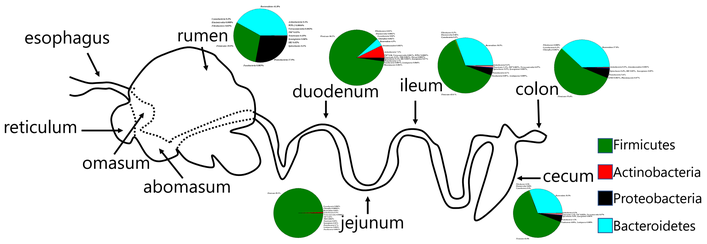
Pipelines, Workflows and Virtualization to Build Institutional Informatics Capacity

PRESENTER: **Aaron** Dickey

THE WORKFLOW

Role of Alimentary Tract Microbiota in Cattle Feed Efficiency

- The senior researcher (Harvey Freetly) works through a published bioinformatic template using 16S data from a single tissue. The bioinformatician writes scripts as requested to add new analyses and extract customized outputs.
- The final workflow is applied to each tissue.



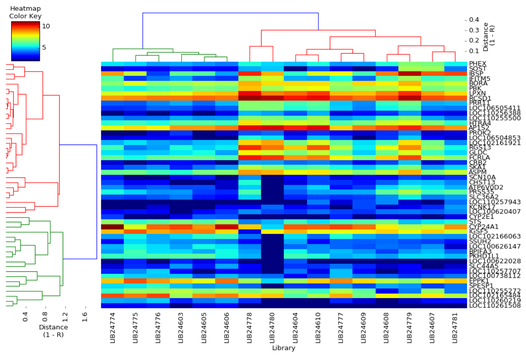
- Link to Freetly et al. 2019 (full manuscript under review)

THE PIPELINE

Olfactory Epithelium Transcriptome of Normal Cycling and Acyclic Gilts

- The pipeline is a bash script which accesses RNASeq data on the server and processes the data using software installed on the server. The last step produces gene count tables for each library.
- The senior researcher (Dan Nonneman) analyzes the count data.

Few Differentially Expressed Genes in Prepubertal Gilts vs Controls



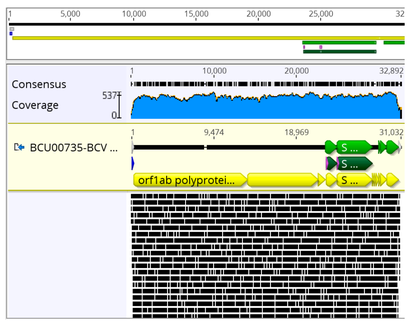
- Link to Nonneman et al. 2019

THE VIRTUALIZED PIPELINE

Viral Metagenomics of Bovine Respiratory Disease

- The pipeline is a bash script to access sequence data on the server and process it using software in a virtual container.
- The senior researcher (Aspen Workman) validates viral 'hits' in Geneious™.

Reads mapped to a Bovine Coronavirus reference



Strengths of Workflows:

- Customizable
- Real-time control

Strengths of Pipelines:

- Rapid run-time
- Continual inputs not required

Strengths of Virtualization:

- Server administrator does not need to maintain software/dependencies
- Software in Singularity containers can be run by non-root users

Conclusions:

Individual user and project needs and comfort level will determine the best approach. All can be used by the staff bioinformatician to efficiently serve the needs of multiple Principal investigators.

This photo by Tomas Castelazo is licensed under CC BY-SA

Pipelines offer **shorter** run time while **workflows** allow for **custom** statistics and outputs.

Take a picture to download the full poster

aaron.dickey@usda.gov

ABSTRACT

Volmers *et al.*, 2017 defined the "bioinformatics middle class" as being comprised of competent and informed users rather than tool developers. Increasingly, these middle class bioinformaticians are being employed in supporting roles where they can collaborate to advance the research programs of multiple principal investigators across an institution to access, manipulate and analyze large datasets. The daily routine of the middle class bioinformatician may vary with both individual strengths as well as client and institutional needs but will often comprise a variety of activities. Such activities might include delivering trainings, scripting, developing pipelines and curating databases. Middle class bioinformaticians occupy a similar role to departmental statisticians and may face some of the same professional challenges; among them, maintaining an active publication record.

A data analysis pipeline is an end-to-end multi-step data management solution where the data is not manually inspected between each step. In contrast, a user interacts with the data at each step of a data analysis workflow. Either methodological class can take advantage of computer platform virtualization. The purpose of this presentation is to summarize multiple research projects where the bioinformatic support was in the form of a pipeline or workflow with the goal of highlighting differences between these two approaches. Use of virtualization in different projects is also highlighted. Workflows have greater real-time flexibility for integrating custom statistics and outputs whereas pipelines offer greater speed.

THE WORKFLOW

- Template for 16S microbiome analysis in R (Callahan *et al.*, 2016).
- Work through template for a single tissue, discarding analyses that don't apply. Bioinformatician writes custom R scripts to add follow-up analyses and extract customized outputs from R objects as requested.
- The final workflow is applied to each tissue type.

Example R Script for Follow-up Analysis

```r
#The two_level_taxa_sum function takes a prevalence dataframe,
#two taxonomic levels, and the value of the higher taxonomic level
#as arguments and prints a two column dataframe to the console of
#total read abundance of all OTUs by the lower level when limiting
#the higher taxonomic level by the specified value. Classes
#within Firmicutes
two_level_taxa_sum <- function(p = prevdf, L_one = Phylum,
                               L_one_id = "Firmicutes",
                               L_two = "Class") {
  l1 <- substitute(L_one)
  p2 = subset(p, eval(l1) == L_one_id)
  plyr::ddply(p2, L_two, function(df1){sum(df1$TotalAbundance)}) }

#default use case... by Class within Firmicutes
two_level_taxa_sum()
#by Genus within Clostridiales
two_level_taxa_sum(prevdf, Order, "Clostridiales", "Genus")
```

THE PIPELINE

- The RNAseq pipeline consists of a bash script which accesses raw single-end reads on the USMARC server by run date, groups across lanes (Illumina NextSeq), trims adapters with BBDuk, then aligns reads to the genome and extracts gene count tables for each library with STAR.
- Post-pipeline, the data analyzed with DESeq2.
  - 18,484 genes expressed at base mean > 5
  - Only 4 differentially expressed genes (DEGs) between gilts at the same ovarian stage with different reproductive status (Behavioral Anestrus and Prepubertal) vs normal cycling gilts
  - 2497 DEGs in different epithelial tissues: Luteal (1351 elevated) vs. Follicular (1146 elevated) olfactory epithelium indicating tissue expression under hormonal control

A Portion of the RNA-Seq Pipeline Bash Script

```
#Trim the adapters off the nextseq reads for each library
cd ${grouped} && for lib in ${library[*]}
do echo "Adapter Trimming sample ${lib}"
sh ${home}/bbmap/bbduk.sh \
    in1=${lib}_R1.fq.gz \
    in2=${lib}_R2.fq.gz \
    out1=${bbduk}${lib}_R1.fq.gz \
    out2=${bbduk}${lib}_R2.fq.gz \
    ref=${home}/bbmap/resources/adapters.fa \
    ktrim=r k=23 mink=11 hdist=1 pigz=f unpigz=f overwrite=true tpe tbo
    done
#Align reads to the pig genome with STAR
cd ${bbduk} && for lib in ${library[*]}
do echo "Processing sample ${lib}"
STAR --runThreadN 32 --genomeDir ${db} \
    --readFilesIn ${lib}_R1.fq.gz ${lib}_R2.fq.gz \
    --readFilesCommand zcat --outFilterIntronMotifs RemoveNoncanonical \
    --quantMode TranscriptomeSAM GeneCounts \
    --outSAMtype BAM SortedByCoordinate \
    --outFileNamePrefix ${STAR}${lib}/
    done
```

THE VIRTUALIZED PIPELINE

- The pipeline consists of a bash script which accesses raw Illumina reads on the USMARC server by run date and platform (MiSeq or NextSeq) and processes the data using software in a Singularity container.
- 'Hits' validated by visualizing in Geneious™.
- Validated hits include novel sequences with similarity to bovine astrovirus, bovine parvovirus, bovine rhinitis virus A and bovine coronavirus.

A Portion of the Virus Finding Pipeline Bash Script

```
###Screen viral metagenomes against the host database.
if [ $orientation == 'single_end' ]; then cd ${bbduk}; h2p=$(echo "scale=0; $nprocs*0.75/1" | bc)
for lib in ${library[*]}; do echo "Screening ${lib} against the host genome"
singularity exec ${vmloc}/drvm.simg \
    hisat2 -q -x ${hostdb} \
    -U ${lib}.fq.gz \
    -S ${hisat2}${lib}_${host}.sam -p ${h2p}; done
elif [ $orientation == 'paired_end' ]; then cd ${bbduk}; h2p=$(echo "scale=0; $nprocs*0.75/1" | bc)
for lib in ${library[*]}; do echo "Screening ${lib} against the host genome"
singularity exec ${vmloc}/drvm.simg \
    hisat2 -q -x ${hostdb} \
    -1 ${lib}_R1.fq.gz -2 ${lib}_R2.fq.gz \
    -S ${hisat2}${lib}_${host}.sam -p ${h2p}; done
else echo "Check your orientation parameter, it needs to be single_end or paired_end"; fi; wait

###Extract non-host reads from sam files.
cd ${hisat2}; for lib in ${library[*]}; do echo "Extracting non_host reads from ${lib}"
singularity exec ${vmloc}/drvm.simg \
    reformat.sh in=${lib}_${host}.sam \
    out=${hisat2}${lib}_nohost.fq unmappedonly primaryonly overwrite=true; done

###Run drvm
for lib in ${library[*]}; do echo "Running drVM on ${lib}"
SINGULARITYENV_MyDB=${vmloc} \
    singularity exec ${vmloc}/drvm.simg \
    drVM.py -i ${hisat2}${lib}_nohost.fq -t 8;
outpre="myOutput_"
out=${hisat2}${lib}${pre}*
drvmdir=`echo ${out}`
mv ${drvmdir} ${drvm}${lib}_drvm_output; done; rm *.log
```

REFERENCES:

Callahan BJ, Sankaran K, Fukuyama JA, *et al.* Bioconductor workflow for microbiome data analysis: from raw reads to community analyses [version 2; peer review: 3 approved]. *F1000Research.* 2016; **5**: 1492.

Dobin A, Davis CA, Schlesinger F, *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics.* 2013; **29**(1): 15-21.

Kearse M, Moir R, Wilson A, *et al.* Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics.* 2012; **28**(12): 1647-1649.

Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. *PLoS ONE.* **12**(5): e0177459.

Volmers J, Wiegand S, Kaster A. Comparing and evaluating metagenome assembly tools from a microbiologist's perspective – not only size matters. *PLoS* ONE. **12**(1): e0169662.

Aaron Dickey,
Dan Nonneman,
Harvey Freetly,
Aspen Workman,
Larry Kuehn

USDA United States Department of Agriculture
Agricultural Research Service