

Haplotype Calling Plugin

USER MANUAL

User manual for CLC Haplotype Calling 22.0 beta 1

Windows, macOS and Linux

January 4, 2022

This software is for research purposes only.

QIAGEN Aarhus
Silkeborgvej 2
Prismet
DK-8000 Aarhus C
Denmark



Contents

I	Introduction	5
1	Introduction	6
1.1	The concept of CLC Haplotype Calling	6
1.2	Contact information	6
1.3	System requirements and installation	7
1.3.1	System requirements	7
1.3.2	Installation of plugins	7
1.3.3	Uninstalling plugins	8
1.4	Reference data management	9
1.4.1	The Reference Data Manager	9
II	Import and Export	11
2	Data import	12
2.1	Import	12
2.1.1	VCF	12
2.1.2	Standard CLC Variant Track	13
3	Data export	14
3.1	Export Genotype Track as VCF	14
3.2	Export Allele Table as CSV	15
3.3	Export Marker Genotypes as CSV or TSV	15

III Haplotype Calling	17
4 Genotype Track	18
4.1 Genome Model	18
4.2 Locus Table	20
4.3 Allele Table	21
4.4 Header View	23
4.5 Track View	24
5 Microhaplotype Caller	25
5.1 Microhaplotype Caller - Method	25
5.2 Microhaplotype Caller - Options	26
5.3 Microhaplotype Caller - Filters	27
5.3.1 General filters	27
5.3.2 Detection filters	27
5.3.3 Noise filters	28
5.4 Microhaplotype Caller - Annotations	30
6 Glossary	34
6.1 Glossary	34
IV Prebuilt workflows	35
7 Human Identity workflows	36
7.1 Detect QIAseq Human Identity SNPs and Microhaplotypes	36
Bibliography	38

Part I

Introduction

Chapter 1

Introduction

Welcome to CLC Haplotype Calling 22.0 beta 1 – a software package providing a haplotype-aware and VCF-friendly framework that enables you to perform detailed genome analysis and easily share genome information with external applications.

This first version of CLC Haplotype Calling is intended solely for analysis of QIAseq Human Identity (HID) targeted panel data. Analysis of other types of data is not yet supported.

Note that the functionality of this plugin is in beta. It is under active development and subject to change without notice. Generated results may not be accessible when later versions of the plugin are installed.

1.1 The concept of CLC Haplotype Calling

This software package revolves around the **Genotype track**, which is a VCF inspired representation of alleles in database and sample genome context. This suite of tools, so far including **Microhaplotype Caller**, standard CLC variant track converter, VCF importer and exporter, is intended to serve as a framework to perform detailed haplotype-aware analysis of genomic variation.

1.2 Contact information

CLC Haplotype Calling is developed by:

QIAGEN Aarhus
Silkeborgvej 2
Prismet
8000 Aarhus C
Denmark

<https://digitalinsights.qiagen.com/>

Email: ts-bioinformatics@qiagen.com

The QIAGEN Aarhus team continuously improves products with your interests in mind. We welcome feedback and suggestions for new features or improvements. How to contact us

is described at: http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Contact_information_citation.html.

You can also make use of our online documentation resources, including:

- Core product manuals <https://digitalinsights.qiagen.com/technical-support/manuals/>
- Plugin manuals <https://digitalinsights.qiagen.com/products-overview/plugins/>
- Tutorials <https://digitalinsights.qiagen.com/support/tutorials/>
- Frequently Asked Questions <http://helpdesk.clcbio.com/index.php?pg=kb>

1.3 System requirements and installation


1.3.1 System requirements

To work with CLC Haplotype Calling you will need to have CLC Genomics Workbench 22.0 with Biomedical Genomics Analysis 22.0 installed on your computer. With exception of the requirements below, the system requirements of CLC Haplotype Calling are the same as the ones required for the CLC Genomics Workbench (http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=System_requirements.html)

- 16 GB RAM recommended

1.3.2 Installation of plugins

Note: In order to install plugins and modules, the *CLC Workbench* must be run in administrator mode. On Windows, you can do this by right-clicking the program shortcut and choosing "Run as Administrator". On Linux and Mac, it means you must launch the program such that it is run by an administrative user.

Plugins and modules are installed and uninstalled using the Workbench Plugin Manager. To open the Plugin Manager, click on the **Plugins ()** button in the top Toolbar, or go to the menu option:

Utilities | Manage Plugins... ()

The Plugin Manager has two tabs at the top:

- **Manage Plugins** An overview of your installed plugins and modules is provided under this tab.
- **Download Plugins** Plugins and modules available to download and install are listed in this tab.

To install a plugin, click on the **Download Plugins** tab (figure 1.1). Select a plugin. Information about it will be shown in the right hand panel. Click on the **Download and Install** button to install the plugin.

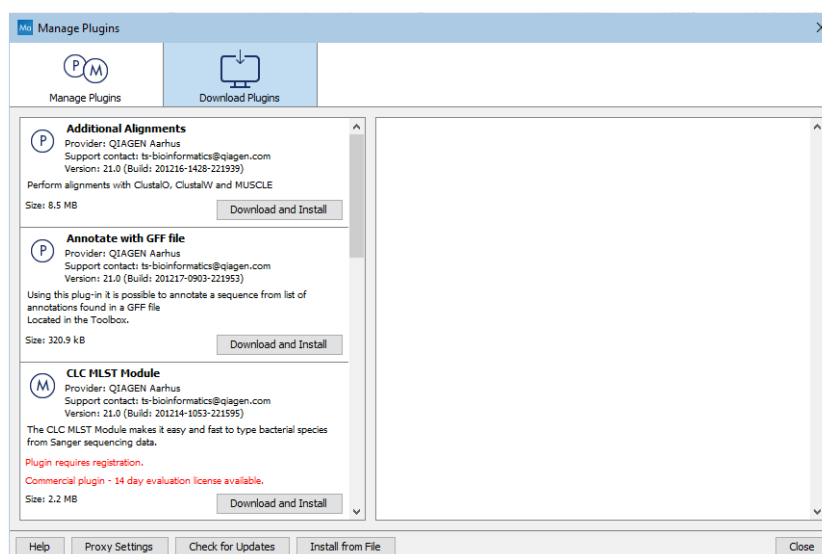


Figure 1.1: Plugins and modules available for installation are listed in the Plugin Manager under the Download Plugins tab.

Accepting the license agreement

The End User License Agreement (EULA) must be read and accepted as part of the installation process. Please read the EULA text carefully, and if you agree to it, check the box next to the text **I accept these terms**. If further information is requested from you, please fill this in before clicking on the **Finish** button.

Installing a cpa file


If you have a .cpa installer file for CLC Haplotype Calling, you can install it by clicking on the **Install from File** button at the bottom of the Plugin Manager.

If you are working on a system not connected to the internet, plugin and module .cpa files can be downloaded from <https://digitalinsights.qiagen.com/products-overview/plugins/> using a networked machine, and then transferred to the non-networked machine for installation.

Restart to complete the installation

Newly installed plugins and modules will be available for use after restarting the software. When you close the Plugin Manager, a dialog appears offering the opportunity to restart the *CLC Workbench*.

1.3.3 Uninstalling plugins

Plugins and modules are uninstalled using the Workbench Plugin Manager. To open the Plugin Manager, click on the **Plugins** () **button** in the top Toolbar, or go to the menu option:

Utilities | Manage Plugins... ()

This will open the Plugin Manager (figure 1.2). Installed plugins and modules are shown under the Manage Plugins tab of the Plugins Manager.

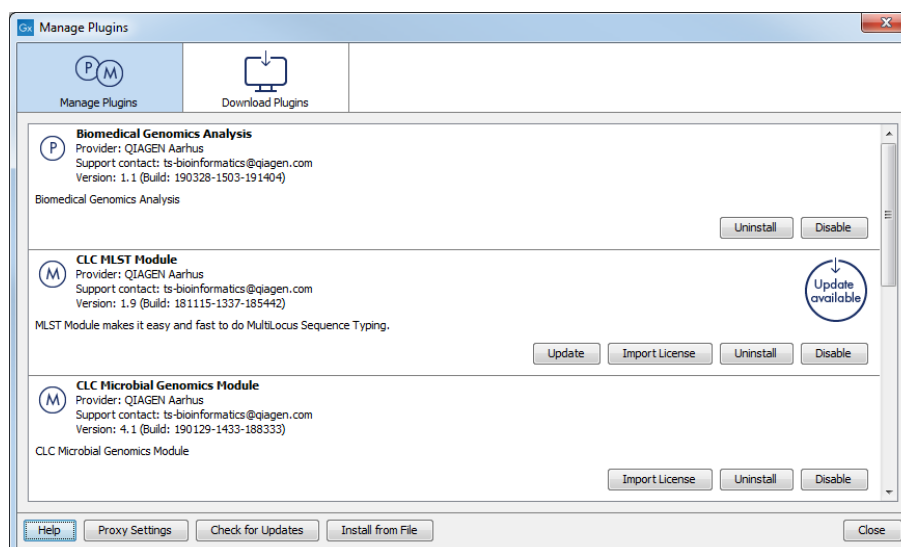


Figure 1.2: Installed plugins and modules are listed in the Plugins Manager under the Manage Plugins tab.

To uninstall a plugin or module, click on its entry in the list, and click on the **Uninstall** button.

Plugins and modules are not uninstalled until the Workbench is restarted. When you close the Plugin Manager, a dialog appears offering the opportunity to restart the *CLC Workbench*.

Disabling a plugin without uninstalling it

If you do not want a plugin to be loaded the next time you start the Workbench, select it in the list under the Manage Plugins tab and click on the **Disable** button.

1.4 Reference data management

Workflows need to be configured with the relevant reference data.

You can download reference data using the Reference Data Manager before running a workflow. For workflows configured to use particular Reference Data Sets, as the template workflows are, the download of such reference data can also be launched via the workflow wizard.

The following section covers aspects of the Reference Data Manager relevant when using the template workflows delivered with the CLC Haplotype Calling plugin. For the full documentation relating to this tool, please see the References management chapter of the CLC Genomics Workbench manual at https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=References_management.html.

1.4.1 The Reference Data Manager

The **QIAGEN Sets Reference Data Library** tab gives access to the reference data used with the CLC Haplotype Calling plugin template workflow. From the wizard you can download and configure the reference data. For the full documentation relating to QIAGEN Sets, please see the QIAGEN Sets chapter of the CLC Genomics Workbench manual at https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QIAGEN_Sets.html

Haplotype Calling reference data in Reference Data Manager

Reference data for **QIAseq Human Identity panels** is available for download in the element **QIAseq Human Identity Panels hg19**.

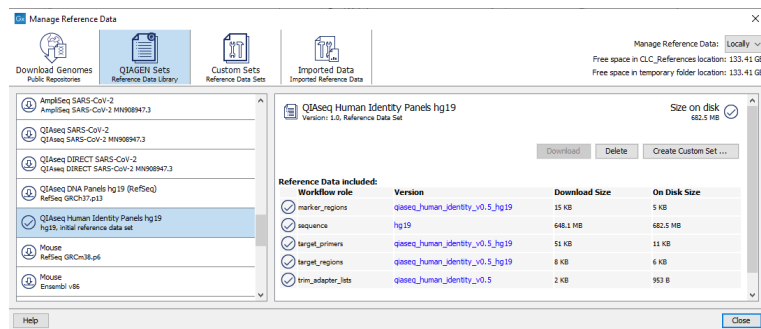


Figure 1.3: Reference data for QIAseq Human Identity Panels in the Reference Data Manager.

Part II

Import and Export

Chapter 2

Data import

Contents

2.1 Import	12
2.1.1 VCF	12
2.1.2 Standard CLC Variant Track	13

This chapter describes only import functionality specific to the CLC Haplotype Calling.

2.1 Import

Genotype tracks can be imported from:

- VCF
- Standard CLC variant track

2.1.1 VCF

Using **Import Variants (VCF) (beta)** it is possible to import variants in a VCF file to **Genotype track** format. Below are some significant differences in the way Import Variants (VCF) (beta) handles VCF compared to the standard VCF importer in the workbench:

- If phasing information is encoded in the VCF file as described in the VCF 4.3 specifications (<https://samtools.github.io/hts-specs/VCFv4.3.pdf>) this information is retained in the Genotype track.
- When filters are specified in the VCF, the corresponding elements in the Genotype track will have applied filters and be hidden as default.
- The option "Optimize for large tracks" allows import of very large VCF files without requiring excessive amounts of memory.
- There is no detection of reference overlap (complex variants) implemented yet.
- It is not possible to import VCF files in batches.

Learn more about how the workbench importer handles VCF format here: http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/1203/index.php?manual=VCF_import.html

To import variants from VCF to Genotype track format launch the tool **Import Variants (VCF) (beta)** using the Import button or the Launch button. It is not possible to start the tool from the Toolbox or from Import Tracks from File which is used for standard VCF import.

File | Import () | Import Variants (VCF) (beta)...

In the wizard step Settings specify the VCF file and the Reference Track. For very large VCF files (e.g. dbSNP) check the option Optimize for large tracks. This option allows import of very large files by reducing memory consumption at the cost of speed and disk space.

2.1.2 Standard CLC Variant Track

It is possible to convert a standard CLC variant track to a **Genotype track**. As no phasing information is available in the standard variant track, this will also not be available in the resulting Genotype track.

To convert a variant track to a Genotype track run the Convert to Genotype Track (beta) tool:

Toolbox | Track Tools () | Convert to Genotype Track (beta) ()

Once the tool wizard has opened, choose one or more variant tracks and press **Next**. You can choose to save or open the results. Clicking **Finish** will start the conversion of the variant track.

Chapter 3

Data export

Contents

3.1 Export Genotype Track as VCF	14
3.2 Export Allele Table as CSV	15
3.3 Export Marker Genotypes as CSV or TSV	15

Exporters exist for the following:

- VCF
- Allele table
- Marker genotypes

3.1 Export Genotype Track as VCF

Using **VCF (Genotype track) (beta)** it is possible to export variants in a Genotype track to VCF file. Below are the main differences between VCF (Genotype track) (beta) and the standard VCF exporter in the workbench:

- Only Genotype tracks can be exported.
- Variant files are exported in VCF 4.3 format.
- Phasing information from the Genotype track is encoded in the VCF.
- The Genotype track locus table (see section 4.2) reflects very well what the contents of the exported VCF will be.
- It is not possible to enforce ploidy.
- Complex variants are represented as described for Reference overlap in standard VCF export: http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/1203/index.php?manual=Complex_variant_representations_VCF_reference_overlap.html.

Learn more about the standard VCF exporter here: http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/1203/index.php?manual=Export_in_VCF_format.html.

To export variants in a Genotype track to VCF, click Export in the Toolbar and choose the tool VCF (Genotype track) (beta). Specify the Genotype tracks that should be exported and click Next. This will open a dialog as shown in figure 3.1.

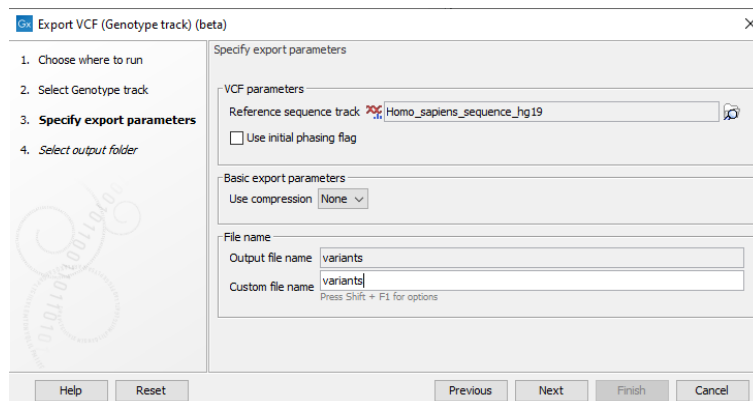


Figure 3.1: Setting parameters for export.

The following parameters can be adjusted in the dialog:

- **Reference sequence track:** Since the VCF format specifies that reference and allele sequences cannot be empty, deletions and insertions have to be padded with bases from the reference sequence. The export needs access to the reference sequence track in order to find the neighboring bases.
- **Use Initial phasing flag:** Enable this option to specify phasing status for the first allele in haploid and mixed phasing genotypes by using an initial phasing flag (e.g. GT=|1 and GT=/0|1|2). Disable to increase compatibility with external applications at the cost of lost phasing information. When disabled, some phased alleles in mixed phasing genotypes are specified as unphased, and phased haploid genotypes are specified using a missing allele (e.g. GT=1|. and GT=0/1|2).

3.2 Export Allele Table as CSV

It is possible to export the allele table of a Genotype track to CSV format.


Launch the standard export functionality by clicking on the Export button on the toolbar, or selecting the menu option: **File | Export** (📄). Select CSV as export format and select the tool **Allele CSV (Genotype track) (beta)**.

Once the tool wizard has opened, choose one or more Genotype tracks and press **Next**. In the following steps it is possible to specify if all or selected columns should be exported and you can choose to save or open the results. Clicking **Finish** will start the export of the allele table.

3.3 Export Marker Genotypes as CSV or TSV

It is possible to export genotypes present in a Genotype track for a specified set of markers, to a flexible CSV or TSV format that enables import in various external applications, e.g. for ancestry,

phenotype, and kinship analysis.

Launch the standard export functionality by clicking on the Export button on the toolbar, or selecting the menu option: **File | Export** . Select TSV or CSV as export format and select the tool **Marker genotypes (beta)**.

Once the tool wizard has opened, choose one or more Genotype tracks and press **Next**. In the following steps it is possible to specify which markers should be exported and you can choose format details for the exported file. Clicking **Finish** will start the export of the marker genotypes.

Part III

Haplotype Calling

Chapter 4

Genotype Track

Contents

4.1 Genome Model	18
4.2 Locus Table	20
4.3 Allele Table	21
4.4 Header View	23
4.5 Track View	24

The Genotype track uses concepts similar to those in the VCF specification to facilitate import and export of the commonly used format (<https://samtools.github.io/hts-specs/VCFv4.3.pdf>).

Genotype tracks exist in either sample or database form, corresponding to VCF files with or without the FORMAT column, respectively.

Databases such as ClinVar, dbSNP, and Cosmic are usually made available in VCFs without the FORMAT and sample specific columns. Database variants are without genome context, in the sense that it varies from sample to sample if they are heterozygous or homozygous and which alleles at other loci they form haplotypes with. These variants are also referred to as conceptual variants, and the annotations they possess (VCF INFO column) are typically database or population specific as opposed to specific for a single sample.

Sample Genotype tracks describe the genome of a single sample. Sample variants can have both database and sample specific annotations and genome context is provided in genotypes and haplotypes.

4.1 Genome Model

The Genotype track genome model consists of four elements as shown in figure 4.1. A variant locus and an allele variant always have a conceptual component and may have a sample specific component. Haplotype alleles represent instances of allele variants in a sample, and when haplotype alleles are present in the same DNA molecule they form a haplotype together (i.e. the haplotype alleles are phased).

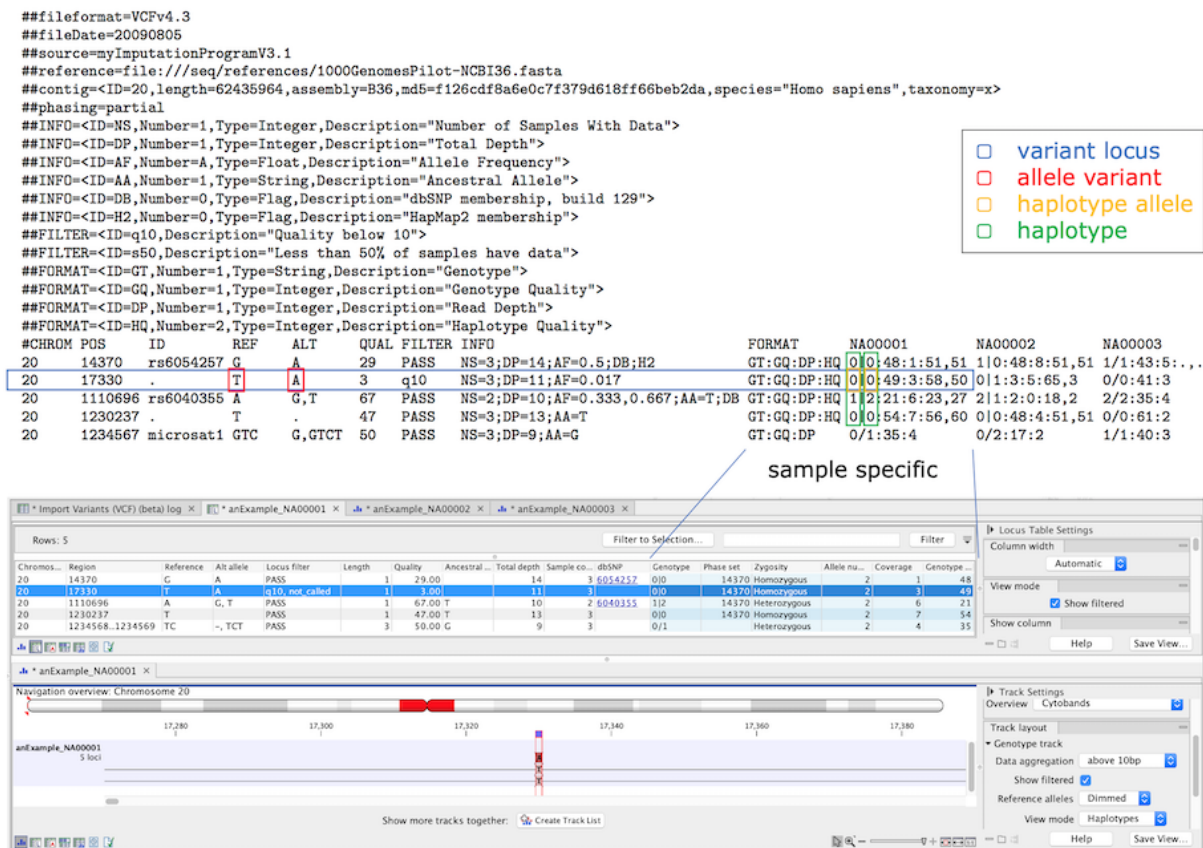


Figure 4.1: The four elements in the genome model and their relation to VCF. The variant locus has a number of allele variants that in a sample has a number of haplotype alleles, and phased haplotype alleles at several loci form a haplotype. In the lower part, the VCF has been imported and a Genotype track locus table and track view are showing a filtered locus. Sample specific annotations are colored blue in the Genotype track tables.

Filtering Just as it is possible to specify filters in the FILTER column of a VCF file, it is possible to apply filters to a locus in the Genotype track. Additionally, allele variants and haplotype alleles can have filters directly applied.

Elements can also have indirectly applicable filters if either the parent element or all child elements have applied filters. For example an allele variant has indirectly applicable filters if all constituent haplotype alleles have applied filters.

That a filter applies indicates that the element should be ignored and is not considered to be present in the sample. Elements with applicable filters are not shown by default, though can be displayed by enabling 'Show filtered' in the side panels of tables and track view (figure 4.1).

Read about variant filtering in the Microhaplotype Caller under [Microhaplotype Caller - Filtering](#).

Complex representation In a sample Genotype track, multiple overlapping loci together constitute a complex locus. For haplotype alleles in different overlapping loci that describe the same haplotype, there is typically a number of loci with the preferred representation of the allele, and the remaining loci have overlap alleles for that haplotype. Overlap alleles are marked with the 'Overlap allele' annotation (see section 4.3). The actual sequence may be unavailable for

the overlap alleles, in which case they are represented as a chosen overlap variant. The overlap allele representation in the Genotype track is 'reference overlap', meaning that overlap alleles where the actual sequence is unavailable are represented as the reference allele variant. Other used overlap allele representations are star alleles and symbolic alleles, however these options are not available in the Genotype track yet.

4.2 Locus Table

The locus table displays all Genotype track loci, one per row, and the displayed columns are locus annotations as default. Loci where filters apply are displayed when the option 'Show filtered' is enabled. Blue columns are sample specific annotations, corresponding to FORMAT annotations in VCF.

Below are descriptions of general locus table annotations. Common locus table annotations created by a specific tool can be found here: [Microhaplotype Caller - Annotations](#)

Chromosome The name of the reference sequence where the variant locus is situated.

Region The reference sequence positions of the variant locus. The region may be either a 'single position', a 'region' or a 'between position region'.

Reference The reference allele nucleotide sequence of the locus. Maximally 20 nucleotides are shown. Longer sequences can be obtained in their entirety by copy-pasting the table cell.

Alt allele Alternatives to the reference allele.

Filter List of filters that are directly or indirectly applicable to the locus. The value 'PASS' specifies that the variant passed all filters.

Alt count Number of non-reference allele variants at this locus.

Length Maximum length of all alleles at the locus.

Sample annotations

The following annotations are available in all sample Genotype tracks.

Genotype The VCF genotype that lists alleles present in the sample genome at this locus. Alleles with applied filters are hence excluded from the genotype.

The genotype is encoded as allele values separated by either of '/' or '|'. The allele values are 0 for the reference allele, 1 for the first allele listed in the 'Alt allele' annotation, 2 for the second allele listed in the 'Alt allele' annotation and so on. For diploid calls examples could be 0/1, 1|0, or 1/2, etc.

Genotype phasing compatible with all ploidy levels

One way to interpret VCF phasing encoded in the genotype (GT) field, is to consider the separators (/ or |) as phasing flags for the following allele, similar to the way phasing is encoded in BCF. For example in case of GT=0/1|2/3 we would know that 1 and 3 are unphased while 2 is phased. This interpretation, however, leaves the phasing status of the first GT allele poorly defined.

Assumptions are frequently made about the phasing status of the first allele in diploid scenarios. For example GT=0|1 is commonly interpreted to mean that both alleles are phased, and GT=0/1 that both alleles are unphased.

Considering the above, the genotype is written so that:

- At loci where all alleles except the first have same phasing status: the appropriate phasing flag is prepended. For example GT=|0/1/2 or GT=/0|1|2 or GT=/0|1
- At loci where alleles have mixed phasing status, and the first allele is phased: the appropriate phasing flag is prepended. For example GT=|0|1/2 or GT=|0/2
- In the case of haploid loci: if the allele is phased, either the appropriate phasing flag is prepended (e.g. GT=|1), or a missing allele is appended with the appropriate phasing flag (e.g. GT=1|.).

Thus, when encountering a genotype where the first allele has no prepended phasing flag, we can determine phasing status of the first allele to be:

- phased, if phasing flags are present and indicate that all other alleles are phased.
- unphased, in all other cases.

Phase set Identifier for a set of phased genotypes that together describe a set of overlapping haplotypes.

Zygoty The zygoty of the sample genome locus. This will be 'Homozygous' when only one allele variant is called at the locus, and 'Heterozygous' when more than one variant is called.

Allele number Total number of alleles in called genotype.

Complex Complex region, if locus is part of a complex of overlapping loci.

4.3 Allele Table

The allele table displays all Genotype track alleles, one variant per row, and the displayed columns are allele variant and haplotype allele annotations as default. Variants where filters apply are displayed when the option 'Show filtered' is enabled. Blue columns are sample specific annotations, corresponding to FORMAT annotations in VCF.

Below are descriptions of general allele table annotations. Common allele table annotations created by a specific tool can be found here: [Microhaplotype Caller - Annotations](#)

Chromosome The name of the reference sequence where the allele is situated.

Region The reference sequence positions of the locus. The region may be either a 'single position', a 'region' or a 'between position region'.

Type Allele variants are classified into five different types:

- SNV. A single nucleotide variant. This is also often referred to as a SNP.
- MNV. A multi nucleotide variant, which has the same number of nucleotides as the reference allele.

- **Insertion.** A variant that solely differs from the reference by having one or more adjacent bases inserted. The reference allele in a locus with zero length is called a reference insertion and also has type Insertion.
- **Deletion.** A variant that solely differs from the reference by having one or more adjacent bases deleted.
- **Replacement.** A variant that differs from the reference by length, and by having one or more bases replaced by one or more bases. This is also referred to as a delins. An example could be AAA->CC.

Reference The reference allele nucleotide sequence of the locus. Maximally 20 nucleotides are shown. Longer sequences can be obtained in their entirety by copy-pasting the table cell.

Allele The allele. Nucleotide sequence is shown if this is a simple small variant.

Reference allele Describes whether the allele is identical to the reference with a 'Yes' or 'No'. This can be useful for table filtering and sorting.

Alteration Length Maximum length of the allele and the reference allele.

Filter' List of filters that are directly or indirectly applicable to the allele variant. The value 'PASS' specifies that the variant passed all filters.

Sample annotations

The following annotations are available in sample Genotype tracks.

Allele count Count of alleles at this locus, identical to this allele variant. Based on separation of haplotypes the variant is part of, as well as the number of haplotype copies.

Haplotypes Haplotype identifiers. All sample alleles are part of one or more haplotypes. Haplotypes are groups of phased alleles that are linked together in the same DNA molecule. An allele may be part of multiple haplotypes describing different aspects of the same DNA molecule.

Locus number Haplotype locus number. The number of loci this haplotype spans, including filtered.

Separation Haplotype separation. Identifiers of all incompatible haplotypes, listed for each haplotype and in the same order. Two haplotypes are incompatible if they are separated by having different alleles at any position. Any enforced separation is included, whereas single locus haplotypes are not included (unless enforced).

Enforced separation Enforced haplotype separation. Identifiers of separated haplotypes, listed for each haplotype and in the same order. Haplotype separation may be enforced based on knowledge about haplotype incompatibility in parts of the chromosome that are not included in the Genotype track, for example when a homozygous locus that is part of two separated haplotypes is extracted using 'Create Track from Selection'.

Copies Number of identical allelic copies represented by a haplotype. Listed for each haplotype and in the same order. Defaults to one and includes any additional enforced copies.

Enforced copies Enforced haplotype copies. Number of additional identical allelic copies represented by a haplotype. Listed for each haplotype and in the same order. Haplotype copies can for example be enforced based on an expected ploidy.

Haplotype allele filter Applicable haplotype allele specific filters.

Overlap allele Presence of this annotation indicates that this is an overlap allele, meaning that parts of the allele sequence at this locus are described by alleles at overlapping loci, and that the allele sequence at this locus may be unavailable.

The annotation value '<NS>' indicates that the haplotype allele sequence at this locus is not available and must be derived from other loci in the complex region. The haplotype allele is in that case represented as the overlap variant specified by the track **complex representation**.

When the sequence is available for this overlap allele, then the annotation value specifies the actual haplotype allele sequence. The overlap allele can in that case either be represented as the proper allele variant with actual sequence, or as the overlap variant specified by the track **complex representation**.

Note that when the overlap allele is represented as the overlap variant, *the actual haplotype allele sequence may differ from that of the variant it is represented as*.

While specification of overlap allele sequence may be considered redundant, other allele annotations can provide locus specific information that is not.

In Track view, overlap alleles are only shown in 'Allele count'-mode, and only when the actual sequence is available.

4.4 Header View

The header view shows details of Genotype track annotations and filters. The following information is provided:

Key The annotation or filter ID.

Type The annotation value type. Square brackets indicates a list.

Export tag Tag used as ID when the annotation is exported, for example to VCF.

Category Specifies whether it is a sample/database annotation at the locus/variant/alt variant/haplotype allele/haplotype level, or a filter.

Mode Calculation mode of the annotation value:

- NOT_CALCULATABLE - Annotation values can only be stored, not calculated.
- CALCULATE_AND_STORE - Values are calculated and stored when the track is created.
- CALCULATE_ONTHEFLY_IF_STORED_VALUE_MISSING - If an element has the annotation, calculate the value on-the-fly if there is no stored value, otherwise use the stored value.
- CALCULATE_ONTHEFLY_NEVER_STORE - If an element has the annotation, calculate the value on-the-fly when requested.

All annotated Are all elements that fit the category considered to have the annotation, or must each annotated element carry the annotation key.

Importance Affects default display of the annotation:

- **HIGH** - annotation will be shown as default in all tables.
- **MEDIUM** - annotation will be shown as default in the appropriate table.
- **LOW** - annotation will not be shown as default in any table.

CLC reserved Is this a reserved CLC annotation.

Description Description of the annotation or filter.

4.5 Track View

The Track view of the Genotype track has the following options in the Track layout side panel:

Data aggregation Allows you to specify whether the information in the track should be shown in detail or whether you wish to aggregate data. By aggregating data you decrease the detail level but increase the speed of the data display process, which is of particular interest when working with big data sets. The threshold (in bp) for when data should be aggregated can be specified with the drop-down box. The threshold describes the unit (or "bucket") size in base pairs, above which the data will start being aggregated. The bucket size depends on the track length and the zoom level. Hence, a data aggregation threshold with a low value will only show details when zoomed in, whereas a high value means that you can see details even when zoomed out. Please note that when using the high values, it will take longer time to display the data on the screen.

Show filtered When enabled, alleles with applicable filters will be shown crossed out. Filtered alleles are hidden as default.

Reference alleles Show reference alleles in a dimmed color to make them more distinguishable.

View mode The view mode selection specifies whether the multiplicity of the displayed allele is based on its allele count in the sample (see 'Allele count' annotation in section 4.3), or on the number of haplotypes it is part of. An allele may be part of multiple haplotypes describing different aspects of the same DNA molecule (see 'Haplotypes' annotation in section 4.3).

- **Allele count** For each locus the alleles are displayed according to their allele count in the sample. Haplotype details are not available in this view mode.
- **Haplotypes** Haplotypes are displayed as linked haplotype alleles. Annotation and filter details are available in the tooltip when hovering over a haplotype allele. If the haplotypes are not separate (see 'Separation' annotation in section 4.3) they may describe the same DNA molecule in the sample genome.

Annotation color Makes it possible to change the allele color.

Chapter 5

Microhaplotype Caller

Contents

5.1 Microhaplotype Caller - Method	25
5.2 Microhaplotype Caller - Options	26
5.3 Microhaplotype Caller - Filters	27
5.3.1 General filters	27
5.3.2 Detection filters	27
5.3.3 Noise filters	28
5.4 Microhaplotype Caller - Annotations	30

The Microhaplotype Caller identifies sample alleles at variant loci as well as specified marker loci, uses phasing information available in the mapped reads to infer haplotypes, and produces a **Genotype track** with all detected alleles, including, for inspection, those not called due to applied filters.

5.1 Microhaplotype Caller - Method

The Microhaplotype Caller takes advantage of the same probabilistic approach as the Low Frequency Variant Detection tool, it is therefore suitable for analysis of mixed tissue samples in which low frequent variants are likely to be present, as well as for samples for which the ploidy is unknown or not well defined. For method details, see: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=_Low_Frequency_Variant_Detection_tool_Models_methods.html

Key features distinguishing the Microhaplotype Caller method:

- The long variant detection responsible for calling MNVs, replacements, and indels in the Low Frequency Variant Detection tool, has been adapted to also call longer more general haplotypes. This enables high resolution allele detection, so multiple haplotype alleles may be distinguished per allele variant and reported with detailed annotations.
- Detected alleles are reported in a locus based representation, making genotypes readily available, e.g. for VCF export.

- When filters are applied, the affected alleles are retained for easy user inspection.
- Forced variant loci can be specified to allow detection of alleles at any site, also homozygous reference loci.

The current version of the Microhaplotype Caller has certain limitations:

- If phasing regions become too large the workbench may run out of memory. Caution must therefore be taken when increasing the 'Maximum phasing distance' parameter from the default value (see section 5.2).
- There is not yet any special homopolymer handling, so length variation artifacts must be considered when analysing loci around longer homopolymers.

5.2 Microhaplotype Caller - Options

The Microhaplotype Caller has three options to select information available in the output (figure 5.1):

- **Forced loci:** Report alleles for the specified loci, including homozygous reference alleles. Alleles detected at other variant loci will still be reported. Either a Genotype track or an Annotation track can be selected.
- **Maximum phasing distance:** The maximum positional distance from one locus to nearest neighbour locus, to include them in the same phasing region. For example, to ensure alleles are phased within a codon, set this parameter to 2.
WARNING: Setting this parameter high may lead to performance issues. This parameter must be kept around its default value when a high density of variant loci is expected to be detected, for example due to a combination of high read coverage, poor read quality, and high sensitivity detection parameters.
- **Restrict calling to target regions:** Variant detection will only be performed in the specified regions.

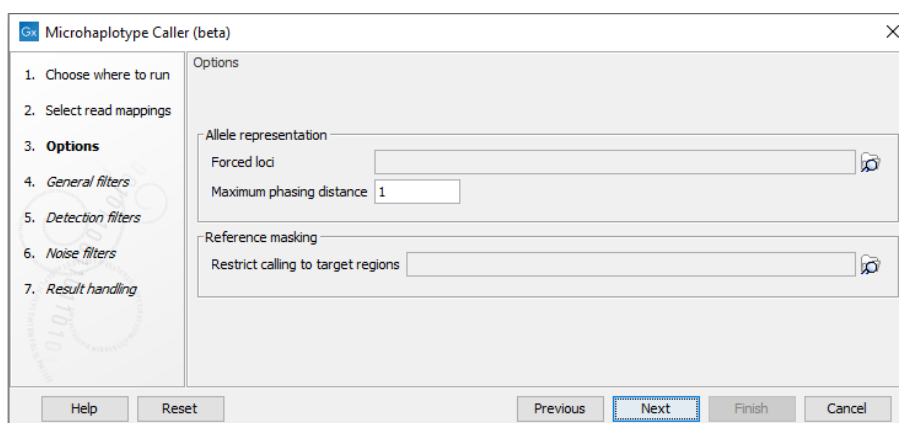


Figure 5.1: The Microhaplotype Caller options.

5.3 Microhaplotype Caller - Filters

5.3.1 General filters

There are three main properties by which the output can be filtered, and each of these have a detection and a call threshold (figure 5.2). The detection threshold specifies if elements are reported or not, whereas the call threshold specifies whether an element is marked with a filter to be disregarded as noise. Elements with applied filters are however still reported for inspection, which can be useful for evaluating borderline calls.

- **Minimum count (detection):** Alleles observed in less than this number of reads will not be detected and can therefore not be reported.
- **Minimum count (call):** Alleles observed in less than this number of reads will not be called though will be reported as filtered if detected.
- **Minimum allele fraction (detection):** Alleles observed at less than this fraction will not be detected and can therefore not be reported.
- **Minimum allele fraction (call):** Alleles observed at less than this fraction will not be called though will be reported as filtered if detected.
- **Minimum haplotype quality (detection):** Haplotypes with lower quality will not be detected and can therefore not be reported.
- **Minimum haplotype quality (call):** Haplotypes with lower quality will not be called though will be reported as filtered if detected.

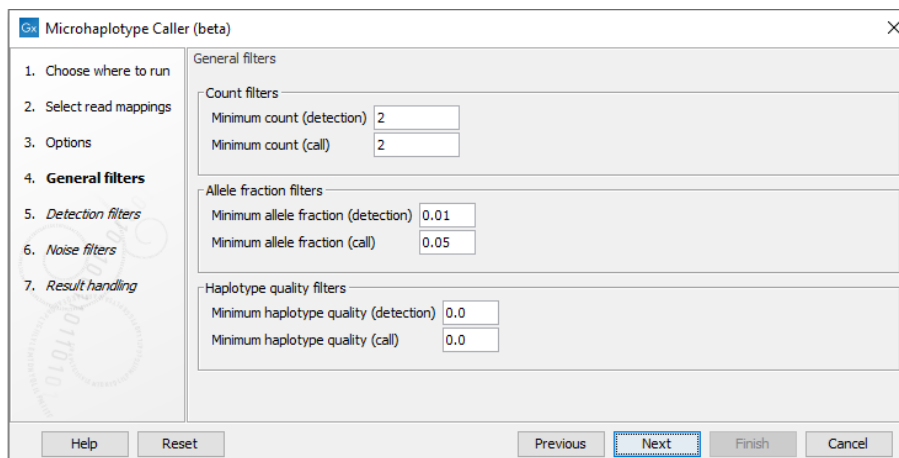


Figure 5.2: The Microhaplotype Caller general filters.

5.3.2 Detection filters

The detection filters affect the sensitivity of the Microhaplotype Caller (figure 5.3):

- **Required Significance:** this parameter determines the cut-off value for the statistical test for the variant not being due to sequencing errors. Only variants that are at least this significant will be called. The lower you set this cut-off, the fewer variants will be called.

- **Ignore broken pairs:** When enabled, reads from broken pairs are ignored. Broken pairs may arise for a number of reasons, one being erroneous mapping of the reads. In general, variants based on broken pair reads are likely to be less reliable, so ignoring them may reduce the number of spurious variants called. However, broken pairs may also arise for biological reasons (e.g. due to structural variants) and if they are ignored some true variants may go undetected. Please note that ignored broken pair reads will not be considered for any non-specific match filters.
- **Non-specific match filter:** Non-specific matches are likely to come from repeat region whose exact mapping location is uncertain. In general, variants based on non-specific matches are likely to be less reliable. However, as there are regions in the genome that are entirely perfect repeats, ignoring non-specific matches may have the effect that true variants go undetected in these regions.

There are three options for specifying to which 'extent' the non-specific matches should be ignored:

- 'No': they are not ignored.
 - 'Reads': they are ignored.
 - 'Region': when this option is chosen no variants are called in regions covered by at least one non-specific match. In this case, the minimum length of reads that are allowed to trigger this effect has to be stated, as really short reads will usually be non-specific even if they do not stem from repeat regions.
- **Base quality filter:** The base quality filter can be used to ignore the reads whose nucleotide at the potential variant position is of dubious quality. This is assessed by considering the quality of the nucleotides in the region around the nucleotide position. There are three parameters to determine the base quality filter:
 - **Neighborhood radius:** This parameter determines the region size. For example if a neighborhood radius of five is used, a nucleotide will be evaluated based on the nucleotides that are 5 positions upstream and 5 positions downstream of the examined site, for a total of 11 nucleotides. Note that, near the end of the reads, eleven nucleotides will still be considered by offsetting the region relative to the nucleotide in question.
 - **Minimum central quality:** Reads whose central base has a quality below the specified value will be ignored. This parameter does not apply to deletions since there is no 'central base' in these cases.
 - **Minimum neighborhood quality:** Reads for which the minimum quality of the bases is below the specified value will be ignored.

5.3.3 Noise filters

These filter thresholds specify when a filter is applied to a genome element for it to be disregarded as noise (figure 5.4):

- **Minimum coverage:** Only variants in regions covered by at least this many reads are called.

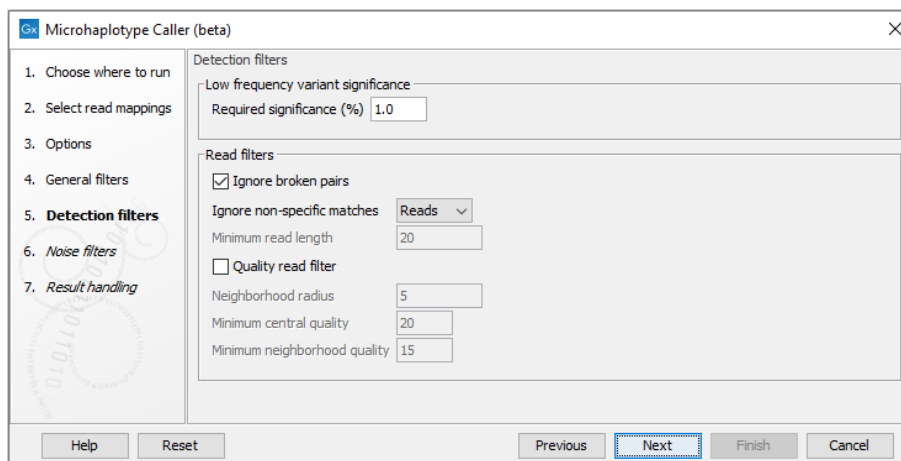


Figure 5.3: The Microhaplotype Caller detection filters.

- **Maximum coverage:** All positions with coverage above this value will be ignored when inspecting the read mapping for variants. The option is highly useful in cases where you have a read mapping which has areas of extremely high coverage as are areas around centromeres for example.
- **Minimum average base quality:** Alleles that have an average base quality below this threshold are disregarded as noise.
- **Read direction filter:** The read direction filter removes alleles that are almost exclusively present in either forward or reverse reads. For many sequencing protocols such alleles are most likely to be the result of amplification induced errors. Note, however, that the filter is **NOT suitable for amplicon data**, as for this you will not expect coverage of both forward and reverse reads. The filter has a single parameter:
 - **Direction frequency:** A filter is applied to alleles that are not supported by at least this frequency of reads from each direction.
- **Relative read direction filter:** The relative read direction filter attempts to do the same thing as the 'Read direction filter', but does this in a statistical, rather than absolute, sense: it tests whether the distribution among forward and reverse reads of the variant carrying reads is different from that of the total set of reads covering the site. The statistical, rather than absolute, approach makes the filter less stringent. The filter has one parameter:
 - **Significance:** A filter is applied to alleles whose read direction distribution is significantly different from the expected with a test at this level. The lower you set the significance cut-off, the fewer alleles will be filtered out.
- **Read position filter:** The read position filter is a filter that attempts to remove systematic errors in a similar fashion as the 'Read direction filter', *but* that is also **suitable for hybridization-based data**. It removes alleles that are located differently in the reads carrying it than would be expected given the general location of the reads covering the variant site. This is done by categorizing each sequenced nucleotide (or gap) according to the mapping direction of the read and also where in the read the nucleotide is found; each read is divided in five parts along its length and the part number of the nucleotide is recorded. This gives a total of ten categories for each sequenced nucleotide and a given site will have a distribution between these ten categories for the reads covering the

site. If a distinct allele is present in the site, you would expect the allele nucleotides to follow the same distribution. The read position filter carries out a test for whether the read position distribution of the allele carrying reads is different from that of the total set of reads covering the site. The filter has one parameter:

- **Significance:** A filter is applied to alleles whose read position distribution is significantly different from the expected with a test at this level. The lower you set the significance cut-off, the fewer alleles will be filtered out.

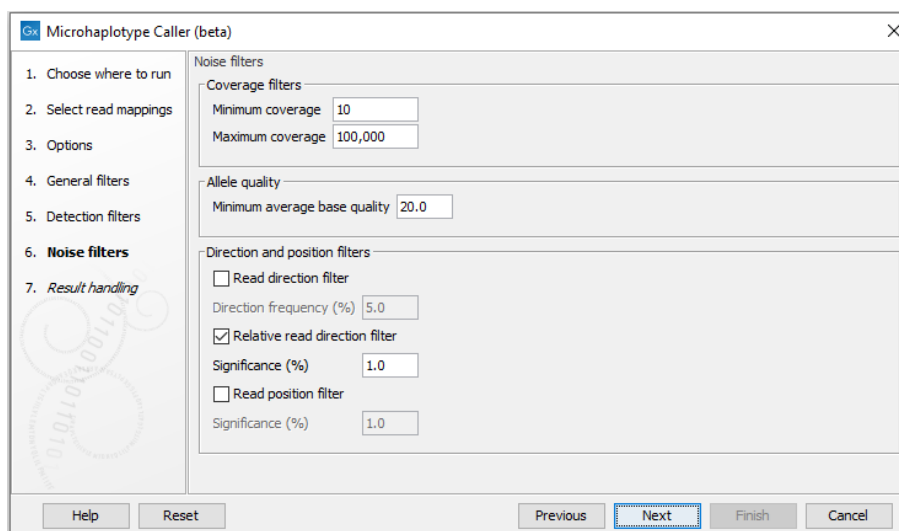


Figure 5.4: The Microhaplotype Caller noise filters.

5.4 Microhaplotype Caller - Annotations

The Microhaplotype Caller output (figure 5.5) is in many ways similar to that of the Low Frequency Variant Detection tool, though a significant difference is that the main annotation level is the high-resolution haplotype alleles. Some annotations are given both on allele variant and haplotype allele level, exemplified by Count and Count', respectively, where the apostrophe indicates the higher detail annotation level.

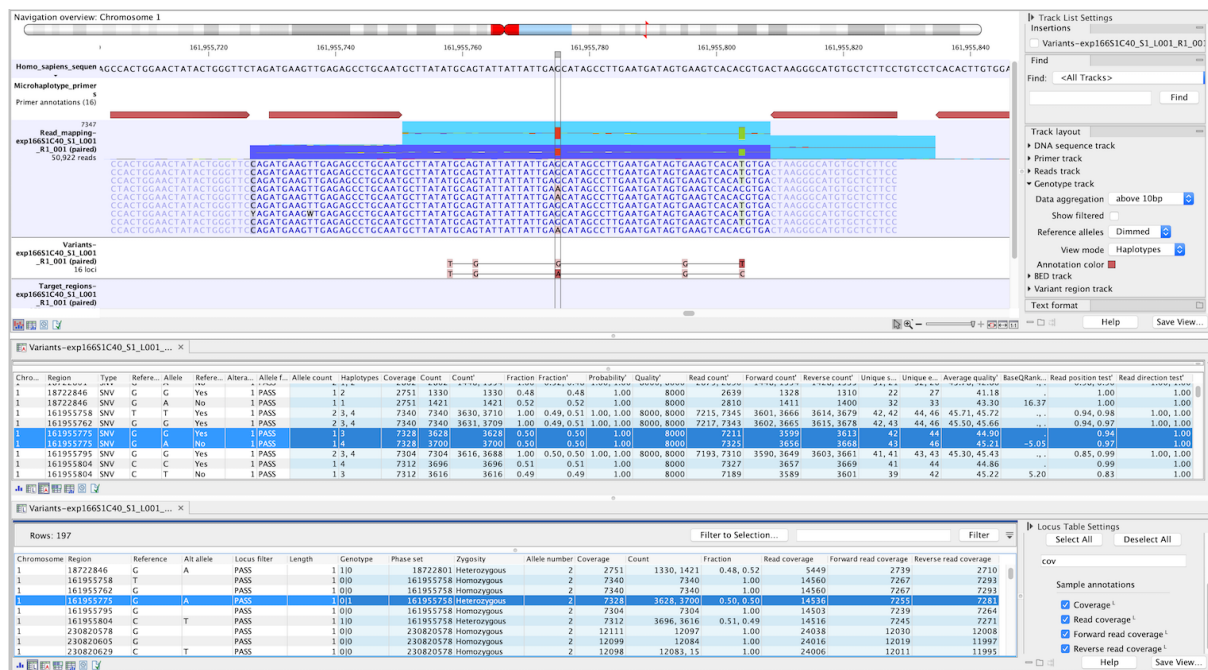


Figure 5.5: Genotype track produced by the Microhaplotype Caller. The figure shows a track list (top) with sequence, primer, reads and Genotype track. The Genotype track was produced by running the Microhaplotype Caller on the reads track. The Genotype track has been opened in two separate table views (locus and allele) by double-clicking on it in the track list and selecting the desired view in the lower left corner. By selecting a row in the Genotype track table, the track list view is centered on the corresponding locus.

Locus annotations

The following locus annotations are created by the Microhaplotype Caller. Note that general annotations present in all Genotype tracks are described in section 4.2.

Coverage The **fragment** coverage at this position. Only **countable fragments** are considered.

Read coverage The read coverage at this position. Only **countable** reads are considered.

Forward and **Reverse read coverage** The **countable** forward or reverse reads covering this locus.

Allele annotations

The following allele annotations are created by the Microhaplotype Caller. Note that general annotations present in all Genotype tracks are described in section 4.3.

Count Number of **fragments** supporting the allele variant, also known as the allele depth. When the allele variant is present in the genotype, counts of haplotype alleles with applied filters are excluded.

Count' Number of **fragments** supporting the haplotype allele.

Fraction Allele variant fraction at locus, calculated as the number of **fragments** supporting this variant divided by the total number of **fragments** supporting all called alleles.

Fraction' Haplotype allele fraction at locus, calculated as the number of **fragments** supporting this allele divided by the total number of **fragments** supporting all called alleles.

Probability' The Microhaplotype Caller makes statistical tests for the various possible explanations for each position. For a given single position variant, the probability is calculated as the sum of probabilities for all the explanations containing that variant. So if a G variant is called, the reported probability is the sum of probabilities for these configurations: G, A/G, C/G, G/T, A/C/G, A/G/T, C/G/T, and A/C/G/T (and also all the configurations containing deletions together with G). For multi position alleles (e.g. deletions) and haplotypes in general, an estimate is made of the probability of observing the same read data if the allele or haplotype did not exist and all observations were due to sequencing errors (based on the generated sequencing error model). The probability column contains one minus this estimated probability. If this value is less than 50%, the variant might as well just be the result of sequencing errors and it is not reported at all.

Quality' Measure of the significance of an allele, i.e., a quantification of the evidence (read count) supporting the allele, relative to the coverage and what could be expected to be seen by chance, given the error rates in the data.

Quality is calculated as $-10\log_{10}(1-p)$, p being the allele probability. Qual is capped at 8000 for $p=1$, with 8000: highly significant, 0: insignificant. A Quality value of 10 indicates a 1 in 10 chance that the called allele is an error, while a Quality of 100 indicates a 1 in 10^{10} chance that the called allele is an error.

Read count' The number of reads supporting the allele. Only **countable** reads are considered. Note that each read in an overlapping pair contribute 1.

Forward and **Reverse read count'** The number of **countable** forward or reverse reads supporting the allele.

Forward/reverse balance' The minimum of the forward and the reverse fraction of all reads supporting the allele. Some systematic sequencing errors can be triggered by a certain combination of bases. This means that sequencing one strand may lead to sequencing errors that are not seen when sequencing the other strand. In order to evaluate whether the distribution of forward and reverse reads is approximately random, this value is calculated as the minimum of the number of forward reads divided by the total number of reads and the number of reverse reads divided by the total number of reads supporting the variant. An equal distribution of forward and reverse reads for a given allele would give a value of 0.5.

Unique starts' The number of unique start positions for **countable fragments** that support the allele. This value can be important to look at in cases with low coverage. If all reads supporting the allele have the same start position, you could suspect that it is a result of an amplification error.

Unique ends' The number of unique end positions for **countable fragments** that support the allele. This value can be important to look at in cases with low coverage. If all reads supporting the allele have the same end position, you could suspect that it is a result of an amplification error.

Average quality' The average base quality score of the bases supporting a haplotype allele. The average quality score is calculated by adding the Q scores of the nucleotides supporting the haplotype allele, and dividing this sum by the number of nucleotides. In the case

of a deletion, the quality score reported is the lowest average quality of the two bases neighboring the deletion. Similarly for insertions, the quality in reads where the insertion is absent is inferred from the lowest average of the two bases on either side of the position.

If there are no values in this column, it is probably because the sequencing data was imported without quality scores.

BaseQRankSum' The BaseQRankSum column contains an evaluation of the quality scores in the reads that have a called allele compared with reference allele quality scores. Reference alleles and variants for which no corresponding reference allele is called do not have a BaseQRankSum value. The score is a z-score derived using the Mann-Whitney U test, so a value of -2.0 indicates that the observed qualities for the allele are two standard deviations below what would be expected if they were drawn from the same distribution as the reference allele qualities. A negative BaseQRankSum indicates an allele with lower quality than the reference, and a positive z-score indicates higher quality than the reference.

Read position test' The test probability for the test of whether the distribution of read positions of this allele in the supporting reads is different from that of all the reads covering the locus.

Read direction test' The test probability for the test of whether the distribution among forward and reverse reads of the allele carrying reads is different from that of all the reads covering the locus. This value reflects a balanced presence of the allele in forward and reverse reads (1: well-balanced, 0: un-balanced). This p-value is based on a statistic that we assume follows a Chi-square(df=2) distribution under the null hypothesis of the allele having equal frequency on reads from both direction. Note that GATK uses a Fisher's exact test for the same purpose. The difference between both approaches lead to a potential overestimation of p-values output by the workbench's variant detection tools.

Chapter 6

Glossary

6.1 Glossary

Complex region A region with two or more overlapping loci.

Countable 'Countable' reads are sometimes also referred to as filtered reads. Which reads are 'countable' depends on the user settings when variant calling is performed - if e.g. the user has chosen 'Ignore broken pairs', reads belonging to broken pairs are not 'countable'.

Fragment A DNA fragment that gives rise to a single read or read pair. Note that, although overlapping paired reads have two reads in their overlap region, they are only counted as one when counting fragments in that region.

VCF Variant Call Format. The file format specification can be found at GitHub <https://samtools.github.io/hts-specs/>.

Part IV

Prebuilt workflows

Chapter 7

Human Identity workflows

Contents

7.1 Detect QIAseq Human Identity SNPs and Microhaplotypes	36
--	-----------



7.1 Detect QIAseq Human Identity SNPs and Microhaplotypes

The workflow Detect QIAseq Human Identity SNPs and Microhaplotypes (beta) (figure 7.1) can both be used to analyse data from SNP and microhaplotype panels. The default parameter values are set for mixed sample sensitivity that allows detection of a human genome present in 10% of the reads.

Note:

- In order to detect alleles present in 5% of the reads, the read coverage of the locus must be at least 200.
- There is not yet any special homopolymer handling, so length variation artifacts must be considered when analysing loci around longer homopolymers.

The Detect QIAseq Human Identity SNPs and Microhaplotypes (beta) workflow can be found here:

Toolbox | Template Workflows | Biomedical Workflows  | **Detect QIAseq Human Identity SNPs and Microhaplotypes (beta)** 

Double-click on the workflow to run the analysis.

The following parameters can be adjusted when running the workflow:

Minimum group size Increase to disregard UMI reads with low number of supporting reads.

Minimum average quality score Increase to disregard UMI reads with low quality.

Maximum phasing distance The default value is set for microhaplotypes. It can be set to 1 when analysing SNP panels where marker loci are close to each other, and phasing is not of interest. This may for example be useful when analysing data from the Ancestry and VISAGE SNP panels. Reducing this value may decrease processing time.

Minimum count (call) Read support threshold for called alleles. Alleles with fewer supporting reads can be inspected by enabling 'Show filtered' in the side panels.

Minimum allele fraction (call) Allele fraction threshold for called alleles. Increase this parameter (e.g. to 0.15) to remove noise when the sample only contains the genome of a single individual.

Minimum haplotype quality (call) Quality threshold for called haplotype alleles. Haplotypes with lower quality can be inspected by enabling 'Show filtered' in the side panels.

Minimum coverage Loci with lower coverage are marked as filtered and can be inspected by enabling 'Show filtered' in the side panels.

Minimum average base quality Apply a filter to an allele when UMI read bases supporting the allele have lower average quality than this threshold.

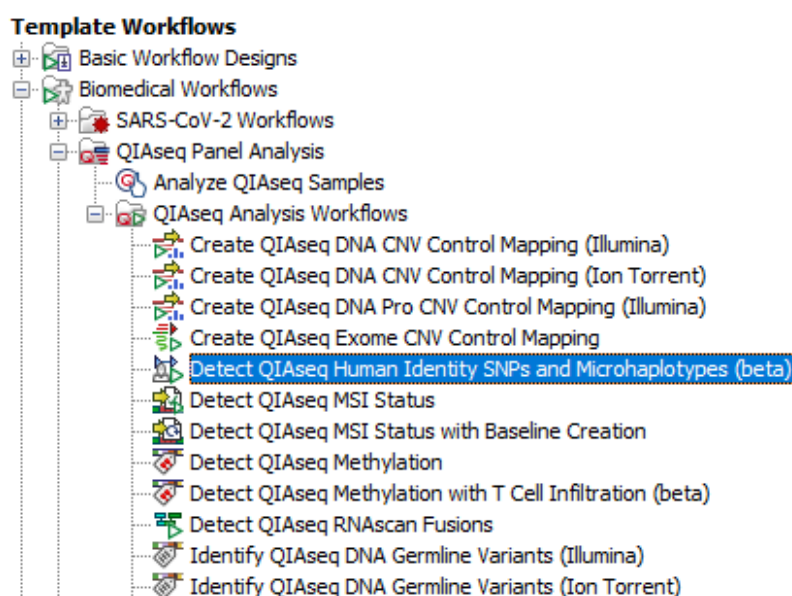


Figure 7.1: The QIaseq SNP and microhaplotype panel analysis workflow.

Bibliography