

CLC **Single Cell Analysis** Module

USER MANUAL

User manual for CLC Single Cell Analysis Module 25.0

Windows, macOS and Linux

March 6, 2025

This software is for research purposes only.

QIAGEN Aarhus AS
Kalkværksvej 5, 11.
DK - 8000 Aarhus C
Denmark



Contents

I	Introduction	10
1	Introduction	11
1.1	The concept of CLC Single Cell Analysis Module	11
1.2	Contact information	12
1.3	System requirements	13
1.4	Installing modules	13
1.4.1	Licensing modules	15
1.4.2	Uninstalling modules	16
1.5	Installing server extensions	17
1.5.1	Licensing server extensions	19
2	Reference data management	21
2.1	QIAGEN Sets	22
3	Running tools and workflows	26
II	Import and Export	27
4	Data import	28
4.1	Import Immune Reference Segments	29
4.1.1	IMSEQ	30
4.1.2	IMGT	31
4.1.3	Output from Import Immune Reference Segments	32
4.2	Import Cell Annotations	34
4.3	Import Cell Clusters	35
4.4	Import Cell Clonotypes	36

4.5	Import Expression Matrix	37
4.5.1	HDF5 formats	40
4.5.2	Other formats	42
4.6	Import Peak Count Matrix	44
4.7	Import Space Ranger	47
4.8	Cell format in importers	48
4.9	On-the-fly import in workflows	51
5	Data export	53
5.1	Export Cell Ranger HDF5 Expression Matrix	54
5.2	Export AnnData Expression Matrix	55
5.3	Export h5Seurat Expression Matrix	55
5.4	Export Loom Expression Matrix	56
5.5	Export MEX Expression Matrix	56
5.6	Export Plain Text Table Expression Matrix	57
5.7	Export Peak Count Matrix	57
III	Single Cell Analysis	59
6	Prepare Reads	60
6.1	Annotate Single Cell Reads	60
6.1.1	Read structure	61
6.1.2	Barcodes from names	63
6.1.3	Barcode correction	63
6.1.4	Sample name	65
6.1.5	The output of Annotate Single Cell Reads	65
7	Creating a Gene Expression Matrix	69
7.1	Single Cell RNA-Seq Analysis	69
7.1.1	The Single Cell RNA-Seq Analysis report	72
7.1.2	The Single Cell RNA-Seq Analysis algorithm	76
7.2	QC for Single Cell	77
7.2.1	Empty droplets filter	78

7.2.2	Count-based and extra-chromosomal filters	80
7.2.3	Doublets filter	82
7.2.4	The output of QC for Single Cell	84
7.2.5	Choosing barcodes to retain	92
7.2.6	Cell calling	93
7.2.7	Automatic thresholds	94
7.2.8	Doublet calling	95
7.3	Demultiplex Parse Bio Samples	96
7.4	Normalize Single Cell Data	98
7.4.1	When is batch correction appropriate?	100
7.4.2	The output of Normalize Single Cell Data	100
7.4.3	The Normalize Single Cell Data algorithm	103
7.5	The Expression Matrix element	105
8	Cell Type Classification	108
8.1	Browse QIAGEN Cell Ontology	108
8.2	Predict Cell Types	109
8.2.1	The output of Predict Cell Types	111
8.2.2	Cell type refinement	112
8.3	Train Cell Type Classifier	113
8.3.1	Features used for training and prediction	116
8.3.2	The output of Train Cell Type Classifier	117
8.3.3	SVMs for cell type classification	120
8.4	Update Cell Type Classifier	120
8.5	The Cell Type Classifier element	122
9	Expression Analysis	124
9.1	Differential Expression for Single Cell	124
9.1.1	The output of Differential Expression for Single Cell	127
9.1.2	The differential expression algorithm	128
9.2	Create Expression Plot	128
9.2.1	The Heat Map output of Create Expression Plot	131
9.2.2	The Dot Plot output of Create Expression Plot	132

9.2.3 The Violin Plot output of Create Expression Plot	133
10 Velocity Analysis	137
10.1 Single Cell Velocity Analysis	138
10.1.1 The output of Single Cell Velocity Analysis	140
10.1.2 The velocity estimation algorithm	140
10.2 Differential Velocity for Single Cell	141
10.3 Score Velocity Genes	142
10.3.1 The output of Score Velocity Genes	143
10.4 Create Phase Portrait Plot	143
10.5 The Velocity Matrix element	144
11 Spatial Transcriptomics	147
11.1 The Spatial Transcriptomics Plot element	147
12 Chromatin Accessibility	151
12.1 Single Cell ATAC-Seq Analysis	151
12.1.1 The output of Single Cell ATAC-Seq Analysis	153
12.1.2 The report output from Single Cell ATAC-Seq Analysis	153
12.1.3 The Single Cell ATAC-Seq Analysis algorithm	156
12.2 Split Read Mapping by Cell	158
12.3 Differential Accessibility for Single Cell	162
12.3.1 The differential accessibility algorithm	162
12.4 The Peak Count Matrix element	164
13 Immune Repertoire	166
13.1 Single Cell V(D)J-Seq Analysis	168
13.1.1 The report output from Single Cell V(D)J-Seq Analysis	169
13.1.2 The clonotype identification algorithm	170
13.2 Filter Cell Clonotypes	172
13.3 Combine Cell Clonotypes	174
13.4 Compare Cell Clonotypes	175
13.4.1 The output of Compare Cell Clonotypes	176
13.5 Convert Clonotypes to Cell Annotations	177

13.6 The Cell Clonotypes element	179
13.6.1 Primary and secondary clonotypes	179
13.6.2 Cell Clonotypes tables	180
13.6.3 Cell Clonotypes alignments	182
13.6.4 Cell Clonotypes Sankey plot	185
14 Noise reduction through feature selection and dimensionality reduction	188
14.1 Feature selection and dimensionality reduction	188
14.1.1 Calculation of estimated biological variation	191
15 Cell Annotation	193
15.1 Cluster Single Cell Data	193
15.1.1 The Cluster Single Cell Data algorithm	195
15.2 Create Heat Map for Cell Abundance	195
15.2.1 The output of Create Heat Map for Cell Abundance	196
15.3 Create Cell Annotations from Hashtags	200
15.4 The Cell Annotations element	201
15.5 The Cell Clusters element	202
16 Dimensionality reduction	203
16.1 UMAP for Single Cell	203
16.2 tSNE for Single Cell	207
17 Single cell low-dimensional plots functionality	209
17.1 Manual annotation	210
17.2 Visualizing different types of matrices	221
17.3 Create Subset	223
17.4 Extract to Table	224
17.5 Launching of Create Expression Plot	225
17.6 Launching of Create Heat Map for Cell Abundance	226
17.7 Launching of Differential Accessibility for Single Cell	227
17.8 Launching of Differential Expression for Single Cell	228
17.9 Launching of Differential Velocity for Single Cell	229
17.10 Launching of Score Velocity Genes	230

18 Utility tools	231
18.1 Combine Cell Annotations	231
18.2 Update Cell Annotations	232
18.3 Convert Metadata to Cell Annotations	233
18.4 Combine Cell Clusters	234
18.5 Update Cell Clusters	234
18.6 Add Information to Plot	236
18.7 Update Single Cell Sample Name	236
 IV Template Workflows	 240
19 Single cell template workflows from reads	241
19.1 Expression Analysis from Reads	241
19.1.1 Configuring the batch units for Expression Analysis from Reads	243
19.1.2 Output from Expression Analysis from Reads	245
19.2 Chromatin Accessibility Analysis from Reads	246
19.2.1 Configuring the batch units for Chromatin Accessibility Analysis from Reads	247
19.2.2 Output from Chromatin Accessibility Analysis from Reads	247
19.3 Chromatin Accessibility and Expression Analysis from Reads	250
19.3.1 Configuring the batch units for Chromatin Accessibility and Expression Analysis from Reads	251
19.3.2 Output from Chromatin Accessibility and Expression Analysis from Reads .	252
19.3.3 Importing reads	252
19.4 Immune Repertoire Analysis from Reads (10xV(D)J)	257
19.4.1 Output from Immune Repertoire Analysis from Reads (10xV(D)J)	258
19.5 Immune Repertoire and Expression Analysis from Reads (10xV(D)J)	259
19.5.1 Configuring the batch units for Immune Repertoire and Expression Analysis from Reads (10xV(D)J)	260
19.5.2 Output from Immune Repertoire and Expression Analysis from Reads (10xV(D)J)	261
 20 Single cell template workflows from imported data	 264
20.1 Expression Analysis from Matrix	264
20.1.1 Output from Expression Analysis from Matrix	266

20.2 Chromatin Accessibility and Expression Analysis from Matrix	266
20.2.1 Output of Chromatin Accessibility and Expression Analysis from Matrix . .	268
20.3 Immune Repertoire and Expression Analysis from Clonotypes and Matrix	268
20.3.1 Output from Immune Repertoire and Expression Analysis from Clonotypes and Matrix	270
Bibliography	272

Part I

Introduction

Chapter 1

Introduction

Contents

1.1 The concept of CLC Single Cell Analysis Module	11
1.2 Contact information	12
1.3 System requirements	13
1.4 Installing modules	13
1.4.1 Licensing modules	15
1.4.2 Uninstalling modules	16
1.5 Installing server extensions	17
1.5.1 Licensing server extensions	19

Welcome to CLC Single Cell Analysis Module 25.0 – a software package supporting your daily bioinformatics work.

1.1 The concept of CLC Single Cell Analysis Module

CLC Single Cell Analysis Module 25.0 enables the study of single cell RNA (scRNA-Seq) data, including RNA velocity, spatial transcriptomics, Assay for Transposase-Accessible Chromatin (scATAC-Seq) samples, and T and B cell receptors (scTCR-Seq and scBCR-Seq, collectively known as scV(D)J-Seq). Tools are provided for analyzing the different types of data both separately and jointly.

scRNA-Seq

Tools are available for quality control and normalization, noise reduction and feature selection, clustering, cell type prediction, and differential expression. UMAP and tSNE plots can be overlaid with clusters, predicted cell types, or the expression of individual genes. Marker genes can be identified through analyses of differential gene expression and by gathering information from expression plots. Alternatively, cell type classifiers can be trained from pre-labeled cells. Two pre-trained classifiers are provided, and can be extended. Further, velocity analysis can be performed for data including both spliced and un-spliced reads, and an interactive phase portrait plot is produced. In addition, high-dimensional vector that predicts the future state of individual cells are added to the dimensionality reduction plots. The provided workflows can be easily

adjusted to fit the chemistry and protocol of the data and are a good starting point from either raw FASTQ or an imported Expression Matrix.

Spatial transcriptomics

A tool is available for importing spatial transcriptomics data from Space Ranger spatial outputs. The resulting plot can be overlaid with clusters, predicted cell types, or the expression of individual genes. Additionally, the plot can be linked to a UMAP or tSNE plot, such that the same visualization can be applied simultaneously to both plots.

scATAC-Seq

A complete pipeline from peak calling and footprinting to analysis of differential accessibility is provided. The peak read mappings can be slit into minor sub-populations and visualized in a Tracklist. It is also possible to generate UMAP and tSNE plots from the peak matrix. Further, three workflows are provided starting either from reads or imported matrices.

scV(D)J-Seq

After the initial identification of clonotypes in the sample, these can be further filtered, combined across samples and the sample-level immune repertoires can be compared with regards to diversity estimates, gene usage, etc. UMAP and tSNE plots from matched scRNA-Seq data can be overlaid with clonotype information, once this is converted to Cell Annotations. The provided workflows are a good starting point from either raw FASTQ or imported Cell Clonotypes. Additionally, workflows are available for the joint analysis of both scRNA-Seq and scV(D)J-Seq data.

Selection of algorithms

The algorithms implemented have been selected to be the best performing at the time of development as assessed by independent paper reviews. All algorithms have been re-implemented in Java with the aim of being able to scale to large data sets and run on a wide range of hardware. Internal benchmarks have been conducted to select the best performing algorithm for predicting cell types, which is one of the key features in this software package. The manual provides detailed descriptions of the chosen algorithms, how to adjust parameters for better performance, and how to interpret results.

The CLC Single Cell Analysis Module is frequently updated. A detailed list of new features, improvements, bug fixes, and changes is available at <https://digitalinsights.qiagen.com/clc-single-cell-analysis-module-latest-improvements/>.

1.2 Contact information

CLC Single Cell Analysis Module is developed by:

QIAGEN Aarhus A/S
Kalkværksvej 5, 11.
DK - 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: https://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 <https://qiagen.my.salesforce-sites.com/KnowledgeBase/KnowledgeNavigatorPage>

1.3 System requirements

In addition to meeting the system requirements of the *CLC Genomics Workbench* or the *CLC Genomics Server*, the following requirements must be met:


- 64 GB RAM recommended (32 GB RAM required).
- We recommend running the following on a *CLC Genomics Server*:
 - Single Cell RNA-Seq Analysis, see section 7.1.
 - Analysis of datasets exceeding 50000 cells.

Compatibility

The CLC Single Cell Analysis Module 25.0 plugin and the CLC Single Cell Analysis Server Extension 25.0 can be installed on *CLC Genomics Workbench* 25.0 and *CLC Genomics Server* 25.0, respectively, and on later versions in the same major release line.

1.4 Installing modules

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.

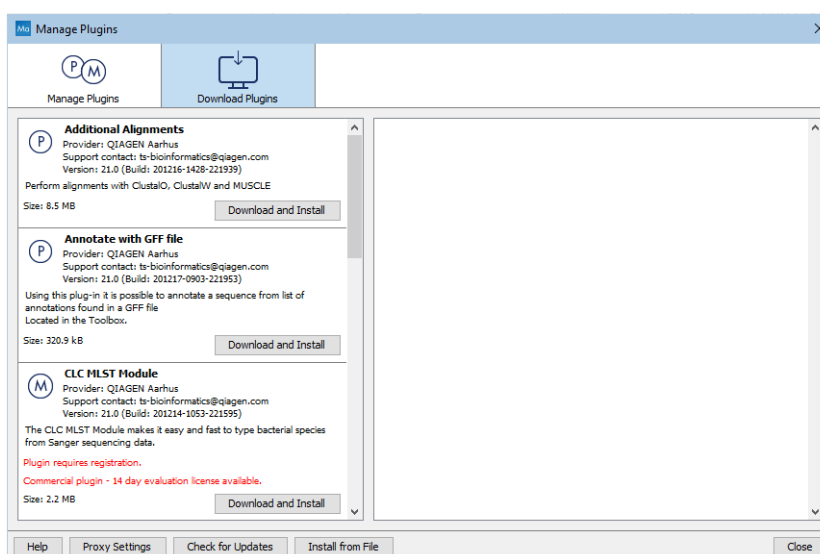


Figure 1.1: Plugins and modules available for installation are listed in the Plugin Manager under the Download Plugins tab.

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 Single Cell Analysis Module, 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*.

1.4.1 Licensing modules

When you have installed the CLC Single Cell Analysis Module and start a tool from that module for the first time, the License Assistant will open (figure 1.2).

The License Assistant can also be launched by opening the Workbench Plugin Manager, selecting the installed module from under the Manage Plugins tab, and clicking on the button labeled *Import License*.

To install a license, 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.

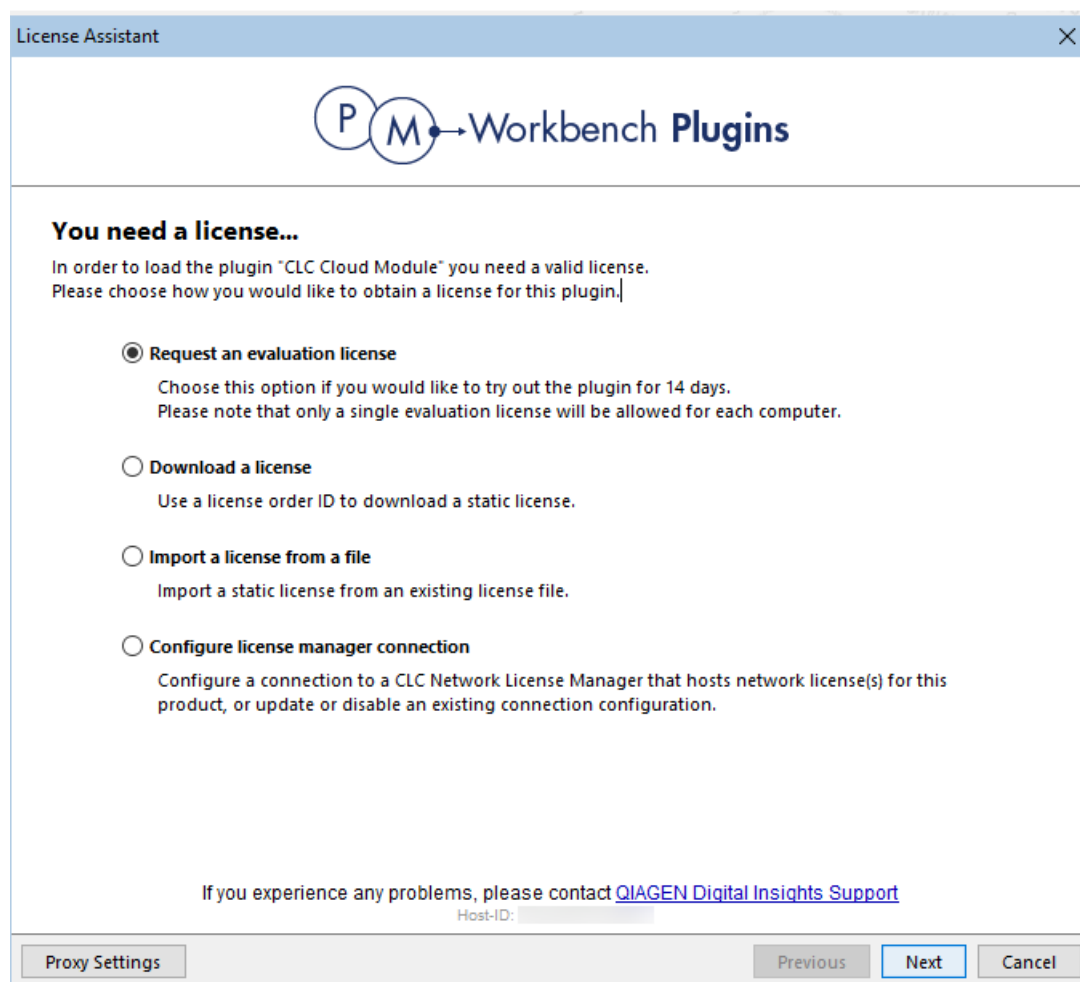


Figure 1.2: The License Assistant provides options for licensing modules installed on the Workbench.

The following options are available:

- **Request an evaluation license.** Request a fully functional, time-limited license.

- **Download a license.** Use the license order ID received when you purchased the software to download and install a license file.
- **Import a license from a file.** Import an existing license file, for example a file downloaded from the web-based licensing system.
- **Configure license manager connection.** If your organization has a *CLC Network License Manager*, select this option to configure the connection to it.

These options are described in detail in sections under https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Workbench_Licenses.html.

To download licenses, including evaluation licenses, your machine must have access to the external network. To install licenses on non-networked machines, please see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Download_static_license_on_non_networked_machine.html.

1.4.2 Uninstalling modules

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

Utilities | Manage Plugins... (P)

This will open the Plugin Manager (figure 1.3). Installed plugins and modules are shown under the Manage Plugins tab of the Plugins Manager.

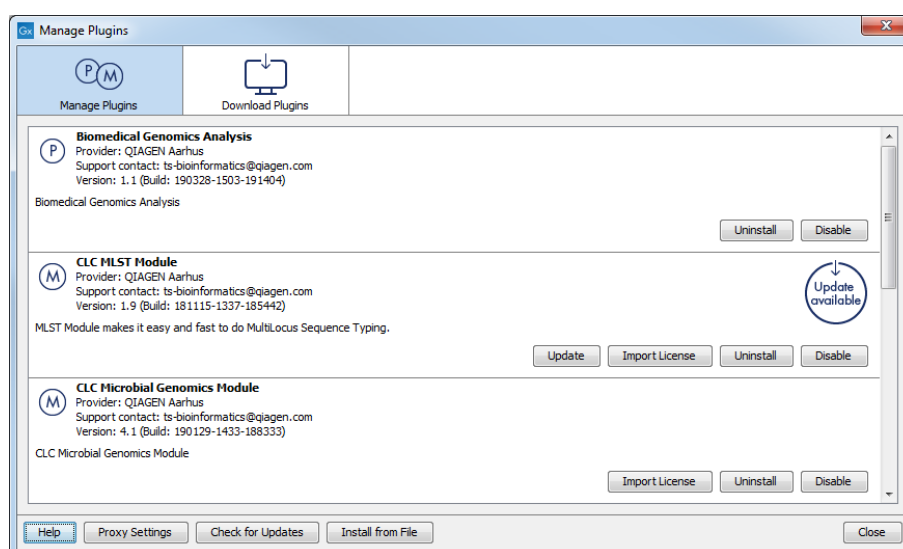


Figure 1.3: Installed plugins and modules are listed in the Plugins Manager under the Manage Plugins tab.

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.

1.5 Installing server extensions

To use the tools and functionalities of CLC Single Cell Analysis Module on a *CLC Server*:

1. You need to purchase a license to run tools delivered by the CLC Single Cell Analysis Server Extension.
2. A *CLC Server* administrator must install the license on the single server, or on the master node in a job node or grid node setup, as described in section 1.5.1.
3. A *CLC Server* administrator must install the CLC Single Cell Analysis Server Extension on the *CLC Server*, as described below.

Download and install server plugins and server extensions

Plugins, including server extensions (commercial plugins), are installed by going to the **Extensions** (🔧) tab in the web administrative interface of the single server, or the master node of a job node or grid node setup, and opening the **Download Plugins** (📁) area (figure 1.4).

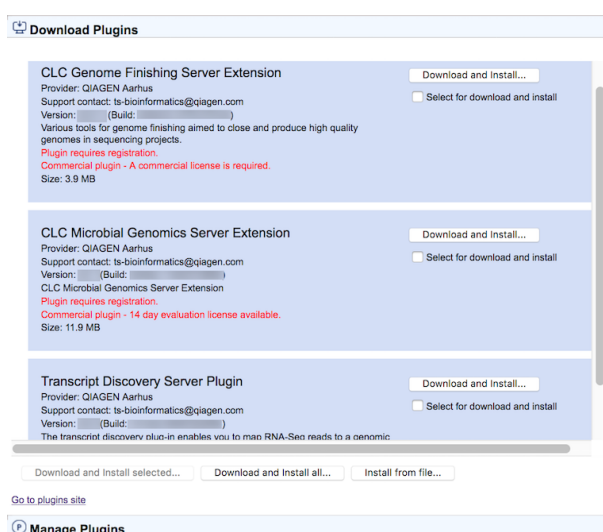



Figure 1.4: Installing plugins and server extensions is done in the Download Plugins area under the Extensions tab.

If the machine has access to the external network, plugins can be both downloaded and installed via the *CLC Server* administrative interface. To do this, locate the plugin in the list under the **Download Plugins** (📁) area and click on the **Download and Install...** button.

To download and install multiple plugins at once on a networked machine, check the "Select for download and install" box beside each relevant plugin, and then click on the **Download and Install All...** button.

If you are working on a machine without access to the external network, server plugin (.cpa) files can be downloaded from: <https://digitalinsights.qiagen.com/products->

[overview/plugins/](#) and installed by browsing for the downloaded file and clicking on the **Install from File...** button.

The *CLC Server* must be restarted to complete the installation or removal of plugins and server extensions. All jobs still in the queue at the time the server is shut down will be dropped and would need to be resubmitted. To minimize the impact on users, the server can be put into Maintenance Mode. In brief: running in Maintenance Mode allows current jobs to run, but no new jobs to be submitted, and users cannot log in. The *CLC Server* can then be restarted when desired. Each time you install or remove a plugin, you will be offered the opportunity to enter Maintenance Mode. You will also be offered the option to restart the *CLC Server*. If you choose not to restart when prompted, you can restart later using the option under the **Server maintenance** () tab.

For job node setups only:

- Once the *master CLC Server* is up and running normally, then restart each *job node CLC Server* so that the plugin is ready to run on each node. This is handled for you if you restart the server using the functionality under



Management () | Server maintenance ()

- In the web administrative interface on the *master CLC Server*, check that the plugin is enabled for each job node.

Installation and updating of plugins on connected job nodes requires that direct data transfer from client systems has been enabled, which is done by the *CLC Server* administrator, under the "External data" tab.

Grid workers will be re-deployed when a plugin is installed on the master server. Thus, no further action is needed to enable the newly installed plugin to be used on grid nodes.

Managing installed server plugins

Installed plugins can be updated or uninstalled, from under the **Manage Plugins** () area (figure 1.5), under the **Extensions** () tab.

The list of tools delivered with a server plugin can be seen by clicking on the **Plugin contents** link to expand that section. Workflows delivered with a server plugin are not shown in this listing.

Links to related documentation

- Logging into the *CLC Server* web administrative interface: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Logging_into_administrative_interface.html
- Maintenance Mode: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Server_maintenance.html
- Restarting the server: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Starting_stopping_server.html
- Plugins on job node setups: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Installing_Server_plugins_on_job_nodes.html

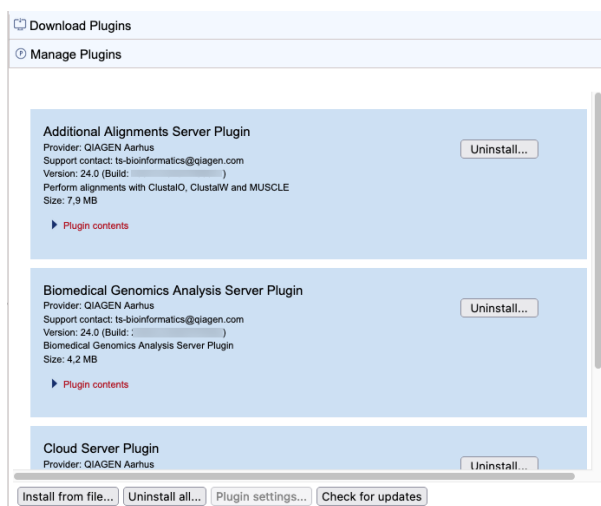


Figure 1.5: Managing installed plugins and server extensions is done in the Manage Plugins area under the Extensions tab. Clicking on Plugin contents opens a list of the tools delivered by the plugin.

- Grid worker re-deployment: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Overview_Model_II.html

Plugin compatibility with the server software

The version of plugins and server extensions installed must be compatible with the version of the CLC Server being run. A message is written under an installed plugin's name if it is not compatible with the version of the CLC Server software running.

When upgrading to a new major version of the CLC Server, all plugins will need to be updated. This means removing the old version and installing a new version.

Incompatibilities can also arise when updating to a new bug fix or minor feature release of the CLC Server. We recommend opening the **Manage Plugins** area after any server software upgrade to check for messages about the installed plugins.

Licensing server extensions is described in section 1.5.1.

1.5.1 Licensing server extensions

Licenses are installed on a single server or on the master node of a job node or grid node setup.

To download and install a license:

- Log into the web administrative interface of the single server or master node as an administrative user.
- Under the **Management** (🔧) tab, open the **Download License** (📄) tab.
- Enter the Order ID supplied by QIAGEN into the Order ID field and click on the "Download and Install License..." button (figure 1.6).

Please contact ts-bioinformatics@qiagen.com if you have not received an Order ID.

The CLC Server must be restarted for new license files to be loaded. Details about restarting can be found at https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Starting_stopping_server.html.

Each time you download a license file, a new file is created in the `licenses` folder under the CLC Server installation area. *If you are upgrading an existing license file, delete the old file from this area before restarting.*

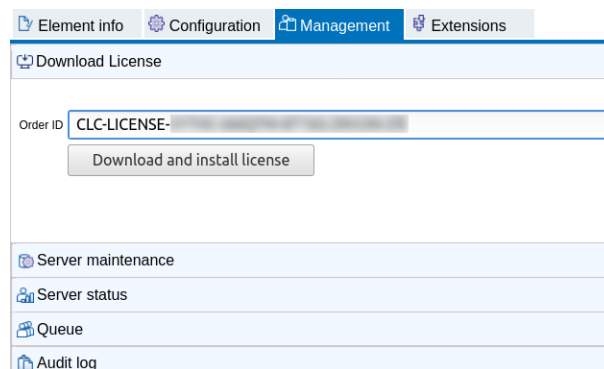


Figure 1.6: License management is done under the Management tab.

Chapter 2

Reference data management

Template workflows delivered by the CLC Single Cell Analysis Module are configured to use QIAGEN Reference Sets, making them simple to launch while helping ensure that the same reference data is used consistently. This reference data can be easily obtained using the Reference Data Manager in the *CLC Genomics Workbench*. Reference data for a specific workflow can also be downloaded via workflow launch wizards. These features are described in detail in section 2.1.

QIAGEN Sets for single cell analyses

Single cell data sets for Human and Mouse are called **Single Cell hg38 (Ensembl)** and **Single Cell Mouse (Ensembl)**, respectively, and are available under the **QIAGEN Sets** tab of the Reference Data Manager (figure 2.1).

The reference data sets contain:

- The reference sequence, gene track, and mRNA track, used for mapping scRNA-Seq data.
- A pre-trained classifier with cell types from QIAGEN Cell Ontology (see section 8.1) to use when predicting cell types or training with more cell types.
- A gene ontology that can be used together with differential expression data to analyze GO terms.
- Reference V (variable), D (diversity), J (joining) and C (constant) gene segments, used for mapping scTCR-Seq data.
- A peak shape filter used for calling scATAC-Seq peaks.

Further detail about working QIAGEN reference data is provided in section 2.1.

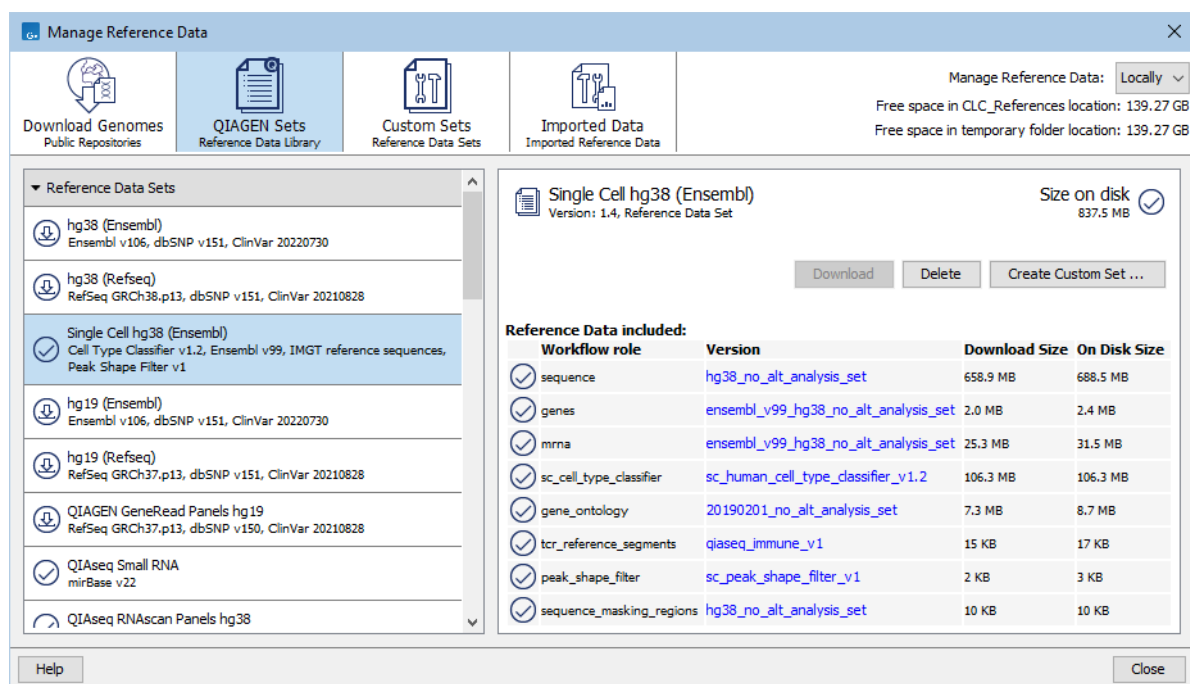


Figure 2.1: A Single Cell reference data set viewed via the QIAGEN Sets tab of the Reference Data Manager.

2.1 QIAGEN Sets

QIAGEN provides access to much common reference data using functionality under the **QIAGEN Sets** tab of the Reference Data Manager. Data is distributed as **Reference Data Elements**, which can be individually downloaded, or downloaded as part of a **Reference Data Set**. Many template workflows are configured to make use of QIAGEN Reference Sets, making them simple to launch while helping to ensure that the same reference data is used consistently. Using such workflows, the relevant reference data can also be downloaded via the workflow launch wizard (figure 2.2). When logged into a CLC Server with a CLC_References location defined, you can choose whether to download the data to the Workbench or Server.

Using the Reference Data Manager for QIAGEN reference data

To access **QIAGEN Sets**, open the Reference Data Manager by clicking on the **Manage Reference Data** (📁) button in the top Toolbar or go to the **Utilities** menu and select **Manage Reference Data** (📁). Then click on the **QIAGEN Sets** tab at the top left. Under this tab, there are subsections for **Reference Data Sets** and **Reference Data Elements** (figure 2.3).

When a Reference Data Set is selected, information about it is displayed in the right hand pane. This includes the size of the whole data set, and a table listing the workflow roles defined in the set, with information about the data element specified for each role. Further details about the element assigned to a role can be found by clicking on the link in the Version column. An icon to the left of each set indicates whether data for this set has already been downloaded (☑) or not (📄). The same icons are used to indicate the status of each element in a Reference Data Set (figure 2.4).

If you have permission to delete downloaded data, the **Delete** button will be enabled. When

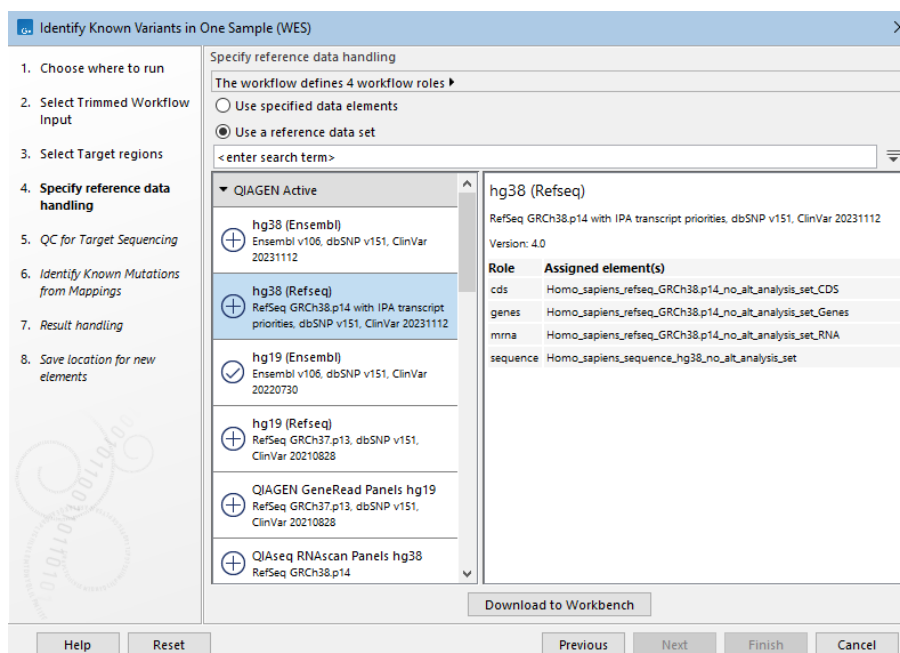


Figure 2.2: When launching workflows configured to use data from Reference Data Sets, the relevant reference data can be downloaded via the workflow launch wizard.

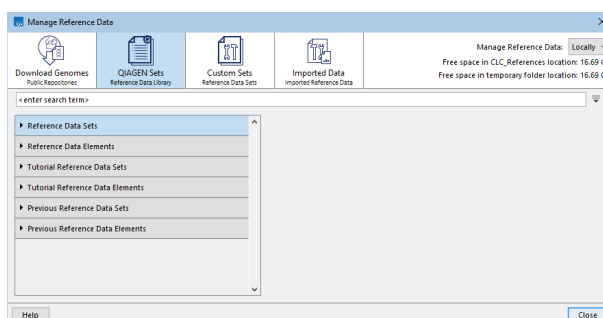


Figure 2.3: Subheadings under the QIAGEN Sets tab provide access to Reference Data Sets and Reference Data Elements

reference data is stored on a CLC Server, you need be logged in from the Workbench as an administrative user to delete reference data.

Searching for data available under the QIAGEN Sets tab

Use the search field under the top toolbar to search for terms in element and set names, workflow role names, and versions. To search for just an exact term, put the term in quotes.

The results include the name of the element or set the term was found in, followed in brackets by the tab it is listed under, e.g. (Reference Data Elements), (Tutorial Reference Data Sets), etc. Hover the cursor over a hit to see what aspect of the result matched the search term (figure 2.5). Double-click on a search result to open it.

Downloading resources

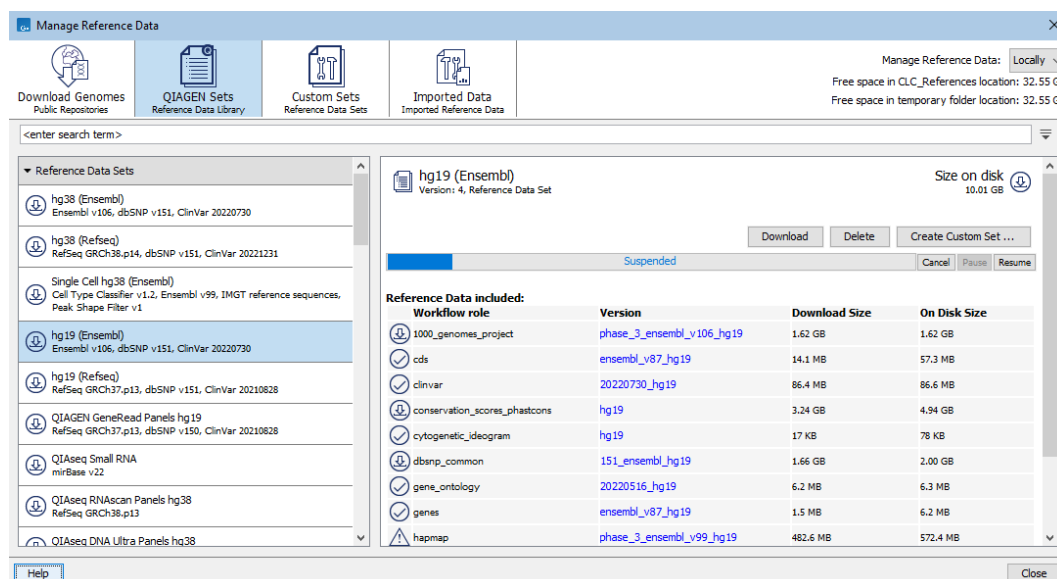


Figure 2.4: The elements in a Reference Data Set are being downloaded. The full size of the data set is shown at the top, right hand side. The size of each element is reported in the "On Disk Size" column. Below the row of tabs at the top is a search field that can be used to search for data sets or elements.

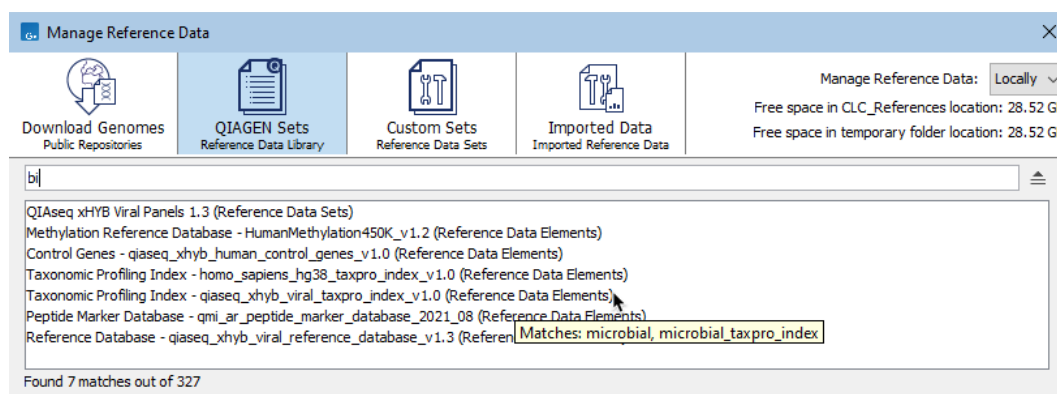


Figure 2.5: Terms entered in the search field when the QIAGEN Sets tab is selected are searched for in element and set names, workflow role names, and versions of the resources available under that tab. Hovering the cursor over a hit reveals a tooltip with information about the match.

To download a Reference Data Element or a Reference Data Set (i.e. all elements in that set), select it and click on the **Download** button.

The progress of the download is indicated and you have the option to **Cancel**, **Pause** or **Resume** the download (figure 2.4).

When the "Manage Reference Data" option at the top of the Reference Data Manager is set to "Locally", data is downloaded to the CLC_References location in the CLC Workbench. When set to "On Server", the data is downloaded to the CLC_References location in the CLC Server.

Additional information

The HapMap (<https://www.sanger.ac.uk/data/hapmap-3/>) databases contain more than one file. QIAGEN Reference Data Sets that include HapMap are initially configured with all the populations available. You can specify specific populations to use when launching a workflow, or you can create a custom reference set that contains only the populations of interest.

General information about Reference Data Sets, and creating Custom Sets, can be found at https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Reference_Data_Sets_defining_Custom_Sets.html.

General information about the Reference Data Manager is at https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=References_management.html.

Chapter 3

Running tools and workflows

When CLC Single Cell Analysis Module is installed, tools and template workflows for single cell analysis are made available. These include:

- Single cell data related importers. See chapter 4.
- Single cell data related exporters. See chapter 5
- Single cell analysis tools, available from the Single Cell Analysis folder under the Tools menu.
- Template workflows for analyzing scRNA-Seq, scATAC-Seq, and scV(D)J-Seq data, available from the Single Cell Workflows subfolder, under Template Workflows, in the Workflows menu.

Information specific to the tools and workflows provided by the CLC Single Cell Analysis Module is covered in this manual. The default settings in the tools and workflows provided are suitable for most use cases. However, some settings need to be adjusted to reflect the sample preparation or sequencing technology used.

General information about

- Launching tools and handling results can be found at: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_tools_handling_results_batching.html.
- Launching workflows can be found at: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Launching_workflows_individually_in_batches.html.

See also section 4.9 for information specific to importing single cell data on the fly in workflows.

If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

Part II

Import and Export

Chapter 4

Data import

Contents

4.1 Import Immune Reference Segments	29
4.1.1 IMSEQ	30
4.1.2 IMGT	31
4.1.3 Output from Import Immune Reference Segments	32
4.2 Import Cell Annotations	34
4.3 Import Cell Clusters	35
4.4 Import Cell Clonotypes	36
4.5 Import Expression Matrix	37
4.5.1 HDF5 formats	40
4.5.2 Other formats	42
4.6 Import Peak Count Matrix	44
4.7 Import Space Ranger	47
4.8 Cell format in importers	48
4.9 On-the-fly import in workflows	51

This chapter describes import functionality specific to the CLC Single Cell Analysis Module. For other importers, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Importing_data.html.

Clicking the Import button in the top toolbar will bring up a list of the available importers, see figure 4.1.

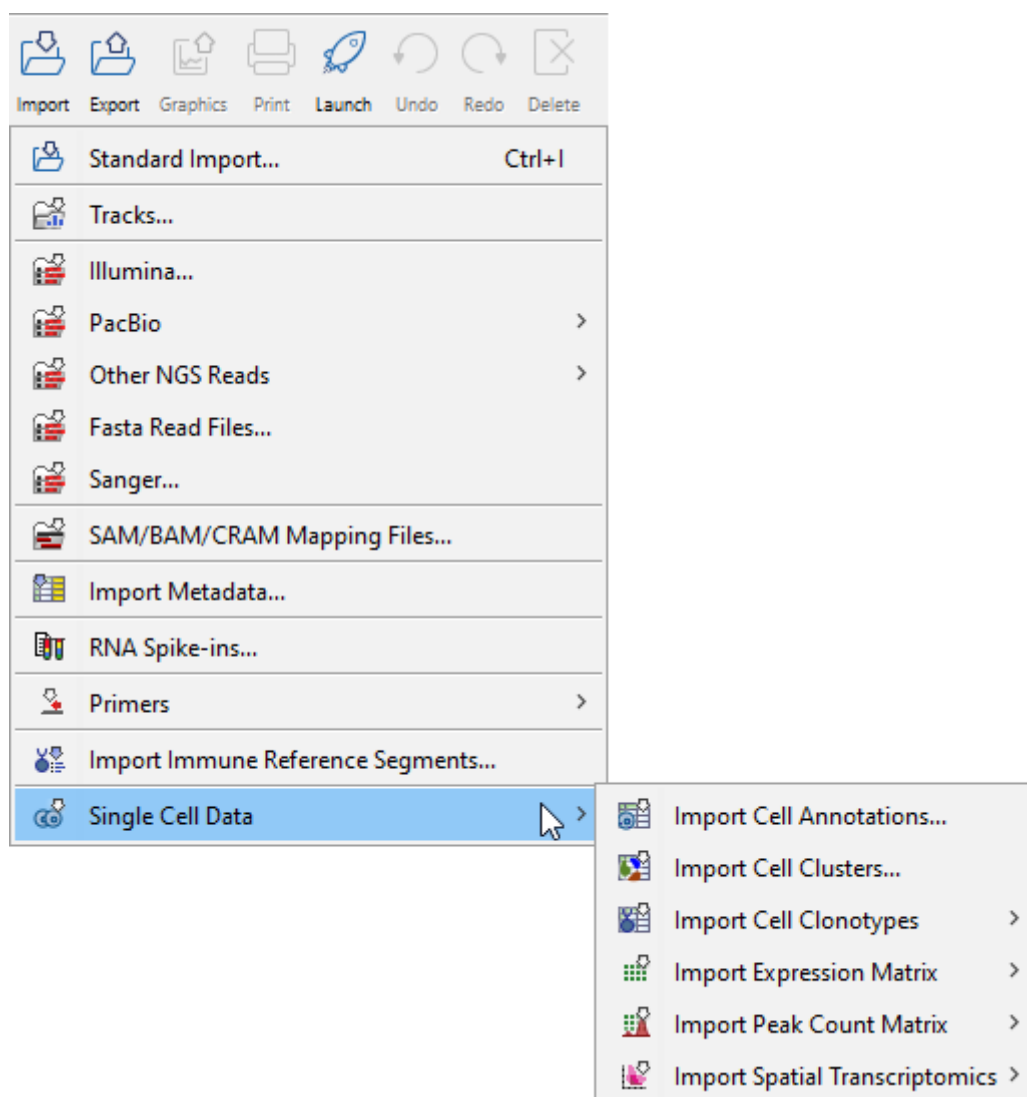


Figure 4.1: Importers available for formats specific to single cell data are located under Import | Single Cell Data.

4.1 Import Immune Reference Segments

Import Immune Reference Segments can import reference sequences for V, D, J and C gene-segments from a fasta file. The sequences are needed when running Single Cell V(D)J-Seq Analysis (section 13.1) for either T or B cell receptor repertoires (TCR and BCR, respectively).

The importer can be found here:

Import (📁) | **Import Immune Reference Segments** (🧬)

The importer can be used to import fasta files that are either in the IMSEQ [Kuchenbecker et al., 2015] or IMGT [Lefranc et al., 2009] format (see figure 4.2).

Both formats support allele numbering for the gene segments. If **Import only the first allele** is ticked, only segments without an allele or those with an allele defined as the number "1" (i.e "01" is also valid) will be imported. Otherwise, all segments are imported.

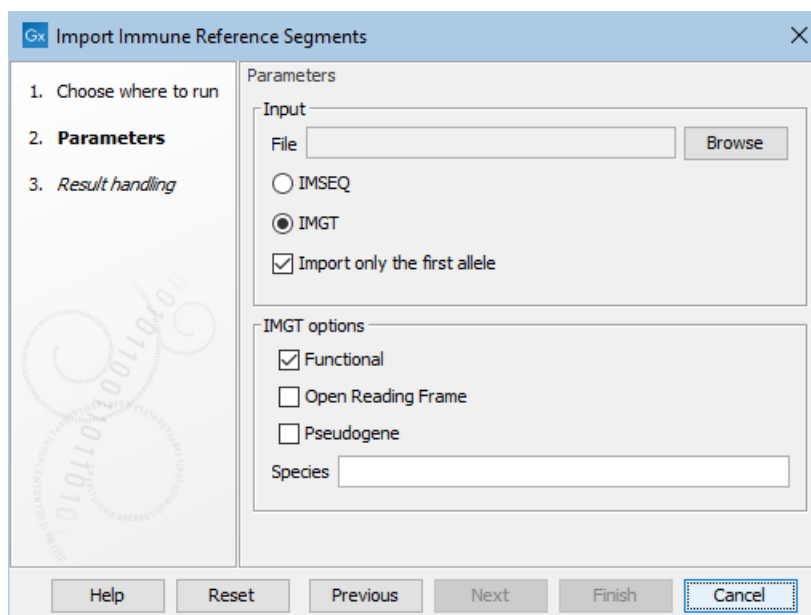


Figure 4.2: The available options when importing immune reference segments.

The two formats differ in how the sequence header is parsed for identifying the gene segment and related information, and how the conserved amino acids in the V and J segments are identified.

When saving the results, the reference data for either TCR, or BCR, or both, can be saved. The wizard will show an error message if an output option is ticked for which no relevant reference sequences are available.

The importer can only handle one fasta file at a time, but if two or more fasta files are imported, the resulting sequence lists can subsequently be combined to one list using the **Create Sequence List** tool.

4.1.1 IMSEQ

For the IMSEQ format, the header contains the following elements, separated by "|":

- The chain: TRA, TRB, TRG or TRD for T cells, and IGH, IGK and IGL for B cells.
- The segment type: V, D, J, C.
- The segment ID. For B cells constant genes, the segment ID should also contain the letter corresponding to the encoded isotype.
- The segment allele.
- For J and V segments, the position of the first base of the conserved amino acid, counting from 0.

Currently only the heavy (IGH) and light κ and λ (IGK and IGL) chain types are supported for B cells.

Any segments with an unsupported chain or segment type are silently ignored.

4.1.2 IMGT

For the IMGT format, the header contains 15 elements, separated by "|". Only the following are read and used during import:

- (1) Accession number(s).
- (2) The segment name, including chain, segment type, ID and allele, in the format: `<chain><type><ID>*<allele>`, for example "TRAV1*01".
Chain and segment type are the same as for IMSEQ. For B cells constant genes, the segment type contains instead the letter corresponding to the encoded isotype.
- (3) Species.
- (4) Allele functionality: F (functional), P (pseudogene) or ORF (open reading frame).
- (5) Extracted label(s): EX1, CH1 and C-REGION for C segments, and V-REGION, D-REGION, J-REGION for V, D, J segments, respectively.
- (8) The start of the codon, counting from 1, or "NR" for non coding labels.
- (9) The number of nucleotides added in 5' in the format `+n`.

The IMGT database contains chains, segment types and labels that are not listed above and are not supported. These are silently ignored.

While the IMSEQ format provides the position of the conserved amino acid, this needs to be calculated for the IMGT format. For this, the V region needs to be provided with gaps such that the conserved amino acid is found at approximately position 104 in the translated amino acid sequence. When downloading sequences from the IMGT database in fasta format, the "F+ORF+in-frame P nucleotide sequences with IMGT gaps" should be used. Alternatively, the corresponding "nt-WithGaps-F+ORF+inframeP" flat file can be downloaded from IMGT/GENE-DB.

If using custom reference data that is not downloaded from the IMGT database, it is recommended to use the IMSEQ format and specify the position of the conserved amino acid.

When importing files in the IMGT format, the following options are available (see figure 4.2):

- Which allele functionality(ies) should be imported. At least one must be chosen.
- Which species should be imported. After choosing the fasta file, the desired species can be chosen from the list of species identified in the file.

If element (9) in the header is not empty, the corresponding number of nucleotides are removed from the 5' end of the sequence.

Identification of the conserved amino acid

The nucleotide sequence (with IMGT gaps for the V segments), starting from position in element (8) in the header, is first translated to amino acids using the standard genetic code. The position of the conserved amino acid is calculated, and, if identified, translated to the position of the first nucleotide in the corresponding codon. Segments where the amino acid cannot be identified are silently ignored.

For the V segments, the amino acid position is calculated as follows:

- If the amino acid at position 104 is C, then position 104 is used.
- Otherwise, the position of the last occurrence of C after position 104 is used, if present.
- Otherwise, if the amino acid at position 104, 105 or 103 (in this order) is one base pair mutation away from C and not a stop codon (i.e. R, S, C, F, G, W, Y), then this position is used.

For the J segments, all 3 open reading frames (starting from nucleotide position 1, 2 or 3) are used. Note that "." below denotes any amino acid. The amino acid position is calculated as follows:

- The amino acid sequence "(F|W)G.G", if present, is identified.
 - The open reading frame that contains the amino acid sequence, no stop codon and has the lowest nucleotide starting position, if any, is used.
 - Otherwise, the open reading frame that contains the amino acid sequence and at least one stop codon, if any, is used. If multiple open reading frames match this criteria, none are used.
- Otherwise, the amino acid sequences "(F|W)X.G" and "(F|W)G.X", if present, are identified. Here, X denotes the amino acids that are one base pair mutation away from F/W and not a stop codon (i.e. A, R, S, C, D, E, V, W).
 - For each of the two amino acid sequences, the position is calculated as above.
 - If both amino acid sequences are present, the position that is closest to the end of the sequence is used.

V and J segments for which the amino acid position cannot be successfully identified are silently ignored.

4.1.3 Output from Import Immune Reference Segments

The importer outputs a sequence list that can be used for immune repertoire analysis. These can be added to a custom reference data set, to be used in workflows. See https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html for details.

The sequence list contains the reference sequences for the V, D, J and C segments, named in the format <chain>-<type>-<ID>*<allele>, for example "TRA-V-1*01". Note that for B

cells constant genes, the letter corresponding to the encoded isotype will be used instead of the segment type.

If the gene segment does not have an allele or **Import only the first allele** is ticked, `*<allele>` is not added to the name.

By ticking "Show annotations" and "Region" in the Side Panel "Annotation layout" and "Annotation types" groups, respectively, the location of the conserved amino acid can be visualized (see figure 4.3).

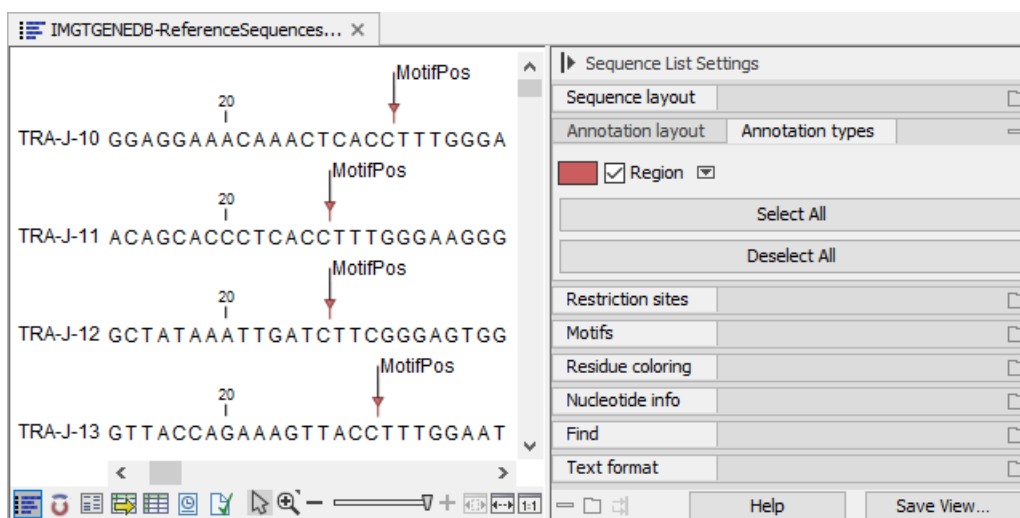



Figure 4.3: Visualizing the location of the conserved amino acid.

The table view of the sequence list shows the chain and segment type of each sequence, and for the IMGT format, also the accession number(s) and species (see figure 4.4).

Name	Size	Accession	Latin name	Chain	Segment type
TRA-C	272	X02883	Homo sapiens	TRA	C
TRA-J-1	62	X02884	Homo sapiens	TRA	J
TRA-J-10	64	M94081	Homo sapiens	TRA	J
TRA-J-11	60	M94081	Homo sapiens	TRA	J
TRA-J-12	60	X02885	Homo sapiens	TRA	J
TRA-J-13	63	M94081	Homo sapiens	TRA	J
TRA-J-14	52	M94081	Homo sapiens	TRA	J
TRA-J-15	60	X05775	Homo sapiens	TRA	J
TRA-J-16	60	M94081	Homo sapiens	TRA	J
TRA-J-17	63	X05773	Homo sapiens	TRA	J
TRA-J-18	66	M94081	Homo sapiens	TRA	J
TRA-J-19	60	M94081	Homo sapiens	TRA	J
TRA-J-2	66	X02884	Homo sapiens	TRA	J
TRA-J-20	57	M94081	Homo sapiens	TRA	J
TRA-J-21	55	M94081	Homo sapiens	TRA	J
TRA-J-22	63	X02886	Homo sapiens	TRA	J
TRA-J-23	63	M94081	Homo sapiens	TRA	J
TRA-J-24	63	X02887	Homo sapiens	TRA	J

Figure 4.4: Table view of imported sequence list showing the name, species and accession number when imported using the IMGT format.

4.2 Import Cell Annotations

Import Cell Annotations can import annotations for each cell. The importer produces **Cell Annotations**  that can be used to define groups of cells for use in many tools such as Differential Expression for Single Cell. Cell Annotations can also be visualized in a Dimensionality Reduction Plot.

Often the same file can be imported as either Cell Clusters or Cell Annotations. The principal advantage of Cell Annotations is that they can represent numerical data. An example might be the probability of a cell being in a particular cell cycle phase.

The importer can be found here:

Import  | **Single Cell Data**  | **Import Cell Annotations** 

The following options are available:


- **Data file.** A single file in .csv, .tsv or .xlsx format. The first row in the file is a header. Each subsequent row describes a cell. Empty lines are ignored.
- **First column defines sample.** When this is enabled, the sample name and barcode are read from the first and second columns in the file, respectively. Otherwise, the first column is used to extract the barcode and optionally the sample, as defined in **Cell format**. Subsequent columns represent categories containing information about the cells.
- **Cell format** and **Sample.** How cells are identified, see section 4.8 for more details.
- **Matrix** (Optional). When a matrix is supplied, the sample name is taken from the matrix. If the previous options also provide sample names:

- If different from the matrix, the importer fails with an error. This can be useful when checking that the file being imported matches the supplied matrix.
- Rows in the file describing cells that are not in the matrix are skipped.

Note that the sample name can only be set using only one of the **First column defines sample**, **Cell format**, or **Sample** options. The **Matrix** option is mandatory if none of the previous options is used for defining the sample and then the matrix must be for one sample only.

The importer can be used in workflows. See section 4.9 for details.

4.3 Import Cell Clusters

Import Cell Clusters can import clusters for each cell. The importer produces **Cell Clusters**  that can be used to define groups of cells for use in many tools such as Differential Expression for Single Cell. Cell Clusters can also be visualized in a Dimensionality Reduction Plot.

Often the same file can be imported as either Cell Clusters or Cell Annotations. The principal advantage of Cell Clusters is that they can be edited within the Dimensionality Reduction Plot.

The importer can be found here:

Import  | **Single Cell Data**  | **Import Cell Clusters** 

The following options are available:

- **Data file.** A single file in .csv, .tsv or .xlsx format. The first row in the file is a header. Each subsequent row describes a cell. Empty lines are ignored.
- **First column defines sample.** When this is enabled, the sample name and barcode are read from the first and second columns in the file, respectively. Otherwise, the first column is used to extract the barcode and optionally the sample, as defined in **Cell format**. Subsequent columns represent categories containing information about the cells.
- **Cell format** and **Sample.** How cells are identified, see section 4.8 for more details.
- **Matrix** (Optional). When a matrix is supplied, the sample name is taken from the matrix. If the previous options also provide sample names:
 - If different from the matrix, the importer fails with an error. This can be useful when checking that the file being imported matches the supplied matrix.
 - Rows in the file describing cells that are not in the matrix are skipped.

Note that the sample name can only be set using only one of the **First column defines sample**, **Cell format**, or **Sample** options. The **Matrix** option is mandatory if none of the previous options is used for defining the sample and then the matrix must be for one sample only.

- **Map clusters to QIAGEN Cell Ontology.** When this is checked, clusters will be translated, if possible, to the QIAGEN Cell Ontology (see section 8.1). The translation attempts to match each cluster with a QIAGEN cell type based on the name and known synonyms. For example, ‘alveolar epithelial cells’ are also called ‘pneumocytes’. If this option is

selected, the ‘alveolar epithelial cells’ cluster, if present, will be named ‘pneumocytes’. This option can be useful when standardizing clusters from different sources. It is especially recommended if clusters will be used to extend a QIAGEN Cell Type Classifier using the Train Cell Type Classifier tool (section 8.3). See figures 4.5 and 4.6 for an example.

Sample	Barcode	Leiden (resolution= 0.1)	Leiden (resolution= 0.5)	Manual annotation
demo	AAACCTGAGCGCTCCA-1	2	3	T cell
demo	AAACCTGGTGATAAAC-1	1	2	B cell
demo	AAACGGGGTTTGTGTG-1	3	4	alveolar epithelial cell
demo	AAAGATGAGTACTTGC-1	3	4	alveolar epithelial cell
demo	AAAGCAAGTCTCTTAT-1	3	1	macrophage
demo	AAAGCAATCCACGAAT-1	1	2	B cell
demo	AAAGTAGGTAGCAAAT-1	1	2	B cell
demo	AAATGCCGTCTAGAGG-1	2	3	T cell
demo	AACACGTCACCTCGGA-1	2	3	T cell



Figure 4.5: An example of a file that could be imported with Import Cell Clusters. The file contains three different clusterings.

Sample	Barcode	Leiden (resolution=0.1)	Leiden (resolution=0.5)	Manual annotation
demo	AAACCTGAGCGCTCCA-1	2	3	T lymphocytes
demo	AAACCTGGTGATAAAC-1	1	2	B lymphocytes
demo	AAACGGGGTTTGTGTG-1	3	4	pneumocytes
demo	AAAGATGAGTACTTGC-1	3	4	pneumocytes
demo	AAAGCAAGTCTCTTAT-1	3	1	macrophages
demo	AAAGCAATCCACGAAT-1	1	2	B lymphocytes
demo	AAAGTAGGTAGCAAAT-1	1	2	B lymphocytes
demo	AAATGCCGTCTAGAGG-1	2	3	T lymphocytes
demo	AACACGTCACCTCGGA-1	2	3	T lymphocytes

Figure 4.6: The result of importing the file shown in figure 4.5 using the option ‘Map clusters to QIAGEN Cell Ontology’. Note that all cell types have been translated to terms in the QIAGEN Cell Ontology. For example, ‘T cells’ have been standardized to ‘T lymphocytes’, and ‘alveolar epithelial cells’ have been standardized to ‘pneumocytes’.

The importer can be used in workflows. See section 4.9 for details.

4.4 Import Cell Clonotypes

Several formats produced by Cell Ranger can be imported into a **TCR Cell Clonotypes**  or **BCR Cell Clonotypes**  element using the following importers:

- AIRR: **Import Cell Clonotypes in AIRR Format** .
- CSV contig: **Import Cell Clonotypes in Cell Ranger Contig Format** .

The importers can be found here:

Import  | **Single Cell Data**  | **Import Cell Clonotypes** 

Most options are common to both importers. The following options can be adjusted:

- **AIRR rearrangements / Contig annotations.** The input .tsv or .csv file, respectively, following the official Cell Ranger format.

- **Reference segments** (Optional). A reference data element downloaded from the Reference Data Manager (see chapter 2) containing the corresponding QIAGEN V, D, J and C genes. When supplied, the gene names in the input file will be mapped to those used in the provided element. This is important when comparing imported Cell Clonotypes with those produced by the CLC Single Cell Analysis Module, see section 13.4.
- **Cell format** and **Sample**. How cells are identified, see section 4.8 for more details.
- **Matrix** (Optional). The sample name will be obtained from the supplied matrix. The importer does not check that the barcodes present in the input file match those in the matrix. If the matrix contains multiple samples, the importer will fail with a relevant message.

If sample name is not defined through either **Cell format**, **Sample**, or **Matrix**, the importer sets the sample name to the name of the input file. The sample name can be set using only one of these three options.








When matched scRNA-Seq data is available, it is important that the Cell Clonotypes and corresponding Expression Matrix/Dimensionality Reduction Plot have the same sample name, see section 13.5.

Note that the productive status is calculated from the CDR3 amino acid sequence found in the input file.

The importer can be used in workflows. See section 4.9 for details.

4.5 Import Expression Matrix

Several formats can be imported into an **Expression Matrix** () using the following importers:

- AnnData: **Import Expression Matrix in AnnData format** ();
- Cell Ranger HDF5: **Import Expression Matrix in Cell Ranger HDF5 format** ();
- h5Seurat: **Import Expression Matrix in h5Seurat format** ();
- Loom: **Import Expression Matrix in Loom format** ();
- MEX: **Import Expression Matrix in MEX format** ();
- Parse Biosciences MTX: **Import Expression Matrix in Parse Bio MTX format** ();
- Plain Text Table: **Import Expression Matrix in Plain Text Table format** ().

The importers can be found here:

Import () | **Single Cell Data** () | **Import Expression Matrix** ()

The importers can be used in workflows. See section 4.9 for details.

Some other commonly encountered formats are specific to a programming language or software package. These can usually be exported from that software package as Loom files. For example, .rds/.Robj formats are from the R programming language and can often be written to Loom using the LoomR package, or methods in the same R package that was used to generate the files.

General options

The following options are common to all expression matrix importers:

- **Gene or Transcript track.** Genes or transcripts in the imported data are matched with features in the provided track to the extent possible. When a match is found, the genomic coordinates of the gene/transcript will be recovered. Matches are only found when the identification of the gene/transcript in the imported data with the feature in the track is unambiguous: one-to-many and many-to-one matches between the imported data and the provided track are not supported. This means, for example, that if a gene is present on two chromosomes of the track, then neither set of genomic coordinates will be recovered.

Matching is used to:

- View the Expression Matrix as a Track. For more information on tracks, see <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tracks.html>.
- Define the mitochondrial chromosome when calculating the proportion of reads mapped to mitochondria in the QC for Single Cell tool (section 7.2.2).
- Recover identifiers (e.g. ENSG00000243485 for ENSEMBL genes) when these are not present in the input data. As identifiers are often more specific than e.g. gene names, this can help when training Cell Type Classifiers using the Train Cell Type Classifier tool (section 8.3), and when predicting cell types using a Cell Type Classifier.

Although it is generally beneficial to provide a Gene or Transcript track that matches many of the genes or transcripts in the imported data, it is still possible to analyze data in the CLC Single Cell Analysis Module even when no matches are found.

The matching algorithm works by choosing an approach from the following list that maximizes the number of one-to-one matches between features in the provided track and features in the imported data:

- Matching names from the track with identifiers from the imported data
- Matching identifiers from the track with identifiers from the imported data
- Matching names from the track with unversioned identifiers from the imported data. An unversioned identifier is obtained by removing anything from or after the first ‘.’ in the identifier. For example, ENSG00000243485 is the unversioned identifier for ENSG00000243485.5.
- Matching identifiers from the track with unversioned identifiers from the imported data
- Matching names from the track with names from the imported data
- Matching identifiers from the track with names from the imported data

In the case of a tie, the first equally good approach from the above list is used. If no matches are found, check that the correct Gene or Transcript track has been supplied.

- **Spike-in controls** (Optional). Genes or transcripts in the imported data are also matched against the spike-in controls provided here. This is used when calculating the proportion of reads mapped to spike-in controls in the QC for Single Cell tool (section 7.2.2). It is also used to remove the spike-in controls from downstream analysis. For details on how to import spike-in controls, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Import_RNA_spike_in_controls.html.



- **Cell format** and **Sample**. How cells are identified. See section 4.8 for more details. When a file contains multiple samples, it is recommended to extract the sample name from the cell name. This allows the QC for Single Cell to process each sample separately, enables coloring of cells by sample in the Dimensionality Reduction Plot, and may simplify configuration of batch correction.

Options for importing cell annotations and clusters

AnnData, h5Seurat, Loom, and Parse Bio MTX can contain metadata about cells, and this can be imported as **Cell Annotations**  or **Cell Clusters** . These importers share the following options:

- **Create clusters for**. A comma-separated list of attributes to be imported as Cell Clusters. Any other cells metadata will be imported as Cell Annotations.
- **Map clusters to QIAGEN Cell Ontology**. When this is checked, clusters will be translated, if possible, to the QIAGEN Cell Ontology (see section 8.1). The translation attempts to match each cluster with a QIAGEN cell type based on the name and known synonyms. For example, ‘alveolar epithelial cells’ are also called ‘pneumocytes’. If this option is selected, the ‘alveolar epithelial cells’ cluster, if present, will be named ‘pneumocytes’. This option can be useful when standardizing clusters from different sources. It is especially recommended if clusters will be used to extend a QIAGEN Cell Type Classifier using the Train Cell Type Classifier tool (section 8.3).

Options for importing spliced and unspliced counts

Loom and MEX formats can contain both the total expression, spliced, and unspliced counts. The importers can be configured with which type of data to import and produce either an **Expression Matrix** , or an **Expression Matrix with spliced and unspliced counts** .

- **Import expressions**. Enables import of total expression from the relevant file. This is needed when:
 - spliced/unspliced counts are not available;
 - the total expression of a gene cannot be obtained purely from the spliced and, if selected, unspliced counts. For example, the expression has been normalized.
- **Import spliced/unspliced**. Enables import of spliced and unspliced counts from the relevant file(s). If the file(s) do not contain spliced/unspliced counts, the import will fail with a relevant message.
- **Include unspliced counts in total expression**. By default, the total expression of a gene is obtained from the spliced counts. When this option is checked, the unspliced counts are also added to the total expression. This option is recommended for single nucleus RNA sequencing (snRNA-Seq), where data is usually analyzed by counting expression from both exons and introns [Bakken et al., 2018]. This option has no effect when both ‘Import expressions’ and ‘Import spliced/unspliced’ are checked, where the total expression is read directly from the file.

4.5.1 HDF5 formats

AnnData, Cell Ranger HDF5, h5Seurat and Loom are HDF5 formats, with specific requirements regarding structure of the data. An HDF5 file is organized in a hierarchical structure with:

- groups, containing zero or more groups or datasets;
- datasets: multidimensional arrays of data elements.

Metadata for groups and datasets is stored in associated attribute lists. Groups and datasets can often be themselves semantically interpreted as attributes.

The HDF5 file

All HDF5 importers contain an **Expression matrix** option, used for specifying the HDF5 file to be imported.

The AnnData, h5Seurat and Loom importers can be customized to import different attributes from the file. These attributes can be previewed by clicking the ‘Preview’ button.

The preview (figure 4.7) shows the available attributes in a table. One column corresponds to one attribute, for either features or cells, as selected in the menu to the left.

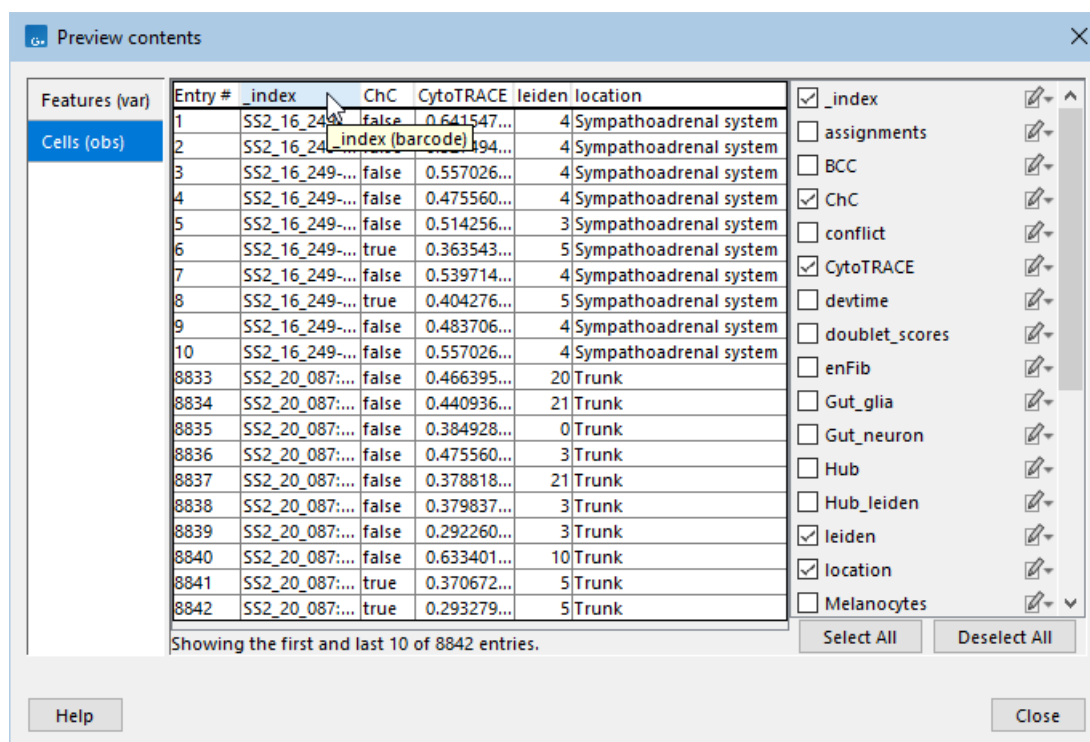


Figure 4.7: Previewing cell attributes found under the ‘obs’ group for the GSE201257 AnnData expression matrix from the Gene Expression Omnibus repository. The ‘_index’ attribute defines the barcode, as shown in the tooltip.

Hovering the cursor over a column name, either at the top of the table or on the menu to the right, displays a tooltip with the type of data stored in the attribute (for example, boolean or integer) and if the attribute is always used by the importer (for example, for the barcode or the sample).

Right-clicking on the column name at the top of the table, or clicking on the edit icon (✎), displays a menu from which the attribute can be added to or removed from relevant wizard options (figure 4.8).

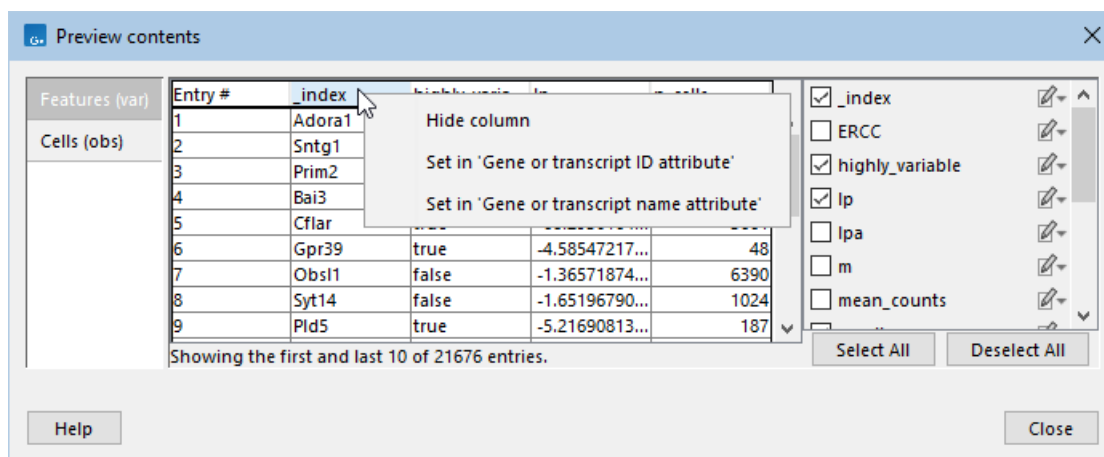


Figure 4.8: Previewing feature attributes found under the ‘var’ group for the GSE201257 AnnData expression matrix from the Gene Expression Omnibus repository. Right-clicking on the ‘_index’ column name displays a menu.

AnnData importer

The expression matrix in an AnnData (h5ad) is in a sparse dataset ‘X’, while features and cells are described using the ‘var’ and ‘obs’ groups, respectively. See <https://anndata.readthedocs.io/> for more details.

The ‘_index’ attribute on group ‘obs’ defines the cell identification, and the interpretation of this is specified by the **Cell format**.

- **Gene or transcript ID attribute** (Optional). A ‘var’ attribute describing an identifier for a gene or transcript (e.g., ENSG00000243485 for ENSEMBL).
- **Gene or transcript name attribute** (Optional). A ‘var’ attribute describing the name for a gene or transcript. If left empty, the ‘_index’ attribute on group ‘var’ is used.

h5Seurat importer

A h5seurat file may contain multiple assays and each assay may contain multiple expression matrices, e.g., counts and normalized expressions. The matrices can be sparse or dense. See <https://mojaveazure.github.io/seurat-disk/articles/h5Seurat-spec.html> for more details.

Only one assay and matrix can be imported at a time. The h5Seurat importer expects the format version 4.0.0.

The ‘cell.names’ attribute contains the cell identification, and the interpretation of this is specified by the **Cell format**. If the sample is not set through **Cell format** or **Sample**, the sample for each cell is read from the ‘orig.ident’ attribute on group ‘meta.data’.

The gene or transcript names are read from the ‘features’ attribute of the selected assay.

- **Assay** (Optional). The name of the assay to import. If left empty, the assay in the 'active.assay' attribute will be used.
- **Import expressions from** (Optional). The matrix for the selected assay to import. The matrix may be sparse (e.g., 'counts' or 'data') or dense (e.g., 'scale.data'). If left empty, the importer will use 'counts'.

Loom importer

A Loom file has an internal structure consisting of a main matrix, optional 'layers' of the same size as the main matrix and row and column attributes (describing features and cells, respectively). See <https://linnarssonlab.org/loompy/format/index.html> for details on the format.

The Loom importer expects the Loom format version 3.0.0.

- **Spliced layer**. The layer where the spliced counts are stored.
- **Unspliced layer**. The layer where the unspliced counts are stored.
- **Cell ID attribute** (Optional). A column attribute identifying the cell by its barcode and sample. The interpretation of this value is specified by the **Cell format**.
- **Gene or transcript ID attribute** (Optional). A row attribute describing an identifier for a gene or transcript (e.g., ENSG00000243485 for ENSEMBL).
- **Gene or transcript name attribute**. A row attribute describing the name for a gene or transcript. If no names are present, then it is also possible to set this to the same value as the Gene or transcript ID attribute.

4.5.2 Other formats

MEX importer

This importer requires the following files to be supplied:

- **Barcodes file**. A file with the extension .tsv, conventionally named barcodes.tsv. It contains tab-separated columns, and has one row per barcode. It can optionally contain a header. The barcodes are read from the first column. Empty lines are ignored.
Use the **Cell format** option to control how the barcodes should be interpreted - for example if it also includes information about the sample.
- **Feature file**. A file with the extension .tsv, conventionally named features.tsv or genes.tsv. It contains one row per feature. It can optionally contain a header. Empty lines are ignored. It contains multiple tab-separated columns:
 - One column: the feature name.
 - Two columns: the feature identifier and name.
 - Three columns: the feature identifier, name, and type. Of the commonly used feature types, "Gene Expression", "Transcript Expression", and "Spike-in" are the most important ones. Other features, such as "Antibody Capture" will be silently ignored by most tools.

For 10x Multiome files there will be six columns. The last three consist of genome coordinates and will be ignored. Lines with feature type "Peaks" will also be ignored. They should instead be imported as a **Peak Count Matrix** (see section 4.6).

- **Matrix file(s).** File(s) with the extension .mtx in the Matrix Market Exchange Coordinate Format, see <https://math.nist.gov/MatrixMarket/formats.html> for details. Features must be in the first dimension (rows) and cells in the second (columns).

Expressions and/or spliced and unspliced counts can be imported using:

- **Matrix file** for expressions, conventionally named matrix.mtx.
- **Matrix file (spliced)** for spliced counts, conventionally named spliced.mtx.
- **Matrix file (unspliced)** for unspliced counts, conventionally named unspliced.mtx.

See 'Options for importing spliced and unspliced counts' from section 4.5 for more details.

Additional options are:

- **Name.** The name of the imported matrix. If **Cell format** is not configured to parse a sample name from each barcode in the barcodes file, then this will also be the sample name for all the imported barcodes.
- **Files are in same directory.** This option is provided for convenience and works for local files. When checked, if any file option is updated to a file in a new directory, the other files are automatically updated, if files with the conventional names can be found in the directory.

Parse Bio MTX importer

This importer requires three files to be supplied:

- **Cell metadata file.** A file with the extension .csv and comma separated columns, with one row per barcode. It must contain headers. The following options relating to the cell metadata file are available:
 - **Barcode column.** The name of the column containing the barcodes.
 - **Cell metadata has sample name.** If checked, the sample name is read from the file. Otherwise, the sample name is defined by the general options (see section 4.8).
 - **Sample column** (Optional). The name of the column containing the sample names.
- **Feature file.** A file with the extension .csv and comma separated columns, with one row per feature. The following options relating to the feature file are available:
 - **Feature id column** (Optional). The name of the column containing the feature identifiers (e.g., ENSG00000243485 for ENSEMBL).
 - **Feature name column.** The name of the column containing the feature names.
- **Matrix file.** A file containing the expression with the extension .mtx in the Matrix Market Exchange Coordinate Format. Cells must be in the first dimension (rows) and features in the second (columns). See <https://math.nist.gov/MatrixMarket/formats.html> for details of the Matrix Market Exchange Coordinate Format.

Batches and samples: QC for Single Cell, see section 7.2, runs separately for each sample detected in the input Expression Matrix. This might not be appropriate for Parse Biosciences data, where samples are sequenced together in one batch. If the matrix to be imported is not filtered, we recommend to:

- Import the data by running the importer with **Cell metadata has sample name** unchecked and **Create cell annotations** checked.
- Filter the matrix by running QC for Single Cell on the imported matrix.
- Update the sample name of the filtered matrix to that in the imported annotations using Update Single Cell Sample Name, see section 18.7.

Plain Text Table importer



This importer supports import of text data in a full plain text table format.

- **Expression matrix.** A single file to be imported.
- **Table layout.** Choose whether the table has cells in columns and features in rows, or is transposed such that features are in columns and cells are in rows.

Working with spreadsheets Be careful to check that all the data is present before import if the file originates from a spreadsheet program. Such programs often impose limits on the number of rows and columns.

4.6 Import Peak Count Matrix

Several formats can be imported into a **Peak Count Matrix** () using the following importers:

- Cell Ranger HDF5: **Import Peak Count Matrix in Cell Ranger HDF5 format** ();
- MEX: **Import Peak Count Matrix in MEX format** ();

The importers can be found here:

Import () | **Single Cell Data** () | **Import Peak Count Matrix** ()

The importers can be used in workflows. See section 4.9 for details.

General options

The following options are common to all peak matrix importers (figures 4.9 and 4.10):

- **Gene track** Positions in the imported data are matched with the provided track. Matching is used to:

The dialog box is titled "Import Peak Count Matrix in Cell Ranger HDF5 Format". It has a sidebar on the left with four steps: 1. Choose where to run, 2. **Cell Ranger HDF5 import**, 3. Peak import, and 4. Result handling. The main area is titled "Cell Ranger HDF5 import" and contains three sections: "General options" with fields for "Gene track", "Cell format" (set to "{barcode}" with a hint "Press Shift + F1 for options"), and "Sample"; "Cell Ranger HDF5 options" with a "Peak count matrix" field and a "Browse" button; and "Preview cells" with the text "No file selected" and a "Disable" button. At the bottom are buttons for "Help", "Reset", "Previous", "Next", "Finish", and "Cancel".

Figure 4.9: The Cell Ranger Peak Count importer. The General options are common to all the peak matrix importers.

- View the Peak Count Matrix as a Track. For more information on tracks, see <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tracks.html>.
- Identify nearby genes if these are not explicitly supplied.
- **Cell format** and **Sample**: How cells are identified. See section 4.8 for more details.

The dialog box is titled "Import Peak Count Matrix in Cell Ranger HDF5 Format". The sidebar on the left shows step 3, **Peak import**, selected. The main area is titled "Peak import" and contains three sections: "Nearby genes" with an "mRNA track" field, an unchecked "Import nearby genes" checkbox, and a "Peak annotations" field with a "Browse" button; "Transcription factors" with an unchecked "Import transcription factors" checkbox and a "Peak-motif associations" field with a "Browse" button; and "Filtering" with a "Minimum peak count" field set to "1,000". At the bottom are buttons for "Help", "Reset", "Previous", "Next", "Finish", and "Cancel".

Figure 4.10: Additional options common to all the peak matrix importers.

- **Nearby genes** Nearby genes are determined in one of two ways:
 - By searching for nearby genes using the **Gene track** and an accompanying mRNA track

- By supplying nearby genes in a selected tab-separated file (.tsv).

The file must consist of either:

- 6 columns: name of the chromosome prefixed with "chr" (e.g., "chr1"), start and end position of the peak, the name of the gene, distance and type of peak.
- 4 columns: name of the chromosome together with start and end positions of the peak (e.g., "chr1:123-456"), the name of the gene, distance and type of peak.

The first line must be column headers.

The distance is the number of base positions from the start or end of the peak to the start or end of the gene, whichever is closest. It is signed and will be negative if the peak is before the gene.

The type of the peak can be either "promoter" or "distal". Other values are ignored.

If there are multiple nearby genes per peak, they can either be on separate lines or be grouped on one line, with gene name, distance and peak types separated by semi-colon.

- **Transcription factors** If enabled, transcription factors will be imported from the selected tab-separated bed file. Each line consists of the name of the chromosome (e.g., "chr1"), start and end positions of the peak, and the name of the transcription factor.

If not enabled, the peak matrix will not have transcription factors.

The data to be imported may either consist of peak data only or it may be a mixture of peaks and gene expressions, as is the case for 10x Multiome files. In the latter case, the gene expressions must be imported into a separate **Expression Matrix** (see section 4.5).

HDF5 importer

The Cell Ranger HDF5 importer requires one file to be supplied:

- **Peak count matrix** The peak count matrix in HDF5 format.

MEX importer

The MEX importer requires three files to be supplied:

- **Barcodes file** A file with the extension .tsv and tab-separated columns, with one row per barcode. It can optionally contain a header. The barcodes are read from the first column. Empty lines are ignored.


Use the **Cell format** option to control how the barcodes should be interpreted - for example if it also includes information about the sample.

- **Feature or peak file** This should be one of:
 - A feature file with extension .tsv and six tab-separated columns, with one row per feature or peak. These are relevant for mixtures of peaks and expressions, e.g. 10x Multiome. The columns are: identifier (e.g., "chr1:123-456"); name (same as identifier for peaks); type, e.g. "Gene Expression" or "Peaks"; chromosome (e.g., "chr1"); start and end position of the feature or peak. The file can optionally contain a header. Empty lines are ignored.

- A peak file with extension `.bed` and three tab-separated columns: the chromosome, start and end position.
- **Matrix file** A file containing the expression with the extension `.mtx` in the Matrix Market Exchange Coordinate Format.

See <https://math.nist.gov/MatrixMarket/formats.html> for details of the Matrix Market Exchange Coordinate Format.

4.7 Import Space Ranger

Import Space Ranger can import spatial transcriptomics data from Space Ranger spatial outputs containing processed tissue images and barcode locations in those images. The importer produces a **Spatial Transcriptomics Plot** () (see section 11.1).

The importer can be found here:

Import () | **Single Cell Data** () | **Import Spatial Transcriptomics** () | **Import Space Ranger** ()

The following options are available:

- **Tissue positions.** A file in one of the following formats:
 - CSV format: extension `.csv` and comma separated columns. The file contains one row per barcode. It may optionally contain headers.
 - Parquet format: extension `.parquet`. The file contains one record per barcode.

Information is extracted from the following columns in the file, using the column names for CSV files with headers and Parquet files, or column numbers for CSV files without headers:

 - `barcode` (column 1): the barcode.
 - `in_tissue` (column 2): zero or one indicating whether the barcode is present in the tissue. Rows with zero are ignored.
 - `pxl_row_in_fullres` (column 5): the y position in the full resolution image.
 - `pxl_col_in_fullres` (column 6): the x position in the full resolution image.
- **Include image.** Include a processed tissue image when checked.
 - **Processed image file.** A processed tissue image in png format.
 - **Scale factors file.** A JSON file containing a map of scale factors for processed tissue images. The scale factor with the map key matching the tissue image file name is used. Both the file name and map keys are split into components using `"_"`, and matching is based on identical components. For example, for a tissue image file `tissue_hires_image.png`, the key is typically `tissue_hires_scalef`.
- **Cell format** and **Sample.** How cells are identified, see section 4.8 for more details.
- **Matrix** (Optional). The sample name will be obtained from the supplied matrix. The importer does not check that the barcodes present in the input file match those in the matrix. If the matrix contains multiple samples, the importer will fail with a relevant message.

The sample name has to be defined through either **Cell format**, **Sample**, or **Matrix**. The sample name can be set using only one of these three options.

It is important that the Spatial Transcriptomics Plot and corresponding Expression Matrix/Dimensionality Reduction Plot have the same sample name, see section 11.1.

The importer can be used in workflows. See section 4.9 for details. Note that if **Use archive file** is checked, then:

- If **Include image** is checked and there are multiple processed image files, the smallest image is used, i.e., the image with the smallest scale factor.
- The sample name can be omitted from **Cell format**, **Sample**, and **Matrix**. In this case, it will be based on the archive file name.

4.8 Cell format in importers

Most importers in the CLC Single Cell Analysis Module import information about cells. Cells are identified by a combination of their barcode, e.g. "AAGCT", and their sample name.

Importers share the following common options:

- **Cell format**. Specify how to extract the barcode and, optionally, the sample name from the name of the cell. A combination of placeholders and text can be used (figures 4.11 and 4.12) to extract the part(s) of the name corresponding to the barcode/sample. Hover the mouse cursor over the field to see a tooltip with several examples. Simultaneously pressing Shift and F1 displays all available placeholders.

By default, the entire name of the cell is used as barcode and the sample name is based on the name of the imported file.

- **Sample** (Optional). This can be used for specifying a custom sample name. It should only be used when the file contains just one sample. It overrides the default sample name.
This is relevant e.g. when jointly analyzing an imported Expression Matrix and Peak Count Matrix, where cells must have the same sample name.

Importers contain a **Preview cells** section displaying how the cell names are parsed into sample and barcode (figure 4.11). The preview helps ensure that the configured **Cell format** matches the input (figure 4.13). If the configured format is invalid, the preview may fail to determine the sample and/or barcode (figure 4.14).

The preview can be disabled if not needed. This is useful for input files that are large, where generating the preview may take some time.

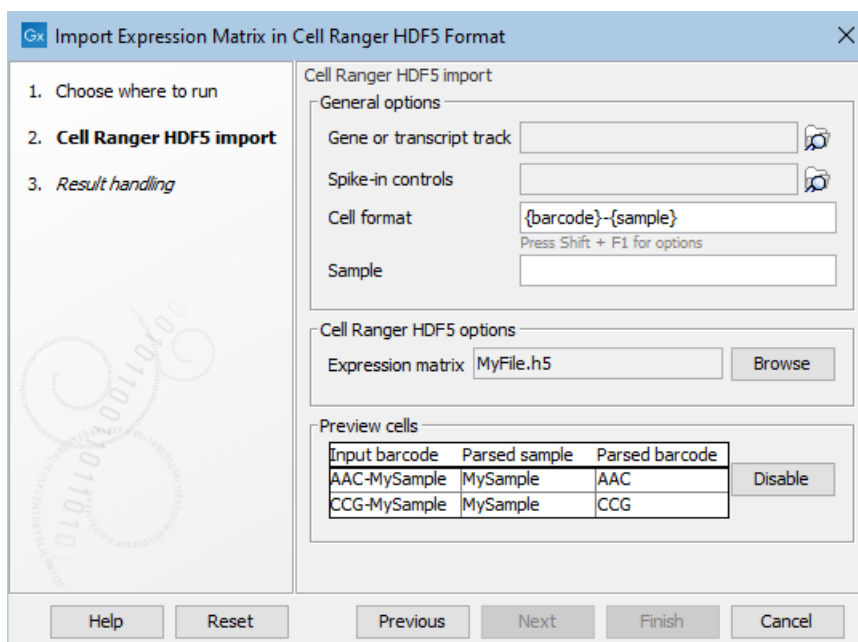


Figure 4.11: Extracting the barcode and sample from the name of the cell. 'Preview cells' shows how the cell names are parsed into sample and barcode.

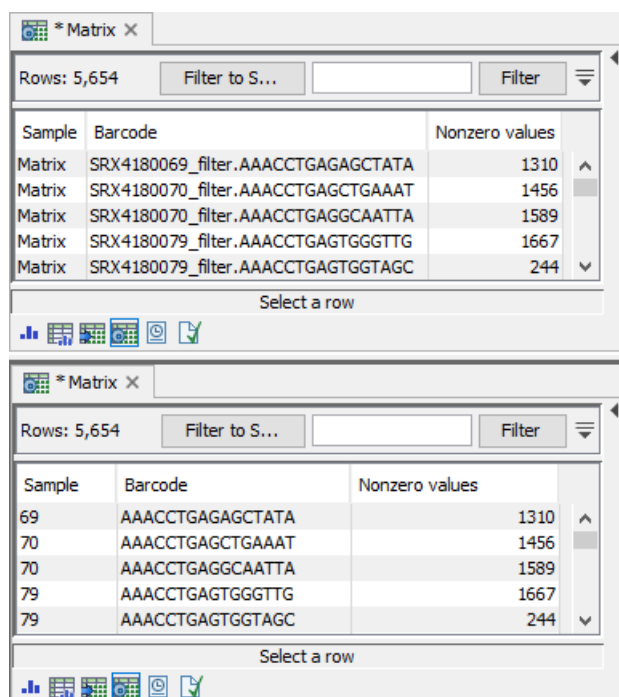


Figure 4.12: The top panel shows the results of importing a matrix file with Cell format = {barcode}. After import, the sample name is the name of the file that was imported, and the barcode is the entire name of the cell. In the bottom panel, Cell format = SRX41800{sample}_filter.{barcode}. Here, the sample name and the barcode are extracted from the name of the cell, and other parts of the name are discarded.

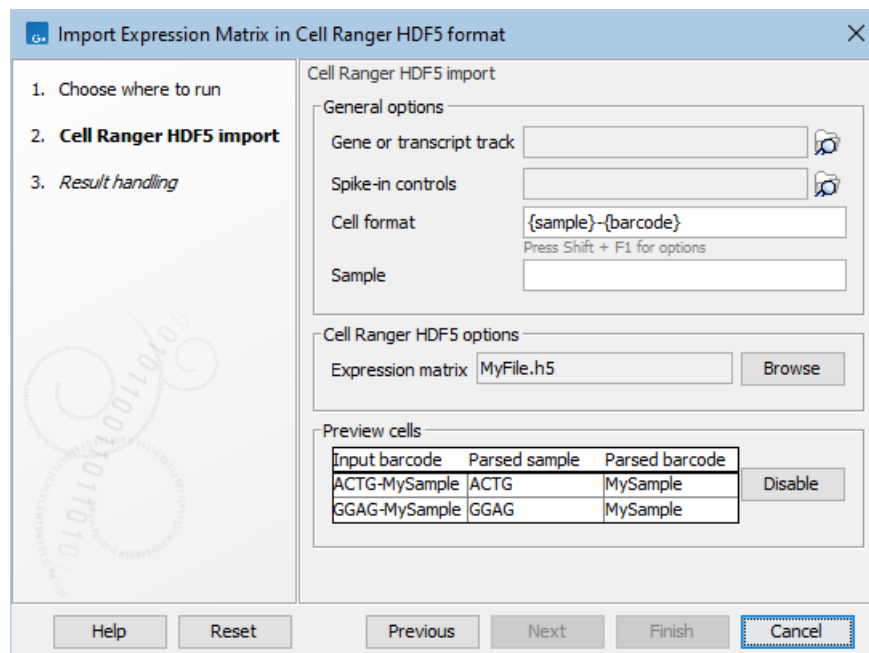


Figure 4.13: The preview helps identify that the sample and barcode have been swapped.

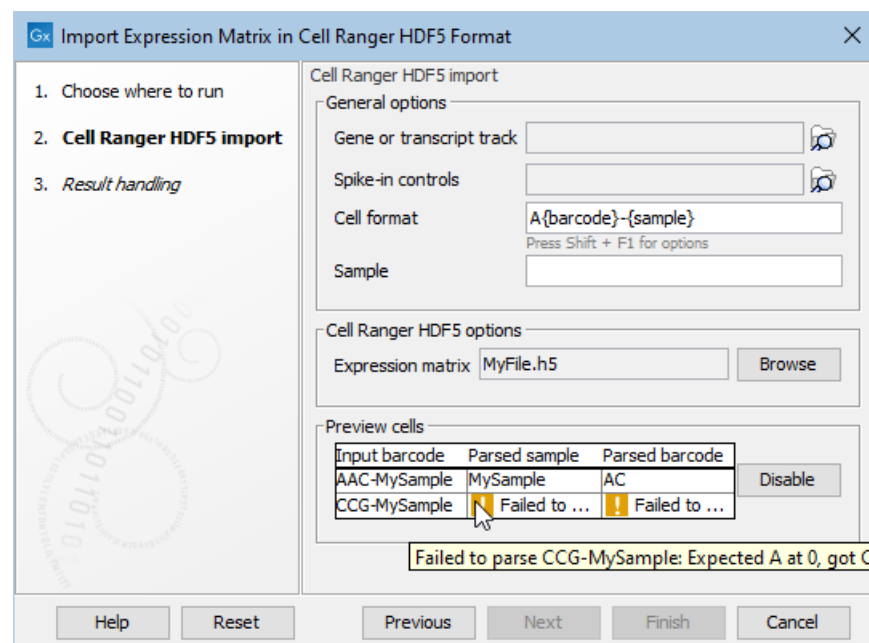


Figure 4.14: The preview helps identify that the configured cell format is not valid. The tooltip contains a detailed error message.

4.9 On-the-fly import in workflows

Most single cell data importers can be used in workflows for importing data on the fly. See https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Launching_workflows_individually_in_batches.html for general information about on-the-fly import. This section provides information specific to the single cell data importers.

Note that when importing data on the fly, the **Preview cells** section, see section 4.8, is not available.

Importing multiple datasets

When importing multiple datasets, the same import options are used for all datasets. For example, if the files to be imported do not share the same **Cell format** and/or **Sample** (see section 4.8), they need to be imported separately before running the workflow.

Most importers use a single file for importing data. To import multiple datasets, select all the files to be imported.

Some importers allow optional additional files, and others require several files. For example, **Import Peak Count Matrix in Cell Ranger HDF5 format** imports the data from an HDF5 file, and can optionally import nearby genes and/or transcription factors from additional files, while **Import Expression Matrix in MEX format** requires at least three files: barcodes, features and matrix files.

Importers that accept more than one file have the following additional options when used in a workflow:

- **Use archive file.** Enables import of an archive file instead of individual files.
- **Archive file.** The archive file to be imported. Hover the mouse cursor over the option to see a tooltip with a description of the files the importer will use from the archive (figure 4.15).

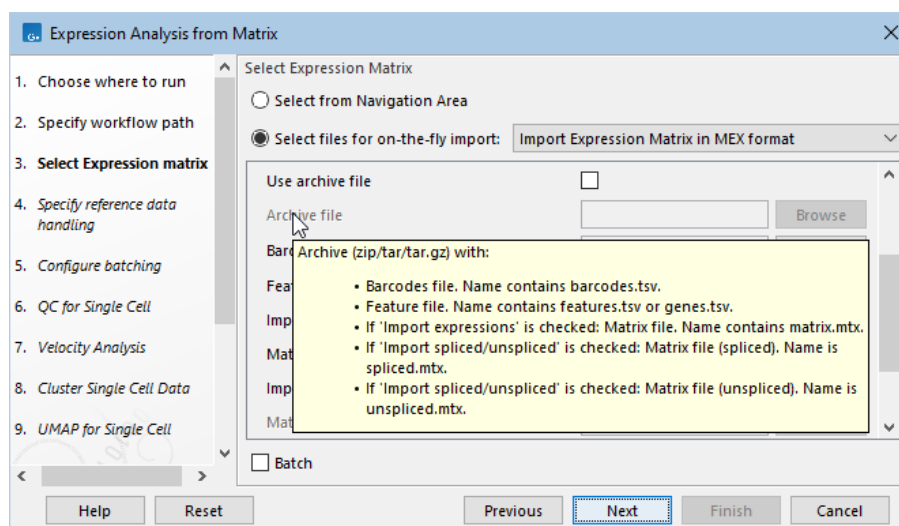


Figure 4.15: Hovering the mouse cursor over the 'Archive file' option reveals a tooltip with a description of the files the importer will use from the archive.

To import multiple datasets using such importers, check the **Use archive file** option and select all the archive files to be imported.

Batching

When importing multiple datasets, batching can be used to analyze the datasets separately.

Note that most template workflows provided by the CLC Single Cell Analysis Module include an **Iterate** control flow element, which automatically handles batching when multiple files are imported.

Batching can sometimes be disallowed (figure 4.16).

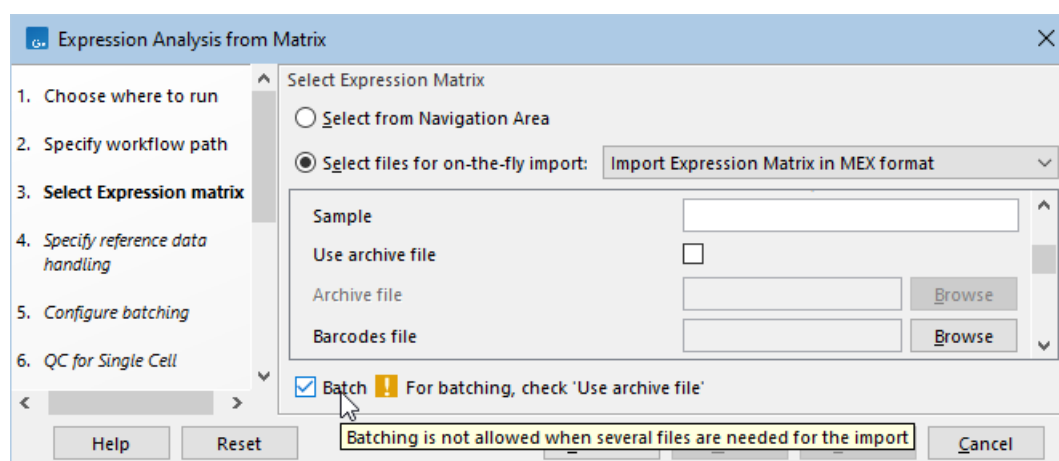


Figure 4.16: Hovering the mouse cursor over the 'Batch' option reveals a tooltip. When batching is disallowed, the tooltip and an info message offer a description of why batching is disallowed and what is required for batching.

For more information on running workflows in batch mode, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html.

Chapter 5

Data export

Contents

5.1 Export Cell Ranger HDF5 Expression Matrix	54
5.2 Export AnnData Expression Matrix	55
5.3 Export h5Seurat Expression Matrix	55
5.4 Export Loom Expression Matrix	56
5.5 Export MEX Expression Matrix	56
5.6 Export Plain Text Table Expression Matrix	57
5.7 Export Peak Count Matrix	57

This chapter describes only exporters functionality specific to the CLC Single Cell Analysis Module.

Exporters exist for exporting **Expression Matrix** () / () to the following formats:

- **Cell Ranger HDF5.** It is the most used format for single cell expression matrices and compatible with most open source tools. It is a highly compressed binary data structure that is easy to navigate compared to text formats like MEX.
- **Loom.** A format that in addition to the Expression Matrix, raw or normalized, offers to export cell cluster and cell annotation information. This is useful for third party visualization tools.
- **AnnData.** An HDF5 format used by e.g. Scanpy.
- **h5Seurat.** An HDF5 format used by Seurat.
- **MEX.** A format for export of raw or normalized Expression Matrices with barcode and genes listed in .tsv files for indexing.
- **Plain Text Table.** A plain text table format with comma, semicolon or tab separators for export of raw Expression Matrices.

Note: While it is possible to export normalized expression values, this is only provided for interoperability with other software. If such values are re-imported, the resulting Expression Matrix may not work as intended in tools that assume values look like counts (such as Normalize Single Cell Data) or are equal to or larger than zero (such as Predict Cell Types). For this reason, when sharing data with other users of CLC Single Cell Analysis Module, it is recommended to export in CLC format. This preserves both the raw expression values and the normalized values.

Additionally, exporters exist for exporting a **Peak Count Matrix** (📊) to **Cell Ranger HDF5** and **MEX**.

Launch the Export tool by clicking on the Export button in the toolbar, or going to the **File** menu and choosing **Export....** To easily locate relevant exporters, type "matrix" into the search field at the top of the tool (figure 5.1).

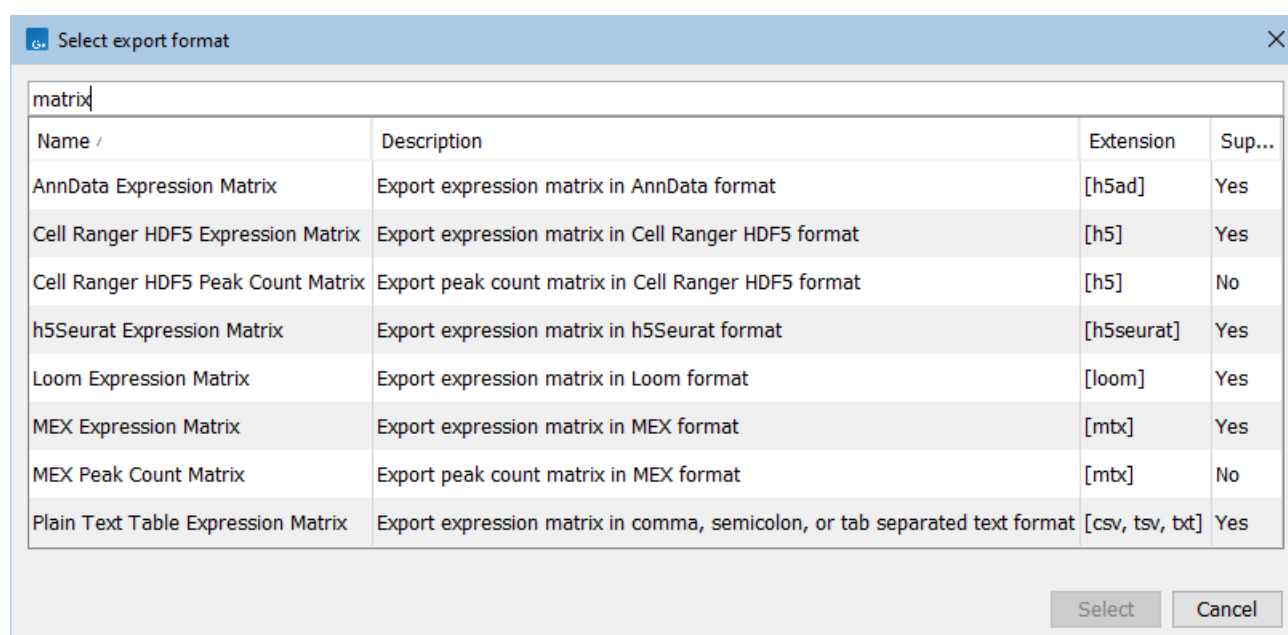


Figure 5.1: Exporters available for exporting single cell expression matrices

5.1 Export Cell Ranger HDF5 Expression Matrix

For export in Cell Ranger HDF5 format, a number of options can be specified:

- **Cell format.** Can be specified to include both the sample and barcode. Hover the mouse over this setting to see the possible options. Often the barcode in itself is sufficient as a unique identifier of cells. However, when an expression matrix contains multiple samples, the cell format must also include the sample, otherwise the export will fail.
- **Use compression.** Select among gzip, zip or no compression which is default.
- **Output file name.** Shows the name of the file. This can be customized by changing the default pattern in **Custom file name**.

5.2 Export AnnData Expression Matrix

For export in AnnData format (h5ad), a number of options can be specified:



- **Cell format.** Can be specified to include both the sample and barcode. Hover the mouse over this setting to see the possible options. Often the barcode in itself is sufficient as a unique identifier of cells. However, when an expression matrix contains multiple samples, the cell format must also include the sample, otherwise the export will fail.
- **Use compression.** Select among gzip, zip or no compression which is default.
- **Output file name.** Shows the name of the file. This can be customized by changing the default pattern in **Custom file name**.

The AnnData exporter produces an hdf5 file with:

- Feature names written at `var/index`.
- Feature ids written at `var/gene_ids`.

5.3 Export h5Seurat Expression Matrix

For export in h5Seurat format (h5seurat), a number of options can be specified:



- **Cell format.** Can be specified to include both the sample and barcode. Hover the mouse over this setting to see the possible options. Often the barcode in itself is sufficient as a unique identifier of cells. However, when an expression matrix contains multiple samples, the cell format must also include the sample, otherwise the export will fail.
- **Clusters.** A **Cell Clusters**  element to be exported with the matrix.
- **Cell annotations.** A **Cell Annotations**  element to be exported with the matrix.
- **Use compression.** Select among gzip, zip or no compression which is default.
- **Output file name.** Shows the name of the file. This can be customized by changing the default pattern in **Custom file name**.

The h5Seurat exporter creates a h5seurat file format version 4.0.0, see <https://mojaveazure.github.io/seurat-disk/articles/h5Seurat-spec.html> for details. In particular:



- A single assay at `assays/RNA` is created.
- The feature names are written at `assays/RNA/features`.
- The raw counts are written as a sparse matrix at `assays/RNA/counts`.
- The cell ID, composed of sample and barcode, as specified by **Cell format**, is written at `cell.names`.
- The cells' samples are written at `meta.data/orig.ident`.
- Any exported clusters and annotations are written as additional HDF5 groups at `meta.data`.

5.4 Export Loom Expression Matrix

For export in Loom format, a number of options can be specified:

- **Cell format.** Can be specified to include both the sample and barcode. Hover the mouse over this setting to see the possible options. Often the barcode in itself is sufficient as a unique identifier of cells. However, when an expression matrix contains multiple samples, the cell format must also include the sample, otherwise the export will fail.
- **Clusters.** A **Cell Clusters**  element to be exported with the matrix.
- **Cell annotations.** A **Cell Annotations**  element to be exported with the matrix.
- **Export normalized.** If selected and the matrix has been normalized with Normalize Single Cell Data (see section 7.4), the normalized expressions will be exported instead of the raw counts. If the matrix has not been normalized, the export will fail.
- **Use compression.** Select among gzip, zip or no compression which is default.
- **Output file name.** Shows the name of the file. This can be customized by changing the default pattern in **Custom file name**.

The Loom exporter creates a Loom file in format version 3.0.0, see <https://linnarssonlab.org/loompy/format/index.html> for details. In particular:

- The version number is written as an HDF5 dataset at `/attrs/LOOM_SPEC_VERSION`.
- The expression data, either raw counts or normalized, is written as an HDF5 dataset at `/matrix`.
- When exporting an **Expression Matrix** , an empty `layers` HDF5 group is created.
- When exporting an **Expression Matrix with spliced and unspliced counts** , the spliced and unspliced counts are written as HDF5 datasets at `/layers/spliced` and `/layers/unspliced`, respectively.
- The cell ID, composed of sample and barcode, as specified by **Cell format**, is written as an HDF5 group at `col_attrs/CellID`.
- The feature IDs and names are written as HDF5 groups at `col_attrs/Accession` and `col_attrs/Gene`, respectively.
- Any exported clusters and annotations are written as additional HDF5 groups at `col_attrs`.

5.5 Export MEX Expression Matrix

For export in MEX format, a number of options can be specified:

- **Cell format.** Can be specified to include both the sample and barcode. Hover the mouse over this setting to see the possible options. Often the barcode in itself is sufficient as a unique identifier of cells. However, when an expression matrix contains multiple samples, the cell format must also include the sample, otherwise the export will fail.

- **Export normalized.** If selected and the matrix has been normalized with Normalize Single Cell Data (see section 7.4), the normalized expressions will be exported instead of the raw counts. If the matrix has not been normalized, the export will fail.
- **Use compression.** Select among gzip, zip or no compression which is default.
- **Output file name.** Shows the name of the file. This can be customized by changing the default pattern in **Custom file name**. The default configuration will export to a sub-directory named after the expression matrix ({name}).

The export outputs barcodes, features, and expressions to barcodes.tsv, features.tsv, and matrix.mtx, respectively. If the matrix is an **Expression Matrix with spliced and unspliced counts** (📊), two additional files are produced containing spliced (spliced.mtx) and unspliced (unspliced.mtx) counts.

5.6 Export Plain Text Table Expression Matrix

Expression matrices can be exported in plain-text table format. A number of options can be specified:

- **Cell format.** Can be specified to include both the sample and barcode. Hover the mouse over this setting to see the possible options. Often the barcode in itself is sufficient as a unique identifier of cells. However, when an expression matrix contains multiple samples, the cell format must also include the sample, otherwise the export will fail.
- **Feature format.** Choose whether the features' (genes or transcripts) names (e.g., DDX11L1) or ids (e.g., ENSG00000223972.5) will be exported.
- **Table layout.** Choose whether the table should have cells in columns and features in rows, or if it should be transposed such that features are in columns and cells are in rows.
- **Separator.** Choose the column separator.
- **Use compression.** Select among gzip, zip or no compression which is default.
- **Output file name.** Shows the name of the file. This can be customized by changing the default pattern in **Custom file name**.

Note that the matrix exported will be dense and that it will also include values for unexpressed cells and features. It may take up a lot of space compared to the other export formats which are all sparse.

5.7 Export Peak Count Matrix

A peak count matrix can be exported to the following formats:

- Cell Ranger HDF5.
- MEX.

A number of options can be specified for both exporters:

- **Cell format.** Can be specified to include both the sample and barcode. Hover the mouse over this setting to see the possible options. Often the barcode in itself is sufficient as a unique identifier of cells. However, when an expression matrix contains multiple samples, the cell format must also include the sample, otherwise the export will fail.
- **Peak annotations.** If selected, the nearby genes will be exported to a separate file named `peak_annotations.tsv`.
- **Peak-motif associations.** If selected, the transcription factors will be exported to a separate file named `peak_motif_mapping.bed`.

The format of peak annotations and peak-motif annotations is as described in Import Peak Count Matrix (see section 4.6). The peak annotations have 6 columns.

- **Use compression.** Select among gzip, zip or no compression which is default.
- **Output file name.** Shows the name of the file. This can be customized by changing the default pattern in **Custom file name**.

Part III

Single Cell Analysis

Chapter 6

Prepare Reads


Contents

6.1 Annotate Single Cell Reads	60
6.1.1 Read structure	61
6.1.2 Barcodes from names	63
6.1.3 Barcode correction	63
6.1.4 Sample name	65
6.1.5 The output of Annotate Single Cell Reads	65

6.1 Annotate Single Cell Reads

Annotate Single Cell Reads is available from:

Tools | Single Cell Analysis  | **Annotate Single Cell Reads** 

The tool takes as input one or more Sequence Lists (). For each input it outputs an 'annotated reads' sequence list, where reads only contain the biological sequence and are annotated with cell barcode, UMI and/or hashtag, as relevant. These reads can be used as input for:

- **Single Cell RNA-Seq Analysis**, see section [7.1](#)
- **Map Reads to Reference**, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Map_Reads_Reference.html, to produce a read mapping for **Single Cell ATAC-Seq Analysis**, see section [12.1](#)
- **Single Cell V(D)J-Seq Analysis**, see section [13.1](#)
- **Create Cell Annotations from Hashtags**, see section [15.3](#)

The cell barcode can be obtained as a combination of nucleotides from the read structure (see section [6.1.1](#)), part of the read name, and part of the input name (see section [6.1.2](#)). The cell barcode needs to be defined through at least one of these three options.

6.1.1 Read structure

The input reads usually contain the biological sequence and optionally a cell barcode, UMI, and/or hashtag. Their location on the reads is configured in the **Read structure** dialog (figure 6.1). See section 6.1.2 if the cell barcode is not part of the input reads.

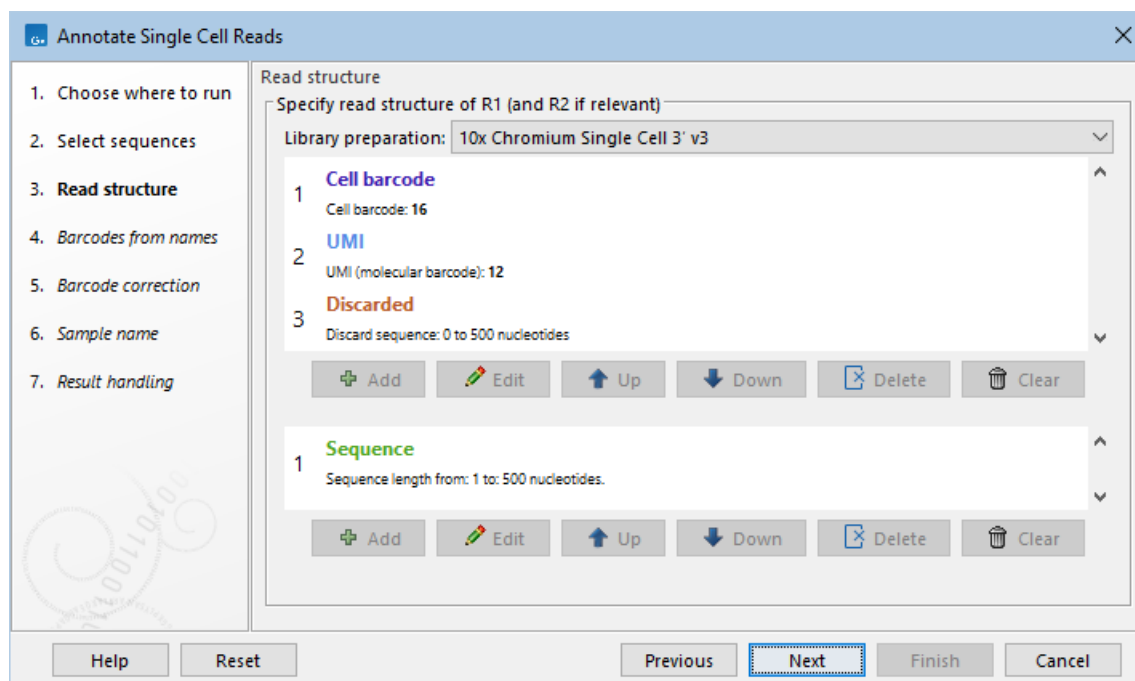


Figure 6.1: The read structure for a 10x 3' gene expression protocol.

Under **Library preparation**, predefined read structures are available for several protocols.

Multiome ATAC

When **Library preparation** is set to '10x Chromium Single Cell Multiome ATAC', the 10x Multiome ATAC barcodes are translated to 10x Multiome GEX barcodes.

This enables the combined analysis of ATAC and GEX data, for example in dimensionality reduction plots, see chapter 16. This is illustrated in the workflow Chromatin Accessibility and Expression Analysis from Reads, see section 19.3. Reads with barcodes that cannot be translated are discarded.

It is not possible to customize this read structure whilst retaining the barcode translation.

Parse Biosciences

When **Library preparation** is set to one of the Parse Biosciences options, the barcodes are translated to the well number they originate from.

The barcode order on the read structure is reversed compared to the barcoding rounds. Thus, the first barcode on R2 corresponds to the third barcoding round. During barcode translation, the barcode order is reversed so that it corresponds to the order of the barcoding rounds.

It is not possible to customize the Parse Biosciences read structures whilst retaining the barcode

translation.

After the third barcoding round, cells/nuclei are pooled and optionally further split into distinct populations, known as sublibraries. Each sublibrary has a unique fourth barcode (the Illumina sample index). When sublibraries are used, it is crucial to check one of the options from **Barcodes from names**, see 6.1.2, to correctly annotate the reads with the fourth barcode.

Custom read structure

When **Library preparation** is set to 'Custom', the read structure of the protocol previously selected can be updated in the two panels below.

The top panel contains the structure of R1 of a pair, or single-end reads. The bottom panel describes R2. For single-end reads, the configuration in the bottom panel is ignored.

Five different types of tags can be used for defining the read structure:

- Sequence
- Cell barcode
- UMI
- Hashtag
- Discarded

Only the **Sequence** part of the read will be retained in the 'annotated reads' list. The parts of the reads corresponding to the other tags are removed. **Cell barcode**, **UMI** and **Hashtag** are added as annotations on the read to be used by downstream tools.

Consider the read structure from figure 6.1. R1 consists of a 16 nt cell barcode, followed by a 12 nt UMI, with an additional sequence of variable length (0 to 500 nt) that is discarded. Read pairs with an R1 shorter than 28 (=16+12) nt or longer than 528 (=16+12+500) nt do not match the read structure and will be discarded. As only R2 contains a 'Sequence' tag, the output will be single-end reads containing R2 from the original pairs, and annotated with the cell barcode and UMI from R1.

It is important that the read structure describes the full length of the read. If the variable length 'Discarded' tag was not included in R1 (figure 6.1), only read pairs with an R1 of exactly 28 nt would match the read structure.

Many different library structures are listed at https://teichlab.github.io/scg_lib_structs/. Figure 6.2 shows the configuration for Microwell-seq, as described in the resource at the time of writing: R1 contains 6 nt cell barcode + 15 nt adapter + 6 nt cell barcode + 15 nt adapter + 6 nt cell barcode + 6 nt UMI + polyA, while R2 contains the biological insert. The tool would construct a single 18 nt cell barcode from the three tags of 6nt each. More general constructions are also possible. For example, if two UMI tags are defined, one on R1 and one on R2, then a single UMI will be constructed from both parts.

```

1  Cell barcode
   Cell barcode: 6
2  Discarded
   Discard sequence: 15 to 15 nucleotides
3  Cell barcode
   Cell barcode: 6
4  Discarded
   Discard sequence: 15 to 15 nucleotides
5  Cell barcode
   Cell barcode: 6
6  UMI
   UMI (molecular barcode): 6
7  Discarded
   Discard sequence: 0 to 500 nucleotides

```

Figure 6.2: Configuration for R1 from Microwell-seq. A single 18 nt cell barcode will be constructed from the three tags of 6nt each.

6.1.2 Barcodes from names

In some protocols, the cell barcode is not included in the read. For instance, the Illumina sample index can be used as a barcode:

- For Smart-seq2, the cell barcode is the sample index.
- For Parse Biosciences, the sample index can optionally be used as the fourth barcode, see section 6.1.1.

Software converting Illumina output to FASTQ typically includes the sample index in the read name for both R1 and R2. Alternatively, separate FASTQ files can be generated for each sample index.

The cell barcode can be extracted from the name of the reads and/or input sequence list by adjusting the options in the **Barcodes from names** dialog (figure 6.3):

- **Read/Input names define barcodes.** When checked, cell barcodes are extracted from the read/input names, according to the structure configured in **Read/Input name structure**.
- **Read/Input name structure.** Specify how to extract cell barcodes from the read/input names. The cell barcode does not have to consist of nucleotides. A combination of placeholders and text can be used (figure 6.3) to extract the part(s) of the name containing the cell barcode. Hover the mouse cursor over the field to see a tooltip with several examples. Simultaneously pressing Shift and F1 displays all available placeholders.

The **Barcodes from names** dialog has two previews, displaying how the read/input names are parsed into the cell barcode (figure 6.3). These previews help ensure that the name structure matches the input. If the configured structure is invalid, the preview may fail to determine the cell barcode.

6.1.3 Barcode correction

Sequencing errors and low-quality bases can result in one cell barcode being represented as distinct barcodes in the data, with:

Annotate Single Cell Reads

1. Choose where to run
2. Select sequences
3. Read structure
4. **Barcodes from names**
5. Barcode correction
6. Sample name
7. Result handling

Barcodes from read names

☒ Read names define barcodes

Read name structure:

Press Shift + F1 for options

Preview barcodes from read names

Input	Cell barcode
A26:19:H5TH:1:10:136:10 1:N:0:GAGTGG	GAGTGG
A26:19:H5TX:1:12:249:34 2:N:0:CACCGG	CACCGG

Barcodes from input names

☒ Input names define barcodes

Input name structure:

Press Shift + F1 for options

Preview barcodes from input names

Input	Cell barcode
MySample-CellA-123	A
MySample-CellZ-987	Z

Buttons: Help, Previous, **Next**, Finish

Figure 6.3: Extracting cell barcodes from the read and input names. The two previews show how the names are parsed into cell barcodes.

- Ambiguous nucleotides.
- Incorrectly sequenced nucleotides, e.g. an 'A' in the barcode that was sequenced as a 'C'.

Such errors can be corrected in each barcode component independently using the options in the **Barcode correction** dialog:

- **No correction.** Do not correct the barcodes.
- **Use sample.** Correct barcodes to other barcodes found in the sample.
A barcode 'A' is corrected to a barcode 'B' without ambiguous nucleotides if they differ at positions where 'A' either has:
 - A non-ambiguous nucleotide. Only one difference is allowed. 'B' must have at least 4 times as many UMIs/reads as 'A'.
 - Ambiguous nucleotides.
- **Use whitelist.** Correct barcodes not on the whitelist to other barcodes found in the sample and on the whitelist.
A barcode 'A' is corrected to a barcode 'B' if they differ at positions where 'A' either has:
 - A non-ambiguous nucleotide. Only one difference is allowed.
 - Ambiguous nucleotides.

Barcodes not on the whitelist after correction are discarded.

The whitelist file must contain one cell barcode per line, arranged alphabetically.

If one barcode can be corrected to several barcodes, the one with the highest number of UMIs/reads is used.

Several of the predefined read structures under **Library preparation** have inbuilt whitelists. For these, barcode correction is enforced.

6.1.4 Sample name

The 'annotated reads' are also annotated with a sample name (see section 6.1.5). When jointly analyzing scATAC-Seq or scV(D)J-Seq with matched scRNA-Seq data, it is important that reads originating from the same sample are annotated with the same sample name. This can be configured in the **Sample name** dialog:




- **Sample name.** Specify how to set the sample name. A combination of placeholders and text can be used to set the sample name. Hover the mouse cursor over the field to see a tooltip with several examples. Simultaneously pressing Shift and F1 displays all available placeholders. Different placeholders are available when the tool is run from the Tools menu or as part of a workflow.

Parse Biosciences

When using a Parse Biosciences kit, multiple samples can be present in a single sequence list, but only one sample is assigned by Annotate Single Cell Reads. The final sample name can be set using Demultiplex Parse Bio Samples, see section 7.3.

6.1.5 The output of Annotate Single Cell Reads

Annotate Single Cell Reads produces the following outputs:

- A **Sequence List** () of 'annotated reads'.
- Optionally, a **Sequence List** () of 'unmatched reads' that did not match the configured options.
- Optionally, a **Report** () containing various summaries.

The annotated reads

The 'annotated reads' contain just the 'Sequence' part of the read structure for the input reads that matched the configured options. These reads are suitable for use in several tools, see section 6.1. Note that the output reads are sorted by their cell barcode, UMI and hashtag, if used, so they will appear shuffled compared to the input.

The table view of the sequence list (see https://resources.qiagenbioinformatics.com/manuals/clogenomicsworkbench/current/index.php?manual=Table_view_sequence_lists.html) contains the annotations added to the reads.

The barcode in the 'Cell barcode' column is formed by 'Cell barcode' tags in the read structure and/or the barcode extracted from names, joined by a '-'.

Barcodes that are corrected (see section 6.1.3) and/or translated (see 'Multiome ATAC' and 'Parse Biosciences' in section 6.1.1) contain the final barcode under 'Cell barcode', and the uncorrected and untranslated barcode under 'Original barcode'.

The unmatched reads

The 'unmatched reads' contain the input reads that did not match the configured options:

- The reads did not match the **Read structure**. Typically these reads are too short. If longer reads are present, it may be worth verifying if the read structure includes a variable length tag.
- The read names did not match the **Read name structure**.
- The cell barcodes for the reads were discarded because they are not on the whitelist.

The report

The report includes the following sections:

Summary This section contains information about the number of:

- Input, annotated and unmatched reads.
- Distinct cell barcodes, as well as hashtags, if used. If multiple barcode components are defined, the total number of barcodes reflects the final joined cell barcodes.

Barcode correction This section is present if barcode correction was used. It contains, for each barcode component, the number of barcodes that were:

- Identified.
- Corrected.
- Discarded, if a whitelist was used.
- Retained.

Barcode ranks A plot ranking the cell barcodes in decreasing order of the number of reads (figure 6.4). The number of cells present in the data can be approximated by the location of a sharp fall in the plot.

Nucleotide counts Plots showing the distributions of different nucleotides at each position for the tags extracted using the **Read structure** (cell barcode, UMI, hashtag) for the 'annotated reads'.

The distributions for cell barcodes and UMIs are both expected to be roughly uniform, while the distribution for hashtags varies depending on the number of distinct hashtags in the data. The distribution becomes more uniform with more hashtags, but if only a few hashtags are expected, for example when the hashtag represents the sample, the distribution is likely to be skewed and should reflect the known expected hashtags.

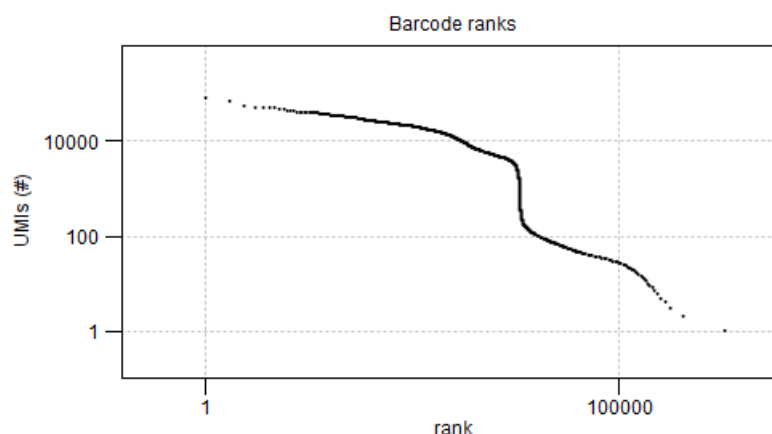


Figure 6.4: A barcode ranks plot. A sharp transition from an average of a bit less than 10,000 reads to less than 100 reads per barcode is seen at $x = 5,000$, suggesting that there are approximately 5,000 cells in the data.

For simplicity, the remainder of this section will talk about 'barcodes', but the description is equally true for UMIs.

Typically, barcodes are randomly generated, or else designed to be very different from each other, such that all nucleotides are observed at each barcode position, and in approximately equal amounts. Errors may be detected when the barcode plots do not show this behavior, such as in figure 6.5, where position 1 in the barcode is mostly 'A', position 2 is mostly 'A', position 3 is mostly 'G' etc. It appears that one barcode contains almost all the reads in the sample. In this case, the cell barcode part of the read structure has been misconfigured to read an adapter with sequence 'AAGCAGTGGT'. The same plot with the correct read structure is shown in figure 6.6.

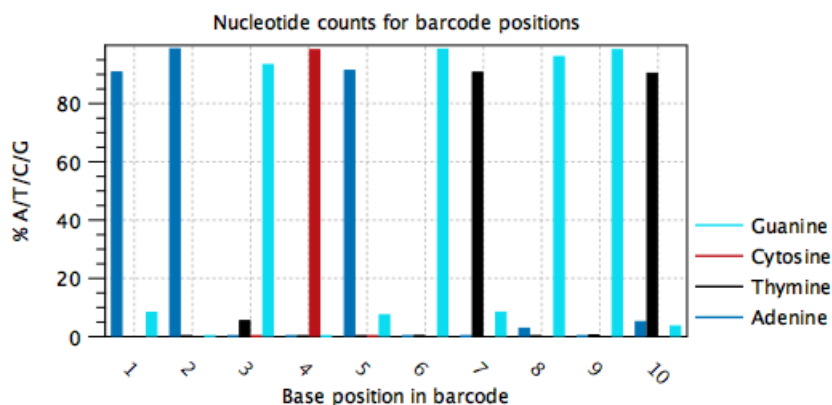


Figure 6.5: Nucleotide counts plot for a misconfigured barcode. One barcode with sequence 'AAGCAGTGGT' is present in most of the reads. In this case the barcode was misconfigured to be part of an adapter.

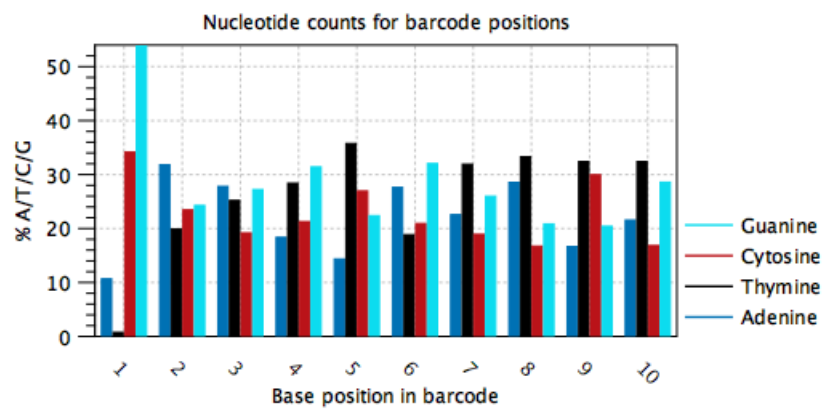


Figure 6.6: Nucleotide counts plot for the same data as in figure 6.5. All nucleotides are seen at all positions of the barcode with comparable frequencies, except for at position 1. This dataset is from a 96-well protocol where the barcodes for each well are known in advance and the skewed nucleotide distribution at position 1 is as expected.

Chapter 7

Creating a Gene Expression Matrix




Contents

7.1 Single Cell RNA-Seq Analysis	69
7.1.1 The Single Cell RNA-Seq Analysis report	72
7.1.2 The Single Cell RNA-Seq Analysis algorithm	76
7.2 QC for Single Cell	77
7.2.1 Empty droplets filter	78
7.2.2 Count-based and extra-chromosomal filters	80
7.2.3 Doublets filter	82
7.2.4 The output of QC for Single Cell	84
7.2.5 Choosing barcodes to retain	92
7.2.6 Cell calling	93
7.2.7 Automatic thresholds	94
7.2.8 Doublet calling	95
7.3 Demultiplex Parse Bio Samples	96
7.4 Normalize Single Cell Data	98
7.4.1 When is batch correction appropriate?	100
7.4.2 The output of Normalize Single Cell Data	100
7.4.3 The Normalize Single Cell Data algorithm	103
7.5 The Expression Matrix element	105

7.1 Single Cell RNA-Seq Analysis

Single Cell RNA-Seq Analysis is available from:

Tools | Single Cell Analysis  | **Gene Expression**  | **Cell Preparation**  | **Single Cell RNA-Seq Analysis** 

The tool takes as input one or more Sequence Lists () of reads that have been annotated using **Annotate Single Cell Reads**. It outputs an Expression Matrix with spliced and unspliced counts () for gene expressions, and optionally an Expression Matrix () for transcript expressions, a report, and unmapped reads.

Sample: All input sequence lists must originate from the same sample, which is set when executing the **Annotate Single Cell Reads** tool (see section 6.1). This is because **Single Cell RNA-Seq Analysis** assumes that reads with the same cell barcode that are present in different inputs represent the same cell. The wizard does not allow executing the tool with inputs that are annotated with different samples.

It is important to provide all the data for a sample to **Single Cell RNA-Seq Analysis** at the same time. For example, if one sample was sequenced on 4 lanes of an Illumina sequencer, then all 4 lanes should be supplied together. This allows reads originating from the same cell, but coming from different lanes, to be analyzed jointly such that amplification duplicates are detected using UMIs and only give one count in the output Expression Matrix.

Matrix with spliced and unspliced counts: The Expression Matrix with spliced and unspliced counts (📊) is an extension of the Expression Matrix (📊) containing separate information about the spliced and unspliced reads for each cell and gene. Reads mapping to transcripts are counted towards the spliced expression of a gene, while reads mapping to a gene but not a transcript, such as introns of known transcripts, or upstream/-downstream of known transcripts, are counted towards the unspliced expression. The Expression Matrix with spliced and unspliced counts (📊) can be used as input to any tool that accepts an Expression Matrix (📊).

Filtering: The output matrix should be filtered by **QC for Single Cell** before being used in any other tool in the CLC Single Cell Analysis Module. This is because sequencing errors often lead to many barcodes that have few counts, and which do not represent real cells. If no filtering is performed, the large number of barcodes can cause downstream tools to run extremely slowly and results can be negatively affected by the added noise.

Barcode whitelists: In some protocols, the set of valid barcodes is known in advance, and available as a barcode whitelist. In CLC Single Cell Analysis Module, it is not possible to directly use such a list. Instead, **QC for Single Cell** is usually able to detect the barcodes that correspond to cells using the Empty droplets filter (see section 7.2.1), and to prevent specific barcodes from being filtered away (see section 7.2.5).

The tool requires a genome - supplied as **References**, and both a **Gene track** and a corresponding **mRNA track**. These data can be obtained in two ways:

- Directly downloaded as tracks using the Reference Data Manager (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Download_Genomes.html).
- Imported as tracks from fasta and gff/gff3/gtf files (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Import_tracks.html).

The following additional options are available:

- **Use spike-in controls.** Includes spike-in controls in the output, which can be used downstream in the QC for Single Cell tool. A spike-in section is also added to the report produced by this tool.

- **Spike-in controls** The spike-in controls. To learn how to import spike-in control files, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Import_RNA_spike_in_controls.html.
- **Strand setting.** This option controls whether the reads should be mapped in the same orientation as the transcript from which they originate (forward), in the reverse direction (reverse), or to both directions (both). The ‘forward’ and ‘reverse’ options allow assignment of reads to the correct gene in cases where overlapping genes are located on different strands. Without the strand-specific protocol, this would not be possible (see [Parkhomchuk et al., 2009]). For many single cell library preparations, one read of a pair, which is usually discarded, binds to the polyA tail of transcripts. This means that the remaining read should usually be mapped with strand specific ‘forward’.
- **Coverage bias.** The expected coverage bias determines whether it is possible to produce an Expression Matrix for transcript expressions, and also affects the quality control applied to the ‘Gene/transcript length coverage’ section of the report.
 - **Unbiased.** An Expression Matrix for transcript expressions can be produced. The expected coverage is uniform across the bodies of transcripts.
 - **Targeted.** An Expression Matrix for transcript expressions cannot be produced. This is because several transcripts may be amplified by the same primers, meaning it is often not possible to determine the transcript of origin for a read. The expected coverage has no particular bias.
 - **3’ bias.** An Expression Matrix for transcript expressions cannot be produced. This is because several transcripts may end at the same genomic position, meaning it is often not possible to determine the transcript of origin for a read. The expected coverage is 3’ biased.
- **Include intronic reads in total expression** By default, the total expression of a gene is given by the spliced expression. When this option is enabled, the total expression is set instead to the sum of spliced and unspliced counts. This option is recommended for single nucleus RNA sequencing (snRNA-Seq), where data is usually analyzed by counting expression from both exons and introns [Bakken et al., 2018].
- **Group by UMIs.** When enabled, reads with the same cell barcode and UMI are counted as 1 such that the output expressions have no amplification bias. When disabled, reads with the same cell barcode and UMI are counted separately.
- **Output report.** When enabled, a detailed report is produced, see section 7.1.1.
- **Output transcript matrix.** When enabled, an Expression Matrix for transcript expressions is produced.
- **Output unmapped reads.** When enabled, up to two lists with reads that did not map are produced, one for paired reads and one for single reads. The unmapped reads consist of those reads that did not map to the reference at all, or that mapped equally well to more than 10 distinct places in the reference sequence.

For paired reads, pairs that mapped to different genes are output in the unmapped paired reads list, while members of broken pairs are output in the unmapped single reads list.

7.1.1 The Single Cell RNA-Seq Analysis report

An example of an scRNA-Seq report is shown in figure 7.1.

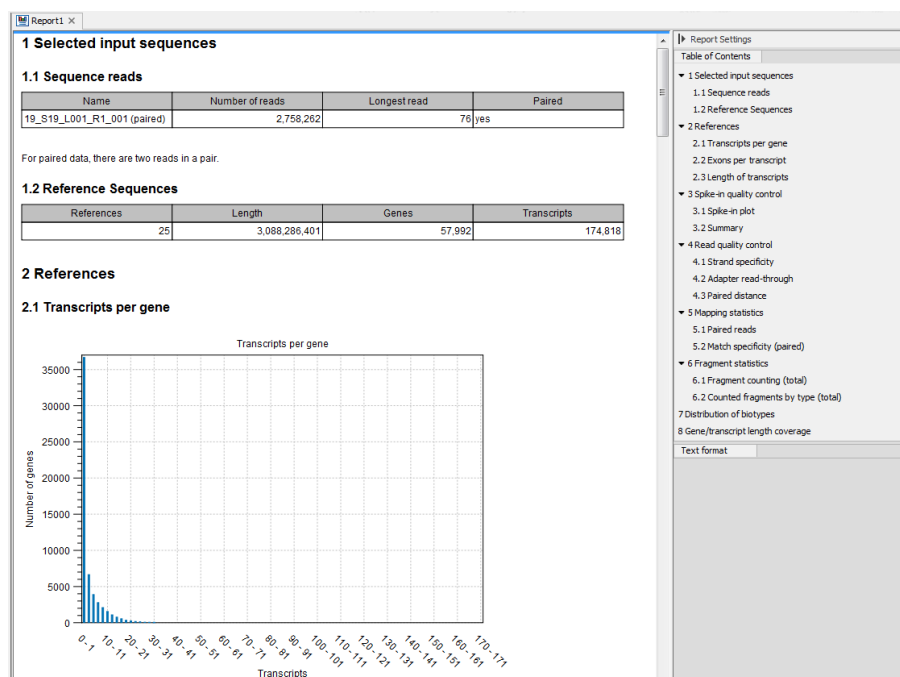


Figure 7.1: Report of an RNA-Seq run.

The report is a collection of the sections described below, some sections included only based on the input provided when starting the tool. If a section is flagged with a pink highlight, it means that something has almost certainly gone wrong in the sample preparation or analysis. A warning message tailored to the highlighted section is added to the report to help troubleshoot the issue. The report can be exported in PDF or Excel format.

Selected input sequences

Information about the sequence reads provided as input, including the number of reads in each sample, as well as information about the reference sequences used and their lengths.

References

Information about the total number of genes and transcripts found in the reference:

- **Transcripts per gene.** A graph showing the number of transcripts per gene.
- **Exons per transcript.** A graph showing the number of exons per transcript.
- **Length of transcripts.** A graph showing the distribution of transcript lengths.

Spike-in quality control

- **Spike-in plot.** A plot shows the expression of each spike-in as a function of the known concentration of that spike-in (see figure 7.2 to see an optimal spike-in plot).

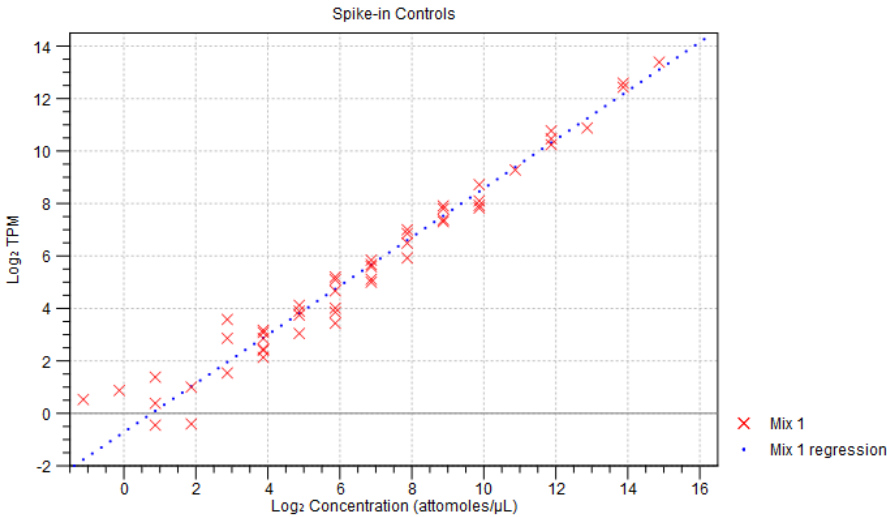
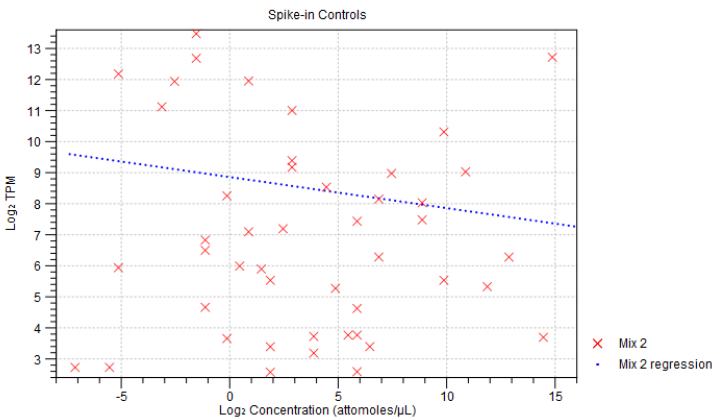


Figure 7.2: Spike-in plot showing how the points fall close to the regression line at high concentration.

- **Summary table.** A table provides more details on the spike-in detection. Figure 7.3 shows a failed spike-in control, with a table where results that require attention are highlighted in pink.



3.2 Summary

Number of spike-ins detected	46/92
R ²	0.05
Reads mapped to spike-ins	7,336
% of reads mapped to spike-ins	3.10
Lower limit of detection (attomoles/μL)	0.03

Figure 7.3: Summary table where less than optimal results are highlighted.

Under the table, a **warning message** explains what the optimal value was, and offers some troubleshooting measures: When samples have poor correlation ($R^2 < 0.8$) between known and measured spike-in concentrations, it indicates problems with the spike-in protocol, or a more serious problem with the sample. To troubleshoot, check that the correct spike-in file has been selected, and control the integrity of the sample RNA. Also, if fewer than 10000 reads mapped to spike-ins, check that the correct spike-in sequences are specified, and consider using more spike-in mix in future experiments.

Read quality control

This section includes:

- A **strand specificity table** that indicates the direction of the RNA fragment that generated the read. Strandedness can only be defined for reads that map to a gene or transcript. Of these reads, the number of "Reads with known strand" is used in determining the percentage of reads ignored due to being on the wrong strand, and the subsequent percentage of reads with the wrong strand. In a strand-specific protocol, almost all reads are generated from a specific orientation, but otherwise a mix of both orientations is expected.
 - A warning message will appear if over 90% of reads were mapped in the same orientation but the tool was run without using a strand specific setting ("Forward"/"Reverse").
 - If over 25% of the reads were filtered away due to the strand specific setting, try to re-run the tool with strand specific setting "Both". However, if a strand-specific protocol was used, library preparation may have failed.
- A **percentage of mapped paired-end reads containing read-through adapters**. If present in above 10% of the reads, adapters may lead to false positive variant calls or incorrect transcript quantification (because reads must align within transcript annotations to be counted towards expression). Read-through adapters can be removed using the Trim Reads tool. In future experiments, consider selecting fragments that are longer than the read size. Note that single base extensions such as TA overhangs will also be classed as read-through adapters, and in these cases the additional base should be trimmed. Note also that trimming of read starts (5' trim) can lead to spurious detection of read-through adapters, because the trimming increases the number of read pairs where the end of one read aligns over the (trimmed) start of the other.
- A **paired distance graph** (only included if paired reads are used) shows the distribution of paired-end distances, which is equivalent to the distribution of sequenced RNA fragment sizes. There should be a single broad peak at the target fragment size. An asymmetric peak may indicate problems in size selection.

Mapping statistics

Shows statistics on:

- **Paired reads** or **Single reads**. The table included depends on the reads used. The table shows the number of reads mapped or unmapped, and in the case of paired reads, how many reads mapped in pairs and in broken pairs.

If over 50% of the reads did not map, and the correct reference genome was selected, this indicates a serious problem with the sample. To troubleshoot, the report offers the following options:

 - Check that the correct reference genome and any relevant gene/mRNA tracks have been provided.
 - The mapping parameters may be too strict. Try resetting them to the default values.
 - Try mapping the unmapped reads against possible contaminants. If the sample is contaminated, enrich for the target species before library preparation in future experiments.

- Library preparation may have failed. Check the quality of the sample RNA.

In case paired reads are used and over 40% of them mapped as broken pairs, the report hints that there could be problems with the tool settings, a low quality reference sequence, or incomplete gene/mRNA annotations. It could also indicate a more serious problem with the sample. To troubleshoot, it is suggested to:

- Check that the correct reference genome and any relevant gene/mRNA tracks have been provided.
 - Try re-running the tool with the "Auto-detect paired distances" option selected.
 - Check that the paired-end distances on the reads are set correctly. These are shown in the "Element Information" view on the reads. If these are correct, try re-running the tool without the "Auto-detect paired distances" option.
 - Try mapping the reads against possible contaminants. If the sample is contaminated, enrich for the target species before library preparation in future experiments.
- **Match specificity.** Shows a graph of the number of match positions for the reads. Most reads will be mapped 0 or 1 time, but there will also be reads matching more than once in the reference.

Fragment statistics

- **Fragment counting.** Lists the total number of fragments used for calculating expression, divided into uniquely and non-specifically mapped reads, as well as uncounted fragments (see the point below on match specificity for details).
- **UMI fragment counting.** Lists the total number of distinct UMI fragments used for calculating expression. This table is only included if the Library type setting is 3' sequencing and if the input reads are single end reads annotated with UMIs by tools of the Biomedical Genomics Analysis plugin.
- **Counted fragments by type.** Divides the fragments that are counted into different types, e.g., uniquely mapped, non-specifically mapped, mapped. A last column gives the percentage of fragments mapped for a particular type.
 - **Total gene reads.** All reads that map to the gene.
 - **- Intron.** From the total gene reads, reads that fall partly or entirely within an intron.
 - **- Exon.** From the total gene reads, reads that fall entirely within an exon or in an exon-exon junction.
 - **- - Exon.** From the total gene - exon reads, reads that map completely within an exon
 - **- - Exon-exon.** From the total gene - exon reads, reads that map across an exon junction .
 - **Intergenic.** All reads that map partly or entirely between genes.
 - **Total.** Total amount of reads for a particular type.

- **Counted UMI fragments by type.** Divides the distinct UMI fragments that are counted into different types, e.g., uniquely mapped, non-specifically mapped, mapped. The table contains the same rows as the 'Counted fragments by type' table (see above). It is only included in the report if the Library type setting is 3' sequencing and if the input reads are single end reads annotated with UMIs by tools of the Biomedical Genomics Analysis plugin.

Distribution of biotypes

Table generated from biotype annotations present on the input gene or mRNA tracks. If using both gene and mRNA tracks, the biotypes in the report are taken from the mRNA track.

- For genes, biotypes can be any of the following columns: "gene_biotype", "biotype", "gbkey", "type". The first one in this list is chosen.
- For transcripts, biotypes can be any of the following columns: "transcript_biotype", "biotype", "gbkey", "type". The first one in this list is chosen.

The biotypes are "as a percentage of all transcripts" or "as a percentage of all genes". For a poly-A enrichment experiment, it is expected that the majority of reads correspond to protein-coding regions. For an rRNA depletion protocol, a variety of non-coding RNA regions may also be observed. The percentage of reads mapping to rRNA should usually be <15%.

If over 15% of the reads mapped to rRNA, it could be that the poly-A enrichment/rRNA depletion protocol failed. To troubleshoot the issues in future experiments, check for rRNA depletion prior to library preparation. Also, if an rRNA depletion kit was used, check that the kit matches the species being studied.

Gene/transcript length coverage

Plot showing the normalized coverage across a gene/transcript body for four different groupings of gene/transcript length (figure 7.4).

To generate this plot, every transcript is rescaled to have a length of 100. For every read that is assigned to a transcript, we get its start and end coordinates in this "transcript-length-normalized" coordinate system [0,100]. We then increment counters from the read start position to the read end position. After all the reads have been counted, the average 5' count is the average value of the counters at position 0,1,2...49. The average 3' count is the value at positions 51,52,53...100. The difference between average 3' and 5' normalized counts is the difference between these values as a percentage of the maximum number of counts seen at any position.

7.1.2 The Single Cell RNA-Seq Analysis algorithm

Single Cell RNA-Seq Analysis uses the same algorithm as the **RNA-Seq Analysis** tool of the CLC Genomics Workbench. Briefly, the tool extracts the sequence of all transcripts from the provided mRNA track. Reads are then simultaneously aligned to both this transcriptome and the full genome (and spike-in sequences if these have been provided).

Each read may have multiple equally high scoring alignments, some to transcripts and others to the genome. These alignments are translated back into genomic coordinates. In many cases, all the alignments refer to the same genomic coordinates and the read is considered 'uniquely

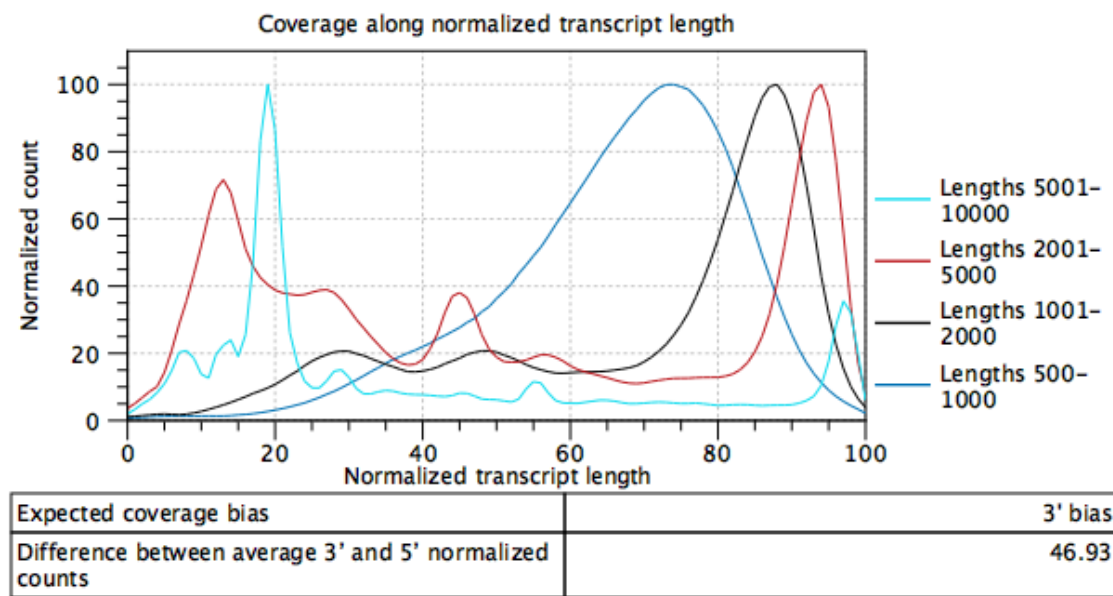


Figure 7.4: Gene/transcript length coverage plot for data with a 3' bias.

mapped'. If there are more than 10 distinct alignments in genomic coordinates, then the read is discarded.

When a read can be aligned equally well to multiple transcripts or multiple genes, it is counted towards only one of these, with the 'lucky' transcript being chosen by an Expectation Maximization (EM) method similar to RSEM and eXpress. This works as follows:

- An 'ambiguity graph' is built that links transcripts that could have given rise to the same reads. At this stage all reads are considered together without reference to their barcodes or UMIs.
- The abundance of each transcript is estimated from this graph.
- The reads are distributed to the different transcripts according to their estimated abundances. Reads that map to genes, but are incompatible with known transcripts are ignored unless the option **Count intronic reads** is enabled. When the option is enabled, these reads are assigned to a gene based on the estimated abundances of the transcripts for each gene.

At this stage, if the option **Group by UMIs** is enabled, then reads with the same barcode and UMI are only counted once. After the first read has been assigned, subsequent reads with the same barcode and UMI are ignored.

The final gene expression is the sum of the expressions of the transcripts for that gene. When the option **Count intronic reads** is enabled, expression from introns and UTRs is also included.

7.2 QC for Single Cell

QC for Single Cell performs quality control and removes barcodes that are either deemed to not contain RNA from a single cell (section 7.2.1 and section 7.2.3), or that are deemed to be of low

quality, based on different metrics (section 7.2.2). These barcodes can contribute to misleading results in the downstream analysis and should be removed as a first step after the Expression Matrix has been created.

The QC for Single Cell tool is available from:

Tools | Single Cell Analysis  | **Gene Expression**  | **Cell Preparation**  | **QC for Single Cell** 

The tool takes an Expression Matrix  /  as input.

Batches and samples: The distribution of the QC metrics can vary between batches. Therefore, QC for Single Cell analyzes each batch separately by treating each sample detected in the input Expression Matrix as a different batch. This might not be appropriate for samples that are sequenced together in one batch, as can be the case for Parse Biosciences data, see section 7.3, or multiplexed samples, see section 18.7. For such data, we recommend running QC for Single Cell before updating the sample name.

7.2.1 Empty droplets filter

In droplet-based data, barcodes can correspond to droplets containing one cell, more cells or no cells at all. In the first dialog of QC for Single Cell, the **Empty droplets filter** can be enabled and customized to remove the droplets that are detected to not contain any cells. This filter should be skipped for single-cell protocols that are not droplet-based.

Note that each droplet is assigned one barcode and these terms can be used interchangeably for droplet-based protocols.

Non-zero counts in empty droplets are obtained from ambient (i.e., extracellular) RNA, that can be captured and sequenced during the protocol. Sequenced empty droplets contain significantly fewer reads, and this can be seen as a sharp transition in the rank plot, shown in figure 7.9.

Droplets can be classified in three categories (see figure 7.9):

- **Ambient:** removed droplets that have a low number of reads which are assumed to only contain ambient RNA.
- **Cells:** retained droplets that have a high number of reads.
- The remaining droplets with an intermediate number of reads can either be cells with low RNA content or empty droplets, and this cannot be determined purely based on the number of reads.

Droplets with a low number of reads are removed as ambient droplets. The threshold for this is usually obtained automatically from the histogram of number of reads, see section 7.2.6 for details.

Droplets with a high number of reads are automatically retained as cell-containing droplets. The threshold for this is usually obtained from the automatically inferred knee from the rank plot, see figure 7.9 and section 7.2.6 for details.

To detect cells with low RNA content, first an ambient RNA profile is estimated from the ambient droplets. The remaining droplets with an intermediate number of reads can be tested against

this profile and are assigned simulation-based FDR-corrected p-values, from which non-empty droplets are identified.

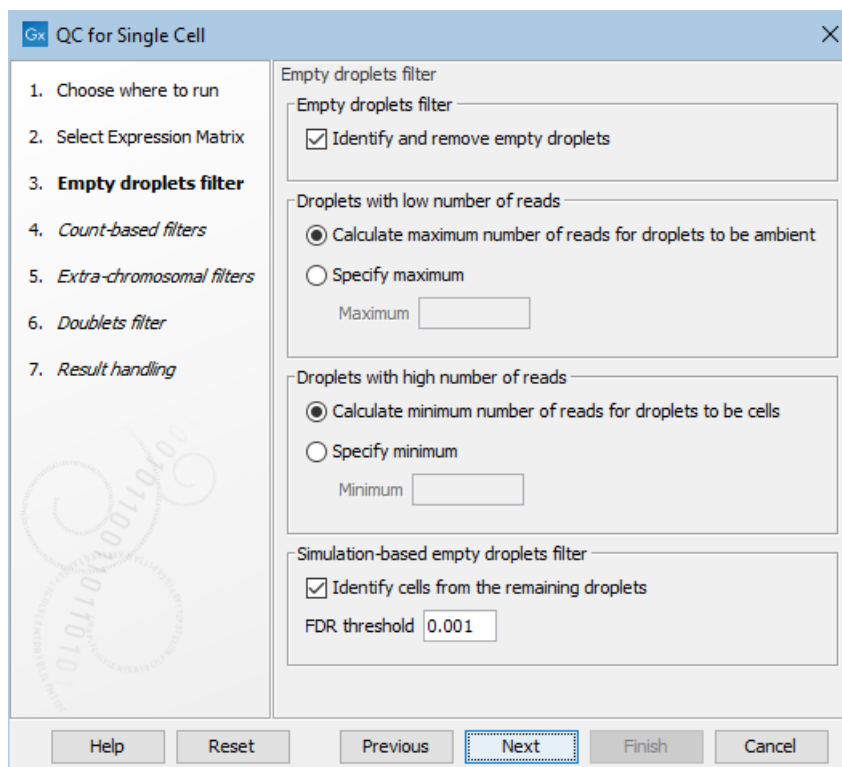


Figure 7.5: The default settings in the Empty droplets filter dialog.

The following options can be adjusted in the **Empty droplets filter** dialog (figure 7.5):

- **Identify and remove empty droplets.** Enables filtering of the empty droplets. This should be unchecked for single-cell protocols that are not droplet-based.
- **Droplets with low number of reads.** Droplets with a total number of reads below a threshold are removed as ambient droplets. The threshold can be calculated automatically (see section 7.2.6 for details) by choosing **Calculate maximum number of reads for droplets to be ambient**, or can be specified manually in the **Maximum** parameter by choosing **Specify maximum**.
- **Droplets with high number of reads.** Droplets with a total number of reads above a threshold are retained as cell-containing droplets. The threshold can be calculated automatically (see section 7.2.6 for details) by choosing **Calculate minimum number of reads for droplets to be cells**, or can be specified manually in the **Minimum** parameter by choosing **Specify minimum**.
- **Identify cells from the remaining droplets.** Enables the simulation-based detection of cells with low RNA content.
- **FDR threshold.** Droplets with FDR-corrected p-values larger than this are removed as empty droplets.

The generated rank plot and summary (see figures 7.9 and 7.10) can be used to identify when the automatic thresholds are not suitable and manual thresholds are required.

After applying the **Empty droplets filter**, only droplets that are identified as non-empty are retained for the remaining filters (see section 7.2.2). Note that this filter does not concern the quality of the retained cells. The **Empty droplets filter** already removes cells with low number of reads, or, by association, low number of expressed features, and enabling the **Count-based filters** is not strictly necessary. The **Extra-chromosomal filters** provide the most additional benefit in this situation.

For removing droplets containing more than one cell, the **Doublets filter** can be used, see section 7.2.3.

7.2.2 Count-based and extra-chromosomal filters

Low quality barcodes can arise for various reasons, such as damaged cells or library preparation problems. Potential low quality barcodes can be identified using the distributions of the following metrics:

- Total number of reads. Barcodes with few reads result from losing RNA during library preparation.
- Total number of expressed features. Barcodes with few expressed features indicate that the diverse transcript population has not been successfully captured.
- Percentage of reads mapped to spike-in control regions. When spike-in controls are used, barcodes with proportionally many reads mapped to the spike-in controls are symptomatic of loss of endogenous RNA, as the same amount of spike-in RNA should have been added to each cell.
- Percentage of reads mapped to features indicative of low quality. Barcodes with proportionally many reads mapped to certain features are indicative of low quality cells. For example, loss of cytoplasmic RNA from perforated cells can lead to high expression of mitochondrial genes in eukaryotes [Islam et al., 2014, Ilicic et al., 2016].

Count-based filters

In this dialog of QC for Single Cell, the filters using the total number of reads and expressed features can be enabled and customized.

The dialog first allows for manually specifying a list of barcodes to be retained as cells in **Barcodes to retain** (figure 7.6). These would typically be barcodes that are otherwise removed by any of the filters applied. See section 7.2.5 for details. The barcodes used in the **Barcodes to retain** have to meet the following criteria:

- Either all barcodes are prepended by the sample name and a "-", or no barcodes contain the sample name.
- Barcodes can be separated by any white-space characters, ",", and ";". This consequently requires that no barcodes contain any of the allowed separators.

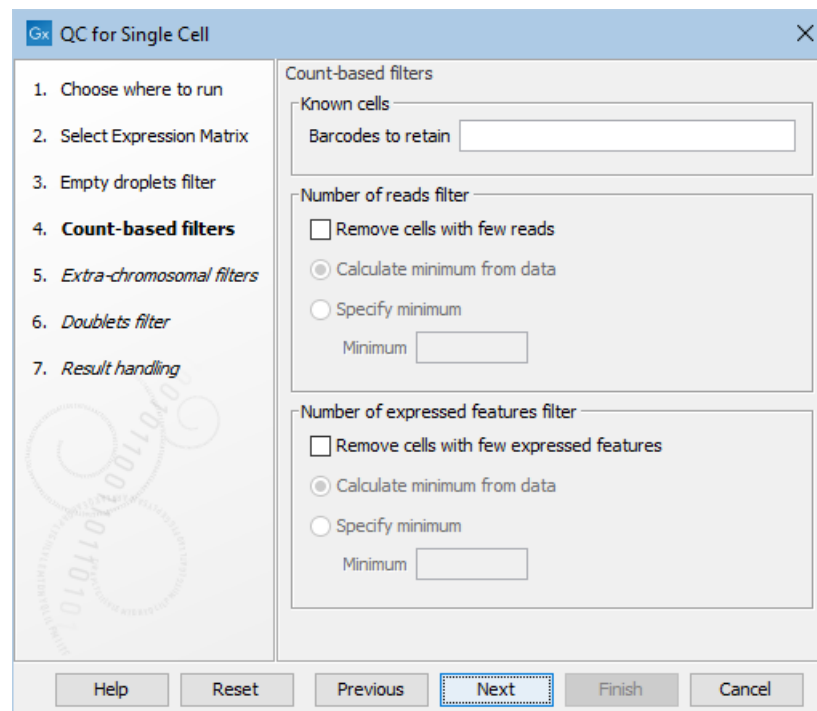


Figure 7.6: The default settings in the *Count-based filters* dialog.

The following options can be adjusted for the **Count-based filters** (figure 7.6):

- **Remove cells with few reads.** When checked, barcodes with fewer reads than a minimum threshold are removed.
- **Remove cells with few expressed features.** When checked, barcodes with fewer expressed features than a minimum threshold are removed.
- The minimum threshold can be:
 - Calculated automatically from the distribution of number of reads/expressed features by using **Calculate minimum from data**. See section 7.2.7 for details.
 - Specified manually by using **Specify minimum**.

Extra-chromosomal filters

In this dialog of QC for Single Cell, the filters using the percentage of reads mapped to spike-in controls and features indicative of low quality can be enabled and customized.

The following options can be adjusted in the **Extra-chromosomal filters** dialog (figure 7.7):

- **Remove cells with many spike-in reads (%).** When checked, barcodes with a percentage of reads mapped to spike-in controls that is greater than a maximum threshold are removed.
- **Remove cells with many reads mapping to features indicative of low quality (%).** When checked, barcodes with a percentage of reads mapped to features indicative of low quality that is greater than a maximum threshold are removed. Features indicative of low quality are defined using at least one of the following options:

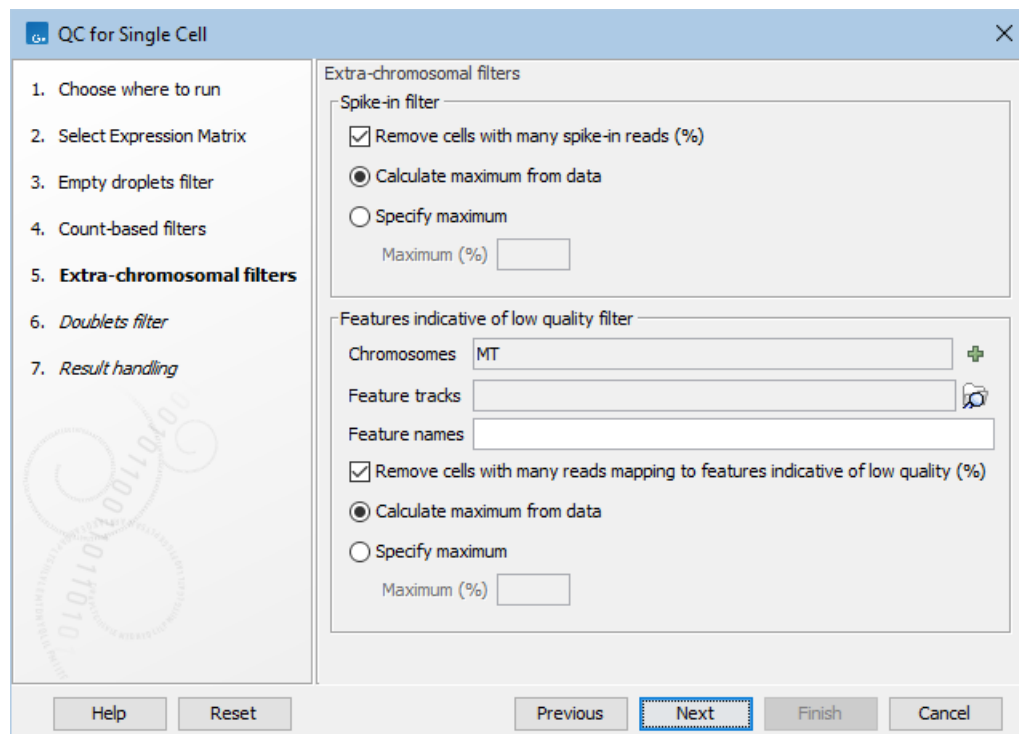


Figure 7.7: The default settings in the Extra-chromosomal filters dialog.

- **Chromosomes.** The name of the mitochondria chromosome and/or other chromosomes containing only features indicative of low quality cells. Can be left empty or multiple chromosomes can be chosen.
- **Feature tracks.** Feature tracks containing only features indicative of low quality cells. Can be left empty or multiple tracks can be chosen.
- **Feature names.** Names or ids for features indicative of low quality cells. Any white-space characters, ",", and ";" are accepted as separators.
- The maximum threshold can be:
 - Calculated automatically from the distribution of percentage of reads mapped to spike-in controls/features indicative of low quality by using **Calculate maximum from data**. See section 7.2.7 for details.
 - Specified manually by using **Specify maximum**.

7.2.3 Doublets filter

Certain single-cell protocols can assign the same barcode to two or more cells. For example, in droplet-based data, each droplet has a unique barcode, but droplets can contain more than one cell. For data obtained using combinatorial barcoding, there is a chance that two cells will travel together through all barcoding rounds, creating a doublet.

In this dialog of QC for Single Cell, the **Doublets filter** can be enabled and customized to remove the barcodes that are detected as being assigned to two cells. This filter should be skipped for single-cell protocols that do not generate doublets.

Note that QC for Single Cell cannot remove barcodes that are assigned to more than two cells. However, these are expected to be present at negligible rates.

There are two types of doublets:

- homotypic doublets are formed by two cells with similar expression profiles;
- heterotypic doublets are formed by two cells with different expression profiles.

Doublet-removal software, which relies on gene expression to detect doublets, cannot identify homotypic doublets, as their expression profiles are indistinguishable from those of other cells. Alternative approaches are required to detect homotypic doublets, such as cell hashing [Stoeckius et al., 2018] and SNPs in multiplexed samples [Kang et al., 2018].

The **Doublets filter** simulates heterotypic doublets by averaging the expression of two random barcodes that are sufficiently different from each other. These artificial doublets are then used for predicting which of the input barcodes are doublets.

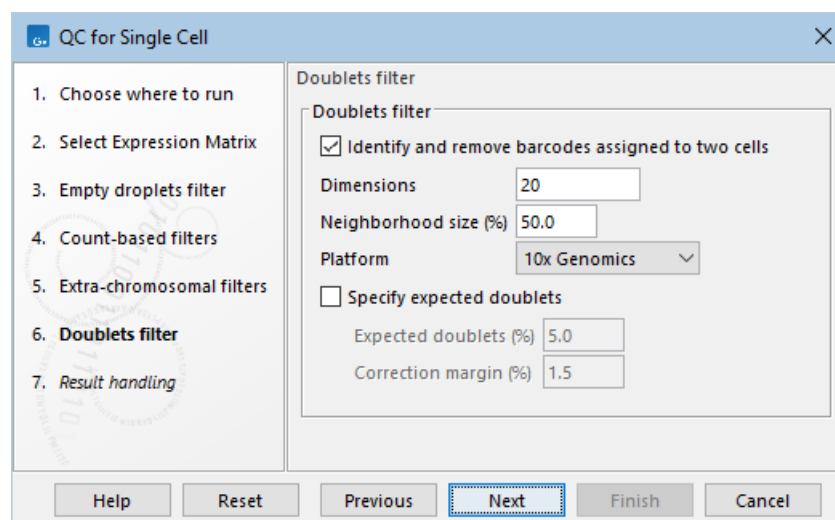


Figure 7.8: The default settings in the Doublets filter dialog.

The following options can be adjusted in the **Doublets filter** dialog (figure 7.8):

- **Identify and remove barcodes assigned to two cells.** Enables filtering of the doublets. This should be unchecked for single-cell protocols that do not generate doublets.
- **Dimensions.** The number of PC dimensions to be used when reducing the dimensions of the expression data.
- **Neighborhood size (%).** Simulated doublets are obtained from barcodes that are not in each other's neighborhood. The size of the neighborhood is specified as % of input barcodes. Note that this is relative to the number of barcodes that pass all previous filters of QC for Single Cell. The optimal neighborhood size is data-set specific and would typically depend on the number of clusters in the data.
- **Platform.** The percentage of barcodes that are expected to be doublets depends on the platform used for generating the single-cell data. Choosing the platform sets the

default values for **Expected doublets (%)** and **Correction margin (%)**, even though **Specify expected doublets** is not checked (see below). The following options are available:

- **10x Genomics.** **Expected doublets (%)** is set to 1% per 1000 captured cells, and **Correction margin (%)** is set to half of this.
 - **Parse Biosciences.** **Expected doublets (%)** is set differently according to the kit size:
 - * 3% for at most 10,000 captured cells,
 - * 5% for at most 100,000 captured cells,
 - * 7% otherwise.**Correction margin (%)** is set to half of the **Expected doublets (%)**.
 - **Other.** No default values are used. It is recommended to specify the expected doublets by checking **Specify expected doublets**.
- **Specify expected doublets.** Check this option to specify approximately how many doublets are present in the data. This option should be used whenever a reasonable expectation is known, as it is very important for an accurate detection of doublets.
 - **Expected doublets (%).** The percentage of barcodes that are expected to be doublets, relative to the number of captured cells.
 - **Correction margin (%).** The percentage of predicted doublets will lie in the interval given by 'Expected doublets (%)' \pm 'Correction margin (%)'.

Note: **Expected doublets (%)** is relative to the number of captured cells, with estimates dependent on the platform:

- **10x Genomics.** The number of barcodes passing the **Empty droplets filter**.
- **Parse Biosciences.** The number of barcodes passing the **Number of reads filter**.




If the above estimates are not appropriate, it is recommended to check **Specify expected doublets** and set the options based on the number of the experiment's target capture cells. For example, if 5000 cells were targeted using 10x Genomics, **Expected doublets (%)** should be set to 5% and **Correction margin (%)** to 2.5%.

The **Doublets filter** receives as input only the high quality cells that pass all filters.

For more details on how doublets are detected, see section [7.2.8](#).

7.2.4 The output of QC for Single Cell

The tool produces the following outputs:

- An **Expression Matrix** ( / ) containing only the barcodes that passed all filters.
- Optionally, a **Cell Annotations** () element containing the different QC metrics used by the filters for the barcodes that passed all filters. Using this Cell Annotations, the barcodes can be colored in a Dimensionality Reduction Plot (see chapter [16](#)) using the QC metrics.

- Optionally, a **Report** (📄), summarizing the filters applied and providing diagnostic plots for each type of filter, as detailed below. The report contains information separately for each sample: summary tables contain one row per sample (figure 7.11), while plots are added per sample.

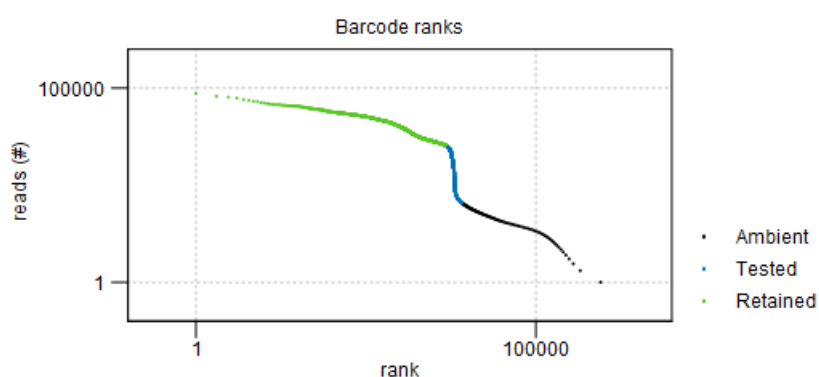
Empty droplets filter

Note that for droplet-based protocols, each droplet is assigned one barcode and these terms can be used interchangeably.

If the **Empty droplets filter** was enabled, the report contains the following information.

The report first shows the barcode rank plot, as seen in figure 7.9.

1 Barcode ranks



1 Barcode ranks

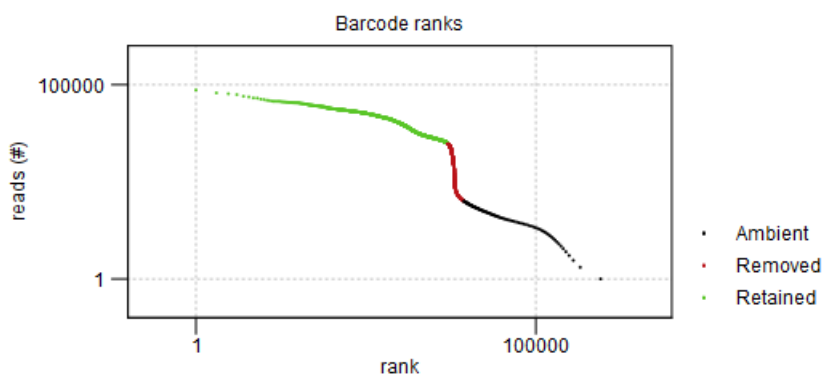


Figure 7.9: Barcode rank plot: log-log plot of the total number of reads for each barcode vs the rank of the barcode, in decreasing order of the number of reads. The barcodes are colored according to whether they are empty droplets containing only ambient RNA (Ambient, black) or retained as cells because they contain a high number of reads (Retained, green). When "Identify cells from the remaining droplets" is checked, remaining barcodes are shown in blue and are tested for being empty droplets (top). Otherwise, these barcodes are shown in red and are removed as empty droplets (bottom).

A summary of the empty droplet filtering and the identified cells is then shown, see figure 7.10 and figure 7.11.

2 Cell calling for droplet data

2.1 Summary

Sample	Renal tumor
Min reads for droplets to be cells	2,972
Max reads for droplets to be ambient*	100
Ambient droplets	1,268,110
Estimated cells	6,651
Droplets with significant FDR p-value	1,510
Reads in cells (%)	85.27
Median reads per cell	4,110.00
Median expressed features per cell	1,518.00

*Droplets with a low number of reads are assumed to be empty and only contain ambient RNA. The ambient droplets are used for estimating an ambient RNA profile.

Figure 7.10: Table summarizing the performed empty droplets filter and identified cells. "Droplets with significant FDR p-value" is reported only when "Identify cells from the remaining droplets" is checked.

2 Cell calling for droplet data

2.1 Summary

Sample	Min reads for droplets to be cells	Max reads for droplets to be ambient*	Ambient droplets	Estimated cells	Droplets with significant FDR p-value	Reads in cells (%)	Median reads per cell	Median expressed features per cell
S1	2,972	100	1,268,110	6,651	1,510	85.27	4,110.00	1,518.00
S2	2,728	100	848,319	6,027	1,451	85.98	3,900.00	1,119.00

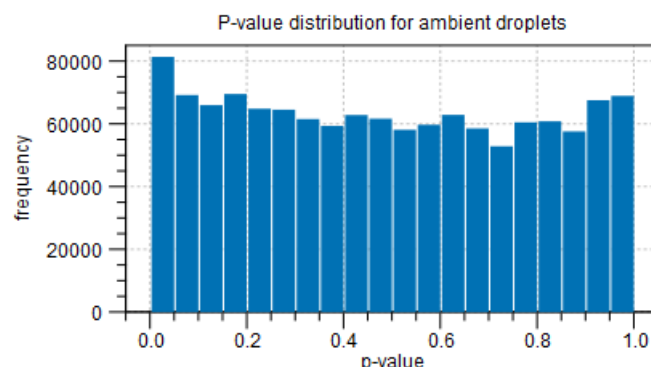
*Droplets with a low number of reads are assumed to be empty and only contain ambient RNA. The ambient droplets are used for estimating an ambient RNA profile.

Figure 7.11: Table summarizing the performed empty droplets filter and identified cells, for input matrix containing two samples.

If any automatic threshold was used (see section 7.2.1), the barcode rank plot and summary table can indicate if this was successful or not. If any of the thresholds are not appropriate, they can be changed as detailed in section 7.2.1.

When **Identify cells from the remaining droplets** is checked, the p-values are simulation-based. The number of simulations to be performed is calculated automatically based on the **FDR threshold**. The report shows the p-value distribution for the ambient droplets. This is expected to be roughly uniformly distributed. Peaks close to 0 indicate that the assumption is invalid and the value for considering barcodes as being empty droplets should be reduced (see section 7.2.1).

2.2 P-value distribution for ambient droplets



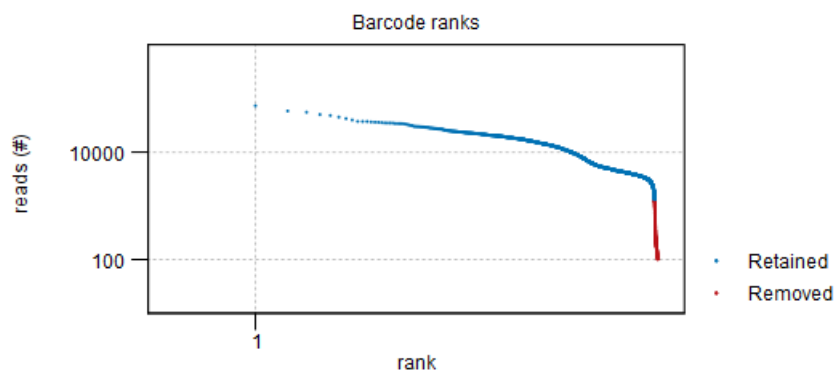
The p-value distribution for ambient droplets should be roughly uniform. Peaks near one can originate from droplets with very few reads and are not a cause for concern. Large peaks near zero indicate that 'Maximum number of reads for droplets to be ambient' should be decreased.

Figure 7.12: Histogram of the p-values calculated for the barcodes from which the ambient RNA profile is built.

Count-based and extra-chromosomal filters

If the **Empty droplets filter** was not enabled, the report first shows the barcode rank plot, as seen in figure 7.13.

1 Barcode ranks



Cells with fewer reads than 1,300 have been removed.

Figure 7.13: Barcode rank plot: log-log plot of the total number of reads for each barcode vs the rank of the barcode, in decreasing order of the number of reads. The barcodes are colored according to whether they are removed (red) or retained (blue), as determined by the number of reads filter.

The report then lists a summary regarding the performed **Count-based filters** and **Extra-chromosomal filters**, as shown in figure 7.14.

Following are histograms of all QC metrics, regardless of whether they have been used for filtering or not. If filtering was enabled, the histograms indicate the threshold used, see figure 7.15. When this threshold is calculated automatically (see section 7.2.2), the histograms can indicate

3 QC filtering

3.1 Summary

Sample	Renal tumor
Input cells	6,651
Known cells*	0
Retained cells†	5,553
Min reads	973
Cells with too few reads	507
Min expressed features	489
Cells with too few expressed features	491
Max reads mapping to features indicative of low quality (%)	11.04
Cells with too many reads mapping to features indicative of low quality	748

*The number of barcodes that are retained due to 'Barcodes to retain'.

†The total number of barcodes that passed QC filtering.

Figure 7.14: Table summarizing the performed count-based and extra-chromosomal filters.

if the threshold is appropriate or not.

3.3 Expressed features

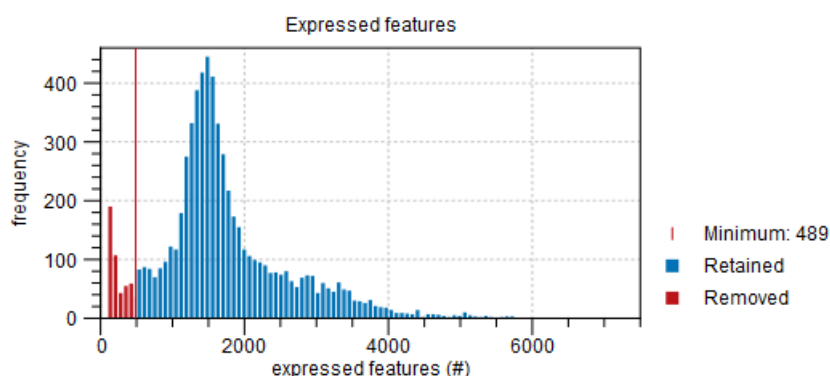
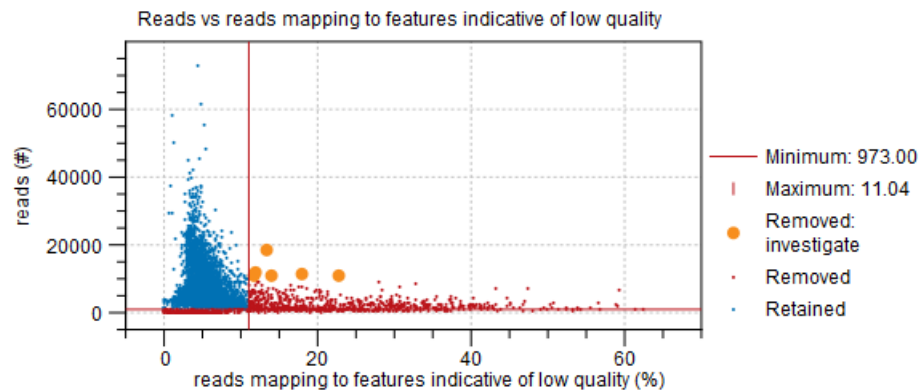


Figure 7.15: Histogram of the number of expressed features for all barcodes.

If the features indicative of low quality filter is enabled, barcodes with too many reads mapped to these features are removed. However, high quality cells can be highly metabolically active, leading to the incorrect removal of barcodes. The report contains plots showing the relations between the percentage of reads mapped to features indicative of low quality and the other QC metrics, where barcodes that might have been incorrectly removed are highlighted (figure 7.16). The highlighted barcodes are identified as having extreme values for the QC metrics, using an automatic threshold calculated in a similar manner to the approach described in section 7.2.7. See section 7.2.5 on how to specify barcodes that should not be removed.

3.6.1 Reads vs features indicative of low quality



3.6.2 Expressed features vs features indicative of low quality

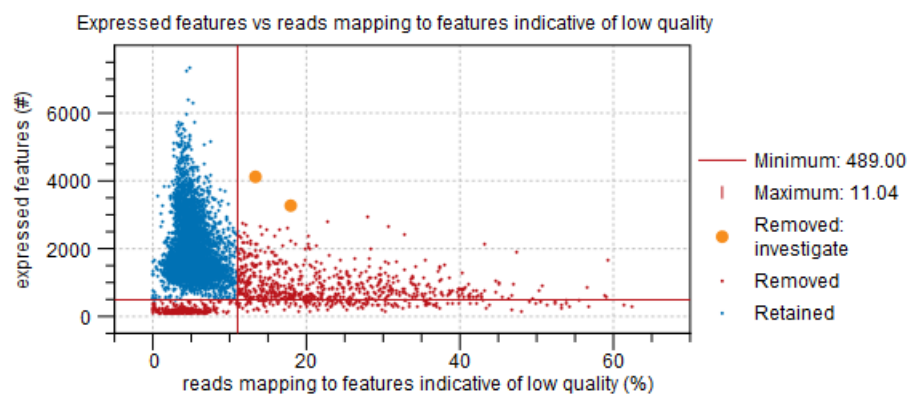


Figure 7.16: The percentage of reads mapped to features indicative of low quality vs the total number of reads (top) and expressed features (bottom). Barcodes in red have been removed and those in blue have been retained. The thresholds for removing barcodes are shown as horizontal and vertical red lines. Barcodes highlighted in orange have been removed, but might correspond to high quality cells that are highly metabolically active that should be retained.

Doublets filter

If the **Doublets filter** was enabled, the report contains the following information:

- A summary regarding the performed filter and the identified cells (figure 7.17).
- A histogram showing the doublet scores (figure 7.18), which can indicate if doublet filtering was successful.
- Relations between the doublet score and number of reads and expressed features (figure 7.19). Typically, barcodes with a high number of reads and/or expressed features are more likely to be removed as doublets.

These diagnostic plots can serve as a guide in adjusting the options for the doublet filter.

4.1 Summary

Sample	Renal tumor
Estimated cells	6,651
Expected doublets (%)	6.65 ± 2.78
Expected doublets	442 ± 185
Identified doublets	367
Max doublet score	0.00
Input cells	5,553
Known cells*	0
Retained cells†	5,186

*The number of barcodes that are retained due to 'Barcodes to retain'.

†The total number of barcodes that passed QC filtering.

Figure 7.17: Table summarizing the performed doublets filter and identified cells.

4.2 Doublet score distribution

- Ideally, there should be a clear distinction between the doublet score distributions from the simulated artificial doublets and the input cells, and the joint distribution should be bimodal.
- Some overlap between the two distributions is expected, originating from input cells that are predicted to be doublets and subsequently removed.

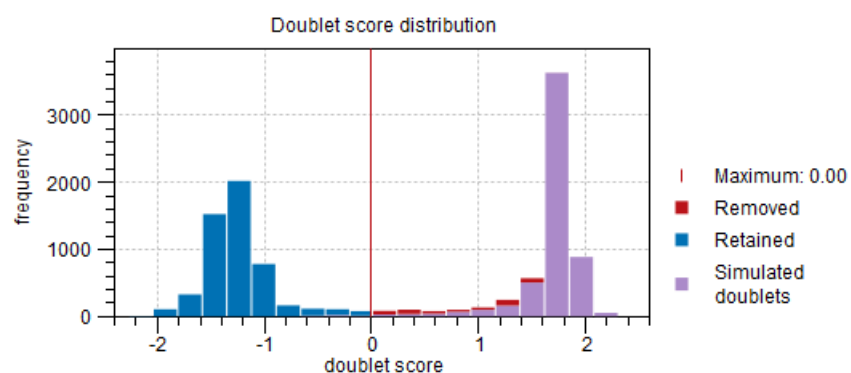
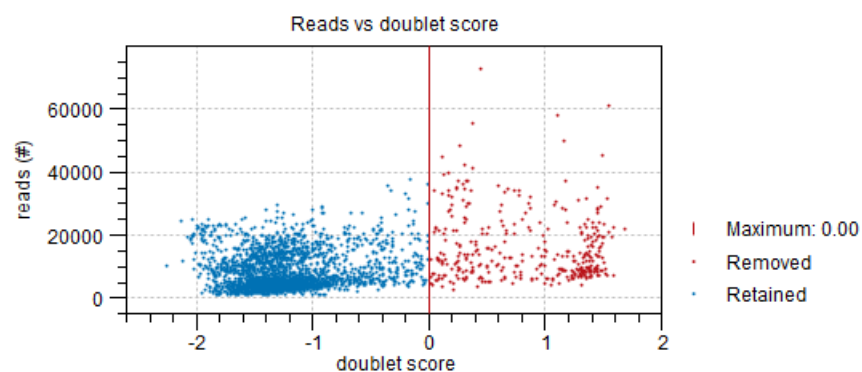


Figure 7.18: Histogram of the doublet score for all barcodes and simulated artificial doublets. The threshold for removing barcodes is shown as a vertical red line.

4.3 Reads vs doublet score



4.4 Expressed features vs doublet score

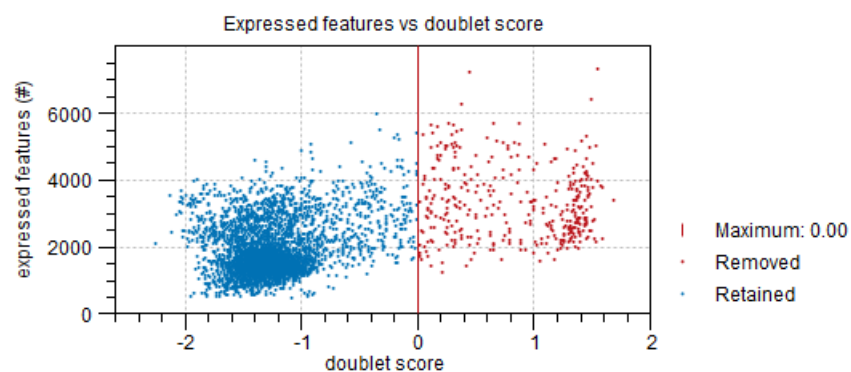

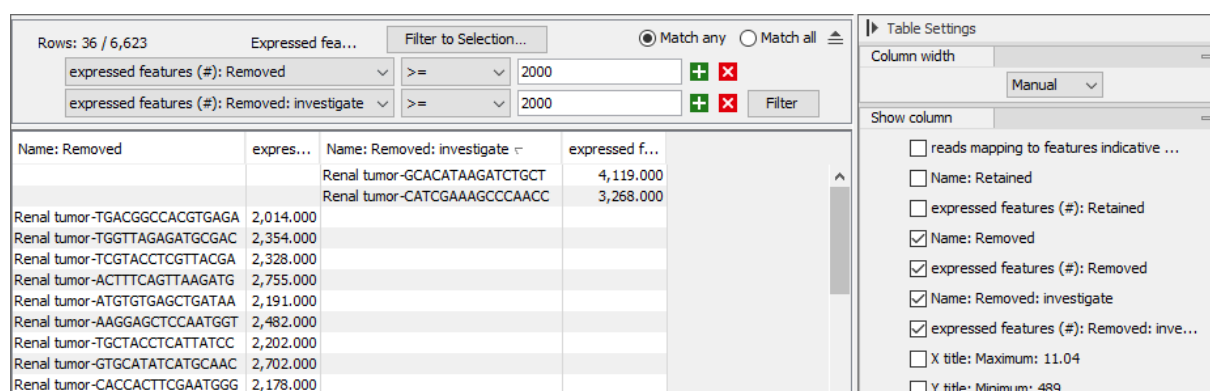


Figure 7.19: The doublet score vs the total number of reads (top) and expressed features (bottom). Barcodes in red have been removed and those in blue have been retained. The threshold for removing barcodes is shown as vertical red lines.

7.2.5 Choosing barcodes to retain

In the dialog for **Count-based filters** (see section 7.2.2), a list of barcodes to be retained as cells can be specified. The desired barcodes can be obtained from the plots in the report:

- Double-click on the desired plot in the report to open it.
- Change to the plot table view ().
- Use standard table filtering tools to list the barcodes of interest (figure 7.20). See https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Filtering_tables.html for details.
- Copy the barcodes and paste them in the **Barcodes to retain** field.



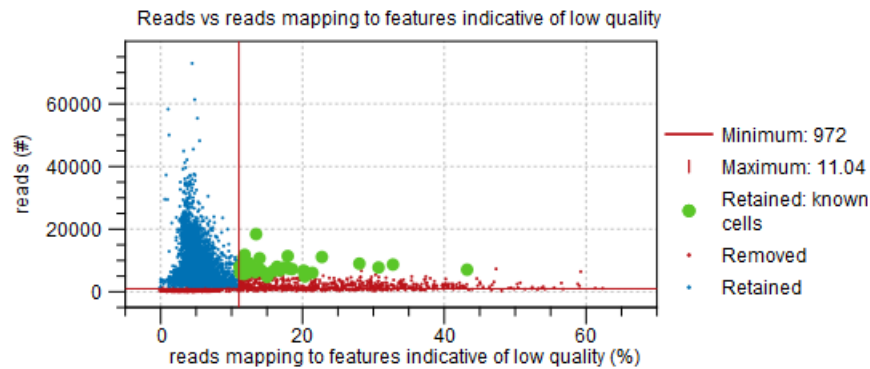
Name: Removed	expressed features (#): Removed	Name: Removed: investigate	expressed features (#): Removed: investigate
Renal tumor-TGACGGCCACGTGAGA	2,014.000	Renal tumor-GCACATAAGATCTGCT	4,119.000
Renal tumor-TGGTTAGAGATGCGAC	2,354.000	Renal tumor-CATCGAAAGCCCAACC	3,268.000
Renal tumor-TCGTACCTCGTTACGA	2,328.000		
Renal tumor-ACTTTCAGTTAAGATG	2,755.000		
Renal tumor-ATGTGTGAGCTGATAA	2,191.000		
Renal tumor-AAGGAGCTCCAATGGT	2,482.000		
Renal tumor-TGCTACCTCATTATCC	2,202.000		
Renal tumor-GTGCATATCATGCAAC	2,702.000		
Renal tumor-CACCACTTCGAATGGG	2,178.000		

Figure 7.20: Table view of the "Expressed features vs features indicative of low quality" plot from figure 7.16. The table is filtered to only show barcodes that have been removed and have at least 2,000 expressed features.

When QC for Single Cell is run with retaining these barcodes, it will produce a report that highlights the retained barcodes as known cells, as shown in figure 7.21.

Note that this option also affects the **Doublets filter**. All barcodes to be retained as cells will be retained regardless of their doublet score. Such barcodes are highlighted in the doublet score relations plots.

3.6.1 Reads vs features indicative of low quality



3.6.2 Expressed features vs features indicative of low quality

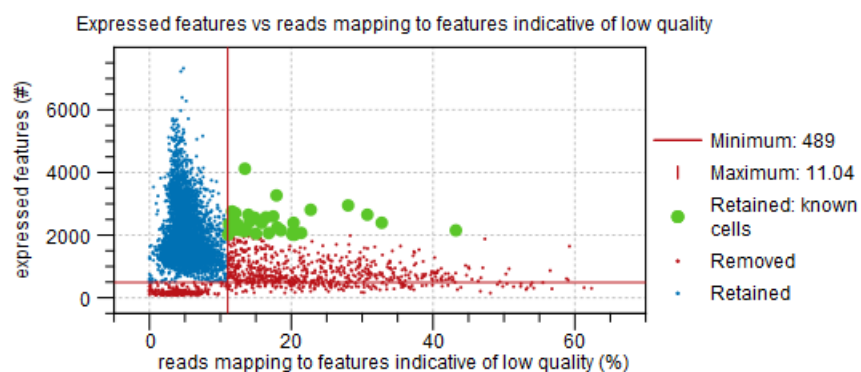


Figure 7.21: The percentage of reads mapped to features indicative of low quality vs the total number of reads (top) and expressed features (bottom). Barcodes in red have been removed and those in blue have been retained. The thresholds for removing barcodes are shown as horizontal and vertical red lines. Barcodes highlighted in green have been retained as specified known cells.

7.2.6 Cell calling

Barcodes with a low number of reads are always removed as ambient droplets. If **Calculate maximum number of reads for droplets to be ambient** is selected (see section 7.2.1), an automatically estimated threshold is used for detecting such barcodes. The threshold is set to 100, or identified from the histogram of number of reads for those droplets that have at most 500 reads, using the Otsu method [Otsu, 1979], whichever is largest. When the threshold is calculated automatically, the following need to be met:

- The minimum number of reads across all droplets is at most 100.
- At least 10% of all droplets have at most 500 reads.
- At least 100 barcodes are identified as ambient droplets.

If any of the above checks are not met, the threshold is set such that no barcodes is an ambient droplet and hence cell calling is not performed.

Barcodes with a high number of reads are always retained as cell-containing droplets. If **Calculate minimum number of reads for droplets to be cells** is selected (see section 7.2.1),

the automatically estimated knee is used for detecting such barcodes. The knee is identified from the smoothed log-log rank data (figure 7.9) where the ambient droplets are removed. An adaptation of the [Satopaa et al., 2011](#) algorithm implemented in <https://github.com/mariolpantunes/ml> is used.

The algorithm for testing if barcodes with an intermediate number of reads are cells is based on EmptyDrops [[Lun et al., 2019](#)]:

- The ambient RNA profile is estimated from the ambient droplets. The expressions from these droplets are added together and a proportion vector for the ambient profile is obtained using the Good Turing algorithm [[Gale and Sampson, 1995](#)].
- Barcodes with an intermediate number of reads are tested for significant deviations from the ambient profile. For each barcode, the probability of obtaining its expression profile from the ambient is calculated. A p-value is obtained from the probabilities of ambient simulated barcodes containing the same total number of reads.
- FDR correction is applied to the p-values for barcodes that are not part of the ambient profile.
- Barcodes with FDR-corrected p-values below the provided value in **FDR threshold** (see section 7.2.1) are retained as non-empty droplets.

7.2.7 Automatic thresholds

Low quality barcodes can be identified using the distributions of the metrics listed below, see section 7.2.2 for more details.

- Total number of reads
- Total number of expressed features
- Percentage of reads mapped to spike-in control regions
- Percentage of reads mapped to features indicative of low quality

When determining an automatic threshold for a metric distribution, the aim is to identify a point within the distribution that separates the low quality barcodes. The following approach is used:

- Compute the median absolute deviation (MAD) threshold: three times the MAD above or below the median value.

This approach is suitable when the data exhibits a normal distribution.

For this, the entire distribution of the metric is used. The total number of reads/expressed features metrics are logarithmically transformed to achieve a normal distribution.

- Check the validity of the MAD threshold by ensuring that:
 - It does not result in the removal of all barcodes.
 - It is greater than a predefined target threshold: 100 for the total number of reads and expressed features and 1% for the percentage of reads mapped to spike-in control regions and features indicative of low quality.

- Use the MAD threshold if valid for all four metrics. Otherwise,
- Use the threshold calculated using the Otsu method [Otsu, 1979].

This approach is suitable when the data exhibits a bimodal distribution. Under the assumption that one of the modes originates from the low quality barcodes, the threshold optimally separates the low quality barcodes.

For this, only barcodes with a moderate number of reads are considered. Barcodes with total number of reads below the ambient threshold or above the knee are excluded. See section 7.2.6 for the calculation of the ambient threshold and knee.

When calculating the threshold for the percentage of reads mapped to spike-in control regions and features indicative of low quality, within the barcodes with a moderate number of reads, only barcodes with moderate percentages (between 5% and 50%) are considered.

7.2.8 Doublet calling

The algorithm for doublet calling contains the following steps.

Doublet simulation

The input expression data is first normalized by using $\log(1 + \text{scaled expression})$. Scaling is performed such that the total expression per barcode is 10000. This normalization procedure is very simple, but sufficient for doublet calling. Note that it is different than the normalization described in section 7.4.

The dimension of the data is then reduced by projecting it into PC space. See section 14.1 for more details. Note that feature selection is not used here.

Heterotypic doublets are afterwards simulated: one doublet is obtained by averaging the expression of two random barcodes that are sufficiently different from each other. For this, a k-nearest neighbor graph is calculated and two barcodes are considered sufficiently different if they are not found within each other's neighborhoods. The value of k is set from 'Neighborhood size (%)'. Note that simulation might fail if this is set too high.

Simulated doublets are normalized and projected into the PC space.

Doublet features calculation

A k-nearest neighbor graph is calculated for all input barcodes and simulated doublets using a pre-defined set of values for k. For each input barcode and simulated doublet, the following doublet features are calculated:

- Is the nearest neighbor a simulated doublet?
- Distance to the nearest neighbor.
- Ratio between the distance to the nearest simulated doublet and nearest input barcode.
- For each value of k, the percentage of neighbors that are simulated doublets.
- For each value of k, the sum of the distances to the neighbors that are simulated doublets, divided by the sum of the distance to all neighbors.

Doublet classification

A Support Vector Machine (SVM) binary classifier is trained using the doublet features from above. Training is performed iteratively:

- In the first iteration, all input barcodes are used in the training data as singlets.
- In the subsequent iterations, input barcodes that are predicted as doublets are removed from the training data.
- The model's performance from each iteration is evaluated by the number of incorrect predictions it makes. There are three kinds of incorrect predictions:
 - how many simulated doublets are predicted as singlets;
 - how many input barcodes used in the training data are predicted as doublets (as these were assumed to be singlets);
 - how many input barcodes not used in the training data are predicted as singlets (as these were assumed to be doublets).
- Training ends when the input barcodes that are predicted as doublets do not change or the performance of the model does not improve after a number of iterations.
- Doublets are finally predicted using the model with the best performance. All input barcodes that are predicted as doublets are removed.



The SVM produces a doublet score where a positive value indicates a doublet. Input barcodes are predicted as doublets as follows:

- 'Specify expected doublets' is unchecked: Input barcodes that have a positive score.
- 'Specify expected doublets' is checked: Input barcodes are sorted according to the score and barcodes with the highest scores are assigned as doublets.
 - During training, 'Expected doublets (%)' determines how many barcodes are predicted as doublets.
 - For the final prediction, a doublet score threshold is calculated such that the number of input barcodes with a doublet score above this threshold falls in the interval given by 'Expected doublets (%)' \pm 'Correction margin (%)' and the threshold is as close to 0 as possible.

7.3 Demultiplex Parse Bio Samples

When using a Parse Biosciences kit, the first step is to distribute the samples into wells, such that each well contains material from only one sample. A well-specific barcode is added. The **Demultiplex Parse Bio Samples** tool assigns the correct sample to the cells, according to the well of origin. The tool is available from::

Tools | Single Cell Analysis  | **Gene Expression**  | **Cell Preparation**  | **Demultiplex Parse Bio Samples** 

It takes an Expression Matrix  /  as input and outputs a copy of the input with updated sample names.

Batches and samples: QC for Single Cell, see section 7.2, runs separately for each sample detected in the input Expression Matrix. This might not be appropriate for Parse Biosciences data, where samples are sequenced together in one batch. For such data, we recommend running QC for Single Cell before updating the sample name.

Defining the samples

The sequenced samples are defined by filling in the sample table at the top of the dialog (figure 7.22).

#	Sample	From well	To well	
1	23/08 d45 E	A1	A2	<input checked="" type="checkbox"/>
2	23/08 d45 F	A3	A4	<input checked="" type="checkbox"/>
3	21/08 d45 E	A5	A6	<input checked="" type="checkbox"/>
4	21/08 d45 F	A7	A7	<input checked="" type="checkbox"/>
5	6/07 d45 C	A8	A9	<input checked="" type="checkbox"/>
6	6/07 d45 D	A10	A10	<input checked="" type="checkbox"/>
7	Test	B2	B5	<input checked="" type="checkbox"/>

Import... Add Remove Clear Select All Deselect All

Display Cell barcodes (#) ▾

	1	2	3	4	5	6	7	8	9	10	11	12
A	11470	12430	16352	13759	17086	17425	18162	17956	18148	18362	18224	18072
B												
C												
D												
E												
F												
G												
H												

Help Reset Previous Next Finish Cancel

Figure 7.22: Demultiplexing the GSM7730635 Parse Biosciences expression matrix from the Gene Expression Omnibus repository. The provided sample loading table was used to populate the dialog. The original sample 7, corresponding to wells A11–A12, was removed, and another sample, named "Test", was added for wells B2–B5. The wells in the plate configuration at the bottom are colored according to the sample they belong to, if the sample is checked in the sample table at the top. The plate configuration quickly reveals that the "Test" sample does not contain any cells, while several cells present in the input matrix will not be part of the output. The well right-click menu from the plate configuration provides options to fill in 'From well' and 'To well' for the selected row of the sample table.

Each row contains information for one sample, and the following is needed:

- The name of the sample, set in the **Sample** column.
- The wells the sample was distributed to. The wells have to be consecutive and are defined by the first and last well, set in the **From well** and **To well** columns, respectively.

The sample table can be filled in from a Parse Biosciences sample loading table by using the **Import...** button.

Alternatively, the table can be filled in manually. More rows can be added by using the **Add** button. The selected row can be deleted by using the **Remove** button. The **Clear** button removes all rows.

Choosing output samples

The check boxes in the sample table's right column can be used to select and deselect samples. Only the selected samples will be added to the output. This can be useful when the input contains unrelated samples that should not be analyzed jointly.

The **Select All** and **Deselect All** can be used to select or deselect all samples, respectively.

Plate configuration

At the bottom of the dialog, the plate with wells A1–H12 is shown. The wells can show different information as selected in **Display**:

- **Cell barcodes #.** The number of cells, if any, from this well that are found in the input matrix.
- **Sample.** The sample number, if any, that this well belongs to.
- **Well.** The number of this well, ranging from 1 to 96. This corresponds to the barcode from the first barcoding round.

The wells are colored based on their sample, if the sample is checked in the sample table. Their border is black if they are not empty, i.e. the input matrix has cells for the well.

If the input matrix is not available, for example in workflows, information about the number of cells will not be displayed.

The well right-click menu provides the following options (figure 7.22):





- **Set in 'From well'.** Sets 'From well' in the selected row of the sample table to the current well.
- **Set in 'To well'.** Sets 'To well' in the selected row of the sample table to the current well.

7.4 Normalize Single Cell Data

The Normalize Single Cell Data tool transforms count data so as to remove the effect of sequencing depth and, optionally, the effect of batch factors. It is recommended to use this tool prior to downstream analysis.

Normalize Single Cell Data is available from:

Tools | Single Cell Analysis  | **Gene Expression**  | **Cell Preparation**  | **Normalize Single Cell Data** 

The tool takes at least one Expression Matrix ( / ) as input, and produces a single Expression Matrix ( / ) as output. If multiple Expression Matrixes are provided as input, the single Expression Matrix output will be filtered to only contain those genes that are present in all of the inputs. A report can optionally also be output.

There are three ways of using the Normalize Single Cell Data tool, which differ in how batch correction is performed:

- **None.** Batch correction is not applied, but count data is transformed so as to remove the effect of sequencing depth. For a new dataset, it is often sensible to first try this setting, and then only apply a batch correction if a batch effect is evident in the Dimensionality Reduction Plot. For more details see section 7.4.1.
- **Each sample is a batch.** Batch correction is performed by choosing one sample as the ‘baseline’. Transformations for each additional sample are applied to make them resemble the baseline. This is appropriate when each sample is expected to have systematic changes in gene expression compared with all other samples, and when these changes are uninteresting for downstream analysis. For example, this setting may be appropriate for combining samples of the same tissue created by different investigators.
- **Using metadata.** A flexible batch correction is applied, where each batch can consist of several inputs (Sample level metadata), or where batches can be specified at the level of individual cells (Cell level metadata).

Sample level metadata can be supplied as a **Metadata table**. For details on how to create a Metadata Table, see <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Metadata.html>. To use sample level metadata, multiple inputs must be provided, because each batch will consist of at least one input.

Batch factors can be supplied in the **Correct for** field. These correspond to columns of a Metadata table. The use of more than one batch factor is not advised as it is easy to over-parameterize the model, see section 7.4.2.

It is also possible to supply **Do not correct for** factors. When these are present, the tool will warn if correcting for the specified batch effect would remove all variation due to these factors in at least one sample (because they are confounded). It will also explicitly model the effect of these factors on expression, which helps to prevent variation due to these factors from being removed by the batch effect correction. This should be regarded as ‘advanced’ functionality because it is easy to over-parameterize the model, see section 7.4.2.

A typical use case for sample level metadata might be when combining samples of the same tissue prepared by different investigators, but where each investigator might have prepared multiple samples. Here it would make sense to ‘Correct for = investigator’. If each investigator prepared a mixture of treated and control samples, then it would make sense to ‘Correct for = investigator’ and ‘Do not correct for = Treatment/Control’.

Cell level metadata Batch factors can also be specified from categories in Cell Clusters and Cell Annotations. Numerical categories of Cell Annotations are not supported, so it is not possible to, for example, regress out ‘Mitochondrial counts (%)’, but this practice is also not advised [Germain et al., 2020]. Multiple inputs of each type are supported, so it is not necessary to ‘combine’ Cell Clusters and Cell Annotations before the tool is run.

It is easiest to explain the batch correction process with an example. If correcting for a cell cycle annotation, possible values might be "G2/M, G1, S". Cells without an annotated cell

cycle state, or with an annotation that is rare (shared by < 20 cells) are given an additional value "Unknown". One of these four cell cycle states (G2/M, G1, S1, Unknown) is then chosen as the baseline, and transformations for each additional value are applied to make the other cell cycle states resemble the baseline.

7.4.1 When is batch correction appropriate?

If in doubt, apply batch correction only when normalization alone proves unsuitable. The suitability of normalization can often be evaluated by looking at how well cells from different samples are mixed within clusters in a Dimensionality Reduction Plot. Two examples follow:

Example 1: figure 7.23 shows clusters colored by sample, where each sample consists of a single cell type. After batch correction using "Each sample is a batch", the cell types are mixed. This is obviously undesired, so batch correction is inappropriate in this case - any effect of batch on expression is confounded with the effect of cell type on expression, and it is not possible to remove one without also removing the other.

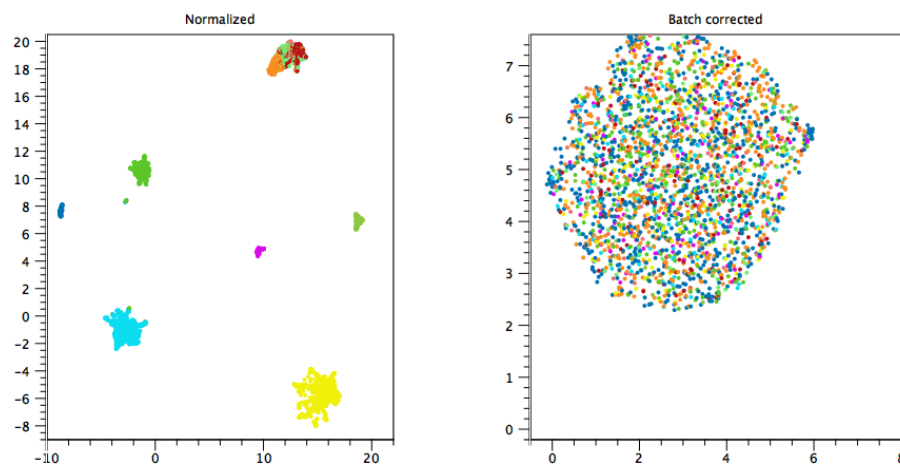





Figure 7.23: An example of when batch correction is undesirable. Each color corresponds to a sample of one cell type. Batch correcting the different samples also removes differences due to cell type, leading to a single cluster.

Example 2: figure 7.24 shows clusters colored by sample. After batch correction using "Each sample is a batch", the clusters are mixed. If the samples described the same tissue type and experimental treatment then we would suspect a batch effect was present, and that batch correction is appropriate. If instead the samples described a difference in experimental treatment, then it would not be possible to determine whether the clusters were separated due to the effect of the treatment, or due to a batch, and deciding whether to apply batch correction would be more difficult.

7.4.2 The output of Normalize Single Cell Data

Normalize Single Cell Data produces the following outputs:

- A single **Expression Matrix** ( / ) named with the extension '(residuals)'.
- A single **Report** () providing diagnostic plots.

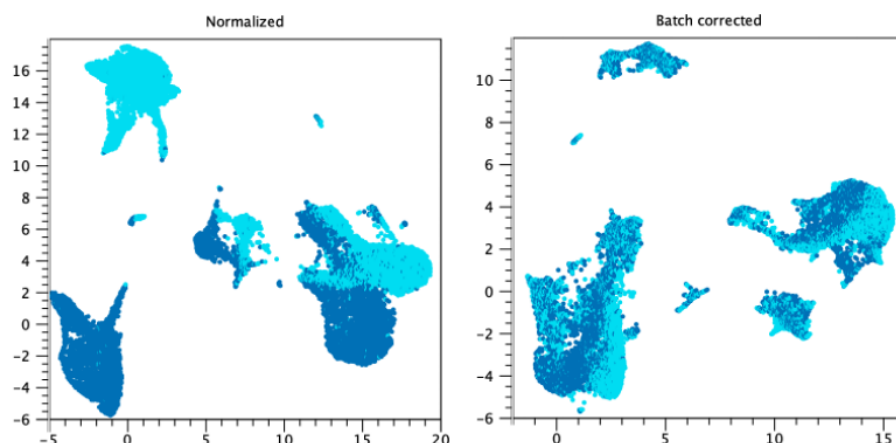


Figure 7.24: An example of when batch correction may be desired. Several clusters can be seen for each of two samples. After batch correction clusters contain a mixture of both samples. Data is from a Seurat tutorial data set (https://satijalab.org/seurat/archive/v3.2/immune_alignment.html).

The Expression Matrix output

It is not possible to see the normalized expressions in the output Expression Matrix - instead the transformation is stored ‘invisibly’, and the output file is typically not much larger (on disk) than the inputs. Tools whose results benefit from normalized data will automatically use the transformation when available. Normalize Single Cell Data ignores any transformations already stored on its inputs, so it is safe to, for example, run the tool on a mixture of samples, some of which have already been normalized, and others of which have not.

The report

Because normalization involves fitting a model to data, bad results can be obtained when the model is inappropriate. This can be diagnosed by the residual variance plot in the report. The variance of the Pearson Residuals is expected to be ~ 1 after normalization for the majority of genes (because the majority are expected to have relatively stable expression across all cells in the data).

There are several ways in which the model can be inappropriate:

The distribution of counts is not negative binomial Usually this is not a problem, as the negative binomial (NB) model is quite flexible. However, the NB model is most appropriate for UMI data. Figure 7.25 shows a dataset where there are no UMIs, and where the sequencing is very deep. Normalization may still be beneficial in this case, but it is worth checking whether the plot is indicating problems with the data.

The normalization is over-parameterized Each batch adds a term in the model that must be estimated from the data. The fewer cells for that batch, the more inaccurate that estimation will be.

The model is under-parameterized Figure 7.26 shows the variance of Pearson Residuals after normalizing a mixture of several protocols (primarily 10X v2, 10X v3, Drop-Seq, Seq-Well and inDrop), but without fitting batch terms. Figure 7.27 shows the variance of Pearson

Residuals for the same data after batch correction. The batch corrected data are more tightly clustered around the expected line.



Figure 7.25: *Residual variance plot for a dataset that is not well modeled by the negative binomial distribution*

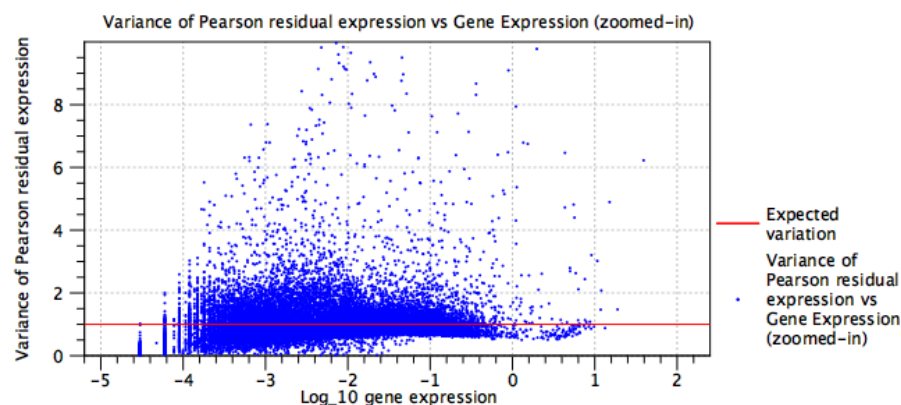


Figure 7.26: *Residual variance plot for an under-parameterized model*

In addition to the residual variance plot, the report contains a list of the most highly variable genes found in the data after correcting for sequencing depth and batch effects. This list should typically be enriched for marker genes of the cell types present in the data. It is sometimes possible to spot problems here. For example, if multiple samples were supplied as input to the tool, and the list was enriched for rRNA genes, then maybe some of the samples had a higher amount of rRNA. Further investigation might then reveal that the samples were prepared by two different individuals, and this might be added as a batch factor.

The remaining sections of the report plot the fitted values of each term in the model (blue points), and, when relevant, ‘regularized’ values (red lines, see section 7.4.3 for details). These allow the number of terms in the model to be seen, which can be useful when evaluating if the model is likely to be over-parameterized. Three terms are always present - the ‘intercept’, the sequencing depth term ‘log10_expression’ and the dispersion term ‘Log10(theta)’. By double-clicking on each plot and switching to the table view, it is also possible to extract all the fitted values. Examples of such plots are shown in figure 7.28 and figure 7.29.

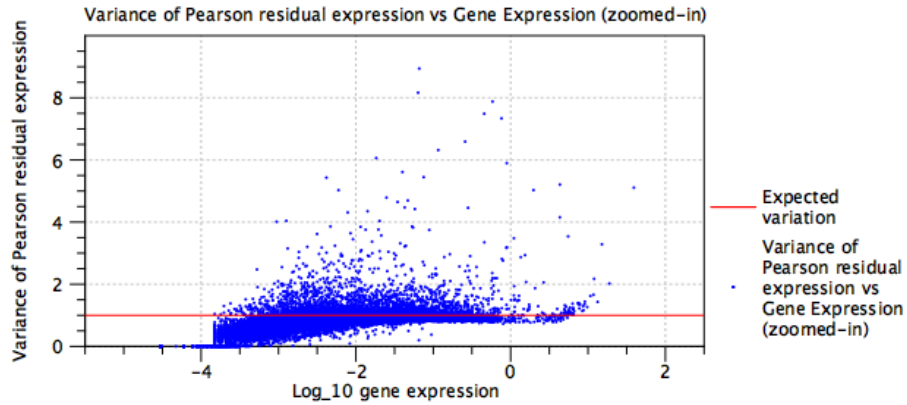


Figure 7.27: Residual variance plot for the same data as in figure 7.26 after adding batch correction terms to the model

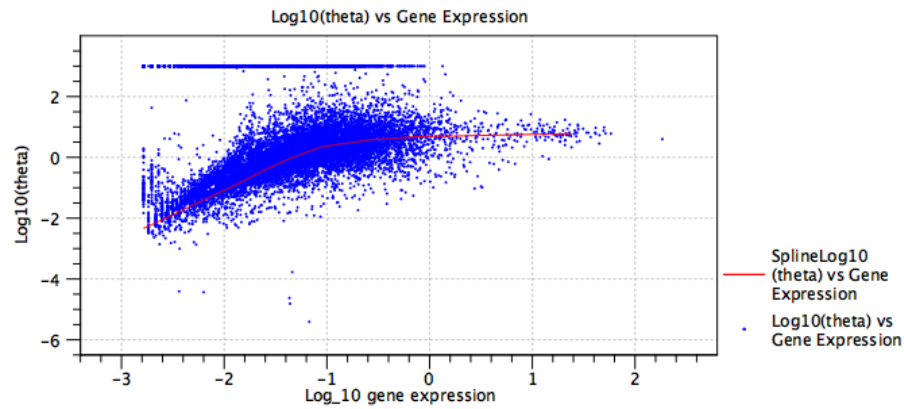


Figure 7.28: A plot of the fitted dispersion parameter, θ , of a model. Each blue point is a gene. The red line is a ‘regularized’ trendline. When the average gene expression is low, the expression of some genes is consistent with a Poisson distribution, which is seen here by the band of genes at $\theta = 10^3$. The trendline provides a more robust estimate of the dispersion, with θ actually decreasing at low expression. See section 7.4.3 for more details.

7.4.3 The Normalize Single Cell Data algorithm

The algorithm is based on **sctransform** [Hafemeister and Satija, 2019]. Briefly, a negative binomial (NB) generalized linear model (GLM) is fit to 2000 genes, uniformly sampled across a range of expressions. The form of the model for each gene is:

$$\log \mathbb{E}(y_i) = \beta_0 + \beta_1 \log_{10} m_i,$$

where y_i are the observed counts for the gene for a cell i that has m_i total counts. The dispersion parameter $\gamma = 1/\theta$ of the NB distribution is estimated during fitting using the Cox-Reid penalized adjusted likelihood [Robinson et al., 2010]. When $\gamma = 0$ ($\theta = \infty$) the NB distribution reduces to the Poisson distribution.

LOWESS regression is then used to estimate the intercept β_0 , the log-sequencing-depth coefficient β_1 , and the dispersion as a function of the average expression. The regression serves as a form of regularization that avoids over-fitting the model, which happens especially for low expression

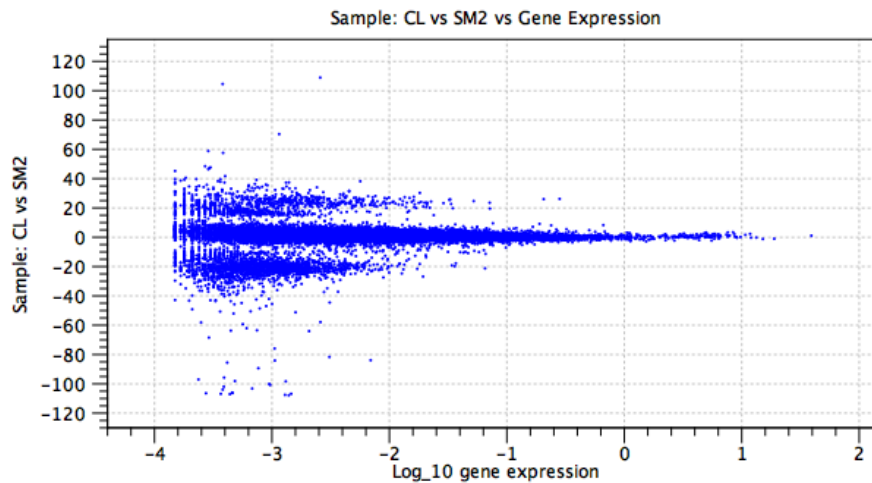


Figure 7.29: A plot of a batch effect parameter. In this example, the y-axis shows the natural logarithm of the fold change for each gene (blue point) in the ‘CL’ sample as compared to the baseline ‘SM2’ sample. There is no ‘regularized’ trendline. The large fold changes concentrated in two bands at $y = 20$ and $y = -20$ are due to genes that are expressed in only one sample, or in neither.

genes.

For batch correction there are some differences from both the above and from `sctransform`:

- A GLM is fit to every gene, because any gene might have a batch effect - though genes with expression < 5 counts are ignored.
- Batch effect terms are added to the model, and these cannot be regularized because each gene might have a batch effect very different from those of genes with similar expression.
- LOWESS regression is only applied to the dispersion, because otherwise there is a mix of regularized and non-regularized terms that cannot be disentangled. The end result is, roughly speaking, that the price to pay for batch correction is a tendency to over-correct data. However, the more data there is, the less this will be a problem, and batch correction is typically performed on large amounts of data.

Normalized/batch corrected values are Pearson Residuals. For each gene, these are defined as follows:


$$\begin{aligned}
 z_i &= \frac{y_i - \exp(\beta_0 + \beta_1 \log_{10} m_i)}{\sigma} \\
 &= \frac{y_i - \hat{y}_i}{\sigma} \\
 &= \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i(1 + \gamma \hat{y}_i)}}
 \end{aligned}$$

Note that Pearson Residuals have several properties that may be unexpected. They are:


- decimals e.g. 123.4 rather than integers e.g. 123

- negative when a gene in a cell has lower expression than predicted by the underlying GLM (though usually not very negative)
- zero for all cells in the unlikely event that expression can be perfectly predicted by the GLM from the provided combination of sequencing depth and batch factors
- only defined in the context of a data set - they cannot be compared across data sets. For example if we have three data sets A, B and C, then running the tool on (A + B) might say that a particular cell + gene in set A has a normalized expression of 100, while running the same tool on (A + C) might say that the same cell + gene has a normalized expression of 0.

7.5 The Expression Matrix element





Expression Matrix () elements are Tracks that hold information about expression of genes or transcripts. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tracks.html> for more information on Tracks.

An Expression Matrix () contains one expression value for each cell and feature.

An Expression Matrix with spliced and unspliced counts () contains multiple expression values:

- The overall expression, which is determined by **Include intronic reads in total expression**, see section 4.5 and section 7.1.
- The amount of spliced and unspliced mRNA.

The spliced and unspliced counts are used by Single Cell Velocity Analysis, see section 10.1.

Both Expression Matrix () / () element types contain table views centered around features () and cells (). Feature expressions for individual cells can be viewed in one of these two tables, described below. See https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Working_with_tables.html for general information on working with tables.

Feature Table

The Feature Table () for an Expression Matrix () / () contains one row for each feature, and has the following columns:

- **Name.** The name of the feature.
- **Id.** A feature identifier. The identifier is either determined upon import, see section 4.5, or from the gene/mRNA track used when running Single Cell RNA-Seq Analysis, see section 7.1.
- **Type.** The feature type.
- **Chromosome.** The chromosome that the feature is on.
- **Region.** The region that the feature spans.

The Feature Table for an Expression Matrix () additionally contains the following columns:

- **Min/Max/Avg row counts.** The minimum/maximum/average feature expression across all cells.
- **Cells.** The number of cells that express the feature.

The Feature Table for an Expression Matrix with spliced and unspliced counts (📊) instead contains the following columns:

- **Min/Max/Avg row counts/spliced/unspliced.** The minimum/maximum/average feature expression across all cells, considering the overall/spliced/unspliced expression.
- **Cells w/ row counts/spliced/unspliced.** The number of cells that express the feature, considering the overall/spliced/unspliced expression.

Clicking on a row opens a separate table, listing the cells expressing the feature (figure 7.30).

To create a new matrix element from a row selection in the Feature Table, use the **Create Matrix from Selection** option in the right-click menu. Cells that express at least one of the selected features are included.

The screenshot shows the 'Feature Table' interface. The top table lists features with columns: Name, Id, Type, Chromosome, Region, and Cells. The row for 'EIF4A1' is selected. Below this, a detailed view for the selected row shows columns: Sample, Barcode, and Raw counts. The 'Feature Table Settings' panel on the right shows column width set to 'Automatic' and a list of columns to show, including Name, Id, Type, Chromosome, Region, Min row counts, Max row counts, Avg row counts, and Cells.

Name	Id	Type	Chromosome	Region	Cells
LOC124903894	124903894	Gene	NC_000017	complement(846485..853020)	1
SLC43A2	124935	Gene	NC_000017	complement(1569254..1630088)	1
OR1D2	4991	Gene	NC_000017	complement(3088484..3104422)	1
OR1P1	8391	Gene	NC_000017	complement(3153890..3154882)	1
RNASEK	440400	Gene	NC_000017	7012624..7014532	1
EIF4A1	1973	Gene	NC_000017	7572825..7579006	1
MPDU1	9526	Gene	NC_000017	7583647..7588212	1
FXR2	9513	Gene	NC_000017	complement(7591230..7614897)	1
LOC124903918	124903918	Gene	NC_000017	8767584..8768927	1

Sample	Barcode	Raw counts
Sample	AAG	1.00

Figure 7.30: Feature Table for an Expression Matrix. One feature expressed by one cell is selected. The table at the bottom shows the cell and its expression level.

Cell Table

The Cell Table (📊) for an Expression Matrix (📊)/ (📊) contains one row for each cell, and has the following columns:

- **Sample.** The sample that the cell is from.
- **Barcode.** The cell barcode.

The Cell Table for an Expression Matrix (📊) additionally contains the following column:

- **Nonzero values.** The number of features expressed by the cell.

The Cell Table for an Expression Matrix with spliced and unspliced counts (📊) instead contains the following columns:

- **Nonzero (Raw counts/Spliced/Unspliced)**. The number of features expressed by the cell, considering the overall/spliced/unspliced expression.

Clicking on a row opens a separate table, listing the features expressed by the cell (figure 7.31).

To create a new matrix element from a row selection in the Cell Table, use the **Create Matrix from Selection** option in the right-click menu.

Rows: 10 Filter to Selection... Filter		
Sample	Barcode	Nonzero values
Sample	AAC	7
Sample	AAG	7
Sample	AAT	5
Sample	ACG	3
Sample	AGG	7
Sample	ATG	6
Sample	ACT	8
Sample	AGT	5
Sample	ATT	8
Sample	CCG	7

Rows: 7 Filter		
Feature	Feature Id	Raw counts
LOC124903894	124903894	1.00
OR1D2	4991	1.00
SNHG29	125144	1.00
GRAPL	400581	1.00
NUFIP2	57532	1.00
PHB1	5245	1.00
OTOP3	347741	1.00

Cell Table Settings

Column width

Automatic

Show column

☒ Sample
☒ Barcode
☒ Nonzero values

Select All

Deselect All

Figure 7.31: Cell Table for an Expression Matrix. One cell expressing seven features is selected. The table at the bottom shows the features and their expression levels.

Chapter 8

Cell Type Classification

Contents

8.1 Browse QIAGEN Cell Ontology	108
8.2 Predict Cell Types	109
8.2.1 The output of Predict Cell Types	111
8.2.2 Cell type refinement	112
8.3 Train Cell Type Classifier	113
8.3.1 Features used for training and prediction	116
8.3.2 The output of Train Cell Type Classifier	117
8.3.3 SVMs for cell type classification	120
8.4 Update Cell Type Classifier	120
8.5 The Cell Type Classifier element	122

8.1 Browse QIAGEN Cell Ontology

CLC Single Cell Analysis Module has a built-in **QIAGEN Cell Ontology** of cell types curated by QIAGEN, associated with

- tissue information;
- synonyms;
- OmicSoft **controlled vocabulary** term;
- CL **Cell Ontology** term;
- parent cell type.

The QIAGEN Cell Ontology can be browsed by using the Browse QIAGEN Cell Ontology tool available from:

Tools | Single Cell Analysis  | **Gene Expression**  | **Cell Type Classification**  | **Browse QIAGEN Cell Ontology** 

The tool launches a wizard from where the content of the ontology can be browsed (see figure 8.1). The desired cell types can be quickly identified by using the search functionality. The displayed cell types can be restricted by using several text fields to only show cell types and their subtypes containing the text in the following properties:

- Cell type: "Cell type", "Synonyms", "OmicSoft" or "CL".
- Tissue: "Tissue".
- Free text: any property of the cell type, including "Definition" and "Comment".

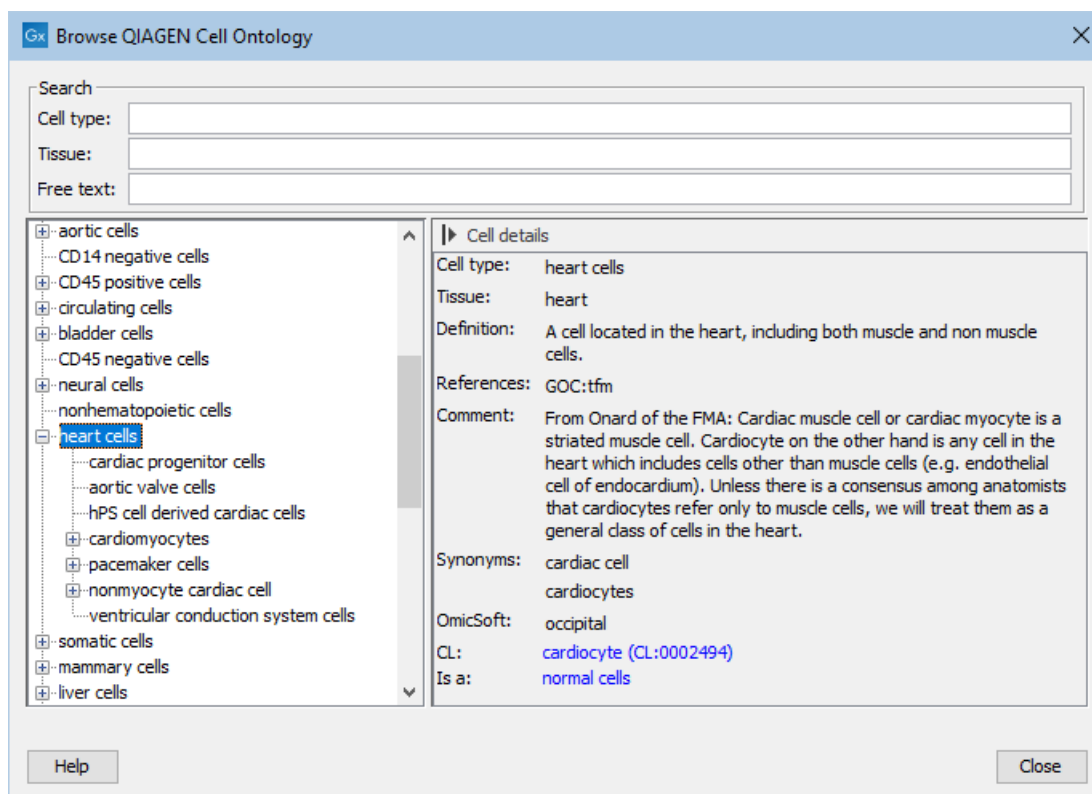


Figure 8.1: View of heart cells in the QIAGEN Cell Ontology.

The pre-trained classifiers available as reference data (see chapter 2) contain only cell types from QIAGEN Cell Ontology.

8.2 Predict Cell Types

The Predict Cell Types tool uses a **Cell Type Classifier** (🧠) to automatically assign cell types to the cells in the Expression Matrix (📊) / (📊) provided as input.

Predict Cell Types is available from:

Tools | **Single Cell Analysis** (🧠) | **Gene Expression** (📊) | **Cell Type Classification** (🧠) | **Predict Cell Types** (🧠)

There are a number of options that can be adjusted (figure 8.2).

Under 'Classifier':

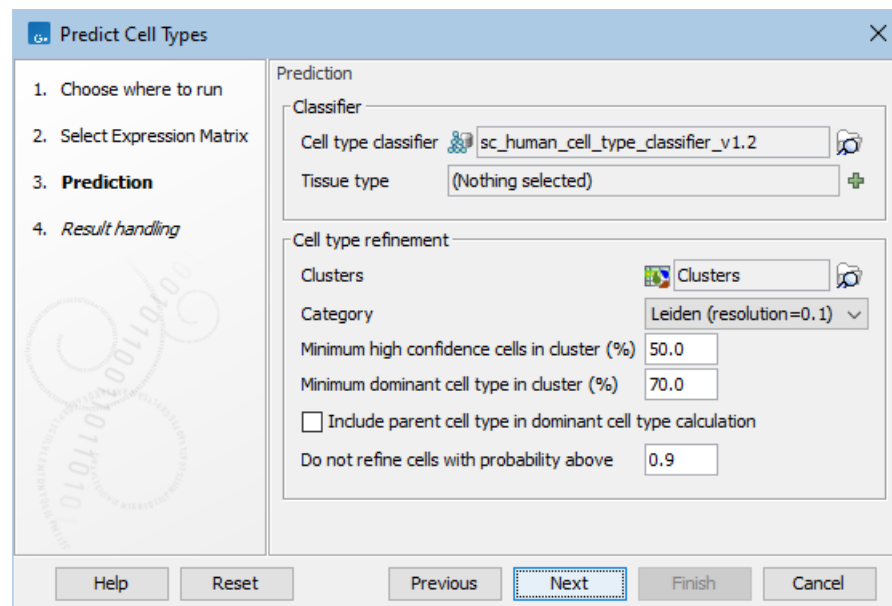


Figure 8.2: The options in the dialog of the Predict Cell Types tool. A Cell Type Classifier for human data downloaded from the Reference Data Manager has been selected.

- **Cell type classifier.** A classifier downloaded from the Reference Data Manager (see chapter 2) or produced by the Train Cell Type Classifier tool (see section 8.3). Note that the features in the input matrix and those used for training the classifier should be matching, see section 8.3.1.
- **Tissue type.** Many cell types from the QIAGEN Cell Ontology (see section 8.1) are associated with specific tissues. When one or more tissues are selected, cell types associated with other tissues will no longer be predicted. For example, hepatocytes are associated with the liver. If **Tissue type = Heart**, then no cells will be predicted as "hepatocytes". A list of cell types that would have been predicted had no tissue type been specified can be found on the History view (📄) of the outputs.

Under 'Cell type refinement', the cell type can optionally be refined (see section 8.2.2):

- **Clusters** (Optional). A Cell Clusters element containing clusters for the input matrix.
- **Category** (Optional). The category from the Cell Clusters element which contains the clusters for cell type refinement.
- **Minimum high confidence cells in cluster (%)** (Optional). Cell type refinement for high confidence cell types (see section 8.2.1) is only performed for clusters where the percentage of cells that have a predicted high confidence cell type is larger than or equal to this.
- **Minimum dominant cell type in cluster (%)** (Optional). Cell type refinement is only performed for clusters where the percentage of cells that have the dominant cell type, i.e. the most frequent cell type, is larger than or equal to this.
- **Include parent cell type in dominant cell type calculation** (Optional). When enabled, cell type refinement for clusters where the percentage of cells that have the dominant cell



type is lower than 'Minimum dominant cell type in cluster (%)', will be performed using the dominant parent cell type from the QIAGEN Cell Ontology, if its percentage is larger than or equal to **Minimum dominant cell type in cluster (%)**.

- **Do not refine cells with probability above.** Cell type refinement is only applied to cells for which the probability of the predicted cell type is equal to or below this. When set to 1.0, cell type refinement will be used for all cells.

Note that for **Tissue type** and **Include parent cell type in dominant cell type calculation** to have an effect, cell types need to be from the QIAGEN Cell Ontology, see section 8.5.

8.2.1 The output of Predict Cell Types

Predict Cell Types outputs:

- A **Cell Clusters**  element containing two categories:
 - "Cell type (all)" contains the predicted cell types, for each cell in the input matrix.
 - "Cell type (high confidence)": if cell type refinement is not performed, it contains the same predicted cell types as "Cell type (all)", but with predictions with probability below 0.5 being replaced with "Unknown". Otherwise, see section 8.2.2. High confidence cell types can be useful for detecting novel cell types.
- Optionally, a **Cell Annotations**  element with the probabilities assigned for a subset of relevant cell types, for each cell in the input matrix. A cell type is considered relevant if:
 - The cell type is the predicted cell type for at least one cell in the "Cell type (all)" category, or
 - There is at least one cell with a probability of at least 0.1 for the cell type.

If cell type refinement is performed, the Cell Annotations element additionally contains the following categories:

- Refined (high confidence): "Yes" if "Cell type (high confidence)" was changed through refinement, "No" otherwise.
- Dominant cell type % (high confidence): The percentage of cells in the cluster with the dominant high confidence cell type before refinement.
- Dominant parent cell type % (high confidence): The percentage of cells in the cluster with the dominant parent cell type as parent or ancestor, with high confidence, before refinement.
- Refined (all): "Yes" if "Cell type (all)" was changed through refinement, "No" otherwise.
- Dominant cell type % (all): The percentage of cells in the cluster with the dominant cell type before refinement.
- Dominant parent cell type % (all): The percentage of cells in the cluster with the dominant parent cell type as parent or ancestor before refinement.

Using the outputs, the cells can be colored in a Dimensionality Reduction Plot (see chapter 16):

- Using the Cell Clusters: by the predicted cell type.
- Using the Cell Annotations: by the probability of having a specific cell type or by whether the initial predicted cell type has been changed due to refinement.

‘Dominant cell type % (high confidence)’ and ‘Dominant cell type % (all)’ can be used for determining a more appropriate ‘Minimum dominant cell type in cluster (%)’.

The predicted cell types can be manually further refined in the Dimensionality Reduction Plot (see section 17.1).

For details on how cell types are predicted, see section 8.3.3.

8.2.2 Cell type refinement

When a Cell Clusters element is provided in **Clusters**, the initial predicted cell types are refined where possible, such that cells in the same cluster either have the same predicted cell type or share a parent cell type:

- Cell type refinement is applied, independently, to both "Cell type (all)" and "Cell type (high confidence)", see section 8.2.1.
- First, a dominant cell type is identified for each cluster:
 - The most frequent cell type, if it is predicted for at least ‘Minimum dominant cell type in cluster (%)’ cells. Otherwise:
 - The most frequent parent cell type that is not "normal cells", if ‘Include parent cell type in dominant cell type calculation’ is enabled and at least ‘Minimum dominant cell type in cluster (%)’ cells have this cell type or are descendants from it.

If no dominant cell type is identified, refinement is not performed for this cluster.

- All cells in the cluster are reassigned to the dominant cell type if:
 - The probability of the initial predicted cell type is at most ‘Do not refine cells with probability above’. And:
 - The initial predicted cell type is not a descendant of the dominant cell type.
- When applying cell type refinement to "Cell type (high confidence)":
 - Cells predicted as "Unknown" (see section 8.2.1) are disregarded when calculating frequencies for the dominant cell type.
 - Refinement is performed only in clusters where the percentage of cells that are not predicted as "Unknown" is equal to or above ‘Minimum high confidence cells in cluster (%)’.

Consider a cluster where the following cell types are predicted:

- 20% of the cells are alpha-beta T lymphocytes,
- 35% of the cells are T lymphocytes,

- 30% of the cells are B lymphocytes, and
- the remaining 15% are other cell types that are not descendants of lymphocytes.

T lymphocytes are the most frequent cell type, but since they constitute of less than 70% (set by 'Minimum dominant cell type in cluster (%)') of the cells, refinement will not be performed.

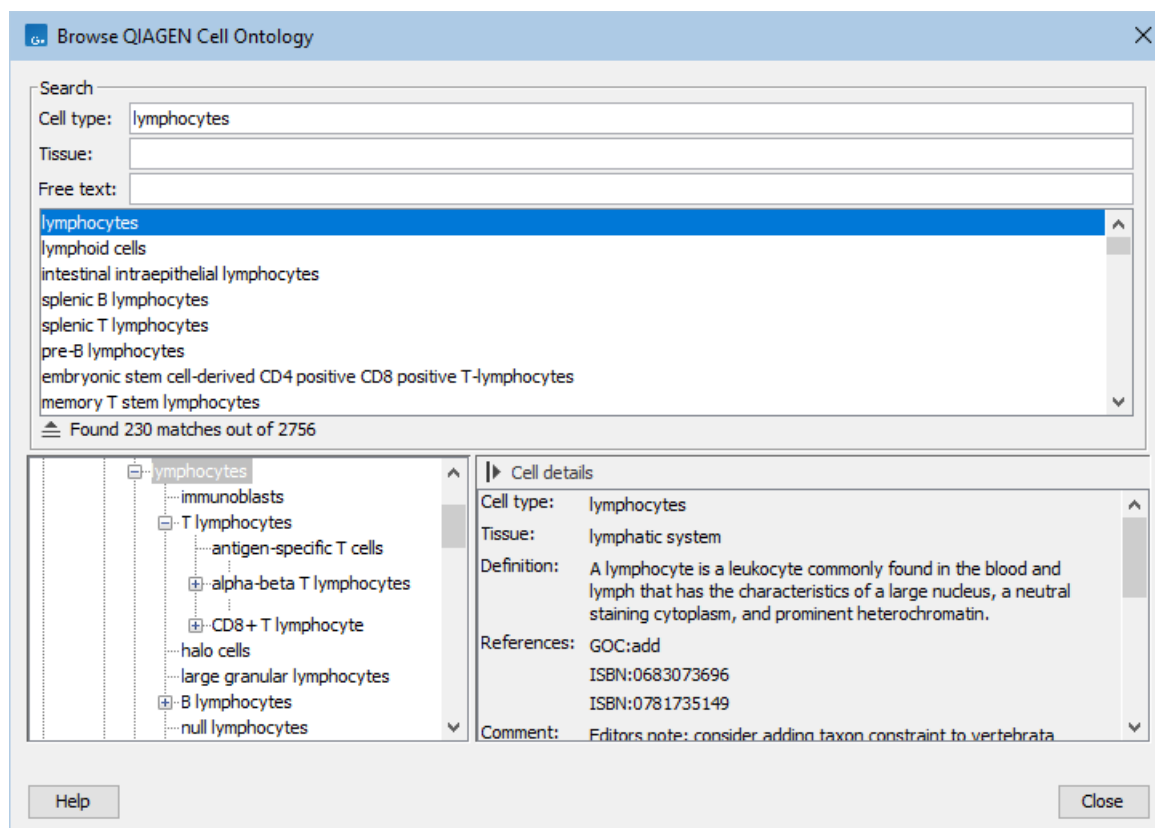


Figure 8.3: View of the QIAGEN Cell Ontology showing lymphocytes, T lymphocytes, alpha-beta T lymphocytes and B lymphocytes.

However, alpha-beta T lymphocytes have T lymphocytes as a parent cell type, and T lymphocytes and B lymphocytes have lymphocytes as a parent cell type (figure 8.3). Using 'Include parent cell type in dominant cell type calculation', the frequency for the parent cell types is calculated:

- 55% of cells are predicted as T lymphocytes or are descendants of T lymphocytes.
- 85% of cells are descendants of lymphocytes.

Lymphocytes represent the dominant cell type since their percentage exceeds 'Minimum dominant cell type in cluster (%)'. The remaining 15% of the cells will therefore be refined as lymphocytes.

8.3 Train Cell Type Classifier

The Train Cell Type Classifier tool trains a Cell Type Classifier which can be used in the Predict Cell Types tool (see section 8.2).

The tool learns to distinguish different cell types by learning specific expression patterns from the expression values of cells that are already assigned a cell type.

Train Cell Type Classifier is available from:

Tools | Single Cell Analysis (📁) | **Gene Expression** (📁) | **Cell Type Classification** (📁) | **Train Cell Type Classifier** (👤)

The tool takes an Expression Matrix (📊) / (📊) as input. There are a number of options that can be adjusted (figure 8.4).

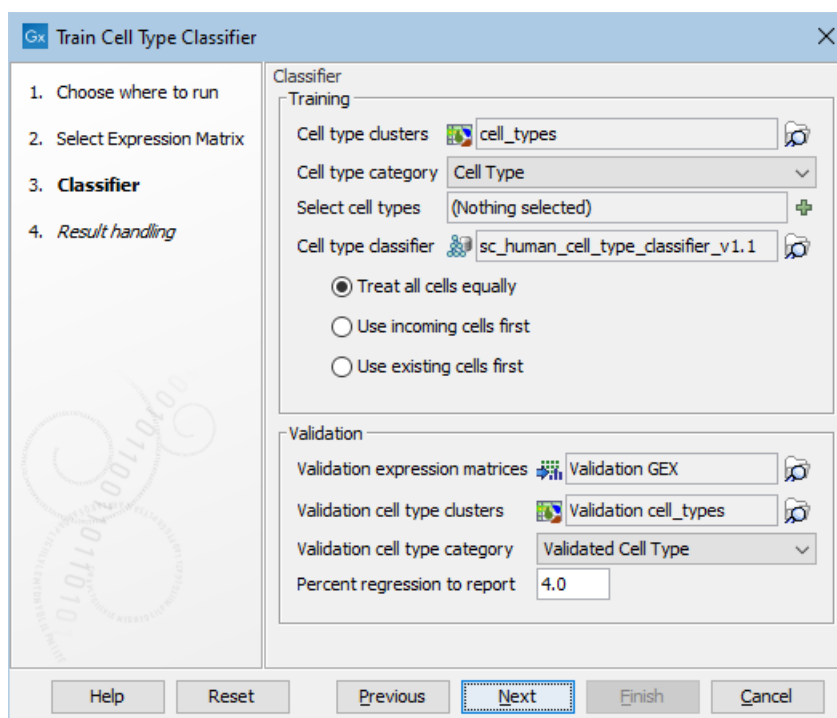


Figure 8.4: The options in the dialog of the Train Cell Type Classifier tool. A Cell Type Classifier for human data downloaded from the Reference Data Manager has been selected.

Under 'Training':

- **Cell type clusters.** A Cell Clusters element containing clusters for the input matrix.
- **Cell type category.** The category from the Cell Clusters element which contains the clusters representing cell types. The tool cannot distinguish the clusters that are true cell types, and therefore this category should only contain clusters that truly represent cell types. It is not required for all cells to belong to a cluster and cells with unknown cell types should be left unannotated, rather than being clustered in an "unknown" cluster.

We recommend using QIAGEN Cell Ontology cell types (see section 8.1). For this, use 'Map clusters to QIAGEN Cell Ontology' when importing clusters (see section 4.3) and use the ontology when defining new clusters in a Dimensionality Reduction Plot (see section 17.1).

- **Select cell types** (Optional). Only train using the selected clusters. If no cell types are selected, all will be used. This can be used to remove undesired "unknown" clusters, or to remove cell types that are found to reduce performance when added to an existing Cell Type Classifier.



Note: In order to keep the running times and the size of the resulting Cell Type Classifier low, the tool uses up to approximately 50 training cells per cell type, which are chosen randomly to include cells from every sample present in the data. If the data contains more than 50 samples, one cell will be chosen randomly from each sample.

- **Cell type classifier** (Optional). A Cell Type Classifier downloaded from the Reference Data Manager (see chapter 2) or produced by this tool. This allows extending existing classifiers with new data. The cells to be used during training can be preferentially chosen from the classifier or the input data as follows:
 - **Treat all cells equally.** The tool will use cells from both the classifier and the input data in a as uniform manner as possible. This is the default option and it ensures that all samples present in both the classifier and input data are represented in the training cells.
 - **Use incoming cells first.** The tool will preferentially use cells from the input data. If there are less than 50 cells for any given cell type, further cells will be chosen from the classifier.
 - **Use existing cells first.** The tool will preferentially use cells from the classifier. If there are less than 50 cells for any given cell type, further cells will be chosen from the input data.

The impact of these options can be investigated in the sample columns of the resulting Cell Type Classifier's table view, see section 8.5.

Note that the features in the input matrix and those used in the classifier should be matching, see section 8.3.1.

Under 'Validation', the trained classifier can optionally be validated:

- **Validation expression matrices** (Optional). One or more Expression Matrices  /  containing cells on which to evaluate the performance of the Cell Type Classifier, and to detect regressions from the existing Cell Type Classifier if present. The tool performs validation only for cells that are not used during training, including the data present in the existing classifier, if provided. For validation it is therefore recommended to use all existing training data that has been previously used for training a classifier.

The report produced can be used to detect inconsistencies in the annotation between the input matrix, the data present in the existing classifier, if used, and the validation matrices.

Note: The features in the input matrix and validation matrices should be matching, see section 8.3.1.

- **Validation cell type clusters** (Optional). One or more Cell Clusters elements containing clusters for the validation matrices. It is not required for there to be a one-to-one correspondence between validation matrices and validation cell type clusters. For example, a single validation Cell Clusters element may contain clusters for some of the cells in several of the validation matrices. Only cells that are present in both a validation matrix and a validation cell type cluster are used for validation.

- **Validation cell type category** (Optional). The category from the Cell Clusters element(s) which contains the clusters representing cell types. The tool cannot distinguish the clusters that are true cell types, and therefore this category should only contain clusters that truly represent cell types. It is not required for all cells to belong to a cluster and cells with unknown cell types should be left unannotated, rather than being clustered in an "unknown" cluster.
- **Percent regression to report** (Optional). Only relevant when a Cell Type Classifier has been provided. If the threshold is $x\%$, then in the report:
 - Cell types with $\geq x\%$ less sensitivity on the validation matrix are colored red.
 - Cell types with $\geq x\%$ more sensitivity on the validation matrix are colored green.
 - Cell types with $\geq x\%$ less sensitivity in any validation matrix are listed in a table.
 - For each validation matrix with $\geq x\%$ less sensitivity, a list of incorrect predictions to cell types that are present in the input matrix is produced. The list is filtered to only contain cell types that are predicted incorrectly with $\geq x\%$ more.

Setting the threshold to 0 will produce a very detailed report of all regressions. It is recommended to use a small non-zero value so that it is easier to spot significant regressions.

8.3.1 Features used for training and prediction

An expression matrix has a set of features associated with it, which is either specified as a gene or transcript track when importing a matrix (see section 4.5), or as a gene track when creating a matrix by mapping reads using the Single Cell RNA-Seq Analysis (see section 7.1).

As feature expression is used for training a classifier and for predicting the cell types of new cells, it is important that the features used for training, validating and predicting are compatible. The two sets of features are mapped against each other to find matching features.

In order to do this, the ids of the features are used. If fewer than 50% of the features are found to be matching, several mappings are created:



- Both feature sets are mapped to three standard gene annotation databases: Ensembl, Entrez and HGNC, and an internal mapping between these databases is used to then match the features from the two sets.
- Features are mapped by name.

The mapping resulting in the largest percentage of matching features from the classifier is used. If this percentage is less than 50%, the tool will fail with a relevant warning message. This means that the two feature sets are incompatible.

Two pre-trained cell type classifiers are available through the Reference Data Manager (see chapter 2), one for human and another one for mouse. These classifiers have been trained on a subset of genes, which are protein coding and are found in both the Ensembl and Entrez gene annotations databases. Therefore, these classifiers should be compatible with most data sets.

8.3.2 The output of Train Cell Type Classifier

Train Cell Type Classifier produces the following outputs:

- A single **Cell Type Classifier**  element, see section 8.5.
- A single **Report**  summarizing the cell types added to the classifier, the performance of the new classifier on validation data (if provided), and any regressions compared to an existing classifier (if provided).

The report has up to 4 sections depending on whether validation data or an existing classifier were provided.

Input data cell types

The input data are the matrix and clusters from which cells are added to the classifier. They are distinct from the validation data. The training data is the subset of the input data that is added to the classifier, and the data already present in the existing Cell Type Classifier, if used.

The first table in this section lists the cell types in the input that have the exact same name as a term in the QIAGEN Cell Ontology. The second table lists the remaining input cell types.

When both tables have entries, it is recommended to check for spelling mistakes or redundancy. For example, in figure 8.5, some cells are annotated by a spelling mistake of "T lymphocytes", and others are annotated as "perithelial cells" - which is a synonym of the term "pericytes" from the first table. The classifier will have attempted to learn all four types separately, which will likely harm performance.

1.1 Cell types in the QIAGEN Cell Ontology

Cell type	Already in classifier?	Cells (#)
pericytes	No	1,902
T lymphocytes	No	834

1.2 Cell types not in the QIAGEN Cell Ontology

Cell type	Already in classifier?	Cells (#)
perithelial cell	No	1,187
T lymphocytes	No	2,631

Figure 8.5: The "Input data cell types" section of the report. In this case, section 1.2 contains cell types that are spelling mistakes and synonyms.

The tables have the following columns:

- **Cell type.** The name of the cell type.
- **Already in classifier?** This column is only shown when an existing classifier is provided and indicates whether the classifier has already been trained with this cell type.
- **Cells (#).** The number of cells in the input with this type. The classifier will be trained with approximately 50 cells per cell type. If the cell type is already in the classifier, as little as one additional cell of this type may be added during training.

The predictions of cell types that are already in the classifier and do not have a very small number of cells (e.g. < 20), are likely to be more accurate than predictions of new cell types with few cells.

Validation data cell types

This section is only present when validation data is supplied to the tool.

Where possible, a performance assessment of the new classifier is made for each cell type in the validation data.

When no assessment is possible, a table lists the affected cell types and the reasons why assessment is not possible. The reasons are:

- **No validation data.** All the cells in the validation data are part of the training data. A performance assessment based on training data would not provide a good estimate of the actual classifier performance or whether a regression occurred.
- **Cell type is not in the classifier.** The new classifier (and therefore also the existing classifier, if supplied) does not contain the cell type. The classifier(s) will never predict this cell type correctly.

The **Performance summary for validation data cell types** table lists the remaining cell types in alphabetical order. Performance is measured based on the classifiers' prediction of the cell type for each cell - no cells are left unlabeled. This corresponds to the "Cell type (all)" category of the Cell Clusters element produced by the Predict Cell Types tool.

Three columns are always present:

- **Cell type.** The name of the cell type.
- **In input data?** Whether the input data contains this cell type.
- **Cells (#).** The number of cells used for validation.

When no existing classifier is provided, the following column is shown:

- **Correct (%).** The overall correct (%) from all validation data.

When an existing classifier is provided, the following additional columns are shown:

- **Regressed matrices (#).** The number of validation matrices where the correct (%) worsened by more than the allowed threshold.

In general, regressions can be explained by misannotations in either the input or validation data. A regression seen in many matrices e.g., "4 (of 5)" is likely to indicate misannotation in the input data. If the regression is reported for "1 (of 1)", it is hard to determine whether the misannotation is in the input or validation data.

A row is added in the subsequent tables for each regressed matrix, together with more information which can help understanding the possible cause of the regression.

- **Change correct (%)**. The difference between **New correct (%)** and **Old correct (%)**.

If the absolute value is larger than the allowed threshold, the corresponding entry in the 'Cell type' column will be colored green if the change is positive (better prediction), or red, if the change is negative (worse prediction).

- **New correct (%)**. The correct (%) for this cell type using the newly trained classifier.
- **Old correct (%)**. The correct (%) for this cell type using the previous classifier.

The correct (%) is calculated as the number of cells that are correctly predicted with the respective cell type out of the the total number of cells that are annotated with the cell type. When multiple validation matrices are used, matrices with more cells will have more influence. This is because each cell is weighted equally. Note that this allows an arbitrary weighting of the validation matrices by choosing subsets of cells in the desired proportions.

Note that large apparent regressions in performance may be spurious if the number of cells in the validation data is very low. For example, if there are 5 cells, a 20% regression indicates that only a single additional cell was predicted incorrectly.

Regressions for cell types not/in input data

These tables are only produced when both validation data and an existing classifier are provided. They list cell types in alphabetical order contain a row for each matrix in the **Regressed matrices (#)** column of the **Performance summary for validation data cell types** table.

For each matrix, the additional % of incorrect predictions is listed, if:

- the % exceeds the allowed threshold;
- the predicted cell type is in the input data.

These are divided into three categories, depending on the relationship between the validation and predicted cell type. Direct relationships describe whether two cell types are more or less specific descriptions of the same type. They are found by mapping the two cell types to the QIAGEN Cell Ontology via a list of known synonyms.

- **Incorrect**. No direct relationship can be determined. Either the mapping has been successful and the cell types are not directly related, or the mapping is not possible.
It may be that an "Incorrect" cell type prediction is acceptable. For example, the validation cell type may differentiate into the predicted one, or the two types may be subtypes of the same cell type. Sometimes both explanations are possible, for example "mature B lymphocytes" and "plasma cells" are both subtypes of "B lymphocytes", and mature B lymphocytes differentiate into plasma cells.
- **Less specific**. The predicted cell type may be technically correct, but it is less specific than the validation data cell type e.g., the predicted type is "B lymphocyte", but the validation type is "mature B lymphocyte".
- **More specific**. The predicted cell type is a more specific type than the validation data cell type e.g., the predicted type is "mature B lymphocyte", but the validation type is "B lymphocyte".

Ideally, no cell types should be listed in the **Less specific** and **More specific** categories for the **Regressions for cell types in input data** table. This is because the input data includes both the validation and predicted cell type, so the use of the validation cell type instead of the predicted cell type was deliberate. Such cases always merit investigation.

The presence of cell types in the **Less specific** category is always a cause for concern. It suggests that the newly trained classifier has lost some of the existing classifier's ability to predict cells specifically.

The presence of cell types in the **More specific** category can be benign. It suggests that the newly trained classifier has gained the ability to predict cells specifically. Care should be taken to ensure that this explanation is plausible, for example, perhaps the more specific cell type has just been added to the classifier and/or was absent in the matrix for which the regression occurred. If the more specific cell type is localized to a particular tissue e.g. "ovarian vascular surface endothelial cells" instead of "endothelial cells", then it can be checked whether the validation matrix is expected to include cells from that tissue and whether the classifier contains a more appropriate cell type that was not predicted.

8.3.3 SVMs for cell type classification

The algorithm for cell type classification uses Support Vector Machines (SVMs). Independent benchmarking [Abdelaal et al., 2019] has shown that SVM classifiers can outperform most other types of classifiers for cell type prediction. The implementation uses liblinear (<https://github.com/bwaldvogel/liblinear-java>) with a series of additions:

- Feature expression is log transformed and scaled using the maximum observed expression, such that values are placed in the interval $[-1, 1]$.
- Weights are used to balance the size of the training classes.
- Platt scaling is applied on the decision values to obtain probabilities [Platt et al., 1999]. Note that probabilities are not normalized: the sum of the probabilities of all cell types for one cell does not necessarily sum up to one.
- For each cell, the most likely cell type consistent with the chosen tissue(s) is assigned from the Platt probabilities. Cells are labeled as unknown (see section 8.2) when the most likely cell type has a Platt probability below 0.5.

Note that the algorithm uses the raw, not normalized, expression values.

8.4 Update Cell Type Classifier

The Update Cell Type Classifier tool takes one **Cell Type Classifier** (🧬) element as input and outputs a new **Cell Type Classifier** (🧬) element containing updated training data.

Update Cell Type Classifier is available from:

Tools | Single Cell Analysis (🧬) | **Gene Expression** (🧬) | **Cell Type Classification** (🧬) | **Update Cell Type Classifier** (🧬)

In the first wizard step 'Update classifier', the following options can be adjusted:

- **Remove cell types.** The cells annotated with the selected cell types are removed from the training data. The classifier will be re-trained.
- **Remove samples.** The cells from the selected samples are removed from the training data. Cell types for which all training cells are from the removed samples will be subsequently removed. The classifier will be re-trained.
- **Rename cell types.** One or more cell types are renamed. The **Add** button can be used to add additional cell types to be renamed.
- **Map cell types to QIAGEN Cell Ontology.** When this is checked, cell types will be translated, if possible, to the QIAGEN Cell Ontology (see section 8.1). The translation attempts to match each cell type with a QIAGEN cell type based on the name and known synonyms. For example, ‘alveolar epithelial cells’ are also called ‘pneumocytes’. If this option is selected, the ‘alveolar epithelial cells’ cell type, if present, will be named ‘pneumocytes’. This option can be useful when standardizing cell types from different sources.

Cell types with the same name will be merged into one and the classifier will be re-trained. Cell types are merged when:

- A cell type is renamed to an existing name.
- Two cell types are renamed to the same name.
- Two cell type names are synonyms for the same QIAGEN cell type and cell types are mapped to the ontology.


In the next wizard step ‘Validation’, the tool can be configured to validate the resulting classifier, as done for the Train Cell Type Classifier tool (see section 8.3). Regressions are calculated using the input classifier.

The options above can be mixed and matched to obtain the desired output. For example, a cell type can be first renamed and then the new name can be mapped to the ontology.

Renaming and/or mapping cell types to the ontology does not require updates to the underlying classification model. However, the classifier will be re-trained when the training data is changed:

- Cells are removed, by removing cell types and/or samples.
- Cell types are merged. Note that the classifier will be trained with approximately 50 cells per cell type, see section 8.3. When cell types are merged, some cells might be discarded during training.

The Update Cell Type Classifier tool produces a report summarizing the performed updates and, if validation data was provided, the performance of the updated classifier as described in section 8.3.2.

The Update Cell Type Classifier tool can also be started from the table view of the **Cell Type Classifier**  element (see section 8.5), where the dialog is automatically filled in with relevant selections:

- The selected cell types can be removed or renamed.
- The sample over which the mouse hovers can be removed.

8.5 The Cell Type Classifier element

The table view of the Cell Type Classifier gives a summary of (figure 8.6):

- the cell types the classifier has been trained on;
- the type of cell type: "QIAGEN cell type" if it is from the QIAGEN Cell Ontology, "Other" otherwise;
- which features the classifier uses to distinguish the cell types;
- how many cells the classifier has been trained on;
- how many and which samples the classifier has been trained on.

Cell type	Type	Top features supporting this cell type	Number of cells	Number of samples	HCL-A...	HCL-A...
fasciculata cells	QIAGEN cell type	STAR, CYP17A1, FDX1, HSD3B2,	50	17		5
fibroblasts	QIAGEN cell type	DCN, GSN, APOD, IGFBP6, TIMP3,	72	72	1	1
gastric chief cells	QIAGEN cell type	PGC, PGA5, LIPF, PGA3, CXCL17,	50	2		
goblet cells	QIAGEN cell type	AGR2, TFF2, TFF1, SPINK1, CYSTM1,	50	21		
inflammatory	QIAGEN cell type	FABP4, THRSP, GOS2, RBP4, DGAT2,	50	1		
intercalated cells	QIAGEN cell type	TMEM213, ATP5V0D2, MMP9, ACP5,	50	13		1
M2 macrophages	QIAGEN cell type	APOC1, CTSD, CSTB, LGALS3, LYZ,	50	24		3
macrophages	QIAGEN cell type	C1QB, RNASE1, CCL3L3, SPP1, FTL,	95	95	1	1
mast cell	Other	TPSB2, TPSAB1, CPA3, HPGDS, SRGN,	56	56	1	
mesenchymal stem	QIAGEN cell type	COL1A2, COL1A1, LGALS1, COL3A1,	52	52		
mesothelial cells	QIAGEN cell type	ITLN1, PRG4, HP, C3, OGN, CPB1,	50	5		
monocytes	QIAGEN cell type	CTSS, SAT1, FCN1, HCST, FCER1G,	91	91	1	1
myeloid cells	QIAGEN cell type	RARRES2, CSTB, RPS27, MRPL33,	50	17		13
neurons	QIAGEN cell type	STMN2, TUBA1A, S100B, CHGB,	50	47		1
neutrophils	QIAGEN cell type	MPO, S100A8, SRGN, S100A9, LTF,	74	74	1	1
oligodendrocytes	QIAGEN cell type	PLP1, PTGDS, TF, SCD, CRYAB, CNP,	50	2		
plasma cells	QIAGEN cell type	JCHAIN, MZB1, SSR4, IGLL5,	59	59	1	1
pre-Sertoli cells	QIAGEN cell type	BEK1, AMH, ZFAND5, PRDX6, FSCN1,	50	4		
primordial germ cells	QIAGEN cell type	RPL7, MSMB, HSP90AA1, RPL37,	50	49		
progenitor intestinal	QIAGEN cell type	OLFM4, FXYD3, PYY, EPCAM, BIRC3,	50	26		

Figure 8.6: The table view of a Cell Type Classifier trained on HCL data (<http://bis.zju.edu.cn/HCL>) containing 106 different samples. For cell types present in more than 50 samples, one cell is chosen from each sample. The sample columns (here, starting with "HCL-") contain the number of training cells used from the respective sample. The "Top features" columns list the most important features used by the classifier to distinguish each cell type from the rest.

Cell types that are in the QIAGEN Cell Ontology (see section 8.1) are clickable and the links open the ontology browser with the corresponding cell type selected. In figure 8.6, 'mast cell' is missing a link because this cell type is named 'mast cells' in the ontology. For ensuring that the link between cell types and the ontology is present where possible, use Update Cell Type Classifier with 'Map clusters to QIAGEN Cell Ontology', see section 8.4. Note that this tool can be started from the table view's right-click menus.

The classifier assigns weights to each feature according to how informative it is for distinguishing each cell type from the rest. The "Top features supporting this cell type" and "Top features supporting another cell type" list up to 10 features with the largest weights. If a cell has high expression for the features in "Top features supporting this cell type", it is a good indication that it is of that specific cell type, while if it has high expression for the features in "Top features supporting another cell type", it is a good indication that it is not of that specific cell type. Note that the classifier uses more information than that summarized in these two columns, and the combined expression for all features together with the assigned weights is used for predicting cell types.

If the top features are assigned ids from either Ensembl or Entrez, the feature names are clickable and the link opens the corresponding Ensembl or Entrez webpage.

Top features and markers. The top features identified by the classifier are different than the markers identified by the Differential Expression for Single Cell tool (see section 9.1). A cell type marker has different expression in the cell type compared to all other cell types, and this is calculated independently for each feature. The classifier top features are useful jointly in recognizing a specific cell type, but might not necessarily be very informative on their own.

Let us consider the following cell types A-D with the given average expression for features X-Z. The cell types might then have the listed top features and markers:

	A	B	C	D
X	4	4	0	8
Y	2	4	2	6
Z	0	0	2	4
Top features supporting this cell type	X	X, Y	Y, Z	X, Y, Z
Top features supporting another cell type	Y, Z	Z	X	-
Markers	-	Y	X, Z	X, Y, Z





Chapter 9

Expression Analysis

Contents


9.1 Differential Expression for Single Cell	124
9.1.1 The output of Differential Expression for Single Cell	127
9.1.2 The differential expression algorithm	128
9.2 Create Expression Plot	128
9.2.1 The Heat Map output of Create Expression Plot	131
9.2.2 The Dot Plot output of Create Expression Plot	132
9.2.3 The Violin Plot output of Create Expression Plot	133

9.1 Differential Expression for Single Cell



Differential Expression for Single Cell detects differentially expressed features using expressions from an input **Expression Matrix** () / () and groupings provided by **Cell Clusters** () or **Cell Annotations** ().

It is often most natural to run the tool from a Dimensionality Reduction Plot by right-clicking on the plot, see section 17 for details. However, it can also be found under the Tools menu at:

Tools | Single Cell Analysis () | **Gene Expression** () | **Expression Analysis** () | **Differential Expression for Single Cell** ()

The tool tests if each feature is differentially expressed and outputs Statistical Comparison Tables ().

The first set of options narrow down the focus of the tool:

- **Clusters** and **Cell annotations**. At least one of these must be supplied. **Clusters** accepts **Cell Clusters** () and **Cell annotations** accepts **Cell Annotations** (.
- **Test differential expression due to** a single category from the supplied Cell Clusters or Cell Annotations. Categories that only contain true/false values or numerical data are not supported. Tests will be performed between the groups of cells with different labels in this category.

- **Select groups** (Optional). This can be supplied to reduce the number of groups of cells considered or to control the order in which comparisons are made.

It is easiest to understand the effects of these settings with example data from figure 9.1. If the table shown there were supplied as either 'Clusters' or 'Cell annotations', then the possible values of 'Test differential expression due to' would be 'Sample', 'Status' or 'Cell type' ('Barcode' is special and is excluded). If 'Cell type' were chosen, then possible groups in 'Select groups' would be 'T cell', 'B cell' and 'Platelet'.

Sample	Barcode	Status	Cell type
demo	AAAA	Case	T cell
demo	AAGA	Case	B cell
demo	AACA	Case	Platelet
demo	AATA	Control	T cell
demo	AATT	Control	B cell
demo	AAGG	Control	Platelet

Figure 9.1: Example data consisting of cells with different cell types coming from either Case or Control samples

From now on, we will continue with this example, assuming that **Test differential expression due to = Cell type**. There are two possible types of tests: 'All group pairs' and 'Identify marker genes'.

All group pairs

In the example, there are three groups: 'T cell', 'B cell' and 'Platelet'. When **All group pairs** is selected, up to 6 pairwise comparisons can be performed. Only three of these will be output, for example 'T cell vs B cell', 'T cell vs Platelet', and 'B cell vs Platelet'. The other three tests, 'B cell vs T cell', 'Platelet vs T cell', and 'Platelet vs B cell' will not be produced - this is because the only difference between, for example, 'T cell vs B cell' and 'B cell vs T cell' is the sign of the fold change.

It is possible to control exactly which comparisons are performed by using the **Select groups** option. The order of any selected groups determines the direction of the comparisons. For example, if **Select groups = Platelet, B cell, T cell**, then the comparisons will be 'Platelet vs B cell', 'Platelet vs T cell', and 'B cell vs T cell'. If **Select groups = T cell, B cell, Platelet**, then the comparisons will be 'T cell vs B cell', 'T cell vs Platelet', and 'B cell vs Platelet'.

The **Select groups** option can also be used to restrict the number of comparisons. If **Select groups = B cell, Platelet**, then the outputs will be reduced to just those involving the selected groups. In this case there would only be one output: 'B cell vs Platelet'.

Identify marker genes

In the CLC Single Cell Analysis Module, marker genes are considered to be genes that are differentially expressed in the group of interest when compared to all other groups. This does not necessarily mean that they are only expressed in the group of interest, or are up-regulated in the group of interest; marker genes may also have abnormally low expression (though this is unlikely), or have an expression that, by being lower than in some groups and higher than in others, is distinctive to the group of interest.

In practice, the requirement that marker genes are differentially expressed compared to *all* other groups can be overly strict. For example, a group might contain so few cells that it is never possible to detect differential expression compared to this group. To avoid this problem, groups are excluded if they have no significant differentially expressed genes relative to a majority of the other groups. Here, significant means that the FDR p-value is less than 0.05.

Select groups determines the groups for which the markers have to be differentially expressed. For example if **Select groups = Platelet, B cell, T cell** then three sets of markers will be output 'Platelet vs rest', 'B cell vs rest' and 'T cell vs rest'. The markers for 'Platelet vs rest' will only be differentially expressed when compared to B cells or T cells - if there was another cell type in the data that had been excluded from the selected groups, then it is possible that the markers in 'Platelet vs rest' would not be useful for distinguishing platelets from this additional cell type.

Marker genes are identified by first running 'All group pairs' and collecting the pairwise results into marker results as detailed above.

Performing separate tests between conditions for each cell type

It is possible to make comparisons between conditions (e.g. Case vs Control) for each cell type using the option **Perform a separate test for each group in**. Again this is easiest to illustrate with reference to figure 9.1.

Using 'All group pairs' with **Test differential expression due to = Status** and **Perform a separate test for each group in = Cell type** will give the outputs 'T cell: Case vs Control', 'B cell: Case vs Control', and 'Platelet: Case vs Control'.

Selecting genes to be tested


When a gene is expressed in too few cells, there could be too little information to reliably detect if it is differentially expressed. A minimum number of cells expressing the gene can be set using **Minimum number of cells** and **Minimum percentage of cells**. A gene is considered to have insufficient expression in a group if one of the following is true:

- the number of cells expressing the gene is less than **Minimum number of cells**;
- the percentage of cells expressing the gene is less than **Minimum percentage of cells**.

When a pairwise comparison is performed, tests are not performed for genes with insufficient expression in both groups, and the p-value is set to NaN (not a number).

This also affects 'Identify marker genes', as the markers are obtained from pairwise tests. For markers, the test for a gene is not performed when the group of interest and at least one other group have insufficient expression.

9.1.1 The output of Differential Expression for Single Cell

Differential Expression for Single Cell produces one or more Statistical Comparison Tables ()

Differentially expressed genes and clustering. Groups are often defined based on clusters found using a clustering algorithm. Because clustering and differential expression analysis are performed on the same data, they are not independent. This means that, even for simulated data generated from the same distribution, random differences in expression between genes may drive the formation of clusters, and these same genes will then be found to be differentially expressed between the clusters. One remedy for this is to perform clustering on half the data and differential expression on the other half. However, it is more common to simply be cautious about over-interpreting results.

A similar warning can be made for groups defined based on cell types predicted by Predict Cell Types - the tool works by learning the expression pattern of different genes in different cell types. Therefore, it is likely that many differentially expressed genes between cell types assigned by Predict Cell Types have been implicitly learned by the tool, and may not be specific to the dataset being analyzed.

The Statistical Comparison Table element





For each gene, the table has several columns whose interpretation depends on whether the tests performed are 'All group pairs' or 'Identify marker genes'. The difference in interpretation arises because the output of 'Identify marker genes' is a summary of several pairwise comparisons of the kind produced by 'All group pairs'.



For example, with three groups: 'Platelet', 'B cell', and 'T cell', 'All group pairs' will perform tests such as 'Platelet vs B cell', whereas 'Identify marker genes' will perform tests such as 'Platelet vs rest'. 'Platelet vs rest', will be a summary of the pairwise comparisons 'Platelet vs B cell' and 'Platelet vs T cell'.

- **Case (#)** , **Case (%)**, **Control (#)**, and **Control (%)**. For each group in the statistical comparison, the number (#) and percentage (%) of cells expressing the gene is calculated. For 'Platelet vs B cell', the case is 'Platelet' and the control is 'B cell'. For 'Platelet vs rest', the case group is 'Platelet', and the control groups are 'B cell' and 'T cell'. When there are multiple control groups, the minimum observed values for Control (#) and Control (%) are reported. Note that these two values might originate from two different control groups.
- **Max group mean.** For each group in the statistical comparison, the average expression value is calculated. For 'Platelet vs B cell' the groups are 'Platelet' and 'B cell'. For 'Platelet vs rest' the groups are 'Platelet', 'B cell' and 'T cell'. The 'Max Groups Mean' is the maximum of the average values.
- **Log2 fold change.** The logarithmic fold change.
- **Fold change.** The (signed) fold change. Genes that are not expressed in any cells used in the comparison have undefined fold changes and are reported as NaN (not a number). For an output of 'Identify marker genes', the fold change for a gene is the smallest magnitude fold change found in its component pairwise comparisons.

- **P-value.** Standard p-value. Genes that are not expressed in sufficient cells are reported as NaN (not a number). For an output of 'Identify marker genes', the p-value for a gene is the least significant p-value among the pairwise comparisons.
- **FDR p-value.** The false discovery rate corrected p-value. This is calculated directly from the values in the P-value column.
- **Bonferroni.** The Bonferroni corrected p-value. This is calculated directly from the values in the P-value column.

Downstream analyses using Statistical Comparison Tables

- Visualize the relationship between the p-values and the \log_2 fold changes using the **Volcano plot** view.
- Identify over-represented GO terms using **Gene Set Test** (.
- Compare differentially expressed genes from multiple Statistical Comparison Tables using **Create Venn Diagram for RNA-Seq** (.
- Compare the p-values and fold changes of all genes from multiple Statistical Comparison Tables using the table view of the Venn diagram produced by **Create Venn Diagram for RNA-Seq** (.
- Investigate pathways associated with differentially expressed genes by uploading Statistical Comparison Tables to an existing **QIAGEN Ingenuity Pathway Analysis** account using **Upload to IPA** () from the Biomedical Genomics Analysis plugin.





Note: Settings in **Gene Set Test** () and **Upload to IPA** () for filtering features using the 'Max group mean' need to be adjusted, as default values are based on the TPM measure of expression, which is rarely appropriate for single cell data.

9.1.2 The differential expression algorithm




The Differential Expression for Single Cell tool performs Mann-Whitney U tests (also known as Wilcoxon rank-sum tests) for each feature. If data is not normalized using the Normalize Single Cell Data tool (see section 7.2), the raw expression data is used. In this situation, results should be interpreted with caution, as differences could be solely driven by library size and other sample preparation specific factors.

When the data is normalized using the Normalize Single Cell Data tool, the Pearson residuals (see section 7.4.3) are used for performing the Mann-Whitney U tests. The residuals are however difficult to interpret, and therefore the 'Fold change' and 'Max group mean' are calculated using the estimated expression values if all cells had approximately 1,000 reads.

9.2 Create Expression Plot

Create Expression Plot generates visualizations of gene expressions for a small number of genes. It uses expressions from an input **Expression Matrix** ( / ) and groupings provided by **Cell Clusters** () or **Cell Annotations** (.



The tool can output:

- A **Heat Map** () with one row per gene and one column per cell.
- A **Dot Plot** () with one row per gene and one column per grouping of cells.
- A **Violin Plot** () with one violin distribution curve per combination of gene and group.

It is often most natural to run the tool from a Dimensionality Reduction Plot by right-clicking on the plot, see section 17 for details. However, it can also be found under the Tools menu at:

Tools | Single Cell Analysis () | **Gene Expression** () | **Expression Analysis** () | **Create Expression Plot** ()

The first set of options control how cells are grouped. The groupings are shown at the top of the Heat Map, form the columns of the Dot Plot and define groups in the Violin Plot. These options are:

- **Clusters** and **Cell annotations**. At least one of these must be supplied. **Clusters** accepts **Cell Clusters** () and **Cell annotations** accepts **Cell Annotations** ()
- **Group by**. One or more categories from the supplied Cell Clusters or Cell Annotations. Categories that only contain non-integer numerical data are not supported. If Cell Clusters contained a category 'Cell type' with values 'T cell', 'B cell' and 'Platelet', and Cell Annotations contained a category 'Status' with values 'Case' and 'Control', then selecting **Group by = Cell type, Status** would give groups 'T cell - Case', 'T cell - Control', 'B cell - Case', 'B cell - Control', 'Platelet - Case', and 'Platelet - Control'.
- **Select groups** (Optional). This can be supplied to reduce the number of groups of cells in the plot to only those of interest, or to control the order in which the groups are shown. For example, if the aim of the plot is to show how expression changes in T cells as a function of case / control, the 'T cell - Case' and 'T cell - Control' groups can be selected. If left empty, all groups will be displayed.

The genes in the output Heat Map or Dot Plot are clustered such that genes with similar expression patterns are found on adjacent rows. The clustering has a tree structure that is generated by

1. Letting each gene be a cluster.
2. Calculating pairwise distances between all clusters.
3. Joining the two closest clusters into one new cluster.
4. Iterating 2-3 until there is only one cluster left, containing all the genes.

In the Heat Map, the clustering is drawn as a tree where distances between clusters are reflected by the lengths of the branches in the tree.

The above algorithm requires a distance measure and a 'linkage' that describes how to apply the distance measure to clusters.

There are three kinds of **distance measures**:

- **Euclidean distance.** The length of the segment connecting two points. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Euclidean distance between u and v is

$$|u - v| = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}.$$

- **Manhattan distance.** The distance between two points measured along axes at right angles. If $u = (u_1, u_2, \dots, u_n)$ and $v = (v_1, v_2, \dots, v_n)$, then the Manhattan distance between u and v is

$$|u - v| = \sum_{i=1}^n |u_i - v_i|.$$

- **1 - Pearson correlation.** The Pearson correlation coefficient between $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$ is defined as

$$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \cdot \left(\frac{y_i - \bar{y}}{s_y} \right)$$

where \bar{x}/\bar{y} and s_x/s_y are the average and sample standard deviation, respectively, of the values in x/y values.

The Pearson correlation coefficient ranges from -1 to 1, with high absolute values indicating strong correlation, and values near 0 suggesting little to no relationship between the elements.

Using $1 - |\text{Pearson correlation}|$ as the distance measure ensures that highly correlated elements have a shorter distance, while elements with low correlation are farther apart.



The distance between two clusters is determined using one of the following linkage types:

- **Single linkage.** The distance between the two closest elements in the two clusters.
- **Average linkage.** The average distance between elements in the first cluster and elements in the second cluster.
- **Complete linkage.** The distance between the two farthest elements in the two clusters.


There are usually too many cells for all of them to be viewed in a Heat Map on a standard computer display. **Max cells in heat map** constructs the Heat Map by sampling the given number of cells from the full Expression Matrix. This option has no effect on the Dot Plot. Sampling works by sampling a fixed percentage of the cells in each grouping. For example, if there are 10 000 cells in the input, and 'Max cells in heat map = 1 000', then sampling will aim to recover 1 000 / 10 000 = 10% of the cells for each grouping. In this example, a group with <5 cells would be omitted, because 10% of <5 would be rounded down to 0.

There are also usually too many features to allow for a meaningful visualization of all genes. Therefore several options can be used to select the most informative genes to visualize:

- **Fixed number of features.** A specified number of features are kept.

- **Number of features.** This option is only available when data have been normalized by Normalize Single Cell Data. The given number of highly variable genes (HVGs) are selected according to the variance of their normalized values, from highest variance to lowest variance.
- **Filter features by statistics.** Features that are differentially expressed according to the specified thresholds are kept. All the thresholds must be satisfied in at least one of the input Statistical Comparison Tables.
 - **Statistical comparison.** One or more Statistical Comparison Tables, such as are produced by Differential Expression for Single Cell.
 - **Minimum absolute fold change.** Only features with an absolute fold change of this or higher are kept.
 - **Threshold.** Only features with a p-value of this or lower are kept. The p-value type can be specified.
- **Specify features.** A set of features, as specified by either an **Annotation Track**  or by plain text, are kept.
 - **Feature track.** Any features defined in the **Annotation Track**  are kept.
 - **Feature names.** A plain text list of case sensitive feature names. Any white-space characters, comma, and semicolon are accepted as separators.

9.2.1 The Heat Map output of Create Expression Plot

In a Heat Map each row corresponds to a gene and each column to a cell. The color in the i 'th row and j 'th column reflects the z-score normalized expression of feature i in cell j . This allows the relative expression of genes with very different average expressions to be visualized in the same plot, but means that expression values cannot be compared between genes - only between cells for the same gene. The normalized expression can be seen in the table view .

There are a number of options to change the appearance of the Heat Map. At the top of the **Side Panel**, you find the **Heat Map** group (see figure 9.2).

- **Lock width to window** When you zoom in the Heat Map, you will per default only zoom in on the vertical level. This is because the width of the Heat Map is locked to the window. If you uncheck this option, you will zoom both vertically and horizontally.
- **Lock height to window** This is the corresponding option for the height. Note that if you check both options, you will not be able to zoom at all, since both the width and the height are fixed.
- **Lock headers and footers** This will ensure that you are always able to see the cell and feature names and the trees when you zoom in.
- **Colors** The expression levels are visualized using a gradient color scheme, where the right side color is used for high expression levels and the left side color is used for low expression levels. You can change the coloring by clicking the box, and you can change the relative coloring of the values by dragging the two knobs on the white slider above.

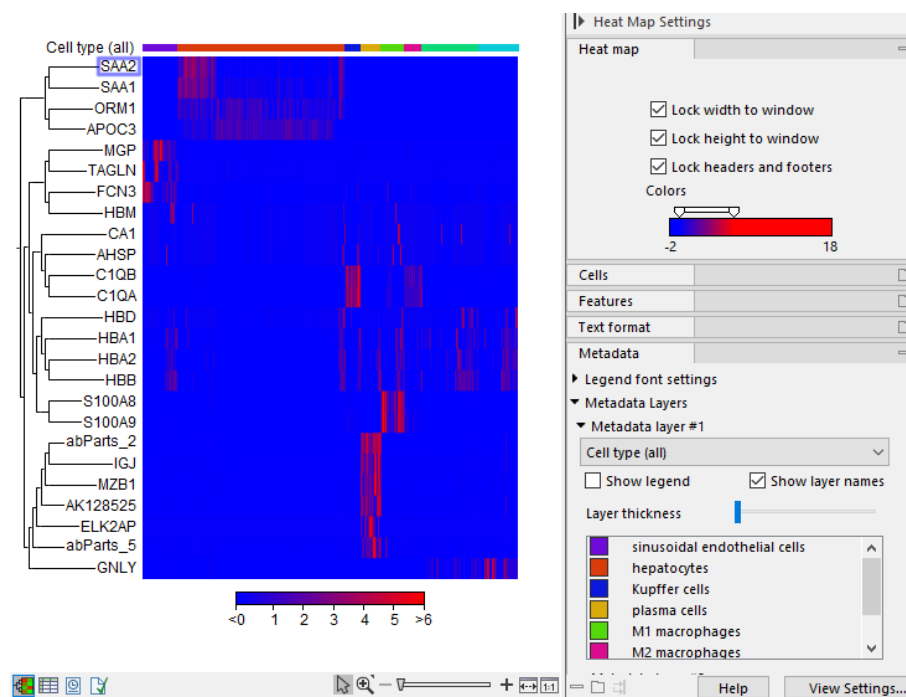


Figure 9.2: A heat map visualization of data from *MacParland et al., 2018*.

Below you find the **Cells** and **Features** groups. They contain options to show names, color legends, and, in the case of Features, trees at the left or right of the heat map. The tree options also control the **Tree size**, including the option of showing the full tree, no matter how much space it will use.

The **Metadata** group makes it possible to visualize all the information in the Cell Clusters and Cell Annotations supplied when the Heat Map was created:

- **Legend font settings** adjusts the label settings.
- **Metadata layers** Adds a color bar, colored according to the chosen metadata.

9.2.2 The Dot Plot output of Create Expression Plot

A Dot Plot summarizes the expression of all the cells in a grouping for each gene (see figure 9.3). You may need to scroll downwards or to the right to view all the data in the plot. Alternatively **Export Graphics** can be used to export the entire plot in an image format such as png, for more details see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Export_graphics_files.html.

In a Dot Plot, the expression values are z-score normalized like they are in the Heat Map. This allows the relative expression of genes with very different average expressions to be visualized in the same plot, but means that expression values cannot be compared between genes - only between cells for the same gene.

Each combination of gene and cell grouping is represented by a circle whose diameter is proportional to the percentage of cells in the grouping that express that gene. Note that scaling by diameter means that a gene expressed in 50% of cells will have one quarter the area of a gene

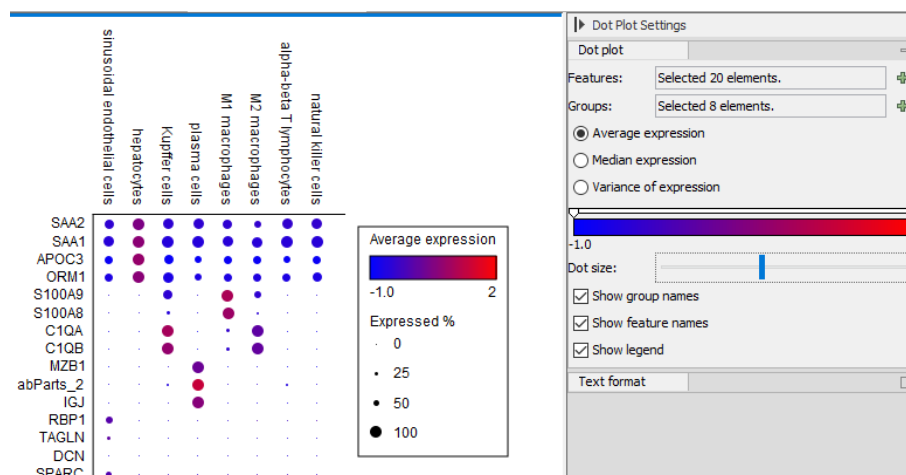


Figure 9.3: A Dot Plot visualization of data from *MacParland et al., 2018*.

expressed in 100% of cells. Good marker genes will typically be present in a large percentage of the cells for a cell type.

From the **Side Panel**, it is possible to change the values by which cells are colored. The options are:

Average expression The average expression of cells with at least some expression of the gene.

Median expression The median expression of cells with at least some expression of the gene. The median is more robust than the average in the sense that its value is less affected by outliers.

Variance of expression The sample variance of cells with at least some expression of the gene. In some cases, when a grouping of cells has high variance for a gene, then this may be evidence that the grouping contains more than one population of cells. Otherwise it may indicate that the gene's expression changes rapidly.

The order in which the clusters or genes are arranged is adjustable in the Features and Groups selection. Click the green plus and use the arrows to remove, add or rearrange the order of the visualized genes or clusters. The coloring can also be changed by clicking the color gradient in the Side Panel. The relative coloring of the values can be adjusted by dragging the two knobs on the white slider above the color gradient.

9.2.3 The Violin Plot output of Create Expression Plot

Violin plots superimpose a kernel density plot on a box plot in order to provide more insight into the distribution of expressions in a sample. The box plot shows the median as a filled black square, the interquartile ranges as an unfilled black box, and the range of other non-outlier measurements as whiskers. Surrounding the box plot is the estimated kernel density that shows the shape of the data. In places with a wide distribution the probability to find data points is much larger compared to the narrower sections (figure 9.4).

A number of options exist when looking at the Violin Plot **Side Panel**.

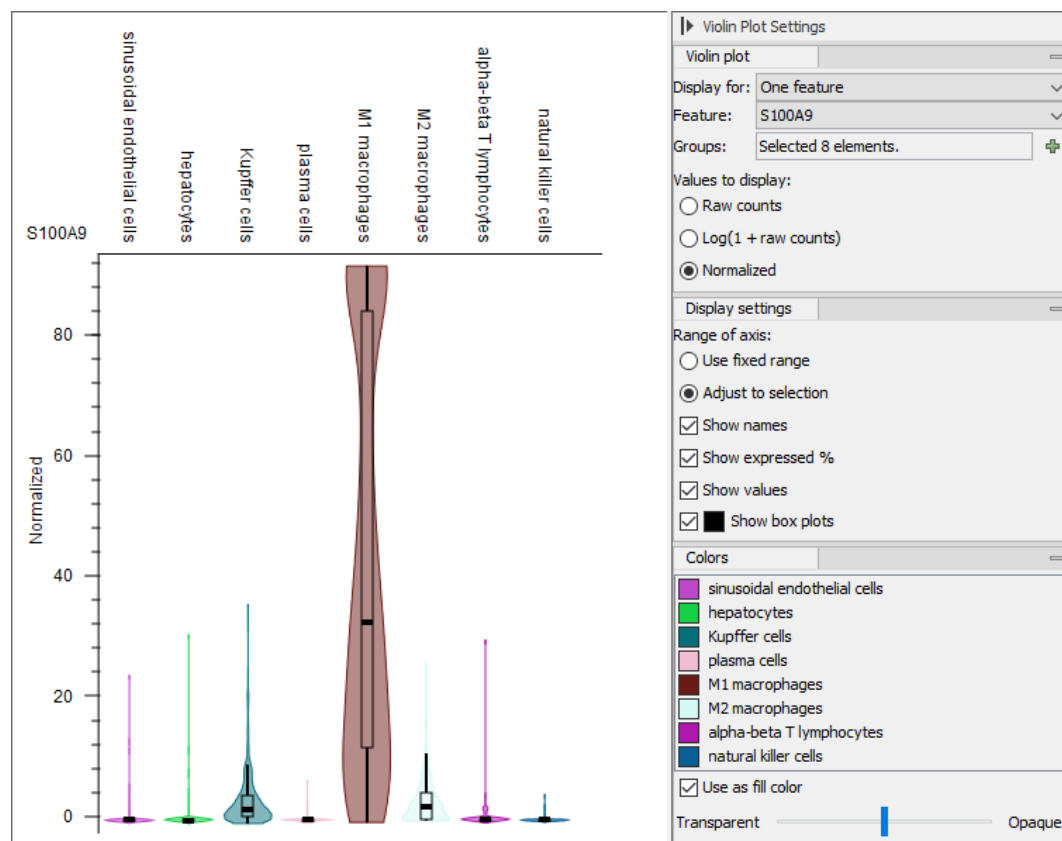


Figure 9.4: A Violin Plot visualization based on data from *MacParland et al., 2018* showing a specific feature across different groups of cells. Note the options in the Side Panel.

Violin plot Choose features, groups or a mixture to be displayed. Select how the data should be represented: as raw counts, $\log(1+\text{raw counts})$ or normalized.

Display settings Add a legend and values to the plot. Show and hide the box plot.

Colors Customize the colors of the individual plots.

A number of options for zooming and adjusting the size of the plot is provided. By clicking the small icons (+/-) or (fix) it is possible to fix either the x-axis or y-axis when zooming.

Several features can be displayed for one group of cells, as shown in figure 9.5. This can be useful for identifying marker genes when the cell types are not known or they need to be confirmed.

It is possible to pick and choose violins from different Violin Plots and to show them together in one plot. In order to do this, click on a feature or group label and right-click to bookmark the violin (see figure 9.6). All bookmarked violins will then appear in the same plotting area when selecting the bookmark option from the "Display for:" drop-down menu in the **Side Panel**. An example of a mixture plot is shown in figure 9.7.

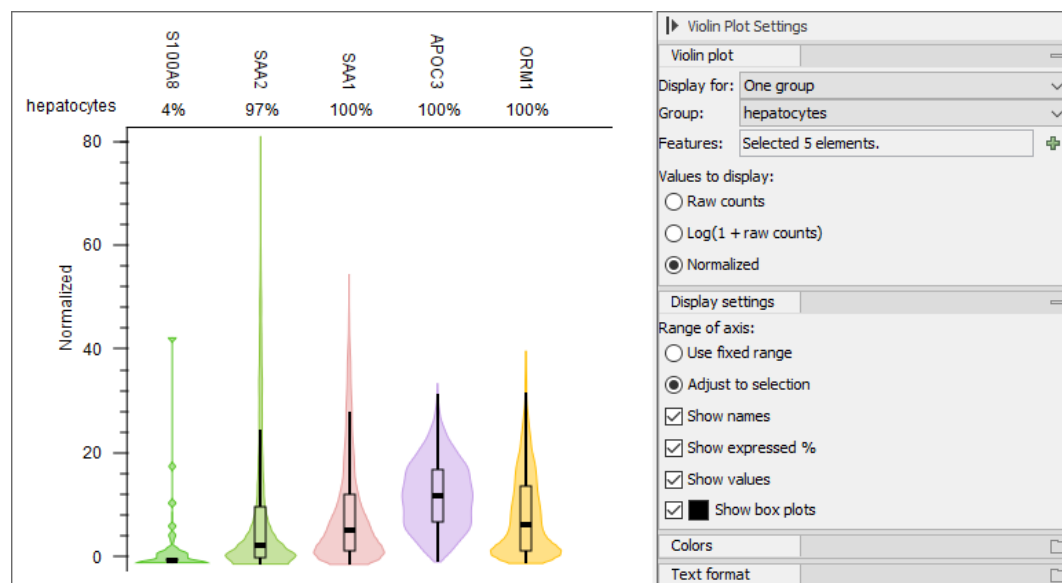


Figure 9.5: A Violin Plot visualization based on data from [MacParland et al., 2018](#). Here, selected features are shown for hepatocytes. The percentage of cells expressing each feature is shown on top of the violins.

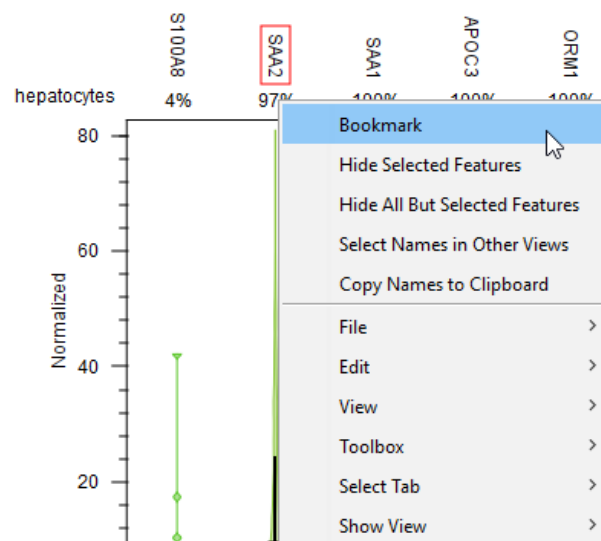


Figure 9.6: A Violin Plot visualization based on data from [MacParland et al., 2018](#) showing how to bookmark a violin.

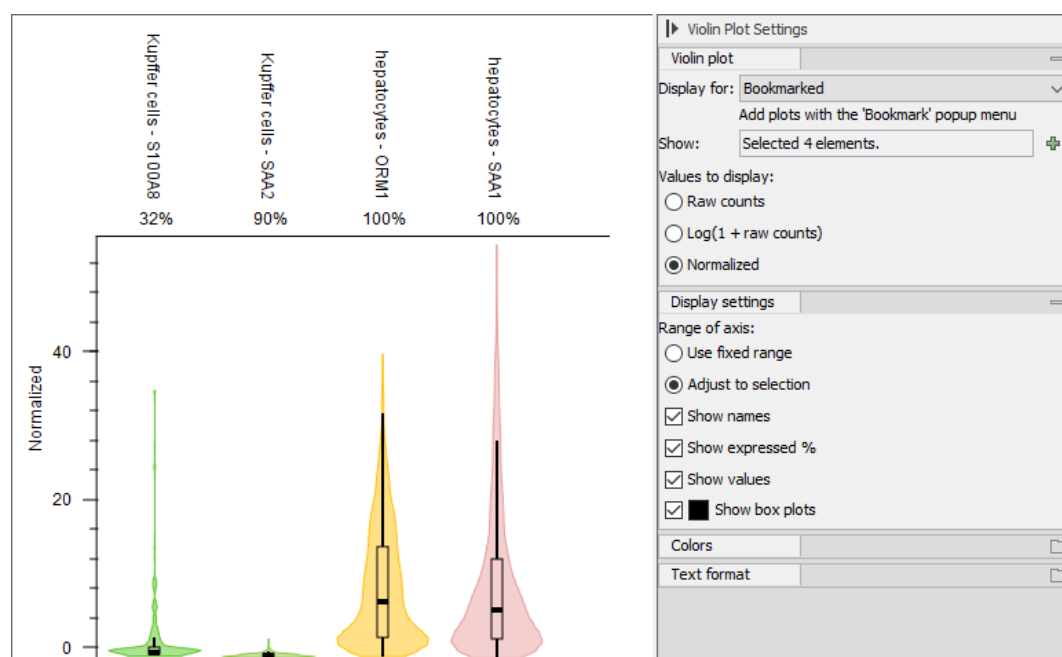


Figure 9.7: A Violin Plot visualization based on data from [MacParland et al., 2018](#) showing bookmarked violins with a mixture of both features and cell groups.

Chapter 10

Velocity Analysis

RNA velocity is a high-dimensional vector and a powerful approach to predict the future state of the individual cells on a timescale of hours, from the static snapshot provided by scRNA-Seq. This can help analyze time-resolved phenomena such as embryogenesis or tissue regeneration [La Manno et al., 2018, Bergen et al., 2020]. Visualizing this high-dimensional vector as arrows in a Dimensionality Reduction Plot provides an easy interpretation of the moving cell system (figure 10.1). Arrows show the direction and speed of movement of each cell, which can reveal differences between near-terminal cells, where arrows are short, and transient cells, where arrows are longer.

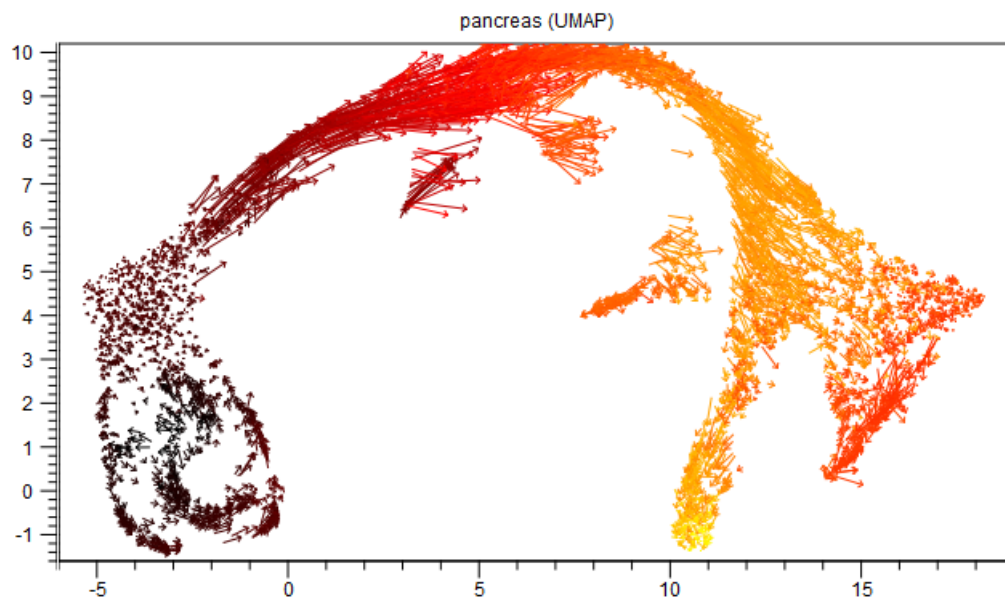





Figure 10.1: UMAP plot of the pancreas data set [Bastidas-Ponce et al., 2019] built-in scVelo [Bergen et al., 2020]. Arrows show the direction and speed of movement of an individual cell. The real time cells experience as they differentiate is approximated by the latent time, shown here in the 0 (black) to 1 (yellow) range.

Contents

10.1 Single Cell Velocity Analysis	138
10.1.1 The output of Single Cell Velocity Analysis	140

10.1.2 The velocity estimation algorithm	140
10.2 Differential Velocity for Single Cell	141
10.3 Score Velocity Genes	142
10.3.1 The output of Score Velocity Genes	143
10.4 Create Phase Portrait Plot	143
10.5 The Velocity Matrix element	144

10.1 Single Cell Velocity Analysis

Single Cell Velocity Analysis estimates velocities for studying cellular dynamics. It takes an **Expression Matrix with spliced and unspliced counts** () as input and produces a **Velocity Matrix** () and **Cell Annotations** (). We recommend normalizing the input with the Normalize Single Cell Data tool (see section 7.2).

Single Cell Velocity Analysis is available from:

Tools | Single Cell Analysis () | **Gene Expression** () | **Velocity Analysis** () | **Single Cell Velocity Analysis** ()

The tool offers options to run dimensionality reduction or feature selection prior to velocity estimation. To perform feature selection through highly variable genes (HVGs), the data has to be normalized first with the Normalize Single Cell Data tool. When HVGs are used, velocity is estimated only for these genes. This can greatly speed up the calculations. We recommend using HVGs whenever possible. Note that top velocity genes are not necessarily top HVGs. The default value of 2,000 is a good starting point - a too small value can lead to missing important velocity genes, while a too high value will diminish the computation gain. For details on dimensionality reduction or feature selection, please see section 14.1.

The following additional options are available (figure 10.2):

- **Neighborhood size.** The number of cells ‘k’ used in the k-nearest neighbor graph for imputing spliced and unspliced counts. This determines the granularity of the imputation.
- **Model.** Two models are available to estimate velocities:
 - **Steady-state model:** infers a steady-state ratio of unspliced to spliced mRNA levels, and determines the velocities as deviations from this ratio [La Manno et al., 2018]. It is fast but can be less accurate.
 - **Dynamical model:** performs a likelihood-based inference of the full splicing kinetics and generalizes RNA velocity estimation to transient systems. Unlike the steady-state model, it is robust to non-observed steady-states [Bergen et al., 2020], but is much slower.

See section 10.1.2 for details.

- **Calculate velocity for each sample independently.** If multiple samples are present in the input and this is enabled, the k-nearest neighbor graph, normalization and imputation (see section 10.1.2 for details) will be performed for each sample independently, while the remaining velocity estimation will be performed using all the cells jointly. Otherwise, all cells are used jointly throughout the entire algorithm. We recommend running with inputs

Figure 10.2: The options in the dialog of the Single Cell Velocity Analysis tool.



containing just one sample, and caution should be used otherwise when interpreting the output, see discussion below.

Multi-sample input: There are no well-established approaches for joint batch correction of spliced and unspliced counts. We recommend caution when analyzing a matrix containing multiple samples. If the matrix is batch corrected using the Normalize Single Cell Data tool (see section 7.2), then the correction is only applied to the total gene expression, which is used for k-nearest neighbor graph construction, and not to the spliced and unspliced counts, which are used for velocity estimation. See [Bergen et al., 2021] for a discussion on batch correction for velocity estimation.

Single nucleus RNA sequencing (snRNA-Seq): Velocity estimation has been developed for scRNA-Seq data and it is yet to be determined how well the method works for snRNA-Seq, where the assumptions of the model might not hold [Bergen et al., 2021]. We recommend caution when analyzing and interpreting the results for this type of data.

10.1.1 The output of Single Cell Velocity Analysis

Single Cell Velocity Analysis outputs:

- A **Velocity Matrix** () element containing the estimated velocities for the velocity genes, as well as the imputed spliced and unspliced counts for all sufficiently expressed genes that, if **Use highly variable genes** was checked, are also HVGs. See section 10.5 for information on Velocity Matrix elements.
- A **Cell Annotations** () element containing the velocity coherence and length. If the dynamical model was run, the output also contains the estimated latent time. See section ?? for information on Cell Annotations elements.

See section 10.1.2 for details.

When using the Velocity Matrix in a Dimensionality Reduction Plot (see chapter 16), the velocities are projected onto the embedding and visualized as arrows. The velocity coherence, length and, if available, the latent time can also be visualized in the Dimensionality Reduction Plot as any other Cell Annotations. See section 17.1 for details.

10.1.2 The velocity estimation algorithm

The velocity estimation algorithm closely follows the approach implemented in scVelo [Bergen et al., 2020] and it contains multiple steps, covering different scVelo methods:

1. Genes are filtered such that only the genes that are HVGs (if used) and have a sufficient spliced and unspliced count are retained for the velocity calculations. Spliced and unspliced counts are then normalized to correct for sequencing depth. Each type of count is divided by the total observed count of the cell, and multiplied to the median total count across all cells. The total counts are obtained by summing over the genes retained for the velocity calculations.
2. A k-nearest neighbor graph is calculated using the pairwise Euclidean distance between all cells, using either the raw or normalized, if available, total gene expression, after any optional HVGs selection and dimensionality reduction. Using each cell's nearest neighbors, the spliced and unspliced counts are imputed as the average normalized spliced and unspliced counts across the neighborhood.
3. Velocity is estimated for each gene according to the chosen model:
 - Steady-state model: A linear regression on the extreme quantiles of the spliced and unspliced counts is used to determine the steady-state ratio. Genes are considered velocity genes if the inferred ratio and R^2 are above 0.01.
 - Dynamical model: To reduce computational cost, the dynamical model is only estimated for genes that are considered velocity genes based on the steady-state model. If the gene likelihood is larger than 0.001, the gene is considered a velocity gene.

Downstream analysis is only performed using the velocity genes.

4. Transition probabilities are calculated from the cosine similarity, which measures how well the change in gene expression can be explained by the estimated velocity vector.

5. Velocity coherence (how a velocity vector correlates with its neighboring velocities) and length (speed or rate of differentiation) are calculated for each cell.
6. If the dynamical model was used, the gene-shared latent time representing the cells' internal clocks is estimated by using the inferred dynamics.

Note that inference of the terminal states requires calculating the eigenvector of the transition probability matrix corresponding to an eigenvalue 1. If the estimated eigenvalue is not sufficiently close to 1, the resulting terminal states are not trustworthy and hence the latent time is not calculated.

The above steps are equivalent to running the following commands in scVelo 0.2.4:

```
import scVelo as scv

scv.pp.filter_and_normalize(adata, min_shared_counts=20, n_top_genes=2000)
scv.pp.moments(adata, n_neighbors=30, n_pcs=20)

# only for the steady-state model
scv.tl.velocity(adata, mode='deterministic')

# only for the dynamical model
scv.tl.recover_dynamics(adata)
scv.tl.velocity(adata, mode='dynamical')




scv.tl.velocity_graph(adata)
scv.tl.get_transition_matrix(adata, scale=10, self_transitions=True, use_negative_counts=True)

scv.tl.velocity_confidence(adata)

# only for the dynamical model
scv.tl.latent_time(adata)
```

Note that small differences to scVelo are expected in the results due to the different normalized total gene expression. Additionally, the dynamical model uses numerical optimization and this can lead to different estimated kinetic parameters and hence estimated velocities.


10.2 Differential Velocity for Single Cell

Differential Velocity for Single Cell performs differential analysis from an input **Velocity Matrix** () and groupings provided by **Cell Clusters** () or **Cell Annotations** ()

Note: Differential Velocity for Single Cell is complementary to Score Velocity Genes, which reports likelihoods. Differential Velocity for Single Cell performs statistical tests to report p-values and identify genes that show different velocity patterns in between groups of cells.

It is often most natural to run the tool from a Dimensionality Reduction Plot by right-clicking on the plot, see section 17 for details. However, it can also be found under the Tools menu at:

Tools | Single Cell Analysis  | **Gene Expression**  | **Velocity Analysis**  | **Differential Velocity for Single Cell** 

The tool performs a differential analysis for the velocity of each gene and outputs Statistical Comparison Tables .




The available options specify the type of test to be performed and how genes can be filtered before testing, in a similar manner as for Differential Expression for Single Cell; see section 9.1 for details. Note that Differential Velocity for Single Cell can only run an 'Identify marker genes' analysis and the 'All group pairs' option is not available.

The tool performs pairwise comparisons by using the estimated velocities for each gene to calculate:

- **Max group mean.** The maximum of the average velocities of the two groups. Can be negative.
- **Fold change.** The (signed) fold change, calculated as the ratio between the average velocities of the two groups. Note that if one group has a positive average, while the other group has a negative average, the fold change is reported as NaN (not a number).
- **P-value.** The p-value is obtained from a Mann-Whitney U test (also known as Wilcoxon rank-sum test).

See section 9.1.1 for more details on the output and section 9.1 for details on how the pairwise comparisons are used to 'Identify marker genes'.

10.3 Score Velocity Genes

Score Velocity Genes produces likelihoods for the velocity genes found in an input **Velocity Matrix**  produced with the dynamical model. It uses groupings provided by **Cell Clusters**  or **Cell Annotations** .



Steady-state model: It is not possible to run the tool on a matrix produced with the steady-state model. For this, use Differential Velocity for Single Cell instead.

Note: Score Velocity Genes is complementary to Differential Velocity for Single Cell, which performs statistical tests to report p-values. Score Velocity Genes can be used to identify the genes driving the observed dynamics, either for the entire data set or a group of cells, by ranking the genes (from largest to smallest) according to the likelihood. Some genes might be equally important for two different sets of cells, without them showing differential velocity patterns.


It is often most natural to run the tool from a Dimensionality Reduction Plot by right-clicking on the plot, see section 17 for details. However, it can also be found under the Tools menu at:

Tools | Single Cell Analysis  | **Gene Expression**  | **Velocity Analysis**  | **Score Velocity Genes** 


The set set of options narrow down the focus of the tool:

- **Clusters** and **Cell annotations**. At least one of these must be supplied. **Clusters** accepts **Cell Clusters**  and **Cell annotations** accepts **Cell Annotations** .
- **Score velocity genes for** a single column from the supplied Cell Clusters or Cell Annotations. Columns that only contain true/false values or numerical data are not supported. Tests will be performed between the groups of cells with different labels in this column.
- **Select groups** (Optional). This can be supplied to reduce the number of groups of cells considered or to control the order in which comparisons are made.

For details on how groups of cells can be defined, see section 9.1.

The tool outputs the gene likelihoods obtained from the dynamical model to a table , both for the defined groups of cells, and the entire data set. The performed calculations closely follow those from scVelo's `rank_dynamical_genes` method [Bergen et al., 2020].

10.3.1 The output of Score Velocity Genes

Score Velocity Genes produces one table , with one gene per row.

For each gene, the table has several columns, depending on how the groups of cells have been defined.

For example, if an input Velocity Matrix named 'velocity matrix' was used and three groups 'Platelet', 'B cell', and 'T cell' were defined, the output will contain the following columns:



- *Name and Identifier*: the gene name and identifier, as present in the input Velocity Matrix;
- *velocity matrix*: the score for the entire data set;
- *Platelet, B cell and T cell*: the scores calculated using the cells belonging to the three groups.

Note that only the velocity genes for which velocity estimates are present in the input Velocity Matrix are present in the output.

10.4 Create Phase Portrait Plot

Create Phase Portrait Plot is available from:

Tools | Single Cell Analysis  | **Gene Expression**  | **Velocity Analysis**  | **Create Phase Portrait Plot** 

The tool takes a **Velocity Matrix**  as input and produces a **Phase Portrait Plot**  containing phase portraits for all genes with imputed spliced and unspliced counts (see section 10.1.1 for details). The estimated velocities and inferred dynamics can also be visualized for the velocity genes.

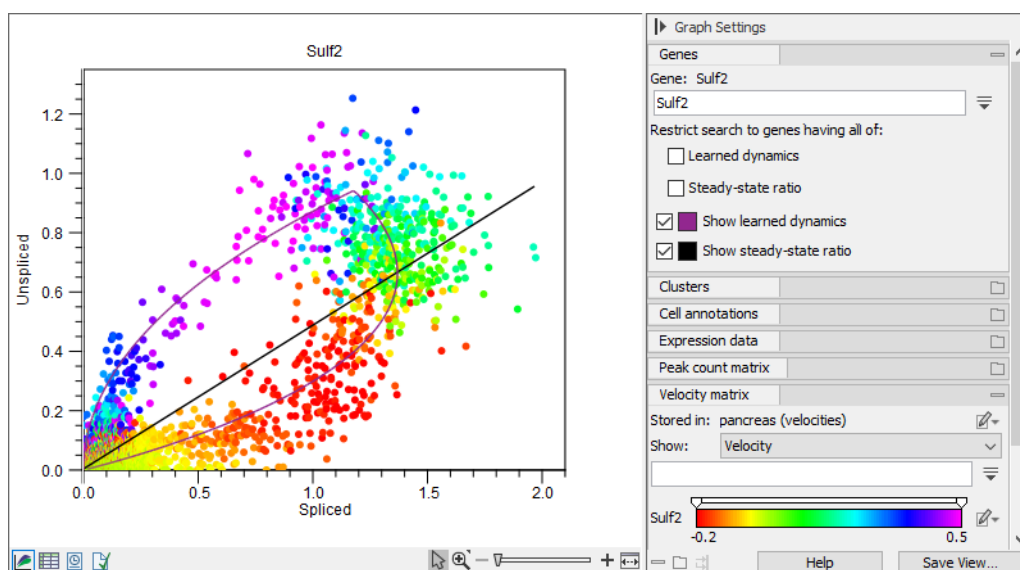


Figure 10.3: A phase portrait with inferred dynamics for the pancreas data set [Bastidas-Ponce et al., 2019] built-in scVelo [Bergen et al., 2020].

An example output is shown in figure 10.3.

The gene to be shown in the phase portrait can be chosen under the ‘Genes’ group at the top right of the Side Panel. When a new gene is selected, the cells are automatically colored by the velocity component for that gene, if available.

The gene search can be limited to only showing the genes for which velocity has been estimated, by selecting ‘Learned dynamics’ and/or ‘Steady-state ratio’ under ‘Restrict search to genes having all of:’. Hit space in the search field to list all genes, subject to these restrictions (if any).

The inferred dynamics, if available, and steady-state ratio can be shown or hidden by toggling the ‘Show learned dynamics’ and ‘Show steady-state ratio’.

When changing to the table (📊) view of the plot, all genes for which a phase portrait is available will be listed per row. Choosing one row in this table will update the plot to show the phase portrait for the selected gene.

Using a Phase Portrait Plot, various aspects of the data can be visualized, see chapter 17 for details. Note that it is not possible to edit clusters or launch tools using Phase Portrait Plots.

10.5 The Velocity Matrix element

Velocity Matrix (📊) elements are Tracks that hold information about gene velocity, as well as imputed spliced and unspliced expressions. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tracks.html> for more information on Tracks. Velocity Matrix elements are produced by Single Cell Velocity Analysis, see section 10.1.

Velocity Matrix elements contain various table views, described below. See https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Working_with_tables.html for general information on working with tables.

Feature Table

The Feature Table () for a Velocity Matrix () contains one row for each feature, and has the following columns:

- **Name.** The name of the feature.
- **Id.** A feature identifier.
- **Type.** The feature type.
- **Chromosome.** The chromosome that the feature is on.
- **Region.** The region that the feature spans.



Additionally, for gene velocity, as well as imputed spliced and unspliced expressions, the table contains the following columns:

- **Min/Max/Avg.** The minimum/maximum/average velocity or expression for the feature across all cells.
- **Cells w/.** The number of cells with a non-zero velocity or expression for the feature.

Clicking on a row opens a separate table, listing the cells expressing the feature.

To create a new matrix element from a row selection in the Feature Table, use the **Create Matrix from Selection** option in the right-click menu. Cells that express at least one of the selected features are included.

Cell Table


The Cell Table () for a Velocity Matrix () contains one row for each cell, and has the following columns for gene velocity, as well as imputed spliced and unspliced expressions:

- **Nonzero.** The number of features for which the cell has a non-zero velocity or expression.

Clicking on a row opens a separate table, listing the features expressed by the cell.

To create a new matrix element from a row selection in the Cell Table, use the **Create Matrix from Selection** option in the right-click menu.

Transition Probabilities Table

The Transition Probabilities Table () contains one row for each cell, and has the following columns:

- **Sample.** The sample that the cell is from.
- **Barcode.** The cell barcode.
- **Non-zero incoming probability.** The number of cells with a non-zero probability of transitioning towards the cell.

- **Non-zero outgoing probability.** The number of cells that the cell has a non-zero probability of transitioning towards.

Clicking on a row opens a separate table, listing the cells with a non-zero incoming/outgoing probability from/towards the selected cell.

Chapter 11

Spatial Transcriptomics



Contents

11.1 The Spatial Transcriptomics Plot element	147
--	------------

The location of a cell in a multicellular organism is crucial for its function. Towards the goal of fully characterizing cells' functions and understanding tissue architecture, spatial transcriptomics exposes tissue heterogeneity by quantifying and localizing the gene expression in the tissue context.

CLC Single Cell Analysis Module offers a tool for importing spatial transcriptomics data from Space Ranger spatial outputs (see section 4.7). Various aspects of the data can be visualized using the resulting Spatial Transcriptomics Plot. Additionally, the plot can be linked to a Dimensionality Reduction Plot, such that the same visualization can be applied simultaneously to both plots.

11.1 The Spatial Transcriptomics Plot element

A Spatial Transcriptomics Plot () represents each barcode as one dot, with its position determined by the spatial position within the tissue, optionally overlayed on an image of the corresponding tissue (figure 11.1). The spatial position of each barcode can be seen in the table () view.

Using a Spatial Transcriptomics Plot, various aspects of the data can be visualized, cells can be manually annotated and various tools can be started using the information selected in the plot, see chapter 17 for details.

When **Overlay on image** is checked, the tissue image, if available, is shown and the barcodes are displayed on top of it. If the image contains the fiducial markers, **Fit to tissue** can be used to clip the image to only show the tissue where barcodes have been detected. Using the sliders, the brightness, contrast, saturation and transparency of the image, as well as the transparency of the dots, can be controlled.

Linking to a Dimensionality Reduction Plot

A Spatial Transcriptomics Plot can be linked to a Dimensionality Reduction Plot, such that the

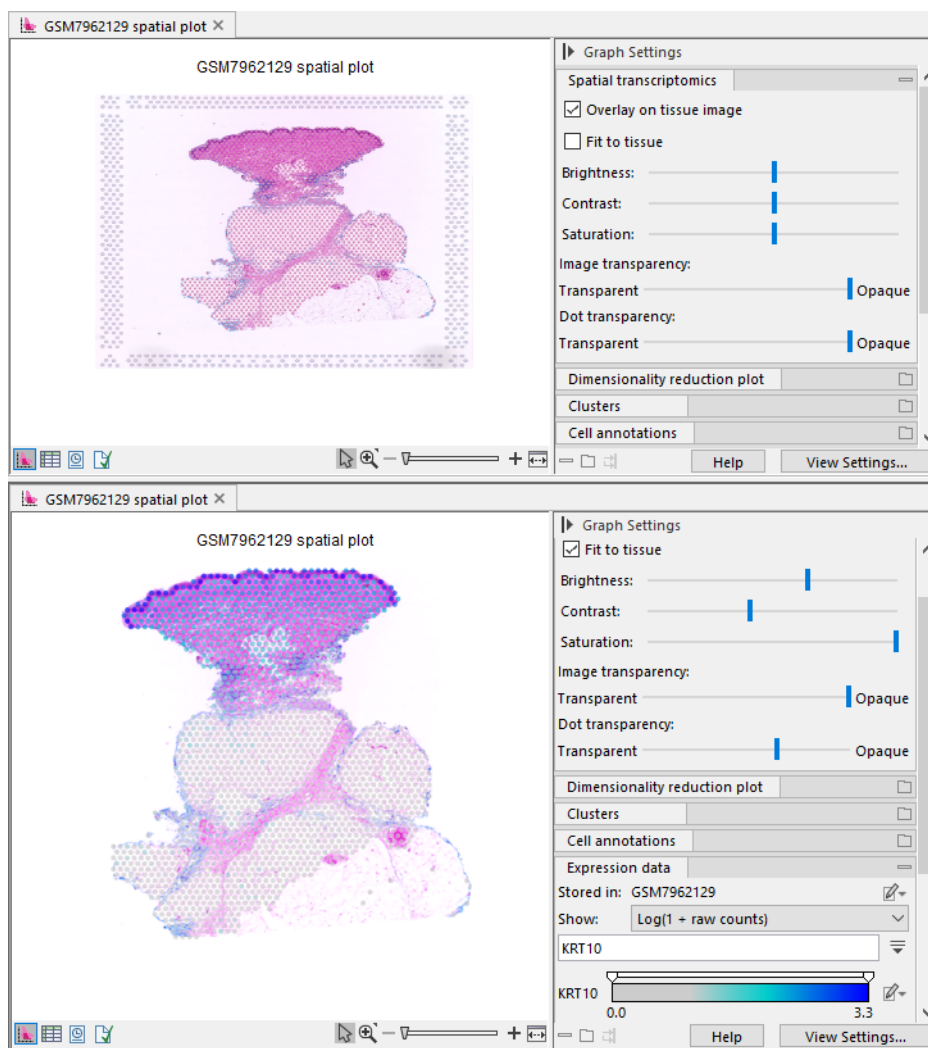


Figure 11.1: A Spatial Transcriptomics Plot of the GSM7962129 data from the Gene Expression Omnibus repository. Top: Default view. Bottom: The image is fit to the tissue, and brightness, contrast, saturation and dot transparency are adjusted. The barcodes are colored using the expression of KRT10.

options and selections are mirrored in both plots. For example:

- The source of colors in the Spatial Transcriptomics Plot is controlled from the Side Panel of the Dimensionality Reduction Plot.
- Lasso selection of barcodes in one plot is reflected in the other.

The barcodes in the Dimensionality Reduction Plot and those in the Spatial Transcriptomics Plot need to have the same sample name. Ideally, it should be ensured that these share the sample name as a first step when importing the Spatial Transcriptomics Plot element (see section 4.7). If this has not been done, the sample name can be updated using the Update Single Cell Sample Name tool (see section 18.7).

To link the plots (figure 11.2):

- Open the Dimensionality Reduction Plot, if it is not already open. To visualize the plots side by side, use a split view, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Arrange_views_in_View_Area.html for details.
- Associate the Dimensionality Reduction Plot to the Spatial Transcriptomics Plot by dragging the Dimensionality Reduction Plot into the 'Dimensionality reduction plot' Side Panel group of the Spatial Transcriptomics Plot. The association can be saved by saving the changes to the Spatial Transcriptomics Plot.
- Check **Link to plot** to link the two plots.

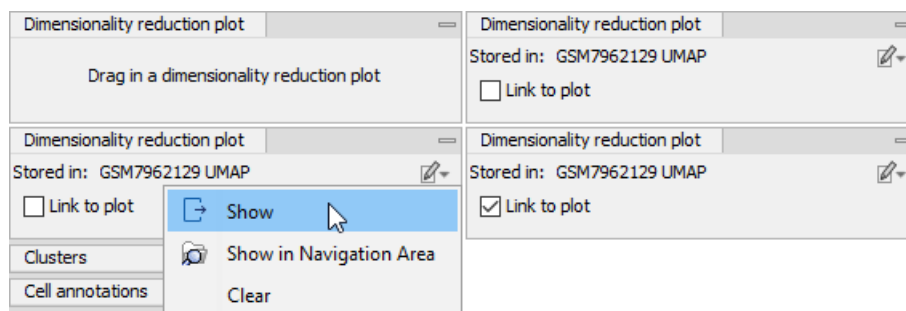


Figure 11.2: A Dimensionality Reduction Plot can be associated to a Spatial Transcriptomics Plot. Top left: Initial view. Top right: A plot has been dragged into the Side Panel and is now associated. Bottom left: An associated plot can be opened from the menu in the top right corner of the Side Panel group. Bottom right: The link is active.

On subsequent uses of the Spatial Transcriptomics Plot, the Dimensionality Reduction Plot can be opened from the Side Panel (figure 11.2).

The link is active if 'Link to plot' is checked and enabled (figure 11.2). When there is an active link, most of the Side Panel groups for the Spatial Transcriptomics Plot are disabled and the coloring is based on the options chosen in the Dimensionality Reduction Plot (figure 11.3).

When the active link is disabled by unchecking **Link to plot**, the Side Panel groups for the Spatial Transcriptomics Plot are enabled again and the coloring is reverted to the state before the link was activated.

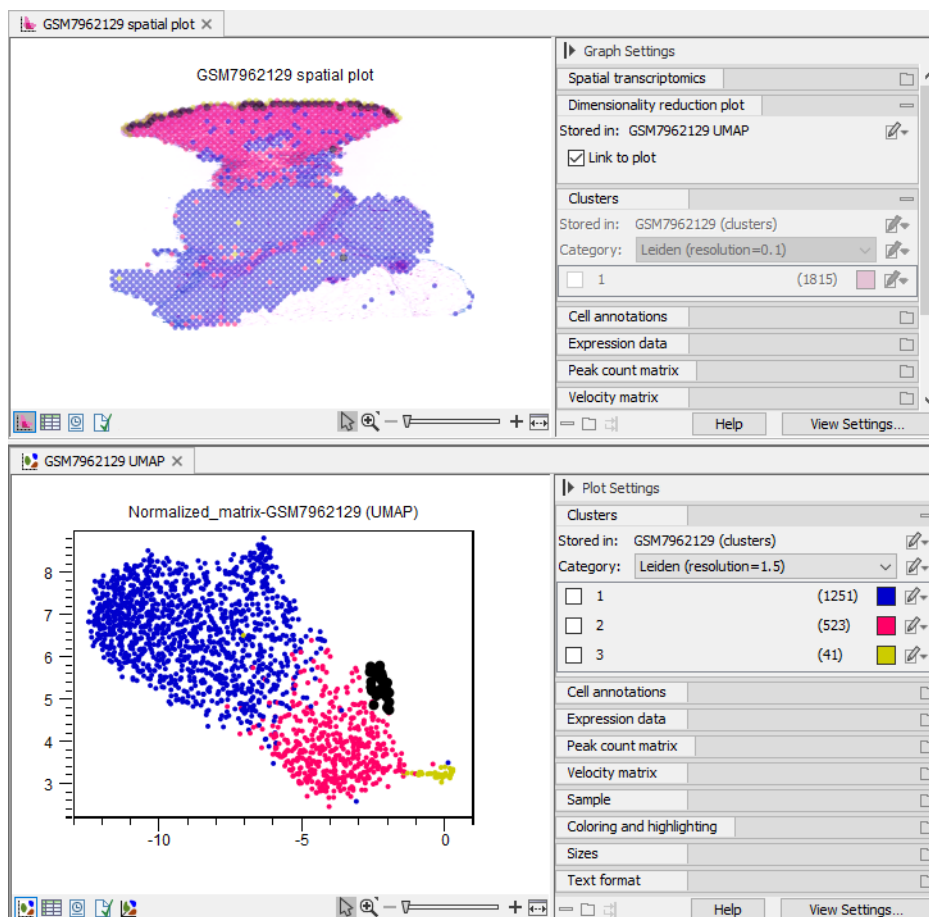


Figure 11.3: The source of colors in the Spatial Transcriptomics Plot is controlled from the Side Panel of the Dimensionality Reduction Plot. Lasso selections in either plot is reflected in both plots.

Chapter 12

Chromatin Accessibility


Contents






12.1 Single Cell ATAC-Seq Analysis	151
12.1.1 The output of Single Cell ATAC-Seq Analysis	153
12.1.2 The report output from Single Cell ATAC-Seq Analysis	153
12.1.3 The Single Cell ATAC-Seq Analysis algorithm	156
12.2 Split Read Mapping by Cell	158
12.3 Differential Accessibility for Single Cell	162
12.3.1 The differential accessibility algorithm	162
12.4 The Peak Count Matrix element	164

12.1 Single Cell ATAC-Seq Analysis

Single Cell ATAC-Seq Analysis is available from:




Tools | Single Cell Analysis  | **Chromatin Accessibility**  | **Single Cell ATAC-Seq Analysis** 

The tool takes as input a single read mapping  of reads that have been annotated using **Annotate Single Cell Reads**. The tool outputs:

- A **Peak Count Matrix**  with annotated nearby genes and transcription factors.
- The **Read Mapping**  that was used for peak calling.
- An **Annotation Track**  of transcription factor motifs found within the peaks.
- A **Graph Track**  showing the footprint score at each position.
- A **Report**  providing a summary of the data and diagnostic plots for quality control.

It is important that the input read mapping contains all the samples that will be used in a downstream analysis. This is because it is not possible to combine Peak Count Matrices as they will typically have different coordinates for shared peaks. There are two ways to generate a single read mapping from multiple samples:

1. Provide multiple read lists to **Map Reads to Reference** https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Map_Reads_Reference.html.
2. Merge existing read mappings using **Merge Read Mappings** https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Merge_Read_Mappings.html.

The tool requires a **Peak Shape Filter** () for calling scATAC-Seq peaks, and both a **Gene track** () and a corresponding **mRNA track** () for assigning nearby genes to peaks. These data can be directly downloaded using the Reference Data Manager (see chapter 2).


It is also possible to supply custom **Peak Shape Filter**, **Gene track** and **mRNA track** as follows:


- **Peak Shape Filters** can be generated by **Learn Peak Shape Filter** (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Learn_Peak_Shape_Filter.html).
- **Gene and mRNA tracks** can be imported from gff/gff3/gtf files (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Import_tracks.html).



The following additional options are available:

- **Maximum P-value for peak calling.** The threshold for reporting peaks, higher values will increase the number of called peaks.
- **Minimum peak count.** The number of peaks a barcode must have to be called as a cell. Barcodes that do not have this many peaks will not be present in the Peak Count Matrix. This option is the scATAC-Seq equivalent of QC for Single Cell. It is effective despite its simplicity because:
 1. Peaks must be shared by other cells to have been detected by the peak caller, meaning that this metric is not affected by the presence of large numbers of randomly mapping reads.
 2. The minimum number of peaks is related to the amount of open chromatin per cell, which is presumed to have a high lower bound for any active cell.
 3. Sequencing is expected to sample peaks uniformly, so identifying non-cells is easier than for gene expression, where a cell might have so much expression of one gene that it is hard to detect others even though they are present.
- **Chromosomes to ignore.** As it lacks chromatin, many reads map to the mitochondria chromosome. Ignoring the mitochondria chromosome can therefore speed up analysis and improve results by removing the possibility that peaks are called there. If viral genomes have been added to the reference as decoys, then these should also be ignored. When configuring this option in a workflow, multiple chromosome names can be provided as a comma-separated list.


12.1.1 The output of Single Cell ATAC-Seq Analysis


The main output of Single Cell ATAC-Seq Analysis is a **Peak Count Matrix** (). This can be used directly in **Cluster Single Cell Data**, **tSNE for Single Cell**, and **UMAP for Single Cell**.

The **Read Mapping** () that was used for calling peaks is also produced. We recommend using this rather than the original read mapping in later analyses, because it is smaller and the shape of peaks can be seen more clearly.

The **Motif Track** () output shows the location of all transcription factor motifs found within peaks. Each row in the table view () contains the following information:

- **Chromosome.** The chromosome on which the motif was found.
- **Region.** The region matching the motif.
- **Name.** The name of the transcription factor.
- **Score.** The score for matching the region against the motif.
- **Score threshold.** The score threshold that must be reached for the match to be significant at p-value 0.0001. This is specific to each motif - for example, longer motifs will typically require a higher score to be as significant as shorter ones. All reported motifs have a score higher than the score threshold. The difference between the score and score threshold for several overlapping motifs may give an indication of which motif is the 'best fit' for a region.
- **Footprint score.** The footprint score at the middle of the motif. Higher scores show more evidence of transcription factor binding.
- **Bound.** "Yes" if the footprint score is higher than a threshold determined from the data, and otherwise "No". Only transcription factors for each peak that are bound are reported in the Peak Count Matrix.

The **Footprint Graph Track** () output shows the footprint score at all positions on the genome. This is mainly provided for visualization.

The **Report** () is useful for quality control, and is described separately in section [12.1.2](#).

12.1.2 The report output from Single Cell ATAC-Seq Analysis

The report contains the following sections:

Reads

For each sample, the following information is shown:

- **Input read pairs.** The number of paired reads in the read mapping. This includes pairs that are mapped ambiguously, but excludes pairs on chromosomes supplied to the **Chromosomes to ignore** option.
- **Unique read pairs.** The number of paired reads after removing reads that map ambiguously.

- **% Unique.** "Unique read pairs" / "Input read pairs" x 100. If the sample was PCR amplified, then a low "% Unique" indicates that most fragments in the sample were sequenced.

Comparing these values across samples may reveal biases. For example, if control samples have more reads than case samples, then one might expect to see a higher proportion of cells for each peak for the control samples.

Fragments

A single fragment size distribution plot is shown for all the data. This plot has a characteristic shape for scATAC-Seq data, as seen in figure 12.1. The absence of this shape may indicate failed library preparation.

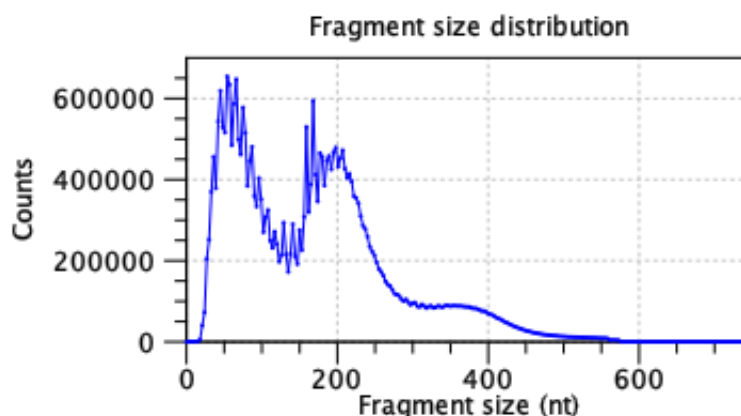


Figure 12.1: A characteristic ATAC-seq fragment size distribution. The fragment size distribution should have few fragments <30 nt as this is too small for the Tn5 transposase to bind. Short fragments are usually most abundant. A peak should be seen at about 180 nt. Subsequent peaks may be present with nucleosome spacing i.e. a new peak approximately 147 nt after each previous peak. A high frequency periodicity may be observed for small fragment sizes. This is related to the DNA helix pitch. Data is for two samples from [Taavitsainen et al., 2021](#).

Two additional metrics are shown per sample:

- **Fragments in peaks.** The total number of read pairs counted per peak and barcode. This is calculated on all barcodes before any filtering by the **Minimum peak count** option. For details of which read pairs are counted, see section 12.1.3.
- **% In peaks.** "Fragments in peaks" / "Unique read pairs" x 100. If this is low, check the Read Mapping output to see whether reads map to a specific chromosome that should be ignored, are distributed evenly across the genome (which may indicate failed library preparation), or are piled up at particular regions of high coverage. The latter is expected to some degree, but is not expected to affect downstream analysis. Lists of known high coverage regions are available from the ENCODE project for human and mouse [[Amemiya et al., 2019](#)].

Tn5 bias correction

The Tn5 enzyme has a bias towards certain sequences. This should be seen in the "before" lines of the nucleotide frequency plots (figure 12.2). An absence of a detectable bias indicates problems with library preparation. A different bias may reflect use of a different enzyme.

The "after" lines should show markedly less bias. Bias correction is used to improve the assignment of transcription factors to peaks via footprinting. Failure to correct for bias may lead to more transcription factors being associated with each peak.

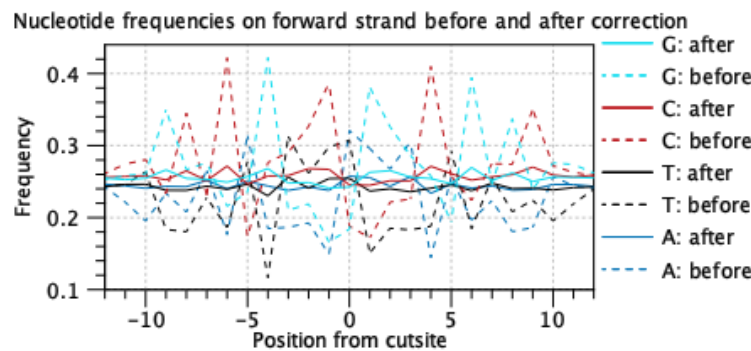


Figure 12.2: A characteristic Tn5 insertion bias is seen in the "before" lines. This is reduced after bias correction as part of footprinting. Data is for two samples from [Taavitsainen et al., 2021](#).

Cells

A barcode rank plot is shown for all the samples. An example is shown in figure 12.3. The red horizontal line shows the cutoff specified by the **Minimum peak count** option. All barcodes above the red line are retained as cells, and all barcodes below the line are discarded. The lines for each sample should be nearly vertical at the point where they cross the threshold line, indicating an abrupt fall in the number of peaks at the threshold. If this is not the case, consider re-running the tool with a different **Minimum peak count**.

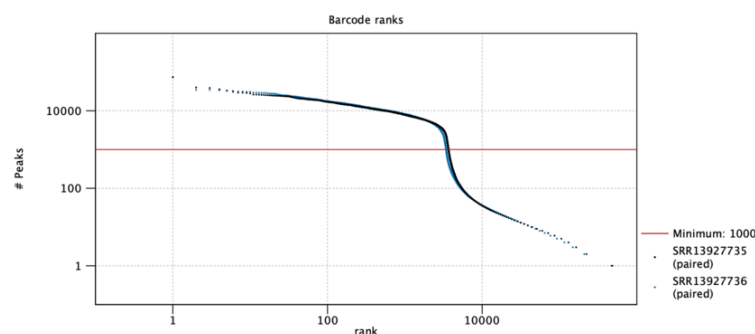


Figure 12.3: A barcode rank plot is a log-log plot of the total number of peaks for each barcode vs the rank of the barcode, in decreasing order of the number of peaks. Barcodes above the red threshold line are retained as cells. Data is for two samples from [Taavitsainen et al., 2021](#).

Two additional metrics are shown per sample:

- **Barcodes before filtering.** The total number of barcodes for each sample.
- **Cells after filtering.** The number of cells after filtering. This is the x value at which the red threshold line meets the sample line on the plot.

Peaks

A summary table is shown for all peaks:

- **Peaks.** The total number of peaks seen in the sample.
- **Peaks with nearby gene.** The number of these peaks that were annotated with a nearby gene.
- **% with nearby gene.** "Peaks with nearby gene" / "Peaks" x 100.
- **Peaks with transcription factor.** The number of peaks that were annotated with a transcription factor.
- **% with transcription factor.** "Peaks with transcription factor" / "Peaks" x 100.

Details are provided for peaks with nearby genes:

- **Peaks with nearby gene.** The number of peaks that were annotated with a nearby gene. This is the same number as in the summary table.
- **Peaks at TSS.** The number of peaks whose center was within -1000nt to +100nt of a gene's transcription start site (TSS).
- **% at TSS.** "Peaks at TSS" / "Peaks with nearby gene" x 100.

12.1.3 The Single Cell ATAC-Seq Analysis algorithm

As a first step, Single Cell ATAC-Seq Analysis de-duplicates reads if they have the same "logical" start positions and the same cell barcode. De-duplication is necessary because otherwise duplicated reads can pile up at a position and look like peaks. One read is kept from each set of duplicates. Ambiguously mapping reads and reads that are not in pairs are also discarded. All subsequent analysis is performed on this de-duplicated read mapping, which is also one of the tool's outputs.

Peaks are then called using the CLC Shape-based Peak Caller [Strino and Lappe, 2016]. As the peak caller does not explicitly depend on read coverage, there is no merging of nearby peaks to rescue regions where multiple small peaks would not meet a coverage threshold. This means that more peaks will typically be called than in approaches that merge peaks, and peaks will more often contain just 0, 1 or 2 reads per cell as expected.

Counting reads per peak and cell

It is standard practice to correct the read start site by +4bp for forward reads and -5bp for reverse reads such that the reads start at the center of the transcription factor binding site [Buenrostro et al., 2013]. Reads are counted for a peak if the corrected read start for either read in a pair is contained within the peak.

Finding transcription factors

The sequence of each peak is scanned for transcription factor binding motifs from the HOCOMOCO v11 Human Core Collection (i.e. those with quality A, B or C) [Kulakovskiy et al., 2018] using the SPRY-SARUS library. It is not possible to provide a custom motif database.

HOCOMOCO provides two types of matrices: mononucleotide, which scores positions individually, and dinucleotide, which scores two positions at a time. The use of dinucleotide matrices is preferred because their matches are more precise. For example, a mononucleotide matrix "AG(C/G)(T/C)A" might match the sequences "AGCTA", "AGGTA", "AGCCA", and "AGGCA". The equivalent dinucleotide matrix might know that the C at position 3 is always followed by a T, whereas the G is always followed by a C and so the valid matching sequences are "AGCTA" and "AGGCA". We use dinucleotide matrices when they are available, and otherwise fall back to the mononucleotide matrix.

Although only human HOCOMOCO motifs are searched, there is considerable orthology between species, such that results can be informative for other species.

Motifs are reported if they meet a score threshold corresponding to a p-value of 0.0001. All motifs exceeding this threshold are output by Single Cell ATAC-Seq Analysis in an annotation track.

A footprinting algorithm is used to detect "valleys" within the peaks that suggest the presence of transcription factor binding. The intuition is that the Tn5 transposase cannot cut the DNA at a position where a transcription factor is bound, and so the peak signal is lower where binding occurs. A "footprint" score is calculated at each position within a peak and used to filter away transcription factor binding sites for which there is little evidence of binding. The reported transcription factors for a peak are those with a footprint score above the calculated threshold.

The footprinting algorithm used is a Java re-implementation of the ATAcCorrect, scoreBigWig and BINDetect (for one sample) modules of TOBIAS [Bentsen et al., 2020]. Briefly, the insertion bias of the Tn5 transposase is learned from the cut sites in the mapped reads. The observed cut sites are then corrected for this bias. The footprinting score is calculated over a window. Better scores are obtained if the number of corrected cut sites is high in the flanks of the window and low in the center. A footprint score threshold is calculated from a background of randomly sampled positions in peaks, under the assumption that most positions in peaks do not have bound transcription factors.

There are some minor differences compared to the TOBIAS v0.12.10 implementation. The most notable is that the determination of the footprinting score threshold starts with a simple Gaussian fit as opposed to a 2-component Gaussian mixture model. This is because peak calling is performed as part of Single Cell ATAC-Seq Analysis and so there is no need to model peaks that are unobserved in the input.

Finding nearby genes

Nearby genes for a peak are likely to be regulated by that peak. Genes are assigned to peaks as follows:

1. If the peak center is within -1000nt to +100nt of a gene's transcription start site (TSS), then the gene is a promoter gene for that peak. There may be multiple promoter genes for one peak. Transcription start sites are here defined as being the first nucleotide in any of

the supplied mRNA or gene annotations.




2. If no promoter gene is found:

- we look for a TSS within -200kb to +200kb of the peak center. The closest gene (or genes) within that range are distal genes for the peak. There is usually only one closest TSS.
- we also look for genes and transcripts overlapping the peak center. Any such genes are distal genes for the peak. There may be many such genes, but usually there are none or one.

Note that it is possible to precisely control which genes and transcripts are used for finding nearby genes by providing custom gene and mRNA tracks to Single Cell ATAC-Seq Analysis.



The distinction between promoter and distal genes is only used in the Single Cell ATAC-Seq Analysis report, and on export for compatibility with third party tools. It cannot be viewed in the Peak Count Matrix.

12.2 Split Read Mapping by Cell



Split Read Mapping by Cell splits an input **Read Mapping** () according to groupings provided by **Cell Clusters** () or **Cell Annotations** (). It is available from:


Tools | Single Cell Analysis () | **Chromatin Accessibility** () | **Split Read Mapping by Cell** ()

There are two types of output:

- A **Graph Track** () suitable for visualizing scATAC-Seq peaks per grouping.
- A **Read Mapping** () per grouping, which can be used as input to Single Cell ATAC-Seq Analysis to analyze a subset of previously analyzed data.

The options control the groups of cells for which an output is produced:

- **Clusters** and **Cell annotations**. **Clusters** accepts **Cell Clusters** () and **Cell annotations** accepts **Cell Annotations** ()
- **Group by**. One or more categories from the supplied Cell Clusters or Cell Annotations. If neither is supplied, then it is only possible to group by 'Sample'. Categories that only contain non-integer numerical data are not supported. If Cell Clusters contained a category 'Cell type' with values 'T cell', 'B cell' and 'Platelet', and Cell Annotations contained a category 'Status' with values 'Case' and 'Control', then selecting **Group by = Cell type, Status** would give groups 'T cell - Case', 'T cell - Control', 'B cell - Case', 'B cell - Control', 'Platelet - Case', and 'Platelet - Control'.
- **Select groups** (Optional). This can be supplied to reduce the number of groups to only those of interest. If left empty, all groups will be output.

The tool also outputs a **Report** () summarizing the input and the resulting cell groups.

Peak graph tracks

The **Create peak graph tracks** option creates a graph of fragment coverage for each group of cells. Only paired end reads are used to create the graph - broken pairs are discarded. Fragments are corrected to the cut site by offsetting read start sites by +4nt for forward reads and -5nt for reverse reads. The peak graph track often provides a more intuitive visualization of peaks than a Read Mapping and uses much less disk space. The visualization is more intuitive because the unsequenced part of each fragment that lies between the two reads of a pair is counted towards the coverage of the peak graph, but does not count towards the coverage of the Read Mapping.

It is recommended to only create peak graph tracks on read mappings that have been produced by Single Cell ATAC-Seq Analysis, as otherwise the presence of duplicate reads can make peaks less clear.

Peak graph tracks can be scaled in two ways. Scaling does not affect the relative height of peaks within the same track, and so is only useful when comparing peaks in two different tracks:

- **No scaling.** The height of the graph track corresponds to the number of fragments sequenced at each position. With this scaling, if one track has 5 times more reads in a peak than the other, then the height of the peak will be 5 times greater. This allows the signal strength for each peak for a group of cells to be seen.
- **Scale by number of cells.** The height of each graph track is scaled by the number of cells in a group. With this scaling, if one track has 5 times more reads in a peak than the other, but also 5 times more cells in the group, then the heights of the peaks will be the same. This allows the shapes of peaks from large and small groups of cells to be compared.

To visualize the effect of scaling in a Track List, all graph tracks must be shown on the same scale. To do this, check the **Fix graph bounds** option in the Side Panel. The effect of different settings is shown in figures [12.4-12.6](#).

Reads tracks

The **Create reads tracks** option creates a Read Mapping for each group of cells. Unlike **Create peak graph tracks**, no filtering or post-processing of the reads is applied: the output includes paired end reads and broken pairs, and the original alignment coordinates are preserved (i.e. there is no correction to the cut site).

Report

The report lists how many fragments and cells were found in the input Read Mapping:

- Fragments tables will be produced separately for paired and single reads, if there are any such reads. Both paired reads and single reads count as one fragment. Note that a broken pair of reads will be listed as two separate single reads and so will count as two fragments.
- Cells are split into matched and unmatched cells. If single reads are present (for example, due to the presence of broken pairs), then the unmatched cells will be further split into cells that are unmatched because they are not part of any group, and cells that are unmatched because they have no paired reads.

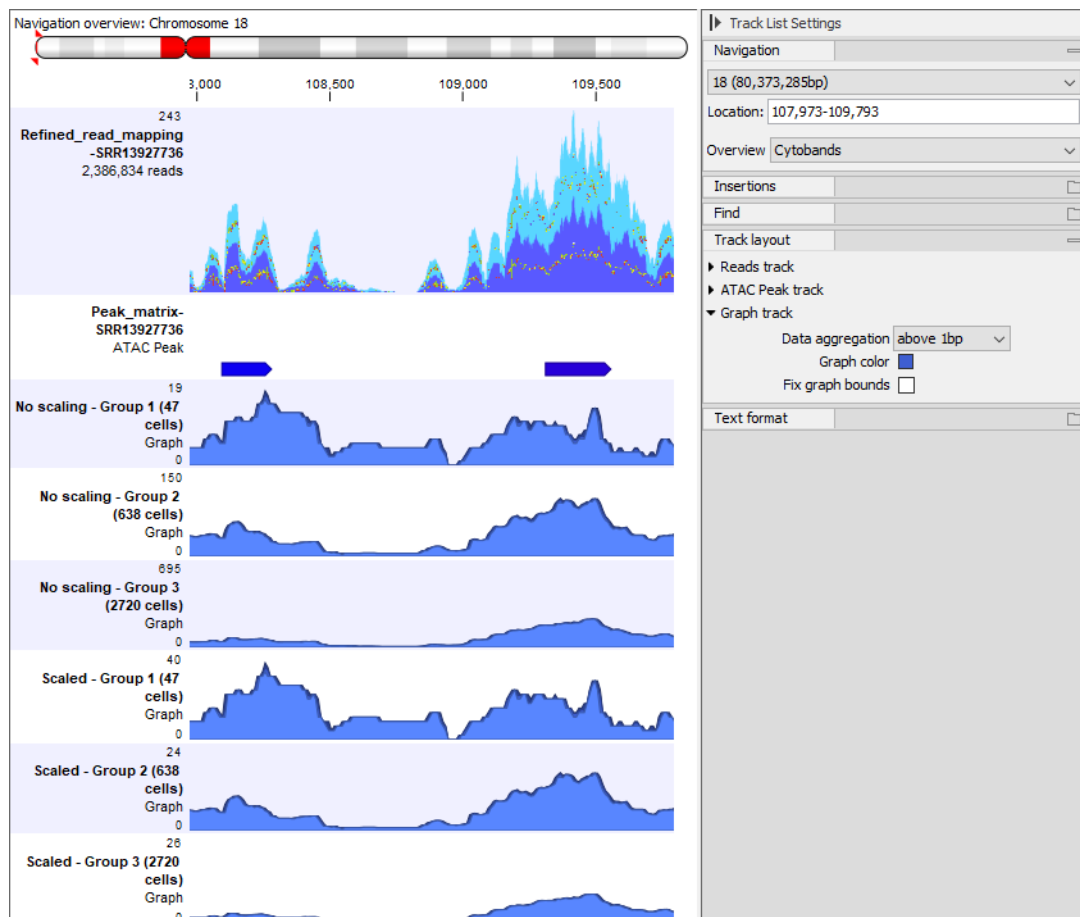


Figure 12.4: A Track List showing the Read Mapping coverage graph (top), called peaks, and peak graph tracks for three groups of cells of very different sizes. Fix graph bounds is not checked in the Side Panel, so each graph track is independently rescaled to use the available space. This means that the graph tracks for each group appear the same regardless of whether they have no scaling or are scaled by number of cells. Data is for one sample from [Taavitsainen et al., 2021](#).

For each resulting cell group, the number of cells in the group is reported.

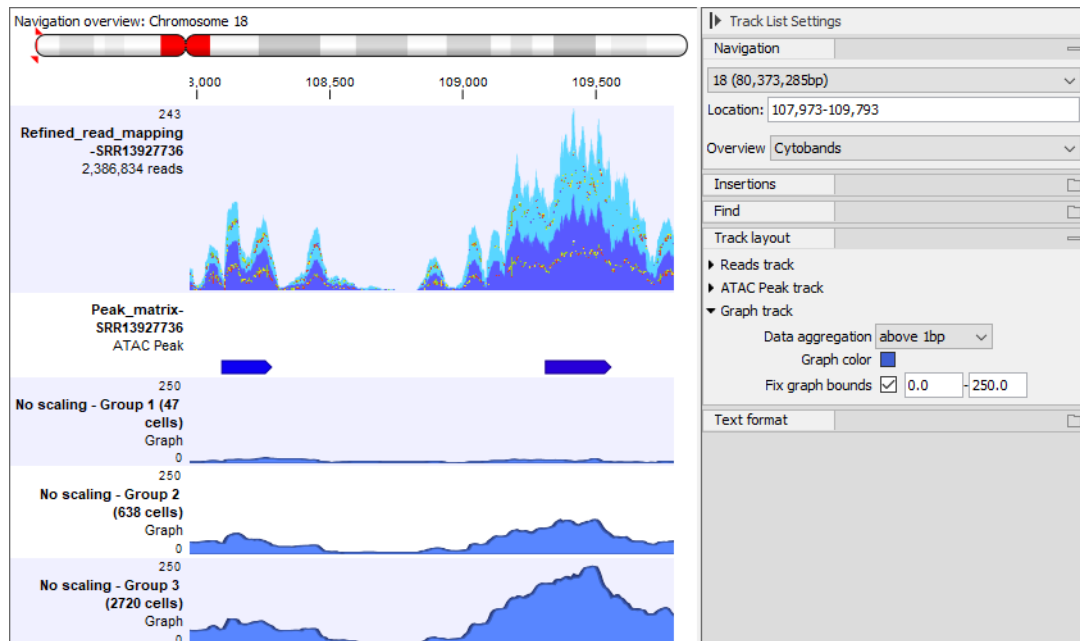


Figure 12.5: The same Track List as in figure 12.4, but only showing the graph tracks without scaling and with Fix graph bounds checked in the Side Panel. There are many more cells in group 3 than in group 1, and this is reflected by the heights of the graphs - the signal at each of the two peaks is much stronger in group 3 than in group 1.

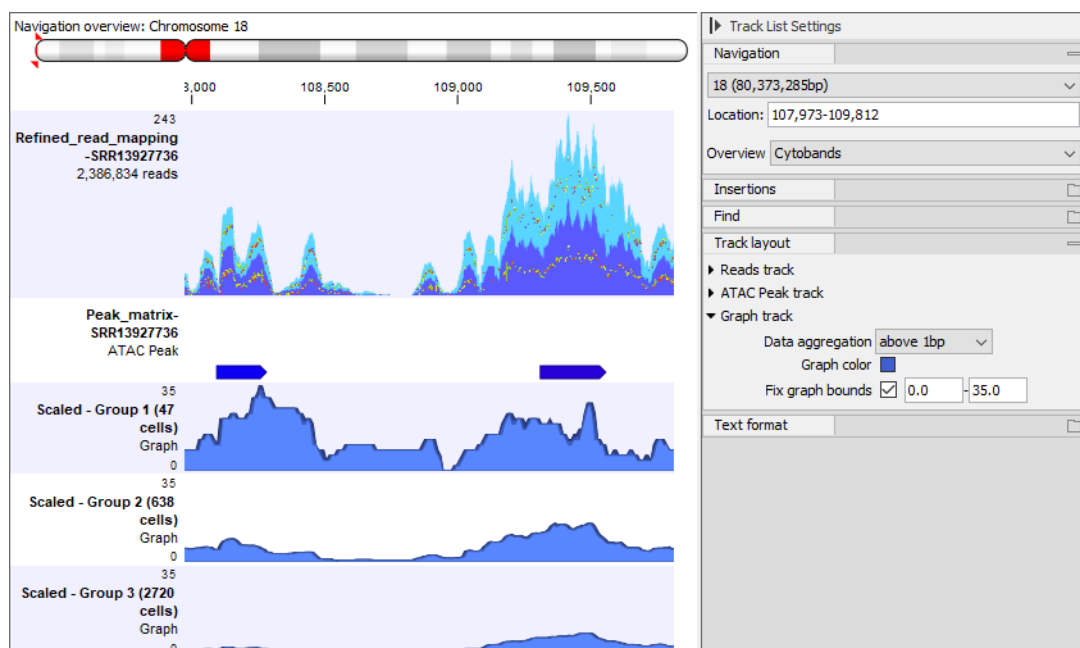






Figure 12.6: The same Track List as in figure 12.4, but only showing the graph tracks with scaling and with Fix graph bounds checked in the Side Panel. The heights of the graphs are much greater in group 1 than in group 3. This is because a greater fraction of the cells in group 1 than in group 3 have reads in the peaks.

12.3 Differential Accessibility for Single Cell

Differential Accessibility for Single Cell performs differential analysis from an input **Peak Count Matrix** () and groupings provided by **Cell Clusters** () or **Cell Annotations** ()

It is often most natural to run the tool from a Dimensionality Reduction Plot by right-clicking on the plot, see section 17 for details. However, it can also be found under the Tools menu at:

Tools | Single Cell Analysis () | **Chromatin Accessibility** () | **Differential Accessibility for Single Cell** ()

The tool performs tests for differentially accessible peaks, nearby genes or transcription factors, as specified in the ‘Data type’ options group. The tests are summarized in the output Statistical Comparison Tables () , see section 9.1.1 for details.

The remaining options specify the type of test to be performed and how features can be filtered before testing, in a similar manner as done for Differential Expression for Single Cell, see section 9.1 for details.

Note that features that are present in few cells can lead to bands in the volcano plot, showing the relationship between the p-values and the \log_2 fold changes, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Volcano_plots.html for details. Such features can span a wide range of fold changes but often have high p-values. To remove these bands, the features that are not present in sufficient cells can be filtered before testing, as detailed above.

12.3.1 The differential accessibility algorithm

The Differential Accessibility for Single Cell tool performs different types of tests for the different data types.

Peaks

As peaks are either present or not in a cell and their counts are not relevant, only the peak presence / absence is used when performing the differential accessibility test.

The observed presence / absence is modeled using logistic regression. Let Y be the presence / absence of the peak and $p = \mathbb{P}(Y = 1)$, then the form of the model for each peak is:

$$\text{logit } p = \ln \frac{p}{1-p} = \beta_0 + \beta_1 g_i + \beta_2 \log_{10} m_i ,$$

where for cell i , g_i denotes the group it belongs to, and m_i its total peak count. The total peak count is used as a proxy for the total sequencing depth of the cell.

Note that the logistic regression is applied in a pairwise fashion, where g_i is either 0 or 1.

The probability that the peak is present in a specific group $p_g = \mathbb{P}(Y_g = 1)$ is then estimated as

$$\text{logit } p_g = \beta_0 + \beta_1 \mathbf{1}_{g=1} + \beta_2 \overline{M} ,$$

where $\mathbf{1}$ is the indicator function and \overline{M} is the average $\log_{10} m_i$ over all cells.

The following are reported:

- **Max group mean.** The maximum of the two estimated probabilities.
- **Fold change.** The ratio between the two estimated probabilities.
- **P-value.** The p-value that $\beta_1 \neq 0$.

Nearby Genes and Transcription Factors

When comparing nearby genes or transcription factors, the count data is first normalized using a negative binomial (NB) generalized linear model.

The form of the model for each feature is:

$$\log \mathbb{E}(y_i) = \beta_0 + \beta_1 \log_{10} m_i ,$$

where y_i are the observed counts for the feature for a cell i . The dispersion parameter $\gamma = 1/\theta$ of the NB distribution is estimated during fitting using the Cox-Reid penalized adjusted likelihood [Robinson et al., 2010]. When $\gamma = 0$ ($\theta = \infty$), the NB distribution reduces to the Poisson distribution.

To obtain the normalized values, the Pearson residuals are calculated as follows:

$$\begin{aligned} z_i &= \frac{y_i - \exp(\beta_0 + \beta_1 \log_{10} m_i)}{\sigma} \\ &= \frac{y_i - \hat{y}_i}{\sigma} \\ &= \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i(1 + \gamma \hat{y}_i)}} \end{aligned}$$

The Pearson residuals are, however, difficult to interpret, and therefore the following is used for calculating average counts for each group:

$$\log \tilde{y}_i = \beta_0 + \beta_1 \overline{M} .$$


The following are reported for pairwise comparisons:

- **Max group mean.** The maximum of the average \tilde{y}_i of the two groups.
- **Fold change.** The ratio between the average \tilde{y}_i of the two groups.
- **P-value.** The p-value obtained from a Mann-Whitney U test (also known as Wilcoxon rank-sum test) on the Pearson residuals.

Note that when identifying markers, the reported ‘Max group mean’, ‘Fold change’ and ‘P-value’, regardless of the data type used for the test, are aggregated across all pairwise comparisons, as detailed in section 9.1.


For more details on the outputs, see section 9.1.1.

12.4 The Peak Count Matrix element

Peak Count Matrix () elements are Tracks that hold information about peaks, as well as their nearby genes and transcription factors. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tracks.html> for more information on Tracks.

Peak Count Matrix elements contain various table views, described below. See https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Working_with_tables.html for general information on working with tables.


Nearby Genes Table

The Nearby Genes Table () contains a row for each gene near a peak, and has the following columns:

- **Nearby gene.** The name of the gene.
- **Min/Max/Avg expression.** The minimum/maximum/average gene expression across all cells.
- **Nonzero values.** The number of cells with peaks near the gene.

Clicking on a row opens a separate table, listing the cells with peaks near the gene.

Peak Count Table


The Peak Count Table () contains one row for each peak, and has the following columns:

- **Name.** The name of the peak.
- **Chromosome.** The chromosome that the peak is on.
- **Region.** The region that the peak spans.
- **Nearby gene.** The genes near the peak.
- **Transcription factor.** The transcription factors for the peak.
- **Min/Max/Avg expression.** The minimum/maximum/average peak expression across all cells.
- **Cells.** The number of cells with evidence of open chromatin at the peak.

Clicking on a row opens a separate table, listing the cells with evidence of open chromatin at the peak.

To create a new matrix element from a row selection in the Peak Count Table, use the **Create Matrix from Selection** option in the right-click menu. Cells with evidence of open chromatin at one or more of the selected peaks are included.

Transcription Factors Table

The Transcription Factors Table () contains a row for each transcription factor, and has the following columns:

- **Transcription factor.** The name of the transcription factor.
- **Min/Max/Avg expression.** The minimum/maximum/average transcription factor expression across all cells.
- **Nonzero values.** The number of cells with peaks having the transcription factor.

Clicking on a row opens a separate table, listing the cells with peaks having the transcription factor.

Cell Table

The Cell Table () for a Peak Count Matrix () contains one row for each cell, and has the following columns:

- **Sample.** The sample that the cell is from.
- **Barcode.** The cell barcode.
- **Nonzero values.** The number of peaks expressed by the cell.

Clicking on a row opens a separate table, listing the peaks expressed by the cell.

To create a new matrix element from a row selection in the Cell Table, use the **Create Matrix from Selection** option in the right-click menu.

Chapter 13

Immune Repertoire

Contents

13.1 Single Cell V(D)J-Seq Analysis	168
13.1.1 The report output from Single Cell V(D)J-Seq Analysis	169
13.1.2 The clonotype identification algorithm	170
13.2 Filter Cell Clonotypes	172
13.3 Combine Cell Clonotypes	174
13.4 Compare Cell Clonotypes	175
13.4.1 The output of Compare Cell Clonotypes	176
13.5 Convert Clonotypes to Cell Annotations	177
13.6 The Cell Clonotypes element	179
13.6.1 Primary and secondary clonotypes	179
13.6.2 Cell Clonotypes tables	180
13.6.3 Cell Clonotypes alignments	182
13.6.4 Cell Clonotypes Sankey plot	185

T and B cells form our acquired immune response. They both contain highly variable receptors (see figure 13.1) with binding sites that recognize antigens.

T and B cell receptors (TCR and BCR, respectively) are composed of multiple polypeptide chains: TCRs contain one pair, while BCRs contain two copies of a pair:

- TCR: α (TRA) and β (TRB), or γ (TRG) and δ (TRD).
- BCR: light and heavy. There are two types of light chains in humans: κ (IGK) and λ (IGL), while other animals also contain other types of light chains. Once set, the light chain class remains fixed for the life of the B cell. There are five types of heavy chains (IGH) for mammals: γ , δ , α , μ and η , defining the class of the receptor.

The chains are encoded by genes that undergo somatic recombination. During this process, gene segments are joined with random nucleotides at the junction sites. There are two types of recombination (see figure 13.2):

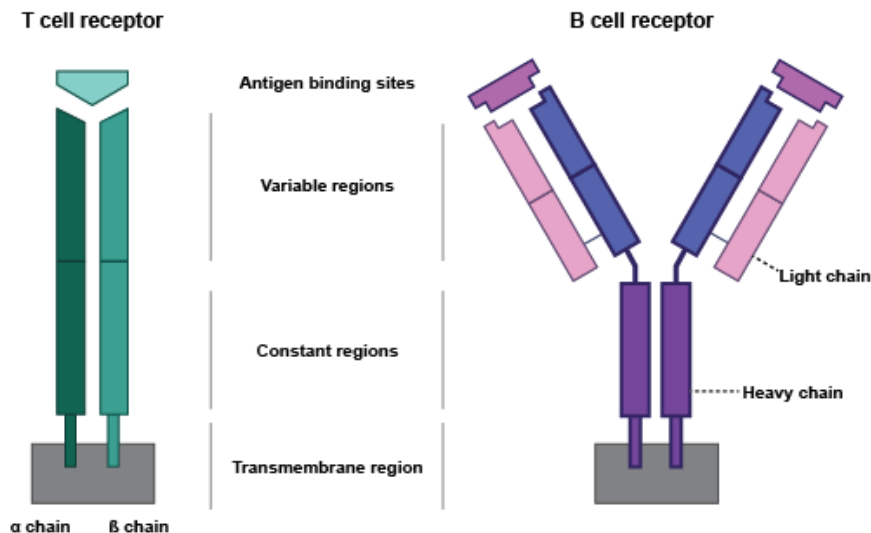


Figure 13.1: *T and B cell receptors structure. Each pair of chains forms an antigen binding site that binds to specific antigens.*

- VJ recombination, where one V (variable) gene segment is joined to a J (joining) gene segment;
- VDJ recombination, where a D (diversity) gene segment is added between the V and J gene segments.

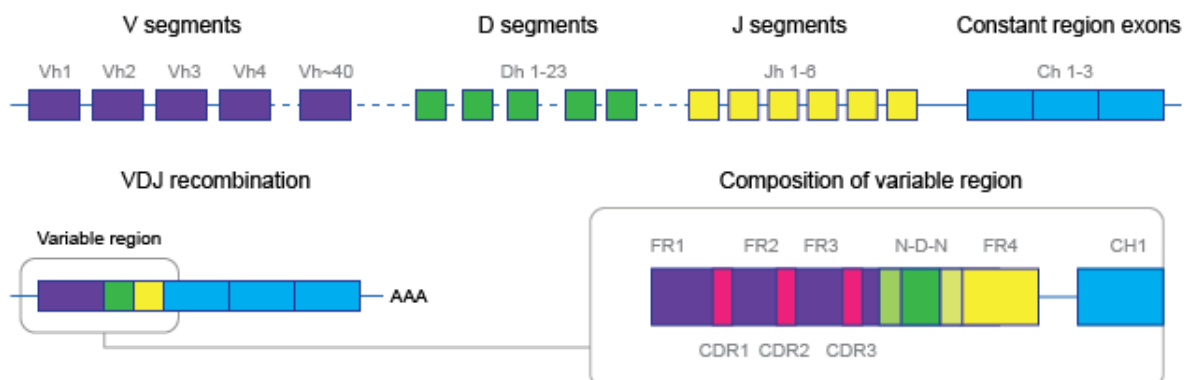


Figure 13.2: *VDJ recombination brings together a V, D, J and C gene segment.*

For both types of recombination, a C (constant) gene segment is also added following the J segment.

The TRA and TRG chains are the result of VJ recombination, while the TRB and TRD chains are the result of VDJ recombination. The V, D, J and C gene segments are specific for each TCR chain type.

BCR light chains are the result of VJ recombination, while BCR heavy chains are the result of VDJ recombination. BCR heavy chains have three to four C gene segments. The V, D, J and C gene segments are specific for each BCR light chain type, while they are shared by the BCR heavy chains.

Each chain contains three "complementary-determining regions" (CDRs) (figure 13.2) which form loops in the antigen binding sites. The V(D)J recombination junction is located in the third CDR (CDR3). Due to inclusion of random nucleotides at the junctions between segments, the CDR3 is the most diverse among the three CDRs. Its beginning and end are marked by a conserved cysteine (C) and phenylalanine/tryptophan (F/W) amino-acid in the V and J segments, respectively.




CLC Single Cell Analysis Module offers tools to clonotype reads and characterize the T or B cell receptor repertoire (section 13.1), filter the repertoires (section 13.2), combine them across samples (section 13.3), compare them (section 13.4) and convert them to cell annotations (section 13.5) for easy visualization on Dimensionality Reduction Plots.

Here, clonotyping consists of identifying which V, D, J and C segments from the reference data (section 13.1) are used, and extracting the CDR3 region found between the conserved amino acids.

13.1 Single Cell V(D)J-Seq Analysis

Single Cell V(D)J-Seq Analysis is available from:

Tools | Single Cell Analysis  | **Immune Repertoire**  | **Single Cell V(D)J-Seq Analysis** 

The tool takes as input one or more Sequence Lists  of reads that have been annotated using **Annotate Single Cell Reads**. It outputs a **TCR Cell Clonotypes**  or **BCR Cell Clonotypes**  element (see section 13.6), and optionally a report.

Sample: All input sequence lists must originate from the same sample, which is set when executing the **Annotate Single Cell Reads** tool (see section 6.1). This is because **Single Cell V(D)J-Seq Analysis** assumes that reads with the same cell barcode that are present in different inputs represent the same cell. The wizard does not allow executing the tool with inputs that are annotated with different samples.

It is important to provide all the data for a sample to **Single Cell V(D)J-Seq Analysis** at the same time. For example, if one sample was sequenced on 4 lanes of an Illumina sequencer, then all 4 lanes should be supplied together. This allows reads originating from the same cell, but coming from different lanes, to be analyzed jointly and leads to a more accurate clonotype identification.

Barcode whitelists: In some protocols, the set of valid barcodes is known in advance, and available as a barcode whitelist. In CLC Single Cell Analysis Module, it is not possible to directly use such a list. Instead, the **Filter Cell Clonotypes** can be used for filtering the Cell Clonotypes output such that only barcodes that are identified as cells are retained, such as those identified as cells in matched scRNA-Seq data. Additionally, the Filter Cell Clonotypes can be used for retaining only the desired types of clonotypes, for example only those that are productive. See section 13.2 for details.

Note: Different runs can result in slightly different results. This is caused by multi-threading of the program combined with the use of probabilistic data structures. The overall content of the Cell Clonotypes should not be markedly different.

The following options can be adjusted (figure 13.3):

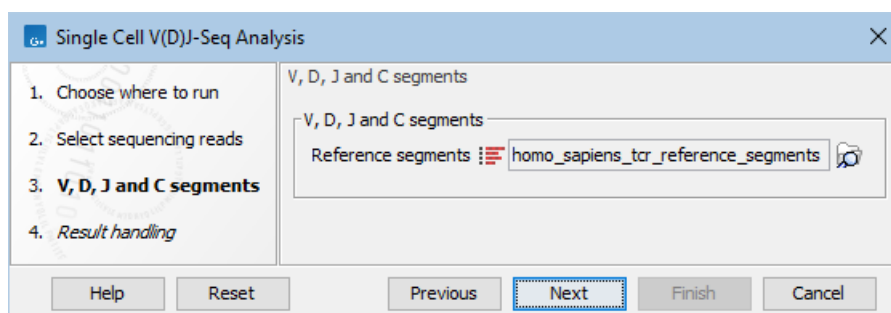


Figure 13.3: The options in the dialog of the Single Cell V(D)J-Seq Analysis tool. Human reference data downloaded from the Reference Data Manager has been selected.

- **Reference segments.** The V (variable), D (diversity), J (joining) and C (constant) segments. These are used during clonotype identification and determine whether the tool outputs a **TCR Cell Clonotypes** (🧬) or **BCR Cell Clonotypes** (🧬) element. The reference segments can either be
 - imported using **Import Immune Reference Segments** (see section 4.1);
 - downloaded from the Reference Data Manager (see chapter 2).

13.1.1 The report output from Single Cell V(D)J-Seq Analysis

The optional report includes information for different chain types:

- For TCR Cell Clonotypes: TRA + TRB, TRA, TRB, TRG + TRD, TRG, and TRD;
- For BCR Cell Clonotypes: IGH + IGK, IGH + IGL, IGH, IGK, and IGL.

Only the chain types that are found in the Cell Clonotypes are present in the report.

The following information is provided for each different chain type:

- **Summary.** Summary tables with information about the performed assembly, trimming and identified clonotypes. See section 13.1.2 for more details.
- **Diversity indices.** Several diversity indices, as listed below. The extrapolated diversity gives a projection of what the diversity would have been if the sample had been sequenced deeply enough to identify all clonotypes.
 - Distinct clonotypes: The number of different clonotypes detected.
 - Extrapolated diversity (chaoE): The extrapolated number of detected distinct clonotypes as described in [Chao, 1987].

- Lorenz curve at 50% of total: The fraction of all detected clonotypes that account for 50% of the total count. Also sometimes denoted as D50.
- Inverse Simpson's index: Let c_i denote the count for the i th distinct clonotype and let $n = \sum_i c_i$. Then the inverse Simpson's index is defined as:

$$\sum_i \frac{1}{c_i/n}.$$

- Extrapolated Inverse Simpson's index (chaoE): The extrapolated inverse Simpson's index as described in [Chao et al., 2014].
- Shannon-Wiener index: With c_i and n defined as above, the Shannon-Wiener index is defined as:

$$\sum_i \frac{c_i}{n} \ln \left(\frac{c_i}{n} \right).$$

- Extrapolated Shannon-Wiener index (chaoE): The extrapolated Shannon-Wiener index as described in [Chao et al., 2013].
- **Rarefaction.** Rarefaction curves, also known as species accumulation curves. They show the expected number of distinct clonotypes discovered as a function of the total number of detected clonotypes, together with the confidence interval (CI), obtained from a normal approximation. The curve is
 - interpolated down to 0 clonotypes;
 - extrapolated to twice the total number of detected clonotypes.
- **CDR3 length.** The distribution of the length of the CDR3 nucleotide sequences for all detected clonotypes. Peaks are expected every 3 nucleotides due to repertoires consisting predominantly of in-frame CDR3 sequences.
- **V, D, J and C usage.** Bar plots showing the V, D, J and C segment usage for all detected clonotypes.
- **Frequencies.** The percentage of all detected clonotypes that are unique and the clonotype abundance: how many distinct clonotypes are found with abundance (count) i . Most clonotypes are expected to be unique, so the percentage is close to 100% and most clonotypes have abundance 1.
- **Productive summary.** The percentage of all detected clonotypes that have productive CDR3 nucleotide sequences, and the percentage of barcodes with at least one productive CDR3 nucleotide sequence.

Note that for diversity indices and rarefaction, the number of distinct clonotypes is used. For the rest of the report, the number of detected clonotypes contains all clonotypes for all barcodes, where if more than one barcode has the same clonotype, this is counted multiple times.

13.1.2 The clonotype identification algorithm

The algorithm for identifying the clonotypes is composed of three sequential steps described below.

Assembly

All reads originating from the same barcode are collected and:

- Barcodes containing ambiguous nucleotides are discarded.
- Barcodes with less than 5 UMIs are discarded.
- Barcodes with more than 80,000 reads are down-sampled to about 80,000 reads.
- Remaining reads are de novo assembled into contigs.
- Contigs shorter than 60 nucleotides are discarded.
- The reads are mapped back to the valid contigs and the contigs are adjusted by the mapped reads.
- Contigs and barcodes of low quality are discarded. The following are required:
 - Contigs should have an average coverage of at least 5.
 - Contigs should have at least 20 mapped reads.
 - If more than four contigs are assembled, contigs should have an average coverage of at least the median average coverage of all contigs.
 - Barcodes should have at least 3 UMIs mapped to high-quality contigs.

The assemble summary reports:

- the number of input barcodes and reads;
- the number of processed barcodes (those without ambiguous nucleotides) and reads (those left after down-sampling);
- the number of barcodes that have been discarded;
- the number of barcodes and high-quality contigs that have been successfully assembled;

Trimming

Prior to clonotype identification, the contigs are trimmed with the following settings:

- The ends of the contigs are trimmed using 0.05 "Quality limit" and 2 "Maximum number of ambiguities", see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Quality_trimming.html.
- Contigs shorter than 60 nucleotides after trimming are discarded.

The trimming summary reports the average length of the contigs before and after trimming, and how many barcodes and contigs remain after trimming.

Clonotype identification

Clonotyping a contig consists of identifying which V, D, J and C segments from the reference data are used, and extracting the CDR3 region found between the conserved amino acids.

The identification of the segments is done by mapping the contigs against the references provided in "Reference segments".

Depending on the length and diversity of the segment that is covered by the contig, it might not be possible to unambiguously detect the segment. In this case, all possible segments are reported.

The V and J segments are required for successfully clonotyping a read, because otherwise the CDR3 cannot be determined.



The D and C segments are optional. Note that the (lack of) identification of these two segment types can lead to the tool reporting clonotypes as the same or different clonotypes:

- If two cells have the same assigned V and J segments and share the CDR3 sequence, they would typically be considered to have the same clonotype. However, if for one cell the C segment is successfully identified, but the contigs for the other cell did not cover the C segment, their two clonotypes will be reported separately.
- If two contigs for the same cell have the same assigned V and J segments and a CDR3 sequence that is almost the same, they would typically be merged and be considered to have the same clonotype (see below). However, due to the non-identical CDR3 sequence, one contig might have a D segment assigned, while the other might not, hence the two clonotypes will be considered to be distinct.

After the initial clonotyping of the contigs, merging of clonotypes identified for the same barcode is performed as follows:

- If a clonotype has ambiguously assigned segments, it will be merged, if possible, into a clonotype with the same CDR3 and less ambiguous segments that are a subset of the former clonotype's segments.
- If two clonotypes exist with the same segments, but differing by a single nucleotide in the CDR3 sequence, the clonotype with fewer contigs will be merged into the other.

13.2 Filter Cell Clonotypes

Sometimes it can be desirable to restrict **TCR Cell Clonotypes**  or **BCR Cell Clonotypes**  to only a specific subset, for example only productive clonotypes, or only barcodes that are present in matched scRNA-Seq data. This can be achieved with the Filter Cell Clonotypes tool. Alternatively, the clonotypes can be filtered to a selection in the Cell Clonotypes tables (section 13.6.2) by using the **Create Clonotypes from Selection** option in the right-click menu.

The Filter Cell Clonotypes is available from:

Tools | Single Cell Analysis (🧬) | Immune Repertoire (🧬) | Filter Cell Clonotypes (🧬)

The tool takes a Cell Clonotypes element as input and produces a filtered element.

The following options can be adjusted (figure 13.4):

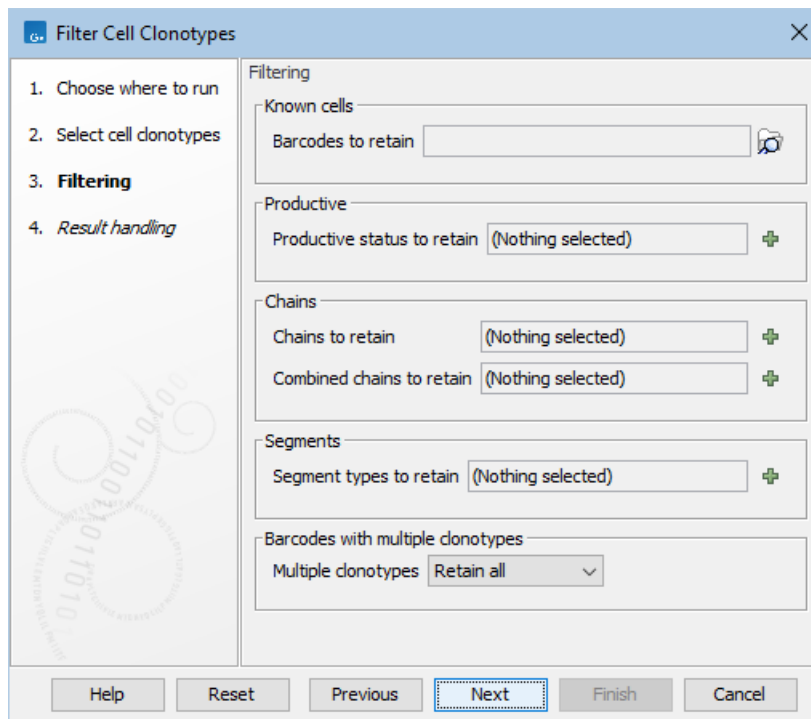


Figure 13.4: The options in the dialog of the Filter Cell Clonotypes tool.

- **Barcodes to retain.** Multiple elements containing cells can be provided, such as Expression Matrices, Cell Clusters and Cell Annotations. From these, a set of valid cells, identified through the sample and barcode, is obtained as the intersection of the cells in the chosen elements. When used, only the clonotypes for the valid cells are retained in the output.
- **Productive status to retain.** A mixture of 'Productive', 'Out of frame' and 'Premature stop codon' can be chosen and only the clonotypes with the respective productive status will be retained. If left empty, no filter is applied.

- **Chains to retain.** A mixture of:

- for TCR Cell Clonotypes: TRA, TRB, TRG and TRD;
- for BCR Cell Clonotypes: IGH, IGK and IGL;

can be chosen. Only the clonotypes with the respective chains will be retained. If left empty, no filter is applied.

- **Combined chains to retain.** A mixture of:

- for TCR Cell Clonotypes: TRA + TRB and TRG + TRG;
- for BCR Cell Clonotypes: IGH + IGK and IGH + IGL;

can be chosen. Only clonotypes that contain all chains in the combination are retained. This can be used for removing barcodes for which not all desired chains have been identified. If left empty, no filter is applied.

- **Segment types to retain.** A mixture of 'V', 'D', 'J' and 'C' can be chosen and only the clonotypes that have identified segments for all respective segment types will be retained. This means that, for example, if 'D' is chosen, only chains for which the D segment is used will be retained, and for those chains, only the clonotypes for which the identification of the D segment was successful will be retained. If left empty, no filter is applied.
- **Multiple clonotypes.** Barcodes can have more than one clonotype associated with them, see section 13.6.1. Different types of filters can be chosen:
 - **Retain all.** No filter is applied and all clonotypes are retained.
 - **Retain primary.** Only the primary clonotypes are retained.
 - **Retain secondary.** Only the secondary clonotypes are retained.
 - **Retain primary and secondary.** Both the primary and the secondary clonotypes are retained.
 - **Retain none.** Only barcodes containing just primary clonotypes are retained.

The options above can be mixed and matched to obtain the desired output. Note that the filters are applied in the order given above.

For example, assume we want to only use the primary TRB clonotypes with D segments. This can be obtained by setting "Chains to retain" to "TRB", "Segment types to retain" to "D", and "Multiple clonotypes" to "Retain primary". If one barcode A has a primary TRB clonotype without D segments and a secondary TRB clonotype with D segments, the former will be removed first and the second with D segments will become the primary one. Hence, "Retain primary" will have no effect on this barcode. If another barcode has two TRB clonotypes with D segments, "Retain primary" will remove the secondary clonotype.

If the desired behavior is that barcode A should be entirely removed from the output, as its primary TRB clonotype does not have D segments, the tool can be run multiple times such that the filters are applied in a different order. By running the tool with "Multiple clonotypes" set to "Retain primary" first, the barcode will have only the clonotype without D segments. A second execution of the tool with "Segment types to retain" set to "D" will entirely remove the barcode.



The Filter Cell Clonotypes tool can optionally produce a report for each sample found in the input element, summarizing the clonotypes left after filtering. The output report includes the same information as the report produced by the Single Cell V(D)J-Seq Analysis tool, minus the assembly and trimming summaries (see section 13.1.1).

To obtain a report summarizing clonotypes across samples, see section 13.4.

13.3 Combine Cell Clonotypes

The Combine Cell Clonotypes tool is available from:

Tools | Single Cell Analysis  | **Immune Repertoire**  | **Combine Cell Clonotypes** 

The tool takes as input multiple **TCR Cell Clonotypes**  or **BCR Cell Clonotypes**  elements and outputs a single Cell Clonotypes element. This can be useful to reduce the number of elements needed to describe a set of cells.

Note that TCR Cell Clonotypes and BCR Cell Clonotypes cannot be mixed and only one type should be used at a time.



The tool is very flexible and it supports:

- Different clonotypes for the same cells.
- Clonotypes for different cells.

Cells are considered to be the same if they have the same sample and barcode.



If different clonotypes are identified for the same cell, they will all be collected in the output. While a cell can have up to two different clonotypes for the same chain type (see section 13.6.1), this tool can lead to barcodes having an arbitrary number of clonotypes. Clonotypes marked as "Subsequent" are biologically unlikely and if the output Cell Clonotypes element contains subsequent clonotypes, it is an indication that the incorrect elements have been combined. Subsequent clonotypes can be removed using the Filter Cell Clonotypes tool, see section 13.2.

13.4 Compare Cell Clonotypes



Compare Cell Clonotypes contrasts properties, such as diversity and similarity, of the immune repertoires identified for groups of cells, as determined by the sample or, when available, through **Cell Clusters**  or **Cell Annotations** .

The Compare Cell Clonotypes tool is available from:

Tools | Single Cell Analysis  | **Immune Repertoire**  | **Compare Cell Clonotypes** 

The tool takes a **TCR Cell Clonotypes**  or **BCR Cell Clonotypes**  element as input. If the clonotypes to be compared are found in different elements, these can be combined using the Combine Cell Clonotypes tool, see section 13.3 for details.

The following options can be adjusted (figure 13.5):

- **Clusters** and **Cell annotations** (Optional). **Clusters** accepts **Cell Clusters**  and **Cell annotations** accepts **Cell Annotations** . These can be created within the CLC Single Cell Analysis Module for scRNA-Seq data, but they can also be imported, see section 4.2 and section 4.3 for details.

- **Group by**. Any categories from the supplied Cell Clusters or Cell Annotations. "Sample" can be additionally chosen, even if no Cell Clusters or Cell Annotations are provided.

If Cell Annotations contained a category 'Infection' with values 'Pre' and 'Post', and another category 'Individual' with values 'Ind1' and 'Ind2', then selecting **Group by = Infection, Individual** would give groups 'Pre - Ind1', 'Pre - Ind2', 'Post - Ind1' and 'Post - Ind2'.

The chosen options need to induce at least two groups of cells. Otherwise, the tool will fail with a relevant message.

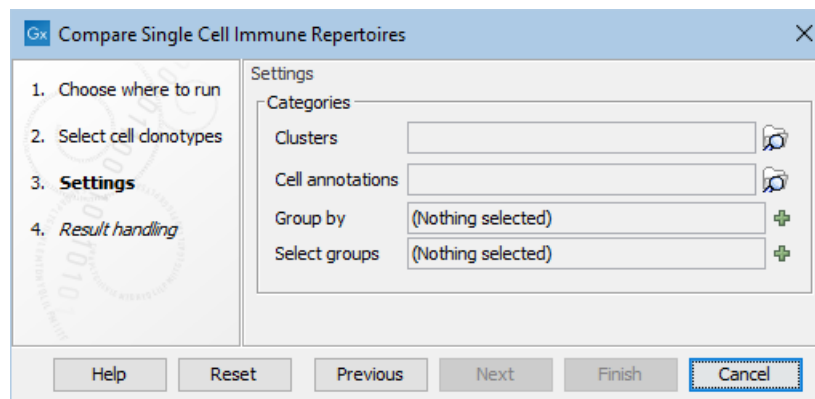


Figure 13.5: The default options in the dialog of the Compare Cell Clonotypes tool. Note that "Group by" needs to contain at least one value before proceeding. This can always be set to "Sample".

Note that category from Cell Annotations that only contain non-integer numerical data are not supported.

- **Select groups** (Optional). This can be supplied to reduce the number of groups of cells in the outputs to only those of interest, or to control the order in which the groups are shown. For example, if the aim is to investigate the infection effect in individual 1, the 'Pre - Ind1' and 'Post - Ind1' groups can be selected. If left empty, all groups will be used.

13.4.1 The output of Compare Cell Clonotypes

Compare Cell Clonotypes produces a report contrasting the immune repertoire properties, and optionally, a heat map and/or table summarizing the similarity of the immune repertoires.

The report

The report contains some of the same information provided in section 13.1.1, for each group of cells as defined by the configured options. It additionally contains:

- **Sample composition summary.** Information about the samples identified in groups of cells.
- **Interpolated to lowest group diversity.** Diversity index interpolated to the lowest number of clonotypes observed among all groups.
- **CDR3 length.** Tables containing summary statistics of the distributions.

Note that diversity is not reported for groups of cells containing more than one sample.

When a group of cells has a name that is too long to be suitable for figure legends, numbers are used in the legend, and the mapping between the numbers and the group names is listed below each figure.

If the "Group by" option leads to more than nine groups of cells, the figures will not have legends. The underlying information can be recovered by double-clicking on the desired figure and switching to the table view.

Heat map

For each pair of groups, the weighted Jaccard similarity between the two is computed. Let X_i, Y_i denote the relative frequencies of the i 'th clonotype in the first and second group respectively. The weighted Jaccard similarity is defined as:

$$J(X, Y) = \frac{\sum_{i=1}^n \min(X_i, Y_i)}{\sum_{i=1}^n \max(X_i, Y_i)}. \quad (13.1)$$

The weighted Jaccard distance is defined as:

$$D(X, Y) = 1 - J(X, Y).$$

The heat map is obtained using the Jaccard distance, where groups are clustered hierarchically.




Similarity table

A table showing the Jaccard similarity (eq. 13.1) between each pair of groups.

13.5 Convert Clonotypes to Cell Annotations

The Convert Clonotypes to Cell Annotations tool is available from:


Tools | Single Cell Analysis  | **Immune Repertoire**  | **Convert Clonotypes to Cell Annotations** 

The tool takes a **TCR Cell Clonotypes**  or **BCR Cell Clonotypes**  element as input and produces a single **Cell Annotations**  element summarizing the clonotypes.

The output contains multiple categories that summarize the primary clonotypes for each barcode (see section 13.6.1). For converting secondary clonotypes, run first Filter Cell Clonotypes with 'Multiple clonotypes' set to 'Retain secondary', see section 13.2 for details.

The cells can be colored by any of the available categories in a Dimensionality Reduction Plot (see chapter 16) obtained from matched scRNA-Seq data for the same cells (see figure 13.7).

The cells in the Dimensionality Reduction Plot and those in the Cell Annotations need to have the same sample name. Ideally, it should be ensured that these share the sample name as a first step in the analysis pipeline, when running the Annotate Single Cell Reads tool (see section 6.1), or when importing the Cell Clonotypes element (see section 4.4). If this has not been done, the sample name can be updated using the Update Single Cell Sample Name tool (see section 18.7).

The **Cell Annotations**  element contains the **Clone size** for each barcode: the number of barcodes it shares its primary clonotype with. Let us consider the following example with TCR clonotypes, where each column represents one barcode, and the rows identify the TRA and TRB clonotypes by name:

	B1	B2	B3
Primary TRA clonotype	TRA-1	TRA-2	TRA-1
Primary TRB clonotype	TRB-1	TRB-2	None
Secondary TRA clonotype	TRA-2	None	None
Secondary TRB clonotype	TRB-2	None	None

Each barcode in this example has a clone size of one, because they are not sharing the primary TRA + TRB clonotypes with any other barcode, even though they have TRA and TRB clonotypes in common. To obtain the clone size for only one chain at a time, run first Filter Cell Clonotypes with 'Chains to retain' set accordingly, see section 13.2.

For each identified chain, the **Cell Annotations**  element additionally contains the following categories with information about the primary clonotype (see figure 13.6):

- productive status;
- V, D, J and C segments;
- CDR3 length;
- the CDR3 amino acid sequence;
- the number of UMIs and reads supporting the clonotype (only for single chains).

The possible identified chains are:

- chain combinations TRA + TRB and TRG + TRD for T cells;
- individual chains TRA, TRB, TRG and TRD for T cells;
- chain combinations IGH + IGK and IGH + IGL for B cells;
- individual chains IGH, IGK and IGL for B cells.

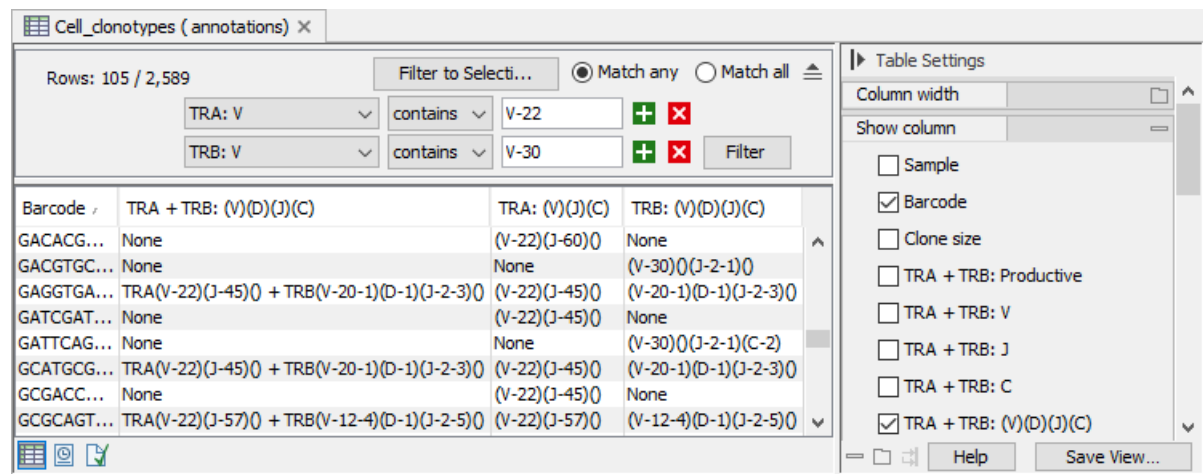


Figure 13.6: View of Convert Clonotypes to Cell Annotations output from a TCR Cell Clonotypes, filtered to specific V segments and where the segments are shown for the primary clonotypes and the TRA + TRB, TRA and TRB chains. Not all barcodes have identified clonotypes for both TRA and TRB chains.

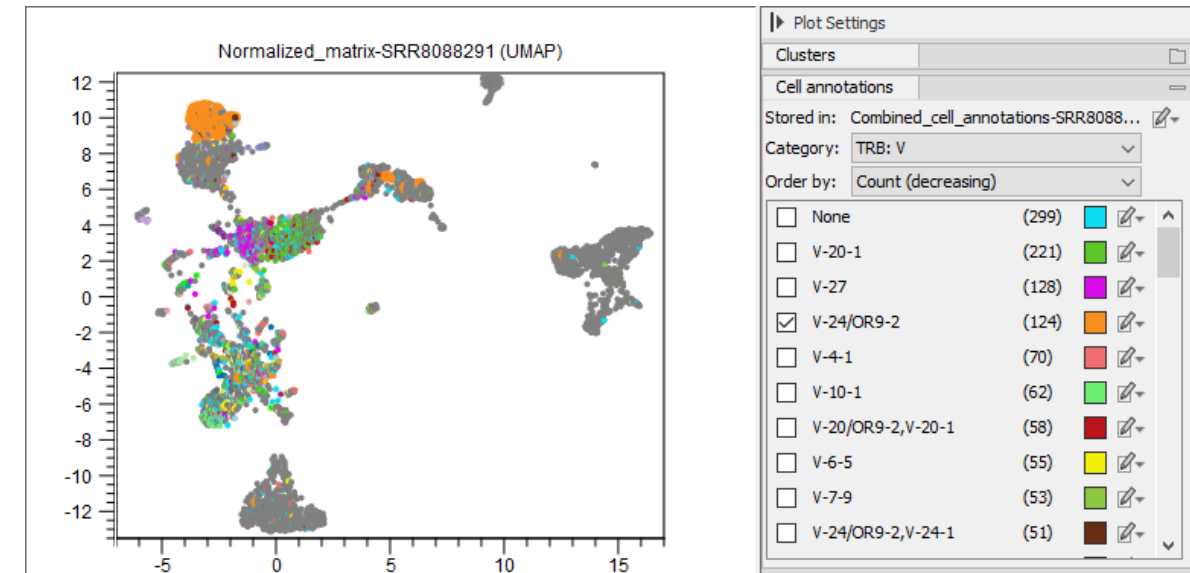


Figure 13.7: UMAP view of scRNA-Seq data, where cells are colored by the V segment from the TRB chain.

13.6 The Cell Clonotypes element

Both **TCR Cell Clonotypes** and **BCR Cell Clonotypes** elements contain the clonotypes and have a number of views, displaying different properties / summaries of the clonotypes.

13.6.1 Primary and secondary clonotypes

During the somatic recombination, the two chromosomes recombine independently, and typically this process leads to only one functional gene for each chain type. The second copy of the gene, most often non-functional, can still be expressed and captured by scV(D)J-Seq. Hence, for each cell, it is possible for up to two different copies of the same chain type to be present in a cell:

- **Primary:** The copy that leads to a productive CDR3. If no productive CDR3s are identified, then the copy with the highest UMI count.
- **Secondary:** The second copy, if present.

For example:

- If a T cell contains two productive TRB chains, the chain with the highest number of UMIs will be part of the primary clonotype, while the chain with the lowest number will be part of the secondary clonotype.
- If a B cell contains two light chains and only one is productive, the productive chain will be part of the primary clonotype, while the unproductive chain will be part of the secondary clonotype.

The Cell Clonotypes created by the Single Cell V(D)J-Seq Analysis tool only contain primary / secondary clonotypes. Imported clonotypes (see section 4.4) and those produced by Combine Cell Clonotypes (see section 13.3) can have barcodes with more clonotypes. These additional clonotypes are presented as "Subsequent" and are biologically unlikely. They can be removed using the Filter Cell Clonotypes, see section 13.2 for details.

13.6.2 Cell Clonotypes tables

The Cell Clonotypes elements contain two table views, centered around the clonotypes (📊) and barcodes (📊). Multiple clonotypes can be identified for a barcode, where each clonotype is for one chain. These are the cell-level clonotypes (📊). Clonotypes (📊) can have more than one chain (see below). For a given chain, the clonotype consists of multiple cell-level clonotypes with the same characteristics. The different chains present in one clonotype are supported by the corresponding cell-level clonotypes being present in the same barcode.

Both views contain the following information (see figure 13.8):

- **Clonotype #.** A unique number identifying the clonotype.
- **Chain:** Which chain the clonotype belongs to. Can be:
 - For TCR Cell Clonotypes: TRA, TRB, TRA + TRB, TRG, TRD, and TRG + TRD;
 - For BCR Cell Clonotypes: IGH, IGK, IGL, IGH + IGK, and IGH + IGL.

Cell-level clonotypes have only one chain.

- **V / D / J / C.** The identified V, D, J and C reference segment(s), respectively. If a single unambiguous segment cannot be identified, the segments are separated by a comma.
- **CDR3 nucleotide sequence.** The nucleotide sequence for CDR3 including the V- and J region-encoded conserved motifs.
- **CDR3 amino acid sequence.** The translated amino acid sequence for the CDR3 nucleotide sequence, provided that it is in-frame.
- **CDR3 length.** The length of the CDR3 nucleotide sequence.

Top View: Rows: 2,753

C...	Chain	V	D	J	C	Barcodes
9	TRA + TRB	TRAV-12-2 + TRBV-20-1	TRBD-1	TRAJ-42 + TRBJ-2-3	None + TRBC-2	78
10	TRB	TRBV-24/OR9-2	TRBD-1	TRBJ-1-4	TRBC-1	67
11	TRA	TRAV-12-2		TRAJ-42	TRAC	61

Rows: 156

Cell-level clonotype #	Sample	Barcode	Chain	Reads	UMIs
856	Renal tumor	AAAGATGAGATAGGGT	TRA	30	14
7126	Renal tumor	AAAGATGAGATAGGGT	TRB	23	19

Bottom View: Rows: 8,937

Cell-level clonotype #	Barcode	Clonotype #	Chain	V	D	J	C	UMIs
2400	AAAGATGAGATACACA	6	TRA	V-22		J-45		2
6653	AAAGATGAGATACACA	5	TRB	V-20-1	D-1	J-2-3		31
856	AAAGATGAGATAGGGT	9	TRA	V-12-2		J-42		14
7126	AAAGATGAGATAGGGT	9	TRB	V-20-1	D-1	J-2-3	C-2	19
2244	AAAGATGAGATCGGGT	802	TRA	V-20		J-42	C	35
3501	AAAGATGAGATCGGGT	802	TRB	V-3-1	D-1	J-2-7	C-2	48
7616	AAAGATGAGATCGGGT	47	TRB	V-20...	D-1	J-2-3		75

Figure 13.8: Views of the same TCR Cell Clonotypes element. Note that not all table columns are shown. Clonotype with number 9 is highlighted in both views. Top: View centered around the identified clonotypes, sorted after the number of barcodes. All identified chains for the clonotype are shown. For example, clonotype 9 contains both a TRA and TRB chain, while clonotypes 10 and 11 contain only a TRB and TRA chain, respectively. When a clonotype is selected, a second table lists the barcodes with the corresponding clonotype. Bottom: View centered around the barcodes, sorted by barcode. Rows with the same barcode have the same background color when the table is sorted after the barcode.

- **Productive.** One of three categories are used to characterize the CDR3 nucleotide sequence:
 - *Productive.* Sequences that are in frame and do not contain a premature stop codon.
 - *Out-of-frame.* Sequences that have a length that is not a multiple of three.
 - *Premature stop codon.* Sequences that contain an in-frame premature stop codon.

Note that the Filter Cell Clonotypes can be used for retaining only the productive clonotypes, see section 13.2 for details.

The view centered around the identified clonotypes (📊) additionally contains:

- **Barcodes.** The number of barcodes with the given clonotype.
- **Primary (%).** The percentage of clonotypes that were the primary clonotype for their respective barcode, see section 13.6.1.


Clicking on a row in this view opens a new table listing the corresponding barcodes (see figure 13.8).

The cell-level clonotypes view (📄) also provides (see figure 13.8):

- **Cell-level clonotype #.** A unique number identifying the barcode and clonotype.
- **Sample / Barcode.** The sample and barcode.
- **Reads.** The number of reads from the barcode that mapped to the contig from which the specific CDR3 sequence was detected.
- **UMIs.** The number of unique UMIs the aligned reads correspond to.
- **Copy rank.** Primary, Secondary or Subsequent, see section 13.6.1.

To create a new Cell Clonotypes element from a row selection in a Cell Clonotypes table, use the **Create Clonotypes from Selection** option in the right-click menu.

13.6.3 Cell Clonotypes alignments

The alignments view () shows all assembled contigs mapping to a specific clonotype (figure 13.9).

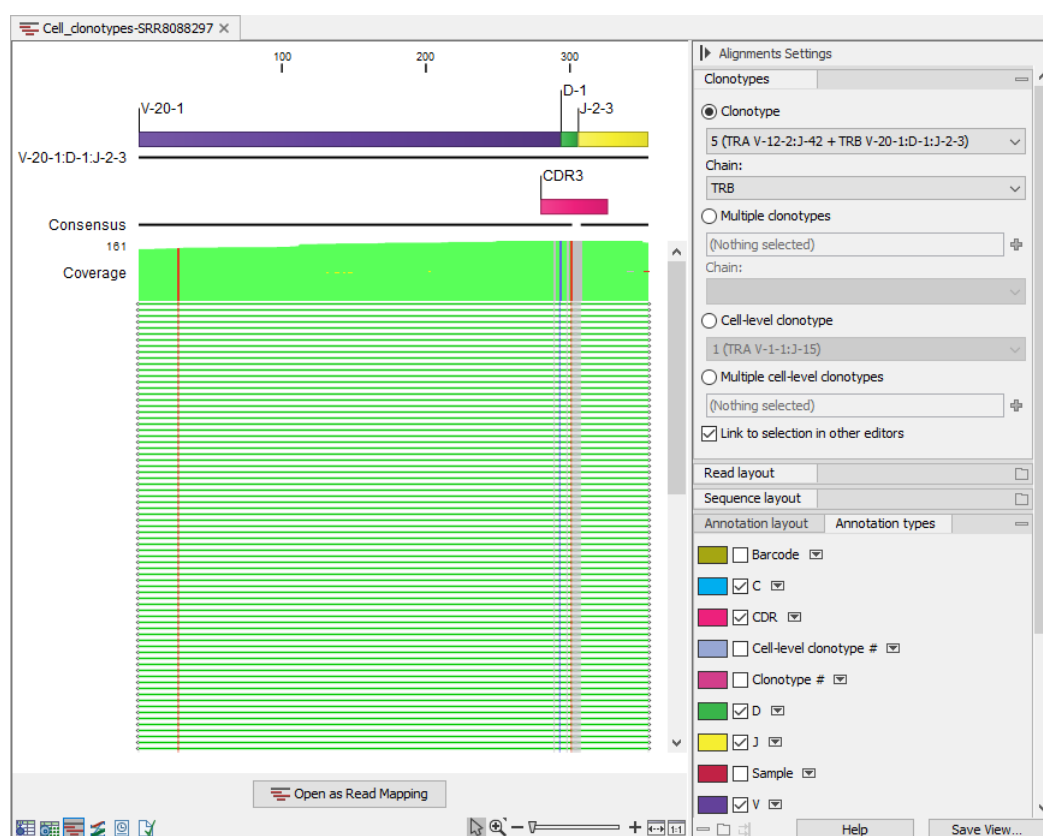


Figure 13.9: Read mapping for the TRB clonotype from a clonotype containing both TRA and TRB. V, D, J and C segments are annotated on the reference sequence and the CDR3 is annotated on the consensus.

The alignment contains:

- The reference sequence consisting of the identified V(D)JC segments. Annotations indicate the location of the different segment types. For clonotypes with ambiguous segments, only one of the identified segments is used.

- The consensus sequence with an annotation indicating the CDR3 region.
- The aligned contigs.

The clonotypes for which the alignment should be shown can be selected from the drop-down menus in the Side Panel, or from one of the clonotype tables (section 13.6.2) while using a split view (figure 13.10).

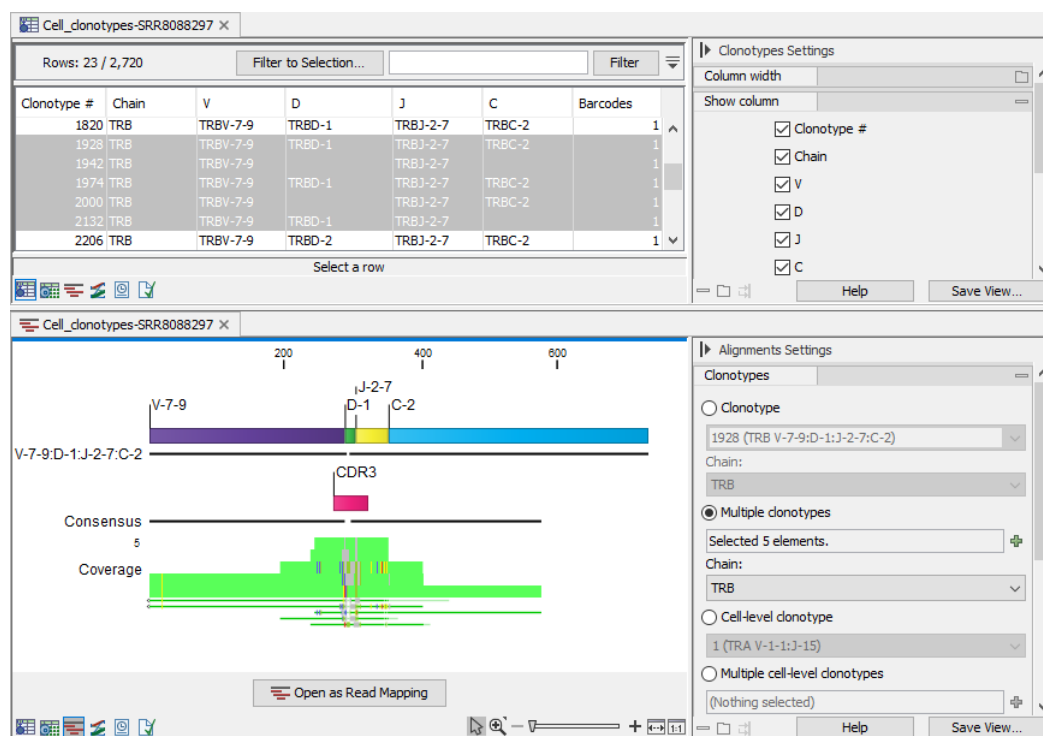


Figure 13.10: Clonotypes split view. Top: multiple clonotypes sharing reference segments are selected in the table view. Bottom: Alignment view for the clonotypes selected in the table view.

Alignments for multiple clonotypes can be shown together provided that they have the same chain, V and J segments and the D / C segments are not contradictory: either the D / C segment is identified and the same, or it is missing (figure 13.10).

When viewing alignments for multiple clonotypes, it can be useful to change "Compactness" to "Not compact" and tick the "Sample", "Barcode", "Cell-level clonotype #" and "Clonotype #" from the Side Panel. This way, it is easy to see this information for each of the aligned contigs (figure 13.11).

Figure 13.10 shows an alignment for multiple clonotypes. Some of the contigs do not span past the J segment and using the "Clonotype #" annotation (figure 13.11), we can confirm that these contigs belong to clonotypes for which the C segment has not been identified. Some of the clonotypes share the D segment, while others do not have an identified D segment. They all have different CDR3 sequences. Using the alignment view, it is straightforward to spot the differences between the CDR3 sequences.

For further processing, the alignments can be opened and saved as a stand-alone read mapping by using the "Open as Read Mapping" button. Using the **Extract Reads** tool (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Extract_Reads.html),




Figure 13.11: Alignment view for multiple clonotypes where "Compactness" is set to "Not compact" and "Clonotype #" is ticked.

the contigs can be extracted from the stand-alone read mapping as a Sequence List (📄), which can be used as input to Single Cell V(D)J-Seq Analysis. The tool will then skip the assembly and trimming and only clonotype the contigs (see section 13.1.2). This allows for custom processing of the contigs, where additional trimming can be performed before clonotyping.

Various settings controlling how the alignment, consensus and reference are displayed can be configured in the Side Panel, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=View_settings_in_Side_Panel.html.

13.6.4 Cell Clonotypes Sankey plot

The Sankey plot view ()

- shows how the segments of different types form the clonotypes for a given chain, when "Show column per" is set to "Grouping property" in the Side Panel (figure 13.12);
- compares clonotypes frequencies across samples, when "Show column per" is set to "Sample" in the Side Panel (figure 13.14). This option is available only for Cell Clonotypes containing more than one sample.

Note that only primary clonotypes (see section 13.6.1) are included in the Sankey plot. For visualizing secondary clonotypes, run first Filter Cell Clonotypes with 'Multiple clonotypes' set to 'Retain secondary', see section 13.2 for details.

To keep the plot size manageable, it is recommended to filter the clonotypes using the Filter Cell Clonotypes tool.

Grouping property

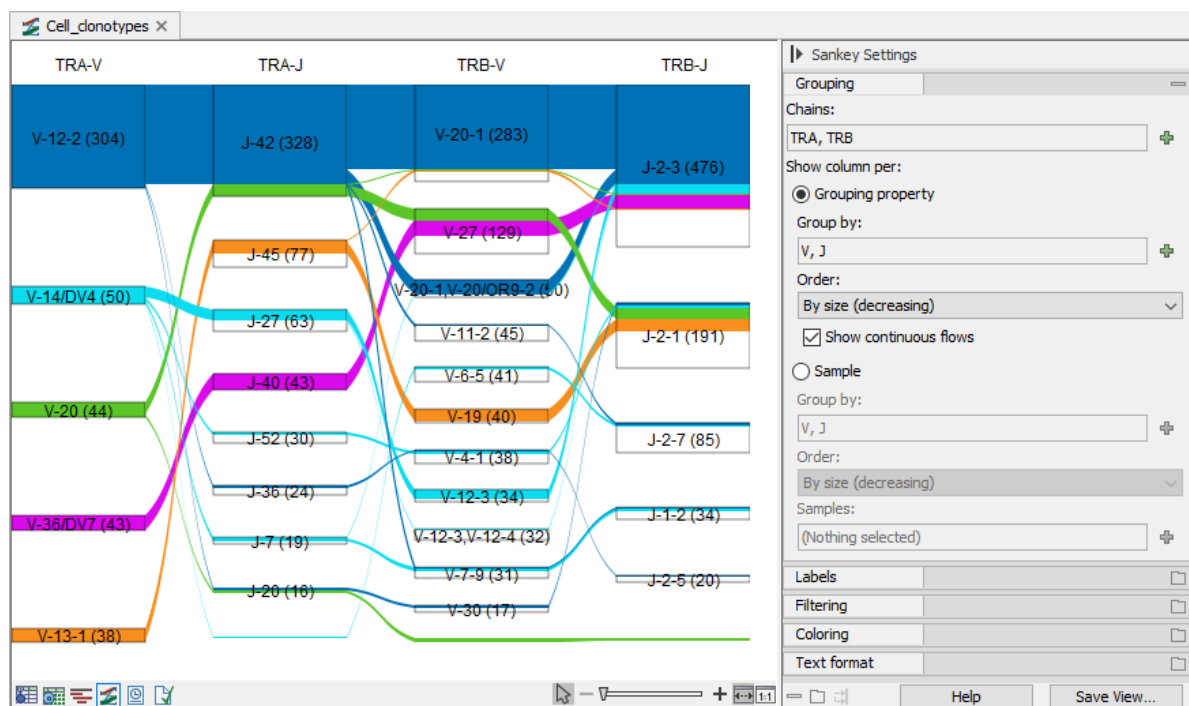


Figure 13.12: Sankey plot for the TRA and TRB chains showing the V and J segments. Numbers in brackets show the total barcode count. Flows show how many barcodes have clonotypes with the specific chain and segment combinations. The plot is restricted to showing only the most common 5 TRA-V segment. The box for TRA-J-45 contains a white region because there are barcodes with TRA-J-45 that have a TRA-V segment that is different than those present in the plot.

For each selected segment type, the plot has a column that contains boxes for each segment. The box height reflects the total number of cells containing clonotypes with the given segment. The boxes are connected with flows. The color of a flow indicates the element where the flow

- If not ticked, there are flows between boxes in consecutive columns for clonotypes having segments corresponding to the boxes. The height of the flow indicates the total number of cells for these clonotypes.
- If ticked, the flows start from the fixed column defined by "Flows start at". When the flow starts at the leftmost column, flows between boxes in the first two columns reflect clonotypes with segments corresponding to the two boxes. Flows between boxes in the second and third column reflect clonotypes corresponding to the boxes in both the first, second and third column, and so on.

Boxes can be removed from the plot by using the options under "Filtering" in the Side Panel (figure 13.12). The plot will show only boxes for the selected segments and the boxes to which the selected segments have a flow. If multiple filters are used, boxes are subject to all the restrictions (figure 13.13).

Cell_donotypes X

TRA-VJc TRA-CDR3 AA TRB-VJc TRB-CDR3 AA

(V-12-2)(J-42)(I) CAVNGGSGGNLIF (V-20-1)(D-1)(J-2-3)(I) CSACREDTDTQYF

(V-12-2)(J-42)(C) (V-20-1)(D-1)(J-2-3)(C-2)

None (V-36/DV7)(J-40)(I) (V-27)(D-2)(J-2-3)(I) CALSDLAIQGAGKLVF (V-27)(D-2)(J-2-3)(C-2) CASSFFGTSVTDTQYF

(V-9-2)(J-54)(I)

Sankey Settings

Grouping

Chains: TRA, TRB

Show column per:

☒ Grouping property

Group by: V(D)Jc, CDR3 AA

Order: By size (decreasing)

☒ Show continuous flows

☐ Sample

Group by: V, J

Order: By size (decreasing)

Samples: (Nothing selected)

Labels

Filtering

Coloring

Text format

Help Save View...

Figure 13.13: Sankey plot for the TRA and TRB chains showing the V(D)JC segments and CDR3. The plot is filtered to show only the most common 5 TRA and TRB V(D)JC segments. Note that only 4 boxes for TRA-VJC and TRB-VDJC are present in the plot, because there are no barcodes containing both of the missing TRA-VJC and TRB-VDJC. Clonotypes can have CDR3s that are out of frame and are hence missing a CDR3 AA. These are shown in the None box.

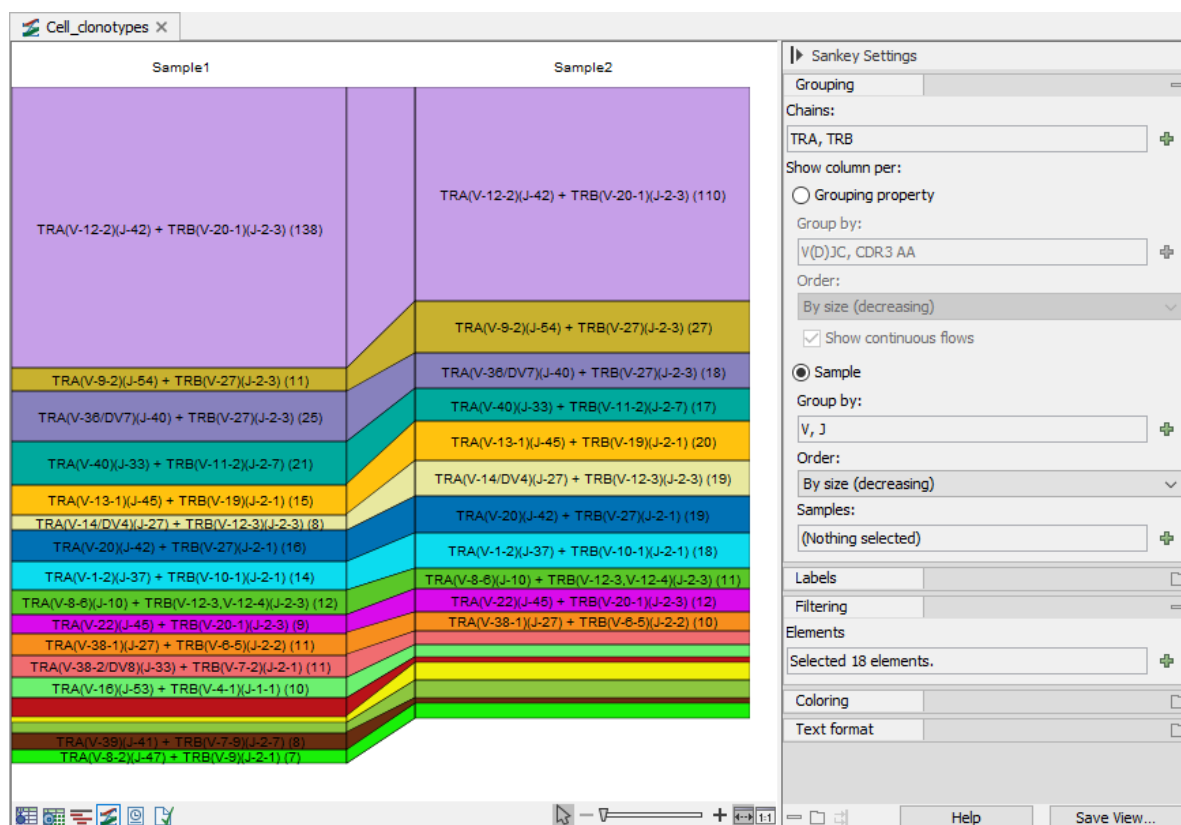


Figure 13.14: Sankey plot for the TRA and TRB chains showing the clonotypes count with specific V and J segments, compared across samples. Number in brackets show the total barcode count.

Sample

For each selected sample, the plot has a column that contains boxes for each group of clonotypes (hereafter referred to as simply clonotypes) with the selected properties. The properties, such as the segment type or the CDR3 amino acid sequence, are selected from the Side Panel under "Group by".

The height of a box indicates the frequency of the clonotypes in the sample. The frequency is defined as the number of barcodes with the specific clonotype, divided by the total number of barcodes found in the sample.

Chapter 14

Noise reduction through feature selection and dimensionality reduction

Contents

14.1 Feature selection and dimensionality reduction	188
14.1.1 Calculation of estimated biological variation	191

14.1 Feature selection and dimensionality reduction

Several tools provide options for speeding up calculations and reducing noise by decreasing the amount of data used. The available options are:

- Feature selection, where only highly variable genes (HVGs) are used.
- Dimensionality reduction, where data is projected into a lower dimensional space, through either principal component analysis (PCA) for expression data, or latent semantic indexing (LSI) for peak data.

At least one of these options must be used.

Highly variable genes (HVGs)

Not all genes are equally informative when clustering or visualizing cells. For example, house-keeping genes, whose expression levels are approximately constant across different cell types, are not informative for distinguishing between cell types. It is therefore often possible to get qualitatively the same results from an analysis by only using genes whose expression levels are highly variable across cells.

In order to use HVGs, data must first have been normalized by Normalize Single Cell Data. **Use highly variable genes** is not selected by default, but may be appropriate when speed is a priority, or when results using all genes appear unsatisfactory. The **Number of highly variable genes to use** must be specified. Values in the range 1000-5000 are typically sufficient to capture most variation from most data sets. Setting this value too low may exclude genes that are weakly informative, such as those that have small fold changes in rare cell types.

Highly variable peaks HVGs can only be used for expression data, and not for peak data. The only exception to this is when a tool works with expression and peak data simultaneously, and where no dimensionality reduction is applied. In this case, which is not recommended, the same number of peaks as HVGs are chosen at random to be "highly variable peaks".

When a tool is run, the log will contain estimates of the amount of signal and noise removed by choosing a certain number of HVGs (figure 14.1), which may help when choosing an appropriate value.

```
Estimated biological variation (as percentage of total variation): 16.7%
Using 1000 highly variable genes (HVGs)
Estimated biological variation after selecting HVGs (as percentage of total variation): 15.8%
Estimated noise removed by selecting HVGs (as percentage of total variation): 75.1%
```

Figure 14.1: An example of information provided in a tool log. Here, using 1000 HVGs reduced the total amount of variation in the data. However, the majority of the removed variation was estimated to be noise (75.1% of the original variation) and only a small amount of signal was lost ($16.7 - 15.8 = 0.9\%$ of the original variation). For more details on variation estimates, see section 14.1.1.

Genes are selected to be HVGs according to the variance of their normalized values, from highest variance to lowest variance. Genes with variance ≤ 1 are never selected, as this is consistent with random noise - this means that the number of HVGs used in an analysis may be lower than the number specified.

Note that using HVGs in one part of an analysis does not limit the number of genes available in downstream steps. For example, after constructing a visualization with HVGs, it is still possible to visualize the expressions of all genes.

Dimensionality reduction by PCA or LSI

In most circumstances it is recommended to **Use dimensionality reduction** as it provides a substantial increase in speed without affecting accuracy. Exceptions might include analysis of targeted expression data, where the expression of only a few hundred genes is measured.

Dimensionality reduction is by PCA for expression data, and LSI for peak data, using k number of dimensions as set in the **Dimensions** option. When both data types are present, k PCA components and k LSI components are used, and each cell is represented by $2 * k$ features.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) projects data into a lower dimensional space while preserving as much variation as possible.

Not all PCA dimensions are equal - the first dimension contains most of the variation and each subsequent dimension contains less of the variation than the previous one. For this reason, it often makes little difference to results whether k is set to 50 or 500, but large differences can be observed if too few PCA dimensions are used. Values in the range 20-50 are suitable for most applications. If the data has been normalized by Normalize Single Cell Data, the log will contain estimates of the amount of biological variation in the data, which can be compared to the amount

of variation captured by the chosen number of PCA dimensions (figure 14.2). For details on how biological variation is estimated, see section 14.1.1.

```
Estimated biological variation (as percentage of total variation): 16.7%
Target variation to be captured by PCA (as percentage of total variation): 16.7%
PCA dimensionality reduction: 16063 -> 20
Estimated biological variation in data: 16.7%, variation captured by PCA: 16.0%
```

Figure 14.2: An example of information provided in a tool log. Here, using 20 PCA dimensions captured 16.0% of variation in the data. This is comparable with the estimated amount of biological variation in the data.

PCA is performed using an implementation of Algorithm 971 [Li et al., 2017]. This is an extremely fast and accurate algorithm for finding the first PCA dimensions, but its accuracy decreases for higher dimensions. For this reason, it is advised to keep the number of PCA dimensions small compared to the number of expressed genes.

When data have been normalized by Normalize Single Cell Data it is additionally possible to **Automatically select PCA dimensions**. This chooses a number of dimensions ≤ 50 that contain the same amount of variation as the estimated biological variation. An example log is shown in figure 14.3.

```
Estimated biological variation (as percentage of total variation): 68.2%
Target variation to be captured by PCA (as percentage of total variation): 68.2%
PCA dimensionality reduction: 22094 -> 50
Estimated biological variation in data: 68.2%, variation captured by PCA: 49.4%
Insufficient PCA dimensions to capture estimated biological variation in the data.
At least 262 PCA dimensions are required.
```

Figure 14.3: An example of information provided in a tool log when selecting PCA dimensions automatically. Here, using 50 PCA dimensions captured 49.4% of variation in the data, which was lower than the estimated biological (i.e. non-noise) variation in the data. At least 262 PCA dimensions are required to capture all 68.2% of the variation estimated to be biological. However, the estimates are upper bounds and in practice 50 dimensions is likely to be sufficient.

Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) is often applied in natural language processing to a "document-term" matrix, which tabulates the number of times different words (terms) are seen in different documents. The technique returns a lower-dimensional representation of the matrix, with a reduced number of terms. These terms are linear combinations of original terms that are often found in the same documents.

In the context of scATAC-Seq, the peaks are terms and the cells are documents. The reduced dimensions are therefore linear combinations of peaks that are often found in groups of cells.

We construct a "document-term" matrix from the Peak Count Matrix using a Term Frequency - Inverse Document Frequency (TF-IDF) weighting:

$$D_{cp} = \mathbf{1}_{X_{cp} > 0} \log \frac{N}{1 + |c \in C : p \in c|}.$$

Here N is the number of cells, X_{cp} is the Peak Count Matrix element for cell c and peak p , $\mathbf{1}_{X_{cp} > 0}$

is the indicator function that returns 1 if $X_{cp} > 0$ or else 0, and $|c \in C : p \in c|$ is the number of cells containing peak p .

LSI with k dimensions is then achieved by taking the first k -components of the \mathbf{U} matrix returned by singular value decomposition of \mathbf{D} :

$$\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

To correct for the effect of sequencing depth, the \mathbf{U} matrix is re-normalized such that each cell is represented by a unit vector:

$$U_{ck} = \frac{1}{\sqrt{\sum_k U_{ck}^2}} U_{ck}.$$

Combining HVGs and PCA or LSI

It is possible to use HVGs and dimensionality reduction together. When this is done, HVGs are selected and then dimensionality reduction is run only on the HVGs. Note that, because using HVGs already removes a lot of noise, the log may show that even a relatively large number of **Dimensions** is insufficient to capture all the estimated biological variation. It may be worth experimenting with increasing the number of dimensions slightly to check whether this has an impact on the results.

Using both expression and peak data

When feature selection and/or dimensionality reduction is applied to both an expression and peak matrix, only cells that are in common to both matrices are used.

It is possible to run only feature selection without dimensionality reduction. To ensure that the two data types contribute equally to the downstream analysis, the number of peaks is also reduced to the same number as the genes. Since no corresponding method for feature selection exists for peak data, the peaks are chosen randomly.

When dimensionality reduction is applied, all peaks are used, regardless of the HVGs settings, as the dimensionality reduction ensures equal contribution of both data types.

14.1.1 Calculation of estimated biological variation

Genes that have been normalized by Normalize Single Cell Data have an expected variance of ~ 1 from random noise. In reality many genes have larger variance because they do not perfectly fit the model used in normalization. This is expected because the model only expects expression to vary due to sequencing depth and (optionally) batch effects - it does not account for expressions differing across different cell types or treatments.

We define the ‘estimated biological variation’ v_{bio} in a normalized sample to be the fraction of the total variance that is above the expected variance due to random noise for each gene

$$v_{\text{bio}} = \frac{\sum_g \max(\text{Var}(z_g) - 1, 0)}{\sum_g (\text{Var}(z_g))}.$$

Here, z_g are the normalized expressions of gene g . Note that this estimate assumes that all

variation remaining after normalization is of ‘biological’ origin. This is unlikely in practice, and the estimate will often be too high.

Chapter 15





Cell Annotation

Contents

15.1 Cluster Single Cell Data	193
15.1.1 The Cluster Single Cell Data algorithm	195
15.2 Create Heat Map for Cell Abundance	195
15.2.1 The output of Create Heat Map for Cell Abundance	196
15.3 Create Cell Annotations from Hashtags	200
15.4 The Cell Annotations element	201
15.5 The Cell Clusters element	202

15.1 Cluster Single Cell Data

Cluster Single Cell Data uses a graph-based clustering to automatically cluster cells. Typically the aim is to recover clusters that describe cells of different types or with different behavior.

The tool takes an Expression Matrix ( / ) or a Peak Count Matrix () or both types of matrix as input, and produces a Cell Clusters () result. Note that when both types of matrices are provided, only cells that are in common to both matrices are used.

Cluster Single Cell Data is available from:

Tools | Single Cell Analysis () | **Cell Annotation** () | **Cluster Single Cell Data** ()

The tool offers options to run dimensionality reduction or feature selection prior to clustering. For details on these options, please see section 14.1. The following additional options are available:

- **Distance measure.** The algorithm starts from a k-nearest neighbor graph, and the distance measure is used to find the ‘nearest’ neighbors. The ‘1-Pearson correlation’ distance is less sensitive to changes in the scale of expression between cells than Euclidean distance (for example, if one cell has exactly twice the expression of another for each gene, the ‘1 - Pearson correlation’ distance is 0 while the Euclidean distance may be very large) and may be better at finding more distant neighbors. Conversely, Euclidean distance may provide higher resolution for distinguishing similar cell types.

- **Neighborhood size.** The number of cells 'k' used in the k-nearest neighbor graph. This determines the granularity of the visualization. Smaller values may be better at recovering small clusters, but may also lead to larger clusters becoming fragmented.
- **Use fixed resolution** The resolution controls the coarseness of the clustering, with smaller values of the resolution leading to fewer clusters. When this option is disabled, results for several different resolutions from 0.1 to 1.5 are returned. Only when none of these resolutions appear appropriate would a fixed resolution typically be required.
- **Resolution** The fixed resolution to use.

The result of clustering is a Cell Clusters (📊) element containing clusters at different resolutions. It is easiest to view these in a Dimensionality Reduction Plot (📊).

Generally speaking, a good clustering will have distinct clusters for each large clump of cells that appears to form a cluster by eye in the Dimensionality Reduction Plot. If this is not the case, the resolution may be too low (as in figure 15.1, compared with figure 15.2). Unfortunately, it can be hard to tell when the resolution is too high, but generally one or more of the clusterings at a default resolution will be suitable for downstream analysis.

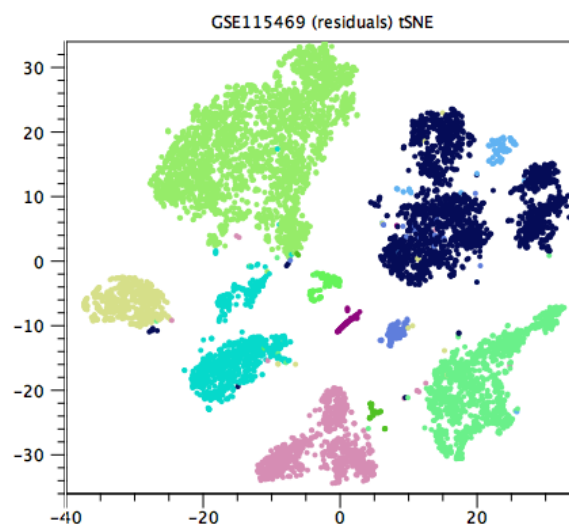


Figure 15.1: Clustering with too low resolution. Clusters that are distinct by eye are given the same color. Examples include the three dark blue clusters at the top-right corner of the plot, and the two turquoise clusters at $x=-20$. Data is from [MacParland et al., 2018](#).

As the aim of clustering is usually to have clusters that correspond to different cell types, it is possible, from the Dimensionality Reduction Plot, to redraw the boundaries between clusters, to add new clusters, and to rename clusters. These changes might be based on insights from other sources of information such as:

- Predicted cell types from the Predict Cell Types tool.
- The expression of known marker genes for a cell type.
- Marker genes that have been detected from the clusters by Differential Expression for Single Cell.

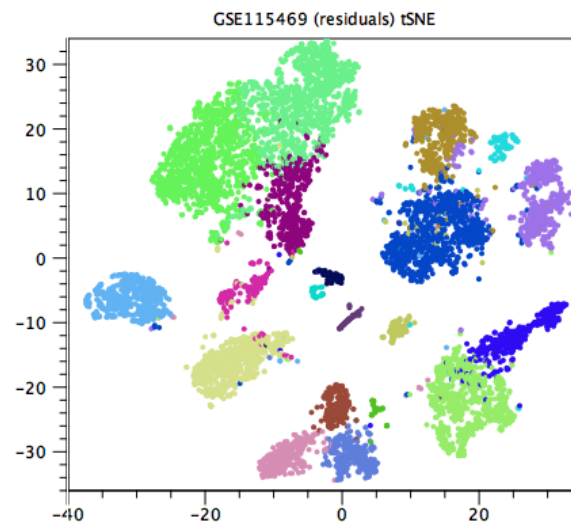


Figure 15.2: A higher resolution clustering of the same data as in figure 15.1. Each cluster that seems distinct by eye is now given its own color. The resolution is no longer too low. It can be difficult to determine whether the resolution is too high.

15.1.1.1 The Cluster Single Cell Data algorithm

Cluster Single Cell Data is a graph-based clustering method. It proceeds in three phases:

1. Construction of a k-nearest neighbor graph (kNN), where each cell is a node in the graph with edges to its k nearest neighbors.
2. Construction of a Shared Nearest Neighbor (SNN) graph from the kNN graph using the method of [Xu and Su, 2015](#). Briefly, in the SNN graph each cell is again a node, but two cells are only connected by an edge if they share a nearest neighbor in the kNN graph. Neighbors of each cell in the kNN graph are ranked from 1 (the same cell, because each cell is its own closest neighbor) to k (the most distant neighbor). Edges in the SNN graph are weighted according to the best of the average ranks of their shared neighbors. Edges connecting cells that share close nearest neighbors are weighted higher than edges connecting cells that only share distant nearest neighbors.
3. Application of Leiden community detection to the weighted SNN graph [[Traag et al., 2019](#)].


The Leiden community detection algorithm has two hyperparameters. These are set as follows:

- **Iterations:** 3
- **Randomness:** $\theta = 0.01$

15.2 Create Heat Map for Cell Abundance

Create Heat Map for Cell Abundance compares two groupings of the same cells. It outputs a heat map showing the cell abundance for the two groupings. The tool is useful for matching

automated clusters with predicted cell types or for comparing analysis that has been run in third party tools with the results obtained using the CLC Single Cell Analysis Module.

The tool produces a **Heat Map** () of cell abundance. It is often most natural to run the tool from a Dimensionality Reduction Plot by right-clicking on the plot, see section 17 for details. However, it can also be found under the Tools menu at:

Tools | Single Cell Analysis () | **Cell Annotation** () | **Create Heat Map for Cell Abundance** ()

The tool requires at least one **Cell Clusters** () and/or **Cell Annotations** (). A number of options are available to choose and order the groups:

- **Group by (row)**. Select which category should be displayed as rows.
- **Select groups (row)** (Optional). This can be supplied to reduce the groups in the plot to only those of interest, or to control their order.
- **Group by (column)**. Select which category should be displayed as columns.
- **Select groups (column)** (Optional). This can be supplied to reduce the groups in the plot to only those of interest, or to control their order.


While numerical categories can be selected from Cell Annotations, it is often most relevant to choose discrete traits.

Each colored rectangle in the heat map represents the number of cells found in both groups. Three options exist for scaling the numbers:

- **By all**. All entries will sum to 100%.
- **Per row**. All entries in each row will sum to 100%.
- **Per column**. All entries in each column will sum to 100%.

When selecting "By all", the most abundant pairs of groups will be most noticeable, whereas "Per row / column" highlights how well different groupings match and shows the composition of one of the groupings as a function of the other.

15.2.1 The output of Create Heat Map for Cell Abundance

Figure 15.3 is an example of a plot showing how a third party tool's cell type annotations (columns) compare to the Leiden clusters with resolution=0.5 (rows) scaled using the "Per column" option. There seems to be a good concordance between the groups: there is mostly only one red rectangle per column. The cell abundance values can be seen in the table view ()

Reordering the cell types can improve the visualization and make it easier to interpret the results (see figure 15.4). The red diagonal makes it clear that only minor differences exist between the two groupings.

Creating a new heat map using the "By all" scaling option shows whether most cells fall into the expected categories. Figure 15.5 illustrates how frequent each group combination is, with "15 - CD4 Naive T cells" and "6 - CD14 Mono cells" being most abundant.

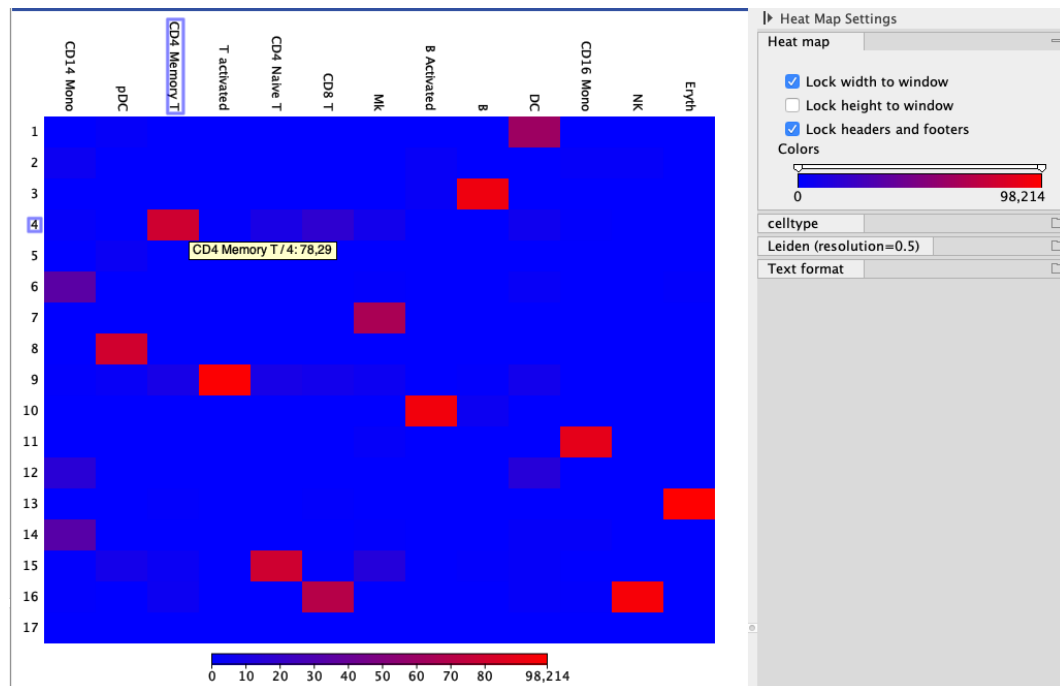


Figure 15.3: A heat map of cell type abundance showing Leiden clusters with $\text{resolution}=0.5$ calculated using the CLC Single Cell Analysis Module and compared to the cell types predicted by Seurat, using a tutorial data set (https://satijalab.org/seurat/archive/v3.2/immune_alignment.html). Hovering over the rectangle reveals the abundance of the selected combination, also indicated by the color.

The heat maps of cell abundance indicate that "CD14 Mono cells" are split into four clusters, 2, 6, 12 and 14. These four clusters are adjacent in a UMAP plot (see figure 15.6), suggesting that they represent sub-types of "CD14 Mono cells".

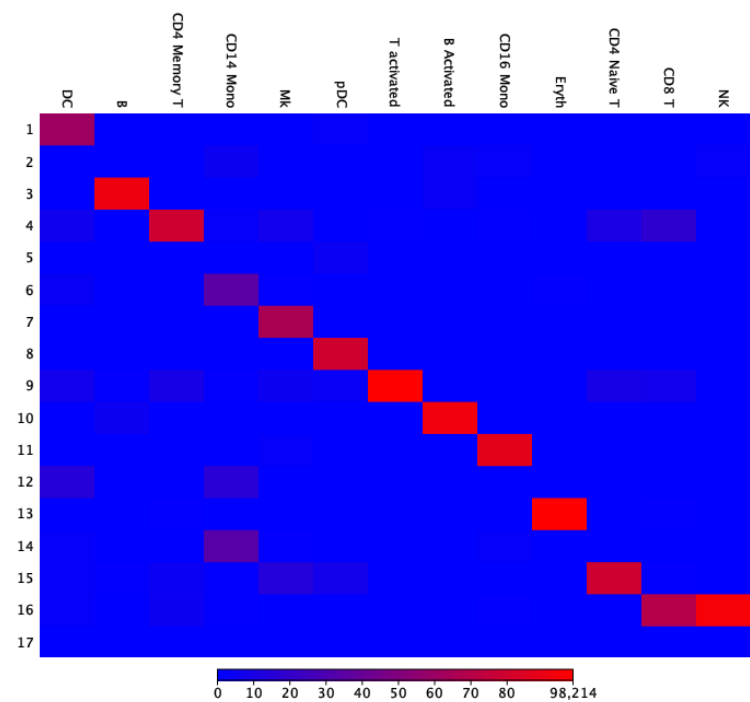


Figure 15.4: The heat map from figure 15.3 has been rearranged for easier interpretation by changing the order of the Seurat cell types.

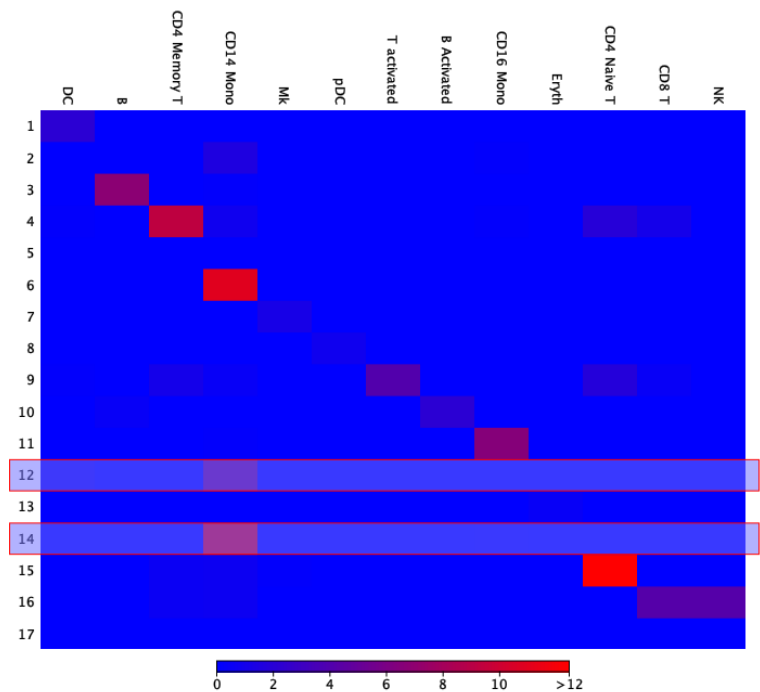


Figure 15.5: The heat map from figure 15.4 with all abundance entries summing up to 100. The highlighted clusters, 12 and 14, are annotated as CD14 Mono cells and have a relatively high abundance.

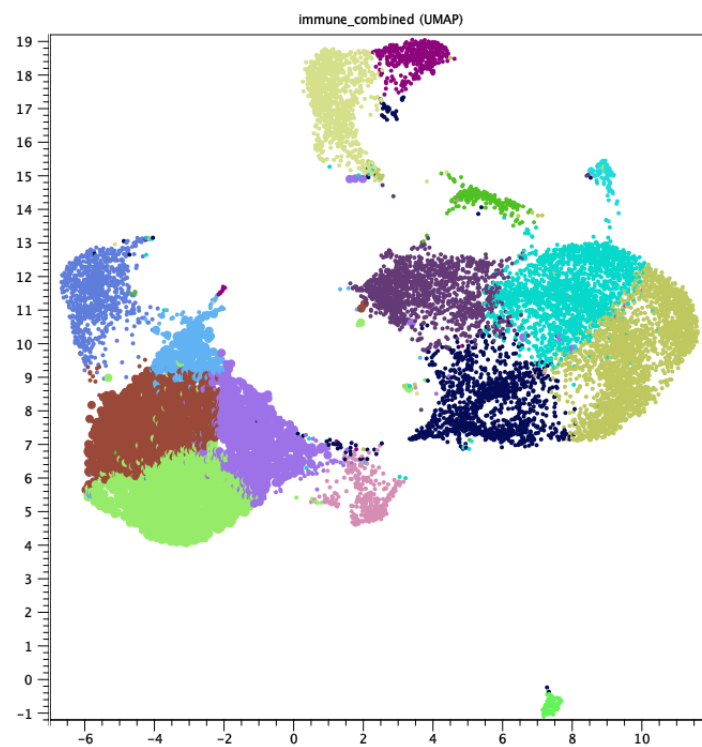


Figure 15.6: UMAP of the same data as in figure 15.5. The blue, green, lilac, and brown cells in the lower left corner correspond to clusters 2, 6, 12 and 14 respectively. Highlighted cells (larger points) are also "CD14 Mono cells".

15.3 Create Cell Annotations from Hashtags

Create Cell Annotations from Hashtags is available from:

Tools | Single Cell Analysis (🔍) | **Cell Annotation** (🌐) | **Create Cell Annotations from Hashtags** (🔍)

The tool takes as input one or more sequence lists (📄) of reads that have been annotated with cell barcodes and hashtags using **Annotate Single Cell Reads**.

Using a file that translates hashtags to annotations, it produces a Cell Annotations (📄) element containing, for each cell, the corresponding annotations according to the hashtags found on the reads. Note that when the hashtag represents the sample, Update Single Cell Sample Name can be used with the output produced by this tool, see section 18.7.

A number of options are available (figure 15.7).

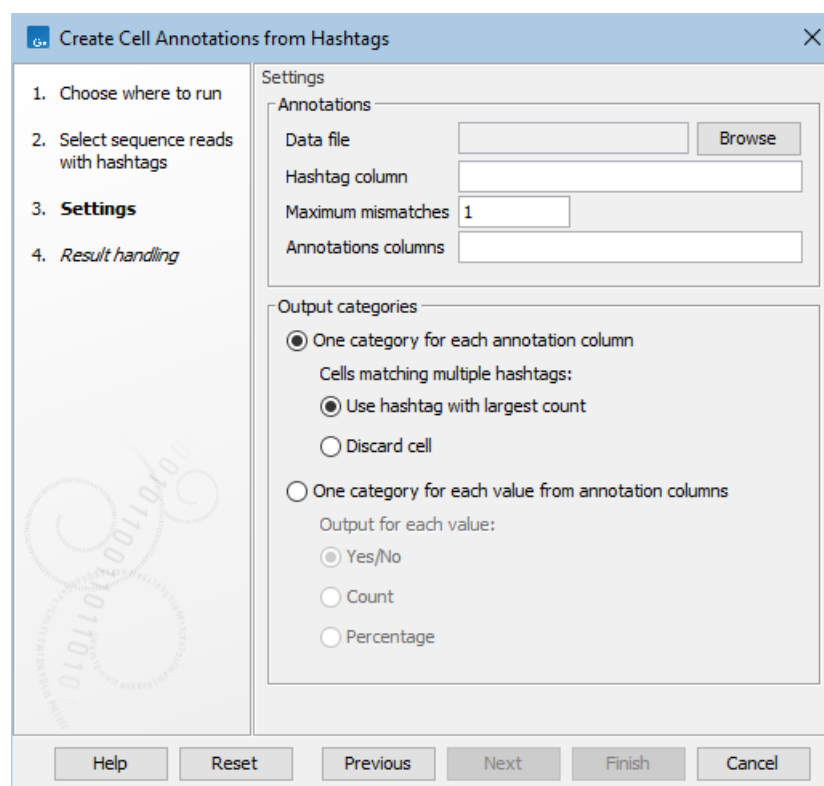


Figure 15.7: The options in the dialog of the Create Cell Annotations from Hashtags tool.

Under ‘Annotations’:

- **Data file.** A single file in .csv, .tsv or .xlsx format.
- **Hashtag column.** The name of the column containing the hashtags.
- **Maximum mismatches.** When matching hashtags from reads with those from the data file, this many sequence mismatches are allowed. Note that since the hashtags have a fixed length, insertions and deletions end up counting for 2 mismatches. Hashtags are matched such that there are as few mismatches as possible.

- **Annotation columns.** The names of the columns to output as cell annotations.

Under ‘Output categories’, the categories and content of the output Cell Annotations can be configured:

- **One category for each annotation column.** If selected, one category will be added for each selected annotation column. One hashtag is used for each cell. If a cell has multiple hashtags:
 - **Use hashtag with largest count.** The hashtag with the largest ‘Count’ (see below) is used.
 - **Discard cell.** The cell is not added to the output.


The annotation for a cell will be the value found in the annotation column for the row with the hashtag of the cell.

- **One category for each value from annotation columns.** If selected, one category will be added for all unique values found in each selected annotation column. E.g., if a column name is "HTO" with content "A", "B" and "C", there will be three categories "HTO: A", "HTO: B" and "HTO: C". For each cell, all identified hashtags are used. The annotation for a cell is configured as follows:
 - **Yes/No.** ‘Yes’ if the cell has reads with the corresponding hashtag, ‘No’ otherwise.
 - **Count.** The number of reads with the corresponding hashtag. Reads are collapsed and counted as one when they are for the same cell and have the same UMI and same hashtag.
 - **Percentage.** The ‘Count’ transformed to percentage.

15.4 The Cell Annotations element

Cell Annotations () elements contain information about cells and their annotations.

The Cell Annotations Table


The Cell Annotations Table () contains one row for each annotation, and has the following columns:

- **Annotation.** The value of the annotation.
- **Cells.** The number of cells with this annotation value.

Clicking on a row opens a separate table, listing the cells with the annotation value.

To create a new Cell Annotations element from a row selection in the Cell Annotations Table, use the **Create Cell Annotations Element from Selection** option in the right-click menu.

Cell Table


The Cell Table () for a Cell Annotations () element contains one row for each cell, and has the following columns:

- **Sample.** The sample that the cell is from.
- **Barcode.** The cell barcode.

They are followed by one column per annotation category, displaying the value of the annotation.

To create a new Cell Annotations element from a row selection in the Cell Table, use the **Create Cell Annotations Element from Selection** option in the right-click menu.

15.5 The Cell Clusters element



Cell Clusters () elements contain information about cells and the clusters they belong to.

Cell Clusters Table The Cell Clusters Table () contains one row for each cluster, and has the following columns:

- **Cluster.** The name of the cluster.
- **Cells.** The number of cells in this cluster.

Clicking on a row opens a separate table, listing the cells in the cluster.

To create a new Cell Clusters element from a row selection in the Cell Clusters Table, use the **Create Cell Clusters Element from Selection** option in the right-click menu.

Cell Table The Cell Table () for a Cell Clusters () element contains one row for each cell, and has the following columns:

- **Sample.** The sample that the cell is from.
- **Barcode.** The cell barcode.

They are followed by one column per cluster category, displaying the cluster containing the cell (if any).

To create a new Cell Clusters element from a row selection in the Cell Table, use the **Create Cell Clusters Element from Selection** option in the right-click menu.


Chapter 16

Dimensionality reduction

Contents




16.1 UMAP for Single Cell	203
16.2 tSNE for Single Cell	207

16.1 UMAP for Single Cell

Uniform Manifold Approximation and Projection, UMAP, is a general purpose algorithm for visualizing high dimensional data in 2D or 3D [McInnes et al., 2018]. In the CLC Single Cell Analysis Module, it is one of two ways of constructing a Dimensionality Reduction Plot () , with the other being tSNE. The choice between tSNE and UMAP is purely visual - it has no effect on downstream analysis. Therefore it is recommended to use the tool that produces the visualization you prefer.

The UMAP for Single Cell tool is available from:

Tools | Single Cell Analysis () | **Dimensionality Reduction** () | **UMAP for Single Cell** ()

The tool takes an Expression Matrix () / () , or a Peak Count Matrix () , or both types of matrix as input. Note that when both types of matrices are provided, only cells that are in common to both matrices are used.

UMAP for Single Cell offers options to run dimensionality reduction or feature selection prior to the UMAP algorithm. For details on these options, please see section 14.1. The following additional options are available:

- **Produce 3D plot.** Perform a second UMAP calculation in three dimensions. As this involves a full re-calculation of UMAP in a higher dimension, the runtime is approximately doubled when this option is selected. Note that the 3D plot has special system requirements, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=System_requirements.html.
- **Distance measure.** UMAP works on a k-nearest neighbor graph, and the distance measure is used to find the 'nearest' neighbors. The '1-Pearson correlation' distance is less

sensitive to changes in the scale of expression between cells than Euclidean distance (for example, if one cell has exactly twice the expression of another for each gene, the '1 - Pearson correlation' distance is 0 while the Euclidean distance may be very large) and may be better at finding more distant neighbors. Conversely, Euclidean distance may provide higher resolution for distinguishing similar cell types.

- **Neighborhood size.** The number of cells 'k' used in the k-nearest neighbor graph. This determines the granularity of the visualization. Smaller values will recover more local structure, but will lose the 'big picture'. Larger values may average out fine structure.
- **Random seed.** The algorithm contains a random component determined by the seed. This means that each value of the seed leads to a slightly different visualization.
- **Minimum distance.** The effective minimum distance between embedded points. Smaller values result in tighter clusters.
- **Spread.** The effective scale of embedded points. Smaller values result in tighter clusters.
- **Epochs.** The algorithm works by repeating the same steps a predefined number of times. There is, unfortunately, no good rule for determining how many iterations are appropriate. More iterations do no harm, but too few iterations may lead to clusters of cells failing to separate. Doubling the number of iterations approximately doubles the runtime.

An example output is shown in figure 16.1.

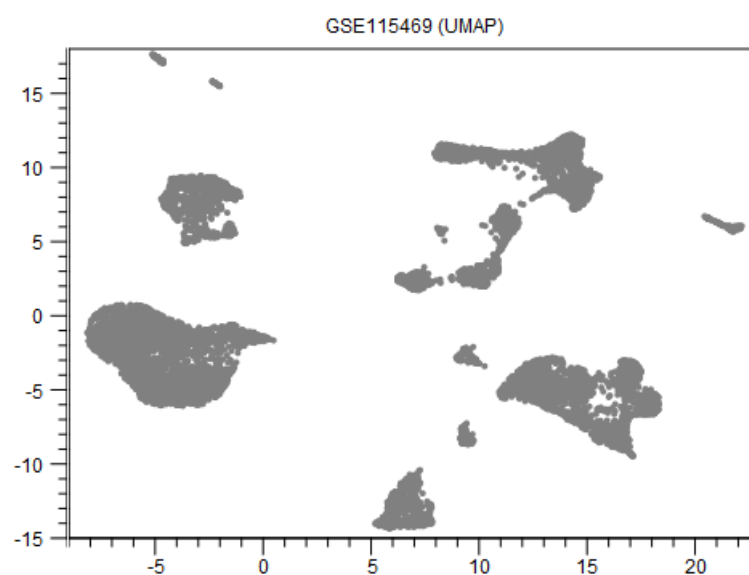


Figure 16.1: A UMAP visualization of data from *MacParland et al., 2018*.

Tuning the visualization

Although reducing **Spread** and **Minimum distance** give tighter clusters, they do so in different ways. Therefore it can be useful to try changing both parameters. An example of this is given in figures 16.2-16.4.

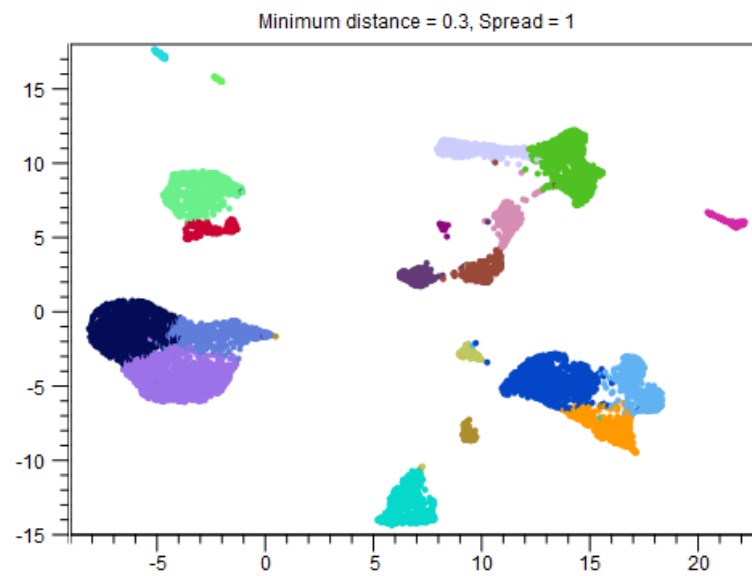


Figure 16.2: UMAP with *Minimum distance* = 0.3 and *Spread* = 1. This is the same plot as in figure 16.1, but with clusters overlaid.

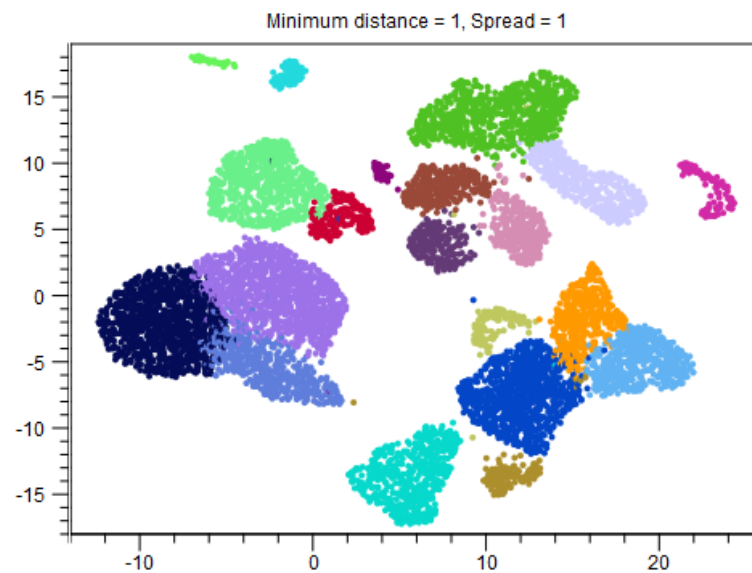


Figure 16.3: UMAP with *Minimum distance* = 1 and *Spread* = 1. The overall structure of the clusters is the same as in figure 16.2, but the points are more separated.

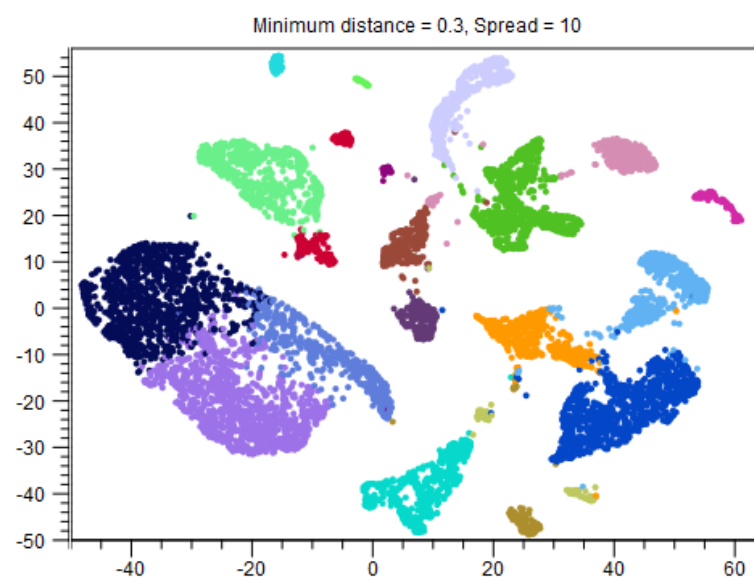


Figure 16.4: UMAP with Minimum distance = 0.3 and Spread = 10. Both points and clusters are more separated than in figure 16.2. Whether this is desirable is likely to depend on the application. For example, it is easier to see that the dark blue, light blue and orange clusters are different cell types, which may help with cell type annotation, but their proximity in the other figures may have indicated a shared developmental lineage, which it is not possible to see here. Note that other clusters, such as the red cluster, are now also split in two, compared to the other figures.

16.2 tSNE for Single Cell

t-Distributed Stochastic Neighbor Embedding, tSNE, is a general purpose algorithm for visualizing high dimensional data in 2D or 3D [Maaten and Hinton, 2008]. In the CLC Single Cell Analysis Module, it is one of two ways of constructing a Dimensionality Reduction Plot (📊), with the other being UMAP. The choice between tSNE and UMAP is purely visual - it has no effect on downstream analysis. Therefore it is recommended to use the tool that produces the visualization you prefer.

The tSNE for Single Cell tool is available from:

Tools | Single Cell Analysis (📁) | **Dimensionality Reduction** (📊) | **tSNE for Single Cell** (🔍)

The tool takes an Expression Matrix (📄) / (📄), or a Peak Count Matrix (📄), or both types of matrix as input. Note that when both types of matrices are provided, only cells that are in common to both matrices are used.

tSNE for Single Cell offers options to run dimensionality reduction or feature selection prior to the tSNE algorithm. For details on these options, please see section 14.1. The following additional options are available:

- **Produce 3D plot.** Perform a second tSNE calculation in three dimensions. As this involves a full re-calculation of tSNE in a higher dimension, the runtime is approximately doubled when this option is selected. Note that the 3D plot has special system requirements, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=System_requirements.html.
- **Random seed.** The algorithm contains a random component determined by the seed. This means that each value of the seed leads to a slightly different visualization.
- **Iterations.** The algorithm works by repeating the same steps a predefined number of times. There is, unfortunately, no good rule for determining how many iterations are appropriate. More iterations do no harm, but too few iterations may lead to clusters of cells failing to separate. Doubling the number of iterations approximately doubles the runtime.
- **Automatically select perplexity.** automatically chooses a value for the perplexity based on the number of cells, n . This is set to $(n - 2)/3$ for $n < 92$ (the highest value allowed by the data), 30 for $92 \leq n \leq 3000$ (a commonly used value in the literature), $n/100$ from $3000 \leq n \leq 10000$ (suggested by Kobak and Berens, 2019), and 100 for $n \geq 10000$.
- **Perplexity.** The perplexity roughly corresponds to the number of close neighbors (in expression space) that each cell has. Generally speaking, smaller values of the complexity lead to a tendency to form more clusters.

An example output is shown in figure 16.5. When interpreting tSNE plots, it is important to be aware that the tightness of clusters and distances between them may not reflect the actual intra- and inter-cluster similarities. Some examples of this are provided by Wattenberg et al., 2016.

Implementation details Barnes-Hut tSNE is implemented [Van Der Maaten, 2014]. If dimensionality reduction has been selected, the initial guess at the optimal layout is seeded using PCA and/or LSI (plus a small amount of random variation), and otherwise is uniformly random

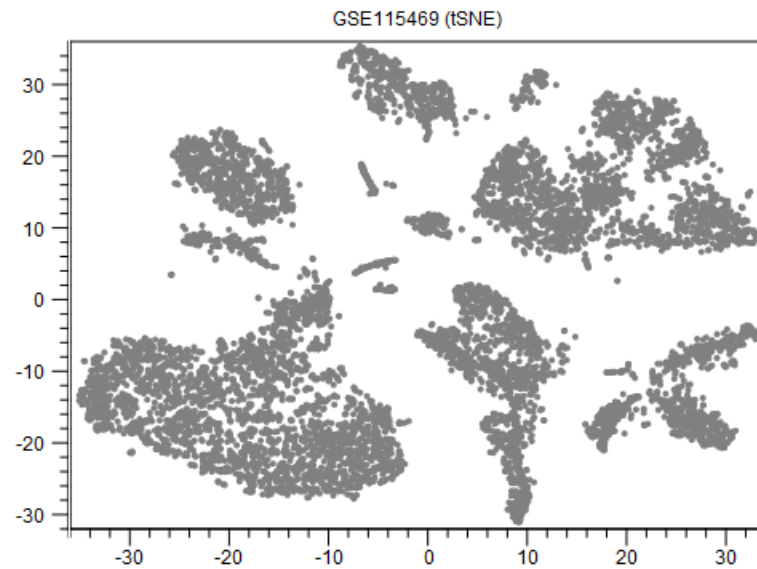


Figure 16.5: A tSNE visualization of data from *MacParland et al., 2018*.

in the range 0 - 0.0001. The use of dimensionality reduction is recommended, because several authors have reported improved conservation of global structure in tSNE visualizations when PCA initialization is used.

tSNE has several hyperparameters, which are set as follows:

Early exaggeration factor: $\alpha = 12$

Learning rate: $\nu = \max(200, n/\alpha)$ where n is the number of cells

Iterations for early exaggeration: 250

Momentum during early exaggeration: 0.5

Momentum for subsequent iterations: 0.8




Chapter 17

Single cell low-dimensional plots functionality

Contents

17.1 Manual annotation	210
17.2 Visualizing different types of matrices	221
17.3 Create Subset	223
17.4 Extract to Table	224
17.5 Launching of Create Expression Plot	225
17.6 Launching of Create Heat Map for Cell Abundance	226
17.7 Launching of Differential Accessibility for Single Cell	227
17.8 Launching of Differential Expression for Single Cell	228
17.9 Launching of Differential Velocity for Single Cell	229
17.10 Launching of Score Velocity Genes	230

CLC Single Cell Analysis Module can produce three types of plots, where each cell or barcode is represented as one point in a low-dimensionality representation:

- **Dimensionality Reduction Plot** () , see chapter 16.
- **Phase Portrait Plot** () , see section 10.4.
- **Spatial Transcriptomics Plot** () , see section 11.1.

The title of a Dimensionality Reduction Plot or Spatial Transcriptomics Plot is the same as the name of the plot element, and it can be updated by renaming the element. The title of a Phase Portrait Plot is the name of the displayed gene.

Using such a low-dimensional plot:

- different aspects of the data can be visualized;
- cells can be manually annotated;

- various tools can be started using the information selected in the plot.

This chapter describes this functionality, showcasing specific aspects.

Note that it is not possible to edit clusters or launch tools using Phase Portrait Plots.

17.1 Manual annotation

This section showcases the different functionalities available from a low-dimensional plot using a UMAP plot of an Expression Matrix (📊) generated from data from [MacParland et al., 2018](#).

The cells in a plot can be colored using different sources of information, such as Cell Clusters, Cell Annotations, features expression and sample of origin. To enable this coloring, the relevant elements are associated with the plot by dragging and dropping them in the corresponding groups of the Side Panel (see figure 17.1).

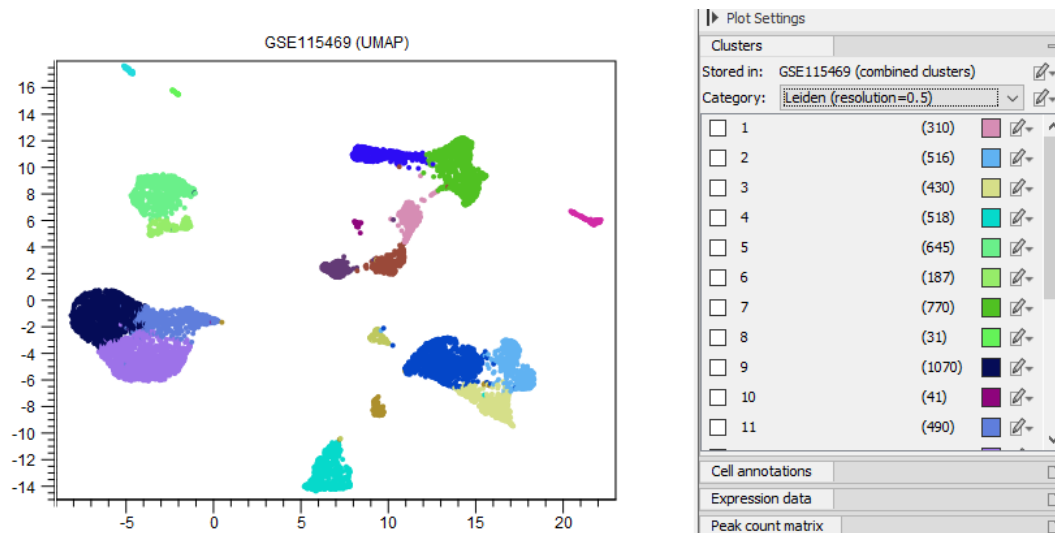


Figure 17.1: Cells are colored by clusters produced by Cluster Single Cell Data (see section 15.1). The category 'Leiden (resolution=0.5)' was chosen in the Side Panel Clusters group.

To visualize the same low-dimensional plot using different sources of information for coloring, the plot can be opened multiple times and the windows can be rearranged by dragging and dropping (see figure 17.2).

On mouse hover, a tooltip shows summaries for the nearby cells (see figure 17.3). The tooltip can be disabled by unchecking 'Show tooltip' in the 'Coloring and highlighting' group at the bottom of the Side Panel. The same type of summary can be obtained for a group of selected cells by choosing 'Show Information for Selected' from the plot right-click menu (see figure 17.4).

On right-click on the plot, a series of options are available for launching tools or performing various actions on selected cells (see figure 17.4).

Selecting cells

Cells can be selected in multiple ways:

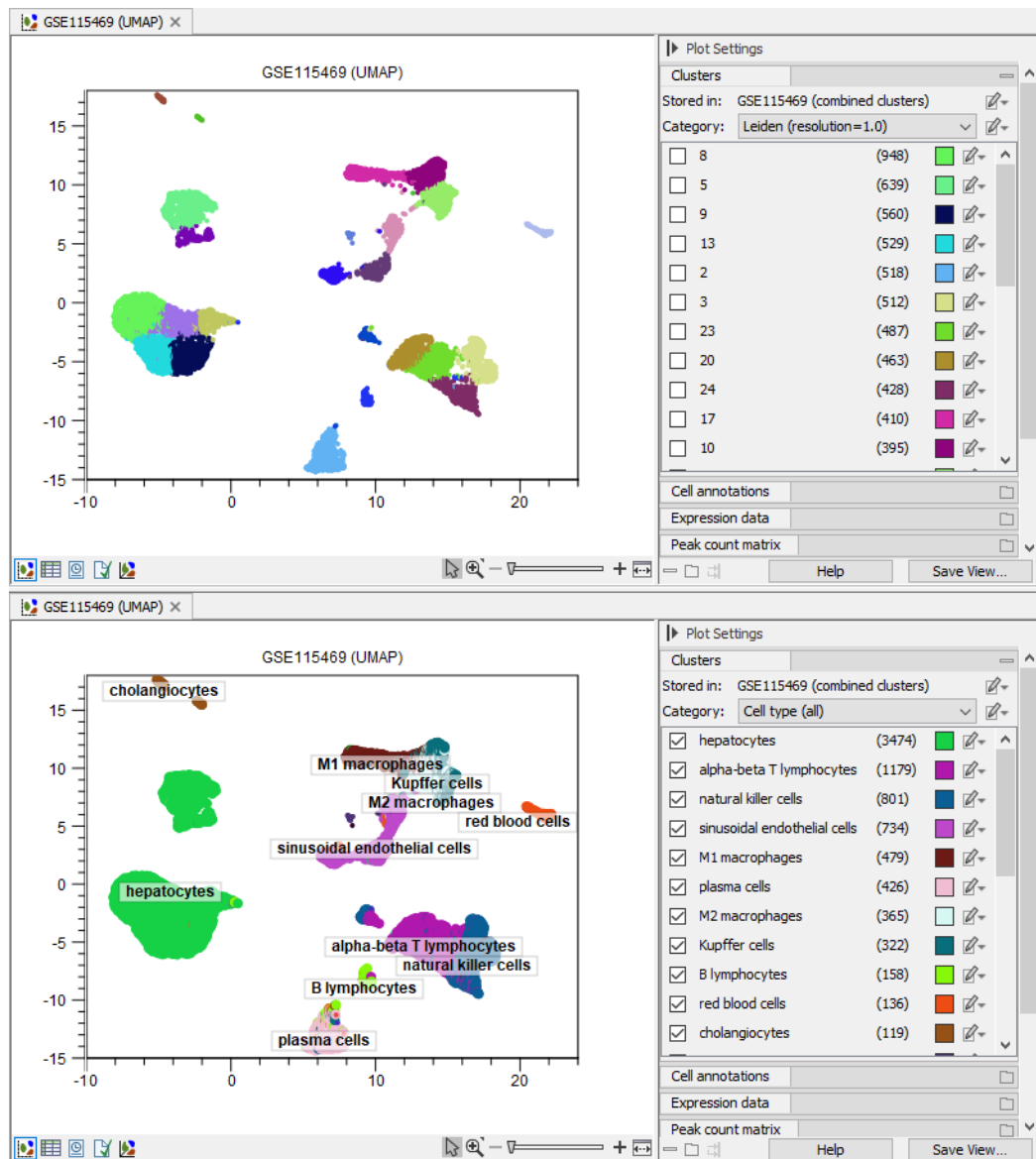


Figure 17.2: The same UMAP plot is opened two times, and cells are colored by the automated clusters with ‘Leiden (resolution=1.0)’ (top) and the predicted cell types (bottom) produced by Predict Cell Types (see section 8.2) using the human pre-trained cell classifier (see chapter 2). The label for the selected cell types is added on the plot by choosing ‘Show labels from: Clusters’ in the ‘Coloring and highlighting’ Side Panel group.

- Cells can be highlighted using the Side Panel, for example by choosing one or multiple clusters, using specific ranges for cell annotations (see figure 17.11) or feature expression (see figure 17.13), or choosing specific samples. Once the cells are highlighted, they can be selected by choosing ‘Selected Highlighted’ from the plot right-click menu (see figure 17.4).
- Cells can be selected using the lasso tool (see figure 17.5).
- A larger amount of cells can be selected by making a small selection and choosing ‘Invert Selection’ from the plot right-click menu (see figure 17.4).

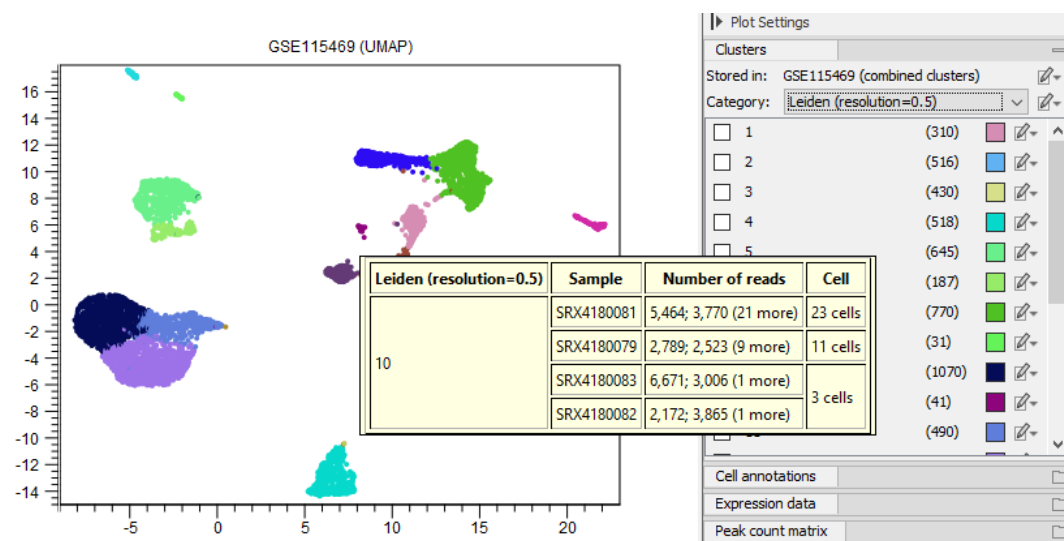


Figure 17.3: When the mouse hovers over the plot, a tooltip is displayed summarizing the nearby cells, containing information from the elements associated to the plot, and the feature expression for any selected features.

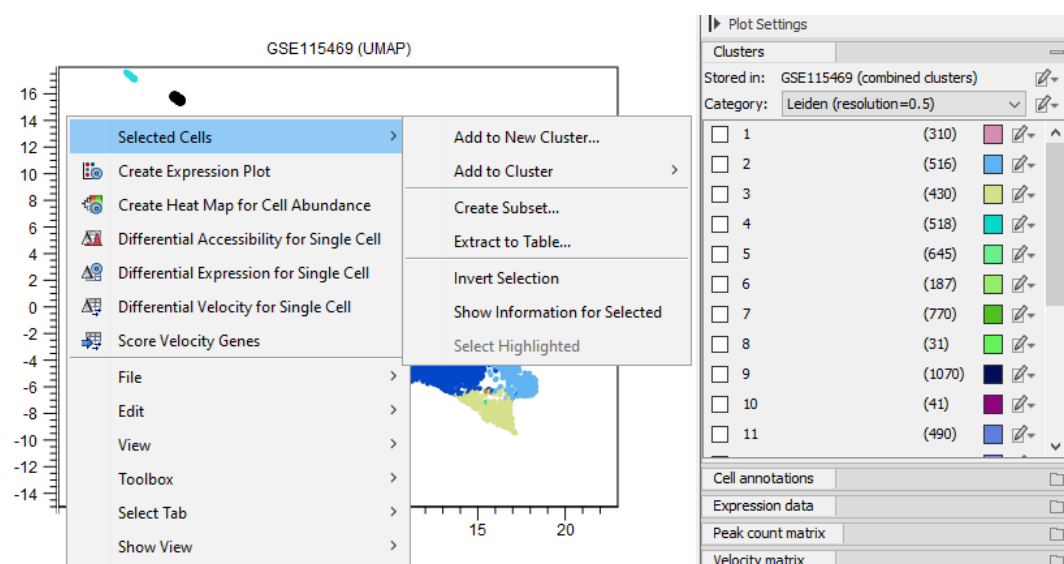


Figure 17.4: Available options in the plot right-click menu.

Working with clusters and annotations

Selected cells can be reassigned to existing clusters by choosing ‘Add to Cluster’ or added to a completely new cluster by choosing ‘Add to New Cluster’ (see figure 17.6) from the plot right-click menu (see figure 17.4), either as a free text or a cell type from the QIAGEN Cell Ontology (see section 8.1). When reassigning to existing clusters, the clusters are listed in the same order as in the Side Panel (see figure 17.8).

An existing cluster can be renamed by choosing ‘Rename Cluster’ in the cluster edit menu (see figure 17.9). A dialog similar to that in figure 17.6 opens, where the cluster can be renamed either using free text or a cell type from the QIAGEN Cell Ontology. When a cluster represents an ontology cell type, details about it (as those shown in figure 17.7) can be obtained by choosing

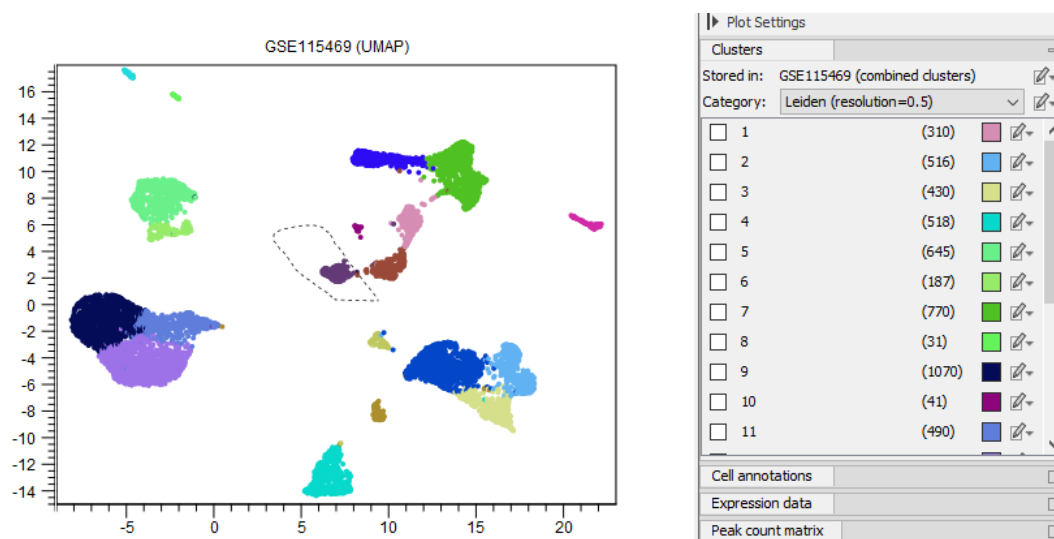


Figure 17.5: Cells can be selected using the lasso tool.

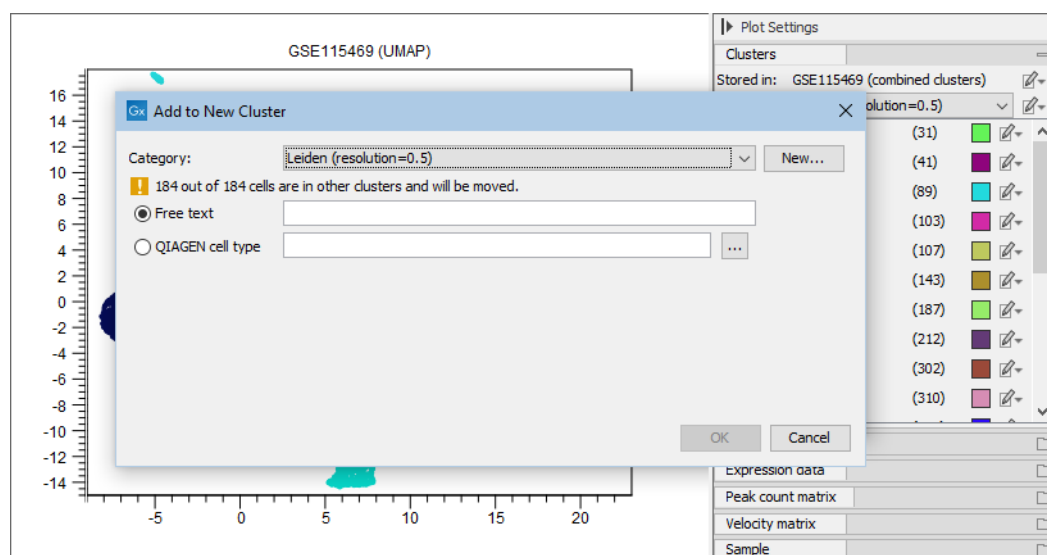


Figure 17.6: Cells can be added to a new cluster defined either by using free text or a cell type from the QIAGEN Cell Ontology by choosing 'QIAGEN cell type'. Clicking on the browse button ('...') opens up the ontology browser (see figure 17.7).

'Show in QIAGEN Cell Ontology' (see figure 17.9).

Any of the changes made using the above actions can be undone using the 'Undo' button. When clusters are changed, the plot name is marked with an '*' indicating that it contains an element that needs to be saved. By clicking 'Save', a new Cell Clusters element can be created.

The coloring of a cluster can be changed by clicking on the color box next to its name in the Side Panel. The newly chosen color can be saved using the View Settings, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Side_Panel_view_settings.html. Note that when the plot is closed and opened again, the default color is used, and to recover the custom color, the previously saved view settings need to be re-applied to the plot.

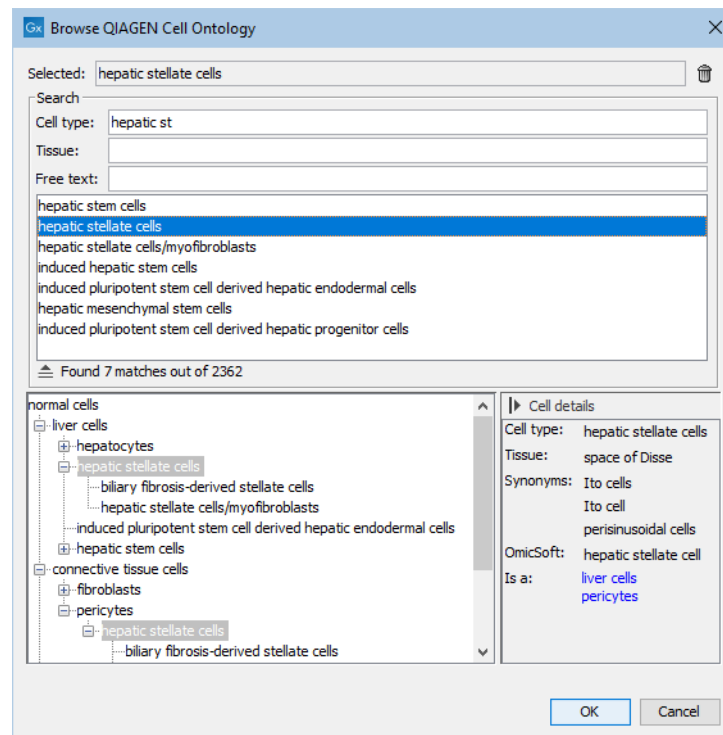


Figure 17.7: View of the QIAGEN Cell Ontology, showing details for ‘hepatic stellate cells’. The ‘Cell type’ text field can be used to quickly identify the desired cell type. Only cell types specific to a certain tissue can be shown by filling in ‘Tissue’. By using ‘Free text’, all cell types not containing the given text anywhere in their details are removed from the ontology structure shown at the bottom.

Cells can be also colored using information from cell annotations (see figures 17.10 and 17.11).

Visualizing expressions

Visualizing the expression of marker genes and selecting cells that express a set of marker genes above a specific value can be used to manually annotate the cell types. The plot can show the full expression of a specific gene across all cells using a gradient (see figure 17.12) or using one color for the cells with one or multiple gene expressions in a specific interval (see figure 17.13).

Multiple genes can be selected by:

- Adding them manually by choosing ‘Add to Selected Features’, see figure 17.12.
- Loading them from a file (see figure 17.14). The file should contain one gene name per line (see figure 17.15).
- Selecting them from other views showing feature expression, see below.

Selecting features in other views

Features can be selected from various elements showing their expression, and this is done in a synchronized manner, such that all opened elements showcasing feature expression will highlight the corresponding features, if available. This can be done from an Expression Matrix

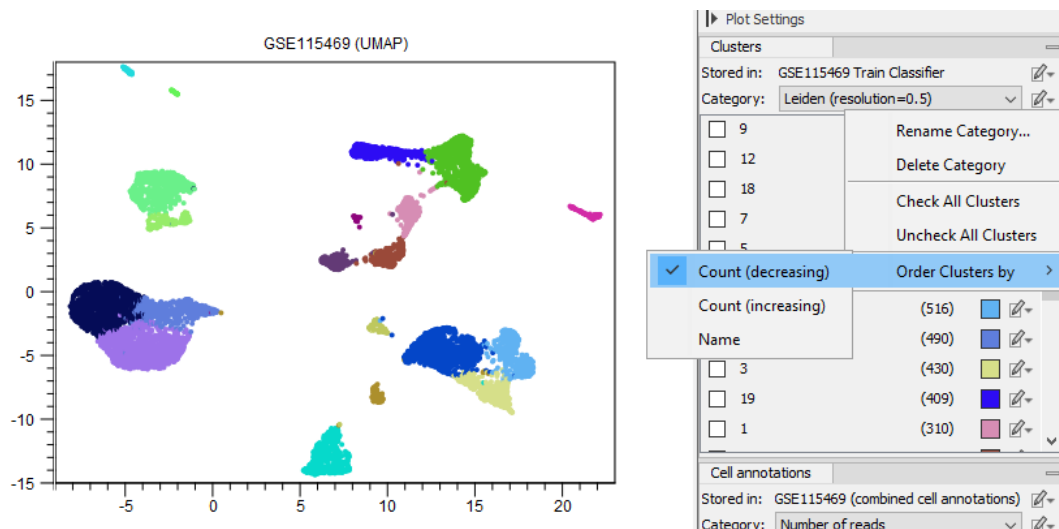


Figure 17.8: Setting the order of clusters in the Side Panel so the clusters with most cells are listed first.

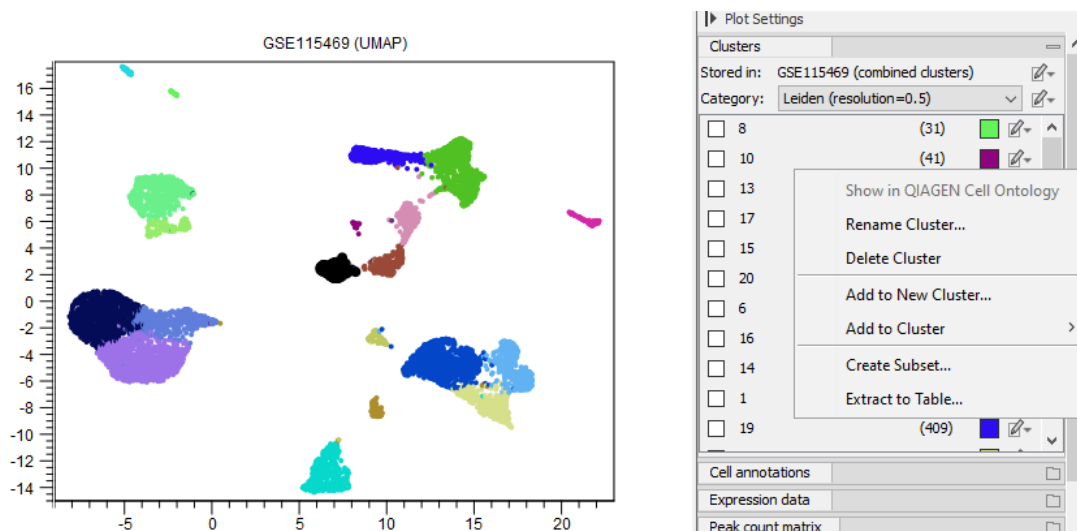


Figure 17.9: The available options for editing a single cluster from the Side Panel. 'Show in QIAGEN Cell Ontology' is grayed out because the corresponding cluster is not part of the ontology.

(see figure 17.16), a Dot Plot (see figure 17.17), a Heat Map (see figure 17.18), or a Violin Plot (see figure 17.19).

The UMAP plot, Dot Plot, Heat Map, and Violin Plot all show that 'ACTA2', 'COL1A1', 'TAGLN', 'COL1A2', 'COL3A1', 'SPARC', 'RBP1', 'DCN', 'MYL9' are highly expressed in cluster 10. These genes were identified as markers for hepatic stellate cells [MacParland et al., 2018] and cluster 10 is confirmed to contain hepatic stellate cells by the Predict Cell Types tool (see figure 17.11). This can be further confirmed by investigating a Cell Abundance Heat Map (see section 15.2), as shown in figure 17.20.

Once the cells co-expressing specific markers are highlighted, a new cluster with the corresponding cell type can be created, as described above.

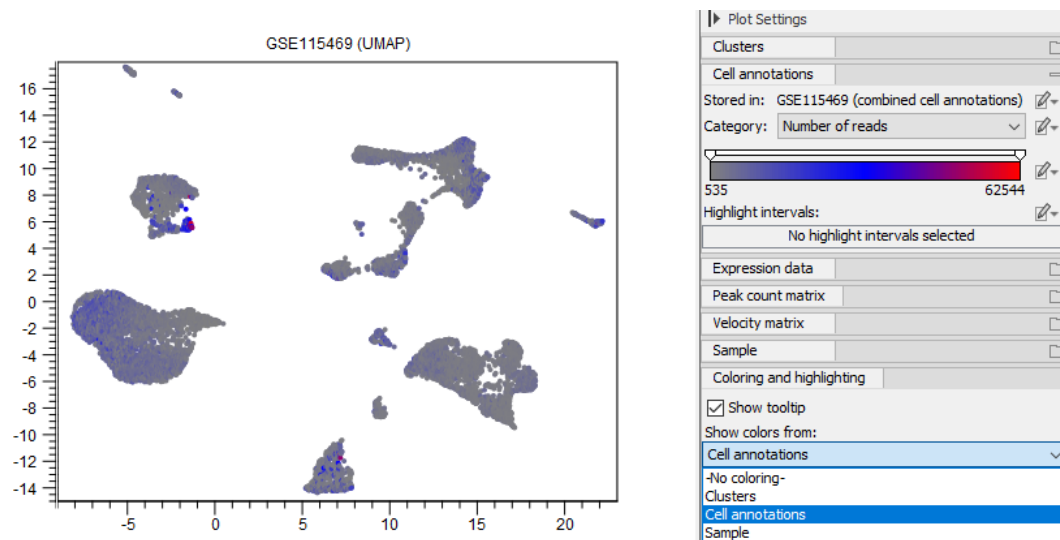


Figure 17.10: Cells are colored by the number of reads from the annotation produced by QC for Single Cell (see section 7.2). What information the cells are colored by can be chosen in the ‘Coloring and highlighting’ group at the bottom of the Side Panel.

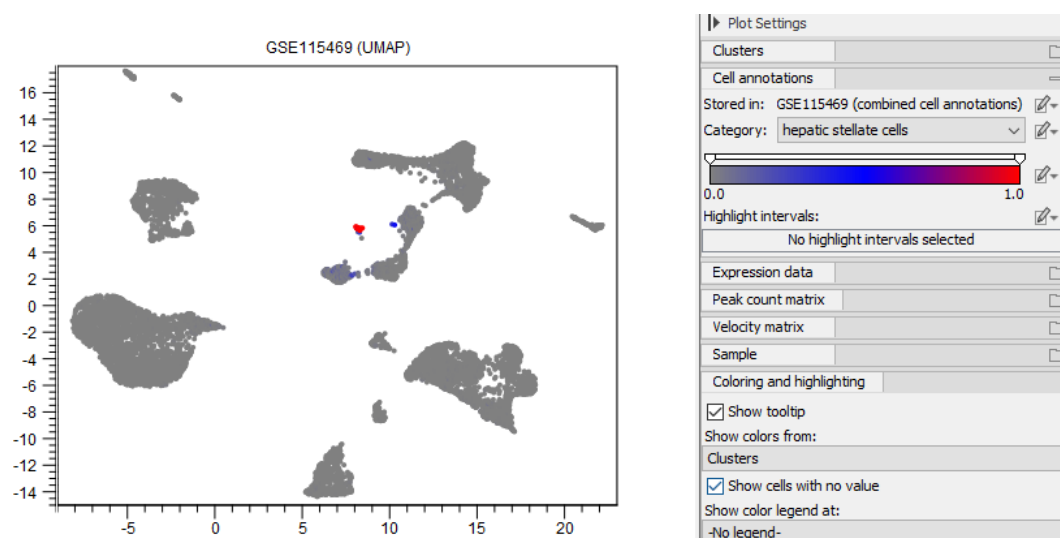


Figure 17.11: Cells are colored by the probability of having the type ‘hepatic stellate cells’ from the annotation produced by Predict Cell Types (see section 8.2) using the human pre-trained cell classifier (see chapter 2). Cells with a probability of at least 0.3 are highlighted. The highlight interval is inclusive.

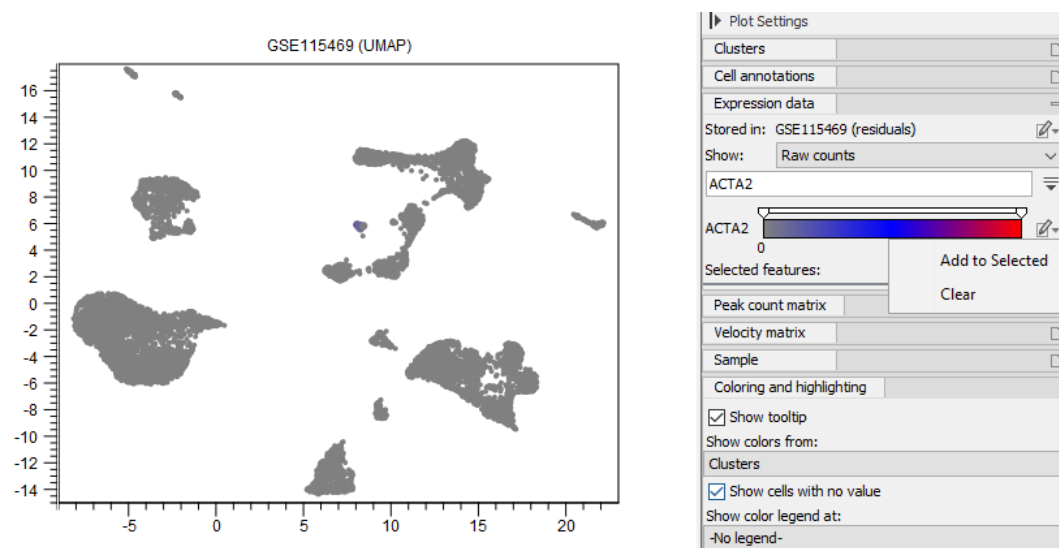


Figure 17.12: Cells are colored by the expression of ‘ACTA2’. The gene is selected by typing its name in the search box under the ‘Expression data’ group in the Side Panel. The relative coloring of the values can be changed by dragging the two knobs on the white slider above. ‘ACTA2’ can be added to ‘Selected features’ (see figure 17.13).

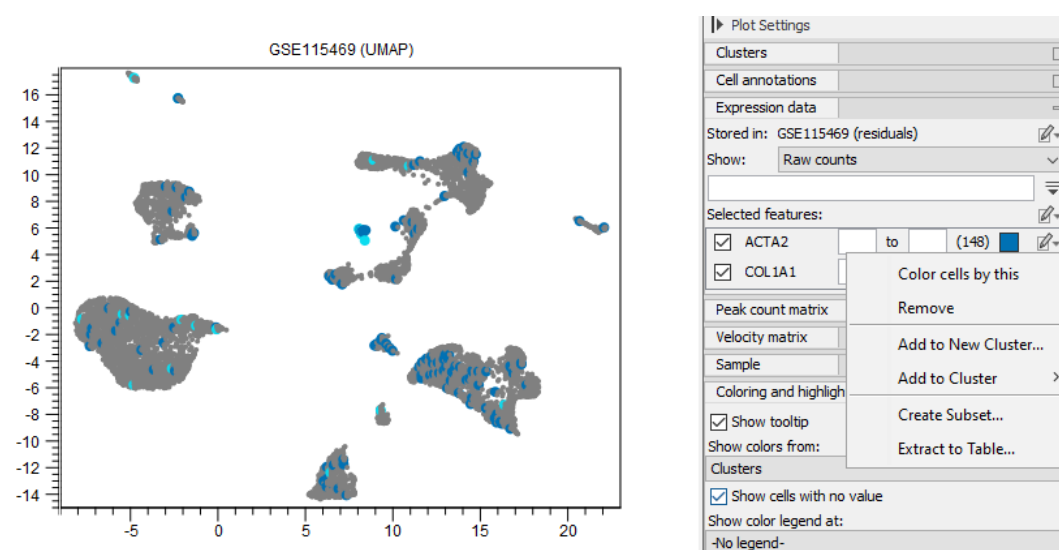


Figure 17.13: Cells that have an expression of at least 5 for ‘ACTA2’ and 1 for ‘COL1A1’ are colored in the plot. Different options are available for manipulating the cells expressing a particular feature. Selected cells can be required to express both genes, or just one of them, by choosing ‘Selected in all (intersection)’ or ‘Selected in any (union)’ at the bottom of the Side Panel. Note that the expression range is inclusive: setting the minimum expression to 0 will include the cells not expressing the gene. Choosing the interval $[0, 0]$ will highlight only the cells that do not express the gene.

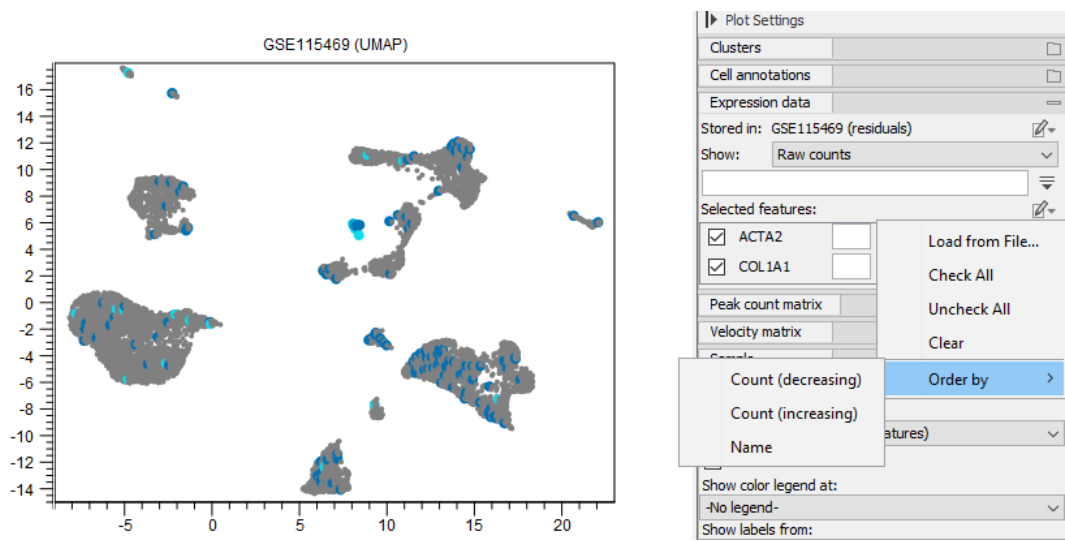


Figure 17.14: Side Panel options for 'Selected features'.

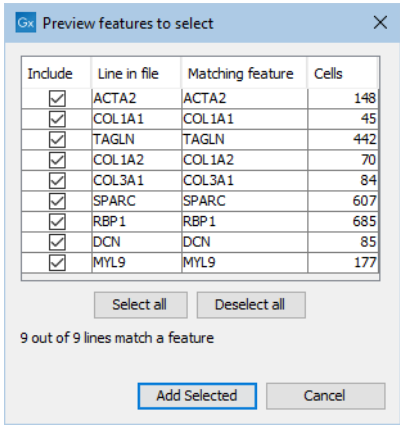


Figure 17.15: Dialog for loading feature names from a file.

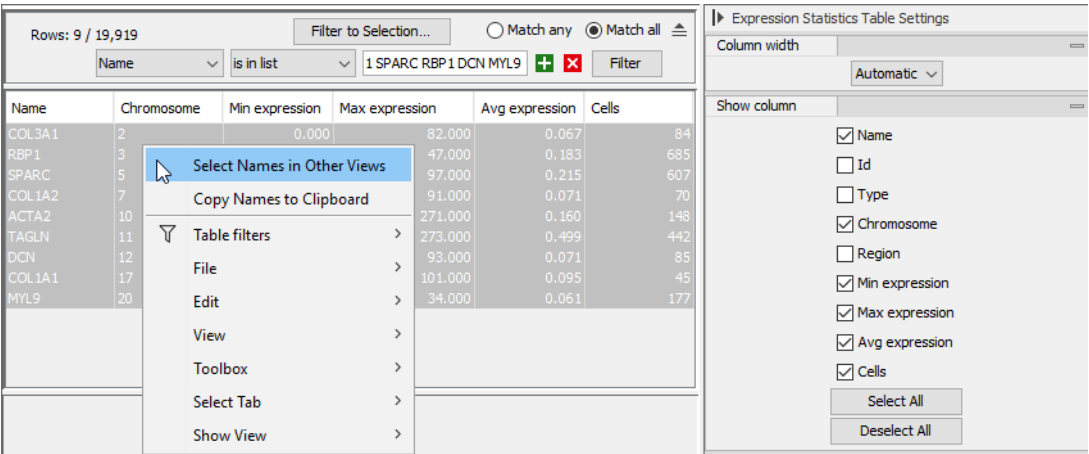


Figure 17.16: Right-click menu for selecting genes from the Expression Statistics Table view of an Expression Matrix to be synchronized with other views.

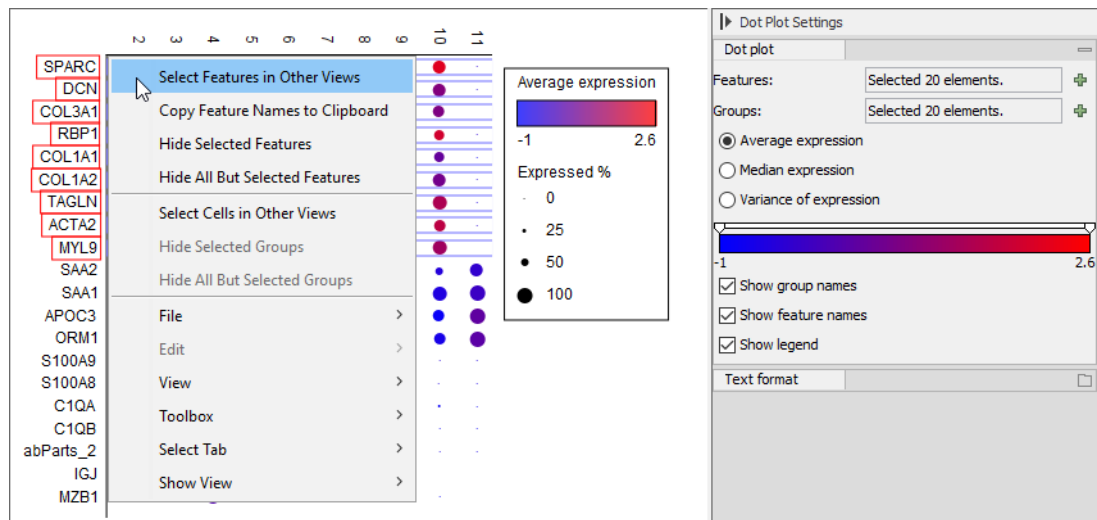


Figure 17.17: Right-click menu for selecting genes from a Dot Plot to be synchronized with other views.

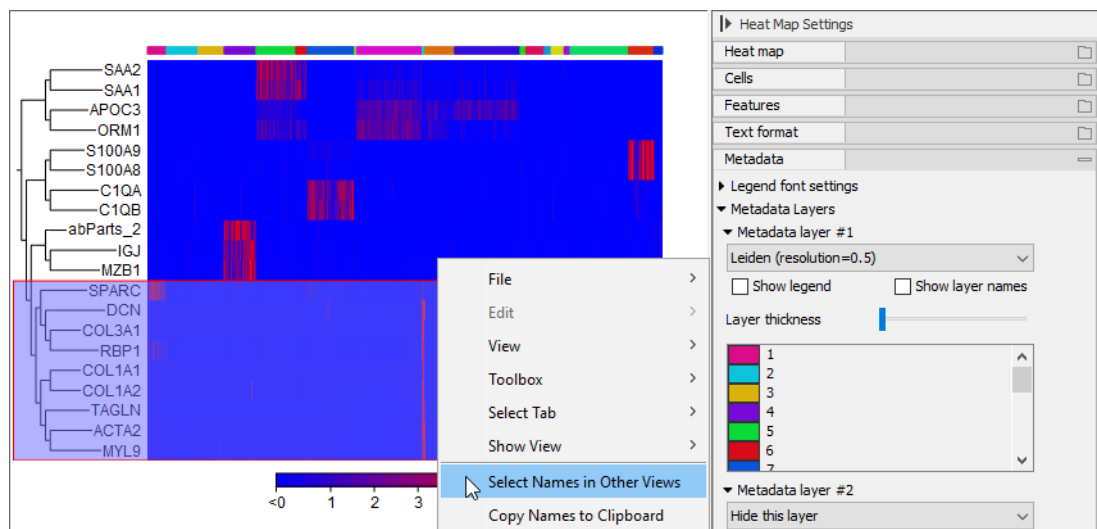


Figure 17.18: Right-click menu for selecting genes from a Heat Map to be synchronized with other views.

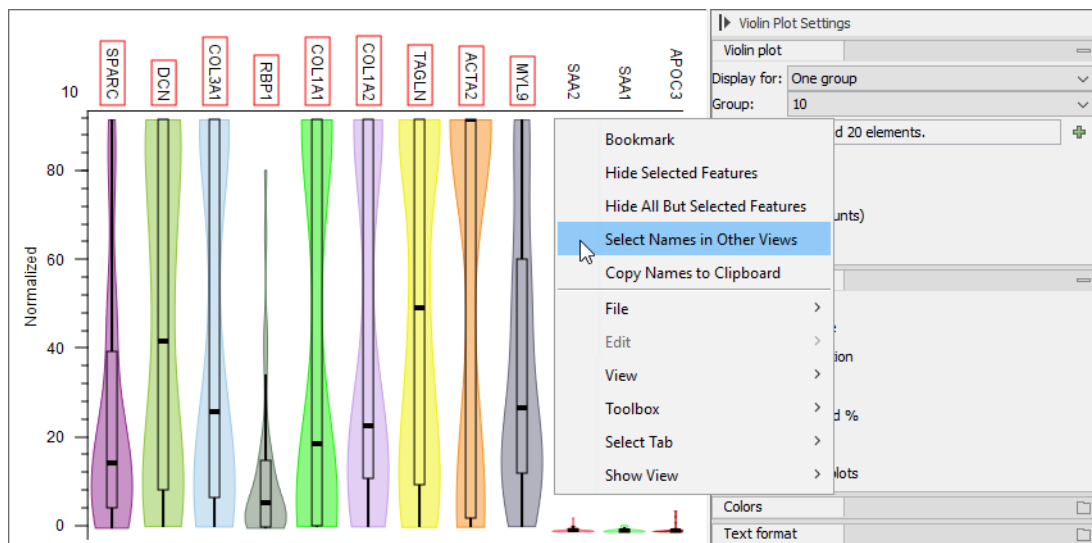


Figure 17.19: Right-click menu for selecting genes from a Violin Plot to be synchronized with other views.

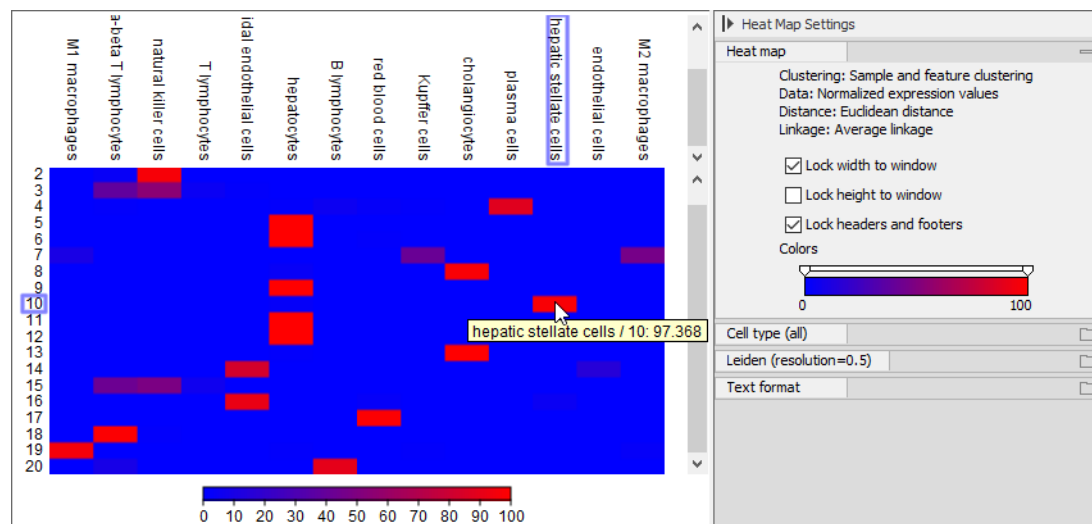


Figure 17.20: Cell Abundance Heat Map comparing the 'Leiden (resolution=0.5)' clusters to the cell types produced by Predict Cell Types. Hovering over a rectangle reveals the abundance of the selected combination.

17.2 Visualizing different types of matrices

The low-dimensional plots can show information stored in multiple types of matrices. To enable this, the relevant elements are associated with the plot by dragging and dropping them in the corresponding groups of the Side Panel:

- Expression Matrix (📊): ‘Expression data’ group;
- Expression Matrix with spliced and unspliced counts (📊📄): ‘Expression data’ group;
- Peak Count Matrix (📊📍): ‘Peak count matrix’ group;
- Velocity Matrix (📊🚀): ‘Velocity matrix’ group.

Note that only one element at a time can be associated with each group, but all groups can have associations concurrently.

The Expression Matrix with spliced and unspliced counts (📊📄) shares the same group as the Expression Matrix (📊) because it is an extension which additionally contains separate information about the spliced and unspliced counts for each cell and gene.

The Manual Annotation section showcases how the expression of genes found in an Expression Matrix (📊) can be used for coloring and selecting cells (see section 17.1). The remaining matrices can display:

- Multiple types of expression values for the same feature (figure 17.21);
- Multiple types of features (figure 17.22);
- Other type of information (figure 17.23);

all of which can be chosen from a drop down menu.

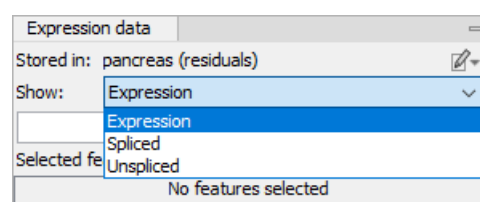


Figure 17.21: The options available in the ‘Expression data’ group for an Expression Matrix with spliced and unspliced counts.

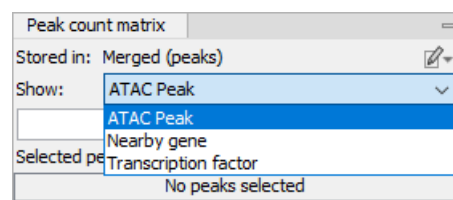


Figure 17.22: The options available in the ‘Peak count matrix’ group for a Peak Count Matrix.

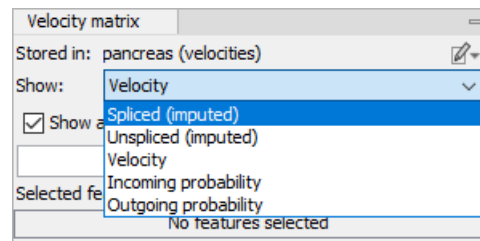


Figure 17.23: The options available in the ‘Velocity matrix’ group for a Velocity Matrix.

The same functionality for coloring and selecting cells available for expression data, as detailed in the Manual Annotation section, is also available for all the options in the drop down menus.

The transition probabilities present in the Velocity Matrix (figure 17.23) are the only options that are cell-based and not feature-based. To color cells after such probabilities, one cell can be selected in the plot using the lasso tool (see figure 17.5) and the option to show the probabilities can be chosen from the plot right-click menu (figure 17.24 and 17.25).

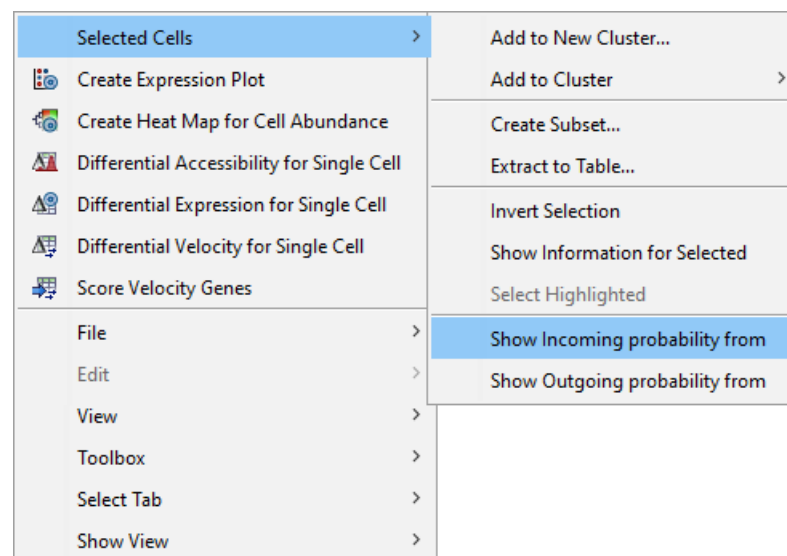


Figure 17.24: Showing transition probabilities from the plot right-click menu. Options are available when just one cell is selected.

The ‘Incoming probability’ colors cells that have a non-zero probability of transitioning towards the selected cell by that probability. The ‘Outgoing probability’ colors cells that the selected cell has a non-zero probability of transitioning towards by that probability. Note that probabilities can be negative, indicating that the cell is transitioning in the opposite direction.

When a Velocity Matrix is associated with a Dimensionality Reduction Plot, the ‘Show arrows’ option can be used to display the projected velocities at cellular level. Each arrow is obtained from the cell’s ‘Outgoing probabilities’ and summarizes them into a projected direction for the cell and its speed of movement. The arrows can reveal differences between near-terminal cells, where arrows are short, and transient cells, where arrows are longer (figure 17.26).

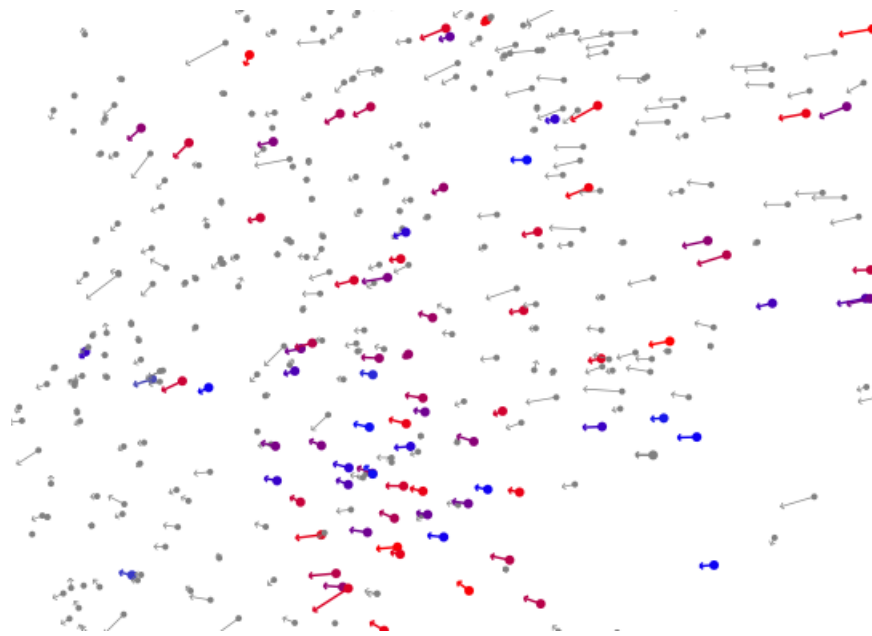


Figure 17.25: Coloring by incoming transition probabilities from a selected cell

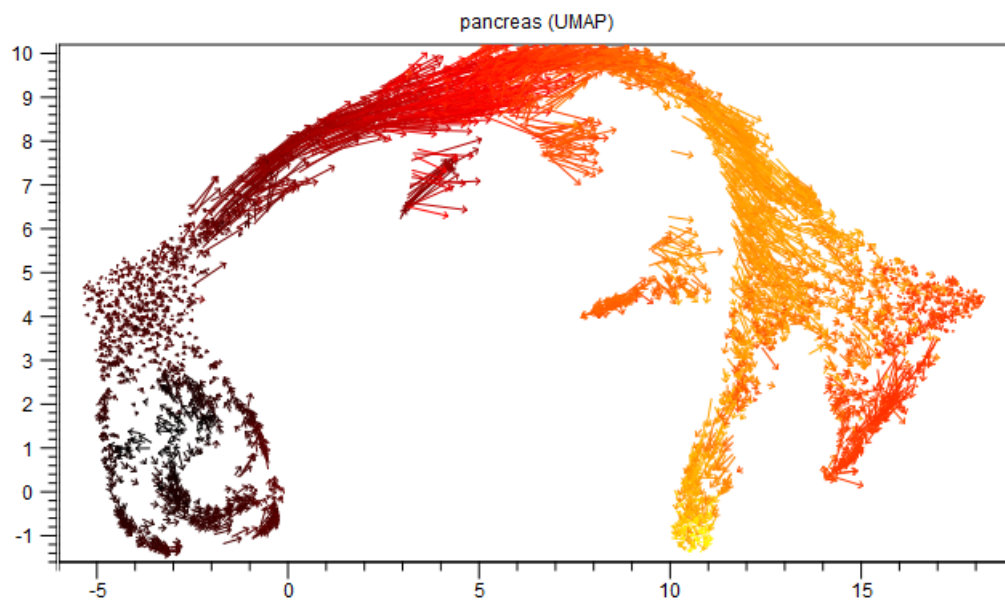


Figure 17.26: UMAP plot of the pancreas data set [Bastidas-Ponce et al., 2019] built-in scVelo [Bergen et al., 2020]. Arrows show the direction and speed of movement of an individual cell. The real time cells experience as they differentiate is approximated by the latent time, shown here in the 0 (black) to 1 (yellow) range.

17.3 Create Subset

By choosing ‘Create Subset’ from the low-dimensional plot right-click menu (see figure 17.4), new matrices and associated elements, as relevant, can be created, containing only the corresponding selected cells (see figure 17.27)

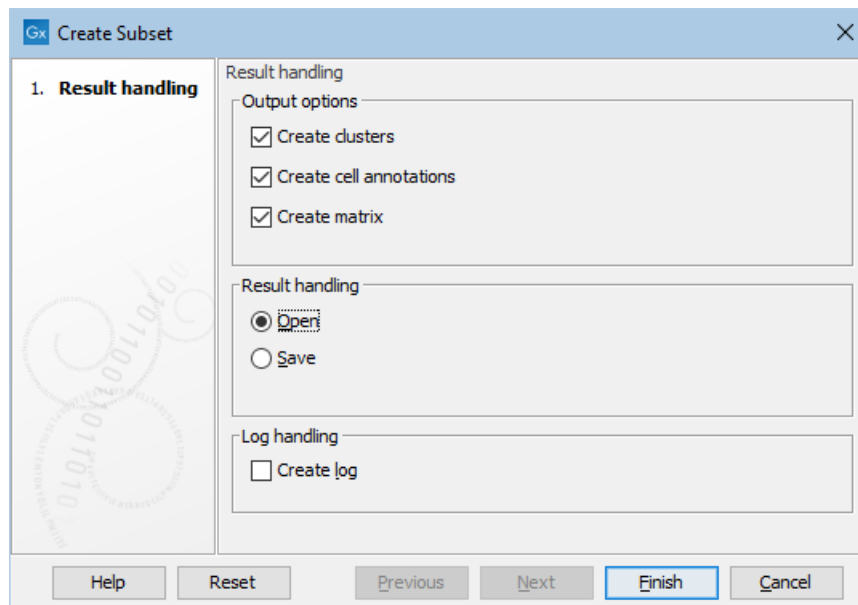


Figure 17.27: Dialog for Create Subset when started from a Dimensionality Reduction Plot. The Output options allow choosing which of the elements that are associated with the plot the subset will be created for.

17.4 Extract to Table

By choosing 'Extract to Table' from the low-dimensional plot right-click menu (see figure 17.4), a table can be created, with the relevant information for the corresponding selected cells (see figures 17.28 and 17.29).

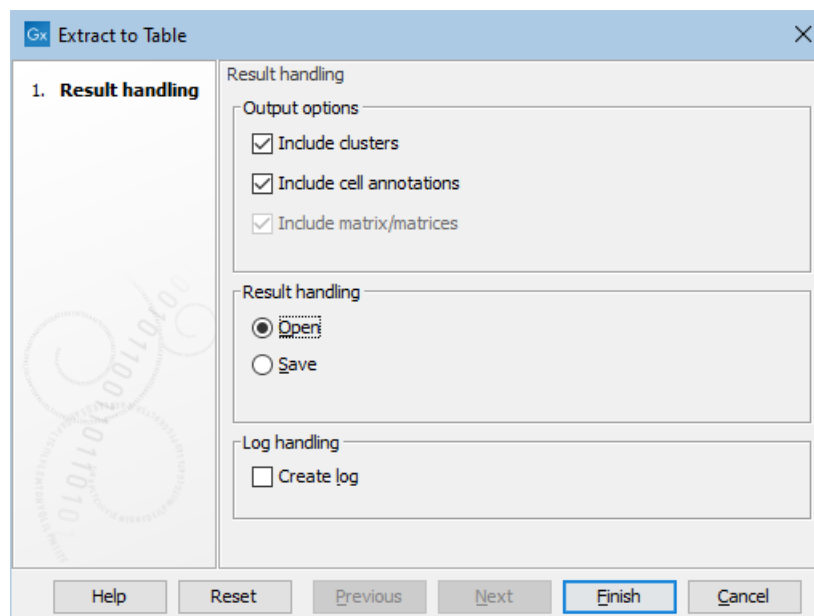


Figure 17.28: Dialog for Extract to Table. The Output options allow choosing which available information will be included in the resulting plot.

Rows: 40							Filter to Selection...		Filter
Sample	Barcode	Leiden (r...	hepatic s...	ACTA2	COL1A1	TAGLN			
SRX4180082	CTCGTCAGTGATGTGG	10	4.430E-3	0.000	2.000	1.000			
SRX4180081	TTGTAGGCAGGTTTCG	10	1.000	27.000	56.000	54.000			
SRX4180081	GTTCTCGGTAACGCGA	10	1.000	271.000	88.000	193.000			
SRX4180081	GGGTCTGCAAGTCTGT	10	1.000	0.000	1.000	1.000			
SRX4180081	GATCTAGAGTACGTTT	10	1.000	0.000	0.000	6.000			
SRX4180081	TCTGAGACACAGACTT	10	1.000	15.000	28.000	14.000			
SRX4180081	TCTGGAAGTTACGCGC	10	1.000	19.000	0.000	35.000			
SRX4180083	AGACGTTGTTACAGGCC	10	1.000	7.000	24.000	9.000			
SRX4180081	GTATCTTCATTCTCAT	10	1.000	0.000	4.000	2.000			
SRX4180079	GTCACGGGTATGAATG	10	0.267	0.000	0.000	0.000			
SRX4180081	TGCTGCTAGAGTAAGG	10	1.000	15.000	1.000	23.000			
SRX4180081	GATGCTAGTAATCACC	10	1.000	49.000	0.000	44.000			
SRX4180083	AGGTCCGTCGCCTGTT	10	1.000	17.000	18.000	14.000			
SRX4180083	GTGAAGGTCAACACCT	10	1.000	19.000	12.000	22.000			
SRX4180081	ACAGCTAGTTACAGACT	10	1.000	19.000	36.000	18.000			
SRX4180079	TTCTCCTTCCCGGATG	10	1.000	5.000	0.000	4.000			
SRX4180081	CCTATTACATCGGTTA	10	1.000	227.000	68.000	185.000			
SRX4180079	ATTCTACTCTGCCAGG	10	1.000	3.000	0.000	4.000			
SRX4180081	CACAAACAGGGAACA	10	1.000	59.000	101.000	101.000			
SRX4180079	CATGACAAGGGCACTA	10	1.000	25.000	1.000	120.000			
SRX4180079	AGCTCTCCAACAACCT	10	0.129	0.000	1.000	0.000			
SRX4180082	CAGATCAAGTGTTC	10	1.000	18.000	0.000	66.000			
SRX4180079	AGAATAGGTTTCGCTC	10	0.064	0.000	1.000	0.000			

Table Settings

Column width: Automatic

Show column:

- ☒ Sample
- ☒ Barcode
- ☐ Leiden (resolution=0.1)
- ☐ Leiden (resolution=0.2)
- ☐ Leiden (resolution=0.3)
- ☐ Leiden (resolution=0.4)
- ☒ Leiden (resolution=0.5)
- ☐ Leiden (resolution=0.6)
- ☐ Leiden (resolution=0.7)
- ☐ Leiden (resolution=0.8)
- ☐ Leiden (resolution=0.9)
- ☐ Leiden (resolution=1.0)
- ☐ Leiden (resolution=1.1)
- ☐ Leiden (resolution=1.2)
- ☐ Leiden (resolution=1.3)

Figure 17.29: The output of Extract to Table. Only the ‘Sample’, ‘Barcode’, ‘Leiden (resolution=0.5)’ cluster, probability of ‘cardiomyocytes’, and expression of ‘Myl2’ columns are shown, as selected in the Side Panel.

17.5 Launching of Create Expression Plot

By choosing ‘Create Expression Plot’ from the low-dimensional plot right-click menu (see figure 17.4), the tool Create Expression Plot can be started (see section 9.2). The dialog is automatically filled in with the relevant information from the plot (see figure 17.30).

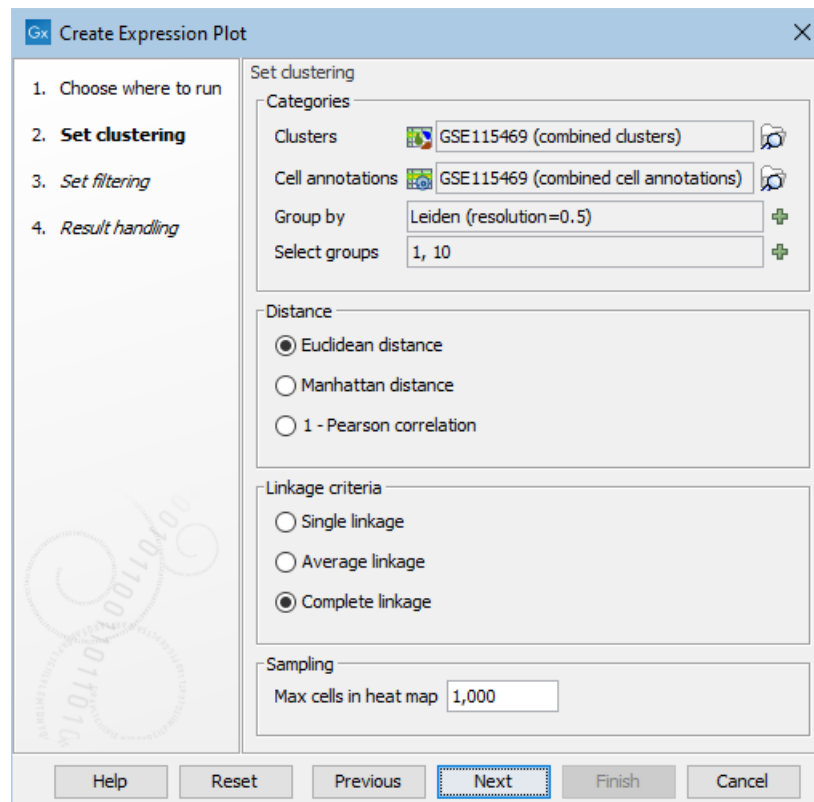


Figure 17.30: Dialog for Create Expression Plot (see section 9.2) when started from the plot. The clusters and annotations that are associated with the plot are automatically filled in. The selected cluster is automatically added to ‘Group by’.

17.6 Launching of Create Heat Map for Cell Abundance

By choosing ‘Create Heat Map for Cell Abundance’ from the low-dimensional plot right-click menu (see figure 17.4), the tool Create Heat Map for Cell Abundance can be started (see section 15.2). The dialog is automatically filled in with the relevant information from the plot (see figure 17.31).

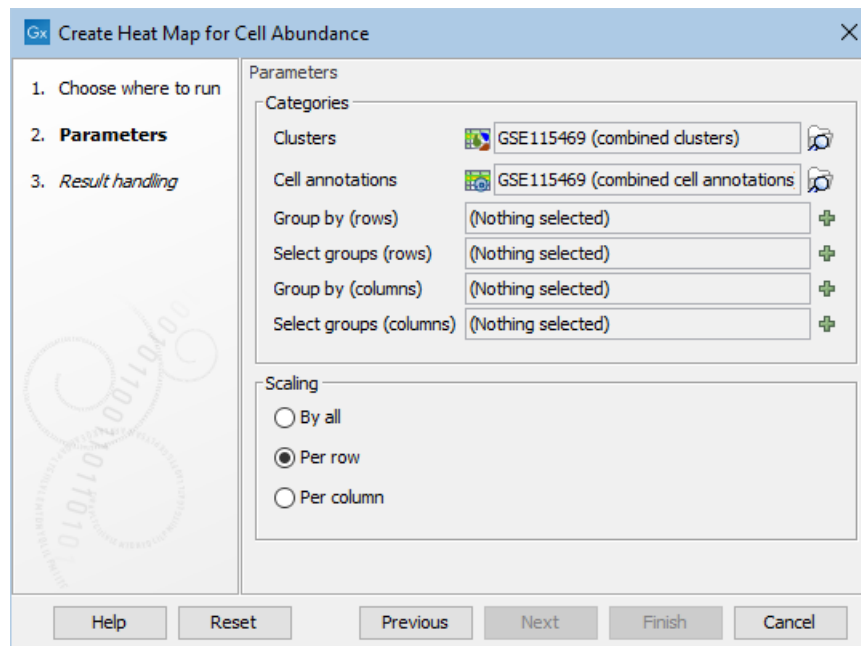


Figure 17.31: *Dialog for Create Heat Map for Cell Abundance* (see section [15.2](#)) when started from the plot. The clusters and annotations that are associated with the plot are automatically filled in. Note that ‘Group by’ and ‘Scaling’ need to be configured to obtain the desired analysis.

17.7 Launching of Differential Accessibility for Single Cell

By choosing ‘Differential Accessibility for Single Cell’ from the low-dimensional plot right-click menu (see figure [17.4](#)), the tool Differential Accessibility for Single Cell can be started (see section [12.3](#)). The dialog is automatically filled in with the relevant information from the plot (see figure [17.32](#)).

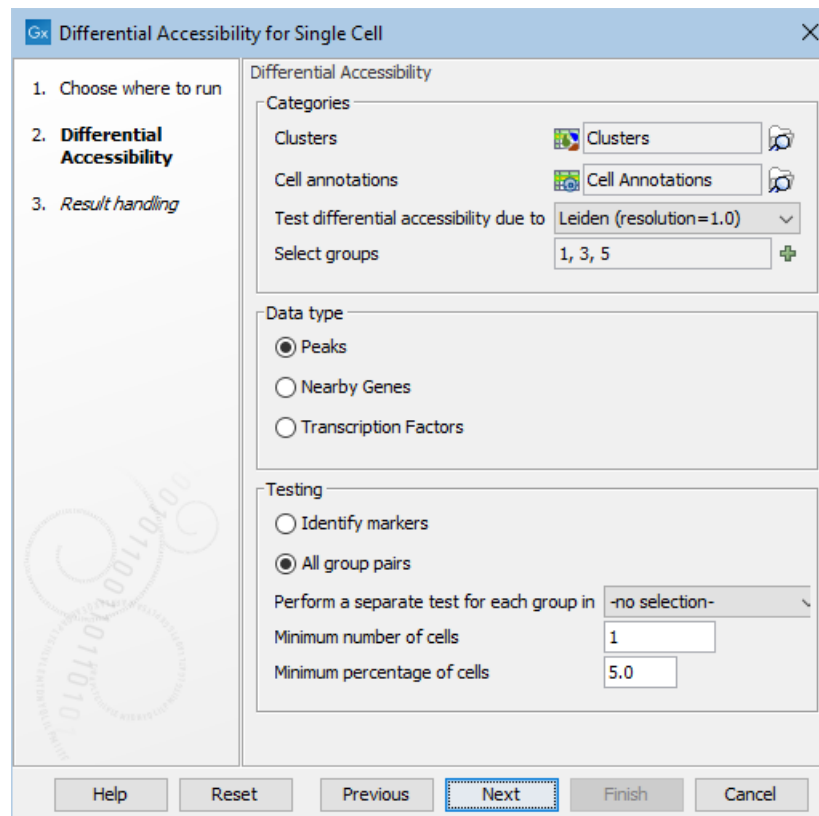


Figure 17.32: Dialog for Differential Accessibility for Single Cell (see section 12.3) when started from the plot. The clusters and annotations that are associated with the plot are automatically filled in. The selected clusters are automatically added to ‘Select groups’.

17.8 Launching of Differential Expression for Single Cell

By choosing ‘Differential Expression for Single Cell’ from the low-dimensional plot right-click menu (see figure 17.4), the tool Differential Expression for Single Cell can be started (see section 9.1). The dialog is automatically filled in with the relevant information from the plot (see figure 17.33).

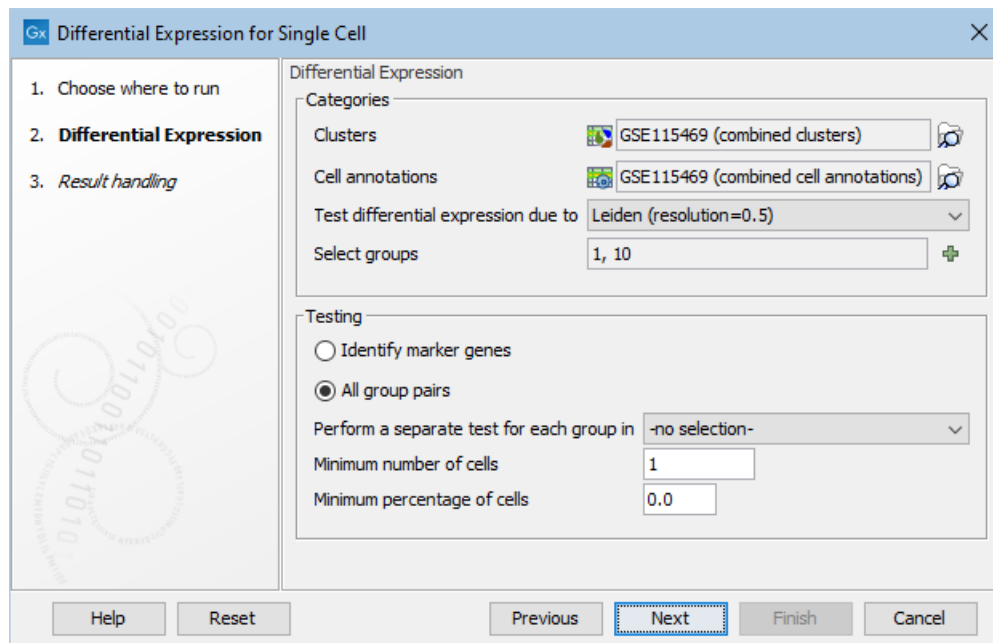


Figure 17.33: Dialog for Differential Expression for Single Cell (see section 9.1) when started from the plot. The clusters and annotations that are associated with the plot are automatically filled in. The selected clusters are automatically added to ‘Select groups’.

17.9 Launching of Differential Velocity for Single Cell

By choosing ‘Differential Velocity for Single Cell’ from the low-dimensional plot right-click menu (see figure 17.4), the tool Differential Velocity for Single Cell can be started (see section 10.2). The dialog is automatically filled in with the relevant information from the plot (see figure 17.34).

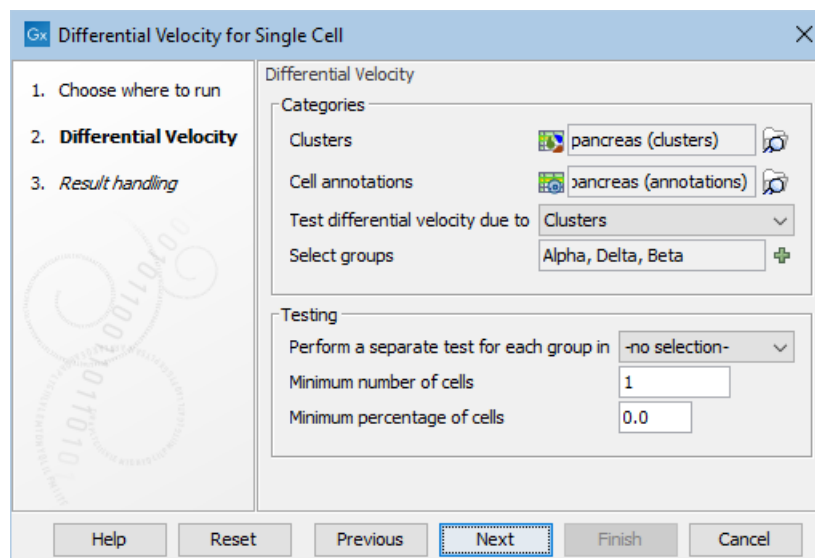


Figure 17.34: Dialog for Differential Velocity for Single Cell (see section 10.2) when started from the plot. The clusters and annotations that are associated with the plot are automatically filled in. The selected clusters are automatically added to ‘Select groups’.

17.10 Launching of Score Velocity Genes

By choosing ‘Score Velocity Genes’ from the low-dimensional plot right-click menu (see figure 17.4), the tool Score Velocity Genes can be started (see section 10.3). The dialog is automatically filled in with the relevant information from the plot (see figure 17.35).

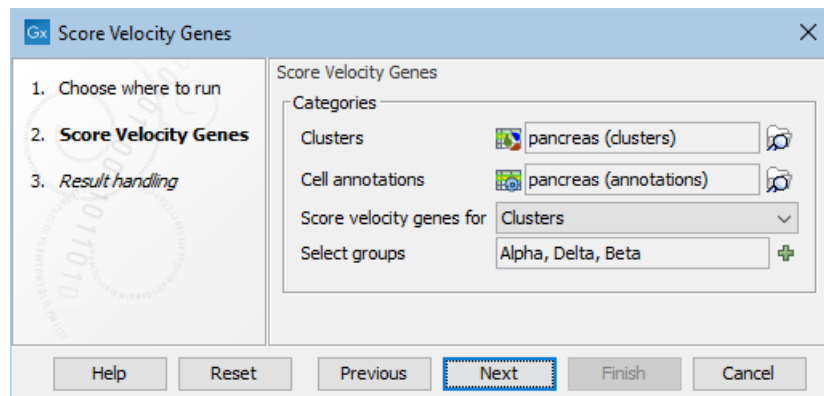


Figure 17.35: Dialog for Score Velocity Genes (see section 10.3) when started from the plot. The clusters and annotations that are associated with the plot are automatically filled in. The selected clusters are automatically added to ‘Select groups’.

Chapter 18

Utility tools



Contents

18.1 Combine Cell Annotations	231
18.2 Update Cell Annotations	232
18.3 Convert Metadata to Cell Annotations	233
18.4 Combine Cell Clusters	234
18.5 Update Cell Clusters	234
18.6 Add Information to Plot	236
18.7 Update Single Cell Sample Name	236

18.1 Combine Cell Annotations

Combine Cell Annotations can be found at:

Tools | Single Cell Analysis  | **Utility Tools**  | **Combine Cell Annotations** 

The tool takes as input multiple **Cell Annotations**  and outputs a single **Cell Annotations**  element. This can be useful to reduce the number of elements needed to describe a set of cells.

The combining is very flexible. For example, it supports:

- Different pieces of information for the same cells. An example could be QC metrics from QC for Single Cell with cell type probabilities from Predict Cell Types.
- The same pieces of information from different cells.
- Different pieces of information from different cells.

Cells are considered to be the same if they have the same sample and barcode. Note that if two Cell Annotations describe the same cell with contradictory information in the same category, then combining will fail with a warning.

18.2 Update Cell Annotations

The Update Cell Annotations tool takes one **Cell Annotations**  element as input and outputs a new **Cell Annotations**  element containing updated categories and/or annotations.

Update Cell Annotations is available from:

Tools | Single Cell Analysis  | **Utility Tools**  | **Update Cell Annotations** 


In the first wizard step ‘Update categories’, the following options can be adjusted:


- **Remove categories.** The selected categories are removed.
- **Remove all empty categories.** Categories without any cell annotations are removed. Categories are also removed if empty after removal of cell annotations (see below).
- **Rename categories.** One or more categories are renamed. The **Add** button can be used to add additional categories to be renamed. Note that multiple categories cannot be renamed to the same name.
- **Rename all categories.** All categories are renamed according to the given pattern. This can be used for both prepending and appending to the name. E.g., if set to ‘MyPrefix {categoryName} MySuffix’, a category ‘MyCategory’ will be renamed to ‘MyPrefix MyCategory MySuffix’. This is done after the changes specified by **Rename categories**.


In the next wizard step ‘Update annotations’, annotations containing text can be updated. Boolean (yes/no) and numeric annotations cannot be updated. The following options can be adjusted:

- **Update annotations for.** The updates can be applied to:
 - **No categories.** Skip updating cell annotations.
 - **All categories.** Update the cell annotations for all categories.
 - **Selected category.** Update the cell annotations only for the category selected in **Update annotations for category**.
- **Remove annotations.** The selected annotations are removed.
- **Change annotations.** One or more annotations are changed. The **Add** button can be used to add additional annotations to be changed. Note that a annotation may be changed to an existing annotation and multiple annotations may be changed to the same annotation.

The options above can be mixed and matched to obtain the desired output. For example, a category can be first renamed and then the annotations it contains can be updated.

The Update Cell Annotations tool can also be started from the different views of the **Cell Annotations**  element, with dialogs automatically filled in with relevant selections:

- From the annotations table view :
 - annotations can be removed or changed from the right-click menu. Multiple rows can be selected before using the right-click menu.

- A category can be removed or renamed using the **Remove** and **Rename** buttons from the Side Panel.
- From the cell-level view () , the category over which the mouse hovers can be removed or renamed from the right-click menu.


18.3 Convert Metadata to Cell Annotations

Metadata is often present at both the sample and cell level. However, it is always possible to convert sample level metadata into cell level metadata. For example, the knowledge that a sample comes from ‘Lab A’ can be captured by annotating the entire sample with ‘Lab A’, or alternatively by annotating all the cells in the sample with ‘Lab A’.

For simplicity, most tools in the CLC Single Cell Analysis Module only accept cell level metadata. Convert Metadata to Cell Annotations is provided to easily transform sample level metadata in the form of Metadata Tables (see <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Metadata.html> for more details) into cell level metadata in the form of Cell Annotations. Convert Metadata to Cell Annotations can be found at:

Tools | Single Cell Analysis () | **Utility Tools** () | **Convert Metadata to Cell Annotations** ()

The tool can take an **Expression Matrix** () / () , **Velocity Matrix** () , or **Peak Count Matrix** () as input and produces a single **Cell Annotations** () element. An example of this process is shown in figure 18.1.



MyExpressionMatrix X		
Sample metadata table		
Fixed Fields		
▶ Name Edit		
▶ Description Edit		
▼ Metadata Refresh		
From 'Sample metadata table' Refresh Delete Edit		
This Expression data is 'Sample data' for:		
Sample :	MyExpressionMatrix	
Lab :	Lab A	
Date :	11.12.2010	

Rows: 20				
Sample	Barcode	Sample	Lab	Date
Sample	AAC	MyExpressionMatrix	Lab A	11.12.2010
Sample	AAG	MyExpressionMatrix	Lab A	11.12.2010
Sample	AAT	MyExpressionMatrix	Lab A	11.12.2010
Sample	ACG	MyExpressionMatrix	Lab A	11.12.2010
Sample	AGG	MyExpressionMatrix	Lab A	11.12.2010
Sample	ATG	MyExpressionMatrix	Lab A	11.12.2010
Sample	ACT	MyExpressionMatrix	Lab A	11.12.2010
Sample	AGT	MyExpressionMatrix	Lab A	11.12.2010
Sample	ATT	MyExpressionMatrix	Lab A	11.12.2010
Sample	CCG	MyExpressionMatrix	Lab A	11.12.2010
Sample	GCG	MyExpressionMatrix	Lab A	11.12.2010
Sample	TCG	MyExpressionMatrix	Lab A	11.12.2010
Sample	CGG	MyExpressionMatrix	Lab A	11.12.2010
Sample	GGG	MyExpressionMatrix	Lab A	11.12.2010
Sample	TGG	MyExpressionMatrix	Lab A	11.12.2010
Sample	CTG	MyExpressionMatrix	Lab A	11.12.2010
Sample	GTG	MyExpressionMatrix	Lab A	11.12.2010
Sample	TTG	MyExpressionMatrix	Lab A	11.12.2010
Sample	CCT	MyExpressionMatrix	Lab A	11.12.2010
Sample	GCT	MyExpressionMatrix	Lab A	11.12.2010

Figure 18.1: A Metadata Table (left) annotates a sample as being produced by ‘Lab A’ on ‘11.12.2010’. Convert Metadata to Cell Annotations converts this metadata into Cell Annotations (right) where each cell in the sample is annotated with the same information.

The tool does not require the sample level metadata to be explicitly provided. Instead:

In a workflow the sample level metadata is taken from a Metadata Table provided to the workflow, if present.

Otherwise the sample level metadata is collated from all the Metadata Tables that reference the input Expression Matrix. If multiple such tables exist, their annotations are combined.

If the annotations are in conflict, for example one table says the sample has ‘Lab = Lab A’ and another says ‘Lab = Lab B’, then the Lab will be missing (and hence unknown) in the combined table. Note that after connecting to a CLC Server, additional metadata tables, only present on the server, may be found by the tool.

18.4 Combine Cell Clusters

Combine Cell Clusters can be found at:

Tools | Single Cell Analysis  | **Utility Tools**  | **Combine Cell Clusters** 



The tool takes as input multiple **Cell Clusters**  and outputs a single **Cell Clusters**  element. This can be useful to reduce the number of elements needed to describe a set of cells.

The combining is very flexible. For example, it supports:

- Different pieces of information for the same cells. An example could be predicted cell types from Predict Cell Types with clusters from Cluster Single Cell Data.
- The same pieces of information from different cells.
- Different pieces of information from different cells.

Cells are considered to be the same if they have the same sample and barcode. Note that if two Cell Clusters describe the same cell with contradictory information in the same category, then combining will fail with a warning.

18.5 Update Cell Clusters

The Update Cell Clusters tool takes one **Cell Clusters**  element as input and outputs a new **Cell Clusters**  element containing updated categories and/or clusters.

Update Cell Clusters is available from:

Tools | Single Cell Analysis  | **Utility Tools**  | **Update Cell Clusters** 

In the first wizard step ‘Update categories’, the following options can be adjusted:

- **Remove categories.** The selected categories are removed.
- **Remove all empty categories.** Categories without any clusters are removed. Categories are also removed if empty after removal of clusters (see below).
- **Rename categories.** One or more categories are renamed. The **Add** button can be used to add additional categories to be renamed. Note that multiple categories cannot be renamed to the same name.
- **Rename all categories.** All categories are renamed according to the given pattern. This can be used for both prepending and appending to the name. E.g., if set to ‘MyPrefix {categoryName} MySuffix’, a category ‘MyCategory’ will be renamed to ‘MyPrefix MyCategory MySuffix’. This is done after the changes specified by **Rename categories**.


In the next wizard step ‘Update clusters’, the clusters can be updated. The following options can be adjusted:



- **Update clusters for.** The updates can be applied to:
 - **No categories.** Skip updating clusters.
 - **All categories.** Update the clusters for all categories.
 - **Selected category.** Update the clusters only for the category selected in **Update clusters for category**.
- **Remove clusters.** The selected clusters are removed.
- **Remove all empty clusters.** Clusters with no cells are removed.
- **Rename clusters.** One or more clusters are renamed. The **Add** button can be used to add additional clusters to be renamed.
- **Map clusters to QIAGEN Cell Ontology.** When this is checked, clusters will be translated, if possible, to the QIAGEN Cell Ontology (see section 8.1). The translation attempts to match each cluster with a QIAGEN cell type based on the name and known synonyms. For example, ‘alveolar epithelial cells’ are also called ‘pneumocytes’. If this option is selected, the ‘alveolar epithelial cells’ cluster, if present, will be named ‘pneumocytes’. This option can be useful when standardizing clusters from different sources. It is especially recommended if clusters will be used to extend a QIAGEN Cell Type Classifier using the Train Cell Type Classifier tool (section 8.3).

If two distinct clusters are given the same name, they are combined into one. This can happen if:

- A cluster is renamed to an existing name.
- Two clusters are renamed to the same name.
- Two cluster names are synonyms for the same QIAGEN cell type and clusters are mapped to the ontology.

The options above can be mixed and matched to obtain the desired output. For example, a cluster can be first renamed and then the new name can be mapped to the ontology.

The Update Cell Clusters tool can also be started from the different views of the **Cell Clusters**  element, with dialogs automatically filled in with relevant selections:




- From the clusters table view :
 - Clusters can be removed or renamed from the right-click menu. Multiple rows can be selected before using the right-click menu.
 - A category can be removed or renamed using the **Remove** and **Rename** buttons from the Side Panel.
- From the cell-level view , the category over which the mouse hovers can be removed or renamed from the right-click menu.

18.6 Add Information to Plot








Add Information to Plot can be found at:

Tools | Single Cell Analysis  | **Utility Tools**  | **Add Information to Plot** 

The tool takes as input a plot of one of the following types:

- **Dimensionality Reduction Plot** , see chapter 16.
- **Phase Portrait Plot** , see section 10.4.
- **Spatial Transcriptomics Plot** , see section 11.1.

The information to be added to the plot is set in the **Information** option. Multiple elements can be chosen, at most one of the following types:

- Cell Clusters ;
- Cell Annotations ;
- Expression Matrix  or Expression Matrix with spliced and unspliced counts ;
- Peak Count Matrix ;
- Velocity Matrix ;
- Only when the input is a Spatial Transcriptomics Plot: Dimensionality Reduction Plot .

Add Information to Plot produces a copy of the input plot, with associations to the selected elements, such that they are available in the Side Panel. Note that dimensionality reduction and phase portrait plots are already associated with the matrices that were used to produce the plots.



The tool is intended for use in workflows, so that the association of the various elements can be automated. Outside of a workflow setting, it is usually easier to achieve the same result by dragging the elements from the Navigation Area into the Side Panel, see chapter 17 for details.

18.7 Update Single Cell Sample Name

There are a number of situations where the sample name needs to be updated:

- When analyzing
 - scATAC-Seq or scV(D)J-Seq with matched scRNA-Seq, data originating from the same sample must be annotated with the same sample name.
 - spatial transcriptomics data, the Spatial Transcriptomics Plot and corresponding Expression Matrix/Dimensionality Reduction Plot must have the same sample name.











Ideally, this should be done as a first step in the analysis pipeline, when running **Annotate Single Cell Reads**, see section 6.1, or when importing the data (see chapter 4). If this has not been done, the sample names must be updated subsequently.

- Multiple samples may be multiplexed with a hashtag identifying the sample. The hashtags are first translated to Cell Annotations () (see section 15.3) and then the samples are demultiplexed using these annotations.
- When analyzing multiple samples produced using a Parse Biosciences kit. The matrix created by **Demultiplex Parse Bio Samples**, see section 7.3, can be used to update the sample for other elements, such as the Cell Annotations () containing QC metrics generated by **QC for Single Cell**, see section 7.2.

Update Single Cell Sample Name can be used in such situations. It can be found at:

Tools | Single Cell Analysis () | **Utility Tools** () | **Update Single Cell Sample Name** ()

The tool takes as input an element of one of the types below:

- Sequence List () that has been annotated with Annotate Single Cell Reads;
- Expression Matrix () / ();
- Peak Count Matrix ();
- Velocity Matrix ();
- Cell Clusters ();
- Cell Annotations ();
- Cell Clonotypes () / ();
- Spatial Transcriptomics Plot ().

The output is a copy of the input where the sample name is updated.




The sample name can be provided in one of three ways (figure 18.2).

Specify sample name

When the input contains only one sample, the sample name can be set as specified in the **Sample name** option. A combination of placeholders and text can be used to set the sample name. Hover the mouse cursor over the field to see a tooltip with several examples. Simultaneously pressing Shift and F1 displays all available placeholders. Different placeholders are available when the tool is run from the Tools menu or as part of a workflow.

From element

The sample name can be updated to that found in a second element provided in the **Element** option. The element can be of one of the types below:

- Sequence List () that has been annotated with Annotate Single Cell Reads;
- Expression Matrix () / ();

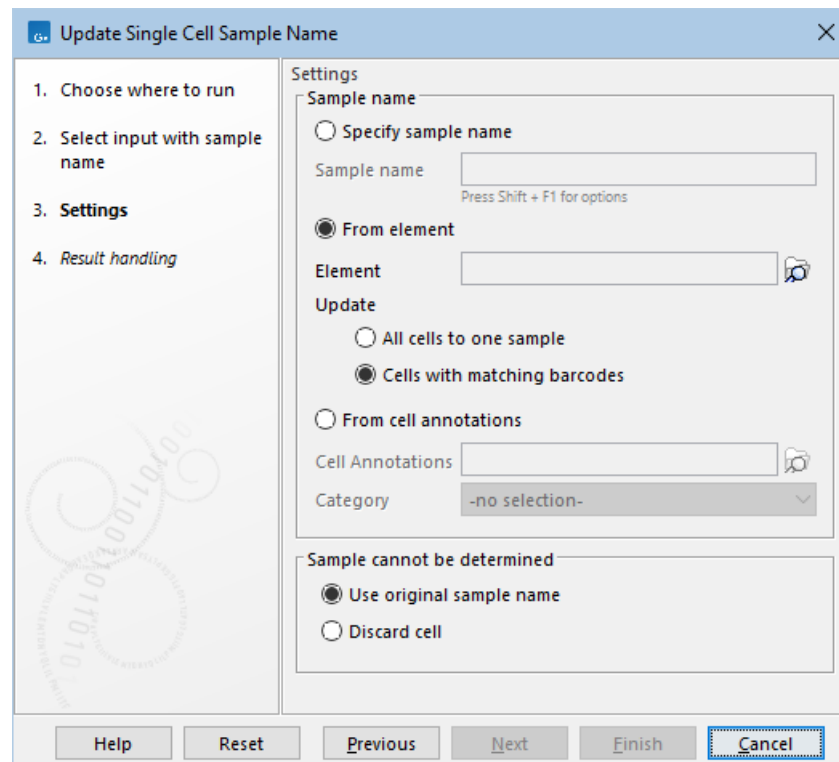



Figure 18.2: Options for updating the sample name.

- Peak Count Matrix (📊);
- Velocity Matrix (📊);
- Cell Clusters (📊);
- Cell Annotations (📊);
- Cell Clonotypes (📊) / (📊);
- Dot Plot (📊);
- Violin Plot (📊);
- Dimensionality Reduction Plot (📊);
- Spatial Transcriptomics Plot (📊).


The following options for how to **Update** the sample are available:

- **All cells to one sample.** The sample name for all input cells is set to the sample name found in the element. This requires that both the input and the element contain one sample each.
- **Cells with matching barcodes.** The sample name for the input cells is set to the sample name of the matching barcode found in the element. Only barcodes that are unique in both the input and element can be matched. Input cells with non-unique barcodes are handled according to **Sample cannot be determined**, see below.


As Sequence Lists () can only represent one sample, their sample name cannot be set using this option.

Using an input with one sample and an element containing multiple samples makes it possible to demultiplex the input into multiple samples.

From cell annotations

The sample name can be set from a category from a Cell Annotations () element, making it possible to demultiplex to multiple samples. For this, the input cells are matched with the cells from the Cell Annotations as follows:

- Cells with the same barcode that are unique in both the input and Cell Annotations are matched. Input cells with non-unique barcodes are handled according to **Sample cannot be determined**, see below.
- Cells with the same barcode and sample in both the input and Cell Annotations are matched. Input cells not found in the Cell Annotations are handled according to **Sample cannot be determined**, see below.

As Sequence Lists () can only represent one sample, their sample name cannot be set using Cell Annotations.

Sample cannot be determined

The following options are available for cells where the sample name cannot be determined:

- **Use original sample name.** The sample name in the output will be the same as in the input.
- **Discard cell.** The cells will not be part of the output.

Part IV

Template Workflows

Chapter 19

Single cell template workflows from reads








Contents

19.1 Expression Analysis from Reads	241
19.1.1 Configuring the batch units for Expression Analysis from Reads	243
19.1.2 Output from Expression Analysis from Reads	245
19.2 Chromatin Accessibility Analysis from Reads	246
19.2.1 Configuring the batch units for Chromatin Accessibility Analysis from Reads	247
19.2.2 Output from Chromatin Accessibility Analysis from Reads	247
19.3 Chromatin Accessibility and Expression Analysis from Reads	250
19.3.1 Configuring the batch units for Chromatin Accessibility and Expression Analysis from Reads	251
19.3.2 Output from Chromatin Accessibility and Expression Analysis from Reads	252
19.3.3 Importing reads	252
19.4 Immune Repertoire Analysis from Reads (10xV(D)J)	257
19.4.1 Output from Immune Repertoire Analysis from Reads (10xV(D)J)	258
19.5 Immune Repertoire and Expression Analysis from Reads (10xV(D)J)	259
19.5.1 Configuring the batch units for Immune Repertoire and Expression Analysis from Reads (10xV(D)J)	260
19.5.2 Output from Immune Repertoire and Expression Analysis from Reads (10xV(D)J)	261

19.1 Expression Analysis from Reads

The workflow Expression Analysis from Reads takes Reads as input and starts by annotating them with cell barcode and UMI, followed by trimming and mapping to create one or more Expression Matrix (📊) / (📊). Then it performs quality control, normalization, clustering, and cell type prediction. The workflow uses iterate functionality and allows for a combined analysis of multiple samples to produce:

- a single, multi-sample, normalized **Expression Matrix** (📊) / (📊);

- a **Dimensionality Reduction Plot** () associated with the automated clusters, predicted cell types and additional cell annotations;
- a **Heat Map** () , a **Dot Plot** () , and a **Violin Plot** () with the predicted cell types as cell groups;
- a **Cell Abundance Heat Map** () with the automated clusters and predicted cell types as cell groups.
- If velocity analysis is run:
 - a **Phase Portrait Plot** () with per gene information on the velocity dynamics;
 - a **Velocity Genes Scores** () element allowing identification of velocity genes driving the dynamics.

The workflow can be found here:

Template Workflows | Single Cell Workflows () | **From Reads** () | **Expression Analysis from Reads** ()


If you are connected to a *CLC Server* via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a *CLC Server* when possible.

Using a Fork element, the workflow offers the option to run velocity analysis. To enable this, set **Velocity Analysis** to **Run** in the **Specify Workflow Path** wizard step. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomics/current/index.php?manual=Fork.html> for details.

You can choose either one or more Sequence lists or **Select files for on-the-fly import** and select FASTQ files for importing.

The workflow offers a number of options. Note that not all parameters can be configured. Open parameters indicate places where customization may be necessary for different samples, but default settings are suitable in most cases.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html. When creating custom reference data sets, the chosen gene track needs to match the gene annotations used for training the provided **Cell Type Classifier** () (see section 8.3.1).

The workflow allows the analysis of multiple samples and you can specify metadata during workflow execution for configuring which inputs belong to which sample. When there is only one library per sample, metadata is not necessary and "Use organization of input data" can be used, but metadata can still be useful, as it is converted to cell annotations and can be used for coloring the cells in the Dimensionality Reduction Plot. For more details on configuring workflow execution with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html. Make sure to inspect the batch overview to check that the analysis will be performed correctly.

Examples for how to use metadata for workflow execution can be found in section [19.1.1](#).

It is important to select the proper read structure for annotating the reads with cell barcode and UMI. If the data has not been prepared using one of the predefined protocols, a custom read structure can be specified as detailed in section [6.1](#), where a list of many different single cell protocols is also linked. However, this requires editing the workflow, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Creating_editing_workflows.html for details.

Spike-in controls can be provided, if used during sample preparation. To learn how to import spike-in control files, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Import_RNA_spike_in_controls.html.

The strand specificity and expected coverage bias must be specified. Strand specific "Forward" is most common, though 5' sequencing often requires strand specific "Reverse". For 5' sequencing, we recommend setting coverage bias to "Targeted". If an unsuitable strand specificity or coverage bias is chosen, warnings may be shown in the output RNA-Seq report (for details see section [7.1.1](#).)

An option to count intronic reads towards gene expression is also present. This is recommended when many transcripts are expected to be unprocessed, as is the case for single nucleus RNA sequencing.

For quality control a number of options exist. The option to remove empty droplets is not suitable for protocols that do not use droplets, and removing barcodes with low number of reads or expressed features might be more appropriate. Quality Control (QC) uses the number of reads mapped to the mitochondria, and for this the name of the mitochondria chromosome needs to be provided. The default value is often the correct name. After quality control, the matrices are collected and normalized jointly. Note that batch correction is not performed. Read more about QC and normalization in chapter [7](#).

For clustering and creation of the Dimensionality Reduction Plot plot, it is possible to restrict analysis to highly variable genes. The data is then projected to a lower dimensional space using PCA. You can read about this feature in section [14.1](#).

The high confidence predicted cell types ("Cell type (high confidence)") are used to group the cells in the expression plots (Heat Map and Dot Plot) and Cell Abundance Heat Map, as well as for scoring the velocity genes. The Cell Abundance Heat Map additionally groups the cells based on the automated clusters obtained with resolution 1.0 ("Leiden (resolution=1.0)"). Any of these groups can be changed to:

- all predicted cell types ("Cell type (all)");
- automated clusters obtained with a different resolution x ("Leiden (resolution= x)"). All resolutions $0.1 \leq x \leq 1.5$ are produced, in steps of 0.1.

19.1.1 Configuring the batch units for Expression Analysis from Reads

When there is only library per sample, metadata is not necessary for workflow execution. Let us consider the FASTQ files shown in figure [19.1](#).

The files can be automatically imported during workflow execution by choosing "Select files for on-the-fly import", selecting the Illumina importer and enabling "Paired reads" and "Join reads

☐ S1_L001_R1.fastq ☐ S2_L001_R1.fastq ☐ S3_L001_R1.fastq
☐ S1_L001_R2.fastq ☐ S2_L001_R2.fastq ☐ S3_L001_R2.fastq
☐ S1_L002_R1.fastq ☐ S2_L002_R1.fastq
☐ S1_L002_R2.fastq ☐ S2_L002_R2.fastq

Figure 19.1: Example of ten FASTQ files for paired reads, originating from multiple lanes and three libraries.

from different lanes". Selecting "Use organization of input data" when defining the batch units will lead to the input files being grouped in three libraries, as shown in figure 19.2. Note that if any of the samples has more than 1 billion paired reads, the metadata approach described below should be used instead.

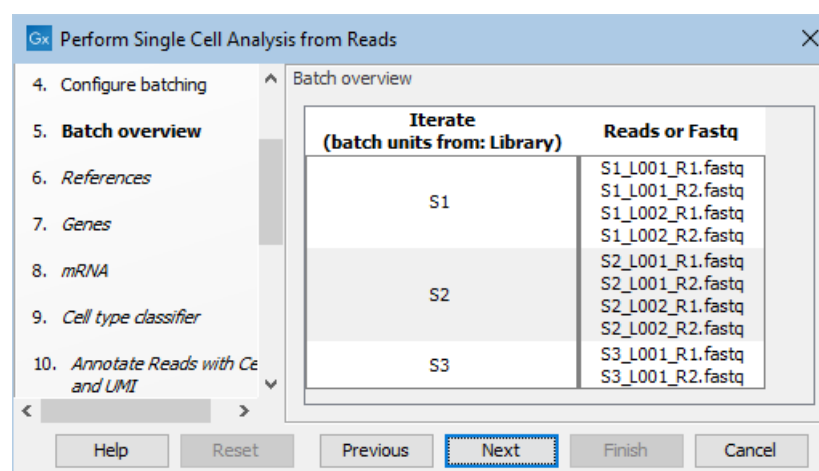


Figure 19.2: Batch overview when importing the FASTQ files and choosing "Use organization of input data".

Now let us consider the metadata shown in figure 19.3.

File	Library	Lane	Library and lane	Read	Sex	Time point
S1_L001_R1	S1	L001	S1_L001	R1	Male	T1
S1_L001_R2	S1	L001	S1_L001	R2	Male	T1
S1_L002_R1	S1	L002	S1_L002	R1	Male	T1
S1_L002_R2	S1	L002	S1_L002	R2	Male	T1
S2_L001_R1	S2	L001	S2_L001	R1	Female	T1
S2_L001_R2	S2	L001	S2_L001	R2	Female	T1
S2_L002_R1	S2	L002	S2_L002	R1	Female	T1
S2_L002_R2	S2	L002	S2_L002	R2	Female	T1
S3_L001_R1	S3	L001	S3_L001	R1	Female	T2
S3_L001_R2	S3	L001	S3_L001	R2	Female	T2

Figure 19.3: Example of metadata for the files from figure 19.1.

Metadata can be imported directly from an Excel or txt file during workflow execution and for this example, the "Library" metadata column should be used for defining the batch units (see figure 19.4).

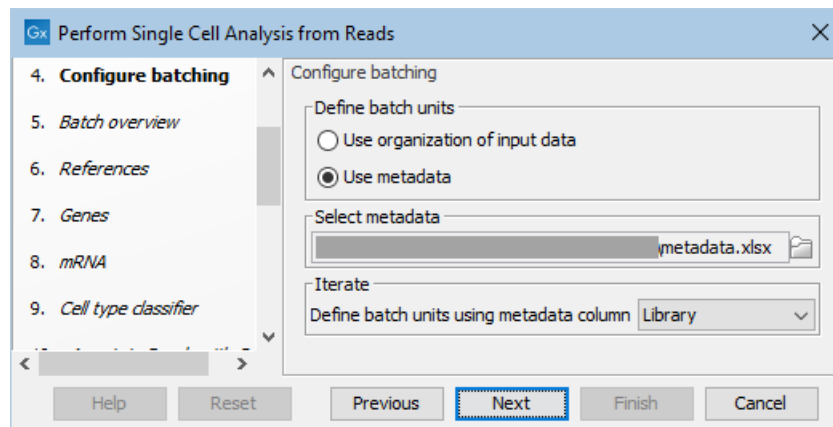


Figure 19.4: Configuring the workflow execution using metadata.

The workflow will automatically associate the input files with the correct rows in the metadata based on the first column and a batch overview similar to that in figure 19.2 will be shown. The additional metadata columns will be converted to cell annotations.

Note that in this workflow it is not possible to freely choose the batch units. Instead, each batch must correspond to one sample. The reason for the restriction is that each read is linked to a cell by the cell barcode. Batching by sample is required in order to inform the workflow that if the same cell barcode is present in multiple files, it is because it is the same cell.

Failing to batch by sample will likely lead to misleading results. For example, in figure 19.4 it would be necessary to batch by library. If we batched by "Time point", then two cells with the same barcode at time point T1 would be treated as being identical, even if one came from sample S1 and the other from sample S2. If, on the other hand, we batched by "Library and Lane", then a cell from sample S1 that was sequenced on both lanes would be split up into two cells - one for each lane.

The workflow combines all inputs to produce just one matrix. All metadata, including "Sex" and "Time point" in the provided example, will be available in the output cell annotations.

The FASTQ and metadata files can also be imported manually and used for the workflow execution.

19.1.2 Output from Expression Analysis from Reads

The workflow creates several output elements stored in a specific folder structure as indicated in figure 19.5. The reports are valuable for assessing whether the appropriate parameters have been used.

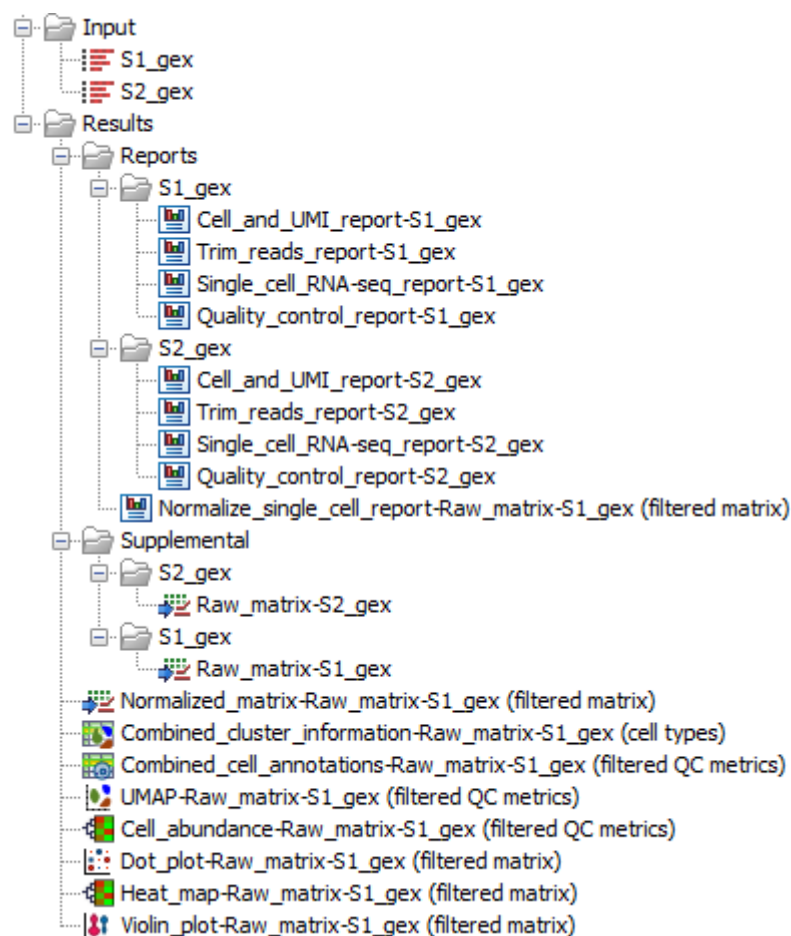


Figure 19.5: The Results folder contains the output elements produced by the Expression Analysis from Reads workflow when running using the sequence lists in the Input folder. The Reports and Supplemental folders contain a subfolder for each of the batch units defined during execution. Here, the "Use organization of input data" was used. Root elements originate from the combined analysis of all inputs.

19.2 Chromatin Accessibility Analysis from Reads

The workflow Chromatin Accessibility Analysis from Reads takes Reads as input and starts by annotating them with cell barcode and UMI, followed by trimming and mapping to create a **Peak Count Matrix** (📊). Then it performs clustering. The workflow uses iterate functionality and allows for a combined analysis of multiple samples to produce:

- a single multi-sample **Peak Count Matrix** (📊);
- a **Dimensionality Reduction Plot** (📊) associated with the automated clusters.

The workflow can be found here:

Template Workflows | Single Cell Workflows (🔍) | **From Reads** (📁) | **Chromatin Accessibility Analysis from Reads** (📊)

If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

You can choose either one or more Sequence lists or **Select files for on-the-fly import** and select FASTQ files for importing.

The workflow offers a number of options. Note that not all parameters can be configured. Open parameters indicate places where customization may be necessary for different samples, but default settings are suitable in most cases.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html.

The workflow allows the analysis of multiple samples and you can specify metadata during workflow execution for configuring which inputs belong to which sample. When there is only one library per sample, metadata is not necessary and "Use organization of input data" can be used. For more details on configuring workflow execution with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html. Make sure to inspect the batch overview to check that the analysis will be performed correctly.

19.2.1 Configuring the batch units for Chromatin Accessibility Analysis from Reads

Consider the FASTQ files shown in figure 19.6.













 S1_L001_R1.fastq	 S1_L001_R2.fastq	 S1_L001_R3.fastq
 S1_L002_R1.fastq	 S1_L002_R2.fastq	 S1_L002_R3.fastq
 S2_L001_R1.fastq	 S2_L001_R2.fastq	 S2_L001_R3.fastq
 S2_L002_R1.fastq	 S2_L002_R2.fastq	 S2_L002_R3.fastq

Figure 19.6: Example of twelve FASTQ files for paired reads split in three files each. They originate from two libraries with two lanes each.

The files can be automatically imported during workflow execution by choosing "Select files for on-the-fly import", selecting the Illumina importer and enabling "Paired reads" and "Join reads from different lanes". Note that for 10x ATAC reads "Use custom reads options" must be checked and "Custom reads options" must be set to "R1,R2 R3", as shown in figure 19.7. Selecting "Use organization of input data" when defining the batch units will lead to the input files being grouped in two libraries, as shown in figure 19.8.

The FASTQ files can also be imported manually and used for the workflow execution.

19.2.2 Output from Chromatin Accessibility Analysis from Reads

The workflow creates several output elements stored in a specific folder structure as indicated in figure 19.9.

Chromatin Accessibility Analysis from Reads

1. Choose where to run
- 2. Select Reads**
3. Select reference data set
4. Configure batching
5. Annotate Single Cell Reads
6. Single Cell ATAC-Seq Analysis
7. Result handling
8. Save location for new elements

Select sequences

☐ Select from Navigation Area

☒ Select files for on-the-fly import: **Illumina**

Select files: S2_L002_R1.fastq, S2_L002_R2.fastq, **Browse**

Discard read names: ☐

Discard quality scores: ☐

Paired reads: ☒

Read orientation: **Forward Reverse**

Minimum distance: 1

Maximum distance: 1,000

Remove failed reads: ☒

Quality scores: **NCBI/Sanger or Illumina Pipeline 1.8 and later**

MiSeq de-multiplexing: ☐

Trim reads: ☐

Join reads from different lanes: ☒

Custom read structure: ☒

Structure definition: R1,R2 R3

☐ Batch

Help **Reset** **Previous** **Next** **Finish** **Cancel**

Figure 19.7: Example of importing on-the-fly twelve FASTQ files for paired reads, originating from multiple lanes and two libraries.

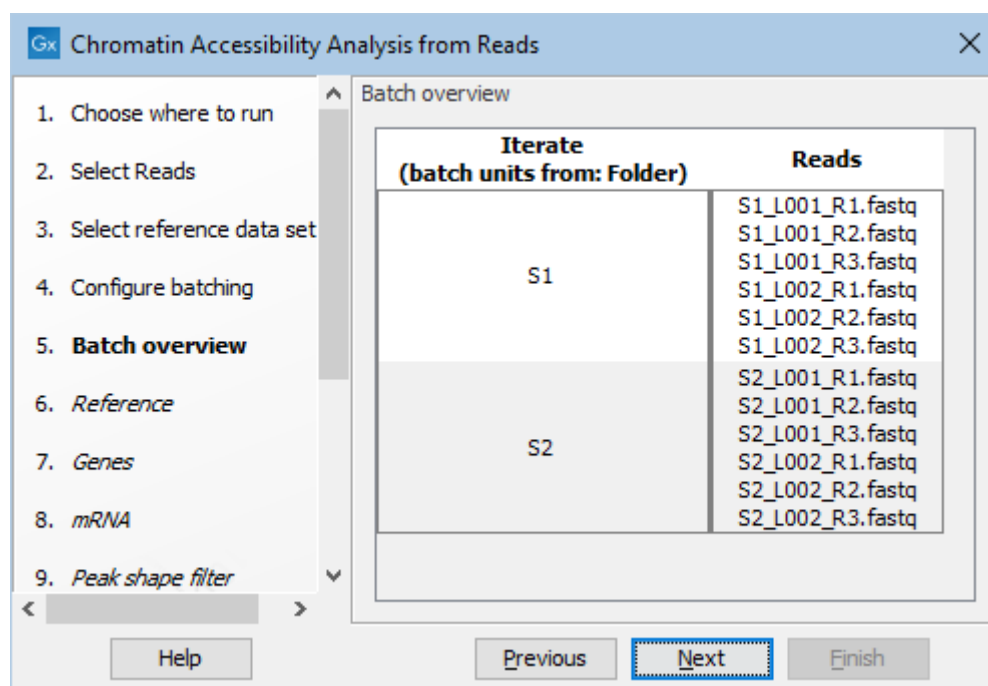


Figure 19.8: Batch overview when importing the FASTQ files and choosing "Use organization of input data".

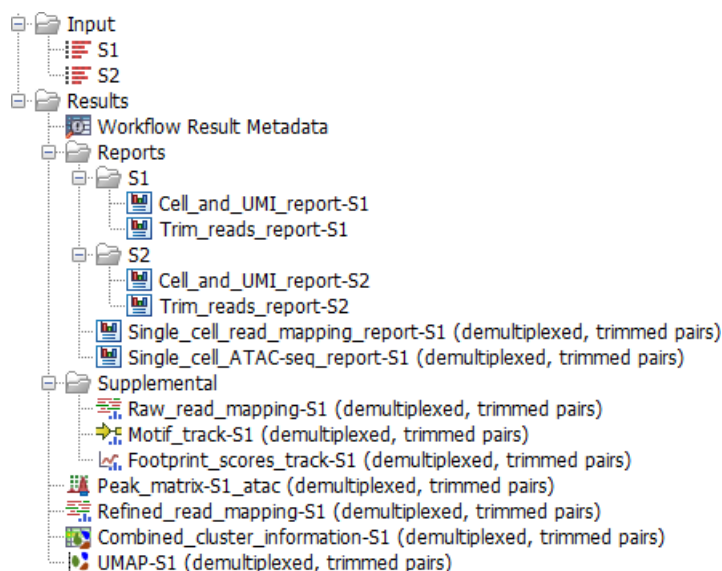



Figure 19.9: The Results folder contains the output elements produced by the Chromatin Accessibility Analysis from Reads workflow when running using the sequence lists in the Input folder. The Reports folder contains a subfolder for each of the batch units defined during execution and combined reports for all samples. The Supplemental folder contains elements that can be viewed in a track list. The read mapping can be further analyzed by sub-grouping with Split Read Mapping by Cell (section 12.2).










19.3 Chromatin Accessibility and Expression Analysis from Reads

The workflow Chromatin Accessibility and Expression Analysis from Reads takes 10x Multiome ATAC and gene expression (GEX) reads as input and starts by annotating them with cell barcode and UMI, followed by trimming.

During the annotation the barcodes from the ATAC reads are translated to barcodes that match the cell barcodes of GEX reads. The ATAC reads are then mapped and, in case of multiple samples, combined into one before producing one Peak Count Matrix ()

The GEX reads are analyzed as described in section 19.1. Clustering and dimensionality reduction are performed using both expression and peak matrices.

The workflow allows for a combined analysis of multiple samples to produce:

- a single **Peak Count Matrix** ()
- a single normalized **Expression Matrix** ()
- a **Dimensionality Reduction Plot** () associated with the automated clusters, predicted cell types and additional cell annotations;
- a **Heat Map** () , a **Dot Plot** () , and a **Violin Plot** () with the predicted cell types as cell groups;
- a **Cell Abundance Heat Map** () with the automated clusters and predicted cell types as cell groups.
- If velocity analysis is run:
 - a **Phase Portrait Plot** () with per gene information on the velocity dynamics;
 - a **Velocity Genes Scores** () element allowing identification of velocity genes driving the dynamics.

The workflow can be found here:

Template Workflows | Single Cell Workflows () | **From Reads** () | **Chromatin Accessibility and Expression Analysis from Reads** ()

If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

Using a Fork element, the workflow offers the option to run velocity analysis. To enable this, set **Velocity Analysis** to **Run** in the **Specify Workflow Path** wizard step. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomics/current/index.php?manual=Fork.html> for details.

You can choose either one or more Sequence lists or **Select files for on-the-fly import** and select FASTQ files for importing.

The workflow offers a number of options. Note that not all parameters can be configured. Open parameters indicate places where customization may be necessary for different samples, but default settings are suitable in most cases.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html. When creating custom reference data sets, the chosen gene track needs to match the gene annotations used for training the provided **Cell Type Classifier** (🧠) (see section 8.3.1).

The workflow allows the analysis of multiple samples. Metadata must always be specified for configuring which inputs belong to which sample. In addition to group the input, metadata is converted to cell annotations and can be used for coloring the cells in the Dimensionality Reduction Plot.

For more details on configuring workflow execution with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html. Make sure to inspect the batch overview to check that the analysis will be performed correctly.

19.3.1 Configuring the batch units for Chromatin Accessibility and Expression Analysis from Reads

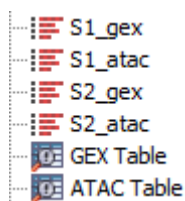


Figure 19.10: ATAC and GEX reads for two samples along with metadata

Assume as an example that the input consists of ATAC and GEX reads for two samples as shown in figure 19.10. The reads must be linked by metadata tables. That is, there must be one metadata table with all ATAC reads and one with all GEX reads and the two must have a common column linking them (e.g., Sample).

The batching can be configured as shown in figure 19.11. In this example:

- The batching is based on the column "Sample" in the metadata table for the ATAC reads. There will be one iteration per distinct value, which typically means per row.
- The ATAC and GEX reads are matched by the same "Sample" column. It must be present in both metadata tables. The matching column does not need to be the same as the batching column, but it typically is.

There will be two iterations, one for the ATAC and GEX reads for sample S1 and one for S2 as shown in figure 19.12.

It is also possible to use the same metadata table for ATAC and GEX reads. Then it must be selected twice in the **Configure batching** wizard step.

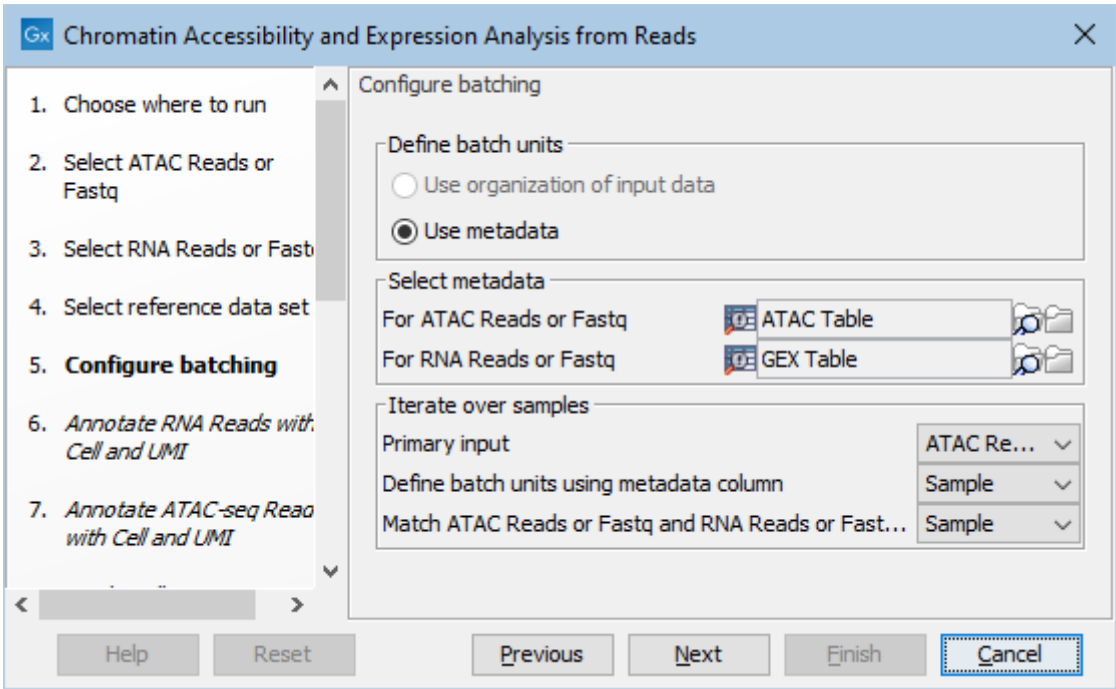


Figure 19.11: Configuring batching of coupled ATAC and GEX reads

Iterate over samples (batch units from: Sample)	Iterate over samples (matching on: Sample)	ATAC Reads or Fastq	RNA Reads or Fastq
S1	S1	S1_atac	S1_gex
S2	S2	S2_atac	S2_gex

Figure 19.12: Batch overview for coupled ATAC and GEX reads

19.3.2 Output from Chromatin Accessibility and Expression Analysis from Reads

The workflow creates several output elements stored in a specific folder structure as indicated in figure 19.13.

19.3.3 Importing reads

The ATAC or GEX reads or both can be imported with on-the-fly imports when running the workflow. This section will describe an example for the reads shown in figure 19.14.

The two sets of reads must be matched with metadata as shown in figure 19.15 and 19.16. Both tables have a common column "Sample" which will be used for matching. Note that the import of 10x ATAC reads must be configured with "Custom reads options" set to "R1,R2 R3" as shown in figure 19.17.

The two sets of reads will be grouped in two batches as shown in figure 19.18.

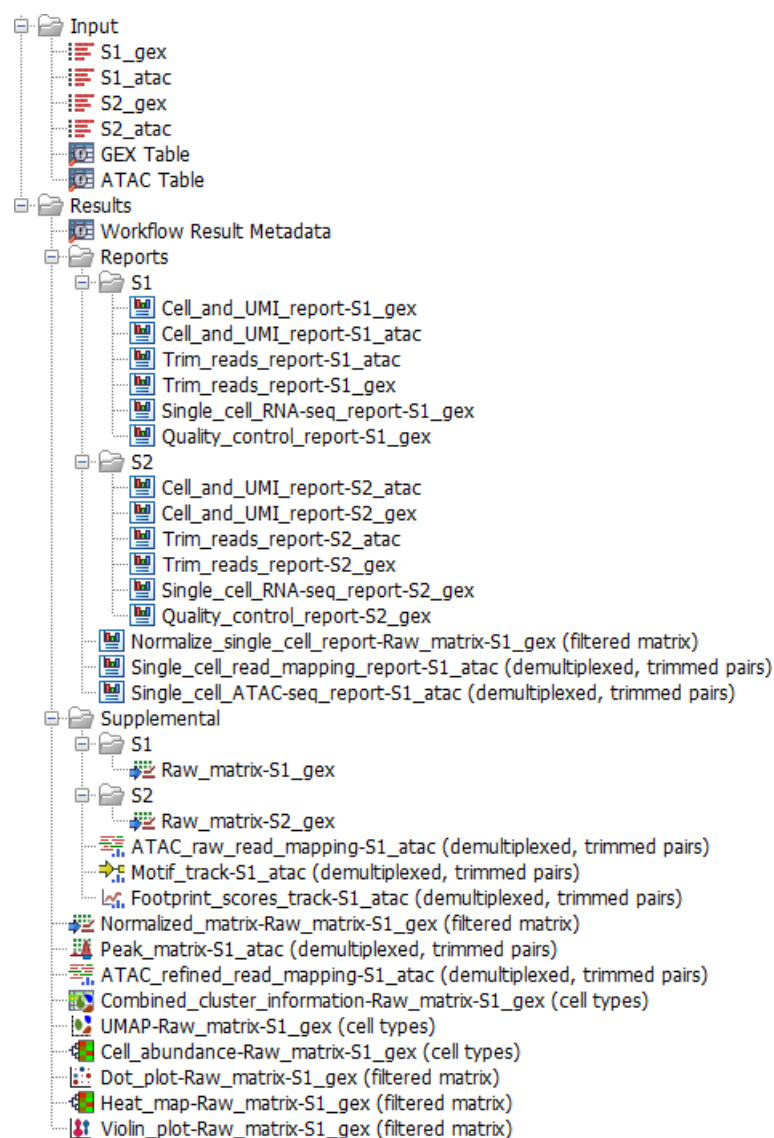


Figure 19.13: The Results folder contains the output elements produced by the Chromatin Accessibility and Expression Analysis from Reads workflow

S1_ATAC_L001_R1.fastq S1_GEX_L001_R1.fastq
 S1_ATAC_L001_R2.fastq S1_GEX_L001_R2.fastq
 S1_ATAC_L001_R3.fastq S1_GEX_L002_R1.fastq
 S1_ATAC_L002_R1.fastq S1_GEX_L002_R2.fastq
 S1_ATAC_L002_R2.fastq S2_GEX_L001_R1.fastq
 S1_ATAC_L002_R3.fastq S2_GEX_L001_R2.fastq
 S2_ATAC_L001_R1.fastq S2_GEX_L002_R1.fastq
 S2_ATAC_L001_R2.fastq S2_GEX_L002_R2.fastq
 S2_ATAC_L001_R3.fastq
 S2_ATAC_L002_R1.fastq
 S2_ATAC_L002_R2.fastq
 S2_ATAC_L002_R3.fastq

Figure 19.14: FASTQ files for two samples with two lanes for both ATAC and GEX reads

File	Sample	Read	Lane
S1_ATAC_L001_R1.fastq	S1	R1	L001
S1_ATAC_L001_R2.fastq	S1	R2	L001
S1_ATAC_L001_R3.fastq	S1	R3	L001
S1_ATAC_L002_R1.fastq	S1	R1	L002
S1_ATAC_L002_R2.fastq	S1	R2	L002
S1_ATAC_L002_R3.fastq	S1	R3	L002
S2_ATAC_L001_R1.fastq	S2	R1	L001
S2_ATAC_L001_R2.fastq	S2	R2	L001
S2_ATAC_L001_R3.fastq	S2	R3	L001
S2_ATAC_L002_R1.fastq	S2	R1	L002
S2_ATAC_L002_R2.fastq	S2	R2	L002
S2_ATAC_L002_R3.fastq	S2	R3	L002

Figure 19.15: Metadata for ATAC FASTQ files

File	Sample	Read	Lane
S1_GEX_L001_R1.fastq	S1	R1	L001
S1_GEX_L001_R2.fastq	S1	R2	L001
S1_GEX_L002_R1.fastq	S1	R1	L002
S1_GEX_L002_R2.fastq	S1	R2	L002
S2_GEX_L001_R1.fastq	S2	R1	L001
S2_GEX_L001_R2.fastq	S2	R2	L001
S2_GEX_L002_R1.fastq	S2	R1	L002
S2_GEX_L002_R2.fastq	S2	R2	L002

Figure 19.16: Metadata for GEX FASTQ files

Chromatin Accessibility and Expression Analysis from Reads

1. Choose where to run

2. **Select ATAC Reads or Fastq**

3. Select RNA Reads or Fastq

4. Select reference data set

5. Configure batching

6. Annotate scRNA-Seq Reads

7. Annotate scATAC-Seq Read

8. Single Cell RNA-Seq Analysis

9. QC for Single Cell

10. Single Cell ATAC-Seq Analysis

11. Enable Velocity Analysis

12. Single Cell Velocity Analysis

13. Create Heat Map for Cell Abundance

Select sequences

☐ Select from Navigation Area

☒ Select files for on-the-fly import: Illumina

Select files: 02_R1.fastq, S2_ATAC_L002_R2.fastq Browse

Discard read names ☐

Discard quality scores ☐

Paired reads ☒

Read orientation: Forward Reverse

Minimum distance: 1

Maximum distance: 1,000

Remove failed reads ☒

Quality scores: NCBI/Sanger or Illumina Pipeline 1.8 and later

MiSeq de-multiplexing ☐

Trim reads ☐

Join reads from different lanes ☒

Custom read structure ☒

Structure definition: R1,R2 R3

☐ Batch



Help Reset Previous **Next** Finish Cancel

Figure 19.17: Import settings for 10x ATAC FASTQ files




Iterate over samples (batch units from: Sample)	Iterate over samples (matching on: Sample)	ATAC Reads or Fastq	RNA Reads or Fastq
S1	S1	S1_ATAC_L001_R1.fastq S1_ATAC_L001_R2.fastq S1_ATAC_L001_R3.fastq S1_ATAC_L002_R1.fastq S1_ATAC_L002_R2.fastq S1_ATAC_L002_R3.fastq	S1_GEX_L001_R1.fastq S1_GEX_L001_R2.fastq S1_GEX_L002_R1.fastq S1_GEX_L002_R2.fastq
S2	S2	S2_ATAC_L001_R1.fastq S2_ATAC_L001_R2.fastq S2_ATAC_L001_R3.fastq S2_ATAC_L002_R1.fastq S2_ATAC_L002_R2.fastq S2_ATAC_L002_R3.fastq	S2_GEX_L001_R1.fastq S2_GEX_L001_R2.fastq S2_GEX_L002_R1.fastq S2_GEX_L002_R2.fastq

Figure 19.18: Batch overview for ATAC and GEX FASTQ files

19.4 Immune Repertoire Analysis from Reads (10xV(D)J)

The workflow Immune Repertoire Analysis from Reads (10xV(D)J) takes 10xV(D)J reads as input and starts by annotating them with cell barcode and UMI, followed by clonotype identification and filtering. The workflow uses iterate functionality and allows for a combined analysis of multiple samples to produce a single, multi-sample, filtered **TCR Cell Clonotypes**  or **BCR Cell Clonotypes**  element.

The workflow can be found here:

Template Workflows | **Single Cell Workflows**  | **From Reads**  | **Immune Repertoire Analysis from Reads (10xV(D)J)** 

If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

You can choose either one or more Sequence lists or **Select files for on-the-fly import** and select FASTQ files for importing.


The workflow is configured for 10x Chromium Single Cell V(D)J reads and only clonotype filtering is customizable. Adjustments can be made in a workflow copy, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Creating_editing_workflows.html.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html. Reference V, D, J and C gene segments for other species or for B cells can be imported using **Import Immune Reference Segments** (see section 4.1).

The workflow allows the analysis of multiple samples and you can specify metadata during workflow execution for configuring which inputs belong to which sample. When there is only one library per sample, metadata is not necessary and "Use organization of input data" can be used. For more details on configuring workflow execution with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html. Make sure to inspect the batch overview to check that the analysis will be performed correctly.

Examples for how to use metadata for workflow execution can be found in section 19.1.1.

The **Filter Cell Clonotypes**  tool in the workflow is configured with most parameters locked, see 19.19. There are two open options. "Chains to retain" allows selecting the chains that are expected to be found, and removes noise from the results. "Multiple clonotypes" allows different handling of barcodes with more than one clonotype. The default option **Retain primary** retains only the clonotype with the highest number of reads. See section 13.2 for details.

Configure Filter Cell Clonotypes

1. Filtering (Filter Cell Clonotypes)

Filtering

Known cells

Barcodes to retain

Productive

Productive status to retain (Nothing selected)

Chains

Chains to retain (Nothing selected)

Combined chains to retain (Nothing selected)

Segments

Segment types to retain (Nothing selected)

Barcodes with multiple clonotypes

Multiple clonotypes Retain primary

Help Reset Previous Next Finish Cancel

Figure 19.19: Workflow settings of the Filter Cell Clonotypes tool.

19.4.1 Output from Immune Repertoire Analysis from Reads (10xV(D)J)

The workflow creates several output elements stored in a specific folder structure as indicated in figure 19.20. The reports are valuable for assessing whether the appropriate parameters have been used.

Read more about the clonotype reports in section 13.1.1 and **TCR Cell Clonotypes** (🔍) or **BCR Cell Clonotypes** (🔍) in section 13.6.

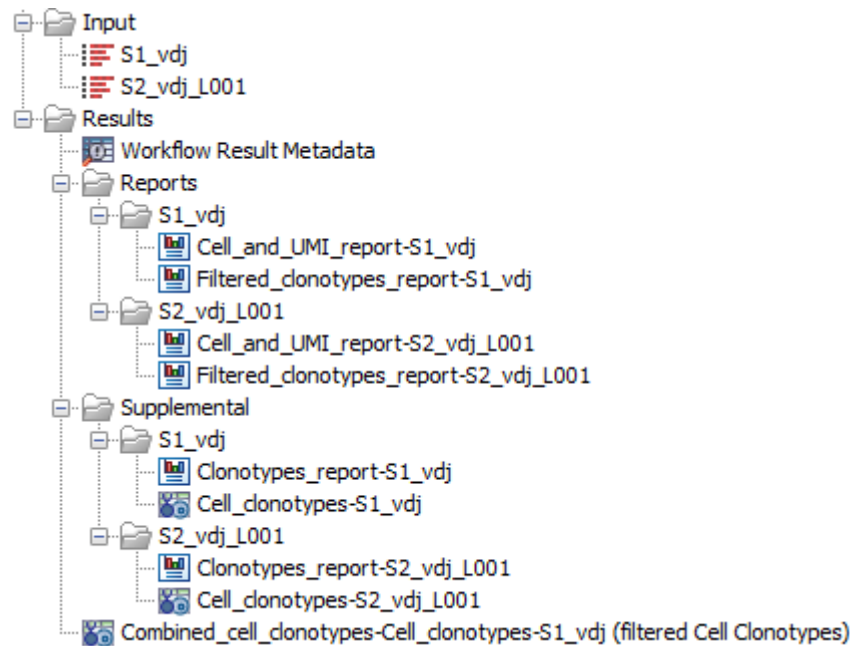


Figure 19.20: The Results folder contains the output elements produced by the Immune Repertoire Analysis from Reads (10xV(D)J) workflow when running using the sequence lists in the Input folder. The Reports and Supplemental folders contain a subfolder for each of the batch units defined during execution. Here, the "Use organization of input data" was used. Root elements originate from the combined analysis of all inputs.



19.5 Immune Repertoire and Expression Analysis from Reads (10xV(D)J)

The workflow Immune Repertoire and Expression Analysis from Reads (10xV(D)J) takes reads as input and jointly analyzes scRNA-Seq and scV(D)J-Seq data originating from the same sample. The reads are first annotated with cell barcode and UMI, after which they are sent on two different paths, one for each type of data. The workflow splits the reads according to sample and data type, as given through metadata.

The scRNA-Seq and scV(D)J-Seq paths follow the same analysis described in section 19.1 and section 19.4, respectively.

The workflow uses the iterate functionality and allows for a combined analysis of multiple samples to produce:

- a single, multi-sample, normalized **Expression Matrix** (📊) / (📊);
- a single, multi-sample, filtered **TCR Cell Clonotypes** (📊) or **BCR Cell Clonotypes** (📊) element;
- a **Dimensionality Reduction Plot** (📊) associated with the automated clusters, predicted cell types, identified clonotypes and additional cell annotations;
- a **Heat Map** (📊), a **Dot Plot** (📊), and a **Violin Plot** (📊) with the predicted cell types as cell groups;
- a **Cell Abundance Heat Map** (📊) with the automated clusters and predicted cell types as cell groups.

- If velocity analysis is run:
 - a **Phase Portrait Plot** () with per gene information on the velocity dynamics;
 - a **Velocity Genes Scores** () element allowing identification of velocity genes driving the dynamics.

The workflow can be found here:

Template Workflows | Single Cell Workflows () | **From Reads** () | **Immune Repertoire and Expression Analysis from Reads (10xV(D)J)** ()


If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

Using a Fork element, the workflow offers the option to run velocity analysis. To enable this, set **Velocity Analysis** to **Run** in the **Specify Workflow Path** wizard step. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomics/current/index.php?manual=Fork.html> for details.

You can choose either one or more Sequence lists or **Select files for on-the-fly import** and select FASTQ files for importing.

The workflow is configured for 10x Chromium Single Cell V(D)J data. For the scRNA-Seq path, a number of options are customizable, see section 19.1. For the scV(D)J-Seq path, only clonotype filtering is customizable, see section 19.4. Adjustments can be made in a workflow copy, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Creating_editing_workflows.html.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html. When creating custom reference data sets, the chosen gene track needs to match the gene annotations used for training the provided **Cell Type Classifier** () (see section 8.3.1). Reference V, D, J and C gene segments for other species or for B cells can be imported using **Import Immune Reference Segments** (see section 4.1).

The workflow allows the analysis of multiple samples and you must specify metadata during execution for configuring which reads belong to which sample and data type, see section 19.5.1.

19.5.1 Configuring the batch units for Immune Repertoire and Expression Analysis from Reads (10xV(D)J)

The Immune Repertoire and Expression Analysis from Reads (10xV(D)J) workflow requires metadata containing information about the sample of origin and data type: scRNA-Seq or scV(D)J-Seq (figure 19.21). This can be created a priori, or it can be imported directly from an Excel or txt file during workflow execution. For more information about Metadata Tables, see <https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Metadata.html>.

The screenshot shows a software window titled 'workflow_metadata'. It contains two main tables and a settings panel on the right.

Top Table (Metadata Table):

ID	Sample	Type
S1_gex	S1	Gene Expression
S1_vdj	S1	V(D)J
S2_gex	S2	Gene Expression
S2_vdj_L001	S2	V(D)J
S2_vdj_L002	S2	V(D)J

Bottom Table (Associated Elements):

ID	Role	Type	Name	Path
S1_gex	Sample data	CLC	S1_gex	CLC_Data/Input
S1_vdj	Sample data	CLC	S1_vdj	CLC_Data/Input
S2_gex	Sample data	CLC	S2_gex	CLC_Data/Input
S2_vdj_L001	Sample data	CLC	S2_vdj_L001	CLC_Data/Input
S2_vdj_L002	Sample data	CLC	S2_vdj_L002	CLC_Data/Input

Metadata Table Settings Panel:

- Column width: Automatic
- Show column:
 - ☒ ID
 - ☒ Sample
 - ☒ Type
- Buttons: Select All, Deselect All

Figure 19.21: A Metadata Table associating the reads to samples and data type. The sample of origin is given in the "Sample" column, while the data type is given in the "Type" column, with values "Gene Expression" for scRNA-Seq data, and "V(D)J" for scV(D)J-Seq data. The associated elements are opened on the bottom.

During the workflow execution, it must be configured which metadata columns define the sample: "Iterate over Sample" and data type: "Iterate over RNA and V(D)J", and which values from the data type column correspond to scRNA-Seq and scV(D)J-Seq data, respectively. For more details on configuring workflow execution with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html. Make sure to inspect the batch overview to check that the analysis will be performed correctly.

Figures 19.22 and 19.23 show how the workflow should be configured using the input reads and metadata from figure 19.21, and the resulting batch overview is given in figure 19.24.

An additional example for importing the reads and metadata during workflow execution can be found in section 19.1.1.

19.5.2 Output from Immune Repertoire and Expression Analysis from Reads (10xV(D)J)

The workflow creates several output elements stored in a specific folder structure as indicated in figure 19.25. The reports are valuable for assessing whether the appropriate parameters have been used.

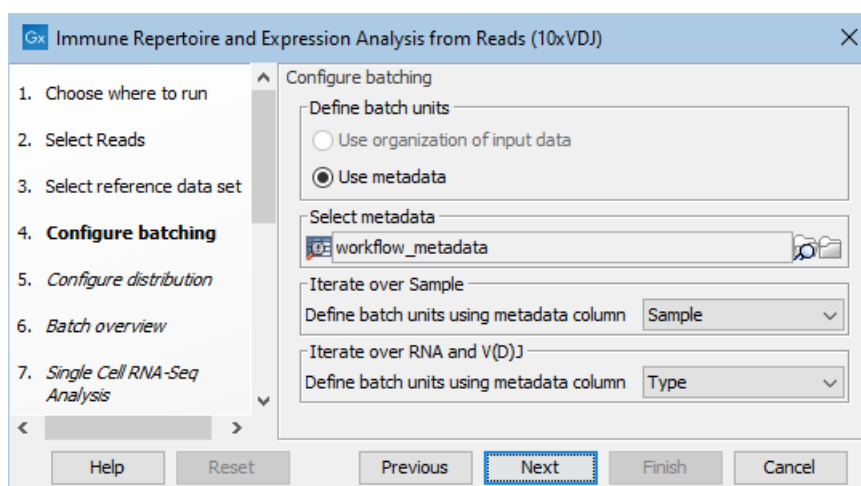


Figure 19.22: Configuring sample and data type during workflow execution using the metadata from figure 19.21.

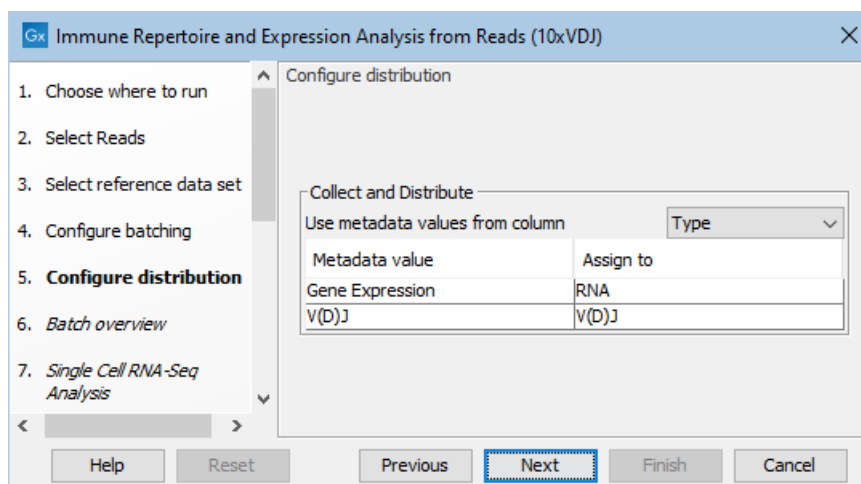


Figure 19.23: Configuring scRNA-Seq and scV(D)J-Seq reads distribution during workflow execution using the metadata from figure 19.21.

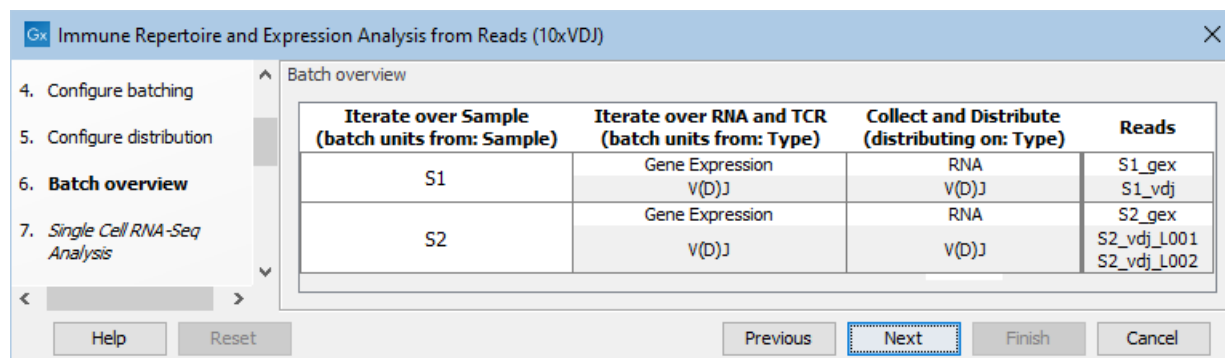


Figure 19.24: Batch overview for the configuration shown in figures 19.22 and 19.23.

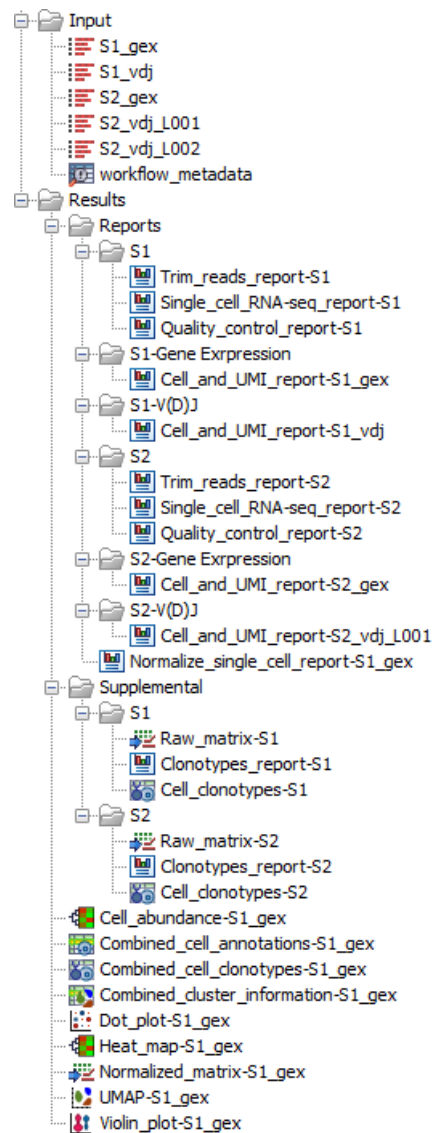


Figure 19.25: The Results folder contains the output elements produced by the Immune Repertoire and Expression Analysis from Reads (10xV(D)J) workflow when running using the sequence lists in the Input folder. The Reports and Supplemental folders contain a subfolder for each of the batch units defined during execution. Here, the metadata from figure 19.21 was used. Root elements originate from the combined analysis of all inputs.



Chapter 20









Single cell template workflows from imported data


Contents

20.1 Expression Analysis from Matrix	264
20.1.1 Output from Expression Analysis from Matrix	266
20.2 Chromatin Accessibility and Expression Analysis from Matrix	266
20.2.1 Output of Chromatin Accessibility and Expression Analysis from Matrix	268
20.3 Immune Repertoire and Expression Analysis from Clonotypes and Matrix	268
20.3.1 Output from Immune Repertoire and Expression Analysis from Clonotypes and Matrix	270

20.1 Expression Analysis from Matrix

The workflow Expression Analysis from Matrix takes one or more **Expression Matrix** ( / ) as input and performs quality control, normalization, clustering, and cell type prediction. The workflow uses iterate functionality and allows for a combined analysis of multiple samples to produce:

- a single, multi-sample, normalized **Expression Matrix** ( / );
- a **Dimensionality Reduction Plot** () associated with the automated clusters, predicted cell types and additional cell annotations;
- a **Heat Map** () , a **Dot Plot** () , and a **Violin Plot** () with the predicted cell types as cell groups;
- a **Cell Abundance Heat Map** () with the automated clusters and predicted cell types as cell groups.
- If velocity analysis is run:
 - a **Phase Portrait Plot** () with per gene information on the velocity dynamics;



- a **Velocity Genes Scores**  element allowing identification of velocity genes driving the dynamics.

The workflow can be found here:

Template Workflows | Single Cell Workflows  | **From Imported Data**  | **Expression Analysis from Matrix** 


If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

Using a Fork element, the workflow offers the option to run velocity analysis. To enable this, set **Velocity Analysis** to **Run** in the **Specify Workflow Path** wizard step. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomics/current/index.php?manual=Fork.html> for details.

Choose either one or more **Expression Matrix**  /  or **Select files for on-the-fly import** and select the format that is compatible with the selected inputs. Read more about import options in section 4.9.

The workflow offers a number of options. Note that not all parameters can be configured. Open parameters indicate places where customization may be necessary for different samples, but default settings are suitable in most cases.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html. When creating custom reference data sets, the chosen gene track needs to match the gene annotations used for training the provided **Cell Type Classifier**  (see section 8.3.1).

The workflow allows the analysis of multiple samples and you can specify metadata during workflow execution. This is converted to cell annotations and can be used for coloring the cells in the Dimensionality Reduction Plot. However, the workflow expects each sample to be present in just one Expression Matrix, and attempting to define batch units containing more than one Expression Matrix will lead to a failure during execution.

For more details on configuring workflow execution with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html. Make sure to inspect the batch overview to check that the analysis will be performed correctly.

For quality control a number of options exist. The option to remove empty droplets is not suitable for protocols that do not use droplets, and removing barcodes with low number of reads or expressed features might be more appropriate. Quality Control (QC) uses the number of reads mapped to the mitochondria, and for this the name of the mitochondria chromosome needs to be provided. The default value is often the correct name. After quality control, the matrices are collected and normalized jointly. Note that batch correction is not performed. Read more about QC and normalization in chapter 7.

For clustering and creation of the Dimensionality Reduction Plot plot, it is possible to restrict analysis to highly variable genes. The data is then projected to a lower dimensional space using PCA. You can read about this feature in section 14.1.

The high confidence predicted cell types ("Cell type (high confidence)") are used to group the cells in the expression plots (Heat Map and Dot Plot) and Cell Abundance Heat Map, as well as for scoring the velocity genes. The Cell Abundance Heat Map additionally groups the cells based on the automated clusters obtained with resolution 1.0 ("Leiden (resolution=1.0)"). Any of these groups can be changed to:

- all predicted cell types ("Cell type (all)");
- automated clusters obtained with a different resolution x ("Leiden (resolution= x)"). All resolutions $0.1 \leq x \leq 1.5$ are produced, in steps of 0.1.

20.1.1 Output from Expression Analysis from Matrix

The workflow creates several output elements stored in a specific folder structure as indicated in figure 20.1. The reports are valuable for assessing whether the appropriate parameters have been used.

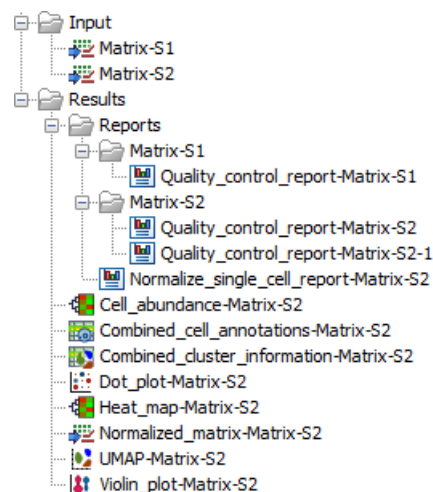










Figure 20.1: The Results folder contains the output elements produced by the Expression Analysis from Matrix workflow when run using the Expression Matrices in the Input folder. The Reports folder contains a subfolder for each of the batch units defined during execution. Here, the "Use organization of input data" was used. Note that Matrix-S2 contained two samples and so two QC reports are produced. Root elements originate from the combined analysis of all inputs.

20.2 Chromatin Accessibility and Expression Analysis from Matrix

The workflow Chromatin Accessibility and Expression Analysis from Matrix takes a pair of an **Expression Matrix** (📊) / (📊) and a **Peak Count Matrix** (📊) as input to jointly analyze scRNA-Seq and scATAC-Seq data originating from the same sample or samples.

The expression matrix is analyzed as described in section 20.1. Clustering and dimensionality reduction are performed using both expression and peak matrices.

The workflow allows for a combined analysis to produce:




- a single normalized **Expression Matrix** ();
- a **Dimensionality Reduction Plot** () associated with the automated clusters, predicted cell types and additional cell annotations;
- a **Heat Map** () , a **Dot Plot** () , and a **Violin Plot** () with the predicted cell types as cell groups;
- a **Cell Abundance Heat Map** () with the automated clusters and predicted cell types as cell groups.
- If velocity analysis is run:
 - a **Phase Portrait Plot** () with per gene information on the velocity dynamics;
 - a **Velocity Genes Scores** () element allowing identification of velocity genes driving the dynamics.

The workflow can be found here:


Template Workflows | Single Cell Workflows () | **From Imported Data** () | **Chromatin Accessibility and Expression Analysis from Matrix** ()

If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

Using a Fork element, the workflow offers the option to run velocity analysis. To enable this, set **Velocity Analysis** to **Run** in the **Specify Workflow Path** wizard step. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomics/current/index.php?manual=Fork.html> for details.

Choose either one or more **Expression Matrix** () / () and **Peak Count Matrix** () or **Select files for on-the-fly import** and select the format that is compatible with the selected inputs. Read more about import options in section 4.9.

Note that the sample in the inputs must be the same for cells originating from the same sample. This can be achieved in different ways, depending on how the elements were generated:

- If the input elements were generated in the CLC Single Cell Analysis Module, the sample name can be set when running Annotate Single Cell Reads, see section 6.1.4.
- If the input elements are imported, the sample name can be set during import through the **Cell format** or **Sample** options, see section 4.8.
- The tool **Update Single Cell Sample Name** () can be used for updating the sample name in either input element, see section 18.7.

The workflow offers a number of options. Note that not all parameters can be configured. Open parameters indicate places where customization may be necessary for different samples, but default settings are suitable in most cases.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html. When creating custom reference data sets, the chosen gene track needs to match the gene annotations used for training the provided **Cell Type Classifier** (🧠) (see section 8.3.1).

20.2.1 Output of Chromatin Accessibility and Expression Analysis from Matrix

The workflow creates several output elements stored in a specific folder structure as indicated in figure 20.2.

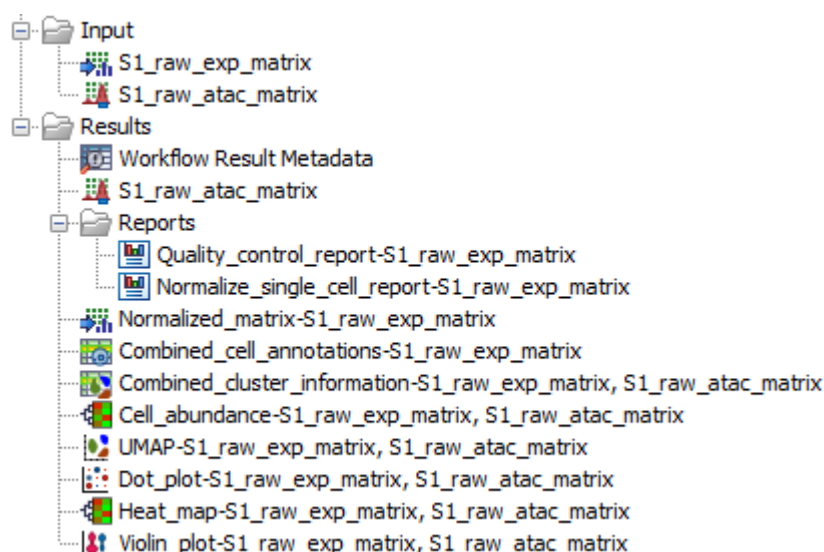









Figure 20.2: The Results folder contains the output elements produced by the Chromatin Accessibility and Expression Analysis from Matrix workflow

20.3 Immune Repertoire and Expression Analysis from Clonotypes and Matrix

The workflow Immune Repertoire and Expression Analysis from Clonotypes and Matrix takes one or more **Expression Matrix** (📊) / (📊) and **TCR Cell Clonotypes** (🧬) or **BCR Cell Clonotypes** (🧬) elements as input to jointly analyze scRNA-Seq and scV(D)J-Seq data originating from the same sample or samples. The Expression Matrices and Cell Clonotypes are sent on two different paths, for scRNA-Seq and scV(D)J-Seq data, respectively. The scRNA-Seq path follows the same analysis described in section 20.1, while the scV(D)J-Seq path ensures that clonotypes are filtered accordingly.

The workflow uses the iterate functionality and allows for a combined analysis of multiple samples to produce:

- a single, multi-sample, normalized **Expression Matrix** (📊) / (📊);
- a single, multi-sample, filtered **TCR Cell Clonotypes** (🧬) or **BCR Cell Clonotypes** (🧬) element;





- a **Dimensionality Reduction Plot** () associated with the automated clusters, predicted cell types, identified clonotypes and additional cell annotations;
- a **Heat Map** () , a **Dot Plot** () , and a **Violin Plot** () with the predicted cell types as cell groups;
- a **Cell Abundance Heat Map** () with the automated clusters and predicted cell types as cell groups.
- If velocity analysis is run:
 - a **Phase Portrait Plot** () with per gene information on the velocity dynamics;
 - a **Velocity Genes Scores** () element allowing identification of velocity genes driving the dynamics.

The workflow can be found here:


Template Workflows | Single Cell Workflows () | **From Imported Data** () | **Immune Repertoire and Expression Analysis from Clonotypes and Matrix** ()

If you are connected to a CLC Server via the CLC Workbench, you will be asked where you would like to run the analysis. We recommend that you run the analysis on a CLC Server when possible.

Using a Fork element, the workflow offers the option to run velocity analysis. To enable this, set **Velocity Analysis** to **Run** in the **Specify Workflow Path** wizard step. See <https://resources.qiagenbioinformatics.com/manuals/clcgenomics/current/index.php?manual=Fork.html> for details.


Choose either one or more **Expression Matrix** () / () and **TCR Cell Clonotypes** () or **BCR Cell Clonotypes** () elements or **Select files for on-the-fly import** and select the format that is compatible with the selected inputs. Read more about import options in section 4.9.

Note that the sample in the inputs must be the same for cells originating from the same sample. This can be achieved in different ways, depending on how the elements were generated:

- If the input elements were generated in the CLC Single Cell Analysis Module, the sample name can be set when running Annotate Single Cell Reads, see section 6.1.4.
- If the input elements are imported, the sample name can be set during import through the **Cell format** or **Sample** options, see section 4.8.
- The tool **Update Single Cell Sample Name** () can be used for updating the sample name in either input element, see section 18.7.

For the scRNA-Seq path, a number of options are customizable, see section 20.1. For the scV(D)J-Seq path, only clonotype filtering is customizable, as described in section 19.4. Adjustments can be made in a workflow copy, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Creating_editing_workflows.html.

The workflow can be run using **Single Cell hg38 (Ensembl)** or **Single Cell Mouse (Ensembl)** reference data sets (see chapter 2).

Note: Reference data elements cannot be configured during workflow execution. If other elements than those provided in the default reference data sets are needed, a custom reference data set can be used, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Custom_Sets.html. When creating custom reference data sets, the chosen gene track needs to match the gene annotations used for training the provided **Cell Type Classifier** () (see section 8.3.1). Reference V, D, J and C gene segments for other species or for B cells can be imported using **Import Immune Reference Segments** (see section 4.1).

The workflow allows the analysis of multiple samples and you can specify metadata during workflow execution. This is converted to cell annotations and can be used for coloring the cells in the Dimensionality Reduction Plot. However, the workflow expects each sample to be present in just one Expression Matrix, and attempting to define batch units containing more than one Expression Matrix will lead to a failure during execution. Similarly, each sample is expected to be present in just one Cell Clonotypes element.

For more details on configuring workflow execution with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html. Make sure to inspect the batch overview to check that the analysis will be performed correctly.

20.3.1 Output from Immune Repertoire and Expression Analysis from Clonotypes and Matrix

The workflow creates several output elements stored in a specific folder structure as indicated in figure 20.3. The reports are valuable for assessing whether the appropriate parameters have been used.

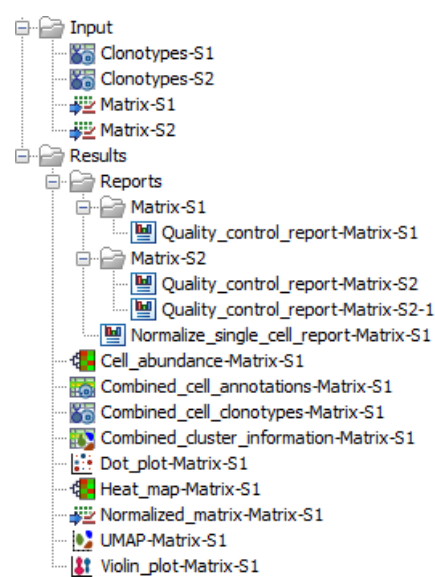


Figure 20.3: The Results folder contains the output elements produced by the Immune Repertoire and Expression Analysis from Clonotypes and Matrix workflow when run using the Expression Matrices and Cell Clonotypes in the Input folder. The Reports folder contains a subfolder for each of the batch units defined during execution. Here, the "Use organization of input data" was used. Note that Matrix-S2 contained two samples and so two QC reports are produced. Root elements originate from the combined analysis of all inputs.

Bibliography

- [Abdelaal et al., 2019] Abdelaal, T., Michielsen, L., Cats, D., Hoogduin, D., Mei, H., Reinders, M. J., and Mahfouz, A. (2019). A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome biology*, 20(1):194.
- [Amemiya et al., 2019] Amemiya, H. M., Kundaje, A., and Boyle, A. P. (2019). The encode blacklist: identification of problematic regions of the genome. *Scientific reports*, 9(1):1–5.
- [Bakken et al., 2018] Bakken, T. E., Hodge, R. D., Miller, J. A., Yao, Z., Nguyen, T. N., Aevermann, B., Barkan, E., Bertagnolli, D., Casper, T., Dee, N., et al. (2018). Single-nucleus and single-cell transcriptomes compared in matched cortical cell types. *PloS one*, 13(12):e0209648.
- [Bastidas-Ponce et al., 2019] Bastidas-Ponce, A., Tritschler, S., Dony, L., Scheibner, K., Tarquis-Medina, M., Salinno, C., Schirge, S., Burtscher, I., Böttcher, A., Theis, F. J., et al. (2019). Comprehensive single cell mrna profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development*, 146(12):dev173849.
- [Bentsen et al., 2020] Bentsen, M., Goymann, P., Schultheis, H., Klee, K., Petrova, A., Wiegandt, R., Fust, A., Preussner, J., Kuenne, C., Braun, T., et al. (2020). Atac-seq footprinting unravels kinetics of transcription factor binding during zygotic genome activation. *Nature communications*, 11(1):1–11.
- [Bergen et al., 2020] Bergen, V., Lange, M., Peidli, S., Wolf, F. A., and Theis, F. J. (2020). Generalizing rna velocity to transient cell states through dynamical modeling. *Nature biotechnology*, 38(12):1408–1414.
- [Bergen et al., 2021] Bergen, V., Soldatov, R. A., Kharchenko, P. V., and Theis, F. J. (2021). Rna velocity-current challenges and future perspectives. *Molecular systems biology*, 17(8):e10282.
- [Buenrostro et al., 2013] Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y., and Greenleaf, W. J. (2013). Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics. *Nature methods*, 10(12):1213.
- [Chao, 1987] Chao, A. (1987). Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, pages 783–791.
- [Chao et al., 2014] Chao, A., Gotelli, N. J., Hsieh, T., Sander, E. L., Ma, K., Colwell, R. K., and Ellison, A. M. (2014). Rarefaction and extrapolation with hill numbers: a framework for sampling and estimation in species diversity studies. *Ecological monographs*, 84(1):45–67.
- [Chao et al., 2013] Chao, A., Wang, Y., and Jost, L. (2013). Entropy and the species accumulation curve: a novel entropy estimator via discovery rates of new species. *Methods in Ecology and Evolution*, 4(11):1091–1100.

- [Gale and Sampson, 1995] Gale, W. A. and Sampson, G. (1995). Good-turing frequency estimation without tears. *Journal of quantitative linguistics*, 2(3):217–237.
- [Germain et al., 2020] Germain, P., Sonrel, A., and Robinson, M. (2020). pipecomp, a general framework for the evaluation of computational pipelines, reveals performant single cell rna-seq preprocessing tools. *Genome Biol*, 21(227).
- [Hafemeister and Satija, 2019] Hafemeister, C. and Satija, R. (2019). Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. *Genome Biol*, 20(296).
- [Ilicic et al., 2016] Ilicic, T., Kim, J. K., Kolodziejczyk, A. A., Bagger, F. O., McCarthy, D. J., Marioni, J. C., and Teichmann, S. A. (2016). Classification of low quality cells from single-cell RNA-seq data. *Genome biology*, 17(1):1–15.
- [Islam et al., 2014] Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., Lönnerberg, P., and Linnarsson, S. (2014). Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature methods*, 11(2):163.
- [Kang et al., 2018] Kang, H. M., Subramaniam, M., Targ, S., Nguyen, M., Maliskova, L., McCarthy, E., Wan, E., Wong, S., Byrnes, L., Lanata, C. M., et al. (2018). Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1):89.
- [Kobak and Berens, 2019] Kobak, D. and Berens, P. (2019). The art of using t-sne for single-cell transcriptomics. *Nature communications*, 10(1):1–14.
- [Kuchenbecker et al., 2015] Kuchenbecker, L., Nienen, M., Hecht, J., Neumann, A. U., Babel, N., Reinert, K., and Robinson, P. N. (2015). Imseq—a fast and error aware approach to immunogenetic sequence analysis. *Bioinformatics*, 31(18):2963–2971.
- [Kulakovskiy et al., 2018] Kulakovskiy, I. V., Vorontsov, I. E., Yevshin, I. S., Sharipov, R. N., Fedorova, A. D., Rumynskiy, E. I., Medvedeva, Y. A., Magana-Mora, A., Bajic, V. B., Papatsenko, D. A., et al. (2018). Hocomoco: towards a complete collection of transcription factor binding models for human and mouse via large-scale chip-seq analysis. *Nucleic acids research*, 46(D1):D252–D259.
- [La Manno et al., 2018] La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., Lidschreiber, K., Kastrioti, M. E., Lönnerberg, P., Furlan, A., et al. (2018). Rna velocity of single cells. *Nature*, 560(7719):494–498.
- [Lefranc et al., 2009] Lefranc, M.-P., Giudicelli, V., Ginestoux, C., Jabado-Michaloud, J., Folch, G., Bellahcene, F., Wu, Y., Gemrot, E., Brochet, X., Lane, J., et al. (2009). Imgt®, the international immunogenetics information system®. *Nucleic acids research*, 37(suppl_1):D1006–D1012.
- [Li et al., 2017] Li, H., Linderman, G. C., Szlam, A., Stanton, K. P., Kluger, Y., and Tygert, M. (2017). Algorithm 971: An implementation of a randomized algorithm for principal component analysis. *ACM Transactions on Mathematical Software (TOMS)*, 43(3):1–14.
- [Lun et al., 2019] Lun, A. T., Riesenfeld, S., Andrews, T., Gomes, T., Marioni, J. C., et al. (2019). Emptydrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biology*, pages 1–9.
- [Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

- [MacParland et al., 2018] MacParland, S. A., Liu, J. C., Ma, X.-Z., Innes, B. T., Bartczak, A. M., Gage, B. K., Manuel, J., Khuu, N., Echeverri, J., Linares, I., et al. (2018). Single cell rna sequencing of human liver reveals distinct intrahepatic macrophage populations. *Nature communications*, 9(1):1–21.
- [McInnes et al., 2018] McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- [Otsu, 1979] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.
- [Parkhomchuk et al., 2009] Parkhomchuk, D., Borodina, T., Amstislavskiy, V., Banaru, M., Hallen, L., Krobisch, S., Lehrach, H., and Soldatov, A. (2009). Transcriptome analysis by strand-specific sequencing of complementary dna. *Nucleic Acids Res*, 37(18):e123.
- [Platt et al., 1999] Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- [Robinson et al., 2010] Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.
- [Satopaa et al., 2011] Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B. (2011). Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE.
- [Stoeckius et al., 2018] Stoeckius, M., Zheng, S., Houck-Loomis, B., Hao, S., Yeung, B. Z., Mauck, W. M., Smibert, P., and Satija, R. (2018). Cell hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. *Genome biology*, 19(1):1–12.
- [Strino and Lappe, 2016] Strino, F. and Lappe, M. (2016). Identifying peaks in*-seq data using shape information. *BMC bioinformatics*, 17(5):343–361.
- [Taavitsainen et al., 2021] Taavitsainen, S., Engedal, N., Cao, S., Handle, F., Erickson, A., Prekovic, S., Wetterskog, D., Tolonen, T., Vuorinen, E., Kiviahio, A., et al. (2021). Single-cell atac and rna sequencing reveal pre-existing and persistent cells associated with prostate cancer relapse. *Nature communications*, 12(1):1–16.
- [Traag et al., 2019] Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.
- [Van Der Maaten, 2014] Van Der Maaten, L. (2014). Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245.
- [Wattenberg et al., 2016] Wattenberg, M., Viñes, F., and Johnson, I. (2016). How to use t-sne effectively. *Distill*.
- [Xu and Su, 2015] Xu, C. and Su, Z. (2015). Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12):1974–1980.