Introduction to de novo RNA-seq assembly
Introduction
Ideal day for a molecular biologist

Any type of biological material → Genetic material with high quality and yield → "Ideal Sequencer" → Complete genome sequence

Real day for a molecular biologist

- We can’t sequence full-length native DNA (Our technology is limited)
- But we can get the sequence of short fragments (depending on the technology we use)

<table>
<thead>
<tr>
<th>Short Fragments</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 at a time</td>
<td>Sanger</td>
</tr>
<tr>
<td>100,000 at a time</td>
<td>Roche 454</td>
</tr>
<tr>
<td>1,000,000 at a time</td>
<td>PGM</td>
</tr>
<tr>
<td>10,000,000 at a time</td>
<td>Proton, MiSeq</td>
</tr>
<tr>
<td>100,000,000 at a time</td>
<td>HiSeq</td>
</tr>
</tbody>
</table>
How Many Species Have Been Sequenced?

During the last 250 years, 1.2 million eukaryotic species have been identified and taxonomically classified. Number of species estimated to exist on Earth: bacterial and archaea species, from 100,000 to 10 million\(^1,2\); eukaryotic species, approximately 8.7 million (including 2.2 million marine organisms; ± 1.3 million, total)\(^1\).

2. Bergey’s International Society for Microbial Systematics (2014)
3. *PLOS Biology* (August 2011)

*as of April 24, 2014

<table>
<thead>
<tr>
<th></th>
<th>Total species (estimated)</th>
<th>Identified / described species</th>
<th>Species with complete genome sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bacteria, Archaea</strong></td>
<td>100,000 to 10 million</td>
<td>12,000 (460 cultured Archaea)</td>
<td>17,420 bacteria, 362 Archaea genomes</td>
</tr>
<tr>
<td><strong>Fungi</strong></td>
<td>1.5 million</td>
<td>100,000</td>
<td>356</td>
</tr>
<tr>
<td><strong>Insects</strong></td>
<td>10 millions</td>
<td>1 million</td>
<td>98</td>
</tr>
<tr>
<td><strong>Plants</strong></td>
<td>435, 000 (land plants and green algae)</td>
<td>300,000</td>
<td>150</td>
</tr>
<tr>
<td><strong>Terrestrial vertebrates, fish</strong></td>
<td>80,500 (5,500 mammalian)</td>
<td>62,345 (5,487 mammalian)</td>
<td>235 (80 mammalian)</td>
</tr>
<tr>
<td><strong>Marine invertebrates</strong></td>
<td>6.5 million</td>
<td>1.3 million</td>
<td>60</td>
</tr>
<tr>
<td><strong>Other invertebrates</strong></td>
<td>1 million nematode, several thousand <em>Drosophila</em></td>
<td>23,000 nematode, 1,300 <em>Drosophila</em></td>
<td>17 nematode, 21 <em>Drosophila</em></td>
</tr>
</tbody>
</table>

We have a limited number of complete sequenced genomes!
De novo assembly

Reference genome available

A reference-guided assembly
Assembly Strategies
Reference genome available

No

File of short sequence reads with high quality-value

Data structure

String-based model

Greedy-extension

Graph-based model

De Bruijin graphs assembly (DBG)

Overlap-Layout-Consensus Assembly (OLC)
A genome is a compilation of millions of nucleotides. Genomes often have long repetitive regions, and they are hard to assemble using overlap between reads.

- **Sequencing errors**: find the pieces with imperfections/errors
- **Sequencing bias**: Find the missing pieces
- **Overlap**: Find reads which fit together

The genome assembly problem

General steps to reconstruct the genome jigsaw puzzle!
Overlap problem

Solution 1
Sequencing with longer reads but we have a limit current of sequencing technology

Solution 2
The paired-end (PE) sequencing

Intensive computational time requirement problem

Solution 1
Parallelization is implemented in a couple of graph-based assemblers (Multithreaded or Distributable)

Solution 2
Implementation of better algorithms
Overview of *de novo* short reads assemblers

Zhang et al., 2011 A Practical Comparison of De Novo Genome Assembly Software Tools for *Next-Generation* Sequencing Technologies
Do you know what a de Bruijn Graph is?
Kaliningrad is a Russian city and a seaport that lies in a Russian enclave bordering between Poland and Lithuania. In the eighteenth century this city was in the Prussian empire and known as Königsberg.
De Bruijn graphs: from the seven bridges to the genome assembly

Konigsberg had a special geographic peculiarity. It lay on 4 separate landmasses, two of which were islands and the other two the banks of the River Pregel.

Königsberg's residents enjoyed strolling through their city, and they wondered if every part of the city could be visited by walking across each of the seven bridges exactly once and returning to one's starting location.

In 1735, Leonhard Euler represented each landmass as a point (called a node) and each bridge as a line segment (called an edge) connecting the appropriate two points. This creates a graph—a network of nodes connected by edges.

Euler's ideas were subsequently adapted by Dutch mathematician Nicolaas De Bruijn. De Bruijn became interested in the ‘superstring problem’: find a shortest circular ‘superstring’ that contains all possible ‘substrings’ of length k (k-mers) over a given alphabet.

However as the figure shows, the four segments of Konigsberg have an odd number of bridges, three of them have three and one has five. Thus, he showed, it was impossible to cross the bridges without a second crossing of at least one of the bridges.
Repeats
• A segment of DNA which occurs more than once in the genome sequence
• Examples: Transposons (self replicating genes)
  Satellites (repetitive adjacent patterns)
  Gene duplication (paralogs)
• Error assembly due to repetitive regions: An **overlap-consensus-layout assembler** needs to constantly guess whether slight variation between two overlapping segments is due to **repeats or error**.

Graph
4 nodes / vertices
• A, B, C, D

7 edges / arcs
• 1, 2, 2, 4, 5, 6, 7
What is a de Bruijn graph?

De Bruijn graph is a directed graph, where an edge exists from all nodes of form A=(a1, a2, a3, ....an) to B=(b1,b2,b3,...,bn) with B being a left-shifted version of A, or b1=a2, b2=a3,......bn-1=an.

If we consider the assembly of nucleotides, it means that two connected nodes have overlaps of n-1 'nucleotides'.

- **Genome** can be split into its k-mer components.
- Then various **k-mers are connected based on whether they have k-1 common nucleotides**.
- Each line and circle represents collection of many k-mer nodes.
How does the De Bruijn graph-based genome assembly algorithm work?

**Step 1:** short reads are broken into small pieces (k-mers) and a de Bruijn graph is constructed from those short pieces.

**Step 2:** the genome is constructed based on the de Bruijin graph.

**Sequence**
- AACCGG

**K-mers (k=4)**
- AAC, ACC, CCG, CCGG

**Graph**

AACC → ACCG → CCGG

General
Sequence

- AATAAATA

K-mers (k=4)

- AATA ATAA TAAT AATA (repeat)

Graph
Sequence
- CAATATG
K-mers (k=3)
- CAA AAT ATA TAT ATG
Graph
De Bruijn graphs of Transcriptomic or RNAseq Libraries

RNAseq data contains sequences from thousands of genes.

Consider a transcriptome with only two genes, among which one is expressed 100 times more than the other. The de Bruijn graph of each gene can be constructed separately, as shown in the above figure. If two genes do not have any common k-mer, our two-gene transcriptome consists of two separate graphs without any cross-link.

An example of five genes along with their coverage. Genes 2 and 3 are either alternatively spliced versions of the same gene, or two highly similar genes from the same family.
De Bruijn graph of a genome

- Split it into all k-mers (k=7)
- A direct graph with those 7-mers as nodes
- Edges are drawn between node pairs in such a way that the connected nodes have overlaps of 6 (=k-1) nucleotides.
- Here, only the adjacent 7-mers from the original sequence got connected in the graph.

The 5’-most and 3’-most 7-mers are identical (blue).
- Merge the two identical nodes (one less node)
- The original sequence are still connected in the graph, but in addition the graph forms a loop by connecting its two ends.
A very important fact is that each node is double-stranded!

The genome assemblers avoid even \( k \), because with even \( k \), many \( k \)-mers become reverse complements of their own sequences. The strand-specificity of the graph might be affected by these ambiguities, thus odd \( k \)-values are preferred.

How to simplify a DBG graph?

- **Remove tips or spurs**
  - Dead ends in graphs due to errors at read end
- **Collapse bubbles**
  - Errors in middle of reads
  - But could be true SNPs or diploidity
- **Remove low coverage paths**
  - Possible contamination
- **Makes final Eulerian path easier**
  - And get more accurate contigs

A \( k=1 \) de Bruijn graph is not useful as you can see from the following example:
Lets put it together!

**Diagram Description:**

- **Align-then-assemble**
  - Align reads to the genome
  - Reference-based assembly of aligned reads
  - De novo assembly
  - Unaligned reads
  - De novo assembly
  - Comprehensive assembly

- **Assemble-then-align**
  - De novo assembly
  - Scaffold contigs
  - Unassembled reads
  - Extend contigs

---

Identification of key molecular markers for needle retention in firs using RNA-seq data
**Field component**

**Lab component**

---

**Needle Abscission Zone**

5 NAZ-N + 5 NAZ-S / Clone / Rep.

---

**Library preparation and sequencing**

- Indexed separately (barcoded)
- Pooled together
- Sequenced in 2 PE100 lanes
Quick first impression of the data
http://www.bioinformatics.babraham.ac.uk

No reference genome

**Section I**

**de novo assembly**

Assembly (Trinity) (Individually and merged)

Clustering (USEARCH)

Centroids

Merged

Individual

Sailfish
- Sailfish index
- Quantification

Bowtie2 + RSEM
- Indexing
- Quantification

DESeq2
Detection of differentially expressed genes

Use sickle (paired-end aware) to trim reads based on quality. Remove low quality portions while preserving the longest high quality part of a NGS read.

Method to do an efficient and robust *de novo* reconstruction of transcriptomes from RNA-seq data. (Grabherr MG et al., 2011, Nature Bio.)

Group contings by similarity and evaluated the redundancy level.

Divides a set of sequences into clusters. A cluster is defined by one sequence, known as the centroid or representative sequence (longer sequence)

Quantify the relative transcript abundance without mapping reads. Faster method. Counts k-mers from reads against a k-mer index (Patro et al., 2014, Nature Bio.)

Quantify gene and isoform abundances by mapping reads. Reads of each sample are aligned to the assembled transcriptome using Bowtie. Finally, abundance values are calculated with RSEM. (Li, B. & Dewey, CN, 2011, BMC Bio.)
1. Assembles the RNA-seq data into unique contigs.

2. Clusters/groups Inchworm contigs into clusters based on alternative splicing or gene duplication. Constructs complete de Bruijn graphs for each cluster.

3. Processes the individual graphs in parallel, reports full-length transcripts for alternatively spliced isoforms, and teasing apart transcripts that corresponds to paralogous genes.
Practical example
Computational analysis of bacterial RNA-Seq data

Ryan McClure1,2, Divya Balasubramanian3, Yan Sun3, Maksym Bobrovskyy3, Paul Sumby4, Caroline A. Genco1,2, Carin K. Vanderpool3 and Brian Tjaden5,**

1Department of Microbiology, Boston University School of Medicine, Boston, MA 02118, USA, 2Department of Medicine, Section of Infectious Diseases, Boston University School of Medicine, Boston, MA 02118, USA, 3Department of Microbiology, University of Illinois, Urbana, IL 61801, USA, 4Department of Pathology, Center for Molecular and Translational Human Infectious Diseases Research, The Methodist Hospital Research Institute, Houston, TX 77030, USA and 5Computer Science Department, Wellesley College, Wellesley, MA 02481, USA

Received January 14, 2013; Revised March 26, 2013; Accepted May 1, 2013

ABSTRACT

Recent advances in high-throughput RNA sequencing (RNA-seq) have enabled tremendous leaps forward in our understanding of bacterial transcriptomes. However, computational methods for analysis of bacterial transcriptome data have not kept pace with the large and growing data sets generated by RNA-seq technology. Here, we present new algorithms, specific to bacterial gene structures and transcriptomes, for analysis of RNA-seq data. The algorithms are implemented in an open source software system called Rockhopper that supports various stages of bacterial RNA-seq data analysis, including aligning sequencing reads to a genome, constructing transcriptome maps, quantifying transcript abundance, testing for differential gene expression, determining operon structures and visualizing results. We demonstrate the performance of Rockhopper using 2.1 billion sequenced reads from 75 RNA-seq experiments conducted with Escherichia coli, Neisseria gonorrhoeae, Salmonella enterica, Streptococcus pyogenes and Xenorhabdus nematophila. We find

INTRODUCTION

Transcriptome assays are increasingly being performed by high-throughput RNA sequencing (RNA-seq) methods. As compared with microarrays, high-throughput sequencing technologies have a number of advantages, including single base pair resolution, low background signal, a large dynamic range of expression levels over which transcripts can be detected, higher levels of reproducibility, smaller sample requirements for starting RNA and no limitation in detecting transcripts that do not correspond to a previously sequenced genome (1). High-throughput sequencing technologies, including systems from Illumina, ABI and Roche 454, have been used to conduct bacterial RNA-seq experiments with a wide range of applications (2,3). For example, bacterial RNA-seq experiments have been conducted with protein-bound RNA (4), with size-selected RNA when small RNAs are of interest (5–8), with 5’ end selected primary transcripts, e.g. via selection of RNA carrying a 5’ tri-phosphate group (9), with RNA from both pathogens and hosts for investigations of host-pathogen interactions (10) and with RNA from whole environments in metatranscriptome studies (11–13).

One of the challenges associated with RNA-seq experiments is analysis of the large resulting data sets. There are
De novo assembly of bacterial transcriptomes from RNA-seq data

Brian Tjaden

Abstract
Transcriptome assays are increasingly being performed by high-throughput RNA sequencing (RNA-seq). For organisms whose genomes have not been sequenced and annotated, transcriptomes must be assembled de novo from the RNA-seq data. Here, we present novel algorithms, specific to bacterial gene structures and transcriptomes, for analysis of bacterial RNA-seq data and de novo transcriptome assembly. The algorithms are implemented in an open source software system called Rockhopper 2. We find that Rockhopper 2 outperforms other de novo transcriptome assemblers and offers accurate and efficient analysis of bacterial RNA-seq data. Rockhopper 2 is available at http://cs.wellesley.edu/~btjaden/Rockhopper.
Bacterial RNA-seq experiments

High-throughput sequencing technologies, including systems from Illumina, ABI and Roche 454, have been used to conduct bacterial RNA-seq experiments with a wide range of applications:

- Selection of RNA carrying an specific group
- RNA from both pathogens and hosts for investigations of host-pathogen interactions
- RNA from whole environments in metatranscriptome studies

What is different?

**Neighboring genes often overlap:** It is difficult to distinguish the start of one gene transcript from the end of another.

**Polycistronic messages:** Different promoters may drive expression of a gene or operon under different conditions.

**Inappropriate eukaryotic RNA gene models:** gene models are not appropriate to explain small regulatory RNAs (sRNAs) in bacteria.
1. Assemble candidate transcripts and store them in a Burrows-Wheeler index

2. Second pass through the sequencing reads, aligning each read to the index.
   (In the case of paired-end reads, it is required that the paired-ends for each read form a scaffold consistent with the transcript)!!!

3. Sufficiently long regions of candidate transcripts are retained as high quality finalized transcripts.

4. Transcript abundance levels are estimated using a measure similar to RPKM (reads per kilobase per million), which sums the number of reads for a transcript and divides by the transcript’s length and a normalization factor.

5. Tests for differential transcript expression in pairs of conditions using the algorithm of DESeq
ROCKHOPPER

A system for analyzing bacterial RNA-seq data

Rockhopper is a comprehensive and user-friendly system for computational analysis of bacterial RNA-seq data. As input, Rockhopper takes RNA sequencing reads output by high-throughput sequencing technology (FASTQ, QSEQ, FASTA, SAM, or BAM files). Rockhopper supports the following tasks:

- Reference based transcript assembly (when one or more reference genomes are available)
- Aligning reads to genomes
- Assembling transcripts
- Identifying transcript boundaries and novel transcripts such as small RNAs
- De novo transcript assembly (when reference genomes are unavailable)
- Normalizing data from different experiments
- Quantifying transcript abundance

It is recommended that your computer has Java version 1.6 or later and your computer has at least 2 gigabytes of RAM. If you do not have Java or you need to update to a more recent version, you can do so by clicking the Java icon on the right:

Download Latest Release (Rockhopper version 2.0.3)

Rockhopper for Windows

Rockhopper for Mac

Rockhopper for any platform

To execute the GUI version of Rockhopper, use the following command:

```
java -Xmx1200m -jar Rockhopper.jar
```

To execute the command line version of Rockhopper, use the following command:

```
java -Xmx1200m -cp Rockhopper.jar Rockhopper
```

Rockhopper source code

To extract the source code from the JAR file, use the following command:

```
jar xf Rockhopper.jar
```

To extract the source code from the compressed TAR archive, use the following command:

```
tar xjf Rockhopper-2.0.3.tar.bz2
```
Perform de novo transcript assembly without reference replicons

Look In: data_rockhop_tuberculosis

- MtubRNAseq.fq.gz
- read1.fq.gz
- read2.fq.gz

File Name: read2.fq.gz

Files of Type: All Files

[Submit]
Initializing RNAseq analysis...

Assembling transcripts from reads in files:
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read1.fq.gz
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read2.fq.gz

Aligning reads to assembled transcripts using files:
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read1.fq.gz
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read2.fq.gz

Total reads in files: 2022724
Perfectly aligned reads: 757680 37%

Total number of assembled transcripts: 450
Average transcript length: 939
Median transcript length: 555
Total number of assembled bases: 422728

Summary of results written to file: Rockhopper_Results/summary.txt
Details of assembled transcripts written to file: Rockhopper_Results/transcripts.txt

FINSIHED.
Assembling transcripts from reads in files:
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read1.fq.gz
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read2.fq.gz

Aligning reads to assembled transcripts using files:
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read1.fq.gz
C:\Users\LILIAN\Documents\Lila-2015\BIT815_2015\data_rockhop_tuberculosis\read2.fq.gz

Total reads in files: 2022724
Perfectly aligned reads: 810305 40%

Total number of assembled transcripts: 1122
Average transcript length: 670
Median transcript length: 397
Total number of assembled bases: 751969

Summary of results written to file: Rockhopper_Results/summary.txt
Details of assembled transcripts written to file: Rockhopper_Results/transcripts.txt
FINISHED.