gotCloud?

http://gotcloud.org
What is **genomes on the Cloud**?

- **gotCloud** is a sequence analysis pipeline
  - *Integrative* Alignment, QC, Variant Calling, Phasing
  - *Seamless* Requires only simple configuration files
  - *Robust* ..against unexpected failures & stops
  - *Scalable* ..to many thousands of genomes

- **gotCloud** also provides...
  - A set of many useful software tools
  - Software library (C++) for sequence analysis
How can I use gotCloud?
GotCloud Alignment Pipeline

- End-to-end analysis
  - Fully automated and parallelized (with quality controls)
  - Requires only a simple fastq list file

<table>
<thead>
<tr>
<th>SAMPLE_ID</th>
<th>FASTQ1</th>
<th>FASTQ2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample1</td>
<td>Sm1_Run1_1.fastq.gz</td>
<td>Sm1_Run1_1.fastq.gz</td>
</tr>
<tr>
<td>Sample1</td>
<td>Sm1_Run2_1.fastq.gz</td>
<td>Sm1_Run2_1.fastq.gz</td>
</tr>
<tr>
<td>Sample2</td>
<td>Sm2.fastq.gz</td>
<td>.</td>
</tr>
<tr>
<td>Sample3</td>
<td>Sm3_1.fastq.gz</td>
<td>Sm3_2.fastq.gz</td>
</tr>
</tbody>
</table>
Running GotCloud Alignment Pipeline

Example command

`.gotcloud align --list fastq.txt --out outputDir --numjobs 3 --threads 2`

- `.gotcloud` run GotCloud
- `align` alignment pipeline
- `--list fastq.txt` list of FASTQs
- `--out outputDir` where to write output
- `--numjobs 3` run 3 samples concurrently
- `--threads 2` 2 CPU threads per sample

CRAM support is in beta testing
What to Expect from GotCloud Alignment Pipeline

- Aligned and post-processed BAM files
- Summary statistics and graphs from qplot

- Contamination checking from verifyBamID
  - With and/or without external genotype data
GotCloud Variant Calling Pipelines

Overview of SNP Calling Pipeline:

- **Processed BAM** (one per sample)
  - **samtools**
- **Filtered VCF**
  - **infoCollector**
- **Unfiltered VCF**
  - **glfMultiples**
- **Genotype Likelihood** (one per sample)
  - **SVM**
- **Phased VCF**
  - **beagle** / **thunder**
- **Association Results**
  - **EPACTS (External)**

### Summary

- **End-to-end analysis**
- **Very efficient**
- **Small memory** (<1G)
- **Scalable to >1,000s**
- **High parallelization**
- **Fault-tolerant**
- **Requires only list of BAMs**
GotCloud Variant Calling Pipelines

Example commands

./gotcloud snpcall --list bams.txt --out outputDir --numjobs 10
./gotcloud indel --list bams.txt --out outputDir --numjobs 10
./gotcloud genomestrip --list bams.txt --out outputDir --numjobs 10
./gotcloud lddefine --list bams.txt --out outputDir --numjobs 10
./gotcloud mei --list bams.txt --out outputDir --numjobs 10*

– snpcall/indel/genomestrip/mei variant caller to run
– ldrefine run beagle/thunderVCF genotype refinement
– --list bams.txt list of bams per sample
– --out outputDir where to write output
– --numjobs 10 run 10 jobs concurrently

* Coming soon..
# GotCloud for Large-scale Sequencing

<table>
<thead>
<tr>
<th>Study</th>
<th>Genome</th>
<th>Exome</th>
<th>N</th>
<th>Populations</th>
<th># SNPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000 Genomes</td>
<td>~6x</td>
<td>~40x</td>
<td>2,535</td>
<td>Many</td>
<td>69.1M</td>
</tr>
<tr>
<td>Type 2 Diabetes</td>
<td>~5x</td>
<td>~80x</td>
<td>2,850</td>
<td>Europeans</td>
<td>26.7M</td>
</tr>
<tr>
<td>Exome Sequencing Project</td>
<td>.</td>
<td>~80x</td>
<td>6,916</td>
<td>EUR+AFR</td>
<td>1.92M</td>
</tr>
<tr>
<td>Sardinian Sequencing</td>
<td>~4x</td>
<td>.</td>
<td>3,520</td>
<td>Sardinians</td>
<td>23.1M</td>
</tr>
<tr>
<td>Bipolar Sequencing</td>
<td>~12x</td>
<td>.</td>
<td>2,825</td>
<td>Europeans</td>
<td>43.7M</td>
</tr>
<tr>
<td>Nephrotic Syndrome</td>
<td>~4x</td>
<td>.</td>
<td>464</td>
<td>Many</td>
<td>25.6M</td>
</tr>
<tr>
<td>Age-related Macular Degeneration</td>
<td>~6x</td>
<td>.</td>
<td>3,000</td>
<td>Europeans</td>
<td>36.2M</td>
</tr>
<tr>
<td>HUNT</td>
<td>~4x</td>
<td>.</td>
<td>1,200</td>
<td>Norwegians</td>
<td>23.0M</td>
</tr>
</tbody>
</table>
Can I detect clinically important variants using gotCloud?
Variant Call Examples using GotCloud

- Examples from *APOL1* gene
  - Nephrotic syndrome associated genes
  - Available at [http://genome.sph.umich.edu/wiki/GotCloud:_Amazon_Demo](http://genome.sph.umich.edu/wiki/GotCloud:_Amazon_Demo)

- SNPs: Nephrotic syndrome risk allele – *APOL* G1 allele

- Indels: Nephrotic syndrome risk allele – *APOL* G2 allele

- Structural Variants nearby *APOL1* loci
Variant Call Examples using GotCloud

• Examples from *APOL1* gene
  – Nephrotic syndrome associated genes
  – Available at [http://genome.sph.umich.edu/wiki/GotCloud:_Amazon_Demo](http://genome.sph.umich.edu/wiki/GotCloud:_Amazon_Demo)

• **SNPs**: Nephrotic syndrome risk allele – *APOL* G1 allele
  22 36661906 . A G 18 PASS AN=124;AC=2;AF=0.013827...

• **Indels**: Nephrotic syndrome risk allele – *APOL* G2 allele
  22 36662041 . AATAATT A 756 PASS AN=114;AC=2;AF=0.017544...

• **Structural Variants** nearby *APOL1* loci
  22 36133488 . C <DEL> . PASS AN=124;AC=2;END=36144254...
How can I use gotCloud on the Cloud?
GotCloud on the Cloud

• GotCloud supports for many high-computing cluster systems
  – MOSIX
  – SLURM
  – Sun Grid Engine (SGE)
  – Portable Batch System (PBS)

• GotCloud also supports the Amazon Cloud
  – In 1000G low-pass genomes examples...
  – Average cost is $20 per genome
  – GotCloud Amazon Machine Images (AMI) available
• Getting started...
  – GotCloud AMI is publicly available
  – The AMI contains example input files
    • You can run today’s demo on your own

• Adding computing power for your own analysis...
  – GotCloud AMI supports StarCluster (SGE-compatible)
  – You can add as many nodes as you need.
  – You will need to upload your own files to Amazon
    • Have your data mounted or use wget/curl/lftp
Thanks!

http://gotcloud.org/

or https://github.com/statgen/gotcloud

https://drive.google.com/file/d/0B9LqMcsR6cysalhwazB5WkVPOTg/view?usp=sharing

• Gonçalo Abecasis
• Mary Kate Wing
• Hyun Min Kang
• Goo Jun

• Adrian Tan
• Terry Gliedt
• Tom Blackwell
• Alan Kwong
GotCloud on Amazon Cloud: Launch Amazon Instance

Assuming that you set up your account, security key, and also that you selected GotCloud AMI and machine size (instance type)...

Step 7: Review Instance Launch

Selected Instance Type

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>ECUs</th>
<th>vCPUs</th>
<th>Memory (GiB)</th>
<th>Instance Storage (GB)</th>
<th>EBS-Optimized Available</th>
<th>Network Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>c3.xlarge</td>
<td>14</td>
<td>4</td>
<td>7.5</td>
<td>2 x 40</td>
<td>Yes</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

Security Groups

Instance Details

Storage

Tags

Click Launch When Ready
GotCloud on Amazon Cloud: Connecting to Your Instance

- Click "Connect" to Launch the terminal when Ready
- After Launching, Check if the State is "Running"
GotCloud on Amazon Cloud: Connecting to Your Instance

I would like to connect with:

- A standalone SSH client
- A Java SSH Client directly from my browser (Java required)

Enter the required information in the fields below to connect to your instance. AWS automatically detects the key pair name, and Public DNS for your instance. You need to enter the location and name of the .pem file containing your private key.

- **Public DNS**: ec2-54-160-249-250.compute-1.amazonaws.com
- **User name**: root
- **Key name**: CSG.pem
- **Private key path**: eg. C:\KeyPairs\CSG.pem
- **Save key location**: [on/off]

[Launch SSH Client] [Close]

**Username (e.g. ubuntu)**

**Path to your key (previously selected)**
GotCloud on Amazon Cloud: Running GotCloud

ubuntu@ip-10-47-131-201:~$ head -n 10 example/bam.list
HG00551 example/bams/HG00551.recal.bam
HG00553 example/bams/HG00553.recal.bam
HG00554 example/bams/HG00554.recal.bam
HG00637 example/bams/HG00637.recal.bam
HG00638 example/bams/HG00638.recal.bam
HG00640 example/bams/HG00640.recal.bam
HG00641 example/bams/HG00641.recal.bam
HG00734 example/bams/HG00734.recal.bam
HG00736 example/bams/HG00736.recal.bam
HG00737 example/bams/HG00737.recal.bam

Check if the demo input files are available
GotCloud on Amazon Cloud: Running GotCloud

```bash
ubuntu@ip-10-47-131-201:~$ gotcloud snpcall --conf example/test.conf --out output --numjobs 8
```

Key configurations:
- **GOTCLOUD_ROOT**: /home/ubuntu/gotcloud
- **OUT_DIR**: output
- **BAM_LIST**: example/bam.list
- **REF**: /home/ubuntu/gotcloud/gotcloud.ref/human.glk.v37.fa
- **CHRS**: 22
- **BATCH_TYPE**: local
- **BATCH_OPTS**: 

Processing the following steps...
1: RUN_PILEUP
2: RUN_PILEUP
3: RUN_GLFMULTIPLES
4: RUN_VCFPILEUP
5: RUN_FILTER
6: RUN_SVM
7: RUN_SPLIT

Generating commands for chr22...
GotCloud on Amazon Cloud: Checking Output Files

Starting sorting values
Writing output values
Analysis finished on Thu Oct 9 18:06:20 2014

loading /home/ubuntu/example/ref22/dbsnp_135.b37.chr22.vcf.gz as a VCF input.
finished loading dbSNP for 676395 variants
TOTAL 4074 4074 2029 1045 723 120 85 47 22 3
Commands finished in 445 secs with no errors reported

```
ubuntu@ip-10-63-151-167:~$ ls output/
glfs pvcfs umake.snpcall.conf umake.snpcall.Makefile.cluster vcfs
jobfiles split umake.snpcall.Makefile umake.snpcall.Makefile.log
ubuntu@ip-10-63-151-167:~$ ls output/vcfs
chr22
ubuntu@ip-10-63-151-167:~$ ls output/vcfs/chr22/
10000000.15000000 50000001.51304566 1,50000000 50000001.10000000
15000001.20000000 chr22.filtered.sites.vcf
20000001.25000000 chr22.filtered.sites.vcf.norm.log
25000001.30000000 chr22.filtered.sites.vcf.summary
30000001.35000000 chr22.filtered.vcf.gz
35000001.40000000 chr22.filtered.vcf.gz.OK
40000001.45000000 chr22.filtered.vcf.gz.tbi
45000001.50000000 chr22.hardfiltered.sites.vcf
```

Examine the output directory
Examine the variant of interest
GotCloud on Amazon Cloud: Don’t Forget to Terminate