USING HADOOP TO IDENTIFY FALSE POSITIVES IN PYROPRINTING

Colin Adams & Skyler Gordon

IAB Presentation

Spring 2016

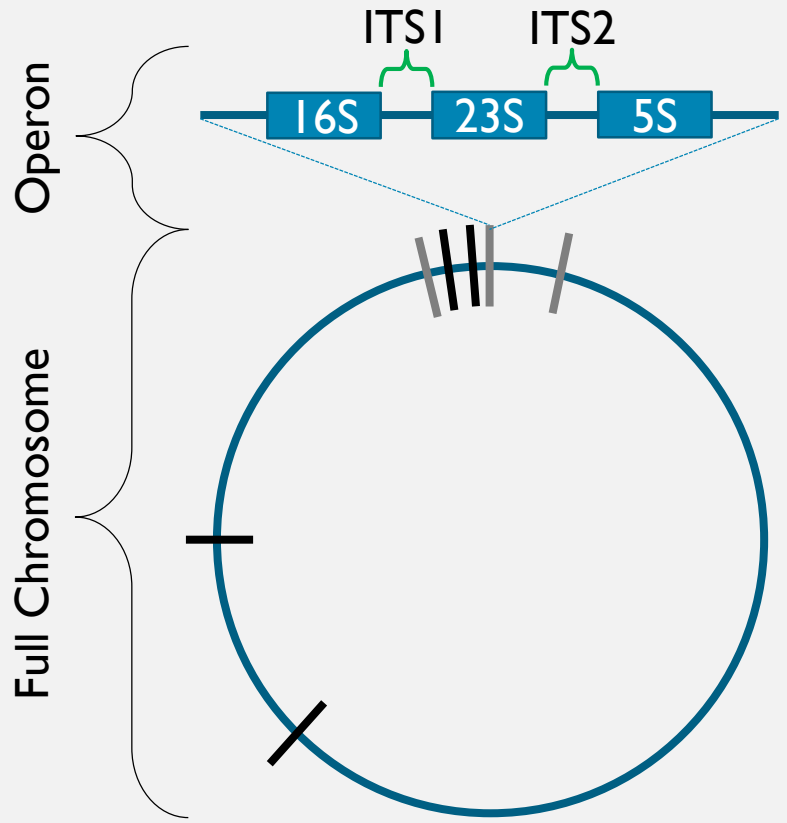
AGENDA

- Background
  - *E. coli* and Pyroprinting
- Implementation
  - Hadoop
- Results

BACKGROUND

E. COLI DNA

- Structure
  - Circular chromosome
  - Seven copies of rRNA operons
  - Operon structure
- **Interested in copies of ITS1 and ITS2 alleles**
- Allele Ratio
  - 7 - all seven alleles are same sequence
  - 6:1 – six of one allele, one different
  - 5:2 – five of one allele, two of another
  - …
  - 1:1:1:1:1:1:1 – every allele is different



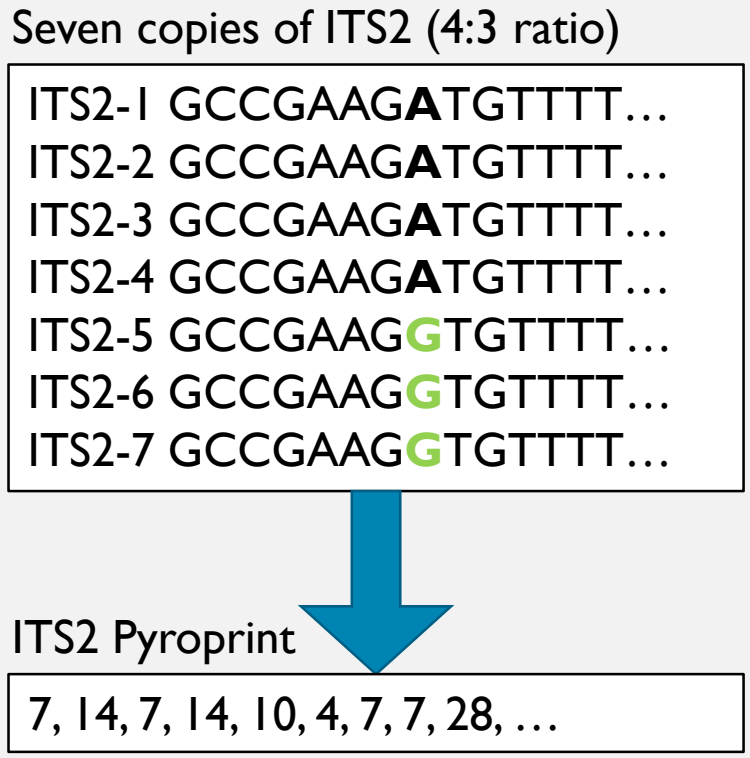
Adapted from Black, M., et al. Journal of Microbiological Methods (2014)

PYROPRINTING

- Rapid, inexpensive strain typing method
- Creates fingerprint from ITS1 or ITS2 alleles

- Two *E. coli* with matching pyroprints considered same strain

Question: How often do the pyroprints between *E. coli* match but the allele ratio or sequences differ? In other words, how frequent is a false positive?

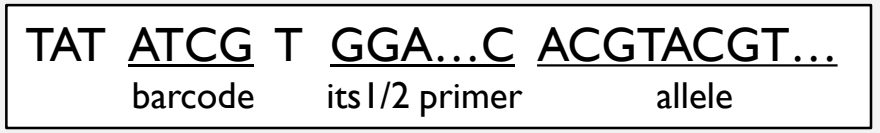


Adapted from Black, M., et al. Journal of Microbiological Methods (2014)

SOLUTION STRUCTURE

- Determine alleles and allele ratio for each *E. coli* isolate
  - If two isolates have matching pyroprints but different alleles/ratio → false positive

- Process
  - Choose 100 *E. coli* isolates
  - Amplify & Modify ITS Regions (PCR)
  - Have sequenced (Illumina/Mr. DNA)
    - 5-20 million sequences
  - Analyze sequences (Hadoop)
    - Count alleles for each isolate and region
    - Determine allele ratio

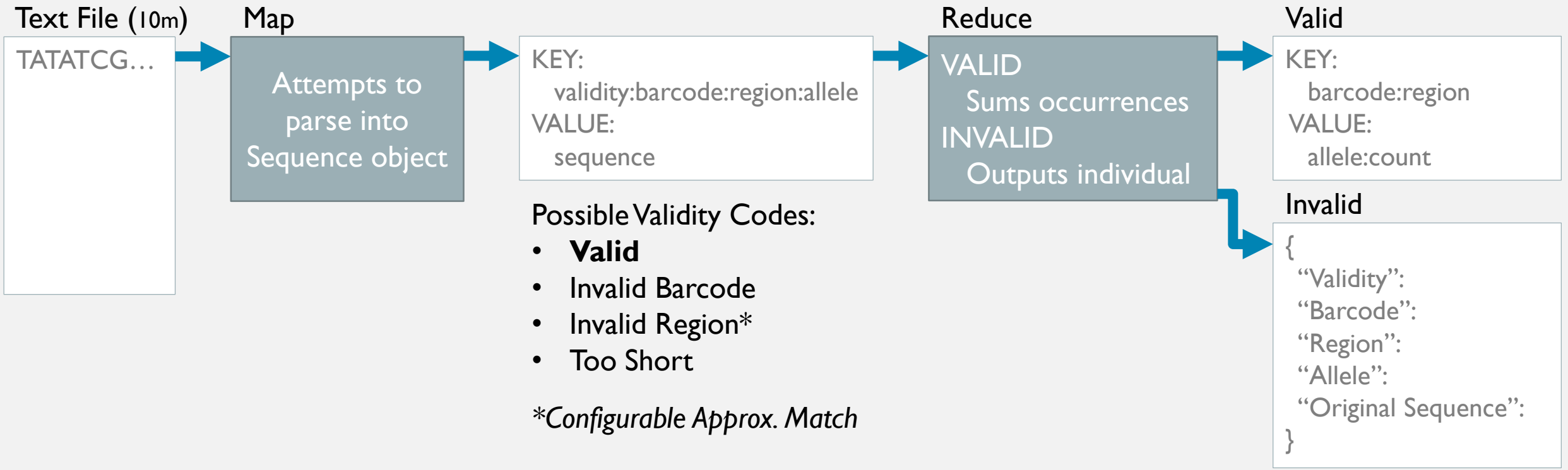


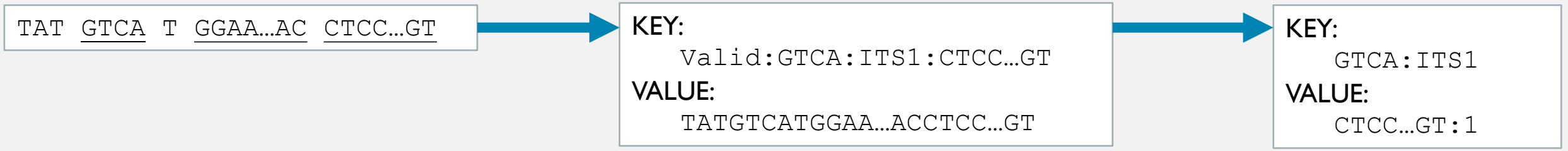
IMPLEMENTATION

HADOOP BACKGROUND

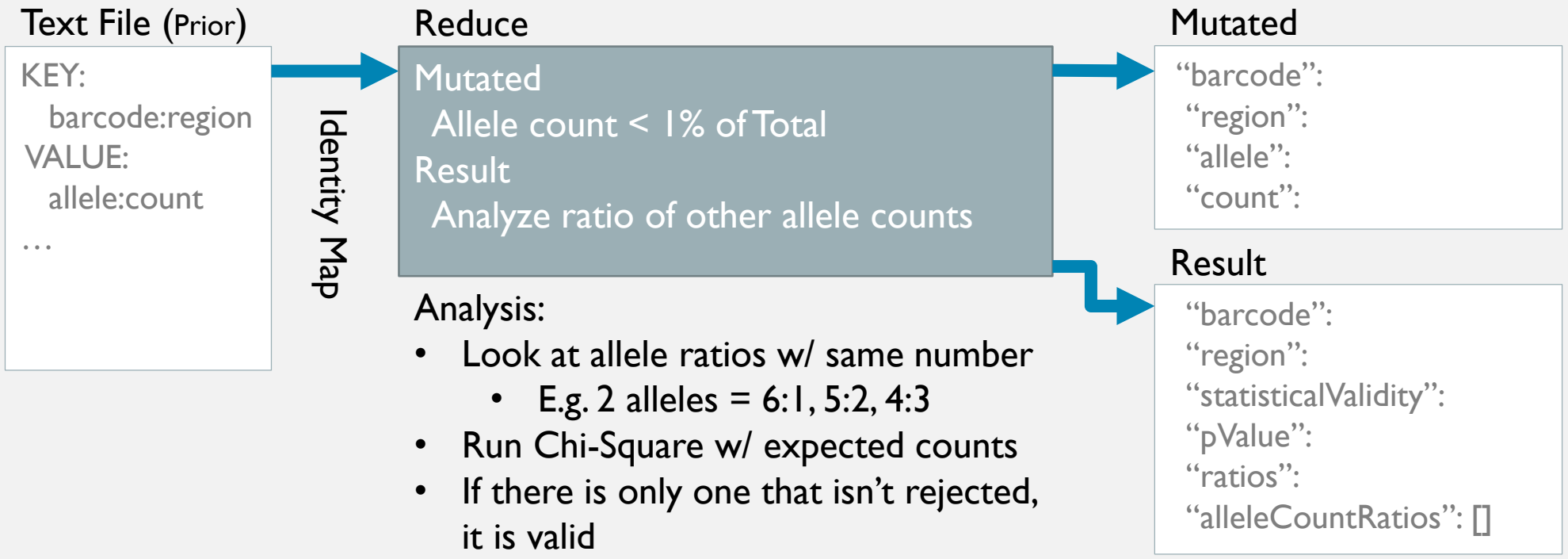
- Distributed storage and processing framework
- Simple data processing pipeline
  - Map(obj) -> <key, value>…
    - Transforms each input object into any number of <key, value> pairs
  - Shuffle
    - Map() output is shuffled so all objects with same key are grouped for Reduce()
  - Reduce(key, values…) -> <key, value>…
    - Performs reduction on values with matching key and outputs any number of results

FIRST HADOOP JOB





SECOND HADOOP JOB




```
GTCA:ITS1    CTCC…GT:40
GTCA:ITS1    CTCC…AT:30
```

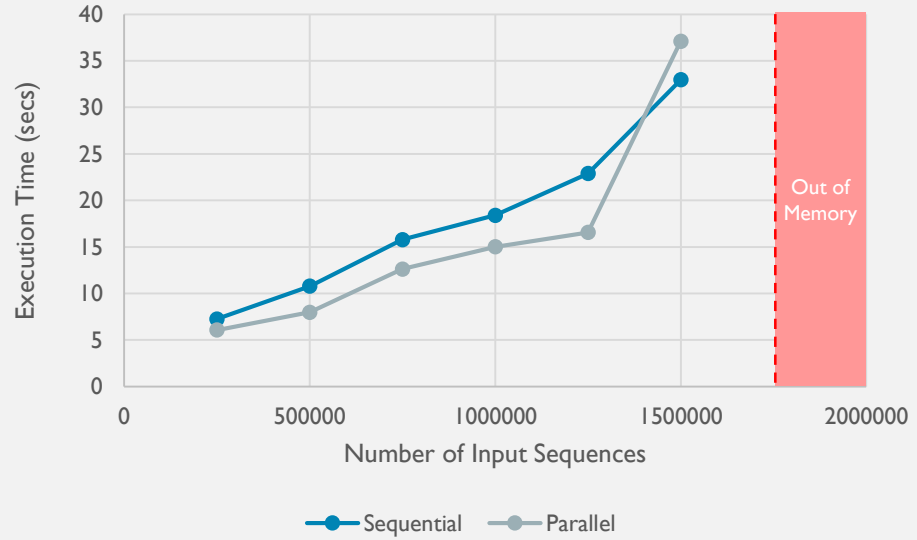
```
{
  "barcode": GTCA,
  "region": ITS1,
  "statisticalValidity": true,
  "pValue": 1.0,
  "ratios": [4:3],
  "alleleCountRatios": [
    {"allele": "CTCC…GT", "count": 40, "ratio": 4},
    {"allele": "CTCC…AT", "count": 30, "ratio": 3}
  ]
}
```

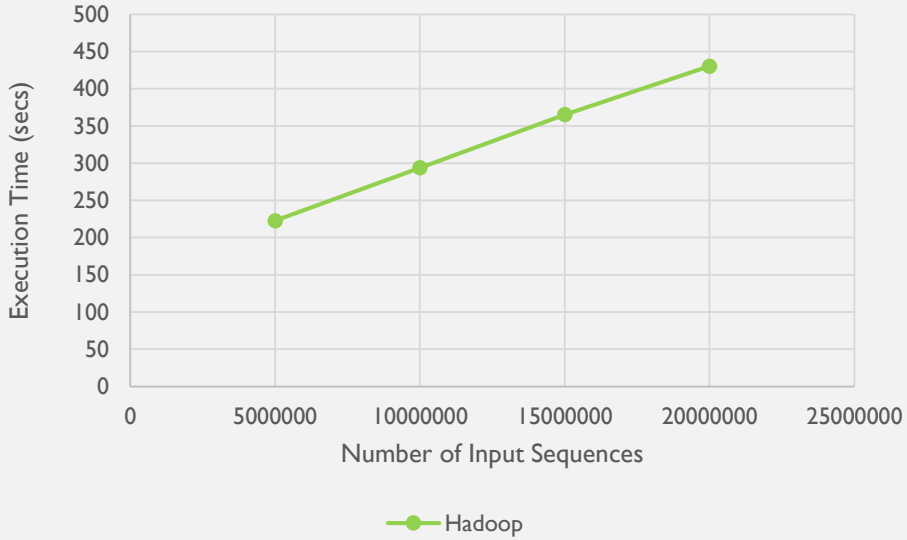
RESULTS

RAW RESULTS


```
{"catalogNumber":9,"region":"ITS1","strainName":"Av-009","statisticallyValid":false,
 "ratios":["THREE_ONE_ONE_ONE_ONE"],"pValue":0.0,"barcode":"GTCA",
 "alleleCountRatios":[
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGAAGTGCTCACACAGATTGTCTGATAGAAAGTGAAAAGCAAGGCGT","count":6483,"ratios":[3]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGAAGTGCTCACACAGATTGTCTGATGAAAATGAGCAGTAAAACCTCT","count":1370,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTACTTTGCAGTGCTCACACAGATTGTCTGATGAAAATGAGCAGTAAAA","count":1072,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGCAGTGCTCACACAGATTGTCTGATGAAAAGTAAATAGCAAGG","count":227,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTACTTTGCAGTGCTCACACAGATTGTCTGATAGAAAGTGAAAAGCAA","count":210,"ratios":[1]}
]}
{"catalogNumber":9,"region":"ITS1","strainName":"Cw-053","statisticallyValid":false,
 "ratios":["THREE_ONE_ONE_ONE_ONE"],"pValue":0.0,"barcode":"CGAG",
 "alleleCountRatios":[
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGAAGTGCTCACACAGATTGTCTGATAGAAAGTGAAAAGCAAGGCGT","count":6497,"ratios":[3]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGAAGTGCTCACACAGATTGTCTGATGAAAATGAGCAGTAAAACCTCT","count":1413,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTACTTTGCAGTGCTCACACAGATTGTCTGATGAAAATGAGCAGTAAAA","count":1035,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTACTTTGCAGTGCTCACACAGATTGTCTGATAGAAAGTGAAAAGCAA","count":201,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGCAGTGCTCACACAGATTGTCTGATGAAAAGTAAATAGCAAGG","count":171,"ratios":[1]}
]}
{"catalogNumber":9,"region":"ITS1","strainName":"Pg-085","statisticallyValid":false,
 "ratios":["THREE_ONE_ONE_ONE_ONE"],"pValue":0.0,"barcode":"GCCA",
 "alleleCountRatios":[
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGAAGTGCTCACACAGATTGTCTGATAGAAAGTGAAAAGCAAGGCGT","count":6269,"ratios":[3]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGAAGTGCTCACACAGATTGTCTGATGAAAATGAGCAGTAAAACCTCT","count":1276,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTACTTTGCAGTGCTCACACAGATTGTCTGATGAAAATGAGCAGTAAAA","count":1030,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTACTTTGCAGTGCTCACACAGATTGTCTGATAGAAAGTGAAAAGCAA","count":222,"ratios":[1]},
   {"allele":"CTCCTTACCTTAAAGAAGCGTTCTTTGCAGTGCTCACACAGATTGTCTGATGAAAAGTAAATAGCAAGG","count":154,"ratios":[1]}
]}
```

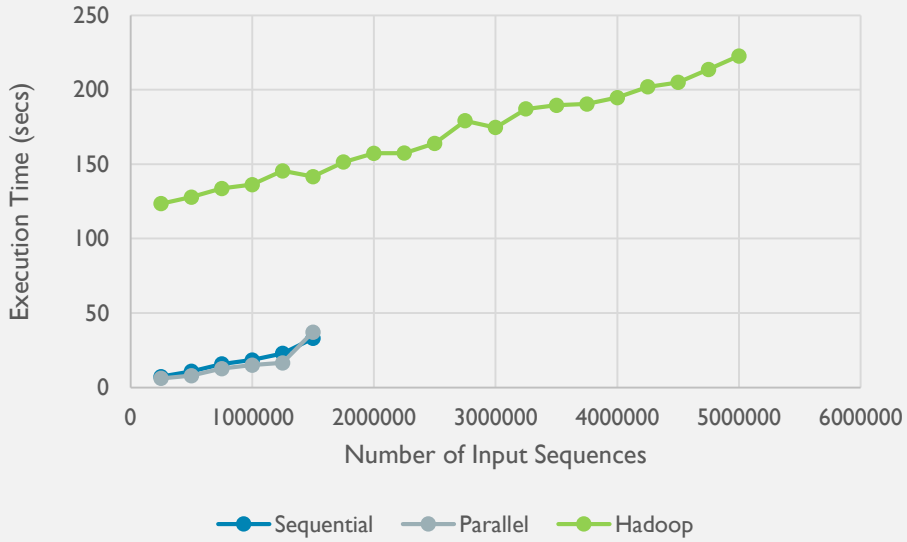
PERFORMANCE





Sequential & Parallel

- Single-Machine
  - Intel i7
  - 8 GB RAM
- Java 8 Streams API



Hadoop

- Google Cloud Compute
  - 1 Master
  - 3 Workers
  - 8 GB RAM each

LESSONS LEARNED

COLIN

- Technical
  - Hidden Complexity
  - Sequential → Hadoop
  - Cloud services
- Interdisciplinary
  - Be involved early and often
  - Many problems need CS
    - Efficiency

SKYLER

- Future Implications

THANK YOU