Explore Spark for Metagenome assembly

Zhong Wang, Ph.D.

Group Lead, Genome Analysis
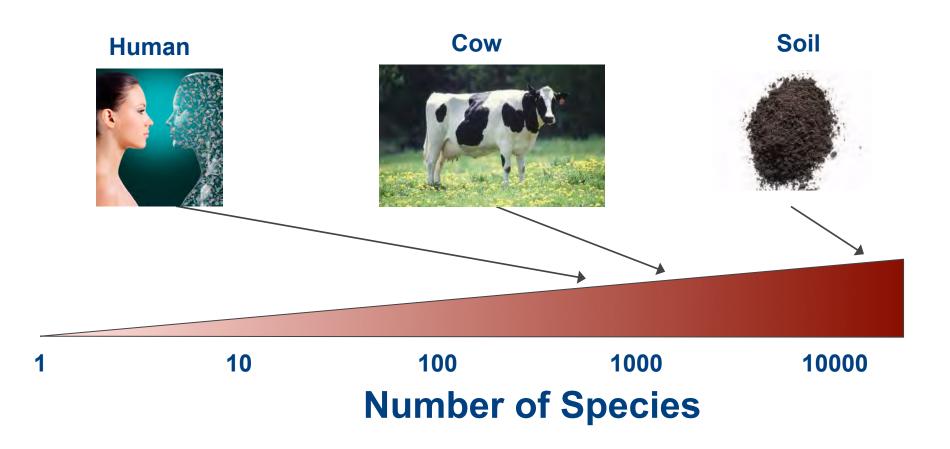
National Microbiome Initiative
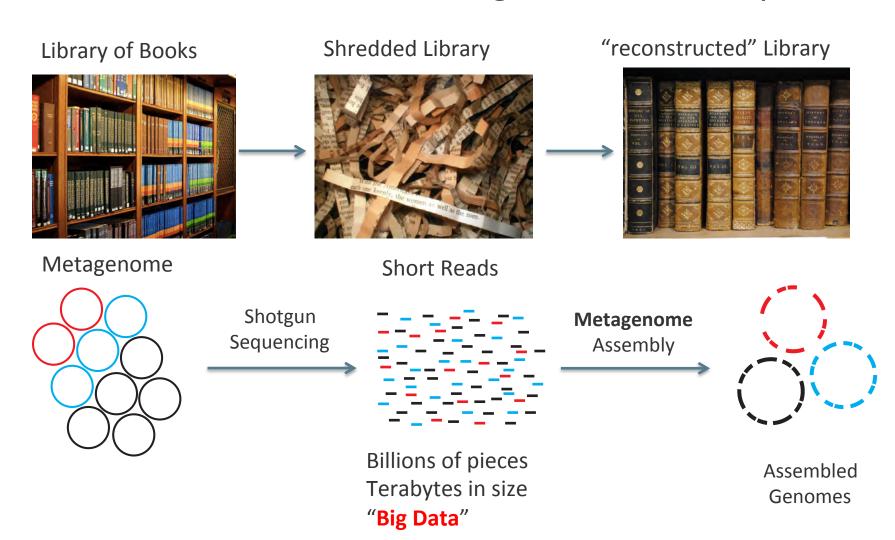
Decode Metagenome

Genome ~= Book   Metagenome ~= Library
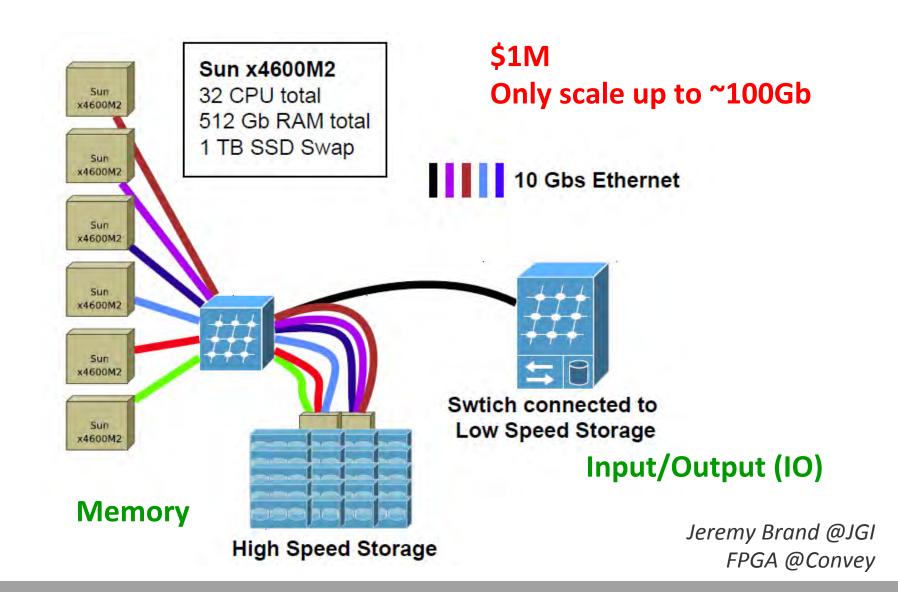
Library of Books   Shredded Library   "reconstructed" Library

Metagenome   Short Reads

Shotgun Sequencing

Metagenome Assembly

Billions of pieces
Terabytes in size
"**Big Data**"

Assembled Genomes

- **Easy to develop**
- **Robust**
- **Scale to big data**
- **Efficient**

Sun x4600M2
32 CPU total
512 Gb RAM total
1 TB SSD Swap

$1M
Only scale up to ~100Gb

10 Gbs Ethernet

Swtich connected to
Low Speed Storage

Input/Output (IO)

Memory

High Speed Storage

Jeremy Brand @JGI
FPGA @Convey

**MPI version
412 Gb, 4.5B reads
2.7 hours on 128x24 cores
NESRC Supercomputer**

Fast, scalable

Problems:

- **Experienced software engineers**
- **Six months of development time**
- **One task fails, all tasks fail**

Rob Egan @JGI

- **Google MapReduce**
  - Data Parallel programming model to process petabyte data
  - Generally has a map and a reduce step
- **Apache Hadoop**
  - Distributed file system (HDFS) and job handling for scalability and robustness
  - Data locality to bring compute to data, avoiding network transfer bottleneck

finding out top 5 websites young people visit

```
Users = load 'users' as (name,
age);
Fltrd = filter Users by
          age >= 18 and age <= 25;
Pages = load 'pages' as (user,
url);
Jnd = join Fltrd by name, Pages by
user;
Grpd = group Jnd by url;
Smmd = foreach Grpd generate
group,
          COUNT(Jnd) as clicks;
Srtd = order Smmd
by clicks desc;
Top5 = limit Srtd
5;
store Top5 into
'top5sites';
```

BioPIG

Robust

| BioPig-Blaster | BioPig-Assembler | BioPig-Extender |
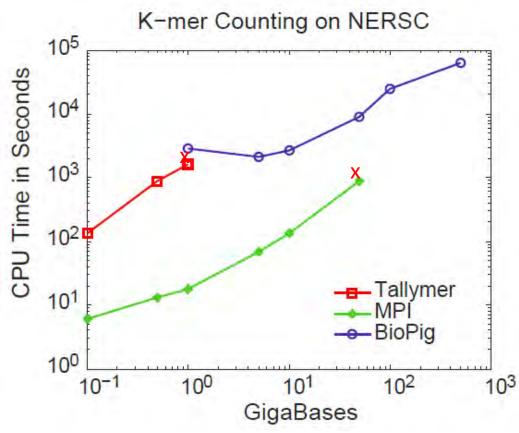
Programmability

BioPig: 61 lines of code
MPI-extender: ~12,000 lines
(vs 31 in BioPig)

Scalability

K-mer Counting on NERSC

Legend:
- Tallymer
- MPI
- BioPig

X-axis: GigaBases
Y-axis: CPU Time in Seconds

Karan Bhatia, Henrik Nordberg, Kai Wang
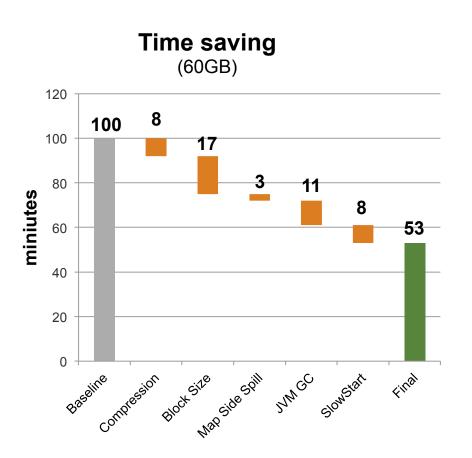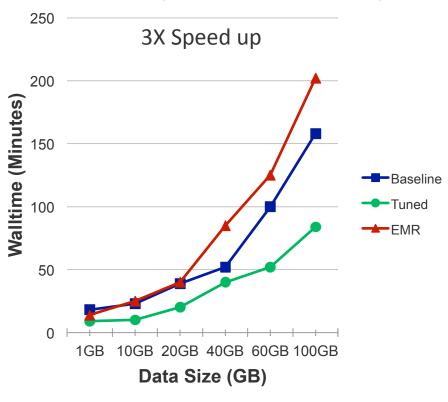
- 2-3 orders of magnitude slower than MPI
- IO optimization, e.g., reduce data copying
- Some problems do not easily fit into map/reduce framework, e.g., graph-based algorithms
- Runs on AWS, but cost $$$ if not optimized

Optimizing BioPig

Time saving
(60GB)

100 8 17 3 11 8 53

Baseline Compression Block Size Map Side Spill JVM GC SlowStart Final

miniutes

Still very low efficiency!

3X Speed up

Walltime (Minutes)

Baseline
Tuned
EMR

Data Size (GB)

1GB 10GB 20GB 40GB 60GB 100GB

Lizhen Shi, Weikuan Yu @FSU

Addressing big data: Apache Spark

- New **scalable** programming paradigm
  - Compatible with Hadoop-supported storage systems
- Improves **efficiency** through:
  - In-memory computing primitives
  - General computation graphs
- Improves **usability** through:
  - Rich APIs in Java, Scala, Python
  - Interactive shell

Goal: Metagenome read clustering

- **Data characteristics:**
  - Total data size typically 100Gb – 1Tb
  - >1 billion short pieces (reads, each 100-200bp)
  - >1,000 different species, some species are more similar than others
  - Sequence errors 1-2%
- **Proposed approach: Divide-and-conquer**
  - Cluster reads from each genome (Clustering)
  - Assemble each cluster in parallel (Assembly)

Read clustering with Spark: idea

- **Local information: overlap**

  Spark Metagenome

  Metagenome Assemb

  genome Assembler

  Me, Too!

- **Global information: covariance**

Me 2

Sample1	Sample2	Sample3
1 Spark	2 Spark	10 Spark
1 Metagenome	2 Metagenome	10 Metagenome
1 Assembler	2 Assembler	10 Assembler
10 de	2 de	5 de
10 bruijn	2 bruijn	5 bruijn

Toy test datasets

- **Species:**
  - 6 bacterial species
  - Synthetic communities with random proportions of each

- **Data: single genome sequence data (synthetic & real reads)**

NMI: normalized mutual information

Mutual Information: How pure the different clusters are

$$\text{NMI}(\Omega, \mathbb{C}) = \frac{I(\Omega; \mathbb{C})}{[H(\Omega) + H(\mathbb{C})]/2}$$
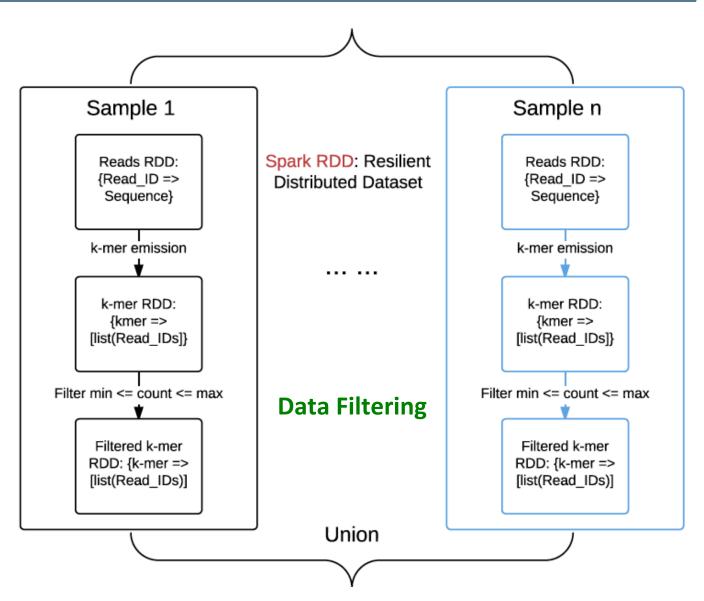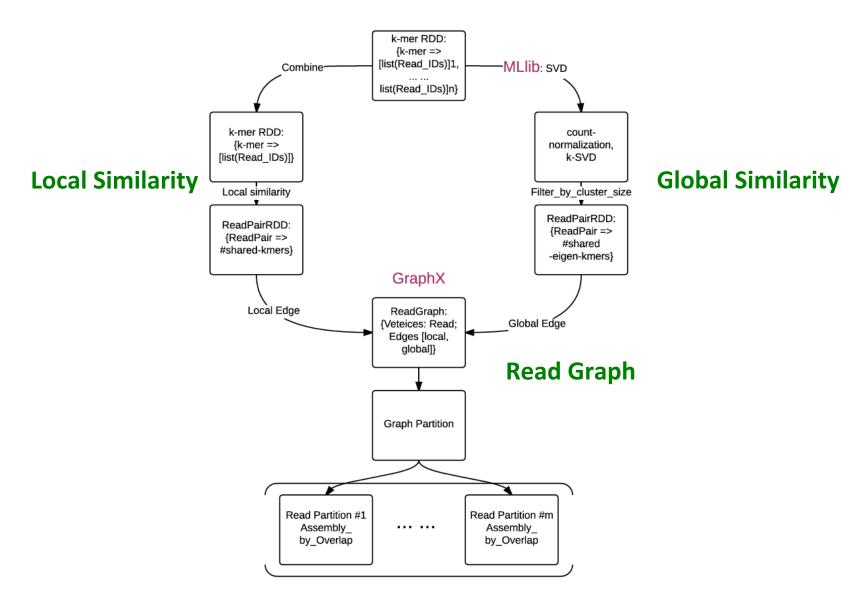
Entropy: Penalizes having small clusters

http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

Testing Environments
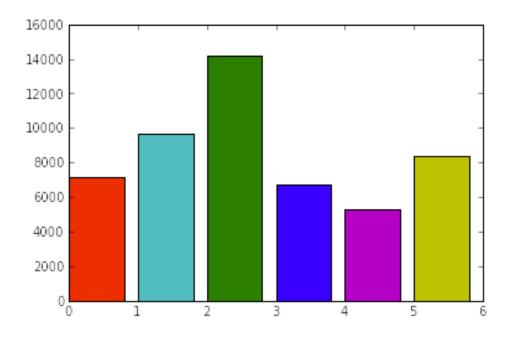
- **Local**
  - *Algorithm development*
  - 32-core
  - 256GB memory
- **HPC-Lawrencium**
  - *Small scale analysis*
  - CPU: INTEL XEON E5-2670
  - 16-core per node
  - 64GB memory per node
  - Infiniband FDR
- **NERSC-Cori**
  - *Large scale analysis*
  - CPU: Cray Haswell
  - 32-core per node
  - 128GB memory per node
  - Cray Aries high-speed interconnect with Dragonfly topology

Local Similarity

In ideal situation (no errors, no repetitive sequences, sufficient sequence coverage): read clustering with local similarity works perfectly.

With real-world situations where:
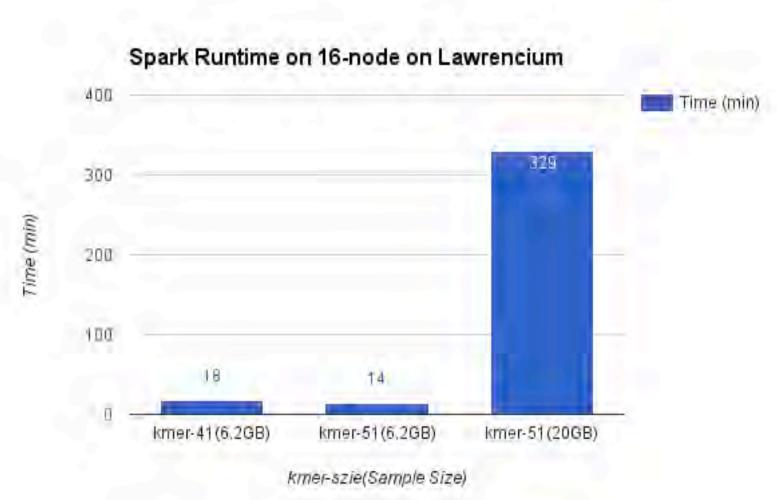
Sequencing coverage is low, many small clusters may form from a same genome, leads to

False Negatives

Different genome share sequences, they can fall into the same cluster, leads to

False Positives.

Reads of the same color belong to the same genome

Some performance metrics

Spark Runtime on 16-node on Lawrencium

Needs 500-700X of memory – optimization is needed

Xiandong Meng @JGI
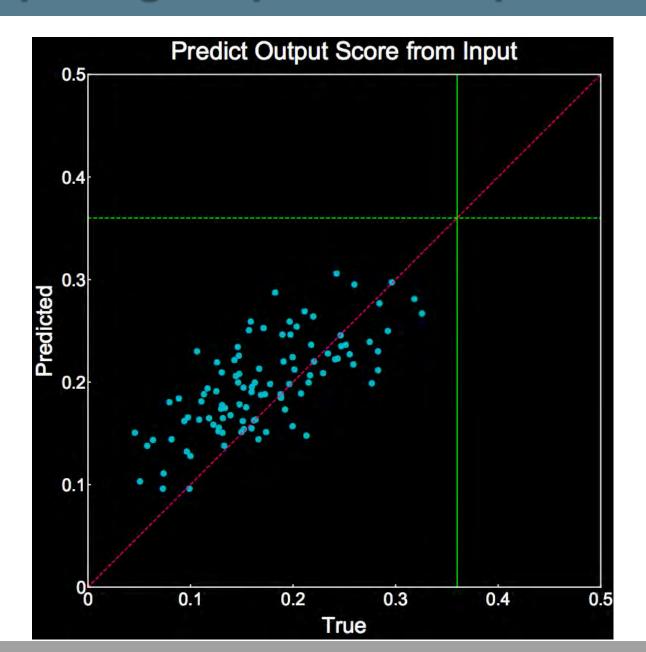
Global Similarity: input Parameters

No samples	1-1000
K-mer Length	20-50
K-Means Clusters	10-500
Eigen K-mers to sample	1-10,000
Eigen Reads to sample	100-60,000
Global Weight	0-150
Power Iteration Clusters	10-150
Power Iteration Steps	0-50

Exploring the parameter space

Predict Output Score from Input

Jordan Hoffman
@Harvard

Overall impression of Spark

✓ **Easy to develop**

? **Robust**

? **Scale to big data**

? **Efficient**

- ▪ VS Hadoop/PIG
- ▪ VS MPI

Acknowledgements

BioPig Team
Henrik Nordberg
Kai Wang
Karan Bhat @ Amazon
Lizhen Shi, Weikuan Yu @ FSU

Spark Team
Xiandong Meng
Jordan Hoffman

Lisa Gerhardt , Evan Racah
Shane Cannon @ NERSC

Gary Jung, Greg Kurtzer
Bernard Li, Yong Qin @ HPC