Extending the GATK to support Genomics X PRIZE variation comparisons

Brad Chapman
Bioinformatics Core
Harvard School of Public Health

23 February 2012

Genomics X PRIZE

ARCHON GENOMICS X PRIZE PRESENTED BY medco

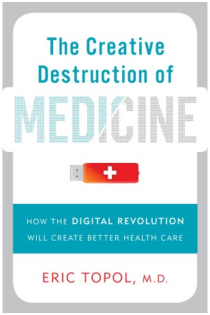
EdgeBio

http://genomics.xprize.org/

Clinical grade genome

- 98 percent genome coverage
- 1 error per million bases (SNPs + small indels)
- Full haplotype phasing
- Structural variations

Sequencing for patients


The Creative Destruction of
MEDICINE
HOW THE DIGITAL REVOLUTION WILL CREATE BETTER HEALTH CARE
ERIC TOPOL, M.D.

http://www.amazon.com/Creative-Destruction-Medicine-Digital-Revolution/dp/0465025501/

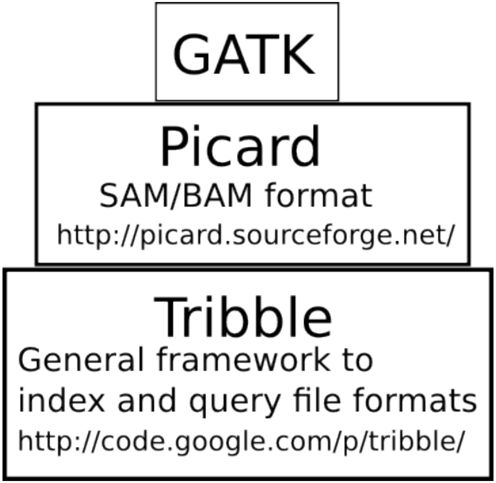
Reference genome

- Pooled 40kb fosmids
- Multiple technologies – Ilumina, SOLiD, Complete Genomics, Genotyping
- ABI validation

Variation evaluation

- Variant normalization
- Comparisons – concordant, discordant
- Haplotypes
- Custom structural variation comparison

GATK software stack


GATK
Picard
SAM/BAM format
http://picard.sourceforge.net/
Tribble
General framework to
index and query file formats
http://code.google.com/p/tribble/

GATK

- Toolkit for variation data
- Java API
  - Map/Reduce
  - File formats + variation objects

http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit

Clojure

- Dynamic programming language
- Java Virtual Machine
- Lisp
- Immutable data structures
- Functional programming

`http://clojure.org/`

Example code with Java

```
(defn java-string-interop []
  (let [example "GATc"]
    (println (.startsWith example "GAT"))
    (println (.toUpperCase example))))

user> (java-string-interop)
true
GATC
```

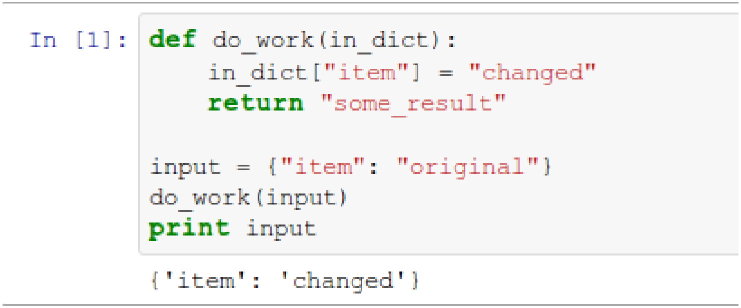
Development tools

Build	Leiningen
	https://github.com/technomancy/leiningen
Testing	Midje
	https://github.com/marick/Midje
Integration	Travis CI
	http://travis-ci.org/
Deployment	Heroku
	http://www.heroku.com/

Dependency resolution

```
:dependencies [[org.clojure/clojure "1.3.0"]
               [org.clojure/math.combinatorics "0.0.2"]
               [org.clojars.chapmanb/gatk "1.4.20"]
               [org.clojars.chapmanb/picard "1.58"]
               [incanter/incanter-core "1.3.0-SNAPSHOT"]
               [incanter/incanter-charts "1.3.0-SNAPSHOT"]
               [fs "1.1.2"]
               [clj-yaml "0.3.1"]
               [doric "0.7.0-SNAPSHOT"]
               [ordered "1.0.0"]
               [compojure "1.0.1"]
               [ring "1.0.2"]
               [enlive "1.0.0"]]
```

Python mutability



```
In [1]: def do_work(in_dict):
            in_dict["item"] = "changed"
            return "some_result"

        input = {"item": "original"}
        do_work(input)
        print input

        {'item': 'changed'}
```

Immutable data structures

```
(defn do-work [in-dict]
  (let [new-dict (assoc in-dict "item" "changed")]
    "some_result"))

(let [input {"item" "original"}]
  (do-work input)
  (println input))

{"item" "original"}
```

Python and Clojure side by side

```
def words(text):
    return re.findall("[a-z]+", text.lower())

(defn words [text]
  (re-seq #"[a-z]+" (.toLowerCase text)))
```

http://en.wikibooks.org/wiki/Clojure_Programming/Examples/Norvig_Spelling_Corrector

http://tin.nu/sudoku.html

MapReduce

- Software framework for distributed computing
- Introduced by Google
- Parallelization of large datasets
- Hadoop – open source

Functional: Map

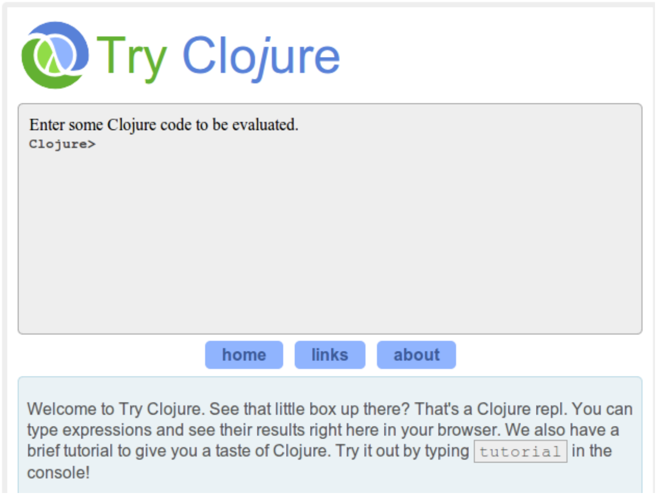
```
width = 1 + max(len(values[s])
                for s in squares)

(inc (apply max
            (map (comp count values)
                 squares)))
```

Functional: Reduce

```
model = collections.defaultdict(lambda: 1)
for f in features:
    model[f] += 1

(reduce (fn [model x]
          (assoc model x
                 (inc (get model x 1))))
        {}
        features))
```



Enter some Clojure code to be evaluated.

```
Clojure>
```

Welcome to Try Clojure. See that little box up there? That's a Clojure repl. You can type expressions and see their results right here in your browser. We also have a brief tutorial to give you a taste of Clojure. Try it out by typing `tutorial` in the console!

http://tryclj.com/

Variant comparison architecture

- Automated pipeline
- Configuration file describing inputs
- Web front end to build configuration and run

Configuration YAML

```
---
outdir: test/data/grading
outdir-prep: test/data/grading/prep
experiments:
  - sample: NA00001
    ref: test/data/GRCh37.fa
    intervals: test/data/phasing-reference-regions.bed
    calls:
      - name: reference
        file: test/data/phasing-reference.vcf
      - name: contestant
        file: test/data/phasing-contestant.vcf
        intervals: test/data/phasing-contestant-regions.bed
```

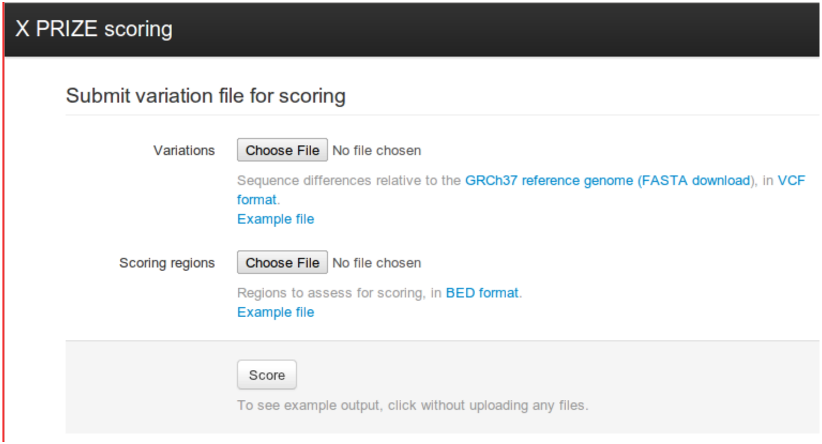
General comparison YAML

```
experiments:
  - sample: Test1
    ref: test/data/hg19.fa
    intervals: test/data/target-regions.bed
    align: test/data/aligned-reads.bam
    calls:
      - name: gatk
        file: test/data/gatk-calls.vcf
        refcalls: false
      - name: freebayes
        file: test/data/freebayes-calls.vcf
        annotate: true
        filters:
          - HRun > 5.0
          - QD < 2.0
          - FS > 60.0
          - MQRankSum < -12.5
```

Web: implementation

- Clojure web framework
  - Compojure http://compojure.org/
  - Enlive https://github.com/cgrand/enlive
- ClojureScript: Clojure to Javascript
  - http://clojurescriptone.com/
- CSS framework: Twitter Bootstrap
  - http://twitter.github.com/bootstrap/

Web: submission


X PRIZE scoring

Submit variation file for scoring

Variations No file chosen

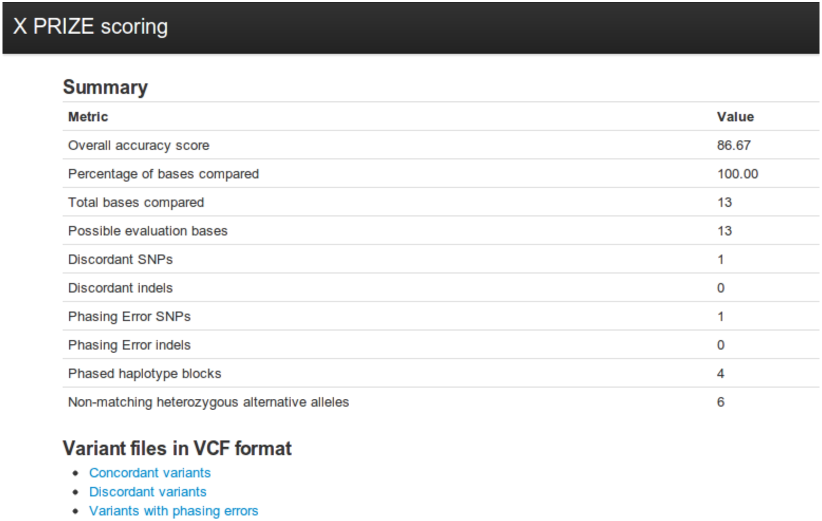
Sequence differences relative to the GRCh37 reference genome (FASTA download), in VCF format.
Example file

Scoring regions No file chosen

Regions to assess for scoring, in BED format.
Example file

To see example output, click without uploading any files.

Web: results



X PRIZE scoring

Summary

Metric	Value
Overall accuracy score	86.67
Percentage of bases compared	100.00
Total bases compared	13
Possible evaluation bases	13
Discordant SNPs	1
Discordant indels	0
Phasing Error SNPs	1
Phasing Error indels	0
Phased haplotype blocks	4
Non-matching heterozygous alternative alleles	6

Variant files in VCF format

- Concordant variants
- Discordant variants
- Variants with phasing errors

Code perusal

Source	https://github.com/chapmanb/bcbio.variation
Docs	http://chapmanb.github.com/bcbio.variation