986.

- Applied Biosystems' produced the first fully automated sequencing machine, the ABI 370, in 1987, followed by the ABI Prism 373, (1990), ABI Prism 377 (1995), ABI Prism 310 (also 1995) represented the first capillary sequencer, ABI Prism 3700 (1999, the workhorse of the human genome project), ABI 3730xl DNA analyzer (2002) @ **2M bases per day**.

HISTORY

- Primer Walking
  - Design a primer that matches the sequence neighboring the unknown sequence
  - Sequence the short DNA strand using the Sanger method
  - The new sequenced portion is used to design a new primer and repeat
- *de novo* sequencing or “shotgun sequencing”
  - High-molecular weight DNA is sheared into random fragments
  - ‘Shorter' fragments are cloned into a vectors
  - clones are sequenced from both ends, creating two ”reads”
  - original sequence is reconstitutes by ”assembling” the reads

ROCHE 454

- The first massively parallel method (aka “next generation sequencing”) to become commercially available was developed by 454 Life Sciences in 2005 (acquired by Roche in 2007) and is based on the pyrosequencing technique. Similar to the Sanger method, sequencing is carried out using primed synthesis by DNA polymerase. However in the 454 pyrosequencing method, the DNA fragments are presented with each of the four dNTPs sequentially and without a dye-terminator, as is done with Sanger sequencing, allowing for multiple incorporation in the same flow. The amount of the incorporation is monitored by luminometric detection of the pyrophosphate released (hence the name ”pyrosequencing”).
- 700 Mb/day

ILLUMINA (SOLEXA)

- The second “next generation” sequencing technology to be released (in 2006) was Illumina Solexa sequencing. A key difference between Roche 454 and Illumina sequencing was the use of chain-terminating nucleotides. The fluorescent label on the terminating base can be removed to leave an unblocked 3’ terminus, mating the chain termination a reversible process. The method thus sequences one base at a time, rather than 0 or more bases as does Roche 454.
- > 400Gb/day

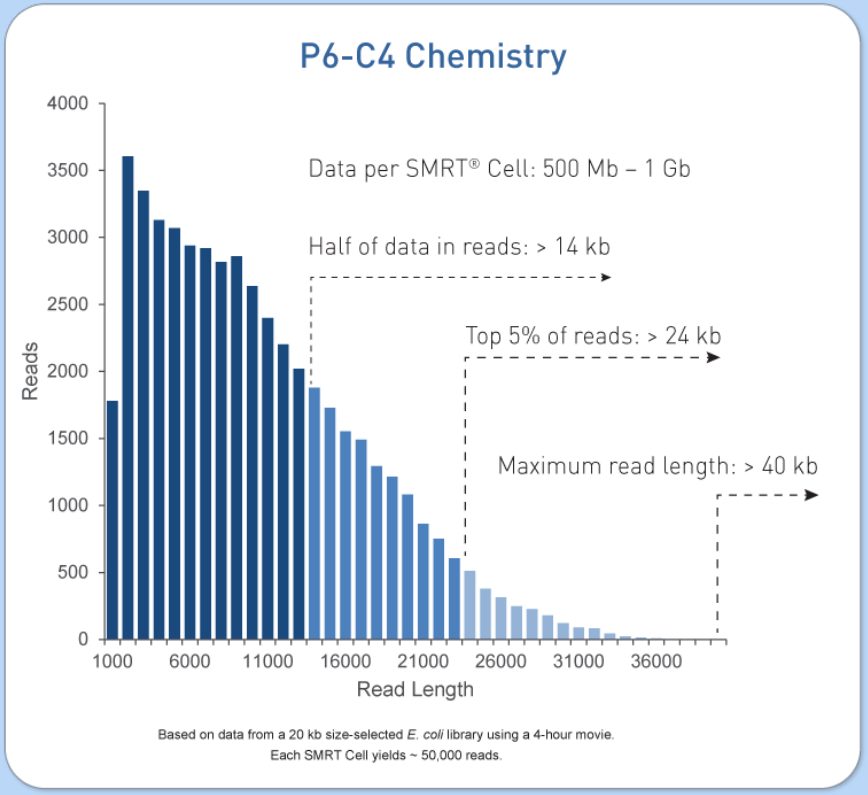
LIFE TECHNOLOGIES

- Ion Torrent PGM, first available in 2011, generates up to 400bp reads (reported) and up to 2Gb (5.5m reads) per run. Cheap fast runs. Ion Proton system can generate up to 10Gb per run. Generates flowgrams and SFF files similar to Roche 454 data as well as the standard fastq files. Ion Torrent is also considered a second generation sequencer.
- ~ 64Gb/day

PACIFIC BIOSYSTEMS

- Pacific Biosystems is so far the most successful third generation DNA sequencing system. Key differences are that its a single molecule, real time (SMRT) technology and capable of producing sequences of multi-kilobases.
- ~ 2Gb/day (20Gb/day)

Third generation sequencers are single molecule sequencers.



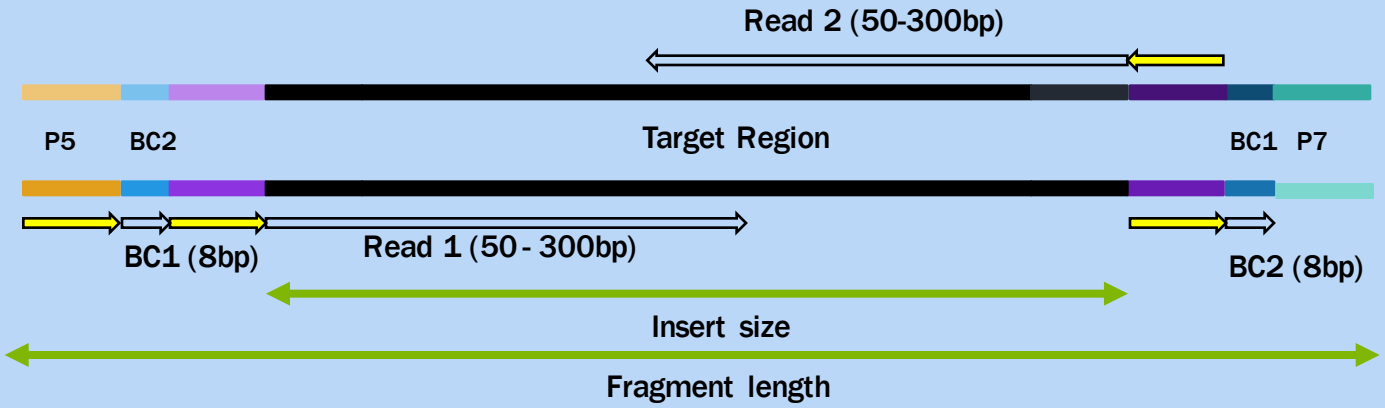
OXFORD NANOPORE

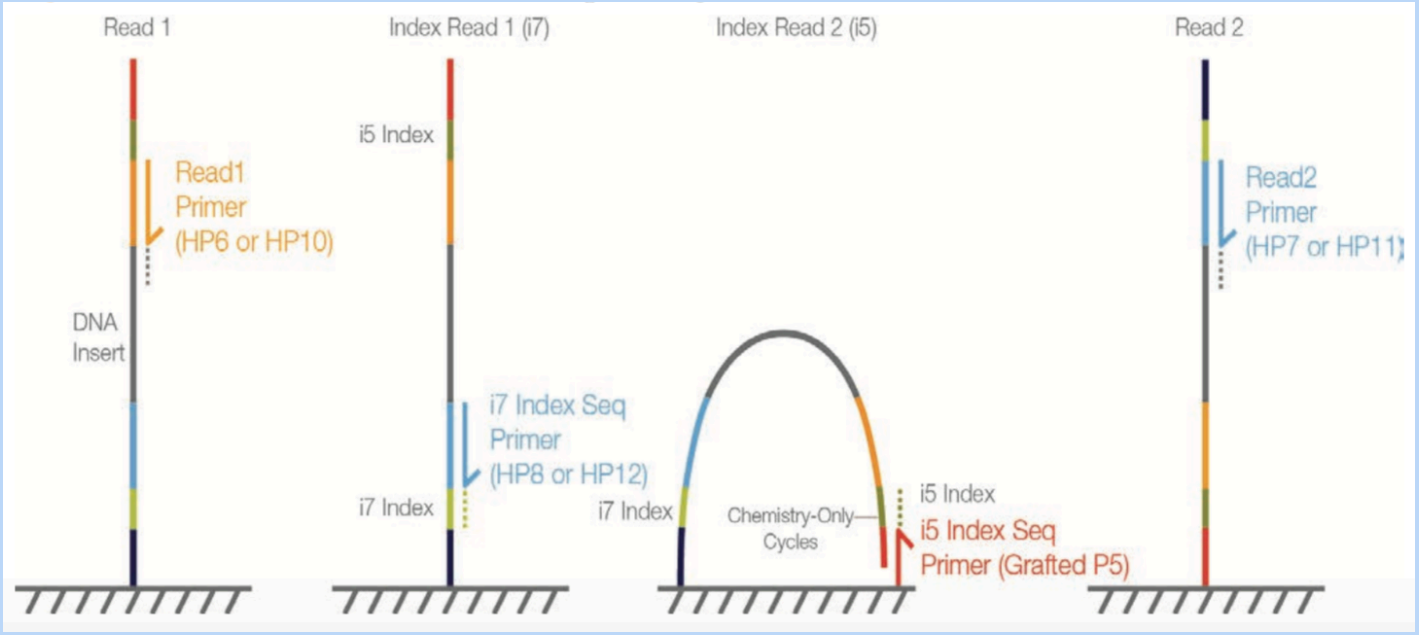
- Another 3rd generation sequencer, founded in 2005 and recently (May 2015) produced a commercial product, the MinION. The sequencer uses nanopore technology developed in the 90's to sequence single molecules.
- > 1Gb per flowcell

FYI: 4th generation sequencing is being described as "In-situ sequencing", for applications such as "spatial transcriptomics"

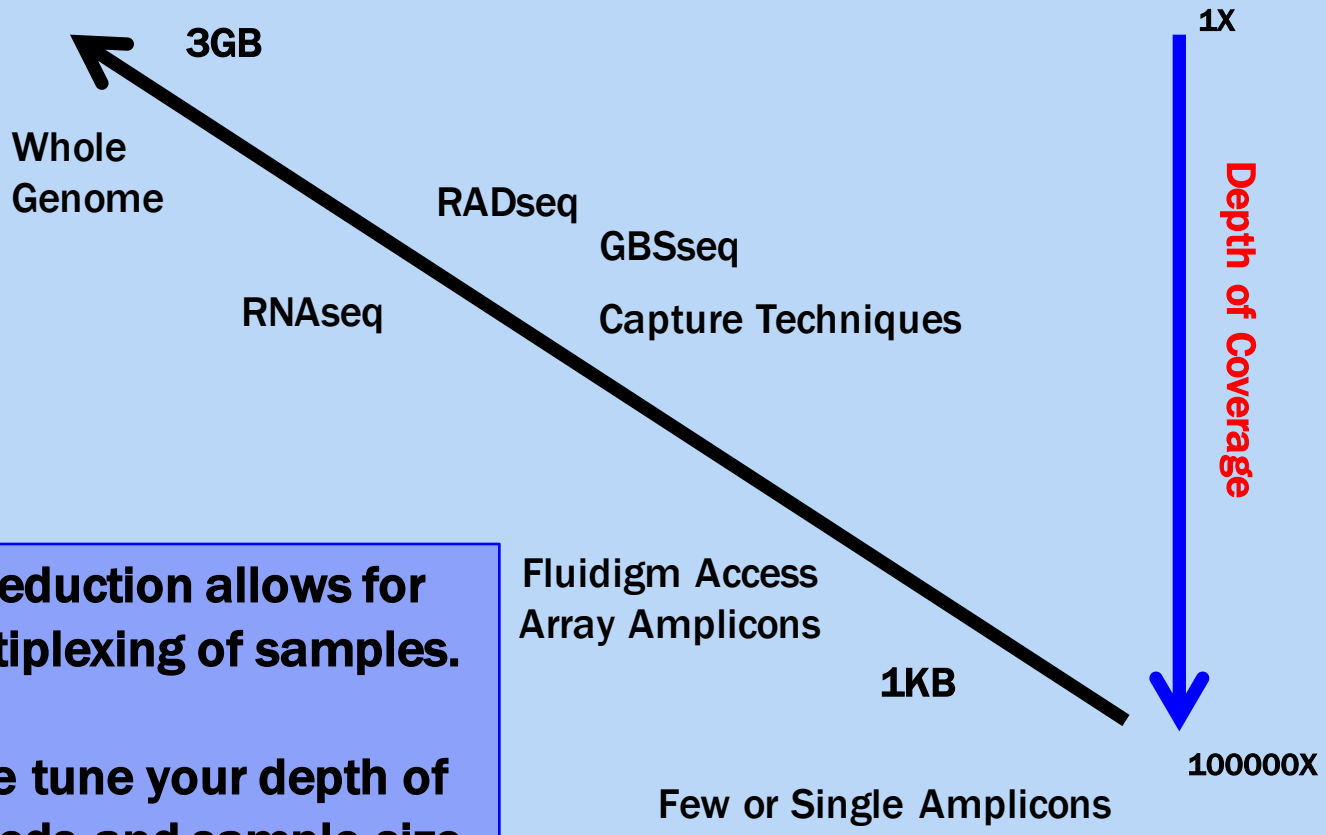
ILLUMINA SEQUENCING

- Illumina SBS





GENOMIC REDUCTION



Genomic reduction allows for greater multiplexing of samples.

You can fine tune your depth of coverage needs and sample size with the reduction technique

ILLUMINA HISEQ SEQUENCING

- http://www.illumina.com/systems/hiseq-3000-4000/specifications.html

2500

MiSeq

	HISEQ 3000 SYSTEM	HISEQ 4000 SYSTEM
No. of Flow Cells per Run	1	1 or 2
Data Yield: 2 × 150 bp 2 × 75 bp 1 × 50 bp	 650-750 Gb 325-375 Gb 105-125 Gb	 1300-1500 Gb 650-750 Gb 210-250 Gb
Clusters Passing Filter (Single Reads) (8 lanes per flow cell)	2.1-2.5 billion	4.3-5 billion
Quality Scores: 2 × 50 bp 2 × 75 bp 2 × 150 bp	 ≥ 85% bases above Q30 ≥ 80% bases above Q30 ≥ 75% bases above Q30	 ≥ 85% bases above Q30 ≥ 80% bases above Q30 ≥ 75% bases above Q30
Daily Throughput	> 200 Gb	> 400 Gb
Run Time	< 1-3.5 days	< 1-3.5 days
Human Genomes per Run*	up to 6	up to 12
Exomes per Run**	up to 48	up to 96
Transcriptomes per Run***	up to 50	up to 100

ILLUMINA MISEQ SEQUENCING

MISEQ REAGENT KIT V2

READ LENGTH	TOTAL TIME*	OUTPUT
1 × 36 bp	~4 hrs	540-610 Mb
2 × 25 bp	~5.5 hrs	750-850 Mb
2 × 150 bp	~24 hrs	4.5-5.1 Gb
2 × 250 bp	~39 hrs	7.5-8.5 Gb

MISEQ REAGENT KIT V3

READ LENGTH	TOTAL TIME*	OUTPUT
2 × 75 bp	~21 hrs	3.3-3.8 Gb
2 × 300 bp	~56 hrs	13.2-15 Gb

Reads Passing Filter†

MISEQ REAGENT KIT V2

Single Reads	12-15 M
Paired-End Reads	24-30 M

MISEQ REAGENT KIT V3

Single Reads	22-25 M
Paired-End Reads	44-50 M

Quality Scores††

MISEQ REAGENT KIT V2

> 90% bases higher than Q30 at 1 x 36 bp
> 90% bases higher than Q30 at 2 x 25 bp
> 80% bases higher than Q30 at 2 x 150 bp
> 75% bases higher than Q30 at 2 x 250 bp

MISEQ REAGENT KIT V3

> 85% bases higher than Q30 at 2 x 75 bp
> 70% bases higher than Q30 at 2 x 300 bp

Perform

Visit the Illumina website

- How many sequencing machines does Illumina Have?
- What are the differences between the HiSeq 2500 series and HiSeq 3000/4000 series machines?
- What are differences in read length options between the MiSeq and HiSeq?
- What are the differences between the HiSeq's High Output Run Mode and the Rapid Run Mode
- How many reads (and bp) do you get on one lane of a HiSeq 2500 in High Output Run Mode for 1x50 and 2x100, compare those to what you get on the HiSeq 3000
- How many reads (and bp) do you get from one run of a MiSeq 2x300.

HOMEWORK

Getting to know Illumina

EXPERIMENTAL DESIGN: COMMUNITY PROFILING

Section

EXPERIMENTAL DESIGN

- In high throughput biological work (Microarrays, Sequencing, HT Genotyping, etc.), what may seem like small technical artifacts introduced during sample extraction/preparation can lead to large changes, or technical bias, in the data.
  - Not to say this doesn't occur with smaller scale analysis such as Sanger sequencing or qRT-PCR, but they do become more apparent and may cause significant issues during analysis.

GENERAL RULES FOR PREPARING SAMPLES

- Prepare more samples then you are going to need, i.e. expect some will be of poor quality, or outright fail
- Preparation stages should occur across all samples at the same time (or as close as possible) and by the same person
- Spend time practicing a new technique to produce the highest quality product you can, reliably
- Quality should be established using Fragment analysis traces (pseudo-gel images, FOR RNA RIN > 7.0)
- DNA/RNA should not be degraded
  - 260/280 ratios for RNA should be approximately 2.0 and 260/230 should be between 2.0 and 2.2. Values over 1.8 are acceptable
- Quantity should be determined with a Fluorometer, such as a Qubit.

BE CONSISTENT ACROSS ALL SAMPLES!!!

SEQUENCING DEPTH

- The first and most basic question is how many base pairs of sequence data will I get
  Factors to consider are:
  - 1. Number of reads being sequenced
  - 2. Read length (if paired consider then as individuals)
  - 3. Number of samples being sequenced
  - 4. Expected percentage of usable data

$$bpPerSample = \frac{readLength * readCount}{sampleCount} * 0.8$$

- The number of reads and read length data are best obtained from the manufacturer's website (search for specifications) and always use the lower end of the estimate.

SEQUENCING COVERAGE

- Once you have the number of base pairs per sample you can then determine expected coverage
  Factors to consider are:
  - 1. Length of the genome (in bp)
  - 2. Any extra-genomic sequence, or contamination

$$expectedCoverage = \frac{bpPerSample}{totalGenomicContent}$$

- Coverage is determined differently for "Counting" based experiments (RNAseq, amplicons) where an expected number of reads per sample is typically more suitable.

MG SEQUENCING DESIGNS

- Taxonomic Identification
  - Amplicon based (e.g. 16s variable regions)
  - Shotgun Metagenomics
- Functional Characterization
  - Shotgun Metagenomics
  - Shotgun Metatranscriptomics (active)
- Genome Assembly, Function and Variation
  - Shotgun Metagenomics
  - Shotgun Metatranscriptomics

PROPOSAL STAGE

Considerations (when a literature search turns up nothing)

- Proportion that is host (non-microbial genomic content)
- Proportion that is microbial (genomic content of interest)
  - Number of species
  - Genome size of each species
  - Relative abundance of each species

The back of the envelope calculation

$$\#\text{Reads} = \frac{\text{Coverage} * \text{Average\_Genome\_Size}}{0.8 * \text{Read\_Len} * \text{Dillution\_factor} * \text{1-Host\_prop}}$$

RAREFACTION CURVES

Plot rarefactions curves of Amplicons, or reconstituted 16S sequence to determine if saturation is achieved

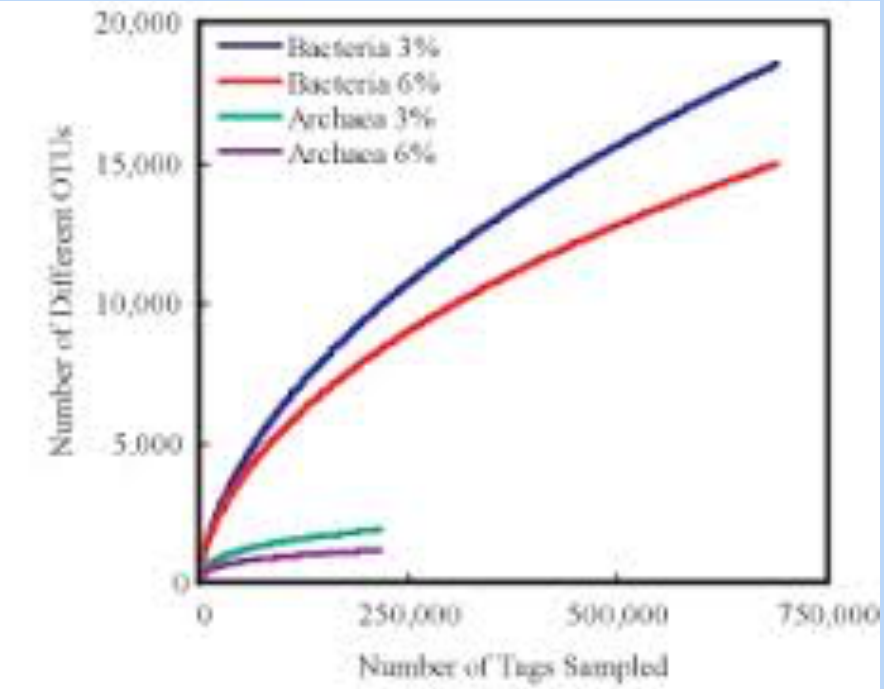
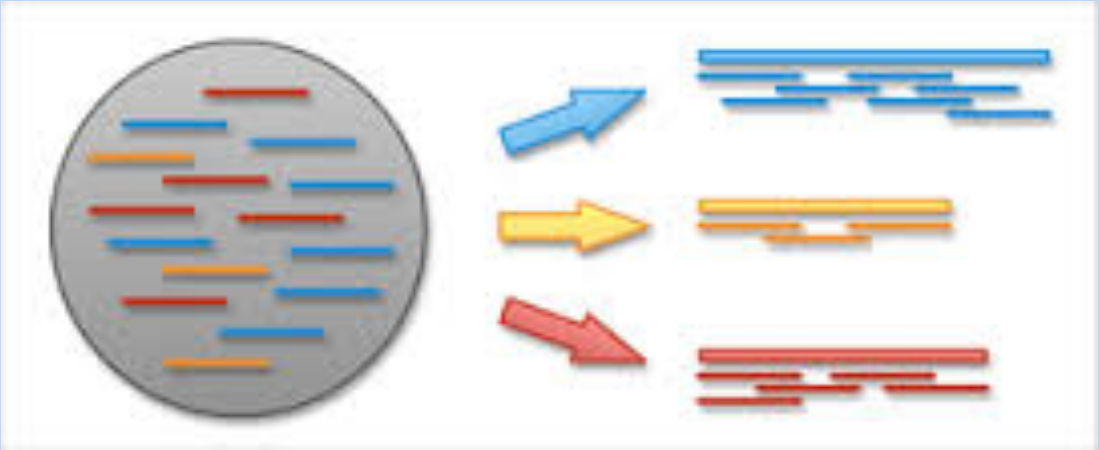


Figure 2

Miller CS, Baker BJ, Thomas BC, Singer SW, Banfield JF (2011) EMIRGE: reconstruction of full-length ribosomal genes from microbial community short read sequencing data. Genome biology 12: R44 doi:10.1186/gb-2011-12-5-r44.

METAGENOMICS ASSEMBLY



To determine if you've sequenced 'enough' to re-assemble 'most' of the community member's genetic content, look to what is left over - proportionally

TAKE HOMES

- Experience and/or literature searches (other peoples experiences) will provide the best justification for estimates on needed depth.
- Don't forget about 'contamination' can often be a large source of reads.
- 'Longer' reads are better than short reads, especially for assembly.
- Libraries can be sequenced again, so do a pilot, estimate 'contamination', compute rarefaction/16s coverage then sequence more accordingly.

COST ESTIMATION

- DNA/RNA extraction QA/QC (Bioanalyzer/Gels)
- Metatranscriptomes: Enrichment of RNA of interest and RNA library preparation
  - Library QA/QC (Bioanalyzer and Qubit)
  - Pooling ($10/library)
- Metagenomes: DNA library preparation
  - Library QA/QC (Bioanalyzer and Qubit)
  - Pooling ($10/library)
- Community Profiling: PCR reactions
  - Library QA/QC (Bioanalyzer and Qubit/microplate reader)
  - Pooling
- Sequencing (Number of Lanes / runs)
- Bioinformatics (General rule is to estimate the same amount as data generation, i.e. double your budget)

http://dnatech.genomecenter.ucdavis.edu/prices/

Perform

- Search the web for 16s reads per sample recommendations.
- Cost estimate a metagenomics (DNA) project analysis on 8 samples, 2x100 on a HiSeq 3000 using the number of reads target determined in the last homework

HOMEWORK

COVERAGE

SHOTGUN READ PROCESSING

Section

PREPROCESS READS

- We have found that aggressively ”cleaning” and processing reads can make a large difference to speed and quality of assembly and mapping results. Cleaning your reads means, removing reads/bases

  that are:

  - not of primary interest (contamination)
  - originate from PCR duplication
  - artificially added onto sequence of primary interest (vectors, adapters, primers)
  - low quality bases
  - other unwanted sequence (polyA tails in RNA-seq data)
  - join short overlapping paired-end reads

READ PREPROCESSING STRATEGIES

- Identity and remove contaminant and vector reads
  - Reads which appear to fully come from extraneous sequence should be removed.
- Quality trim/cut
  - "end" trim a read until the average quality > Q (Lucy)
  - remove any read with average quality < Q
- eliminate singletons/duplicates
  - If you have excess depth of coverage, and particularly if you have at least x-fold coverage where x is the read length, then eliminating singletons is a nice way of dramatically reducing the number of error-prone reads.
  - Read which appear the same (particularly paired-end) are often more likely PCR duplicates and therefor redundant reads.
- eliminate all reads (pairs) containing an "N" character
  - If you can afford the loss of coverage, you might throw away all reads containing Ns.
- Identity and trim off adapter and barcodes if present
  - Believe it or not, the software provided by Illumina, either does not look for, or does a mediocre job of, identifying adapters and removing them.

READ PREPROCESSING PIPELINE

- The pipeline we've developed (expHTS) for read preprocessing employs the following steps for each read:

  Contaminant screening (phiX minimum) → PCR duplicate detection/removal → 5' and 3' end quality trimming→ Join (when possible) and remove adapters from paired end reads → Final cleanup, remove too short sequence, second polyA/T removal. Finally generate statistics for each sample.

- This pipeline is realized using the following applications

  screen.py (part of expHTS), extract_unmapped_reads.py (part of expHTS), Super Deduper, Sickle, Flash, cleanupWrapper.py (part of expHTS)

- This pipeline is also under active development:

https://github.com/msettles/expHTS

Processes multiple samples within a experiment, assumes a structure by default.

PREPROCESSING

- screen.py and extract_unmapped_reads.py
  - Remove contaminants (at least PhiX), uses bowtie2 then extracts all reads (pairs) that are marked as unmapped.
- Super-Deduper
  - Remove PCR duplicates (we use bases 10-35 of each paired read)
- Sickle
  - Trim sequences (5' and 3') by quality score (I like Q20)
- FLASH2
  - Join and extend, overlapping paired end reads
  - If reads completely overlap they will contain adapter, remove adapters
  - Identify and remove any adapter dimers present
- cleanupWrapper.py
  - Remove any reads that are less then the minimum length parameter
  - Run a polyA/T screen (if RNA)
  - Produce run statistics

FILES AND DIRECTORY STRUCTURE

I use a strict directory structure to show the relationship between results and input, expHTS assumes this directory structure though it can be changed.

- PARENT folder, name of the experiment
  - 00-RawData
    - SEQUENCE_ID_1
      - Fastq Files
    - SEQUENCE_ID_2
      - Fastq Files
  - 02-Cleaned
    - SAMPLE_ID_1
      - Fastq Files
    - SAMPLE_ID_2
      - Fastq Files
    - Preprocessing_Summary.log

METADATA FILE

- Variables that describe each sample, including those that will be used to compare samples to each other (experimental factors). Also good to include all technical factors that may influence experimental results (ex. Day of RNA isolation) in order to test for effect later.

- samples.txt – is a plain text tab delimited metadata file that will be used within the workshop to run expHTS and the R differential expression analysis. Rows are samples, columns are metadata
- Two REQUIRED columns, add more columns as you need
  - "SEQUENCE_ID" – folder name containing the sequences
  - "SAMPLE_ID" – Name in which to assign the sample

QA/QC

- Beyond generating better data for downstream analysis, cleaning statistics also give you an idea as to the quality of the sample, library generation, and sequencing technique used to generate the data.
- This can help inform you of what you might do in the later processing steps.
- I've found it best to perform QA/QC on both the sequencing run as a whole (poor samples can affect other samples) and on the samples themselves as they compare to other samples (REMEMBER, BE CONSISTANT).
  - Reports such as Basespace for Illumina, are great ways to evaluate the runs as a whole.
  - PCA/MDS plots of the preprocessing summary are a great way to look for technical bias across your experiment, expect randomness

METAGENOMICS SOFTWARE

- Remember this is “Data Science”, focus on the questions you wish to answer, the path to answering those questions and not on what is the ‘best’ software to use.
  - Software is rarely generated for the ’generic’ situation
  - Often untied to type of sequencing data, though often dependant
  - Often one and done, look for software that is currently being supported.
- Work in a comparative framework, consistency of results across samples indicates comparability, biases are at least consistent and/or unassociated with the project factors.

TAXONOMIC ASSIGNMENT

KRAKEN

- A taxonomic classifier using k-mers, current db contains > 75Gb of microbial genome data (unique kmers).
- Requires a large server, 128Gb to 256Gb of memory
- Assigns each read to its lowest common ancestor in the tree in a taxonomic tree based on the set of kmers in a read
- Accepts 'single' read fasta format (flags for pairs and fastq)
  - Output is 'unusable', meant for additional processing
  - Kraken-translate to output taxonomic assignment for every read, output can then be used to build abundance tables
- Kraken-filter will move a read up the tree based on confidence of mapping (loosely based on proportion of kmers)
- Can build your own database
- Kraken-report and kraken-mpa-report for abundance table construction

MetaPhlAn

- Classifies by using a set of marker genes – measures species abundances

ASSEMBLY

- Many assemblers to choose from and more each day
  - Recent tutorial using cloud computing

  http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4496567/
- Most metagenomics assemblers use kmers
  - Either normalize reads by kmers (remove what appears to be redundant information)
  - Or first bin by kmers (each bin is assumed to be a unique species), then assemble each bin (first normalizing by kmers).
- Map reads back to assembly to estimate coverage/count
- BUT then you have to do some with ambiguous contigs
  - Identify ORFs, marker genes, etc. to characterize gene/taxon content
  - IT IS all about the databases!!

MG-RAST

The MG-RAST system provides answers to the following scientific questions:

- Who is out there? Identifying the composition of a microbial community either by using amplicon data for single genes or by deriving community composition from shotgun metagenomic data using sequence similarities.
- What are they doing? Using shotgun data (or metatranscriptomic data) to derive the functional complement of a microbial community using similarity searches against a number of databases.
- Who is doing what? Based on sequence similarity searches, identifying the organisms encoding specific functions.
- Finally compare samples to each other

MG-RAST

- Can upload
  - 16s amplicons
  - Metagenomes/Metatranscriptomes
  - Assembled contigs
  - Raw reads
- Use their resources for analysis, don't have to have your own computational resources
- More of a black box, but can download many of output data options
- Subjected to their philosophy for analysis of metagenomic/transcriptomic data

A downloadable alternative to MG-RAST is MEGAN5

Perform

- Launch read preprocessing via
  - expHTS preprocess
- When complete perform QA/QC

HOMEWORK 8

Running preprocessing

WORKSHOP DATASET

Section

SLASHPILE

- Slash Pile – Accumulated debris from cutting brush or trimming trees
  - Measured, bacteria and fungal communities using 5 amplicons
    - 16sV1V3
    - 16sV4V5
    - ITS1
    - ITS2
    - LSU
- 3 – slashpiles
- 2 depths
- Distance from slashpile

Perform

- Briefly scan the metadata files on the github page
- Upload the fastq reads files to AWS, then extract them
  - https://drive.google.com/open?id=0B_3lTFvg_WhxMVQwUEtZaV9rQzA

HOMEWORK

Workshop Data

THE COMMAND LINE

SECTION

THE COMMAND LINE

- The command line is powerful and the preferred way to run bioinformatics tools

BASICS:

Prompt msettles@MacBook-Pro:~$

~ is your home director

… $ command [parameters] [files] ENTER

parameters begin with a - short parameter or -- long parameter

Help … $ man command

… $ command –h

… $ command –help

Tab complete!!

THE COMMAND LINE

Command input/output:

- file/folder on the command line, either as a positional argument or a parameter, or defaults
- stdin aka 'standard in', input pipe
- stdout aka 'standard out', output pipe
- stderr aka 'standard error', pipe for error messages

Special characters:

|	vertical bar is the pipe, it pipes the stdout of one command to the stdin of another cmd
< > 2>	feeds stdin, stdout, stderr to the command
>>	append
&	at the end of a command will run the command in the background
.	Current directory
..	Up one folder
/	folder delimiter
*	wild character, match anything

When naming files/folders avoid special characters and spaces (use . and _)!!

THE COMMAND LINE

Basics Commands:

pwd	print the working directory (current dir)
ls [file/dir]	list the current contents of the directory
ls –lah [file/dir]	long list, human readable, show hidden of the directory
cd to	change directory
cp f1 f2	copy file f1 to file f2
mv from to	mv directory (also way to rename)
mkdir dir	make directory
rm file	remove file
rmdir dir	remove an empty directory
rm –r dir	recursively remove a non-empty directory
nano file	edit a document
cat f1 [f2]	print to stdout file f1 then if provided file f2 (concatenate)
less file	view file, one page at a time
head file	view the first 10 lines of a file
tail file	view the last 10 lines of a file

CHEAT SHEETS

- Cheat Sheets are available on the github page

Perform

- Look at your home directory, what files are already there, what files are hidden
- Create a new folder
- Move into that folder and create a file (write anything)
- Show that file, using less and cat
- Go back to your home directory
- Create a second folder
- Copy the file from the first directory to the new directory
- Move to your home directory
- Remove both folders
- Learn to use the command 'wc' with 'man wc'
- Play with the new commands

HOMEWORK

Playing with the command line

GIT

- Git is an open source program for tacking changes in text files. Its was written by the author of the Linux operating system, Linus Torvalds. Use git for:
  - Tracking software changes
  - Tracking notes changes
  - Tracking document changes
- Github is a web-based Git repository hosting service, free to use for open repositories
  - This workshop is available via git and github
  - https://github.com/msettles/Workshop_Community-Profiling

GIT

- git clone [repo] –to clone a copy of a repository from the remote server (github) onto your workstation. With your clone you can edit the documents as you please
- git status – displays the differences between your repositories and the current working head (changes you've made).
- git pull – when you fetch in changes to the repository and merging them in. for example if I edited the workshop repository, you can then pull in those changes to your workstation
- git checkout – to checkout different branches

Perform

- Install system software
  - sudo apt-get install bowtie2
  - sudo apt-get install openjdk-7-jdk
  - sudo apt-get install ant
- Using git clone, install from repositories on github
  - FLASH2
    - Using Make, link onto the Path
  - RDPtools
    - Using Make
  - dbcAmplicons
    - python setup.py install

HOMEWORK

Install expHTS using git and python

DBCAMPLICONS

Section

MICROBIAL COMMUNITY ANALYSIS

Goal: A culture independent method for profiling the diversity of a community.

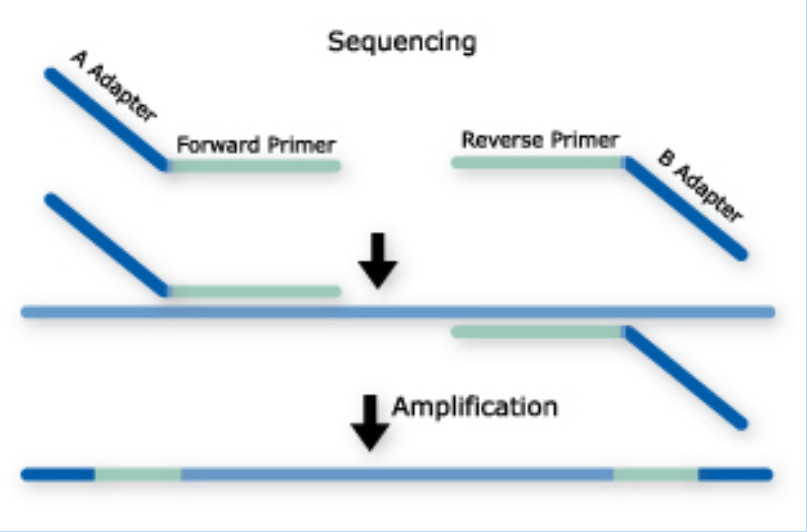
High-throughput sequencing technologies (such as Illumina) can generate millions of amplicons, across thousands of samples in a single run, and are today our best approach to deeply assess the environmental or clinical diversity of complex microbial assemblages of archaea, bacteria, and eukaryotes.

Using sequence variation within a common gene (e.g. 16s) to assign and count community members rather than counting individual cells. Assume each sequence variant is one community member.

AMPLICONS: COMMON APPROACH

- Single PCR
- Long primer sequences (~75bp) that contain barcodes and sequencing adapters
- Single or dual barcodes
  (dual barcode often within read 2)

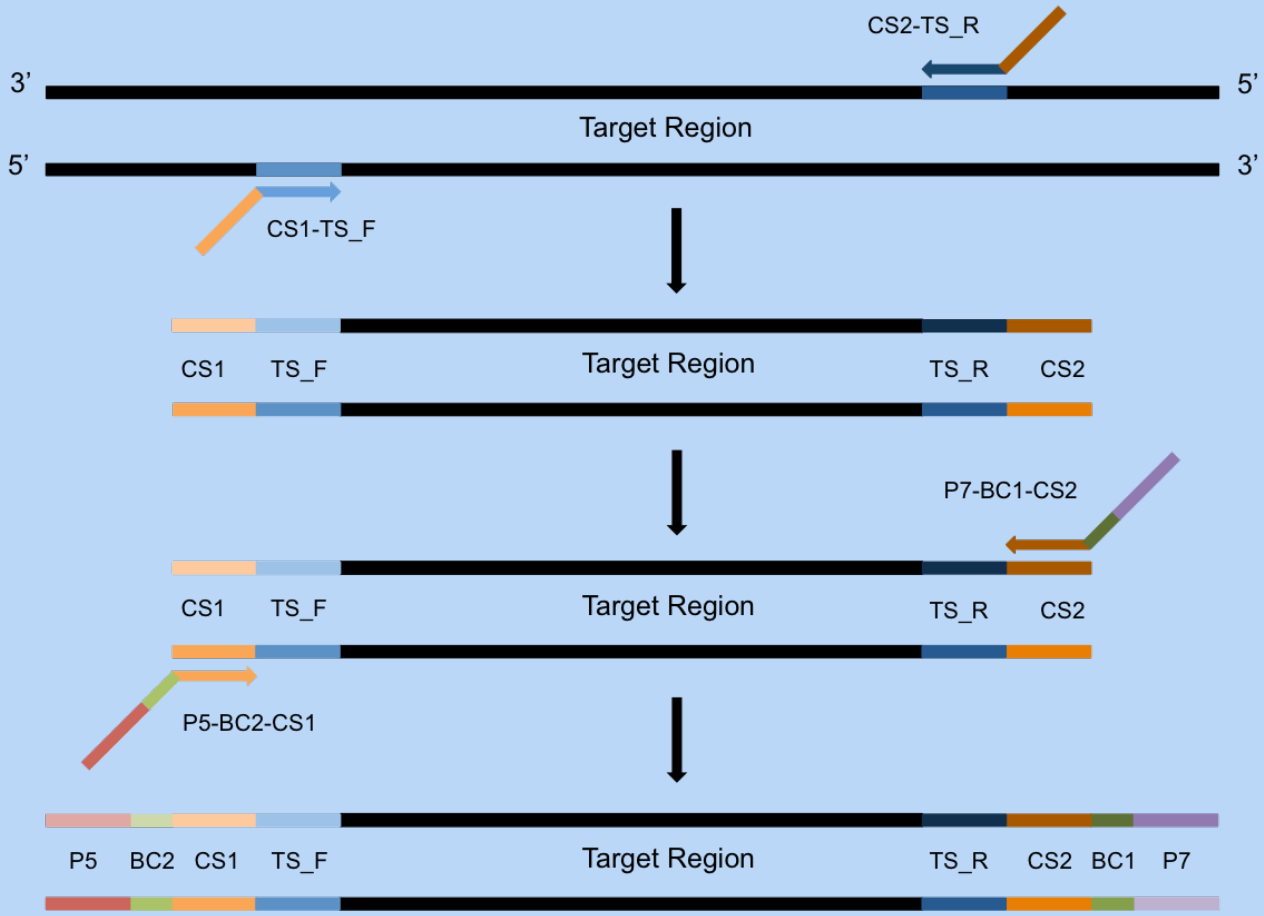
Common approach



DBCAMPLICONS

- Originally conceived in late 2012 to lower per sample costs on relatively short targeted (PCR) regions
  - 16S, ITS, LSU, 18S, etc. Community profiling
  - Extraction of mitochondria, virae, chloroplast regions, plasmids by PCR
  - Genotyping of samples for phylogenomics, genome to phenotype interactions
- Uses the Illumina platform, capably of pooling thousands, or even tens of thousands of barcoded samples/targets per sequencing run.
- Core Facility friendly, facilitates interactions between and across individual labs, standardizing workflows.

AMPLICONS: ALTERNATIVE APPROACH



PCR-1
Template with overhang

PCR-2
Add barcodes and adapters

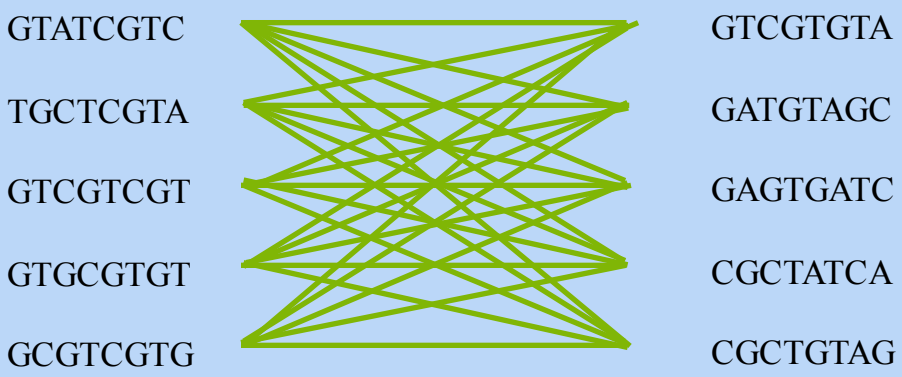
Barcodes and adapters are added in the second round of PCR

MULTIPLEX SAMPLES

Dual barcoding allows for massively multiplexing of samples using only a relatively few primers

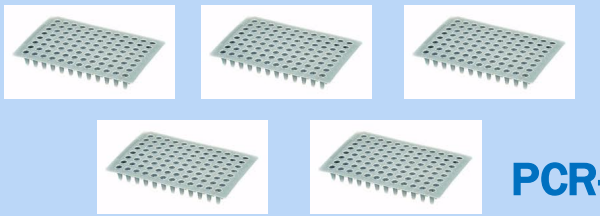
Pairing of BC1 and BC2 uniquely identifies sample

8bp [Error-Correcting] Barcodes

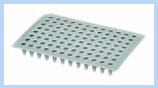


5 Pairs of Barcodes allows for multiplexing of **25 samples.**
32 Pairs can multiplex **1024 samples** in the same sequencing reaction

MULTIPLEX AMPLICON TARGETS

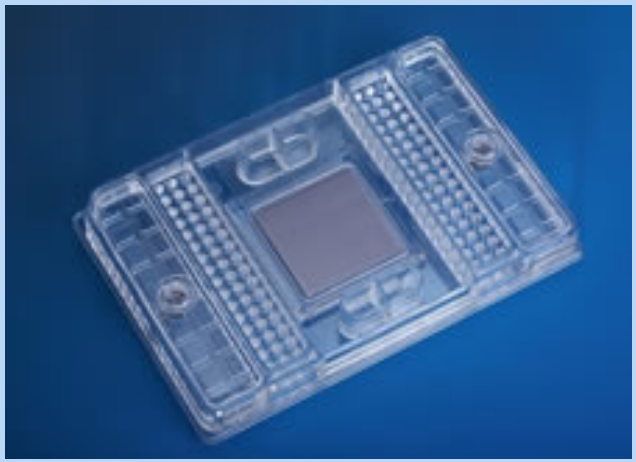


PCR-1 product



PCR-2 product

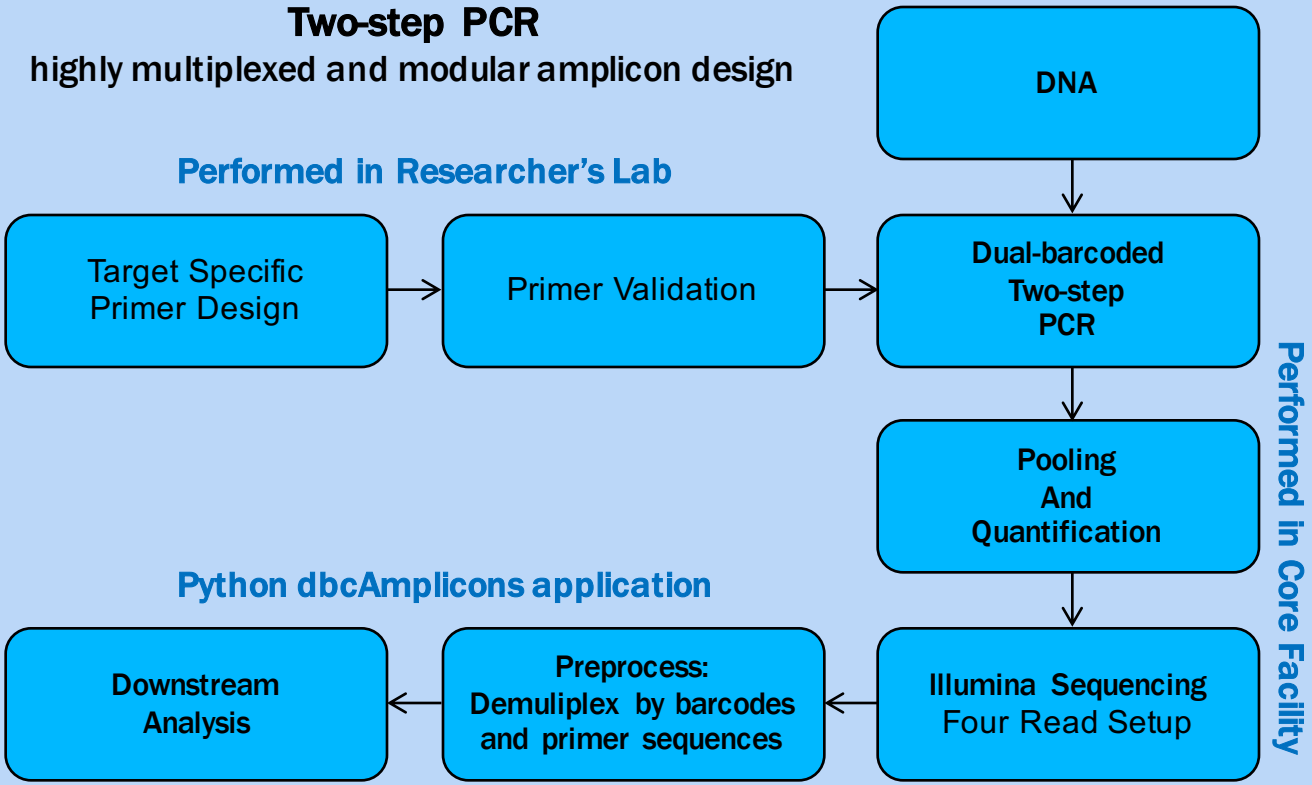
Fluidigm Access Array System



48 samples X 48 amplicons
2304 Two-step PCR reactions

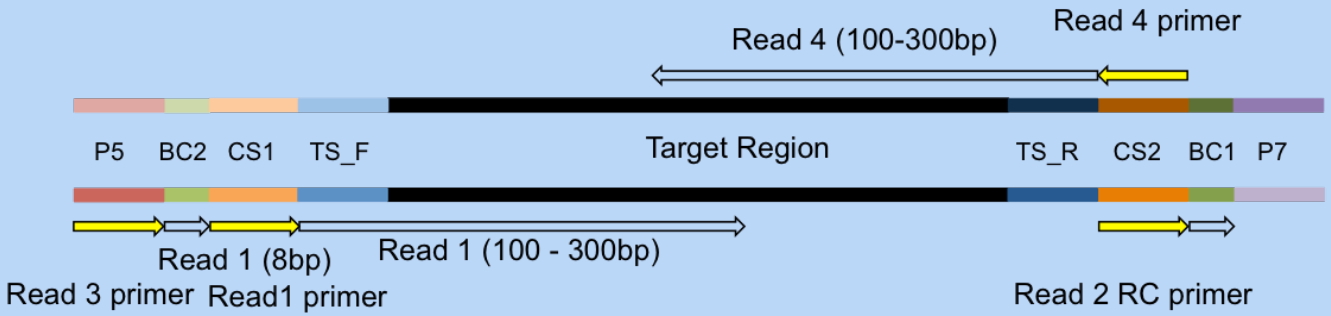
Pool many samples

DBCAMPLICONS



SEQUENCING

Illumina four read sequencing run



Read	Sequencing Primer
Read1 primer	CS1 - 5' ACACTGACGACATGGTTCTACA
Read2 primer	CS2 - 5' TACGGTAGCAGAGACTTGGTCT
BC1 primer	CS2rc - 5' AGACCAAGTCTCTGCTACCGTA
BC2 primer	Uses the P5 amplification primer

BIOINFORMATICS

Illumina bcl2fastq 1.8.4

Four Fastq files

Barcode Table

Primer Table

Sample Sheet

"Preprocess" Demultiplex barcode and primer sequences

Overlap paired reads (wrapper around custom Flash2)

Downstream Analysis

dbcAmplicons
Python Application

Barcode	Match determined by edit-distance, allowing for 1 mismatch per barcode (by default)
Primers	Match determined by Levenshtein Distance allowing for 4 mismatches (by default) with the final 4 (by default) bases required to be perfect matches

https://github.com/msettles/dbcAmplicons

DOWNSTREAM ANALYSIS

Population Community Profiling (i.e. microbial, bacterial, fungal, etc.)

dbcAmplicons Python Application

Screen	Using Bowtie2, screen targets against a reference fasta file, separating reads by those that produce matches and those that do not match sequences in the reference database.
Classify	Wrapper around the MSU Ribosomal Database Project (RDP) Classifier for Bacterial and Archaeal 16S rRNA sequences, Fungal 28S rRNA, fungal ITS regions
Abundance	Reduce RDP classifier results to abundance tables (or biom file format), rows are taxa and columns are samples ready for additional community analysis.

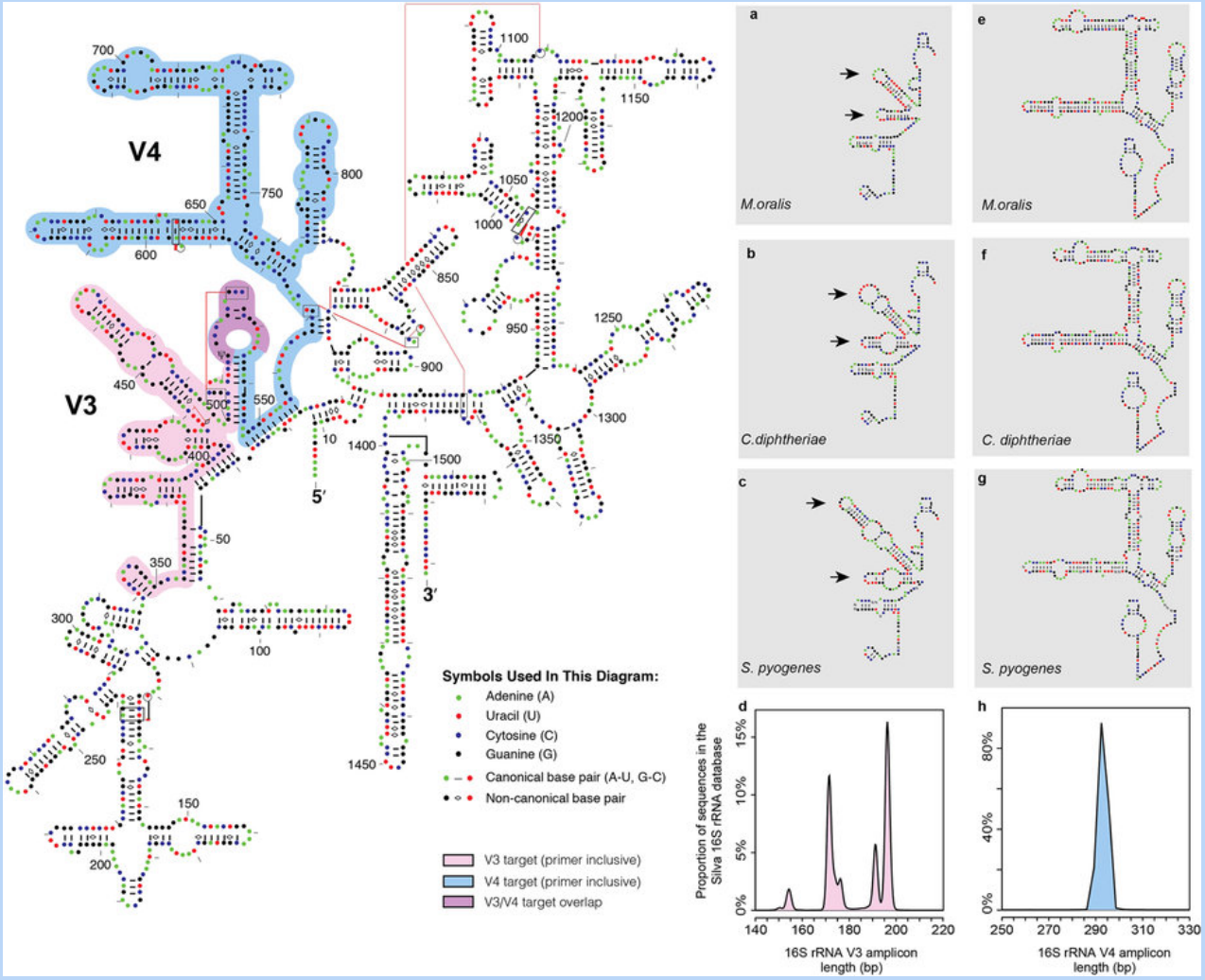
Targeted Re-sequencing

	Consensus - Reduce reads to consensus sequence for each sample and amplicon
R-functions to be added into dbcAmplicons	**Most Common** – Reduce reads to the most commonly occurring read in the sample and amplicon (that is present in at least 5% and 5 reads, by default)
	Haplotypes – Impute the different haplotypes in the sample and amplicon

SUPPLEMENTAL SCRIPTS

- convert2Readto4Read.py
  - For when samples are processed by someone else
- splitReadsBySample.py
  - To facilitate upload to the SRC

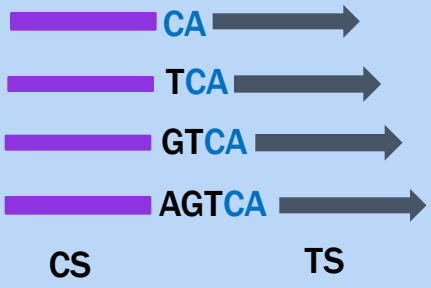
PRIMER DESIGN



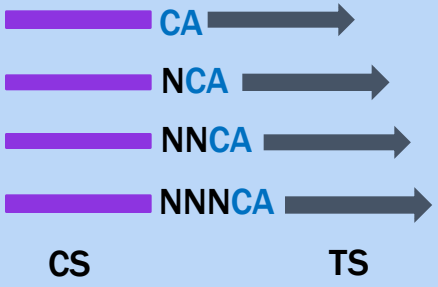
TEMPLATE SPECIFIC PRIMER DESIGN

- Each primer pair contains the following parts
  - CS1 or CS2 to attach second adapter/barcode primer
  - Phase-shifting bases [see below]
  - Linker sequence
  - Template specific primer sequence

Phase shifting primer



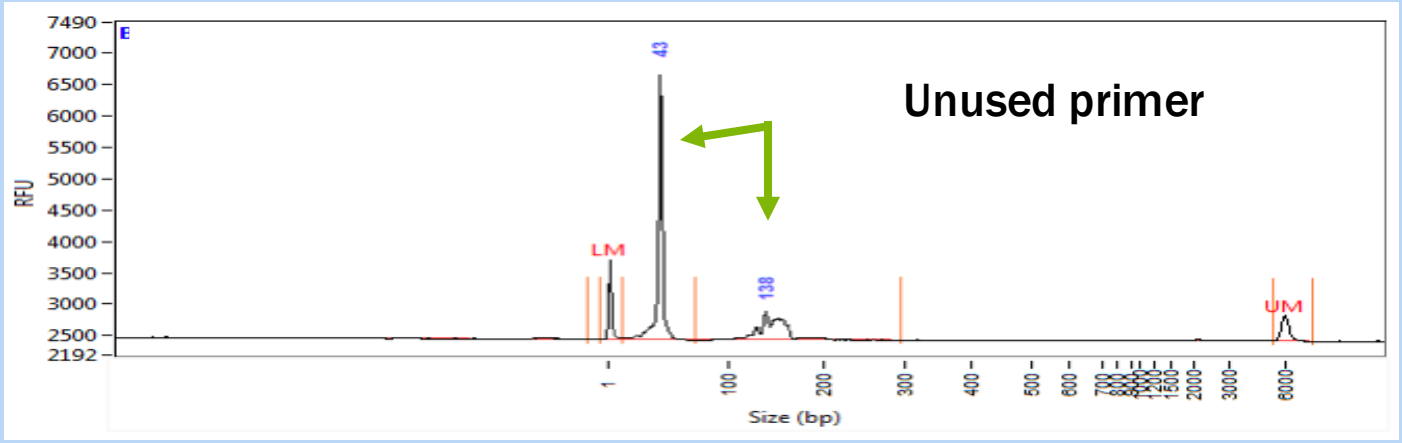
Phase shifting primers with PCR duplicate detection



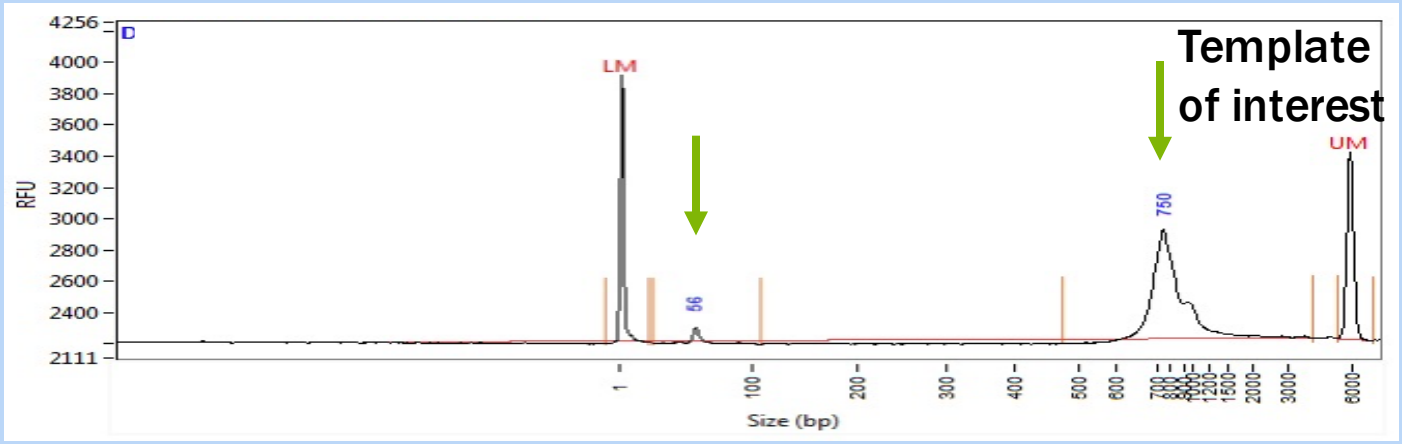
Ns, resolve to be PCR duplicate keys and should only ever appear once

QC: WHAT IS A "GOOD" LIBRARY?

Unused primers and adapters



BAD!



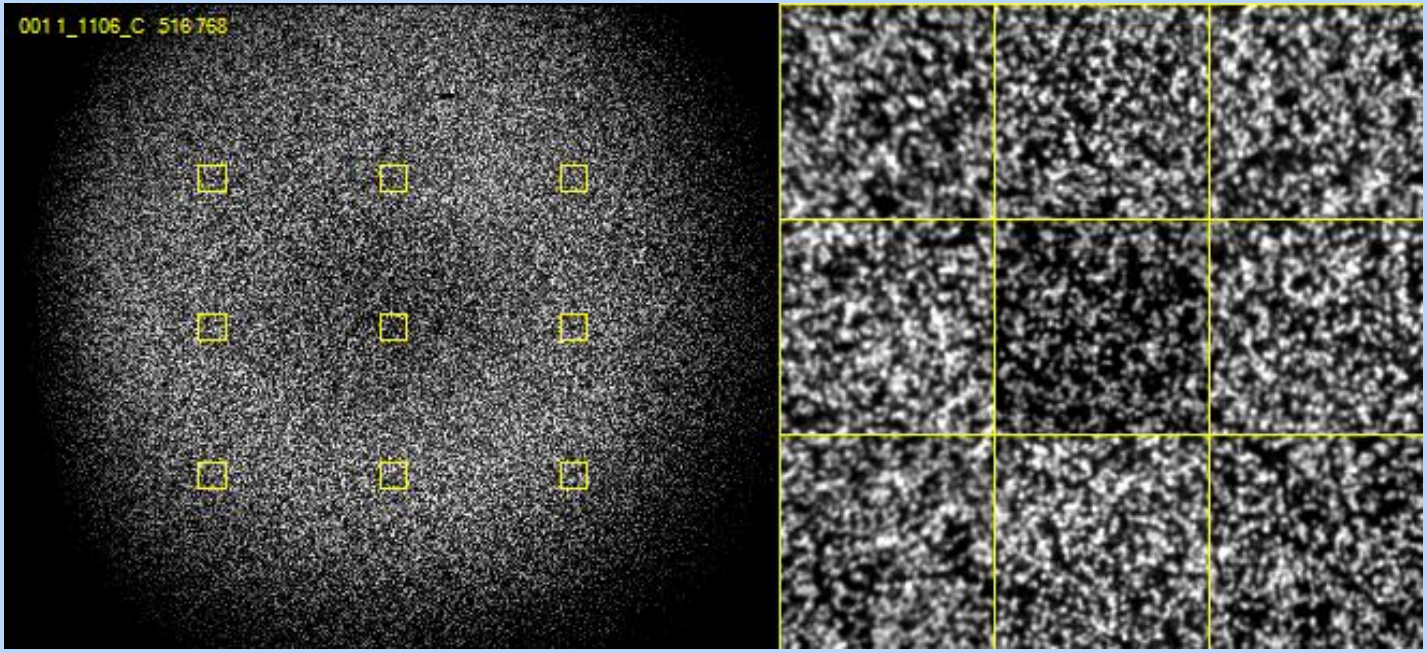
Good!

POOLING SAMPLES/AMPLICONS

- Even amplicon representation is important and difficult to achieve. Amplicon counts can vary from sample to sample by 100x
- Each amplicon should be evaluated by quality (ideally by trace) and quantity (fluourometery). Both qualities will effect final counts.
- Best practices
  - First group amplicons by quality/quantity profiles
  - Pool each group separately
  - If a small number of groups consider qPCR on each group for final pooling concentrations
  - If a large number re-quantify and pool to final pool.

NUCLEOTIDE DIVERSITY

Critically important for imaging clusters



AMPLICON RUNS ARE LOW COMPLEXITY

- Sequencing with lower cluster density
- Adding in highly diverse libraries (shotgun libraries ~15%)
- Phase-shifting template specific primers:

What is nucleotide diversity?

The term "nucleotide diversity" describes the proportion of each nucleotide (A, T, C, and G) at each position in a sequencing library. A "balanced" or "diverse" library has *equal proportions* of A, C, G, and T nucleotides at each base position in a sequencing library. Figure 9 illustrates cluster images and SAV data by cycle plots from diverse/ balanced, low-diversity, and unbalanced libraries.

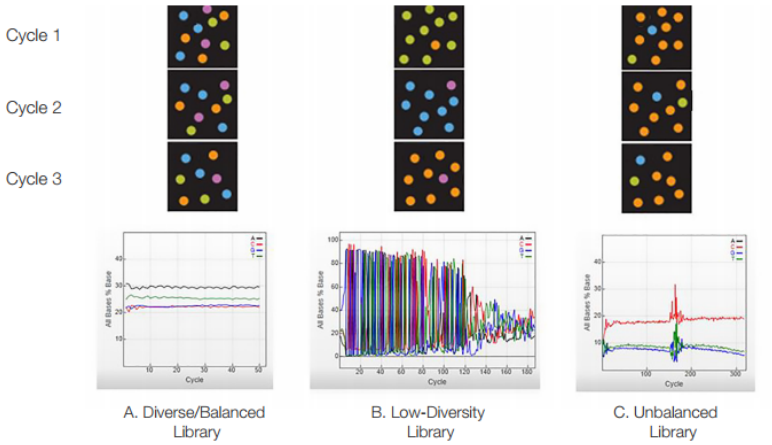


Figure 9: Nucleotide Diversity and Data by Cycle: % Base Plots. A) Diverse/balanced libraries contains equal proportions of A, T, C, and G nucleotides. The Data by Cycle: % Base plot shows even, horizontal curves centered around 25%. B) Low-diversity libraries, such as amplicon, enriched/targeted, or ChIP libraries, have an uneven proportion of nucleotides across the flow cell from one cycle to the next. The Data by Cycle: % Bases plot shows large intensity spikes at each cycle. C) Unbalanced libraries, such as bisulfite converted or Tetrahymena libraries, have one base at a much lower percentage than the others. This example shows a library with a low percentage of the "A" nucleotide.

BENEFITS/DRAWBACKS

Benefits

- Maximum Flexibility, fewer target specific primers needed.
- Dual barcoding, allowing for massively multiplexing of samples to occur.
- Pool multiple targets per run
- Software for demultiplexing

Drawbacks

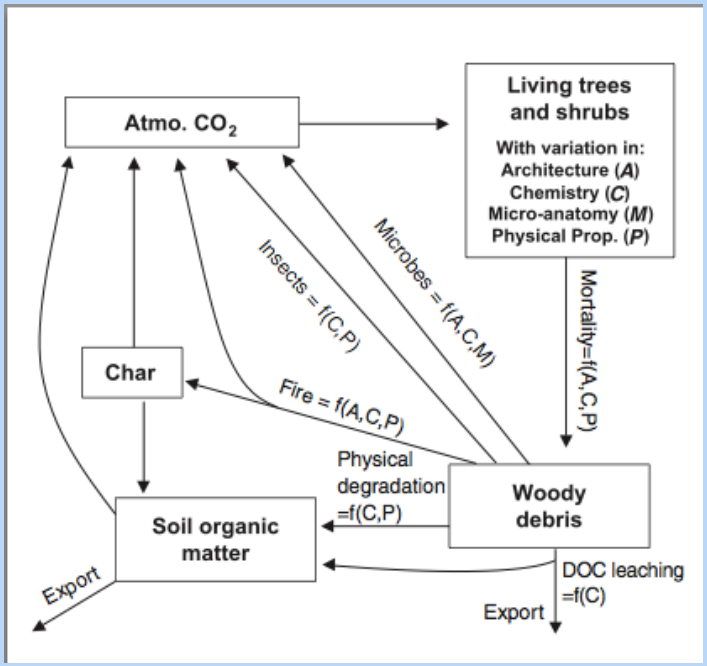
- Two – step PCR reaction
- Sequence the target specific primer

COMMUNITY PROFILING (ZANNE LAB GWU)

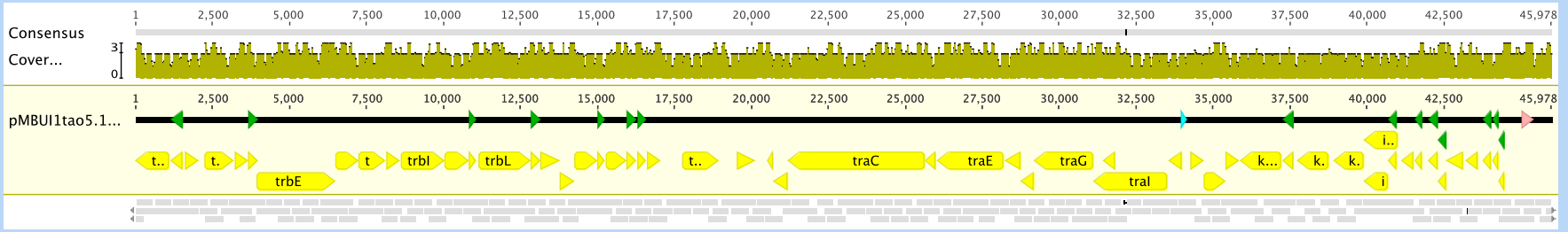
Examine differences in wood decomposition rates across 21 species in two environmental settings (ridge tops and valley bottoms) and how these rates relate to plant functional traits and their fungal and microorganism communities.

Characterize the bacterial, fungal and 'other' microorganisms (2x16S, 2xITS, and LSU)

576 samples X 5 amplicons
2,880 unique amplicon products
~6,000 reads per amplicon



SMALL GENOME SEQUENCING (TOP LAB UI)



Resequencing of the pMBUI1 plasmid from three evolution experiments in E. coli, C. necator and a combination of the two (i.e. host switching) in order to identify plasmid encoded mutations that may be involved in plasmid host range expansion and evolution

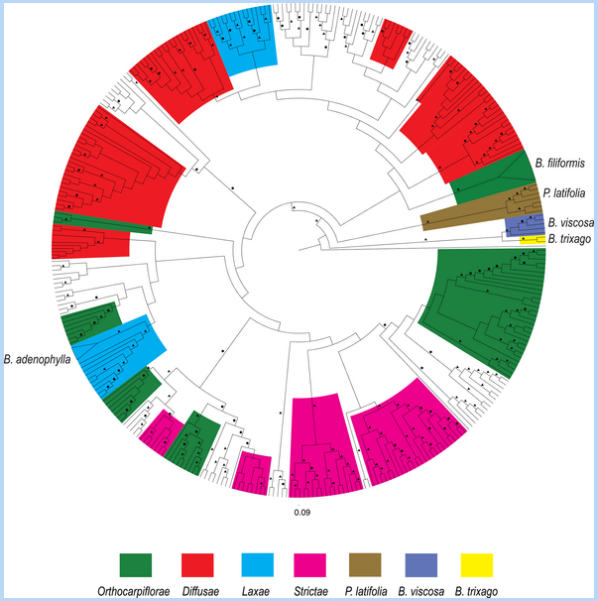
1 – Fluidigm Chip
47 samples X 192 amplicons
9,024 unique amplicon products
~900 reads per amplicon

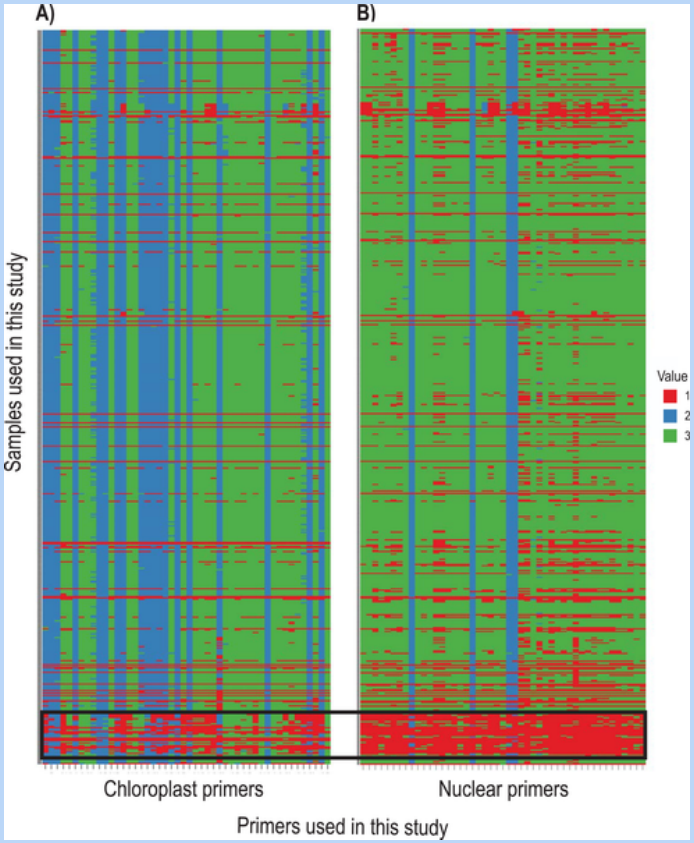
Fluidigm Chip	$710
Pooling/Primers	$220
½ MiSeq run (2x300)	$1,100
Total	$2,030
	$43/genome

PHYLOGENOMICS (TANK LAB UI)

24 – Fluidigm Chips
576 samples X 96 amplicons
55,296 unique amplicon products
~300 reads per amplicon ~$35/sample

Sequence phylogenetically informative regions (Chloroplast and Nuclear) across many, many samples





Uribe-Convers S, Settles ML, Tank DC (2016) A Phylogenomic Approach Based on PCR Target Enrichment and High Throughput Sequencing: Resolving the Diversity within the South American Species of Bartsia L. (Orobanchaceae). PLoS ONE 11(2): e0148203. doi:10.1371/journal.pone.0148203
http://journals.plos.org/plosone/article?id=info:doi/10.1371/journal.pone.0148203

FUTURE DIRECTIONS

- Include error-correcting barcodes in demultiplexing
- Identification of PCR duplicates
- Replace RDP classification with another scheme
  - Have ideas (for years) but no time
- Extend to community analysis (maybe a Shiny app). dbcAmplicons is a data reduction pipeline, not a complete analysis pipeline
- Incorporate the R genotyping pipeline into dbcAmplicons
  - Extend to inferring copy number (or ploidy levels)

DBCAMPLICONS ANALYSIS

- Approach is an DNA-result pipeline
- Requires
  - Illumina bcl2fastq
  - Python
    - Few python libs, like biom
  - Flash2 (https://github.com/dstreett/FLASH2)
  - Bowtie2 – for off-target screening
  - RDP (https://github.com/rdpstaff/RDPTools) - for classification
- dbcAmplicons
  - https://github.com/msettles/dbcAmplicons

```
mattsettles@Matts-MBP:~$ dbcAmplicons -h
usage: dbcAmplicons [-h] [--version]
                    {validate,preprocess,join,screen,classify,abundance} ...

dbcAmplicons, a python package for preprocessing of massively multiplexed,
dual barcoded Illumina Amplicons

positional arguments:
  {validate,preprocess,join,screen,classify,abundance}
                        commands
    validate            validate the sample, barcode and primer sheets
    preprocess          Preprocess four read raw amplicon data, identifying
                        barcode and primer sequence
    join                join reads using flash2
    screen              screen reads using bowtie2 and a reference sequence
                        file
    classify            classify reads using RDP generating a fixrank formated
                        file
    abundance           Generate an abundance table from a fixrank formated
                        file

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

For questions or comments, please contact Matt Settles <settles@ucdavis.edu>
mattsettles@Matts-MBP:~$
```

Stages of the Pipeline

BARCODES SHEET

#BarcodeID	Read2RC	Read3
Alpha1	TAAGGCGA	TAGATCGC
Alpha2	TAAGGCGA	CTCTCTAT
Alpha3	TAAGGCGA	TATCCTCT
Alpha4	TAAGGCGA	AGAGTAGA
Alpha5	TAAGGCGA	GTAAGGAG
Alpha6	TAAGGCGA	ACTGCATA
Alpha7	TAAGGCGA	AAGGAGTA
Alpha8	TAAGGCGA	CTAAGCCT
Alpha9	TAAGGCGA	TGAACCTT
Alpha10	TAAGGCGA	TGCTAAGT
Alpha11	TAAGGCGA	TGTTCTCT
Alpha12	TAAGGCGA	TAAGACAC
Alpha13	TAAGGCGA	CTAATCGA
Alpha14	TAAGGCGA	CTAGAACA
Alpha15	TAAGGCGA	TAAGTTCC
Alpha16	TAAGGCGA	TAGACCTA
Alpha17	TAAGGCGA	TATAGCCT
Alpha18	TAAGGCGA	ATAGAGGC
Alpha19	TAAGGCGA	CCTATCCT
Alpha20	TAAGGCGA	GGCTCTGA
Alpha21	TAAGGCGA	AGGCGAAG
Alpha22	TAAGGCGA	TAATCTTA
Alpha23	TAAGGCGA	CAGGACGT

- Tab-separated text file, 3 columns
  - Barcode name
  - sequence of read 2
  - sequence of read 3
- Orientation is important, read 2 comes off in reverse complement

PRIMER SHEET

#Read	Pair_ID	Primer_ID	Sequence
P5	16S	27F_YM1	GTAGAGTTTGATCCTGGCTCAG
P5	16S	27F_YM2	CGTAGAGTTTGATCATGGCTCAG
P5	16S	27F_YM3	ACGTAGAGTTTGATTCTGGCTCAG
P5	16S	27F_YM4	TACGTAGAGTTTGATTATGGCTCAG
P5	16S	27F_Bif	GTACGTAGGGTTCGATTCTGGCTCAG
P5	16S	27F_Bor	CGTACGTAGAGTTTGATCCTGGCTTAG
P5	16S	27F_Chl	ACGTACGTAGAATTTGATCTTGGTTCAG
P7	16S	534R_1	CCATTACCGCGGCTGCTGG
P7	16S	534R_2	GCCATTACCGCGGCTGCTGG
P7	16S	534R_3	TGCCATTACCGCGGCTGCTGG
P7	16S	534R_4	ATGCCATTACCGCGGCTGCTGG
P7	16S	534R_5	CATGCCATTACCGCGGCTGCTGG
P7	16S	534R_6	TCATGCCATTACCGCGGCTGCTGG
P7	16S	534R_7	ATCATGCCATTACCGCGGCTGCTGG

- Tab-separated text file, 4 columns
  - Read from machine
  - primer pair ID
  - individual primer ID
  - sequence of primer (without CS tag!)
- accepts IUPAC ambiguity codes

SAMPLE METADATA SHEET

- Tab-separated text file
- 4 columns required, additional metadata in additional columns will be added to biom file
- Formatting is important! (no #$%^\|{}; characters)
- can search multiple primers (comma separated or * [any]) or no primer (-)
- can make subdirectories for individual projects

SampleID	BarcodeID	PrimerPairID	ProjectID
1	Delta146	16S	subfolder/match_16S
2	Delta147	16S	subfolder/match_16S
3	Delta148	16S	subfolder/match_16S
4	Delta149	16S	subfolder/match_16S
5	Delta150	16S	subfolder/match_16S
6	Delta151	16S	subfolder/match_16S
7	Delta152	16S	subfolder/match_16S
8	Delta153	16S	subfolder/match_16S
9	Delta154	16S	subfolder/match_16S
10	Delta155	16S	subfolder/match_16S
11	Delta156	16S	subfolder/match_16S
12	Delta157	16S	subfolder/match_16S
13	Delta158	16S	subfolder/match_16S
14	Delta159	16S	subfolder/match_16S
15	Delta160	16S	subfolder/match_16S

```
mattsettles@Matts-MBP:~$ dbcAmplicons validate -h
usage: dbcAmplicons validate [-h] -B FILENAME -P FILENAME -S FILENAME [-v]
                             [--debug]

optional arguments:
  -h, --help            show this help message and exit
  -B FILENAME, --barcodes_file FILENAME
                        file with barcodes
  -P FILENAME, --primer_file FILENAME
                        file with primers
  -S FILENAME, --sample_metadata FILENAME
                        file with sample metadata
  -v, --silent          silences verbose output [default: False]
  --debug               show traceback on error
mattsettles@Matts-MBP:~$
```

dbcAmplicons validate command

- Generate the
  - Barcode, primer and sample input files
- Use dbcAmplicons validate to validate the files

HOMEWORK

Validate

SEQUENCING READ FILES

fasta files

```
>sequence1
ACCCATGATTTGCGA
```

qual files

```
>sequence1
40 40 39 39 40 39 40 40 40 40 20 20 36 39 39
```

fastq files

```
@sequence1
ACCCATGATTTGCGA
+
IIHHIHIIII55EHH
```

QUALITY SCORES

$$Q = -10 log_{10} P$$

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%

QSCORE CONVERSION

$Q_{sanger} = -10log_{10}P$ - based on probability (aka phred)
$Q_{solexa} = -10log_{10}\frac{P}{1-P}$ - based on odds

S - Sanger	Phred+33,	raw reads typically (0, 40)
X - Solexa	Solexa+64,	raw reads typically (-5, 40)
I - Illumina 1.3+	Phred+64,	raw reads typically (0, 40)
J - Illumina 1.5+	Phred+64,	raw reads typically (3, 40)
L - Illumina 1.8+	Phred+33,	raw reads typically (0, 41)

ILLUMINA READ NAMING CONVENTIONS

CASAVA 1.8 Read IDs

- @EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
  - EAS139 the unique instrument name
  - 136 the run id
  - FC706VJ the flowcell id
  - 2 flowcell lane
  - 2104 tile number within the flowcell lane
  - 15343 'x'-coordinate of the cluster within the tile
  - 197393 'y'-coordinate of the cluster within the tile
  - 1 the member of a pair, 1 or 2 (paired-end or mate-pair reads only)
  - Y Y if the read fails filter (read is bad), N otherwise
  - 18 0 when none of the control bits are on, otherwise it is an even number
  - ATCACG index sequence

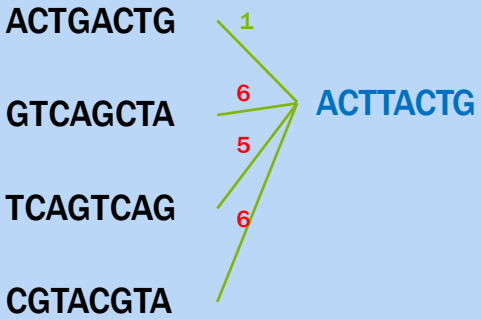
```
mattsettles@Matts-MBP:~$ dbcAmplicons preprocess -h
usage: dbcAmplicons preprocess [-h] -B FILENAME [-d BARCODEDIFF] [-P FILENAME]
                               [-D PRIMERDIFF] [-e PRIMEREND] [-f DEDUP_FLOAT]
                               [-S FILENAME] [-q MINQ] [-l MINL]
                               [-b BATCHSIZE] [-O PREFIX] [-U] [-u] [-v] -1
                               read1 [read1 ...] [-2 read2 [read2 ...]]
                               [-3 read3 [read3 ...]] [-4 read4 [read4 ...]]
                               [--test] [--keepPrimers] [--debug]

optional arguments:
  -h, --help            show this help message and exit
  -B FILENAME, --barcodes_file FILENAME
                        file with barcodes
  -d BARCODEDIFF, --barcodediff BARCODEDIFF
                        max hamming dist from barcode [default: 1]
  -P FILENAME, --primer_file FILENAME
                        file with primers
  -D PRIMERDIFF, --primerdiff PRIMERDIFF
                        max hamming dist from primer [default: 4]
  -e PRIMEREND, --primerend PRIMEREND
                        required number of matching bases at end of primer
                        [default: 4]
  -f DEDUP_FLOAT, --dedup_float DEDUP_FLOAT
                        number of bases preceeding primer used to deduplicate
                        reads [default: 0]
  -S FILENAME, --sample_metadata FILENAME
                        file with sample metadata
  -q MINQ, --minQ MINQ  trim 3' end of sequences to minQ [default: None]
  -l MINL, --minL MINL  if minQ is not None, only keep reads that are at least
                        minL length [default: 0]
  -b BATCHSIZE, --batchsize BATCHSIZE
                        batch size to process reads in [default: 100000]
  -O PREFIX, --output_prefix PREFIX
                        output file basename [default: fastq_prefix]
  -U, --output_unidentified
                        output unidentified reads [default: False]
  -u, --uncompressed    leave output files uncompressed [default: False]
  -v, --silent          silences verbose output [default: False]
  -1 read1 [read1 ...]  read1 of an amplicon fastq four file set
  -2 read2 [read2 ...]  read2 of an amplicon fastq four file set
  -3 read3 [read3 ...]  read3 of an amplicon fastq four file set
  -4 read4 [read4 ...]  read4 of an amplicon fastq four file set
  --test                exit after the first batch in order to test the inputs
  --keepPrimers         Don't cut off the primer sequence, leave it as a part
                        of the read [default: 0]
  --debug               show traceback on error
mattsettles@Matts-MBP:~$
```

dbcAmplicons preprocess command

BARCODE/PRIMER COMPARISON

Barcode Comparison



Compares each barcode to all possible barcodes and returns the best match < desired edit distance

Primer Comparison

GGCTTGGTCATTTAGAGGAAGTAA — Primer 1

TACGGCTTGGTCATTTAGAGGAAGTAA — Primer 2

CGGCTTGGTCATTTAGAGGAAGTAA — Primer 3

ACGGCTTGGTCATTTAGAGGAAGTAA — Primer 4

TACGGACTTG_TCATTTACAGGAAGTAAAAGTCGTAA — Read

Compares the beginning (primer region) of each read to all possible primers and returns the best match < specified maximimum Levenshtein disteance + final 4 exact match

PREPROCESS COMMAND

The Minimal preprocess command

Takes:

- 4 fastq files (Read 1, Index 1, Index 2, Read 2)
- Table of barcodes (indices) used in the run
- Table ofsequences and names of template-specific primers
- Table of sample names and the assigned barcodes and primers for each

```
dbcAmplicons preprocess \
  -B barcodeLookupTable.txt \
  -P primerLookupTable.txt \
  -S sampleLookupTable.txt \
  -1 data_R1_001.fastq.gz
```

STANDARD OUTPUT

```
alida@alida:~/Documents/Idaho/IBEST/dbcAmplicons workshops/tests$ dbcAmplicons preprocess -b 10000 -B barcodeLookupTable.txt
-P primerLookupTable.txt -S sampleLookupTable.txt -1 Test100K_16S_R1_001.fastq.gz
barcode table length: 864
primer table length P5 Primer Sequences:7, P7 Primer Sequences:7
sample table length: 130, and 3 projects.
processed 10000 total reads, 6859.0 Reads/second, 4901 identified reads(49.0%), 5099 unidentified reads
processed 20000 total reads, 6686.0 Reads/second, 9948 identified reads(49.7%), 10052 unidentified reads
processed 30000 total reads, 6656.0 Reads/second, 14981 identified reads(49.9%), 15019 unidentified reads
processed 40000 total reads, 6666.0 Reads/second, 20041 identified reads(50.1%), 19959 unidentified reads
processed 50000 total reads, 6641.0 Reads/second, 25166 identified reads(50.3%), 24834 unidentified reads
processed 60000 total reads, 6640.0 Reads/second, 30166 identified reads(50.3%), 29834 unidentified reads
processed 70000 total reads, 6593.0 Reads/second, 35203 identified reads(50.3%), 34797 unidentified reads
processed 80000 total reads, 6584.0 Reads/second, 40309 identified reads(50.4%), 39691 unidentified reads
processed 90000 total reads, 6580.0 Reads/second, 45322 identified reads(50.4%), 44678 unidentified reads
processed 100000 total reads, 6554.0 Reads/second, 50353 identified reads(50.4%), 49647 unidentified reads
100000 reads processed in 0.25 minutes, 50353 (50.4%) identified

29914 reads (29.9% of total run) found for project      match_twoprimer
2195 reads (2.2% of total run) found for project        subfolder/match_16S
18244 reads (18.2% of total run) found for project      match_wildcard
Cleaning up.
```

OUTPUTS: IDENTIFIED_BARCODES.TXT

Barcode	X16S	None
Delta146	44	0
Delta147	44	0
Delta148	44	0
Delta149	63	0
Delta150	50	0
Delta151	29	0
Delta152	49	0
Delta153	60	0
Delta154	32	0
Delta155	95	0
Delta156	63	0
Delta157	110	0
Delta158	67	0
Delta159	74	0
Delta160	14	0
Delta161	1	0
Delta162	130	0
Delta163	171	0
Delta164	100	0
Delta165	115	0
Delta166	118	0

- Tab-separated text file listing all barcodes that had at least 1 read
- Rows list number of reads per barcode
- Columns list the primers provided

OUTPUTS: DIRECTORIES AND FILES

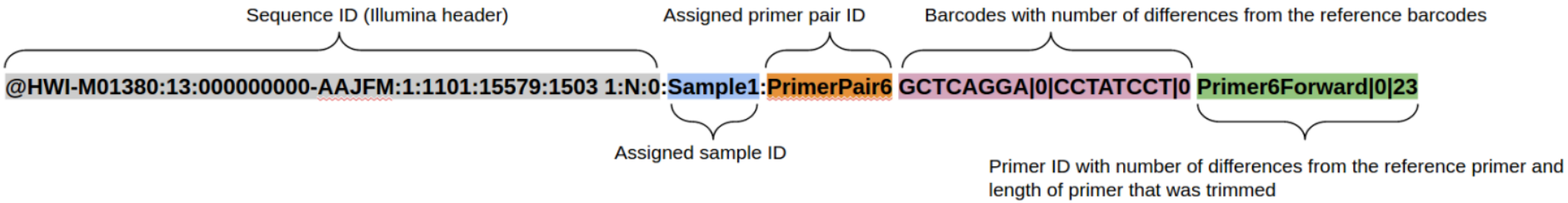
```
alida  23K Feb 11 11:11 barcodeLookupTable.txt
alida 2.0K Feb 11 13:04 Identified_Barcodes.txt
alida 3.9M Feb 11 13:04 match_twoprimer_R1.fastq.gz
alida 4.1M Feb 11 13:04 match_twoprimer_R2.fastq.gz
alida 2.5M Feb 11 13:04 match_wildcard_R1.fastq.gz
alida 2.6M Feb 11 13:04 match_wildcard_R2.fastq.gz
alida  579 Feb 11 11:34 primerLookupTable.txt
alida 5.2K Feb 11 13:03 sampleLookupTable.txt
alida 4.0K Feb 11 13:03 subfolder/
alida 3.3K Feb 11 12:02 table.txt
alida  15M Feb 11 11:11 Test100K_16S_R1_001.fastq.gz
alida 794K Feb 11 11:11 Test100K_16S_R2_001.fastq.gz
alida 893K Feb 11 11:11 Test100K_16S_R3_001.fastq.gz
alida  16M Feb 11 11:11 Test100K_16S_R4_001.fastq.gz
```

- ProjectID column from the sample sheet determines the resulting output structure
- A “folder/name” structure creates a directory with the named files within it

```
alida@alida:~/Documents/Idaho/IBEST/dbcAmplicons workshops/tests$ ll subfolder/
total 620K
drwxrwxr-x 2 alida alida 4.0K Feb 11 13:17 ./
drwxrwxr-x 3 alida alida 4.0K Feb 11 13:28 ../
-rw-rw-r-- 1 alida alida 295K Feb 11 13:17 match_16S_R1.fastq.gz
-rw-rw-r-- 1 alida alida 314K Feb 11 13:17 match_16S_R2.fastq.gz
```

OUTPUTS: FASTQ FILES

Header format of all identified sequences:




```
@HWI-M01380:50:000000000-A641U:1:1101:17127:1556 1:N:0:Sample97:16S GTCGTGAT|0|TAAGTTCC|0 27F_Bif|0|26
GATGAACGCTAGCTACAGGCTTAACACATGCAAGTCGAGGGGCATCAGGAAGAAAGCTTGCTTTCTTTGCTGGCGACCGGCGCACGGGTGAGTAACACGTATC
```

SUMMARY OF PREPROCESS

- Read 1 and Read 2 files for each unique Project_ID
- Each sequence identified by barcode, primer and/or by sample name
- Table of identified barcodes from the run
- A number of parameters available, however defaults work most of the time

- Run
  - dbcAmplicons preprocess
- Try --test to play with a few parameters

HOMEWORK

preprocess

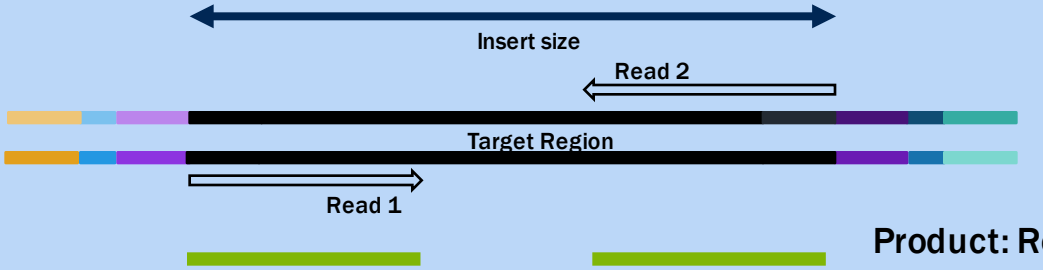
NEXT: OVERLAPPING READS

- dbcAmplicons join builds off of preprocess

- Uses FLASH2, a modified version of flash (Fast Length Adjustment of SHort reads, http://ccb.jhu.edu/software/FLASH/) to merge paired-end reads.
- Modification include:
  - Performs complete overlaps with adapter trimming
  - Allows for different sized reads (after cutting primer off)
  - Discards reads with > 50% Q of 2 (changed to Q10 in dbcAmplicons), which are indicative of adapter/primer dimers

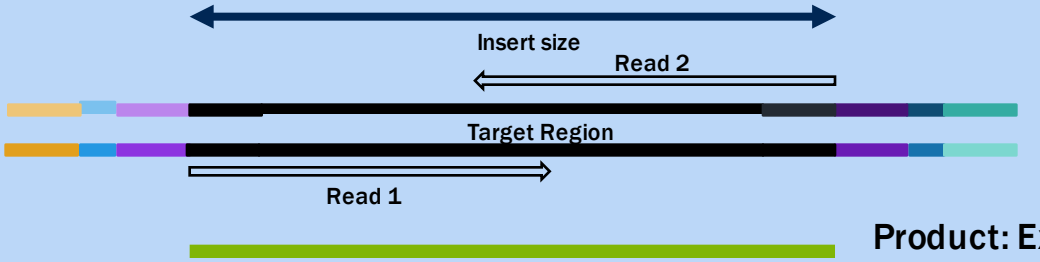
FLASH2 – OVERLAPPING OF READS

Insert size > length of the number of cycles



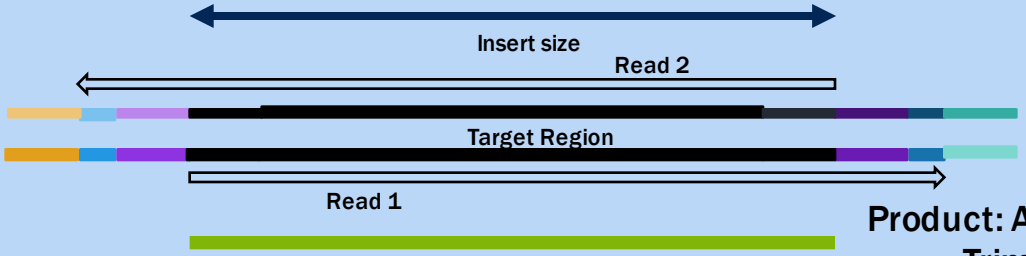
Product: Read Pair

Insert size < length of the number of cycles (10bp min)



Product: Extended Single

Insert size < length of the read length



Product: Adapter Trimmed Single

```
mattsettles@Matts-MBP:~$ dbcAmplicons join -h
usage: dbcAmplicons join [-h] [-O PREFIX] [-u] [-x NUM] [-t NTHREADS] [-v] -1
                         read1 [-2 read2]

optional arguments:
  -h, --help            show this help message and exit
  -O PREFIX, --output_path PREFIX
                        path for output files [default: joined]
  -u, --uncompressed    leave output files uncompressed [default: False]
  -x NUM, --max-mismatch-density NUM
                        Maximum allowed ratio between the number of mismatched
                        base pairs and the overlap length. Two reads will not
                        be combined with a given overlap if that overlap
                        results in a mismatched base density higher than this
                        value. Note: Any occurrence of an "N" in either read
                        is ignored and not counted towards the mismatches or
                        overlap length. [default:0.25]
  -t NTHREADS, --threads NTHREADS
                        Set the number of worker threads. [default: 1]
  -v, --verbose         verbose output [default: True]
  -1 read1              read1 of an amplicon fastq (or fastq.gz) two file set
  -2 read2              read2 of an amplicon fastq (or fastq.gz) two file set
mattsettles@Matts-MBP:~$
```

dbcAmplicons join parameters

EXECUTING DBCAMPLICONS JOIN

```
dbcAmplicons join -O join/match -1 match_16S_R1.fastq.gz -2
  match_16S_R2.fastq.gz -v
```

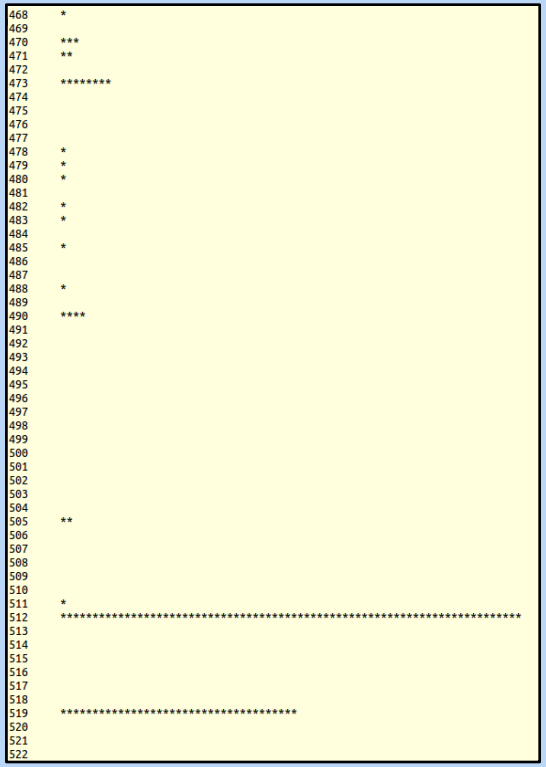
```
alida@alida:~/Documents/Idaho/IBEST/dbcAmplicons workshops/tests/subfolder$ dbcAmplicons join
-O join/match -1 match_16S_R1.fastq.gz -2 match_16S_R2.fastq.gz -v
Using Flash_version:v2.2.00
Min_overlap:10
Min_overlap_outie:35
Max_overlap:600
Max_mismatch_density:0.250000
Allow_"outie"_pairs:true
Cap_mismatch_quals:false
Combiner_threads:1
Input_format:FASTQ, phred_offset=33
Output_format:FASTQ, phred_offset=33, gzip
Total_pairs:2195
Combined_pairs:1893
Innie_pairs:1847 (97.57% of combined)
Outie_pairs:46 (2.43% of combined)
Uncombined_pairs:302
Percent_combined:86.24%
```

DBCAMPLICONS JOIN OUTPUTS

```
alida 451K Feb 11 13:55 match.extendedFrags.fastq.gz
alida  695 Feb 11 13:55 match.hist
alida  469 Feb 11 13:55 match.hist.innie
alida 2.6K Feb 11 13:55 match.histogram
alida 1.5K Feb 11 13:55 match.histogram.innie
alida 1.9K Feb 11 13:55 match.histogram.outie
alida  226 Feb 11 13:55 match.hist.outie
alida  54K Feb 11 13:55 match.notCombined_1.fastq.gz
alida  51K Feb 11 13:55 match.notCombined_2.fastq.gz
```

- 3 fastq files: 1 of overlapped pairs [extendedFrags], 2 remaining single [notCombined_1/2] reads that didn't overlap
- Histogram showing the length of reads that overlapped and the frequency
  Ignore "innie" and "outie" designations for now

HISTOGRAM



- "more match.histogram" will scroll through the file showing the lengths (symbolic)
- "match.hist" gives the number of sequences that fall into each numerical category
- Useful for determining the lengths of the amplicons that were sequenced

- Using dbcAmplicons
  - Join the preprocessed reads
- Modify the parameter
  - -x see how the results change

HOMEWORK

Join

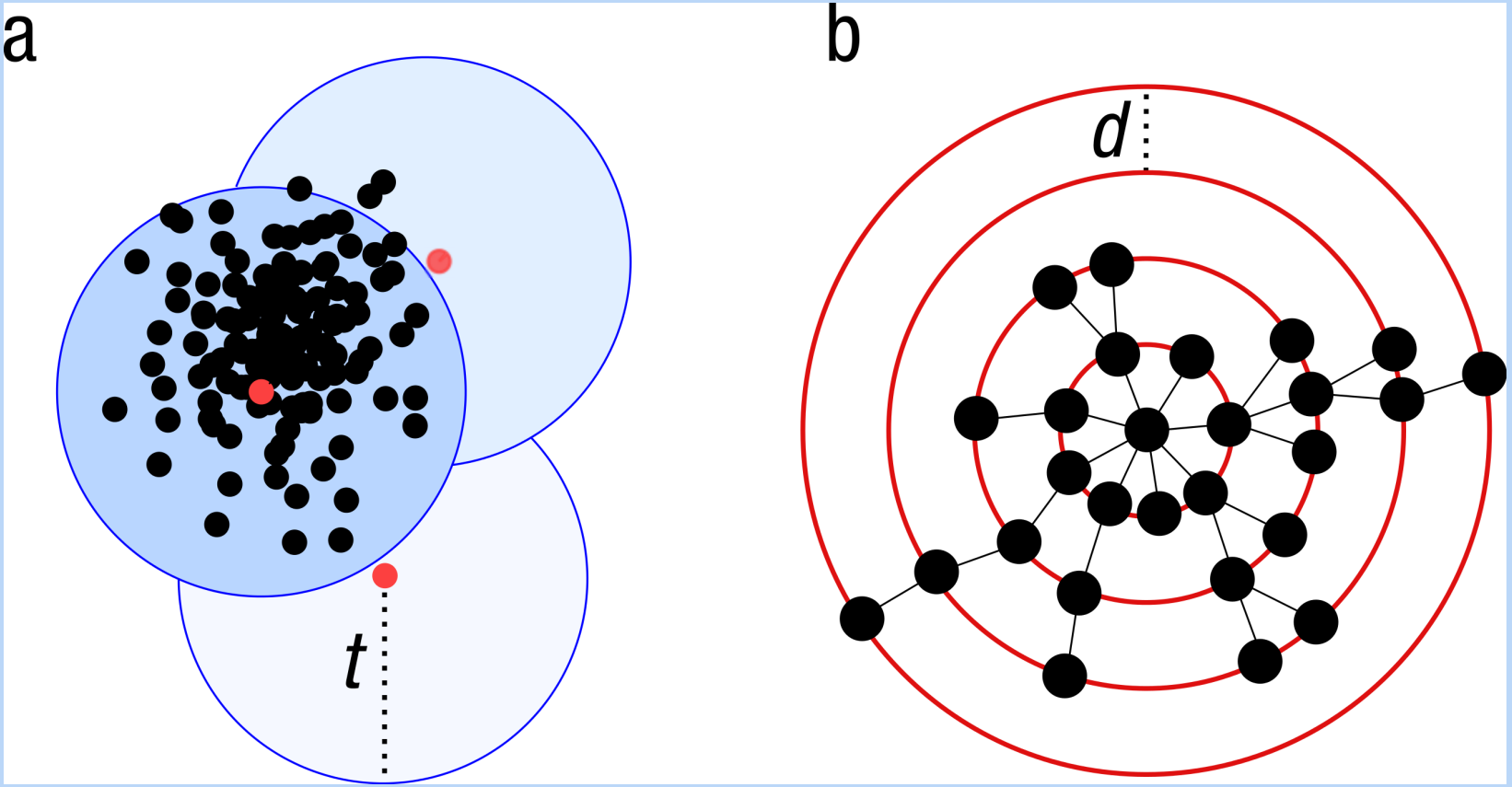
DBCAMPLICONS CLASSIFY

- Uses the RDP (Ribosomal Database Project) classifier for bacterial and archaeal 16S, fungal LSU, ITS warcup/unite databases.
- Classifies sequences to the closest taxonomic reference provides a bootstrap score for reliability
- Concatenates Paired end reads

DIRECT CLASSIFICATION VS CLUSTERING

- Clustering – "Because of the increasing sizes of today's amplicon datasets, fast and greedy *de novo* clustering heuristics are the preferred and only practical approach to produce OTUs". Shared steps in these current algorithms are:
  1. An amplicon is drawn out of the amplicon pool and becomes the center of a new OTU (centroid selection)
  2. This centroid is then compared to all other amplicons remaining in the pool.
  3. Amplicons for which the distance is within a global clustering threshold, *t (e.g. 3%)*, to the centroid are moved from the pool to the OUT
  4. The OTU is then closed. These steps are repeated as long as amplicons remain in the pool.
- Problems as I see them, no biological rational to any of the clustering parameters, dependent on ordering, does not consider sequencing errors.

CLASSIFICATION



COMPARING CLUSTERING ALGORITHMS

	Clone43				Simclone15_1			
	Expected OTUs	Inferred* OTUs(2%)	inferred OTUs(3%)	inferred OTUs(4%)	Expected OTUs	inferred OTUs(2%)	inferred OTUs(3%)	inferred
Mothur	43	1882	720	369	15	63	41	20
Muscle+Mothur		2478	1418	784		117	89	54
ESPRIT		4474	4397	1733		131	131	55
ESPRIT-Tree		2301	1096	279		96	29	16
SLP		286	245	227		17	17	15
Uclust		2177	1883	597		80	75	51
CD-HIT		1473	1464	481		50	49	32
DNAClust		3768	3658	1103		239	225	53
GramCluster		2119	2071	2071		70	70	70
CROP		339	133	62		21	15	15

*: all the listed numbers of OTU are the average numbers over xx simulations.
doi:10.1371/journal.pone.0070837.t002

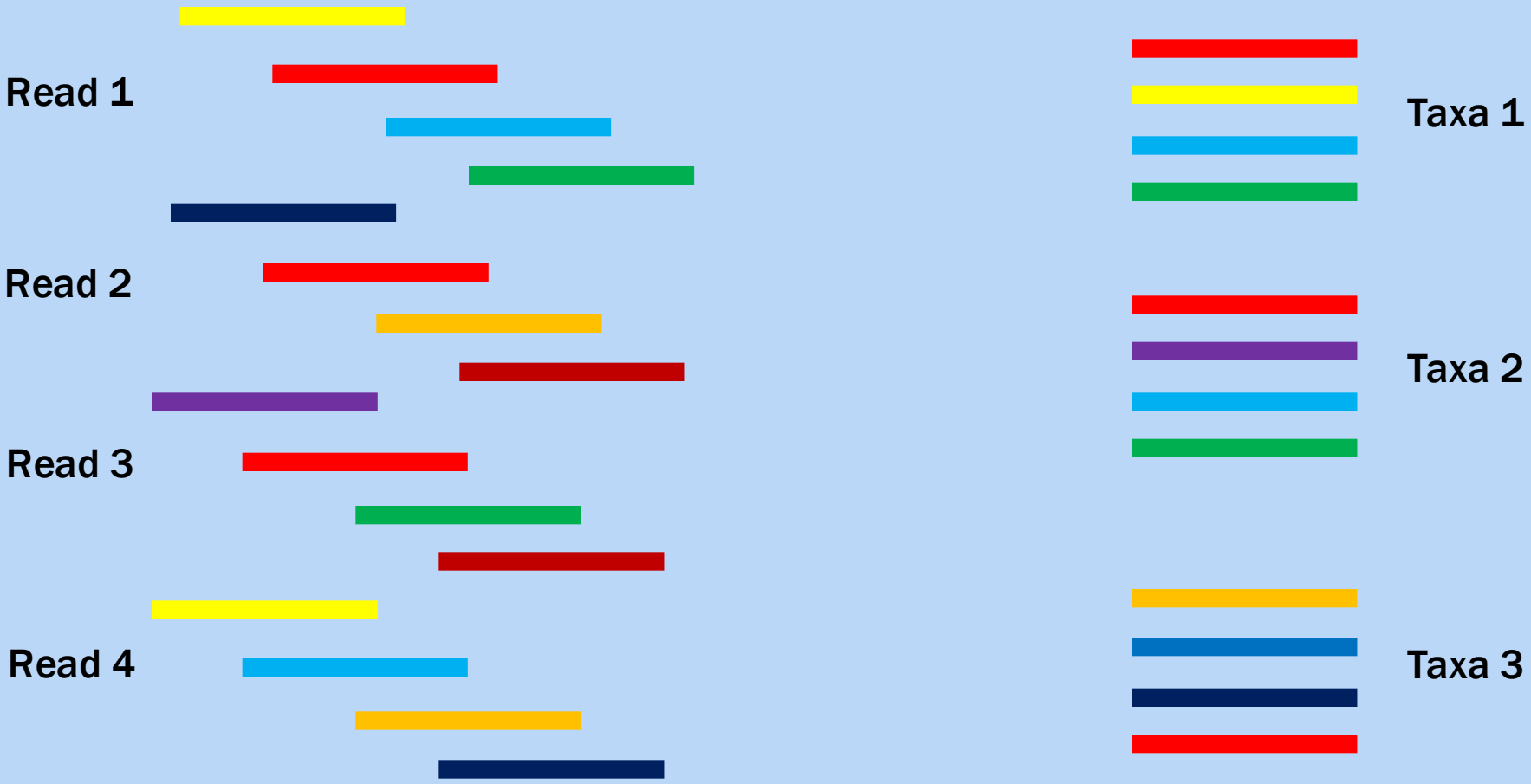
Chen W, Zhang CK, Cheng Y, Zhang S, Zhao H (2013) A Comparison of Methods for Clustering 16S rRNA Sequences into OTUs. PLoS ONE 8(8): e70837. doi:10.1371/journal.pone.0070837
http://journals.plos.org/plosone/article?id=info:doi/10.1371/journal.pone.0070837

PLOS | ONE

DIRECT CLASSIFICATION

- Ribosomal Database Project (RDP) - naïve Bayesian Classifier
  - Compares each read to a database
    - Database is updates periodically
  - Compares by k-mers (15 mers)
  - 100 bootstraps to establish confidence in result
- Order does not matter, no 3% !
- Drawbacks
  - Accepts only fasta files
  - Can be slow
  - Down to genus only
  - Kmers are based on whole 16s

DIRECT CLASSIFICATION



```
attsettles@Matts-MBP:~$ dbcAmplicons classify -h
sage: dbcAmplicons classify [-h] [-b BATCHSIZE] [-q MINQ] [-l MINL]
                            [-p PROCS] --rdpPath PATH [-g <arg>] [-t TRAIN]
                            [-O outputPrefix] [-1 read1 [read1 ...]]
                            [-2 read2 [read2 ...]] [-U single [single ...]]
                            [-v] [--debug]

ptional arguments:
 -h, --help            show this help message and exit
 -b BATCHSIZE, --batchsize BATCHSIZE
                       batch size to process reads in [default: 100000]
 -q MINQ, --minQ MINQ  trim 3' end of sequences to minQ (paired-end reads
                       only) [default: None]
 -l MINL, --minL MINL  if minQ is not None, only keep reads that are at least
                       minL length (paired-end reads only) [default: 0]
 -p PROCS, --processors PROCS
                       number of processors to use, [default: 1]
 --rdpPath PATH        path to the RDP classifier
 -g <arg>, --gene <arg>
                       16srrna, fungallsu, fungalits_warcup, or
                       fungalits_unite [default: 16srrna]
 -t TRAIN, --train TRAIN
                       RDP property file containing the mapping of the
                       training files for a custom RDP training set, if not
                       using the default.
 -O outputPrefix, --output_path outputPrefix
                       path for output files [default: classify]
 -1 read1 [read1 ...]  read1 of an amplicon fastq two file set
 -2 read2 [read2 ...]  read2 of an amplicon fastq two file set
 -U single [single ...]
                       single-end amplicon, typically from joined paired
                       reads
 -v, --silent          silences verbose output [default: False]
 --debug               show traceback on error
attsettles@Matts-MBP:~$
```

dbcAmplicons classify command

USING DBCAMPLICONS CLASSIFY

```
dbcAmplicons classify -b 7500
    -p 1
    --rdpPath RDPTools/classifier.jar
    -O classify/classified
    -U join/match.extendedFrags.fastq.gz
    -1 join/match.notCombined_1.fastq.gz
    -2 join/match.notCombined_2.fastq.gz
```

USING DBCAMPLICONS CLASSIFY

```
Starting rdp for file classify/classified.1893.fasta
Finished processing classify/classified.1893.fasta in 0.16 minutes
Starting rdp for file classify/classified.2195.fasta
Finished processing classify/classified.2195.fasta in 0.08 minutes
Combining temporary files
2195 reads processed in 0.25 minutes
Cleaning up.
alida@alida:~/Documents/Idaho/IBEST/dbcAmplicons workshops/tests/subfolder$ ll classify/
total 448K
drwxrwxr-x 2 alida alida 4.0K Feb 11 15:21 ./
drwxrwxr-x 4 alida alida 4.0K Feb 11 14:46 ../
-rw-rw-r-- 1 alida alida 440K Feb 11 15:21 classified.fixrank
```

FIXRANK FILES

```
HWI-M01380:50:000000000-A641U:1:1101:18594:1568|Sample26:16S                Bacteria
   domain 1.0        "Actinobacteria"        phylum     1.0        Actinobacteria          class
          1.0        Actinomycetales         order      1.0        Actinomycetaceae
          family     1.0        Varibaculum  genus      1.0
HWI-M01380:50:000000000-A641U:1:1101:17948:1574|Sample28:16S                Bacteria
   domain 1.0        "Actinobacteria"        phylum     0.96       Actinobacteria          class
          0.96       Bifidobacteriales       order      0.9        Bifidobacteriaceae
          family     0.9        Aeriscardovia            genus      0.23
HWI-M01380:50:000000000-A641U:1:1101:14301:1587|Sample12:16S                Bacteria domain
          1.0        Firmicutes  phylum      0.98       Bacilli    class
          0.97Lactobacillales    order       0.86       Lactobacillaceae        family     0.81
          Lactobacillus          genus       0.61
HWI-M01380:50:000000000-A641U:1:1101:18207:1603|Sample19:16S                Bacteria
   domain 1.0        Firmicutes  phylum      1.0        Bacilli    class       1.0
          Lactobacillales        order       1.0        Lactobacillaceae        family     1.0
          Lactobacillus          genus       1.0
HWI-M01380:50:000000000-A641U:1:1101:17585:1630|Sample17:16S                Bacteria
   domain 1.0        Firmicutes  phylum      1.0        Bacilli    class       1.0
          Lactobacillales        order       1.0        Lactobacillaceae        family     1.0
          Lactobacillus          genus       1.0
HWI-M01380:50:000000000-A641U:1:1101:22144:1652|Sample26:16S                Bacteria
   domain 1.0        "Actinobacteria"        phylum     1.0        Actinobacteria          class
          1.0        Actinomycetales         order      1.0        Actinomycetaceae
          family     1.0        Actinomyces  genus      1.0
```

- Classify your sequences using dbcAmplicons classify
- View the fixrank file with more

HOMEWORK

Classify

```
mattsettles@Matts-MBP:~$ dbcAmplicons abundance -h
usage: dbcAmplicons abundance [-h] [-r <arg>] [-t VALUE] [-m VALUE] [-M VALUE]
                              [-S FILE] [-O FILE_PREFIX] -F FILE [FILE ...]
                              [-b] [-v] [--debug]

optional arguments:
  -h, --help            show this help message and exit
  -r <arg>, --rank <arg>
                        taxonomic rank to build table from, allowable values
                        are (domain, phylum, class, order, family, genus, and
                        species{if performed}, [default: genus]
  -t VALUE, --threshold VALUE
                        Threshold bootstrap value to use for assignment, first
                        taxon level greater than threshold
  -m VALUE, --minsize VALUE
                        Min size of amplicon to consider
  -M VALUE, --maxsize VALUE
                        Max size of amplicon to consider
  -S FILE, --sample_metadata FILE
                        file with sample metadata
  -O FILE_PREFIX, --output_path FILE_PREFIX
                        path for output files [default: table]
  -F FILE [FILE ...]    fixrank formated classification file generated by
                        classify
  -b, --biom            output biom formatted file [default: False]
  -v, --silent          verbose output [default: False]
  --debug               show traceback on error
mattsettles@Matts-MBP:~$
```

dbcAmplicons abundance

GENERATING ABUNDANCE TABLES

- Abundance tables
  - Rows are taxa
  - Columns are samples
  - Counts of the number of amplicons for each taxa/samples
- Proportions tables
  - Same as abundance but each cell is the proportion of amplicons
- Biom file (Biological Observation Matrix)
  - JSON file format for microbiome files
  - http://biom-format.org
  - Abundance tables are 0 heavy, a biom file removes the 0's as well as stores extra metadata

DBCAMPLICONS ABUNDANCE

```
dbcAmplicons abundance -F
  classify/classified.fixrank
```

Outputs 3 files:
table.abundance.txt
table.proportions.txt
table.taxa_counts.txt

TABLE.ABUNDANCE.TXT

	A	B	C	D	E	F	G	H	I	J	K	L
1	Taxon_Name	Level	MeanBootstraptValue	Sample1	Sample10	Sample11	Sample12	Sample13	Sample14	Sample15	Sample16	Sample17
2	Acidovorax	genus	0.973	0	0	0	0	0	0	0	0	0
3	Actinobaculum	genus	1	0	0	0	0	0	0	0	0	0
4	Actinomyces	genus	0.988	0	0	0	0	0	0	0	0	0
5	Actinomycetaceae	family	0.83	0	0	0	0	0	0	0	0	0
6	Actinomycetales	order	0.923	0	0	0	0	0	0	0	0	0
7	Aeriscardovia	genus	0.68	0	0	0	0	0	0	0	0	0
8	Aerococcus	genus	1	0	0	0	0	0	1	0	0	0
9	Alloscardovia	genus	1	0	3	0	0	0	0	0	0	0
10	Lactobacillus	genus	0.977	41	75	55	96	56	66	12	0	114
11	Anaerococcus	genus	0.946	0	1	0	0	0	1	0	0	0
12	Anaerosphaera	genus	0.582	0	0	0	0	0	0	0	0	0
13	Archaea	domain	0.587	0	0	0	0	0	0	0	0	0
14	Atopobium	genus	1	0	0	0	0	0	0	0	0	0
15	Bacilli	class	0.606	0	1	1	2	1	0	0	0	5
16	Bacillus	genus	0.9	0	0	0	0	0	0	0	0	0

Can open file in spreadsheet; lists lowest taxonomically classified level for each sample by read count

TABLE.PROPORTIONS.TXT

	A	B	C	D	E	F	G
1	Taxon_Name	Level	MeanBootstraptValue	Sample1	Sample10	Sample11	Sample12
2	Acidovorax	genus	0.973	0	0	0	0
3	Actinobaculum	genus	1	0	0	0	0
4	Actinomyces	genus	0.988	0	0	0	0
5	Actinomycetaceae	family	0.83	0	0	0	0
6	Actinomycetales	order	0.923	0	0	0	0
7	Aeriscardovia	genus	0.68	0	0	0	0
8	Aerococcus	genus	1	0	0	0	0
9	Alloscardovia	genus	1	0	0.0315789474	0	0
10	Anaerococcus	genus	0.946	0	0.0105263158	0	0
11	Anaerosphaera	genus	0.582	0	0	0	0
12	Archaea	domain	0.587	0	0	0	0
13	Atopobium	genus	1	0	0	0	0
14	Bacilli	class	0.606	0	0.0105263158	0.0158730159	0.0181818182
15	Bacillus	genus	0.9	0	0	0	0
16	Bacteria	domain	0.885	0.0227272727	0	0.0158730159	0.0090909091
17	Bacteroidales	order	0.57	0	0	0	0
18	Bacteroidetes	phylum	0.85	0	0	0	0
19	Bifidobacteriaceae	family	0.905	0	0	0	0
20	Bifidobacterium	genus	0.965	0	0	0	0
21	Blautia	genus	0.99	0.0227272727	0	0	0
22	Burkholderiales	order	0.76	0	0	0	0

Similar to abundance.txt, but instead lists proportions of each taxonomic level found per sample

TABLE.TAXA_COUNTS.TXT

	A	B
1	Taxon_Name	Count
2	Klebsiella|genus	21
3	Aerococcus|genus	1
4	Porphyromonadaceae|family	2
5	Clostridiales|order	2
6	Varibaculum|genus	7
7	Turicella|genus	1
8	Burkholderiales|order	1
9	Paralactobacillus|genus	4
10	Phyllobacteriaceae|family	1
11	Lactobacillales|order	68
12	Bacillus|genus	2
13	Peptoniphilus|genus	11
14	Clostridium XIX|genus	2
15	Bacteria|domain	95
16	Parascardovia|genus	1
17	Thermoprote|class	1
18	Leclercia|genus	3
19	Bacilli|class	19
20	Actinobaculum|genus	1
21	Propionibacterium|genus	8
22	Bacteroidales|order	1
23	Acidovorax|genus	8
24	Mobiluncus|genus	1
25	Dechloromonas|genus	2
26	Clostridiales_Incertae Sedis XI|family	1
27	Lachnospiraceae|family	1
28	Unknown|Root	38
29	Lactobacillaceae|family	28

Summary of each classification and the number of counts for that level across all samples

- Use dbcAmplicons abundance to modify abundance tables
  - Threshold
  - Rank
- Generate a biom formatted file
- Use the python biom package to convert the biom file to a new tab delimited file

HOMEWORK

Abundance tables

CONVERT2READTO4READ

Back-converts a set of 2 Illumina paired-end files that have already been preprocessed by *dbcAmplicons* into 4 reads, splitting the 2 barcodes into separate reads.

Useful for reprocessing datasets and analyzing barcode counts.

Can be used to back covert non-dbcAmplicons reads to acceptable by dbcAmplicons

SPLITREADSBYSAMPLE
(UPLOAD TO SRA)

Takes processed reads from *dbcAmplicons* and bins each read by its sample ID. If no sample IDs are assigned, reads will be split by individual barcode.

Each sample receives a Read 1 and Read 2.

Designed for easy upload to the SRA