

User manual for

Ab Initio Transcript Discovery Plug-in 2.0

Windows, Mac OS X and Linux

February 13, 2012

This software is for research purposes only.

CLC bio
Finlandsgade 10-12
DK-8200 Aarhus N
Denmark

Contents

1 Introduction to the *Ab Initio Transcript Discovery Plug-in* **4**

2 Large gap mapper **5**

2.1 Overview 5

2.2 Parameters 5

3 Transcript discovery **8**

3.1 Algorithm and parameter description 8

3.2 Results 16

3.2.1 Annotations on the read mappings 16

3.2.2 Predicted genes table 17

3.2.3 Regions and events table 18

3.2.4 Summary report 19

3.2.5 Extract annotated reference sequences 19

3.3 Mitochondrion 19

4 Installation **21**

5 Uninstall **23**

Bibliography 24

Chapter 1

Introduction to the *Ab Initio Transcript Discovery Plug-in*

The *Ab Initio Transcript Discovery Plug-in* is designed to discover transcripts by mapping RNA-Seq sequencing reads to a genomic reference, allowing large gaps (for introns), followed by a transcript discovery process where transcripts are inferred from the read mappings. Relying heavily on reads mapped with a gap as evidence for transcripts, it is primarily developed for eukaryotic genomes.

The proposed work flow for using the *Ab Initio Transcript Discovery Plug-in* in combination with the existing RNA-Seq tool in the *CLC Genomics Workbench* is this:

1. Run the large gap mapper using all your RNA-Seq reads and a genomic reference sequence.
2. Run the transcript discovery algorithm on the resulting read mapping to predict transcripts and genes.
3. Inspect the results and if necessary re-run the transcript discovery to refine the settings to produce the desired result.
4. Part of the result from the transcript discovery is a copy of the reference genome including the new transcript and gene annotations. This can now be used as a common reference for measuring gene expression using the existing RNA-Seq tool in the Workbench

If you have sequenced several samples that need to be compared, we suggest using the reads from all samples for the large gap mapping and subsequent transcript discovery. In this way, you can establish a common set of reference transcripts and genes that makes it possible to compare gene expression levels across samples (using the RNA-Seq tool in the *CLC Genomics Workbench*). The initial read mapping created by the large gap mapper is then no longer used and can be deleted, unless you wish to be able to go back and double-check the basis of the prediction.

The current release is a beta version with full functionality **for single reads**. If you have paired reads, they are treated as single reads.

Chapter 2

Large gap mapper

2.1 Overview

The large gap mapper maps reads to a reference, while allowing for large gaps in the mapping. It is developed to support transcript discovery using RNA-seq data, since it is able to map RNA-seq reads that span introns without requiring prior transcript annotations. However, it may equally well be used for mapping any other type of reads, for example to find large deletions in genomic data.

The large gap mapper works by iteratively applying the standard read mapper of the *CLC Genomics Workbench* to each read as follows:

1. Find the best match for the read.
2. If the match is good enough (according to the settings, see below), the read is mapped to this position.
3. If there is an unaligned end which is long enough for the mapper to handle (17 bp for standard mapping, 18 bp for mapping in color space), this part of the read is used as input to step 1.
4. This continues until no reads have unaligned ends that are longer than 17/18 bp. Usually for 100 bp reads it will be maximum three rounds of mapping (corresponding to spanning two introns).

The matched region of the read identified in the first round of the mapping is called the seed segment (or just 'seed'). Matched regions found in later rounds are called non-seed segments.

2.2 Parameters

The large gap mapper is started from the Toolbox:

Toolbox | **High-throughput Sequencing ()** | **Large Gap Mapper**

After having specified the reads and the reference to which the reads should be mapped, the user must specify two parameters related to the mapped segments of a read (see figure 2.1:

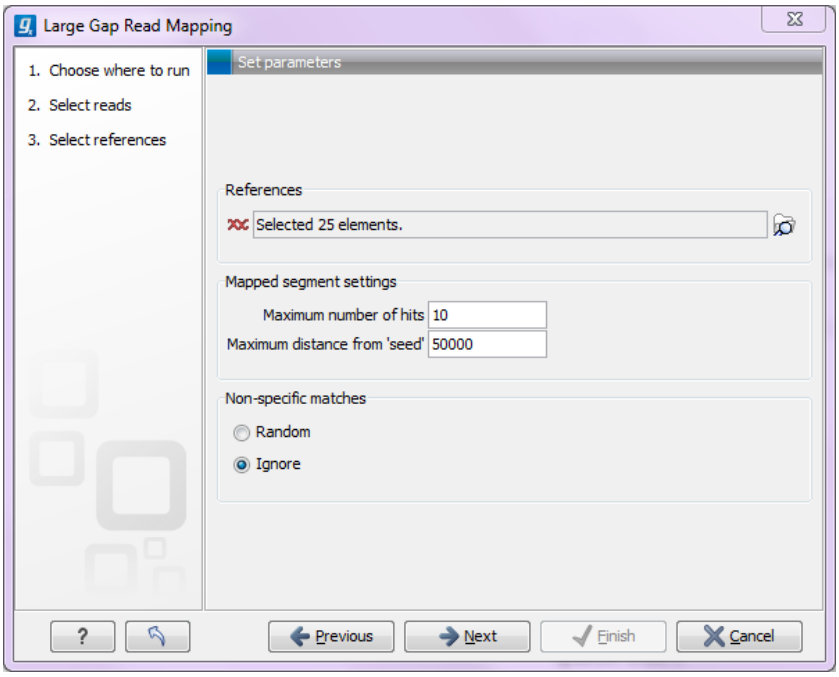


Figure 2.1: *Specifying parameters for the large gap mapper.*

Maximum number of hits is the maximum number of hits that a segment is allowed to have in order for the read to be mapped. If, for a non-seed segment, this number is exceeded, the read is classified as unmapped. If it is not exceeded, all the multiple hit positions will be considered. If the seed makes up the full read it may map in up to 'Maximum number of hits' positions.

Maximum distance from seed is the maximum distance allowed between seed and non-seed segments. Matches that are found further away from the seed that this value are discarded.

You can also specify whether **non-specific matches** should be distributed randomly or ignored.

Click **Next** to specify parameters related to the mapping quality. This is done in the **Mapping settings** step (see figure 2.2).

Here, the can specify the mapping settings. We refer to the user manual of *CLC Genomics Workbench* for further detail (you can find the manual in the **Help** menu or at `http://www.clcbio.com/usermanuals`). However, the the minimum similarity and length fractions need some more explanation: The **similarity fraction** is the required similarity between a mapped *segment* and the reference. This means that all segments must fulfill this requirement. Since segments can be as short as 17 bp, this threshold should not be set too strict (setting the threshold at 0.9 means that two errors for a segment of 17 bp would discard the match). The **length fraction** is the required fraction of the *full read* that should be mapped.

In addition to these user specified mapping settings, the large gap mapper requires that each mapped segment must comprise at least 10% of the read *and* must be of minimum length 17 bp (18 for color space).

Click **Next** to specify output options. In addition to the read mapping, the user can specify that a report on the mapping should be created, and that lists of unmapped and invalid mapped reads should be produced. The **unmapped read** list contains the reads which the large gap mapper

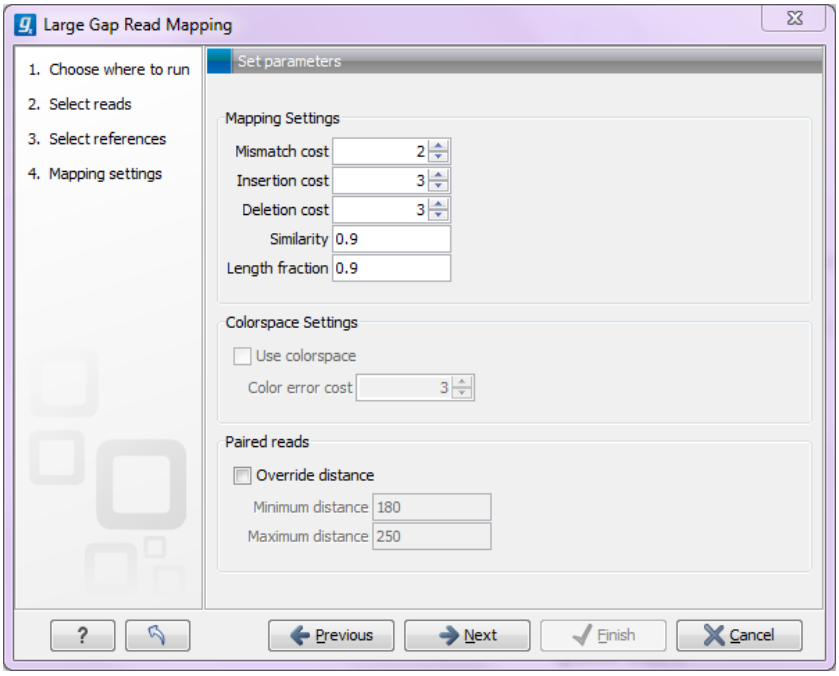


Figure 2.2: *Specifying parameters for the large gap mapper.*

was not able to map. The reads for which the large gap mapper was able to find a mapping, but for which the mappings of the segments where *incompatible*, are put in the **invalid mapped reads** list. The mappings of the segments of a read are incompatible if their positions are not consecutive along the reference, or if they do not have the same direction.

Chapter 3

Transcript discovery

The Transcript discovery tool is the second component in the set of tools in the *Ab Initio Transcript Discovery Plug-in*. The tool takes read mappings as input and produces gene and mRNA annotations. The annotations are generated by examining the read mapping and identifying likely regions of genes, their exons and splice sites, and, for each gene region, a set of transcript annotations that explain the observed exons and splice sites in this region.

3.1 Algorithm and parameter description

To start the transcript discovery:

Toolbox | **High-throughput Sequencing (**  **)** | **Transcript Discovery**

Select the read mapping produced by the large gap mapper and click **Next**[1]. You are now presented with choices regarding the overall mode of analysis as shown in figure 3.1.

Protocol If a strand specific protocol was used for generating the reads in the mapping, select this option. It means that the strandedness of the predicted exons and splice sites will be defined by the strands on which the reads map. If a strand specific protocol was not used, the splice sites (see below) will be used to determine the strandedness.

Use existing annotations This option allows you to enrich existing gene and mRNA annotations on the reference sequences of the mappings. It means that all existing gene and mRNA annotations will be kept, and new ones added only when the mapped reads suggest a new transcript or gene. When this option is not selected, existing gene and mRNA annotations — if present — will be ignored, and annotations will be generated solely based on the mapping of the reads.

Splice sites The algorithm will examine each gap in the read mapping to see if the gap is placed at, or without cost can be moved to, a valid splice site based on the ones you select in this list. Gaps that are placed at or moved to one of these splice sites are defined as *certain splice sites*.

Exclude uncertain splice sites For some gaps it will not be possible to place the gap at one of the defined splice sites, and these will be considered as gaps with *uncertain splice sites*

[1] Please note that if your mapping used paired data, the reads will be treated as single reads.

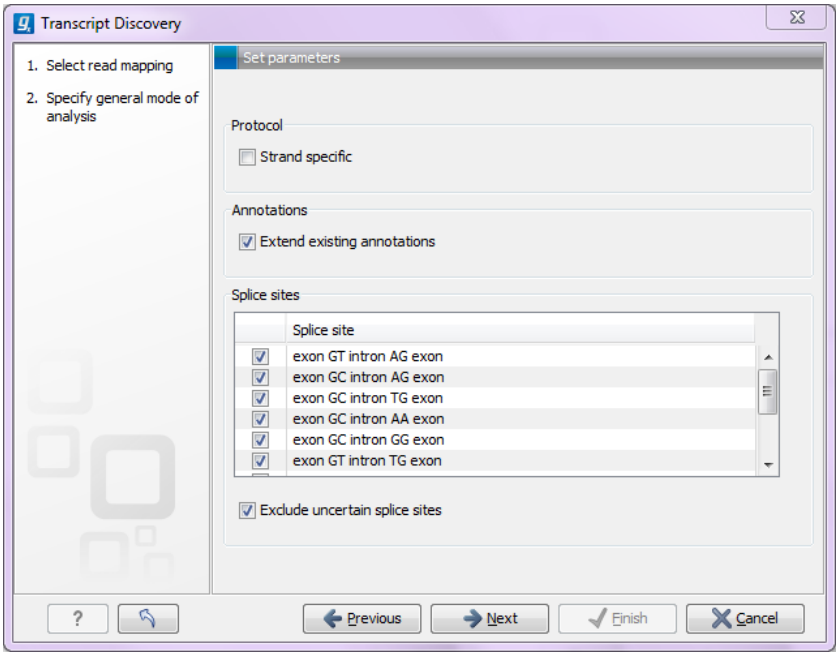


Figure 3.1: *Specifying the overall mode of analysis.*

and can be ignored by selecting this option.

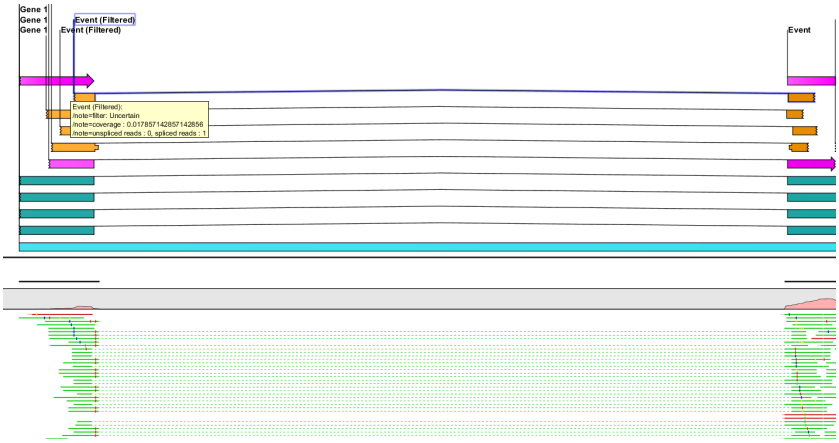


Figure 3.2: *Events with certain and uncertain splice sites.*

Click **Next** allows you to choose whether you wish to adjust a whole range of filtering parameters, or whether you prefer running the transcript discovery with default parameters (see figure 3.3).

If you select **Manual**, you will be able to adjust all the filters (explained below). The filters are mainly in place to eliminate as much noise as possible and provide thresholds for defining valid genes and transcripts. If you select **Automatic**, default settings will be used. Since the result is quite sensitive to the values used in the filters, it will often be beneficial to adjust these. Often, it will require running the analysis a few times, inspecting the results and adjusting the filters accordingly.

Before proceeding with describing all the filters, some introduction to the underlying algorithm is needed. It consists of ten steps, outlined below. The step-wise examination will be supplemented with explanations of the options in the dialogs.

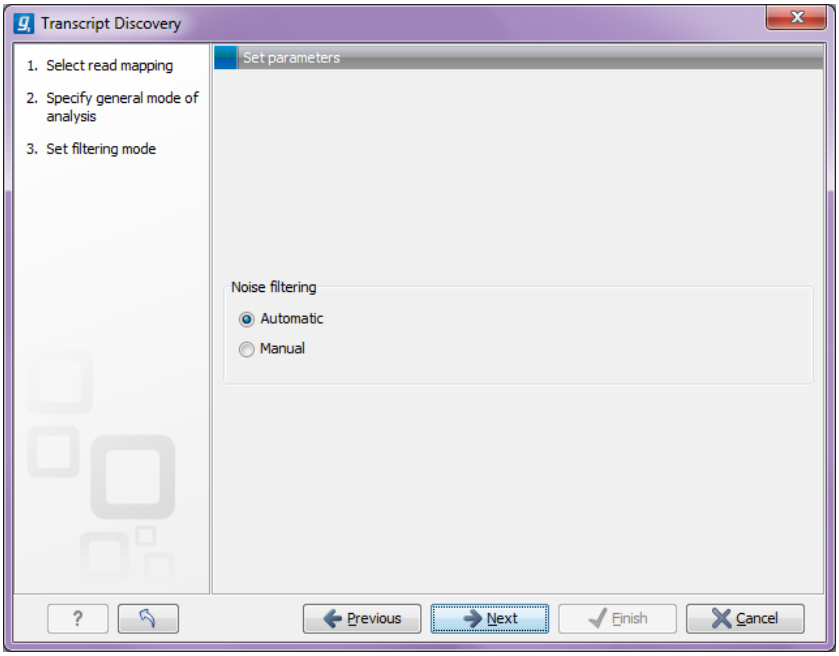


Figure 3.3: *Specifying if filtering should be automatic or manual.*

1. *Define events*. This step examines each mapped read and converts it to a predicted *event*. **In this first version of the plug-in paired reads are treated as two single reads, so that two independent events are created for reads in a pair.** An event has a 'region', defined by the mapping region(s) of the read. If the read mapping is un-gapped, the event is just called an 'event' and its region consists of a single interval. If the read mapping has gaps, the predicted event will be called a spliced event, and its region will consist of more intervals. If you have specified canonical splice sites (see figure 3.1), the mapping will be checked to see if the gap is placed at, or without cost can be moved to, one of these. Events also have *supporting read counts* of *spliced* and *un-spliced reads*. At this point they are 1 and 0 for spliced reads and 0 and 1 for un-spliced reads. The supporting read counts are used in later steps for filtering and merging events. If the **Use existing annotations** option is selected, a gene region is defined for each annotated gene, and an event is produced for each annotated transcript. These are referred to as *known* events.

   The 'Read filtering' options in the 'Event filtering' step are related to defining events:

   Ignore match duplicates For reads that are 100% identical, only one copy is used to define events. This is relevant for the 'supporting read counts' that are used when filtering events. When ticked, identical reads will only be counted as '1' in the read counts.

   Ignore non-specific matches Reads that have an equally good match elsewhere on the reference genome (these reads are colored yellow in the mapping view) can be ignored in the analysis. Whether you include these reads or not will be a tradeoff between sensitivity and specificity. Including them may lead to the prediction of transcripts that are not correct, whereas excluding them may mean that you will loose some true transcripts.

   The rest of the event filter settings in figure 3.4 will be explained further when the event filters are described below (step 7). Figure 3.5 shows the transcript and gene filters. These will be explained in more detail in under step 10.

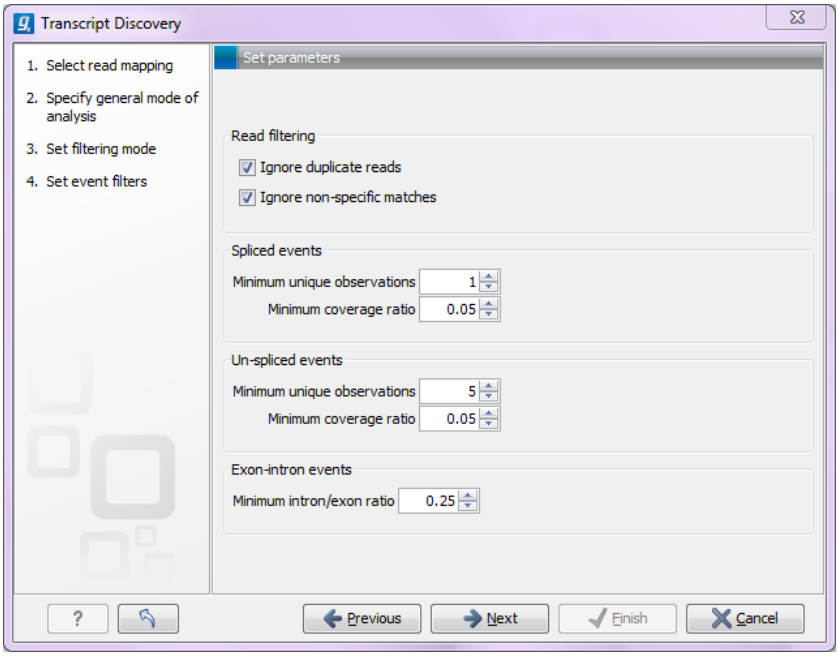


Figure 3.4: *Specifying event filters.*

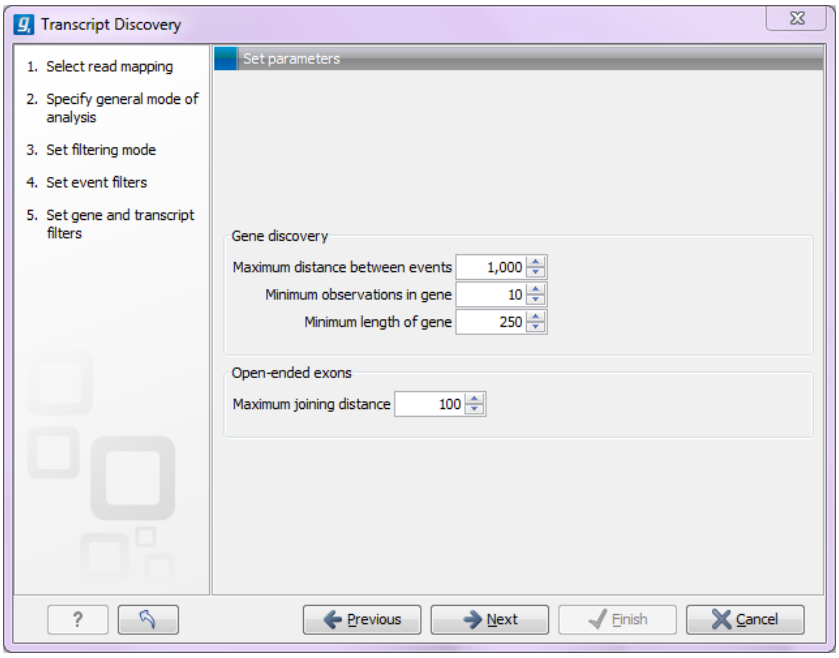


Figure 3.5: *Specifying transcript and gene filters.*

2. *Identify regions of coverage*. A *region of coverage* is a region in the mappings that contains events that are close enough to belong to the same gene. Regions of coverage are identified by iterating through the predicted events while calculating the distance between the current and the next event. If this is less than the **Maximum distance between events** value (see figure 3.5) the current region is extended. If not, a new region is started. If the 'Use annotations' option is ticked, the annotated gene regions take priority over the distance between events requirement; that is, if the next event lies within a region of an annotated gene that is different to that of the current event, a new region of coverage is started, even though the events are closer than the 'maximum distance between events' value.

Please note that a 'region of coverage' is *not* a gene region — it may contain more gene genes. The potential gene regions within a region of coverage are found in a later step (see step 6 below).

3. *First round of merging events*. To reduce the number of events before proceeding with the analysis, we merge events that unequivocally support the same splice sites. While doing this, the supporting read counts for the merged events are summarized. In this first round of merging, we merge *strictly overlapping events*. Two events are strictly overlapping if (1) the events overlap, (2) all introns and exons of the events in the overlapping region are the same and (3) the non-overlapping parts of the events do not extend across any splice site positions of any other events in the coverage region. The requirement (3) ensures that we do not merge an event with another event in cases where there are more events supporting different splice sites with which it could be merged.

4. *Fix end- and splice-points*. Many events will have end points that are slightly off, mostly because the first few (respectively last) bases of the intron and the end of the following (respectively preceding) exon are identical, or because they really should have been mapped with a gap, but a too short part of the read was present on one side of the gap for the mapper to map it, so instead it was mapped, possibly with a few nucleotides into an intron, and an unaligned end (see an example in figure 3.6).

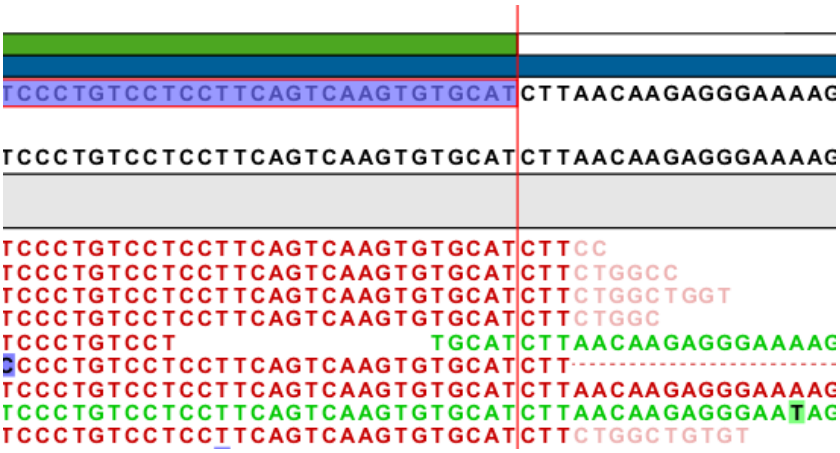


Figure 3.6: *The four reads at the top and the read at the bottom all have aligned three nucleotides into the intron (based on the existing transcript annotation) and with an unaligned end. These nucleotides actually map perfectly at the next exon, but the large gap mapper is not able to align parts that are smaller than 17 bp.*

Also, splice sites may not have been found, due to errors in the read. To account for this we fix end and splice points of the events. This is done for each coverage region, by collecting all the *certain* splice sites observed in the region (based on the sequence of the splice site, see figure 3.1) in a list. For each event in the region, it is then examined whether its end points or splice positions are close (less than 9 bp) to one of the sites in this list of observed certain splice sites. If so, the end or splice points of the event are moved, and the event is given the note **Boundaries are moved based on events in vicinity** (this can be seen on the event annotation as shown in figure 3.7).

5. *Second round of merging events*. The fixing of end and splice points above alters the set of events within a coverage region, and typically causes events that were not mergable before the fixing took place to now be mergable. To further reduce the number of events, we carry

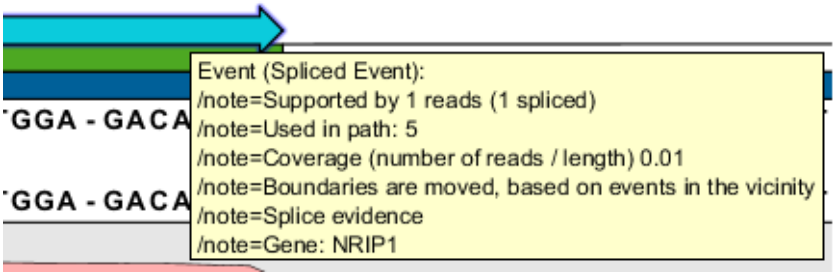


Figure 3.7: *The annotation tells you that the boundaries for this event were moved in the Fix end- and splice points step.*

out a second merging of events that unequivocally support the same splice sites, while again adding read counts of merged events. This time the merging consists of two steps:

Merge contained events For each event, it is examined if it is *contained* in another event. If so, it is merged with the other event, meaning that the supporting read counts of the contained event is added to those of the containing event. An event is contained in another event if all its introns and exons are present in the other event. An event can be contained in several other events, which means that the supporting read counts will be added to each of those events.

Merge strictly overlapping events see above ('First round of merging').

6. *Split in genes*. As some genes lie close and/or overlap, a coverage region may contain more genes. The next step is to split the events within a region of coverage into, possibly several, potential gene regions. If no annotations are used, the split-in-genes step just splits the events on strands:

   (a) Assign all events that are on the forward strand to a 'forward strand' region.

   (b) Assign all events that are on the reverse strand to a 'reverse strand' region.

   (c) For an un-stranded event: if it intersects with the region spanned by the forward stranded events, and not with that of the reverse stranded event, assign it to the forward strand region. It the opposite is the case, assign it to the reverse strand region. Else, assign it to the region whose' events have the highest summed read count.

   (d) Create a gene region for the events assigned to the forward strand, and another for those assigned to the reverse.

   If annotations are used, the procedure is more complicated:

   (a) Create a list of the regions of the known genes in the coverage region.

   (b) For each event in the region, collect a list of those of the known genes that it might belong to; the 'possible' genes. If the event is stranded, this list contains all of the known genes in the region that have the same strand as the event, and whose' region intersects with the event. If the event is un-stranded, the list contains all genes whose' region intersects with the event.

   (c) If there is just one possible gene for an event, we assign it to that gene. If there are more possible events, we prefer a gene that completely overlaps with the event. If there are more of these, we prefer the shortest. If there are no possible genes for an

event, but it intersects with an event that has been assigned to a gene and that is not on a different strand, assign it to the gene of that event. If there are events left after this that have not been assigned to a gene, use the 'split on strand' procedure described above, to create gene regions for them.

7. *Filter events*. Some events in the gene regions may be supported by just a few reads. To remove these *noise* events you can apply a number of event filters (see figure 3.4):

   Spliced events: Minimum unique observations (Filtering of spliced events with weak evidence). The minimum number of *unique* spliced reads that must support an spliced event. Events that do not meet this requirement are ignored. A read is unique if it 'counts' as specified by the 'Read filtering' options in 3.1: If the 'Ignore duplicate reads' option is ticked, identical spliced reads are counted as 1, and if the 'Ignore non-specific matches' is ticked, non-specific matches are not counted.

   Minimum coverage ratio (Filtering of spliced events with weak relative evidence). The *spliced coverage of a region* is calculated as the number of spliced reads in the gene region, divided by the total length of the region consisting of the union of the exons in the events in the region. Similarly, the spliced coverage of an event is calculated as the number of spliced reads supporting the event, divided by the length of the exon regions of the event. If the the spliced coverage of the event divided by the spliced coverage of the region is smaller than the user specified value, the event is ignored. Compared to the filter on absolute read count above, the coverage ratio filter allows filtering of events with weak evidence in regions of high coverage.

   Un-spliced events: Minimum unique observations (Filtering of events with weak evidence). The minimum number of unique un-spliced reads that must support an un-spliced event. Events that do not meet this requirement are ignored.

   Minimum coverage ratio (Filtering of events with weak relative evidence). The *un-spliced coverage of a region* is calculated as the number of un-spliced reads in the transcript event region, divided by the total length of the region consisting of the union of the exons in the events in the region. Similarly, the un-spliced coverage of an event is calculated as the number of un-spliced reads supporting the event, divided by the length of the exon regions of the event. If the the un-spliced coverage of the event divided by the un-spliced coverage of the region is smaller than the user specified value, the event is ignored.

   Exon-intron events: (Filtering of events with relatively weak evidence that extend across exon-intron boundaries). For each un-spliced event, it is examined if there are spliced events with which it is incompatible (if there is, the un-spliced event is an event that extends across an exon-intron boundary, but that does not have an alternative splice site). If so, the *coverage of the incompatible events* is calculated as the sum of the read counts of the incompatible events, divided by the length of the union of the exons in the incompatible events. If the ratio of the coverage of the un-spliced to that of 'coverage of the incompatible events' is smaller than the user-specified 'Minimum intron/exon ratio' value, the un-spliced event is ignored (see 3.8).

   In addition to these user controlled filters the algorithm applies two more filters:

   Exclude internal un-spliced All events that lie within introns of other events, and do not overlap with any other event are excluded. This ensures that spurious expression within introns is ignored.

Exclude external External events without spliced reads are excluded if their coverage is less than 25% of that of the event with maximum coverage in the gene region. External events are those that lie upstream or downstream of all events with splice evidence. This ensures that spurious weak expression most upstream or downstream in the region of coverage is ignored.

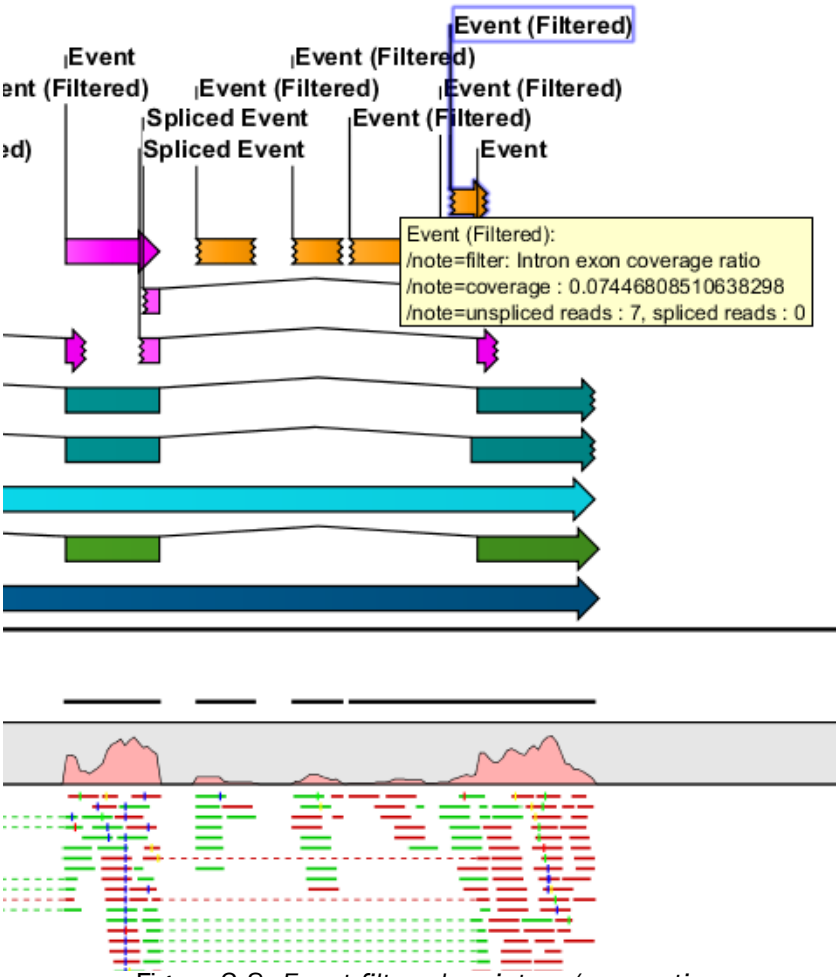


Figure 3.8: *Event filtered on intron/exon ratio.*

8. *Third merging of events*. Having excluded events in the filtering step, some events may now be mergable, that were not before the filtering. We thus apply a third round of merging of strictly overlapping events (described above).

9. *Predicting transcript annotations*. For each gene region, transcript annotations are created by a two-step procedure:

   Identifying events belonging to a transcript First we build a directed graph which has as nodes the events in the graph. Events are sorted by starting positions. A directed edge is added between nodes (from the first to the last) if the events are compatible. Two events are compatible if they, in their overlap, support the exact same splice junctions. We use the Dijkstra algorithm [Dijkstra, 1959] to identify a set of paths through the graph that completely covers the nodes. While building paths, the *supporting read counts* of the events are used as weights, and the path with the highest weight is

preferred. Each resulting path contains a set of events that belong to the same transcript.

Converting paths to transcripts For each path, a transcript is defined by merging the events in the path. Open-ended exons are joined if their distance is below the **Maximum joining distance** (see figure 3.5).

10. *Identifying genes*. Finally, a gene is predicted if the sum of the read counts of the events in the transcripts for the gene is above the **Minimum observations in gene** value (see figure 3.5), and the length of the gene (from beginning of first exon to the end of the last exon) is larger than the **Minimum length of gene** value (see figure 3.5).

3.2 Results

Before clicking **Finish**, you can decide which forms of output you want (see 3.9).

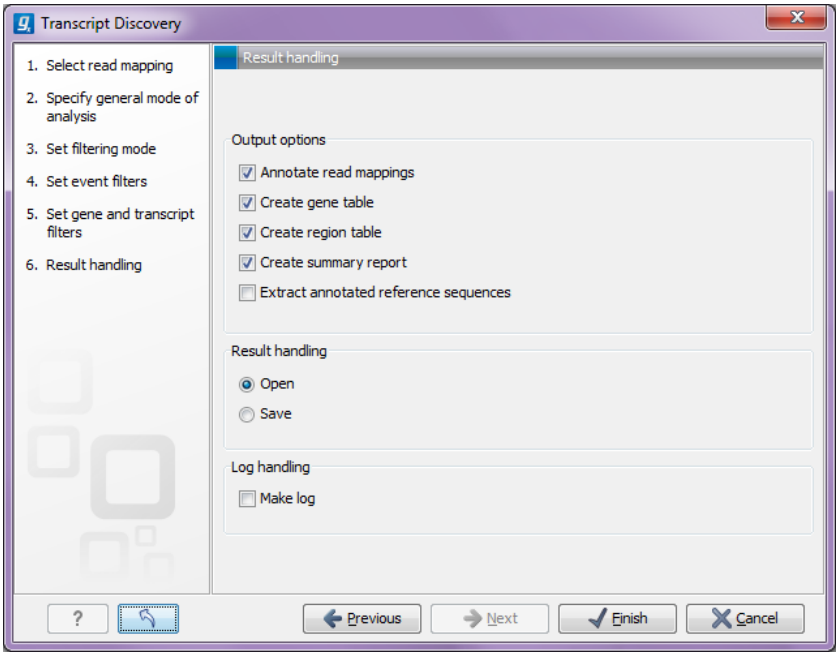


Figure 3.9: *Output options.*

The various outputs are described in the sections below.

3.2.1 Annotations on the read mappings

It can be very handy to see annotations reflecting the various steps in the transcript discovery directly in the read mapping view. An example is shown in figure 3.10

There are different kinds of annotations shown:

Event An 'Event' annotation is added for each event that is used in the graph building procedure in the step *Predicting transcript annotations*.

Event (filtered) An 'Event (filtered)' annotation is added for each event that was filtered in the step *Filter events*.

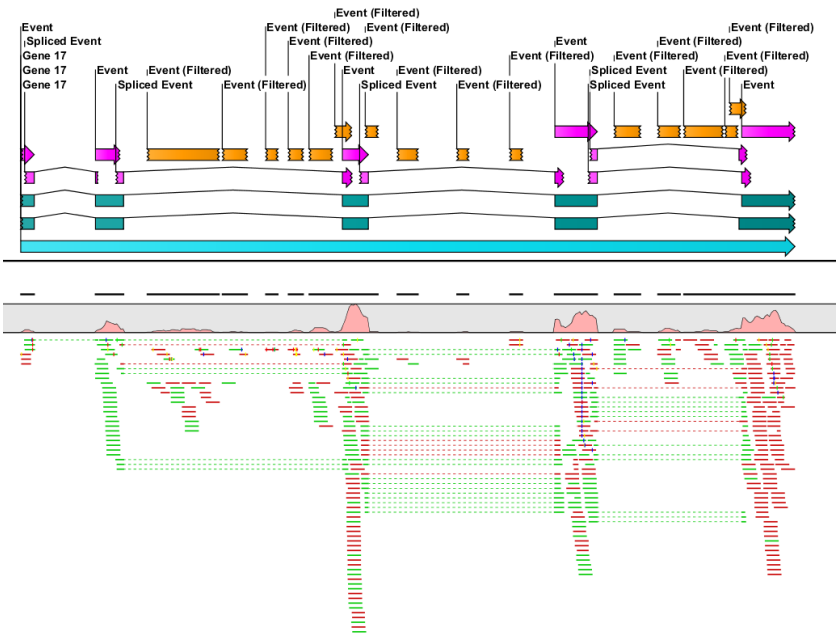


Figure 3.10: *All the annotations showing final predictions and the intermediate events.*

Event region (filtered) An 'Event region (filtered)' annotation is added for each gene region that was filtered in the step *Identifying genes*.

Predicted gene A 'Predicted gene' annotation is added for each predicted gene (that is, for each gene region that passed the filter in the *Identifying genes* step.

Predicted mRNA A 'Predicted mRNA' annotation is added for each predicted transcript for the predicted genes.

3.2.2 Predicted genes table

This table contains a row for each predicted gene. If the **Use existing annotations** option is selected, the existing gene annotations are also included in the table. When the 'Use annotations' option has been used the gene table has the following columns:

Reference The name of the mapping in which the gene was predicted.

Gene The name of the gene if it was annotated prior to the analysis. If it is a new predicted gene the name will be 'Gene' followed by a number (e.g. 'Gene 1').

Unknown No if the gene was annotated prior to the analysis; yes if it is a new predicted gene.

Length The length of the gene region.

Start The start of the gene region.

End The end of the gene region.

Strand The strand on which the gene was predicted.

Transcript The number of transcripts for the gene (including prior annotated as well as new predicted).

Known transcripts The number of prior annotated transcripts for the gene.

Unknown transcripts The number of new predicted transcripts for the gene.

Longest transcripts The length of the longest transcript for the gene.

Unknown events The number of unknown events (that is, events that are not contained in prior annotated transcripts) for the transcripts of the gene.

Unknown spliced events The number of unknown spliced events (that is, spliced events that are not contained in prior annotated transcripts) for the transcripts of the gene.

Reads The sum of the read counts of the events from which the transcript annotations were built.

Spliced reads The sum of the spliced read counts of the events from which the transcript annotations were built.

New 5' sequence Yes, if the gene region extends 5' of the prior gene annotation if there was one, else no.

New 3' sequence Yes, if the gene region extends 3' of the prior gene annotation if there was one, else no.

Splicing description A summary of the types of new splice sites found for transcripts for the gene ('Alternative acceptor/donor' and/or 'new exon'.

When the 'Use annotations' option was not used the columns 'Unknown', 'Known transcripts', 'Unknown transcripts', 'Unknown events', 'Unknown spliced events', 'New 5' sequence', 'New 3' sequence' and 'Splicing description' are not present.

3.2.3 Regions and events table

This table has a row for each *coverage region* that has been examined for potential genes and transcripts (see step 2 in section 3.1). It has the following columns:

Region the region of the coverage region.

Overlapping genes The names of overlapping genes, if any.

Total reads The number of reads in the events of the coverage region.

Un-spliced The number of spliced reads in the events of the coverage region.

Spliced The number of spliced reads in the events of the coverage region.

Events The number of events in the coverage region.

Events (after filtering) The number of events after filtering in the coverage region.

In addition, for each gene region within the coverage region, it will have a set of the following five columns:

Region The region of the gene.

Gene The name of the gene.

Events The number of events in the gene region before filtering.

Filtered The number of events in the gene region after filtering.

Filtered description A summary of how many event were filtered by each of the filters (e.g. 'Observations(2), Uncertain(2)').

3.2.4 Summary report

The summary report holds various statistics on the annotations generated in the analysis, such as distributions of the lengths of genes, the numbers of transcripts per gene and the numbers of exons per transcripts. Also, there are statistics on the filtering of events, spliced events, and genes. These summaries can be used to get an overview of the overall performance of the generation of annotations, and may give a rough indication of whether the filtering was appropriate for the user's particular aim. 3.11)

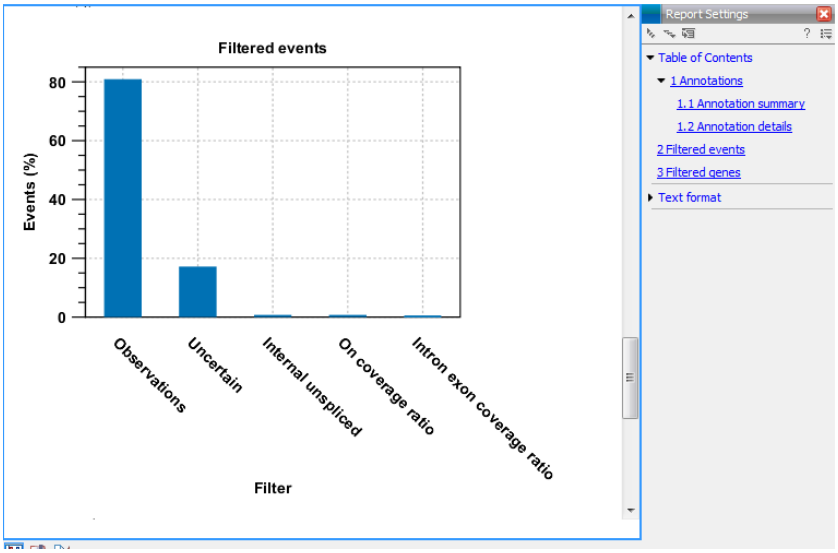


Figure 3.11: *An overview of the event filtering from the Ab initio transcript assembly report.*

3.2.5 Extract annotated reference sequences

When ticked, the reference sequences of the read mappings will be copied, the predicted gene and transcript annotations will be added to them, and they will be put in a sequence list. *These sequences can then be used as references in a subsequent RNA-seq analysis*. New predicted annotations will have the note 'Predicted by CLC bio transcript discovery'.

3.3 Mitochondrion

You will typically have very high coverage on the mitochondrion genome and this quickly becomes a problem for the transcript discovery. The high coverage means that there can often be millions of events that will take a very long time to analyze. For this reason, the transcript discovery will skip the reference sequences that have an average coverage higher than 1000.

For a standard human data set, this limit will target the mitochondrion genome specifically.

Please note that it is not recommended to exclude the mitochondrion genome as reference for the mapping since all the reads that map well on this reference sequence will try to find matches on other chromosomes. This will lead to false positive matches.

Chapter 4

Installation

The *Ab Initio Transcript Discovery Plug-in* is installed as a plug-in. Plug-ins are installed using the plug-in manager[1]:

Help in the Menu Bar | Plug-ins and Resources... ()

or **Plug-ins () in the Toolbar**

The plug-in manager has four tabs at the top:

- **Manage Plug-ins.** This is an overview of plug-ins that are installed.
- **Download Plug-ins.** This is an overview of available plug-ins on CLC bio's server.
- **Manage Resources.** This is an overview of resources that are installed.
- **Download Resources.** This is an overview of available resources on CLC bio's server.

To install a plug-in, click the **Download Plug-ins** tab. This will display an overview of the plug-ins that are available for download and installation (see figure 4.1).

Clicking a plug-in will display additional information at the right side of the dialog. This will also display a button: **Download and Install**.

Click the *Ab Initio Transcript Discovery Plug-in* and press **Download and Install**. A dialog displaying progress is now shown, and the plug-in is downloaded and installed.

If the *Ab Initio Transcript Discovery Plug-in* is not shown on the server, and you have it on your computer (e.g. if you have downloaded it from our web-site), you can install it by clicking the **Install from File** button at the bottom of the dialog. This will open a dialog where you can browse for the plug-in. The plug-in file should be a file of the type ".cpa".

When you close the dialog, you will be asked whether you wish to restart the CLC Workbench. The plug-in will not be ready for use before you have restarted.

[1] In order to install plug-ins on Windows Vista, the Workbench must be run in administrator mode: Right-click the program shortcut and choose "Run as Administrator". Then follow the procedure described below.

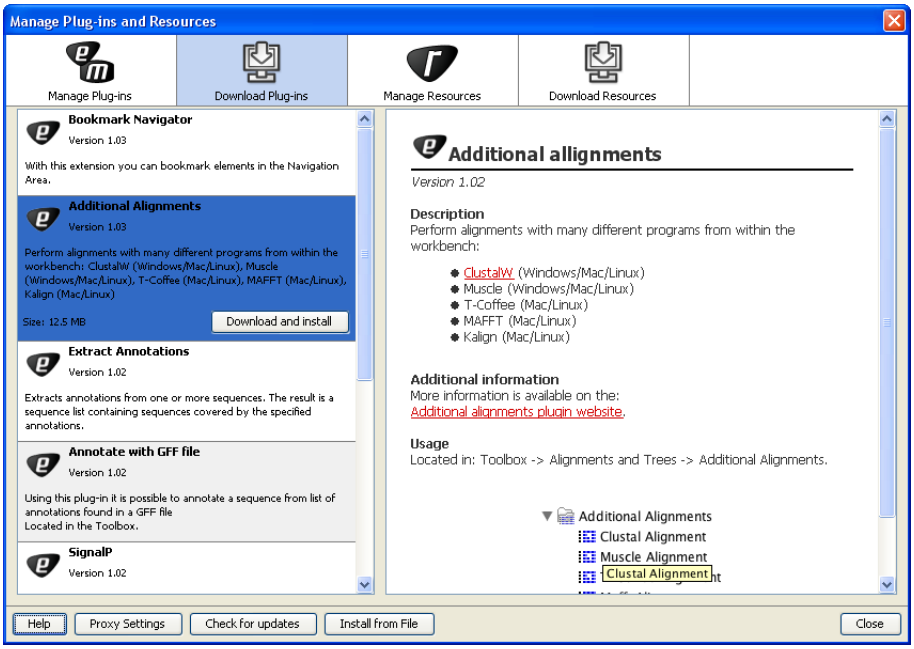


Figure 4.1: *The plug-ins that are available for download.*

Chapter 5

Uninstall

Plug-ins are uninstalled using the plug-in manager:

Help in the Menu Bar | **Plug-ins and Resources... (****)**

or **Plug-ins (****) in the Toolbar**

This will open the dialog shown in figure 5.1.

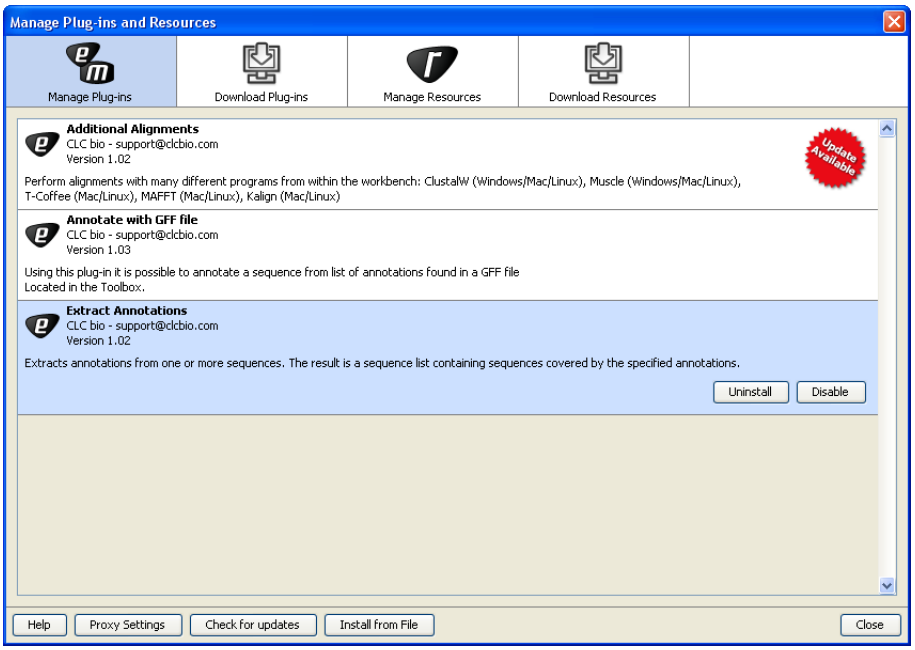


Figure 5.1: *The plug-in manager with plug-ins installed.*

The installed plug-ins are shown in this dialog. To uninstall:

Click the *Ab Initio Transcript Discovery Plug-in* | **Uninstall**

If you do not wish to completely uninstall the plug-in but you don't want it to be used next time you start the Workbench, click the **Disable** button.

When you close the dialog, you will be asked whether you wish to restart the workbench. The plug-in will not be uninstalled before the workbench is restarted.

Bibliography

[Dijkstra, 1959] Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.