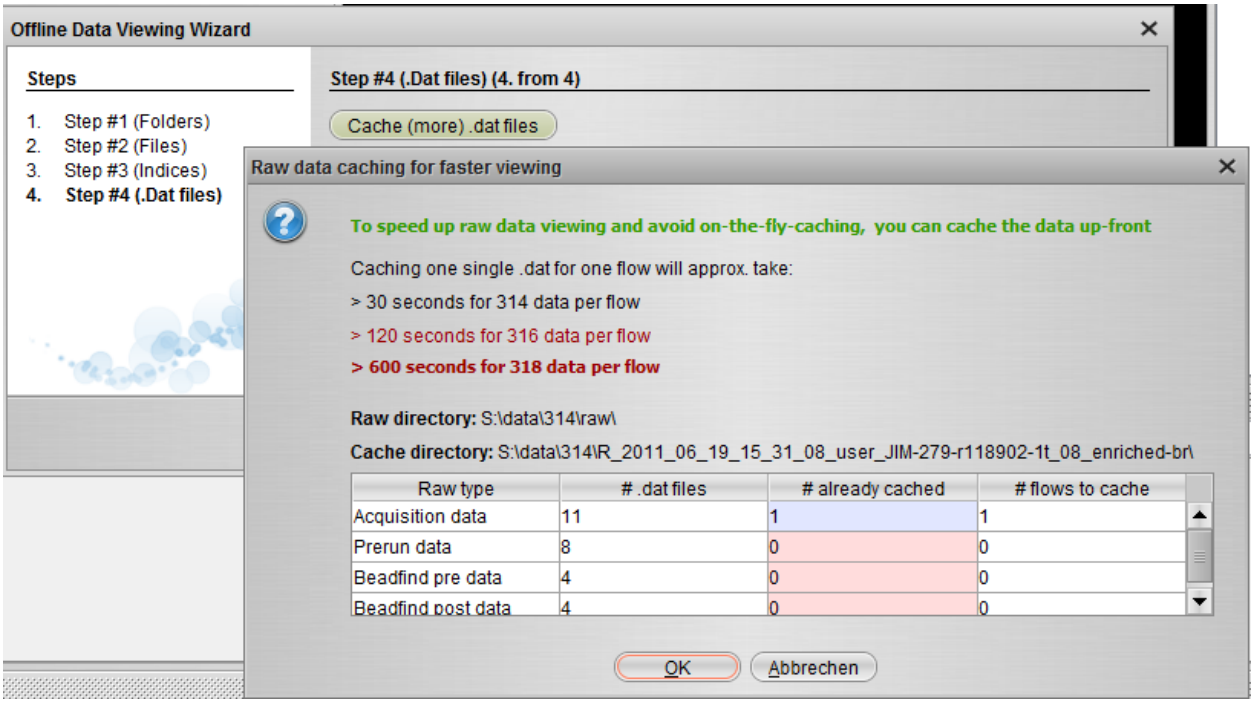
Torrent Scout Additional Use Cases

Table of Contents

Viewing the Raw Signal from the .dat files

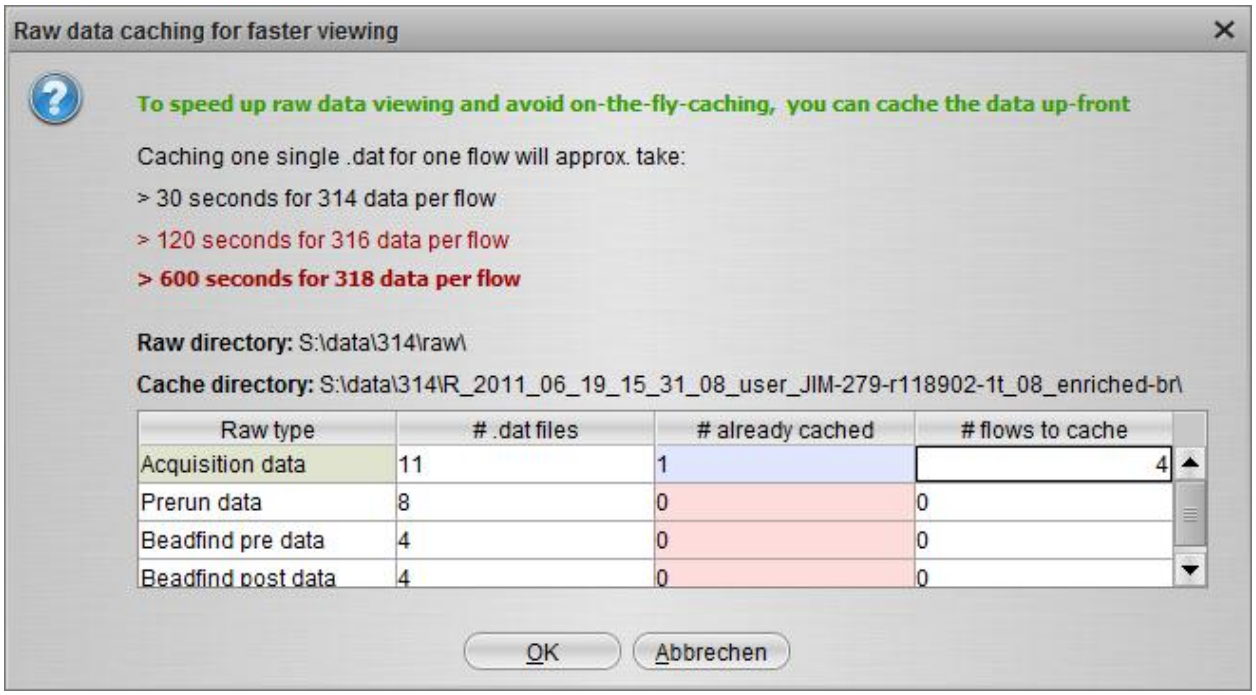
Preparing the Data For Viewing

- Before the data can be viewed efficiently, the data needs to be prepared for viewing
- In the offline component, in step 4 of the Indexing and Caching Wizard, make sure you select the "Cache (more) .dat files" button



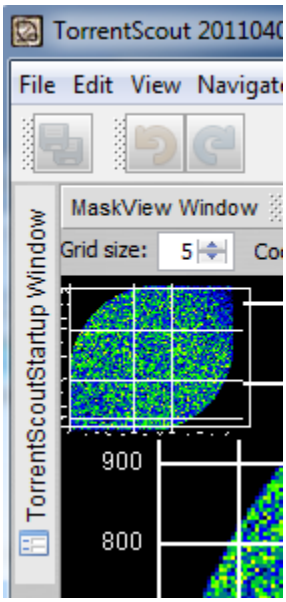
- In the first column, you see how many files you have (11 in our case). The second column shows how many flows we have already prepared (1), and in the last column, you can specify how many flows you wish to prepare for viewing. Let's put in 4: select the cell, enter 4, and make sure you

either hit enter or leave the cell so that the value is entered:

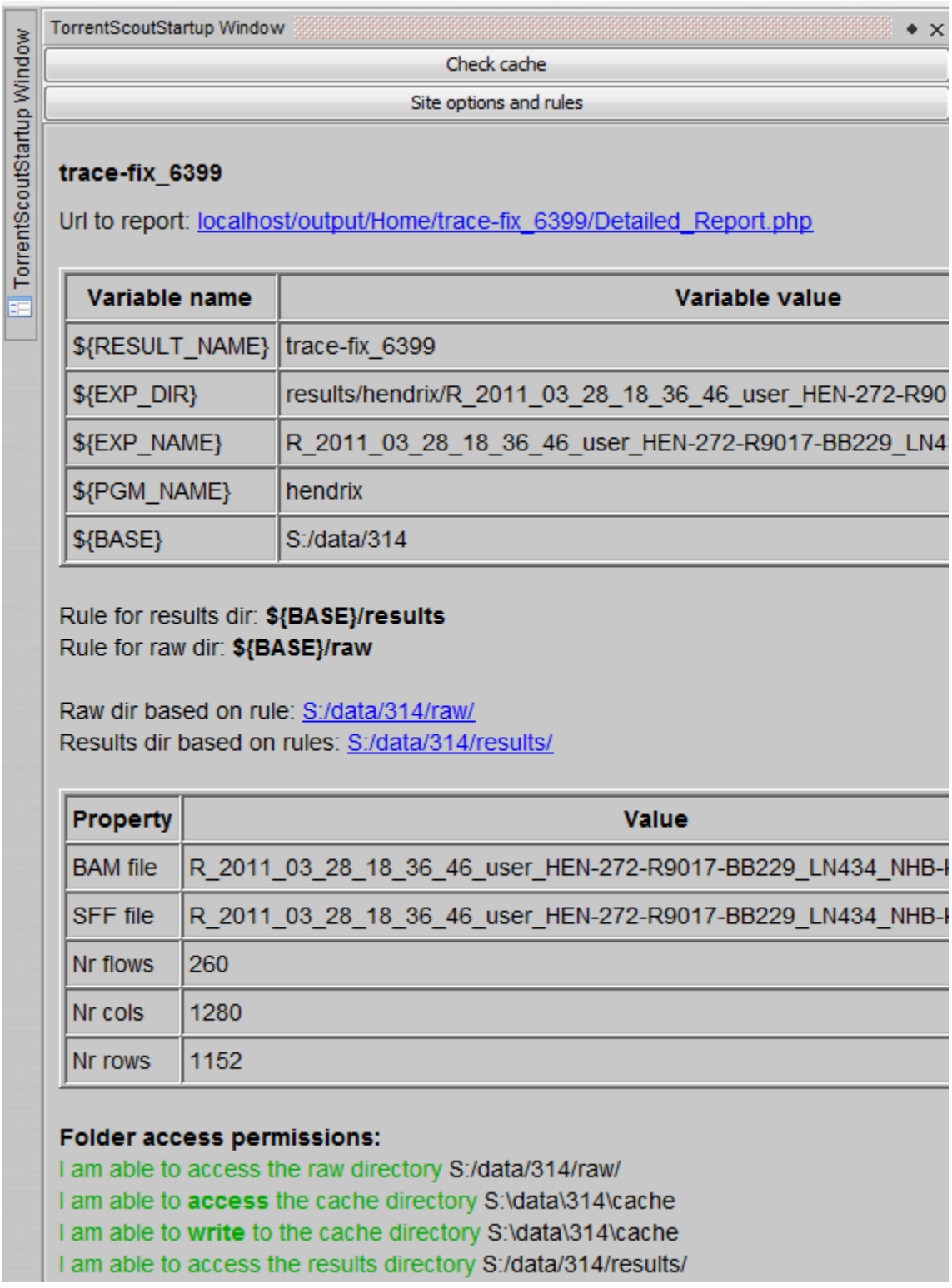


Note that the .dat processing can take a very long time if you select many flows. In can take hours if you select larger chips (this also depends on the computer used). If your computer does not have at least 2-3GB of memory, it may not be able to process the larger chips!

- As an alternative, you can also click on the TorrentScoutStartup Window on the very left side of the application:

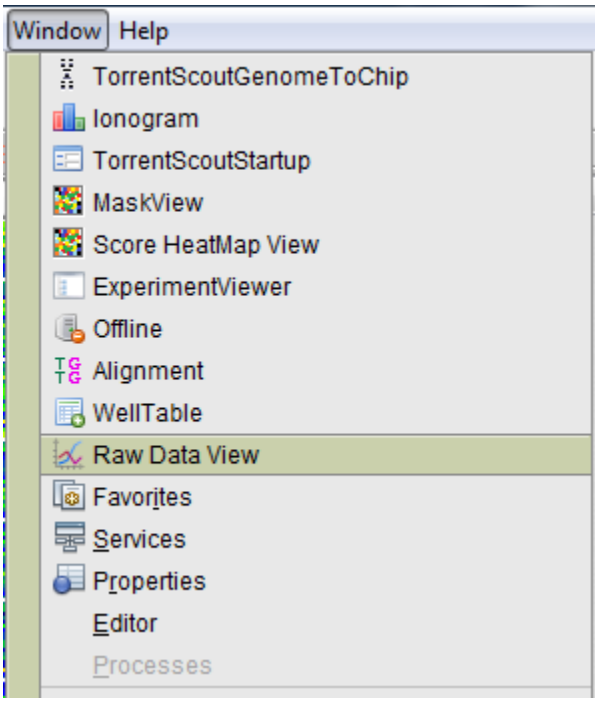


- This opens a page to show you basic information about the experiment context, and there is also a “Check cache” button on top that opens the same caching panel as before:

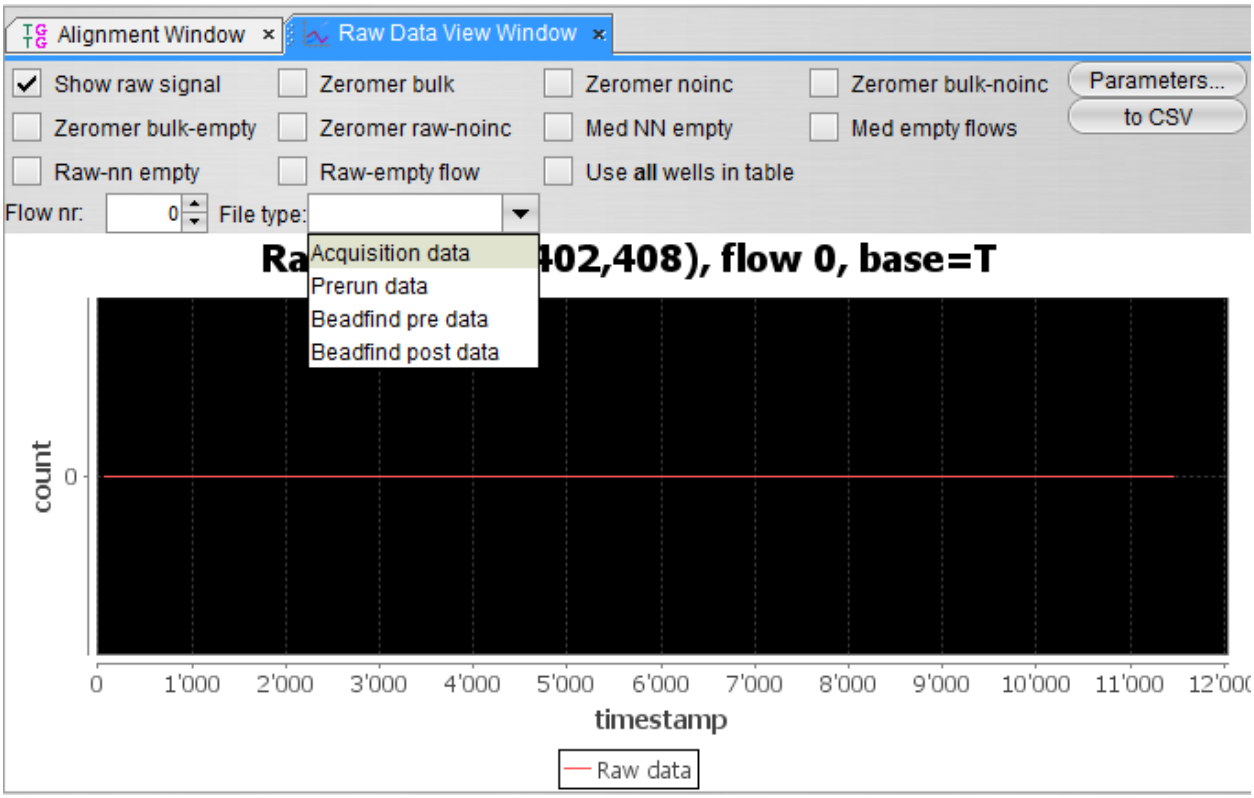


Opening the Raw Data Viewer

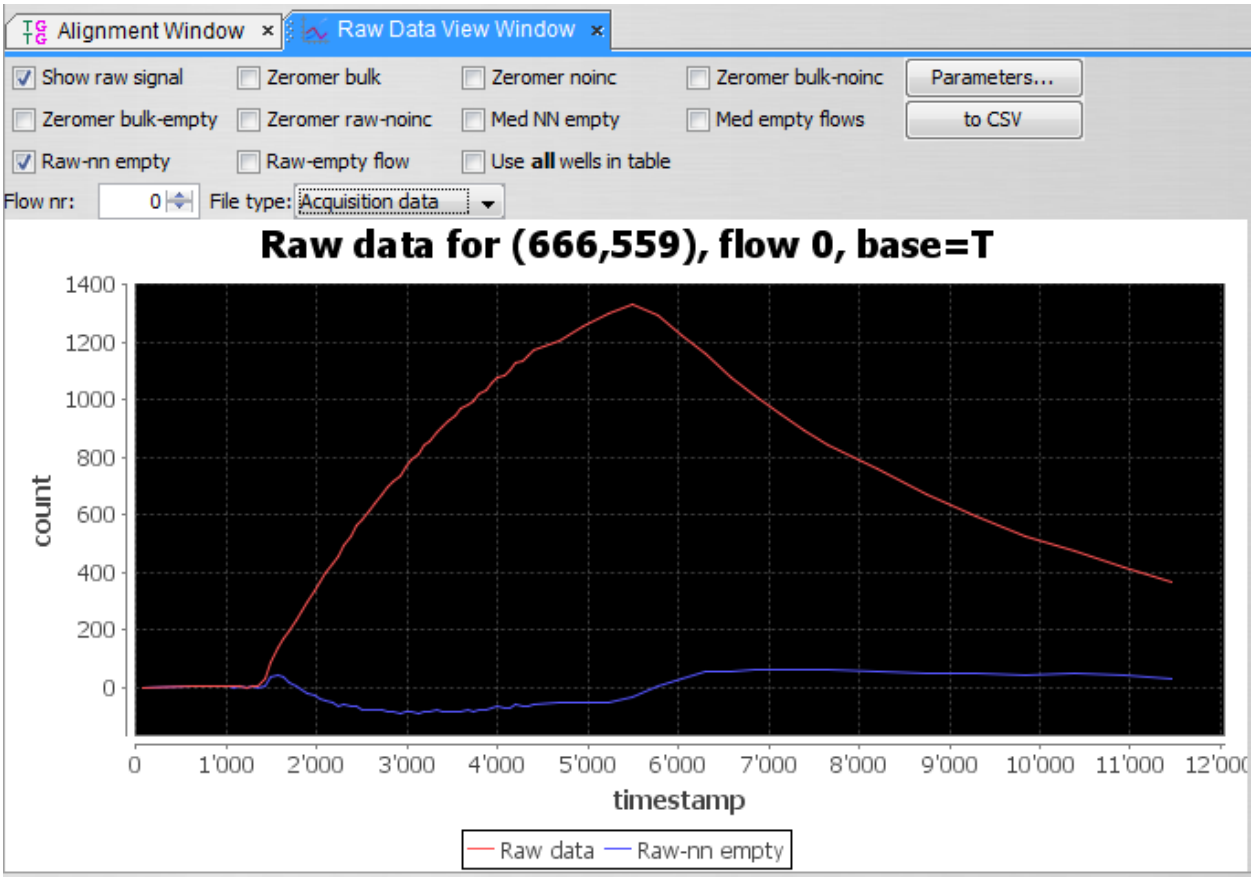
- In the menu Windows, select Raw Data View



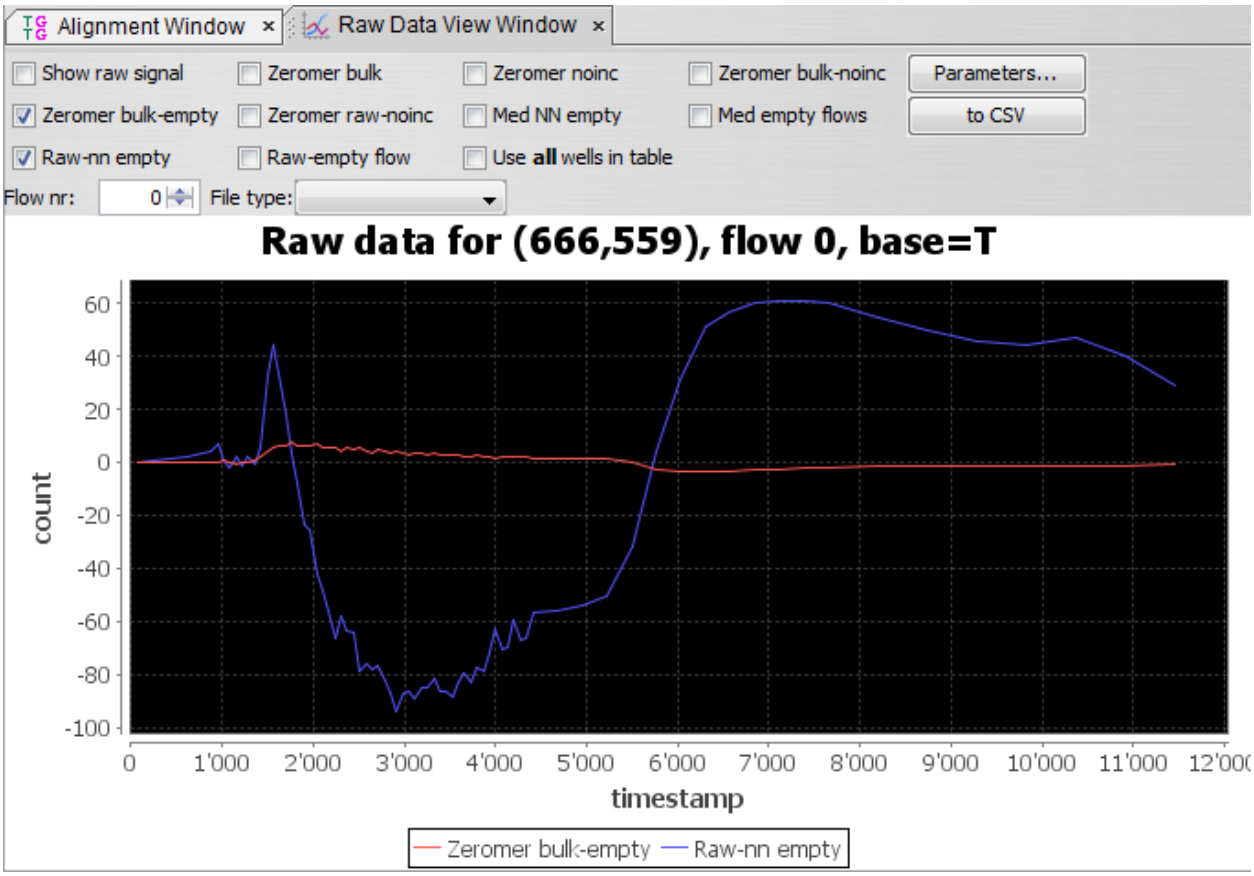
- In the dropdown box labeled “File type:”, select the type of raw data you wish you view (acquisition, prerun, beadfind pre or beadfind post). Let’s select acquisition data:



- Select a well If your choice and then you should see the raw, unprocessed signal (red curve). Not that by default, it only shows the raw signal, and does not compute any bg subtraction. To see the background signal subtracted of the neighboring wells, select “Raw-nn empty”:



- To see the background subtracted signal, **deselect** “Show raw signal”, and select both “Raw-nn empty” and Zeromer bulk-empty:

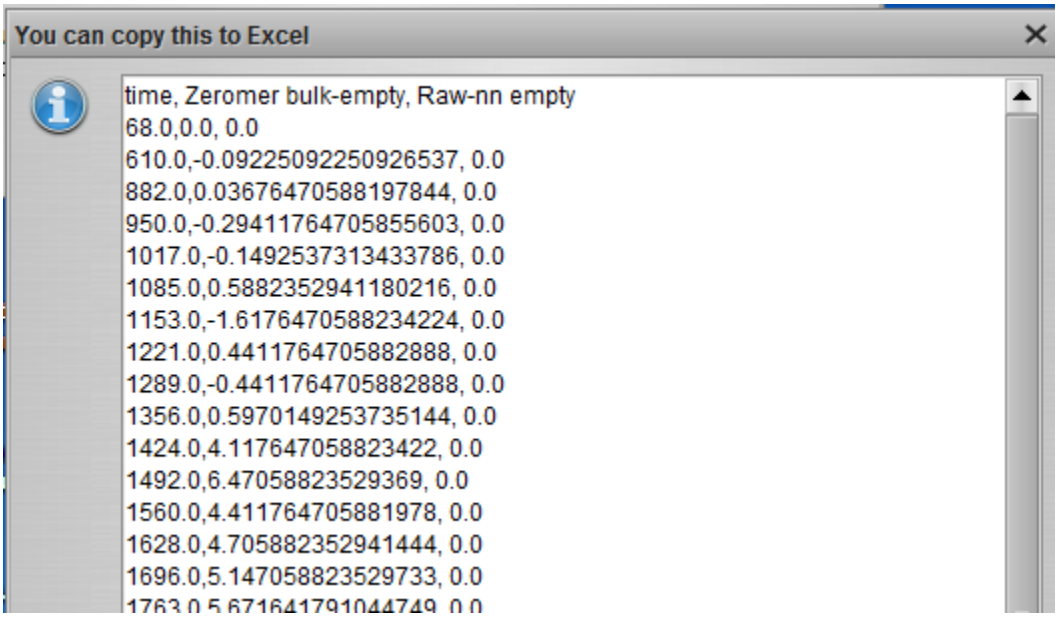


- The zeromer bulk-empty is the signal one would expect for a zeromer (red line). The blue line now shows the raw-nn subtraction. The difference then shows the actual signal. So the **blue spike** at the beginning is the actual incorporation signal .

Exporting the data to a file

- In the raw data viewer, click on “to CSV”

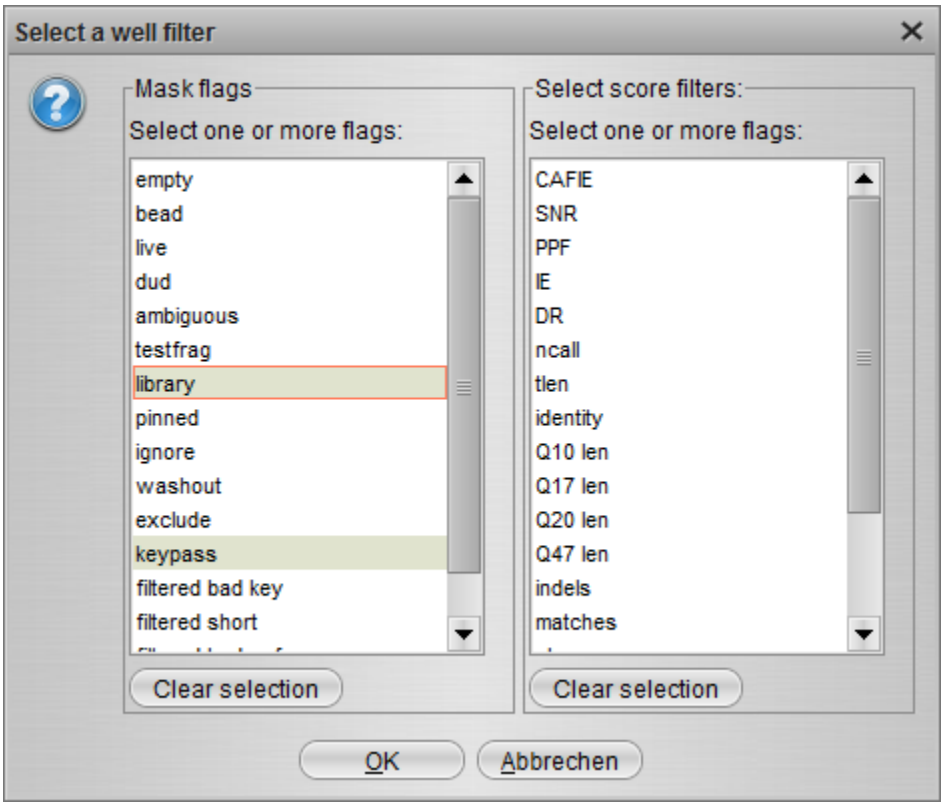
- I will show you the content which you can now select and copy/paste into Excel



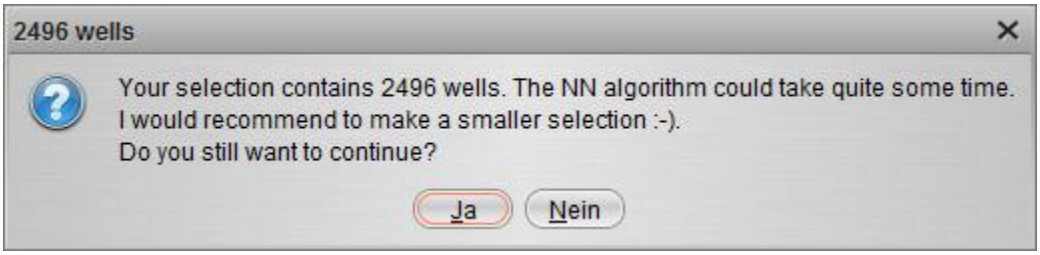
Background Subtraction For Chip Area

- If you want to compute the average signal of an entire area, and subtract the average background for this entire area, you can select “use all wells in table”.
- This will use all wells shown in the WellTable Window to use as “signal”, and it will use all ”empty” wells in the entire area for the background subtraction.
- Note that if you select a large area, this could take quite some time to compute!
- First, select an area of say 20x20 in the MaskView or via the HeatMap (or by entering the coordinates)
- Next, go to the WellTable Window and make sure the load scores is selected.
- Click on the filter icon.

- We only want to use signals for wells with the library and keypass flag. Feel free to filter further by any other parameters, such as identity score or Q17 len etc.



- Now the table only contains those wells as specified by the filter criteria.
- Now, in the raw data view windows, select "Use all wells in table"
- Depending on the size of the chip area, it will warn you about continuing:



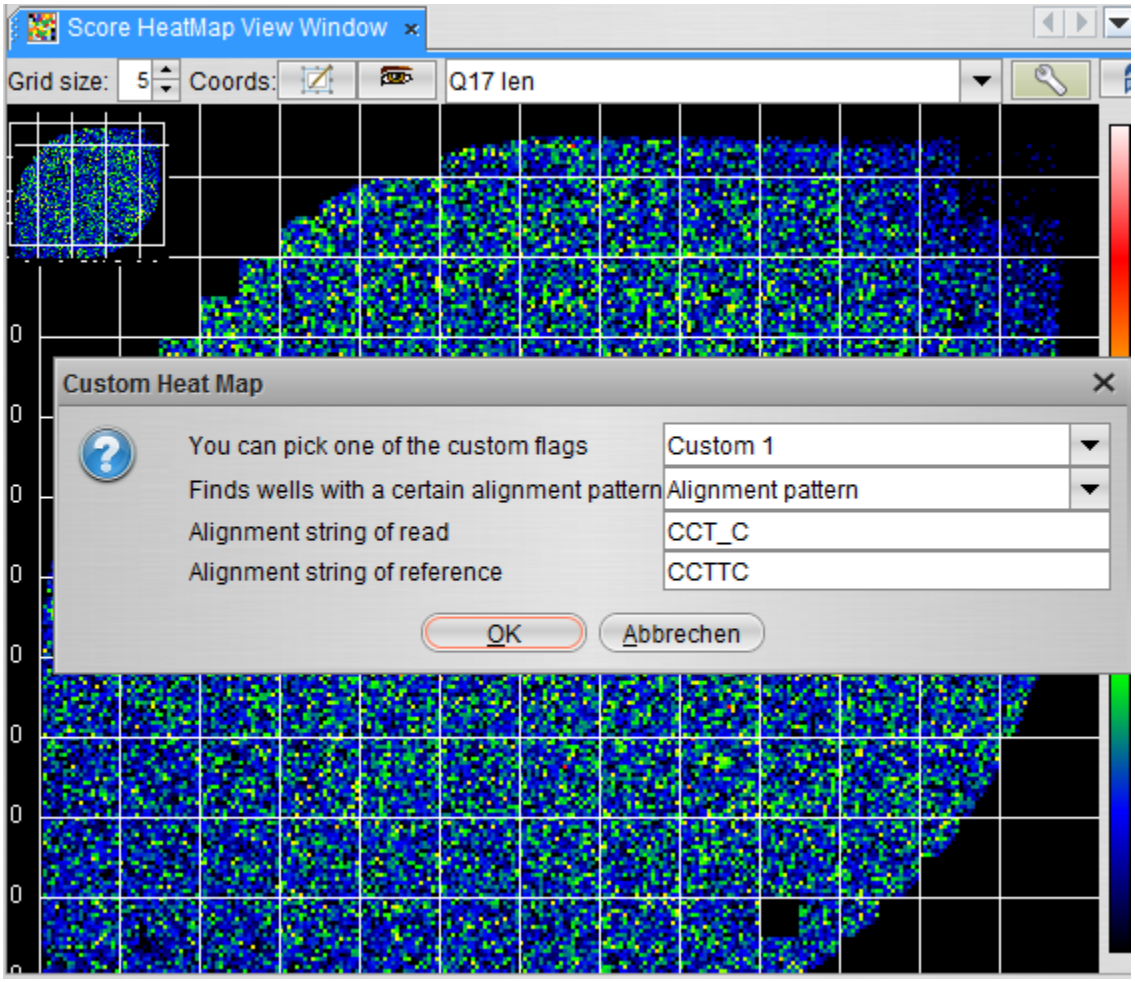
- If you select too many wells, it could take a long time to compute.

Score Heat Map Use Cases

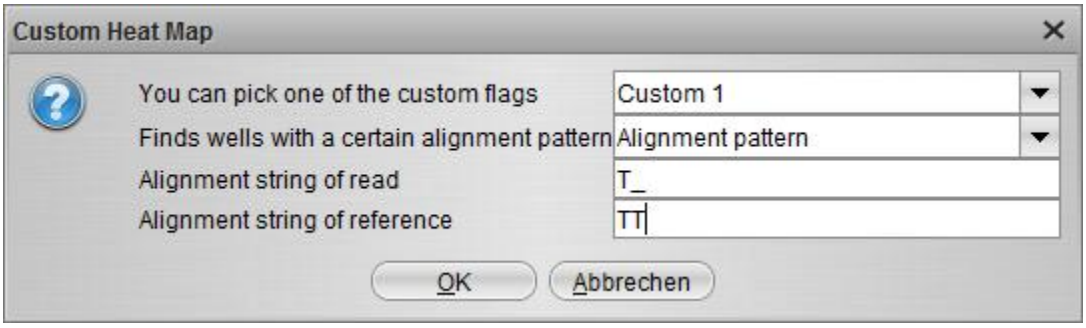
Finding Alignment Patterns on the Chip

- Select the Score Heat Map View Window (Menu Windows/Score Heat Map)

- Click on the little tool icon on the top, which opens the Custom Heat Map panel:

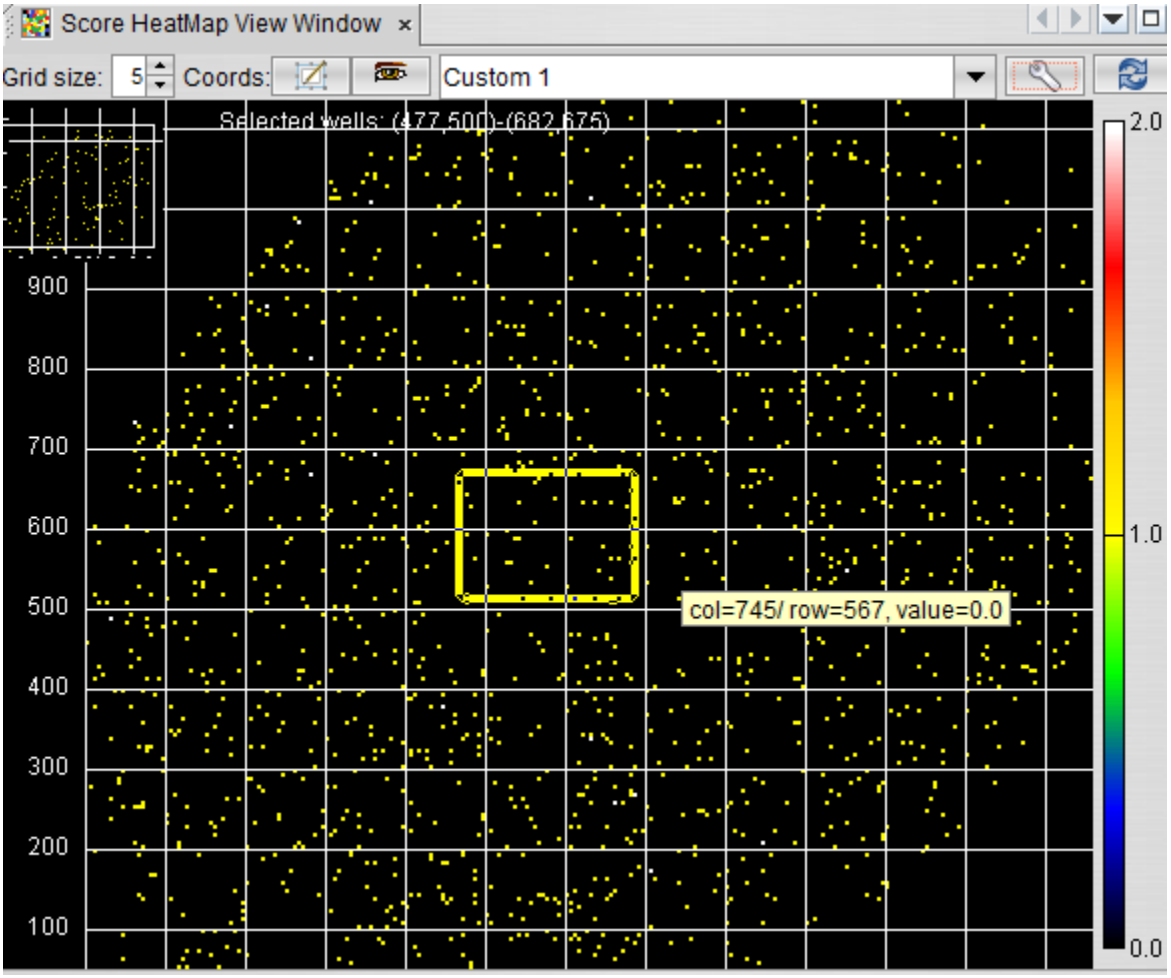


- Pick the alignment pattern in the drop down box
- Now enter an alignment pattern for the read (_ represent gaps)
- In our example, enter the following: T_ for the read, and TT for the reference

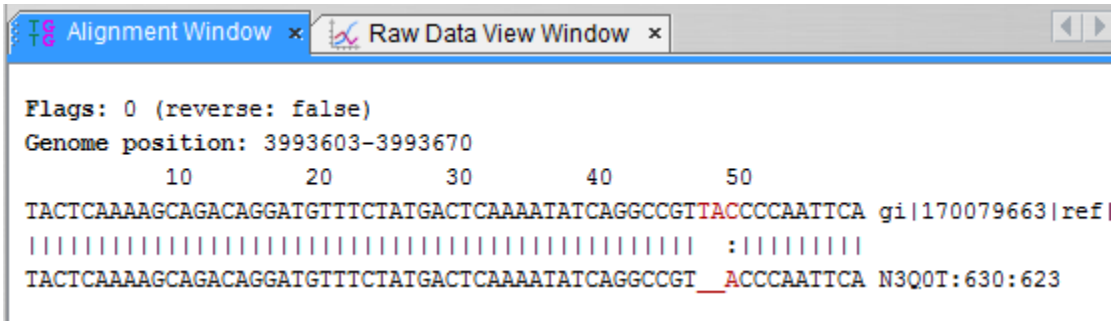


- When you click OK , it will now search all alignments on the entire chip for this alignment pattern. So this could take a few minutes

- You should now see something like this:



- Each pixel represents the location where this alignment pattern was found
- Select an area and and select a row in the WellTable window
- Open the Alignment Viewer and you will see that every alignment you select now contains this pattern above: (The base T followed by a gap in the read)



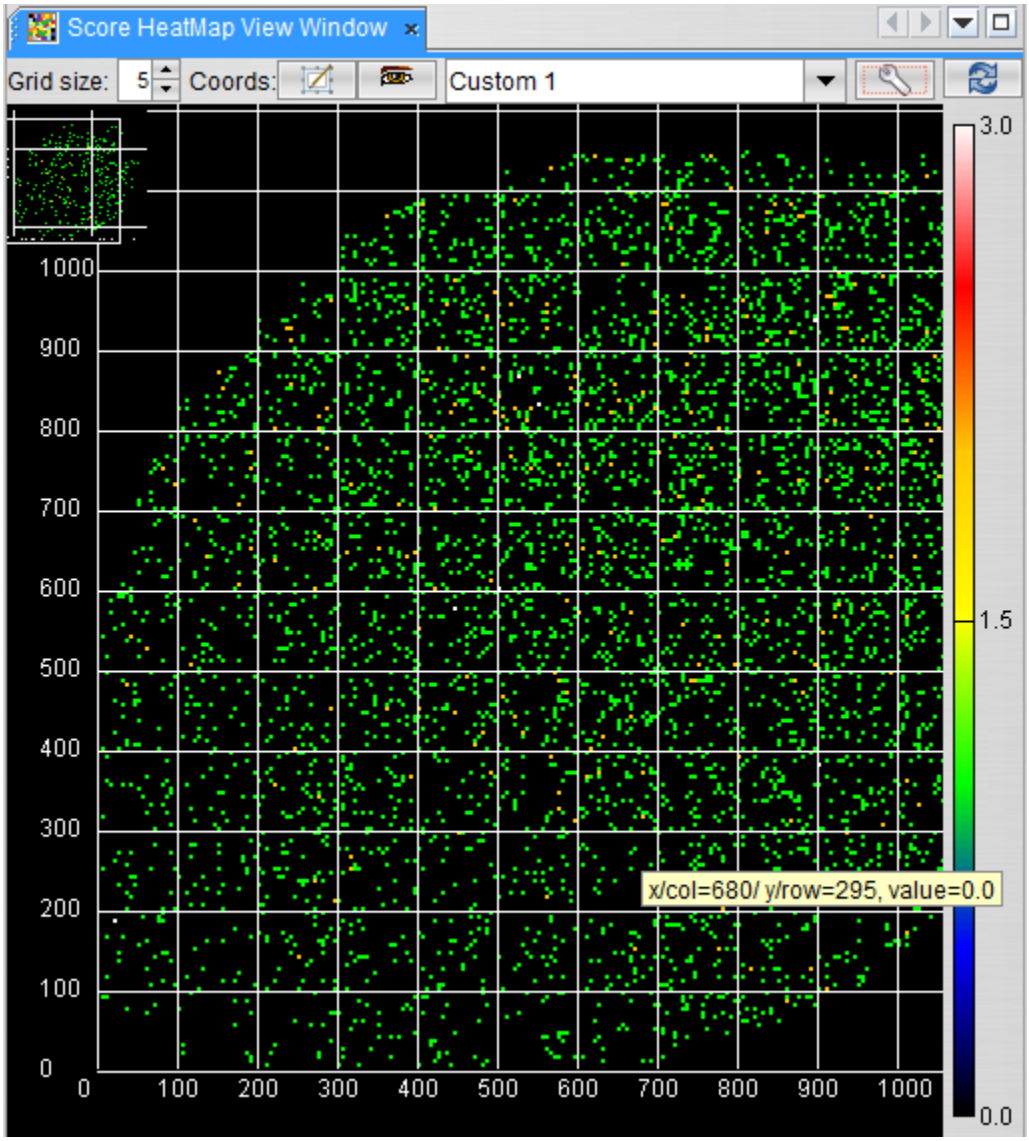
Finding "Perfect Reads"

- Similar to the use case above, open the Scores Heat Ma and go to the custom panel

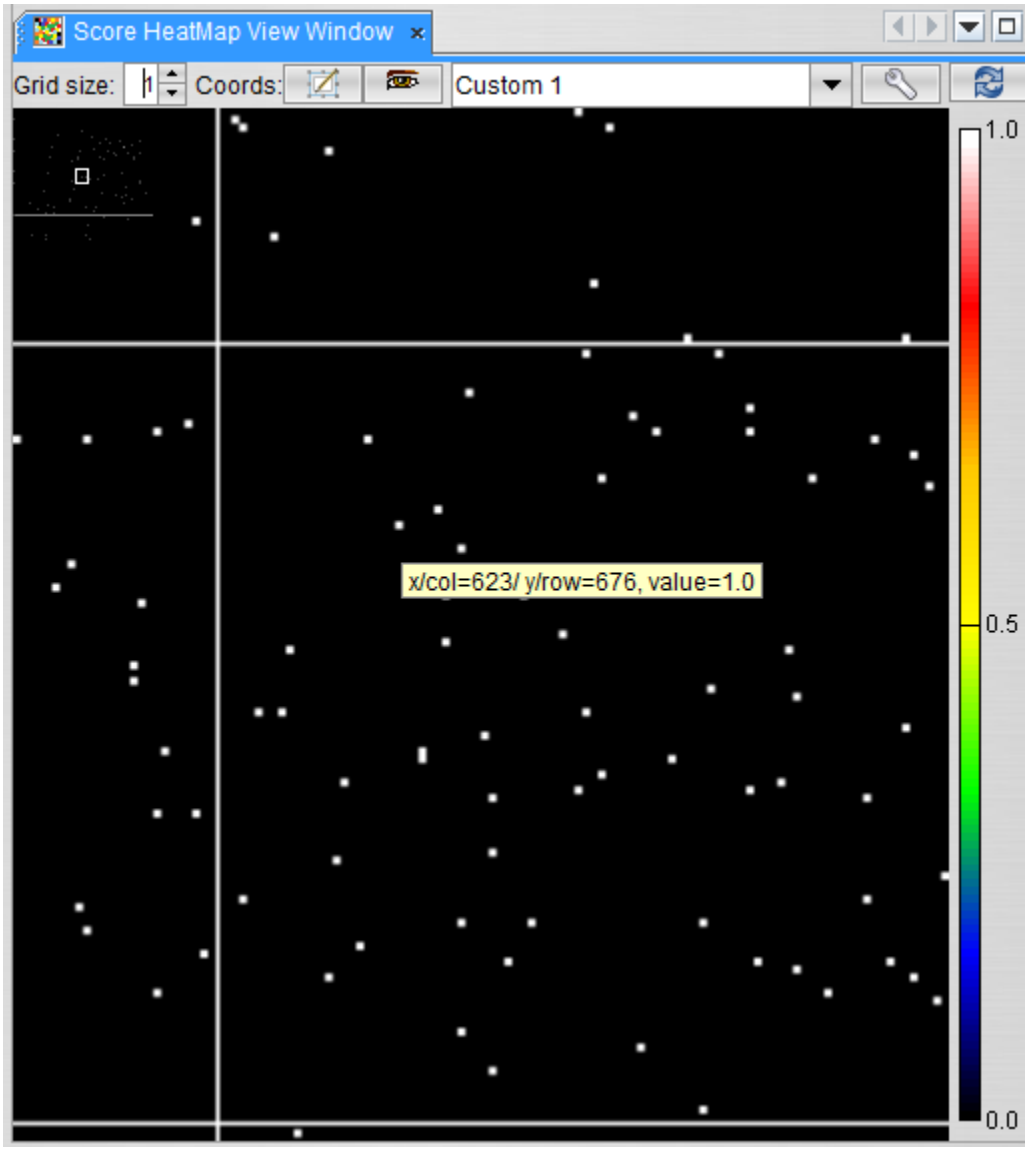
- Pick the “Perfect Read Pattern” from the drop down box:

- Let’s find all reads that have at least a stretch of 100 bp in a row without any errors (no mismatches and no indels)
- After you click ok you will see a message “reading BAM file…”which could take a minute or more depending on your chip size.

- The resulting heat map might look something like this:

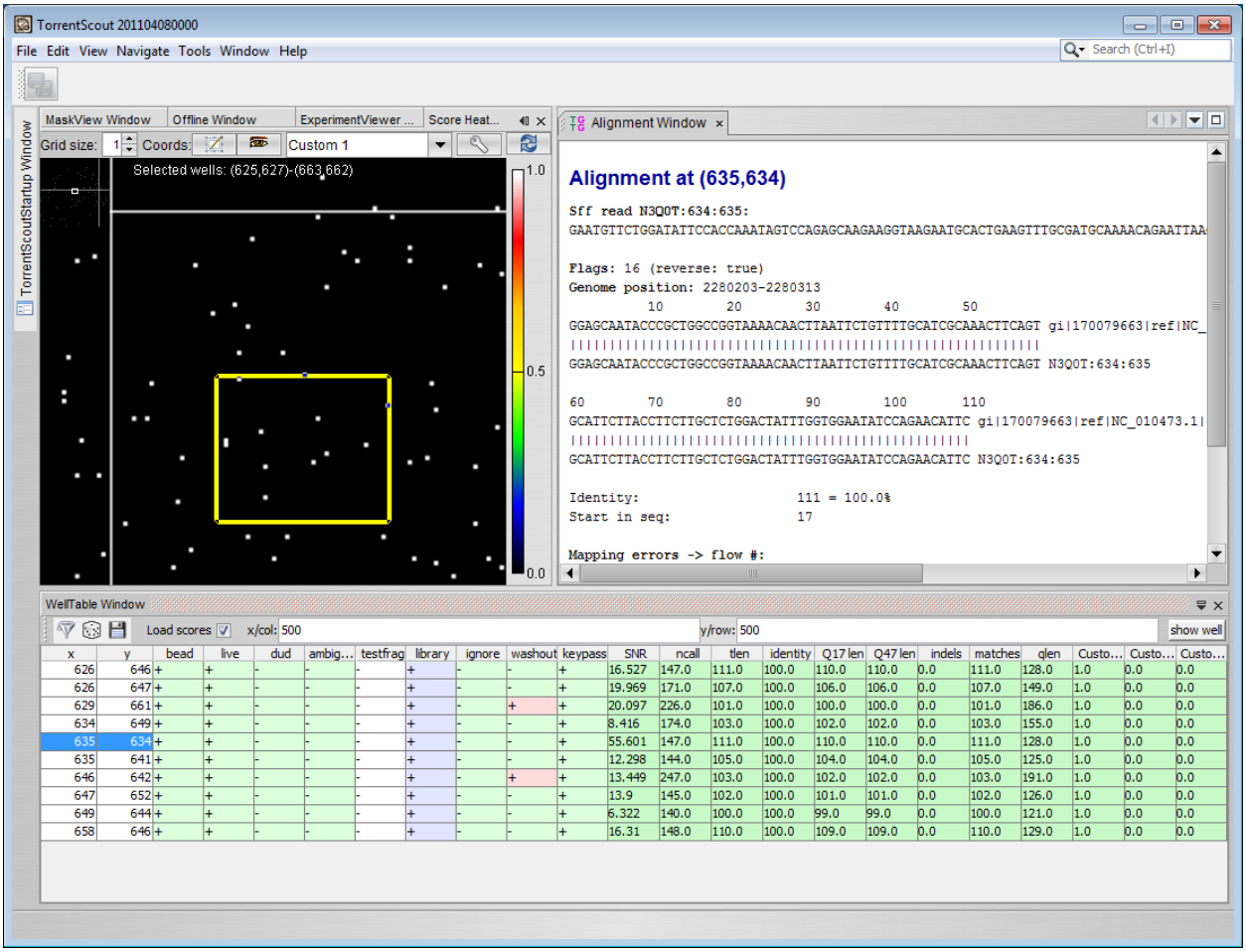


- Each pixel on the map is an area where a read with at least 100bp without errors was found. If you change the grid size to 1, then each pixel stands for one read:



- As before, you can select an area by drawing a rectangle, and the Well Table will now only show reads that satisfy that search criteria. So all reads in the table will have at least a 100bp perfect

stretch. Example:



x	y	bead	live	dud	ambig...	testfrag	library	ignore	washout	keypass	SNR	ncall	tlen	identity	Q17 len	Q47 len	indels	matches	qlen	Custo...	Custo...	Custo...
626	646	+	+	-	-	-	+	-	-	+	16.527	147.0	111.0	100.0	110.0	110.0	0.0	111.0	128.0	1.0	0.0	0.0
626	647	+	+	-	-	-	+	-	-	+	19.969	171.0	107.0	100.0	106.0	106.0	0.0	107.0	149.0	1.0	0.0	0.0
629	661	+	+	-	-	-	+	-	+	+	20.097	226.0	101.0	100.0	100.0	100.0	0.0	101.0	186.0	1.0	0.0	0.0
634	649	+	+	-	-	-	+	-	-	+	8.416	174.0	103.0	100.0	102.0	102.0	0.0	103.0	155.0	1.0	0.0	0.0
635	634	+	+	-	-	-	+	-	-	+	55.601	147.0	111.0	100.0	110.0	110.0	0.0	111.0	128.0	1.0	0.0	0.0
635	641	+	+	-	-	-	+	-	-	+	12.298	144.0	105.0	100.0	104.0	104.0	0.0	105.0	125.0	1.0	0.0	0.0
646	642	+	+	-	-	-	+	-	+	+	13.449	247.0	103.0	100.0	102.0	102.0	0.0	103.0	191.0	1.0	0.0	0.0
647	652	+	+	-	-	-	+	-	-	+	13.9	145.0	102.0	100.0	101.0	101.0	0.0	102.0	126.0	1.0	0.0	0.0
649	644	+	+	-	-	-	+	-	-	+	6.322	140.0	100.0	100.0	99.0	99.0	0.0	100.0	121.0	1.0	0.0	0.0
658	646	+	+	-	-	-	+	-	-	+	16.31	148.0	110.0	100.0	109.0	109.0	0.0	110.0	129.0	1.0	0.0	0.0

Finding reads with a certain subsequence