BALSAMIC

unknown

Apr 10, 2024

INSTALLATION AND USAGE

BALSAMIC is basically a wrapper for its core workflow manager. The goal is to have a package with well defined cli to make it reproducible for user to run somatic calling regaradless of the workflow manger at its core. Right now, BALSAMIC is using Snakemake as its core. So one can run the sample using workflows available within this package and standard Snakemake cli given that there is a proper config file created.

Source code	https://github.com/Clinical-Genomics/BALSAMIC
Version	
Author	Hassan Foroughi Asl
Development model	Gitflow
Build status	
Container latest release status	
Container master and develop status	
Code coverage	
Documentation	
Dependencies	
Contributors	@ashwini06, @ivadym, @khurrammaqbool, @keyvanelhami, @mropat, @imsarath, @rannick, @fevac, @mathiasbio

This section describes steps to install BALSAMIC (**version** = 15.0.0)

CHAPTER

ONE

SOFTWARE REQUIREMENTS

- Conda >=version 4.5.0: For detailed software and python requirements please see `setup.py` and `BALSAMIC/conda/balsamic.yaml`
- Singularity >=version 3.0.0: BALSAMIC uses singularity to run vairous parts of the workflow.
- Python 3.11
- BALSAMIC is dependent on third-party bioinformatics software `Sentieon-tools`. Example: for running wgs variant calling using `TNScope`, and to execute `UMIworkflow`.

`Note: Set Sentieon envionment variables in your ~/.bashrc file by adding following two lines`

```
export SENTIEON_INSTALL_DIR=path_to_sentieon_install_dir
export SENTIEON_LICENSE=IP:Port
```

CHAPTER

TWO

STEP 1. INSTALLING BALSAMIC

1. Create a conda environment:

```
conda create -c conda-forge -c defaults --name S_balsamic python==3.11 pip pygraphviz␣
↪wkhtmltopdf
```

2. Activate environment:

```
conda activate S_BALSAMIC
```

3. Install BALSAMIC using `pip` within the newly created environment:

```
pip install --no-cache-dir -U git+https://github.com/Clinical-Genomics/BALSAMIC
```

Or if you have repository cloned and want it in editable mode:

```
pip install -e .
```

CHAPTER

THREE

STEP 2. GENERATE BALSAMIC CACHE AND PULL CONTAINERS

First generate your own COSMIC database key via: https://cancer.sanger.ac.uk/cosmic/help/file_download The following commands will create and download reference directory at `~/balsamic_cache` (change this path if you want it to be created in another location):

NOTE: This process can take couple of hours

```
# Note:
# 1. COSMIC key is in variable $COSMIC_KEY
# 2. For genome version hg38, set --genome-version to hg38
# 3. For using develop container version, set --cache-version to develop
# 4. For submitting jobs to slurm cluster, use option --account

balsamic init --outdir ~/balsamic_cache \
  --cosmic-key "${COSMIC_KEY}" \
  --genome-version hg19 \
  --run-analysis \
  --account development

# Generate cache locally instead of slurm job submission
balsamic init --outdir ~/balsamic_cache \
  --cosmic-key "${COSMIC_KEY}" \
  --genome-version hg19 \
  --run-analysis \
  --run-mode local \
  --snakemake-opt "--cores 16"
```

CHAPTER

FOUR

SHORT TUTORIAL

Here a short tutorial is provided for BALSAMIC (**version** = 15.0.0).

4.1 Regarding fastq-inputs

Previous versions of BALSAMIC only accepted one fastq-pair per sample, which required concatenation of fastq-pairs if multiple existed.

The current version BALSAMIC takes `--fastq-path` instead which is a path to a directory containing ALL fastq-files you want to include in the analysis, for tumor and normal (if it exists for the analysis).

NOTE: The fastq-files in `--fastq-path` need to contain the names from `--tumor-sample-name [sample_name]` and `--normal-sample-name [sample_name]` as a sub-string in the fastq-names to correctly assign them to their respective sample.

4.2 Running a test sample

Example config demo case:

```
balsamic config case \
  --analysis-dir demo/
  --balsamic-cache ~/balsamic_cache
  --fastq-path tests/test_data/fastq/
  --case-id demo_run_balsamic
  --gender female
  --analysis-workflow balsamic
  --genome-version hg19
  --tumor-sample-name S1
  --panel-bed tests/test_data/references/panel/panel.bed
```

Let's try a dry run and see everything is in place:

```
balsamic run analysis --sample-config demo/demo_run_balsamic/demo_run_balsamic.json
```

Command above should exit a similar output as below:

```
Job counts:
count jobs
1 BaseRecalibrator
```

(continues on next page)

(continued from previous page)

```
1 CollectAlignmentSummaryMetrics
1 CollectHsMetrics
1 CollectInsertSizeMetrics
1 IndelRealigner
1 MarkDuplicates
1 RealignerTargetCreator
1 all
1 bwa_mem
1 cnvkit_single
1 fastp
1 fastqc
13  haplotypecaller
1 haplotypecaller_merge
1 manta_germline
1 manta_tumor_only
1 mergeBam_tumor
1 mergeBam_tumor_gatk
1 multiqc
1 mutect2_merge
13  mutect2_tumor_only
1 sambamba_exon_depth
1 sambamba_panel_depth
1 samtools_sort_index
1 somatic_snv_indel_vcf_merge
1 split_bed_by_chrom
1 strelka_germline
1 vardict_merge
13  vardict_tumor_only
7 vep
72
This was a dry-run (flag -n). The order of jobs does not reflect the order of execution.
```

And now run balsamic through SLURM. Make sure you set your SLURM project account using `--account` if your local settings require it:

```
balsamic run analysis --sample-config demo/demo_run_balsamic/demo_run_balsamic.json \
  --profile slurm --qos low --account development --run-analysis
```

And now run balsamic through QSUB. Make sure you set your QSUB project account using `--account` if your local settings require it:

```
balsamic run analysis --sample-config demo/demo_run_balsamic/demo_run_balsamic.json \
  --profile qsub --qos low --account development --run-analysis
```

And running workflow without submitting jobs. Set number of cores by passing an argument to snakemake as seen below:

```
balsamic run analysis --sample-config demo/demo_run_balsamic/demo_run_balsamic.json \
  --run-mode local --snakemake-opt "--cores 8" --run-analysis
```

CHAPTER

FIVE

COMMAND LINE ARGUMENTS

5.1 BALSAMIC

Balsamic 15.0.0: Bioinformatic Analysis Pipeline for Somatic Mutations in Cancer

```
BALSAMIC [OPTIONS] COMMAND [ARGS]...
```

Options

--log-level <log_level>

Logging level in terms of urgency

Default
INFO

Options
NOTSET | DEBUG | INFO | WARNING | ERROR | FATAL | CRITICAL

--version

Show the version and exit.

5.1.1 config

Create config files required for running the pipeline.

```
BALSAMIC config [OPTIONS] COMMAND [ARGS]...
```

case

Create a sample config file from input sample data

```
BALSAMIC config case [OPTIONS]
```

Options

--adapter-trim, --no-adapter-trim

Trim adapters from reads in FASTQ file

Default
True

--analysis-dir <analysis_dir>

Required Path to store the analysis results

-w, --analysis-workflow <analysis_workflow>

Balsamic analysis workflow to be executed

Default
balsamic

Options
balsamic | balsamic-qc | balsamic-umi

-b, --background-variants <background_variants>

Background set of valid variants for UMI

--balsamic-cache <balsamic_cache>

Required Path to BALSAMIC cache

--cache-version <cache_version>

Cache version to be used for init or analysis. Use 'develop' or 'X.X.X'.

Default
15.0.0

--cadd-annotations <cadd_annotations>

Path of CADD annotations

--cancer-germline-snv-observations <cancer_germline_snv_observations>

VCF path of cancer germline SNV normal observations (WGS analysis workflow)

--cancer-somatic-snv-observations <cancer_somatic_snv_observations>

VCF path of cancer SNV tumor observations (WGS analysis workflow)

--cancer-somatic-sv-observations <cancer_somatic_sv_observations>

VCF path of cancer SV observations (WGS analysis workflow)

--case-id <case_id>

Required Sample ID for reporting, naming the analysis jobs, and analysis path

--clinical-snv-observations <clinical_snv_observations>

VCF path of clinical SNV observations (WGS analysis workflow)

--clinical-sv-observations <clinical_sv_observations>

VCF path of clinical SV observations (WGS analysis workflow)

--exome

Assign exome parameters to TGA workflow

--fastq-path <fastq_path>

Required Path to directory containing unconcatenated FASTQ files

--gender <gender>

Sample associated gender

Default
female

Options
female | male

-g, --genome-version <genome_version>

Type and build version of the reference genome

Default
hg19

Options
hg19 | hg38 | canfam3

--genome-interval <genome_interval>

Genome 100 bp interval-file (created with gatk PreprocessIntervals), used for GENS pre-processing.

--gens-coverage-pon <gens_coverage_pon>

GENS PON file, either male or female (created with gatk CreateReadCountPanelOfNormals), used for GENS pre-processing.

--gnomad-min-af5 <gnomad_min_af5>

Gnomad VCF filtered to keep >= 0.05 AF, used for GENS pre-processing.

--normal-sample-name <normal_sample_name>

Normal sample name

-p, --panel-bed <panel_bed>

Panel bed file of target regions

--pon-cnn <pon_cnn>

Panel of normal reference (.cnn) for CNVkit

--quality-trim, --no-quality-trim

Trim low quality reads in FASTQ file

Default
True

--swegen-snv <swegen_snv>

VCF path of Swegen SNV frequency database

--swegen-sv <swegen_sv>

VCF path of Swegen SV frequency database

--tumor-sample-name <tumor_sample_name>

Required Tumor sample name

--umi, --no-umi

UMI processing steps for samples with UMI tags. For WGS cases, UMI is always disabled.

Default
True

--umi-trim-length <umi_trim_length>
Trim N bases from reads in FASTQ file

Default
5

pon

Create a sample config file for PON analysis

```
BALSAMIC config pon [OPTIONS]
```

Options

--adapter-trim, --no-adapter-trim
Trim adapters from reads in FASTQ file

Default
True

--analysis-dir <analysis_dir>
Required Path to store the analysis results

--balsamic-cache <balsamic_cache>
Required Path to BALSAMIC cache

--cache-version <cache_version>
Cache version to be used for init or analysis. Use 'develop' or 'X.X.X'.

Default
15.0.0

--case-id <case_id>
Required Sample ID for reporting, naming the analysis jobs, and analysis path

--fastq-path <fastq_path>
Required Path to directory containing unconcatenated FASTQ files

-g, --genome-version <genome_version>
Type and build version of the reference genome

Default
hg19

Options
hg19 | hg38 | canfam3

--genome-interval <genome_interval>
Genome 100 bp interval-file (created with gatk PreprocessIntervals), used for GENS pre-processing.

-p, --panel-bed <panel_bed>
Panel bed file of target regions

--pon-workflow <pon_workflow>
Required Specify which PON to create.

Options
CNVkit | GENS_male | GENS_female

-v, --version <version>
Version of the PON file to be generated

--quality-trim, --no-quality-trim
Trim low quality reads in FASTQ file

Default
True

--umi, --no-umi
UMI processing steps for samples with UMI tags. For WGS cases, UMI is always disabled.

Default
True

--umi-trim-length <umi_trim_length>
Trim N bases from reads in FASTQ file

Default
5

5.1.2 init

Validate inputs and download reference caches and containers.

```
BALSAMIC init [OPTIONS]
```

Options

-o, --out-dir <out_dir>
Required Output directory for singularity containers and reference files

--cache-version <cache_version>
Cache version to be used for init or analysis. Use 'develop' or 'X.X.X'.

Default
15.0.0

--account <account>
Cluster account to run jobs

--cluster-config <cluster_config>
Cluster configuration JSON file path

--mail-user <mail_user>
User email to receive notifications from the cluster

--mail-type <mail_type>
The mail type triggering cluster emails

Options
ALL | BEGIN | END | FAIL | NONE | REQUEUE | TIME_LIMIT

-p, --profile <profile>
Cluster profile to submit jobs

Default
slurm

Options
slurm | qsub

--qos <qos>
QOS for cluster jobs

Default
low

Options
low | normal | high | express

-c, --cosmic-key <cosmic_key>
Cosmic DB authentication key

--force-all
Force execution. This is equivalent to Snakemake –forceall.

Default
False

-g, --genome-version <genome_version>
Type and build version of the reference genome

Default
hg19

Options
hg19 | hg38 | canfam3

-q, --quiet
Instruct Snakemake to not output any progress or rule information

-r, --run-analysis
Flag to run the actual analysis

Default
False

--run-mode <run_mode>
Run mode to execute Balsamic workflows

Default
cluster

Options
cluster | local

-S, --snakefile <snakefile>
Custom Snakefile for internal testing

--snakemake-opt <snakemake_opt>
Options to be passed to Snakemake

5.1.3 report

Command to generate delivery files and check analysis status.

```
BALSAMIC report [OPTIONS] COMMAND [ARGS]...
```

deliver

Report deliver command to generate output analysis files.

```
BALSAMIC report deliver [OPTIONS]
```

Options

--disable-variant-caller <disable_variant_caller>

Run workflow with selected variant caller(s) disable. Use comma to remove multiple variant callers. Valid values are: ['tnscope_umi', 'tnscope', 'dnascope', 'manta', 'cnvkit', 'vardict', 'manta_germline', 'haplotypecaller', 'dellysv', 'tiddit', 'dellycnv', 'ascat', 'cnvpytor', 'igh_dux4', 'svdb']

-r, --rules-to-deliver <rules_to_deliver>

Specify the rules to deliver. The delivery mode selected via the –delivery-mode option.

Options

multiqc | collect_custom_qc_metrics | mergeBam_tumor_umiconsensus | mergeBam_normal_umiconsensus | bam_compress_tumor | bam_compress_normal | vcfheader_rename_germline | vep_annotate_germlineVAR_tumor | vep_annotate_germlineVAR_normal | bcftools_view_split_variant | bcftools_filter_tnscope_research_tumor_only | bcftools_filter_tnscope_research_tumor_normal | bcftools_filter_tnscope_clinical_tumor_only | bcftools_filter_tnscope_clinical_tumor_normal | vardict_merge | bcftools_filter_vardict_research_tumor_only | bcftools_filter_vardict_research_tumor_normal | bcftools_filter_vardict_clinical_tumor_only | bcftools_filter_vardict_clinical_tumor_normal | sentieon_tnscope_umi | sentieon_tnscope_umi_tn | bcftools_filter_TNscope_umi_research_tumor_only | bcftools_filter_TNscope_umi_research_tumor_normal | bcftools_filter_TNscope_umi_clinical_tumor_only | bcftools_filter_TNscope_umi_clinical_tumor_normal | svdb_merge_tumor_only | svdb_merge_tumor_normal | bcftools_filter_sv_research | bcftools_filter_sv_clinical | tiddit_sv_tumor_only | tiddit_sv_tumor_normal | delly_cnv_tumor_only | delly_cnv_tumor_normal | ascat_tumor_normal | cnvpytor_tumor_only | vcf2cytosure_convert_tumor_only | vcf2cytosure_convert_tumor_normal | cnvkit_segment_CNV_research | cnvkit_call_CNV_research | vcf2cytosure_convert | finalize_gens_outputfiles | tmb_calculation | merge_cnv_pdf_reports

-s, --sample-config <sample_config>

Required Sample configuration file

status

Analysis status CLI command.

```
BALSAMIC report status [OPTIONS]
```

Options

-p, --print-files

Print list of analysis files. Otherwise only final count will be printed.

Default
False

-s, --sample-config <sample_config>

Required Sample configuration file

-m, --show-only-missing

Only show missing analysis files.

Default
False

5.1.4 run

Run Balsamic analysis on a provided configuration file.

```
BALSAMIC run [OPTIONS] COMMAND [ARGS]...
```

analysis

Run BALSAMIC workflow on the provided sample's config file.

```
BALSAMIC run analysis [OPTIONS]
```

Options

--benchmark

Profile slurm jobs. Make sure you have slurm profiler enabled in your HPC.

--account <account>

Cluster account to run jobs

--cluster-config <cluster_config>

Cluster configuration JSON file path

--mail-user <mail_user>

User email to receive notifications from the cluster

--mail-type <mail_type>

The mail type triggering cluster emails

Options
ALL | BEGIN | END | FAIL | NONE | REQUEUE | TIME_LIMIT

-p, --profile <profile>

Cluster profile to submit jobs

Default
slurm

Options
slurm | qsub

--qos <qos>

QOS for cluster jobs

Default
low

Options
low | normal | high | express

--disable-variant-caller <disable_variant_caller>

Run workflow with selected variant caller(s) disable. Use comma to remove multiple variant callers. Valid values are: ['tnscope_umi', 'tnscope', 'dnascope', 'manta', 'cnvkit', 'vardict', 'manta_germline', 'haplotypecaller', 'dellysv', 'tiddit', 'dellycnv', 'ascat', 'cnvpytor', 'igh_dux4', 'svdb']

--dragen

Enable dragen variant caller

--force-all

Force execution. This is equivalent to Snakemake –forceall.

Default
False

-q, --quiet

Instruct Snakemake to not output any progress or rule information

-r, --run-analysis

Flag to run the actual analysis

Default
False

--run-mode <run_mode>

Run mode to execute Balsamic workflows

Default
cluster

Options
cluster | local

-s, --sample-config <sample_config>

Required Sample configuration file

-S, --snakefile <snakefile>
Custom Snakefile for internal testing

--snakemake-opt <snakemake_opt>
Options to be passed to Snakemake

CHAPTER

SIX

CALLING AND FILTERING VARIANTS

In BALSAMIC, various bioinfo tools are integrated for reporting somatic and germline variants summarized in the table below. The choice of these tools differs between the type of analysis; *Whole Genome Sequencing (WGS)*, or *Target Genome Analysis (TGA)* and *Target Genome Analysis (TGA) with UMI-analysis activated*.

Table 1: SNV and small-Indel callers

Variant caller	Sequencing type	Analysis type	Somatic/Germline	Variant type
DNAscope	TGA, WGS	tumor-normal, tumor-only	germline	SNV, InDel
TNscope	WGS, TGA (with UMIs activated)	tumor-normal, tumor-only	somatic	SNV, InDel
VarDict	TGA	tumor-normal, tumor-only	somatic	SNV, InDel

Various filters (Pre-call and Post-call filtering) are applied at different levels to report high-confidence variant calls.

Pre-call filtering is where the variant-calling tool decides not to add a variant to the VCF file if the default filters of the variant-caller did not pass the filter criteria. The set of default filters differs between the various variant-calling algorithms.

To know more about the pre-call filters used by the variant callers, please have a look at the VCF header of the particular variant-calling results. For example:

```
##FILTER=<ID=AMPBIAS,Description="Indicate the variant has amplicon bias.">
##FILTER=<ID=Bias,Description="Strand Bias">
##FILTER=<ID=Cluster0bp,Description="Two variants are within 0 bp">
##FILTER=<ID=InGap,Description="The variant is in the deletion gap, thus likely false positive">
##FILTER=<ID=InIns,Description="The variant is adjacent to an insertion variant">
##FILTER=<ID=LongMSI,Description="The somatic variant is flanked by long A/T (>=14)">
##FILTER=<ID=MSI12,Description="Variant in MSI region with 12 non-monomer MSI or 13 monomer MSI">
##FILTER=<ID=NM4.5,Description="Mean mismatches in reads >= 4.5, thus likely false positive">
##FILTER=<ID=Q10,Description="Mean Mapping Quality Below 10">
##FILTER=<ID=SN1.5,Description="Signal to Noise Less than 1.5">
##FILTER=<ID=d3,Description="Total Depth < 3">
##FILTER=<ID=f0.001,Description="Allele frequency < 0.001">
##FILTER=<ID=p8,Description="Mean Position in Reads Less than 8">
##FILTER=<ID=pSTD,Description="Position in Reads has STD of 0">
##FILTER=<ID=q22.5,Description="Mean Base Quality Below 22.5">
##FILTER=<ID=v2,Description="Var Depth < 2">
```

Fig. 1: Pre-call filters applied by the *Vardict* variant-caller is listed in the VCF header.

In the VCF file, the *FILTER* status is *PASS* if this position has passed all filters, i.e., a call is made at this position. Contrary, if the site has not passed any of the filters, a semicolon-separated list of those failed filter(s) will be appended

to the *FILTER* column instead of *PASS*. E.g., *p8;pSTD* might indicate that at this site, the mean position in reads is less than 8, and the position in reads has a standard deviation of 0.

Note: In BALSAMIC, this VCF file is named as `*.<research/clinical>.vcf.gz` (eg: `SNV.somatic.<CASE_ID>.vardict.<research/clinical>.vcf.gz`)

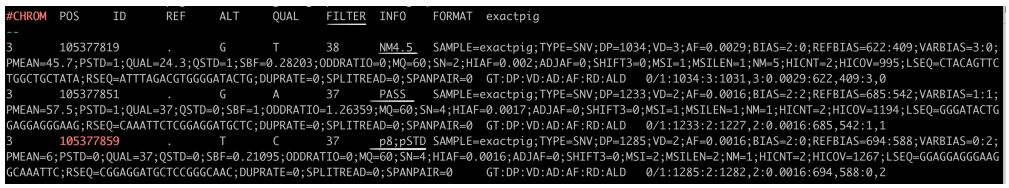


Fig. 2: Vardict Variant calls with different 'FILTER' status underlined in white line (*NM4.5*, *PASS*, *p8;pSTD*)

Post-call filtering is where a variant is further filtered with quality, depth, VAF, etc., with more stringent thresholds.

For *Post-call filtering*, in BALSAMIC we have applied various filtering criteria (*Vardict_filtering*, *TNscope filtering (Tumor_normal)*) depending on the analysis-type (TGS/WGS) and sample-type(tumor-only/tumor-normal).

Note: In BALSAMIC, this VCF file is named as `*.<research/clinical>.filtered.vcf.gz` (eg: `SNV.somatic.<CASE_ID>.vardict.<research/clinical>.filtered.vcf.gz`)

Only those variants that fulfill the pre-call and post-call filters are scored as *PASS* in the *STATUS* column of the VCF file. We filter those *PASS* variants and deliver a final list of variants to the customer either via *Scout* or *Caesar*

Note: In BALSAMIC, this VCF file is named as `*.<research/clinical>.filtered.pass.vcf.gz` (eg: `SNV.somatic.<CASE_ID>.vardict.<research/clinical>.filtered.pass.vcf.gz`)

Table 2: Description of VCF files

VCF file name	Description	Delivered to the customer
.vcf.gz	Unannotated VCF file with pre-call filters included in the STATUS column	Yes (Caesar)
.<research/clinical>.vcf.gz	Annotated VCF file with pre-call filters included in the STATUS column	No
.<research/clinical>.filtered.pass.vcf.g	Annotated and filtered VCF file by excluding all filters that did not meet the pre and post-call filter criteria. Includes only variants with the *PASS* STATUS	Yes (Caesar and Scout)

6.1 Targeted Genome Analysis

6.1.1 Somatic Callers for reporting SNVs/INDELS

Vardict

Vardict is a sensitive variant caller used for both tumor-only and tumor-normal variant calling. The results of *Vardict* variant calling are further post-filtered based on several criteria (*Vardict_filtering*) to retrieve high-confidence variant calls. These high-confidence variant calls are the final list of variants uploaded to Scout or available in the delivered VCF file in Caesar.

There are two slightly different post-processing filters activated depending on if the sample is an exome or a smaller panel as these tend to have very different sequencing depths.

Vardict_filtering

Following is the set of criteria applied for filtering vardict results. It is used for both tumor-normal and tumor-only samples.

Post-call Quality Filters for panels

Mean Mapping Quality (MQ): Refers to the root mean square (RMS) mapping quality of all the reads spanning the given variant site.

```
MQ >= 30
```

Total Depth (DP): Refers to the overall read depth supporting the called variant.

```
DP >= 100
```

Variant depth (VD): Total reads supporting the ALT allele

```
VD >= 5
```

Allelic Frequency (AF): Fraction of the reads supporting the alternate allele

```
Minimum AF >= 0.007
```

Post-call Quality Filters for exomes

Mean Mapping Quality (MQ): Refers to the root mean square (RMS) mapping quality of all the reads spanning the given variant site.

```
MQ >= 30
```

Total Depth (DP): Refers to the overall read depth supporting the called variant.

```
DP >= 20
```

Variant depth (VD): Total reads supporting the ALT allele

```
VD >= 5
```

Allelic Frequency (AF): Fraction of the reads supporting the alternate allele

```
Minimum AF >= 0.007
```

Note: Additionally, the variant is excluded for tumor-normal cases if marked as 'germline' in the `STATUS` column of the VCF file.

Attention: BALSAMIC <= v8.2.7 uses minimum AF 1% (0.01). From Balsamic v8.2.8, minimum VAF is changed to 0.7% (0.007)

Post-call Observation database Filters

GNOMADAF_POPMAX: Maximum Allele Frequency across populations

```
GNOMADAF_popmax <= 0.005  (or) GNOMADAF_popmax == "."
```

SWEGENAF: SweGen Allele Frequency

```
SWEGENAF <= 0.01  (or) SWEGENAF == "."
```

Frq: Frequency of observation of the variants from normal *Clinical* samples

```
Frq <= 0.01  (or) Frq == "."
```

6.1.2 Target Genome Analysis with UMI's into account

Sentieon's TNscope

UMI workflow performs the variant calling of SNVs/INDELS using the *TNscope* algorithm from UMI consensus-called reads. The following filter applies for both tumor-normal and tumor-only samples.

Pre-call Filters

minreads: Filtering of consensus called reads based on the minimum reads supporting each UMI tag group

```
minreads = 3,1,1
```

It means that at least *3* read-pairs need to support the UMI-group (based on the UMI-tag and the aligned genomic positions), and with at least 1 read-pair from each strand (F1R2 and F2R1).

min_init_tumor_lod: Initial Log odds for the that the variant exists in the tumor.

```
min_init_tumor_lod = 0.5
```

min_tumor_lod: Minimum log odds in the final call of variant in the tumor.

```
min_tumor_lod = 4.0
```

min_tumor_allele_frac: Set the minimum tumor AF to be considered as potential variant site.

```
min_tumor_allele_frac = 0.0005
```

interval_padding: Adding an extra 100bp to each end of the target region in the bed file before variant calling.

```
interval_padding = 100
```

Post-call Quality Filters

alt_allele_in_normal: Default filter set by TNscope was considered too stringent in filtering tumor in normal and is removed.

```
bcftools annotate -x FILTER/alt_allele_in_normal
```

Relative tumor AF in normal: Allows for maximum Tumor-In-Normal-Contamination of 30%.

```
excludes variant if: AF(normal) / AF(tumor) > 0.3
```

Post-call Observation database Filters

GNOMADAF_POPMAX: Maximum Allele Frequency across populations

```
GNOMADAF_popmax <= 0.02 (or) GNOMADAF_popmax == "."
```

SWEGENAF: SweGen Allele Frequency

```
SWEGENAF <= 0.01  (or) SWEGENAF == "."
```

Frq: Frequency of observation of the variants from normal *Clinical* samples

```
Frq <= 0.01  (or) Frq == "."
```

The variants scored as *PASS* or *triallelic_sites* are included in the final vcf file (*SNV.somatic.<CASE_ID>.tnscope.<research/clinical>.filtered.pass.vcf.gz*).

6.1.3 Whole Genome Sequencing (WGS)

Sentieon's TNscope

BALSAMIC utilizes the *TNscope* algorithm for calling somatic SNVs and INDELS in WGS samples. The TNscope algorithm performs the somatic variant calling on the tumor-normal or the tumor-only samples.

TNscope filtering (Tumor_normal)

Pre-call Filters

Apply TNscope trained MachineLearning Model: Sets MLrejected on variants with ML_PROB below 0.32.

::

ML model: SentieonTNscopeModel_GiAB_HighAF_LowFP-201711.05.model is applied

min_init_tumor_lod: Initial Log odds for the that the variant exists in the tumor.

```
min_init_tumor_lod = 1
```

min_tumor_lod: Minimum log odds in the final call of variant in the tumor.

```
min_tumor_lod = 8
```

min_init_normal_lod: Initial Log odds for the that the variant exists in the normal.

```
min_init_normal_lod = 0.5
```

min_normal_lod: Minimum log odds in the final call of variant in the normal.

```
min_normal_lod = 1
```

Post-call Quality Filters

Total Depth (DP): Refers to the overall read depth from all target samples supporting the variant call

```
DP(tumor) >= 10 (or) DP(normal) >= 10
```

Allelic Depth (AD): Total reads supporting the ALT allele in the tumor sample

```
AD(tumor) >= 3
```

Allelic Frequency (AF): Fraction of the reads supporting the alternate allele

```
Minimum AF(tumor) >= 0.05
```

alt_allele_in_normal: Default filter set by TNscope was considered too stringent in filtering tumor in normal and is removed.

```
bcftools annotate -x FILTER/alt_allele_in_normal
```

Relative tumor AF in normal: Allows for maximum Tumor-In-Normal-Contamination of 30%.

```
excludes variant if: AF(normal) / AF(tumor) > 0.3
```

Post-call Observation database Filters

GNOMADAF_POPMAX: Maximum Allele Frequency across populations

```
GNOMADAF_popmax <= 0.001 (or) GNOMADAF_popmax == "."
```

```
SWEGENAF <= 0.01  (or) SWEGENAF == "."
```

Frq: Frequency of observation of the variants from normal *Clinical* samples

```
Frq <= 0.01  (or) Frq == "."
```

The variants scored as *PASS* or *triallelic_sites* are included in the final vcf file (*SNV.somatic.<CASE_ID>.tnscope.<research/clinical>.filtered.pass.vcf.gz*).

TNscope filtering (tumor_only)

Pre-call Filters

min_init_tumor_lod: Initial Log odds for the that the variant exists in the tumor.

```
min_init_tumor_lod = 1
```

min_tumor_lod: Minimum log odds in the final call of variant in the tumor.

```
min_tumor_lod = 8
```

The somatic variants in TNscope raw VCF file (*SNV.somatic.<CASE_ID>.tnscope.all.vcf.gz*) are filtered out for the genomic regions that are not reliable (eg: centromeric regions, non-chromosome contigs) to enhance the computation time. This WGS interval region file is collected from gatk_bundles gs://gatk-legacy-bundles/b37/wgs_calling_regions.v1.interval_list.

Post-call Quality Filters

Total Depth (DP): Refers to the overall read depth supporting the variant call

```
DP(tumor) >= 10
```

Allelic Depth (AD): Total reads supporting the ALT allele in the tumor sample

```
AD(tumor) > 3
```

Allelic Frequency (AF): Fraction of the reads supporting the alternate allele

```
Minimum AF(tumor) > 0.05
```

```
SUM(QSS)/SUM(AD) >= 20
```

Read Counts: Count of reads in a given (F1R2, F2R1) pair orientation supporting the alternate allele and reference alleles

```
ALT_F1R2 > 0, ALT_F2R1 > 0
REF_F1R2 > 0, REF_F2R1 > 0
```

SOR: Symmetric Odds Ratio of 2x2 contingency table to detect strand bias

```
SOR < 3
```

Post-call Observation database Filters

GNOMADAF_POPMAX: Maximum Allele Frequency across populations

```
GNOMADAF_popmax <= 0.001 (or) GNOMADAF_popmax == "."
```

Normalized base quality scores: The sum of base quality scores for each allele (QSS) is divided by the allelic depth of alt and ref alleles (AD)

```
SWEGENAF <= 0.01  (or) SWEGENAF == "."
```

Frq: Frequency of observation of the variants from normal *Clinical* samples

```
Frq <= 0.01  (or) Frq == "."
```

The variants scored as *PASS* or *triallelic_sites* are included in the final vcf file (*SNV.somatic.<CASE_ID>.tnscope.<research/clinical>.filtered.pass.vcf.gz*).

Attention: BALSAMIC <= v8.2.10 uses GNOMAD_popmax <= 0.005. From Balsamic v9.0.0, this settings is changed to 0.02, to reduce the stringency. BALSAMIC >= v11.0.0 removes unmapped reads from the bam and cram files for all the workflows. BALSAMIC >= v13.0.0 keeps unmapped reads in bam and cram files for all the workflows.

CHAPTER

SEVEN

STRUCTURAL AND COPY NUMBER VARIANTS

Depending on the sequencing type, BALSAMIC is currently running the following structural and copy number variant callers:

Table 1: SV CNV callers

Variant caller	Sequencing type	Analysis type	Somatic/Germline	Variant type
AscatNgs	WGS	tumor-normal	somatic	CNV
CNVkit	TGA, WES	tumor-normal, tumor-only	somatic	CNV
Delly	TGA, WES, WGS	tumor-normal, tumor-only	somatic	SV, CNV
Manta	TGA, WES, WGS	tumor-normal, tumor-only	somatic, germline	SV
TIDDIT	WGS	tumor-normal, tumor-only	somatic	SV
CNVpytor	WGS	tumor-only	somatic	CNV
igh_dux4 (see note below)	WGS	tumor-normal, tumor-only	somatic	SV

Further details about a specific caller can be found in the links for the repositories containing the documentation for SV and CNV callers along with the links for the articles are listed in bioinfo softwares.

Note that igh_dux4 is not a variant caller itself. This is a custom script that uses samtools to detect read pairs supporting IGH::DUX4 rearrangements. In short, the command identifies discordant reads mapping to the IGH region and to either DUX4 or its homologous DUX4-like regions (see references for details). The inclusion of this feature aims to alleviate the failure of callers to detect this rearrangement. It is important to note, however, that the reported breakpoints are fixed to the IGH and DUX4 coordinates and are, therefore, imprecise and uncertain. Therefore, we advise caution when interpreting this information.

It is mandatory to provide the gender of the sample from BALSAMIC version >= 10.0.0 For CNV analysis.

7.1 Pre-merge Filtrations

The copy number variants, identified using ascatNgs and *dellycnv*, are converted to deletion and duplications before they are merged using *SVDB* with *–bnd_distance = 5000* (distance between end points for the variants from different callers) and *–overlap = 0.80* (percentage for overlapping bases for the variants from different callers).

Tumor and normal calls in *TIDDIT* are merged using *SVDB* with *–bnd_distance 500* and *–overlap = 0.80*. Using a custom made script "filter_SVs.py", soft-filters are added to the calls based on the presence of the variant in the normal, with the goal of retaining only somatic variants as PASS.

Manta calls are filtered using bcftools to only keep variants that have evidence from 3 or more reads.

Table 2: SV filters

Variant caller	Filter added	Filter expression
TIDDIT	high_normal_af_fraction	(AF_N_MAX / AF_T_MAX) > 0.25
TIDDIT	max_normal_allele_frequency	AF_N_MAX > 0.25
TIDDIT	normal_variant	AF_T_MAX == 0 and ctg_t == False
TIDDIT	in_normal	ctg_n == True and AF_N_MAX == 0 and AF_T_MAX <= 0.25
Manta	low_pr_sr_count	SUM(FORMAT/PR[0:1]+FORMAT/SR[0:1]) < 4.0
igh_dux4	samtools_igh_dux4	DV < 1

Further information regarding the TIDDIT tumor normal filtration: As translocation variants are represented by 2 BNDs in the VCF which allows for mixed assignment of soft-filters, a requirement for assigning soft-filters to translocations is that neither BND is PASS.

7.2 Post-merge Filtrations

SVDB prioritizes the merging of variants from SV and CNV callers to fetch position and genotype information, in the following order:

Table 3: SVDB merge caller priority order

TGA, WES tumor-normal	**TGA, WES** tumor-only	**WGS** tumor-normal	**WGS** tumor-only
1. manta	1. manta	1. manta	1. manta
2. dellysv	2. dellysv	2. dellysv	2. dellysv
3. cnvkit	3. cnvkit	3. ascat	3. dellycnv
4. dellycnv	4. dellycnv	4. dellycnv	4. tiddit
		5. tiddit	5. cnvpytor
		6. igh_dux4	6. igh_dux4

The merged *SNV.somatic.<CASE_ID>.svdb.vcf.gz* file retains all the information for the variants from the caller in which the variants are identified, which are then annotated using *ensembl-vep*. The SweGen and frequencies and the frequency of observed structural variants from clinical normal samples are annotated using *SVDB*.

The following filter applies for both tumor-normal and tumor-only samples in addition to caller specific filters.

SWEGENAF: SweGen Allele Frequency

```
SWEGENAF <= 0.02  (or) SWEGENAF == "."
```

Frq: Frequency of observation of the variants from normal *Clinical* samples

```
Frq <= 0.02  (or) Frq == "."
```

The variants scored as *PASS* are included in the final vcf file (*SNV.somatic.<CASE_ID>.svdb.<research/clinical>.filtered.pass.vcf.gz*).

The following command can be used to fetch the variants identified by a specific caller from merged structural and copy number variants.

```
zgrep -E "#|<Caller>" <*.svdb.vcf.gz>
```

7.3 Using GENS for WGS

GENS is a visualization tool similar to IGV, originally developed in Clinical Genomics Lund, and primarily for visualizing genomic copy number profiles from WGS samples.

To visualise the GENS-formatted files from BALSAMIC you need to have GENS installed, and to do this you can follow the instructions on the Clinical-Genomics-Lund GENS-repository:

- Clinical-Genomics-Lund-GENS

Two files per sample are uploaded to GENS, one file with allele-frequencies from SNV & InDel germline-calls (BAF file) which can be used to aid the interpretation of the CN-profile, and one file with the Log2 copy number ratios normalized against a PON. Instructions for how to generate this PON using the BALSAMIC PON workflow can be found here:

Generate GENS PON.

There are three required arguments for creating the input files for GENS: 1. Genome interval file produced by GATK `PreprocessIntervals` (see instructions in GENS PON creation) 2. A gender specific PON (see instructions in GENS PON creation) 3. A population database VCF with variant positions to be reported in the BAF file.

We created the file mentioned in **3** using the file `gnomad.genomes.r2.1.1.sites` filtered with bcftools to only keep variants with an AF above 0.05.

```
bcftools view -i AF>=0.05 -Oz
```

To config BALSAMIC to run with GENS activated you supply these files like this:

```
balsamic config case \
  --case-id <CASE_ID>
  --balsamic-cache </path/reference_cache/>
  --analysis-dir </path/analysis/>
  --fastq-path </path/fastq/>
  --gender <[male/female]>
  --analysis-workflow balsamic
  --genome-version hg19
  --tumor-sample-name <TUMOR_NAME>
  --genome-interval </path/genome_interval>
```

(continues on next page)

(continued from previous page)

```
  --gens-coverage-pon </path/pon_file>
  --gnomad-min-af5 </path/population_vcf.vcf.gz>
```

7.4 Genome Reference Files

How to generate genome reference files for ascatNGS

Detailed information is available from ascatNGS documentation

The file *SnpGcCorrections.tsv* prepared from the 1000 genome SNP panel.

GC correction file:

First step is to download the 1000 genome snp file and convert it from .vcf to .tsv. The detailed procedure to for this step is available from ascatNGS-reference-files (Human reference files from 1000 genomes VCFs)

```
export TG_DATA=ftp://ftp.ensembl.org/pub/grch37/release-83/variation/vcf/homo_sapiens/
↪1000GENOMES-phase_3.vcf.gz
```

Followed by:

```
curl -sSL $TG_DATA | zgrep -F 'E_Multiple_observations' | grep -F 'TSA=SNV' |\
perl -ane 'next if($F[0] !~ m/^\d+$/ && $F[0] !~ m/^[XY]$/);\
next if($F[0] eq $l_c && $F[1]-1000 < $l_p); $F[7]=~m/MAF=([^;]+)/;\
next if($1 < 0.05); printf "%s\t%s\t%d\n", $F[2],$F[0],$F[1];\
$l_c=$F[0]; $l_p=$F[1];' > SnpPositions_GRCh37_1000g.tsv
```

–or–

```
curl -sSL $TG_DATA | zgrep -F 'E_Multiple_observations' | grep -F 'TSA=SNV' |\
perl -ane 'next if($F[0] !~ m/^\d+$/ && $F[0] !~ m/^[XY]$/); $F[7]=~m/MAF=([^;]+)/;\
next if($1 < 0.05); next if($F[0] eq $l_c && $F[1]-1000 < $l_p);\
printf "%s\t%s\t%d\n", $F[2],$F[0],$F[1]; $l_c=$F[0]; $l_p=$F[1];'\
> SnpPositions_GRCh37_1000g.tsv
```

Second step is to use *SnpPositions.tsv* file and generate *SnpGcCorrections.tsv* file, more details see ascatNGS-convert-snppositions

```
ascatSnpPanelGcCorrections.pl genome.fa SnpPositions.tsv > SnpGcCorrections.tsv
```

CHAPTER

EIGHT

ANNOTATION RESOURCES

BALSAMIC annotates somatic single nucleotide variants (SNVs) using `ensembl-vep` and `vcfanno`. Somatic structural variants (SVs), somatic copy-number variants (CNVs) and germline single nucleotide variants are annotated using only `ensembl-vep`. All SVs and CNVs are merged using `SVDB` before annotating for *Target Genome Analysis (TGA)* or *Whole Genome Sequencing (WGS)* analyses.

8.1 gnomAD

BALSAMIC adds the following annotation from *gnomAD* database using `vcfanno`.

Table 1: gnomAD annotations

VCF tag	description
GNOMADAF_popmax	maximum allele frequency across populations
GNOMADAF	fraction of the reads supporting the alternate allele, allelic frequency

8.2 ClinVar

BALSAMIC adds the following annotation from *ClinVar* database using `vcfanno`.

Table 2: ClinVar annotations

VCF tag	description
CLNACC	Variant Accession and Versions
CLNREVSTAT	ClinVar review status for the Variation ID
CLNSIG	Clinical significance for this single variant
CLNVCSO	Sequence Ontology id for variant type
CLNVC	Variant type
ORIGIN	Allele origin

The values for *ORIGIN* are described below:

Table 3: ClinVar ORIGIN

Value	Annotation
0	unknown
1	germline
2	somatic
4	inherited
8	paternal
16	maternal
32	*de-novo*
64	biparental
128	uniparental
256	not-tested
512	tested-inconclusive
1073741824	other

8.3 COSMIC

BALSAMIC uses *ensembl-vep* to add the following annotation from *COSMIC* database.

Table 4: COSMIC annotations

VCF tag	description
COSMIC_CDS	CDS annotation
COSMIC_GENE	gene name
COSMIC_STRAND	strand
COSMIC_CNT	number of samples with this mutation in the *COSMIC* database
COSMIC_AA	peptide annotation

8.4 CADD

BALSAMIC adds the following annotation for SNVs from *CADD* database using `vcfanno`.

Table 5: CADD annotations

VCF tag	description
CADD	Combined Annotation Dependent Depletion

8.5 LoqusDB somatic frequencies (cancer cases)

Table 6: LoqusDB Somatic Annotations

VCF tag	description	variant type
Cancer_Somatic_Frq	Frequency of observation for somatic mutations	SNV, SV
Cancer_Somatic_Obs	allele counts of the somatic variant	SNV, SV
Cancer_Somatic_Hom	allele counts of the homozygous somatic variant	SNV

8.6 LoqusDB germline frequencies (cancer cases)

Table 7: loqusDB germline SNV annotations

VCF tag	description	variant type
Cancer_Germline_Frq	Frequency of observation for germline mutations	SNV
Cancer_Germline_Obs	allele counts of the germline variant	SNV
Cancer_Germline_Hom	allele counts of the homozygous germline variant	SNV

8.7 LoqusDB germline frequencies (non-cancer cases)

BALSAMIC adds the following annotation from database of *non-cancer clinical* samples using `vcfanno` for SNVs and SVDB for SVs.

Table 8: loqusDB germline (non-cancer) SNV annotations

VCF tag	description	variant type
Frq	Frequency of observation of the variants from normal *non-cancer clinical* samples	SNV, SV
Obs	allele counts of the variant in normal *non-cancer clinical* samples	SNV
Hom	allele counts of the homozygous variant in normal *non-cancer clinical* samples	SNV
clin_obs	allele counts	SV

8.8 SWEGEN

BALSAMIC adds the following annotation from *SWEGEN* database using `vcfanno` for SNVs and SVDB for SVs.

Table 9: Swegen SNV annotations

VCF tag	description	variant type
SWEGENAF	allele frequency from 1000 Swedish genomes project	SNV, SV
SWEGE-NAAC_Hom	allele counts of homozygous variants	SNV
SWEGENAAC_Het	allele counts of heterozygous variants	SNV
SWEGE-NAAC_Hemi	allele counts of hemizygous variants	SNV
swegen_obs	allele count	SV

8.9 ENSEMBL-VEP annotations

Where relevant, *BALSAMIC* uses *ensembl-vep* to annotate somatic and germline SNVs and somatic SVs/CNVs from *1000genomes (phase3)*, *ClinVar*, *ESP, HGMD-PUBLIC*, *dbSNP*, *gencode*, *gnomAD*, *polyphen*, *refseq*, and *sift* databases. The following annotations are added by *ensembl-vep*.

VEP has a setting for the maximum size of a structural variant that it will annotate, currently this is set to the size of the size of chromosome 1 (in hg19) (*–max_sv_size 249250621*).

Table 10: ensembl-vep

Annotation	description
Allele	the variant allele used to calculate the consequence
Gene	Ensembl stable ID of affected gene
Feature	Ensembl stable ID of feature
Feature type	type of feature. Currently one of Transcript, RegulatoryFeature, MotifFeature.
Consequence	consequence type of this variant
Position in cDNA	relative position of base pair in cDNA sequence
Position in CDS	relative position of base pair in coding sequence
Position in protein	relative position of amino acid in protein
Amino acid change	only given if the variant affects the protein-coding sequence
Codon change	the alternative codons with the variant base in upper case
Co-located variation	identifier of any existing variants
VARI-ANT_CLASS	Sequence Ontology variant class
SYMBOL	the gene symbol
SYM-BOL_SOURCE	the source of the gene symbol
STRAND	the DNA strand (1 or -1) on which the transcript/feature lies

continues on next page

Table 10 – continued from previous page

Annotation	description
ENSP	the Ensembl protein identifier of the affected transcript
FLAGS	transcript quality flags: cds_start_NF: CDS 5' incomplete cds_end_NF: CDS 3' incomplete
SWISSPROT	Best match UniProtKB/Swiss-Prot accession of protein product
TREMBL	Best match UniProtKB/TrEMBL accession of protein product
UNIPARC	Best match UniParc accession of protein product
HGVSc	the HGVS coding sequence name
HGVSp	the HGVS protein sequence name
HGVSg	the HGVS genomic sequence name
HGVS_OFFSE	Indicates by how many bases the HGVS notations for this variant have been shifted
SIFT	the SIFT prediction and/or score, with both given as prediction(score)
PolyPhen	the PolyPhen prediction and/or score
MOTIF_NAME	The source and identifier of a transcription factor binding profile aligned at this position
MOTIF_POS	The relative position of the variation in the aligned TFBP
HIGH_INF_PC	A flag indicating if the variant falls in a high information position of a transcription factor binding profile (TFBP)
MOTIF_SCORE_C	The difference in motif score of the reference and variant sequences for the TFBP
CANONICAL	a flag indicating if the transcript is denoted as the canonical transcript for this gene
CCDS	the CCDS identifer for this transcript, where applicable
INTRON	the intron number (out of total number)
EXON	the exon number (out of total number)
DOMAINS	the source and identifer of any overlapping protein domains
DISTANCE	Shortest distance from variant to transcript
AF	Frequency of existing variant in 1000 Genomes
AFR_AF	Frequency of existing variant in 1000 Genomes combined African population
AMR_AF	Frequency of existing variant in 1000 Genomes combined American population
EUR_AF	Frequency of existing variant in 1000 Genomes combined European population
EAS_AF	Frequency of existing variant in 1000 Genomes combined East Asian population
SAS_AF	Frequency of existing variant in 1000 Genomes combined South Asian population
AA_AF	Frequency of existing variant in NHLBI-ESP African American population
EA_AF	Frequency of existing variant in NHLBI-ESP European American population
gnomAD_AF	Frequency of existing variant in gnomAD exomes combined population
gnomAD_AFR_AI	Frequency of existing variant in gnomAD exomes African/American population
gnomAD_AMR_A	Frequency of existing variant in gnomAD exomes American population
gnomAD_ASJ_AF	Frequency of existing variant in gnomAD exomes Ashkenazi Jewish population
gnomAD_EAS_AI	Frequency of existing variant in gnomAD exomes East Asian population
gnomAD_FIN_AF	Frequency of existing variant in gnomAD exomes Finnish population
gnomAD_NFE_AI	Frequency of existing variant in gnomAD exomes Non-Finnish European population

continues on next page

Table 10 – continued from previous page

Annotation	description
gnomAD_OTH_A	Frequency of existing variant in gnomAD exomes combined other combined populations
gnomAD_SAS_AF	Frequency of existing variant in gnomAD exomes South Asian population
MAX_AF	Maximum observed allele frequency in 1000 Genomes, ESP and gnomAD
MAX_AF_POI	Populations in which maximum allele frequency was observed
CLIN_SIG	ClinVar clinical significance of the dbSNP variant
BIOTYPE	Biotype of transcript or regulatory feature
APPRIS	Annotates alternatively spliced transcripts as primary or alternate based on a range of computational methods. NB: not available for GRCh37
TSL	Transcript support level. NB: not available for GRCh37
PUBMED	Pubmed ID(s) of publications that cite existing variant
SOMATIC	Somatic status of existing variant(s); multiple values correspond to multiple values in the Existing_variation field
PHENO	Indicates if existing variant is associated with a phenotype, disease or trait; multiple values correspond to multiple values in the Existing_variation field
GENE_PHENO	Indicates if overlapped gene is associated with a phenotype, disease or trait
BAM_EDIT	Indicates success or failure of edit using BAM file
GIVEN_REF	Reference allele from input

continues on next page

Table 10 – continued from previous page

Annotation	description
REF-SEQ_MATCH	the RefSeq transcript match status; contains a number of flags indicating whether this RefSeq transcript matches the underlying reference sequence and/or an Ensembl transcript (more information): • rseq_3p_mismatch: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. Specifically, there is a mismatch in the 3' UTR of the RefSeq model with respect to the primary genome assembly (e.g. GRCh37/GRCh38). • rseq_5p_mismatch: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. Specifically, there is a mismatch in the 5' UTR of the RefSeq model with respect to the primary genome assembly. • rseq_cds_mismatch: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. Specifically, there is a mismatch in the CDS of the RefSeq model with respect to the primary genome assembly. • rseq_ens_match_cds: signifies that for the RefSeq transcript there is an overlapping Ensembl model that is identical across the CDS region only. A CDS match is defined as follows: the CDS and peptide sequences are identical and the genomic coordinates of every translatable exon match. Useful related attributes are: rseq_ens_match_wt and rseq_ens_no_match. • rseq_ens_match_wt: signifies that for the RefSeq transcript there is an overlapping Ensembl model that is identical across the whole transcript. A whole transcript match is defined as follows: 1) In the case that both models are coding, the transcript, CDS and peptide sequences are all identical and the genomic coordinates of every exon match. 2) In the case that both transcripts are non-coding the transcript sequences and the genomic coordinates of every exon are identical. No comparison is made between a coding and a non-coding transcript. Useful related attributes are: rseq_ens_match_cds and rseq_ens_no_match. • rseq_ens_no_match: signifies that for the RefSeq transcript there is no overlapping Ensembl model that is identical across either the whole transcript or the CDS. This is caused by differences between the transcript, CDS or peptide sequences or between the exon genomic coordinates. Useful related attributes are: rseq_ens_match_wt and rseq_ens_match_cds. • rseq_mrna_match: signifies an exact match between the RefSeq transcript and the underlying primary genome assembly sequence (based on a match between the transcript stable id and an accession in the RefSeq mRNA file). An exact match occurs when the underlying genomic sequence of the model can be perfectly aligned to the mRNA sequence post polyA clipping. • rseq_mrna_nonmatch: signifies a non-match between the RefSeq transcript and the underlying primary genome assembly sequence. A non-match is deemed to have occurred if the underlying genomic sequence does not have a perfect alignment to the mRNA sequence post polyA clipping. It can also signify that no comparison was possible as the model stable id may not have had a corresponding entry in the RefSeq mRNA file (sometimes happens when accessions are retired or changed). When a non-match occurs one or several of the following transcript attributes will also be present to provide more detail on the nature of the non-match: rseq_5p_mismatch, rseq_cds_mismatch, rseq_3p_mismatch, rseq_nctran_mismatch, rseq_no_comparison • rseq_nctran_mismatch: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. This is a comparison between the entire underlying genomic sequence of the RefSeq model to the mRNA in the case of RefSeq models that are non-coding. • rseq_no_comparison: signifies that no alignment was carried out between the underlying primary genome assembly sequence and a corresponding RefSeq mRNA. The reason for this is generally that no corresponding, unversioned accession was found in the RefSeq mRNA file for the transcript stable id. This sometimes happens when accessions are retired or replaced. A second possibility is that the sequences were too long and problematic to align (though this is rare).

continues on next page

Table 10 – continued from previous page

Annotation	description
CHECK_REF	Reports variants where the input reference does not match the expected reference
HGNC_ID	A unique ID provided by the HGNC for each gene with an approved symbol
MANE	indicating if the transcript is the MANE Select or MANE Plus Clinical transcript for the gene.
miRNA	Reports where the variant lies in the miRNA secondary structure.

CHAPTER

NINE

PANEL OF NORMALS (PON)

Currently two PON-methods are implemented in BALSAMIC to correct for biases and normalise coverage values:

- For producing more accurate CNV variant-calls using `CNVkit` for TGA cases.
- To produce normalised CN-profiles for WGS cases visualised in `GENS`.

CHAPTER

TEN

CNVKIT PON

BALSAMIC provides a functionality to generate a Panel of Normals (PON) for more accurate copy-number filtering of false positives and that can be used as an input for the `CNVkit` variant caller.

For a more detailed PON use case, please refer to the following documentation:

- CNVkit
- Illumina DRAGEN
- GATK

10.1 PON Generation

When creating a new PON reference file, the next steps have to be followed:

1. Identify the samples to be included in the PON and add their `fastq` files to the `fastq` directory

Note: One needs to fetch normal samples coming from the same origin, tissue or blood

2. Generate the <CASE_ID>_PON.json configuration file:

```
balsamic config pon --pon-workflow CNVkit --case-id <CASE_ID> --balsamic-cache </path/
↪reference_cache/> --analysis-dir </path/analysis/> --fastq-path </path/fastq/> --panel-
↪bed </path/panel.bed>
```

3. Run the BALSAMIC PON workflow:

```
balsamic run analysis -s </path/analysis/<CASE_ID>/<CASE_ID>_PON.json -r
```

4. Check for the PON reference finish and output files:

```
/path/analysis/analysis_PON_finish
/path/analysis/cnv/<panel_name>_CNVkit_PON_reference_<version>.cnn
```

10.2 Using the PON during analysis

BALSAMIC can use a PON reference file if its provided while running CNVkit analysis:

```
balsamic config case --case-id <CASE_ID> --pon-cnn /path/analysis/cnv/<panel_name>_
↪CNVkit_PON_reference_<version>.cnn --balsamic-cache </path/reference_cache/> --
↪analysis-dir </path/analysis/> --panel-bed </path/panel.bed> --tumor-path </path/tumor.
↪fastq>
```

Note: In the absence of a PON reference file, CNVkit is capable of generating a flat reference (tumor-only) or normal reference (tumor-normal) file on its own to correct for GC content and regional coverage

CHAPTER

ELEVEN

GENS PON

In order to produce an accurate CN-profile to visualise in GENS you need to create 2 PONs one for each gender (see instructions below).

The original instructions for how to create this PON, and which has been implemented in this BALSAMIC workflow can be found on the Clinical-Genomics-Lund GENS-repository:

- Clinical-Genomics-Lund-GENS

To create the PON using the GENS PON creation workflow you can follow the guide below.

11.1 PON Generation

To create a GENS PON using the BALSAMIC workflow you need to follow these steps:

1. Create a genome-interval file.

Note:

These are the genome bins within which the coverage will be calculated, and consequently is the lowest resolution of viewing the CN-profile.

This is the setting we used:

```
gatk PreprocessIntervals --reference [ref] --bin-length 100 --interval-merging-rule␣
↪OVERLAPPING_ONLY -O human_g1k_v37_gens_targets_preprocessed_100bp.interval_list
```

2. Identify the samples to be included in the PON and add or link their `fastq` files to the `fastq` directory

Note:

It is recommended to include approximately 100 samples of the same gender, using the same library preparation and sequencing method as your intended analysis-samples.

2. Generate the `<CASE_ID>_PON.json` configuration file:

```
balsamic config pon --pon-creation-type <[GENS_female,GENS_male]> --genome-interval
↪<[path-to-file-from-step1]> --case-id <CASE_ID> --balsamic-cache </path/reference_
↪cache/> --analysis-dir </path/analysis/> --fastq-path </path/fastq/> --panel-bed </
↪path/panel.bed>
```

3. Run the BALSAMIC PON workflow:

Note:

If you are following these instructions using 100 WGS samples, you require access to compute-nodes with a lot of memory (one of our jobs crashed at 117GB).

```
balsamic run analysis -s </path/analysis/<CASE_ID>/<CASE_ID>_PON.json -r
```

This workflow runs trimming and alignment for all samples to be included in the PON. Calculates coverages in bins using `GATK CollectReadCounts` then creates the PON using all read-counts with the tool `GATK CreateReadCountPanelOfNormals`.

4. Check for the PON output files:

```
/path/analysis/analysis_PON_finish
/path/analysis/cnv/gens_pon_100bp.<GENDER>.<VERSION>.hdf5
```

11.2 Using the PON during analysis

This PON is a required input in order to produce the final output-files to be loaded into the GENS platform.

How to run a case using this PON and to activate GENS for your WGS analysis you are referred to this page:

Using GENS for WGS.

CHAPTER

TWELVE

METHOD DESCRIPTION

12.1 Target Genome Analysis

BALSAMIC [1] (**version** = 15.0.0) was used to analyze the data from raw FASTQ files. We first quality controlled FASTQ files using FastQC v0.11.9 [2]. Adapter sequences and low-quality bases were trimmed using fastp v0.23.2 [3]. Trimmed reads were mapped to the reference genome hg19 using sentieon-tools 202010.02 [15]. Duplicated reads were marked using Dedup from sentieon-tools 202010.02 [15]. The final BAM is promptly quality controlled using CollectHsMetrics, CollectInsertSizeMetrics and CollectAlignmentSummaryMetrics functionalities from Picard tools v2.27.1 [6]. Results of the quality controlled steps were summarized by MultiQC v1.12 [7]. Small somatic mutations (SNVs and INDELs) were called for each sample using VarDict v1.8.2 [8]. Apart from the Vardict filters to report the variants, the called-variants were also further second filtered using the criteria (*MQ >= 40, DP >= 100, VD >= 5, Minimum AF >= 0.007, Maximum AF < 1, GNOMADAF_popmax <= 0.005, swegen AF < 0.01*). Only those variants that fulfilled the filtering criteria and scored as *PASS* in the VCF file were reported. Structural variants (SV) were called using Manta v1.6.0 [9] and Dellyv1.0.3 [10]. Copy number variations (CNV) were called using CNVkit v0.9.9 [11]. The variant calls from CNVkit, Manta and Delly were merged using SVDB v2.8.1 [12]. The clinical set of SNV and SV is also annotated and filtered against loqusDB curated frequency of observed variants (frequency < 0.01) from non-cancer cases and only annotated using frequency of observed variants from cancer cases (somatic and germline). All variants were annotated using Ensembl VEP v104.3 [13]. We used vcfanno v0.3.3 [14] to annotate somatic variants for their population allele frequency from gnomAD v2.1.1 [18], CADD v1.6 [24], SweGen [22] and frequency of observed variants in normal samples.

12.2 Whole Genome Analysis

BALSAMIC [1] (**version** = 15.0.0) was used to analyze the data from raw FASTQ files. We first quality controlled FASTQ files using FastQC v0.11.9 [2]. Adapter sequences and low-quality bases were trimmed using fastp v0.23.2 [3]. Trimmed reads were mapped to the reference genome hg19 using sentieon-tools 202010.02 [15]. Duplicated reads were marked using Dedup from sentieon-tools 202010.02 [15]. The BAM file was then realigned using Realign from sentieon-tools 202010.02 [15] and common population InDels. The final BAM is quality controlled using WgsMetricsAlgo and CoverageMetrics from sentieon-tools 202010.02 [15] and CollectWgsMetrics, CollectMultipleMetrics, CollectGcBias-Metrics, and CollectHsMetrics functionalities from Picard tools v2.27.1 [6]. Results of the quality controlled steps were summarized by MultiQC v1.12 [7]. Small somatic mutations (SNVs and INDELs) were called for each sample using Sentieon TNscope [16]. The called-variants were also further second filtered using the criteria (DP(tumor,normal) >= 10; AD(tumor) >= 3; AF(tumor) >= 0.05, Maximum AF(tumor < 1; GNOMADAF_popmax <= 0.001; normalized base quality scores >= 20, read_counts of alt,ref alle > 0). Structural variants were called using Manta v1.6.0 [9], Delly v1.0.3 [10] and TIDDIT v3.3.2 [12]. Copy number variations (CNV) were called using ascatNgs v4.5.0 [17] (tumor-normal), Delly v1.0.3 [10] and CNVpytor v1.3.1 [22] (tumor-only) and converted from CNV to deletions (DEL) and duplications (DUP). The structural variant (SV) calls from Manta, Delly, TIDDIT, ascatNgs (tumor-normal) and CNVpytor (tumor-only) were merged using SVDB v2.8.1 [12] The clinical set of SNV and SV is also annotated and filtered against loqusDB curated frequency of observed variants (frequency < 0.01) from non-cancer cases and only annotated using frequency of observed variants from cancer cases (somatic and germline). All variants were annotated using Ensembl VEP v104.3

[13]. We used vcfanno v0.3.3 [14] to annotate somatic single nucleotide variants for their population allele frequency from gnomAD v2.1.1 [18], CADD v1.6 [24], SweGen [22] and frequency of observed variants in normal samples.

12.3 UMI Data Analysis

BALSAMIC [1] (**version** = 15.0.0) was used to analyze the data from raw FASTQ files. We first quality controlled FASTQ files using FastQC v0.11.9 [2]. UMI tag extraction and consensus generation were performed using Sentieon tools v202010.02 [15]. Adapter sequences and low-quality bases were trimmed using fastp v0.23.2 [3]. The alignment of UMI extracted and consensus called reads to the human reference genome (hg19) was done by bwa-mem and samtools using Sentieon utils. Consensus reads were filtered based on the number of minimum reads supporting each UMI tag group. We applied a criteria filter of minimum reads *3,1,1*. It means that at least three UMI tag groups should be ideally considered from both DNA strands, where a minimum of at least one UMI tag group should exist in each single-stranded consensus read. The filtered consensus reads were quality controlled using Picard CollectHsMetrics v2.27.1 [5]. Results of the quality controlled steps were summarized by MultiQC v1.12 [6]. For each sample, somatic mutations were called using Sentieon TNscope [16], with non-default parameters for passing the final list of variants (–min_tumor_allele_frac 0.0005, –filter_t_alt_frac 0.0005, –min_init_tumor_lod 0.5, min_tumor_lod 4, –max_error_per_read 5 –pcr_indel_model NONE, GNOMADAF_popmax <= 0.02). The clinical set of SNV and SV is also annotated and filtered against loqusDB curated frequency of observed variants (frequency < 0.01) from non-cancer cases and only annotated using frequency of observed variants from cancer cases (somatic and germline). All variants were annotated using Ensembl VEP v104.3 [7]. We used vcfanno v0.3.3 [8] to annotate somatic variants for their population allele frequency from gnomAD v2.1.1 [18], CADD v1.6 [24], SweGen [22] and frequency of observed variants in normal samples. For exact parameters used for each software, please refer to https://github.com/Clinical-Genomics/BALSAMIC. We used three commercially available products from SeraCare [Material numbers: 0710-067110 [19], 0710-067211 [20], 0710-067312 [21]] for validating the efficiency of the UMI workflow in identifying 14 mutation sites at known allelic frequencies.

12.4 References

1. Foroughi-Asl, H., Jeggari, A., Maqbool, K., Ivanchuk, V., Elhami, K., & Wirta, V. BALSAMIC: Bioinformatic Analysis pipeLine for SomAtic MutatIons in Cancer (Version v8.2.10) [Computer software]. https://github.com/Clinical-Genomics/BALSAMIC
2. Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. Accessed June 22, 2020. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
3. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics. 2018;34(17):i884-i890. https://doi.org/10.1093/bioinformatics/bty560
4. Li H. (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. https://doi.org/10.48550/arXiv.1303.3997
5. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J., Homer N., Marth G., Abecasis G., Durbin R. and 1000 Genome Project Data Processing Subgroup (2009) The Sequence alignment/map (SAM) format and SAMtools. Bioinformatics, 25, 2078-9. https://doi.org/10.1093/bioinformatics/btp352
6. Picard Tools - By Broad Institute. Accessed June 22, 2020. https://broadinstitute.github.io/picard/
7. Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics. 2016;32(19):3047-3048. https://doi.org/10.1093/bioinformatics/btw354
8. Lai Z, Markovets A, Ahdesmaki M, Chapman B, Hofmann O, McEwen R, Johnson J, Dougherty B, Barrett JC, and Dry JR. VarDict: a novel and versatile variant caller for next-generation sequencing in cancer research. Nucleic Acids Res. 2016. https://doi.org/10.1093/nar/gkw227

9. Chen, X. et al. (2016) Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics, 32, 1220-1222. https://doi.org/10.1093/bioinformatics/btv710

10. Tobias Rausch, Thomas Zichner, Andreas Schlattl, Adrian M. Stuetz, Vladimir Benes, Jan O. Korbel. DELLY: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics. 2012 Sep 15;28(18):i333-i339. https://doi.org/10.1093/bioinformatics/bts378

11. Talevich, E, Shain, A.H, Botton, T, & Bastian, B.C. CNVkit: Genome-wide copy number detection and visualization from targeted sequencing. PLOS Computational Biology. 2016, 12(4):e1004873. https://doi.org/10.1371/journal.pcbi.1004873

12. Jesper Eisfeldt et.al. TIDDIT, an efficient and comprehensive structural variant caller for massive parallel sequencing data. F1000 research. 2017. https://doi.org/10.12688/f1000research.11168.2

13. McLaren W, Gil L, Hunt SE, et al. The Ensembl Variant Effect Predictor. Genome Biology. 2016;17(1):122. https://doi.org/10.1186/s13059-016-0974-4

14. Pedersen BS, Layer RM, Quinlan AR. Vcfanno: fast, flexible annotation of genetic variants. Genome Biology. 2016;17(1):118. https://doi.org/10.1186/s13059-016-0973-5

15. Donald Freed, Rafael Aldana, Jessica A. Weber, Jeremy S. Edwards. The Sentieon Genomics Tools - A fast and accurate solution to variant calling from next-generation sequence data. Bioinformatics. 2016, Volume 32,Issue 8. https://doi.org/10.1093/bioinformatics/btv710

16. Donald Freed, Renke Pan, Rafael Aldana. TNscope: Accurate Detection of Somatic Mutations with Haplotype-based Variant Candidate Detection and Machine Learning Filtering. bioRvix. https://doi.org/10.1101/250647

17. Keiran MR, Peter VL, David CW, David J, Andrew M, Adam PB , Jon WT, Patrick T, Serena Nik-Zainal, Peter J C. ascatNgs: Identifying Somatically Acquired Copy-Number Alterations from Whole-Genome Sequencing Data. Curr Protoc Bioinformatics. 2016. https://doi.org/10.1002/cpbi.17

18. Karczewski, K.J., Francioli, L.C., Tiao, G. et al. The mutational constraint spectrum quantified from variation in 141,456 humans. Nature 581, 434–443 (2020). https://doi.org/10.1038/s41586-020-2308-7

19. Seraseq ctDNA Complete Reference Material AF 1%. https://www.seracare.com/Seraseq-ctDNA-Complete-Reference-Material-AF1-0710-0671/

20. Seraseq ctDNA Complete Reference Material AF 0.5%. https://www.seracare.com/Seraseq-ctDNA-Complete-Reference-Material-AF05-0710-0672/

21. Seraseq ctDNA Complete Reference Material AF 0.1%. https://www.seracare.com/Seraseq-ctDNA-Complete-Reference-Material-AF01-0710-0673/

22. Ameur, A., Dahlberg, J., Olason, P. et al. SweGen: a whole-genome data resource of genetic variability in a cross-section of the Swedish population. Eur J Hum Genet 25, 1253–1260 (2017). https://doi.org/10.1038/ejhg.2017.130

23. Milovan Suvakov, Arijit Panda, Colin Diesh, Ian Holmes, Alexej Abyzov, CNVpytor: a tool for copy number variation detection and analysis from read depth and allele imbalance in whole-genome sequencing, GigaScience, Volume 10, Issue 11, November 2021, giab074, https://doi.org/10.1093/gigascience/giab074

24. Rentzsch P., Witten D., Cooper G.M., Shendure J., Kircher M. CADD: predicting the deleteriousness of variants throughout the human genome. Nucleic Acids Res. 2018. https://doi.org/10.1093/nar/gky1016. PubMed PMID: 30371827.

CHAPTER

THIRTEEN

CHANGELOG

13.1 [15.0.0]

13.1.1 Added:

- high_normal_tumor_af_frac filter in bcftools for TNscope T+N filtering out more than 30% TINC https://github.com/Clinical-Genomics/BALSAMIC/pull/1289
- New option for exome samples *–exome* with modified bcftools filters compared to standard targeted workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1414
- Custom samtools script for the detection of IGH::DUX4 rearrangements https://github.com/Clinical-Genomics/BALSAMIC/pull/1397

13.1.2 Changed:

- Reduced stringency of minimum MQ for all TGA to 30 from 40 https://github.com/Clinical-Genomics/BALSAMIC/pull/1414
- Removed -u flag from VarDict T+N and T only rules to remove calling only in reverse reads of overlapping mates https://github.com/Clinical-Genomics/BALSAMIC/pull/1414
- Removed -U flag to VarDict T+N rule to start calling SVs https://github.com/Clinical-Genomics/BALSAMIC/pull/1414

13.1.3 Removed:

- alt_allele_in_normal filter from TNscope T+N workflows https://github.com/Clinical-Genomics/BALSAMIC/pull/1289

13.1.4 Fixed:

- initial filter keeping only PASS or triallelic-site from T+N bcftools quality filter rule has been removed https://github.com/Clinical-Genomics/BALSAMIC/pull/1424

13.2 [14.0.1]

13.2.1 Fixed:

- PureCN fail due to bash strict mode https://github.com/Clinical-Genomics/BALSAMIC/pull/1406
- Corrected name of CNVkit container in the CNVkit PON creation workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1412

13.3 [14.0.0]

13.3.1 Added:

- bcftools filters for *PR:SR* evidence in Manta calls https://github.com/Clinical-Genomics/BALSAMIC/pull/1371
- *–exome* argument to Manta runs in TGA cases https://github.com/Clinical-Genomics/BALSAMIC/pull/1371
- MultiQC intermediate files to deliverables https://github.com/Clinical-Genomics/BALSAMIC/pull/1388

13.3.2 Removed:

- Extra bcftools filters that allows MaxDepth filtered variants in the final SV VCF https://github.com/Clinical-Genomics/BALSAMIC/pull/1371
- Unused arguments from *delivery.py* https://github.com/Clinical-Genomics/BALSAMIC/pull/1388

13.3.3 Fixed:

- ASCAT-Ngs container https://github.com/Clinical-Genomics/BALSAMIC/pull/1395
- bcftools in manta_tumor_normal uses correct column for tumor read filtering https://github.com/Clinical-Genomics/BALSAMIC/pull/1400

13.4 [13.0.1]

13.4.1 Added:

- Sleep rule before start to fix key_error https://github.com/Clinical-Genomics/BALSAMIC/pull/1311

13.4.2 Fixed:

- Missing *__init__.py* in *snakemake_rules* folders https://github.com/Clinical-Genomics/BALSAMIC/pull/1383

13.5 [13.0.0]

13.5.1 Added:

- Fastq concatenation https://github.com/Clinical-Genomics/BALSAMIC/pull/1069
- *CADD* SNV references https://github.com/Clinical-Genomics/BALSAMIC/pull/1126
- *CADD* SNV annotation https://github.com/Clinical-Genomics/BALSAMIC/pull/1150
- Samtools *stats*, *flagstat*, *idxstat* to WGS workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Functionality for dynamically assigning fastq-info to sample dict in config from input fastq-dir https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Annotate SNVs with cancer germline SNV observations from Loqusdb https://github.com/Clinical-Genomics/BALSAMIC/pull/1178
- Annotate SNVs with somatic SNV observations from Loqusdb https://github.com/Clinical-Genomics/BALSAMIC/pull/1187
- Tests for Annotation with Cancer germline, somatic and clinical observations, and swegen frequencies https://github/Clinical-Genomics/BALSAMIC/pull/1190
- Annotate SVs with somatic SV observations from Loqusdb https://github.com/Clinical-Genomics/BALSAMIC/pull/1194
- Support singularity bind paths with different destination directories https://github/Clinical-Genomics/BALSAMIC/pull/1211
- Added *–rerun-trigger mtime* option to Snakemake command https://github.com/Clinical-Genomics/BALSAMIC/pull/1217
- *CADD* container https://github.com/Clinical-Genomics/BALSAMIC/pull/1222
- Container ettiquette to ReadtheDocs https://github.com/Clinical-Genomics/BALSAMIC/pull/1232
- *htslib* (samtools, bcftools tabix) container https://github.com/Clinical-Genomics/BALSAMIC/pull/1234
- Release version support for cache generation https://github.com/Clinical-Genomics/BALSAMIC/pull/1231
- *CADD* scores for INDELs https://github.com/Clinical-Genomics/BALSAMIC/pull/1238
- *CADD* reference to tests https://githuc.com/Clinical-Genomics/BALSAMIC/pull/1241
- Add cache version option to config case https://github.com/Clinical-Genomics/BALSAMIC/pull/1244
- *cnvkit* container https://github.com/Clinical-Genomics/BALSAMIC/pull/1252
- *PureCN* container https://github.com/Clinical-Genomics/BALSAMIC/pull/1255
- *GATK* container https://github.com/Clinical-Genomics/BALSAMIC/pull/1266
- Resolved FASTQ paths to sample dictionary (balsamic logging) https://github.com/Clinical-Genomics/BALSAMIC/pull/1275
- Picard HsMetrics and CollectGcBiasMetrics for WGS https://github.com/Clinical-Genomics/BALSAMIC/pull/1288
- *LOH* to TGA workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1278
- CNVs from PureCN to TGA workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1278
- Command-line arguments and rules for creation of GENS files https://github.com/Clinical-Genomics/BALSAMIC/pull/1279

- Somatic and germline Loqusdb annotation to ReadtheDocs https://github.com/Clinical-Genomics/BALSAMIC/pull/1317
- Postprocess step before VarDict in TGA https://github.com/Clinical-Genomics/BALSAMIC/pull/1332
- CNV report for TGA workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1339
- *wkhtmltopdf* to system requirements https://github.com/Clinical-Genomics/BALSAMIC/pull/1339
- Store WGS CNV report plots https://github.com/Clinical-Genomics/BALSAMIC/pull/1347

13.5.2 Changed:

- Changed CN header field in cnvpytor in cnvpytor_tumor_only to be Float instead of Integer https://github.com/Clinical-Genomics/BALSAMIC/pull/1182
- Changed samples in case_config.json from being a dict to a list of dicts https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Updated snakemake version to 7.25.0 https://github.com/Clinical-Genomics/BALSAMIC/pull/1099
- Updated cryptography version to 41.0.1 https://github.com/Clinical-Genomics/BALSAMIC/pull/1173
- Refactor bam and fastq inputs in snakemake to call pydantic model functions https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Standardised alignment workflows to WGS-workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Implemented parallel trimming and alignment in all workflows per lane https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- All bam-QC tools take the final dedup.realign bamfile as input https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Validation of pydantic models done both during config and run https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Refactored fastp rules, and changed order of UMI-trimming and quality trimming https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Fix pydantic version (<2.0) https://github.com/Clinical-Genomics/BALSAMIC/pull/1191
- Refactor constants https://github.com/Clinical-Genomics/BALSAMIC/pull/1174
- Move models to their own folder https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Balsamic init workflow refactoring https://github.com/Clinical-Genomics/BALSAMIC/pull/1188
- Updated cryptography version to 41.0.2 https://github.com/Clinical-Genomics/BALSAMIC/pull/1205
- Refactor snakemake executable command generation https://github/Clinical-Genomics/BALSAMIC/pull/1211
- Updated Python version to 3.11 and its dependencies https://github.com/Clinical-Genomics/BALSAMIC/pull/1216
- Tools versions in doc https:/github.com/Clinical-Genomics/BALSAMIC/pull/1239
- Reuse common Balsamic CLI options https://github.com/Clinical-Genomics/BALSAMIC/pull/1242
- Update *reference.json* file to use relative paths https://github.com/Clinical-Genomics/BALSAMIC/pull/1251
- Update pydantic to v2 while maintaining support for v1 models https://github.com/Clinical-Genomics/BALSAMIC/pull/1253

- *PCT_PF_READS_IMPROPER_PAIRS* QC threshold lowered to 5% https://github.com/Clinical-Genomics/BALSAMIC/issues/1265
- Migrate Metrics models to pydantic v2 https://github.com/Clinical-Genomics/BALSAMIC/pull/1270
- Migrate Snakemake models to pydantic v2 https://github.com/Clinical-Genomics/BALSAMIC/pull/1268
- Migrate Cache models to pydantic v2 https://github.com/Clinical-Genomics/BALSAMIC/pull/1274
- Made BALSAMIC compatible with multiple PON creation workflows https://github.com/Clinical-Genomics/BALSAMIC/pull/1279
- Use StrEnum from python enum https://github.com/Clinical-Genomics/BALSAMIC/pull/1303
- Renamed final cram bamfile to format *<tumor/normal>.<LIMS_ID>.cram* https://github.com/Clinical-Genomics/BALSAMIC/pull/1307
- Updated snakemake version to 7.32.4 https://github.com/Clinical-Genomics/BALSAMIC/pull/1308
- Migrate analysis models to pydantic v2 https://github.com/Clinical-Genomics/BALSAMIC/pull/1306
- Split analysis model into config and params models https://github.com/Clinical-Genomics/BALSAMIC/pull/1306
- Renamed name in sample column of final clincial vcfs https://github.com/Clinical-Genomics/BALSAMIC/pull/1310
- Update Gens HK tags https://github.com/Clinical-Genomics/BALSAMIC/pull/1319
- Increased memory and threads for VarDict https://github.com/Clinical-Genomics/BALSAMIC/pull/1332
- Updated ReadtheDocs with GENS and structural pipeline changes https://github.com/Clinical-Genomics/BALSAMIC/pull/1327
- Migrate WGS CNV report generation to pypdf & pdfkit https://github.com/Clinical-Genomics/BALSAMIC/pull/1346

13.5.3 Fixed:

- vcf2cytosure container https://github.com/Clinical-Genomics/BALSAMIC/pull/1159
- Link external fastqs to case folder & create case directory https://github.com/Clinical-Genomics/BALSAMIC/pull/1195
- vcf2cytosure container missing constants https://github.com/Clinical-Genomics/BALSAMIC/pull/1198
- Bash commands in vep_somatic_clinical_snv https://github.com/Clinical-Genomics/BALSAMIC/pull/1200
- Fix SVDB annotation intermediate rule https://github.com/Clinical-Genomics/BALSAMIC/pull/1218
- Broken documentation links https://github.com/Clinical-Genomics/BALSAMIC/pull/1226
- Updated contributors in main README https://github.com/Clinical-Genomics/BALSAMIC/pull/1237
- CNVpytor container https://github.com/Clinical-Genomics/BALSAMIC/pull/1246
- Restored balsamic container in UMI concatenation rule https://github.com/Clinical-Genomics/BALSAMIC/pull/1261
- CNVpytor container, fixing numpy version https://github.com/Clinical-Genomics/BALSAMIC/pull/1273
- QC workflow store https://github.com/Clinical-Genomics/BALSAMIC/pull/1295
- MultiQC rule missing input files https://github.com/Clinical-Genomics/BALSAMIC/pull/1321

- *gens_preprocessing* rule missing python directive https://github.com/Clinical-Genomics/BALSAMIC/pull/1322
- CADD annotations container path and code smells https://github.com/Clinical-Genomics/BALSAMIC/pull/1323
- Sonarcloud reported issues https://github.com/Clinical-Genomics/BALSAMIC/pull/1348
- Loqusdb SV annotation somatic fields https://github.com/Clinical-Genomics/BALSAMIC/pull/1354

13.5.4 Removed:

- Config folder https://github.com/Clinical-Genomics/BALSAMIC/pull/1175
- Quality trimming of fastqs for UMI workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1176
- Balsamic container https://github.com/Clinical-Genomics/BALSAMIC/pull/1230
- Plugin CLI https://github.com/Clinical-Genomics/BALSAMIC/pull/1245
- Realignment step for TGA workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1272
- Archived/outdated workflows and scripts https://github.com/Clinical-Genomics/BALSAMIC/pull/1296
- Sed command to convert CNVpytor integer to float, deprecated by updated CNVpytor version https://github.com/Clinical-Genomics/BALSAMIC/pull/1310
- Removed max AF 1 filter from bcftools https://github.com/Clinical-Genomics/BALSAMIC/pull/1338
- Extra samtools sort command from WGS cases https://github.com/Clinical-Genomics/BALSAMIC/pull/1334

13.6 [12.0.2]

13.6.1 Fixed:

- Missing *Number* in VCF header for SVs https://github.com/Clinical-Genomics/BALSAMIC/pull/1203

13.6.2 Changed:

- Fix cyvcf2 to version 0.30.22 https://github.com/Clinical-Genomics/BALSAMIC/pull/1206
- Fix pydantic version (<2.0) https://github.com/Clinical-Genomics/BALSAMIC/pull/1206
- Update varcall-cnvkit container versions https://github.com/Clinical-Genomics/BALSAMIC/pull/1207

13.7 [12.0.1]

13.7.1 Added:

- WGS QC criteria for *PCT_PF_READS_IMPROPER_PAIRS* (condition: <= 0.1) https://github.com/Clinical-Genomics/BALSAMIC/pull/1164

13.7.2 Fixed:

- Logged version of Delly (changing it to v1.0.3) https://github.com/Clinical-Genomics/BALSAMIC/pull/1170

13.8 [12.0.0]

13.8.1 Added:

- PIP specific missing tools to config https://github.com/Clinical-Genomics/BALSAMIC/pull/1096
- Filtering script to remove normal variants from TIDDIT https://github.com/Clinical-Genomics/BALSAMIC/pull/1120
- Store TMB files in HK https://github.com/Clinical-Genomics/BALSAMIC/pull/1144

13.8.2 Changed:

- Fixed all conda container dependencies https://github.com/Clinical-Genomics/BALSAMIC/pull/1096
- Changed –max_sv_size in VEP params to the size of chr1 for hg19 https://github.com/Clinical-Genomics/BALSAMIC/pull/1124
- Increased time-limit for sambamba_exon_depth and picard_markduplicates to 6 hours https://github.com/Clinical-Genomics/BALSAMIC/pull/1143
- Update cosmicdb to v97 https://github.com/Clinical-Genomics/BALSAMIC/pull/1147
- Updated read the docs with the changes relevant to mention https://github.com/Clinical-Genomics/BALSAMIC/pull/1153

13.8.3 Fixed:

- Update cryptography version (39.0.1) due to security alert https://github.com/Clinical-Genomics/BALSAMIC/pull/1087
- Bump cryptography to v40.0.2 and gsutil to v5.23 https://github.com/Clinical-Genomics/BALSAMIC/pull/1154
- Pytest file saved in balsamic directory https://github.com/Clinical-Genomics/BALSAMIC/pull/1093
- Fix varcall_py3 container bcftools dependency error https://github.com/Clinical-Genomics/BALSAMIC/pull/1097
- AscatNgs container https://github.com/Clinical-Genomics/BALSAMIC/pull/1155

13.9 [11.2.0]

13.9.1 Fixed:

- Number of variants are increased with triallelic_site https://github.com/Clinical-Genomics/BALSAMIC/pull/1089

13.10 [11.1.0]

13.10.1 Added:

- Added somalier integration and relatedness check: https://github.com/Clinical-Genomics/BALSAMIC/pull/1017
- Cluster resources for CNVPytor tumor only https://github.com/Clinical-Genomics/BALSAMIC/pull/1083

13.10.2 Changed:

- Parallelize download of reference files https://github.com/Clinical-Genomics/BALSAMIC/pull/1065
- Parallelize download of container images https://github.com/Clinical-Genomics/BALSAMIC/pull/1068

13.10.3 Fixed:

- triallelic_site in quality filter for SNV https://github.com/Clinical-Genomics/BALSAMIC/pull/1052
- Compression of SNV, research and clinical, VCF files https://github.com/Clinical-Genomics/BALSAMIC/pull/1060
- *test_write_json* failing locally https://github.com/Clinical-Genomics/BALSAMIC/pull/1063
- Container build and push via github actions by setting buildx *provenance* flag to false https://github.com/Clinical-Genomics/BALSAMIC/pull/1071
- Added buildx to the submodule workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1072
- Change user in somalier container to defaultuser https://github.com/Clinical-Genomics/BALSAMIC/pull/1080
- Reference files for hg38 https://github.com/Clinical-Genomics/BALSAMIC/pull/1081

13.11 [11.0.2]

13.11.1 Changed:

- Code owners https://github.com/Clinical-Genomics/BALSAMIC/pull/1050

13.11.2 Fixed:

- MaxDepth in quality filter for SV https://github.com/Clinical-Genomics/BALSAMIC/pull/1051

13.12 [11.0.1]

13.12.1 Fixed:

- Incorrect raw *TNscope* VCF delivered https://github.com/Clinical-Genomics/BALSAMIC/pull/1042

13.13 [11.0.0]

13.13.1 Added:

- Use of PON reference, if exists for CNVkit tumor-normal analysis https://github.com/Clinical-Genomics/BALSAMIC/pull/982
- Added PON version to CLI and config.json https://github.com/Clinical-Genomics/BALSAMIC/pull/983
- *cnvpytor* to varcallpy3 container https://github.com/Clinical-Genomics/BALSAMIC/pull/991
- *cnvpytor* for tumor only workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/994
- R packages to cnvkit container https://github.com/Clinical-Genomics/BALSAMIC/pull/996
- Missing R packages to cnvkit container https://github.com/Clinical-Genomics/BALSAMIC/pull/997
- add rlang to cnvkit container https://github.com/Clinical-Genomics/BALSAMIC/pull/998
- AnnotSV and bedtools to annotate container https://github.com/Clinical-Genomics/BALSAMIC/pull/1005
- cosmicdb to TNscope for tumor only and tumor normal workflows https://github.com/Clinical-Genomics/BALSAMIC/pull/1006
- *loqusDB* dump files to the config through the balsamic config case CLI https://github.com/Clinical-Genomics/BALSAMIC/pull/992
- Pre-annotation quality filters for SNVs annd added *research* to output files https://github.com/Clinical-Genomics/BALSAMIC/pull/1007
- Annotation of snv_clinical_observations for somatic snv https://github.com/Clinical-Genomics/BALSAMIC/pull/1012
- Annotation of sv_clinical_observations for somatic sv and SV CNV filter rules https://github.com/Clinical-Genomics/BALSAMIC/pull/1013
- Swegen SNV and SV frequency database for WGS https://github.com/Clinical-Genomics/BALSAMIC/pull/1014
- triallelic_sites and variants with MaxDepth to the VCFs https://github.com/Clinical-Genomics/BALSAMIC/pull/1021
- Clinical VCF for TGA workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1024
- CNVpytor plots into the CNV PDF report https://github.com/Clinical-Genomics/BALSAMIC/pull/1023
- Research and clinical housekeeper tags https://github.com/Clinical-Genomics/BALSAMIC/pull/1023
- Cluster configuration for rules https://github.com/Clinical-Genomics/BALSAMIC/pull/1028
- Variant filteration using loqusDB and Swegen annotations https://github.com/Clinical-Genomics/BALSAMIC/pull/1029
- Annotation resources to readsthedocs https://github.com/Clinical-Genomics/BALSAMIC/pull/1031

- Delly CNV rules for TGA workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/103
- cnvpytor container and removed cnvpytor from varcallpy3 https://github.com/Clinical-Genomics/BALSAMIC/pull/1037

13.13.2 Changed:

- Added version number to the PON reference filename (*.cnn*) https://github.com/Clinical-Genomics/BALSAMIC/pull/982
- Update *TIDDIT* to v3.3.0, *SVDB* to v2.6.4, *delly* to v1.1.3, *vcf2cytosure* to v0.8 https://github.com/Clinical-Genomics/BALSAMIC/pull/987
- toml config file for vcfanno https://github.com/Clinical-Genomics/BALSAMIC/pull/1012
- Split *vep_germline* rule into *tumor* and *normal* https://github.com/Clinical-Genomics/BALSAMIC/pull/1018
- Extract number of variants from clinical files https://github.com/Clinical-Genomics/BALSAMIC/pull/1022

13.13.3 Fixed:

- Reverted *pandas* version (from *1.3.5* to *1.1.5*) https://github.com/Clinical-Genomics/BALSAMIC/pull/1018
- Mate in realigned bam file https://github.com/Clinical-Genomics/BALSAMIC/pull/1019
- samtools command in merge bam and names in toml for vcfanno https://github.com/Clinical-Genomics/BALSAMIC/pull/1020
- If statement in *vep_somatic_clinical_snv* rule https://github.com/Clinical-Genomics/BALSAMIC/pull/1022
- Invalid flag second of pair validation error https://github.com/Clinical-Genomics/BALSAMIC/pull/1025
- Invalid flag second of pair validation error using picardtools https://github.com/Clinical-Genomics/BALSAMIC/pull/1027
- Samtools command for mergetype tumor https://github.com/Clinical-Genomics/BALSAMIC/pull/1030
- *varcall_py3* container building https://github.com/Clinical-Genomics/BALSAMIC/pull/1036
- Picard and fastp commands params and cluster config for umi workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/1032
- Set channels in *varcall_py3* container https://github.com/Clinical-Genomics/BALSAMIC/pull/1035
- Delly command for tumor-normal analysis https://github.com/Clinical-Genomics/BALSAMIC/pull/1039
- tabix command in bcftools_quality_filter_TNscope_umi_tumor_only rule https://github.com/Clinical-Genomics/BALSAMIC/pull/1040

13.13.4 Removed:

- case ID from the PON *.cnn* output file https://github.com/Clinical-Genomics/BALSAMIC/pull/983
- *TNhaplotyper* for paired WGS analysis https://github.com/Clinical-Genomics/BALSAMIC/pull/988
- *TNhaplotyper* for tumor only WGS analysis https://github.com/Clinical-Genomics/BALSAMIC/pull/1006
- *TNhaplotyper* for TGS https://github.com/Clinical-Genomics/BALSAMIC/pull/1022

13.14 [10.0.5]

13.14.1 Changed:

- Update *vcf2cytosure* version to v0.8 https://github.com/Clinical-Genomics/BALSAMIC/pull/1010
- Update GitHub action images to *ubuntu-20.04* https://github.com/Clinical-Genomics/BALSAMIC/pull/1010
- Update GitHub actions to their latest versions https://github.com/Clinical-Genomics/BALSAMIC/pull/1010

13.15 [10.0.4]

13.15.1 Fixed:

- Increase *sambamba_exon_depth* rule run time https://github.com/Clinical-Genomics/BALSAMIC/pull/1001

13.16 [10.0.3]

13.16.1 Fixed:

- Input VCF files for cnvkit rules, cnvkit command and container https://github.com/Clinical-Genomics/BALSAMIC/pull/995

13.17 [10.0.2]

13.17.1 Fixed:

- TIDDIT delivery rule names (undo rule name changes made in Balsamic 10.0.1) https://github.com/Clinical-Genomics/BALSAMIC/pull/977
- BALSAMIC readthedocs CLI documentation generation https://github.com/Clinical-Genomics/BALSAMIC/issues/965

13.18 [10.0.1]

13.18.1 Fixed:

- Command and condition for TIDDIT and fixed ReadtheDocs https://github.com/Clinical-Genomics/BALSAMIC/pull/973
- ReadtheDocs and updated the header https://github.com/Clinical-Genomics/BALSAMIC/pull/973

13.18.2 Changed:

- Time allocation in cluster configuration for SV rules https://github.com/Clinical-Genomics/BALSAMIC/pull/973

13.19 [10.0.0]

13.19.1 Added:

- New option *analysis-workflow* to balsamic config case CLI https://github.com/Clinical-Genomics/BALSAMIC/pull/932
- New python script to edit INFO tags in *vardict* and *tnscope_umi* VCF files https://github.com/Clinical-Genomics/BALSAMIC/pull/948
- Added *cyvcf2* and *click* tools to the *varcallpy3* container https://github.com/Clinical-Genomics/BALSAMIC/pull/948
- Delly TIDDIT and vcf2cytosure for WGS https://github.com/Clinical-Genomics/BALSAMIC/pull/947
- *Delly TIDDIT vcf2cytosure* and method to process SVs and CNVs for WGS https://github.com/Clinical-Genomics/BALSAMIC/pull/947
- SV and CNV analysis and *TIDDIT* to balsamic ReadtheDocs https://github.com/Clinical-Genomics/BALSAMIC/pull/951
- Gender to *config.json* https://github.com/Clinical-Genomics/BALSAMIC/pull/955
- Provided gender as input for *vcf2cyosure* https://github.com/Clinical-Genomics/BALSAMIC/pull/955
- SV CNV doc to balsamic READTHEDOCS https://github.com/Clinical-Genomics/BALSAMIC/pull/960
- Germline normal SNV VCF file header renaming to be compatible with genotype uploads https://github.com/Clinical-Genomics/BALSAMIC/issues/882
- Add tabix and gzip to vcf2cytosure container https://github.com/Clinical-Genomics/BALSAMIC/pull/969

13.19.2 Changed:

- UMI-workflow for panel cases to be run only with *balsamic-umi* flag https://github.com/Clinical-Genomics/BALSAMIC/issues/896
- Update *codecov* action version to @v2 https://github.com/Clinical-Genomics/BALSAMIC/pull/941
- QC-workflow for panel cases to be run only with *balsamic-qc* https://github.com/Clinical-Genomics/BALSAMIC/pull/942
- *get_snakefile* function takes the argument *analysis_workflow* to trigger the QC workflow when necessary https://github.com/Clinical-Genomics/BALSAMIC/pull/942
- *bcftools_counts* input depending on *analysis_workflow* https://github.com/Clinical-Genomics/BALSAMIC/pull/942
- UMI output filename *TNscope_umi* is changed to *tnscope_umi* https://github.com/Clinical-Genomics/BALSAMIC/pull/948
- Update *delly* to v1.0.3 https://github.com/Clinical-Genomics/BALSAMIC/pull/950
- Update versions of *delly* in ReadtheDocs https://github.com/Clinical-Genomics/BALSAMIC/pull/951

- Provided gender as input for *ascat* and *cnvkit* https://github.com/Clinical-Genomics/BALSAMIC/pull/955
- Update QC criteria for panel and wgs analysis according to https://github.com/Clinical-Genomics/project-planning/issues/338#issuecomment-1132643330. https://github.com/Clinical-Genomics/BALSAMIC/pull/952
- For uploads to scout, increasing the number of variants failing threshold from 10000 to 50000 https://github.com/Clinical-Genomics/BALSAMIC/pull/952

13.19.3 Fixed:

- GENOME_VERSION set to the different genome_version options and replaced with config["reference"]["genome_version"] https://github.com/Clinical-Genomics/BALSAMIC/pull/942
- *run_validate.sh* script https://github.com/Clinical-Genomics/BALSAMIC/pull/952
- Somatic SV tumor normal rules https://github.com/Clinical-Genomics/BALSAMIC/pull/959
- Missing *genderChr* flag for *ascat_tumor_normal* rule https://github.com/Clinical-Genomics/BALSAMIC/pull/963
- Command in vcf2cytosure rule and updated ReadtheDocs https://github.com/Clinical-Genomics/BALSAMIC/pull/966
- Missing name *analysis_dir* in QC.smk https://github.com/Clinical-Genomics/BALSAMIC/pull/970
- Remove *sample_type* wildcard from the *vcfheader_rename_germline* rule and change genotype file name https://github.com/Clinical-Genomics/BALSAMIC/pull/971

13.19.4 Removed

- Removed *qc_panel* config in favor of standard config https://github.com/Clinical-Genomics/BALSAMIC/pull/942
- Removed cli *–analysis_type* for *balsamic report deliver* command and *balsamic run analysis* https://github.com/Clinical-Genomics/BALSAMIC/pull/942
- Removed *analysis_type*: *qc_panel* and replace the trigger for QC workflow by *analysis_workflow*: *balsamic-qc* https://github.com/Clinical-Genomics/BALSAMIC/pull/942
- Outdated balsamic report files (*balsamic_report.html* & *balsamic_report.md*) https://github.com/Clinical-Genomics/BALSAMIC/pull/952

13.20 [9.0.1]

13.20.1 Fixed:

- Revert *csvkit* tool in align_qc container https://github.com/Clinical-Genomics/BALSAMIC/pull/928
- Automatic version update for balsamic methods https://github.com/Clinical-Genomics/BALSAMIC/pull/930

13.21 [9.0.0]

13.21.1 Added:

- Snakemake workflow to create canfam3 reference https://github.com/Clinical-Genomics/BALSAMIC/pull/843
- Call umi variants using TNscope in bed defined regions https://github.com/Clinical-Genomics/BALSAMIC/issues/821
- UMI duplication metrics to report in multiqc_picard_dups.json https://github.com/Clinical-Genomics/BALSAMIC/issues/844
- Option to use PON reference in cnv calling for TGA tumor-only cases https://github.com/Clinical-Genomics/BALSAMIC/pull/851
- QC default validation conditions (for not defined capture kits) https://github.com/Clinical-Genomics/BALSAMIC/pull/855
- SVdb to the varcall_py36 container https://github.com/Clinical-Genomics/BALSAMIC/pull/872
- SVdb to WGS workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/873
- Docker container for vcf2cytosure https://github.com/Clinical-Genomics/BALSAMIC/pull/869
- Snakemake rule for creating *.cgh* files from *CNVkit* outputs https://github.com/Clinical-Genomics/BALSAMIC/pull/880
- SVdb to TGA workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/879
- SVdb merge SV and CNV https://github.com/Clinical-Genomics/BALSAMIC/pull/886
- Readthedocs for BALSAMIC method descriptions https://github.com/Clinical-Genomics/BALSAMIC/pull/906
- Readthedocs for BALSAMIC variant filters for WGS somatic callers https://github.com/Clinical-Genomics/BALSAMIC/pull/906
- bcftools counts to varcall filter rules https://github.com/Clinical-Genomics/BALSAMIC/pull/899
- Additional WGS metrics to be stored in `<case>_metrics_deliverables.yaml` https://github.com/Clinical-Genomics/BALSAMIC/pull/907
- ascatNGS copynumber file https://github.com/Clinical-Genomics/BALSAMIC/pull/914
- ReadtheDocs for BALSAMIC annotation resources https://github.com/Clinical-Genomics/BALSAMIC/pull/916
- Delly CNV for tumor only workflow https://github.com/Clinical-Genomics/BALSAMIC/pull/923
- Delly CNV Read-depth profiles for tumor only workflows https://github.com/Clinical-Genomics/BALSAMIC/pull/924
- New metric to be extracted and validated: `NUMBER_OF_SITES` (`bcftools` counts) https://github.com/Clinical-Genomics/BALSAMIC/pull/925

13.21.2 Changed:

- Merge QC metric extraction workflows https://github.com/Clinical-Genomics/BALSAMIC/pull/833
- Changed the base-image for balsamic container to 4.10.3-alpine https://github.com/Clinical-Genomics/BALSAMIC/pull/869
- Updated SVdb to 2.6.0 https://github.com/Clinical-Genomics/BALSAMIC/pull/901
- Upgrade black to 22.3.0
- For UMI workflow, post filter *gnomad_pop_freq* value is changed from *0.005* to *0.02* https://github.com/Clinical-Genomics/BALSAMIC/pull/919
- updated delly to 0.9.1 https://github.com/Clinical-Genomics/BALSAMIC/pull/920
- container base_image (align_qc, annotate, coverage_qc, varcall_cnvkit, varcall_py36) to 4.10.3-alpine https://github.com/Clinical-Genomics/BALSAMIC/pull/921
- update container (align_qc, annotate, coverage_qc, varcall_cnvkit,varcall_py36) bioinfo tool versions https://github.com/Clinical-Genomics/BALSAMIC/pull/921
- update tool versions (align_qc, annotate, coverage_qc, varcall_cnvkit) in methods and softwares docs https://github.com/Clinical-Genomics/BALSAMIC/pull/921
- Updated the list of files to be stored and delivered https://github.com/Clinical-Genomics/BALSAMIC/pull/915
- Moved `collect_custom_qc_metrics` rule from `multiqc.rule` https://github.com/Clinical-Genomics/BALSAMIC/pull/925

13.21.3 Fixed:

- Automate balsamic version for readthedocs install page https://github.com/Clinical-Genomics/BALSAMIC/pull/888
- `collect_qc_metrics.py` failing for WGS cases with empty `capture_kit` argument https://github.com/Clinical-Genomics/BALSAMIC/pull/850
- QC metric validation for different panel bed version https://github.com/Clinical-Genomics/BALSAMIC/pull/855
- Fixed development version of `fpdf2` to `2.4.6` https://github.com/Clinical-Genomics/BALSAMIC/issues/878
- Added missing svdb index file https://github.com/Clinical-Genomics/BALSAMIC/issues/848

13.21.4 Removed

- `--qc-metrics/--no-qc-metrics` flag from the `balsamic report deliver` command https://github.com/Clinical-Genomics/BALSAMIC/pull/833
- Unused pon option for SNV calling with TNhaplotyper tumor-only https://github.com/Clinical-Genomics/BALSAMIC/pull/851
- SV and CNV callers from annotation and filtering https://github.com/Clinical-Genomics/BALSAMIC/pull/889
- vcfanno and COSMIC from SV annotation https://github.com/Clinical-Genomics/BALSAMIC/pull/891
- Removed *MSK_impact* and *MSK_impact_noStrelka* json files from config https://github.com/Clinical-Genomics/BALSAMIC/pull/903
- Cleanup of *strelka*, *pindel* , *mutect2* variables from BALSAMIC https://github.com/Clinical-Genomics/BALSAMIC/pull/903

- bcftools_stats from vep https://github.com/Clinical-Genomics/BALSAMIC/issues/898
- QC delivery report workflow (generating the `<case>_qc_report.html` file) https://github.com/Clinical-Genomics/BALSAMIC/issues/878
- `--sample-id-map` and `--case-id-map` flags from the `balsamic report deliver` command https://github.com/Clinical-Genomics/BALSAMIC/issues/878
- Removed *gatk_haplotypecaller* for reporting panel germline variants https://github.com/Clinical-Genomics/BALSAMIC/issues/918

13.22 [8.2.10]

13.22.1 Added:

- *libopenblas=0.3.20* dependency to annotate container for fixing bcftools #909

13.22.2 Fixes:

- bcftools version locked at *1.10* #909

13.22.3 Changed:

- base image of balsamic container to *4.10.3-alphine* #909
- Replaced annotate container tests with new code #909

13.22.4 Removed:

- Removed failed *vcf2cytosure* installation from annotate container #909

13.23 [8.2.9]

13.23.1 Added:

- Added slurm qos tag *express* #885
- Included more text about UMI-workflow variant calling settings to the readthedocs #888
- Extend QCModel to include *n_base_limit* which outputs in config json *QC* dict

13.23.2 Fixes:

- Automate balsamic version for readthedocs install page #888

13.23.3 Changed:

- Upgrade black to 22.3.0
- fastp default setting of *n_base_limit* is changed to *50* from *5*

13.24 [8.2.8]

13.24.1 Added:

- Added the readthedocs page for BALSAMIC variant-calling filters #867
- Project requirements (setup.py) to build the docs #874
- Generate cram from umi-consensus called bam files #865

13.24.2 Changed:

- Updated the bioinfo tools version numbers in BALSAMIC readthedocs #867
- Sphinx version fixed to <0.18 #874
- Sphinx GitHub action triggers only on master branch PRs
- VAF filter for reporting somatic variants (Vardict) is minimised to 0.7% from 1% #876

13.24.3 Fixes:

- cyvcf2 mock import for READTHEDOCS environment #874

13.25 [8.2.7]

13.25.1 Fixes:

- Fixes fastqc timeout issues for wgs cases #861
- Fix cluster configuration for vep and vcfanno #857

13.26 [8.2.6]

13.26.1 Fixes:

- Set right qos in scheduler command #856

13.27 [8.2.5]

- balsamic.sif container installation during cache generation #841

13.27.1 Fixed:

- Execution of *create_pdf* python script inside the balsamic container #841

13.28 [8.2.4]

13.28.1 Added:

- `--hgvsg` annotation to VEP #830
- `ascatNgs` PDF delivery (plots & statistics) #828

13.29 [8.2.3]

13.29.1 Fixed:

- Add default for gender if `purecn` captures dual gender values #824

13.29.2 Changed:

- Updated `purecn` and its dependencies to latest versions

13.30 [8.2.2]

13.30.1 Added:

- `ascatNGS` tumor normal delivery #810

13.30.2 Changed:

- QC metrics delivery tag #820
- Refactor tmb rule that contains redundant line #817

13.31 [8.2.1]

13.31.1 Fixed:

- `cnvkit` gender comparison operator bug #819

13.32 [8.2.0]

13.32.1 Added:

- Added various basic filters to all variant callers irregardless of their delivery status #750
- BALSAMIC container #728
- BALSAMIC reference generation via cluster submission for both reference and container #686
- Container specific tests #770
- BALSAMIC quality control metrics extraction and validation #754
- Delly is added as a submodule and removed from rest of the conda environments #787
- Store research VCFs for all filtered and annotated VCF files
- Added *.,PASS* to all structural variant filter rules to resolve the issues with missing calls in filtered file
- Handling of QC metrics validation errors #783
- Github Action workflow that builds the docs using Sphinx #809
- Zenodo integration to create citable link #813
- Panel BED specific QC conditions #800
- Metric extraction to a YAML file for Vogue #802

13.32.2 Changed:

- refactored main workflow with more readible organization #614
- refactored conda envs within container to be on base and container definition is uncoupled #759
- renamed umi output file names to fix issue with picard HSmetrics #804
- locked requirements for graphviz io 0.16 #811
- QC metric validation is performed across all metrics of each of the samples #800

13.32.3 Removed:

- The option of running umiworkflow independently with balsamic command-line option "-a umi"
- Removed source activate from reference and pon workflows #764

13.32.4 Fixed:

- Pip installation failure inside balsamic container #758
- Fixed issue #768 with missing `vep_install` command in container
- Fixed issue #765 with correct input bam files for SV rules
- Continuation of CNVkit even if `PURECN` fails and fix `PureCN` conda paths #774 #775
- Locked version for `cryptography` package
- Bumped version for `bcftools` in cnvkit container
- Fixed issues #776 and #777 with correct install paths for gatk and manta
- Fixed issue #782 for missing AF in the vcf INFO field
- Fixed issues #748 #749 with correct sample names
- Fixed issue #767 for ascatngs hardcoded values
- Fixed missing output option in bcftools filters for tnhaplotyper #793
- Fixed issue #795 with increasing resources for vep and filter SV prior to vep
- Building `wheel` for `cryptography` bug inside BALSAMIC container #801
- Fixed badget for docker container master and develop status
- ReadtheDocs building failure due to dependencies, fixed by locking versions #773
- Dev requirements installation for Sphinx docs (Github Action) #812
- Changed path for main Dockerfile version in `.bumpversion.cfg`

13.33 [8.1.0]

13.33.1 Added:

- Workflow to check PR tiltes to make easier to tell PR intents #724
- `bcftools stats` to calculate Ti/Tv for all post annotate germline and somatic calls #93
- Added reference download date to `reference.json` #726
- `ascatngs` hg38 references to constants #683
- Added ClinVar as a source to download and to be annotated with VCFAnno #737

13.33.2 Changed:

- Updated docs for git FAQs #731
- Rename panel of normal filename Clinical-Genomics/cgp-cancer-cnvcall#10

13.33.3 Fixed:

- Fixed bug with using varcall_py36 container with VarDict #739
- Fixed a bug with VEP module in MultiQC by excluding #746
- Fixed a bug with `bcftools stats` results failing in MultiQC #744

13.34 [8.0.2]

13.34.1 Fixed:

- Fixed breaking shell command for VEP annotation rules #734

13.35 [8.0.1]

13.35.1 Fixed:

- Fixed context for Dockerfile for release content #720

13.36 [8.0.0]

13.36.1 Added:

- `samtools` flagstats and stats to workflow and MultiQC
- `delly v0.8.7` somatic SV caller #644
- `delly` containter #644
- `bcftools v1.12` to `delly` container #644
- `tabix v0.2.6` to `delly` container #644
- Passed SV calls from Manta to clinical delivery
- An extra filter to VarDict tumor-normal to remove variants with STATUS=Germline, all other will still be around
- Added `vcf2cytosure` to annotate container
- `git` to the container definition
- prepare_delly_exclusion rule
- Installation of `PureCN` rpackage in `cnvkit` container
- Calculate tumor-purity and ploidy using `PureCN` for `cnvkit` call
- `ascatngs` as a submodule #672

- GitHub action to build and test `ascatngs` container
- Reference section to `docs/FAQ.rst`
- `ascatngs` download references from reference_file repository #672
- `delly` tumor only rule #644
- `ascatngs` download container #672
- Documentation update on setting sentieon env variables in `bashrc`
- `ascatngs` tumor normal rule for wgs cases #672
- Individual rules (i.e. ngs filters) for cnv and sv callers. Only Manta will be delivered and added to the list of output files. #708
- Added "targeted" and "wgs" tags to variant callers to provide another layer of separation. #708
- `manta` convert inversion #709
- Sentieon version to bioinformatic tool version parsing #685
- added `CITATION.cff` to cite BALSAMIC

13.36.2 Changed:

- Upgrade to latest sentieon version 202010.02
- New name `MarkDuplicates` to `picard_markduplicates` in `bwa_mem` rule and `cluster.json`
- New name rule `GATK_contest` to `gatk_contest`
- Avoid running pytest github actions workflow on `docs/**` and `CHANGELOG.rst` changes
- Updated `snakemake` to `v6.5.3` #501
- Update `GNOMAD` URL
- Split Tumor-only `cnvkit batch` into individual commands
- Improved TMB calculation issue #51
- Generalized ascat, delly, and manta result in workflow. #708
- Generalized workflow to eliminate duplicate entries and code. #708
- Split Tumor-Normal `cnvkit batch` into individual commands
- Moved params that are used in multiple rules to constants #711
- Changed the way conda and non-conda bioinfo tools version are parsed
- Python code formatter changed from Black to YAPF #619

13.36.3 Fixed:

- post-processing of the umi consensus in handling BI tags
- vcf-filtered-clinical tag files will have all variants including PASS
- Refactor snakemake `annotate` rules according to snakemake etiquette #636
- Refactor snakemake `align` rules according to snakemake etiquette #636
- Refactor snakemake `fastqc` `vep` contest and `mosdepth` rules according to `snakemake` etiquette #636
- Order of columns in QC and coverage report issue #601
- `delly` not showing in workflow at runtime #644
- `ascatngs` documentation links in `FAQs` #672
- `varcall_py36` container build and push #703
- Wrong spacing in reference json issue #704
- Refactor snakemake `quality control` rules according to snakemake etiquette #636

13.36.4 Removed:

- Cleaned up unused container definitions and conda environment files
- Remove cnvkit calling for WGS cases
- Removed the install.sh script

13.37 [7.2.5]

13.37.1 Changed:

- Updated COSMIC path to use version 94

13.38 [7.2.5]

13.38.1 Changed:

- Updated path for gnomad and 1000genomes to a working path from Google Storage

13.39 [7.2.4]

13.39.1 Changed:

- Updated sentieon util sort in umi to use Sentieon 20201002 version

13.40 [7.2.3]

13.40.1 Fixed:

- Fixed memory issue with vcfanno in vep_somatic rule fixes #661

13.41 [7.2.2]

13.41.1 Fixed:

- An error with Sentieon for better management of memory fixes #621

13.42 [7.2.1]

13.42.1 Changed:

- Rename Github actions to reflect their content

13.43 [7.2.0]

13.43.1 Added:

- Changelog reminder workflow to Github
- Snakemake workflow for created PON reference
- Balsamic cli config command(pon) for creating json for PON analysis
- tumor lod option for passing tnscope-umi final variants
- Git guide to make balsamic release in FAQ docs

13.43.2 Changed:

- Expanded multiqc result search dir to whole analysis dir
- Simple test for docker container

13.43.3 Fixed:

- Correctly version bump for Dockerfile

13.43.4 Removed:

- Removed unused Dockerfile releases
- Removed redundant genome version from `reference.json`

13.44 [7.1.10]

13.44.1 Fixed:

- Bug in `ngs_filter` rule set for tumor-only WGS
- Missing delivery of tumor only WGS filter

13.45 [7.1.9]

13.45.1 Changed:

- only pass variants are not part of delivery anymore
- delivery tag file ids are properly matched with sample_name
- tabix updated to 0.2.6
- fastp updated to 0.20.1
- samtools updated to 1.12
- bedtools updated to 2.30.0

13.45.2 Removed:

- sentieon-dedup rule from delivery
- Removed all pre filter pass from delivery

13.46 [7.1.8]

13.46.1 Fixed:

- Target coverage (Picard HsMetrics) for UMI files is now correctly calculated.

13.46.2 Changed:

- TNscope calculated AF values are fetched and written to AFtable.txt.

13.47 [7.1.7]

13.47.1 Added:

- ngs_filter_tnscope is also part of deliveries now

13.47.2 Changed:

- rankscore is now a research tag instead of clinical
- Some typo and fixes in the coverage and constant metrics
- Delivery process is more verbose

13.47.3 Fixed:

- CNVKit output is now properly imported in the deliveries and workflow

13.48 [7.1.6]

13.48.1 Fixed:

- CSS style for qc coverage report is changed to landscape

13.49 [7.1.5]

13.49.1 Changed:

- update download url for 1000genome WGS sites from ftp to http

13.50 [7.1.4]

13.50.1 Changed:

- bump picard to version 2.25.0

13.51 [7.1.3]

13.51.1 Fixed:

- `assets` path is now added to bind path

13.52 [7.1.2]

13.52.1 Fixed:

- umi_workflow config json is set as true for panel and wgs as false.
- Rename umiconsensus bam file headers from {samplenames} to TUMOR/NORMAL.
- Documentation autobuild on RTFD

13.53 [7.1.1]

13.53.1 Fixed:

- Moved all requirements to setup.py, and added all package_data there. Clean up unused files.

13.54 [7.1.0]

13.54.1 Removed

- `tnsnv` removed from WGS analysis, both tumor-only and tumor-normal
- GATK-BaseRecalibrator is removed from all workflows

13.54.2 Fixed

- Fixed issue 577 with missing `tumor.merged.bam` and `normal.merged.bam`
- Issue 448 with lingering tmp_dir. It is not deleted after analysis is properly finished.

13.54.3 Changed

- All variant calling rules use proper `tumor.merged.bam` or `normal.merged.bam` as inputs

13.55 [7.0.2]

13.55.1 Added

- Updated docs with FAQ for UMI workflow

13.55.2 Fixed

- fix job scheduling bug for benchmarking
- rankscore's output is now a proper vcf.gz file
- Manta rules now properly make a sample_name file

13.56 [7.0.1]

13.56.1 Added

- github action workflow to autobuild release containers

13.57 [7.0.0]

13.57.1 Added

- `balsamic init` to download reference and related containers done in PRs #464 #538
- `balsamic config case` now only take a cache path instead of container and reference #538
- UMI workflow added to main workflow in series of PRs #469 #477 #483 #498 #503 #514 #517
- DRAGEN for WGS applications in PR #488
- A framework for QC check PR #401
- `--quiet` option for `run analysis` PR #491
- Benchmark SLURM jobs after the analysis is finished PR #534
- One container per conda environment (i.e. decouple containers) PR #511 #525 #522
- `--disable-variant-caller` command for `report deliver` PR #439
- Added genmod and rankscore in series of two PRs #531 and #533
- Variant filtering to Tumor-Normal in PR #534
- Split SNV/InDels and SVs from TNScope variant caller PR #540
- WGS Tumor only variant filters added in PR #548

13.57.2 Changed

- Update Manta to 1.6.0 PR #470
- Update FastQC to 0.11.9 PR #532
- Update BCFTools to 1.11 PR #537
- Update Samtools to 1.11 PR #537
- Increase resources and runtime for various workflows in PRs #482
- Python package dependenicies versions fixed in PR #480
- QoL changes to workflow in series of PR #471
- Series of documentation updates in PRs #489 #553
- QoL changes to scheduler script PR #491
- QoL changes to how temporary directories are handlded PR #516
- TNScope model apply rule merged with TNScope variant calling for tumor-normal in WGS #540
- Decoupled `fastp` rule into two rules to make it possible to use it for UMI runs #570

13.57.3 Fixed

- A bug in Manta variant calling rules that didn't name samples properly to TUMOR/NORMAL in the VCF file #572

13.58 [6.1.2]

13.58.1 Changed

- Changed hk delivery tag for coverage-qc-report

13.59 [6.1.1]

13.59.1 Fixed

- No UMI trimming for WGS applications #486
- Fixed a bug where BALSAMIC was checking for sacct/jobid file in local mode PR #497
- `readlink` command in `vep_germline`, `vep_somatic`, `split_bed`, and `GATK_popVCF` #533
- Fix various bugs for memory handling of Picardtools and its executable in PR #534
- Fixed various issues with `gsutils` in PR #550

13.59.2 Removed

- `gatk-register` command removed from installing GATK PR #496

13.60 [6.1.1]

- Fixed a bug with missing QC templates after `pip install`

13.61 [6.1.0]

13.61.1 Added

- CLI option to expand report generation for TGA and WES runs. Please see `balsamic report deliver --help`
- BALSAMIC now generates a custom HTML report for TGA and WES cases.

13.62 [6.0.4]

13.62.1 Changed

- Reduces MQ cutoff from 50 to 40 to only remove obvious artifacts PR #535
- Reduces AF cutoff from 0.02 to 0.01 PR #535

13.63 [6.0.3]

13.63.1 Added

- `config case` subcommand now has `--tumor-sample-name` and `--normal-sample-name`

13.63.2 Fixed

- Manta resource allocation is now properly set PR #523
- VarDict resource allocation in cluster.json increased (both core and time allocation) PR #523
- minimum memory request for GATK mutect2 and haplotypecaller is removed and max memory increased PR #523

13.64 [6.0.2]

13.64.1 Added

- Document for Snakemake rule grammar PR #489

13.64.2 Fixed

- removed `gatk3-register` command from Dockerfile(s) PR #508

13.65 [6.0.1]

13.65.1 Added

- A secondary path for latest jobids submitted to cluster (slurm and qsub) PR #465

13.66 [6.0.0]

13.66.1 Added

- UMI workflow using Sentieon tools. Analysis run available via *balsamic run analysis –help* command. PR #359
- VCFutils to create VCF from flat text file. This is for internal purpose to generate validation VCF. PR #349
- Download option for hg38 (not validated) PR #407
- Option to disable variant callers for WES runs. PR #417

13.66.2 Fixed

- Missing cyvcf2 dependency, and changed conda environment for base environment PR #413
- Missing numpy dependency PR #426

13.66.3 Changed

- COSMIC db for hg19 updated to v90 PR #407
- Fastp trimming is now a two-pass trimming and adapter trimming is always enabled. This might affect coverage slightly PR #422
- All containers start with a clean environment #425
- All Sentieon environment variables are now added to config when workflow executes #425
- Branching model will be changed to gitflow

13.67 [5.1.0]

13.67.1 Fixed

- Vardict-java version fixed. This is due to bad dependency and releases available on conda. Anaconda is not yet update with vardict 1.8, but vardict-java 1.8 is there. This causes various random breaks with Vardict's TSV output. #403

13.67.2 Changed

- Refactored Docker files a bit, preparation for decoupling #403

13.67.3 Removed

- In preparation for GATK4, IndelRealigner is removed #404

13.68 [5.0.1]

13.68.1 Added

- Temp directory for various rules and workflow wide temp directory #396

13.68.2 Changed

- Refactored tags for housekeeper delivery to make them unique #395
- Increased core requirements for mutect2 #396
- GATK3.8 related utils run via jar file instead of gatk3 #396

13.69 [5.0.0]

13.69.1 Added

- Config.json and DAG draph included in Housekeeper report #372
- New output names added to cnvkit_single and cnvkit_paired #372
- New output names added to vep.rule #372
- Delivery option to CLI and what to delivery with delivery params in rules that are needed to be delivered #376
- Reference data model with validation #371
- Added container path to install script #388

13.69.2 Changed

- Delivery file format simplified #376
- VEP rules have "all" and "pass" as output #376
- Downloaded reference structure changed #371
- genome/refseq.flat renamed to genome/refGene.flat #371
- reverted CNVKit to version 0.9.4 #390

13.69.3 Fixed

- Missing pygments to requirements.txt to fix travis CI #364
- Wildcard resolve for deliveries of vep_germline #374
- Missing index file from deliverables #383
- Ambiguous deliveries in vep_somatic and ngs_filters #387
- Updated documentation to match with installation #391

13.69.4 Removed

- Temp files removed from list of outputs in vep.rule #372
- samtools.rule and merged it with bwa_mem #375

13.70 [4.5.0]

13.70.1 Added

- Models to build config case JSON. The models and descriptions of their contents can now be found in BALSAMIC/utils/models.py
- Added analysis_type to *report deliver* command
- Added report and delivery capability to Alignment workflow
- run_validate.sh now has -d to handle path to analysis_dir (for internal use only) #361

13.70.2 Changed

- Fastq files are no longer being copied as part of creation of the case config file. A symlink is now created at the destination path instead
- Config structure is no longer contained in a collestion of JSON files. The config models are now built using Pydantic and are contained in BALSAMIC/utils/models.py

13.70.3 Removed

- Removed command line option "–fastq-prefix" from config case command
- Removed command line option "–config-path" from config case command. The config is now always saved with default name "case_id.json"
- Removed command line option "–overwrite-config" from config-case command The command is now always executed with "–overwrite-config True" behavior

13.70.4 Refactored

- Refactored BALSAMIC/commands/config/case.py: Utility functions are moved to BALSAMIC/utils/cli.py Models for config fields can be found at BALSAMIC/utils/models.py Context aborts and logging now contained in pilot function Tests created to support new architecture
- Reduce analysis directory's storage

13.70.5 Fixed

- Report generation warnings supressed by adding workdirectory
- Missing tag name for germline annotated calls #356
- Bind path is not added as None if analysis type is wgs #357
- Changes vardict to vardict-java #361

13.71 [4.4.0]

13.71.1 Added

- pydantic to validate various models namely variant caller filters

13.71.2 Changed

- Variant caller filters moved into pydantic
- Install script and setup.py
- refactored install script with more log output and added a conda env suffix option
- refactored docker container and decoupled various parts of the workflow

13.72 [4.3.0]

13.72.1 Added

- Added cram files for targeted sequencing runs fixes #286
- Added *mosdepth* to calculate coverage for whole exome and targeted sequencing
- Filter models added for tumor-only mode
- Enabling adapter trim enables pe adapter trim option for fastp
- Annotate germline variant calls
- Baitset name to picard hsmetrics

13.72.2 Deprecated

- Sambamba coverage and rules will be deprecated

13.72.3 Fixed

- Fixed latest tag in install script
- Fixed lack of naming final annotated VCF TUMOR/NORMAL

13.72.4 Changed

- Increased run time for various slurm jobs fixes #314
- Enabled SV calls for VarDict tumor-only
- Updated *ensembl-vep* to v100.2

13.73 [4.2.4]

13.73.1 Fixed

- Fixed sort issue with bedfiles after 100 slop

13.74 [4.2.3]

13.74.1 Added

- Added Docker container definition for release and bumpversion

13.74.2 Changed

- Quality of life change to rtfd docs

13.74.3 Fixed

- Fix Docker container with faulty git checkout

13.75 [4.2.2]

13.75.1 Added

- Add "SENTIEON_TMPDIR" to wgs workflow

13.76 [4.2.1]

13.76.1 Changed

- Add docker container pull for correct version of install script

13.77 [4.2.0]

13.77.1 Added

- CNV output as VCF
- Vep output for PASSed variants
- Report command with status and delivery subcommands

13.77.2 Changed

- Bed files are slopped 100bp for variant calling fix #262
- Disable vcfmerge
- Picard markduplicate output moved from log to output
- Vep upgraded to 99.1
- Removed SVs from vardict
- Refactored delivery plugins to produce a file with list of output files from workflow
- Updated snakemake to 5.13

13.77.3 Fixed

- Fixed a bug where threads were not sent properly to rules

13.77.4 Removed

- Removed coverage annotation from mutect2
- Removed source deactivate from rules to suppress conda warning
- Removed `plugins delivery` subcommand
- Removed annotation for germline caller results

13.78 [4.1.0]

13.78.1 Added

- VEP now also produces a tab delimited file
- CNVkit rules output genemetrics and gene break file
- Added reference genome to be able to calculate AT/CG dropouts by Picard
- coverage plot plugin part of issue #75
- callable regions for CNV calling of tumor-only

13.78.2 Changed

- Increased time for indel realigner and base recalib rules
- decoupled vep stat from vep main rule
- changed qsub command to match UGE
- scout plugin updated

13.78.3 Fixed

- WGS qc rules - updated with correct options (picard - CollectMultipleMetrics, sentieon - CoverageMetrics)
- Log warning if WES workflow cannot find SENTIEON* env variables
- Fixes issue with cnvkit and WGS samples #268
- Fix #267 coverage issue with long deletions in vardict

13.79 [4.0.1] - 2019-11-08

13.79.1 Added

- dependencies for workflow report
- sentieon variant callers germline and somatic for wes cases

13.79.2 Changed

- housekeeper file path changed from basename to absolute
- scout template for sample location changed from delivery_report to scout
- rule names added to benchmark files

13.80 [4.0.0] - 2019-11-04

SGE qsub support release

13.80.1 Added

- `install.sh` now also downloads latest container
- Docker image for balsamic as part of ci
- Support for qsub alongside with slurm on `run analysis --profile`

13.80.2 Changed

- Documentation updated
- Test fastq data and test panel bed file with real but dummy data

13.81 [3.3.1] - 2019-10-28

13.81.1 Fixed

- Various links for reference genome is updated with working URL
- Config reference command now print correct output file

13.82 [3.3.0] - 2019-10-24

somatic vcfmerge release

13.82.1 Added

- QC metrics for WGS workflow
- refGene.txt download to reference.json and reference workflow
- A new conda environment within container
- A new base container built via Docker (centos7:miniconda3_4_6_14)
- VCFmerge package as VCF merge rule (https://github.com/hassanfa/VCFmerge)
- A container for develop branch
- Benchmark rules to variant callers

13.82.2 Changed

- SLURM resource allocation for various variancalling rules optimized
- mergetype rule updated and only accepts one single tumor instead of multiple

13.83 [3.2.3] - 2019-10-24

13.83.1 Fixed

- Removed unused output files from cnvkit which caused to fail on targetted analysis

13.84 [3.2.2] - 2019-10-23

13.84.1 Fixed

- Removed target file from cnvkit batch

13.85 [3.2.1] - 2019-10-23

13.85.1 Fixed

- CNVkit single missing reference file added

13.86 [3.2.0] - 2019-10-11

13.86.1 Adds:

- CNVkit to WGS workflow
- get_thread for runs

13.86.2 Changed:

- Optimized resources for SLURM jobs

13.86.3 Removed:

- Removed hsmetrics for non-mark duplicate bam files

13.87 [3.1.4] - 2019-10-08

13.87.1 Fixed

- Fixes a bug where missing capture kit bed file error for WGS cases

13.88 [3.1.3] - 2019-10-07

13.88.1 Fixed

- benchmark path bug issue #221

13.89 [3.1.2] - 2019-10-07

13.89.1 Fixed

- libreadline.so.6 symlinking and proper centos version for container

13.90 [3.1.1] - 2019-10-03

13.90.1 Fixed

- Proper tag retrieval for release ### Changed
- BALSAMIC container change to latest and version added to help line

13.91 [3.1.0] - 2019-10-03

TL;DR:

- QoL changes to WGS workflow
- Simplified installation by moving all tools to a container

13.91.1 Added

- Benchmarking using psutil
- ML variant calling for WGS
- `--singularity` option to `config case` and `config reference`

13.91.2 Fixed

- Fixed a bug with boolean values in analysis.json

13.91.3 Changed

- `install.sh` simplified and will be depricated
- Singularity container updated
- Common somatic and germline variant callers are put in single file
- Variant calling workflow and analysis config files merged together

13.91.4 Removed

- `balsamic install` is removed
- Conda environments for py36 and py27 are removed

13.92 [3.0.1] - 2019-09-11

13.92.1 Fixed

- Permissions on `analysis/qc` dir are 777 now

13.93 [3.0.0] - 2019-09-05

This is major release. TL;DR:

- Major changes to CLI. See documentation for updates.
- New additions to reference generation and reference config file generation and complete overhaul
- Major changes to reposityory structure, conda environments.

13.93.1 Added

- Creating and downloading reference files: `balsamic config reference` and `balsamic run reference`
- Container definitions for install and running BALSAMIC
- Bunch of tests, setup coveralls and travis.
- Added Mutliqc, fastp to rule utilities
- Create Housekeeper and Scout files after analysis completes
- Added Sentieon tumor-normal and tumor only workflows
- Added trimming option while creating workflow
- Added multiple tumor sample QC analysis
- Added pindle for indel variant calling
- Added Analysis finish file in the analysis directory

13.93.2 Fixed

- Multiple fixes to snakemake rules

13.93.3 Changed

- Running analysis through: `balsamic run analysis`
- Cluster account and email info added to `balsamic run analysis`
- `umi` workflow through `--umi` tag. [workflow still in evaluation]
- `sample-id` replaced by `case-id`
- Plan to remove FastQC as well

13.93.4 Removed

- `balsamic config report` and `balsamic report`
- `sample.config` and `reference.json` from config directory
- Removed cutadapt from workflows

13.94 [2.9.8] - 2019-01-01

13.94.1 Fixed

- picard hsmetrics now has 50000 cov max
- cnvkit single wildcard resolve bug fixed

13.95 [2.9.7] - 2019-02-28

13.95.1 Fixed

- Various fixes to umi_single mode
- analysis_finish file does not block reruns anymore
- Added missing single_umi to analysis workflow cli

13.95.2 Changed

- vardict in single mode has lower AF threshold filter (0.005 -> 0.001)

13.96 [2.9.6] - 2019-02-25

13.96.1 Fixed

- Reference to issue #141, fix for 3 other workflows
- CNVkit rule update for refflat file

13.97 [2.9.5] - 2019-02-25

13.97.1 Added

- An analysis finish file is generated with date and time inside (%Y-%M-%d T%T %:z)

13.98 [2.9.4] - 2019-02-13

13.98.1 Fixed

- picard version update to 2.18.11 github.com/hassanfa/picard

13.99 [2.9.3] - 2019-02-12

13.99.1 Fixed

- Mutect single mode table generation fix
- Vardict single mode MVL annotation fix

13.100 [2.9.2] - 2019-02-04

13.100.1 Added

- CNVkit single sample mode now in workflow
- MVL list from cheng et al. 2015 moved to assets

13.101 [2.9.1] - 2019-01-22

13.101.1 Added

- Simple table for somatic variant callers for single sample mode added

13.101.2 Fixed

- Fixes an issue with conda that unset variables threw an error issue #141

13.102 [2.9.0] - 2019-01-04

13.102.1 Changed

- Readme structure and example
- Mutect2's single sample output is similar to paired now
- cli path structure update

13.102.2 Added

- test data and sample inputs
- A dag PDF will be generated when config is made
- umi specific variant calling

13.103 [2.8.1] - 2018-11-28

13.103.1 Fixed

- VEP's perl module errors
- CoverageRep.R now properly takes protein_coding transcatipts only

13.104 [2.8.0] - 2018-11-23

UMI single sample align and QC

13.104.1 Added

- Added rules and workflows for UMI analysis: QC and alignment

13.105 [2.7.4] - 2018-11-23

Germline single sample

13.105.1 Added

- Germline single sample addition ### Changed
- Minor fixes to some rules to make them compatible with tumor mode

13.106 [2.7.3] - 2018-11-20

13.106.1 Fixed

- Various bugs with DAG to keep popvcf and splitbed depending on merge bam file
- install script script fixed and help added

13.107 [2.7.2] - 2018-11-15

13.107.1 Changed

- Vardict, Strelka, and Manta separated from GATK best practice pipeline

13.108 [2.7.1] - 2018-11-13

13.108.1 Fixed

- minro bugs with strelka_germline and freebayes merge ### Changed
- removed ERC from haplotypecaller

13.109 [2.7.0] - 2018-11-08

Germline patch

13.109.1 Added

- Germline caller tested and added to the paired analysis workflow: Freebayes, HaplotypeCaller, Strelka, Manta

13.109.2 Changed

- Analysis config files updated
- Output directory structure changed
- vep rule is now a single rule
- Bunch of rule names updated and shortened, specifically in Picard and GATK
- Variant caller rules are all updated and changed
- output vcf file names are now more sensible: {SNV,SV}.{somatic,germline}.sampleId.variantCaller.vcf.gz
- Job limit increased to 300

13.109.3 Removed

- removed bcftools.rule for var id annotation

13.109.4 Changed

13.109.5 Fixed

13.110 [2.6.3] - 2018-11-01

13.110.1 Changed

- Ugly and godforsaken `runSbatch.py` is now dumping sacct files with job IDs. Yikes!

13.111 [2.6.2] - 2018-10-31

13.111.1 Fixed

- added `--fastq-prefix` option for `config sample` to set fastq prefix name. Linking is not changed.

13.112 [2.6.1] - 2018-10-29

13.112.1 Fixed

- patched a bug for copying results for strelka and manta which was introduced in `2.5.0`

13.113 [2.5.0] - 2018-10-22

13.113.1 Changed

- `variant_panel` changed to `capture_kit`
- sample config file takes balsamic version
- bioinfo tool config moved bioinfotool to cli_utils from `config report`

13.113.2 Added

- bioinfo tool versions is now added to analysis config file

13.114 [2.4.0] - 2018-10-22

13.114.1 Changed

- `balsamic run` has 3 stop points: paired variant calling, single mode variant calling, and QC/Alignment mode.
- `balsamic run [OPTIONS] -S ...` is depricated, but it supersedes `analysis_type` mode if provided.

13.115 [2.3.3] - 2018-10-22

13.115.1 Added

- CSV output for variants in each variant caller based on variant filters
- DAG image of workflow ### Changed
- Input for variant filter has a default value
- `delivery_report` is no created during config generation
- Variant reporter R script cmd updated in `balsamic report`

13.116 [2.3.2] - 2018-10-19

13.116.1 Changed

- Fastq files are now always linked to `fastq` directory within the analysis directory

13.116.2 Added

- `balsamic config sample` now accepts individual files and paths. See README for usage.

13.117 [2.3.1] - 2018-09-25

13.117.1 Added

- CollectHSmetric now run twice for before and after markduplicate

13.118 [2.3.0] - 2018-09-25

13.118.1 Changed

- Sample config file now includes a list of chromosomes in the panel bed file

13.118.2 Fixed

- Non-matching chrom won't break the splitbed rule anymore
- collectqc rules now properly parse tab delimited metric files

13.119 [2.2.0] - 2018-09-11

13.119.1 Added

- Coverage plot to report
- target coverage file to report json
- post-cutadapt fastqc to collectqc
- A header to report pdf
- list of bioinfo tools used in the analysis added to report ### Changed
- VariantRep.R now accepts multiple inputs for each parameter (see help)
- AF values for MSKIMPACT config ### Fixed
- Output figure for coverageplot is now fully square :-)

13.120 [2.1.0] - 2018-09-11

13.120.1 Added

- normalized coverage plot script
- fastq file IO check for config creation
- added qos option to `balsamic run` ### Fixed
- Sambamba depth coverage parameters
- bug with picard markduplicate flag

13.121 [2.0.2] - 2018-09-11

13.121.1 Added

- Added qos option for setting qos to run jobs with a default value of low

13.122 [2.0.1] - 2018-09-10

13.122.1 Fixed

- Fixed package dependencies with vep and installation

13.123 [2.0.0] - 2018-09-05

Variant reporter patch and cli update

13.123.1 Added

- Added `balsamic config sample` and `balsamic config report` to generate run analysis and reporting config
- Added `VariantRep.R` script to information from merged variant table: variant summry, TMB, and much more
- Added a workflow for single sample mode alignment and QC only
- Added QC skimming script to qccollect to generate nicely formatted information from picard ### Changed
- Change to CLI for running and creating config
- Major overhaul to coverage report script. It's now simpler and more readable! ### Fixed
- Fixed sambamba depth to include mapping quality
- Markduplicate now is now by default on marking mode, and will NOT remove duplicates
- Minor formatting and script beautification happened

13.124 [1.13.1] - 2018-08-17

13.124.1 Fixed

- fixed a typo in MSKMVL config
- fixed a bug in strelka_simple for correct column orders

13.125 [1.13.0] - 2018-08-10

13.125.1 Added

- rule for all three variant callers for paired analysis now generate a simple VCF file
- rule for all three variant callers for paired analysis to convert VCF into table format
- MVL config file and MVL annotation to VCF calls for SNV/INDEL callers
- CALLER annotation added to SNV/INDEL callers
- exome specific option for strelka paired
- create_config subcommand is now more granular, it accepts all enteries from sample.json as commandline arguments
- Added tabQuery to the assets as a tool to query the tabulated output of summarized VCF
- Added MQ annotation field to Mutect2 output see #67 ### Changed
- Leaner VCF output from mutect2 with coverage and MQ annotation according to #64
- variant ids are now updated from simple VCF file ### Fixed
- Fixed a bug with sambamba depth coverage reporting wrong exon and panel coverage see #68
- The json output is now properly formatted using yapf
- Strelka rule doesn't filter out PASS variants anymore fixes issue #63

13.126 [1.12.0] - 2018-07-06

Coverage report patch

13.126.1 Added

- Added a new script to retrieve coverage report for a list of gene(s) and transcripts(s)
- Added sambamba exon depth rule for coverage report
- Added a new entry in reference json for exon bed file, this file generated using: https://github.com/hassanfa/GFFtoolkit ### Changed
- sambamba_depth rule changed to sambama_panel_depth
- sambamba depth now has fix-mate-overlaps parameter enabled
- sambamba string filter changed to `unmapped or mate\_is\_unmapped) and not duplicate and not failed\_quality\_control`.

- sambamba depth for both panel and exon work on picard flag (rmdup or mrkdup). ### Fixed
- Fixed sambamba panel depth rule for redundant coverage parameter

13.127 [1.11.0] - 2018-07-05

create config patch for single and paired mode

13.127.1 Changed

- create_config is now accepting a paired|single mode instead of analysis json template (see help for changes). It is not backward compatible ### Added
- analysis_{paired single}.json for creating config. Analysis.json is now obsolete. ### Fixed
- A bug with writing output for analysis config, and creating the path if it doesn't exist.
- A bug with manta rule to correctly set output files in config.
- A bug that strelka was still included in sample analysis.

13.128 [1.10.0] - 2018-06-07

13.128.1 Added

- Markduplicate flag to analysis config

13.129 [1.9.0] - 2018-06-04

13.129.1 Added

- Single mode for vardict, manta, and mutect.
- merge type for tumor only ### Changed
- Single mode variant calling now has all variant calling rules ### Fixed
- run_analaysis now accepts workflows for testing pyrposes

13.130 [1.8.0] - 2018-06-01

13.130.1 Changed

- picard create bed interval rule moved into collect hsmetric
- split bed is dependent on bam merge rule
- vardict env now has specific build rather than URL download (conda doesn't support URLs anymore) ### Fixed
- new logs and scripts dirs are not re-created if they are empty

13.131 [1.7.0] - 2018-05-31

13.131.1 Added

- A source altered picard to generated more quality metrics output is added to installation and rules

13.132 [1.6.0] - 2018-05-30

13.132.1 Added

- report subcommand for generating a pdf report from a json input file
- Added fastqc after removing adapter ### Changed
- Markduplicate now has both REMOVE and MARK (rmdup vs mrkdup)
- CollectHSMetrics now has more steps on PCT_TARGET_BASES

13.133 [1.5.0] - 2018-05-28

13.133.1 Changed

- New log and script directories are now created for each re-run ### Fixed
- Picardtools' memory issue addressed for large samples

13.134 [1.4.0] - 2018-05-18

13.134.1 Added

- single sample analysis mode
- alignment and insert size metrics are added to the workflow ### Changed
- collectqc and contest have their own rule for paired (tumor vs normal) and single (tumor only) sample.

13.135 [1.3.0] - 2018-05-13

13.135.1 Added

- bed file for panel analysis is now mandatory to create analaysis config

13.136 [1.2.3] - 2018-05-13

13.136.1 Changed

- vep execution path
- working directory for snakemake

13.137 [1.2.2] - 2018-05-04

13.137.1 Added

- sbatch submitter and cluster config now has an mail field ### Changed
- `create_config` now only requires sample and output json. The rest are optional

13.138 [1.2.0] - 2018-05-02

13.138.1 Added

- snakefile and cluster config in run analysis are now optional with a default value

13.139 [1.1.2] - 2018-04-27

13.139.1 Fixed

- vardict installation was failing without conda-forge channel
- gatk installation was failing without correct jar file

13.140 [1.1.1] - 2018-04-27

13.140.1 Fixed

- gatk-register tmp directory

13.141 [1.1.0] - 2018-04-26

13.141.1 Added

- create config sub command added as a new feature to create input config file
- templates to generate a config file for analysis added
- code style template for YAPF input created. see: https://github.com/google/yapf

- vt conda env added

13.141.2 Changed

- install script changed to create an output config
- README updated with usage

13.141.3 Fixed

- fastq location for analysis config is now fixed
- lambda rules removed from cutadapt and fastq

13.142 [1.0.3-rc2] - 2018-04-18

13.142.1 Added

- Added sbatch submitter to handle it outside snakemake ### Changed
- sample config file structure changed
- coding styles updated

13.143 [1.0.2-rc2] - 2018-04-17

13.143.1 Added

- Added vt environment ### Fixed
- conda envs are now have D prefix instead of P (develop vs production)
- install_conda subcommand now accepts a proper conda prefix

13.144 [1.0.1-rc2] - 2018-04-16

13.144.1 Fixed

- snakemake rules are now externally linked

13.145 [1.0.0-rc2] - 2018-04-16

13.145.1 Added

- run_analysis subcommand
- Mutational Signature R script with CLI
- unittest to install_conda
- a method to semi-dynamically retrieve suitable conda env for each rule

13.145.2 Fixed

- install.sh updated with gatk and proper log output
- conda environments updated
- vardict now has its own environment and it should not raise anymore errors

13.146 [1.0.0-rc1] - 2018-04-05

13.146.1 Added

- install.sh to install balsamic
- balsamic barebone cli
- subcommand to install required environments
- README.md updated with basic installation instructions

13.146.2 Fixed

- conda environment yaml files

CHAPTER

FOURTEEN

TOOLS AND SOFTWARE

BALSAMIC (**version** = 15.0.0) uses myriad of tools and softwares to analyze fastq files. This section covers why each one is included: usage and parameters, and relevant external links.

14.1 ascatNgs

Source code
GitHub https://github.com/cancerit/ascatNgs

Article
PNAS https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6097604/

Version
4.5.0

14.2 bcftools

Source code
GitHub https://github.com/samtools/bcftools

Article
Bioinformatics https://pubmed.ncbi.nlm.nih.gov/21903627/

Version
>=1.10

14.3 bedtools

Source code
GitHub https://github.com/arq5x/bedtools2

Article
Bioinformatics https://pubmed.ncbi.nlm.nih.gov/20110278/

Version
2.30.0

14.4 bwa

Source code
GitHub https://github.com/lh3/bwa

Article
Bioinformatics https://arxiv.org/abs/1303.3997

Version
0.7.17

14.5 cadd

Source code
GitHub https://github.com/kircherlab/CADD-scripts/

Article
Nature Genetics https://dx.doi.org/10.1038/ng.2892

Version
1.6

14.6 cnvkit

Source code
GitHub https://github.com/etal/cnvkit

Article
PLOS Computational Biology https://doi.org/10.1371/journal.pcbi.1004873

Version
0.9.9

14.7 cnvpytor

Source code
GitHub <https://github.com/abyzovlab/CNVpytor/>

Article
GigaSciences <https://doi.org/10.1093/gigascience/giab074>

Version
1.3.1

14.8 delly

Source code
GitHub https://github.com/dellytools/delly

Article
Bioinformatics https://academic.oup.com/bioinformatics/article/28/18/i333/245403

Version
1.0.3

14.9 ensembl-vep

Source code
GitHub https://github.com/Ensembl/ensembl-vep

Article
Genome Biology https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0974-4

Version
104.3

14.10 fastp

Source code
GitHub https://github.com/OpenGene/fastp

Article
Bioinformatics https://doi.org/10.1093/bioinformatics/bty560

Version
0.23.2

14.11 fastqc

Source code
GitHub https://github.com/s-andrews/FastQC

Article
Babraham https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Version
0.11.9

14.12 gatk

Source code
Github https://github.com/broadinstitute/gatk

Article
Current Protocols in Bioinformatics https://pubmed.ncbi.nlm.nih.gov/25431634/

Version
3.8

14.13 genmod

Source code
Github https://github.com/Clinical-Genomics/genmod

Version
0.2.16

14.14 manta

Source code
GitHub https://github.com/Illumina/manta

Article
Bioinformatics https://doi.org/10.1093/bioinformatics/btv710

Version
1.6.0

14.15 mosdepth

Source code
GitHub https://github.com/brentp/mosdepth

Article
Bioinformatics https://doi.org/10.1093/bioinformatics/btx699

Version
0.3.3

14.16 multiqc

Source code
GitHub https://github.com/ewels/MultiQC

Article
Bioinformatics https://doi.org/10.1093/bioinformatics/btw354

Version
1.12

14.17 picard

Source code
GitHub https://github.com/broadinstitute/picard

Article
-

Version
2.27.1

14.18 sambamba

Source code
GitHub https://github.com/biod/sambamba

Article
Bioinformatics https://pubmed.ncbi.nlm.nih.gov/25697820/

Version
0.8.2

14.19 samtools

Source code
GitHub https://github.com/samtools/samtools

Article
Bioinformatics https://pubmed.ncbi.nlm.nih.gov/19505943/

Version
>1.11

14.20 sentieon-tools

Source code
Commercial Tool https://www.sentieon.com/

Article
Bioinformatics https://www.biorxiv.org/content/10.1101/115717v2

Version
202010.02

14.21 somalier

Source code
Github https://github.com/brentp/somalier

Article
Genome Medicine https://doi.org/10.1186/s13073-020-00761-2

Version
0.2.16

14.22 svdb

Source code
Github https://github.com/J35P312/SVDB

Article
F1000Res https://pubmed.ncbi.nlm.nih.gov/28781756/

Version
2.8.1

14.23 tabix

Source code
GitHub https://github.com/samtools/tabix

Article
Bioinformatics https://academic.oup.com/bioinformatics/article/27/5/718/262743

Version
>=1.11

14.24 tiddit

Source code
Github https://github.com/SciLifeLab/TIDDIT

Article
F1000Res https://pubmed.ncbi.nlm.nih.gov/28781756/

Version
3.3.2

14.25 vardict

Source code
GitHub https://github.com/AstraZeneca-NGS/VarDict

Article
Nucleic Acid Research https://pubmed.ncbi.nlm.nih.gov/27060149/

Version
2019.06.04

14.26 vcfanno

Source code
GitHub https://github.com/brentp/vcfanno

Article
Genome Biology https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0973-5/

Version
0.3.3

14.27 vcf2cytosure

Source code
GitHub https://github.com/NBISweden/vcf2cytosure

Article
-

Version
0.8

CHAPTER

FIFTEEN

REFERENCES AND OTHER RESOURCES

Main resources including knowledge base and databases necessary for pipeline development

1. **MSK-Impact pipeline**: https://www.mskcc.org/msk-impact
2. **TCGA**: https://cancergenome.nih.gov/
3. **COSMIC**: http://cancer.sanger.ac.uk/cosmic
4. **dbSNP**: Database of single nucleotide polymorphisms (SNPs) and multiple small-scale variations that include insertions/deletions, microsatellites, and non-polymorphic variants. https://www.ncbi.nlm.nih.gov/snp/ Download link: ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b150_GRCh38p7/VCF/All_20170710.vcf.gz
5. **ClinVar**: ClinVar aggregates information about genomic variation and its relationship to human health. https://www.ncbi.nlm.nih.gov/clinvar/ Download link: ftp://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/clinvar_20171029.vcf.gz
6. **SweGen**: This dataset contains whole-genome variant frequencies for 1000 Swedish individuals generated within the SweGen project. Download link: https://swefreq.nbis.se/
7. **ExAC**: The Exome Aggregation Consortium (ExAC) is a coalition of investigators seeking to aggregate and harmonize exome sequencing data from a wide variety of large-scale sequencing projects, and to make summary data available for the wider scientific community. http://exac.broadinstitute.org/ Download link: ftp://ftp.broadinstitute.org/pub/ExAC_release/release1/ExAC.r1.sites.vep.vcf.gz
8. **GTEx**: The Genotype-Tissue Expression (GTEx) project aims to provide to the scientific community a resource with which to study human gene expression and regulation and its relationship to genetic variation. https://www.gtexportal.org/home/ Download URL by applying through: https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs000424.v6.p1
9. **OMIM**: OMIM®, Online Mendelian Inheritance in Man®, An Online Catalog of Human Genes and Genetic Disorders. https://www.omim.org/ Download link: https://omim.org/downloads/ (registration required)
10. **Drug resistance**: An effort by Cosmic to annotate mutations identified in the literature as resistance mutations, including those conferring acquired resistance (after treatment) and intrinsic resistance (before treatment). Available through Cosmic: http://cancer.sanger.ac.uk/cosmic/drug_resistance
11. **Mutational signatures**: Signatures of Mutational Processes in Human Cancer. Available through Cosmic: http://cancer.sanger.ac.uk/cosmic/signatures
12. **DGVa**: The Database of Genomic Variants archive (DGVa) is a repository that provides archiving, accessioning and distribution of publicly available genomic structural variants, in all species. https://www.ebi.ac.uk/dgva
13. **Cancer genomics workflow**: MGI's CWL Cancer Pipelines. https://github.com/genome/cancer-genomics-workflow/wiki
14. **GIAB**: The priority of GIAB is authoritative characterization of human genomes for use in analytical validation and technology development, optimization, and demonstration. https://github.com/genome-in-a-bottle

15. **dbNSFP**: dbNSFP is a database developed for functional prediction and annotation of all potential non-synonymous single-nucleotide variants (nsSNVs) in the human genome. https://sites.google.com/site/jpopgen/dbNSFP

16. **1000Genomes**: The goal of the 1000 Genomes Project was to find most genetic variants with frequencies of at least 1% in the populations studied. http://www.internationalgenome.org/ Download link: ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/

17. **HapMap3**: The International HapMap Project was an organization that aimed to develop a haplotype map (HapMap) of the human genome, to describe the common patterns of human genetic variation. HapMap 3 is the third phase of the International HapMap project. http://www.sanger.ac.uk/resources/downloads/human/hapmap3.html Download link: ftp://ftp.ncbi.nlm.nih.gov/hapmap/

18. **GRCh38.p11**: GRCh38.p11 is the eleventh patch release for the GRCh38 (human) reference assembly. https://www.ncbi.nlm.nih.gov/grc/human Download link: ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/001/405/

19. **dbVar**: dbVar is NCBI's database of genomic structural variation – insertions, deletions, duplications, inversions, mobile element insertions, translocations, and complex chromosomal rearrangements https://www.ncbi.nlm.nih.gov/dbvar Download link: https://www.ncbi.nlm.nih.gov/dbvar/content/ftp_manifest/

20. **Drug sensitivity in cancer**: Identifying molecular features of cancers that predict response to anti-cancer drugs. http://www.cancerrxgene.org/ Download link: ftp://ftp.sanger.ac.uk/pub4/cancerrxgene/releases

21. **VarSome**: VarSome is a knowledge base and aggregator for human genomic variants. https://varsome.com/about/

22. **CADD**: CADD is a tool for scoring the deleteriousness of single nucleotide variants as well as insertion/deletions variants in the human genome. CADD can quantitatively prioritize functional, deleterious, and disease causal variants across a wide range of functional categories, effect sizes and genetic architectures and can be used prioritize causal variation in both research and clinical settings.

15.1 Sample datasets

1. **TCRB**: The Texas Cancer Research Biobank (TCRB) was created to bridge the gap between doctors and scientific researchers to improve the prevention, diagnosis and treatment of cancer. This work occurred with funding from the Cancer Prevention & Research Institute of Texas (CPRIT) from 2010-2014. http://txcrb.org/data.html Article: https://www.nature.com/articles/sdata201610

15.2 Relevant publications

Including methodological benchmarking

1. **MSK-IMPACT**:

   - **Original pipeline**: Cheng, D. T., Mitchell, T. N., Zehir, A., Shah, R. H., Benayed, R., Syed, A., … Berger, M. F. (2015). Memorial sloan kettering-integrated mutation profiling of actionable cancer targets (MSK-IMPACT): A hybridization capture-based next-generation sequencing clinical assay for solid tumor molecular oncology. Journal of Molecular Diagnostics, 17(3), 251–264. https://doi.org/10.1016/j.jmoldx.2014.12.006

   - **Case study**: Cheng, D. T., Prasad, M., Chekaluk, Y., Benayed, R., Sadowska, J., Zehir, A., … Zhang, L. (2017). Comprehensive detection of germline variants by MSK-IMPACT, a clinical diagnostic platform for solid tumor molecular oncology and concurrent cancer predisposition testing. BMC Medical Genomics, 10(1), 33. https://doi.org/10.1186/s12920-017-0271-4

- **Case study**: Zehir, A., Benayed, R., Shah, R. H., Syed, A., Middha, S., Kim, H. R., … Berger, M. F. (2017). Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10,000 patients. Nature Medicine, 23(6), 703–713. https://doi.org/10.1038/nm.4333

2. **Application of MSK-IMPACT**: Zehir, A., Benayed, R., Shah, R. H., Syed, A., Middha, S., Kim, H. R., … Berger, M. F. (2017). Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10,000 patients. Nature Medicine, 23(6), 703–713. https://doi.org/10.1038/nm.4333
3. **Review on bioinformatic pipelins**: Leipzig, J. (2017). A review of bioinformatic pipeline frameworks. Briefings in Bioinformatics, 18(3), 530–536. https://doi.org/10.1093/bib/bbw020
4. **Mutational signature reviews**:
   - Helleday, T., Eshtad, S., & Nik-Zainal, S. (2014). Mechanisms underlying mutational signatures in human cancers. Nature Reviews Genetics, 15(9), 585–598. https://doi.org/10.1038/nrg3729
   - Alexandrov, L. B., & Stratton, M. R. (2014). Mutational signatures: The patterns of somatic mutations hidden in cancer genomes. Current Opinion in Genetics and Development, 24(1), 52–60. https://doi.org/10.1016/j.gde.2013.11.014
5. **Review on structural variation detection tools**:
   - Lin, K., Bonnema, G., Sanchez-Perez, G., & De Ridder, D. (2014). Making the difference: Integrating structural variation detection tools. Briefings in Bioinformatics, 16(5), 852–864. https://doi.org/10.1093/bib/bbu047
   - Tattini, L., D'Aurizio, R., & Magi, A. (2015). Detection of Genomic Structural Variants from Next-Generation Sequencing Data. Frontiers in Bioengineering and Biotechnology, 3(June), 1–8. https://doi.org/10.3389/fbioe.2015.00092
6. **Two case studies and a pipeline (unpublished)**: Noll, A. C., Miller, N. A., Smith, L. D., Yoo, B., Fiedler, S., Cooley, L. D., … Kingsmore, S. F. (2016). Clinical detection of deletion structural variants in whole-genome sequences. Npj Genomic Medicine, 1(1), 16026. https://doi.org/10.1038/npjgenmed.2016.26
7. **Review on driver gene methods**: Tokheim, C. J., Papadopoulos, N., Kinzler, K. W., Vogelstein, B., & Karchin, R. (2016). Evaluating the evaluation of cancer driver genes. Proceedings of the National Academy of Sciences, 113(50), 14330–14335. https://doi.org/10.1073/pnas.1616440113
8. **Detection of IGH::DUX4 rearrangement**: Rezayee, F., Eisfeldt, J., Skaftason, A., Öfverholm, I., Sayyab, S., Syvänen, A. C., … & Barbany, G. (2023). Feasibility to use whole-genome sequencing as a sole diagnostic method to detect genomic aberrations in pediatric B-cell acute lymphoblastic leukemia. Frontiers in Oncology, 13, 1217712. https://doi.org/10.3389/fonc.2023.1217712

Resource, or general notable papers including resource and KB papers related to cancer genomics

1. **GIAB**: Zook, J. M., Catoe, D., McDaniel, J., Vang, L., Spies, N., Sidow, A., … Salit, M. (2016). Extensive sequencing of seven human genomes to characterize benchmark reference materials. Scientific Data, 3, 160025. https://doi.org/10.1038/sdata.2016.25

15.3 Methods and tools

Excluding multiple method comparison or benchmarking tools

- **BreakDancer**: Chen, K., Wallis, J. W., Mclellan, M. D., Larson, D. E., Kalicki, J. M., Pohl, C. S., … Elaine, R. (2013). BreakDancer - An algorithm for high resolution mapping of genomic structure variation. Nature Methods, 6(9), 677–681. https://doi.org/10.1038/nmeth.1363
- **Pindel**: Ye, K., Schulz, M. H., Long, Q., Apweiler, R., & Ning, Z. (2009). Pindel: A pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. Bioinformatics, 25(21), 2865–2871. https://doi.org/10.1093/bioinformatics/btp394

- **SVDetect**: Zeitouni, B., Boeva, V., Janoueix-Lerosey, I., Loeillet, S., Legoix-né, P., Nicolas, A., … Barillot, E. (2010). SVDetect: A tool to identify genomic structural variations from paired-end and mate-pair sequencing data. Bioinformatics, 26(15), 1895–1896. https://doi.org/10.1093/bioinformatics/btq293
- **Purityest**: Su, X., Zhang, L., Zhang, J., Meric-bernstam, F., & Weinstein, J. N. (2012). Purityest: Estimating purity of human tumor samples using next-generation sequencing data. Bioinformatics, 28(17), 2265–2266. https://doi.org/10.1093/bioinformatics/bts365
- **PurBayes**: Larson, N. B., & Fridley, B. L. (2013). PurBayes: Estimating tumor cellularity and subclonality in next-generation sequencing data. Bioinformatics, 29(15), 1888–1889. https://doi.org/10.1093/bioinformatics/btt293
- **ANNOVAR**: Wang, K., Li, M., & Hakonarson, H. (2010). ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Research, 38(16), 1–7. https://doi.org/10.1093/nar/gkq603
- **ASCAT**: Van Loo, P., Nordgard, S. H., Lingjaerde, O. C., Russnes, H. G., Rye, I. H., Sun, W., … Kristensen, V. N. (2010). Allele-specific copy number analysis of tumors. Proceedings of the National Academy of Sciences, 107(39), 16910–16915. https://doi.org/10.1073/pnas.1009843107
- **Treeomics**: Reiter, J. G., Makohon-Moore, A. P., Gerold, J. M., Bozic, I., Chatterjee, K., Iacobuzio-Donahue, C. A., … Nowak, M. A. (2017). Reconstructing metastatic seeding patterns of human cancers. Nature Communications, 8, 14114. https://doi.org/10.1038/ncomms14114
- **deconstructSigs**: Rosenthal, R., McGranahan, N., Herrero, J., Taylor, B. S., & Swanton, C. (2016). deconstructSigs: delineating mutational processes in single tumors distinguishes DNA repair deficiencies and patterns of carcinoma evolution. Genome Biology, 17(1), 31. https://doi.org/10.1186/s13059-016-0893-4
- **MutationalPatterns**: Blokzijl, F., Janssen, R., van Boxtel, R., & Cuppen, E. (2017). MutationalPatterns: comprehensive genome-wide analysis of mutational processes. bioRxiv, 1–20. https://doi.org/https://doi.org/10.1101/071761
- **MaSuRCA**: Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. Bioinformatics, 29(21), 2669–2677. https://doi.org/10.1093/bioinformatics/btt476
- **VarDict**: Lai, Z., Markovets, A., Ahdesmaki, M., Chapman, B., Hofmann, O., Mcewen, R., … Dry, J. R. (2016). VarDict: A novel and versatile variant caller for next-generation sequencing in cancer research. Nucleic Acids Research, 44(11), 1–11. https://doi.org/10.1093/nar/gkw227
- **vt**: Tan, A., Abecasis, G. R., & Kang, H. M. (2015). Unified representation of genetic variants. Bioinformatics, 31(13), 2202–2204. https://doi.org/10.1093/bioinformatics/btv112
- **peddy**: Pedersen, B. S., & Quinlan, A. R. (2017). Who's Who? Detecting and Resolving Sample Anomalies in Human DNA Sequencing Studies with Peddy. American Journal of Human Genetics, 100(3), 406–413. https://doi.org/10.1016/j.ajhg.2017.01.017
- **GQT**: Layer, R. M., Kindlon, N., Karczewski, K. J., & Quinlan, A. R. (2015). Efficient genotype compression and analysis of large genetic-variation data sets. Nature Methods, 13(1). https://doi.org/10.1038/nmeth.3654

Tool sets and softwares required at various steps of pipeline development

1. **FastQC**: Quality control tool. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
2. **Cutadapt**: Adapter removal tool. https://cutadapt.readthedocs.io/en/stable/
3. **Trim Galore!**: FastQC and Cutadapt wrapper. https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
4. **Picardtools**: BAM/SAM/VCF/CRAM manipulator. http://broadinstitute.github.io/picard/
    - **MarkDuplicate**: Mark duplicate reads and potentially remove them
    - **LiftoverVcf**: liftover VCF between builds

- **CollectHsMetric**: Collects hybrid-selection (HS) metrics for a SAM or BAM file
- **CollectAlignmentSummaryMetrics**: Produces a summary of alignment metrics from a SAM or BAM file
- **CollectGcBiasMetrics**: Collect metrics regarding GC bias
- **CollectWgsMetrics**: Collect metrics about coverage and performance of whole genome sequencing (WGS) experiments

5. **GATK**: A variant discovery tool: https://gatk.broadinstitute.org/hc/en-us
   - **BaseRecalibrator**: Detect systematic error in base quality score
   - **Somatic Indel Realigner**: Local Realignment around Indels
   - **ContEst**: Estimate cross sample contamination
   - **DepthOfCoverage**: Assess sequence coverage by sample, read group, or libraries
   - **DuplicateReadFilter**: remove duplicated from flag set by MarkDuplicates
6. **Samtools**: Reading/writing/editing/indexing/viewing SAM/BAM/CRAM format http://www.htslib.org/
7. **Sambamba**: Tools for working with SAM/BAM/CRAM data http://lomereiter.github.io/sambamba/
8. **bcftools**: Reading/writing BCF2/VCF/gVCF files and calling/filtering/summarising SNP and short indel sequence variants http://www.htslib.org/doc/bcftools.html
9. **vcftools**: VCFtools is a program package designed for working with VCF files, such as those generated by the 1000 Genomes Project. https://vcftools.github.io/index.html
10. **Delly2**: An integrated structural variant prediction method that can discover, genotype and visualize deletions, tandem duplications, inversions and translocations https://github.com/dellytools/delly
11. **PLINK**: PLINK: Whole genome data analysis toolset https://www.cog-genomics.org/plink2
12. **freebayes**: a haplotype-based variant detector. https://github.com/ekg/freebayes
13. **AscatNGS**: Allele-Specific Copy Number Analysis of Tumors, tumor purity and ploidy https://github.com/cancerit/ascatNgs
14. **MutationalPatterns**: R package for extracting and visualizing mutational patterns in base substitution catalogues https://github.com/UMCUGenetics/MutationalPatterns
15. **desconstructSigs**: identification of mutational signatures within a single tumor sample https://github.com/raerose01/deconstructSigs
16. **treeOmics**: Decrypting somatic mutation patterns to reveal the evolution of cancer https://github.com/johannesreiter/treeomics
17. **controlFreeC**: Copy number and allelic content caller http://boevalab.com/FREEC/
18. **MuTect2**: Call somatic SNPs and indels via local re-assembly of haplotypes https://gatk.broadinstitute.org/hc/en-us/articles/360037593851-Mutect2
19. **Annovar**: annotation of detected genetic variation http://annovar.openbioinformatics.org/en/latest/
20. **Strelka**: Small variant caller https://github.com/Illumina/strelka
21. **Manta**: Structural variant caller https://github.com/Illumina/manta
22. **PurBayes**: estimate tumor purity and clonality
23. **VarDict**: variant caller for both single and paired sample variant calling from BAM files https://github.com/AstraZeneca-NGS/VarDict
24. **SNPeff/SNPSift**: Genomic variant annotations and functional effect prediction toolbox. http://snpeff.sourceforge.net/ and http://snpeff.sourceforge.net/SnpSift.html

25. **IGV**: visualization tool for interactive exploration http://software.broadinstitute.org/software/igv/
26. **SVDetect**: a tool to detect genomic structural variations http://svdetect.sourceforge.net/Site/Home.html
27. **GenomeSTRiP**: A suite of tools for discovering and genotyping structural variations using sequencing data http://software.broadinstitute.org/software/genomestrip/
28. **BreakDancer**: SV detection from paired end reads mapping https://github.com/genome/breakdancer
29. **pIndel**: Detect breakpoints of large deletions, medium sized insertions, inversions, and tandem duplications https://github.com/genome/pindel
30. **VarScan**: Variant calling and somatic mutation/CNV detection https://github.com/dkoboldt/varscan
31. **VEP**: Variant Effect Predictor https://www.ensembl.org/info/docs/tools/vep/index.html
32. **Probablistic2020**: Simulates somatic mutations, and calls statistically significant oncogenes and tumor suppressor genes based on a randomization-based test https://github.com/KarchinLab/probabilistic2020
33. **2020plus**: Classifies genes as an oncogene, tumor suppressor gene, or as a non-driver gene by using Random Forests https://github.com/KarchinLab/2020plus
34. **vtools**: variant tools is a software tool for the manipulation, annotation, selection, simulation, and analysis of variants in the context of next-gen sequencing analysis. https://vatlab.github.io/vat-docs/
35. **vt**: A variant tool set that discovers short variants from Next Generation Sequencing data. https://genome.sph.umich.edu/wiki/Vt and https://github.com/atks/vt
36. **CNVnator**: a tool for CNV discovery and genotyping from depth-of-coverage by mapped reads. https://github.com/abyzovlab/CNVnator
37. **CNVpytor**: a tool for copy number variation detection and analysis from read depth and allele imbalance in whole-genome sequencing. https://github.com/abyzovlab/CNVpytor
38. **SvABA**: Structural variation and indel detection by local assembly. https://github.com/walaj/svaba
39. **indelope**: find indels and SVs too small for structural variant callers and too large for GATK. https://github.com/brentp/indelope
40. **peddy**: peddy compares familial-relationships and sexes as reported in a PED/FAM file with those inferred from a VCF. https://github.com/brentp/peddy
41. **cyvcf2**: cyvcf2 is a cython wrapper around htslib built for fast parsing of Variant Call Format (VCF) files. https://github.com/brentp/cyvcf2
42. **GQT**: Genotype Query Tools (GQT) is command line software and a C API for indexing and querying large-scale genotype data sets. https://github.com/ryanlayer/gqt
43. **LOFTEE**: Loss-Of-Function Transcript Effect Estimator. A VEP plugin to identify LoF (loss-of-function) variation. Assesses variants that are: Stop-gained, Splice site disrupting, and Frameshift variants. https://github.com/konradjk/loftee
44. **PureCN**: copy number calling and SNV classification using targeted short read sequencing https://bioconductor.org/packages/release/bioc/html/PureCN.html
45. **SVCaller**: A structural variant caller. https://github.com/tomwhi/svcaller
46. **SnakeMake**: A workflow manager. http://snakemake.readthedocs.io/en/stable/index.html
47. **BWA**: BWA is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM. http://bio-bwa.sourceforge.net/

48. **wgsim**: Wgsim is a small tool for simulating sequence reads from a reference genome. It is able to simulate diploid genomes with SNPs and insertion/deletion (INDEL) polymorphisms, and simulate reads with uniform substitution sequencing errors. https://github.com/lh3/wgsim
49. **dwgsim**: Whole genome simulation can be performed with dwgsim. dwgsim is based off of wgsim found in SAMtools. https://github.com/nh13/DWGSIM
50. **THetA**: Tumor Heterogeneity Analysis. This algorithm estimates tumor purity and clonal/subclonal copy number aberrations directly from high-throughput DNA sequencing data. https://github.com/raphael-group/THetA
51. **Skewer**: Adapter trimming, similar to cutadapt. https://github.com/relipmoc/skewer
52. **Phylowgs**: Application for inferring subclonal composition and evolution from whole-genome sequencing data. https://github.com/morrislab/phylowgs
53. **superFreq**: SuperFreq is an R package that analyses cancer exomes to track subclones. https://github.com/ChristofferFlensburg/superFreq
54. **readVCF-r**: Read VCFs into R and annotatte them. https://bioconductor.org/packages/release/bioc/html/VariantAnnotation.html
55. **vcfr**: Read VCFs into R. https://github.com/knausb/vcfR
56. **msisensor**: microsatellite instability detection using paired tumor-normal https://github.com/ding-lab/msisensor
57. **MOSAIC**: MicrOSAtellite Instability Classifier https://github.com/ronaldhause/mosaic
58. **MANTIS**: Microsatellite Analysis for Normal-Tumor InStability https://github.com/OSU-SRLab/MANTIS
59. **SBDB**: A toolkit for constricting and querying structural variant databases https://github.com/J35P312/SVDB

CHAPTER

SIXTEEN

DOCUMENTATION GUIDLINE

BALSAMIC uses Sphinx to build the documentation, see the official documentation of Sphinx: https://www.sphinx-doc.org/en/master/index.html

Following steps explains how to build documents locally.

Create a conda environment:

```
conda create -n balsamic_doc -c bioconda -c conda-forge python=3.11 pip pygraphviz␣
↪wkhtmltopdf
conda activate balsamic_doc
```

Install Sphinx and extensions:

```
cd /path/to/BALSAMIC
python -m pip install --upgrade --upgrade-strategy eager --no-cache-dir .
cd docs
pip install -r requirements.txt -r ../requirements-dev.txt
```

Build docs:

```
sphinx-build -T -E -b html -d _build/doctrees-readthedocs -D language=en . _build/html
```

View docs (`open` or similar command from your OS):

```
open _build/html/index.html
```

CHAPTER

SEVENTEEN

CODING ETIQUETTES

- Structure the code properly
- Maintain good and consistent naming convention
- Keep it simple
- Don't repeat yourself

17.1 Git etiquette

17.1.1 Code formatting

BALSAMIC is using Black as code formatter: https://github.com/psf/black

17.1.2 Conventional commits and PRs

PRs should follow the following keywords in the title: https://www.conventionalcommits.org/en/v1.0.0/

Commit messages are recommended to following the following similar to PRs:

1. **feat**: Introducing a new features. This includes but not limited to workflows, SnakeMake rule, cli, and plugins. In other words, anything that is new and fundamental change will also go here. Enhancements and optimizations will go into refactor.
2. **fix**: This is essentially a patch. Included but not limited to: bug fixes, hotfixes, and any patch to address a known issue.
3. **doc**: Any changes to the documentation are part of doc subject line, included but not limited to docstrings, cli-help, readme, tutorial, documentation, CHANGELOG, and addition of ipython/jupyter notebook in the form of tutorial.
4. **test**: Any changes to the tests are part of test subject line. This includes adding, removing or updating of the following: unittests, validation/verification dataset, and test related configs.
5. **refactor**: Refactoring refers to a rather broad term. Any style changes, code enhancement, and analysis optimization.
6. **version**: Any changes to .bumpversion config and or change of version will be specified with this. This includes comments within .bumpversion, structure of .bumpversion, etc.

17.1.3 Scope

Scope is specified within parenthesis. It show the *scope* of the subject line. The following scope are valid:

- cli
- style
- rule (refers to SnakeMake rules)
- workflow (refer to SnakeMake workflows)
- config (refers to configs that are either used or generated by BALSAMIC)
- Relevant scopes that might fit into a scope description

Note: If scope is broad or matching with multiple (it shouldn't, but if it does) one can leave out the scope.

17.1.4 Message

It's better to start Git commit message with the following words:

- added
- removed
- updated

17.2 Snakemake etiquette

The bioinformatics core analysis in BALSAMIC is defined by set of rules written as a Snakemake rules (`*.rule`) and Snakemake workflow as (`*.smk`). Main `balsamic.smk` workflow uses these rules to create sets of output files from sets of input files. Using `{wildcards}` Snakemake can automatically determine the dependencies between the rules by matching file names. The following guidelines describe the general conventions for naming and order of the rules, while writing a Snakemake file in BALSAMIC. For further description of how Snakemake works, please refer to Snakemake official documentation: https://snakemake.readthedocs.io/

17.2.1 Structure of Snakemake rules

```
rule <program>_<function>_<optional_tag>_<optional_tag>:
    input:
        <named_input_1> = ...,
        <named_input_2> = ...,
    output:
        <named_output_1> = ...,
    benchmark:
        Path(benchmark_dir, "<rule_name>_<{sample}/{case_name}>.tsv").as_posix()
    singularity:
        singularity_image
    params:
        <named_param_1> = ...,
        <named_param_1> = ...,
    threads:
        get_threads(cluster_config, '<rule_name>')
```

(continues on next page)

(continued from previous page)

```
    message:
        ("Align fastq files with bwa-mem to reference genome and sort using samtools for␣
↪sample: {sample}"
        "<second line is allowed to cover more description>")
    shell:
         """
    <first_command> <options>;

    <second_command> <options>;

    <a_long_command> <--option-1> <value_1> \
    <--option-2> <value_2> \
    <--option-3> <value_3>;
        """
```

Descriptions

rulename: Rule name briefly should outline the program and functions utilized inside the rule. Each word is seperated by a underscore _. First word is the bioinformatic tool or script's name. The following words describe subcommand within that bioinformatic tool and then followed by workflow specific description. The word length shouldn't exceed more than 3 or 4 words. Make sure rule names are updated within `config/cluster.json` and it is all lowercase. Examples: `picard_collecthsmetrics_umi`, `bcftools_query_calculateaftable_umi`

input: It is strongly recommended to set input and output files as named. Refrain from introducing new wildcards as much as possible.

output: This should follow the same instructions as `input`.

benchmark: Benchmark name is prefixed with rule name and suffixed with '.tsv' file extension.

singularity: Make sure the singularity image does contain a Conda environment with required bioinformatics tools. Do not use this field if `run` is used instead of `shell`.

params: If the defined parameter is a threshold or globally used constant; add it to `utils/constants.py`. Respective class models need to be updated in `utils/models.py`.

threads: Make sure for each rule, the correct number of threads are assigned in `config/cluster.json`. Otherwise it will be assigned default values from `config/cluster.json` . If there is no need for multithreading, this field can be removed from rule.

message: A short message describing the function of rule. Add any relevant wildcard to message to make it readable and understandable. It is also recommended to use `params` to build a more descriptive `message`

shell (run): Code inside the *shell/run* command should be left indented. Shell lines no longer than 100 characters. Break the long commands with \ and followed by a new line. Avoid having long Python code within `run`, instead add it to `utils/` as a Python script and import the function.

Example:

```
java -jar \
-Djava.io.tmpdir=${{tmpdir}} \
-Xms8G -Xmx16G \
$CONDA_PREFIX/share/picard.jar \
MarkDuplicates \
{input.named_input_1} \
{output.named_output_1};
```

Example for external python scripts that can be saved as modules in `utils/*.py` and can use them as definitions in rules as:

```
from BALSAMIC.utils.workflowscripts import get_densityplot
get_densityplot(input.named_input1, params.named_params_1, output.named_output1 )
```

Similarly `awk` or `R` external scripts can be saved in `assets/scripts/*awk` and can be invoked using *get_script_path* as:

```
params:
    consensusfilter_script = get_script_path("FilterDuplexUMIconsensus.awk")
shell:
     """
samtools view -h {input} | \
awk -v MinR={params.minreads} \
-v OFS=\'\\t\' -f {params.consensusfilter_script} | \
samtools view -bh - > {output}
     """
```

References

1. https://snakemake.readthedocs.io/en/stable/snakefiles/rules.html
2. https://snakemake.readthedocs.io/en/stable/snakefiles/writing_snakefiles.html

17.3 Container etiquette

BALSAMIC uses singularity containers to perform the bioinformatics analysis. These containers are built using Docker and pushed to Docker Hub. For more details on building containers using docker, please refer to the official docker documentation: https://docs.docker.com/

17.3.1 Structure of Docker recipe

```
FROM <CONTAINER>:<VERSION>

LABEL base.image="<CONTAINER>:<VERSION>"
LABEL maintainer="Clinical Genomics"
LABEL about.contact="support@clinicalgenomics.se"
LABEL software="<NAME_OF_THE_MAIN_SOFTWARE>"
LABEL software.version="<VERSION_OF_THE_MAIN_SOFTWARE>"
LABEL about.summary="<DESCRIPTION_OF_THE_MAIN_SOFTWARE>"
LABEL about.home="<URL_OF_THE_MAIN_SOFTWARE>""
LABEL about.documentation="<DOCS_URL_OF_THE_MAIN_SOFTWARE>"
LABEL about.license="MIT License (MIT)"

RUN apt-get update && apt-get -y upgrade && \
    apt-get -y install --no-install-recommends && \
    <SOFTWARE_1 SOFTWARE_2> && \
    apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
```

(continues on next page)

(continued from previous page)

```
RUN ....

USER    ubuntu
WORKDIR /home/ubuntu
CMD ["/bin/bash"]
```

It is preferable to:

- Use official image as the base
- Use Ubuntu-LTS as the base image
- Avoid Conda unless necessary
- Add versions
- Avoid building containers with multiple software used in the rules

CHAPTER

EIGHTEEN

SEMANTIC VERSIONING

BALSAMIC is following Semantic Versioning.

Since October 24, 2018 the following changes were added in addition to SemVer also to cover Bioinformatic and data analysis aspect of it:

- **major**:
  - Structural changes to the BALSAMIC workflow. This includes reordering of annotation softwares or sources, variant callers, aligners, quality trimmers, and/or anything other than QC reporting and rule *all*.
  - Addition of annotation softwares or sources, variant callers, aligners, quality trimmers, and/or anything other than QC reporting
- **minor**:
  - Under the hood changes to rules that won't affect output results of workflow.
  - Addition of new bioinfo tools for QC reporting.
  - Updating version of a Bioinformatic software or data resource (including annotation sources)
- **patch**:
  - Any bug fix and under the hood changes that won't impact end-users run.
  - Changes to resource allocation of Scheduler job submission

The rational for versioning is heavily inspired from BACTpipe: DOI: 10.5281/zenodo.1254248 and https://github.com/ctmrbio/BACTpipe)

CHAPTER

NINETEEN

FREQUENTLY ASKED QUESTIONS (FAQS)

19.1 BALSAMIC

19.2 UMIworkflow

What are UMIs

- Unique Molecular Identifiers (UMIs) are short random nucleotide sequences (3-20 bases) that are ligated to the ends of DNA fragments prior to sequencing to greatly reduce the impact of PCR duplicates and sequencing errors on the variant calling process.

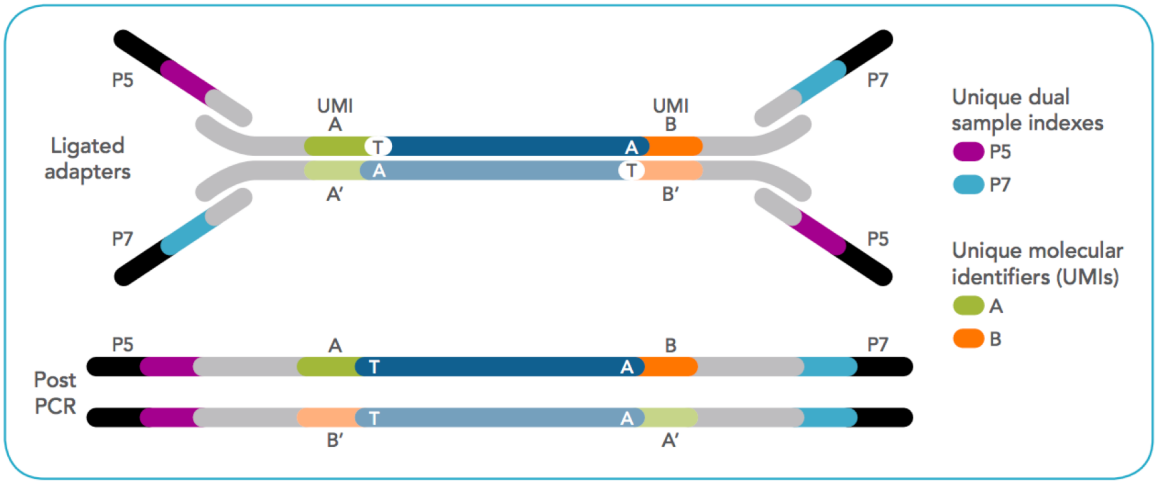


Figure 1. xGen Duplex Seq Adapters enable use of genetic information embedded in both DNA strands. xGen Duplex Seq Adapters are designed to be compatible with standard library construction methods. Adapters contain 3′-dTMP overhangs, which are ligated to 3′-dA-tailed inserts. xGen Duplex Seq Adapters incorporate degenerate UMIs to tag double-sequencing molecules in the ligation step and unique, dual sample indexes are incorporated by PCR amplification. During post-sequencing analysis, the top and bottom strands of original DNA input molecules can be paired back together to help identify errors introduced in the workflow.

Fig. 1: Figure1: Design of UMI adapters in the library preparation. Ref

How is the UMIworkflow implemented

- CG's UMIworkflow is implemented using the commercial software Sentieon. The Sentieon tools provide functionality for extracting UMI tags from fastq reads and performing barcode-aware consensus generation. The workflow is as described:

How is the UMI structure defined

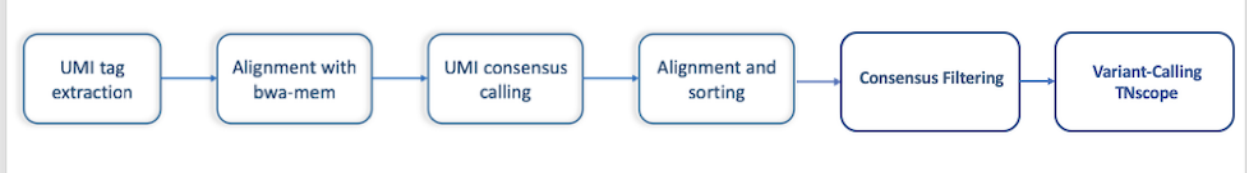


Fig. 2: Figure2: UMI workflow steps.

Our pair-end sequencing read length is about 151 bp and the UMI structure is defined as`3M2S146T, 3M2S146T` where *3M* represents 3 UMI bases, *2S* represents 2 skipped bases, *146T* represents 146 bases in the read.

Are there any differences in the UMI read extraction if the read structure is defined as `3M2S146T, 3M2S146T` or `3M2S+T, 3M2S+T`?

In theory, this should be the same if the read length is always 151bp. But the recommendation is to use *3M2S+T, 3M2S+T* so that UMIworkflow can handle any unexpected input data.

How does the `umi extract` tool handle sequencing adapters? Do the input reads always need to be adapter removed fastq reads

The presence of 5' adapter sequences can cause issues for the Sentieon *umi extract* tool, as the extract tool will not correctly identify the UMI sequence. If 5' adapter contamination is found in the data, before processing with the *umi extract* tool, these adapter sequences needed to be removed with a third-party trimming tool. 3' adapter contamination is much more common and can occur when the insert size is shorter than the sequence read length. The Sentieon *umi consensus* tool will correctly identify and handle 3' adapter/barcode contamination during consensus read creation.

How does Sentieon `umi consensus` tool handles paired-end reads

The *umi consensus* tool will merge overlapping read pairs when it can, but it is not possible for reads with an insert size greater than 2x the read length as there is some unknown intervening sequence. In this case, *umi consensus* will output a consensus read pair where each consensus read in the pair is constructed separately, while other reads in the dataset are collapsed/merged to single-end reads.

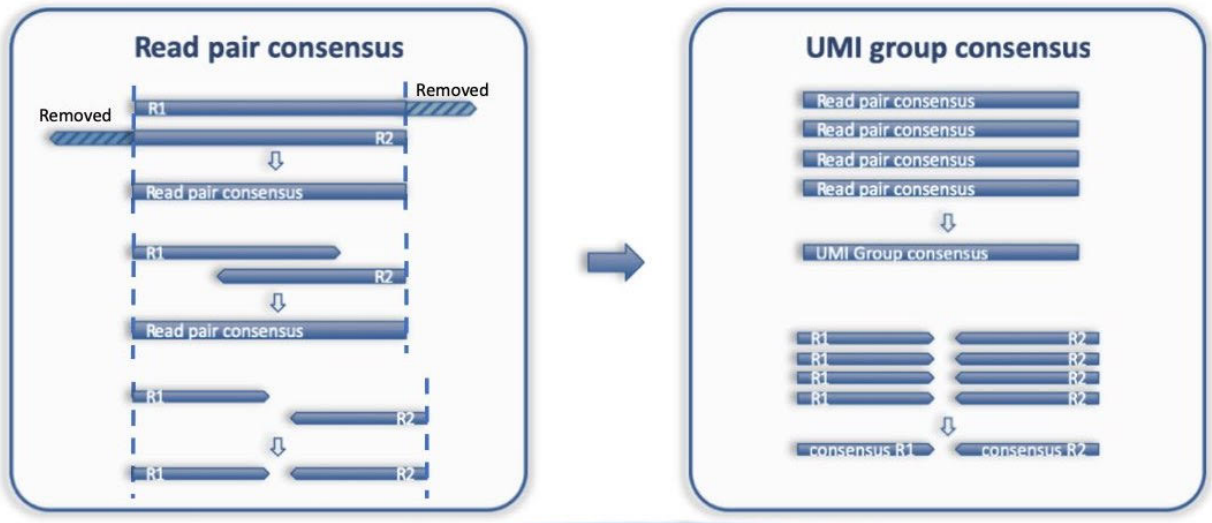


Fig. 3: Figure3: Figure taken from Sentieon document.

Purpose of consensus-filtering step in the UMIworkflow

Mainly to reduce the calling of false-positive variants. Consensus filtering is based on the setting of minimum raw reads (MinR) supporting each UMI group. By default, *MinR* is set as 3,1,1, meaning that the minimum number of

raw reads in both strands should be greater than 1 and the sum of both strands is greater than 3. The default *3,1,1* is a good starting point at lower coverages. This setting can be further adjusted accordingly at higher coverages or if finding false-positive calls due to consensus reads with little read support.

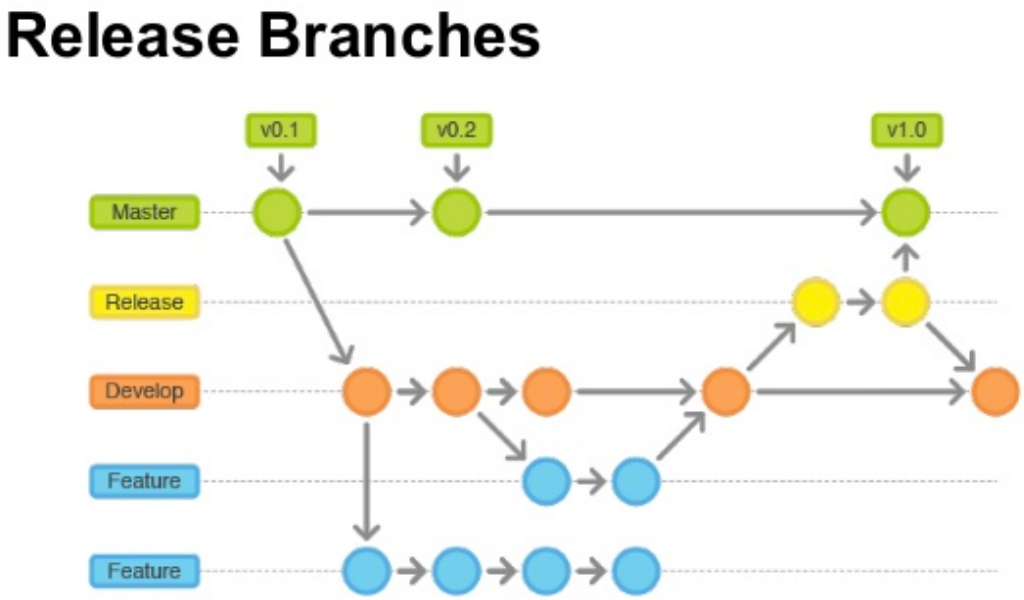
How is the performance of other variant callers for analysing UMI datasets UMI workflow is validated with two datasets (SeraCare and HapMap). The Vardict failed to call the true reference variants while the TNscope performed better. A more detailed analysis is summarized here

We are still investigating other UMI-aware variant callers and maybe in the future, if something works better, additional varcallers will be added to the UMIworkflow.

19.3 Git Related Questions

How to make a new release of balsamic

Our release model looks like in the figure below:



If changes are hotfixes/patches, make a release from the *master* branch. Otherwise, make a release from the *develop* branch.

Release from *develop*:

1. Switch to the master branch *git checkout master*.
2. Pull the latest changes *git pull*.
3. Switch to the *develop* branch: *git checkout develop*.
4. Merge the latest changes from master into develop: *git merge master*.
5. Resolve any conflicts, commit the changes, and push them to the develop branch.
6. Create a new *release* branch: *git checkout -B release_vX.X.X*. Adhere strictly to the naming convention for compatibility with container publishing GitHub Actions. Failure to do so may result in the unavailability of containers for download in this particular version.

7. Update the version number *X.X.X* in the *CHANGELOG.rst*.
8. Open a pull request for the *release* branch, and after obtaining approval, merge it into *master*.
9. Move back to *master* branch: *git checkout master*.
10. Fetch the latest changes: *git pull*.
11. Use *bumpversion* to increment the version (choose *major*, *minor*, or *patch*): *bumpversion –verbose [major/minor/patch]*.
12. Push the version increment changes: *git push*.
13. Push the tags for the new version: *git push –tags*.
14. Merge the changes into the *develop* branch (*git checkout develop && git merge master*).
15. Use the release branch to verify/validate the new version.
16. Once the verification/validation process is successfully completed and approved, proceed with the deployment to production.

Note

- Never force rebase commits into *master* or *develop*.
- For pull requests to *master*, opt for *Create a merge commit* to capture full commit history.
- For pull requests to *develop*, use *Squash and merge* to combine commit messages.

INDEX

Symbols

B