What is benchmarking?
In the area of method development, it is the method for assessing the performance of software tools by comparing it to gold standards or the best practices in the industry. One can use benchmarking to assess the performance of software/bioinformatics tools such as variant callers.

Benchmarking will be a continuous process in order to maintain consistent results from the growing versions of software tools meeting the evolving gold standards. Benchmarking can be achieved by manually comparing the results with gold standards based on specific metrics or using benchmarking tools.

Why do we need benchmarking?
Identify true positives, false positives and false negatives. Decide which tool(s) performs better for calling the variants in the data obtained from clinical cases. Thus, benchmarking is important because it will help identify variants of clinical significance impacting research in personalized medicine.

Data sources: WGS - Illumina’s CLIA NA12878, WES - Personalis, Nextera

MedGAP2.4 Pipeline dependencies: samtools 1.19, picard-tools 1.32, GATK 3.4, tabix 0.2.5, bamtools 2.2.3, sjm 0.1, zlib 1.2.7, Python 2.7, Java 8 and Perl libraries (File-Path-2.08, Statistics-Descriptive-2.6, GD-2.45, GDGraph-1.44, GDGraph-histogram-1.1, GDTextUtil-0.86, Math-CDF-0.1). MedGAP2.4.1 has same dependencies but picard-tools 2.4.1 (More details on MedGAP can be obtained from CGS development team)

Speedseq dependencies: g++ and standard C and C++ development libraries, CMake, GNU awk and core utils, Python 2.7 (numpy, pySam 0.8.0+, scipy), ROOT, Variant Effect Predictor, BWA, FreeBayes, LUMPY, Sambamba, SAMBLASTER, Vawk, GNU Parallel, mbuffer

What are the gold standards available to us?
NA12878 Platinum: NISTv2.18, NISTv2.19, NISTv3.2.2 (Jan 2016)
Fasta sequences from the above NIST versions: hg18, hg19, hg38
Bed files from the above NIST versions (high confidence regions)
Variants (SNPs and INDELS) called in the high confidence regions from the above NIST versions

<table>
<thead>
<tr>
<th>GIAB (NA12878)</th>
<th>No of truth SNPs+indels in GIAB high confidence bed region</th>
<th>No of truth SNPs in GIAB high confidence bed region</th>
<th>No of truth indels in GIAB high confidence bed region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NISTv2.19</td>
<td>3,152,426</td>
<td>2,787,291</td>
<td>365,135</td>
</tr>
<tr>
<td>NISTv3.2.2</td>
<td>4,299,935</td>
<td>3,572,851</td>
<td>727,084</td>
</tr>
</tbody>
</table>

Table 1. Number of SNPs and INDELS in NIST NA12878 truth sets obtained using RTG vcfstats

References
4. Illumina’s hap.py: https://github.com/Illumina/hap.py/blob/master/doc/happy.md
6. LOOM workflow engine: https://pypi.python.org/pypi/loom-engine/0.1.0
8. Vcflib: https://github.com/vcflib/vcflib
11. DOCKER: https://www.docker.com

Acknowledgements
Funding support from SCGPM and CGS
Nathan Hammond and Isaac Liao: Developers of LOOM Computational Resources: SCG4 (SCGPM), SUGI (CGS)