Connecting big data to R, using SparkR, parquet and jupyter - a genomic example

Michal Okoniewski, Scientifc IT Services, ETH Zurich

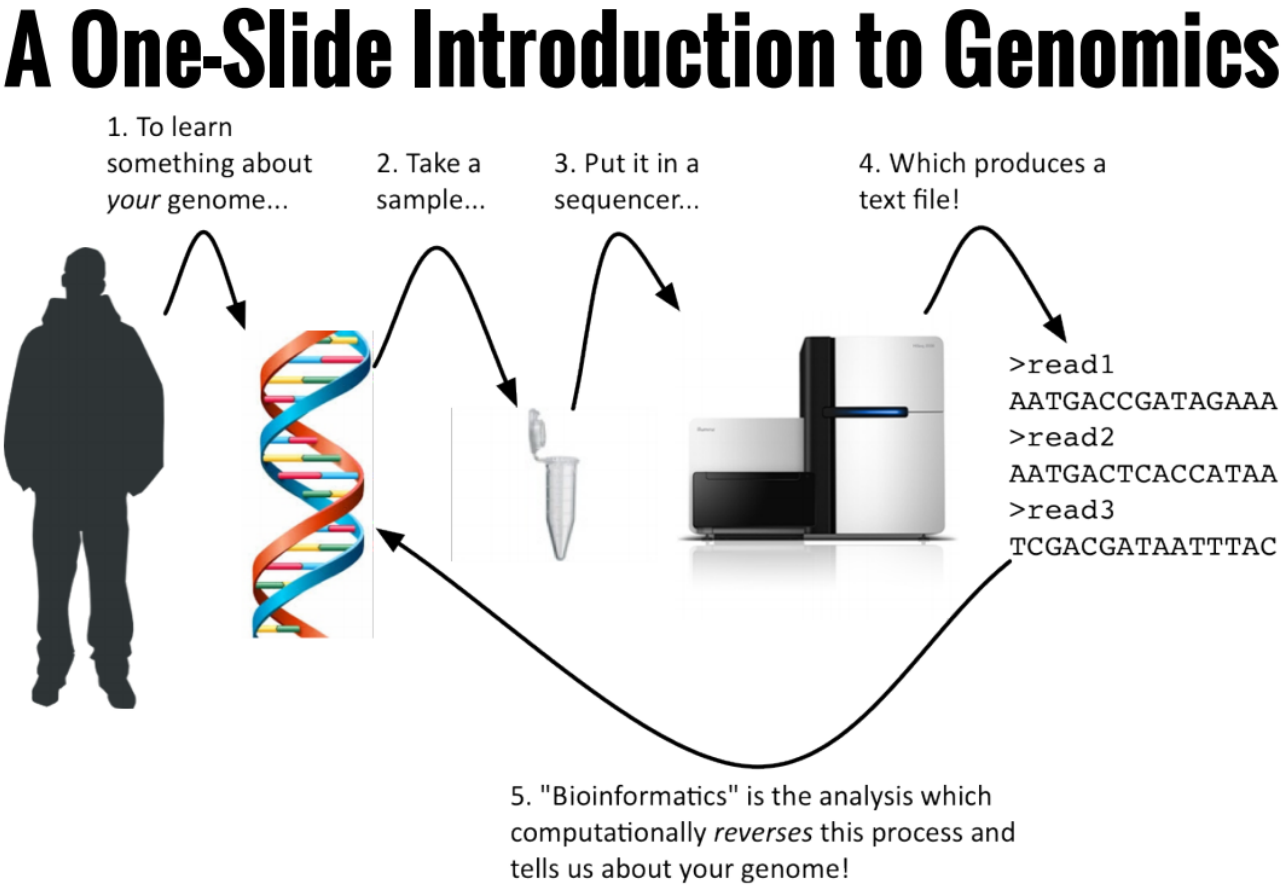
Questions

Can we analyze BIG DATA with R?

Are we technically ready for personalized, genomics-based medicine?

Is R useful for it?

Next-generation sequencing (NGS)

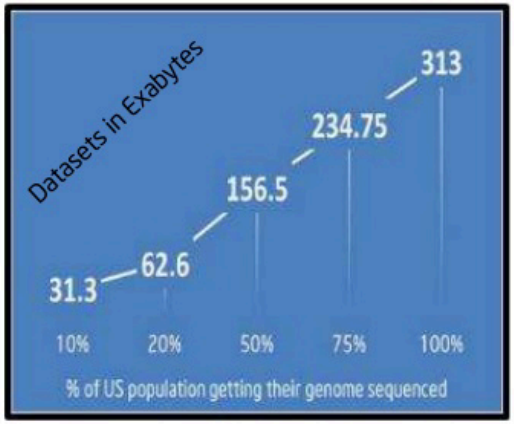


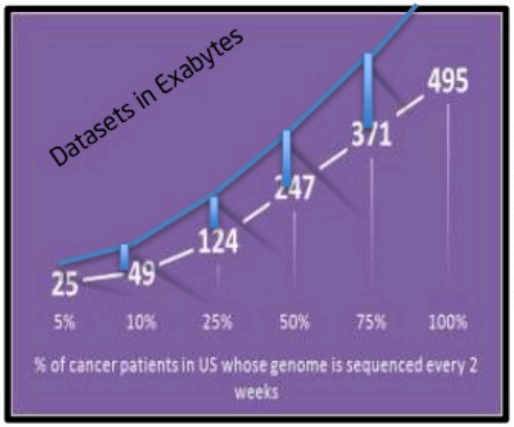
NGS as a big data problem

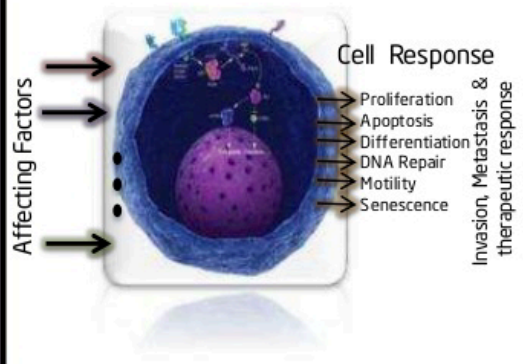
- How many full genomes may be sequenced with 30x coverage

NGS as a big data problem

The day when every newborn gets their DNA sequenced is not far away: http://www.nih.gov/news/health/sep2013/nhgri-04.htm.







313 *Exabytes*
if everyone in the US has their genes sequenced

495 *Exabytes*
if every cancer patient in the US has their genes sequenced every 2 weeks.

Images, Assays and Drug response data will push it further up as shown in Blue line

Complex interaction of *varied & changing* intrinsic and extrinsic factors determine cell response

With Genomic Data growing rapidly, hospitals and research centers need to access the local data (the ones not shared) and the centralized public/private data for various analysis and analytics for Genomic Research/Development/Medicine.

Compute has to be done "where data is" and need to be consistent locally and in the cloud.
Energy, Total Cost of Operation are key

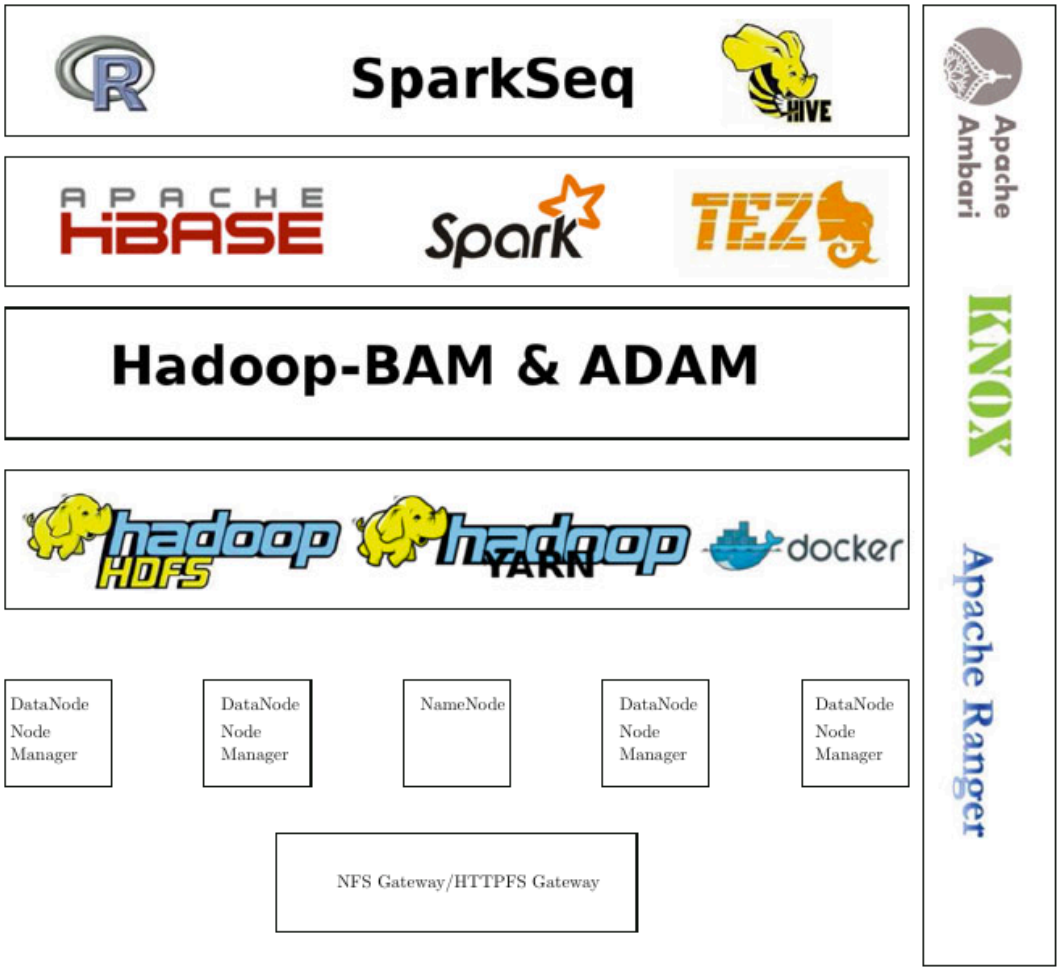
Source: Knights Cancer Institute, Oregon Health Sciences University & Intel

NGS data analysis – clinical applications

- Personalized/precision medicine

- Examples:
  - Personalized cancer therapy
  - Other “systemic diseases”
  - Newborn screening
  - Emergency medicine
  - Personalized food and diet
  - ….

Big data architecture for NGS



Proof-of-concept paper

BIOINFORMATICS APPLICATIONS NOTE

Vol. 30 no. 18 2014, pages 2652–2653
doi:10.1093/bioinformatics/btu343

Genome analysis

Advance Access publication May 19, 2014

SparkSeq: fast, scalable and cloud-ready tool for the interactive genomic data analysis with nucleotide precision

Marek S. Wiewiórka[1,*], Antonio Messina[2], Alicja Pacholewska[3,4], Sergio Maffioletti[2], Piotr Gawrysiak[1] and Michał J. Okoniewski[5]

Big Data Genomics project and ADAM standard

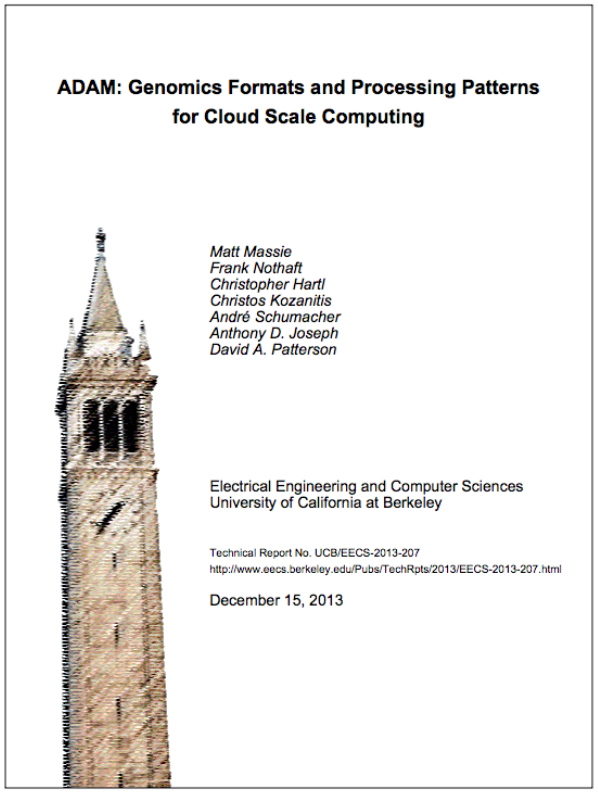
Collaboration of the AMPLab at UC Berkeley, Genome Informatics Lab at UC Santa Cruz, Icahn School of Medicine at Mount Sinai, Microsoft Research, Cloudera, and the Broad Institute.

http://bdgenomics.org/

Matt Massie

http://www.hammerlab.org/

Jeff Hamerbacher


ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing

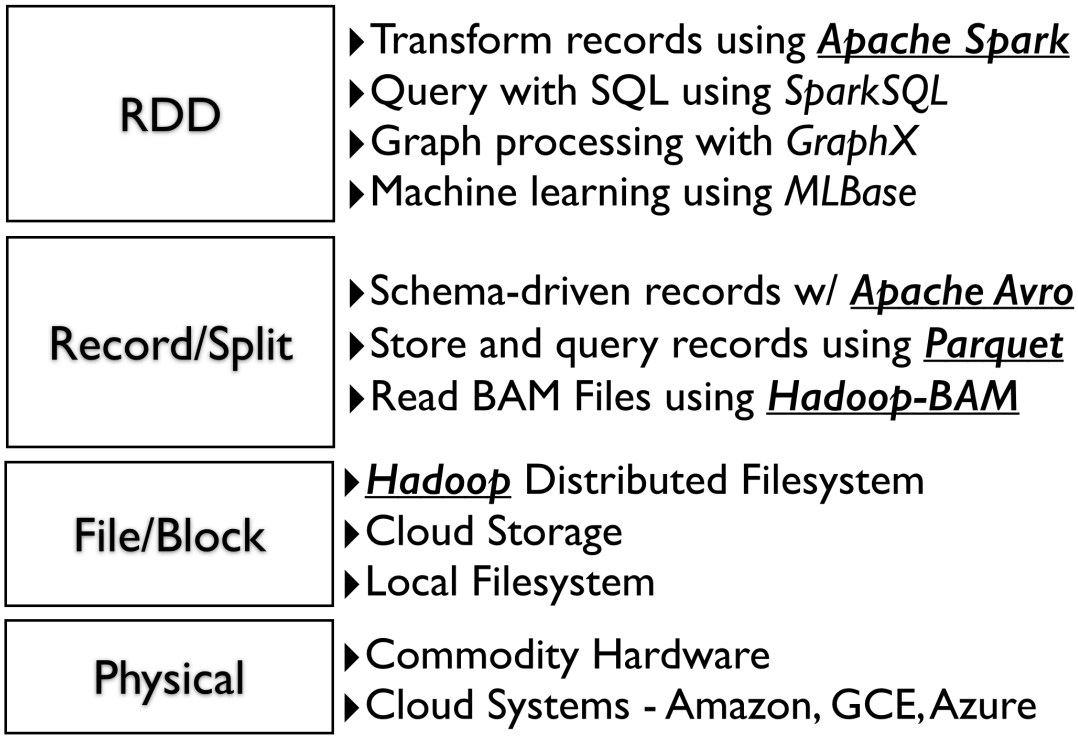
Matt Massie
Frank Nothaft
Christopher Hartl
Christos Kozanitis
André Schumacher
Anthony D. Joseph
David A. Patterson

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-207
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-207.html

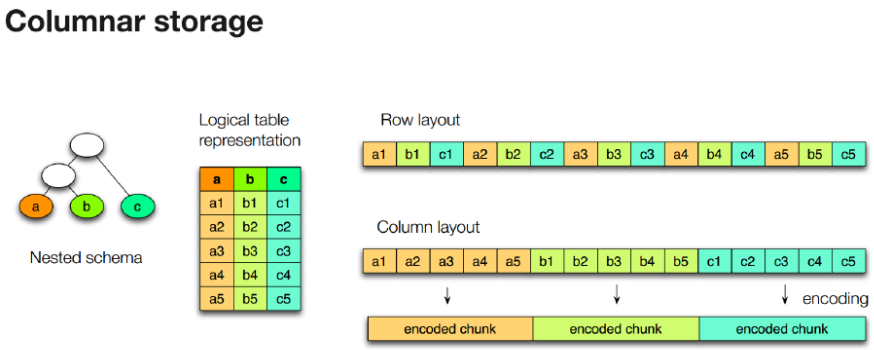
December 15, 2013

Big Data Genomics project and ADAM standard



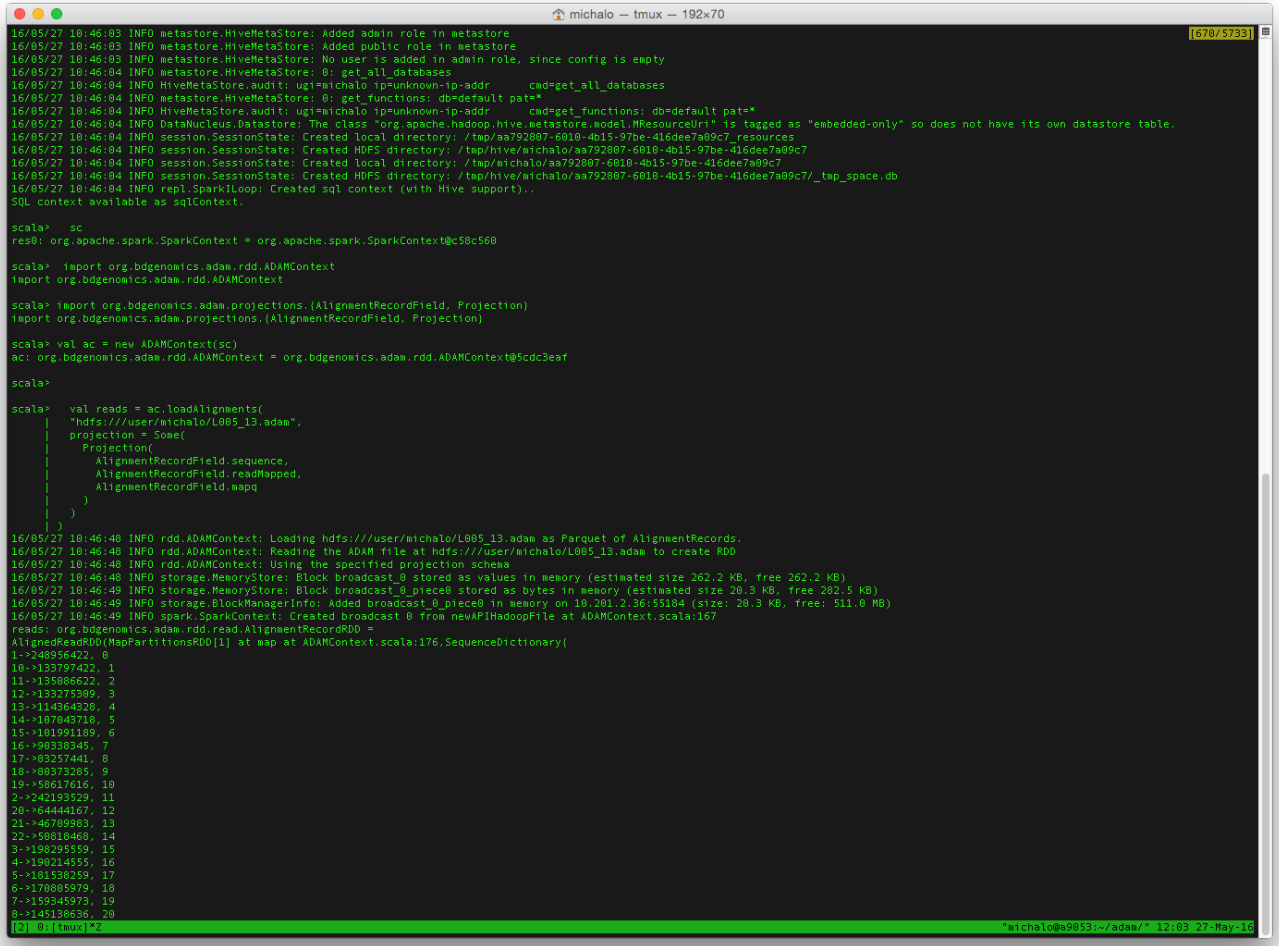
ADAM: Avro + Parquet

```
{"namespace": "example.avro",
"type": "record",
"name": "User",
"fields": [
    {"name": "name", "type": "string"},
    {"name": "favorite_number",  "type": ["int", "null"]},
    {"name": "favorite_color", "type": ["string", "null"]}
]
}
```



http://www.slideshare.net/cloudera/hadoop-summit-36479635

Running ADAM: adam-shell

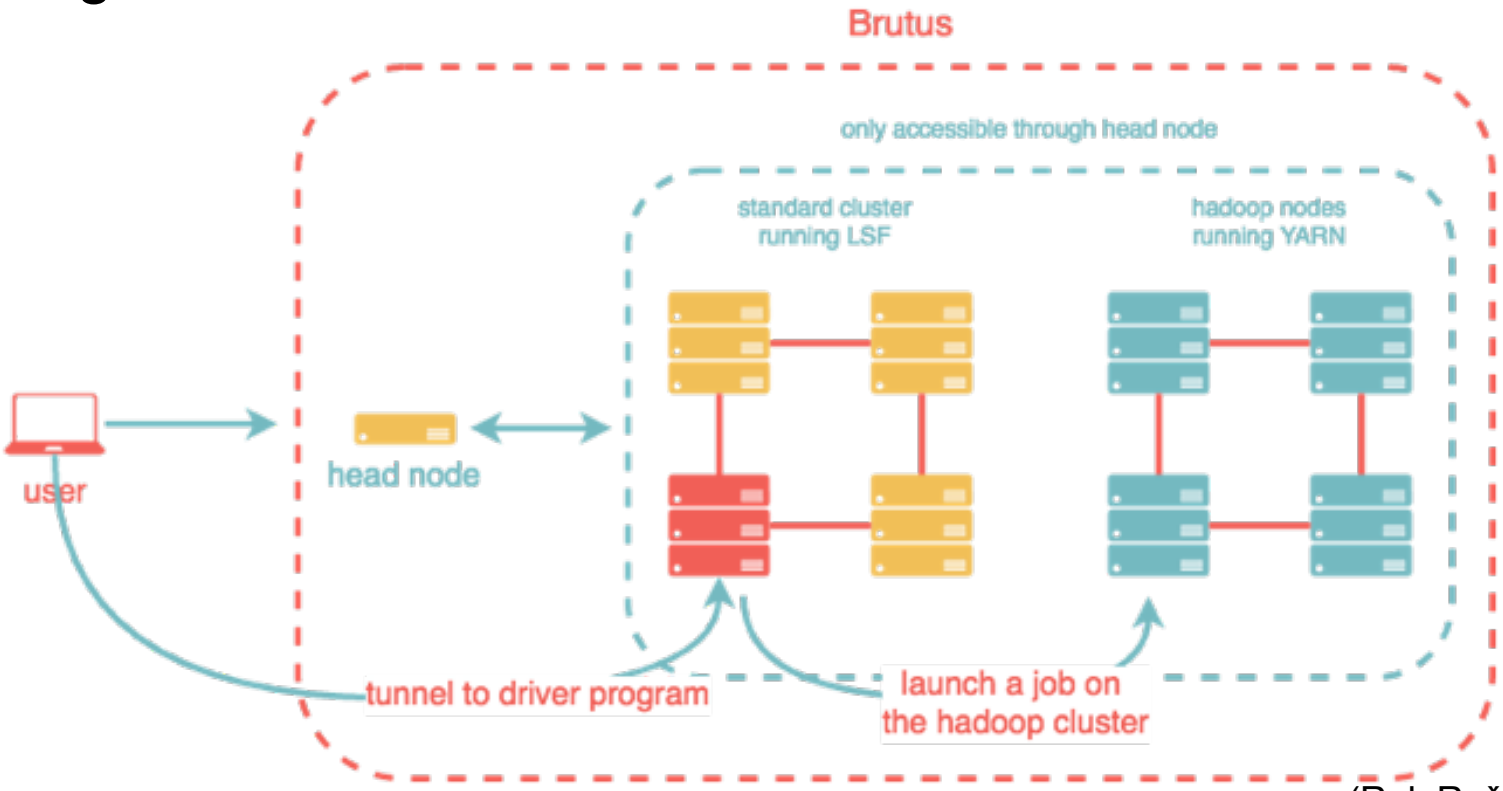


adam-submit: ADAM command line tools

```
[michalo@a9078 bin]$ ./adam-submit --help
Using ADAM_MAIN=org.bdgenomics.adam.cli.ADAMMain
Using SPARK_SUBMIT=/cluster/apps/spark/spark-current/bin/spark-submit
Picked up _JAVA_OPTIONS: -XX:ParallelGCThreads=1
Picked up _JAVA_OPTIONS: -XX:ParallelGCThreads=1
16/05/25 15:04:00 INFO cli.ADAMMain: ADAM invoked with args: "--help"

      e         888~-_          e             e    e
     d8b        888   \        d8b           d8b  d8b
    /Y88b       888    |      /Y88b         d888bdY88b
   /  Y88b      888    |     /  Y88b       / Y88Y Y888b
  /____Y88b     888   /     /____Y88b     /   YY   Y888b
 /      Y88b    888_-~     /      Y88b   /          Y888b

Usage: adam-submit [<spark-args> --] <adam-args>

Choose one of the following commands:

ADAM ACTIONS
             depth : Calculate the depth from a given ADAM file, at each variant in a VCF
       count_kmers : Counts the k-mers/q-mers from a read dataset.
 count_contig_kmers : Counts the k-mers/q-mers from a read dataset.
         transform : Convert SAM/BAM to ADAM format and optionally perform read pre-processing transformations
          adam2fastq : Convert BAM to FASTQ files
            plugin : Executes an ADAMPlugin
           flatten : Convert a ADAM format file to a version with a flattened schema, suitable for querying with tools like Impala

CONVERSION OPERATIONS
          vcf2adam : Convert a VCF file to the corresponding ADAM format
          adam2vcf : Convert an ADAM variant to the VCF ADAM format
         anno2adam : Convert a annotation file (in VCF format) to the corresponding ADAM format
        fasta2adam : Converts a text FASTA sequence file into an ADAMNucleotideContig Parquet file which represents assembled sequences.
       adam2fasta : Convert ADAM nucleotide contig fragments to FASTA files
     features2adam : Convert a file with sequence features into corresponding ADAM format
        wigfix2bed : Locally convert a wigFix file to BED format
    fragments2reads : Convert alignment records into fragment records.
    reads2fragments : Convert alignment records into fragment records.

PRINT
            print : Print an ADAM formatted file
       print_genes : Load a GTF file containing gene annotations and print the corresponding gene models
          flagstat : Print statistics on reads in an ADAM file (similar to samtools flagstat)
        print_tags : Prints the values and counts of all tags in a set of records
          listdict : Print the contents of an ADAM sequence dictionary
       allelecount : Calculate Allele frequencies
              view : View certain reads from an alignment-record file.
```

Hadoop cluster at ETH

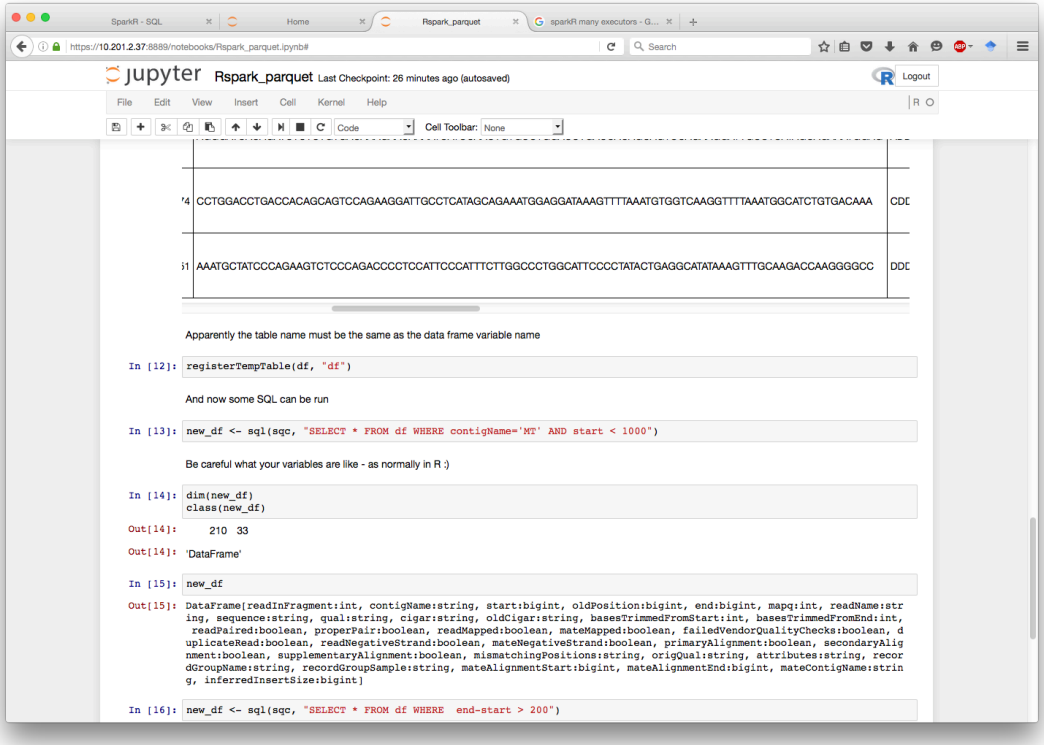
- Fragment of the Brutus cluster



(Rok Roškar)

Using SparkR for secondary analyses

- Via Parquet files and SQL



The same thing in scala

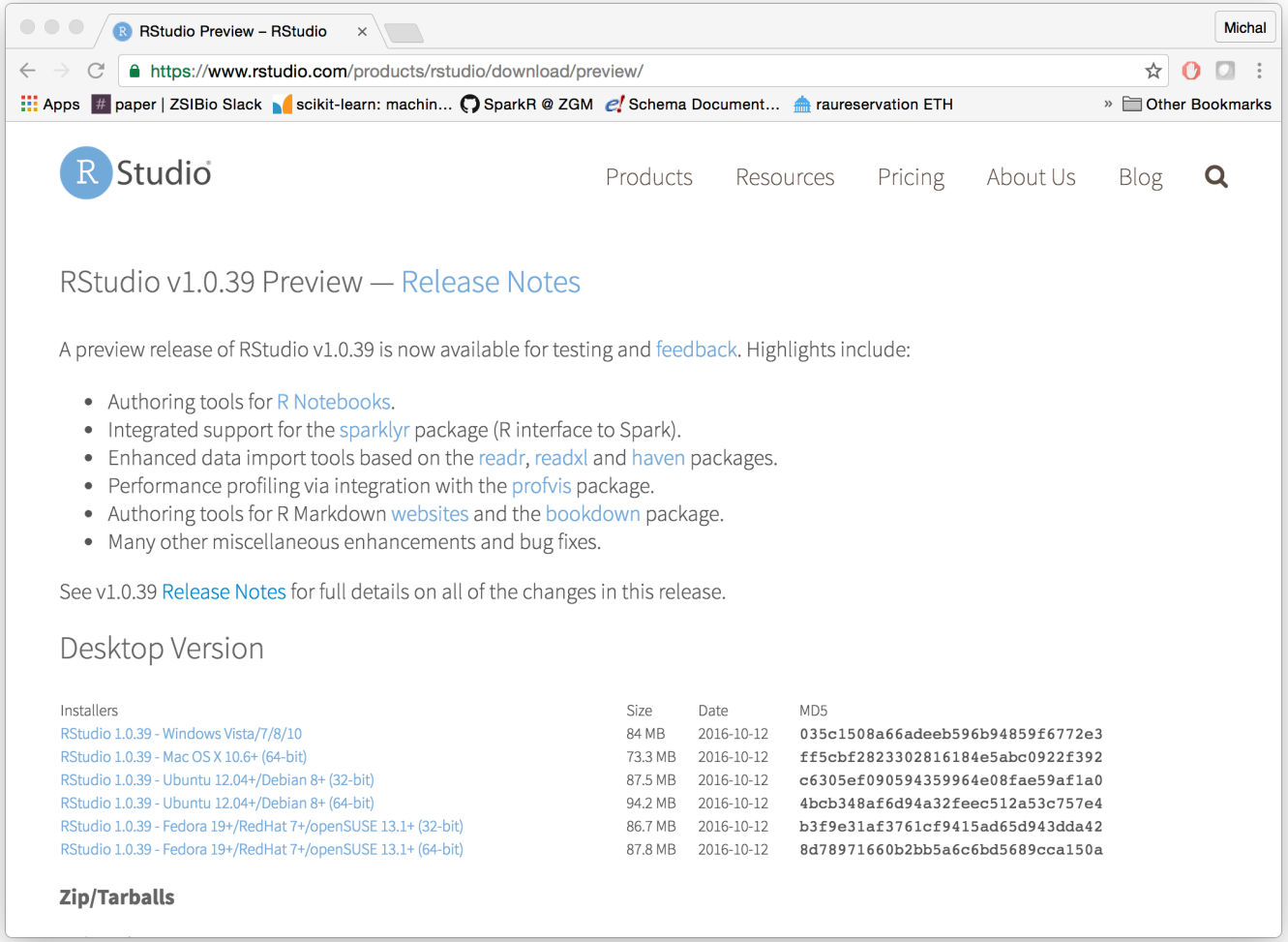
```
val df = sqlContext.read.parquet("hdfs:///user/michalo/cc5Ago2KO_2MAligned.out.sorted.adam")
df.registerTempTable("df")
df.printSchema
val myout = sqlContext.sql("SELECT count(*) FROM df WHERE `end`-`start` > 90")
myout.collect
```

Alternative to SparkR: Rstudio sparklyr

- plyr => dplyr => sparklyr

- Connect to Spark from R — like dplyr backend
- Filter and aggregate Spark datasets then bring them into R for analysis and visualization.
- distributed machine learning from R with Spark MLlib or H2O Sparkling Water
- Create extensions that call the full Spark API and provide interfaces to Spark packages

Info release yesterday….



Summary

- Genomics started generating real big data
- Mechanisms for storing and processing are being developed
- Connecting to “good-old-memory-limited” R is needed
- With SparkR it is possible
  - E.g. to query parquet files on R level and get manageable data frames
- RStudio has just mimicked the receipe described here
- It is not only for genomic use, but many big data cases

Acknowledgements

- Rok Roškar – setting up the Spark and jupyter configuration
- HPC group, Scientific IT Services ETH

- Tomasz Gambin, Marek Wiewiorka– Politechnika Warszawska, Instytut Informatyki

ETH zürich

Thank you for your attention!
Questions/comments?