Accurate and Fast Transcript (and gene) Quantification

Rob Patro

Stony Brook University

ANGUS 2016  Aug. 16 2016
**Whole transcriptome analysis**

- Quantification & differential expression
- Novel txp discovery
  - reference-based
  - de novo
- Variant detection
  - Genomic SNPs
  - RNA editing

- What is dynamic & changing over time (as disease progresses)?
- What is tissue specific (in fetal development but not after)?
- What is condition specific (under stress conditions vs. not)?
Sequencing Reads

pre-proc.

align to ref.

txp. identification

reference-based

quantification

DE, Alternative Splicing, etc.

“Higher-level” analysis

pre-proc.

de novo assembly

align to assembly

de novo

Today
Why do we still need faster analysis?

The real cost of sequencing: scaling computation to keep pace with data generation

Paul Muir, Shantao Li, Shaoke Lou, Daifeng Wang, Daniel J Spakowicz, Leonidas Salichos, Jing Zhang, George M. Weinstock, Farren Isaacs, Joel Rozowsky and Mark Gerstein

In addition to new data, re-analysis of existing experiments often desired: In light of new annotations, discoveries, and methodological advancements.

Short Read Archive @ NCBI
Currently > 5 petabases of data

# From Fig 1 of Muir et al.
Advocating for analysis-efficient computing

• Compute *only* the information required for your analysis; ask what information you *need* to solve your problem, not what output current tools are generating.

• Often the efficiency of the analysis is related to the *size* of the (processed) data’s representation.

• Not all analyses require such efficient solutions, should concentrate on problems where this is actually needed.

I’ll provide some (hopefully) compelling examples:

• **RapMap**: Read alignment → quasi-mapping (get “core” info much faster)

• **Salmon**: Fast, state-of-the-art quantification using quasi-mapping, dual-phase inference & fragment eq. classes

• **RapClust**: Fast, accurate *de novo* assembly clustering using quasi-mapping & fragment eq. classes

We believe these ideas are *general*, and can be applied to many problems.
Advocating for analysis-efficient computing

• Compute *only* the information required for your analysis; ask what information you *need* to solve your problem, not what output current tools are generating

• Often the efficiency of the analysis is related to the *size* of the (processed) data’s representation

• Not all analyses require such efficient solutions, should concentrate on problems where this is actually needed.

I’ll provide a (hopefully) compelling example:

**Boiler** is also a beautiful example of this idea.

When we have a particular analysis in mind — transcript identification & quantification — we can compress data much more aggressively & effectively.

We believe these ideas are **general**, and can be applied to many problems
I promised to show how we can use this yesterday …

**RapMap**: A Rapid, Sensitive and Accurate Tool for Mapping RNA-seq Reads to Transcriptomes

GitHub repository:  https://github.com/COMBINE-lab/RapMap

**Paper**:  [http://bioinformatics.oxfordjournals.org/content/32/12/i192.full.pdf](http://bioinformatics.oxfordjournals.org/content/32/12/i192.full.pdf)  (appeared at ISMB 16)

co-authors (students): Avi Srivastava, Hirak Sarkar, Nitish Gupta
Where might we use quasi-mapping?

We believe there are many places where this replacement can be made. I’ll discuss one in some depth (and mention a second):

1) Transcript-level quantification
   - Determine abundance of transcripts from a collection of RNA-seq reads.
   - The quasi-mapping information is sufficient to yield estimates as accurate as full alignment.

2) De novo transcript clustering
   - Find groups of related contigs likely from the same transcript / gene
   - Such groups help improve downstream analysis (e.g. differential expression testing)

Obviously, alignments are necessary for certain types of analysis (e.g. variant detection).
Transcript Quantification: An Overview

1 gene ⇒ many variants (isoforms)

Gene 1

Gene M

Sample

10s-100s of millions of short (35-300 character) “fragments”

Measurement (RNA-seq)

Inference (e.g. Salmon)

% Gene 1

% Gene M

Abundance Estimates

isoform A

isoform B

isoform C
**Given:**

1. Collection of RNA-Seq fragments
2. A set of known (or assembled) transcript sequences

**Estimate:**

The relative abundance of each transcript
Given: (1) Collection of RNA-Seq fragments  
(2) A set of known (or assembled) transcript sequences

Estimate: The relative abundance of each transcript

Question: If we only care about “gene” abundance, can’t we just count the number of reads mapping / aligning to each gene?
Given: (1) Collection of RNA-Seq fragments
(2) A set of known (or assembled) transcript sequences

Estimate: The relative abundance of each transcript

Question: If we only care about “gene” abundance, can’t we just count the number of reads mapping / aligning to each gene?

Answer: No. I’ll show a general argument (and a few examples) why!
First, consider this non-Biological example

Imagine I have two colors of circle, red and blue. I want to estimate the fraction of circles that are red and blue. I’ll sample from them by tossing down darts.

Here, a dot of a color means I hit a circle of that color. What type of circle is more prevalent? What is the fraction of red / blue circles?
First, consider this non-Biological example

Imagine I have two colors of circle, red and blue. I want to estimate the fraction of circles that are red and blue. I’ll sample from them by tossing down darts.

You’re missing a crucial piece of information! The areas!
First, consider this non-Biological example

Imagine I have two colors of circle, red and blue. I want to estimate the fraction of circles that are red and blue. I’ll sample from them by tossing down darts.

You’re missing a crucial piece of information!

The areas!

There is an analog in RNA-seq, one needs to know the length of the target from which one is drawing to meaningfully assess abundance!
Resolving multi-mapping is fundamental to quantification

Isoform A is half as long as isoform B

<table>
<thead>
<tr>
<th>Condition 1</th>
<th>Condition 2</th>
<th>union-model fold-change</th>
<th>true fold-change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \log_2 \left( \frac{10}{10} \right) = 0 )</td>
<td>( \log_2 \left( \frac{L}{L + 4L} \right) = 0.32 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \log_2 \left( \frac{6}{8} \right) = -0.41 )</td>
<td>( \log_2 \left( \frac{6/L}{8/2L} \right) = 0.58 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \log_2 \left( \frac{5}{10} \right) = -1 )</td>
<td>( \log_2 \left( \frac{5/L}{10/2L} \right) = 0 )</td>
</tr>
</tbody>
</table>

Key point: The length of the actual molecule from which the fragments derive is crucially important to obtaining accurate abundance estimates.

Resolving multi-mapping is fundamental to quantification

These errors can affect DGE calls

Variants of Salmon

Variants of “counting”
Resolving multi-mapping is fundamental to quantification

Can even affect abundance estimation in **absence** of alternative-splicing (e.g. paralogous genes)
How do we do something better than “counting”?

Think about the “ideal” RNA-seq experiment . . .

Experimental Mixture

Read set

Pick a transcript \( t \propto \text{count} \times \text{length} \)

Pick a position \( p \) on \( t \) uniformly “at random”
How do we do something better than “counting”?

Experimental Mixture

<table>
<thead>
<tr>
<th>Length</th>
<th>Copies</th>
<th>Total Length</th>
<th>Percentage</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>6</td>
<td>600 nt</td>
<td>~ 30%</td>
<td>blue</td>
</tr>
<tr>
<td>66</td>
<td>19</td>
<td>1254 nt</td>
<td>~ 60%</td>
<td>green</td>
</tr>
<tr>
<td>33</td>
<td>6</td>
<td>198 nt</td>
<td>~ 10%</td>
<td>red</td>
</tr>
</tbody>
</table>
How do we do something better than “counting”?

We call these values $\eta = [0.3, 0.6, 0.1]$ the nucleotide fractions, they become the primary quantity of interest.

Experimental Mixture

- length( ) = 100 x 6 copies = 600 nt ~ 30% blue
- length( ) = 66 x 19 copies = 1254 nt ~ 60% green
- length( ) = 33 x 6 copies = 198 nt ~ 10% red

We call these values $\eta = [0.3, 0.6, 0.1]$ the nucleotide fractions, they become the primary quantity of interest.
Say we knew the $\eta$, and observed a read that mapped ambiguously, as shown above. What is the probability that it truly originated from $G$ or $R$?

$$\Pr \{ r \text{ from } G \} = \frac{\eta_G}{\text{length}(G)} + \frac{\eta_R}{\text{length}(R)} = \frac{0.6}{66} + \frac{0.1}{33} = 0.75$$

$$\Pr \{ r \text{ from } R \} = \frac{\eta_R}{\text{length}(R)} + \frac{\eta_G}{\text{length}(G)} = \frac{0.1}{33} + \frac{0.6}{66} = 0.25$$

$\text{length}(\text{blue}) = 100 \times 6 \text{ copies} = 600 \text{ nt} \sim 30\% \text{ blue}$

$\text{length}(\text{green}) = 66 \times 19 \text{ copies} = 1254 \text{ nt} \sim 60\% \text{ green}$

$\text{length}(\text{red}) = 33 \times 6 \text{ copies} = 198 \text{ nt} \sim 10\% \text{ red}$
How to assess “abundance”

RPKM — Reads per kilobase per million mapped reads

FPKM — Fragments per kilobase per million mapped reads

Don’t use these measures, TPM measures the “same thing”, but in a better way.

TPM — Transcripts per million

Useful for visualization / assessment etc.

(Estimated) Number of Reads

These are what are used (after normalization) for differential expression. Why can’t we use TPM?
Transcript Quantification

Salmon provides accurate, fast, and bias-aware transcript expression estimates using dual-phase inference

Official website:  http://combine-lab.github.io/salmon/

GitHub repository:  https://github.com/COMBINE-lab/salmon
A probabilistic view of RNA-Seq quantification

We want to find the values of \( \eta \) that maximize this probability. We can do this (at least locally) using the EM algorithm.
Why does $\Pr\{f_j \mid t_i\}$ matter?

Consider the following scenario:

Isoform A

<table>
<thead>
<tr>
<th>Fragment length dist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>200 bp</td>
</tr>
</tbody>
</table>

Isoform B

<table>
<thead>
<tr>
<th>Fragment length dist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1000 bp</td>
</tr>
</tbody>
</table>

Aux. model provides *strong* information about origin of a fragment!

Prob of observing a fragment of size $\sim200$ is **large**

Prob of observing a fragment of size $\sim1000$ is **very small**
Salmon’s “pipeline”

- Raw reads (e.g. fastq files)
- Reference transcripts
- Salmon index
- Quasi-mapping
- Online inference [SCVB0]
- Initial abundances & fragment equiv. classes
- Offline inference [EM or VBEM]
- Aligned reads (e.g. bam file) & reference transcripts

Dual-phase inference
Reads 1 & 3 both map to transcripts B & E
Reads 2 & 4 both map to transcript C

We have 4 reads, but only 2 eq. classes of reads

<table>
<thead>
<tr>
<th>eq. Label</th>
<th>Count</th>
<th>Aux weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>{B,E}</td>
<td>2</td>
<td>(w^{[B,E]}_B, w^{[B,E]}_E)</td>
</tr>
<tr>
<td>{C}</td>
<td>2</td>
<td>(w^{[C]}_C)</td>
</tr>
</tbody>
</table>
We have 4 reads, but only 2 eq. classes of reads

<table>
<thead>
<tr>
<th>eq. Label</th>
<th>Count</th>
<th>Aux weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>{B,E}</td>
<td>2</td>
<td>$w^{(B,E)}_B$, $w^{(B,E)}_E$</td>
</tr>
<tr>
<td>{C}</td>
<td>2</td>
<td>$w^{(C)}_C$</td>
</tr>
</tbody>
</table>
The number of equivalence classes is small

<table>
<thead>
<tr>
<th></th>
<th>Yeast</th>
<th>Human</th>
<th>Chicken</th>
</tr>
</thead>
<tbody>
<tr>
<td># contigs</td>
<td>7353</td>
<td>107,389</td>
<td>335,377</td>
</tr>
<tr>
<td># samples</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Total (paired-end) reads</td>
<td>~36,000,000</td>
<td>~116,000,000</td>
<td>~181,402,780</td>
</tr>
<tr>
<td>Avg # eq. classes (across samples)</td>
<td>5197</td>
<td>100,535</td>
<td>222,216</td>
</tr>
</tbody>
</table>

The number of equivalence classes grows with the complexity of the transcriptome — independent of the number of sequence fragments.

Typically, **two or more orders of magnitude** fewer equivalence classes than sequenced fragments.

The offline inference algorithm scales in the number of fragment equivalence classes.
Transcript inference methods can be very accurate

Results on 20 replicates simulated (RSEM-sim) from parameters learned from NA12716_7 from GEUVADIS. Showing result distributions for kallisto\(^1\), eXpress\(^2\) & salmon\(^3\)

\[ ARD_i = \begin{cases} 
0 & \text{if } x_i = y_i = 0 \\
\frac{|x_i - y_i|}{x_i + y_i} & \text{otherwise} 
\end{cases} \]


Biases abound in RNA-seq data

Biases in prep & sequencing can have a significant effect on the fragments we see.

Fragment gc-bias\(^1\)—The GC-content of the fragment affects the likelihood of sequencing

Sequence-specific bias\(^2\)—sequences surrounding fragment affect the likelihood of sequencing

Positional bias\(^2\)—fragments sequenced non-uniformly across the body of a transcript


Sequence-bias models don't account for fragment-level GC bias

Accuracy difference can be larger with biased data

Simulated data:
2 conditions; 8 samples each

- Simulated transcripts across entire genome with known abundance using Polyester (modified to account for GC bias)
- How well do we recover the underlying relative abundances?
- How does accuracy vary with level of bias?

joint work with Geet Duggal, Mike Love, Rafael Irizarry & Carl Kingsford
Accuracy difference can be larger with biased data

joint work with Geet Duggal, Mike Love, Rafael Irizarry & Carl Kingsford
Simulated data:
2 conditions; 8 replicates each

• set 10% of txps to have fold change of 1/2 or 2 — rest unchanged.

• How well do we recover true DE?

• Since bias is systematic, effect may be even worse than accuracy difference suggests.

Mis-estimates confound downstream analysis
Accuracy difference can be large with biased data!

<table>
<thead>
<tr>
<th>FDR</th>
<th>Sensitivity at given FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Salmon</td>
</tr>
<tr>
<td>0.01</td>
<td>0.326</td>
</tr>
<tr>
<td>0.05</td>
<td>0.409</td>
</tr>
<tr>
<td>0.1</td>
<td>0.454</td>
</tr>
</tbody>
</table>

At the same FDR, accuracy differences of 53 - 450%

joint work with Geet Duggal, Mike Love, Rafael Irizarry & Carl Kingsford
Importance with experimental data

30 samples from the GEUVADIS study:
  15 samples from UNIGE sequencing center
  15 samples from CNAG_CRG sequencing center

Same human population, expect few-to-no real DE (primary differences in sample prep)

DE of data between centers (FDR < 1%) (TPM > 0.1)

<table>
<thead>
<tr>
<th></th>
<th>Salmon</th>
<th>Kallisto</th>
<th>eXpress</th>
</tr>
</thead>
<tbody>
<tr>
<td>All transcripts</td>
<td>1,171</td>
<td>2,620</td>
<td>2,472</td>
</tr>
<tr>
<td>Transcripts of 2 isoform genes</td>
<td>224</td>
<td>545</td>
<td>531</td>
</tr>
</tbody>
</table>

Bias and batch effects are substantial, and must be accounted for.
Importance with *experimental* data

30 samples from the GEUVADIS study:
- 15 samples from UNIGE sequencing center
- 15 samples from CNAG_CRG sequencing center

Same human population, expect few-to-no *real* DE (primary differences in sample prep)

DE of data between centers (FDR < 1%) (TPM > 0.1)

<table>
<thead>
<tr>
<th></th>
<th>Salmon</th>
<th>Kallisto</th>
<th>eXpress</th>
</tr>
</thead>
<tbody>
<tr>
<td>All transcripts</td>
<td>1,171</td>
<td>2,620</td>
<td>2,472</td>
</tr>
<tr>
<td>Transcripts of 2 isoform genes</td>
<td>224</td>
<td>545</td>
<td>531</td>
</tr>
</tbody>
</table>

Bias and batch effects are *substantial*, and I care only about **genes**!

But this is txp-level DE, and I care only about **genes**!
Importance with *experimental* data

30 samples from the GEUVADIS study:
15 samples from UNIGE sequencing center
15 samples from CNAG_CRG sequencing center

Effects seem **at least as extreme** at the gene level

DE of data between centers (FDR < 1%) (TPM > 0.1)

<table>
<thead>
<tr>
<th></th>
<th>Salmon</th>
<th>Kallisto</th>
<th>eXpress</th>
</tr>
</thead>
<tbody>
<tr>
<td>All genes</td>
<td>455</td>
<td>1,200</td>
<td>1582</td>
</tr>
<tr>
<td>Transcripts of 2 isoform genes</td>
<td>224</td>
<td>545</td>
<td>531</td>
</tr>
</tbody>
</table>

**Bias** and **batch effects** are *substantial*, and must be accounted for.
Salmon and kallisto are FAST

Salmon and kallisto are FAST

Consider the following test:

Take all 20 replicates of the RSEM-sim simulated data above, treat them as one, giant sample. This is 20 samples x 30M paired-end reads = 600 million read pairs or 1.2 billion individual reads.

Using 30 threads\(^1\):

kallisto can process this sample in 20 minutes
Salmon can process this sample in 23 minutes

Just *aligning* the reads to use e.g. eXpress, Cufflinks, RSEM etc. would take dozens of hours.

---

\(^1\) Intel Xeon E5-4600 (2.6GHz)
One “issue” with maximum likelihood (ML)

The generative statistical model is a principled and elegant way to represent the RNA-seq process.

It can be optimized efficiently using e.g. the EM / VBEM algorithm.

**but**, these efficient optimization algorithms return “point estimates” of the abundances. That is, there is no notion of how *certain* we are in the computed abundance of transcript.
One “issue” with maximum likelihood (ML)

There are multiple sources of uncertainty e.g.

- **Technical variance**: If we sequenced the *exact* same sample again, we’d get a different set of fragments, and, potentially a different solution.

- **Uncertainty in inference**: We are almost never guaranteed to find a unique, globally optimal result. If we started our algorithm with different initialization parameters, we might get a different result.

We’re trying to find the *best* parameters in a space with 10s to 100s of thousands of dimensions!
One “issue” with maximum likelihood (ML)

If we started here
We’d end up here

but, if we started here
We’d end up here

https://commons.wikimedia.org/wiki/File:Local_search_attraction_basins.png (CC BY-SA 3.0)
Assessing Uncertainty

There are a few ways to address this “issue”

Do a fully Bayesian inference\(^1\):
Infer the entire posterior distribution of parameters, not just a ML estimate (e.g. using MCMC) — too slow!

✓ Posterior Gibbs Sampling:
Starting from our ML estimate, do MCMC sampling to explore how parameters vary — if our ML estimate is good, and taking advantage of equivalence classes, this can be made very fast.

✓ Bootstrap Sampling\(^2\):
Resample (from equivalence class counts) with replacement, and re-run the ML estimate for each sample. This can be made reasonably fast.

Happy to discuss details / implications of this further.

---


2: IsoDE introduced the idea of bootstrapping counts to assess quantification uncertainty. [Al Seesi, Sahar, et al. "Bootstrap-based differential gene expression analysis for RNA-Seq data with and without replicates." BMC genomics 15.8 (2014): 1.], but it was first made practical / fast in kallisto by doing the bootstrapping over equivalence classes.
Salmon addresses the main challenges of quantification

- finding locations of reads (mapping) is slow than necessary → Use quasi-mapping

- alternative splicing and related sequences creates ambiguity about where reads came from → Use dual-phase inference algorithm

- sampling of reads is not uniform or idealized → Use bias models learned from data

- uncertainty in ML estimate of abundances → Use posterior Gibbs sampling or bootstraps to assess uncertainty
Salmon has many other benefits

- Speed of inference makes it possible to use bootstraps or posterior Gibbs sampling to estimate variance (e.g. how certain are we in quantification estimates?).

- Quasi-mapping means no large, intermediate BAM files sitting on disk, or wasting computation time with slow disk I/O.

- Expressive model means new types of bias can be learned and accounted for.

- Separation of mapping / alignment and inference means Salmon can be used with or without existing alignments*. Here I talked only about quasi-mapping, but Salmon can use take BAM input from an aligner (if you really want!).

Many of these improvements (except dual-phase inference) have been back-ported to Sailfish, which is still actively developed!

[https://github.com/kingsfordgroup/sailfish](https://github.com/kingsfordgroup/sailfish)
Thanks!

Collaborators on Salmon

Geet Duggal (CMU / DNAnexus)

Carl Kingsford (CMU)

Mike Love (Harvard / UNC)

Rafael Irizarry (Harvard)
Bonus Slides
De novo transcriptome clustering

RapClust: Fast, Lightweight Clustering of de novo Transcriptomes using Fragment Equivalence Classes

GitHub repository: https://github.com/COMBINE-lab/rapclust
RapClust: clustering contigs in de novo assemblies

Uses the fragment equivalence classes discussed above to cluster contigs in *de novo* assemblies.

This leads to improved downstream analysis (e.g. DE calls)
RapClust is fast

Time *including* quantification (4 threads)

<table>
<thead>
<tr>
<th></th>
<th>Yeast</th>
<th></th>
<th></th>
<th>Human</th>
<th></th>
<th></th>
<th>Chicken</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RapClust</td>
<td>Corset</td>
<td>RapClust</td>
<td>Corset</td>
<td>RapClust</td>
<td>Corset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time(min)</td>
<td>5.12</td>
<td>37.25</td>
<td>22.67</td>
<td>211.67</td>
<td>64.18</td>
<td>453</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Space(Gb)</td>
<td>0.005</td>
<td>5.7</td>
<td>0.092</td>
<td>22</td>
<td>0.49</td>
<td>145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>% of reads</td>
<td>88.17</td>
<td>62.32</td>
<td>93.04</td>
<td>77.94</td>
<td>88.80</td>
<td>60.99</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Time *excluding* quantification

<table>
<thead>
<tr>
<th></th>
<th>Yeast</th>
<th></th>
<th></th>
<th>Human</th>
<th></th>
<th></th>
<th>Chicken</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RapClust</td>
<td>CD</td>
<td>CT</td>
<td>RapClust</td>
<td>CD</td>
<td>CT</td>
<td>RapClust</td>
<td>CD</td>
</tr>
<tr>
<td>Time(min)</td>
<td>0.04</td>
<td>0.2</td>
<td>2.8</td>
<td>0.82</td>
<td>4.02</td>
<td>16.25</td>
<td>5.29</td>
<td>36.5</td>
</tr>
</tbody>
</table>
RapClust is Fast & Lightweight

Time & Space comparison of RapClust with Corset, for all phases (raw reads through quantified clusters — using 4 threads).

<table>
<thead>
<tr>
<th></th>
<th>Yeast</th>
<th></th>
<th>Human</th>
<th></th>
<th>Chicken</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RapClust</td>
<td>Corset</td>
<td>RapClust</td>
<td>Corset</td>
<td>RapClust</td>
</tr>
<tr>
<td>Time(min)</td>
<td>5.12</td>
<td>37.25</td>
<td>22.67</td>
<td>211.67</td>
<td>64.18</td>
</tr>
<tr>
<td>Space(Gb)</td>
<td>0.005</td>
<td>5.7</td>
<td>0.092</td>
<td>22</td>
<td>0.49</td>
</tr>
<tr>
<td>% of reads</td>
<td>88.17</td>
<td>62.32</td>
<td>93.04</td>
<td>77.94</td>
<td>88.80</td>
</tr>
</tbody>
</table>

Not having to output / rely on BAM files means the space footprint of RapClust is orders of magnitude smaller than that of Corset

Time comparison of RapClust (RC), Corset (CT), and CD-HIT EST (CD) for just clustering (using 1 thread).

<table>
<thead>
<tr>
<th></th>
<th>Yeast</th>
<th></th>
<th>Human</th>
<th></th>
<th>Chicken</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RC</td>
<td>CD</td>
<td>CT</td>
<td>RC</td>
<td>CD</td>
</tr>
<tr>
<td>Time(min)</td>
<td>0.04</td>
<td>0.2</td>
<td>2.8</td>
<td>0.82</td>
<td>4.02</td>
</tr>
</tbody>
</table>
RapClust is accurate

Variation of Information\(^\#\) distance between the *true* clustering and the clustering computed by each method (*lower is better*).


<table>
<thead>
<tr>
<th>VI Distance</th>
<th>RapClust</th>
<th>CORSET</th>
<th>CD-HIT EST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chicken</td>
<td>0.127</td>
<td>0.191</td>
<td>2.01</td>
</tr>
<tr>
<td>Human</td>
<td>0.712</td>
<td>0.735</td>
<td>1.24</td>
</tr>
<tr>
<td>Yeast</td>
<td>0.176</td>
<td>0.178</td>
<td>0.216</td>
</tr>
</tbody>
</table>

F1-Score of correct classification (i.e. co-clustering) of contigs from the same gene (*higher is better*).

<table>
<thead>
<tr>
<th>F1-Score</th>
<th>RapClust</th>
<th>CORSET</th>
<th>CD-HIT EST*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chicken</td>
<td>97.17</td>
<td>95.02</td>
<td>13.27</td>
</tr>
<tr>
<td>Human</td>
<td>72.23</td>
<td>70.58</td>
<td>23.97</td>
</tr>
<tr>
<td>Yeast</td>
<td>46.24</td>
<td>45.40</td>
<td>21.48</td>
</tr>
</tbody>
</table>

*Note: RapClust & CORSET only predict clusters on an expressed subset of the data; CD-HIT EST is not directly comparable.*
Phase 1: Online Inference

Process fragments in batches:

Compute local $\eta'$ using $\eta^{t-1}$ & current “bias” model to allocate fragments.

Update global nucleotide fractions: $\eta^t = \eta^{t-1} + a^t \eta'$

Update “bias” model

Place mappings in equivalence classes

- Have access to all fragment-level information when making these updates
- Often converges very quickly.
- Compare-And-Swap (CAS) for synchronizing updates of different batches

Give each transcript appropriate prior mass \( \eta^0 \) (init.)

For each mini-batch \( B^t \) of reads {
    For each read \( r \) in \( B^t \) {
        For each alignment \( a \) of \( r \) {
            compute (un-normalized) prob of \( a \) using \( \eta^{t-1} \), and aux params
        }
        normalize alignment probs & update local transcript weights \( \eta' \)
        add / update the equivalence class for read \( r \)
        sample \( a \in r \) to update auxiliary models
    }
    update global transcript weights given local transcript weights according to “update rule” \( \Rightarrow \eta^t = \eta^{t-1} + w^t \eta' \)
Phase 2: Offline Inference

Repeatedly reallocate fragments according to current abundance estimates & “bias” model until convergence:

\[
\alpha_{i}^{u+1} = \sum_{c_j \in c} d^j \left( \frac{\alpha_{i}^{u} w_i^j}{\sum_{t_k \in t^j} \alpha_{k}^{u} w_k^j} \right)
\]

In practice, we re-estimate the bias terms that depend on the transcript abundances (e.g. seq-specific & fragment-GC) intermittently during optimization.