Webinar

Introduction to Benchmarking of NGS Workflows

20.06.2023
16:00 (CEST)

Kübra Narcı
DKFZ

Outline

- GHGA goals and objectives
- Benchmarking
  - Scope
  - Components
- <QA Break>
- Strategies to find benchmarks
  - Continuous benchmarking
- Benchmark plan of GHGA
  - An example using sarek and NCBench

GHGA

GHGA Goals & Objectives

- Establish a national infrastructure for human omics data
- An ethico-legal and data use framework for
  - Data Sharing
  - Protection
  - Analysis
- Provide standards for:
  - Metadata and Workflows
- Make human omics data

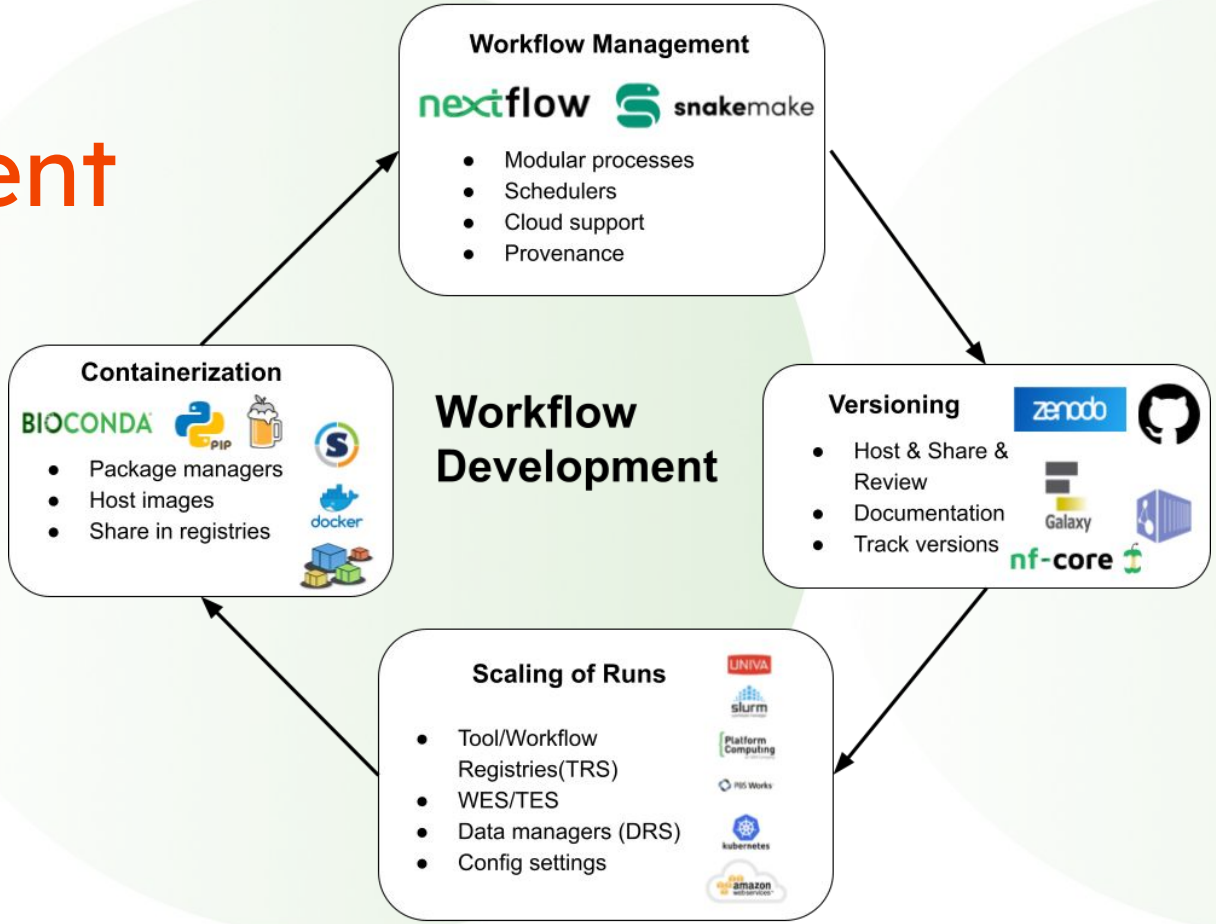
  FAIR and being GA4GH compliant

Global Alliance for Genomics & Health

Collaborate. Innovate. Accelerate.

GHGA Workflow Management

- Accurate
- Scalable
- Reproducible
- Portable



https://github.com/ghga-de

Accuracy

- Use state-of-the-art bioinformatic tools
- Implement best-practice pipelines
- Benchmark!

GHGA

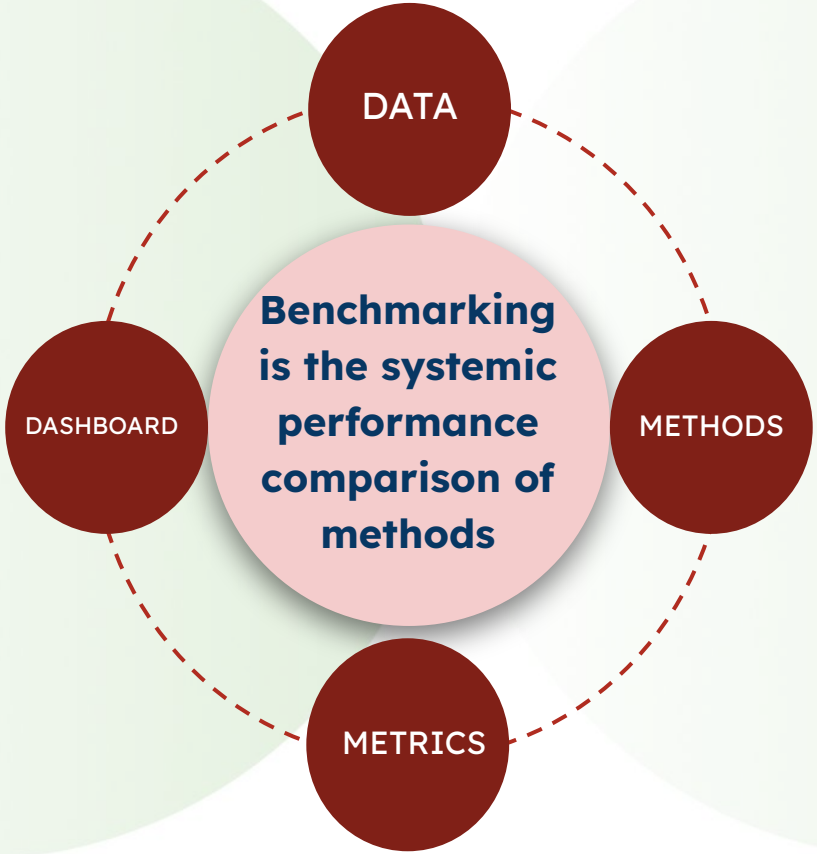
Learning goals

- Importance of **accurate**, **efficient** and **transparent** benchmarking
- Commonly used benchmark datasets, methods and metrics in **variant calling analysis**
- The aspects to consider for **standardized** benchmark analysis
- Importance of **community** developed benchmarking platforms

GHGA

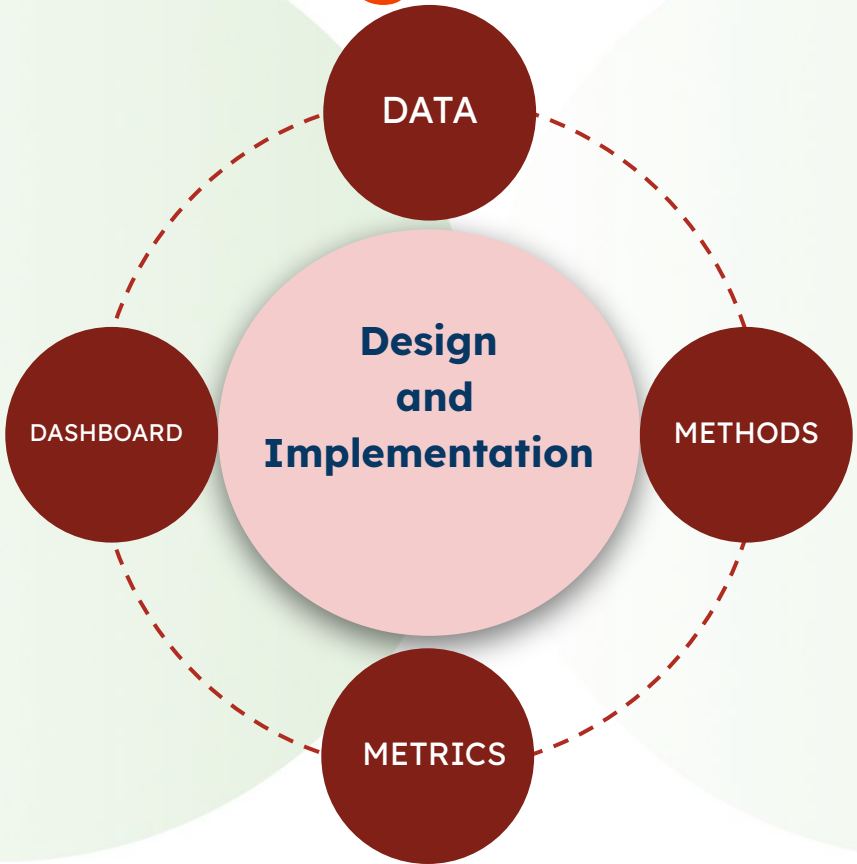
Why benchmarking is needed?

- What is the most accurate tool for my analysis?
- Which parameters to use for best optimization?



Scope of benchmarking

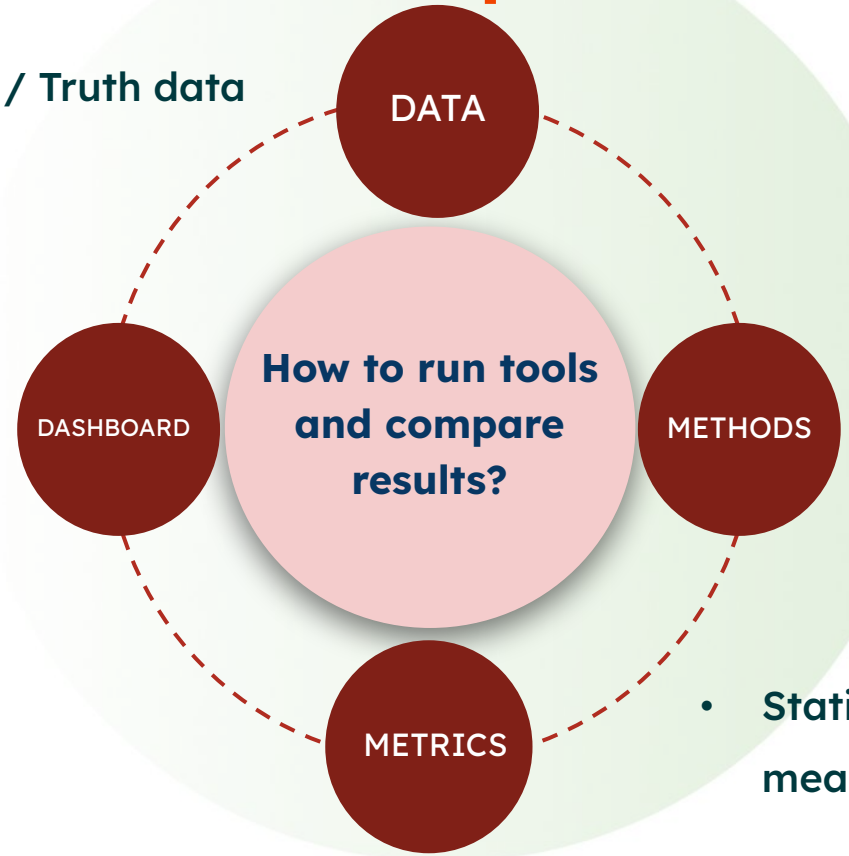
1. Tool developers
    - Highlighting merits and weaknesses
2. Neutral studies
    - Independent groups,
    - Weaknesses in the methods
3. Community efforts
    - Challenges like: DREAM, CASP, CAGI, SEQC2, and PrecisionFDA truth
    - Dashboards



GHGA

Benchmark components

- Test / Truth data



- Sharing results
- Charts

- Evaluation method

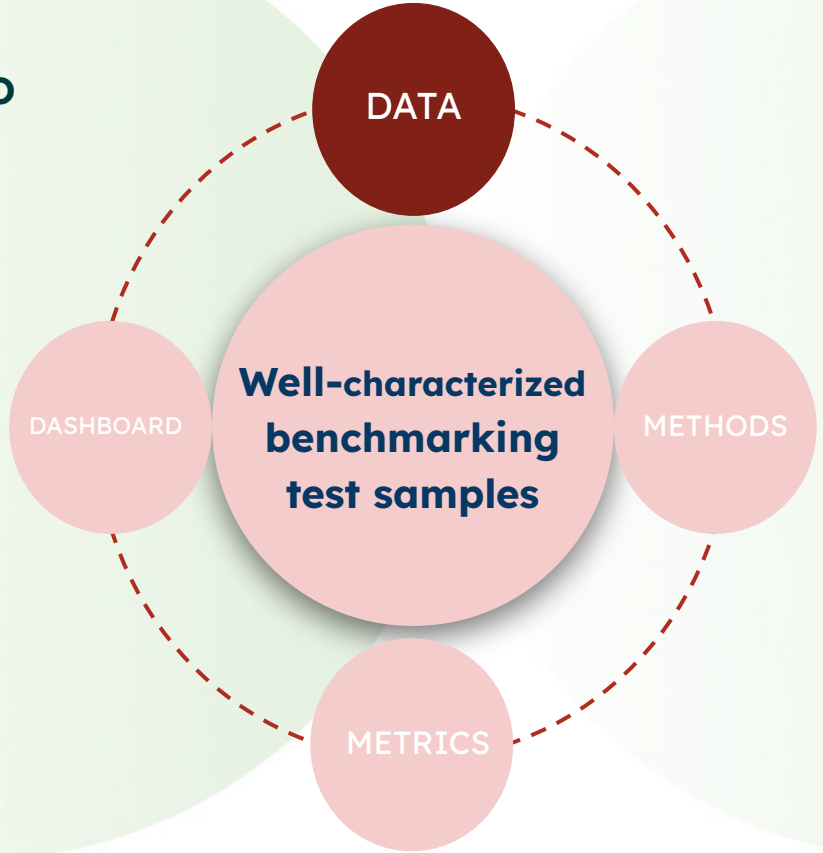
- Statistical measures

GHGA

Test data for variant benchmarking

Set of variants and regions defined to identify performance

- Real samples or synthetic data
- ‘truth’, ‘baseline’ and ‘gold standard’
- ‘high confident regions’, ‘difficult regions’, ‘stratification regions’



GHGA

Benchmarking reference datasets

1. Common calls from multiple sequencers and methods
    a. Genome in a Bottle consortium (GIAB from NIST)

\+ Based on real samples

\- Biased towards easy to call regions

\- Variant callers used to construct the test also used for benchmarking

Genome	Cell line	NIST ID
CEPH	GM12878	HG001
AJ Son	GM24385	HG002
AJ Father	GM24149	HG003
AJ Mother	GM24143	HG004
Chinese Son	GM24631	HG005
Chinese Father	GM24694	HG006
Chinese Mother	GM24695	HG007

GHGA

https://www.nist.gov/programs-projects/genome-bottle

https://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release

Benchmarking reference datasets

2. **Synthetically created benchmark datasets**
    a. **Simulated datasets**
    b. **Syndip/CHM**

\+ **Advantageous to use for difficult regions**

\- **Cannot be simulate real genomic events!**

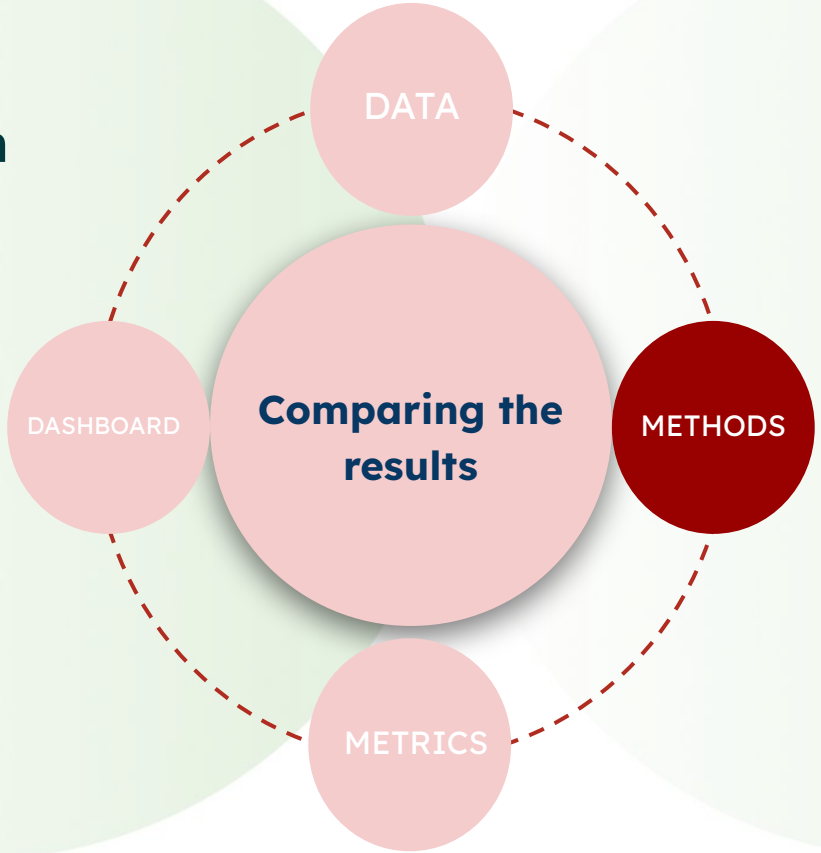
https://github.com/lh3/CHM-eval

GHGA

Analysis of the data

Neutrality should be maintained in comparing tools

- Parameter setting
- Software versions
- Provenance



Benchmarking Methods

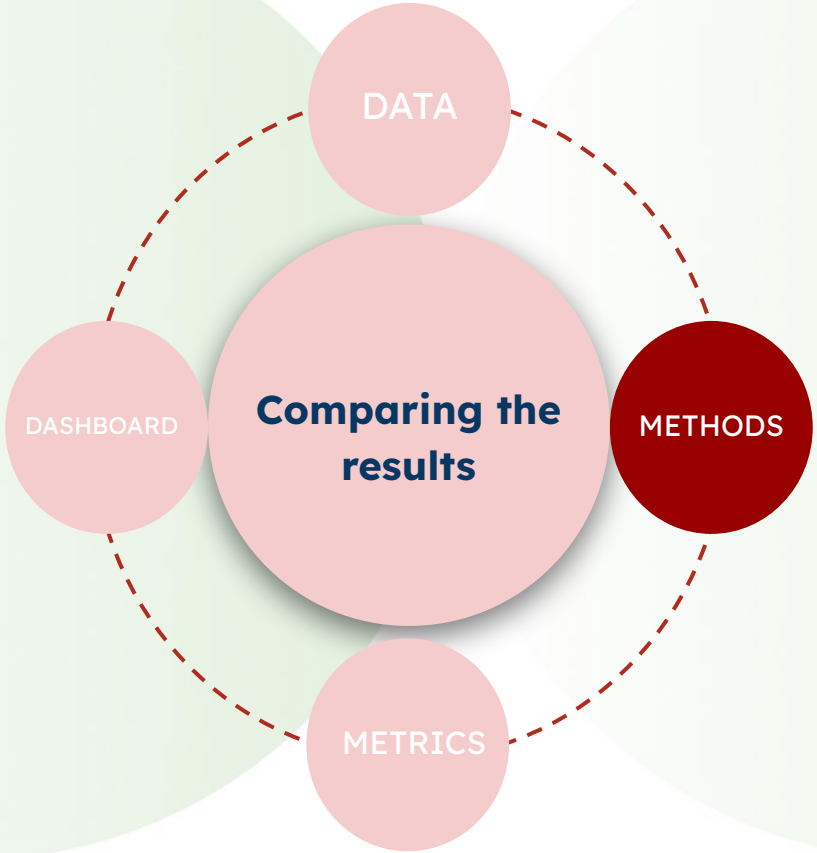
Truth reference set

CHROM	POS	REF	ALT
chr1	56375785	T	A
chr1	61967480	G	A
chr1	61967489	G	A

Benchmark/Query set

CHROM	POS	REF	ALT
chr1	56375785	T	A
chr1	61967489	G	A
chr1	61967500	T	A

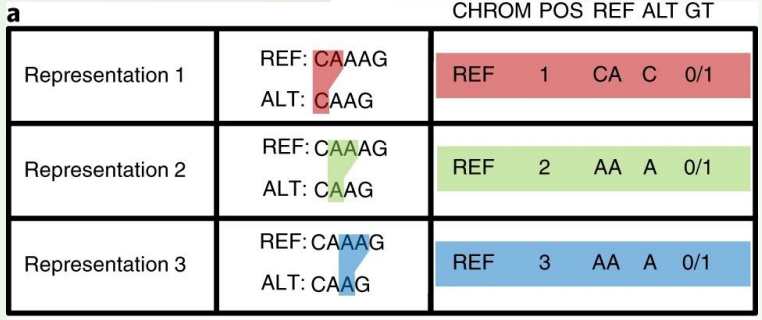
match

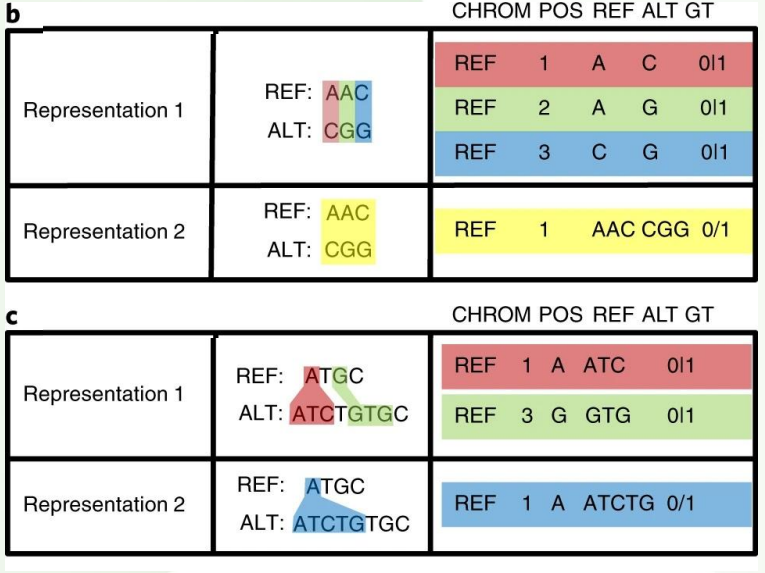


GHGA

Standardization of benchmarking methods

Ambiguity in allelic representations

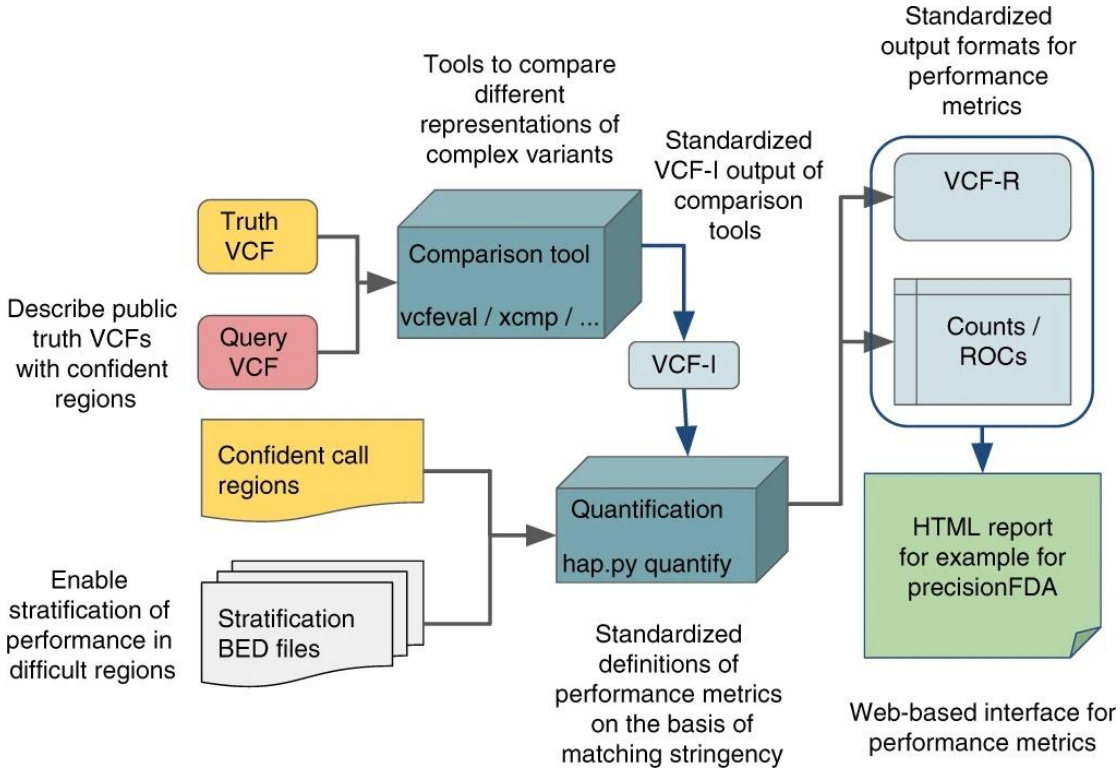




Krusche, P., Trigg, L., Boutros, P.C. et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol 37, 555–560 (2019). https://doi.org/10.1038/s41587-019-0054-x

Standardization of benchmarking methods

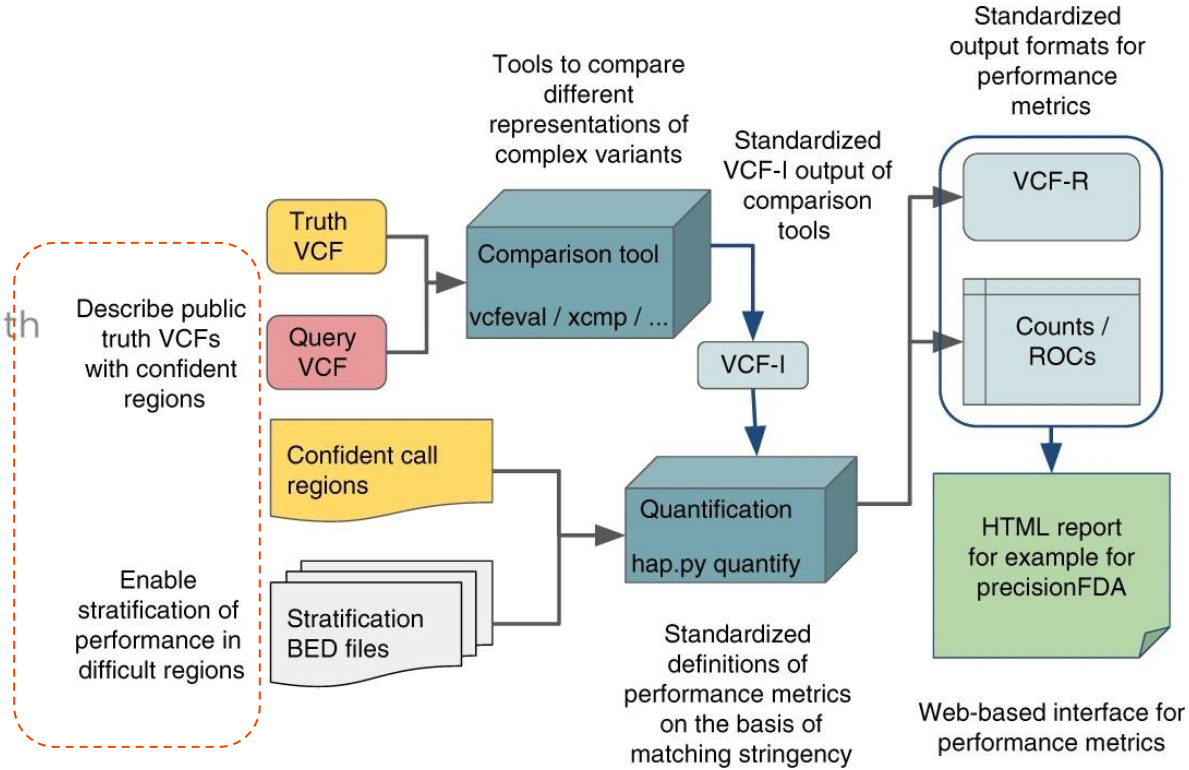
Global Alliance for Genomics & Health
benchmark team



Krusche, P., Trigg, L., Boutros, P.C. et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol 37, 555–560 (2019). https://doi.org/10.1038/s41587-019-0054-x

Standardization of benchmarking methods

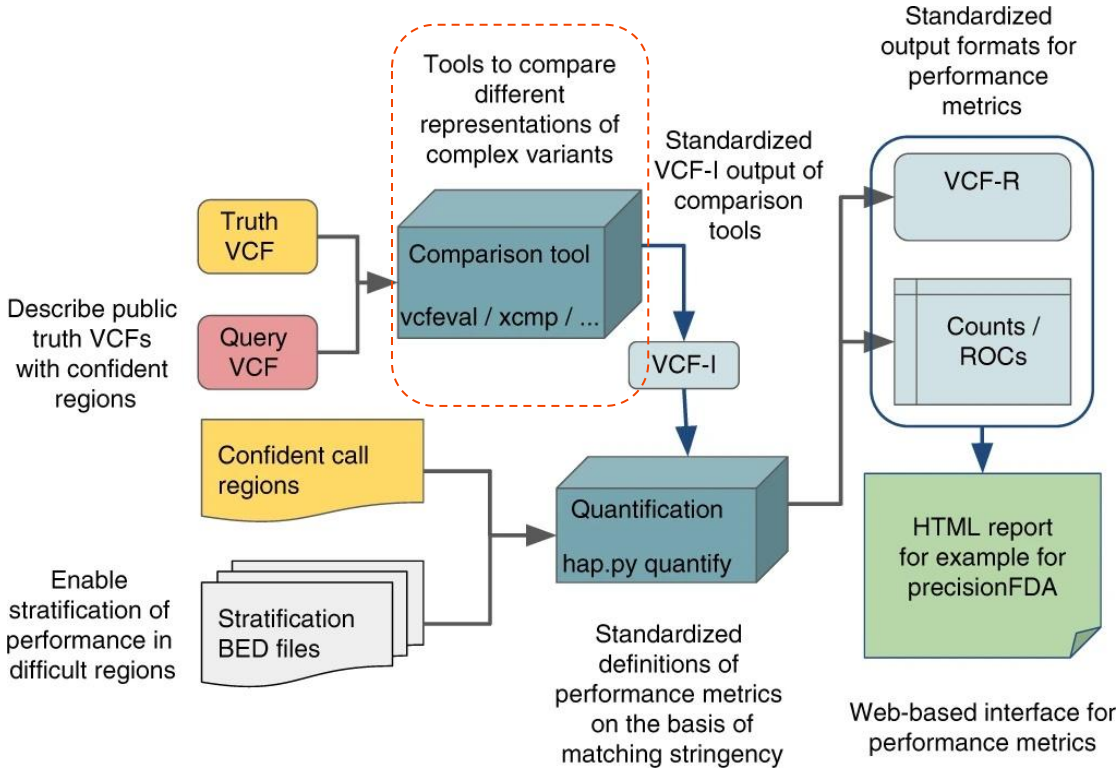
Global Alliance for Genomics & Health
benchmark team



Krusche, P., Trigg, L., Boutros, P.C. et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol 37, 555–560 (2019). https://doi.org/10.1038/s41587-019-0054-x

Standardization of benchmarking methods

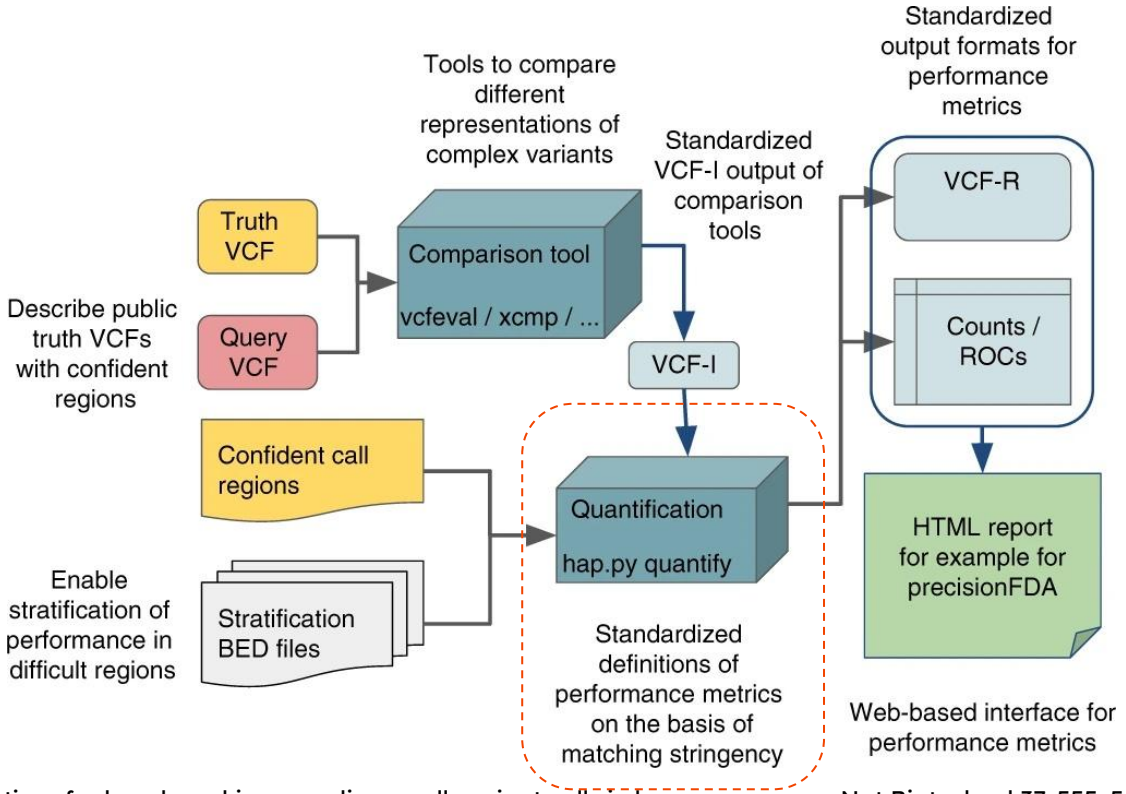
Global Alliance for Genomics & Health
benchmark team



Krusche, P., Trigg, L., Boutros, P.C. et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol 37, 555–560 (2019). https://doi.org/10.1038/s41587-019-0054-x

Standardization of benchmarking methods

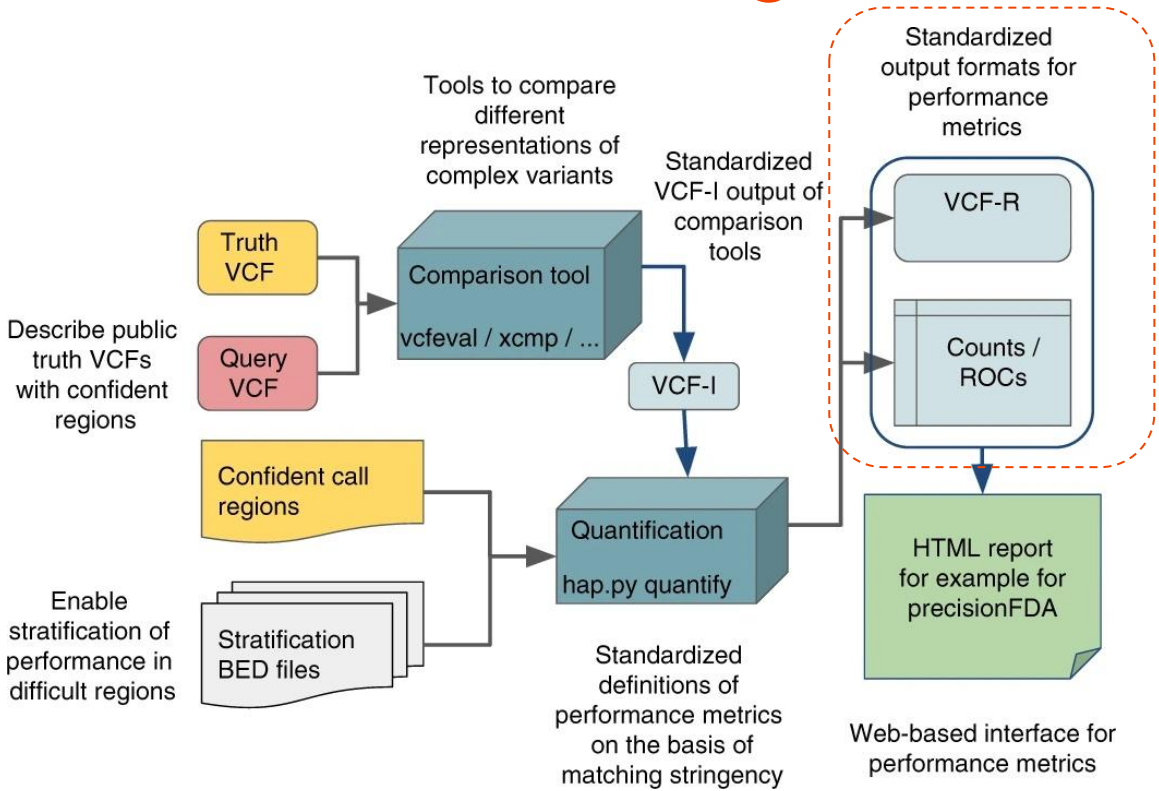
Global Alliance for Genomics & Health
benchmark team



Krusche, P., Trigg, L., Boutros, P.C. et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol 37, 555–560 (2019). https://doi.org/10.1038/s41587-019-0054-x

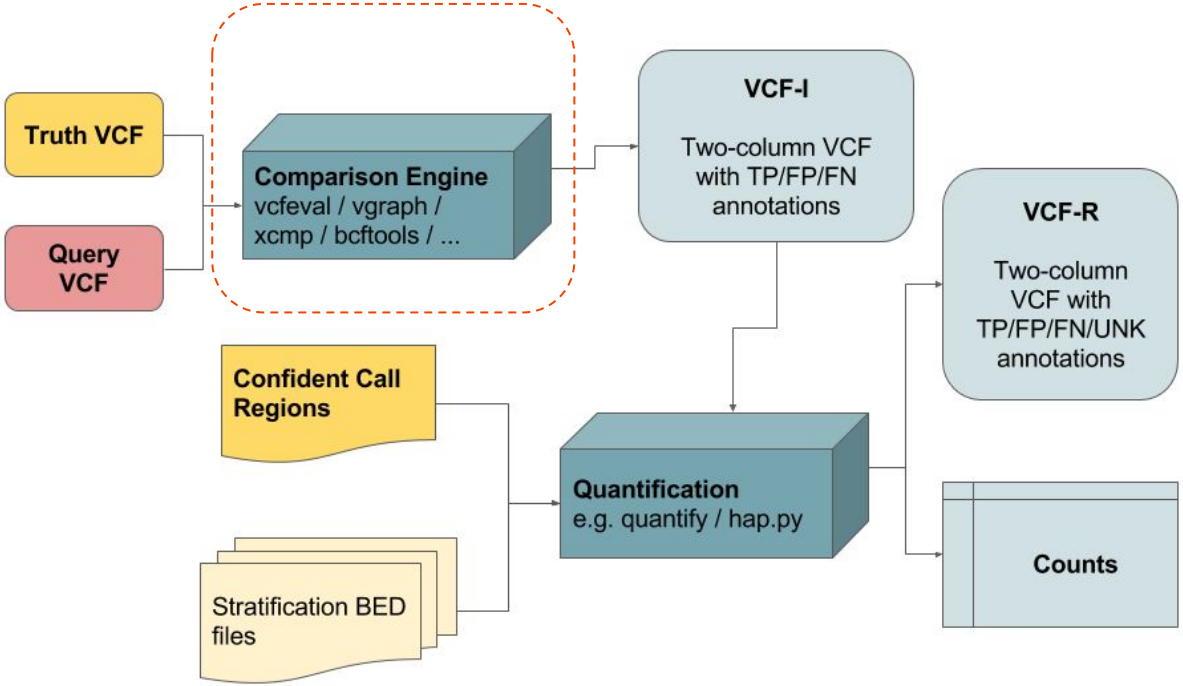
Standardization of benchmarking methods

Global Alliance for Genomics & Health
benchmark team



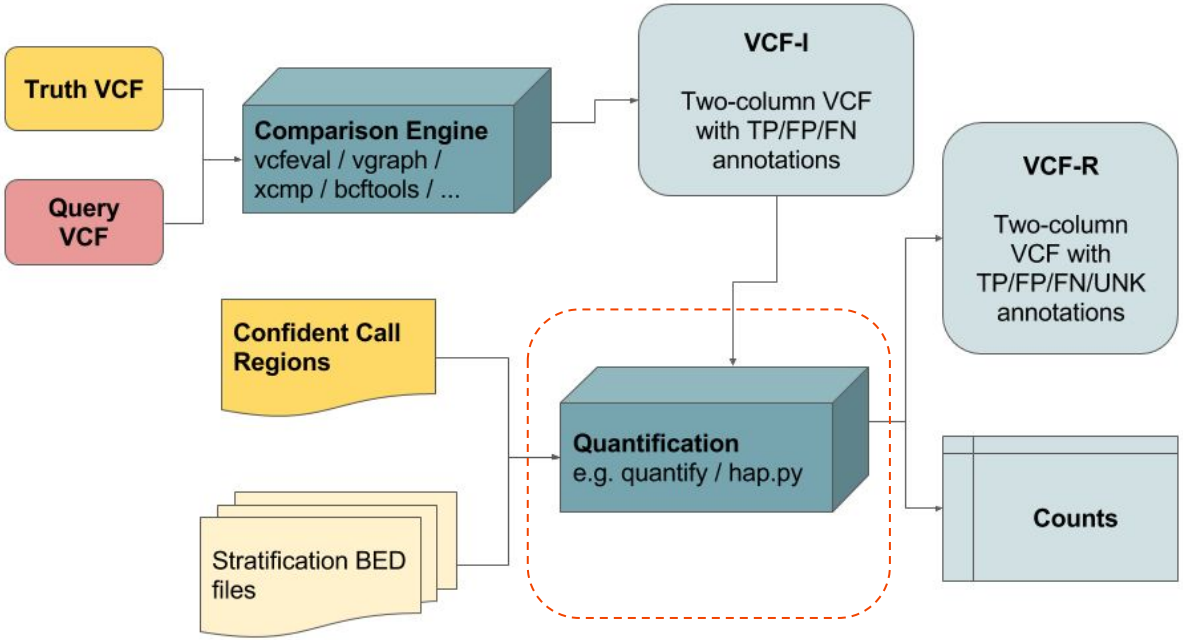
Krusche, P., Trigg, L., Boutros, P.C. et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol 37, 555–560 (2019). https://doi.org/10.1038/s41587-019-0054-x

Standardization of benchmarking methods



https://github.com/ga4gh/benchmarking-tools/tree/master/reporting/basic

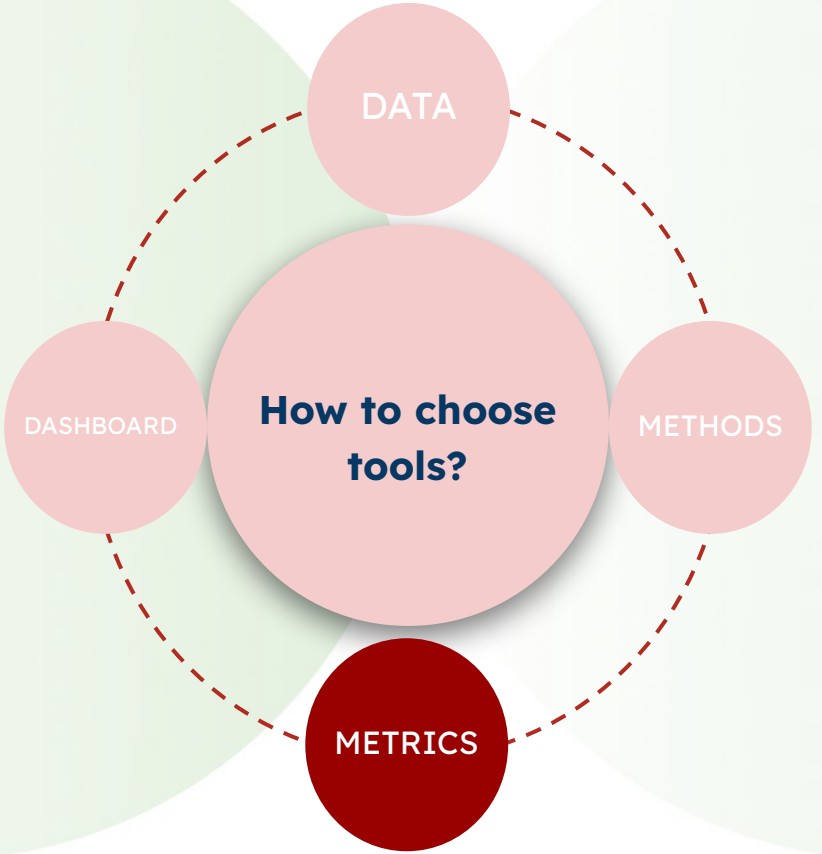
Standardization of benchmarking methods



https://github.com/ga4gh/benchmarking-tools/tree/master/reporting/basic

Measuring performance

1. Qualitative
   + technical perspectives
2. Quantitative
   + performance metrics



Qualitative performance metrics

- User friendliness
- Documentation quality
- Freely available/ open source
- Code quality
- Use of unit testing (CI/CD)
- Adherence to common file types

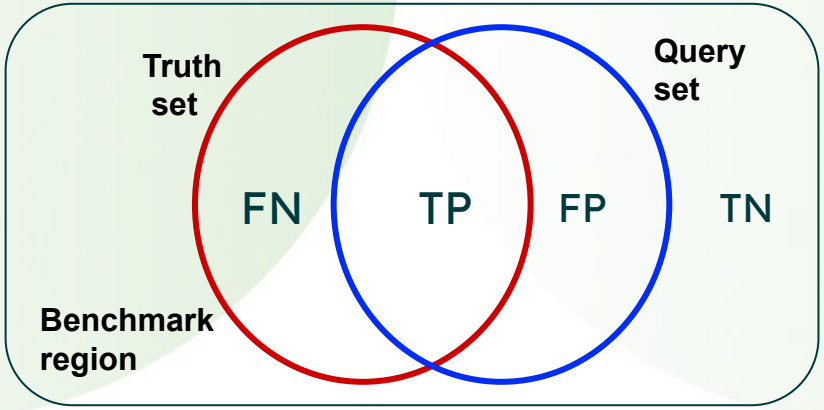
GHGA

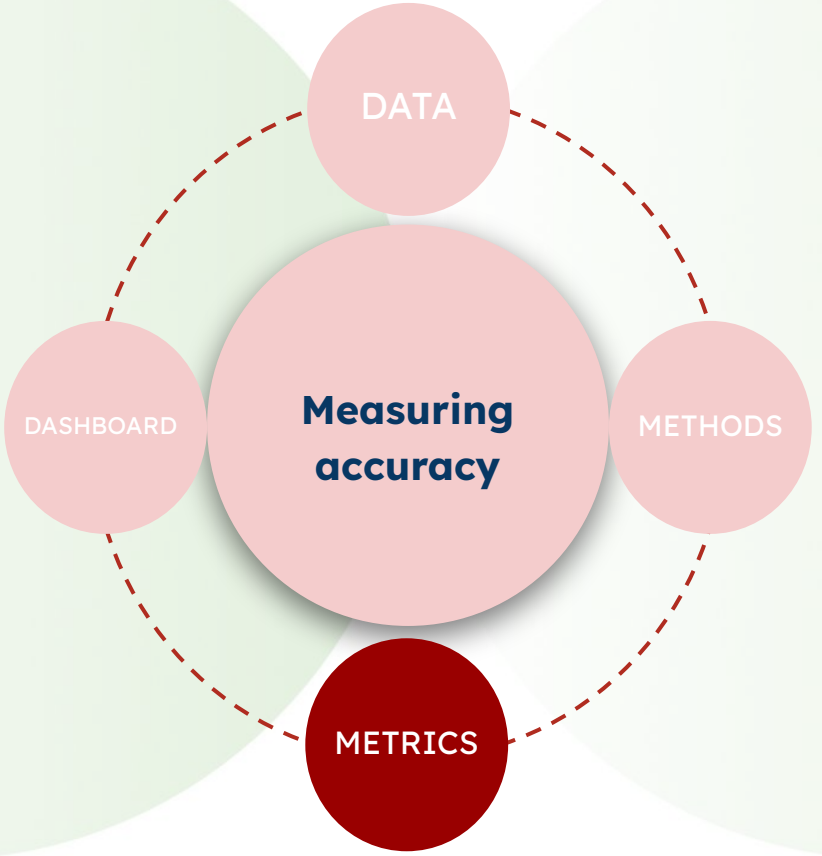
QA

GHGA

Performance metrics

REF	TRUTH	QUERY	COUNT
A/A	C/C	C/C	TP
A/A	A/A	C/C	FP
A/A	C/C	A/A	FN
A/A	A/A	A/A	TN



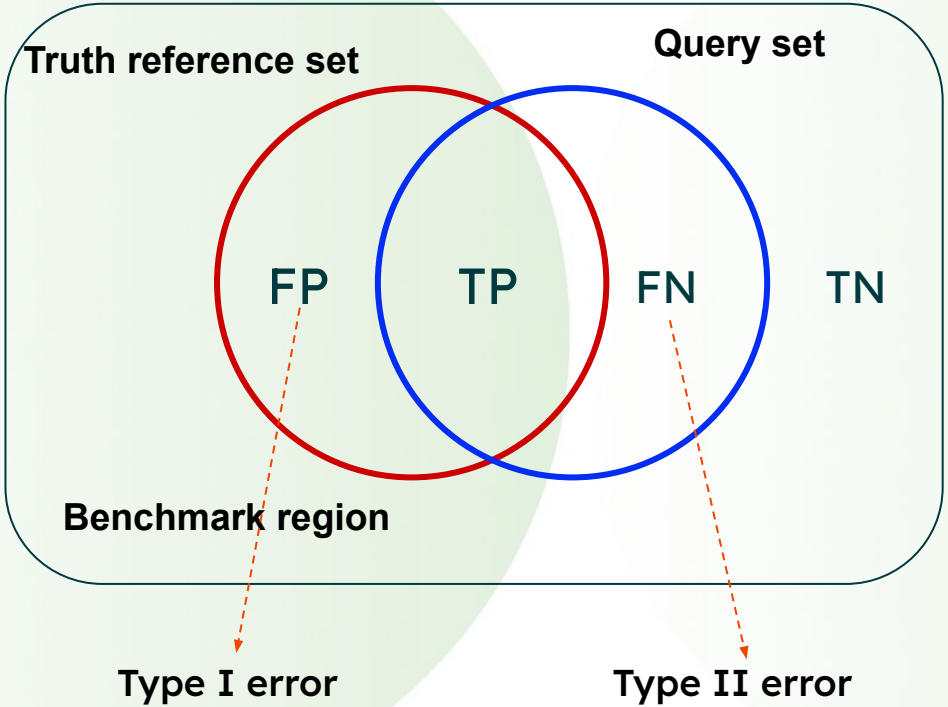


Performance metrics

Confusion matrix

predicted

actual		Query set	
		-	+
Truth set	-	TN	FP
	+	FN	TP



Performance metrics

predicted

actual

		Query set -	Query set +
Truth reference set	-	TN	FP
	+	FN	TP

Specificity
True Negative Rate

$$\frac{TN}{TN + FP}$$

good predictor when you need to find all match cases with the cost of identifying some false positives

Sensitivity (Recall)
True Positive Rate

$$\frac{TP}{TP + FN}$$

Precision

$$\frac{TP}{TP + FP}$$

F1-score

$$\frac{2\ x\ precision\ x\ recall}{precision + recall}$$

GHGA

Performance metrics

predicted

actual

		Query set	
		-	+
Truth reference set	-	TN	FP
	+	FN	TP

good predictor when you need to find only true cases with the cost of missing some true positives

Specificity
True Negative Rate

$$\frac{TN}{TN + FP}$$

Sensitivity (Recall)
True Positive Rate

$$\frac{TP}{TP + FN}$$

Precision

$$\frac{TP}{TP + FP}$$

F1-score

$$\frac{2\ x\ precision\ x\ recall}{precision + recall}$$

GHGA

Performance metrics

a reliably precise and sensitive method will have high F-score!

predicted

actual

		Query set	
		-	+
Truth reference set	-	TN	FP
	+	FN	TP

Specificity
True Negative Rate

$$\frac{TN}{TN + FP}$$

Sensitivity (Recall)
True Positive Rate

$$\frac{TP}{TP + FN}$$

Precision

$$\frac{TP}{TP + FP}$$

F1-score

$$\frac{2\ x\ precision\ x\ recall}{precision + recall}$$

GHGA

Performance metrics

a measure when correct negative cases are too significant to ignore

predicted

actual		Query set -	Query set +
Truth reference set	-	TN	FP
	+	FN	TP

Specificity
True Negative Rate

$$\frac{TN}{TN + FP}$$

Sensitivity (Recall)
True Positive Rate

$$\frac{TP}{TP + FN}$$

Precision

$$\frac{TP}{TP + FP}$$

F1-score

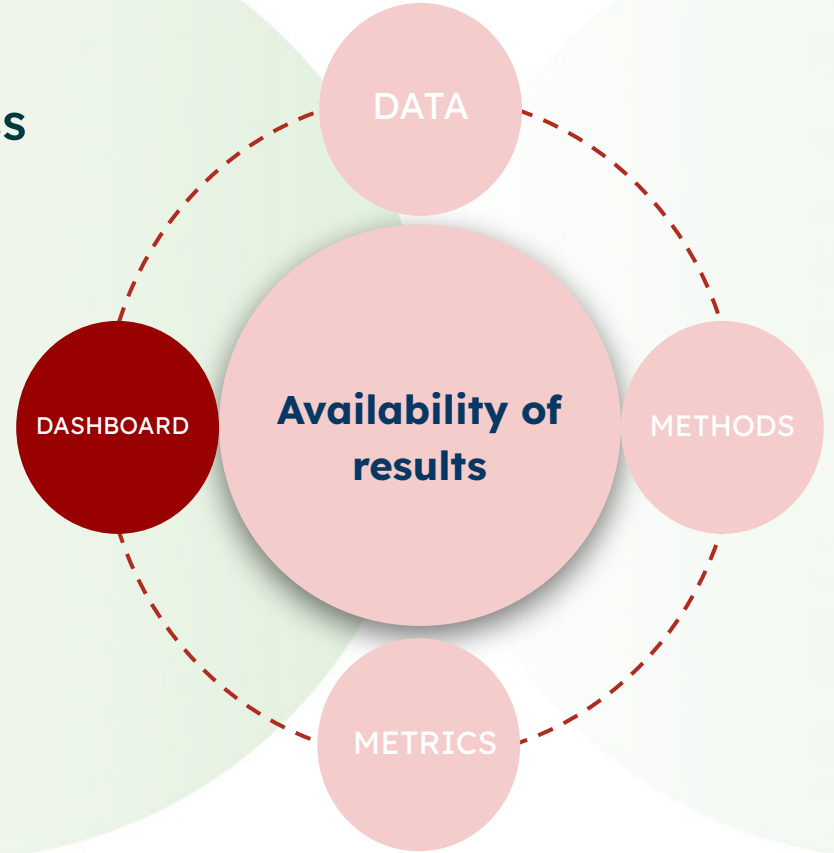
$$\frac{2\ x\ precision\ x\ recall}{precision + recall}$$

GHGA

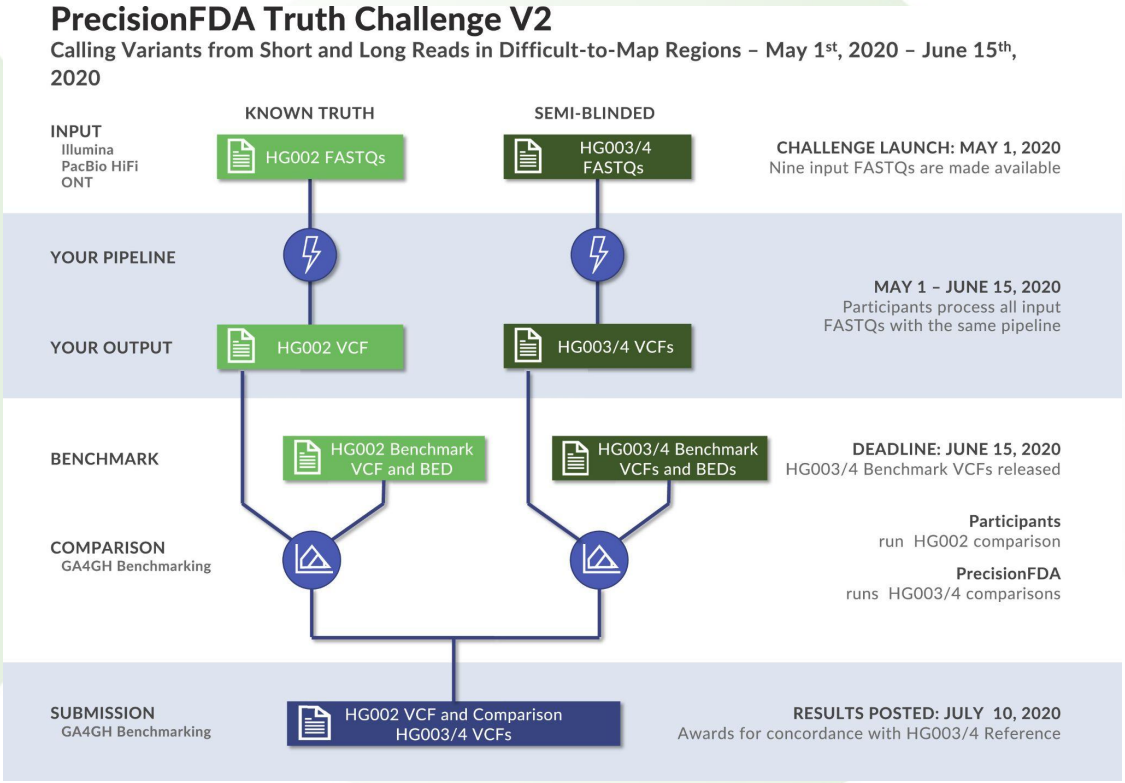
Dashboard

Findability is a condition for FAIRness

- Community driven challenges
- Interactive web-sites
- Continuous benchmarking



Benchmark Challenges

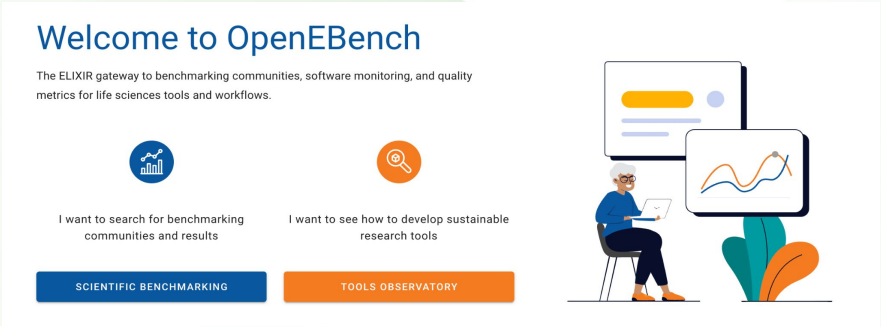


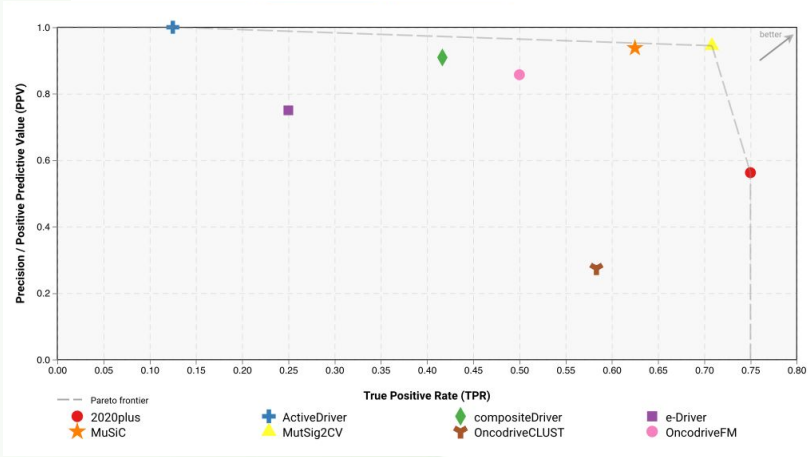
https://precision.fda.gov/challenges/10

GHGA

Interactive Web-sites

- Explore benchmarks
- Perform benchmarks
  - Data
  - Tools
- Monitor and evaluation of software quality
  - FAIRness





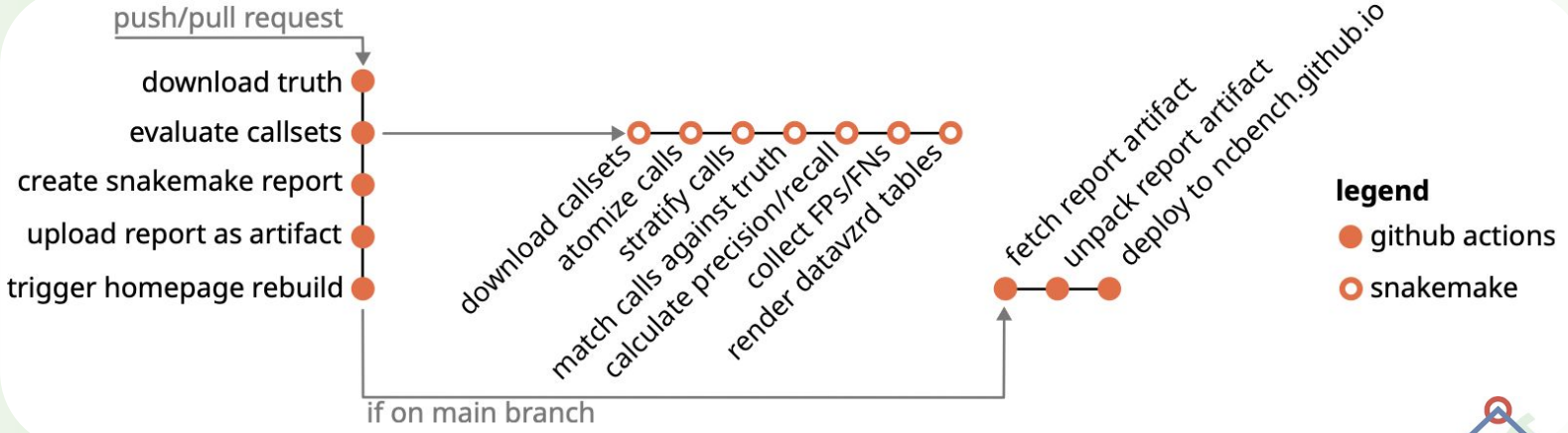
two metrics from the challenge LUSC are represented in the X and Y axis, showing the results from the participating tools in this challenge.

https://openebench.bsc.es/

Continuous Benchmarking

Benchmarking workflow: **NCBench**

zenodo



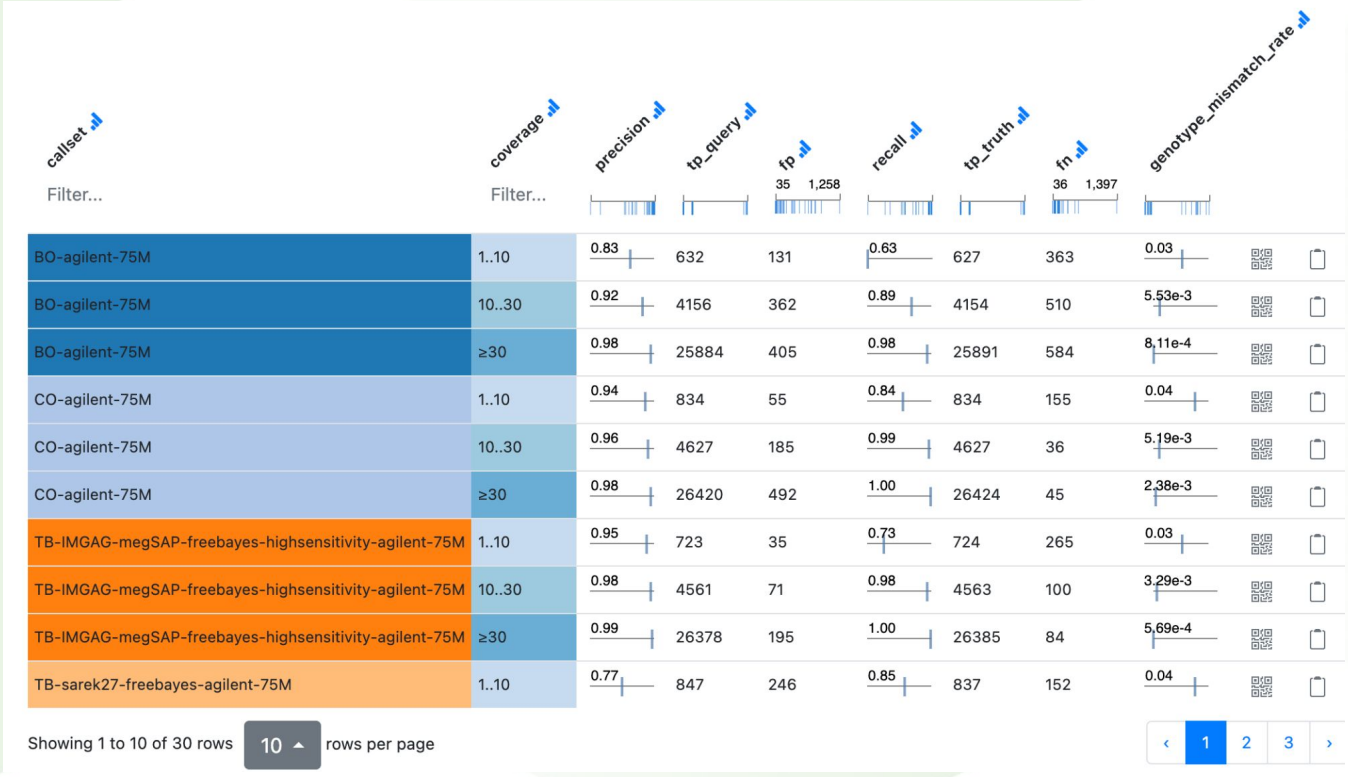
https://github.com/snakemake-workflows/dna-seq-benchmark

https://github.com/ncbench/ncbench-workflow

GHGA

NGS-CN

Continuous Benchmarking



callset	coverage	precision	tp_query	fp	recall	tp_truth	fn	genotype_mismatch_rate
Filter...	Filter...			35 1,258			36 1,397	
BO-agilent-75M	1..10	0.83	632	131	0.63	627	363	0.03
BO-agilent-75M	10..30	0.92	4156	362	0.89	4154	510	5.53e-3
BO-agilent-75M	≥30	0.98	25884	405	0.98	25891	584	8.11e-4
CO-agilent-75M	1..10	0.94	834	55	0.84	834	155	0.04
CO-agilent-75M	10..30	0.96	4627	185	0.99	4627	36	5.19e-3
CO-agilent-75M	≥30	0.98	26420	492	1.00	26424	45	2.38e-3
TB-IMGAG-megSAP-freebayes-highsensitivity-agilent-75M	1..10	0.95	723	35	0.73	724	265	0.03
TB-IMGAG-megSAP-freebayes-highsensitivity-agilent-75M	10..30	0.98	4561	71	0.98	4563	100	3.29e-3
TB-IMGAG-megSAP-freebayes-highsensitivity-agilent-75M	≥30	0.99	26378	195	1.00	26385	84	5.69e-4
TB-sarek27-freebayes-agilent-75M	1..10	0.77	847	246	0.85	837	152	0.04

Showing 1 to 10 of 30 rows 10 rows per page

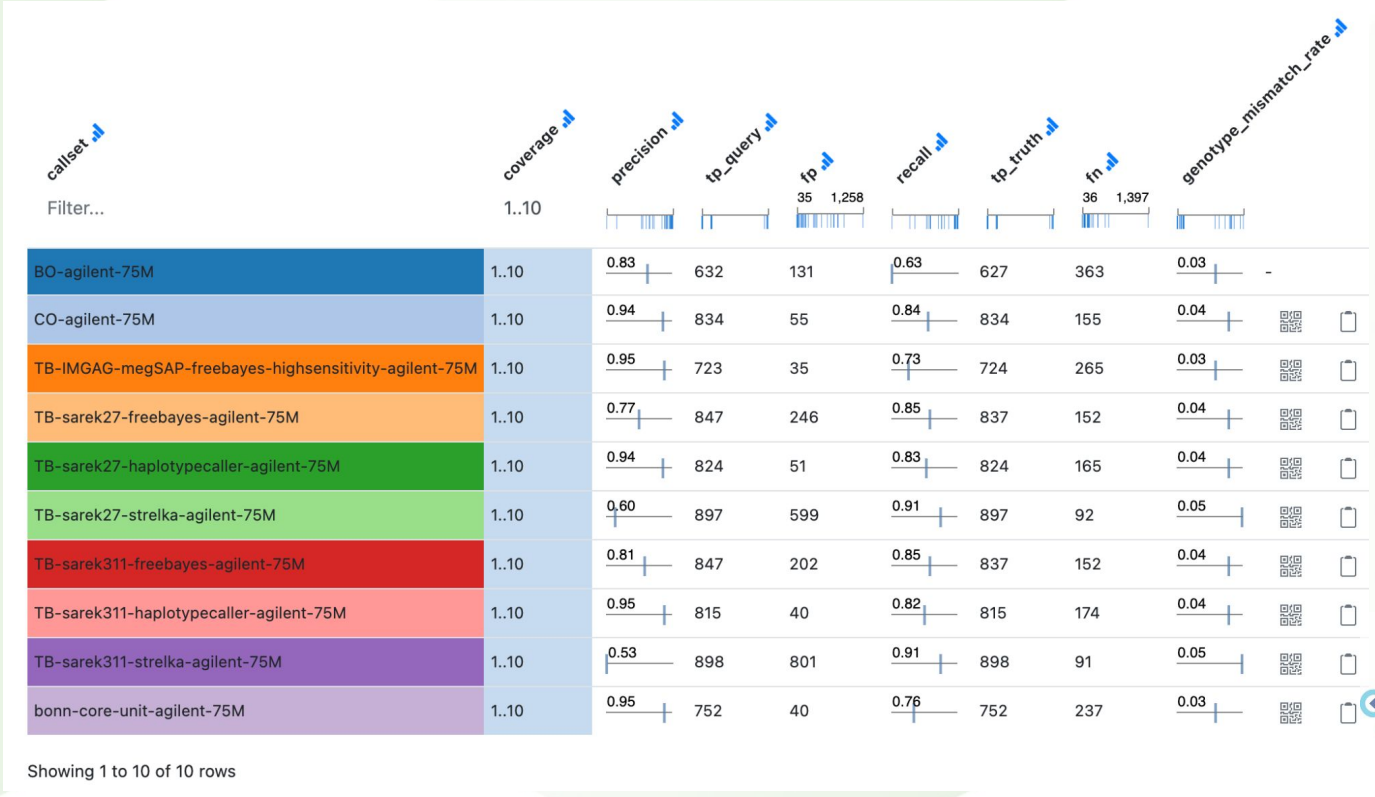
‹ 1 2 3 ›

https://ncbench.github.io/report/report.html#

GHGA

NGS-CN

Continuous Benchmarking



callset	coverage	precision	tp_query	fp	recall	tp_truth	fn	genotype_mismatch_rate
Filter...	1..10			35 1,258			36 1,397	
BO-agilent-75M	1..10	0.83	632	131	0.63	627	363	0.03 -
CO-agilent-75M	1..10	0.94	834	55	0.84	834	155	0.04
TB-IMGAG-megSAP-freebayes-highsensitivity-agilent-75M	1..10	0.95	723	35	0.73	724	265	0.03
TB-sarek27-freebayes-agilent-75M	1..10	0.77	847	246	0.85	837	152	0.04
TB-sarek27-haplotypecaller-agilent-75M	1..10	0.94	824	51	0.83	824	165	0.04
TB-sarek27-strelka-agilent-75M	1..10	0.60	897	599	0.91	897	92	0.05
TB-sarek311-freebayes-agilent-75M	1..10	0.81	847	202	0.85	837	152	0.04
TB-sarek311-haplotypecaller-agilent-75M	1..10	0.95	815	40	0.82	815	174	0.04
TB-sarek311-strelka-agilent-75M	1..10	0.53	898	801	0.91	898	91	0.05
bonn-core-unit-agilent-75M	1..10	0.95	752	40	0.76	752	237	0.03

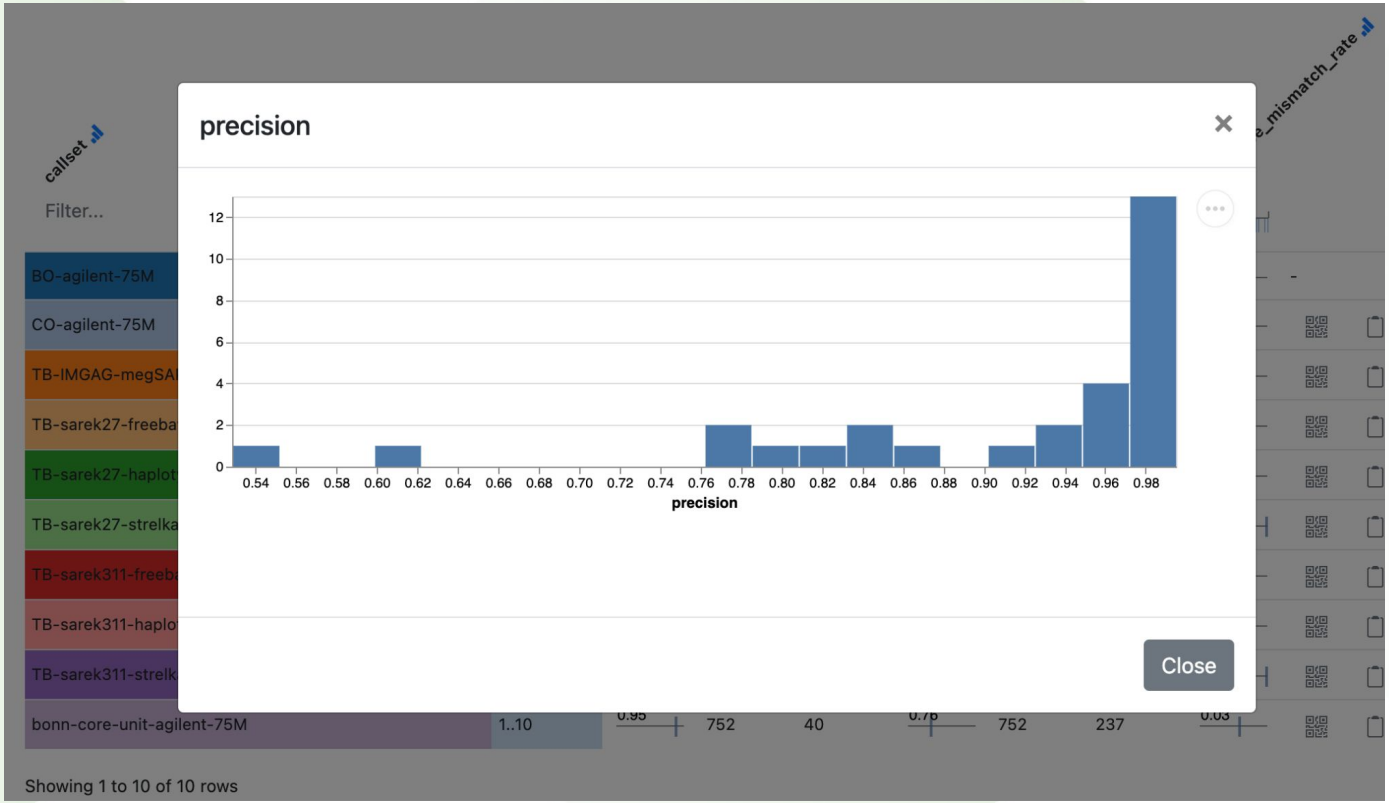
Showing 1 to 10 of 10 rows

https://ncbench.github.io/report/report.html#

GHGA

NGS-CN

Continuous Benchmarking



https://ncbench.github.io/report/report.html#

Benchmarking plan of GHGA

Efforts to make benchmarking FAIR

- Making all benchmark datasets **available**
  - Provide sequencing data through Zenodo (GiaB)

- Make benchmarking workflows **FAIR**
  - Harmonized benchmarking workflows
  - Analysis workflows for reproducibility

- Integration of benchmarking into workflow development
  - Setting up CI/CD for major workflow releases

GHGA

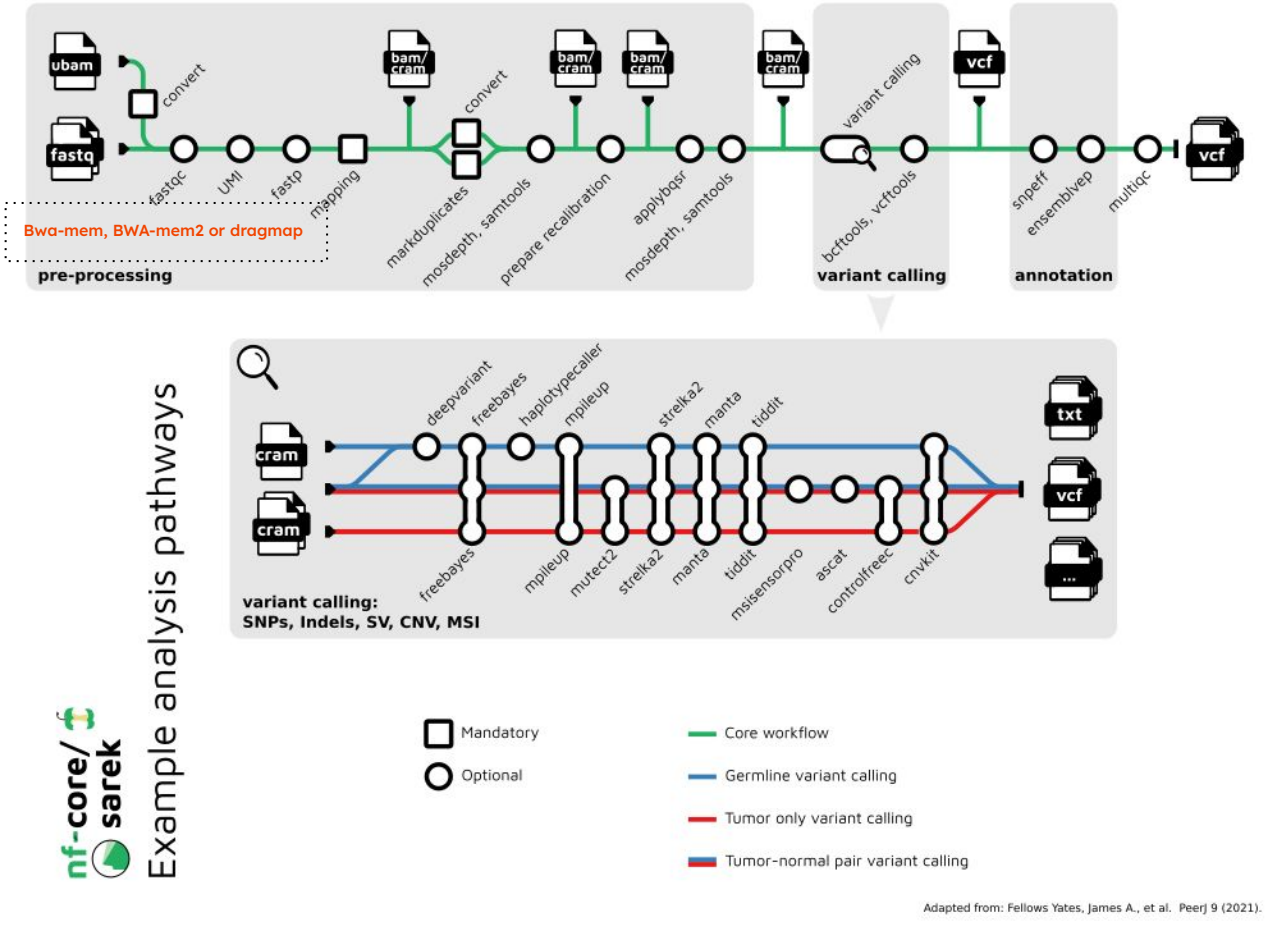
An Example Benchmark Analysis

NCBench and nf-core/sarek

GHGA

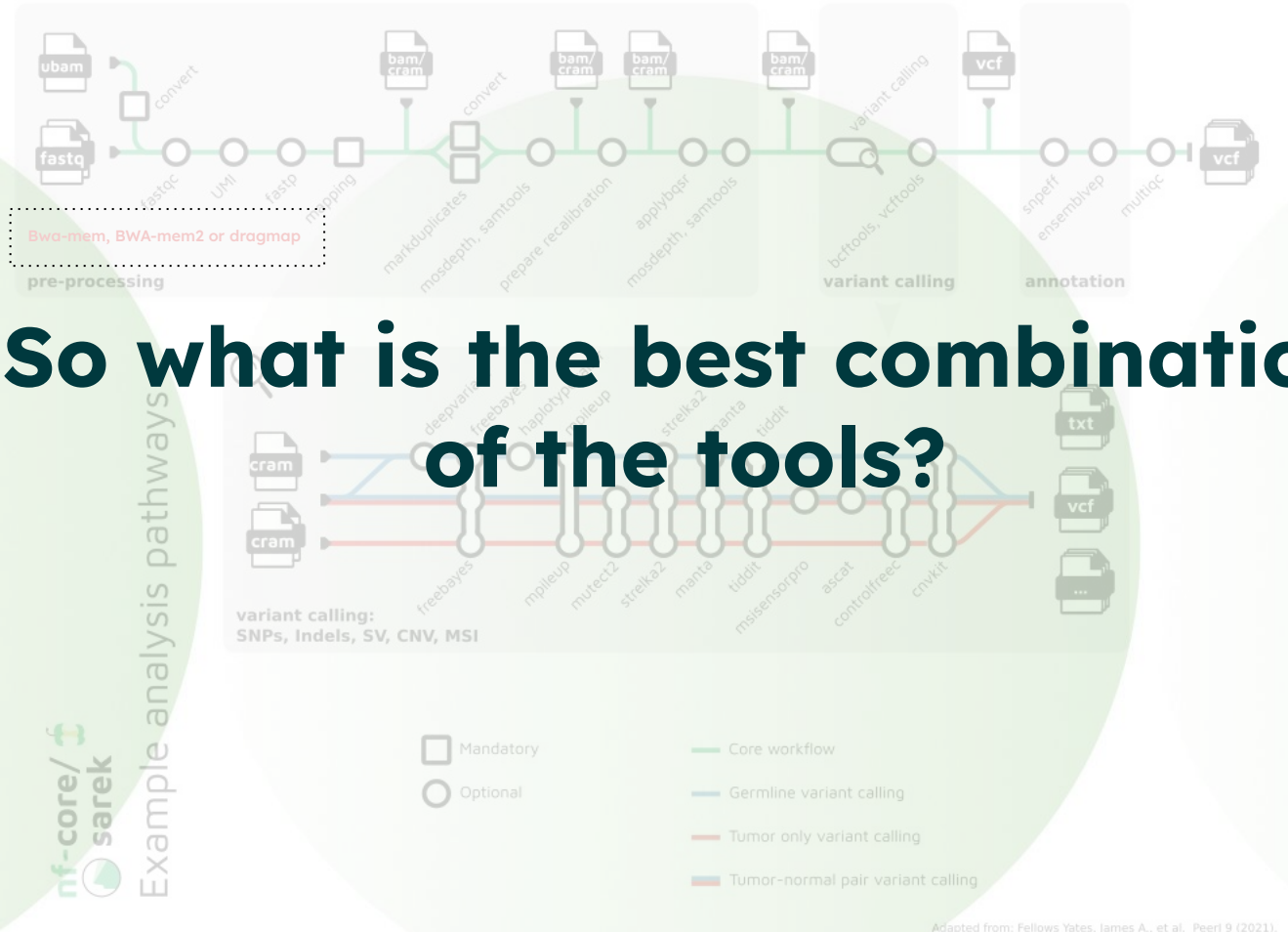
Sarek

Collection of variant callers in a single workflow



https://github.com/nf-core/sarek

So what is the best combination of the tools?



https://github.com/nf-core/sarek

Benchmark plan

Best pipeline to use for germline analysis for small sequencing WES data ?

- Benchmark samples to test:
  - CHM synthetic dataset
  - HG001 GIAB sample sequenced on 2 different platforms (agilent and twist)

3 Aligners		5 Variant callers			
bwa-mem bwa-mem2 dragmap	X	freebayes strelka deepvariant haplotypecaller mpileup	x 3 datasets	=	75 runs!

GHGA

```
> nextflow run nf-core/sarek -r 2.5.2 -profile singularity --input
samplesheet.csv --genome GATK.GRCh38 -outdir results --tools
Mutect2,Strelka,Manta,TIDDIT,ASCAT,ControlFREEC,snpEff,VEP
```

```
>less samplesheet.csv
patient,sample,lane,fastq 1,fastq 2
patient1,test_sample,lane_1,test_L001_1.fastq.gz,test_L001_2.fastq.gz
patient1,test_sample,lane_2,test_L002_1.fastq.gz,test_L002_2.fastq.gz
patient1,test_sample,lane_3,test_L003_1.fastq.gz,test_L003_2.fastq.gz
```

GHGA

```
> nextflow run nf-core/sarek -r 2.5.2 -profile singularity --input
samplesheet.csv --genome GATK.GRCh38 -outdir results --tools
Mutect2,Strelka,Manta,TIDDIT,ASCAT,ControlFREEC,snpEff,VEP
```

```
>less samplesheet.csv
patient,sample,lane,fastq 1,fastq 2
patient1,test_sample,lane_1,test_L001_1.fastq.gz,test_L001_2.fastq.gz
patient1,test_sample,lane_2,test_L002_1.fastq.gz,test_L002_2.fastq.gz
patient1,test_sample,lane_3,test_L003_1.fastq.gz,test_L003_2.fastq.gz
patient2,test_sample2,lane_1,test2_L001_1.fastq.gz,test2_L001_2.fastq.g
z
```

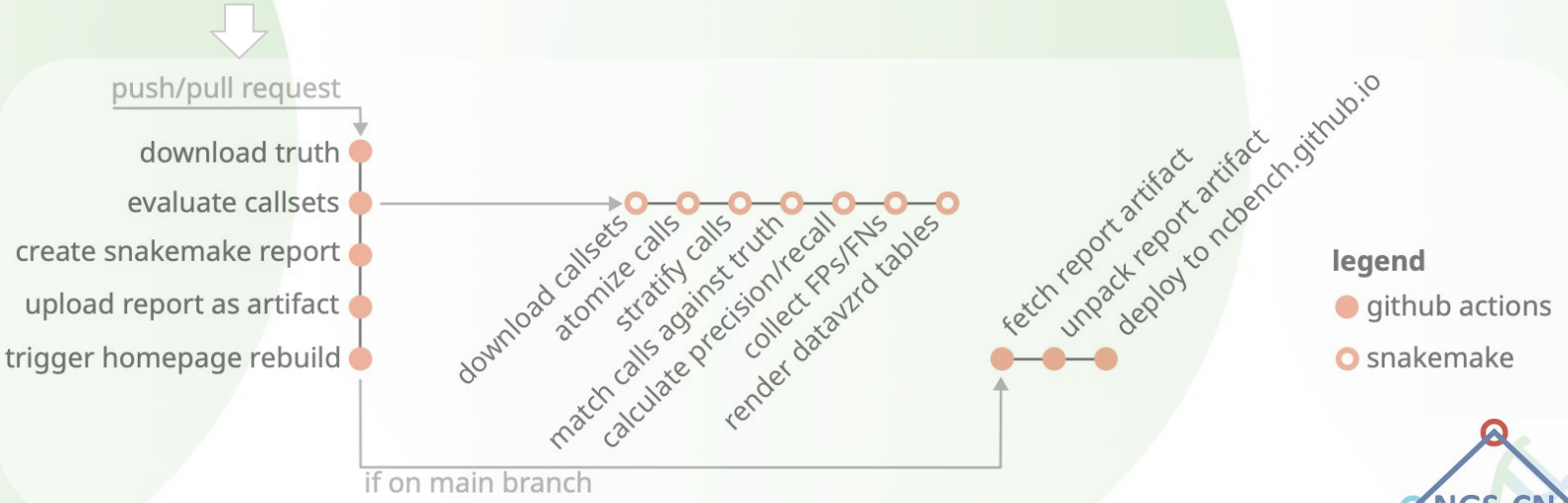
GHGA

```
params {
    config_profile_contact     = 'Kübra Narcı kuebra.narci@dkfz-heidelberg.de'
    config_profile_name        = 'DKFZ cluster'
    max_cpus   = 30
    max_memory = '250.GB'
    max_time   = '48.h'
}
singularity {
  enabled = true
  autoMounts = true
}
process {
  scratch = '$SCRATCHDIR/$LSB_JOBID'
}
executor {
  name = 'lsf'
  perTaskReserve = false
  queueSize = 10
  submitRateLimit = '3 sec'
}
profiles{
    alignment {
        params {
            genome   = 'GATK.GRCh38'
            tools    = 'Strelka'
            aligner  = 'bwa-mem'
        }
    }
```

GHGA

NCBench Benchmark workflow

- Everyone can contribute!
- So how do you do it?



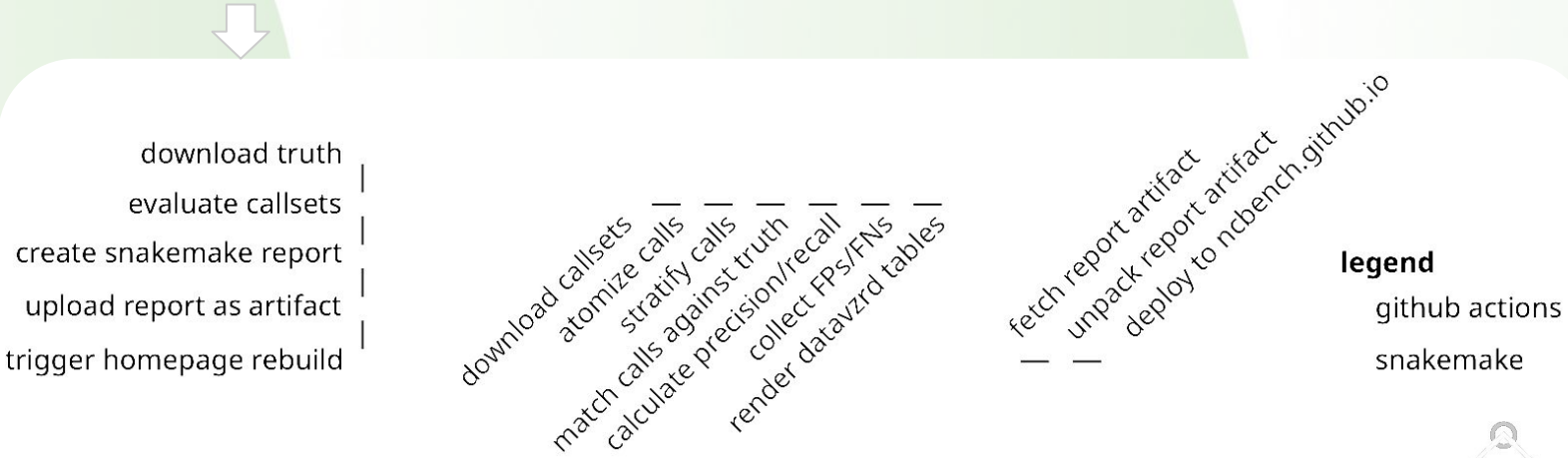
https://ncbench.github.io/report/report.html#

GHGA

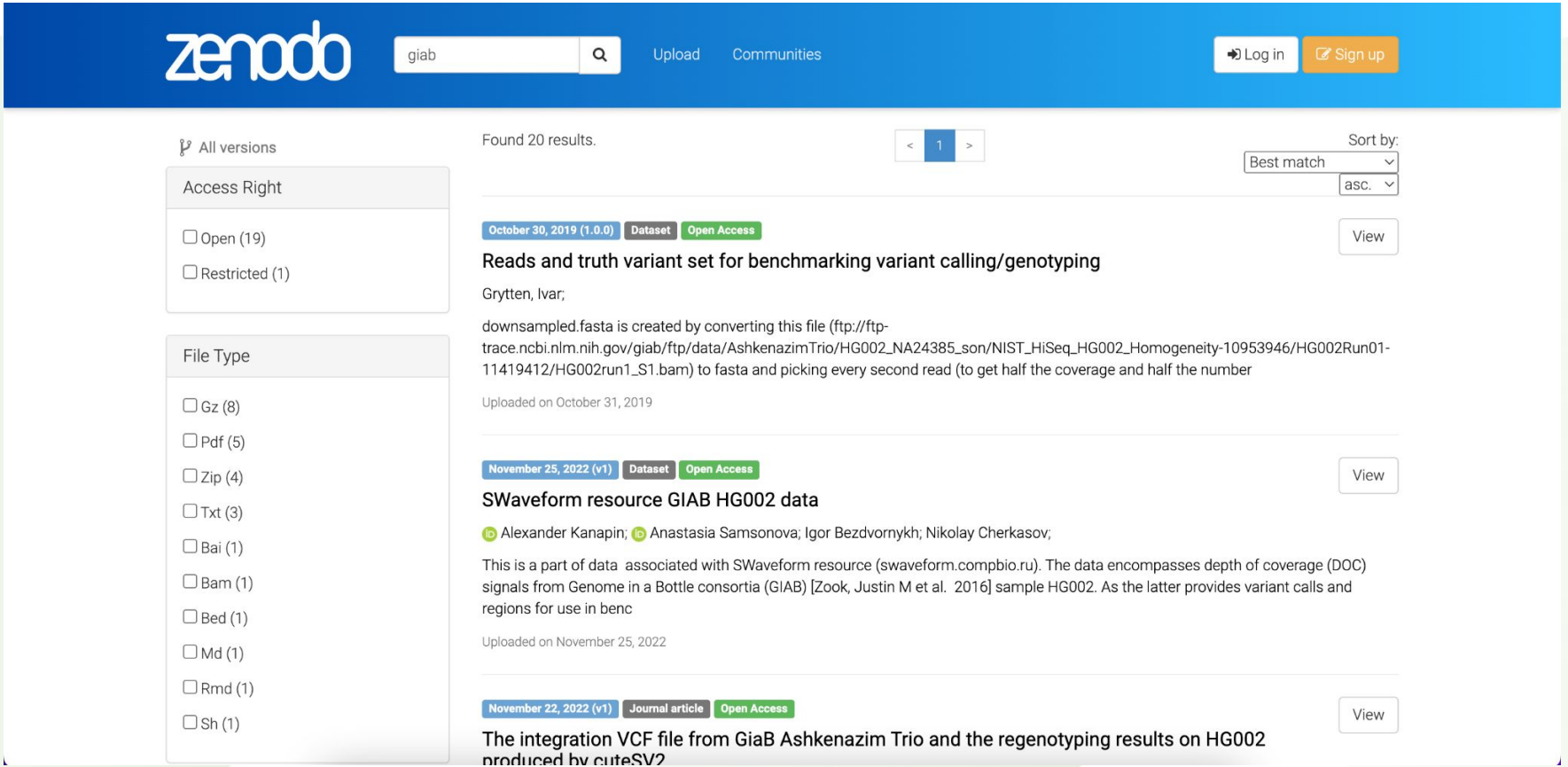
NGS-CN

NCBench Benchmark workflow

1. **Download raw data**
2. **Run your pipeline on it**
3. Upload your results (VCF or BCF) to zenodo
4. Add IDs to NCBench config file, push request will trigger



5. Result will be found here https://ncbench.github.io/report/report.html#



https://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release

https://github.com/lh3/CHM-eval

GHGA

ghga.config

```
profiles{
    alignment_bwa_mem {
        params {
            genome    = 'GATK.GRCh38'
            tools     = 'freebayes,Strelka,deepvariant,haplotypecaller,mpileup'
            aligner   = 'bwa-mem'
        }
    }
profiles{
    alignment_bwa_mem2 {
        params {
            genome    = 'GATK.GRCh38'
            tools     = 'freebayes,Strelka,deepvariant,haplotypecaller,mpileup'
            aligner   = 'bwa-mem2'
        }
    }
profiles{
    alignment_dragmap {
        params {
            genome    = 'GATK.GRCh38'
            tools     = 'freebayes,Strelka,deepvariant,haplotypecaller,mpileup'
            aligner   = 'dragmap'
        }
    }
```

GHGA

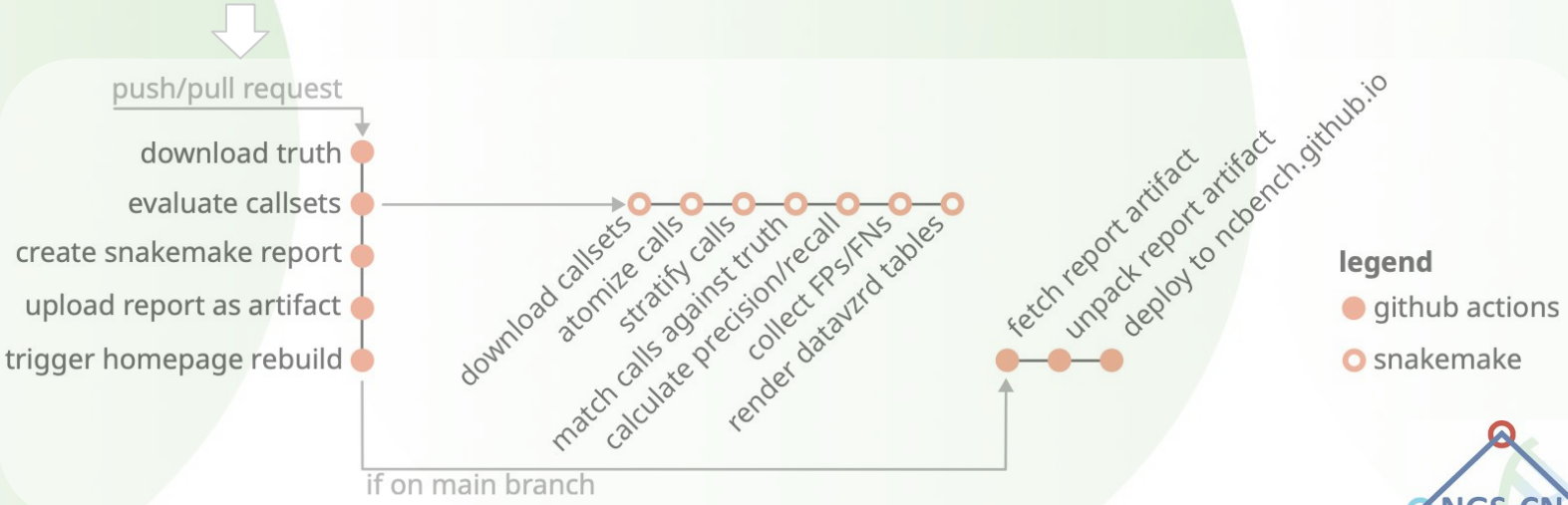
```
> nextflow run nf-core/sarek -r 2.5.2 -profile singularity  --input
samplesheet.csv —-outdir bwa_mem  —-aligner bwa-mem -tools
freebayes,strelka,deepvariant,haplotypecaller,mpileup

> nextflow run nf-core/sarek -r 2.5.2 -profile singularity  --input
samplesheet.csv --outdir bwa_mem2 —-aligner bwa-mem2 -tools
freebayes,strelka,deepvariant,haplotypecaller,mpileup

> nextflow run nf-core/sarek -r 2.5.2 -profile singularity  --input
samplesheet.csv —-outdir dragmap -aligner dragmap -tools
freebayes,strelka,deepvariant,haplotypecaller,mpileup
```

```
>less samplesheet.csv
patient,sample,lane,fastq 1,fastq 2
NA12878 agilent,NA12878,lane 1,NA12878 agilent 1.fastq.gz,NA12878 agilent 2.fastq.gz
NA12878 twist,NA12878,lane 1,NA12878 agilent 1.fastq.gz,NA12878 agilent_2.fastq.gz
ERR1341796,CHM,lane_1,ERR1341796_1.fastq.gz,ERR1341796_2.fastq.gz
```

GHGA

NCBench Benchmark workflow

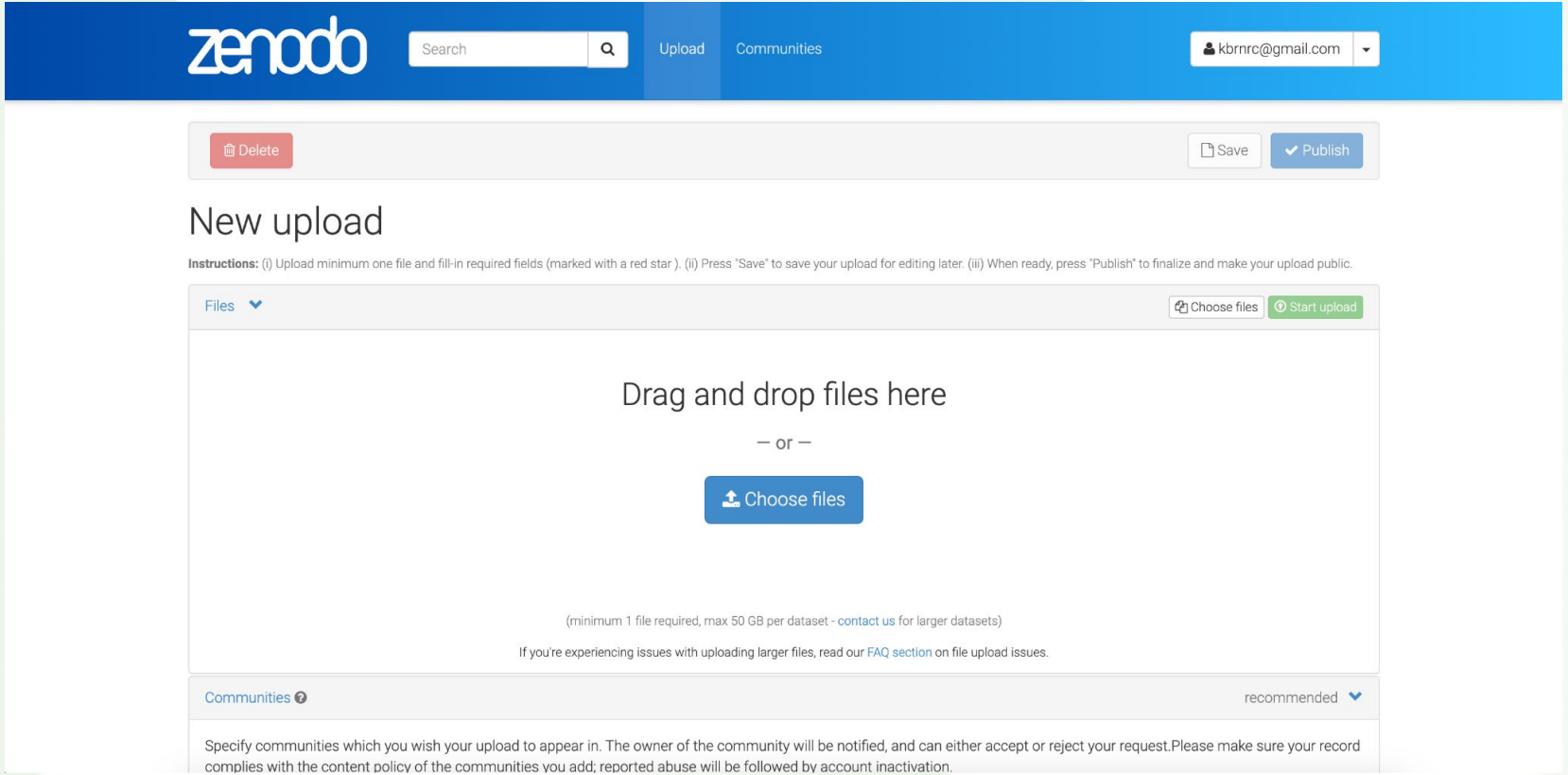
1. Download raw data
2. Run your pipeline on it
3. **Upload your results (VCF) to** zenodo
4. Add IDs to NCBench config file, push request will trigger



5. Result will be found here https://ncbench.github.io/report/report.html#

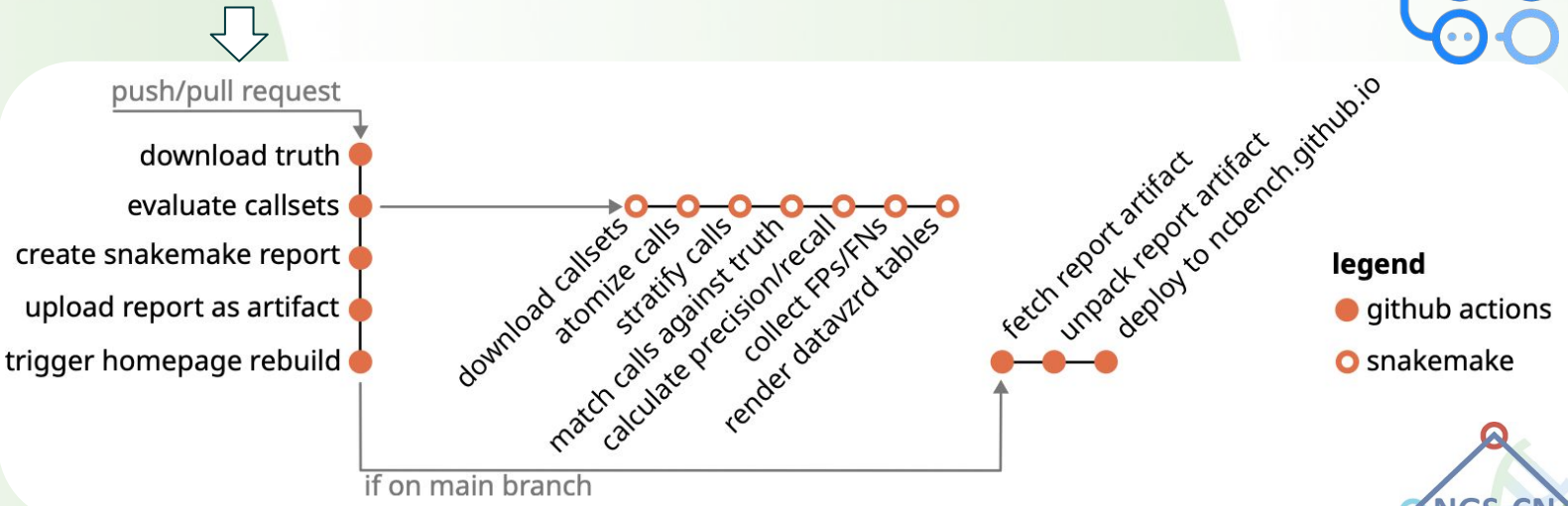
GHGA

NGS-CN


zenodo
Search
Upload
Communities
kbrnrc@gmail.com
Delete
Save
Publish
New upload
Instructions: (i) Upload minimum one file and fill-in required fields (marked with a red star). (ii) Press "Save" to save your upload for editing later. (iii) When ready, press "Publish" to finalize and make your upload public.
Files
Choose files
Start upload
Drag and drop files here
— or —
Choose files
(minimum 1 file required, max 50 GB per dataset - contact us for larger datasets)
If you're experiencing issues with uploading larger files, read our FAQ section on file upload issues.
Communities
recommended
Specify communities which you wish your upload to appear in. The owner of the community will be notified, and can either accept or reject your request.Please make sure your record complies with the content policy of the communities you add; reported abuse will be followed by account inactivation.

NCBench Benchmark workflow

1. Download raw data
2. Run your pipeline on it
3. Upload your results (VCF or BCF) to zenodo
4. **Add IDs to NCBench config file, push request will trigger**



5. Result will be found here https://ncbench.github.io/report/report.html#

GHGA

NGS-CN

1. Add your results to **NCBench config file**
2. Triggers **snakemake-workflows/dna-seq-benchmark** on github actions
3. Creates snakemake reports
4. Homepage rebuilds https://ncbench.github.io/report/report.html#
5. View!

ncbench-workflow / config / config.yaml

Code | Blame | 665 lines (644 loc) · 21.6 KB

```yaml
104     GHGA-sarek-dragmap-haplotypecaller-agilent-200M:
105       labels:
106         site: GHGA-Munich
107         pipeline: nf-core/sarek v3.1 dragmap-haplotypecaller
108         reads: 200M
109       subcategory: NA12878-agilent
110       zenodo:
111         deposition: 7105204
112         filename: NA12878_dragmap.haplotypecaller.filtered.vcf.gz
113       benchmark: giab-NA12878-agilent-200M
114       rename-contigs: resources/rename-contigs/ucsc-to-ensembl.txt
115     GHGA-sarek-dragmap-strelka-agilent-200M:
116       labels:
117         site: GHGA-Munich
118         pipeline: nf-core/sarek v3.1 dragmap-strelka
119         reads: 200M
120       subcategory: NA12878-agilent
121       zenodo:
122         deposition: 7105204
123         filename: NA12878_dragmap.strelka.variants.vcf.gz
124       benchmark: giab-NA12878-agilent-200M
125       rename-contigs: resources/rename-contigs/ucsc-to-ensembl.txt
126     GHGA-sarek-dragmap-freebayes-agilent-200M:
127       labels:
128         site: GHGA-Munich
129         pipeline: nf-core/sarek v3.1 dragmap-freebayes
130         reads: 200M
131       subcategory: NA12878-agilent
132       zenodo:
133         deposition: 7105204
134         filename: NA12878_dragmap.freebayes.vcf.gz
135       benchmark: giab-NA12878-agilent-200M
136       rename-contigs: resources/rename-contigs/ucsc-to-ensembl.txt
137     GHGA-sarek-dragmap-deepvariant-agilent-200M:
138       labels:
139         site: GHGA-Munich
140         pipeline: nf-core/sarek v3.1 dragmap-deepvariant
```

https://ncbench.github.io/report/report.html#

Summary

- NGS data is growing as well as the methods to analyse them
- Challenge is which tool to use when
- Benchmark results from tool developers might not be transparent
- Integration of latest technologies to generate more objective environment
- NGS-CN + GHGA for standardized and open benchmark analysis

GHGA

Thank you!

And thanks to the team and the PIs!

Florian Heyl

Paul Menges

Kübra Narci

Luiz Gadelha

Evangelos Theodorakis

Christian Mertes

GHGA

References

- Krusche, P., Trigg, L., Boutros, P.C. et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol 37, 555–560 (2019). https://doi.org/10.1038/s41587-019-0054-x
- Olson, N.D., Wagner, J., Dwarshuis, N. et al. Variant calling and benchmarking in an era of complete human genome sequences. Nat Rev Genet (2023). https://doi.org/10.1038/s41576-023-00590-0
- Weber, L.M., Saelens, W., Cannoodt, R. et al. Essential guidelines for computational method benchmarking. Genome Biol 20, 125 (2019). https://doi.org/10.1186/s13059-019-1738-8
- Mangul, S., Martin, L.S., Hill, B.L. et al. Systematic benchmarking of omics computational tools. Nat Commun 10, 1393 (2019). https://doi.org/10.1038/s41467-019-09406-4
- Chen, J., Li, X., Zhong, H. et al. Systematic comparison of germline variant calling pipelines cross multiple next-generation sequencers. Sci Rep 9, 9345 (2019). https://doi.org/10.1038/s41598-019-45835-3
- Zhao, S., Agafonov, O., Azab, A. et al. Accuracy and efficiency of germline variant calling pipelines for human genome data. Sci Rep 10, 20222 (2020). https://doi.org/10.1038/s41598-020-77218-4

GHGA