Tutorial

Reference assembly in details

January 27, 2010

CLC bio
Finlandsgade 10-12 · 8200 Aarhus N · Denmark
Telephone: +45 70 22 55 09 · Fax: +45 70 22 55 19
www.clcbio.com · info@clcbio.com

Reference assembly in details

This tutorial goes into detail with reference assembly, also called read mapping, of the *CLC Genomics Workbench*. Whereas the Resequencing tutorial shows the overall work flow including reference assembly, this tutorial will go into much more detail about the reference assembly algorithm. You will learn what the parameters mean, and you will hopefully end up being able to make better decisions about the analysis of your own data.

We recommend going through the resequencing tutorial first, since it includes the basic introduction to doing reference assembly. Find it at http://www.clcbio.com/tutorials.

In this tutorial we use a subset of an *E. coli* data set where we focus on a small part of the genome and use a combination of 454 and Illumina GA reads.

Importing the data

First, download the data set from our web site: http://download.clcbio.com/testdata/Reference_assembly_details_tutorial.zip. Start the *CLC Genomics Workbench* and **Import** () the file. You will now have a folder structure like the one shown in figure 1.

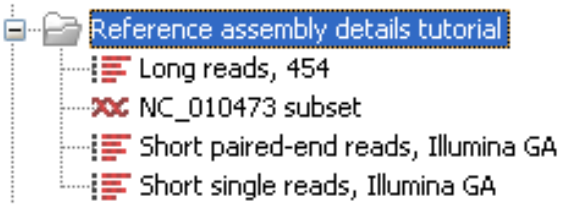


Figure 1: *The data set consisting of reference sequence (subset of genome) and three sets of sequencing reads.*

Assembling long reads

The *CLC Genomics Workbench* uses two different algorithms for assembling reads to a reference depending on the read length. First, we consider long reads. Reads are defined as "long" when they are longer than 55 bp.

First, perform a reference assembly with standard settings using the `Long reads, 454` as input and the `NC_010473 subset` as reference. You can set the parameters to default by clicking the button () at the bottom of the dialog. You do not need to create a log in the last step before you click **Finish**.

Change the compactness in the **Side Panel** to **Low** and deselect all **Annotation types** except **Repeat region**. To save these adjustments so that they take effect next time you open a contig, click the **Save/Restore Settings** button () at the top of the **Side Panel** and click **Save Settings**. Give your settings a name and make sure the check box to **Always apply these settings** is checked.

Now zoom out so that you have a view similar to the one shown in figure 2 and go to around position 2060.

Note that there is something special going on just at the border between the two annotated repeat regions. If you **Zoom in** () to see the details, you can see that for many reads, the end of the read is faded. This is because this part of the read was not aligned to the reference during

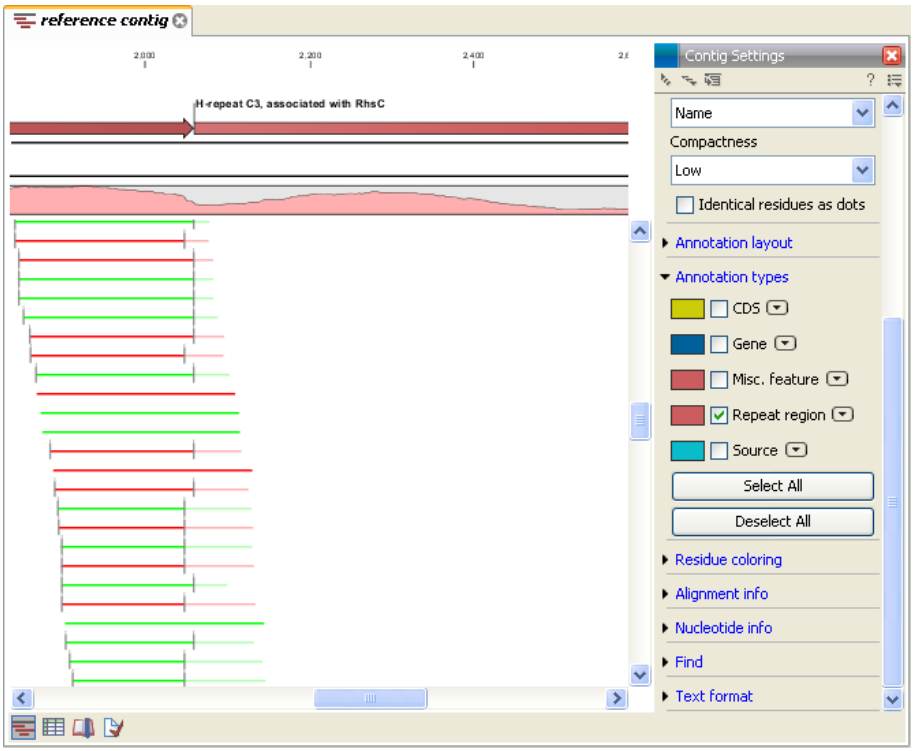


Figure 2: *Assembling reads to a repeat region.*

reference assembly. On closer inspection you can see that this part of the read doesn't match the reference sequence (figure 3).

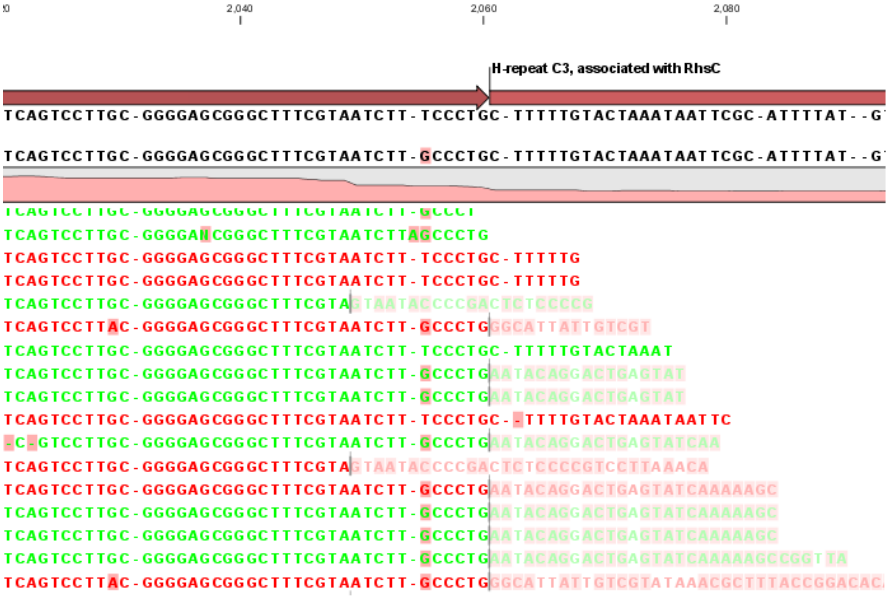


Figure 3: *Unaligned ends.*

The reason why you see this is because the *CLC Genomics Workbench* performs *local alignment* of the reads. This means that it only aligns the part of the read that matches well – the rest is left unaligned.

Assembling long reads in *CLC Genomics Workbench* is a two-step process:

1. For each read, the optimal local alignment between the read and the reference sequence is found.
2. Second, all reads are filtered according to user-defined criteria for length and similarity of the local alignment

To illustrate this, repeat the reference assembly but this time adjust the **Length fraction** to 1 as shown in figure 4.

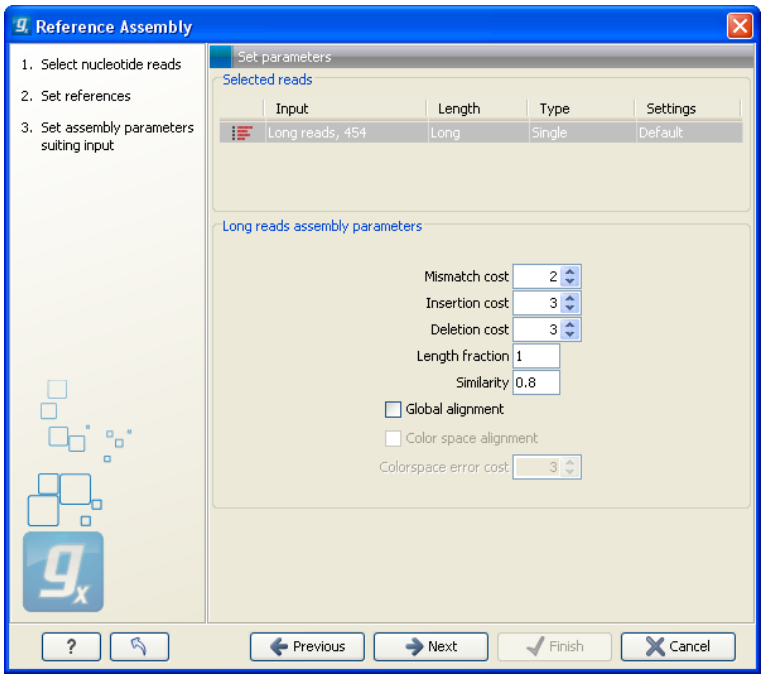


Figure 4: *Stricter alignment settings.*

This means that the reads will be filtered so that only reads matching in their entire length will be included in the contig. Finish the reference assembly with this setting and see how it affects the same region. As expected, all the reads with un-aligned ends are gone (see figure 5), and you are left with less reads (the coverage is around half compared to figure 2).

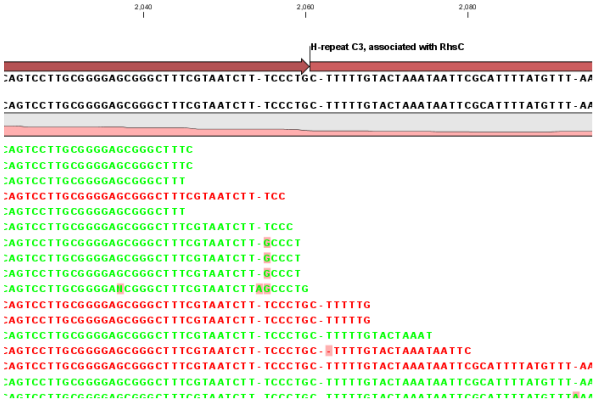


Figure 5: *Stricter alignment settings result in fewer reads assembled.*

If you go back and look at figure 3, you can see that besides the unaligned ends, there are also a number of mismatches in the reads. These mismatches are internal to the local alignment, whereas the unaligned ends were outside the local alignment.

Just as with the unaligned ends, you can control the level of similarity between reads and reference using the **Similarity** setting (see figure 4). Run the assembly again, this time with a **Similarity** of 1. Set the **Length fraction** back to 0.5. You would now expect to have removed all the reads that contain mismatches. But if you look at the result, you will see that this is not the case. Figure 6 shows an example around position 2250, where several reads contain mismatches.

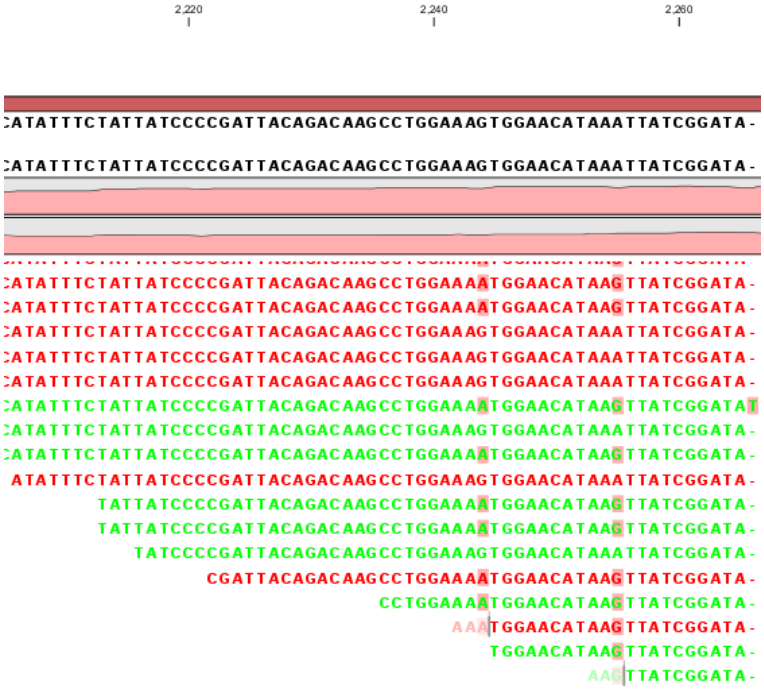


Figure 6: *Still mismatches although similarity was set to 1.*

The reason why you see this is that we have set the **Length fraction** back to 0.5. This means that we have accepted that *half the read should match perfectly to the reference*. The other half is allowed to diverge from the reference.

Besides the **Length fraction** and **Similarity**, you can also specify gap and mismatch costs (see figure 4). These determine how the initial local alignment should be performed. Setting a low mismatch cost and high insertion/deletion costs will favor mismatches over gaps in the local alignment. We are not going into details with these settings in this tutorial.

Assembling short reads

As mentioned in the beginning, the *CLC Genomics Workbench* uses two different algorithms for assembly, depending on the read length. We will now take a look at the short reads. In order to show how different settings affect the result, we are going to perform one reference assembly with very strict settings, requiring a perfect match, and we will then take the unassembled reads and assemble them with more relaxed settings. First, close the open views by pressing Ctrl + Shift + W (⌘ + Shift + W on Mac). You do not need to save the results.

Select the `Short single reads, Illumina GA` and open the reference assembly dialog. Use the same reference sequence as before. In the third step, set **Limit** to 1 and check the **Global alignment** checkbox (see figure 7). Note that we will come back to have a closer look at these settings later, but to summarize it means that we require a perfect match for the full read.

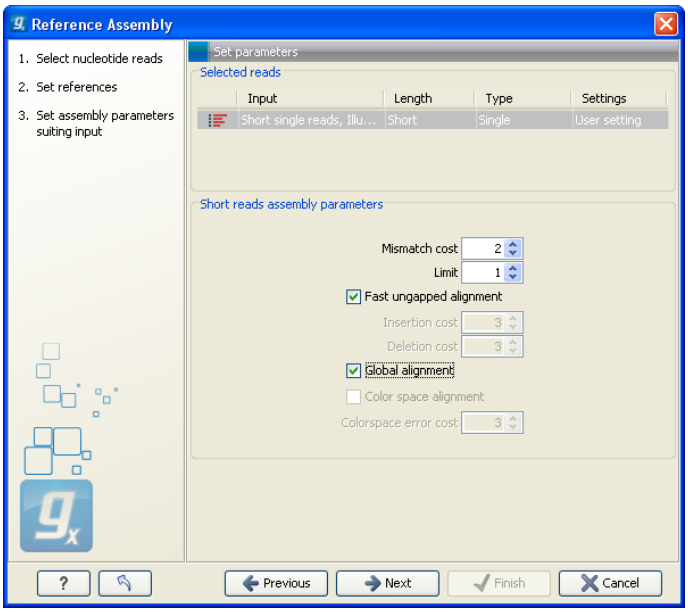


Figure 7: *Very strict alignment settings for short reads.*

Go to the last step and check the **Create list of non-assembled sequences**. The result of the assembly will this time be a contig and a list of the reads that didn't match the reference. Save this list and do a new reference assembly with this list as input. This time, raise the **Limit** to the maximum of 11, uncheck **Fast ungapped alignment** (see figure 8) and uncheck **Global alignment**. This is the most relaxed setting we can make which allows mismatches, gaps and unaligned ends (explained in detail below).

We now have a contig consisting of reads that do not match the reference perfectly. If you go to the same position on the reference sequence as we examined before (2060), you can see what this means for the alignment of the reads (see figure 9).

First of all, this means that some reads have mismatches (see the top read in figure 9 which has two mismatches). Second, it means that there are un-aligned ends just like we saw when aligning the long reads. With the strict assembly settings in figure 7, we selected the global alignment option which means that no unaligned ends are accepted, no matter how the Limit is set. But with the relaxed settings, we are doing local alignment (as you can see from figure 8).

To explain what kind of mismatches and unaligned ends that are accepted, we need to inspect the details of the short read assembly algorithm. It is based on a scoring system where a match is rewarded one point. The other scores for mismatch, insertion and deletion can be adjusted in the dialog (see figure 8). The defaults are:

Match	$+1$ (cannot be changed)
Mismatch	-2
Deletion	-3
Insertion	-3

Let's take a few examples and calculate the score of an alignment. Below is a read of 20 bp

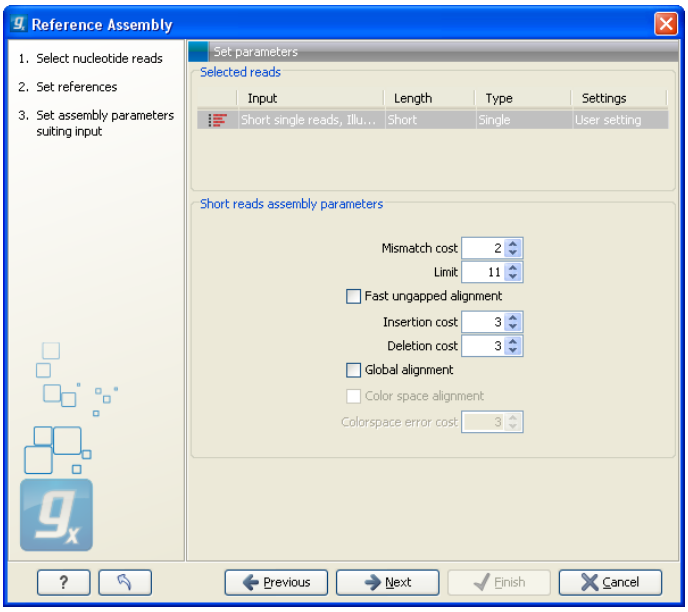


Figure 8: *Very relaxed alignment settings for short reads.*



Figure 9: *Alignment of non-perfect matches.*

perfectly aligned to the reference. The score is thus 20.

```
CGTATCAATCGATTACGCTATGAATG
   ||||||||||||||||||||      20
   ATCAATCGATTACGCTATGA
```

If we introduce a mismatch in the middle, we subtract 2 from the score. In addition, we also lose a match (we only have 19 matches left), so the score ends up at 17:

```
CGTATCAATCGATTACGCTATGAATG
   ||||||||  ||||||||||      17
   ATCAATCGGTTACGCTATGA
```

If we have a mismatch near the end, we will simply exclude this base from the alignment. This means that we do not consider it as a mismatch but only one base less in the alignment. So the score ends at 19:

```
CGTATCAATCGATTACGCTATGAATG
    ||||||||||||||||||      19
   TTCAATCGATTACGCTATGA
```

As an exercise, we have listed a few alignments where you can calculate the score yourself (see answers in the foot note).[1]

```
CGTATCAATCGATTACGCTATGAATG
    |||||||  ||||||||||       A
   CTCAATCGGTTACGCTATGA

CGTATCAATCGATTACGCTATGAATG
    |||||||  ||||  ||||||       B
   TTCAATCGGTTACCCTATGA

CGTATCAATCGATTACGCTATGAATG
    |||||||  ||||  |||||       C
   TTCAATCGGTTACCCTATGC
```

The alignment score is then compared to the read length, and the read is accepted if the difference does not exceed the **Limit** you specify in the dialog (e.g. 11 in figure 8). Example A above with a read length of 20 and a score of 15 would be accepted if the limit is 5 or higher. With the default settings where the limit is 8, it means that up to 2 mismatches are allowed or that 8 bases can be left un-aligned.

Making use of paired-end information

Perform a new round of reference assembly with the `Short single reads, Illumina GA` data set using default parameters. You can set the parameters to default by clicking the button () at the bottom of the dialog.

[1] Answers: A=15, B=13, C=12

In the resulting contig scroll to around position 2020. Here you will see reads colored in yellow (see figure 10).



Figure 10: *Alignment of non-specific hits.*

The yellow color means that these reads could have been matched equally well other places on the reference sequence. They are what we call **Non-specific matches**. We see this because the reference sequence includes repetitive regions. You can show a graph of the frequency of non-specific matches in the **Side Panel** under **Alignment info**. Zoom out and you will see a view as shown in figure 11.

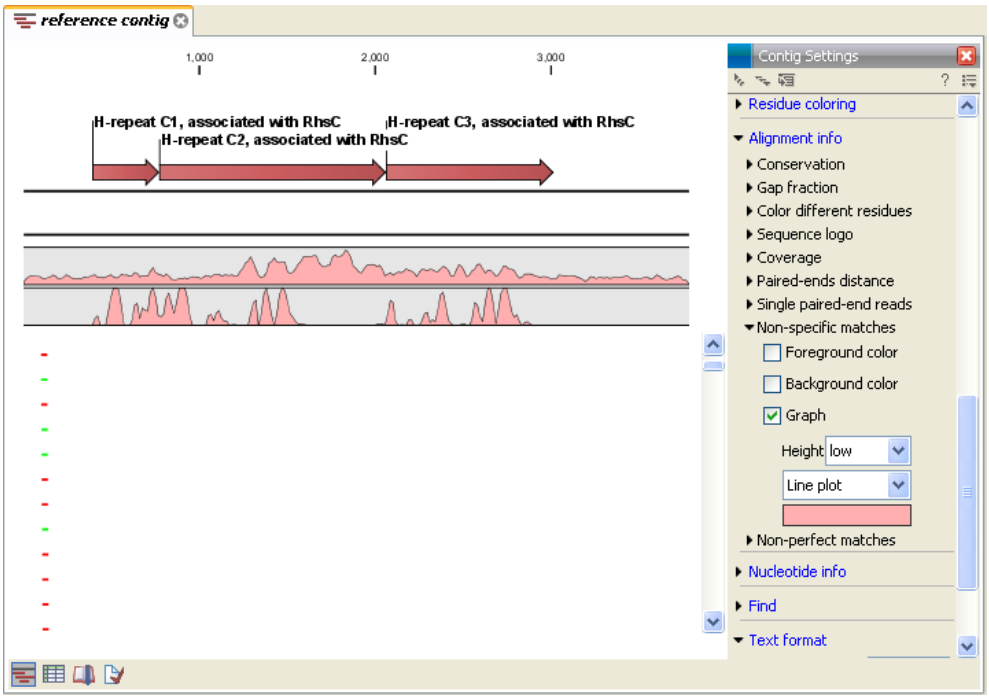


Figure 11: *Showing a graph of the frequency of non-specific hits.*

This graph shows the percentage of the aligned reads that are non-specific. You can **Zoom in** (🔍) by dragging a rectangle around one of the peaks to see the placement of the reads.

The reads in this data set are very short – only 35 bp. Until now we have been working with the `Short single reads, Illumina GA` data set, but let's see what happens if we do the same thing with the paired-end data set. Perform a new reference assembly with default parameters with the `Short paired-end reads, Illumina GA`. This data set is identical to the single reads data set except for the paired-end information. Display the **Non-specific reads** graph and zoom out. Your view should now look like figure 12.

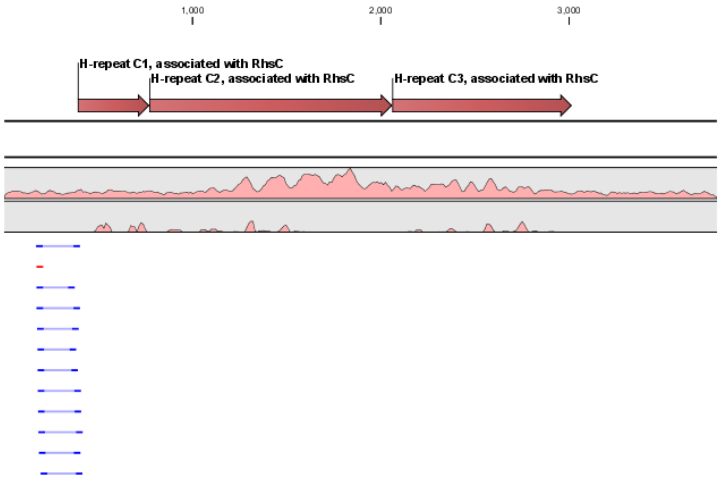


Figure 12: *The number of non-specific reads decreases dramatically when paired-end information is used.*

Compared with the single reads data set, there are considerably less non-specific reads in this paired-end data set. This is because the Workbench now has much more information that can be used to place the reads unambiguously: The paired-end read has 70 bp and spans a region of approximately 215 bp of the reference sequence. Although the total number of base pairs are the same in the two data sets, the extra information in the paired-end read structure produces a contig of much higher quality.

The paired-end data can also be used to visualize genomic re-arrangements. You can read more about this in the user manual by searching for "Interpreting genomic re-arrangements".