ADAM: Fast, Scalable Genome Analysis

Frank Austin Nothaft
AMPLab, University of California, Berkeley, @fnothaft

with: Matt Massie, André Schumacher, Timothy Danford, Carl Yeksigian, Chris Hartl, Jey Kottalam, Arun Aruha, Neal Sidhwaney, Michael Linderman, Jeff Hammerbacher, Anthony Joseph, and Dave Patterson

https://github.com/bigdatagenomics
http://www.bdgenomics.org

What is in ADAM/BDG?

ADAM:
Core API + CLIs

bdg-formats:
Data schemas

avocado:
Distributed local assembler

RNAdam:
RNA analysis on ADAM

bdg-services:
ADAM clusters

xASSEMBLEx:
GraphX-based de novo assembler

Guacamole:
Distributed somatic caller

Design Goals

- Develop processing pipeline that enables efficient, scalable use of cluster/cloud
- Provide data format that has efficient parallel/distributed access across platforms
- Enhance semantics of data and allow more flexible data access patterns

Implementation Overview

- 27K lines of Scala code
- 100% Apache-licensed open-source
- 21 contributors from 8 institutions
- Working towards a production quality release late 2014

ADAM Stack

In-Memory RDD	▸ Transform records using ***Apache Spark*** ▸ Query with SQL using *Shark* ▸ Graph processing with *GraphX* ▸ Machine learning using *MLBase*
Record/Split	▸ Schema-driven records w/ ***Apache Avro*** ▸ Store and retrieve records using ***Parquet*** ▸ Read BAM Files using ***Hadoop-BAM***
File/Block	▸ ***Hadoop*** Distributed Filesystem ▸ Local Filesystem
Physical	▸ Commodity Hardware ▸ Cloud Systems - Amazon, GCE, Azure

Design Principles

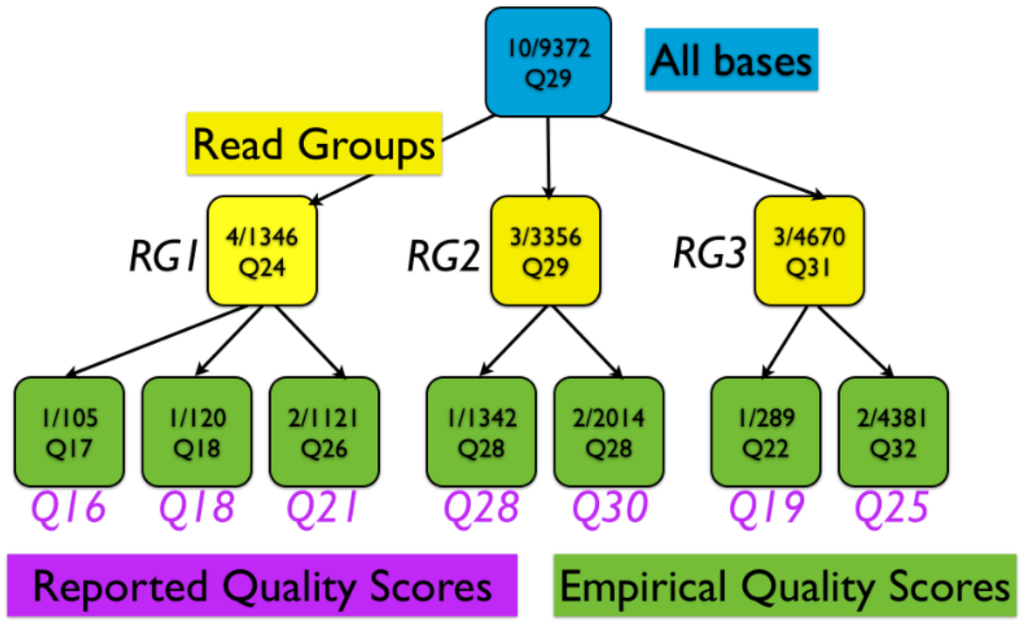
- Abstract as much as possible: schema oriented design makes format easy to evolve
- Provide rich and scalable APIs for manipulating and transforming genomic data and regions
- Don’t lock data in: play nicely with other tools

Parquet

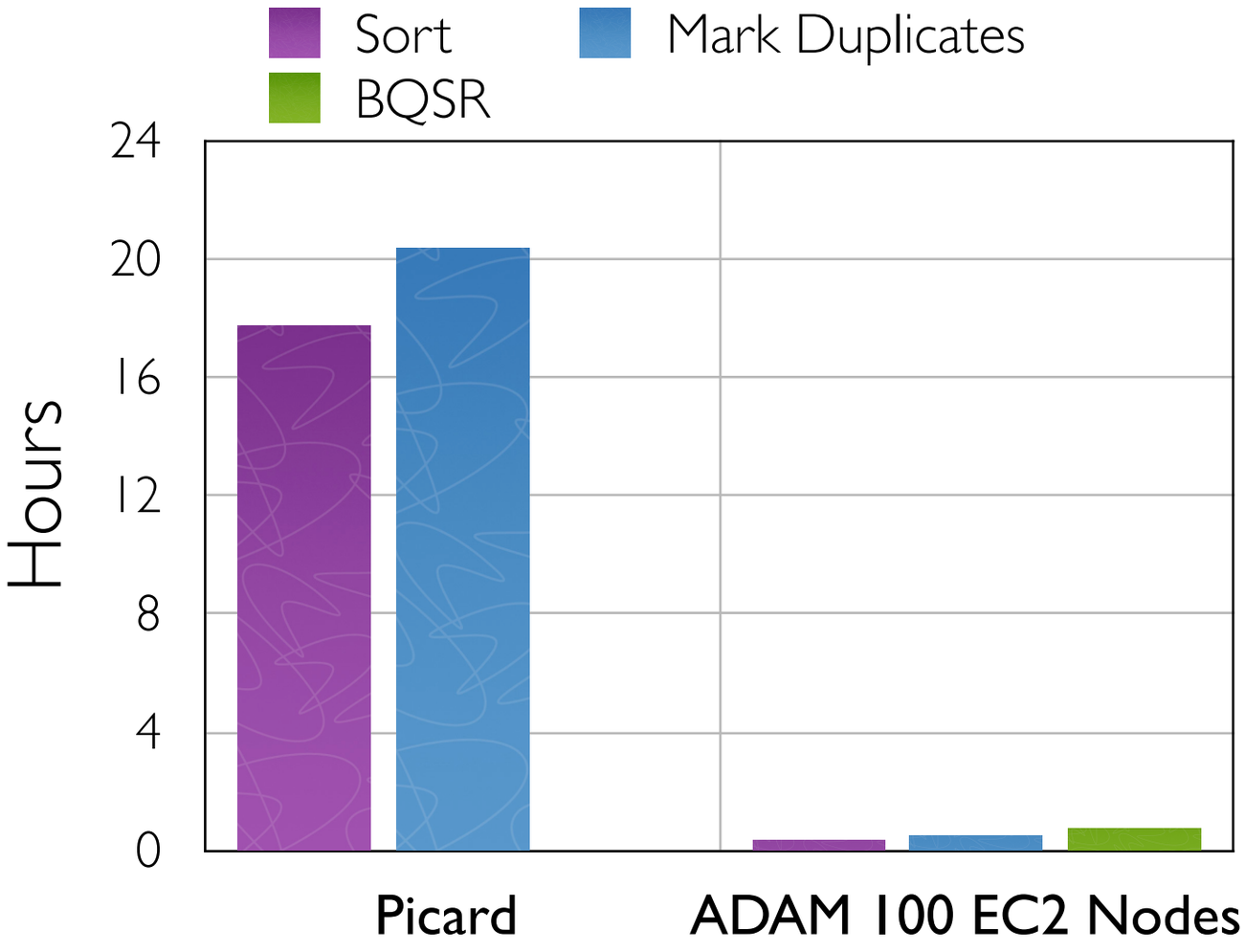
- OSS Created by Twitter and Cloudera, based on Google Dremel, just entered Apache Incubator
- Columnar File Format:
  - Limits I/O to only data that is needed
  - Compresses very well - ADAM files are 5-25% smaller than BAM files without loss of data
  - Fast scans - load only columns you need, e.g. scan a read flag on a whole genome, high-coverage file in less than a minute

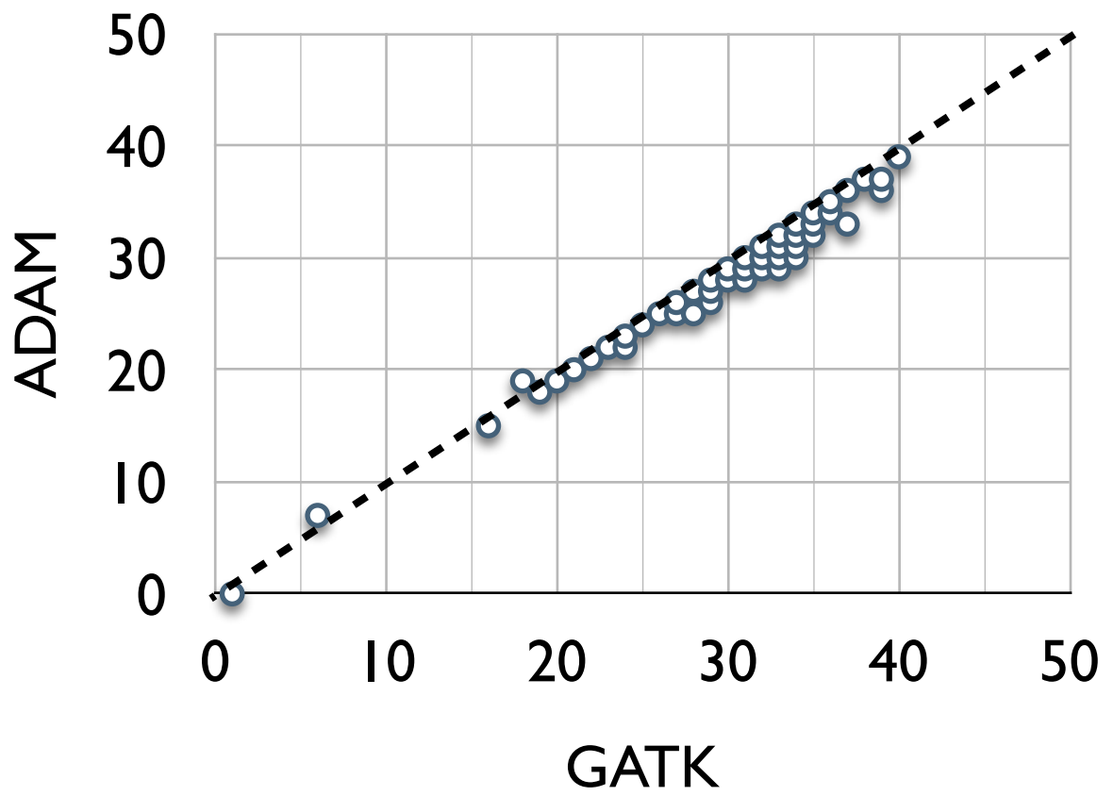
Scaling Genomics: BQSR

- Broadcast 3 GB table of variants, used for masking
- Break reads down to bases and map bases to covariates
- Calculate empirical values per covariate
- Broadcast observation, apply across reads



Performance/Acc'y





- Fully concordant with Picard for MarkDup, >99% concordant with GATK for BQSR

Future Work

- Pushing hard towards production release
- Are building out a complete analysis pipeline
- Plan to release Python bindings
- Work on interoperability with Global Alliance for Genomic Health API (http://genomicsandhealth.org/)

Call for contributions

- As an open source project, we welcome contributions
- We maintain a list of open enhancements at our Github issue trackers
- Github: https://www.github.com/bdgenomics
- UC Berkeley is looking to hire two full time engineers to support this work

Acknowledgements

- **UC Berkeley:** Matt Massie, André Schumacher, Jey Kottalam, Christos Kozanitis
- **Mt. Sinai:** Arun Ahuja, Neal Sidhwaney, Michael Linderman, Jeff Hammerbacher
- **GenomeBridge:** Timothy Danford, Carl Yeksigian
- **The Broad Institute:** Chris Hartl
- **Cloudera:** Uri Laserson
- **Microsoft Research:** Jeremy Elson, Ravi Pandya
- Michael Heuer
- And other open source contributors!

Acknowledgements

This research is supported in part by NSF CISE Expeditions Award CCF-1139158, LBNL Award 7076018, and DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, SAP, The Thomas and Stacey Siebel Foundation, Apple, Inc., C3Energy, Cisco, Cloudera, EMC, Ericsson, Facebook, GameOnTalis, Guavus, HP, Huawei, Intel, Microsoft, NetApp, Pivotal, Splunk, Virdata, VMware, WANdisco and Yahoo!.