

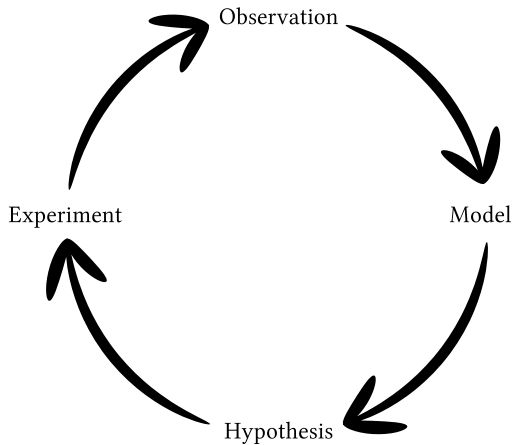
# Practical Bioinformatics

Mark Voorhies

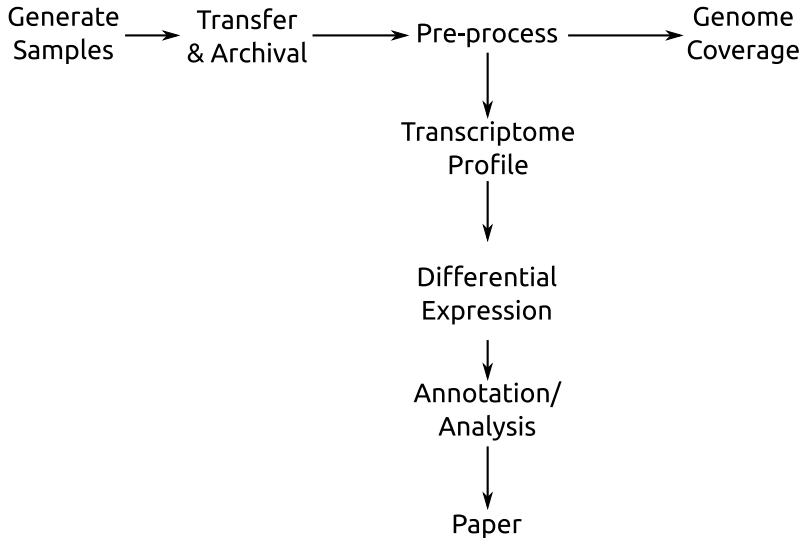
4/16/2018

# JavaTreeView link-out for ENSEMBL Mouse

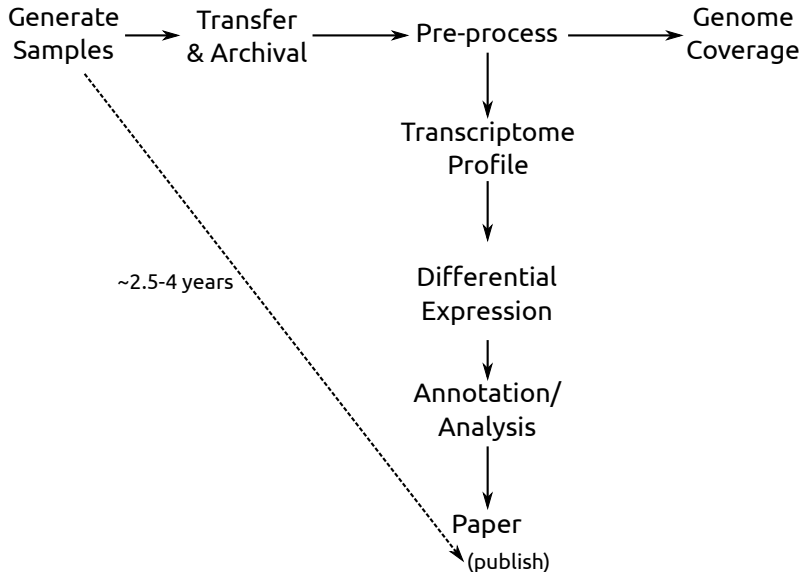
[http://www.ensembl.org/Mus\\_musculus/Gene/Summary?g=HEADER](http://www.ensembl.org/Mus_musculus/Gene/Summary?g=HEADER)



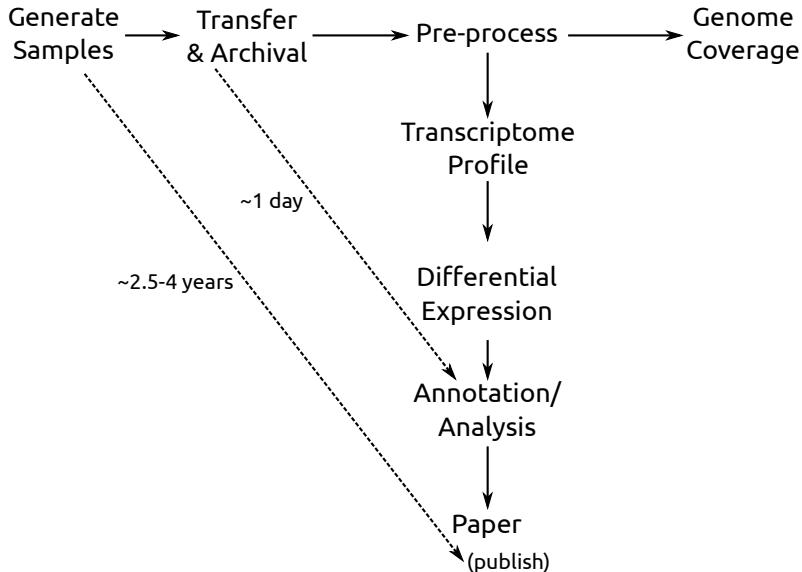
# Example Pipeline: Overview



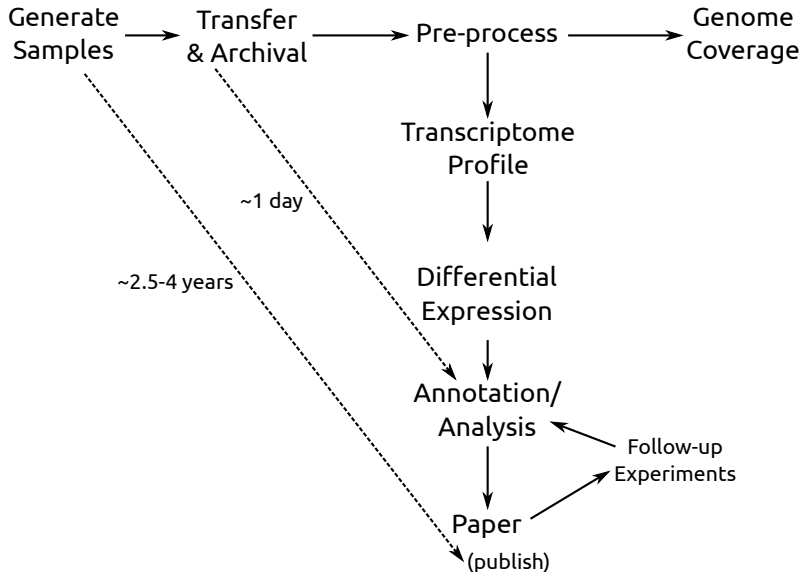
# Example Pipeline: Overview



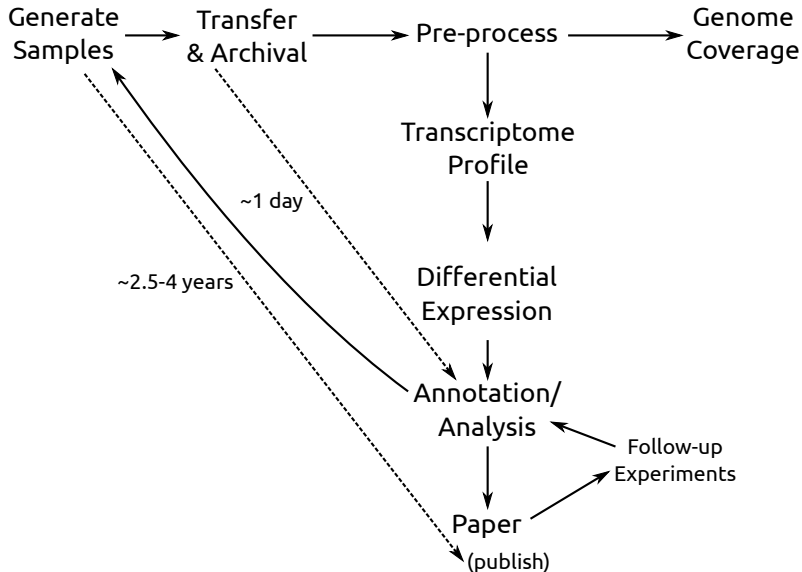
# Example Pipeline: Overview



# Example Pipeline: Overview

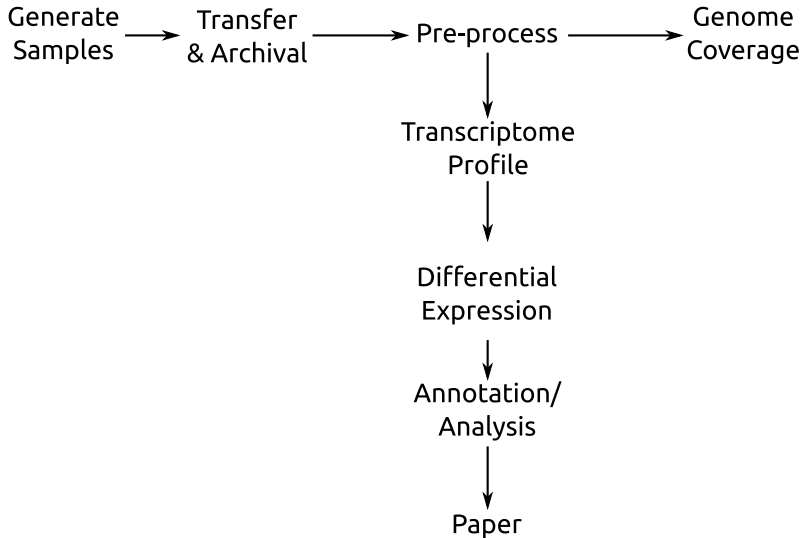


# Example Pipeline: Overview

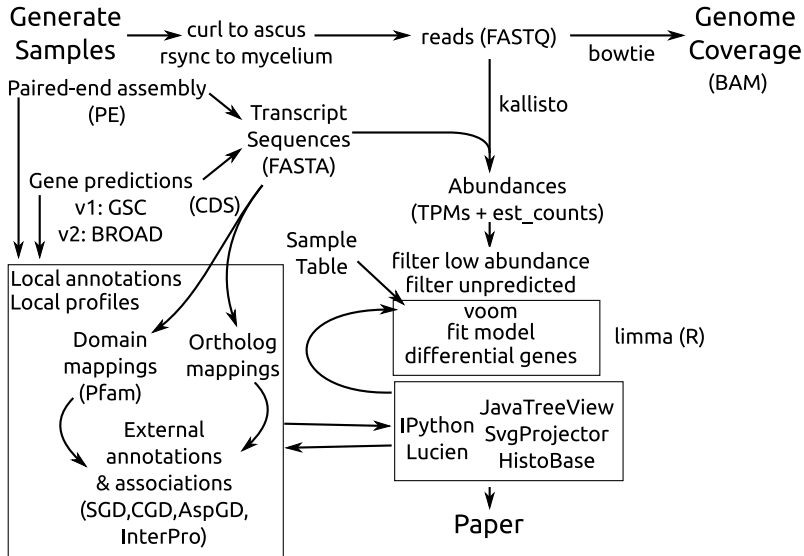




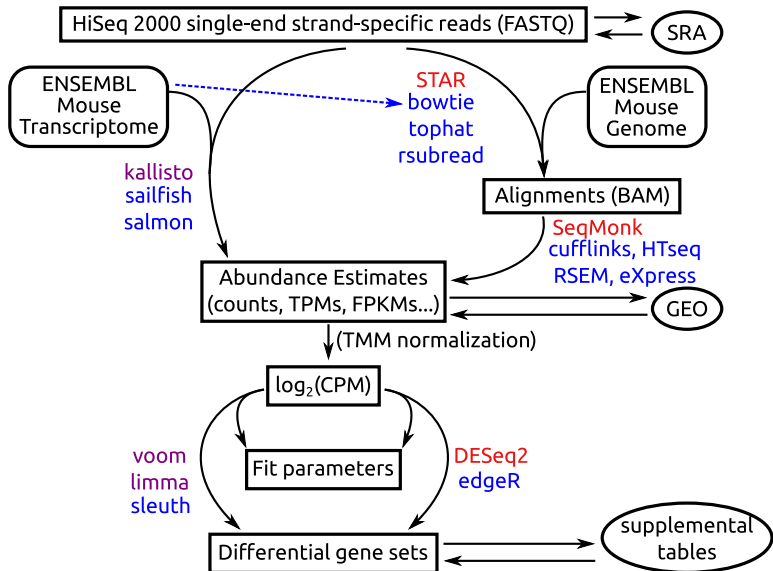
# Example Pipeline: Overview



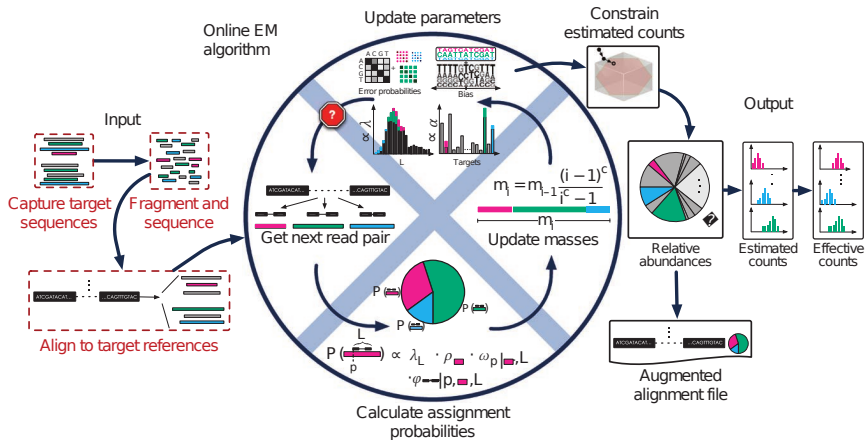
# Example Pipeline: Details



# GSE88801 Pipelines



# EM: Expectation Maximization

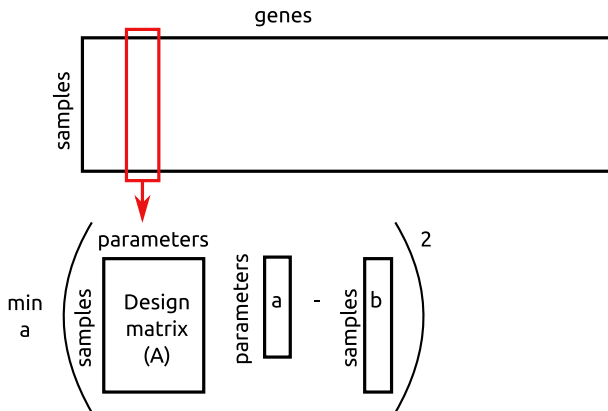


Roberts and Pachter, Nature Methods 10:71

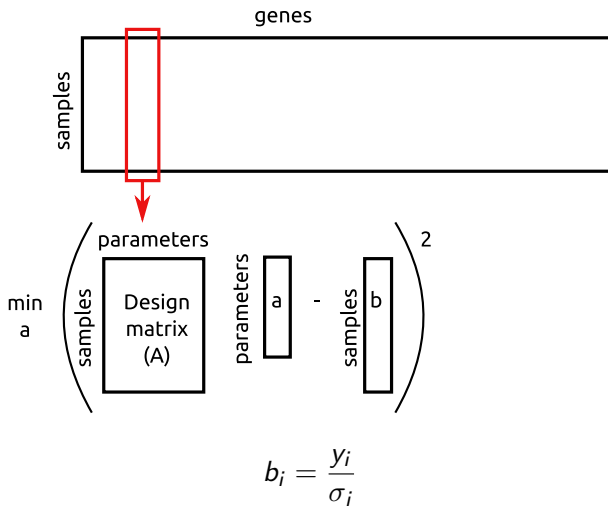
# Abundance estimation with kallisto

```
export transcriptome="GRCm38_all_mRNA"  
while read i; do  
  export jobname="${i}.${transcriptome}.fr"  
  kallisto quant -i "${transcriptome}.idx" \  
    -t 4 --single --fr-stranded -l 250 -s 50  
    -o "${jobname}" "${i}_1.fastq.gz" \  
    > "${jobname}.log" \  
    2> "${jobname}.err"  
done < sample_names.txt
```

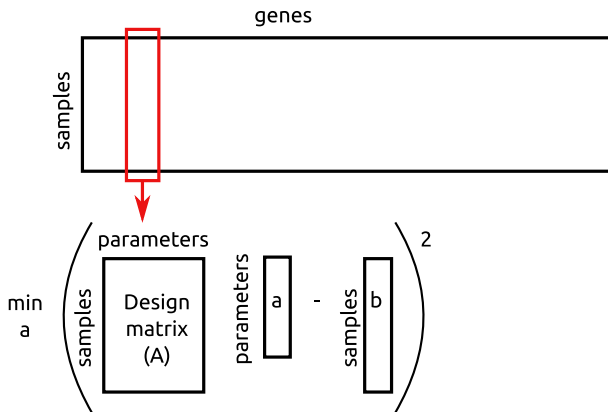
# Linear Least Squares



# Linear Least Squares



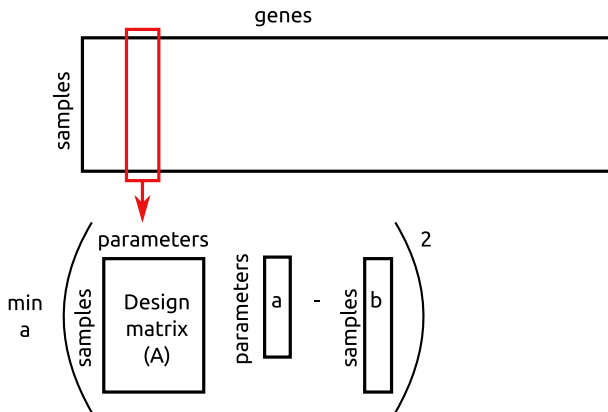
# Linear Least Squares



$$A_{ij} = \frac{f_j(x_i)}{\sigma_i}$$

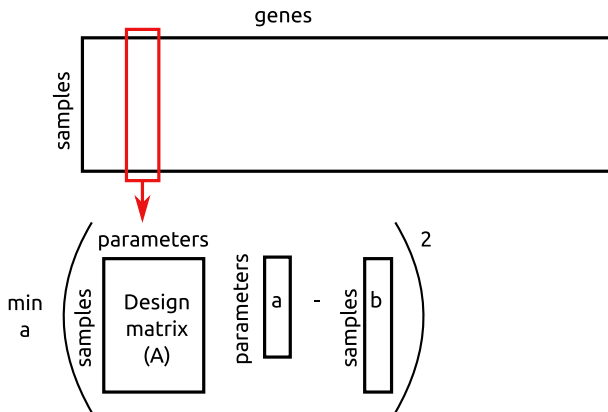


# Linear Least Squares



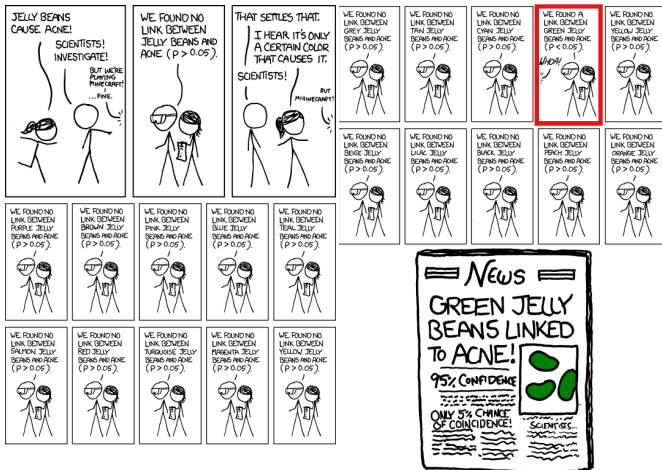
$$\chi^2 = |A \cdot a - b|^2$$

# Linear Least Squares



$$a = \sum_i^M \left( \frac{U_i \cdot b}{s_i} \right) V_i$$

# Multiple Hypothesis Testing



<http://xkcd.com/882/>

# Final Homework

Implement Needleman-Wunsch global alignment with zero gap opening penalties. Try attacking the problem in this order:

- 1 Initialize and fill in a dynamic programming matrix by hand (e.g., try reproducing the example from my slides on paper).
- 2 Write a function to create the dynamic programming matrix and initialize the first row and column.
- 3 Write a function to fill in the rest of the matrix
- 4 Rewrite the *initialize* and *fill* steps to store pointers to the best sub-solution for each cell.
- 5 Write a *backtrace* function to read the optimal alignment from the filled in matrix.

If that isn't enough to keep you occupied, try implementing Smith-Waterman local alignment and/or non-zero gap opening penalties.