Single-cell RNA-seq data analysis in Chipster 9.2.2018

Maria Lehtivaara, Eija Korpelainen
chipster@csc.fi

In this hands-on session you will get familiar with Chipster's tools for Drop-seq based tools, which process data from raw reads (FASTQ files) to digital gene expression matrix (DGE), and Seurat based tools, which filter the DGE and cluster cells in order to find subpopulations.

Please note that the tools are still under development and there will most likely be changes. The Chipster team encourages you to give feedback (chipster@csc.fi), so that we can take your needs into account when developing the tools further.

PART 1: DropSeq data preprocessing

We will process raw Drop-seq data into an expression measurement for each gene in each individual cell ("digital expression matrix"). The data used in the exercises is originally the mouse retinal cell testing data from Dr.Seq tools: (http://www.tongji.edu.cn/~zhanglab/drseq/). The data is small to make things a bit easier and faster.

1. Open Chipster

Go to **chipster.csc.fi**, and **Launch Chipster.** Log in with the course credentials, or use your own.

2. Open example session

We have put the data for you on the Chipster server. Click **Open example session** in the Datasets panel (top left) and select the session **course_single_cell_RNAseq_DropSeq**.
In this example session we have imported two fastq.gz files: read1 with the barcodes and read2 with the actual RNA sequence.

3. Quality control with FastQC

Select both FASTQ files and tool **Quality control / Read quality with FastQC**, and click the **Run for each** button. Select one of the resulting html files and the visualization method **open in external web browser**. Repeat this for the other html file, and compare the FastQC reports.

How many reads are there and how long are they? How is the base quality?

4. Preprocessing DropSeq FASTQ files

Select **both FASTQ files**, and the tool **Single cell RNA-seq / Preprocessing DropSeq FASTQ files**. Click **Show parameters**, and check that the input files are correctly assigned, and that the base range for the cellular and molecular barcodes are correct. **Run** the tool.
This tool combines several steps for preprocessing DropSeq FASTQ files. While waiting, click the **More help** button, and read the manual page.

What is the unaligned BAM file for? What kind of trimming & filtering is performed?

When the 4 result files appear, select **drseq_read_1_unaligned.bam** and open it with **BAM viewer**.

Can you spot read1 and read2 sequences?

Open **tagging_and_trimming_summary.txt** as text.

How many cell barcodes passed the quality filtering? How many reads were filtered out because they were too short?

Open **tagging_and_trimming_histograms.pdf.**

Were there lots of adapters and polyA tails?

Select the new **drseq_read_1.fq.gz** file and run the tool **Quality control / Read quality with FastQC** again.

How many reads are there now and has the length distribution changed?

5. Alignment

Select the **drseq_read_1.fq.gz** file created in the previous step, and the tool **Alignment / HISAT2 for single end reads**. Set the **genome** to **Mus_musculus.GRCm38.90** and run the tool.
Check the **hisat.log** and **drseq_read_1.bam**.

What was the mapping percentage? Can you see the alignment positions for the reads? Is the BAM header different from that of the unaligned BAM?

6. Save the session as cloud session

Click **File** and **Save cloud session...** In the File name field type a name for the session and your name (for example *Maria_dropseq_course*) and click Save.

7. Merge the aligned BAM with the unaligned, tagged BAM

Now we have two BAM files: the unaligned one with the cell and molecular barcode tags, and the aligned BAM which lost the barcode information during the alignment. We want to combine the alignment and barcode information, so we merge these two files. This tool takes only the best alignment for each read from the aligned BAM.

Select **both BAM files**. Select the tool **Single cell RNA-seq / Merge aligned and unaligned BAM**, make sure the files are assigned correctly, and set **genome** to mouse again.

Examine the **drseq_read_1_merged.bam** file with **BAM viewer**.

Does the BAM now have both the alignment and the barcode information?

8. Add annotations (tag reads with gene names)

Next, we use **drseq_read_1_merged.bam** and run tool **Single cell RNA-seq / Tag reads with gene names** (remember to set the **genome** again!).

Examine the **merged_tagged.bam** file with **BAM viewer**.

Can you find a read with a GE field? Which gene is that read mapping to? Notice that there's also this XF-tag now -what info does it hold?

9. BONUS EXERCISE View BAM in genome browser

Select **merged_tagged.bam** and the corresponding **.bai file**. In the visualization window, select Genome browser. Choose the **Mus_musculus.GRCm38.90 (mm10)** as genome and click **Go**. Browse to gene **Rp1**. Zoom in.

Are the reads evenly distributed across the gene? Which transcript isoform seems to be present?

10. Estimate the number of usable cells -check the inflection point

There can be hundreds of thousands of cell barcodes in the BAM file, and we need to extract those that correspond the actual STAMPs (beads with a cell), as opposed to the "empties" (beads with just ambient RNA). To estimate the number of usable cells, we plot a cumulative distribution of reads in cells. We should see a "knee": left side of the knee shows the actual STAMPs, and the right side of the knee are the "empties".

Select the **merged_tagged.bam** and the tool **Single cell RNA-seq / Estimate number of usable cells.**

Check the results in **inflectionPoint.pdf**.

Can you see a "knee"? What is the number of usable cells (=the inflection point)? What fraction of the total reads do they contain?

11. Detect bead synthesis errors & create digital gene expression matrix

Sometimes problems occur in barcode generation. Beads get stuck in some phase and thus miss some synthesis cycles. To get rid of those problematic barcodes, you need to have an estimate for the number of cells in the sample (Number of barcodes = 2 x the expected number of cells). Here we use 2000. Finally, we generate a digital gene expression matrix. We use the information from the previous step to limit the number of cells and thereby the size of the matrix.

Select **merged_tagged.bam** and tool **Single cell RNA-seq / Create digital gene expression matrix.** Set the parameters **How to filter the DGE matrix = number of core barcodes** and **Filtering parameter = 500**.

Examine the summary file (**synthesis_stats_summary.txt**).

How many beads are there in total? How many had some sort of errors?

Examine the **synthesis_stats.txt** file.

How many molecules were detected for the first cell in the list? Are there some biases?

Check the **digital_expression_summary.txt** file.

How many genes and transcripts were there in the cell that had most genes?

Open the **digital_expression.tsv** file. (Click "Visualise", even though it will take some time to open the file.)

How many genes and cells are there? Why do you think the counts are so low? Are there any bigger counts (you can sort the table by clicking the column headers)?

PART 2: Seurat with 10X Genomics data

The 10X Genomics data used in this tutorial is originally the example data for Seurat tools:
http://satijalab.org/seurat/get_started.html
It's a dataset of 2700 peripheral blood mononuclear cells (PBMCs).
We have imported in Chipster a tar package that contains the three 10X Genomics output files.

1. Open example session

Click **Open example session** and select the session **course_single_cell_RNAseq_Seurat**.

2. Setup Seurat object & quality control

Select the **files.tar.gz.** Select tool **Single cell RNA-seq / Seurat -Setup and QC.** Check the parameters, and **name your project** (for example as "PBMC"). **Run** the tool.
Open the **QCplots.pdf** in **external browser**. Look at both pages.

> Based on the plots, what would be the optimal upper limit for the number of genes and mitochondrial transcript percentage? Hint: check the default parameters used in the next tool.

3. Filtering, regression and detection of variable genes.

Select **seurat_obj.Robj** (this is an R-object, which can be exported and opened in R, or just passed to the next tool in Chipster, like we do now). Select the tool **Single cell RNA-seq / Seurat - Filtering, regression and detection of variable genes.** Check if the default parameters are good for this dataset, based on the QCplots? While the tool is running, click the **More help** button, and learn about the three steps this tool performs.
Once the tool is done, open the **Dispersion.pdf** and check also the second page.

> How are the cutoffs working for this data? How many variable genes are there?

4. Principal component analysis

Select **seurat_obj.Robj** from the previous step and run the tool **Single cell RNA-seq / Seurat -PCA.**
Open **PCAplots.pdf** in **external browser**. Look at the heatmaps and the standard deviation of PCs in the last two pages.

> How many principal components should we continue the analysis with (check the elbow in the standard deviation plot, inspect the heatmaps)? Would 10 be ok?

5. Clustering

Select **seurat_obj.Robj**. Select tool **Single cell RNA-seq / Seurat -Clustering**. In the parameters, set **Number of principal components to use** =**10**.
While waiting for the tool to run, you can study the manual (click the **More help** button).

> What are the three main steps of this tool?

When the results are ready, study the **tSNEplot.pdf** (open in external browser).

> How many clusters are there in this data?

6. Markers for a specific cluster

Open **markers.tsv** as a **spreasheet**. Notice that we have all the markers for different clusters in one table.
Now, let's choose only markers for cluster 2. Choose **markers.tsv** and the tool **Utilities / Filter table by column value.** Fill in the parameters accordingly:

> **Column to filter by = cluster**
> **Does the first column have a title = no**
> **Cutoff = 2**
> **Filtering criteria = equal-to.**
> How many biomarkers were recognized for cluster 2?

7. Visualize markers

Choose **seurat_obj.Robj** generated in step 5. Select tool **Single cell RNA-seq / Seurat -Visualize markers**. Type a marker **gene name** to the parameter field (choose one of your own, or try for example with MS4A1, LYZ and PF4). You can run the tool several times for different genes.
Open a **biomarker_plot.pdf** in external browser.

> Is your gene a good marker for that cluster?

BONUS: Try the Seurat tools for the DropSeq data used in Part 1. You can also analyze your own data.

When you start your own analysis: Tips on managing your data and sessions

If your data is really large, the tools can take quite a long time to run. You don't however have to try keep the Chipster client alive, IF you save the session **as a cloud session** after each step. This way you can close the client, and when you open the client and the saved cloud session next time, the output files will appear there when the tool is done. Note, however, that the cloud sessions are not for permanent storage: save your session also locally at some point.

The CSC's Taito can be useful when dealing with large files: store your files in the **$WRKDIR directory**. Note that this directory is not back-upped and is cleaned regularly, and anything older than 90 days is removed. If you want to store your files more permanently and still use them, you can place them in $WRKDIR/DONOTREMOVE directory. For more permanent storage option, take a look at CSC's **HPC archive** and **IDA services**. More info regarding these can be found here: https://research.csc.fi/data-environment

You can use **SUI** (Scientist user interface, https://sui.csc.fi/) to upload and download your files, and **NoMachine** connection to run Chipster in Taito -check out the video tutorial for that in Chipster Youtube channel!

More info regarding data transfer:
https://research.csc.fi/csc-guide-moving-data-between-csc-and-local-environment
NoMachine manual:
https://research.csc.fi/-/nomachine

How to deal with large files:
1) Upload them to your $WRKDIR in Taito (using SUI)
2) Open NoMachine connection, and launch Chipster in Taito-shell (check out the video tutorial for this)
3) Import files to Chipster from your $WRKDIR
4) Save the session with imported files as a cloud session
5) Close the Chipster on NoMachine
6) Now you can open Chipster client on your own computer and continue the analysis
7) Remember to save the cloud session every now and then -always when in doubt that the client won't stay alive!
8) When you are done, save the session also locally in your $WRKDIR or HPC archive -for this you need to open the cloud session on NoMachine+Chipster again, and then save it locally there.

FAQs:

1) I have several pairs of FASTQ files per sample, how do I combine them?
If you have several FASTQ files per sample (from NextSeq you will get 4 FASTQs for read1 and 4 for read2), you can merge them together with the tool **Utilities / Merge FASTQ**. You need to run the tool twice: once for read1 files and once for read2 files. Make sure your paired files are named similarly, so that the alphabetical ordering will work and the pairing information is maintained.

2) My read1 sequences are not 20bp, what should I do?
If you see in the FastQC report that your read1 are of some other length than 20bp, the Tag BAM tool won't work. To get rid of shorter barcode reads, select **both FASTQ files**, and run tool **Preprocessing / Trim reads with Trimmomatic** with parameter **Minimum length of reads to keep = 20**. Continue analysis with the files that are named "--trimmed.fq.gz" (instead of the "--unpaired_trimmed.fq.gz" ones). If there are reads that are longer than 20bp in read1 files, you can trim those from **read1** file after this step by running Trimmomatic with parameter **Number of bases to keep from the start = 20**

3) I would like to import a large tar package of files, but the session looks like a spider net.
You can select **File / Import from / URL directly to server**. You can check what files the tar package contains using the tool **Utilities / List contents of a tar file**. You can selectively extract the files you want with **Utilities / Extract .tar or .tar.gz file**.