# ACEseqDocs Documentation

## *Release 1.2.8*

**Kortine Kleinheinz**

**Jan 13, 2022**

# Contents

# CHAPTER 1

# License

The license of the ACEseq code is *MIT <https://github.com/eilslabs/ACEseqWorkflow/blob/github/LICENSE.txt>*. See *here <https://github.com/eilslabs/ACEseqWorkflow/blob/github/package_LICENSES.md>* licenses of packages used by the workflow.

# CHAPTER 2

# Need help?

In case of question pleae contact Kortine Kleinheinz (k.kleinheinz@dkfz-heidelberg.de)

## CHAPTER 3

# Installation & Run instructions

To run the ACEseq-workflow multiple components are needed:

- The Roddy workflow management framework
- ACEseqWorkflow
- COWorkflowsBasePlugin
- PluginBase
- DefaultPlugin
- Bioinformatic software stack
- Reference data

Section *The Manual Way* describes how these components are manually installed, which gives you the most flexibility. This is probably mostly relevant if you have to run multiple different Roddy-based workflows and in a development setup.

There is an installation with pre-packaged files of ACEseqWorkflow 1.2.10 (see *here*). This installation is mostly interesting for historical reasons and considered "legacy". We can probably not help you much with this installation, if you encounter problems.

Finally, it is possible to run ACEseq in a container. Currently, all available containers wrap the complete workflow including the workflow manager Roddy and even a batch processing system (Sun GridEngine) in a single container. This is done because the workflow manager Roddy does not support submitting containerized cluster jobs. The disadvantage is that the one-size-fits all container for all the cluster jobs may have the wrong size for much of the ACEseq analysis.

There are different variants of these containers:

- There is a set of containers using versions 1.2.8 and 1.2.10, as described in the ACEseq article. See section *Legacy Docker versions*.
- An updated set of containers based on the Conda environment (see *here*).
- ACEseq v1.0.189 was used in the PanCancer Analysis of Whole Genomes (PCAWG) and is part of the PCAWG container.

## 3.1 The Manual Way

### 3.1.1 Roddy Installation

The file `buildinfo.txt` in the ACEseq repository shows you the Roddy API version that you need. We suggest you use the Roddy version with a matching major version number and the highest released minor number. For instance, if the plugin requires Roddy 3.0, use e.g. Roddy 3.6.0. To install that Roddy version please follow the instructions at the Roddy repository or the documentation. Note that the installation also requires you to install the "PluginBase" plugin and the "DefaulPlugin" in the versions stated in the `buildinfo.txt` files of the plugins in the `dist/plugins/` directory of Roddy installation.

With a Roddy installation you can install the ACEseq workflow and its dependencies. To manually resolve the plugin's versions you again need to look at the `buildinfo.txt`. You start from the ACEseqWorkflow plugin and recurse to its dependencies. Usually, you will probably want to install the released versions of these plugins from the zip-archives that can be found in the Github repositories (e.g. for ACEseq) of the respective plugin or component, but you can also use the cloned repositories with the correct release tag (or just commit) checked out.

Note that the ACEseqWorkflow and the COWorkflowBasePlugin can be installed in any directory, as long as all subdirectories there match the pattern for plugin directories of Roddy. So ideally this directory should only contain installations of plugins.

### 3.1.2 Plugin Installation

1. Download the ACEseqWorkflow https://github.com/DKFZ-ODCF/ACEseqWorkflow/tags. Version 1.2.8-4 is used in our production systems. GRCh38/hg38 support is in development and will be available only with version 6. Extract the archive into a directory called "ACEseqWorkflow_$version" with *$version* being the correct version number.
2. Look up the version of the COWorkflowsBasePlugin listed in the "buildinfo.txt" at the root of the ACEseqWorkflow plugin that you just downloaded. Download that version from https://github.com/DKFZ-ODCF/COWorkflowsBasePlugin/tags. Create a "COWorkflowsBasePlugin_$version" directory.

You can create the COWorkflowsBasePlugin and ACEseqWorkflow directories in the *dist/plugins* directory, which was extracted with the RoddyEnv ZIP.

### 3.1.3 Software Stack (Conda)

The software stack is the software that the workflow jobs need to do the bioinformatic work. If you run the workflow on a cluster you need to ensure that the software stack is also available on all cluster nodes that execute jobs. Unfortunately, we do not have ACEseq in a workflow manager that supports job execution in containers.

You can try to rebuild the Conda environment based on the `resources/analysisTools/copyNumberEstimationWorkflow/environments/conda.yml` file. Note, however, that some packages may not be available in the referenced Conda channels anymore. Consequently, the following setup based on the YAML file is really more for the developer:

1. Set up the required channels:

```
conda config --add channels r
conda config --add channels defaults
conda config --add channels conda-forge
conda config --add channels bioconda
```

2. Rebuild the environment

```
cd $PATH_TO_PLUGIN_DIRECTORY
conda env create \
    -n ACEseqWorkflow_1.2.8-4 \
    -f resources/analysisTools/copyNumberEstimationWorkflow/environments/conda.yml
```

The name of the Conda environment is arbitrary but needs to be consistent with the `condaEnvironmentName` variable. You can set the `condaEnvironmentName` variable in any of the loaded configuration files (see Roddy documentation) or even directly in your Roddy call via `--cvalues="condaEnvironmentName:$value"`.

Given the problems with old packages in some Conda channels, we offer a work-around. For reproducibility the software stack has been stored with Conda Pack in an archive.

1. Download the appropriate archive from old-server (e.g. `ACEseqWorkflow_1.2.8-4_conda_4.10.3_x86.tgz`)
2. **Unpack and set up the environment**

   ```
   mkdir $envBaseDir/ACEseqWorkflow_1.2.8-4    # The directory name is the␣
   ↪environment name.
   cd $envBaseDir/ACEseqWorkflow_1.2.8-4
   source bin/activate
   conda unpack
   ```

Now, if you do `conda env list` the environment should be listed. If not make sure you installed the environment into `$CONDA_PREFIX/envs/` or another environments directory can be configured for your Conda installation.

## 3.2 Docker version

Different versions of the ACE-seq workflow have been packaged with other workflows and the reference data. These containers have the required software-stacks installed but run all workflow jobs within the same container. To download these pipelines and reference data see ppcg-server. Instructions for the installation are given in the archives.

## 3.3 Reference files

The workflow uses various files as reference files, such as a reference genome or annotation files. We provide installation scripts in the *installation/* directory. To download and prepare the reference files please check out the ACEseq repository and do

```
bash $PATH_TO_PLUGIN_DIRECTORY/installation/downloadReferences $targetDirectory
```

with *$targetDirectory* being the directory into which you want to install the files. The variable *baseDirectoryReference* in your configurations needs to be set to the *$targetDirectory* path.

Note that the current plugin version is tuned to be run on the hg19 human assembly, but a liftover of all files should probably enable a run on GRch38.

### 3.3.1 Alternative reference files

The reference data can also be downloaded from the ppcg-server.

## 3.4 Legacy Installations

The following installation approaches are kept in the documentation for historical reasons.

### 3.4.1 Prepackaged files (ACEseq 1.2.10 only)

On old-server you can find archives for the 1.2.10 plugin version. The prepackaged zip files contains a full Roddy / Plugin setup and include different scripts to install all necessary software and download the required reference files. Currently, we do not intent to update these prepackaged installation files or the Docker version. Note that the Roddy version packaged is not capable of submitting to LSF. Only Torque/PBS is supported.

### 3.4.2 Stand-alone Roddy for Execution on HTC Cluster

To run the Roddy-based version of ACEseq please download the pre-packed zip file from the old-server. Three steps are required to ensure running of ACEseq.

1. Run the `prepareRoddyInstallation.sh` script.
2. Download all reference files as specified in the section "Reference files" (below).
3. Set up the Conda environment or install the necessary software as specified in the section "Software" (below).

Before running ACEseq a few parameters need to be adjusted in the configuration files. The output directory is specified in `$PATH_TO_ACEseq_RODDY_VERSION/configurations/projectsACEseqTest.xml`. Here the variables `baseDirectoryReference`, `inputBaseDirectory`, `outputBaseDirectory`, `outputAnalysisBaseDirectory` need to be set. If no SVs should be included the following configuration values (`cvalues`) should be included:

```
<cvalue name='runWithSv' value='false' type="boolean"/>
<cvalue name='SV' value='false' type="boolean"/>
```

If you set these values to "true", then "svOutputDirectory" and the SV bedpe filename in the filenames section need to be set, but the SV calls are then used for improved CNV calling.

```
<configurationvalues>
  <cvalue name='svOutputDirectory' value='${outputAnalysisBaseDirectory}/
↪nameOfDirectoryWithSVResults' type="path"/>
</configurationvalues>

<filenames package='de.dkfz.b080.co.files' filestagesbase='de.dkfz.b080.co.files.
↪COFileStage'>
   <filename class="TextFile" onMethod="de.dkfz.b080.co.aceseq.ACESeqMethods.mergeSv"
            selectiontag="svFileTag"
            pattern='${svOutputDirectory}/${pid}_svs.bedpe'/>
</filenames>
```

Technical specifications are set in the file `$PATH_TO_ACEseq_RODDY_VERSION/configurations/applicationProperties.ini`. The path to the project.xml and the path to the plugins (`$PATH_TO_ACEseq_RODDY_VERSION/Roddy/dist/plugins/`) need to be set under `configurationDirectories` and `pluginDirectories`. Finally the job manager and execution host need to be set.

Please have a look at the following default `applicationProperties.ini` file:

```
[COMMON]
useRoddyVersion=current                     # Use the most current version for tests

[DIRECTORIES]
configurationDirectories=[FOLDER_WITH_CONFIGURATION_FILES]
pluginDirectories=[FOLDER_WITH_PLUGINS]

[COMMANDS]
# Choose your job-manager. The first one executes the jobs locally.
jobManagerClass=de.dkfz.roddy.execution.jobs.direct.synchronousexecution.
↪DirectSynchronousExecutionJobManager
#jobManagerClass=de.dkfz.roddy.execution.jobs.cluster.pbs.PBSJobManager
#jobManagerClass=de.dkfz.roddy.execution.jobs.cluster.sge.SGEJobManager
#jobManagerClass=de.dkfz.roddy.execution.jobs.cluster.slurm.SlurmJobManager
#jobManagerClass=de.dkfz.roddy.execution.jobs.cluster.lsf.rest.LSFRestJobManager
commandFactoryUpdateInterval=300
commandLogTruncate=0                    # Truncate logged commands to this length. If
↪<= 0, then no truncation.

[COMMANDLINE]
CLI.executionServiceUser=USERNAME
# The execution service determines how commands are exectuted. Locally, or via SSH.
# SSHExecution service is needed if the host on which you run Roddy is different from␣
↪the
# submission host that allows executing the bsub/qsub command.
CLI.executionServiceClass=de.dkfz.roddy.execution.io.LocalExecutionService
#CLI.executionServiceClass=de.dkfz.roddy.execution.io.SSHExecutionService
CLI.executionServiceHost=[YOURHOST]
CLI.executionServiceAuth=keyfile
#CLI.executionServiceAuth=password
CLI.executionServicePasswd=
CLI.executionServiceStorePassword=false
CLI.executionServiceUseCompression=false
CLI.fileSystemInfoProviderClass=de.dkfz.roddy.execution.io.fs.FileSystemInfoProvider
```

To execute ACEseq run

```
sh $PATH_TO_ACEseq_RODDY_VERSION/Roddy/roddy.sh rerun ACEseq@copyNumberEstimation
↪$pid \
    --useconfig=$PATH_TO_ACEseq_RODDY_VERSION/configuration/applicationProperties.ini␣
↪\
    --cvalues="bamfile_list:$pathToControlBamFile;$pathToTumorBamFile,sample_
↪list:control;tumor,possibleControlSampleNamePrefixes:control,
↪possibleTumorSampleNamePrefixes:tumor"
```

More information on Roddy can be found here.

### 3.4.3 Legacy Docker versions

1. Download all reference files as specified in the section below.
2. Download the Base and ACEseq Docker images from the website: old-server
3. Import both files with (names might differ based on supplied version):

```
docker load < BaseDockerContainer.tar.gz
docker load < ACEseqDockerContainer.tar.gz
```

4. Download the control files archive and extract them. The directory contains the file "roddy.sh". Please call this script with: `bash roddy.sh`. You will see:

```
#!/bin/bash
# 1: Run mode, which might be "run" or "testrun"
# 2: Configuration identifier, normally "ACEseq"
# 3: Configuration directory
# 4: Dataset identifier / PID
# 5: Control bam file
# 6: Tumor bam file
# 7: Control bam sample name
# 8: Tumor bam sample name
# 9: Reference files path
# 10: Output folder
# 11: Optional: The SV file
```

An example call is:

```
bash roddy.sh run ACEseq \
    ./config/ \
    stds \
    /home/roddy/someproject/control_MB99_merged.mdup.bam \
    /home/roddy/someproject/tumor_MB99_merged.mdup.bam \
    control \
    tumor \
    /icgc/ngs_share/assemblies/hg19_GRCh37_1000genomes \
    ./output
```

Here you tell roddy to run the ACEseq configuration using the config folder in the current directory with a control and tumor bam. Also you tell Roddy the samples for both files namely control and tumor. Finally, you supply the path to the reference files and the folder where you will store your output data.

# CHAPTER 4

# QuickStart

To start ACEseq download package from here and install the reference files and conda package as described under *Installation & Run instructions*.

```
sh $PATH_TO_PLUGIN_DIRECTORY/Roddy/roddy.sh \
    rerun \
    ACEseq@copyNumberEstimation \
    $pid \
    --useconfig=$PATH_TO_PLUGIN_DIRECTORY/applicationProperties.ini \
    --cvalues="bamfile_list:$pathToControlBamFile;$pathToTumorBamFile,sample_
↪list:control;tumor,possibleControlSampleNamePrefixes:control,
↪possibleTumorSampleNamePrefixes:tumor"
```

Following parameters should be changed in the project.xml:

- baseDirectoryReference
- outputBaseDirectory
- outputFileGroup (in case all outputfiles should have different group than primary group)

Alternative running modes:

- runWithoutControl/isNoControlWorkflow (in case it should be run without control)
- runWithFakeControl (in case the coverage should be taken from a different control)

# CHAPTER 5

# Requirements

## 5.1 Hardware

ACEseq requires the execution of multiple jobs that are highly parallelized in the beginning but linearize towards the end of the workflow. It requires a maximum of 50g RAM in few of the Jobs. On a HPC cluster with multiple cores available it will usually finish within 24h (100-160 CPU h). The final output usually requires between 4 and 6g memory.

## 5.2 Software

The installation of all required software can be found under *Installation & Run instructions*.

CHAPTER 6

# Alternative Running Modes

Multiple alternative running modes are enabled with ACEseq.

## 6.1 Run With “Fake” Control

We often observed extremely noisy coverage profiles in matched controls from projects outside the ICGC MMML-Seq, possibly due to wrong handling of blood samples, preventing accurate copy number calls based on tumor/control ratios. For such samples ACEseq offers an option to replace the coverage signal from the matched control with an independent control whilst still maintaining the BAFs of the matched control. This control replacement option enables full analysis of these sample pairs including reliable discrimination between runs of homozygosity (ROH) in the germline and somatic loss of heterozygosity (LOH). Furthermore ACEseq can be run without matched control enlarging the spectrum of samples that can be processed.

To run the workflow in this mode, the `runWithFakeControl` option should be set to “true”.

```
<cvalue name="runWithFakeControl" value="true" type="boolean/>
<cvalue name='MALE_FAKE_CONTROL_PRE' value="pathToPID/${pid}/ACEseq/cnv_snp/${pid}.chr
↪" type='path'
        description="path and prefix to chromosome-wise 1kb coverage file used for␣
↪fake control workflow for male patients" />
<cvalue name='FEMALE_FAKE_CONTROL_PRE' value="pathToPID/${pid}/ACEseq/cnv_snp/${pid}.
↪chr" type='path'
        description="path and prefix to chromosome-wise 1kb coverage file used for␣
↪fake control workflow for female patients" />
<cvalue name='FAKE_CONTROL_POST' value=".cnv.anno.tab.gz" type='string'
        description="suffix for chromosome wise 1kb coverage files used for fake␣
↪control workflow"/>
```

The fake control files should thus be located at `${*_FAKE_CONTROL_PRE}${chromosome}${FAKE_CONTROL_POST}`. Each file should be a gzip-compressed TSV with a commented (`#`) header:

```
#chr    pos     end     normal  tumor   map
```

Of these columns the `chr` and `pos` columns are used to combine the analysis results of the tumor with the “fake” control file. The `normal` value from the “fake” control is inserted into the tumor results file (see `resources/analysisTools/copyNumberEstimationWorkflow/replaceControlACEseq.R`).

If you are operating at the DKFZ you will find a path prefix to a suitable generic control in the default configuration of the workflow.

## 6.2 Run Without Control

If no control sample is available, but ACEseq was already used to process other tumor sample pairs, one of their control coverage profiles can be used for normalization. In this case, no BAFs can be used from a matching control sample and also the patient’s sex is not inferred.

For the configuration you need to specify the path and prefix to a control coverage profile for a male and a female patient so it can be matched to the processed sample. To activate this option the configuration value `runWithoutControl` (for versions < 3) or `isNoControlWorkflow` (for versions >= 3) needs to be set to ‘true’, either via the command line execution under cvalues or in the project.xml. Furthermore, the patient’s sex needs to be set explicitly with `PATIENTSEX="male|female|klinefelter"`.

```
<cvalue name="isNoControlWorkflow" value="true" type="boolean
        description="since version 3"/>
<cvalue name="runWithoutControl" value="true" type="boolean"
        description="up to and including version 2; better use a more recent version!
↪"/>
<cvalue name="PATIENTSEX" value="male|female|klinefelter" />
<cvalue name='MALE_FAKE_CONTROL_PRE' value="pathToPID/${pid}/ACEseq/cnv_snp/${pid}.chr
↪" type='path'
        description="path and prefix to chromosome-wise 1kb coverage file used for␣
↪fake control workflow for male patients" />
<cvalue name='FEMALE_FAKE_CONTROL_PRE' value="pathToPID/${pid}/ACEseq/cnv_snp/${pid}.
↪chr" type='path'
        description="path and prefix to chromosome-wise 1kb coverage file used for␣
↪fake control workflow for female patients" />
<cvalue name='FAKE_CONTROL_POST' value=".cnv.anno.tab.gz" type='string'
        description="suffix for chromosome wise 1kb coverage files used for fake␣
↪control workflow"/>
```

Note that if run in no-control mode with SV input (you have `svOutputDirectory` set), then ACEseq does not expect the SV file to be named `svs_${PID}_filtered_somatic_minEventScore3.tsv`, like for the tumor/control case, but `svs_${PID}_filtered_minEventScore3.tsv`. If you use an output directory of the Sophia workflow in no-control mode, you can simply symlink the `svs_${PID}_filtered_minEventScore3.tsv` to create the “somatic”.

## 6.3 Run quality check only

In case you do not want to run the full ACEseq pipeline immediately, but would rather access the sample’s quality first you can start ACEseq with the option “runQualityCheckOnly” set to “true”.

## 6.4 Replace low quality control

If a control sample is very noisy and masks CNAs it can be replaced with the coverage profile from a different control of the same sex. For this run ACEseq with “runWithFakeControl” set to “true” and specify the values “FE-

MALE_FAKE_CONTROL_PRE" and "MALE_FAKE_CONTROL_PRE" as described in the section for analysis without matched control.

## 6.5 Run with/without SV breakpoint incorporation

To process samples with incorporation of SV breakpoints set the following in the project.xml:

```
<configurationvalues>
  <cvalue name='svOutputDirectory' value='${outputAnalysisBaseDirectory}/
↪nameOfDirectoryWithSVResults' type="path"/>
  <cvalue name='runWithSv' value='true' type="boolean"/>
</configurationvalues>

<filenames package='de.dkfz.b080.co.files' filestagesbase='de.dkfz.b080.co.files.
↪COFileStage'>
      <filename class="TextFile" onMethod="de.dkfz.b080.co.aceseq.ACESeqMethods.
↪mergeSv"
                selectiontag="svFileTag"
                pattern='${svOutputDirectory}/${pid}_svs.bedpe'/>
</filenames>
```

If the bedpe file does not exist ACEseq will submit all steps until the bedpe file is required. A rerun once the SV file is generated will start the pipeline up from the point where SV breakpoints are incorporated.

To process a samples without SVs please set the following in the project.xml:

```
<cvalue name='runWithSv' value='false' type="boolean"/>
<cvalue name='SV' value='no' type="string"/>
```

# CHAPTER 7

# Input Parameters

Multiple parameters can be set with ACEseq though not all are necessary to change. This table gives and overview and description for all available parameters

Table 1: “ACEseq parameters”

| name | value | type | description |
|---|---|---|---|
| aceseqOutputDirectory | ${outputAnalysisBaseDirectory}/ACEseq_${tumorSample} | path | |
| svOutputDirectory | ${outputAnalysisBaseDirectory}/SV_calls | path | |
| crestOutputDirectory | ${outputAnalysisBaseDirectory}/crest | path | |
| cnvSnpOutputDirectory | ${aceseqOutputDirectory}/cnv_snp | path | |
| imputeOutputDirectory | ${aceseqOutputDirectory}/phasing | path | |
| plotOutputDirectory | ${aceseqOutputDirectory}/plots | path | |
| runWithoutControl | false | boolean | use control for analysis (false\|true); up to version 2 |
| isNoControlWorkflow | false | boolean | use control for analysis (false\|true); since version 3 |
| minHT | 5 | integer | minimum number of consecutive SNPs to be considered for haploblocks |
| snp_min_coverage | 5 | integer | minimum coverage in control for SNP |
| cnv_min_coverage | 5000 | integer | minimum coverage for 1kb windows to be considered for merging in 10kb windows |
| mapping_quality | 1000 | integer | minimum mapping quality for 1kb windows to be considered for merging in 10kb windows (maximum mappability) |
| min_windows | 5 | integer | minimum number of 1kb windows fullfilling cnv_min_coverage and mapping_quality to obtain merged 10kb windows |
| min_X_ratio | 0.8 | float | minimum ratio for number of reads on chrY per base over number of reads per base over whole genome to be considered as female |
| min_Y_ratio | 0.12 | float | minimum ratio for number of reads on chrY per base over number of reads per base over whole genome to be considered as male |
| LOWESS_F | 0.1 | float | f parameter for R lowess function |
| SCALE_FACTOR | 0.9 | float | scale_factor for R lowess function |
| COVERAGEPLOT_YLIMS | 4 | float | ylims for Rplots in GC-bias plots |
| FILENAME_GC_CORRECTED_PLOT | ${plotOutputDirectory}/${pid}_gc_corrected.png | path | gc-bias plot before/after correction |
| GC_bias_json_key | gc-bias | string | key in GC-bias json |
| FILE_DENSITYBETA | ${aceseqOutputDirectory}/densityBeta.pdf | path | |

Continued on next page

Table 1 – continued from previous page

| name | value | type | description |
|---|---|---|---|
| min_DDI_length | 1000 | integer | minimum length for DEL/DUP/INV to be considered for breakpoint integration |
| selSVColumn | eventScore | string | column from bedpe file to be recored in ${pid}_sv_points.txt file |
| min_seg_width | 2000 | integer | segmentByPairedPSCBS() minwidth parameter in PSCBS R package |
| undo_SD | 24 | integer | segmentByPairedPSCBS() undo.SD parameter in PSCBS R package |
| pscbs_prune_height | 0 | integer | pruneByHClust() parameter h in PSCBS R package |
| min_segment_map | 0.6 | float | minimum average mappability over segment to be kept after segmentation |
| min_seg_length_prune | 9000 | integer | maximum of segment to be considered for merging to neighbouring segment prior to clustering |
| min_num_SNPs | 15 | integer | maximum number of SNPs in segment to be considered for merging to neighbouring segment prior to clustering |
| clustering | yes | string | should segments be clustered (yes\|no), coerage and BAF will be estimated and assigned clusterwide |
| min_cluster_number | 1 | integer | minimum number of clusters to be tried with BIC |
| min_membership | 0.8 | float | obsolete |
| min_distance | 0.05 | float | min_distance |
| haplogroupFilePrefix | haploblocks_chr | string | prefix for file with haplogroups per chromosome |
| haplogroupFileSuffix | txt | string | suffix for file with haplogroups per chromosome |
| haplogroupFilePath | ${imputeOutputDirectory}/${haplogroupFilePrefix} | path | |
| min_length_purity | 1000000 | integer | minimum length of segments to be considered for tumor cell content and ploidy estimation |
| min_hetSNPs_purity | 500 | integer | minimum number of control heterozygous SNPs in segments to be considered for tumor cell content and ploidy estimation |
| dh_stop | max | string | |
| min_length_dh_stop | 1000000 | integer | |
| dh_zero | no | string | |
| purity_min | 0.3 | float | minimum tumor cell content allowed |
| purity_max | 1.0 | float | i |
| ploidy_min | 1.0 | float | |
| ploidy_max | 6.5 | float | |
| SNP_VCF_CNV_PATH | ${cnvSnpOutputDirectory}/${pid}.chr | path | If the value is changed the value for the filename pattern MUST also be changed. |
| SNP_VCF_CNV_PATH_STR | ${SNP_VCF_CNV_PATH} | string | This value must be converted to a string because of a bug. |
| SNP_SUFFIX | snp.tab.gz | string | |
| | | | |
| CHR_NR | ${CHR_PREFIX}${PARM_CHR_INDEX}${CHR_SUFFIX} | string | |
| CHR_NAME | ${PARM_CHR_INDEX} | string | |
| AUTOSOME_INDICES | ( {1..22} ) | bashArray | |
| CREST | yes | string | include SV breakpoints in analysis (yes\|no) |
| mpileup_qual | 0 | integer | quality used for parameter 'Q' in samtools mpileup |
| CNV_MPILEUP_OPTS | "-A -R -B -Q ${mpileup_qual} -q 1 -I " | string | options for mpileup to determine which bases/reads to use |
| FILE_VCF_SUF | vcf | string | suffix for vcf files |
| FILE_TXT_SUF | txt | string | suffix for txt files |
| phasedGenotypesFilePrefix | phased_chr | string | prefix for phased genotypes file |
| unphasedGenotypesFilePrefix | unphased_chr | string | prefix for unphased genotypes file |
| phasedGenotypesFileSuffix | ${FILE_VCF_SUF} | string | suffix for phased genotypes file |

Continued on next page

Table 1 – continued from previous page

| name | value | type | description |
|---|---|---|---|
| unphasedGenotypesFileSuffix | ${FILE_VCF_SUFFIX} | string | suffix for unphased genotypes file |
| BCFTOOLS_OPTS | “-v g<br>“ | string | bcftools options for imputation |
| FAKE_CONTROL_POST | .cnv.anno.tab.gz | string | suffix for chromosome wise 1kb coverage files used for fake control workflow |
| PATIENTSEX | male | string | patient sex used in case of no control workflow (male\|female\|klinefelter) |
| CNV_ANNO_SUFFIX | .cnv.anno.tab.gz | string | suffix for mappability annotated chromosome-wise 1kb coverage files |
| CNV_SUFFIX | .cnv.tab.gz | string | suffix chromosome-wise 1kb coverage files |
| FILE_UNPHASED_PRE | ${imputeOutputDirectory}/${unphasedGenotypesFilePrefix} | path | |
| FILE_UNPHASED_GENOTYPE | ${imputeOutputDirectory}/unphased_genotype_chr | path | |
| FILE_PHASED_PRE | ${imputeOutputDirectory}/${phasedGenotypesFilePrefix} | path | |
| FILE_PHASED_GENOTYPE | ${imputeOutputDirectory}/phased_genotype_chr | path | |
| FILE_INFO | info | string | |
| FILE_INFO_SAMPLE | info_by_sample | string | |
| FILE_HAPS | haps | string | |
| FILE_HAPS_CONF | haps_confidence | string | |
| FILE_SUMMARY | summary | string | |
| FILE_WARNINGS | warnings | string | |
| FILE_PART | part | string | |
| FILE_SAMPLE_G | ${imputeOutputDirectory}/sample_g.txt | path | sample file used by imputation on X chromosome for females |
| MALE_FAKE_CONTROL_PRE | ${path_to_ACEseqResults}/and_smpl/${pid}.chr | path | prefix to chromosome-wise 1kb coverage file used for fake control workflow for male patients |
| FEMALE_FAKE_CONTROL_PRE | ${path_to_ACEseqResults}/and_smpl/${pid}.chr | path | prefix to chromosome-wise 1kb coverage file used for fake control workflow for female patients |
| PLOT_PRE | ${aceseqOutputDirectory}/${pid}_plot | path | |
| FILE_MOST_IMPORTANT_INFO_SEG_PRE | ${pid}_most_important_info | string | |
| FILE_MOST_IMPORTANT_INFO_SEG_POST | .txt | string | |
| FILE_SEGMENT_VCF_PRE | ${aceseqOutputDirectory}/${pid} | path | |
| FILE_SEGMENT_VCF_POST | .cnv.vcf | string | |
| outputUMask | 007 | string | |
| outputFileGroup | $accessGroup | | group for output files and directories |
| outputAccessRights | u+rw,g+rw,o-rwx | | access rights for written files |
| outputAccessRightsForDirectories | u+rwx,g+rwx,o-rwx | | access rights for written directories |
| possibleControlSampleNamePrefixes | ( blood) | bashArray | possible prefix of control bam if named ${prefix}_${pid}_$mergedBamSuffix |
| possibleTumorSampleNamePrefixes | ( tumor ) | bashArray | same as possibleControlSampleNamePrefixes |
| referenceGenome_1KGRef | ${path}/hs37d5.fa | path | reference genome file |
| REFERENCE_GENOME | ${referenceGenome_1KGRef} | | |
| dbSNP_FILE | ${path}/00-All.SNV.vcf.gz | path | |
| MAPPABILITY_FILE | ${path}/wgEncodeCrgMapabilityAlign100mer_chr.bedGraph.gz | path | mappability file |
| CHROMOSOME_LENGTH_FILE | ${path}/chrlengths.txt | path | |
| REPLICATION_TIME_FILE | ${path}/ReplicationTime_10cellines_mean_10KB.Rda | path | replication timing file |
| GC_CONTENT_FILE | ${path}/hg19_GRch37_100genomes_gc_content_10kb.txt | path | |
| GENETIC_MAP_FILE | ${path}/genetic_map_chr${CHR_NAME}_combined_b37.txt | path | impute files |
| KNOWN_HAPLOTYPES_FILE | ${path}/ALL.chr${CHR_NAME}.integrated_phase1_v3.20101123.snps_indels_svs.genotypes.nomono.haplotypes.gz | path | impute files |

Continued on next page

Table 1 – continued from previous page

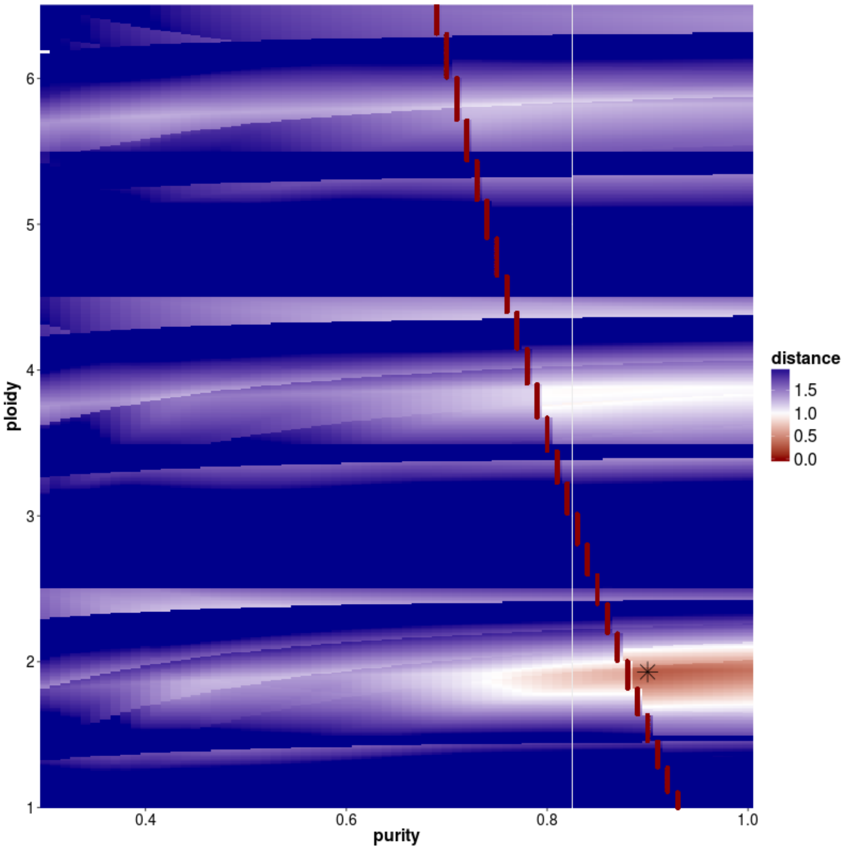
| name | value | type | description |
|---|---|---|---|
| KNOWN_HAPLOTYPES_LEGEND_FILE | ${path}/ALL.chr${CHR_NAME}.integrated_phase1_v3.20101123.snps_indels_svs.genotypes.nomono.legend.gz | path | impute files |
| GENETIC_MAP_FILE_X | ${path}/genetic_map_chrX_nonPAR_combined_b37.txt | path | impute files |
| KNOWN_HAPLOTYPES_FILE_X | ${path}/ALL_1000G_phase1integrated_v3_chrX_nonPAR_impute.hap.gz | path | impute files |
| KNOWN_HAPLOTYPES_LEGEND_FILE_X | ${path}/ALL_1000G_phase1integrated_v3_chrX_nonPAR_impute.legend.gz | path | impute files |
| outputExecutionDirectory | ${path}/exec_${executionTimeString} | path | path to log files |
| imputeBaseDirectory | ${path}/ | path | directory for impute files |
| mergedBamSuffix | merged.mdup.bam | string | A list of all known suffixes for merged bam files. I.e. merged.dupmark.bam, merged.mdup.bam… |
| mergedBamSuffixList | ${mergedBamSuffix} | string | A list of all known suffixes for merged bam files. I.e. merged.dupmark.bam, merged.mdup.bam… |
| defaultMergedBamSuffix | ${mergedBamSuffix} | string | The default suffix for merged bam files when they are created by Roddy. |
| libloc_PSCBS | | string | path to PSCBS library in R |
| libloc_flexclust | | string | path to felxclust library in R |

# CHAPTER 8

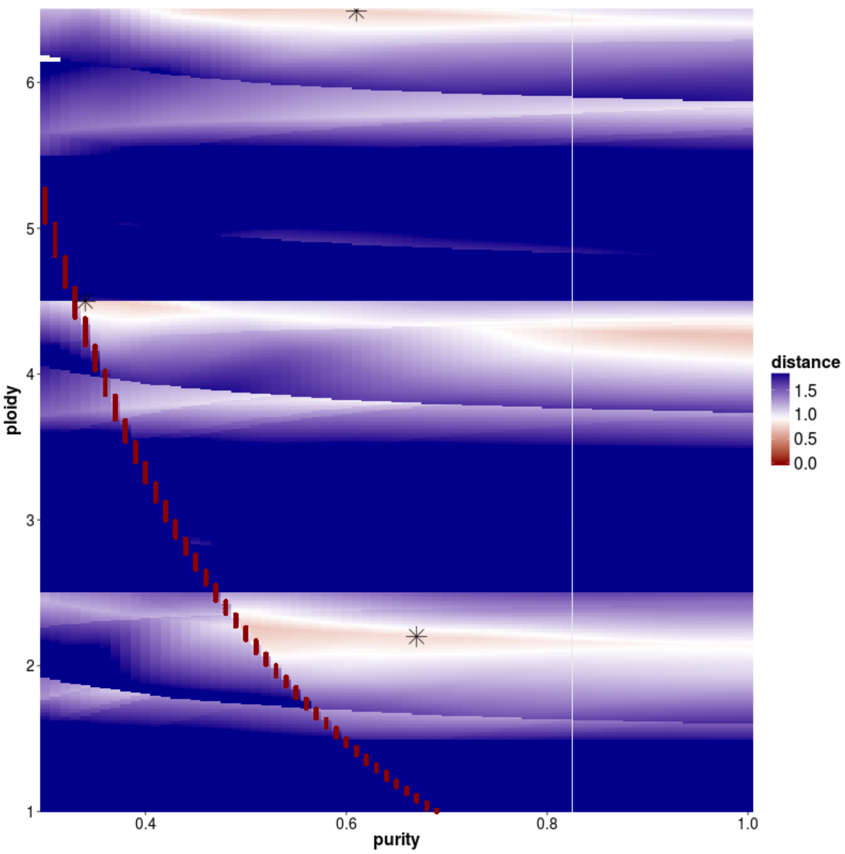
# Purity Evaluation

Depending on the sample, ACEseq might return multiple possible solutions. These solutions can be found in the file ${PatientID}_ploidy_purity_2D.txt within the ACEseq results directory.

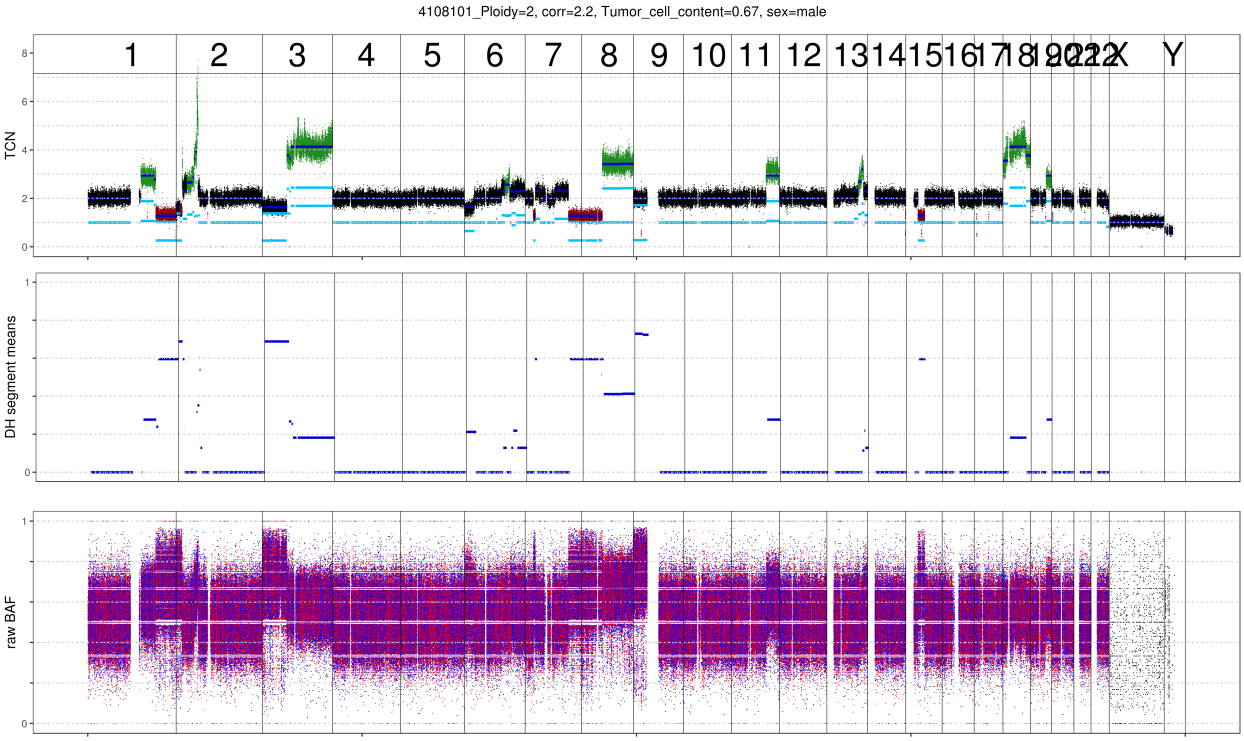
In addition to the table the plot ${PatientID}_tcn_distance_start.png shows a distance matrix. This distance matrix indicates the estimated mean distance per ploidy/tcc combination based on the equations explained in the methods section. The optimally found minima are indicated by a star.



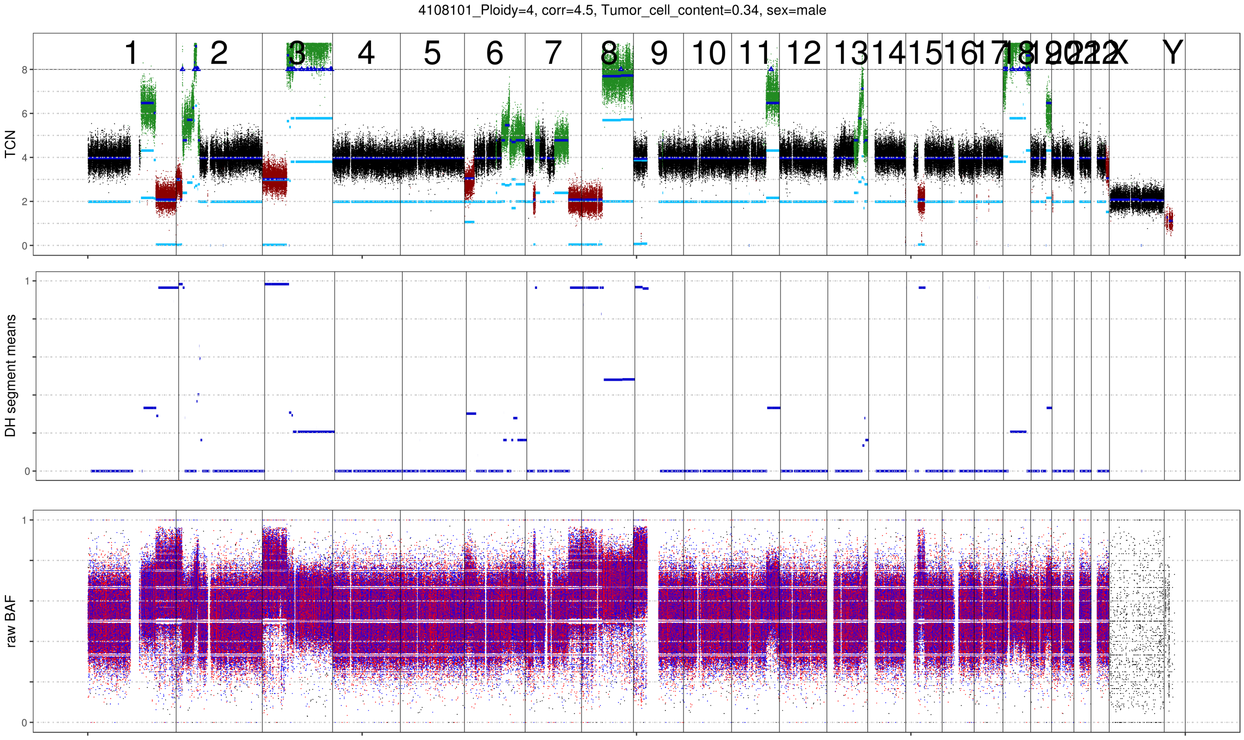
In case of multiple solutions, one can choose between the solution with the lowest distance or do as we suggest and choose the solution that is closest to diplot. It is recommended to make use of the prior knowledge about tumor biology as well as checking the final output as well as the mutant-allele frequency of a patient.



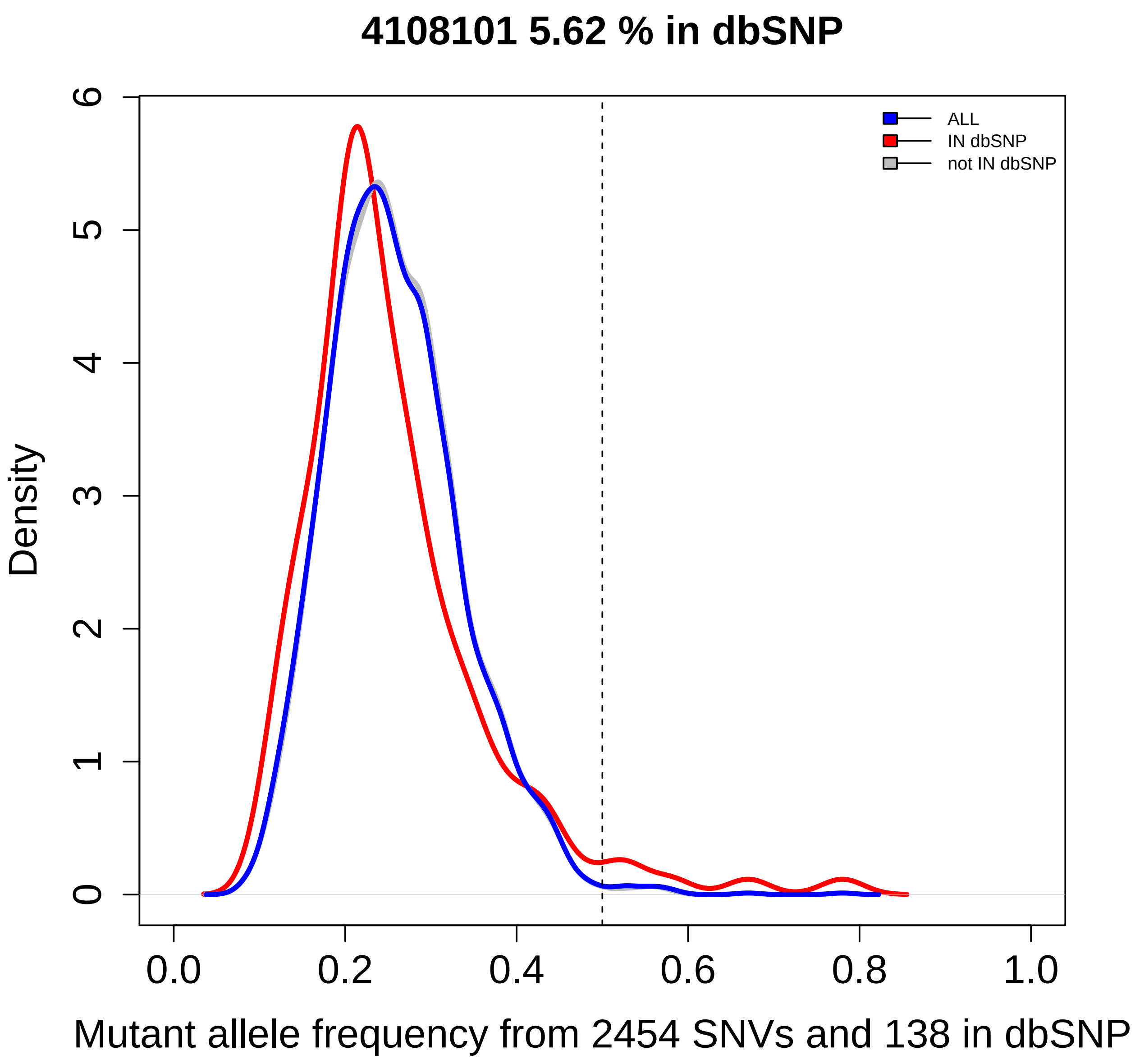
As can be seen below the benefit of increasing the ploidy of this sample to tetraploid leads to a clonal fit of multiple segments though many others remain subclonal (indicated by deviation from integer copy number states). This is often observed for heterogenous samples such as this shown lymphoma sample. Lymphoma tend to be diploid and heterogenous indicating that the diploid solution is correct. In addition we plotted the MAF distribution over balanced segments, that supports our assumption. Diploid solution:



Tetraploid solution:



MAF distribution over balanced segments:

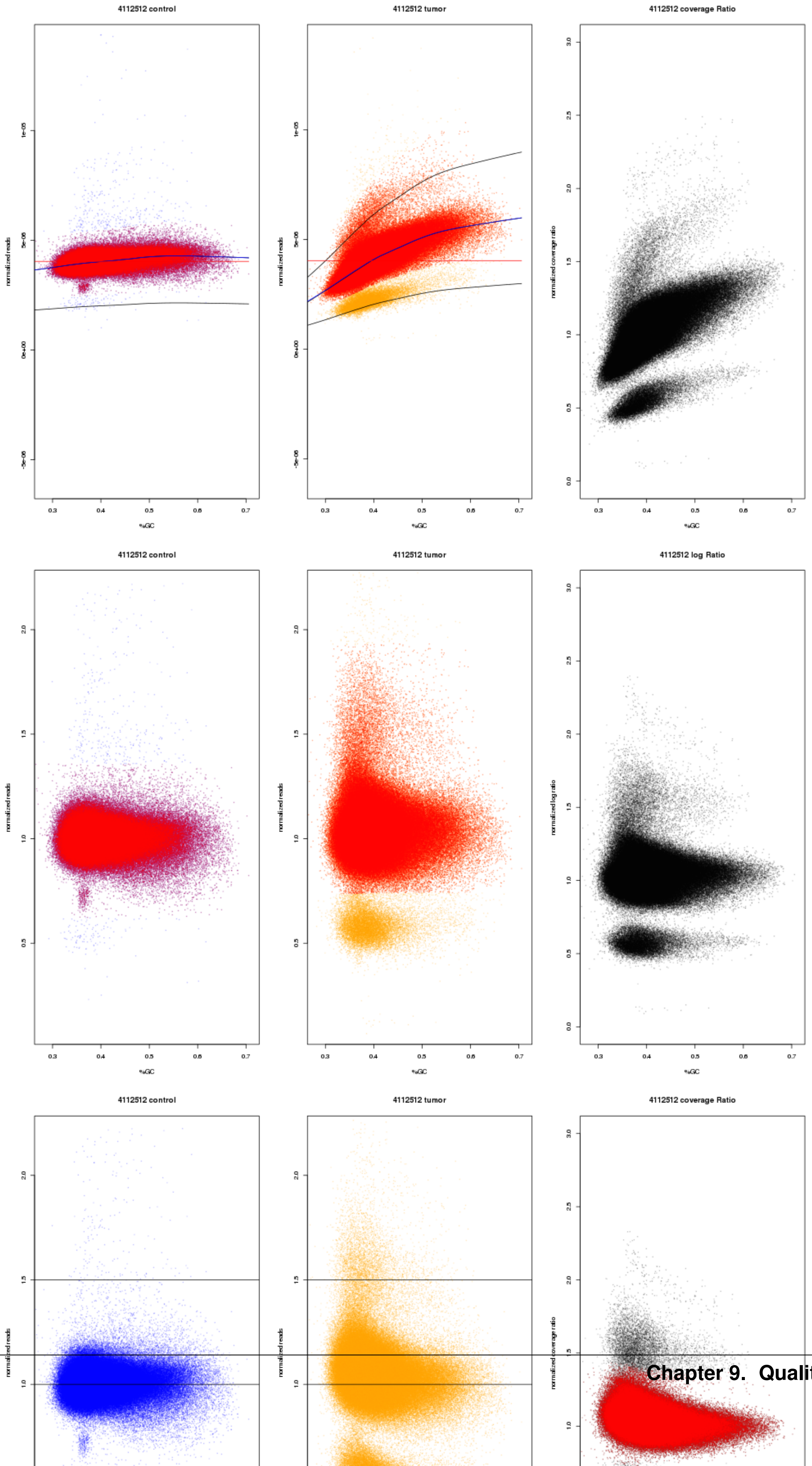

4108101 5.62 % in dbSNP
Density
0
1
2
3
4
5
6
ALL
IN dbSNP
not IN dbSNP
0.0
0.2
0.4
0.6
0.8
1.0
Mutant allele frequency from 2454 SNVs and 138 in dbSNP

# CHAPTER 9

# Quality check

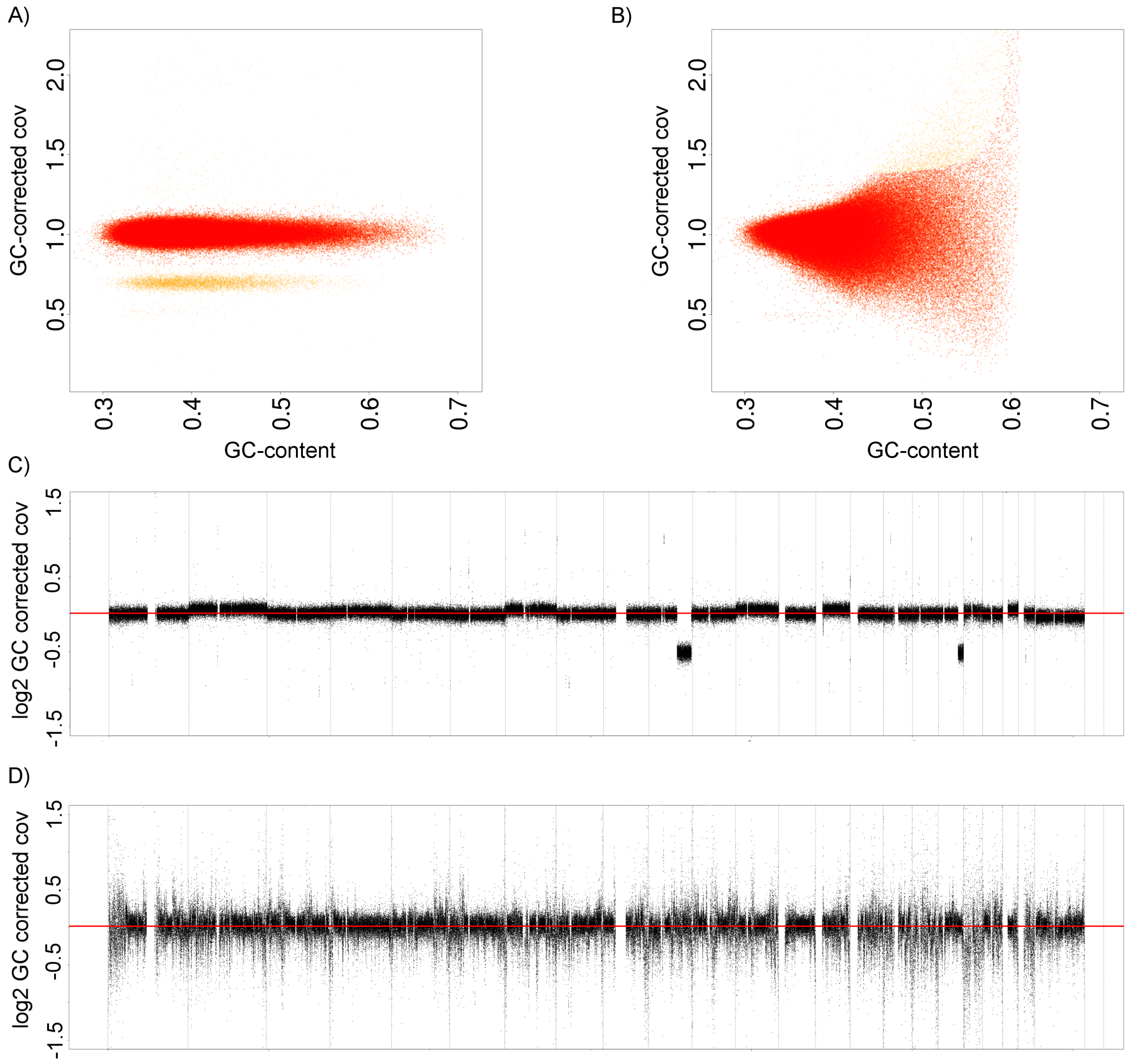
ACEseq provides a thorough quality check of the sample by investigation of the GC-bias: 1.differences in GC-bias between tumor and control 2.evenness of coverage in tumor and control

The following plot shows the raw GC bias for a healthy control (left), a corresponding tumor (middle) and the tumor/control ratio (right). The top row depicts raw data while the middle row indicates GC-bias corrected data and the bottom line indicates GC-bias and RT-bias corrected data.


4112512 control
4112512 tumor
4112512 coverage Ratio
normalized reads
normalized coverage ratio
1e-05
5e-06
0e+00
-5e-06
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.3
0.4
0.5
0.6
0.7
%GC
4112512 control
4112512 tumor
4112512 log Ratio
normalized reads
normalized log ratio
4112512 control
4112512 tumor
4112512 coverage Ratio
normalized reads
normalized coverage ratio

The file ${pid}_qc_gc_corrected.json provides information about slope, curvature and their differences between tumor and control. A strong diffrence between tumor and control can impact sensitivity and specificty of other variant calls.

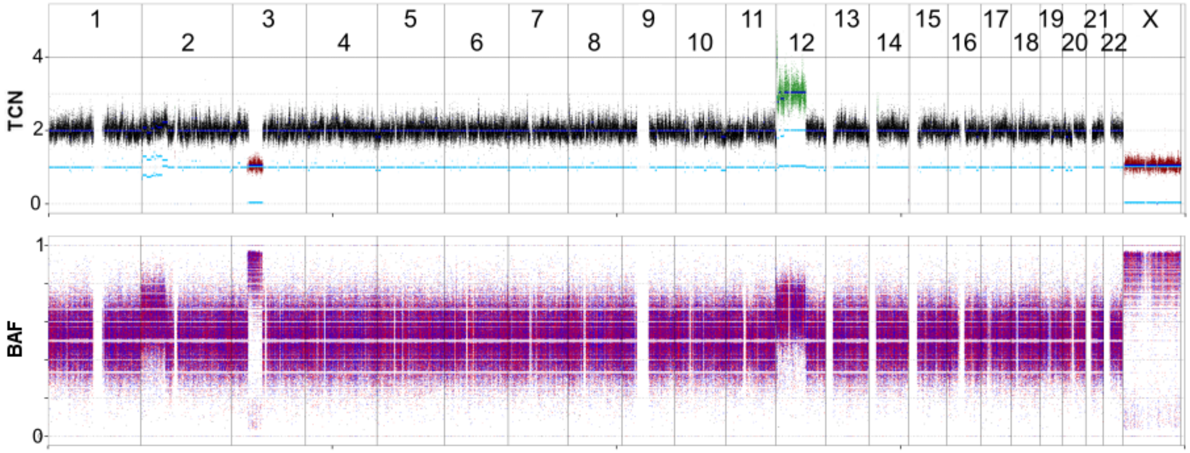
The full width half maximum (FWHM) indicates the noise level within the majority copy number state in a sample. If it exceeds 0.205 in the control or 0.34 in the tumor a sample should be flagged with a warning. Yellow flagged tumors might have decreased specificity and sensitiviy. Samples should be red flagged in case the FWHM exceed 0.29 in healthy controls of 0.54 in tumors. Red flagged tumor samples are very likely to accumulate artifacts due to a noisy coverage profile and should be excluded from further analysis Flagged controls can be rescued by rerunning the pipeline with “rerunWithFakeControl=true”. The following plot shows a sample with low FWHM (A and C) and a sample with noisy coverage and thus high FWHM (B and D). No good copy number estimates can be obtained from the high FWHM sample.



# CHAPTER 10

# Final Output

A graphical presentation of the final results is given for each tcc/ploidy solution. A general overview is give for the whole genome as shown here and per chromosome. Points represent raw SNP data points, colored by their copy number with regard to the majority copy number state (red:loss, black: neutral, green: gain). Segments are indicated by dark and light blue lines for total and allele-specific copy number respectively. Raw BAF are shown in the bottom panel which can be used to evaluate tcc and confirm allele-specific copy numbers.



Final results are provided in two formats. A reduced set of information is contained in the file ${pid}_most_important_info_${ploidy}_${purity}.txt while the full set is presented in ${pid}_comb_pro_extra_${ploidy}_${purity}.txt. A mapping of both headers and a corresponding description is given here.

Table 1: “Final output column headers”

| most_important_info | comb_pro_extra | description |
|---|---|---|
| chromosome | chromosome | |
| start | start | start coordinate |

Continued on next page

Table 1 – continued from previous page

| most_important_info | comb_pro_extra | description |
|---|---|---|
| end | end | end coordinate |
| SV.Type | crest | SV type connected to both or one breakpoint |
| length | length | length of segment |
| TCN | tcnMean | total copy number |
| NbrOfHetSNPs | tcnNbrOfHets | number of control heterozygous SNPs in segment |
| dhSNPs | dhMax | estimated DH |
| minStart | minStart | last SNP prior to segment start |
| maxStart | maxStart | first SNP after segment start |
| minEnd | minStop | last SNP prior to segment end |
| maxEnd | maxStop | first SNP after segment end |
| covRatio | tcnMeanRaw | bias corrected coverage ratio |
| dhEst | dhMean | raw DH |
| c1Mean | c1Mean | minor allele copy number |
| c2Mean | c2Mean | major allele copy number |
| genotype | genotype | ratio of rounded allele copy numbers |
| CNA.type | CNA.type | DUP/DEL/LOH/TCNneutral |
| GNL | GNL | gain/loss/loh compared to diploid |
| | tcnNbrOfSNPs | number of SNPs per segment |
| | tcnNbrOfLoci | number of SNPs per segment |
| | dhNbrOfLoci | heterozygous SNPs per segment |
| | map | mappable/unmappable |
| | cluster | cluster assigned during merging |
| | neighbour | indicates whether neighbouring segments exist on both sides |
| | distDH | distance to main cluster center DH |
| | errorSNP | error for DH direction |
| | distTcn | distance to main cluster center coverage ratio |
| | errorLength | error in coverage ratio direction |
| | totalError | sum of errorSNP and errorLength |
| | area | AUC ratio |
| | peaks | 1 for balanced; 2 for imbalanced |
| | meanCovT | average total coverage per cluster |
| | meanCovB | average total coverage of B allele |
| | AF | allelic factor |
| | BAF | B-allele frequency |
| | A | rounded minor allele copy number |
| | B | rounded major allele copy number |
| | TCN | rounded copy number |
| | ploidy | majority copy number used as reference for CNA.type |
| | Sex | patient sex |
| | cytoband | cytoband |

# CHAPTER 11

# HRD, LST, TAI

HRD, LST and TAI scores are provided in the file ${pid}_HRDscore_${ploidy}_$tcc} for each solution.

HRD and LST are merged based on smoothed segments as suggested by Popova et al. (doi: 10.1158/0008-5472.CAN-12-1470).

TAI are based on unmerged segments. A combination of the three scores might be useful.

# CHAPTER 12

# Methods - Theory

ACEseq can be used to estimate copy-numbers from WGS data using a tumor vs. control approach. Thus a pre-requesite is WGS data from healthy tissue and tumor tissue of the same patient with at least 30x coverage. Samtools [] mpileup is used to determine the coverage for tumor and control sample - position specific for each single nucleotide polymorphism (SNP) position recorded in dbSNP and per 1 kb window. To get chromosome specific allele frequencies, the genotypes of SNP positions are phased with Impute2 [] and A and B allele are assigned accordingly. Haploblocks are defined as regions with consecutively phased SNPs. Subsequently, B-allele frequencies (BAFs) are estimated for all SNP positions in tumor and control with sufficient coverage in the control:

$$BAF = \frac{cov_{SNP}^{B}}{cov_{SNP}^{A} + cov_{SNP}^{B}}$$

These can be converted to the decrease of heterozygosity a measure of the allelic state [Olshen et al.].

$$DH = 2 \times |BAF - 0.5|$$

## 12.1 Pre-processing

To estimate the coverage of each SNP position a general coverage of 10 kb windows was determined. 1 kb coverage windows are merged into 10 kb windows in case enough contributing windows with sufficient coverage and mappability are found in the corresponding region. The resulting coverage values are normalized with the sum of all 10 kb coverage windows for tumor and control respectively. These normalized estimates are subsequently corrected for a possible GC- and replication-timing bias.

## 12.2 GC-/Replication timing bias correction

### 12.2.1 Correction for GC bias

Correction for GC bias

As described in detail by Benjamini and Speed (REF) genomic regions with varying GC content may be sequenced at different depth due to selection bias or sequencing efficiency. Differing raw read counts in these regions even in the absence of copy number alterations can could lead to false positive calls. A GC-bias plot (Figure XY) can be used to visually inspect the bias of a sample. ACEseq first fits a curve to the data using LOWESS (locally weighted scatterplot smoothing, implemented in R) to identify the main copy number state first, which will be used to for a second fit. The second fit to the main copy number state is used for parameter assessment and correction of the data. This two-step fitting is necessary to compensate for large copy number changes that could lead to a misfit. The LOWESS fit as described above interpolates over all 10 kb windows. It thus averages over all different copy number states. If two states have their respective center of mass at different GC content, this first LOWESS fit might be distorted and not well suited for the correction. The full width half maximum (FWHM) of the density over all windows of the main copy number state is estimated for control and tumor. An usual large value here indicates quality issues with the sample.

### 12.2.2 Correction for replication time

Once the data is corrected for GC-bias the replication timing bias is considered. In general, if a fraction of the cells in the analyzed sample is cycling, early replicating regions would be expected to display higher coverage than late replicating regions, as a higher percentage of these would already have undergone replication in the S-phase [Zitat Koren et al.]. For a subtle analysis of copy number alterations, it would be beneficial to correct for this replication timing bias. Large fractions of the genome have common replication timing in different cell types or tissues, but there are regions of tissue or organ specificity [] [Zitat RepliSeq]. In the present work, a consensus replication timing score, the RepliSeq score as described by [] [Zitat RepliSeq], is attributed to every 10 kb window of the genome by averaging over the RepliSeq information from different cell lines. Replication timing bias plots can be generated analogously to the GC bias plots. A LOWESS fit on the already identified main cluster is carried out to correct for this bias (Figure?). This correction is performed on the GC-corrected data to obtain the final corrected coverage data, which will be used in the following.

The two bias correction steps described above are done sequentially. A simultaneous 2D LOWESS or LOESS correction would be desirable, but fails due to computational load (the clusters to be fitted have 106 points). Different parameters such as slope and curvature of the both LOWESS correction curves used are extracted. The GC curve parameters is used as quality measures to determine the suitability of the sample for further analysis whereas the replication timing curve parameters is used to infer the proliferation activity of the tumor. We could show a strong correlation between Ki-67 estimates and the slope of the fitted curve (Figure).

Once corrected a coverage ratio is calculated as the ratio of normalized tumor coverage over normalized control coverage:

$$covR = \frac{covT^{corrected}_{window}}{covN^{corrected}_{window}}$$

Finally SNP and coverage data are merged. Regions without coverage or SNP information are discarded.

## 12.3 Segmentation

Once data pre-processing is completed the genome is segmented with the PSCBS (parent specific circular binary segmentation) [] (Version!!!) algorithm. Prior to the actual segmentation, segment-boundaries due to a lack of coverage are determined. Single outliers among the coverage and very low coverage regions are determined using PSCBS functions. In addition to these, breakpoints that are indicated by previously called structural variations are taken into account. During the actual segmentation step the genome is segmented based on the pre-defined breakpoints, changes in the coverage ratio and DH. DH values are only considered in case the SNP position is heterozygous in the control.

## 12.4 Segment reliability

Homozygous deletions are called in segments that lack mapped reads. These deletions are only considered to be true in case the low read count is unlikely to be caused by a low mappability. Thus, the mappability is assessed for all segments. Regions with mappbility below 60% are considered unmappable and not further considered for copy number estimation. Each SNP position is annotated with the new segment information and mappability.

## 12.5 Segment clustering and merging

In order to avoid over-segmentation short segments (default <9 kb) are attached to the closest neighboring segment according to the coverage ratio. Subsequently, segments from diploid chromosomes are clustered according to the log2 of the coverage ratio and DH. These values are scaled prior to clustering. The DH of a segment is defined as the most commonly found DH value among all SNPs in the segment that are heterozygous in the control. In a first step, c-means clustering is performed. The segments are weighted according to the log2 of their length. A minimum number of one clusters is required allowing up to 20 clusters and the optimal cluster number is determined with BIC clustering **:raw-latex:‘\cite{}‘**. The number is used to cluster the points with cmeans subsequently (with the R fpc package clusterboot function).

To avoid over-fitting a further downstream processing is applied. Firstly, the minimal accuracy defined by the FWHM is taken into account. Cluster with more than 85% of all points within these coverage limits are chosen. Of these the cluster with most segments is defined as main cluster. The other chosen clusters are merged with the main cluster if their the difference between their center and the main cluster center is not bigger than XX times the DH-MAD of the main clusters. Neighboring segments are merged before new cluster centers are determined. In a second step segments that are embedded within main cluster segments are considered for merging. The number of control heterozygous SNP positions and the length are considered here to establish two criteria. Segments with less than 5 heterozygous SNPs are merged with the main cluster if they lie between the FWHM boundaries. Additionally, error values defining the tolerable deviation from the main cluster center is defined both for DH and coverage value as follows:

$$
\begin{aligned}
errorDH &= \frac{1}{\sqrt{numberofheterozygousSNPs}} \\
errorCoverage &= \frac{1}{log2(length)}
\end{aligned}
$$

If the SNP error of a selected segment exceeds the distance in DH and the length error exceeds the coverage difference it is appointed to the main cluster. Again neighboring segments with identical clusters are merged. Finally, a general cluster coverage is estimated from all relevant segments and assigned to the cluster members to further reduce noise in the data.

## 12.6 Allelic adjustment

To get better estimates of a segments allelic state as balanced or imbalanced the phasing and segmentation information are combined. Within an imbalanced segment the more prominent allele should be consistently assigned to the same allele across all haploblocks. For balanced segments a haploblock-wise swap of A- and B-allele should have no effect. Thus, the median tumor BAF is calculated haploblock-wise for all SNP positions that are heterozygous in the control. If it is below 0.5 A- and B-allele are swapped within the haploblock region to get consistency across the haploblocks of a segment. This procedure ensures a more accurate estimation of the allelic state of a region in the next step.

## 12.7 Calling of Allelic Balance and Imbalance

In order to be able to identify the allelic state of a segments, a first test to distinguish between allelic balance and imbalance of a segment independent from the degree of imbalance was implemented. Our method evaluates the area under the BAF density curve left and right of 0.5. Balanced segments should have an equal area and the allelic state of a segment can be defined by equation [eq:areaDiff], i.e. computing the absolute value of the relative difference between the left and right area.

$$diffA_{segment} = \frac{|A_{right} - A_{left}|}{A_{right} + A_{left}}$$

For balanced segments $diffA_{segment}$ should be close to zero, whereas this value should shift more towards one for imbalanced segments. Thus, a cut-off to differentiate between balanced and imbalanced segments is needed. In the following we propose a way to establish a dynamic and sample dependent cut-off. In case a sample has several segments that correspond to different states, e.g one balanced and one imbalanced state, these will be represented by different peaks in the density distribution of $diffA_{segment}$. Hence the minima between the peaks can be used as cut-off. Corresponding to the above reasoning peaks further left in the distribution are more likely to represent balanced states. The minimum that differentiates a balanced from an imbalanced state varies across different samples. Potentially this depends on the relative contribution of copy number states, tumor cell content, contamination, subpopulations and sequencing biases. Empirically the discrimination is optimal for cut-off values in the range of 0.25 and 0.35. The minimum value of the density function within this interval is chosen as cut-off. The allelic state is only evaluated for segments on diplod chromosomes that fullfill certain quality criteria in order to ensure confident calls. Once $diffA_{segment}$ was calculated for a segment and the overall cut-off determined segments that exceed the cut-off are classified imbalanced. Segments below the cut-off are classified as balanced.

## 12.8 Copy Number Estimation

Once the allelic state of a segment is determined it can be used for the computation of tumor cell content and ploidy of the main tumor cell population. The average observed tumor ploidy can be determined with equation [eq:averagePloidy].

$$D_t = p_t \times P_t + 2 \times (1 - p_t)$$

Where p$_t$ is the tumor purity and P$_t$ is the tumor ploidy. Using the observed tumor ploidy and the coverage ratio of a segment (covR:math:_*{segment}*), the total copy number of a segment can be estimated as follows:

$$TCN_{segment} = \frac{covR_{segment} \times D_t - 2 \times (1 - pt)}{p_t}$$

This can be used subsequently to obtain the real BAF value for each segment by converting the coverage data to a copy number. The allelic factor (AF) is introduced for this as a segment-wise conversion measure.

$$AF_{segment} = \frac{\frac{covT^{norm}_{segment}}{10000}}{p_t \times TCN_{segment} + 2 \times (1 - p_t)}$$

covT$^{norm}_{segment}$ represents the observed tumor coverage of a segment. The factor $\frac{1}{10000}$ is introduced to get from the initial 10 kb window coverage to a per base pair coverage. The BAF value of a segment can be calculated as follows.

where covT$^{B}_{segment}$ is the observed tumor coverage of a segment. The BAF value can now be used to calculate the DH of a segment according to [eq:DH]. Finally the allele-specific copy numbers are estimated.

$$TCN^B_{segment} = \frac{1}{2} \times TCN_{segment} \times (1 - DH_{segment})$$
$$TCN^A_{segment} = TCN_{segment} - TCN^B_{segment}$$

## 12.9 Purity and ploidy estimation

To obtain actual copy numbers for each segment ploidy and tumor cell content of the tumor sample have to be inferred from the data. Information about the allelic state of a segment is combined with TCN, DH and allele-specific copy numbers calculations. The combination of ploidy and tumor cell content that can explain the observed data the best is to be found. Possible ploidies in the range from 1 to 6.5 in steps of 0.1 and possible tumor cell content from 30% to 100% in steps of 1% are tested. The evaluation is done based on the distance of all segments from their next plausible copy number state. Imbalanced segments are fitted to a positive integer value.

$$distance_{tcn\_imbalanced} = abs(TCN_{segment} - round(TCN_{segment}))$$

In addition the allele specific copy number is estimated according to equation [eq:TCNb] and [eq:TCNa]. For each allele a distance is defined accordingly:

$$\begin{aligned} distance_{tcn\_a\_imbalanced} &= abs(TCN^{A}_{segment} - round(TCN^{A}_{segment})) \\ distance_{tcn\_b\_imbalanced} &= abs(TCN^{B}_{segment} - round(TCN^{B}_{segment})) \end{aligned}$$

The total distance as quality measure of a fit is defined as the sum of the distances.

$$distance_{segment\_imbalanced} = distance_{tcn\_imbalanced} + distance_{tcn\_a\_imbalanced} + distance_{tcn\_b\_imbalanced}$$

Balanced segments can only be fitted to even total copy numbers. The distance is defined as follows:

$$\begin{aligned} distance_{tcn\_balanced} &= \frac{TCN_{segment}}{2} - floor(\frac{TCN_{segment}}{2}) \\ &?identicalto \\ distance_{tcn\_balanced} &= abs(\frac{TCN_{segment}}{2} - round(\frac{TCN_{segment}}{2})) \times 2 \end{aligned}$$

As both alleles are expected to be present in equal numbers the allele specific copy number as well as the total distance can be derived.

$$\begin{aligned} distance_{tcn\_a\_balanced} = distance_{tcn\_b\_balanced} &= \frac{distance_{tcn\_balanced}}{2} \\ distance_{segment\_balanced} &= distance_{tcn\_balanced} + distance_{tcn\_a\_balanced} + distance_{tcn\_b\_balanced} \\ &= 2 \times distance_{tcn\_balanced} \end{aligned}$$

For each ploidy and tumor cell content combination a mean distance is defined by using the segment length as weights:

$$meanDist(p_t, P_t) = \frac{\sum^{i}_{1:N_{segments}} (distance_{segment_i} * length_{segment_i})}{\sum^{i}_{1:N_{segments}} length_{segment_i}}$$

All segments on diploid chromosomes that exceed a pre-set length and contain a sufficient amount of heterozygous SNP positions are used for the estimation. The smaller the distance the more likely a combination is chosen as final solution. Combinations of ploidy and tumor cell content that lead to negative copy numbers or exceed the DH limits are excluded as solution and used to set a minimum limit.

### 12.9.1 Final output

Once the optimal ploidy and tumor cell content combinations are found the TCN and allele-specific CN will be estimated for all segments in the genome and classified (gain, loss, copy-neutral LOH, loss LOH, gain LOH, sub). If a segments TCN is further than 0.3 away from an integer value it is assumed to originate from subpopulations in the tumor sample that lead to gains or losses in part of the tumor cell population.

# Index