

Knowledge_Progression_handling_RNA-seq_datasets

Aims

I want to see if I can replicate results published on the basis of a bulk RNA-seq dataset in terms of:

- differential gene expression (DGE) analysis
- GO term analysis on differentially expressed genes
- Functional enrichment if possible
- enrichment in disease-related genes or SNPs
- gene network analysis

Finding a suitable study

I searched for clinically relevant, loosely cerebral cortex development-related RNA-seq datasets with attached publication. I selected only datasets uploaded last year and whose publications don't do highly in-depth analyses, as I believe that me working on this data will bring the biggest contribution to expanding our knowledge horizon. As to why I selected datasets that were associated with a publication: this was for me to be able to cross-check the results of my analyses and make sure I am performing my analysis properly.

I found this study: [GSE167193](#) on GEO at NCBI. This study analyzed a potential connection between a pregnancy disorder known as pre-eclampsia and autism spectrum disorders. Pre-eclampsia is a common and dangerous increase in maternal blood pressure that can occur after the 20th week of pregnancy. One in 20 pregnant people are affected by pre-eclampsia, which has a severe negative impact on both the pregnant person and the fetus. There is no treatment known to date, except for the delivery of the placenta, and very little is known about the long-term consequences on the development of the child's brain.

The authors used a pre-eclampsia mouse model in which the disorder was induced in pregnant mice by treating them with a compound called L-NAME. The mouse offspring from mothers that were exposed to L-NAME behaved in ways reminiscent of ASD and scored accordingly on several behavioral tests that are thought to assess metrics related to the condition.

To better understand the relationship between the two conditions, the authors sequenced RNAs from the cortices of mouse embryos at day E 17.5, two days before birth, from mother animals that had or did not have elevated blood pressure, and similarly from the hippocampi of adult offspring. The authors generated cDNA libraries from the cortical RNAs of control and experimental condition embryonic cortices using a kit for stranded mRNA detection. This means that the kit enriches for poly-A-tailed mRNAs and also captures information regarding which genomic DNA strand the mRNAs were transcribed from. This is an excellent fit for my purposes, as I am, for the time being, interested in coding transcripts. Then, the cDNA libraries were fragmented, and the fragments were sequenced on Illumina Hiseq X machines, a process that generated millions of paired-end 150 bp reads (21 million, to be a little more precise [link to post where I open the FastQC files]).

I focused my investigation on the embryonic cortex data, as this is a system I am more familiar with from my own research work.

```
iweber@Lenovo-Ioana:~/Datasets/Pre_eclampsia_mice$ vdb-validate SRR13761520
2024-03-11T18:10:23 vdb-validate.3.1.0 info: Database 'SRR13761520' metadata: md5 ok
2024-03-11T18:10:23 vdb-validate.3.1.0 info: Table 'SEQUENCE' metadata: md5 ok
2024-03-11T18:10:23 vdb-validate.3.1.0 info: Column 'ALTREAD': checksums ok
2024-03-11T18:10:24 vdb-validate.3.1.0 info: Column 'QUALITY': checksums ok
2024-03-11T18:10:25 vdb-validate.3.1.0 info: Column 'READ': checksums ok
2024-03-11T18:10:25 vdb-validate.3.1.0 info: Database 'SRR13761520' contains only unaligned reads
2024-03-11T18:10:25 vdb-validate.3.1.0 info: Database 'SRR13761520' is consistent
iweber@Lenovo-Ioana:~/Datasets/Pre_eclampsia_mice$
```

I downloaded the datasets directly from the entry of this study on GEO in SRA format. As I found out, SRA is an NCBI-specific way to compress FastQ files. These are the ones with the raw reads from the sequencers and I learned that these are precisely what I will need for performing differential sequence analysis. The processing of the files to any other format, such as SAM or BAM, or anything else except for basic quality control, may introduce biases that affect subsequent data analyses.

So what is RNA-seq anyway? And what's the deal with alternative splicing analysis?

🔗 To read:

General info on RNA-Seq analysis <https://academic.oup.com/bib/article/23/2/bbab563/6514404?login=false>
general info RNA-seq, inc long read (third-gen) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10043755/>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4712774/> tximport solves issues with artificially inflated gene counts derived from transcript isoforms. It essentially bundles the counts obtained by Salmon for individual transcripts into counts for the gene that those transcripts were

transcribed from. For this, it needs a so-called tx2gene.tsv table that indicates the relationships between all of the possible transcripts from a genome and the genes from that genome.

Downloading genomes

I knew the next step would be to map the reads to the source genome, and for that I, of course, needed the mouse genome.

I found [here](#) that NCBI has two command line tools for Linux that ease the download of datasets. From within the NGS environment and in a new folder "Genomes", I used `curl` to get the installers as shown on the NCBI page, and then made the binaries executable as instructed.

I then proceeded to download the genomes that are relevant to me (mouse and human) using the option to find them by accession number. I got the accessions for the latest assemblies [here](#) and [here](#). To be on the safe side, I decided to use the more elaborately curated RefSeq assemblies, so the commands looked as follows:

```
(NGS) iweber@iweber-VirtualBox:~/Documents/Genomes$ datasets download genome accession GCF_000001405.40 --filename genome_H_sapiens_GRCh38.zip
```

...and got a big, fat load of nothing xD, because the shell said it can't find the command "datasets". Instead of fumbling to find where the binaries associated with these two tools are, I decided to simply use conda to install them directly into this environment.

```
conda create -n ncbi_datasets conda activate ncbi_datasets conda install -c conda-forge ncbi-datasets-cli
```

```
(NGS) iweber@iweber-VirtualBox:~/Documents/Genomes$ conda activate ncbi_datasets  
(ncbi_datasets) iweber@iweber-VirtualBox:~/Documents/Genomes$ conda install -c conda-forge ncbi-datasets-cli  
Channels:
```

- conda-forge
- bioconda
- defaults

Platform: linux-64

Collecting package metadata (repodata.json): done

Solving environment: done

```
## Package Plan ##
```

```
environment location: /home/iweber/anaconda3/envs/ncbi_datasets
```

```
added / updated specs:  
- ncbi-datasets-cli
```

The following packages will be downloaded:

package	build	
ncbi-datasets-cli-16.14.0	ha770c72_2	15.8 MB conda-forge
	Total:	15.8 MB

The following NEW packages will be INSTALLED:

```
ca-certificates    conda-forge/linux-64::ca-certificates-2024.2.2-hbcc054_0  
ncbi-datasets-cli  conda-forge/linux-64::ncbi-datasets-cli-16.14.0-ha770c72_2
```

Proceed ([y]/n)? y

Downloading and Extracting Packages:

```
Preparing transaction: done  
Verifying transaction: done  
Executing transaction: done  
(ncbi_datasets) iweber@iweber-VirtualBox:~/Documents/Genomes$
```

In the Windows Subsystem for Linux (WSL), getting the SRA Toolkit and SRR run files from the preeclampsia study

Using the documentation at NCBI, I found that one needs a suite of programs called SRA Toolkit in order to losslessly convert the SRA files into FastQ files. I installed the toolkit using the instructions on the GitHub page of the NCBI, <https://github.com/ncbi/sra-tools/wiki/>. I decided to do so under Ubuntu running on the Windows Subsystem for Linux (WSL). I have read that, for optimal memory usage, it is preferable to create a virtual machine running a Linux distribution rather than the WSL. However, given that this was my first shot at analyzing a relatively small SRA dataset (~2.8 GB), I concluded that this is good enough for starters.

I downloaded the SRA toolkit as a .tar.gz archive for Ubuntu from one of the FTP servers of the NCBI:

```
wget --output-document sratoolkit.tar.gz https://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/current/sratoolkit.current-ubuntu64.tar.gz
```

then extracted it in one of my Ubuntu folders with the tar utility (included in Ubuntu),

```
tar -vxzf sratoolkit.tar.gz
```

and could afterwards access its tools over the command line.

As per the instructions of the GitHub page, I modified the PATH variable in my system. This essentially tells Ubuntu that there is a shortcut to the folders where SRA Toolkit has its binaries (bin directory) so that I can then easily access the individual tools and commands from the toolkit via the command line without needing to constantly change directories to that in which I extracted the SRA toolkit.

```
export PATH=$PATH:$PWD/sratoolkit.3.1.0-ubuntu64/bin
```

This should tell the shell that, when looking for binaries (tools, programs, etc that are executable), it should also look in the respective folder of the SRA Toolkit.

To verify whether the binaries can be found by the shell, I used

```
fastq-dump --stdout -X 2 SRR390728
```

...which worked as expected with returning the path to the directory of binaries precisely until I restarted the terminal :) I then found out that "export" only does a temporary modification of the PATH environment variable, which isn't kept in the global namespace. So, after advice from Christoph, I found out I need to only add the exact same line, minus "export", into the .profile file, which should save any customized variants of the PATH variable. And now I have the SRA Toolkit set up to work from any directory, and even after restarting the shell.

Additionally, I created a handy alias for changing the directory to where my datasets are stored by adding the following under "Some more aliases" in the .bashrc file:

```
alias mydatasets='cd /home/iweber/Datasets'
```

So now I can go straight to that folder with only typing "myd" and hitting tab.

I also undertook the Toolkit Configuration as directed by the GitHub page of SRA toolkit using `vdb-dump -i .double check command`. One noteworthy difference to how the configuration window looks like for me vs the website is that I do not have, under Main, the frame with options regarding to Workspace name and Workspace location.

I wonder if this is a matter of working on WSL instead of a full Linux distro?

However, in the tab Cache, I did set the location of the user-repository and process-local to be my Datasets folder, so that I may always find my files easily. I also created a subdirectory for my current study so that my data stays organized.

WSL - Extracting FastQ files from the SRR archives

And now that the tools are set up, I went on to try and extract the FastQ files from my dataset in SRA format. In my Datasets directory, I used the fasterq-dump command together with my dataset as a command-line argument:

```
fasterq-dump SRR13761520
```

And waited. I was afraid nothing is happening and I just went into some infinite loop, but then I navigated in Windows to the Datasets folder in WSL and saw that a temporary directory had been created. Since the terminal also didn't show me a fresh command prompt, I assumed it was just working intensely in the background to convert the data. Also, my ventilators were going crazy. s, when I checked the Task Manager, I saw that nearly 60% of my memory was in use by a process called "VmMem". Sure enough, a brief web search uncovered that this happens precisely when using WSL, and is the Windows process responsible for the virtual memory management [Double check!](#). All of this was happening before switching my RAM chips from a total of 32 GB to a total of 128 GB :) Yes, I am aware that I will need to set up a few more things, such as hyper-threading, for the new RAM to work as effectively as possible, but I was reluctant to do this at first, as my first experience after getting my laptop had been it crashing and refusing to start Windows upon installing a VirtualBox virtual machine and tinkering with the virtualization settings).

I had read on the GitHub page of the SRA Toolkit that the extraction of FastQ files requires a RAM size up to as much as 10 times the size of the original SRA file, at least during the extraction process (scratch space - had run into issues with this with Photoshop before, which kept complaining that the scratch disks were full and it therefore couldn't save whatever new artsy monster file I had created). So I was expecting Task Manager to tell me that around 25 of my 32 GB of RAM (~78% of it) were in use by the VmMem process. The process started out at around 50%. Half an hour of observation later, I realized that it kept increasing over time. I did see it go up to a max of 75.6% with a fairly steady increase over time, so I assume the Toolkit kept adding info to the scratch disks as the process progressed. Looking in the Resource Monitor of Windows, vmmem said it was using a Commit of 10,470,000 and a Working set of about the same size. I suspect Task Manager is not as accurate as the Resource Monitor ?

To check the progress of the unpacking, I peeked at the size of the temporary folder in the SRA folder of this study in Datasets. When I first looked at it, some 3-4 minutes after starting the extraction tool, it was at around 3.8 GB. Some 10 minutes after starting the extraction, it was at 6.5 GB.

At this point, I started wondering where exactly WSL was storing its files in a physical sense (I know, I know, should've done so sooner, etc). It being a subsystem of Windows, I strongly suspected it would also be on my Windows drive, which, unfortunately, is the smaller of my two SSDs, at 250 GB. And indeed, when checking the properties of my Windows drive, I realized that the free space on this drive was decreasing. Being fairly close to the size limit of the drive (~10 GB free), I was concerned that this might completely kill Windows if working on such a full drive or result in a corrupted, incomplete extraction of the FastQ files. So I started uninstalling any programs I saw were using a lot of space with their files and freed up some 10 more GB of RAM.

I also noticed that the Toolkit had also created another folder under Datasets called "sra", with a .sra.cache file of a only slightly larger size as the original SRA dataset (2,829,033 KB for the dataset and 2,829,054 KB for the .sra.cache file). This did not change in time.

It took around one hour for the extraction to finish, and I then got this:

```
iweber@Lenovo-Ioana:~/Datasets/Pre_eclampsia_mice$ fasterq-dump SRR13761520
spots read      : 21,192,945
reads read      : 42,385,890
reads written   : 42,385,890
iweber@Lenovo-Ioana:~/Datasets/Pre_eclampsia_mice$
```

And bam! I had two FastQ files in my study directory, each at exactly 7,954,345 Kb, so 7.9 GB and a total of around 16 GB. All in all, that's around 5.7x larger than the original SRA file.

But why were there two files with the same accession number but with _1_ or _2_ appended to the file name? The answer was fairly easy to find: these were reads from paired-end sequencing runs, as indicated on the NCBI website for this [dataset](#) and the paper's methods. Sequencing both ends of the cDNA fragments resulting from the cortex RNAs allows for more precise alignment [source] and is thus more useful especially in detecting alternative splicing isoforms, which may be of interest to me later.

I opened the two behemoth files by ~~torturing~~ convincing Notepad++ to do so, and, at first glance, they looked identical.

After this step, I also realized I had severely underestimated exactly how many files I'd have to download and extract, and to what space requirements that would amount to :teary-smile:

So, the first thing I did was to...get a larger SSD for my OS. I initially had a 256 GB NVMe and switched to a 4 TB instead.

Temporary fix for space issues: change location where WSL stores its files

But, for the time being: how could I change where Windows saves the files of the WSL? Thank goodness that I'm not yet at a level at which my questions haven't yet been asked by anybody on the Internet. I found this set of instructions by [NotTheDr01ds](#) on SuperUser:

<https://superuser.com/questions/1736576/change-the-storage-location-of-a-wsl2> I decided to execute them on the basis that they also create a backup image of Ubuntu on the WSL as I had it set up before, that the code was clearly commented and easy to understand in terms of what environment variables were being modified how, and that the user seemed to have taken extra time to spell out precautions and build in safety checks. This is what I ran:

```

PS C:\Users\Ioana> wsl -l -v
  NAME      STATE      VERSION
* Ubuntu    Stopped     2
PS C:\Users\Ioana> $WSL_ROOT="D:\WSL"
PS C:\Users\Ioana> $WSL_IMAGE_PATH="${WSL_ROOT}\images"
PS C:\Users\Ioana> $WSL_OLD_DISTRO_NAME="Ubuntu"
PS C:\Users\Ioana> Get-ChildItem HKCU:\Software\Microsoft\Windows\CurrentVersion\Lxss\ |
>>   ForEach-Object {
>>     (Get-ItemProperty $_.PSPATH) | Select-Object DistributionName,BasePath
>>   }

DistributionName BasePath
-----
Ubuntu          C:\Users\Ioana\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu_79rhkp1fndgsc\LocalState

PS C:\Users\Ioana> $WSL_NEW_DISTRO_NAME="Ubuntu_WSL2"
PS C:\Users\Ioana> $WSL_INSTANCE_PATH="${WSL_ROOT}\instances\$WSL_NEW_DISTRO_NAME"
PS C:\Users\Ioana> mkdir $WSL_IMAGE_PATH

  Directory: D:\WSL

Mode                LastWriteTime        Length Name
----                -----          ----- 
d-----        3/14/2024  9:35 PM            images

PS C:\Users\Ioana> mkdir $WSL_INSTANCE_PATH

  Directory: D:\WSL\instances

Mode                LastWriteTime        Length Name
----                -----          ----- 
d-----        3/14/2024  9:35 PM            Ubuntu_WSL2

PS C:\Users\Ioana> wsl --shutdown
PS C:\Users\Ioana> wsl --export $WSL_OLD_DISTRO_NAME "${WSL_IMAGE_PATH}\${WSL_OLD_DISTRO_NAME}.backup.tar"
Export in progress, this may take a few minutes.
The operation completed successfully.
PS C:\Users\Ioana> wsl --import $WSL_NEW_DISTRO_NAME $WSL_INSTANCE_PATH "${WSL_IMAGE_PATH}\${WSL_OLD_DISTRO_NAME}.backup.tar" --version 2
Import in progress, this may take a few minutes.
The operation completed successfully.
PS C:\Users\Ioana> wsl ~ -d $WSL_NEW_DISTRO_NAME
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@Lenovo-Ioana:~# sudo -e /etc/wsl.conf
root@Lenovo-Ioana:~# wsl --shutdown
Command 'wsl' not found, but can be installed with:
apt install wsl
root@Lenovo-Ioana:~# exit
logout
PS C:\Users\Ioana> wsl --terminate $WSL_NEW_DISTRO_NAME
The operation completed successfully.
PS C:\Users\Ioana> wsl --set-default $WSL_NEW_DISTRO_NAME
The operation completed successfully.

```

And it seems to have worked. Also, this was my first time seeing both Windows **and** Linux command lines mixed in the same window and, as the beginner that I am, it blew my mind. I realized that, basically, all I'm doing when starting the Ubuntu terminal I can also do from PowerShell window, provided that I started the WSL within it.

With WSL open in the PowerShell terminal, I did several checks as to whether the new distro at the new location was working properly:

```
PS C:\Users\Ioana> wsl ~
iweber@Lenovo-Ioana:~$ wsl.exe -l -v
  NAME      STATE      VERSION
* Ubuntu_WSL2    Running      2
  Ubuntu     Stopped      2
iweber@Lenovo-Ioana:~$ myfiles_exercises
iweber@Lenovo-Ioana:~/ABI/Exercises_Ioana$ ls -lat
total 48
drwxr-xr-x  4 iweber iweber 4096 Mar  4 10:47 Linux
drwxr-xr-x 12 iweber iweber 4096 Mar  4 09:25 .
drwxr-xr-x  2 iweber iweber 4096 Mar  1 12:32 Bash_scripting_Ioana
drwxr-xr-x  3 iweber iweber 4096 Feb  6 09:56 Databases
drwxr-xr-x  2 iweber iweber 4096 Feb  6 09:55 Biopython
drwxr-xr-x  7 iweber iweber 4096 Feb  6 09:46 ..
drwxr-xr-x  4 iweber iweber 4096 Feb  6 09:45 Alignments
drwxr-xr-x  2 iweber iweber 4096 Feb  6 09:45 Analysis_and_stats
drwxr-xr-x  2 iweber iweber 4096 Feb  6 09:45 NGS
drwxr-xr-x  6 iweber iweber 4096 Feb  6 09:45 Python
drwxr-xr-x  2 iweber iweber 4096 Feb  5 13:01 Structural_Bioinfo
drwxr-xr-x  2 iweber iweber 4096 Feb  5 13:01 Working_w_Strings
iweber@Lenovo-Ioana:~/ABI/Exercises_Ioana$
```

I started WSL, then tested which distro was running using the code provided by [NotTheDr01ds](#). In addition to that, I checked whether I could still use an alias I had set in my .bashrc some time ago, which simply changes the working directory directly to one where I store files for the ABI course. I also listed the contents of the directory, just to be extra sure that they are what I know them to be and that my default user is still the owner and has all of the permissions, as I know this can be difficult to alter should I not be the owner any longer [***source?***](#)

And I then de-registered my old distro from WSL.

A consequence that I hadn't anticipated is that now I cannot use the Ubuntu app from the store directly any longer, at least not without creating a completely new distro...but that's easily fixed by running Ubuntu from the Windows Terminal instead, which I've taken to doing.

Result: successfully changed location, freeing some space on my currently plighted OS drive. I will need to change it back once I exchange the OS drive to a larger one, but at least the datasets I have stored in WSL and all of my course materials are safe for now on the other SSD.

Considering to make regular .tar/VHD backups of the WSL on the second SSD just to be on the safe side

I then wanted to get and extract all of the other SRA archives related to the E 17.5 cortices in this study. To do so, I wrote a Bash script containing the commands for fetching the remaining SRA archives (prefetch command of the SRA toolkit) and to unpack them (fasteq-dump command), set it to executable, and started it from the shell.

Note

script?

I checked the progress on occasion via TeamViewer, and saw things moving along through the continual increase in the size of the export folder. As a side note, here I also saw that the WSL virtual machine was still using up almost half of my RAM during this operation, and I think this is due to the limitations that WSL has in terms of memory usage, at least when compared to a virtual machine, so I decided I must take the plunge and create one. (see linked post about VM)

Quality control of FastQ files with the RNA-seq reads (still on WSL)

Quality control of individual FastQ files

For the quality control of this small number of files, I used [FastQC](#), which is an open-source tool that checks the reads in FastQ files for various quality parameters, such as sequence quality scores, GC content, sequence duplication levels, or the presence of sequences left behind by the oligonucleotide adapters used for the amplification and actual sequencing of the cDNA fragments. I was pleasantly surprised that opening the application from the Ubuntu command line resulted in a user-friendly GUI, with which I could select the sequences to be analyzed. After analyzing the two FastQ files resulting from the first sequencing run (SRR13761520_1 and SRR13761520_2), I got quite different results. Whereas the file ending in 2 passed the quality checks in all but one department ("Per base sequence content" gave a warning), the file ending in 1 had a few more issues. It failed the "Per base sequence content" testing and gave warnings for "Per sequence GC content" and "Sequence duplication levels". I suppose this difference results from one of the files representing one item from each pair of reads and the other file the sequence complementary to it [source?]. As these two reads stem from two separate sequencing reactions [source], it is normal that one set can have different sequencing quality, based on differences in the reaction setup, base composition, or technically-introduced biases that only happened for one of the two sequencing runs.

Nonetheless, I wanted to understand better what the parameters returned by FastQC are and what they mean for the usability of the data for further analyses. To better understand how this data was pre-processed, I foraged a bit in the Series Matrix file provided on the GEO page of the study, but the authors don't give a lot of details about the handling of the data, aside from Reads filtering under criteria removing reads with 20% of the base quality lower than 13". Considering that FastQC did not indicate the presence of any known adapter sequences in the data and that the files were already clearly sorted by experimental condition and subject, (I hope) it's safe to assume that any necessary demultiplexing and adapter trimming has been performed by the authors of the study/the sequencing facility they worked with, plus applying this quality filter they mention in the Series Matrix file.

To centralize all analysis info for all of the FastQ files generated by FastQC, I settled on [MultiQC](#). I envision myself working with single-cell RNA-seq datasets at a later time point, and MultiQC can, as the name suggests, run the analyses in parallel on many files, making it a lot more efficient in handling large numbers of datasets. MultiQC is a part of the bcbio-nextgen Python package, so I wanted to install it in my Python on the WSL Ubuntu as instructed by the package's GitHub [page](#). I thought this installed not only the package but also all dependencies (other modules) that are needed for it to run...but it didn't. It told me it was missing crucial data, such as the genome assemblies for mouse and human, which prevented it from running at all.

During the lengthy installation process, I kept getting a message for all packages "Solving environment: failed with initial frozen solve. Retrying with flexible solve.". Thanks to another kind internet [stranger](#), I found that this is a problem that stems from having a version of Python that's newer than what many of these packages are optimized to work with. [what version do i have on my WSL? what versions needed? conda search python, then conda install python=desired_version_number, and then restart Ubuntu and Py, and update bcbio-nextgen package, hoping that this will resolve dependency issues]. However, the installation of bcbio-nextgen was still running (also, still running after 3h with the WSL virtual machine using 85 GB RAM sounds unusual and a bit alarming).

I realized that this may be due to WSL being less efficient in its memory usage than a full-fledged virtual machine [source?], so I proceeded to kill this process and install Oracle VirtualBox 7 and make an Ubuntu VM on it.

MultiQC: General quality stats of the pre-eclampsia FastQ files

I saw that multiqc takes one full folder as input. I could call `fastqc` from the command line individually for all of my FastQ files, but...why would I do something rather tedious to do for all of the 16 FastQ files? So I quickly wrote a small Bash script to automate the process:

```
#!/bin/bash

directory="/home/iweber/Documents/ABI_files_NGS/6484_fastq/"

for file in "$directory"/*.fastq; do
    if [ -f "$file" ]; then
        fastqc "$file" --outdir /home/iweber/Documents/ABI_files_NGS/FastQC_results
    fi
done
```

This simply cycles through all of the files with a .fastq ending in the indicated directory, stores the name of the file in the variable called `file`, and then calls `fastqc` to operate on the content of the file name variable (`$file`), while outputting the result into the `FastQC_results` folder.

I made it executable with `chmod` and then ran it, and:

```
(NGS) iweber@iweber-VirtualBox:~/Documents/Datasets/Pre_eclampsia_mice$ ./fastqc_loop_folder.sh
null
Started analysis of SRR13761520_1.fastq
Approx 5% complete for SRR13761520_1.fastq
Approx 10% complete for SRR13761520_1.fastq
Approx 15% complete for SRR13761520_1.fastq
Approx 20% complete for SRR13761520_1.fastq
Approx 25% complete for SRR13761520_1.fastq
Approx 30% complete for SRR13761520_1.fastq
Approx 35% complete for SRR13761520_1.fastq
Approx 40% complete for SRR13761520_1.fastq
Approx 45% complete for SRR13761520_1.fastq
Approx 50% complete for SRR13761520_1.fastq
Approx 55% complete for SRR13761520_1.fastq
Approx 60% complete for SRR13761520_1.fastq
Approx 65% complete for SRR13761520_1.fastq
Approx 70% complete for SRR13761520_1.fastq
Approx 75% complete for SRR13761520_1.fastq
Approx 80% complete for SRR13761520_1.fastq
Approx 85% complete for SRR13761520_1.fastq
Approx 90% complete for SRR13761520_1.fastq
Approx 95% complete for SRR13761520_1.fastq
Analysis complete for SRR13761520_1.fastq
null
Started analysis of SRR13761520_2.fastq
Approx 5% complete for SRR13761520_2.fastq
```

It worked! Naturally, I then ran MultiQC to summarize all of the reports obtained with FastQC.

```
(NGS) iweber@iweber-VirtualBox:~/Documents/Datasets/Pre_eclampsia_mice$ multiqc FastQC_results/
/// MultiQC 🔎 | v1.21

multiqc | Search path : /home/iweber/Documents/Datasets/Pre_eclampsia_mice/FastQC_results
searching | 100% 32/32
fastqc | Found 16 reports
multiqc | Report      : multiqc_report.html
multiqc | Data       : multiqc_data
multiqc | MultiQC complete
```

Let's have a more detailed look at the results.

General Statistics

Sample Name	% Dups	% GC	M Seqs
SRR13761520_1	33.6%	48%	21.2 M
SRR13761520_2	28.7%	48%	21.2 M
SRR13761521_1	33.5%	48%	21.6 M
SRR13761521_2	29.6%	49%	21.6 M
SRR13761522_1	34.4%	48%	21.1 M
SRR13761522_2	29.8%	49%	21.1 M
SRR13761523_1	32.8%	48%	19.3 M
SRR13761523_2	28.5%	49%	19.3 M
SRR13761524_1	33.4%	48%	20.4 M
SRR13761524_2	28.3%	49%	20.4 M
SRR13761525_1	33.7%	48%	21.4 M
SRR13761525_2	28.3%	48%	21.4 M
SRR13761526_1	34.4%	48%	20.1 M
SRR13761526_2	32.2%	49%	20.1 M
SRR13761527_1	35.6%	48%	20.7 M
SRR13761527_2	33.6%	49%	20.7 M

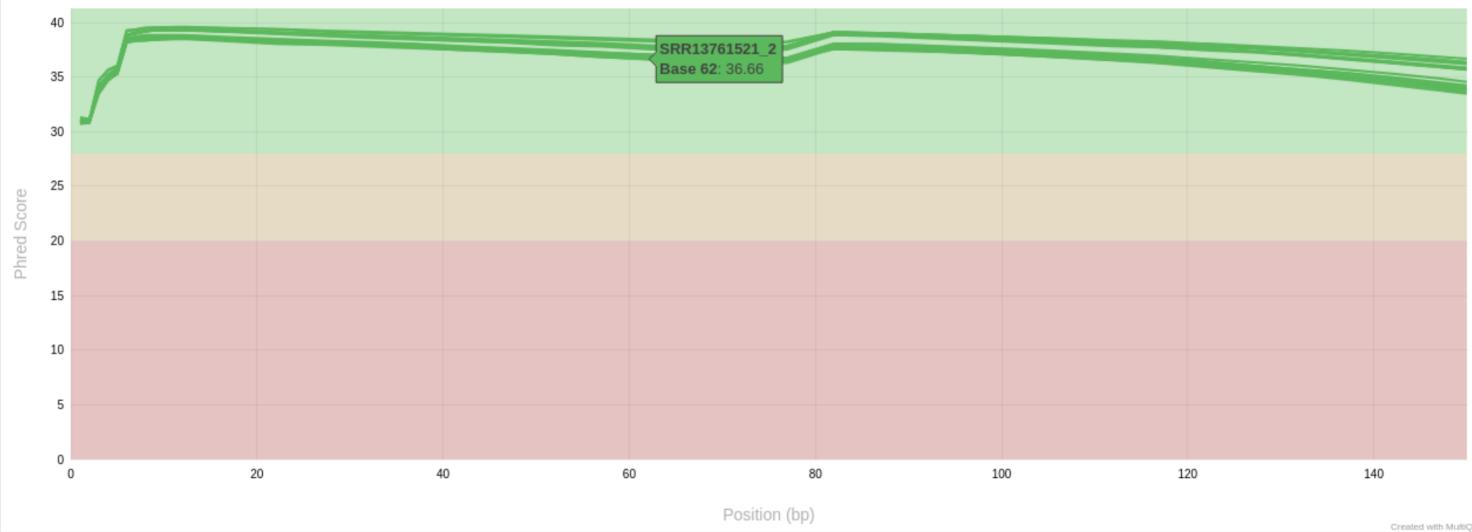
So far, so good. Each of the files contains around 21 million reads, with a GC content that's normal for the mouse ([source?]). The amount of sequences flagged as duplicates is around 30%, which is to be expected due to the amplification steps during the library preparation ([quote from Ioana Lemnian]).

The Phred scores are, on average, also good across the entire length of the reads., even though, as usual, the quality at the beginning, in the first 10 nucleotides or so from the 5' end, is a bit lower than in the rest of the sequence.

The mean quality value across each base position in the read.

[Export Plot](#)

FastQC: Mean Quality Scores



What looked a bit less rosy was the per base sequence content quantification. For all of the `_2` ending samples, FastQC gave a warning for the nucleotide composition, which should ideally be uniform across the read, with each base keeping its proportion percentage and giving a plot with four parallel lines. It's normal that the beginning of the read is a bit more noisy, up to 10 bases, and that's what we see here. Even for the samples ending in `_1`, that were flagged as "failed", the irregularities are restricted to this area. This looks like a good case for trimming these fragment start regions off in subsequent steps.

Per Base Sequence Content

8 8

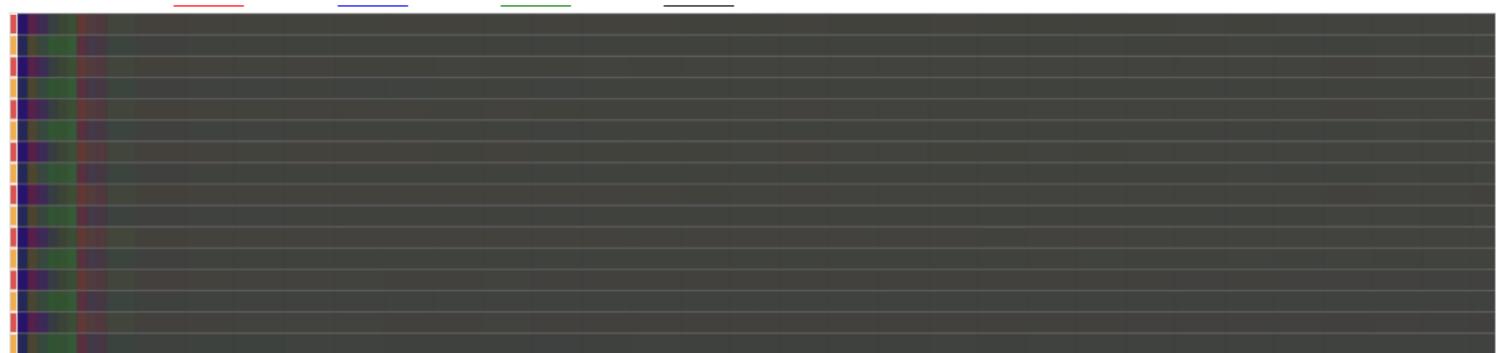
Help

The proportion of each base position for which each of the four normal DNA bases has been called.

Click a sample row to see a line plot for that dataset.

Rollover for sample name

Position: - %T: - %C: - %A: - %G: -

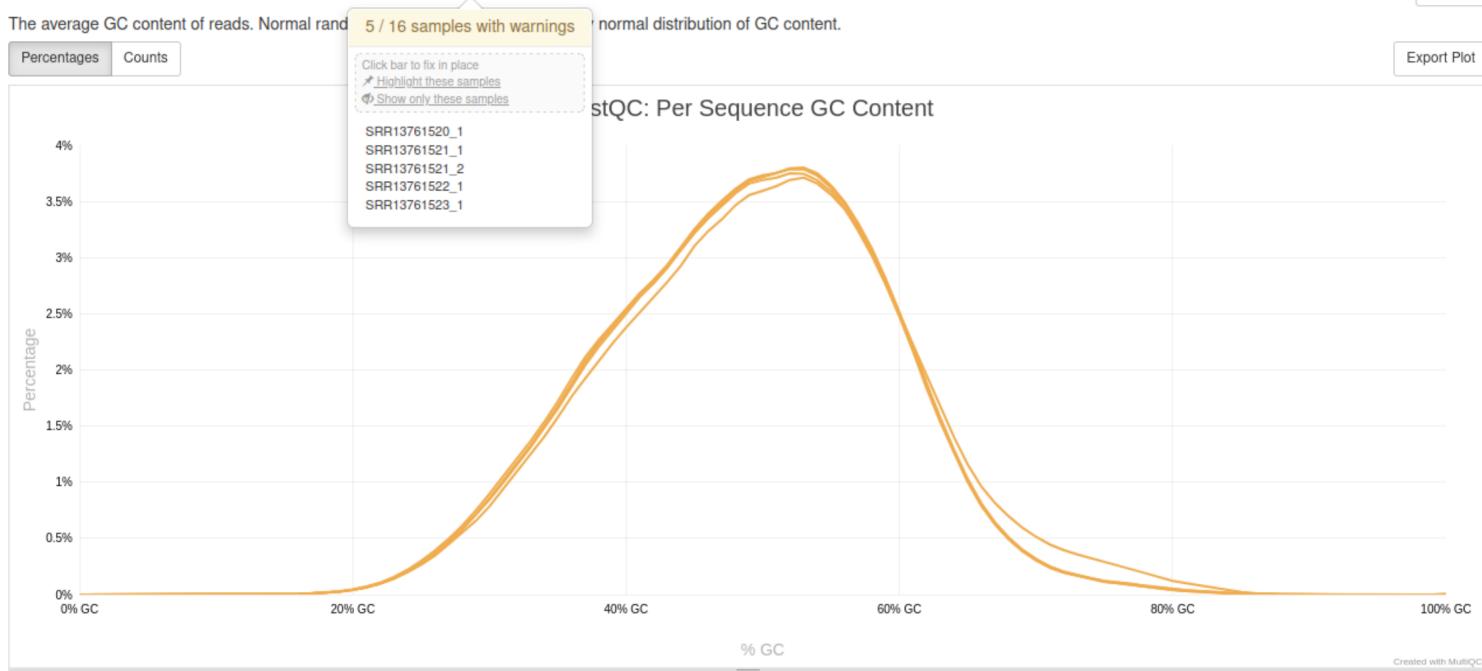


The per sequence GC content looked mostly alright (the reads in 11 of the 16 FastQ files passed with no warning), but there were some where FastQC gave a warning, so I know to keep an eye out in case these files cause some strange, systematic error down the line.

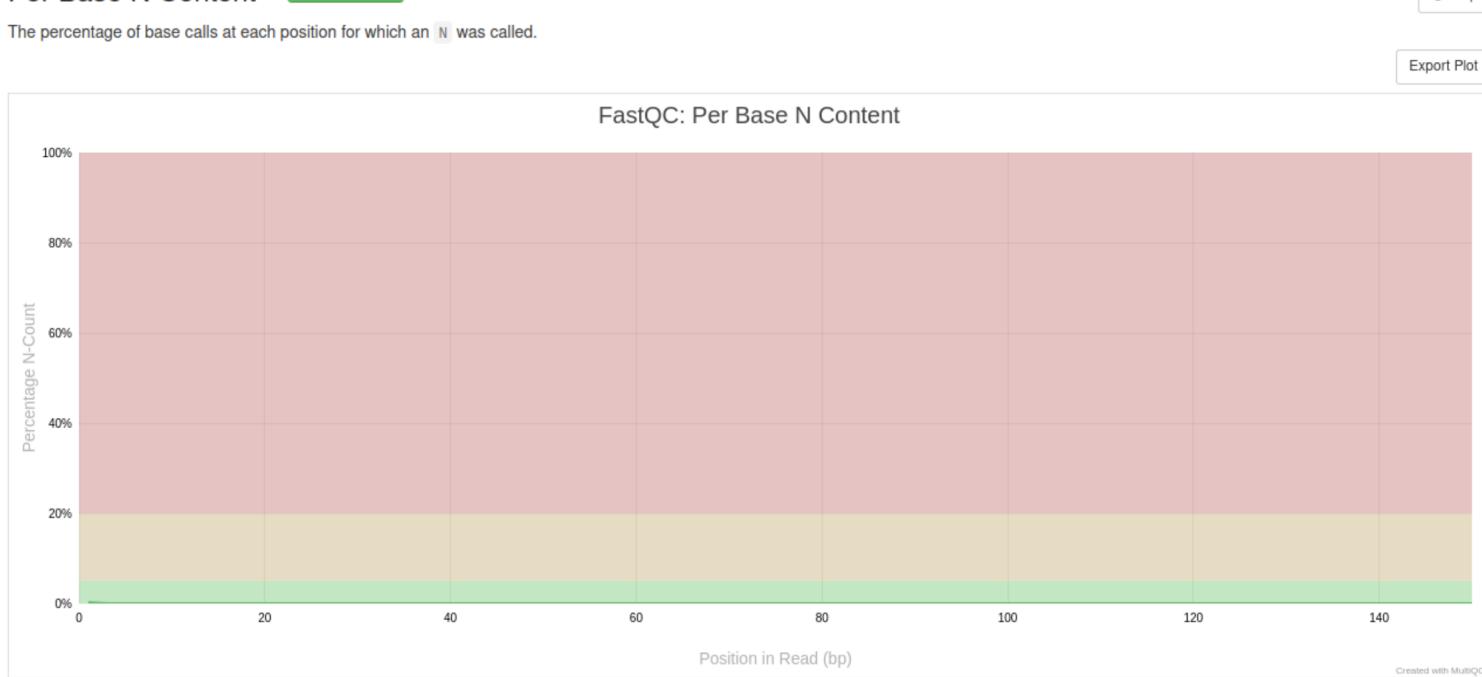
SRR13761520_1
SRR13761521_1
SRR13761521_2
SRR13761522_1

SRR13761523_1

Per Sequence GC Content



Per Base N Content



There were no warnings for overrepresented sequences, and FastQC did not find any of the usual sequences of adapters used in NGS kits in the FastQ files. This tells me that the files were processed before their further use, perhaps directly by the sequencing facility after demultiplexing [maybe **bcl2fastq** or **BCL convert** can also do this? Or maybe the authors just uploaded the already trimmed reads to GEO.]

The total amount of overrepresented sequences found in each library.

16 samples had less than 1% of reads made up of overrepresented sequences

Top overrepresented sequences

Top overrepresented sequences across all samples. The table shows 20 most overrepresented sequences across all samples, ranked by the number of samples they occur in.

[Copy table](#)
[Violin plot](#)

Showing 0% rows.

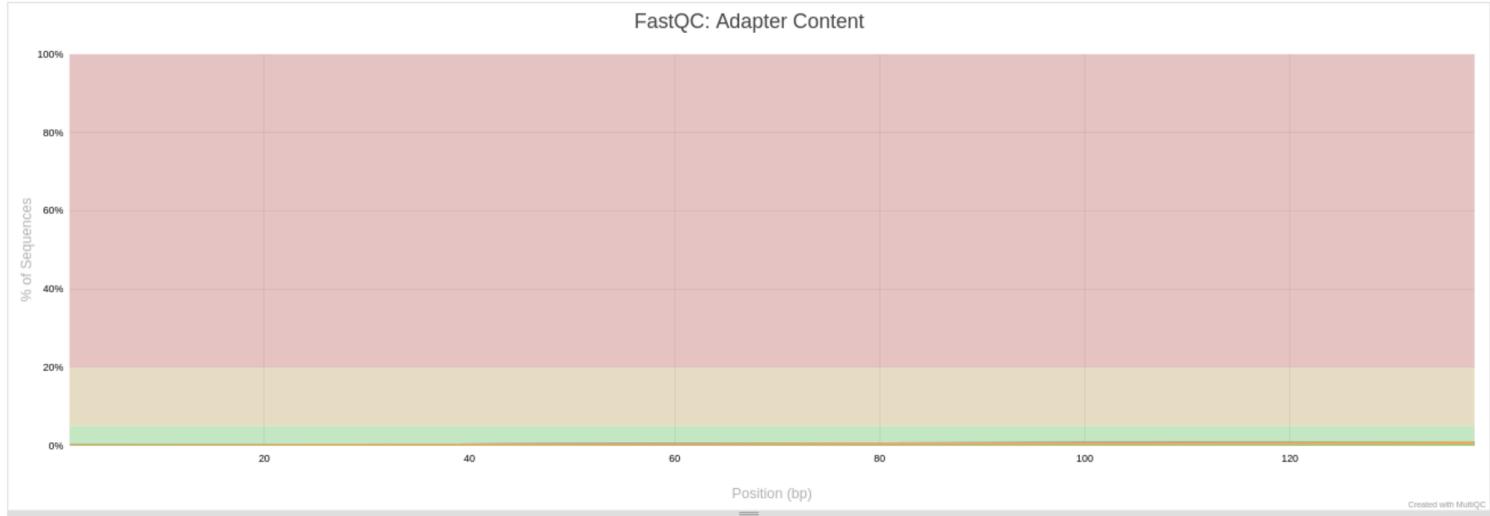
[Export as CSV](#)
[Overrepresented sequence](#)

Adapter Content

16

[Help](#)

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

[Export Plot](#)


Summary of initial quality warnings for the reads

Warnings:

Datasets with warnings for GC content:

- SRR13761520_1
- SRR13761521_1
- SRR13761521_2
- SRR13761522_1
- SRR13761523_1

Dupe warnings:

- SRR13761520_1
- SRR13761521_1
- SRR13761522_1
- SRR13761523_1
- SRR13761524_1
- SRR13761525_1
- SRR13761526_1
- SRR13761526_2
- SRR13761527_1
- SRR13761527_2

Real solution for space issues: Making a VirtualBox virtual machine to run Ubuntu

Before taking the plunge to make a virtual machine, I made some very thorough backups of my system. I made a restore point, then also used File History and Backup and Restore to make images of my WinOS. Finally, I made a recovery USB in case I wouldn't be able to start the OS after making changes to the BIOS to enable virtualization. Maybe I wouldn't be this paranoid if tinkering with the virtualization options hadn't BSOD-ed my back-then three days old laptop right after setting it up out of the box for the very first time...so better safe than sorry.

Got an Ubuntu 22.04 image from the Canonical website, and installed it as VM with the name Ubuntu 22x64 using the Oracle VirtualBox. I allotted it 96 GB RAM, and 6 CPUs, to make sure it has all of the resources it might need to analyze large data sets.

I thought this worked at first, and then the issues started popping up like mushrooms after the rain: the GNOME terminal that comes with the distro didn't work, I couldn't update the Snap Store even though the update was recommended. I installed another terminal emulator called Guake and attempted to install the bcbio-nextgen tool bundle...and it got stuck with conda not being able to solve the environment (this issue seems to be known and caused due to R package dependencies that conda can't solve, or at least not quickly, see [link](#)). After stopping that process, I tried updating Python, which almost completely broke the entirety of the Ubuntu installation: the VM stopped showing me the Ubuntu desktop at all and only did so for the TTY terminal, with which I was not able to refresh the installation of the GUI desktop.

(Side note: I also tried the terminal emulator Terminator, whose options I also enjoyed).

I removed the VM completely and reinstalled everything from scratch...only to find out that, suddenly, my user, created during the VM creation process, had no `sudo` privileges and I had no way of accessing the root to give it said privileges. Removed the entire VM again.

Reinstalled Ubuntu 22.04 on a VM with the name `ubuntu22x64`. Gave it my username `iweber` and my usual password for the purpose, and selected guest additions from the suggested iso. Let's see if I now have `sudo` rights for my user... Nope.

Found a forum entry here saying that the issue can be that this unattended installation option that the VM provides creates an user without `sudo` rights. Skipped this option during the creation of the new VM, leaving me to install Ubuntu manually after the creation of the VM.

Did so, selected the keyboard and then to erase disk and install Ubuntu.

After the installation and restart, noticed that the first DOS-like screen had some errors talking about a graphics device and issues with unsupported hyper-visors, but Ubuntu started nonetheless and I could log in with user `IoWe` (my user, with actual username `iweber`). ***LE: this keeps happening when the virtual machine powers up, but it still boots, and I am currently not seeing any other issues.***

And magic! I could even open the GNOME Terminal now!

I then let Software Updater update everything it said needed updating. It ran and I noticed it again giving some errors regarding Snaps...but then it finalized the update and said it needed to restart to apply all of the updates, so I let it do so.

After the restart, the terminal still works. I could also `whoami` myself AND `sudo whoami`, which, after providing my password, also let me set the user to root, so I now know I can install any software I may be needing down the line.

I went to the Anaconda website to find out how to download packages that I [previously found out](#) need to be installed for the Anaconda Navigator GUI (`anaconda3`) to work. The command I used in the terminal was:

```
sudo apt-get install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 libxcursor1 libcomposite1 libasound2 libxi6 libxtst6
```

and it executed without me seeing any errors.

Tried to download the Anaconda Navigator installer [Anaconda3-2024.02-1-Linux-x86_64.sh](#) as recommended by the Anaconda Navigator page by using

```
curl -O https://repo.anaconda.com/archive/Anaconda3-2024.02-1-Linux-x86_64.sh`
```

...and found out I first had to install the curl package (`cry-laugh`), which I quickly did with

```
sudo apt install curl
```

I could then finally run the curl command above to get the Anaconda Navigator. What was displayed during the installation was this:

```
iweber@iweber-VirtualBox:~$ curl -O https://repo.anaconda.com/archive/Anaconda3-2024.02-1-Linux-x86_64.sh
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload Total   Spent    Left Speed
 8  997M    8 82.9M    0     0  6819k      0  0:02:29  0:00:12  0:02:17 6633k
```

The percentage reached 100 and I had a new command line available, so I knew it was installed. I checked, as per the rec of the Anaconda Navigator install page, whether the checksum is correct using `shasum -a 256 Anaconda3-2024.02-1-Linux-x86_64.sh` (curl always downloads in the working directory). This check returned the expected SHA-256 hash for this version, so I knew I was good to go actually using the installer. Used `bash Anaconda3-2024.02-1-Linux-x86_64.sh` to run the installer, and what I saw was:

```
[/home/iweber/anaconda3] >>>  
PREFIX=/home/iweber/anaconda3  
Unpacking payload ...
```

```
Installing base environment...
```

```
Downloading and Extracting Packages:
```

```
Downloading and Extracting Packages:
```

```
Preparing transaction: done  
Executing transaction: -
```

```
Installed package of scikit-learn can be accelerated using scikit-learn-intelex.  
More details are available here: https://intel.github.io/scikit-learn-intelex
```

```
For example:
```

```
$ conda install scikit-learn-intelex  
$ python -m sklearnex my_application.py
```

```
done  
installation finished.  
Do you wish to update your shell profile to automatically initialize conda?  
This will activate conda on startup and change the command prompt when activated.  
If you'd prefer that conda's base environment not be activated on startup,  
run the following command when conda is activated:
```

```
conda config --set auto_activate_base false
```

```
You can undo this by running `conda init --reverse $SHELL`? [yes|no]  
[no] >>>
```

```
For more information, see the FAQ.
```

I confirmed that I wanted to let conda configure my shell profile so that it can be automatically initialized by entering yes .

```
done
installation finished.
Do you wish to update your shell profile to automatically initialize conda?
This will activate conda on startup and change the command prompt when activated.
If you'd prefer that conda's base environment not be activated on startup,
    run the following command when conda is activated:
```

```
conda config --set auto_activate_base false
```

```
You can undo this by running `conda init --reverse $SHELL`? [yes|no]
```

```
[no] >>> yes
```

```
no change      /home/iweber/anaconda3/condabin/conda
no change      /home/iweber/anaconda3/bin/conda
no change      /home/iweber/anaconda3/bin/conda-env
no change      /home/iweber/anaconda3/bin/activate
no change      /home/iweber/anaconda3/bin/deactivate
no change      /home/iweber/anaconda3/etc/profile.d/conda.sh
no change      /home/iweber/anaconda3/etc/fish/conf.d/conda.fish
no change      /home/iweber/anaconda3/shell/condabin/Conda.ps1
no change      /home/iweber/anaconda3/shell/condabin/conda-hook.ps1
no change      /home/iweber/anaconda3/lib/python3.11/site-packages/xontrib/conda.xsh
no change      /home/iweber/anaconda3/etc/profile.d/conda.csh
modified       /home/iweber/.bashrc
```

```
==> For changes to take effect, close and re-open your current shell. <==
```

```
Thank you for installing Anaconda3!
```

```
iweber@iweber-VirtualBox:~$
```

This seems to have worked. I closed and reopened my shell and checked if I can use the `conda` command to open the manager.

```
Success!
```

```
(base) iweber@iweber-VirtualBox:~$ conda
usage: conda [-h] [-v] [--no-plugins] [-V] COMMAND ...
```

```
conda is a tool for managing and deploying applications, environments and packages.
```

```
options:
```

-h, --help	Show this help message and exit.
-v, --verbose	Can be used multiple times. Once for detailed output, twice for INFO logging, thrice for DEBUG logging, four times for TRACE logging.
--no-plugins	Disable all plugins that are not built into conda.
-V, --version	Show the conda version number and exit.

```
commands:
```

```
The following built-in and plugins subcommands are available.
```

COMMAND	
activate	Activate a conda environment.
build	Build conda packages from a conda recipe.
clean	Remove unused packages and caches.
compare	Compare packages between conda environments.
config	Modify configuration values in .condarc.
content-trust	Signing and verification tools for Conda
convert	Convert pure Python packages to other platforms (a.k.a., subdirs).
create	Create a new conda environment from a list of specified packages.
deactivate	Deactivate the current active conda environment.
debug	Debug the build or test phases of conda recipes.
develop	Install a Python package in 'development mode'. Similar to `pip install --editable`.
doctor	Display a health report for your environment.
index	Update package index metadata files.
info	Display information about current conda install.
Applications	Initialize conda for shell interaction.

And now I finally had hope of being able to install the bcbio-nextgen package.

...but then I realized I still couldn't see the Anaconda Navigator as a GUI version at this point. I saved the state of my VM and restarted it...and quickly realized that this isn't a restore point like in Windows, but a genuinely frozen point in time with all open apps, so more like a hibernation in Windows. So I then truly restarted it from the VirtualBox menu.

I then researched and found that Anaconda Navigator does not show up as an app in the Ubuntu Apps, but can still be opened by running the following very simple command in the terminal:

```
anaconda-navigator
```

And yes, this opened the software! (after some warnings relating to GNOME and rescaling) I chose to update as recommended. However, I did not find the bcbio-nextgen package listed among the packages that could be installed, in spite of creating environments

I discovered that these issues may also be fixed if using mamba as a dependency solver instead of conda. I read the docs of how to install Mamba [here](#). It seems I need to install Miniforge, which seems to be the most recent version, as opposed to using Mambaforge. I read that I must install Miniforge (Mamba) in the so-called base environment and that only it and conda, and no other packages whatsoever, may be installed in the base, because other packages could break both environment managers.

I downloaded the Miniforge .sh binary and ran it in the terminal with `bash Miniforge3-Linux-x86_64.sh`.

It worked:

```
(base) iweber@iweber-VirtualBox:~$ mamba
usage: mamba [-h] [-v] [--no-plugins] [-V] COMMAND ...

conda is a tool for managing and deploying applications, environments and packages.

options:
  -h, --help            Show this help message and exit.
  -v, --verbose         Can be used multiple times. Once for detailed output,
                       twice for INFO logging, thrice for DEBUG logging, four
                       times for TRACE logging.
  --no-plugins          Disable all plugins that are not built into conda.
  -V, --version          Show the conda version number and exit.

commands:
  The following built-in and plugins subcommands are available.

COMMAND
  activate              Activate a conda environment.
  clean                 Remove unused packages and caches.
  compare               Compare packages between conda environments.
  config                Modify configuration values in .condarc.
  create                Create a new conda environment from a list of specified
                       packages.
  deactivate             Deactivate the current active conda environment.
  doctor                Display a health report for your environment.
  info                  Display information about current conda install.
  init                  Initialize conda for shell interaction.
  install               Install a list of packages into a specified conda
                       environment.
  list                  List installed packages in a conda environment.
  notices               Retrieve latest channel notifications.
  package               Create low-level conda packages. (EXPERIMENTAL)
  remove (uninstall)    Remove a list of packages from a specified conda
                       environment.
  rename                Rename an existing environment.
```

Followed installation instructions from the bcbio-nextgen page: <https://bcbio-nextgen.readthedocs.io/en/latest/contents/installation.html#automated>

First, installed Git from git-scm using `sudo apt-get install git`. Checked if I have tar installed by simply typing `tar` in my terminal, and yes, I do.

I then started the installation by calling `python3` and, using it, the Py script that is supposed to install bcbio-nextgen. I used the options `--nodata` to do a minimal installation, without genomes etc, and `--mamba` to use mamba as a package manager. The command looked like:

```
python3 bcbio-nextgen-install.py /home/iweber/bcbio --tooldir=/home/iweber/bcbio/tools --nodata --mamba
```

I noticed it said it was installing mamba, and am not sure why - maybe because I am not installing bcbio within the mamba directory? But I thought that that amounts to installing in the base environment and is not recommended? Either way, the installation did not succeed. It also told me I was using a deprecated version of conda, so I updated it to the last version (24.3 at the time of writing this).

LE: I think the Py installer script for bcbio-nextgen installs Miniconda and uses that to solve the environment/package dependencies, and that's why it pops up as deprecated, even though I just installed the most recent version of conda

Can I somehow force bcbio installer to use the properly installed anaconda3 instead of Miniconda as a solver? - maybe not a good idea. If installer calls for a particular version of conda, it may have a good reason in terms of package dependencies.

After this, whenever I tried running the installation script again, it told me it cannot create the data directory required for the installation, a positional argument that needs to be given upon running the script. So I decided to run it with root privileges using

```
sudo python3 bcbio-nextgen-install.py /bcbio-data --nodata --mamba
```

in order to create a fully new data directory called bcbio-data, and, as before, to not download any genome data and to use mamba as an environment solver instead of conda. And now it seems like I'm stuck in the same solving environment step as has happened every time before when I tried it in the WSL:

```
ruamel_yaml      pkgs/main/linux-64::ruamel_yaml-0.15.100-py37h27cf23_0
setuptools       pkgs/main/linux-64::setuptools-52.0.0-py37h06a4308_0
six              pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_0
sqlite           pkgs/main/linux-64::sqlite-3.36.0-hc218d9a_0
tk               pkgs/main/linux-64::tk-8.6.10-hbc83047_0
tqdm             pkgs/main/noarch::tqdm-4.61.2-pyhd3eb1b0_1
urllib3          pkgs/main/noarch::urllib3-1.26.6-pyhd3eb1b0_1
wheel            pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
xz               pkgs/main/linux-64::xz-5.2.5-h7b6447c_0
yaml             pkgs/main/linux-64::yaml-0.2.5-h7b6447c_0
zlib             pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3

Preparing transaction: done
Executing transaction: done
installation finished.
Installing mamba
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: |
```

I killed the process with `Ctrl+C`. I decided to create a separate conda environment called `bcbio-env` and, within it, to install one of the latest stable versions of Python to see if that helps matters. I had Python 3.10.12 and updated to 3.12.2.

Then, attempted the installation with root privileges in this environment with the same sudo command as before. Same problem with "failed with intial frozen solve. Retrying with flexible solve.".

After that, "failed with repodata from current_repodata.json, will retry with next repodata source"

Downgraded Python in my `bcbio-env` to python 3.9.0 and tried installing again with the sudo command. Also "failed with intial frozen solve. Retrying with flexible solve.". Killed process.

Upgraded Py to 3.12.2. Tried installing bcbio using pip3 ([pip is the Python package manager]). It said it worked, but, when calling `bcbio_nextgen.py --version`, it gave me an error message saying that the "six" package was not installed. I installed it manually with pip3 and tried calling the version of the package...and then got an error saying another package, toolz, was not installed. Did that as well, only to end up with another package error, this time for "yaml". However, when trying to install yaml via pip3, I got an error mesage that "no version could be found that satisfies requirements/no matching distribution found for yaml."

I ran `mamba clean -a` to remove all of these packages.

As the last strategy, I tried installing bcbio from bioconda as described on the [anaconda page](#):

```
conda install bioconda::bcbio-nextgen
```

And, suddenly, I could not only use `which bcbio-nextgen` to see where it is installed (`/home/iweber/miniforge3/envs/bcbio-env/bin/bcbio_nextgen.py`) but **even got a version number** when looking for it with `bcbio_nextgen.py --version`, namely version 1.2.9 ...which, indeed, is the latest version. At this point, I could've cried of happiness.

As life often goes, I realized only moments later that I can't actually download a new genome because the installation probably didn't create the correct folders for the data that bcbio-nextgen needs to run....

(Side note #2: another thing to remember is to start writing a logfile of everything I do in the terminal. For this, every time I start terminal, should run

```
script logfile
```

This automatically closes and creates that logfile when exiting terminal with Ctrl+D.

To get all packages currently installed in my conda environment:

```
conda list --export > my_conda_packages.txt [what was the point of this?]
```

I closed the VM at this point to come back to it another day...only to come back to not being able to open the Anaconda Navigator anymore from bash...

I added the path to the anaconda binaries folder to my PATH variable in the .profile file so that the bash knows to look here for executables as well, because I'd like to type less whenever I want to start the Anaconda Navigator: `export PATH=$PATH:/home/iweber/anaconda3/bin`. This, however, did not solve the issue.

I went ahead and updated the installation with `bash Anaconda3-2024.02-1-Linux-x86_64.sh -u` and got yet another weird error message.

And...the same errors over and over again. I looked further into this pipeline and decided its maintenance is lagging too far behind, as, for many other bioinformatic projects, the main contributors have, in the meanwhile, moved on to other projects.

I have decided to, instead, use Nextflow and established NGS data analysis pipelines that work on it.

Fixing Win-Linux clipboard

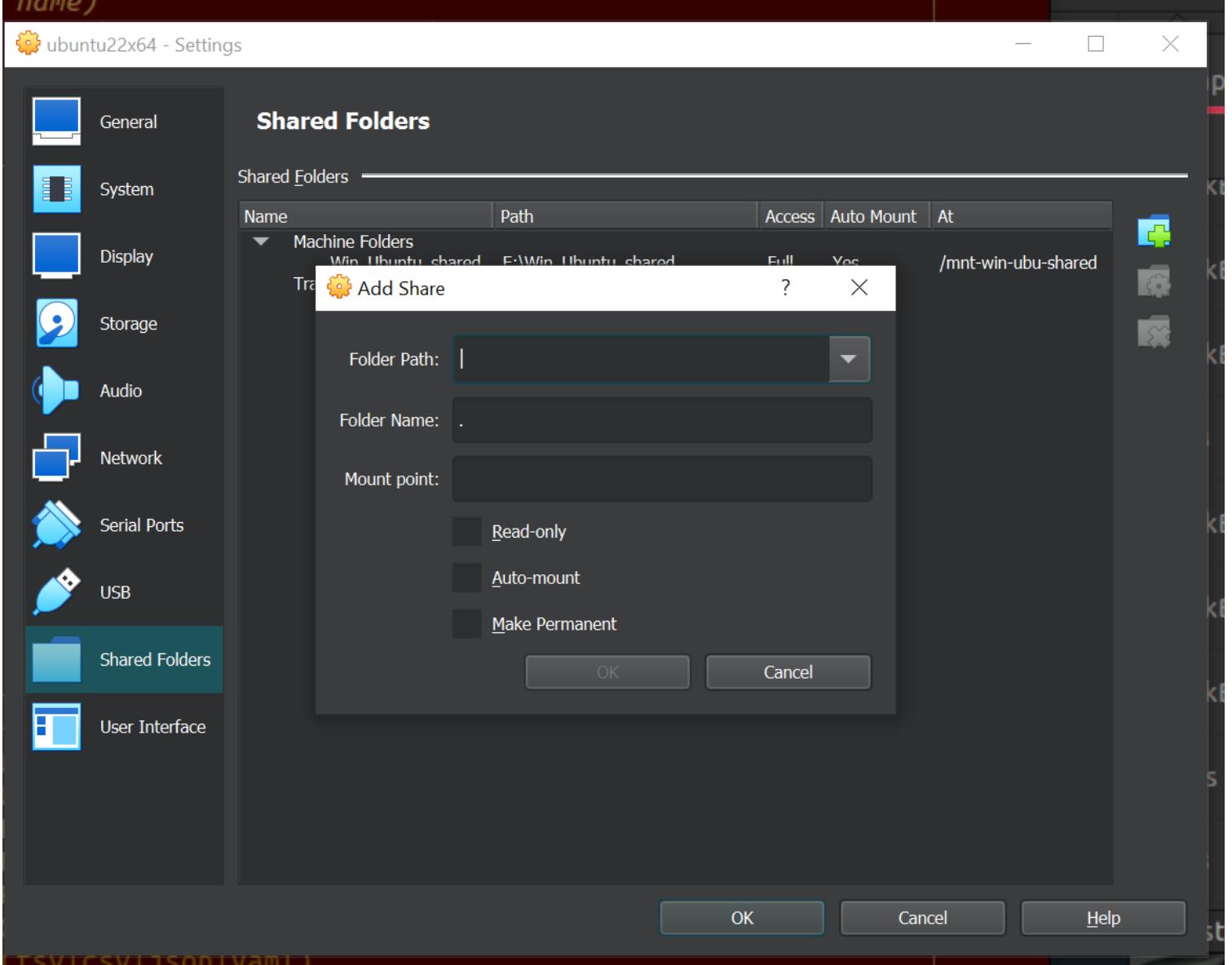
In the meanwhile, I tried fixing my highly annoying copy-paste and drag-and-drop issues between my virtual machine and my Windows host. I first installed:

```
sudo apt-install build-essential dkms linux-headers-generic ,
```

following [this answer](#) and the [page it refers to](#). I "mounted" the VA Guest Box addition disc image and ran the "autorun.sh" bash script it contains as a program (right click -> "Run as program"). After it asked for credentials, the terminal opened and I saw the expected screen. Restarted Ubuntu et voilà! Copying and pasting now finally works.

Creating an inter-OS shared folder between Win and Linux

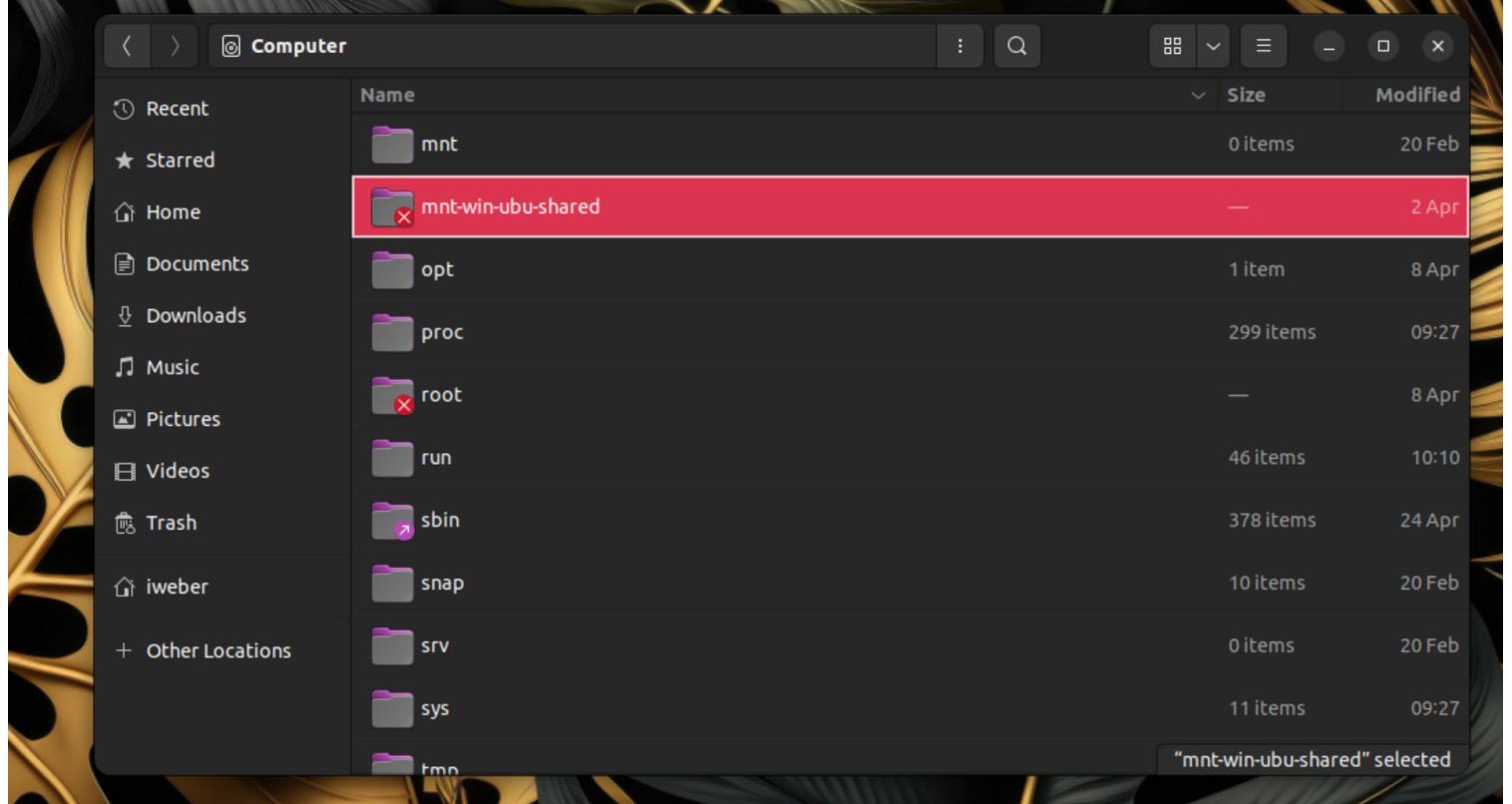
I had previously created a shared folder between my Windows host system and the Ubuntu virtual machine using the VM menu "Devices" -> "Shared folders" -> "Shared folders settings". Clicking the blue folder icon with a plus sign on it, I got such a window:



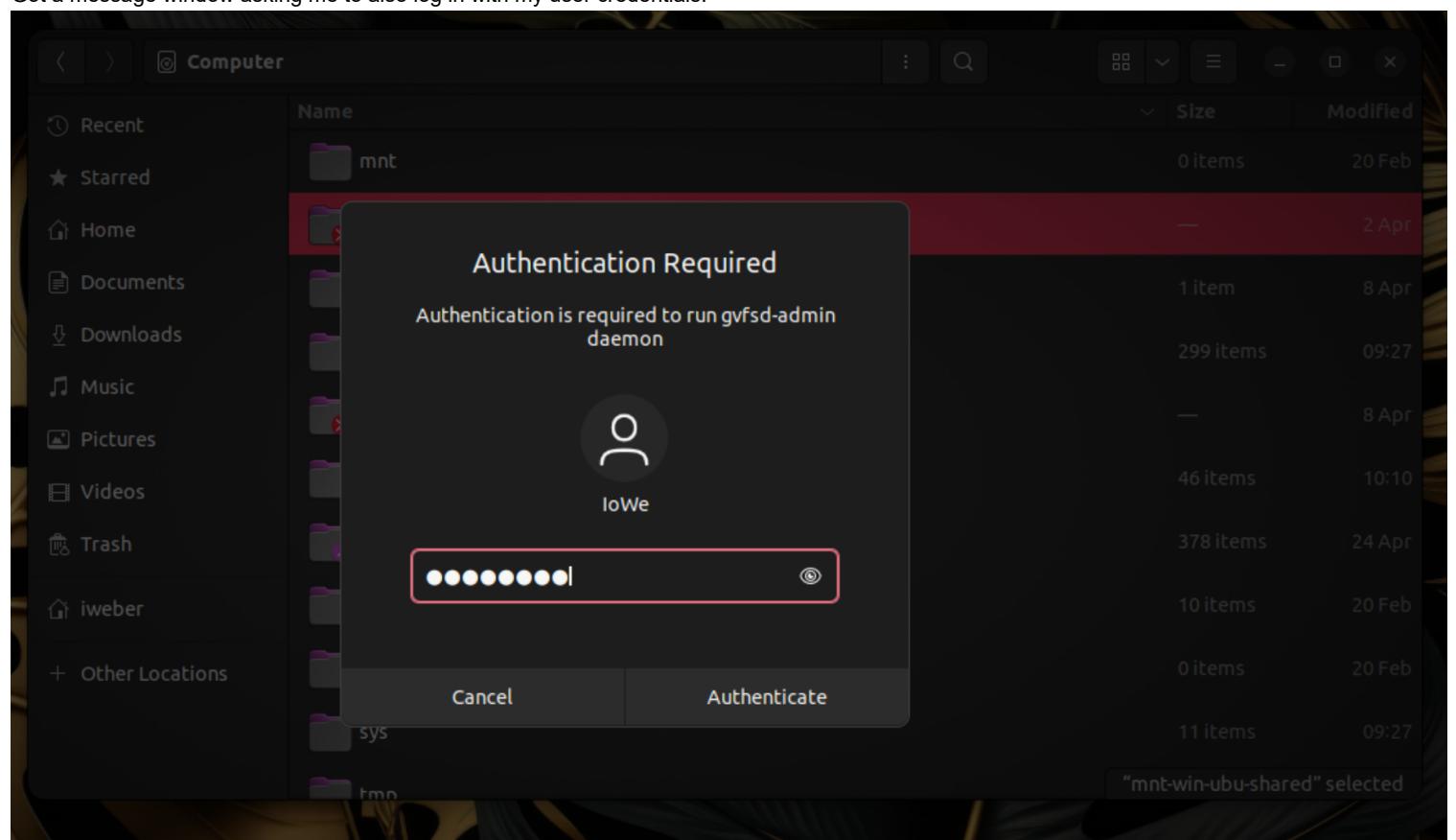
The "Folder Path" field allowed me to select a folder in my host to use as the shared folder. For this purpose, I went to Windows and, on the partition of the SSD that I had allotted to the virtual machine, I created a folder called "Win_Ubuntu_shared". I could then access this from the VM menu displayed above, and selected to auto-mount it and make it permanent in order to have it automatically accessible whenever I run the VM. I also gave it a mount point name in order to find it more easily under Linux, and chose "/mnt-win-ubuntu-shared" for this name.

But I didn't see this folder in my Linux file explorer, not even after restarting the virtual machine. I followed the instructions given [here](#) and ran `sudo usermod -aG vboxsf $USER` in terminal, and then tried accessing folder via Other locations -> Computer, where I finally saw the "mnt-win-ubuntu-shared"

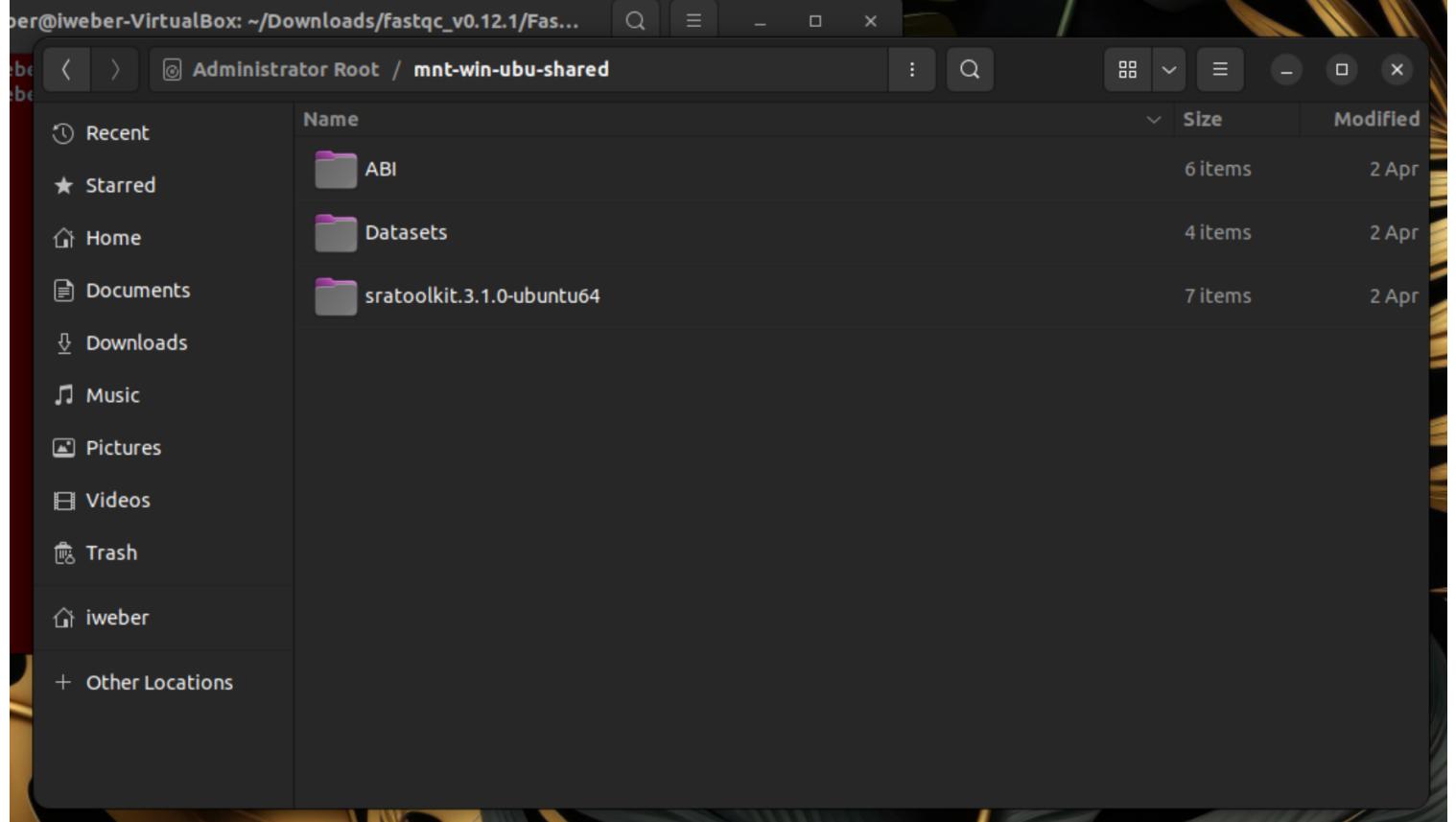
folder (remember, this is the name I gave the folder when setting it up in the virtual machine in the VirtualBox menu "Devices">> "Shared folders").



Got a message window asking me to also log in with my user credentials.



I did so, and I could finally see the contents of the shared folder! I immediately bookmarked this mnt-win-ubuntu-shared folder from the address bar of the explorer (three-dot menu) for easy access later.



I could now *finally* access the FastQ files I had previously generated on the WSL for further analyses!

Making the inter-OS shared folder auto-mount without asking for my password every time

I had previously created a folder for the internship data that's shared between my Windows host OS and Linux , and I realized I wasn't using it as the sole folder for my data in Linux because, every time I wanted to access it, it asked me for my user password. That made me reluctant to use it especially in the context of an automated analysis pipeline, because I feared that Linux suddenly forgetting the password might interrupt the pipeline at inconvenient times, e.g. when I'm not at my laptop. Additionally, this complicated setting up a proper GitHub repo. So, to fix that, I asked ChatGPT.

I used `mount | grep mnt-win-uba-shared` to check where the folder is located. It returned `Win_Ubuntu_shared on /mnt-win-uba-shared type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=0,gid=999,dmode=0770,fmode=0770,tag=VBoxAutomounter)`, so I now knew the address of my mounted folder.

I added my user to the permissions group for the virtual box folder (`vboxsf`) with `sudo usermod -aG vboxsf $USER`, then ran a `chown` command to change ownership of the folder:

```
'sudo chown -R $USER:vboxsf /mnt-win-uba-shared/
```

Now, when checking the permissions, I saw:

```
(base) iweber@iweber-VirtualBox:~$ mount | grep mnt-win-uba-shared
Win_Ubuntu_shared on /mnt-win-uba-shared type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=0,gid=999,dmode=0770,fmode=0770,tag=VBoxAutomounter)
(base) iweber@iweber-VirtualBox:~$ ^C
(base) iweber@iweber-VirtualBox:~$ sudo usermod -aG vboxsf $USER
[sudo] password for iweber:
(base) iweber@iweber-VirtualBox:~$ sudo chown -R $USER:vboxsf /mnt-win-uba-shared/
(base) iweber@iweber-VirtualBox:~$ ls -l /mnt-win-uba-shared/
total 8
drwxrwx--- 1 root vboxsf 4096 Mai 22 13:40 ABI
drwxrwx--- 1 root vboxsf 0 Apr 2 15:22 Datasets
drwxrwx--- 1 root vboxsf 4096 Apr 2 15:23 sratoolkit.3.1.0-ubuntu64
(base) iweber@iweber-VirtualBox:~$
```

so I knew I now have read and write privileges for the directory for both myself and the virtual box group I am in. I copied and pasted some files into this folder from both Windows and Linux to make sure it works bidirectionally, as it should, and it does.

Giving the inter-OS shared folder the correct permissions upon startup

However, upon restarting the virtual machine, Ubuntu still asked me for my password (specifically, "Authentication is required to run gvfsd-admin daemon") when trying to open the inter-OS shared folder...ChatGPT said this is likely due to startup permissions not being set properly and suggested I edit a file called "fstab" with admin privileges and add the path to the inter-OS shared folder. "fstab" is a "file systems table" with information about files, folders, and mount points [other source to confirm?]. I added `Win_Ubuntu_shared /mnt-win-ubu-shared vboxsf uid=1000,gid=1000,dmode=0770,fmode=0770 0 0` at the end of the file. I tried it, restarted the system, and I still got asked for the password...

ChatGPT's next idea was to create a script that will be automatically run at system startup and will mount the inter-OS shared folder with the appropriate permissions from the get-go. I removed the line I previously added to the fstab file and proceeded to create the new startup file. I created the file directly with nano, `sudo nano /usr/local/bin/mount_shared_folder.sh`, and added to it

```
#!/bin/bash
sudo mount -t vboxsf -o uid=1000,gid=1000,dmode=0770,fmode=0770 Win_Ubuntu_shared /mnt-win-ubu-shared
```

I made it executable with `sudo nano /etc/systemd/system/mount_shared_folder.service`, then created a service to run it at startup: `sudo nano /etc/systemd/system/mount_shared_folder.service`, with the contents

```
[Unit]
Description=Mount VirtualBox Shared Folder
After=vboxadd.service
Requires=vboxadd.service

[Service]
Type=oneshot
ExecStart=/usr/local/bin/mount_shared_folder.sh
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Then, reloaded the Systemd and enabled the newly created service:

```
sudo systemctl daemon-reload
sudo systemctl enable mount_shared_folder.service
```

, which returned

```
Created symlink /etc/systemd/system/multi-user.target.wants/mount_shared_folder.service →
/etc/systemd/system/mount_shared_folder.service.
```

And I restarted the service with `sudo systemctl start mount_shared_folder.service`, and then the virtual machine itself. And it still doesn't work without punching in my password. So I decided that, for the time being, if I only have to input the password once, at system startup, that's fine and unlikely enough to affect the pipeline that I'm willing to drop it and move onto something more productive.

I added aliases to my .bashrc file to more easily access the folder, if nothing else. I just pasted into it:

```
alias inter-OS='cd /mnt-win-ubu-shared'
alias inter-OS_data='cd /mnt-win-ubu-shared/Datasets/'
```

Additionally, I installed conda environment name completion. I first tried installing the bash completion just to be sure it works properly with `sudo apt-get install bash-completion`

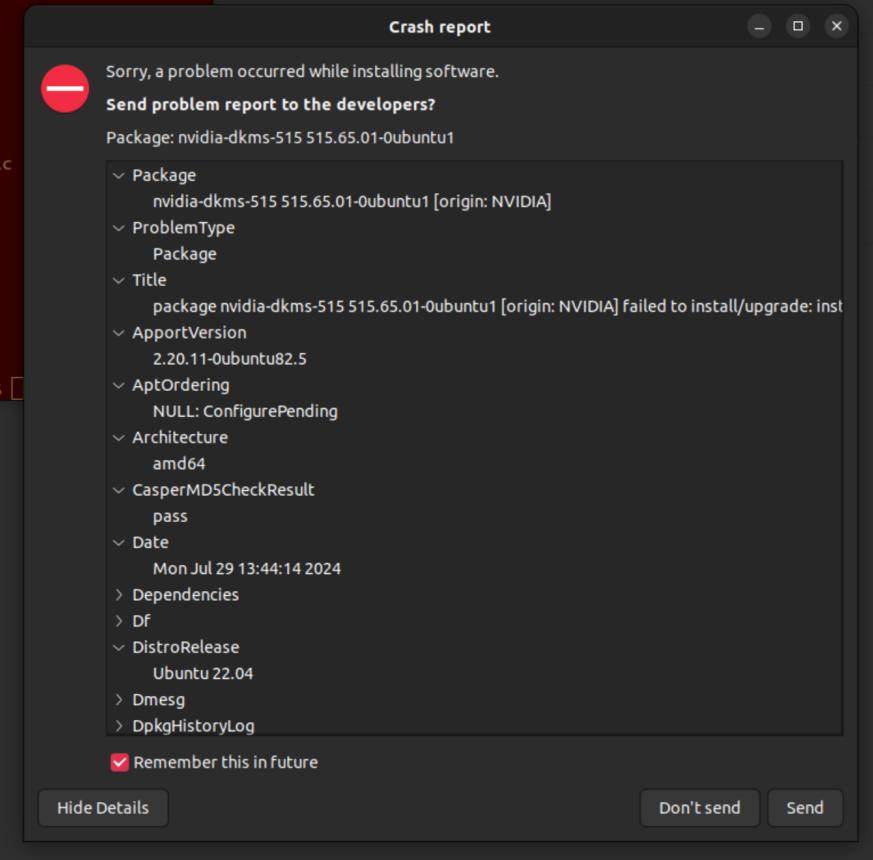
, but that ran into an error:

```
iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared/Datasets$ sudo apt update
cuda-11-7 depends on cuda-demo-suite-11-7 (>= 11.7.91); however:
  Package cuda-demo-suite-11-7 is not configured yet.

dpkg: error processing package cuda-11-7 (--configure):
 dependency problems - leaving unconfigured
dpkg: dependency problems prevent configuration of cuda:
  cuda depends on cuda-11-7 (>= 11.7.1); however:
    Package cuda-11-7 is not configured yet.

dpkg: error processing package cuda (--configure):
 dependency problems - leaving unconfigured
Processing triggers for initramfs-tools (0.140ubuntu13.4) ...
update-initramfs: Generating /boot/initrd.img-6.5.0-44-generic
Errors were encountered while processing:
 nvidia-dkms-515
 cuda-drivers-515
 cuda-drivers
 nvidia-driver-515
 cuda-runtime-11-7
 cuda-demo-suite-11-7
 cuda-11-7
 cuda
E: Sub-process /usr/bin/dpkg returned an error code (1)
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared/Datasets$
```

enomes
BI_files_NGS
ther Locations



The error seems to be related to the Nvidia and Cuda drivers...and I don't have the time to look further into this right now.

I simply proceeded with `conda install bash-completion` in the base environment. It seems to have completed error-free, and the autocompletion now works without errors.

Updating installation for the SRA toolkit

I had initially worked on the WSL and created the VirtualBox virtual machine later, so of course the SRA toolkit was not working any longer. I created a dedicated conda environment for it called `ncbi_SRA`, and installed the package from Bioconda using `conda install sra-tools` and updated it with `conda update sra-tools`

New environment, new life: creating dedicated bioinformatics environments using conda

Since I had all of the aforementioned issues with bcbio, I decided to abandon that approach, and, consequently, deleted the environment with `conda`, just to be on the safe side that I will have no issues later:

```
conda env remove --name bcbio-env
```

I then created a brand new, fresh environment called "NGS" in which to install the QC packages:

```
conda env --create NGS
```

Made a separate conda environment called "NGS" with `conda create --name NGS`

Using the shared folder between my Windows host and my Ubuntu virtual machine, I copied the FastQ files I had previously generated with WSL into the virtual machine.

Installing FastQC on my NGS conda environment within the Linux virtual machine

Perl and Java were already installed on my system (I suspect I installed them when installing conda and it creating its base environment), which is important because FastQC needs to run a small Perl script to find the binaries (executable files). The rest was as easy as 123: downloaded the archive, unzipped it, made the file called "fastqc" executable (`chmod u+x`) and ran it with `./fastqc`.

Added path to the FastQC folder to my PATH environment variable so that I can now only type `fastqc` from any folder and Linux still finds the binaries of fastqc and can open it:

```
sudo ln -s /home/iweber/Downloads/fastqc_v0.12.1/FastQC/fastqc /usr/local/bin/fastqc
```

This worked! I can now call `fastqc` from the command line.

Installing MultiQC on my NGS conda environment within the Linux virtual machine

Once more, this was extra simple from the command line:

```
conda install bioconda::multiqc , after which I could call it from the command line.
```

After that, I wanted to see what options are available for `multiqc`, so I ran `multiqc --help`. I got quite a number of options:

```
(NGS) iweber@iweber-VirtualBox:~/Downloads/fastqc_v0.12.1/FastQC$ multiqc --help

/// MultiQC 🔎 | v1.21

Usage: multiqc [OPTIONS] [ANALYSIS DIRECTORY]

MultiQC aggregates results from bioinformatics analyses across many samples
into a single report.
It searches a given directory for analysis logs and compiles a HTML report.
It's a general use tool, perfect for summarising the output from numerous
bioinformatics tools.
To run, supply with one or more directory to scan for analysis results. For
example, to run in the current working directory, use 'multiqc .'.

Main options
--force          -f  Overwrite any existing reports
--config         -c  Specific config file to load, after those in MultiQC
                     dir / home dir / working dir.
                     (PATH)
--cl-config      -Specify MultiQC config YAML on the command line
                     (TEXT)
--filename       -n  Report filename. Use 'stdout' to print to standard
                     out.
                     (TEXT)
--outdir         -o  Create report in the specified output directory.
                     (TEXT)
--ignore         -x  Ignore analysis files (GLOB EXPRESSION)
--ignore-samples -Ignore sample names (GLOB EXPRESSION)
--ignore-symlinks -Ignore symlinked directories and files
--file-list      -l  Supply a file containing a list of file paths to be
                     searched, one per row

Choosing modules to run
```

Choosing modules to run

--module -m Use only this module. Can specify multiple times.
(MODULE NAME)
--exclude -e Do not use this module. Can specify multiple times.
(MODULE NAME)

Sample handling

--dirs -d Prepend directory to sample names
--dirs-depth -dd Prepend *n* directories to sample names. Negative number to take from start of path.
(INTEGER)
--fullnames -s Do not clean the sample names (*leave as full file name*)
--fn_as_s_name Use the log filename as the sample name
--replace-names TSV file to rename sample names during report generation
(PATH)

Report customisation

--title -i Report title. Printed as page header, used for filename if not otherwise specified.
(TEXT)
--comment -b Custom comment, will be printed at the top of the report.
(TEXT)
--template -t Report template to use.
(default|gathered|geo|highcharts|sections|simple)
--sample-names TSV file containing alternative sample names for renaming buttons in the report
(PATH)
--sample-filters TSV file containing show/hide patterns for the report
(PATH)
--custom-css-file Custom CSS file to add to the final report (PATH)

(PATH)

--custom-css-file Custom CSS file to add to the final report (PATH)

Output files

--flat	-fp	Use only flat plots (<i>static images</i>)
--interactive	-ip	Use only interactive plots (<i>in-browser Javascript</i>)
--export	-p	Export plots as static images in addition to the report
--data-dir		Force the parsed data directory to be created.
--no-data-dir		Prevent the parsed data directory from being created.
--data-format	-k	Output parsed data in a different format. (tsv csv json yaml)
--zip-data-dir	-z	Compress the data directory.
--no-report		Do not generate a report, only export data and plots
--pdf		Creates PDF report with the ' <i>simple</i> ' template. Requires Pandoc to be installed.

MultiQC behaviour

--verbose	-v	Increase output verbosity. (INTEGER RANGE)
--quiet	-q	Only show log warnings
--strict		Don't catch exceptions, run additional code checks to help development.
--development,--dev		Development mode. Do not compress and minimise JS, export uncompressed plot data
--require-logs		Require all explicitly requested modules to have log files. If not, MultiQC will exit with an error.
--profile-runtime		Add analysis of how long MultiQC takes to run to the report
--no-megaqc-upload		Don't upload generated report to MegaQC, even if MegaQC options are found
--no-ansi		Disable coloured log output
--version		Show the version and exit.
--help	-h	Show this message and exit.

Creating a GitHub repository for the project

In order to be able to more easily version my work on the project and to allow my internship supervisor to gain insight into what I was doing, I created a GitHub repo for the newly minted "official project folder" (the inter-OS shared folder I created before). Naturally, I should have started a repository much longer ago, but, as the saying goes, "The best time to plant a tree was 20 years ago. The next-best time is now". I started working on this project while my bioinformatics course was still ongoing and we hadn't talked about git yet - now's that next-best time to catch up on this.

Making the repo under Linux

I first created an empty repo on GitHub and also a fine-grained access token for myself, giving myself all of the privileges necessary for using the repo. Then, from Ubuntu, I initialized a new git repo in the inter-OS shared folder with `git init` and pointed its head to the main branch:

```
git symbolic-ref HEAD refs/heads/main .
```

I added the online repo with

```
git remote add origin https://github.com/i-weber/Internship_project_RNAseq .
```

I added all of the files in the folder to the current staging area with `git add .`, and that might have been a mistake, given that the FastQ files are pretty large in their uncompressed state - the whole folder is around 170 GB *insert clenched teeth smiley*. On the side, I then read one can use the `top` command **in a new terminal window** to check resource usage in Linux, similar to Windows's task manager, and I saw that Git is using 70-80% of all resources of the virtual machine, so I gave it time. Next time, I will use `git add . --verbose` to get a better feel of the progress it is making through the files and folders. Also, can use `iostat -x 1` to investigate disk usage (install beforehand with `sudo apt-get install sysstat`) or `htop`, which apparently is `top` on steroids (also needs prior installation with `sudo apt-get install htop`).

So far, clocking at half an hour run time for the `add` command...let's see how long it takes in total.

At the one hour mark, I decided it was time to stop messing around, so I killed the process and proceeded to add the FastQ files and genomic files to the `.gitignore`. I created it with nano and added to it:

```
Adapters/  
Genomes/  
Nextflow/  
Software/  
Datasets/sra  
Datasets/Pre_eclampsia_mice/Pre_eclampsia_mice_fastq/
```

to avoid some of the largest and recoverable files from the push and commit.

For the future:

Note to self: add them without the slashes, I think git was looking for "empty" name items in these folders and not finding them, without selecting everything *inside* of the folders.

I then tried to push the .gitignore only, and, after completing the command line sign in, immediately got an error because I had previously set GitHub to block commits that might publicize my email address. So I found out that, in order to avoid this issue, I can use a no-reply address that GitHub itself creates for all of its users as my default email. I ran `git config --global user.email "163516184+i-weber@users.noreply.github.com"` to do so, and deactivated the checkbox in my GitHub email settings that supposedly protects one from publicizing their email in the commit metadata (I now know that this no-reply address will be used). And I could finally push the .gitignore file and see it on the website!

I then proceeded to add all other files with `git add . --verbose` at 16.48. It crashed around 17.05 because - and I should've thought of this! - I deleted a file from the folder at some point **facepalm**. One more thing learned...and restarted the same command at 17.06. Annnd it worked!

I proceeded to commit the changes with `git commit -m "All files up to date"`, which returned

```
[main 3a13867] All files up to date  
131 files changed, 98102 insertions(+), 1 deletion(-)
```

It failed, with an error message saying: "Enumerating objects: 129, done. Counting objects: 100% (129/129), done. Delta compression using up to 6 threads Compressing objects: 100% (123/123), done. error: unable to rewind rpc post data - try increasing http.postBuffer error: unable to rewind rpc post data - try increasing http.postBuffer error: RPC failed; HTTP 400 curl 92 Recv failure: Connection reset by peer send-pack: unexpected disconnect while reading sideband packet Writing objects: 100% (127/127), 2.04 GiB | 2.85 MiB/s, done. Total 127 (delta 18), reused 1 (delta 0), pack-reused 0 (from 0) fatal: the remote end hung up unexpectedly Everything up-to-date"

I set the postBuffer to 200 MB using `git config --global http.postBuffer 209715200`, and re-started the commit command at 23:20.

Aaand...it failed again. The message read just as above (I think - I lost the clipboard because I shut the virtual machine off too soon, apparently...).

My next approach was to try the Git Large File System extension (<https://git-lfs.com>). I used the command

..

```
curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh sudo bash
```

to install the package, as directed by the website that git-lfs directed me towards, <https://packagecloud.io/github/git-lfs/install>. That didn't cut it, and neither did trying to install git by using `git lfs install`, as I suspect that will only work once lfs is actually on the system (it gives an error saying it doesn't recognize the command). So I downloaded the tar archive, changed the directory to the Downloads folder, and extracted it with tar (v = verbose, x = extract - otherwise, tar tries creating an archive, z = indicate that archive is in gzip format, f = name to use for the extracted file or folder)

```
'tar -vxzf git-lfs-linux-amd64-v3.5.1.tar.gz'
```

I then navigated to the decompressed folder, made the install.sh script file inside of the folder executable with `chmod u+x install.sh`, then navigated to my inter-OS folder and ran the install.sh with sudo permissions (it told me I could not install it where I had downloaded it, the regular Downloads folder of Ubuntu, hence my choice to change to my folder). I tried now to use `git lfs install` and got the same message as before, Updated Git hooks. Git LFS initialized. It is now ready to use...but, in the end, does not seem so useful for my plans. I don't have individual files at the moment that are extremely large - I simply have folders with lots of relatively small files in them.

To see which files git is currently tracking, I used `git ls-files`, `git ls-tree -d HEAD` (directories only) and `git ls-tree -r HEAD` (directories and files contained therein) to make sure I am now really only tracking the files that I want to track, and not the FastQ files or genome files.

```
(base) iweber@iweber-VirtualBox:/mnt/win-ubuntu-shared$ git ls-tree -r HEAD
100644 blob 0ffed7141f6001530a770f7ec3a3e8533d99c
100644 blob e69de29b2d1d6434b8b29ae775ad8c2e48c5391
100644 blob a179d00b7db49fc9ed8cafcc762beb5278bcd2188
100644 blob 98353cdccff26d29212af41da6792907f0586a
100644 blob 46da2b3455d917d526fd4e26c6e46db0560d3b6c
100644 blob b7b85c975c169e1b08b72267ee2dc0433592f974
100644 blob e0cd473727f4717e05cff0558985d71401b2bd2b
100644 blob 2be6c9af1e2b75c3c1b075ee52bb7c09aa1fc8
100644 blob e993dc4f05fdcf74a9f0efea40a1dc03c0c932265
100644 blob 2188a15c0de0a4d7407b6e4e995e302a799a8a3
100644 blob 271ff6523d376745c06032e9bec813215179d9a
100644 blob a63f91ec8411be49e3fb6430fbfb1b284093845a1
100644 blob fdb69f84f4fb2a279596f4ca43a2ce3f141b2be6a
100644 blob a7076f98c161aea9c97cdc604c697cd9565834
100644 blob 2bf2b7281fc96a306923e1acf24134cad75ffab56
100644 blob 011784fae72d05af50b4f45521407989f4627314
100644 blob 206187ddbdd0156cb13a6893fb94ed282a1aa146
100644 blob 54b5d26c265f73a1c271027494abb3123bef274e
100644 blob a5be721b84be25956713c28ee026b46b4eb9b72e
100644 blob fa633c919f6816bla0833a1adfc763939cca298
100644 blob 1937c15ff7f89a32f3b34179d1ecba448d6008017
100644 blob 7a54a41b7c0b3b30085dfd94018db33b8ca52d0a
100644 blob 58766fdd2bf4ea09cd44e16da58568b236a0
100644 blob da9065aa4ad531169372d79b2805acc034d4913
100644 blob 1534512a9f7930f49b4939be52b63536212133e85
100644 blob c1f62da1a934b6c63d4b5c7b7d8a9cad922ec05c6
100644 blob ce9f9d352028066dfbeecf9f899ad50120e6b917
100644 blob 2c17a0d35018c97b8d4dd53c3d3cc815362b6f0b
100644 blob 8971c7413dfa513ea5c9b1143cef1bf822b762
100644 blob 0bb58f5cbe42a3e1620f565c074e962507091
100644 blob 9f1513f346cdecda487fe45f3aa5e45e3b1cf626
100644 blob 279aa7008335132f255e411a58ffccf5aab4931d6
100644 blob dfe6a72a3c52e7d1a4c472774166f7f544b6333f
100644 blob 3c15cad61e6e84755b2c33067a99420c53a87c8
100644 blob 1a748edd6b96d5517f3ef38c96dcbe547a505b8b
100644 blob c929271f51bf6384b23efc26982717feeb64b7b9
100644 blob 8c30344bb1034cd42b017a25d58421f507302
100644 blob d318b6c1568a59721545fedcb3459819388c9fc8
100644 blob 7158bda53c445e61518e6b6f20f52186f89497e
100644 blob 707fcf79628e8c66e9d701f69f9b1e4bab494b4
100644 blob 18158e1f45bc034e20781b98b19d8367c9dbf8c8
100644 blob 91769a22e5571f014485d44d3028dc8b75f245
100644 blob 686134f691269ab067c71c5a634024e41ddb1b2
100644 blob 7c3b1fb55887921262912ec8a3135025b1afc243
100644 blob e080a1007475d0aaea1818852f3d7683e3503962
100644 blob b8f1c8e90cfbe7b25f4defdf3f627f69f774864
100644 blob ce5cd608d0adb89fe4effc7ae72527bd8535ba28
100644 blob 60b782719eb5453437e415540ab94d0b22b195c
100644 blob 90acb1a09ac468624a064812a99955d455380ca9
```

It looks like I am indeed now avoiding the really large files. I added again everything with `git add .`, then checked the status. It said it has nothing to commit, confirmed when I still tried to commit with `git commit -m "Add remaining changes"` so I went on to push the changes with `git push origin main`. The runtime started at 13:30, this time the "Writing objects" buffer went up to 10%, as opposed to the 7% it used to go up to yesterday. But it crashed again, with the error

"Writing objects: 100% (140/140), 2.04 GiB | 1.08 MiB/s, done.

Total 140 (delta 26), reused 1 (delta 0), pack-reused 0

fatal: the remote end hung up unexpectedly

Everything up-to-date",

so I think I need to reduce the buffer size again. Did it with `git config --global http.postBuffer 157286400`, let's see if this is of any help...nope.

I went through all of the files and removed anything that is not absolutely critical. I then reset the git staging area, and also removed any of the cached versions of any files with `git rm -r --cached .`. Then, I used `find . -type f` to find all files that could be added at that moment. I checked the sizes - nothing is above 100 MB at the moment, not even folders.

Tried again to add and push in small batches, e.g., in the Presentations folder, only some of the PNG images. It stopped at 96% in the writing process and would not progress. I increased the http buffer with `git config --global http.postBuffer 524288000` to 500 MB, but it then got stuck at 76% in the writing process.

This is when I gave up and decided to do a fresh start. I deleted the .git folder, everything in the online version of the repository, and all of the data inside the inter-OS shared folder, since I had it backed up in a different location anyway. I re-initialized a git repo inside of the same folder, added the remote repo I had previously created and which was basically empty at this point, and then started adding the files and folders piecemeal back into the inter-OS shared folder, adding them to the staging area, committing, and pushing after each smaller batch of files. Worked like a charm, took maybe 15 min and solved all of my problems. Presentations folder, at 33 MB, took from 21:36 to 21:37 - nothing like the full hour it took in the previous days and earlier today!

Getting the repo to work under Windows as well

Since I later want to take the data that the analysis pipeline generates and work with it under Windows, where I have R fully set up and ready to rumble, I want to have access to the repo from this OS as well. Additionally, I read that Visual Studio Code allows for a very nice integration with GitHub, including an extension that shows when commits were made and how different branches of a project relate to one another (Git Graph). So I opened the repo in Visual Studio Code, installed the extension, and now can open it with `Ctrl+Shift+P --> Git Graph: View Git Graph` from the dropdown menu. Now I can very easily track the changes to the repo visually, and also have access to the repo from the Windows side - if I select the folder where the repo is in the Explorer part of VSC, I can perform all of the usual git commands from the terminal.

Trying to use Git LFS to track FastQ and other large files

As I want to track changes to my FastQ files as well once I start trimming them and processing them, but also to other large files, such as bam files, that I will generate further in the process, I set up Git LFS to track such files.

I had previously installed git LFS on my VM, and I now went to my inter-OS folder and hit `git lfs install` to initialize it (yes, the naming is somewhat confusing).

I then proceeded to track .fastq files by using `git lfs track "*.fastq"` and it returned `Tracking "*.fastq"`. I checked which files are now tracked by git LFS using `git lfs ls-files`, and it returned...nothing xD.

```
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git lfs ls-files
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ █
```

I checked whether the `.gitattributes` file was correctly created, and it seems so - it is present in my repo and contains `*.fastq filter=lfs diff=lfs merge=lfs -text`, as it should.

I pasted my FastQ files into a raw data folder (`Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/`), together with the FastQC and MultiQC files that resulted from their analysis. I then created a `.gitignore` file to, for now, ignore the FastQs but still be able to stage, commit and push the new directory structure. After doing the usual git steps, I had an updated online repo including the FastQC and MultiQC files, but not the `.fastq` ones.

I removed the `.fastq` rule from the `.gitignore` file. When hitting `git status` now, the only files left unchanged were indeed the newly added FastQ ones.

I ran the `git lfs track "*.fastq"` command again, and got a message saying `"*.fastq" already supported`.

I next added the first FastQ file to the staging area, `SRR13761520_1.fastq`. I started at 12.07, and it took around 5 min to finish. It also took a moment to commit, but, when I tried pushing it to the repo, it failed, saying

```
[f12181f07712304b1aed2f1876165f549942ad8144457878ca37fbe90356a731] Size must be less than or equal to 2147483648: [422] Size
must be less than or equal to 2147483648
error: failed to push some refs to 'https://github.com/i-weber/Internship\_project\_RNAseq'
```

It sounds to me like git LFS is still not managing the FastQ files as it should. In the previous step, Git tried to push a file larger than 270 MB to the repo, which is clearly not what it should be doing, if git LFS were working.

What's happening here? there's some logical connection missing.

The solution may involve using `git lfs migrate` to force the objects that are now managed by git to be managed by git LFS instead. I used `git lfs ls-files --all` to understand if git lfs was already tracking anything

2:17 git status

2:19: It looks like the file that now appears to already be tracked by git lfs does not appear any longer under "untracked files".

Tried adding the pair of this FastQ file (the one ending in `_2`) to the staging area and committing. then running `git lfs ls-files --all` to see if git LFS takes over this file after the commit.

```
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761521_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761521_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761522_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761522_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761523_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761523_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761524_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761524_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761525_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761525_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761526_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761526_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761527_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761527_2.fastq

nothing added to commit but untracked files present (use "git add" to track)
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git add Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_2.fastq
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git commit -m "Pair of first FastQ file used to test if git LFS works"
[main b5b4cc7] Pair of first FastQ file used to test if git LFS works
1 file changed, 3 insertions(+)
create mode 100644 Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_2.fastq
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git lfs ls-files --all
f12181f077 * Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_1.fastq
d3e4f0d7f7 * Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_2.fastq
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761521_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761521_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761522_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761522_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761523_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761523_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761524_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761524_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761525_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761525_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761526_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761526_2.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761527_1.fastq
    Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761527_2.fastq

nothing added to commit but untracked files present (use "git add" to track)
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$
```

And success! The files now don't show up in the regular git tracked files, but do show up as managed by git LFS!

So, what have I learned? Git LFS needs a definition of what kinds of files to look for (the pattern specified in the .gitattributes file). It then keeps a lookout for them, and, when such files are added to the staging area, it picks up on them and creates so-called pointers (pointer files) to these files and the changes in them, which are recorded in the repository instead of the files themselves. A bit like a group of elite dogs getting a sniff at a fabric scrap from persons they should track, then alerting when those persons are nearby, staying with their snouts pointed towards them.

However, they still will not be pushed online. So, even with git LFS, the problem seems to be on the side of GitHub.

```
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git lfs ls-files --all
f12181f077 * Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_1.fastq
d3e4f0d7f7 * Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_2.fastq
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git push origin main
Username for 'https://github.com': i-weber
Password for 'https://i-weber@github.com':
[�12181f07712304b1aed2f1876165f549942ad8144457878ca37fbe90356a731] Size must be less than or equal to 2147483648: [422] Size must be less than or equal to 2147483648
[�3e4f0d7f716d1839f63a3e02f3e5caa6597304d6edb362ee9930f877a] Size must be less than or equal to 2147483648: [422] Size must be less than or equal to 2147483648
error: failed to push some refs to 'https://github.com/i-weber/Internship_project_RNAseq'
```

ChatGPT says that GitHub has a maximal limit of 2 GB even for files managed by git LFS, and this seems to fit the number the push command returns (2147483648 bytes is 2.14 GB).

So I guess I have to add these files to .gitignore and be done with them...

I added *.fastq to my .gitignore file, then used git rm --cached *.fastq to remove any previously tracked fastq files. That took a few minutes in which git-lfs was using a hefty portion of my CPU (77%). It seems to have worked:

```
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git rm --cached *.fastq
rm 'Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_1.fastq'
rm 'Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_2.fastq'
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$
```

I checked the git status, just in case:

```
(base) iweber@iweber-VirtualBox:/mnt-win-ubuntu-shared$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_1.fastq
    deleted:   Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_2.fastq
```

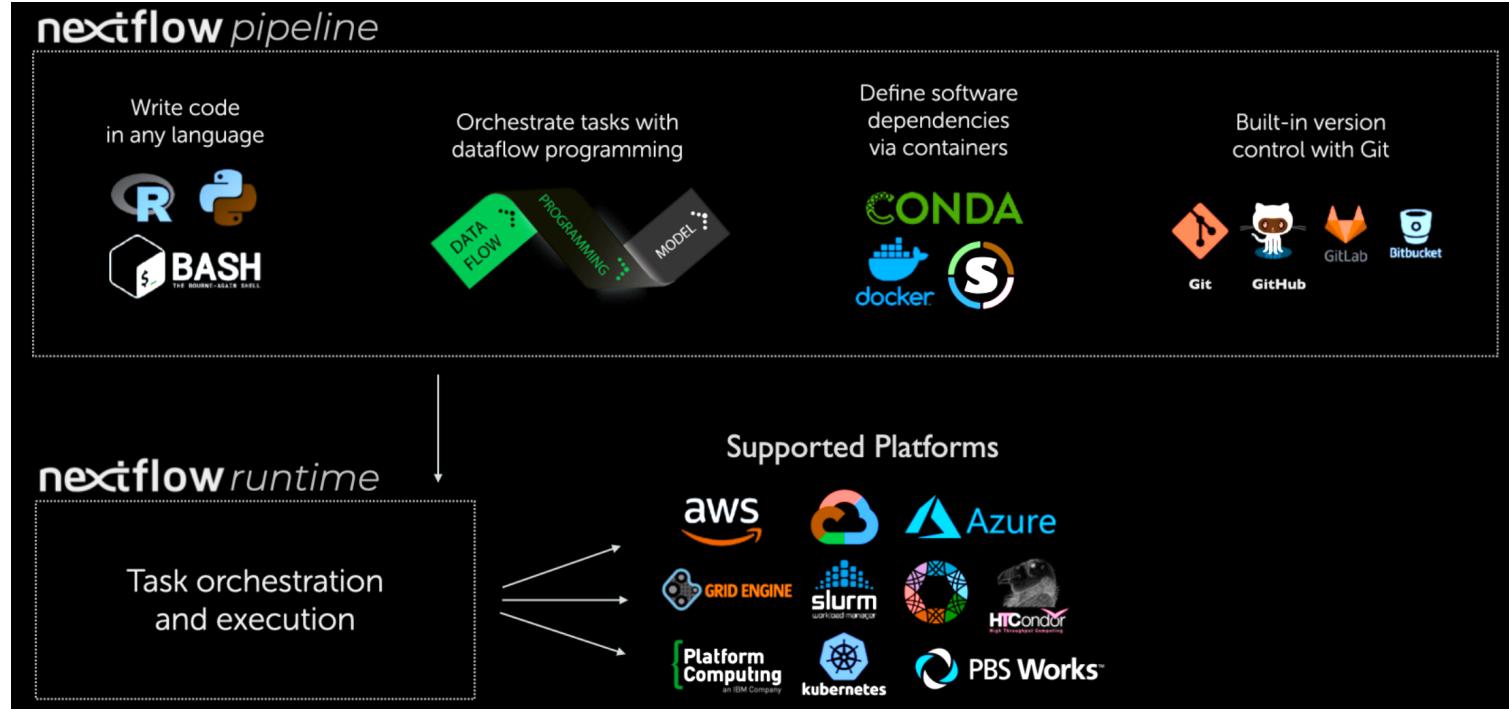
It looks like the next commit should delete these files from my staging area, without touching the files per se in my computer. Tried it out, and, indeed, I could still see my files in my file system using the file explorer and ls -lat .

I also removed `*.fastq` files from the git LFS tracking using `git lfs untrack "*.fastq"`. Got a message in return saying `Untracking "*.fastq"`, and saw that the `.gitattributes` file is now also empty again.

Important to know for the future:

Apparently, [trying to push files from Windows to GitHub that are larger than 4 GB truncates them and causes them getting corrupted!](#)

What is Nextflow?



from the [Nextflow training page](#)

a [domain-specific programming language](#), as opposed to C++, Python, etc, which are general-purpose programming languages

https://github.com/chlazaris/Nextflow_training/blob/main/nextflow_cheatsheet.md

https://github.com/danrlu/Nextflow_cheatsheet/blob/main/nextflow_cheatsheet.pdf

Setting up Nextflow to run bioinformatic analysis pipelines

Installing Nextflow and nf-core

First, installed Java:

```
sudo apt install default-jre
```

I created a dedicated environment in which I will work with Nextflow, that I called as such, using

```
conda create Nextflow
```

After activating it, I installed the actual Nextflow workflow manager with

`conda install -c bioconda nextflow`, and it worked like a charm. Then, to get access to the curated Nextflow pipelines for bioinformatic analyses, I installed nf-core in the same environment with `conda install nf-core`, which also completed without any overt errors.

I also activated shell completions for nf-core by adding the [recommended](#) command to my `.bashrc` file:

```
eval "$_NF_CORE_COMPLETE=bash_source nf-core"
```

and restarted my shell.

And now I could easily list all of the curated pipelines available in nf-core:

```
(base) iweber@iweber-VirtualBox:~$ conda activate Nextflow  
(Nextflow) iweber@iweber-VirtualBox:~$ nf-core list
```

NF-CORE



nf-core/tools version 2.14.1 - <https://nf-co.re>

Pipeline Name	Stars	Latest Release	Released	Last Pulled	Have latest release?
funcscan	62	1.1.6 dev	3 weeks ago yesterday	-	-
variantbench	6	1.4.1	5 months ago	-	-
demultiplex	37	2.10.0 dev	4 weeks ago yesterday	-	-
ampliseq	166	2.2.1 dev	4 days ago	-	-
pairlegenome	0	2.5.2 dev	4 weeks ago	-	-
crisprseq	23	2.0.0 dev	2 years ago 5 months ago	-	-
eager	131	2.2.0 dev	3 days ago	-	-
chipseq	177	2.3.1 dev	3 months ago	-	-
nascent	14	3.4.2 dev	3 months ago	-	-
oncoanalysis	22	3.14.0 dev	3 days ago	-	-
denovotranscript	0	4.1.0 dev	3 days ago	-	-
rangeland	3	4.1.0 dev	3 days ago	-	-
magmap	1	4.1.0 dev	3 days ago	-	-
smrnaseq	70	4.1.0 dev	3 months ago	-	-
sarek	357	4.1.0 dev	1 week ago	-	-
mcmicro	4	4.1.0 dev	2 months ago	-	-
fastquorum	13	4.1.0 dev	7 months ago	-	-
rnaseq	824	4.1.0 dev	1 weeks ago	-	-
multipleseq	11	4.1.0 dev	1 months ago	-	-
demo	1	4.1.0 dev	1 weeks ago	-	-
phaseimpute	16	4.1.0 dev	2 months ago	-	-
airrflow	46	4.1.0 dev	1 weeks ago	-	-

The pipeline I am interested in is called `rnas splice`, and we'll get back to that in a moment.

Containerization with Docker for nf-core/Nextflow?

What caught my eye when reading more about nf-core was the following:

Pipeline software

An analysis pipeline chains the execution of multiple tools together. Historically, all tools would have to be manually installed — often a source of great frustration and a key step where reproducibility between analyses is lost. nf-core pipelines utilise the built-in support for software packaging that Nextflow offers: all can work with Docker and Singularity, and most pipelines also support Conda.

To use any of the below, simply run your nf-core pipeline with `-profile <type>`. For example, `-profile docker` or `-profile conda`.

- Docker

- Typically used locally, on single-user servers, and the cloud
- Analysis runs in a *container*, which behaves like an isolated operating system
- Typically requires system root access, though a “*rootless mode*” is available

- Singularity

- Often used as an alternative to Docker on HPC systems
- Also runs *containers*, and can optionally create these from Docker images
- Does not need root access or any daemon processes

- Apptainer

- Open source version of Singularity (split from Singularity in 2021)

- **⚠ Warning**

Currently, nf-core pipelines run with `-profile apptainer` will build using docker containers instead of using pre-built singularity containers.

To use the singularity containers, use `-profile singularity` instead. This works because `apptainer` simply defines `singularity` as an alias to the `apptainer` command.

- Podman, Charliecloud and Shifter

- All alternatives to Docker, often used on HPC systems

- Conda

- Packaging system that manages environments instead of running analysis in containers
- Poorer reproducibility than Docker / Singularity
 - There can be changes in low-level package dependencies over time
 - The software still runs in your native operating system environment and so core system functions can differ

- Mamba

- A faster reimplementation of Conda

I started working with conda because everyone in the bioinformatic community swears by it, and so did the economics researchers that I had my very first Python course with. It has served me very well so far, but I do also know [quote Adi Dilita] that software developers prefer to work with Docker, Singularity, or Kubernetes to create containers so that their software can always be run, that is, at any time, on any machine. I read more about how Docker, Conda, and Nextflow relate to one another and found out that conda is used so widely in the bioinformatic/scientific community because it integrates seamlessly with Python and R and is a more lightweight solution for reproducibility. This is due to the fact that it manages only the dependencies specifically required by these programming languages. However, Docker containers also incorporate information about the OS and all of the apps and packages installed on it that are required to run a specific program/application. In essence, Docker containers mirror the environment that the developer of a particular software worked in in order to program that software/app etc and isolate this mirror image within another user's OS. This increases the degree of reproducibility dramatically BUT is also more bulky, because a Docker container then stores all of this extra information related to the OS.

Since I want to try out Docker AND because it might be the less buggy option here, I decided to install it on my Linux virtual machine.

As per the instructions on the website, I first ran

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

to install the required packages....and it doesn't work. What I get is:

```
Get:1 file:/var/cuda-repo-ubuntu2004-11-7-local InRelease [1.575 B]
Get:1 file:/var/cuda-repo-ubuntu2004-11-7-local InRelease [1.575 B]
Hit:2 http://de.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Get:4 http://de.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:5 https://packages.microsoft.com/repos/code stable InRelease
Hit:6 http://de.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:7 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Get:8 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:9 https://packagecloud.io/github/git-lfs/ubuntu jammy InRelease
Fetched 257 kB in 2s (139 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.16).
The following packages were automatically installed and are no longer required:
 libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
8 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Setting up nvidia-dkms-515 (515.65.01-0ubuntu1) ...
update-initramfs: deferring update (trigger activated)
```

A modprobe blacklist file has been created at /etc/modprobe.d to prevent Nouveau from loading. This can be reverted by deleting the following file:
/etc/modprobe.d/nvidia-graphics-drivers.conf

A new initrd image has also been created. To revert, please regenerate your initrd by running the following command after deleting the modprobe.d file:
`/usr/sbin/initramfs -u`

```
*****
*** Reboot your computer and verify that the NVIDIA graphics driver can ***
*** be loaded. ***
*****
```

```
INFO:Enable nvidia
DEBUG:Parsing /usr/share/ubuntu-drivers-common/quirks/lenovo_thinkpad
DEBUG:Parsing /usr/share/ubuntu-drivers-common/quirks/put_your_quirks_here
DEBUG:Parsing /usr/share/ubuntu-drivers-common/quirks/dell_latitude
Removing old nvidia-515.65.01 DKMS files...
Deleting module nvidia-515.65.01 completely from the DKMS tree.
Loading new nvidia-515.65.01 DKMS files...
Building for 6.5.0-44-generic
Building for architecture x86_64
```

```

Building initial module for 6.5.0-44-generic
ERROR: Cannot create report: [Errno 17] File exists: '/var/crash/nvidia-dkms-515.0.crash'
Error! Bad return status for module build on kernel: 6.5.0-44-generic (x86_64)
Consult /var/lib/dkms/nvidia/515.65.01/build/make.log for more information.
dpkg: error processing package nvidia-dkms-515 (--configure):
    installed nvidia-dkms-515 package post-installation script subprocess returned error exit status 10
dpkg: dependency problems prevent configuration of cuda-drivers-515:
  cuda-drivers-515 depends on nvidia-dkms-515 (>= 515.65.01); however:
    Package nvidia-dkms-515 is not configured yet.

dpkg: error processing package cuda-drivers-515 (--configure):
  dependency problems - leaving unconfigured
dpkg: dependency problems prevent configuration of cuda-drivers:
  cuda-drivers depends on cuda-drivers-515 (= 515.65.01-1); however:
    Package cuda-drivers-515 is not configured yet.

dpkg: error processing package cuda-drivers (--configure):
  dependency problems - leaving unconfigured
dpkg: dependency problems prevent configuration of nvidia-driver-515:
  nvidia-driver-515 depends on nvidia-dkms-515 (= 515.65.01-0ubuntu1); however:
    Package nvidia-dkms-515 is not configured yet.

dpkg: error processing package nvidia-driver-515 (--configure):
  dependency problems - leaving unconfigured
dpkg: dependency problems prevent configuration of cuda-runtime-11-7:
  cuda-runtime-11-7 depends on cuda-drivers (>No apport report written because the error message indicates its a followup error from a previous failure.
                                              No apport report written because the error message indicates its a followup error from a previous failure.
                                              No apport report written because MaxReports is reached already
                                              No apport report written because MaxReports is reached already
                                              No apport report written because MaxReports is reached already
                                              No apport report written because MaxReports is reached already
                                              No apport report written because MaxReports is reached already
                                              = 515.65.01); however:
    Package cuda-drivers is not configured yet.

dpkg: error processing package cuda-runtime-11-7 (--configure):
  dependency problems - leaving unconfigured
dpkg: dependency problems prevent configuration of cuda-demo-suite-11-7:
  cuda-demo-suite-11-7 depends on cuda-runtime-11-7; however:
    Package cuda-runtime-11-7 is not configured yet.

dpkg: error processing package cuda-demo-suite-11-7 (--configure):
  dependency problems - leaving unconfigured
dpkg: dependency problems prevent configuration of cuda-11-7:
  cuda-11-7 depends on cuda-runtime-11-7 (>= 11.7.1); however:
    Package cuda-runtime-11-7 is not configured yet.
  cuda-11-7 depends on cuda-demo-suite-11-7 (>= 11.7.91); however:
    Package cuda-demo-suite-11-7 is not configured yet.

dpkg: error processing package cuda-11-7 (--configure):
  dependency problems - leaving unconfigured
dpkg: dependency problems prevent configuration of cuda:
  cuda depends on cuda-11-7 (>= 11.7.1); however:
    Package cuda-11-7 is not configured yet.

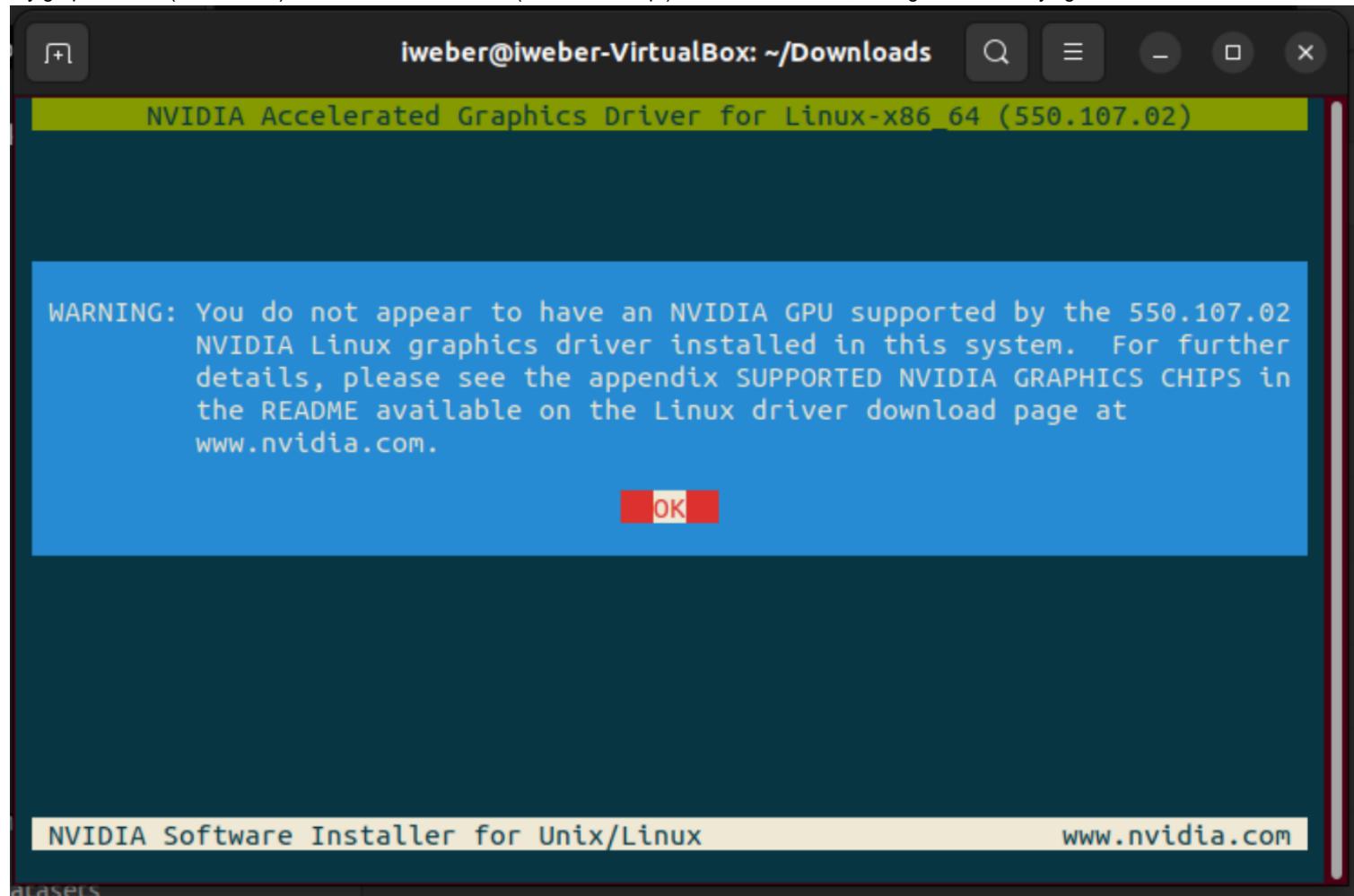
dpkg: error processing package cuda (--configure):
  dependency problems - leaving unconfigured
Processing triggers for initramfs-tools (0.140ubuntu13.4) ...
update-initramfs: Generating /boot/initrd.img-6.5.0-44-generic
Errors were encountered while processing:
  nvidia-dkms-515
  cuda-drivers-515
  cuda-drivers
  nvidia-driver-515
  cuda-runtime-11-7
  cuda-demo-suite-11-7
  cuda-11-7
  cuda
E: Sub-process /usr/bin/dpkg returned an error code (1)

```

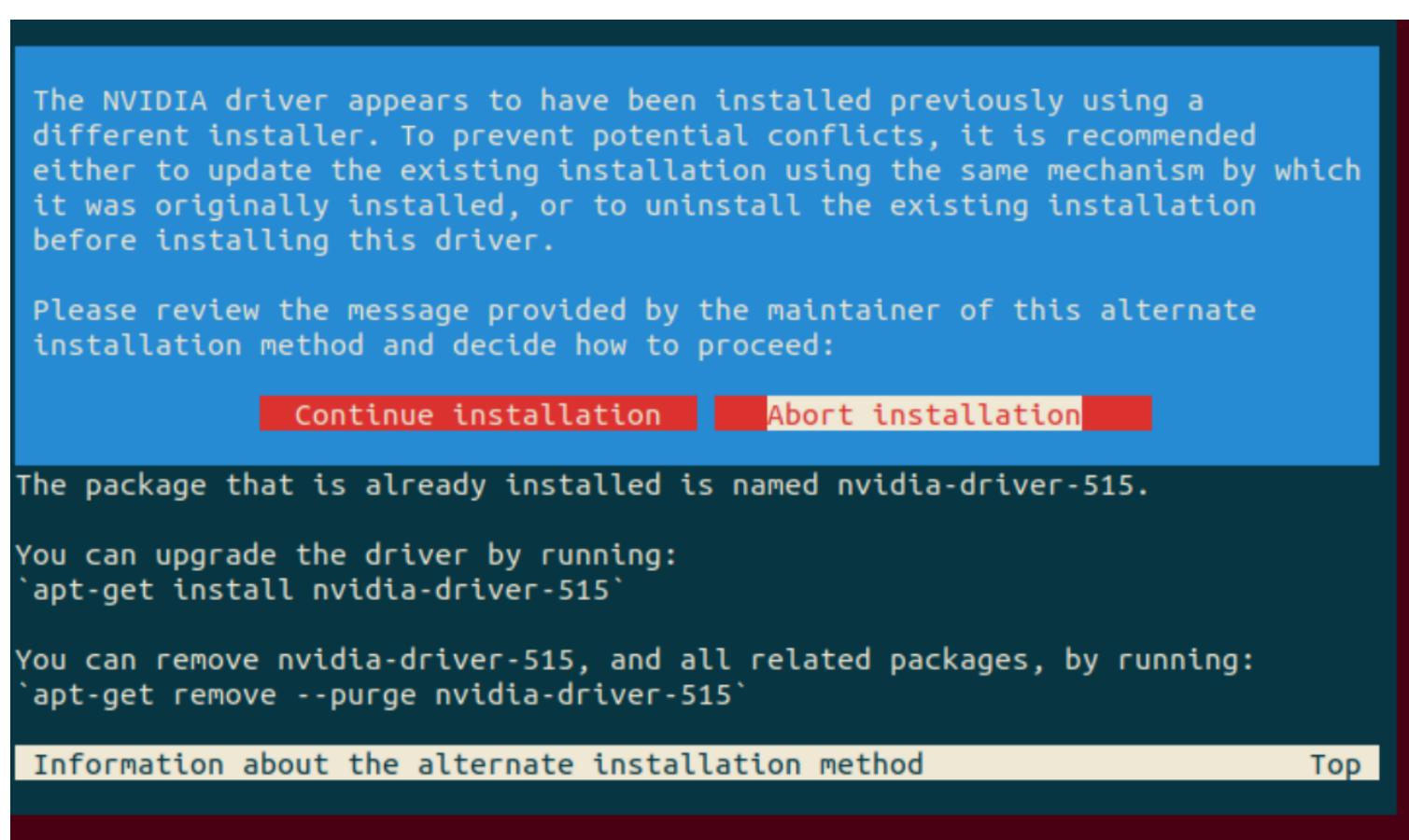
I went online to try and find what all of these issues with Nvidia and Cuda are about, and, sure enough, I found an [Nvidia page](#) about it.

I went through the preliminary checks. Yes, I have an OS version that's supported, yes, I have a GPU that supports Cuda. I also have gcc installed and uname -r returned "6.5.0-44-generic".

I tried finding this mlnx_ofed driver to install, but it was not available anywhere on the NVIDIA pages. I downloaded the dedicated Linux 64-bit driver for my graphics card (RTX A4000) from the NVIDIA website (it's a bash script) and ran it with sudo, but I got an error saying



Upon further reading, I saw that Ubuntu is a bit peculiar in the way in which it uses NVIDIA drivers and that, for some stability reasons, it has its own driver, based on the current driver's predecessor (number 535 instead of 550). But, after hitting OK on this error message, I got another one saying:



I hit "abort installation" and peeked at whether version 515 is indeed the one installed on my system using `ubuntu-drivers list`. Annnd all I got was

```
(base) iweber@iweber-VirtualBox:~/Downloads$ ubuntu-drivers list  
open-vm-tools-desktop  
(base) iweber@iweber-VirtualBox:~/Downloads$
```

As far as I understand, the driver list is different from what a normal, non-virtual OS would have.

I tried upgrading the driver by running `sudo apt-get install nvidia-driver-515` ...annnd got the exact same big stream of errors as above, which suggest rebooting computer and making sure NVIDIA graphics driver can be loaded. This seems to be a general issue with trying to run Cuda on a virtual machine, as, as far as my meagre understanding takes me, the virtual machine does not have direct access to the GPU through the host, especially not on VirtualBox virtual machines (apparently, VMWare is somewhat better, but that's not what I have).

I'm considering trying Singularity instead. Singularity is the containerization app often used on high-performance computing clusters (HPCs), and it makes things a little bit easier because it also does not require root access privileges.

...apparently, Singularity has changed names and is now called Apptainer. I started following the [installation instructions](#), and already the very first step for a setuid installation, `sudo apt install -y software-properties-common` immediately threw the same sequence of errors about NVIDIA and Cuda as above. So I decided the time has come to see if I can clean that up.

Before doing so, I backed up all of my conda environments into my GitHub repository. I created a new folder "Conda_environment_yamls", and pushed that to GitHub

I also created a snapshot of my virtual machine in its current state using VirtualBox, and saved my .vbox file in a safe location. Additionally, I saved the entire virtual machine folder in another location, and exported my virtual machine as an OVF/OVA file (in VirtualBox, File --> Export Appliance). Finally, I made a full clone of my virtual machine to test changing the drivers in, so that I can always just remove it and go back to my unchanged version, should major problems arise (did that also from VirtualBox from the machine options).

It seems like I got myself into another technical rabbit hole here: apparently, the root of the problems is that VirtualBox cannot take the GPU available to my Windows host and pass it through to my virtual machine. This might just be what is causing all of the issues: given the errors the Docker installation throws, it probably needs to access the GPU itself, because it helps it speed up computation quite a lot (inasmuch as my NVIDIA RTX A4000 can do that - it's a good GPU, not the top of the tops, but surely better than the virtualized version that VirtualBox creates). And because the Docker containers are so resource-intensive, this is likely why it won't work without it.

I can likely transfer my virtual machine from VirtualBox to another virtual machine software, VMWare, but that will likely require some more tweaking as well. For the time being, I will try running the pipeline as is in Nextflow under Conda, and then see about anything else.

Transferring entirety of virtual machine to VMWare

I took a snapshot of my current VM (2024-08-01) using VirtualBox, and, in the same program, hit "Export appliance" for this very same machine from the File menu. Started around 18:40, reading about the pipeline in parallel. It's mighty slow - at 19.05, so 25 min later, it was at barely 2%, meaning this will probably take until Saturday...

18:40 - start

19:05 - 2 % --> 1%/12.5 min

19:42 - 4 % --> 1%/18.5 min

21:00 - 7 % --> 3%/1h15, eq to 1%/25 min

...somehow, it seems to be slowing down quite a lot. Maybe because I opened more apps. Should be faster over night, when I'm not actively working on the laptop.

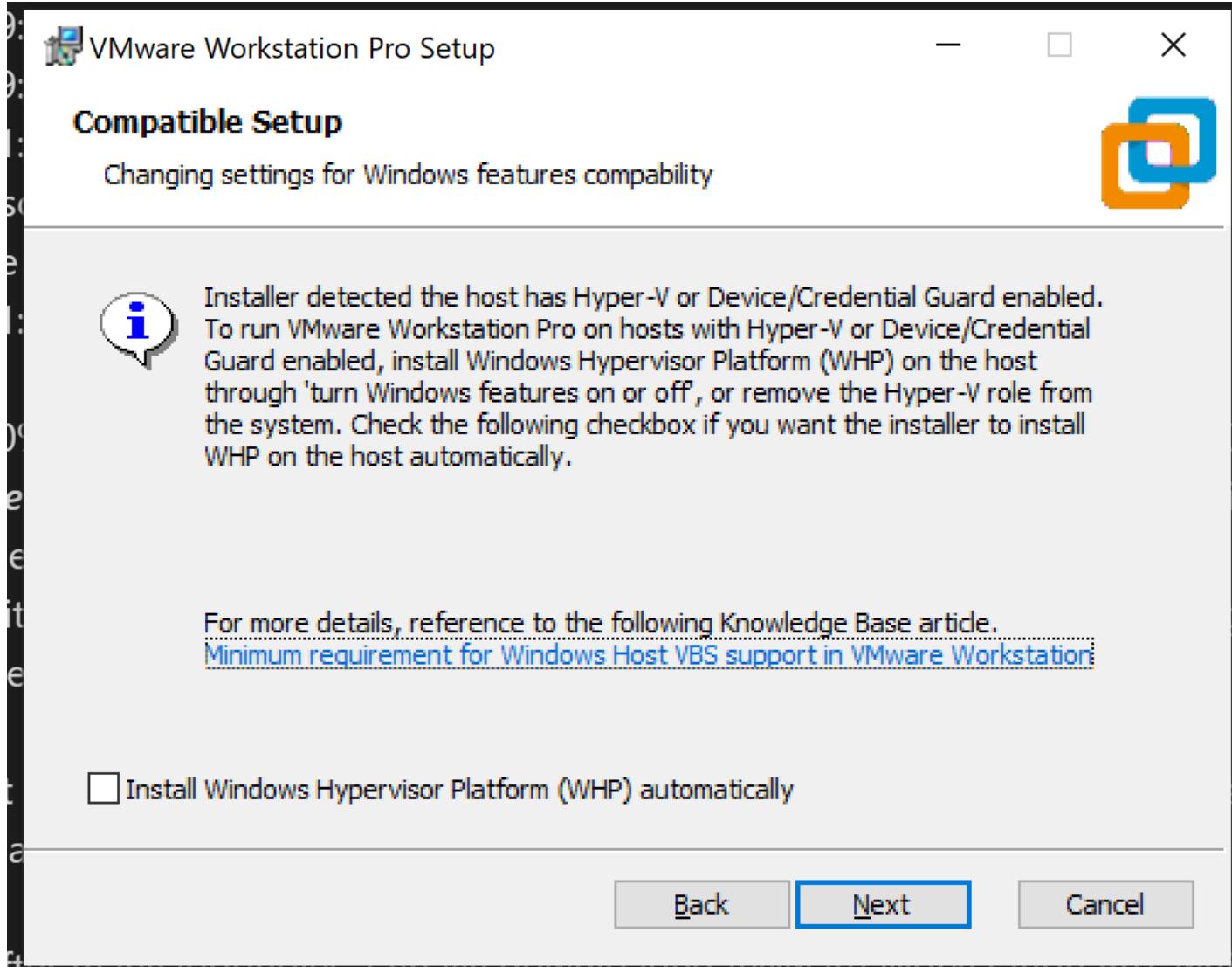
21:42 - 10% --> 7%/42 min, eq to 1%/6 min

90% more to go. At best, this should take 6x90 min = 9 h. At worst, 25x90 min = 37.5 h **insert clenched teeth smiley here** Of course, I closed any app that might be eating any of my laptop's memory/processing power, but the virtual machine was only using 16-30% of my CPU to begin with, so maybe that's not the issue here. Maybe, as with many other things, VirtualBox is rather inefficient in its resource utilization...

At around 2 AM, it had finally finished creating the OVF file (the file format of the so-called virtual machine appliance I just created, which is an [universal format]([How To Convert Virtual Machines Between VirtualBox and VMWare \(howtogeek.com\)](#)) and can be shared between different VM software).

After many loopholes, I finally downloaded VMWare Workstation Pro, which, to my joy, is now available for free for personal use since May of this year.

During the installation of VMware Workstation Pro, I got a message saying



I researched this further and found that, in my Windows Features, Hyper-V was actually switched off. However, since I had previously used WSL, I suspect this is not entirely true. I went on to fully switch it off in an elevated command prompt using

```
bcddedit /set hypervisorlauchtype off
```

Seconds later, I realized it was stupid to not make a Windows restore point right before that, and I tried reverting the command using "on" instead of "off", but it just gave me an error message saying "The integer data is not valid as specified. Run "bcdedit /?" for command line assistance. The parameter is incorrect.". I asked ChatGPT and found I was supposed to run

```
bcddedit /set hypervisorlauchtype auto
```

which, indeed, completed successfully. I swiftly created another restore point. Additionally, I updated my bootable Windows USB drive, which I had created some time back, to be able to easily boot the system in case I suddenly get a BSOD (this happened to me some years back, on the second day of owning this laptop, when I wanted to set up VirtualBox...better safe than sorry).

I also downloaded the VMware OVF Tool to make sure that the OVF I previously generated with VirtualBox is compatible with VMware Workstation Pro. I added its installation directory to my PATH variable under Windows, and could then access it from my command line in PowerShell.

I tried converting the old virtual machine OVA file to a VMware-compatible type using " ovftool ubuntu22x64.ova E:\VMware_ubuntu_VM " (into a new directory I created for the purpose), but got an error: "Opening OVA source: ubuntu22x64.ova Opening VMX target: E:\VMware_ubuntu_VM Error: OVF Package is not supported by target: - Line 25: Unsupported hardware family 'virtualbox-2.2'. Completed with errors "

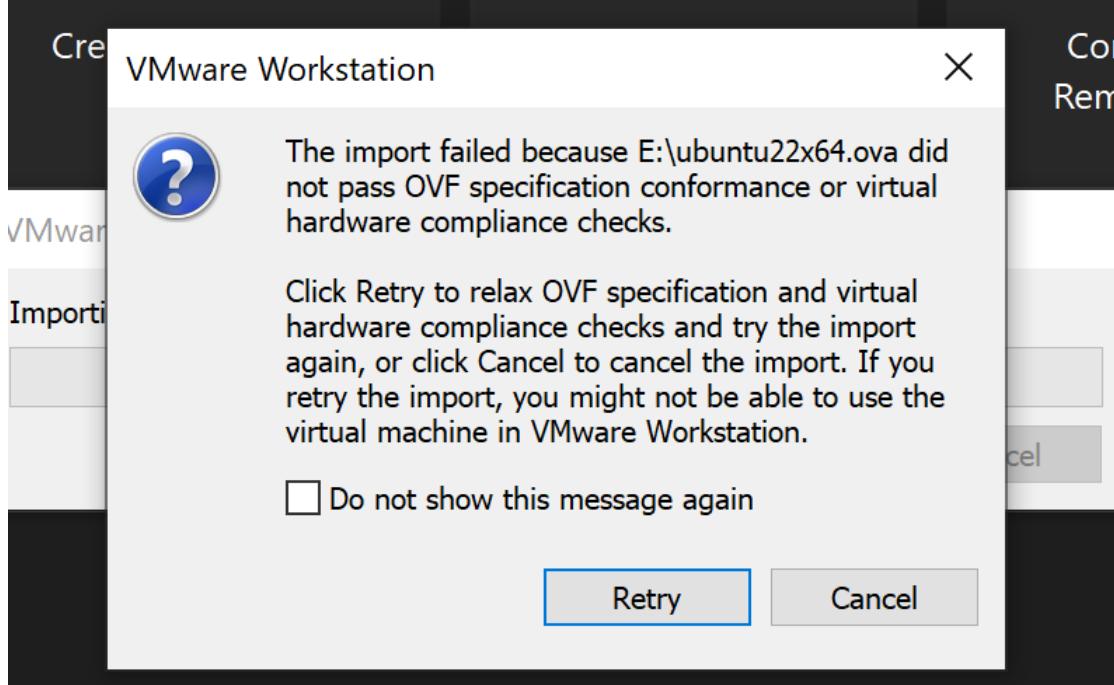
To convert this hardware family to one that's compatible with VMware Workstation Pro, ChatGPT suggested unpacking the OVA archive and directly editing the OVF file that contains the info. While 7zip was doing the unpacking, I did a [bit more reading online on the topic](How To Convert Virtual Machines Between VirtualBox and VMware (howtogeek.com))(<https://www.howtogeek.com/125640/how-to-convert-virtual-machines-between-virtualbox-and-vmware/>). Because this website says it should be possible to open the VM as is in VMware, once it is installed, so I decided to postpone modifying anything in the OVF until I shut off Hyper-V for good with bcddedit, restart my PC, and attempt to install VMware.

7zip kept showing "Unexpected end of data" all the way through.

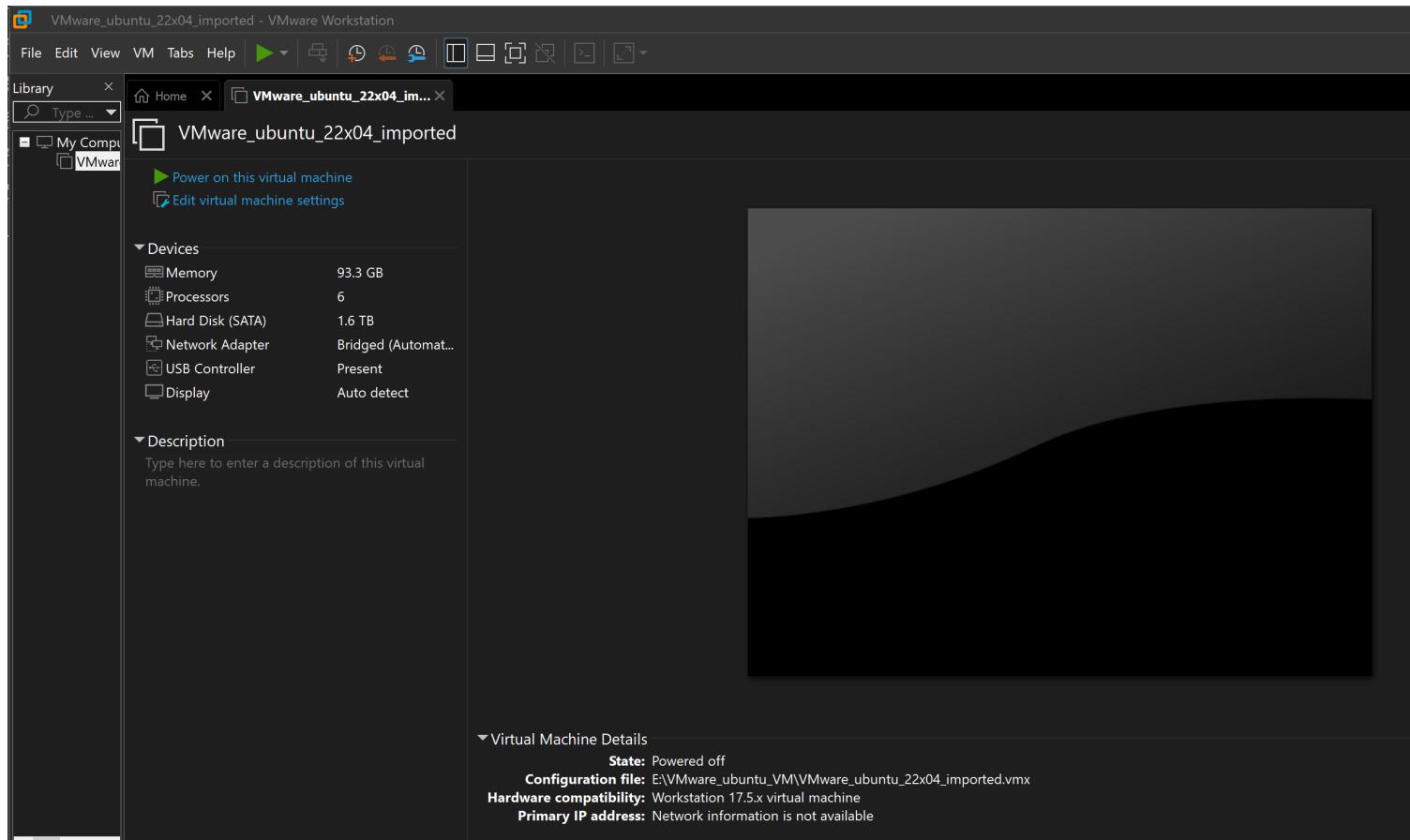
And then I restarted my PC. And got no BSOD, yay! *insert party smiley*

I re-ran the installer for VMware Workstation Pro, and didn't get that error relating to Hyper-V, which means it's now finally switched off and I don't need to worry any longer. I chose to skip adding the optional "Enhanced Keyboard Driver", as I did not find any overtly important reason to do so. The installation completed successfully and I could now finally open the software *!yay!*

I followed the instructions from How To Geek to import the VM from VirtualBox into VMware. I got the error the website predicted:



I hit "Retry" as instructed, which, as it says, makes the importing criteria less strict, and the actual import began. Annnnnnn it worked!



I started the machine and even *conda* was still working as expected in the terminal. What was left to do was to install the VMware Guest Additions from the Linux ISO. For this, with the VM powered off, I went to its settings and added a CD/DVD, which I pointed to the VMware Linux ISO that should contain the additional tools for this Ubuntu virtual machine. The behavior was rather strange after that:

```
(base) iweber@iweber-VirtualBox:~$ vmware-toolbox-cmd -v
12.3.5.46049 (build-22544099)
(base) iweber@iweber-VirtualBox:~$ sudo vmware-uninstall-tools.pl
sudo: vmware-uninstall-tools.pl: command not found
(base) iweber@iweber-VirtualBox:~$ sudo mount /dev/cdrom /mnt
mount: /mnt: WARNING: source write-protected, mounted read-only.
```

Maybe I have a broken installation of the Toolbox, because the option to reinstall it is greyed out in the VM menu, and, while some part of it is installed, returning me some version number, somehow, the uninstallation command isn't. So I tried to remove the Toolbox in its entirety from the system, running these commands one by one:

```
sudo rm -rf /usr/lib/vmware-tools # completed with no errors
sudo rm -rf /etc/vmware-tools # completed with no errors
sudo rm -rf /usr/lib/vmware-tools/modules # same
sudo rm -rf /usr/bin/vmware* # same
```

I then re-mounted the ISO image with

```
sudo mount /dev/cdrom /mnt
```

, which told me that it's already mounted.

Then, I used tar to extract the archive from the ISO:

```
tar -zxvf /mnt/VMwareTools-*.tar.gz -C /tmp
```

, which generated an onslaught of files, but completed fine and gave me a new command prompt.

I changed to the extracted archive directory, made the installation script executable, and ran it. It didn't work, and I realized, based on the messages, that the problem was that the toolbox of VirtualBox was still present on the system:

```
(base) iweber@iweber-VirtualBox:/tmp/vmware-tools-distrib$ sudo ./vmware-install.pl
The installer has detected an existing installation of open-vm-tools packages
on this system and will not attempt to remove and replace these user-space
applications. It is recommended to use the open-vm-tools packages provided by
the operating system. If you do not want to use the existing installation of
open-vm-tools packages and use VMware Tools, you must uninstall the
open-vm-tools packages and re-run this installer.
The packages that need to be removed are:
open-vm-tools
Packages must be removed with the --purge option.
The installer will next check if there are any missing kernel drivers. Type yes
if you want to do this, otherwise type no [yes] yes

INPUT: [yes]

Creating a new VMware Tools installer database using the tar4 format.

Installing VMware Tools.

In which directory do you want to install the binary files?
[/usr/bin] /usr/bin

INPUT: [/usr/bin]
```

!!! I started this kernel driver installer, but I soon realized this was nonsense, and aborted it. I re-ran the rm operations above to purge any existing installed pieces of VMware Toolbox. Then, I proceeded to remove the open-vm-tools installation:

```
sudo apt-get remove --purge open-vm-tools
sudo apt-get autoremove
```

Or...at least I thought I would be doing so. Instead, I got that very same nice error message regarding Nvidia and Cuda that I struggled with on VirtualBox. I tried:

```
sudo dpkg --purge open-vm-tools
```

and at least this one seems to have worked (I got an active command prompt after this):

```
(base) iweber@iweber-VirtualBox:$ sudo dpkg --purge open-vm-tools
(Reading database ... 265933 files and directories currently installed.)
Purging configuration files for open-vm-tools (2:12.3.5-3~ubuntu0.22.04.1) ...
```

I ran a few more commands ChatGPT suggested to completely purge any pieces left from either Toolbox or open-vm-tools:

```
(base) iweber@iweber-VirtualBox:$ sudo dpkg --purge open-vm-tools
(Reading database ... 265933 files and directories currently installed.)
Purging configuration files for open-vm-tools (2:12.3.5-3~ubuntu0.22.04.1) ...
(base) iweber@iweber-VirtualBox:$ sudo rm -rf /etc/vmware-tools
(base) iweber@iweber-VirtualBox:$ sudo rm -rf /usr/lib/open-vm-tools
(base) iweber@iweber-VirtualBox:$ sudo rm -rf /usr/share/open-vm-tools
(base) iweber@iweber-VirtualBox:$ sudo rm -rf /usr/bin/vmware*
(base) iweber@iweber-VirtualBox:$ sudo systemctl disable open-vm-tools
Failed to disable unit: Unit file open-vm-tools.service does not exist.
(base) iweber@iweber-VirtualBox:$ sudo systemctl stop open-vm-tools
Failed to stop open-vm-tools.service: Unit open-vm-tools.service not loaded.
(base) iweber@iweber-VirtualBox:$ sudo rm /etc/systemd/system/multi-user.target.wants/open-vm-tools.service
rm: cannot remove '/etc/systemd/system/multi-user.target.wants/open-vm-tools.service': No such file or directory
(base) iweber@iweber-VirtualBox:$
```

It seems no piece were left anyway - the system said it failed to find the things that I was trying to delete. Also, my shared clipboard between my Windows host and my VM stopped working, as was to be expected.

I attempted to reinstall VMware tools, when I got an interesting message:

```
(base) iweber@iweber-VirtualBox:$ sudo mount /dev/cdrom /mnt
mount: /mnt: /dev/sr0 already mounted on /media/iweber/VMware Tools.
(base) iweber@iweber-VirtualBox:$ cd /tmp/vmware-tools-distrib/
(base) iweber@iweber-VirtualBox:/tmp/vmware-tools-distrib$ ls -lat
total 408
drwxrwxrwt 20 root root 4096 Aug 2 20:26 ..
drwxr-xr-x 9 iweber iweber 4096 Jul 18 2020 .
-rw-r--r-- 1 iweber iweber 147002 Jul 18 2020 FILES
drwxr-xr-x 2 iweber iweber 4096 Jul 18 2020 bin
drwxr-xr-x 2 iweber iweber 4096 Jul 18 2020 doc
drwxr-xr-x 5 iweber iweber 4096 Jul 18 2020 etc
-rw-r--r-- 1 iweber iweber 2538 Jul 18 2020 INSTALL
drwxr-xr-x 2 iweber iweber 4096 Jul 18 2020 installer
-rwrxr-xr-x 1 iweber iweber 227024 Jul 18 2020 vmware-install.pl
drwxr-xr-x 5 iweber iweber 4096 Jul 18 2020 caf
drwxr-xr-x 14 iweber iweber 4096 Jul 18 2020 lib
drwxr-xr-x 3 iweber iweber 4096 Jul 18 2020 vgaauth
(base) iweber@iweber-VirtualBox:/tmp/vmware-tools-distrib$ sudo ./vmware-install.pl
open-vm-tools packages are available from the OS vendor and VMware recommends
using open-vm-tools packages. See http://kb.vmware.com/kb/2073803 for more
information.
Do you still want to proceed with this installation? [no] 
```

I am postponing decisionmaking about this to tomorrow, I'll have to do a bit more research to confirm. This is the website it is sending me to: <https://knowledge.broadcom.com/external/article?legacyId=2073803> and it is last updated in February of this year, so seems fairly recent...

After some further research, I see that VMware indeed recommends open vm tools as the open source implementation of... VMware tools. They also recommend installing it as per the instructions of the OS producer.

After reading [this](#), I chose to go with the desktop version of open-vm-tools, as that offers extended capability for GUI-based systems, whereas, apparently, the simple open-vm-tools package is meant for bare-bones Linux servers.

I first checked what packages need updating before that with `sudo apt update` and listed the packages with `sudo apt list --upgradable`. I got this list, and decided to take a snapshot of the system before the update, just in case something destabilizes it:

minal Mo Aug 5 11:24

iweber@iweber-VirtualBox: ~

```
nf-core: command not found
(base) iweber@iweber-VirtualBox:~$ sudo apt update
[sudo] password for iweber:
Get:1 file:/var/cuda-repo-ubuntu2004-11-7-local InRelease [1.575 B]
Get:1 file:/var/cuda-repo-ubuntu2004-11-7-local InRelease [1.575 B]
Hit:2 http://de.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Hit:4 http://de.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 https://packages.microsoft.com/repos/code stable InRelease
Hit:6 http://de.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:7 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Hit:8 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:9 https://packagecloud.io/github/git-lfs/ubuntu jammy InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
(base) iweber@iweber-VirtualBox:~$ sudo apt list --upgradable
Listing... Done
code/stable 1.92.0-1722473020 amd64 [upgradable from: 1.91.1-1720564633]
libldap-2.5-0/jammy-updates 2.5.18+dfsg-0ubuntu0.22.04.2 amd64 [upgradable from: 2.5.18+dfsg-0ubuntu0.22.04.1]
libldap-common/jammy-updates,jammy-updates 2.5.18+dfsg-0ubuntu0.22.04.2 all [upgradable from: 2.5.18+dfsg-0ubuntu0.22.04.1]
linux-generic-hwe-22.04/jammy-updates 6.5.0.45.45~22.04.1 amd64 [upgradable from: 6.5.0.44.44~22.04.1]
linux-headers-generic-hwe-22.04/jammy-updates 6.5.0.45.45~22.04.1 amd64 [upgradable from: 6.5.0.44.44~22.04.1]
linux-headers-generic/jammy-updates,jammy-security 5.15.0.117.117 amd64 [upgradable from: 5.15.0.116.116]
linux-image-generic-hwe-22.04/jammy-updates 6.5.0.45.45~22.04.1 amd64 [upgradable from: 6.5.0.44.44~22.04.1]
openjdk-11-jre-headless/jammy-updates,jammy-security 11.0.24+8-1ubuntu3~22.04 amd64 [upgradable from: 11.0.23+9-1ubuntu1~22.04.1]
openjdk-11-jre/jammy-updates,jammy-security 11.0.24+8-1ubuntu3~22.04 amd64 [upgradable from: 11.0.23+9-1ubuntu1~22.04.1]
python3-update-manager/jammy-updates,jammy-updates 1:22.04.20 all [upgradable from: 1:22.04.19]
snapd/jammy-updates,jammy-security 2.63+22.04ubuntu0.1 amd64 [upgradable from: 2.63+22.04]
update-manager-core/jammy-updates,jammy-updates 1:22.04.20 all [upgradable from: 1:22.04.19]
update-manager/jammy-updates,jammy-updates 1:22.04.20 all [upgradable from: 1:22.04.19]
(base) iweber@iweber-VirtualBox:~$
```

I hit `sudo apt upgrade` to upgrade all of the packages listed above, and ran into the same issue with the NVIDIA and Cuda drivers not working. And when I tried the actual command to install `open-vm-tools`,

```
sudo apt install open-vm-tools-desktop
```

it happened again...

I had a look in the VMware graphics settings and found that I could change some options regarding 3d acceleration, which I know Nvidia is in charge of, so I switched that on. This is how the updated settings look like:

[Hardware](#) [Options](#)

Device	Summary
Memory	93.3 GB
Processors	6
Hard Disk (SATA)	1.6 TB Using file C:\Program Files (x86...)
CD/DVD (SATA)	
Network Adapter	NAT
USB Controller	Present
Display	Auto detect

3D graphics Accelerate 3D graphics**Monitors** Use host setting for monitors Specify monitor settings:

Number of monitors:

1

Maximum resolution of any one monitor:

2560 x 1600

Graphics memory

Maximum amount of guest memory that can be used for graphics memory:

8 GB (recommended)

Display scaling Stretch mode: Keep aspect ratio stretch

Stretch the virtual machine display while maintaining the user interface aspect ratio

 Free stretch

Stretch the virtual machine display to fill the user interface, without maintaining the user interface aspect ratio

[Add...](#)[Remove](#)[OK](#)[Cancel](#)[Help](#)

After this, I tried switching the machine back on to see if this maybe allows the use of Nvidia and Cuda and lets me install open-vm-ware....nope. Process failed with same problem.

Then, I also saw I have direct virtualization options under "Processors" in the virtual machine's settings. I changed those as follows:

Virtual Machine Settings



Hardware Options

Device	Summary
Memory	93.3 GB
Processors	6
Hard Disk (SATA)	1.6 TB Using file C:\Program Files (x86...)
CD/DVD (SATA)	NAT
Network Adapter	Present
USB Controller	Auto detect
Display	

Processors

Number of processors: Number of cores per processor:
 Total processor cores: 6

Virtualization engine

- Virtualize Intel VT-x/EPT or AMD-V/RVI
- Virtualize CPU performance counters
- Virtualize IOMMU (IO memory management unit)

[Add...](#)[Remove](#)[OK](#)[Cancel](#)[Help](#)

Now, when peeking at the Windows task manager, I can actually see Vmware using the GPU!

But I still can't install anything, not even Ubuntu's very own tool , without the same error about nvidia and cuda popping up again.

I ran

```
sudo apt-get remove --purge '^nvidia-.*'
sudo apt-get remove --purge '^cuda-.*'
sudo apt-get remove --purge '^libcuda-.*'
sudo apt-get remove --purge '^libnvidia-.*'
sudo apt-get autoremove --purge
sudo apt-get clean
```

to clear any installation I might have left on the system for any nvidia and cuda-related packages, and start afresh. Confirmed no more packages left with `lsmod | grep nvidia`, which didn't return anything, and then added the graphics drivers repository to my system with

```
sudo add-apt-repository ppa:graphics-drivers/ppa
sudo apt-get update
```

I then went to [Ubuntu's official Nvidia driver page](#) to see how they recommend installing these drivers from the command line interface (CLI)...no help there. All i got when checking for the recommended drivers for my machine with `sudo ubuntu-drivers install` is...drumroll...open-vm-tools /augh-cry

I suspect this won't do much, but, in sheer despair, I tried forcing the installation of an Nvidia driver with `sudo apt-get install nvidia-driver-535` and rebooted with `sudo reboot`. But running `nvidia-smi` to check the driver gave the same message as before the install, "NVIDIA-SMI has failed

because it couldn't communicate with the NVIDIA driver. Make sure that the latest NVIDIA driver is installed and running."

I shut off the virtual machine and went scouting in my BIOS options to see if anything else relating to Intel VT needs switching on, but it does not look like it - "VT-d" is activated in my BIOS setup.

Again out of lack of better options, I tried installing the nvidia-cuda toolkit directly using `sudo apt install nvidia-cuda-toolkit`. Oddly enough, it downloaded all of the packages and completed successfully. I rebooted the machine and typed `nvidia-smi` to check if the GPU is recognized, but got the same error as above, saying it has failed because it couldn't communicate with the NVIDIA driver.

I also tried installing the CUDA toolkit from https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=22.04&target_type=deb_local, but I do suspect this won't do much if Nvidia can't communicate from my virtual machine with the Nvidia driver.

For the umpteenth time, I purged everything Nvidia related, and installed the Nvidia driver version 550, apparently the latest recommended by Nvidia for Linux 64-bit systems like my VM (I currently have version 535 installed) - `sudo apt-get install nvidia-driver-550`. I also re-installed the CUDA toolkit according to the link above.

And, again, out of sheer desperation, I tried installing Docker as I had previously done on my old VirtualBox virtual machine with

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

and...it gave no errors?!??

I then ran, as indicated on the Docker website,

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

, which also completed without errors, and, finally:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

It...completed?!?? Miracles????

IT WORKED. It actually worked. The test command completed successfully:

```
(base) iweber@iweber-VirtualBox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc69891e79d84cc4cdcec6
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:
<https://hub.docker.com/>

For more examples and ideas, visit:
<https://docs.docker.com/get-started/>

So...now I have Docker on the system. Will perhaps also the pigz installation work now?

It did. Or a prior installation is now working (it looks rather like it tried to update it but didn't find any more recent version):

```
(base) iweber@iweber-VirtualBox:~$ sudo apt install pigz
[sudo] password for iweber:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
pigz is already the newest version (2.6-1).
pigz set to manually installed.
The following packages were automatically installed and are no longer required:
 libatomic1:i386 libdrm-amdgpu1:i386 libdrm-intel1:i386 libdrm-nouveau2:i386
 libdrm-radeon1:i386 libdrm2:i386 libedit2:i386 libelf1:i386 libexpat1:i386 libgl1:i386
 libgl1-mesa-dri:i386 libglapi-mesa:i386 libglvnd0:i386 libglx-mesa0:i386 libglx0:i386
 libicu70:i386 libllvm15:i386 libpciaccess0:i386 libsensors5:i386 libstdc++6:i386
 libx11-xcb1:i386 libxcb-dri2-0:i386 libxcb-dri3-0:i386 libxcb-glx0:i386
 libxcb-present0:i386 libxcb-randr0:i386 libxcb-shm0:i386 libxcb-sync1:i386
 libxcb-xfixes0:i386 libxfixes3:i386 libxml2:i386 libxshmfence1:i386 libxxf86vm1:i386
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

I **immediately** started creating another snapshot of the machine, just in case anything else goes awry.

Ensuring Docker can use GPU

ChatGPT had me test whether Docker can actually use my GPU, in case that's needed (and it likely will be, with the RNA-Seq pipelines) using `sudo docker run --rm --gpus all nvidia/cuda:11.5-base nvidia-smi`, which didn't work ("Unable to find image 'nvidia/cuda:11.5-base' locally docker: Error response from daemon: manifest for nvidia/cuda:11.5-base not found: manifest unknown: manifest unknown. See 'docker run --help'.") - not surprising, considering that `nvidia-smi` still shows the same error message saying "NVIDIA-SMI has failed because it couldn't communicate with the NVIDIA driver. Make sure that the latest NVIDIA driver is installed and running."

So it suggested installing the `nvidia-docker2` package

```

distribution=$( . /etc/os-release;echo $ID$VERSION_ID)
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -
curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list
sudo apt-get update

sudo apt-get install -y nvidia-docker2

sudo systemctl restart docker

```

and then pulling a Docker image that's easily retrievable to test if the GPU is working:

```

sudo docker pull nvidia/cuda:11.0-base
sudo docker run --rm --gpus all nvidia/cuda:11.0-base nvidia-smi

```

Which...did not work. ChatGPT said the exact name of the docker container (its tag) might be at fault, and I then went on the official [GitLab repo of Nvidia with Docker containers to test different Cuda versions](#). Here, I noticed that the Cuda toolkit version that I have (`nvcc -V` -> "Cuda compilation tools, release 11.5, V11.5.119") is not listed under "Ubuntu 22.04" but under earlier versions. I did install the toolkit as recommended on the [Cuda toolkit website](#) but, somehow, still installed version 11.5 instead of 12.6, which seems to be current recommended one.

How did that happen? I did follow the instructions on the Cuda toolkit website to a T - I cycled back to previous commands, and I for sure ran the commands for version 12.6, the latest recommended one. I now ran, out of curiosity, "sudo apt-get -y install cuda-toolkit-12-6", and what I got back was

```

[sudo] password for iweber: Reading package lists... Done Building dependency tree... Done Reading state information... Done

***cuda-toolkit-12-6 is already the newest version (12.6.0-1)***.

```

The following packages were automatically installed and are no longer required: libatomic1:i386 libdrm-amdgpu1:i386 libdrm-intel1:i386 libdrm-nouveau2:i386 libdrm-radeon1:i386 libdrm2:i386 libedit2:i386 libelf1:i386 libexpat1:i386 libffi8:i386 libgl1:i386 libgl1-mesa-dri:i386 libglapi-mesa:i386 libglvnd0:i386 libglx-mesa0:i386 libglx0:i386 libicu70:i386 liblomm15:i386 libnvidia-egl-wayland1 libnvidia-egl-wayland1:i386 libpciaccess0:i386 libsensors5:i386 libstdc++6:i386 libwayland-client0:i386 libwayland-server0:i386 libxcb1:i386 libxcb-dri2-0:i386 libxcb-dri3-0:i386 libxcb-glx0:i386 libxcb-present0:i386 libxcb-randr0:i386 libxcb-shm0:i386 libxcb-sync1:i386 libxcb-xfixes0:i386 libxfixes3:i386 libxml2:i386 libxshmfence1:i386 libxxf86vm1:i386 nvidia-firmware-550-550.107.02 Use 'sudo apt autoremove' to remove them. 0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.

Still, when I do the `nvcc -V` check, what I get is that I have Cuda compilation tools, release 11.5, V11.5.119. I wondered if it is possible that I have two versions on my system, both 12.6 and 11.5, and, if yes, how I would check that. ChatGPT had me run `ls /usr/local/` to see what Cuda folders and present there, and, lo and behold, it was three: `cuda`, `cuda-12`, and `cuda-12.6`. So it is indeed possible that they're all different versions. However, my system apparently can't access either of them directly, since they seem to not be in the PATH variable - running `echo $PATH | grep -o "/usr/local/cuda[^:]*/bin"` to find anything cuda-related in the PATH didn't return anything.

To add the latest version to the PATH variable, I edited my `.bashrc` file and added, through the console, the path to the folder of the latest Cuda version:

```

sudo echo 'export PATH=/usr/local/cuda-12.6/bin${PATH:+:$PATH}' >> ~/.bashrc
sudo echo 'export LD_LIBRARY_PATH=/usr/local/cuda-12.6/lib64${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}' >> ~/.bashrc

```

, then re-loaded my `.bashrc` with `source ~/.bashrc`. Now, when running `echo $PATH | grep -o "/usr/local/cuda[^:]*/bin"`, I got back `"/usr/local/cuda-12.6/bin"` -> success in adding it to the PATH. When I now checked the version with `nvcc -V`, I finally saw

```

(base) iweber@iweber-VirtualBox:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2024 NVIDIA Corporation
Built on Fri_Jun_14_16:34:21_PDT_2024
Cuda compilation tools, release 12.6, V12.6.20
Build cuda_12.6.r12.6/compiler.34431801_0

```

I now tried pulling an existing Cuda Docker container from [their GitLab repository](#) using

```

sudo docker pull nvidia/cuda:12.5.1-base-ubuntu22.04

```

```
(base) iweber@iweber-VirtualBox:~$ sudo docker pull nvidia/cuda:12.5.1-base-ubuntu22.04
12.5.1-base-ubuntu22.04: Pulling from nvidia/cuda
3713021b0277: Pull complete
1a4c6357a507: Pull complete
c13a925c73f2: Pull complete
c6961d6f1272: Pull complete
85bb4fb01b0: Pull complete
Digest: sha256:47181750209e68cab2e36c3e0e8ab0ab8d76bdabbb418e56e4b0ac7b08b97d34
Status: Downloaded newer image for nvidia/cuda:12.5.1-base-ubuntu22.04
docker.io/nvidia/cuda:12.5.1-base-ubuntu22.04
```

(I didn't see a version for Cuda 12.6, so I chose a container running the latest available version, 12.5)

I next tried running the container to see if GPU usage actually works using

```
sudo docker run --rm --gpus all nvidia/cuda:12.5.1-base-ubuntu22.04 nvidia-smi
```

...and it didn't. I restarted the machine, thinking that maybe this will help certain settings load properly, and tried again. I also tried running it without the `nvidia-smi` part. Either way, I got the same error message: "docker: Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: error during container init: error running hook #0: error running hook: exit status 1, stdout: , stderr: Auto-detected mode as 'legacy' nvidia-container-cli: initialization error: nvml error: driver not loaded: unknown."

This is apparently a problem, again, with the Nvidia driver. I could try pulling some other docker container to test this further, but I suspect it wouldn't change much if `nvidia-smi` doesn't show anything useful. And I could go on and on trying this out, but maybe I'm lucky again and maybe the pipeline works even without the GPU usage. So the next important step is to generate some small test data to run a pilot test with the pipeline (see [Pipeline pilot experiment *C. elegans* data from the course](#))

Accessing shared folder from VMware virtual machine

I noticed that I could see my former inter-OS shared folder from the VMware virtual machine, but it appeared as empty, even though I set the path to it in the VM settings in VMware Workstation Pro. I unmounted this folder and then checked the [VMware website and ChatGPT](#) for how to mount the folder. I realized I need to first [check what Linux kernel version I have](#), so I used

```
uname -srn
```

so I could confirm I have the `Linux 6.5.0-45-generic x86_64` kernel.

I created a new mount point with

```
sudo mkdir /mnt/mnt-win-ubu-shared
```

and then, to access my shared folder at this mount point, used:

```
sudo /usr/bin/vmhgfs-fuse .host:/ /mnt/mnt-win-ubu-shared -o subtype=vmhgfs-fuse,allow_other
```

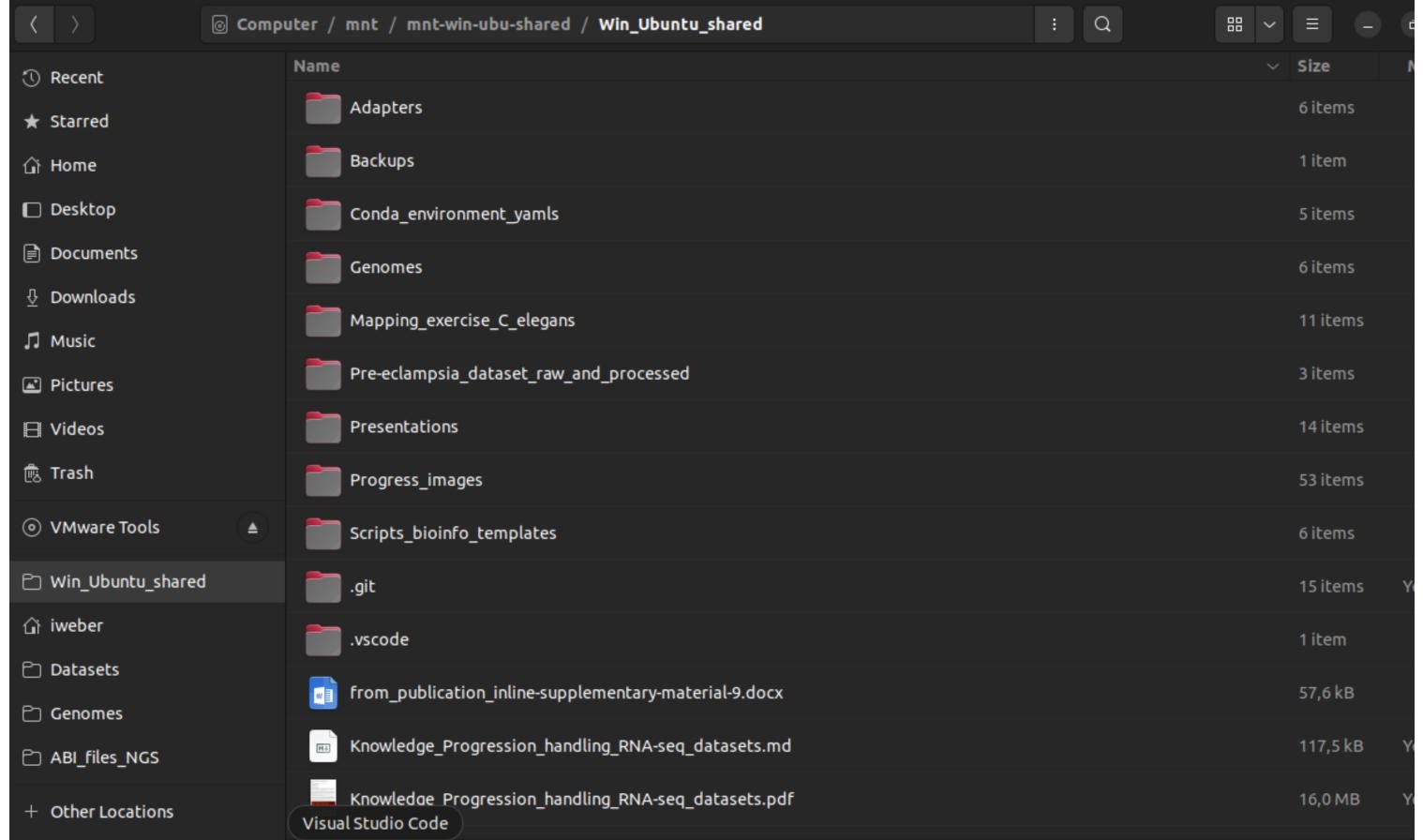
I attempted to make the mounted folder not ask me for my password every time by adding this to `/etc/fstab` (sudo access):

```
.host:/ /mnt/mnt-win-ubu-shared fuse.vmhgfs-fuse defaults,allow_other 0 0
```

and then finally mounted

```
sudo mount -a
```

And restarted my system. I had to go to the mounting point `/mnt/mnt-win-ubu-shared` from the address bar of my file explorer (ctrl+L) but could then see the folder. I swiftly added it to my bookmarks in the left-hand-side panel.



I also changed the alias in my `.bashrc` for the command needed to switch to this inter-OS shared folder:

```
# some more handy aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
alias mydata='cd /home/iweber/Documents'
alias abidata='cd /home/iweber/Documents/ABI_files_NGS'
alias inter-OS='cd /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared'
alias inter-OS_data='cd /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Datasets/'
```

I reloaded the `.bashrc` with `source ~/.bashrc` and set the permissions on the folder for everyone (including my user) to be able to use it:

```
sudo chmod 755 /mnt/mnt-win-ubu-shared
```

Checking that the git repository within the newly mounted inter-OS folder still works

I immediately hit `git status` to make sure my git repo still works, and what I got was an error saying "fatal: detected dubious ownership in repository at '/mnt/mnt-win-ubu-shared/Win_Ubuntu_shared' To add an exception for this directory, call: git config --global --add safe.directory /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared".

To hopefully resolve this, I ran:

```
git config --global --add safe.directory /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared
```

and then tested if it pops up as safe using `git config --global --get-all safe.directory` and its address does show up in the list.

Next, I tested if git is running as it should using `git status`, and it does - I immediately got an overview of untracked changes :) (I added the folders relating to my pilot pipeline run, "Adapters", "Genomes" and "C elegans mapping exercise", to `.gitignore` because I don't care about tracking them. When I repeated the `git status` command, they did not show up any longer). Everything was updated as it should on GitHub :)

The rnassplice pipeline

`rnassplice` is a fairly new pipeline on the nf-core, published just a few months back. This is partially the reason for which I really badly want to run it in a container rather than just in conda: I suspect it probably can still throw a number of bugs when run on a system even remotely different than whatever the developers were using.

I pulled the pipeline from nf-core using

```
nextflow pull nf-core/rnasplice
```

```
(Nextflow) iweber@iweber-VirtualBox:/mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared$ nextflow pull nf-core/rnasplice
Checking nf-core/rnasplice ...
downloaded from https://github.com/nf-core/rnasplice.git - revision: 1d0494ae34 [master]
```

While I am trying to transfer the entirety of my virtual machine from VirtualBox to VMware (see section [Attempting to transfer entirety of virtual machine to VMware](#)), I read about the pipeline to better understand what the individual steps involve.

Pipeline description for rnasplice

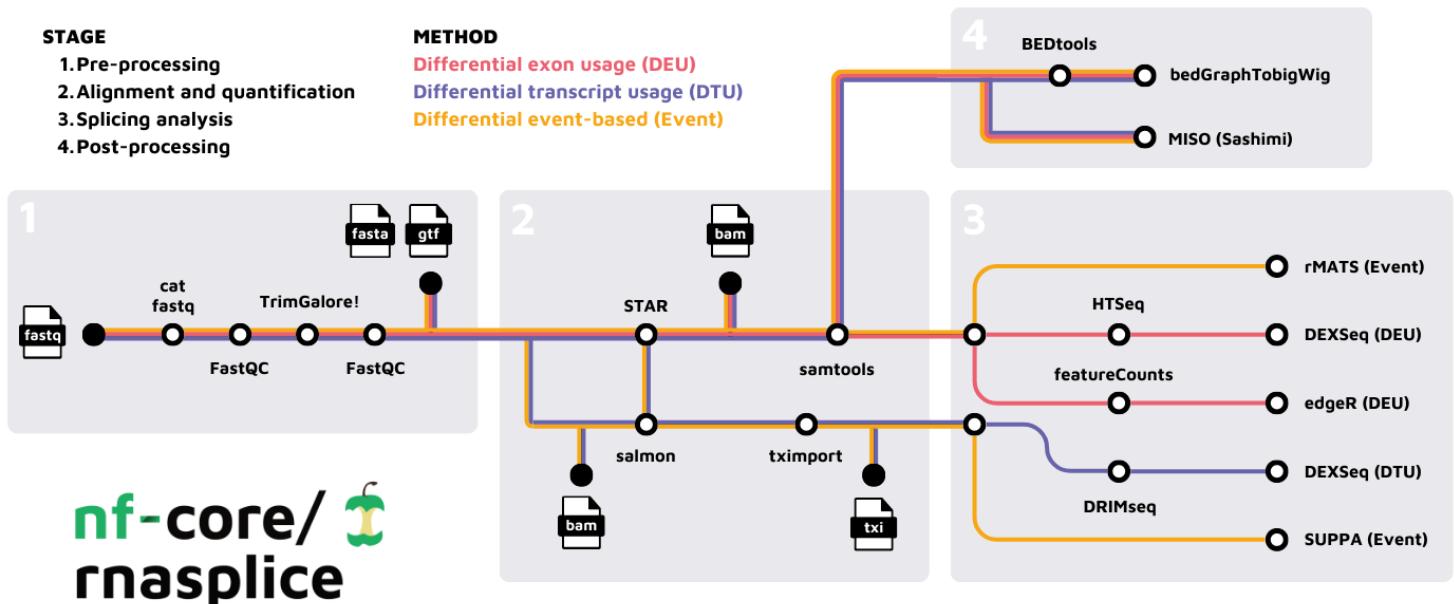
Calling the pipeline could be as easy as typing

```
nextflow run nf-core/rnasplice \
--input samplesheet.csv \
--contrasts contrastsheet.csv \
--genome GRCh37 \
--outdir my/result/directory \
-profile docker
```

but what does it do, and what other parameters need to be set?

Graphical overview

I tried creating an overview here that is somewhat easier to describe with words than the image on the website:



Parts of the pipeline

Also described well at <https://github.com/zifornd/rnasplice/blob/dev/docs/output.md>

Part 1 of pipeline: preprocessing

`cat fastq`

is something I would use if I had several fastq files per sample, coming from several runs of sequencing. This is usually done to increase sequencing depth, but the SRA archives I downloaded contained precisely two files per sample, containing the two parts of a pair of reads each.

`FastQC`

own addition: `MultiQC` to summarize results

I already performed these steps on the WSL, so will not need to do so again. I already know I need to trim 10 bases from the 5' end of the reads in order to solve any problems that the suboptimal per-base distribution of the four nucleotides may create.

`TrimGalore!` - [GitHub](#) and [user guide](#)

- **COMMAND:** `trim_galore [options] <filename(s)>`

- [will set --cores 4 to make maximal use of all of my cores but following the indication on the website that says this is a sweet spot (I set 6 cores to be available to the VM)] *dropped this one as it isn't a TrimGalore option I can automatically set. I set the number of cores as a parameter for the pipeline in general*
- [will use the --paired option because I have paired reads as input] *dropped this one as it isn't a TrimGalore option I can automatically set. This is set for the pipeline in general based on the sample sheet*
- see above - need to trim 10 bases. I previously worked with Trimmomatic for this kind of operation, but TrimGalore also has the option --clip_R1 10 and --clip_R2 10, which will remove these 10 bases from the 5' ends of all of the reads.
- Since it employs FastQC, TrimGalore does an automated check for low quality base calls and trims accordingly. I don't need to specify the Phred score cutoff for base call quality - it automatically sets it to the most modern version, --phred33.
- in my previous checks, FastQC did not detect any significant adapter contamination in the reads. If there are any adapters left, they will be of the Illumina type, since this is what kit was employed. However, I am a little reluctant to just use the --illumina parameter, as this trimming is extremely stringent and an overlap of even only 1 base between the end of the read and the adapter sequences will be removed, potentially losing useful information in the process. I will rather let TrimGalore automatically look for potential overlaps with adapter sequences and set --stringency to 5 to lose less information, since the risk of real adapter contamination seems extremely low AND I am anyway trimming the 5' ends because of the biased base distribution
- it also automatically filters the resulting sequences and, if a read is then shorter than 20 bp, it is automatically removed from the results.
- output: **[does it also create separate files for reads that could successfully be paired up from the two input FastQ files and those that could not?]**

>>>

FastQC

- these two steps are done to check whether the trimming worked exactly as expected, so, in my case, I'd expect the new report generated by MultiQC to show me 140 bp reads with no more issues at the 5' ends.
 - to run FastQC again on the results, use --fastqc_args "--outdir /TrimGalore_output" (TrimGalore automatically creates the output directory if it doesn't exist)
- own addition: MultiQC to summarize results*

To do

I will then need to add a MultiQC step to summarize the results, but I can do that manually

input files: fasta and gtf - see sections [Downloading genomes](#) and [Initial creation of genome and transcriptome indices](#)

[this certainly needs more research for accuracy!] also see <https://github.com/zifornd/rnasplice/blob/dev/docs/output.md#star-and-salmon>
The rest of the pipeline is divided in two branches. Each branch starts with one of two different algorithms, [STAR](#) and [Salmon](#), whose mechanics sound similar at first, but work quite differently.

As explained [here](#) by Devon Ryan, an aligner like STAR genuinely looks at the base-to-base match of a read to a region in the target genome ("does this read from the sequencing run align with this bit of the genome?") and also saves information on whether the read matches the genomic region as-is, from one end of the read to the other, or whether there are gaps or insertions ("does every one of the 150 bases in the read match 150 bases at the location where it's mapped in the genome? Or are, say, only bases 1-57 aligned, with an unmatching portion from bases 58-92, and then 93-150 match again?"). This means that it can later be used to accurately quantify which genes are expressed at what levels. Additionally, it is built so that it can easily identify reads that span exon-exon boundaries, thereby making it able to distinguish which of the different transcripts that can come from the same genomic locus the read could be derived from. This information is usually collected in a SAM file (human-readable) and then, for space reasons, converted into a BAM file (non-human-readable).

To do:

Read more on how Salmon works - the Harvard Chan Bioinfo Core PDF has good visualizations.

I found this excellent [RNA-seq analysis course](#) from the Harvard Chan Bioinformatics Core ([Zenodo](#)). In the [Salmon lesson](#), they explain how to set the parameters:

STAR is the gold standard of the old school of transcript abundance analysis when looking to map reads in a splice-aware manner. Salmon is the new star on the block because it can be used in two different ways: as a pseudo-aligner or as a real aligner. It truly shines as a pseudo-aligner, a mode in which it does not care about the exact fit of the read base-to-base, and does a quantification of "is this read likely to come from this transcript?" based on some more complex background mathematics. This is called pseudo-alignment, and is much faster than an actual base-to-base alignment, like STAR performs. In this mode, Salmon does not compare the reads to a reference genome but to a reference transcriptome, for which it needs a so-called index of the transcriptome. Often, such reference indices are readily available for download for well-studied organisms, such as the mouse (see [Salmon installation and mouse transcriptome index](#) for how I downloaded it).

Salmon also makes some adjustments for normalizing the read counts, e.g. to the length of the transcript, in order to produce accurate estimates of how abundantly different splice isoforms are expressed from a particular gene. It can take the BAM files generated by STAR in order to give these results, which we will obtain with STAR in the first part of the pipeline. [how does tximport play into this?]

Salmon can also operate on raw reads from scratch, as an aligner like STAR would, and I could theoretically let it start its branch of the pipeline from scratch, based on the raw reads, but I will anyway get the BAM files from the STAR branch, [so I could save some time on that.] **does this really save time? pseudo-alignment is much faster - maybe it's better for accuracy, but not time**

The creators of the dataset/authors of the original study used Hisat2 instead of STAR to align the reads to the mm10 mouse genome, and HTSeq to obtain read counts aka expression levels for the genes. The rnassplice pipeline also runs HTSeq after the STAR step, and I am curious how the results will compare to the ones published in the paper.

Part 2a of pipeline: read alignment and read count quantification starting with STAR

STAR

aligns the reads

Parameters to consider:

- `--star_index`: Provide the path to the STAR index built for your reference genome.
- `--genome`: Specify the reference genome (e.g., GRCh38) (but only if using the AWS iGenomes, which seem to be very out of date [source: warning from nextflow pages])
- `--sjdb_overhang`: Adjust based on your read length. A common choice is `read_length - 1`, or else it defaults to 100 bp. So I will use **149**, because I am dealing with 150 bp reads.
- `--save_unaligned true` if wanting to get insight into which reads weren't aligned - this will give me an idea of whether the alignment in general is working reliably
- [Here](#), Renesh Bedre explains more of the other parameters that STAR can use, such as `--runThreadN`, which sets on how many cores the aligner should be run (I'll use 6, because that's how many I have)
- "if `--aligner star_salmon` is specified then all the downstream results will be placed in the `star_salmon/` directory."
(<https://github.com/zifornd/rnassplice/blob/dev/docs/output.md#alignment-post-processing>)

samtools

- <https://www.htslib.org/doc/>
- <https://www.htslib.org/doc/samtools.html>

is a set of programs used, in this pipeline, to process the BAM files for further use. The mapped reads are sorted by coordinate and the BAM files are processed to generate mapping statistics.

Part 3a of pipeline: splicing quantification with edgeR, DEXSeq

rMATS = splicing event quantification

- activated by default in pipeline
- `--rmats_threads`: Set the number of threads to use for **rMATS** [is this really necessary?]

Then follow two branches that deal with finding differentially expressed exons in the data:

HTSeq - part of the DEXSeq package

DEXSeq = differential exon usage

<https://genome.cshlp.org/content/22/10/2008>

- R/Bioconductor package, always active in the pipeline due to the parameter `--dexseq_exon` set to true by default
- "Using the `--aggregation` parameter the pipeline will combine overlapping genes into a single aggregate gene. This approach can alternatively be skipped and any exons that overlap other exons from different genes will be skipped. Other important options to take note of are the `--alignment_quality` parameter which can be set by the user and defines the minimum alignment quality required for reads to be included (defined in 5th column of a given SAM file) (default: 10). Prior to quantification, DEXSeq provides an annotation preparation script which takes a GTF file as input and returns a GFF file. Users may instead wish to define their own GFF file and skip this annotation preparation step by supplying it using the `--gff_dexseq` parameter."

featureCounts

- provides a quantification that edgeR requires
- " is activated when the parameter '`--edger_exon`' is enabled. Please note that as this is aimed at differential exon usage feature type is set as 'exon' and cannot be changed. Please take care to use a suitable attribute to categorize the featureCounts attribute type in your GTF using the option '`--gtf_group_features`' (default: 'gene_id')."
-

edgeR provides info on differential exon usage, based on featureCounts

Part 4 of the pipeline does some processing of the BAM files resulting from STAR in order to quantify splicing events and resulting transcript variants and make them easy to display in a genome browser. This last step is performed by [MISO, which creates so-called Sashimi plots with read densities across the exons and their junctions.][sure-sure?]

Part 4 of pipeline: post-processing after STAR/samtools branch

BEDtools

bedGraphToBigWig

- <https://github.com/zifornd/rnasplice/blob/dev/docs/output.md#bedtools-and-bedgraphtobigwig>
- the bigWig format is an even further compressed format than BAM, based on the BAM file generated by STAR. The bigWig file produced by the pipeline can be used in genome browsers such as the IGV genome browser to visualize the read mapping density across the genome

MISO/Sashimi

- MISO (Mixture of ISOforms) quantifies expression levels of different transcripts resulting from the alternative splicing of a pre-mRNA derived from one gene ([source] (<https://miso.readthedocs.io/en/fastmiso/index.html#what-is-miso>)). It actually performs a Monte Carlo Markov chain estimation [source explaining this concept!].
- like SUPPA, it can also quantify either full transcript variants with all of their splicing particularities, or individual events, such as how frequently one particular exon is included in a type of sample. See [Katz et al] (<http://www.nature.com/nmeth/journal/v7/n12/full/nmeth.1528.html>) for the source publication.
- MISO contains an utility called [sashimi_plot] (<https://miso.readthedocs.io/en/fastmiso/sashimi.html>) (automatically active in the rnasplice pipeline), which creates easy-to-understand plots showing both read densities across certain sequences and also the counts of how often certain splice junctions are paired with each other. This gives information on how frequently parts of a pre-mRNA are included in transcripts or skipped and how the resulting transcripts are structured. Also see [Katz et al] (<http://biorxiv.org/content/early/2014/02/11/002576>)
- the MISO/Sashimi part of the pipeline has two parameters with which users can specify exactly which genes to create the plots for, 'miso_genes' for a list of individual identifiers, given as a 'string in single quotes', and 'miso_genes_file', where one can give a list of identifiers stored in a file. I presume these options come in handy once the analysis is completed and I can look into the genes with the strongest alterations in splicing patterns in the pre-eclampsia mice in comparison to the control ones. See the [pipeline parameters page] (<https://nf-co.re/rnasplice/1.0.4/parameters#miso>) and [this page for potential errors/bugs.] (<https://github.com/nf-core/rnasplice/issues/72#issuecomment-1643637616>)

Part 2b of pipeline: read pseudo-alignment and/or read count quantification per transcript using Salmon

Salmon

- generates read counts per transcript using the BAM files from STAR or an alignment it makes by itself

tximport

- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4712774/>
- <https://bioconductor.org/packages/3.7/bioc/vignettes/tximport/inst/doc/tximport.html>
- it imports the quantifications generated by Salmon from the data and prepares them for differential expression analysis between samples from different conditions
- check this on how to use it independent of the pipeline: <https://divingintogeneticsandgenomics.com/post/downstream-of-bulk-rnaseq-read-in-salmon-output-using-tximport-and-then-deseq2/>
- tximport also solves issues with artificially inflated gene counts due to transcript isoforms derived from the same gene

Part 3b of pipeline: actual splicing junction quantification

DRIMSeq

- <https://f1000research.com/articles/5-1356/v2>
 - an R/Bioconductor package that quantifies changes between experimental conditions in the relative abundance of all of the possible transcript isoforms resulting from a genomic locus, and also monitors SNPs that can lead to such abundance changes (splicing quantitative trait loci, sQTLs)
 - it also [filters low-abundance transcripts]
- (<https://github.com/zifornd/rnasplice/blob/dev/docs/output.md>) and some other categories before we use DEXSeq (see below)

DEXSeq = in this case, used for differential transcript usage

- activated by default in the pipeline due to the parameter --dexseq_dtu being set to true ([Knowledge_Progression_handling_RNA-seq_datasets > Config file for the rnasplice pipeline with absolutely all of the default parameters](#))
-

SUPPA = differential event-based alternative splicing analysis

<https://github.com/comprna/SUPPA>

- can work in two different ways: detecting either

- individual splicing events, e.g. inclusion or skipping of a particular exon, (Skipping Exon = SE, Alternative 5'/3' Splice Sites = A5/A3 (generated together with the option SS), Mutually Exclusive Exons = MX,

Retained Intron = RI, Alternative First/Last Exons = AF/AL (generated together with the option FL),
- or can quantify the abundance of different transcript variants, with their specific composition of
exons or retained introns, alternative splice sites, etc.
- both ways are activated by default in the pipeline ([[Knowledge_Progression_handling_RNA-seq_datasets#Config
file for the rnassplice pipeline with absolutely all of the default parameters]])

Config file for the rnassplice pipeline with absolutely all of the default parameters

Taken directly from <https://github.com/nf-core/rnassplice/blob/1.0.4/nextflow.config>

```
/*
~~~~~ nf-core/rnassplice Nextflow config file ~~~~~
Default config options for all compute environments
*/
// Global default params, used in configs
params {

    // Input options
    input                  = null
    contrasts             = null
    source                = 'fastq'

    // References
    genome                = null
    transcript_fasta       = null
    gtf_extra_attributes  = 'gene_name'
    gtf_group_features    = 'gene_id'
    gencode               = false
    save_reference         = false
    igenomes_base          = 's3://ngi-igenomes/igenomes/'
    igenomes_ignore        = false

    // Trimming
    clip_r1               = null
    clip_r2               = null
    three_prime_clip_r1   = null
    three_prime_clip_r2   = null
    trim_nextseq           = null
    save_trimmed           = false
    skip_trimming          = false
    skip_trimgalore_fastqc = false
    min_trimmed_reads     = 10000

    // Alignment
    aligner               = 'star_salmon'
    pseudo_aligner         = 'salmon'
    bam_csi_index          = false
    seq_center              = null
    salmon_quant_libtype   = null
    star_ignore_sjdbgtf    = false
    skip_alignment          = false
    save_unaligned          = false
    save_align_intermeds   = false
    save_merged_fastq      = false

    // QC
    skip_bigwig            = true
    skip_fastqc             = false

    // rMATs
    rmats                 = true
    rmats_splice_diff_cutoff = 0.0001
    rmats_paired_stats    = true
    rmats_read_len          = 40
    rmats_novel_splice_site = false
    rmats_min_intron_len   = 50
    rmats_max_exon_len     = 500

    // DEXSeq DEU
    dexseq_exon            = true
}
```

```

save_dexseq_annotation      = false
gff_dexseq                 = null
alignment_quality          = 10
aggregation                = true
save_dexseq_plot             = true
n_dexseq_plot               = 10

// edgeR DEU
edger_exon                  = true
save_edger_plot              = true
n_edger_plot                 = 10

// DEXSeq DTU
dexseq_dtu                   = true
dtu_txi                      = 'dtuScaledTPM'

// Miso
sashimi_plot                 = true
miso_genes                    = 'ENSG00000004961, ENSG00000005302, ENSG00000147403'
miso_genes_file               = null
miso_read_len                 = 75
fig_width                     = 7
fig_height                    = 7

// DRIMSeq Filtering
min_samps_feature_expr       = 2
min_samps_feature_prop        = 2
min_samps_gene_expr           = 4
min_feature_expr              = 10
min_feature_prop              = 0.1
min_gene_expr                 = 10

// SUPPA options
suppa                         = true
suppa_per_local_event          = true
suppa_per_isoform              = true
suppa_tpm                      = null

// SUPPA Generate events options
generateevents_pool_genes     = true
generateevents_event_type      = 'SE SS MX RI FL'
generateevents_boundary         = 'S'
generateevents_threshold        = 10
generateevents_exon_length     = 100
psiperevent_total_filter       = 0

// SUPPA Diffsplice options
diffsplice_local_event         = true
diffsplice_isoform              = true
diffsplice_method               = 'empirical'
diffsplice_area                 = 1000
diffsplice_lower_bound           = 0
diffsplice_gene_correction      = true
diffsplice_paired               = true
diffsplice_alpha                 = 0.05
diffsplice_median               = false
diffsplice_tpm_threshold        = 0
diffsplice_nan_threshold        = 0

// SUPPA Cluster options
clusterevents_local_event      = true
clusterevents_isoform           = true
clusterevents_sighthreshold     = null
clusterevents_dpsithreshold    = 0.05
clusterevents_eps                = 0.05
clusterevents_metric              = 'euclidean'
clusterevents_separation         = null
clusterevents_min_pts            = 20
clusterevents_method              = 'DBSCAN'

// MultiQC options
multiqc_config                  = null
multiqc_title                   = null
multiqc_logo                     = null
max_multiqc_email_size           = '25.MB'
multiqc_methods_description      = null

```

```

// Boilerplate options
outdir = null
publish_dir_mode = 'copy'
email = null
email_on_fail = null
plaintext_email = false
monochrome_logs = false
hook_url = null
help = false
version = false

// Config options
config_profile_name = null
config_profile_description = null
custom_config_version = 'master'
custom_config_base = "https://raw.githubusercontent.com/nf-core/configs/${params.custom_config_version}"
config_profile_contact = null
config_profile_url = null

// Max resource options
// Defaults only, expecting to be overwritten
max_memory = '128.GB'
max_cpus = 16
max_time = '240.h'

// Schema validation default options
validationFailUnrecognisedParams = false
validationLenientMode = false
validationSchemaIgnoreParams = 'genomes,igenomes_base'
validationShowHiddenParams = false
validate_params = true

}

// Load base.config by default for all pipelines
includeConfig 'conf/base.config'

// Load nf-core custom profiles from different Institutions
try {
    includeConfig "${params.custom_config_base}/nfcore_custom.config"
} catch (Exception e) {
    System.err.println("WARNING: Could not load nf-core/config profiles: ${params.custom_config_base}/nfcore_custom.config")
}

// Load nf-core/rnasplice custom profiles from different institutions.
// Warning: Uncomment only if a pipeline-specific institutional config already exists on nf-core/configs!
// try {
//     includeConfig "${params.custom_config_base}/pipeline/rnasplice.config"
// } catch (Exception e) {
//     System.err.println("WARNING: Could not load nf-core/config/rnasplice profiles: ${params.custom_config_base}/pipeline/rnasplice.config")
// }

profiles {
    debug {
        dumpHashes = true
        process.beforeScript = 'echo $HOSTNAME'
        cleanup = false
        nextflow.enable.configProcessNamesValidation = true
    }
    conda {
        conda.enabled = true
        docker.enabled = false
        singularity.enabled = false
        podman.enabled = false
        shifter.enabled = false
        charliecloud.enabled = false
        apptainer.enabled = false
    }
    mamba {
        conda.enabled = true
        conda.useMamba = true
        docker.enabled = false
        singularity.enabled = false
        podman.enabled = false
    }
}

```

```

shifter.enabled      = false
charliecloud.enabled = false
apptainer.enabled   = false
}
docker {
    docker.enabled      = true
    conda.enabled       = false
    singularity.enabled = false
    podman.enabled     = false
    shifter.enabled    = false
    charliecloud.enabled = false
    apptainer.enabled  = false
    docker.runOptions   = '-u $(id -u):$(id -g)'
}
arm {
    docker.runOptions   = '-u $(id -u):$(id -g) --platform=linux/amd64'
}
singularity {
    singularity.enabled      = true
    singularity.autoMounts  = true
    conda.enabled            = false
    docker.enabled           = false
    podman.enabled           = false
    shifter.enabled          = false
    charliecloud.enabled     = false
    apptainer.enabled        = false
}
podman {
    podman.enabled      = true
    conda.enabled       = false
    docker.enabled     = false
    singularity.enabled = false
    shifter.enabled    = false
    charliecloud.enabled = false
    apptainer.enabled  = false
}
shifter {
    shifter.enabled      = true
    conda.enabled       = false
    docker.enabled     = false
    singularity.enabled = false
    podman.enabled     = false
    charliecloud.enabled = false
    apptainer.enabled  = false
}
charliecloud {
    charliecloud.enabled = true
    conda.enabled       = false
    docker.enabled     = false
    singularity.enabled = false
    podman.enabled     = false
    shifter.enabled    = false
    apptainer.enabled  = false
}
apptainer {
    apptainer.enabled      = true
    apptainer.autoMounts  = true
    conda.enabled          = false
    docker.enabled         = false
    singularity.enabled   = false
    podman.enabled        = false
    shifter.enabled       = false
    charliecloud.enabled  = false
}
gitpod {
    executor.name      = 'local'
    executor.cpus      = 4
    executor.memory    = 8.GB
}
test           { includeConfig 'conf/test.config' }
test_full      { includeConfig 'conf/test_full.config' }
test_edger     { includeConfig 'conf/test_edger.config' }
test_rmats     { includeConfig 'conf/test_rmats.config' }
test_dexseq    { includeConfig 'conf/test_dexseq.config' }
test_suppa     { includeConfig 'conf/test_suppa.config' }
test_fastq    { includeConfig 'conf/test_fastq.config' }

```

```

test_genome_bam      { includeConfig 'conf/test_genome_bam.config'      }
test_transcriptome_bam { includeConfig 'conf/test_transcriptome_bam.config' }
test_salmon_results   { includeConfig 'conf/test_salmon_results.config'    }

}

// Set default registry for Apptainer, Docker, Podman and Singularity independent of -profile
// Will not be used unless Apptainer / Docker / Podman / Singularity are enabled
// Set to your registry if you have a mirror of containers
apptainer.registry   = 'quay.io'
docker.registry      = 'quay.io'
podman.registry      = 'quay.io'
singularity.registry = 'quay.io'

// Nextflow plugins
plugins {
    id 'nf-validation@1.1.3' // Validation of pipeline parameters and creation of an input channel from a sample sheet
}

// Load igenomes.config if required
if (!params.igenomes_ignore) {
    includeConfig 'conf/igenomes.config'
} else {
    params.genomes = [:]
}

// Export these variables to prevent local Python/R libraries from conflicting with those in the container
// The JULIA depot path has been adjusted to a fixed path '/usr/local/share/julia' that needs to be used for packages in the
// container.
// See https://apeltzer.github.io/post/03-julia-lang-nextflow/ for details on that. Once we have a common agreement on where
// to keep Julia packages, this is adjustable.

env {
    PYTHONNOUSERSITE = 1
    R_PROFILE_USER   = "./Rprofile"
    R_ENVIRON_USER   = "./Renviron"
    JULIA_DEPOT_PATH = "/usr/local/share/julia"
}

// Capture exit codes from upstream processes when piping
process.shell = ['/bin/bash', '-euo', 'pipefail']

// Disable process selector warnings by default. Use debug profile to enable warnings.
nextflow.enable.configProcessNamesValidation = false

def trace_timestamp = new java.util.Date().format( 'yyyy-MM-dd_HH-mm-ss')
timeline {
    enabled = true
    file    = "${params.outdir}/pipeline_info/execution_timeline_${trace_timestamp}.html"
}
report {
    enabled = true
    file    = "${params.outdir}/pipeline_info/execution_report_${trace_timestamp}.html"
}
trace {
    enabled = true
    file    = "${params.outdir}/pipeline_info/execution_trace_${trace_timestamp}.txt"
}
dag {
    enabled = true
    file    = "${params.outdir}/pipeline_info/pipeline_dag_${trace_timestamp}.html"
}

manifest {
    name          = 'nf-core/rnasplice'
    author        = """Ben Southgate, James Ashmore"""
    homePage      = 'https://github.com/nf-core/rnasplice'
    description   = """Alternative splicing analysis using RNA-seq."""
    mainScript    = 'main.nf'
    nextflowVersion = '!>=23.04.0'
    version       = '1.0.4'
    doi           = '10.5281/zenodo.8424632'
}

// Load modules.config for DSL2 module specific options
includeConfig 'conf/modules.config'

```

```

// Function to ensure that resource requirements don't go beyond
// a maximum limit
def check_max(obj, type) {
    if (type == 'memory') {
        try {
            if (obj.compareTo(params.max_memory as nextflow.util.MemoryUnit) == 1)
                return params.max_memory as nextflow.util.MemoryUnit
            else
                return obj
        } catch (all) {
            println "### ERROR ### Max memory '${params.max_memory}' is not valid! Using default value: $obj"
            return obj
        }
    } else if (type == 'time') {
        try {
            if (obj.compareTo(params.max_time as nextflow.util.Duration) == 1)
                return params.max_time as nextflow.util.Duration
            else
                return obj
        } catch (all) {
            println "### ERROR ### Max time '${params.max_time}' is not valid! Using default value: $obj"
            return obj
        }
    } else if (type == 'cpus') {
        try {
            return Math.min( obj, params.max_cpus as int )
        } catch (all) {
            println "### ERROR ### Max cpus '${params.max_cpus}' is not valid! Using default value: $obj"
            return obj
        }
    }
}

```

The Nextflow script for the pipeline and other scripts it calls

Find at <https://github.com/nf-core/rnasplice/blob/1.0.4/workflows/rnasplice.nf>

The other scripts, specific to certain algorithms/steps, such as STAR alignment, can be found either in the "subflows -> local" or "subflows -> nf-core" folders.

In the "local" subflows, we have the scripts for:

- |[align_star.nf](#)
- |[contrasts_check.nf](#)
- |[dexseq_deu.nf](#)
- |[drimseq_dexseq_dtu.nf](#)
- |[edger_deu.nf](#)
- |[input_check.nf](#)
- |[prepare_genome.nf](#)
- |[rmats.nf](<https://github.com/nf-core/rnasplice/blob/1.0.4/subworkflows/local/rmats.nf>)
- |[suppa.nf](#)
- |[tx2gene_tximport.nf](#)
- |[visualise_miso.nf](#)

These are all pipeline-specific processes. There are also processes that are used by many of the nf-core pipelines, which are stored in the "nf-core" workflows:

bam_sort_stats_samtools
bam_stats_samtools
bedgraph_bedclip_bedgraphtobigwig
fastq_fastqc_umi_tools_trimgalore

Initial creation of genome and transcriptome indices

STAR independent installation and mouse genome index

I decided to create my STAR and Salmon indices in advance in order to remove potential sources of pipeline getting stuck, so I had to install both independent of the pipeline.

From the [horse's scientist's mouth](#) who developed STAR, Alexander Dobin, I took the instructions on how to create one's own STAR index, and did so before running the pipeline (see also [here](#))

To first get STAR separately from the Nextflow pipeline, I went into my Nextflow environment and ran:

```
wget https://github.com/alexdobin/STAR/archive/2.7.11b.tar.gz
tar -xzf 2.7.11b.tar.gz
cd STAR-2.7.11b/source
```

Then, as instructed, I ran `make STAR` for the gcc C++ compiler to create a ready-to-run version of STAR on my system. Started around 11:57, took around 3 min to complete.

Got a warning "make: warning: Clock skew detected. Your build may be incomplete."

I reset my system time with `sudo apt-get install ntp` and `sudo service ntp restart`, then re-made the STAR build with `make clean` and `make STAR`, which now worked with no further errors.

I had to add the path to the folder where I compiled the executable to the PATH variable, which I did by adding to my `.bashrc` `export PATH=/home/iweber/Documents/Software/STAR-2.7.11b/source:$PATH`

To make the mouse STAR index, I ran:

```
STAR --runThreadN 6 --runMode genomeGenerate --genomeDir /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus --genomeFastaFiles /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna --sjdbGTFfile /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gtf --sjdbOverhang 149
```

Annnnnnnnd success! Took all of 27 min to complete.

```
STAR --runThreadN 6 --runMode genomeGenerate --genomeDir /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus --genomeFastaFiles /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna --sjdbGTFfile /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gtf --sjdbOverhang 149
    STAR version: 2.7.11b  compiled: 2024-08-07T12:28:19+02:00 :/home/iweber/Documents/Software/STAR-2.7.11b/source
Aug 07 17:23:32 ..... started STAR run
Aug 07 17:23:32 ... starting to generate Genome files
Aug 07 17:24:02 ..... processing annotations GTF
Aug 07 17:24:24 ... starting to sort Suffix Array. This may take a long time...
Aug 07 17:24:31 ... sorting Suffix Array chunks and saving them to disk...
Aug 07 17:39:52 ... loading chunks from disk, packing SA...
Aug 07 17:41:12 ... finished generating suffix array
Aug 07 17:41:12 ... generating Suffix Array index
Aug 07 17:44:38 ... completed Suffix Array index
Aug 07 17:44:39 ..... inserting junctions into the genome indices
Aug 07 17:48:41 ... writing Genome to disk ...
Aug 07 17:48:44 ... writing Suffix Array to disk ...
Aug 07 17:49:05 ... writing SAindex to disk
Aug 07 17:49:07 ..... finished successfully
```

Salmon installation and Salmon index for the mouse transcriptome

For the Salmon index creation, I first got the latest Salmon version independent of the Nextflow pipeline as instructed on the [Salmon website](#) using

```
wget https://github.com/COMBINE-lab/salmon/releases/download/v1.10.0/salmon-1.10.0_linux_x86_64.tar.gz
```

I decompressed the tar archive with `tar -xvzf salmon-1.10.0_linux_x86_64.tar.gz`, and then added the path to Salmon's bin folder to my PATH variable by including `export PATH=/home/iweber/Documents/Software/salmon-latest_linux_x86_64/bin:$PATH` in my `.bashrc` file. I could then see the command autocomplete, so I knew it works on my system :) `salmon --version` also returned "1.10.0", which is correct.

I also tried to get a Docker image with

```
sudo docker pull combinelab/salmon
```

which...did not seem to download anything to my "Software" folder that I ran the command in ***pondering smiley***. I'm sure this rather has something to do with me not understanding how the Docker image is supposed to work.

Next, I went looking for information on how to set up the transcriptomic index with Salmon. The [Salmon manual](#) sent me to this [RefGenie server/repository](#), that has ready-made indices for well-studied species, and it indeed hosts a Salmon index for the mouse, matching the NCBI genome, [here](#). I chose to download the complete [Salmon index generated with selective alignment method](#), because the description says that using it will increase the accuracy of quantification.

I made a new folder, Salmon_index_M_musculus_mm10, in the genome folder (/mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_M_musculus/GCF_000001635.27/), then used wget to download the index from RefGenie:

```
wget http://refgenomes.databio.org/v3/assets/archive/0f10d83b1050c08dd53189986f60970b92a315aa7a16a6f1/salmon_sa_index?tag=default
```

which...didn't work. Even though I saw the 12 GB download and it took around 30 min to do so, I had no files at the end in my working directory and only an odd file that popped up with ls -lha:

```
s/genome_M_musculus/GCF_000001635.27/Salmon_index_M_musculus_mm10$ ls -lha
ls: cannot access 'salmon_sa_index?tag=default': No such file or directory
total 4,0K
drwxrwxrwx 1 root root    0 Aug  7 19:11 .
drwxrwxrwx 1 root root 4,0K Aug  7 19:09 ..
-????????? ? ? ? ? ? ? ? ? 'salmon_sa_index?tag=default'
```

ChatGPT told me that the file with the utterly weird privilege indicators (question marks) and the truncated name is likely the fault of a corrupted download, and that that likely happened because there are special characters (question mark) in the archive name on the website. So it suggested re-downloading the archive with a clearly defined name using

```
wget "http://refgenomes.databio.org/v3/assets/archive/0f10d83b1050c08dd53189986f60970b92a315aa7a16a6f1/salmon_sa_index?
tag=default" -O salmon_sa_index.tgz
```

...but that only downloaded a tiny file, nothing close to the 12 GB I was expecting from the tgz archive. So I went the new-school way and just downloaded it through the browser *sweat drop smiley*

Side note: I found out later that RefGenie is actually a Python package that allows downloading files related to some reference genome assemblies from the command line; [here's how to use it](#) for further reference.

Prerequisites for running the rnassplice pipeline

Create contrast sheet

--contrasts contrastssheet.csv: I first need to create a contrast sheet, which tells the pipeline what kind of an experiment has been performed and how the conditions are called.

🔗 Straight from the pipeline's "contrastsheet input" section:

The contrastsheet has to be a comma-separated file with 3 columns, and a header row as shown in the examples below.

```
contrast,treatment,control
TREATMENT_CONTROL,TREATMENT,CONTROL
```

The contrastsheet can have as many columns as you desire, however, there is a strict requirement for the first 3 columns to match those defined in the table below.

Column	Description
contrast	An arbitrary identifier, will be used to name contrast-wise output files.
treatment	The treatment/target level for the comparison.
control	The control/base level for the comparison.

Therefore, my contrastssheet_preeclampsia.csv looks like:

```
contrast,treatment,control
PREECLAMPSIA_CONTROL,PREECLAMPSIA,CONTROL
```

Creating the sample sheet

--input samplesheet.csv: Next, I created a sample sheet (a .csv file telling the pipeline what the sample files are called and which ones are from the control and which ones from the treated animals). [Note from "Source configuration": when using FastQ files as input, file must be structured exactly as shown here]

Straight from the pipeline's "samplesheet input" section:

```
sample,fastq_1,fastq_2,strandedness,condition
CONTROL_REPO1,AEG588A1_S1_L002_R1_001.fastq.gz,AEG588A1_S1_L002_R2_001.fastq.gz,forward,control
CONTROL_REPO2,AEG588A2_S2_L002_R1_001.fastq.gz,AEG588A2_S2_L002_R2_001.fastq.gz,forward,control
CONTROL_REPO3,AEG588A3_S3_L002_R1_001.fastq.gz,AEG588A3_S3_L002_R2_001.fastq.gz,forward,control'''
```

The samplesheet can have as many columns as you desire, however, there is a strict requirement for at least 3 columns to match those defined in the table below.

Column	Description	
sample	Custom sample name. This entry will be identical for multiple sequencing libraries/runs from the same sample. Spaces in sample names are automatically converted to underscores (_).	
fastq_1	Full path to FastQ file for Illumina short reads 1. File has to be gzipped and have the extension ".fastq.gz" or ".fq.gz".	
fastq_2	Full path to FastQ file for Illumina short reads 2. File has to be gzipped and have the extension ".fastq.gz" or ".fq.gz".	
strandedness	Sample strand-specificity. Must be one of unstranded, forward or reverse.	
condition	The name of the condition a sample belongs to (e.g. 'control', or 'treatment') - these labels will be used for downstream analysis.	
genome_bam	Full path to aligned BAM file, derived from splicing aware mapper (STAR, HiSat, etc). File has to be in ".bam" format.	
transcriptome_bam	Full path to aligned transcriptome file, derived from splicing aware mapper (STAR, HiSat, etc). File has to be in ".bam" format.	
salmon_results	Full path to the result folder produced by salmon quantification.	"

I created the sample sheet in Notepad++, and I even remembered to set the line breaks to Unix line breaks (LF instead of CR LF).

	1	2	5	6	9	
	Run	BioSample	Experiment	GEO_Accession	tissue	
1	SRR13761520	SAMN18024800	SRX10148223	GSM5098819	Cortex offspring from control mother mice	
2	SRR13761521	SAMN18024799	SRX10148224	GSM5098820	Cortex offspring from control mother mice	
3	SRR13761522	SAMN18024798	SRX10148225	GSM5098821	Cortex offspring from control mother mice	
4	SRR13761523	SAMN18024797	SRX10148226	GSM5098822	Cortex offspring from control mother mice	
5	SRR13761524	SAMN18024796	SRX10148227	GSM5098823	Cortex offspring from preeclampsia mother mice	
6	SRR13761525	SAMN18024795	SRX10148228	GSM5098824	Cortex offspring from preeclampsia mother mice	
7	SRR13761526	SAMN18024794	SRX10148229	GSM5098825	Cortex offspring from preeclampsia mother mice	
8	SRR13761527	SAMN18024793	SRX10148230	GSM5098826	Cortex offspring from preeclampsia mother mice	

Therefore, my samplesheet_preeclampsia.csv looks like:

```
sample,fastq_1,fastq_2,strandedness,condition
CONTROL_REPO1,/home/iweber/Documents/Backup_shared_folder/Pre-
eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761520_1.fastq.gz,reverse,CONTROL
CONTROL_REPO2,/home/iweber/Documents/Backup_shared_folder/Pre-
eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761521_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761521_2.fastq.gz,reverse,CONTROL
CONTROL_REPO3,/home/iweber/Documents/Backup_shared_folder/Pre-
eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761522_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761522_2.fastq.gz,reverse,CONTROL
CONTROL_REPO4,/home/iweber/Documents/Backup_shared_folder/Pre-
eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761523_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761523_2.fastq.gz,reverse,CONTROL
PREECLAMPSIA_REPO1,/home/iweber/Documents/Backup_shared_folder/Pre-
eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761524_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761524_2.fastq.gz,reverse,PREECLAMPSIA
PREECLAMPSIA_REPO2,/home/iweber/Documents/Backup_shared_folder/Pre-
eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761525_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclampia_dataset_raw_and_processed/Pre_eclampia_mice_raw_fastq/SRR13761525_2.fastq.gz,reverse,PREECLAMPSIA
PREECLAMPSIA_REPO3,/home/iweber/Documents/Backup_shared_folder/Pre-
```

```
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761526_1.fastq.gz,/home/iweber/Documents/Backup_shared_f  
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761526_2.fastq.gz,reverse,PREECLAMPSIA  
PREECLAMPSIA REP4,/home/iweber/Documents/Backup_shared_folder/Pre-  
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761527_1.fastq.gz,/home/iweber/Documents/Backup_shared_f  
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761527_2.fastq.gz,reverse,PREECLAMPSIA
```

Stranded information?

But where is the strandedness of the libraries indicated? There is no mention of this neither in the publication, nor in its Supplemental Data 1, nor in the Supplementary Materials and Methods. I only found a mention in their regular Materials and Methods section that they used the TruSeq Stranded mRNA Library Prep Kit from Illumina, in whose [protocol](#) I read that it encompasses a first-strand reverse transcription and then the generation of the second strand of the cDNA, complementary to the first. This means that the resulting cDNA has the forward strand identical to the reverse complement of the original mRNA and so should be treated as "reverse" and specified as such in the subsequent process. However, how can I make super sure that this is true? Giving the wrong strandedness will render the entire analysis essentially useless, this is highly important.

[explain the underlying biology]

Side note: how does the kit distinguish between what is the first strand (reverse strand) and the forward strand? According to the TruSeq protocol, this is done by using a deoxyribonucleotide in the second-strand preparation that is different from the first-strand preparation (dUTP instead of dTTP for the first strand - they will both hybridize with adenine, but the presence of Ts in one cDNA strand and the presence of Us in the other makes it possible to tell apart which strand the reads stem from)

The blessings of online searching pointed me to a Python package called [How are we stranded here](#), which analyzes fastq files precisely to understand this. The knack: it also needs some additional information about the organism at hand, such as a gtf annotation file and a kallisto transcriptomic index. This seems like a very complicated option, so I asked [Joana Lemnian](#) whether that is necessary. She said no, because it can only be that the _R1 FastQ files are reverse and _R2 files forward, so the strandedness should be specified as RF (she also sent me some useful resources: [this one from ECSeq](#) and [this from the Broad Institute](#))

One question remained in this context, though: how should one specify this in the sample sheet? It only takes one single indication of strandedness for each sample and both of its _R1 and _R2 files with reads together, and that indication can only be either forward, reverse, or unstranded. I realized nf-core has its very own Slack channel, where I swiftly got a detailed answer from [Thomas Danhorn](#), indicating that the strandedness in this case has to follow that of the R1 file, so, in my case, be reverse.

This is how the sample sheet looks like in the end:

	sample	fastq_1	fastq_2	strandedness	condition
1	CONTROL REP1	SRR13761520_1.fastq	SRR13761520_1.fastq	reverse	CONTROL
2	CONTROL REP2	SRR13761521_1.fastq	SRR13761521_2.fastq	reverse	CONTROL
3	CONTROL REP3	SRR13761522_1.fastq	SRR13761522_2.fastq	reverse	CONTROL
4	CONTROL REP4	SRR13761523_1.fastq	SRR13761523_2.fastq	reverse	CONTROL
5	PREECLAMPSIA REP1	SRR13761524_1.fastq	SRR13761524_2.fastq	reverse	PREECLAMPSIA
6	PREECLAMPSIA REP2	SRR13761525_1.fastq	SRR13761525_2.fastq	reverse	PREECLAMPSIA
7	PREECLAMPSIA REP3	SRR13761526_1.fastq	SRR13761526_2.fastq	reverse	PREECLAMPSIA
8	PREECLAMPSIA REP4	SRR13761527_1.fastq	SRR13761527_2.fastq	reverse	PREECLAMPSIA

Configuration of the type of source files

Since I have FastQ files, I can leave the source configuration to the default, --source fastq.

Gzipping FastQ files for pipeline with pigz

I noticed in the parameters that the pipeline doesn't work with FastQ files directly, but only with their gzipped versions. I started the `gzip SRR13761520_1.fastq` command at 13:09, took around 10 min. So I know that doing the same with the remaining 15 files in sequence should take something like 150 minutes ~ 3 h. Which is why I'll only start after my recovery drive is complete, I switch off hyper-V, and I dare to restart my PC to see if it is indeed off and confirm my system didn't turn into blue pulp (BSOD).

The script I want to use to gzip all files at once: ([pigz, which allows hyperthreading, won't work on my VirtualBox VM because of the same issues with GPU passthrough, so I'll have to take this slow route...](#)):

```
gzip_several_fastqs.sh  
#!/bin/bash  
  
# Specify directory containing FASTQ files  
DIRECTORY="/mnt-win-ubu-shared/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq"  
  
# Find all .fastq files in the specified directory and subdirectories. Write line by line, pipe each line with one file name
```

```

into while loop that compresses it.
find "$DIRECTORY" -type f -name "*.*.fastq" | while read -r FILE; do
    echo "Compressing $FILE"
    gzip "$FILE"
done

```

Scratch the above, I finally fixed all of these issues and now can use pigz, taking advantage of its parallel processing capabilities. So I need only replace the command with pigz:

```

pigz_several_fastqs.sh
#!/bin/bash

# Specify directory containing FASTQ files
DIRECTORY="/mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/"

# Find all .fastq files in the specified directory and subdirectories. Write line by line, pipe each line with one file name
# into while loop that compresses it.
find "$DIRECTORY" -type f -name "*.*.fastq" | while read -r FILE; do
    echo "Compressing $FILE"
    pigz "$FILE"
done

```

GTF file for the mouse mm10 genome

rnas splice also needs a GTF annotation file that indicates where genes, transcripts, exons, etc are located within the mouse genome. Turns out, I had already gotten the mouse gtf a few months back, when I downloaded the mouse genome from NCBI (https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF_000001635.27_GRCm39/). This is good because I have the latest assembly of the mouse genome from NCBI, so it is more consistent to also use the NCBI annotations, especially since I am interested in looking at potentially disease-relevant genes, and I know NCBI has very good integration of its genomic data with its clinical databases, whereas Ensembl is somewhat better at having extremely detailed annotations, which come in handy for comparative/evolutionary studies.

However, the annotation file from NCBI is in [GFF3](#) format, so I might need to convert it to GTF. I found an overview of what these two file types are made of at [Ensembl](#), and it seems that GTF2 and GFF2 files are identical. BUT: reading on on the parameters page, I found that -gff is also an option that allows the use of GFF files instead of GTF, so yay for no conversion needed!

Even better: the GTF file is also included in the NCBI FTP with downloads for this genome assembly.

To get the mouse GTF from the NCBI FTP (already had the genome from them), used `wget`

`https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF_000001635.27_GRCm39/GCF_000001635.27_GRCm39_genomic.gtf.gz` and then `pigz -d` to unzip it.

Pipeline pilot experiment: *C. elegans* data from the course

Side note for the future:

there is a test dataset for this pipeline (human immortalized cell lines, paired-end RNA-seq data from [Shen et al., 2014](#) - the original rMATS paper) that can be called with

```
nextflow run nf-core/rnassplice -profile test_full,<docker/singularity/institute> --outdir <OUTDIR>
```

The *C. elegans* data

In the NGS part of the bioinformatics course, we mapped some reads from the *C. elegans* genome. I'd like to use these as a test dataset for the pipeline, because the data consisted of far fewer reads than what I have for my mouse experiment, and so I should be able to see fast if the pipeline works or not.

I have the reads files stored as fastq.gz files under

```
/mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Mapping_exercise_C_elegans/Fastq_original_C_elegans/
```

and the *C. elegans* genome stored under

```
/mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna
```

C. elegans GTF file

To get the GTF file, an annotation file that specifies where genes and other genomic elements start and end within the genome, I went to Ensembl's FTP page and found that the *C. elegans* GTFs are stored under https://ftp.ensembl.org/pub/current_gtf/caenorhabditis_elegans/Caenorhabditis_elegans.WBcel235.112.gtf.gz. I navigated to my *C. elegans* genome folder and downloaded the file with

```
wget https://ftp.ensembl.org/pub/current_gtf/caenorhabditis_elegans/Caenorhabditis_elegans.WBcel235.112.gtf.gz
```

For the relatively small *C. elegans* genome, this took all of 1 minute. I then unzipped it with pigz:

```
pigz -d Caenorhabditis_elegans.WBcel235.112.gtf.gz
```

STAR index for *C. elegans*

I tried to prepared the STAR index for *C. elegans* using

```
STAR --runThreadN 6 --runMode genomeGenerate --genomeDir /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/STAR_index_C_elegans --genomeFastaFiles /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna --sjdbGTFfile /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/Caenorhabditis_elegans.WBcel235.112.gtf --sjdbOverhang 149
```

but got an error saying something is not right with the GTF file: "Fatal INPUT FILE error, no valid exon lines in the GTF file: /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/Caenorhabditis_elegans.WBcel235.112.gtf

Solution: check the formatting of the GTF file. One likely cause is the difference in chromosome naming between GTF and FASTA file."

As I learned from the Harvard Chan Bioinformatics Core [course on RNA-Seq](#), I need to use the GFF file from NCBI instead, or, at the very least, rename the chromosomes in the GTF files so that they have the same names as they do in the genome. GTF files and genomic files need to be not only from the same build of the genome (so mm10, hg38, etc) but also from the same source, such as NCBI or Ensembl, because the different databases have somewhat different ways of formatting the names of elements in their genomes, such as the names of chromosomes.

So I got the GTF file from NCBI with wget

https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/002/985/GCF_000002985.6_WBcel235/GCF_000002985.6_WBcel235_genomic.gtf.gz, then unzipped it with pigz. I then repeated the index creation using the GTF file from NCBI:

```
STAR --runThreadN 6 --runMode genomeGenerate --genomeDir /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/STAR_index_C_elegans --genomeFastaFiles /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna --sjdbGTFfile /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gtf --sjdbOverhang 149
```

It worked, except for a warning: "!!!! WARNING: --genomeSAindexNbases 14 is too large for the genome size=100286401, which may cause seg-fault at the mapping step. Re-run genome generation with recommended --genomeSAindexNbases 12"

I looked in the STAR manual to gain a better understanding of why this is happening. Apparently, this is because the *C. elegans* genome is very small, hence the request to run the index generation again with this parameter set to 12. So I deleted all of the files that STAR generated for the index, and started afresh with:

```
STAR --runThreadN 6 --runMode genomeGenerate --genomeDir /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/STAR_index_C_elegans --genomeFastaFiles /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna --sjdbGTFfile /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gtf --sjdbOverhang 149 --genomeSAindexNbases 12
```

And ta-da! Two minutes later, with no more warnings, I had the index, a set of files that total some 1.4 GB in size **insert meow_party_smiley**

Salmon index for *C. elegans*

I had less luck with finding a pre-assembled *C. elegans* Salmon (transcriptomic) index than I had with the mouse one, so I needed to build this index from scratch.

Getting transcriptome file for *C. elegans*

For this, according to the Salmon manual, I need a file with all of the annotated *C. elegans* transcripts, so something with RNA or cDNA in the name. I went to NCBI's [genome FTP repository for *C. elegans*](#) and found two files, called "[GCF_000002985.6_WBcel235_rna.fna.gz](#)" and "[GCF_000002985.6_WBcel235_rna_from_genomic.fna.gz](#)", 13 and 14 MB in size, respectively. Which one to use, though? I checked the [FTP FAQ page](#) of the NCBI, but the descriptions about the two files are not necessarily helpful. Since this is just a pilot experiment, I decided to go with the simple rna.fna.gz file and downloaded it with wget.

To do:

clarify this for the future

Getting dependencies for index building

MashMap

To create the index, the Salmon manual talks about two ways to build a so-called *decoy-aware transcriptome*, and, for that, it first needs to build something called a *decoy file*. To do so, it needs a tool called [MashMap](#). I already had GCC and g++, which are needed for compiling the MashMap binaries, but didn't have cmake.

cmake

To get an installation script, I ran

```
wget https://github.com/Kitware/CMake/releases/download/v3.30.2/cmake-3.30.2-linux-x86_64.sh
```

then set the script to executable with `chmod u+x`, ran it with `./cmake-3.30.2-linux-x86_64.sh`, and added the path to the newly created binaries file to my PATH variable in `.bashrc`. I could then call it from the command line.

GNU GSL

MashMap also needs the GNU GSL tool, which I got in its latest stable version, 2.8, from <https://www.gnu.org/software/gsl/> as a tar archive, and immediately unpacked it. I went to the folder that generated, ran `./configure` to read my system properties, `make` to make a configuration suited to my system, which...took a moment, and many lines of code running in the background. It ended with the lines " `make[2]: Leaving directory '/home/iweber/Documents/Software/gsl-2.8'` `make[1]: Leaving directory '/home/iweber/Documents/Software/gsl-2.8'` , and, finally, `sudo make install` to run the installation program that was just compiled. I now have access to `gsl-config --version`, for instance :) The binary is in `/usr/local/bin`, so I did not need to add it to my PATH variable.

Building MashMap

I downloaded MashMap from its git repository with `git clone https://github.com/marbl/MashMap.git` and went into the newly-created directory. Then, as per the [instructions](#) on MashMap's GitHub repo, typed `cmake -H -Bbuild -DCMAKE_BUILD_TYPE=Release` and then `cmake --build build`. This is what I got:

```
(base) iweber@iweber-VirtualBox:~/Documents/Software/MashMap$ cmake --build build
[ 16%] Building CXX object CMakeFiles/mashmap.dir/src/common/utils.cpp.o
[ 33%] Building CXX object CMakeFiles/mashmap.dir/src/map/mash_map.cpp.o
[ 50%] Linking CXX executable bin/mashmap
[ 50%] Built target mashmap
[ 66%] Building CXX object CMakeFiles/mashmap-align.dir/src/common/utils.cpp.o
[ 83%] Building CXX object CMakeFiles/mashmap-align.dir/src/align/align.cpp.o
[100%] Linking CXX executable bin/mashmap-align
[100%] Built target mashmap-align
(base) iweber@iweber-VirtualBox:~/Documents/Software/MashMap$
```

Looking into the MashMap folders, I saw a `/bin` folder and added it to my PATH variable, and now I can call it from the command line.

Get BedTools

Got the archive with

```
wget https://github.com/arq5x/bedtools2/releases/download/v2.31.1/bedtools-2.31.1.tar.gz and unpacked it with tar -vxzf . I changed into the bedtools2 folder and compiled it with make . I got an error ending with
```

```
cram/cram_io.c:57:10: fatal error: bzlib.h: No such file or directory
57 | #include <bzlib.h>
   |          ^
compilation terminated.
make[1]: *** [Makefile:103: cram/cram_io.o] Error 1
```

```
make[1]: Leaving directory '/home/iweber/Documents/Software/bedtools2/src/utils/htslib'
make: *** [src/utils/htslib/htslib.mk:153: src/utils/htslib/libhts.a] Error 2'''
```

So I knew I had to get this bzlib. I tried with

```
sudo apt-get update
sudo apt-get install libbz2-dev
```

and it seems to have worked. Even though it increased my anxiety when I saw it was also getting something related to CUDA...no error messages or warnings in the end.

I cleaned up the previous build with `make clean`, restarted my terminal, and restarted the build with `make`. And got the same issue but for a different library, Izma.h. Got annoyed, cleaned the build, and, as per the [installation page of bedtools](#), decided to just go with a pre-compiled version, which I got using `sudo apt-get install bedtools`, which worked like a charm and installed it into my user binaries (`/usr/bin/`, as I found out with `which bedtools`).

Building the decoys for *C. elegans*

Get SalmonTools

To create the decoys, I got SalmonTools from GitHub with `git clone https://github.com/COMBINE-lab/SalmonTools.git` and went into the newly created directory, `SalmonTools`. ChatGPT advised to make a separate directory for building the compiled tool, so I did with `mkdir build` and changed into it. I configured the build with `cmake .`, and then made it with `make`. It was indeed made, but with quite some errors:

```
[ 7%] Creating directories for 'libspdlog'
[ 15%] Performing download step for 'libspdlog'
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0      0      0 --:--:-- --:--:-- --:--:--    0
100  147k    0  147k    0      0  121k      0 --:--:--  0:00:01 --:--:--  922k
spdlog-v0.12.0.tar.gz: OK
[ 23%] No update step for 'libspdlog'
[ 30%] No patch step for 'libspdlog'
[ 38%] No configure step for 'libspdlog'
[ 46%] No build step for 'libspdlog'
[ 53%] Performing install step for 'libspdlog'
[ 61%] Completed 'libspdlog'
[ 61%] Built target libspdlog
[ 69%] Building CXX object src/CMakeFiles/salmon_tools_core.dir/FastxParser.cpp.o
[ 76%] Building CXX object src/CMakeFiles/salmon_tools_core.dir/ExtractUnmapped.cpp.o
In file included from /home/iweber/Documents/Software/SalmonTools/include/zstr.hpp:16,
                 from /home/iweber/Documents/Software/SalmonTools/src/ExtractUnmapped.cpp:10:
/home/iweber/Documents/Software/SalmonTools/include/strict_fstream.hpp: In static member function 'static void
strict_fstream::detail::static_method_holder::check_peek(std::istream*, const string&, std::ios_base::openmode)':
/home/iweber/Documents/Software/SalmonTools/include/strict_fstream.hpp:128:39: warning: catching polymorphic type 'class
std::ios_base::failure' by value [-Wcatch-value=]
  128 |         catch (std::ios_base::failure e) {}
           |
           |         ^
/home/iweber/Documents/Software/SalmonTools/src/ExtractUnmapped.cpp: In function 'void ExtractUnmapped(const string&,
std::vector<std::basic_string<char> >::const_iterator, std::vector<std::basic_string<char>
>::const_iterator)':
/home/iweber/Documents/Software/SalmonTools/src/ExtractUnmapped.cpp:210:18: warning: catching polymorphic type 'class
args::Help' by value [-Wcatch-value=]
  210 |         catch (args::Help)
           |
           |         ^
/home/iweber/Documents/Software/SalmonTools/src/ExtractUnmapped.cpp:215:29: warning: catching polymorphic type 'class
args::ParseError' by value [-Wcatch-value=]
  215 |         catch (args::ParseError e)
           |
           |         ^
In file included from /home/iweber/Documents/Software/SalmonTools/src/ExtractUnmapped.cpp:8:
/home/iweber/Documents/Software/SalmonTools/include/sparsepp/spp.h: In instantiation of 'void spp::sparseable<T,
Alloc>::resize(spp::sparseable<T, Alloc>::size_type) [with T = std::pair<const std::basic_string<char>,
std::basic_string<char> >; Alloc = spp::libc_allocator<std::pair<const std::basic_string<char>,
std::basic_string<char> >>; spp::sparseable<T, Alloc>::size_type = long unsigned int]':
/home/iweber/Documents/Software/SalmonTools/include/sparsepp/spp.h:2942:25:   required from 'void spp::sparse_hashtable<Value,
Key, HashFn, ExtractKey, SetKey, EqualKey, Alloc>::move_from(spp::sparse_hashtable<Value, Key, HashFn, ExtractKey, SetKey,
EqualKey, Alloc>::MoveDontCopyT, spp::sparse_hashtable<Value, Key, HashFn, ExtractKey, SetKey, EqualKey, Alloc>,
spp::sparse_hashtable<Value, Key, HashFn, ExtractKey, SetKey, EqualKey, Alloc>::size_type) [with Value = std::pair<const
std::basic_string<char>, std::basic_string<char> >; Key = std::basic_string<char>; HashFn =
spp::spp_hash<std::basic_string<char> >; ExtractKey = spp::sparse_hash_map<std::basic_string<char>,
std::basic_string<char> >::SelectKey; SetKey = spp::sparse_hash_map<std::basic_string<char>,
std::basic_string<char> >::SetKey; EqualKey = std::equal_to<std::basic_string<char> >; Alloc =
spp::libc_allocator<std::pair<const std::basic_string<char>, std::basic_string<char> >>;
```

Build decoys

I moved to the "scripts" folder in the main SalmonTools directory to find the script needed to build the decoys, "generateDecoyTranscriptome.sh". I gave myself execution privileges with `chmod u+x`.

To run the script, I used

```
cd /home/iweber/Documents/Software/SalmonTools/scripts/
./generateDecoyTranscriptome.sh -g /mnt/mnt-win-ubuntu-
shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna -t /mnt/mnt-win-ubuntu-
shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_rna.fna -a /mnt/mnt-win-ubuntu-
shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gtf -m
/home/iweber/Documents/Software/MashMap/build/bin/mashmap -b /usr/bin/bedtools -o /mnt/mnt-win-ubuntu-
shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/Salmon_decoys_C_elegans
```

This...kept not working. I got an error from MashMap saying

```
*****
*** getDecoy ***
*****
-g <Genome fasta> = /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna
-t <Transcriptome fasta> = /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_rna.fna
-a <Annotation GTF file> = /mnt/mnt-win-ubuntu-
shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gtf
-m <mashmap binary> = /home/iweber/Documents/Software/MashMap/build/bin/mashmap
-b <bedtools binary> = /usr/bin/bedtools
-o <Output files Path> = /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/Salmon_decoys_C_elegans
[1/10] Extracting exonic features from the gtf
[2/10] Masking the genome fasta
[3/10] Aligning transcriptome to genome
/home/iweber/Documents/Software/MashMap/build/bin/mashmap: error while loading shared libraries: libgsl.so.28: cannot open
shared object file: No such file or directory
```

```
*****
*** ABORTED ***
*****
```

An error occurred. Exiting...

Even after getting that library separately, knowing it's on my system with `find /usr -name "libgsl.so.28"` at `/usr/local/lib/libgsl.so.28`, adding that to my library PATH in .bashrc with `echo 'export LD_LIBRARY_PATH=/usr/local/lib/libgsl.so.28:$LD_LIBRARY_PATH' >> ~/.bashrc`, restarting the terminal, and re-building MashMap, it still would not find it and give the exact same error as above.

I then checked out this other [tutorial](#) from Salmon/Alevin, a tool that's part of Salmon. I tried a similar command to the suggested one:

```
grep '^>' GCF_000002985.6_WBcel235_rna.fna | cut -d " " -f 1 | sed 's/>/\n/g' > decoys.txt
```

and got a text file, decoys.txt, filled with one NCBI transcript identifier per row (NM_...).

Then, I adapted the command for creating the second file, the concatenated transcriptome and genome reference file:

```
cat GCF_000002985.6_WBcel235_rna.fna GCF_000002985.6_WBcel235_genomic.fna | pigz > gentrome.fa.gz
```

Building the decoy-aware transcriptome index for *C. elegans*

As on the [Alevin/Salmon tutorial page](#), I ran

```
salmon index -t gentrome.fa.gz -d decoys.txt -p 12 -i salmon_index --gencode
```

Which gave me many warnings identical to "[2024-08-08 14:09:10.141] [puff::index::jointLog] [warning] Entry with header [NR_131692.1], had length less than equal to the k-mer length of 31 (perhaps after poly-A clipping)" and one critical error saying "[2024-08-08 14:09:10.573] [puff::index::jointLog] [critical] Observed a non-decoy sequence [NC_003279.8] after having already observed a decoy. However, it is required that any decoy target records appear, consecutively, at the end of the input fasta file. Please re-format your input file so that all decoy records appear contiguously at the end of the file, after all valid (non-decoy) records".

This has something to do with the Note on the Alevin page mentioned above: "the genome targets (decoys) should come after the transcriptome targets in the reference"

I installed an app called Seqkit, which eases the handling of FastA and FastQ files (and a language it needs, Go: `sudo apt-get install golang`)

```
wget https://github.com/shenwei356/seqkit/releases/download/v2.8.2/seqkit_linux_amd64.tar.gz  
tar -vxzf seqkit_linux_amd64.tar.gz
```

This immediately popped out the executable for seqkit, so I added the path to it to the PATH variable.

With Seqkit, I performed

```
seqkit grep -v -f decoys.txt gentrome.fa.gz -o non_decoy.fa
```

to extract the non-decoy information from the compressed file I created before (concatenated transcriptome and genome reference file), and

```
seqkit grep -f decoys.txt gentrome.fa.gz -o decoy.fa
```

to extract specifically the decoys.

To re-assemble the compressed file in the correct order, I ran

```
cat non_decoy.fa decoy.fa | pigz > gentrome_corrected.fa.gz
```

And then attempted to build the index with the corrected file:

```
salmon index -t gentrome_corrected.fa.gz -d decoys.txt -p 12 -i salmon_index
```

(I chose to remove the --gencode option - the [Alevin/Salmon tutorial page](#) I got the command from says "NOTE: --gencode flag is for removing extra metadata in the target header separated by | from the gencode reference. You can skip it if using other references.", and I am using NCBI, not GENCODE references, so it should be safe?)

To ask ioana

ask if removing the GENCODE flag was indeed the correct move

The index creation took off at around 2:30 and mightily strained my CPU (~70% usage by the virtual machine, with occasional GPU use), but was done in less than 3 minutes, and I now have a folder called "salmon_index" at around 500 MB size.

If pipeline doesn't work, figure out in more detail what's happening here

The index finished building with the messages

```
[2024-08-08 14:29:48.453] [puff::index::jointLog] [info] Replaced 0 non-ATCG nucleotides  
[2024-08-08 14:29:48.453] [puff::index::jointLog] [info] Clipped poly-A tails from 22 transcripts wrote 37798 cleaned references [2024-08-08 14:29:51.446] [puff::index::jointLog] [info] Filter size not provided; estimating from number of distinct k-mers [2024-08-08 14:30:02.884]  
[puff::index::jointLog] [info] ntHll estimated 97277338 distinct k-mers, setting filter size to 2^31 Threads = 12 Vertex length = 31 Hash functions = 5  
Filter size = 2147483648 Capacity = 2 Files: salmon_index/ref_k31_fixed.fa ----- Round 0, 0:2147483648 Pass Filling Filtering 1 6 16 2 3 1 True junctions count = 522768 False junctions count = 794481 Hash table size = 1317249  
Candidate marks count = 4277706 ----- TBB Warning: The number of workers is currently limited to 5. The request for 11 workers is ignored. Further requests for more workers will be silently ignored until the limit changes. Reallocating bifurcations time: 0 True marks count: 2839486 Edges construction time: 165 ----- Distinct junctions = 522768 TwoPaCo::buildGraphMain:: allocated with scalable_malloc; freeing. TwoPaCo::buildGraphMain:: Calling scalable_allocation_command(TBBMALLOC_CLEAN_ALL_BUFFERS, 0); allowedIn: 15 Max Junction ID: 523362 seen.size():4186905  
kmerInfo.size():523363 approximateContigTotalLength: 38939053 counters for complex kmers: (prec>1 & succ>1)=28528 | (succ>1 & isStart)=70 | (prec>1 & isEnd)=77 | (isStart & isEnd)=31 contig count: 779268 element count: 120440207 complex nodes: 28706 # of ones in rank vector: 779267 [2024-08-08 14:33:20.452] [puff::index::jointLog] [info] Starting the Pufferfish indexing by reading the GFA binary file. [2024-08-08 14:33:20.452] [puff::index::jointLog] [info] Setting the index/BinaryGfa directory salmon_index size = 120440207 ----- | Loading contigs | Time = 33 ms ----- size = 120440207 ----- | Loading contig boundaries | Time = 17.347 ms ----- Number of ones: 779267 Number of ones per inventory item: 512 Inventory entries filled: 1523 779267 [2024-08-08 14:33:20.629] [puff::index::jointLog] [info] Done wrapping the rank vector with a rank9sel structure. [2024-08-08 14:33:20.635] [puff::index::jointLog] [info] contig count for validation: 779,267 [2024-08-08 14:33:20.719] [puff::index::jointLog] [info] Total # of Contigs : 779,267 [2024-08-08 14:33:20.719] [puff::index::jointLog] [info] Total # of numerical Contigs : 779,267 [2024-08-08 14:33:20.729] [puff::index::jointLog] [info] Total # of contig vec entries: 2,890,699 [2024-08-08 14:33:20.729] [puff::index::jointLog] [info] bits per offset entry 22 [2024-08-08 14:33:20.765] [puff::index::jointLog] [info] Done constructing the contig vector. 779268 [2024-08-08 14:33:21.424] [puff::index::jointLog] [info] # segments = 779,267 [2024-08-08 14:33:21.424] [puff::index::jointLog] [info] total length = 120,440,207 [2024-08-08 14:33:21.437] [puff::index::jointLog] [info] Reading the reference files ... [2024-08-08 14:33:21.922] [puff::index::jointLog] [info] positional integer width = 27 [2024-08-08 14:33:21.922] [puff::index::jointLog] [info] seqSize = 120,440,207 [2024-08-08 14:33:21.922] [puff::index::jointLog] [info] rankSize = 120,440,207 [2024-08-08 14:33:21.922] [puff::index::jointLog] [info] edgeVecSize = 0 [2024-08-08 14:33:21.922] [puff::index::jointLog] [info] num keys = 97,062,197 for info, total work write each : 2.331 total work inram from level 3 : 4.322 total work raw : 25.000 [Building BooPHF] 100 % elapsed: 0 min 4 sec remaining: 0 min 0 sec Bitarray 508580416 bits (100.00 %) (array + ranks ) final hash 0 bits (0.00 %) (nb in final hash
```

```
0) [2024-08-08 14:33:25.534] [puff::index::jointLog] [info] mphf size = 60.6275 MB [2024-08-08 14:33:25.626] [puff::index::jointLog] [info] chunk size = 10,036,684 [2024-08-08 14:33:25.626] [puff::index::jointLog] [info] chunk 0 = [0, 10,036,684) [2024-08-08 14:33:25.626]
[puff::index::jointLog] [info] chunk 1 = [10,036,684, 20,073,368) [2024-08-08 14:33:25.626] [puff::index::jointLog] [info] chunk 2 = [20,073,368, 30,110,075) [2024-08-08 14:33:25.626] [puff::index::jointLog] [info] chunk 3 = [30,110,075, 40,146,759) [2024-08-08 14:33:25.626]
[puff::index::jointLog] [info] chunk 4 = [40,146,759, 50,183,443) [2024-08-08 14:33:25.626] [puff::index::jointLog] [info] chunk 5 = [50,183,443, 60,220,129) [2024-08-08 14:33:25.627] [puff::index::jointLog] [info] chunk 6 = [60,220,129, 70,256,813) [2024-08-08 14:33:25.627]
[puff::index::jointLog] [info] chunk 7 = [70,256,813, 80,293,497) [2024-08-08 14:33:25.627] [puff::index::jointLog] [info] chunk 8 = [80,293,497, 90,330,181) [2024-08-08 14:33:25.627] [puff::index::jointLog] [info] chunk 9 = [90,330,181, 100,366,865) [2024-08-08 14:33:25.627]
[puff::index::jointLog] [info] chunk 10 = [100,366,865, 110,403,549) [2024-08-08 14:33:25.627] [puff::index::jointLog] [info] chunk 11 =
[110,403,549, 120,440,177) [2024-08-08 14:33:29.572] [puff::index::jointLog] [info] finished populating pos vector [2024-08-08 14:33:29.572]
[puff::index::jointLog] [info] writing index components [2024-08-08 14:33:29.987] [puff::index::jointLog] [info] finished writing dense pufferfish index
[2024-08-08 14:33:30.022] [jLog] [info] done building index ",
```

but I had a lot of warnings of the form

"R_131708.1], had length less than equal to the k-mer length of 31 (perhaps after poly-A clipping)",

and, at the end, the warnings

```
"[2024-08-08 14:29:48.451] [puff::index::jointLog] [warning] Removed 221 transcripts that were sequence duplicates of indexed transcripts. [2024-08-08 14:29:48.451] [puff::index::jointLog] [warning] If you wish to retain duplicate transcripts, please use the --keepDuplicates flag [2024-08-08 14:29:48.451] [puff::index::jointLog] [warning] There were 221 duplicate decoy sequences.".
```

I asked ChatGPT: where do these warnings come from, and is this index safe to use? Its answer was:

Understanding the Warnings During Salmon Indexing

The warnings you encountered during the Salmon indexing process provide information about potential issues or optimizations that Salmon performed on the input sequences. Let's break them down:

1. Warnings about Sequences Shorter than K-mer Length:

- **Warning:** R_131708.1], had length less than or equal to the k-mer length of 31 (perhaps after poly-A clipping)
- **Explanation:** Salmon uses k-mers (subsequences of length 31 by default) to build its index. If a sequence is shorter than this k-mer length (or exactly equal to it), it cannot be effectively used in the indexing process. This often happens with very short sequences or those that are clipped due to poly-A tail removal. Salmon flags these sequences because they might not contribute useful information to the index.
- **Impact:** These sequences are ignored in the index, but this generally shouldn't be an issue unless these sequences are critical to your analysis. The fact that they are clipped or short suggests they might not be biologically significant or that their inclusion wouldn't provide meaningful quantification data.

2. Warnings about Duplicate Transcripts and Decoys:

- **Warning:** Removed 221 transcripts that were sequence duplicates of indexed transcripts. and There were 221 duplicate decoy sequences.
- **Explanation:** During indexing, Salmon detected sequences that were exact duplicates either within the transcriptome or between the transcriptome and the decoy sequences. By default, Salmon removes these duplicates to avoid redundancy in the index.
- **Impact:** Removing duplicates typically optimizes the index by reducing its size and improving performance. However, if retaining these duplicates is important for your specific analysis, you can use the --keepDuplicates flag to prevent their removal.

Is the Index Safe to Use?

Given the details you provided, the index should generally be safe to use. The warnings indicate that Salmon performed optimizations by removing duplicates and ignoring sequences that couldn't be indexed due to their short length. These are standard procedures that should not negatively impact the majority of analyses.

However, consider the following:

- **Short Sequences:** If the sequences flagged for being shorter than the k-mer length are crucial to your research, you might want to review those sequences to ensure they are not important.
- **Duplicate Sequences:** If the presence of duplicate sequences (either in the transcriptome or decoys) is critical to your analysis, consider re-running the index with the --keepDuplicates flag.

Summary of Steps to Proceed:

- **Proceed with the Current Index:** If the warnings do not raise concerns for your specific analysis, proceed with using the current index.
- **Rebuild with Adjustments:** If you believe the short or duplicate sequences might be important, consider adjusting the Salmon command by including flags like --keepDuplicates and possibly revisiting the input FASTA files to ensure no critical sequences were removed.

You can now proceed with your analysis, but keep these considerations in mind if you encounter unexpected results or if the removed sequences were critical to your research objectives."

Sample sheet and contrasts sheet

I, of course, still need the sample sheet and the contrast sheet for this experiment as well. I have reads from two samples, CHS_1063 and CHS_1109. I tried searching for their names to see what kind of experiment they stemmed from and couldn't find any info on the quick, so I will simply assume that CHS_1063 is "control" and CHS_1109 is "treatment".

Here's how the sample sheet looks like:

```
sample,fastq_1,fastq_2,strandedness,condition
CONTROL_REP1,CHS_1063_R1.fastq.gz,CHS_1063_R2.fastq.gz,reverse,CONTROL
TREATMENT_REP1,CHS_1109_R1.fastq.gz,CHS_1109_R1.fastq.gz,reverse,TREATMENT
```

And the contrasts sheet:

```
contrast,treatment,control
TREATMENT_CONTROL,TREATMENT,CONTROL
```

Running pipeline for the *C. elegans* data

To run it for the *C. elegans* data, I will use

```
sudo nextflow run nf-core/rnasplice -r 1.0.4
--input /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/samplesheet_C_elegans.csv
--contrasts /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/contrastssheet_C_elegans.csv

--fasta /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna
~~~gtf /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/Caenorhabditis_elegans.WBcel235.112.gtf~~ # NO. USE
GFF INSTEAD, because it's for sure from NCBI as well.
--gff /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/

--save-trimmed true # to save the reads after trimming

--aligner star_salmon
--star_index /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/STAR_index_C_elegans/
--salmon_index /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/salmon_index/

--save-unaligned true # to save, if possible, any reads that couldn't be aligned in the results directory for later inspection
--save-align-intermeds # to save BAM files separately

--rmats true # to run rMATS part of pipeline
--outdir p/mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Nextflow_results_test_pipeline/
-profile docker
```

In compact form:

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 --input /mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/samplesheet_C_elegans.csv --contrasts /mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/contrastssheet_C_elegans.csv --fasta /mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna --gff /mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gff --save-trimmed true --aligner
star_salmon --star_index /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/STAR_index_C_elegans/
--salmon_index /mnt/mnt-win-ubu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/salmon_index/ --save-unaligned true --save-
align-intermeds true --rmats true --outdir p/mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Nextflow_results_test_pipeline/ -profile docker
```

I had to run the pipeline with sudo, likely because the data and workflow are located in my inter-OS shared folder. I got some errors saying "ERROR ~ .nextflow/plr/9b802ca2-0d20-4b86-8169-d1dc8af33a6e/nf-validation-1.1.3: Operation not supported", [suggesting that it cannot create so-called symlinks](#), and that may be due to the owner of the inter-OS folder being root, not my user. That didn't fix it, so I had to run the pipeline in my regular Documents folder in the Ubuntu OS, which then did run.

The pipeline ran for a moment and then, predictably, crashed :)

The first time when it crashed, I got an error . Some research in the GitHub issues of Nextflow revealed that this likely has something to do with Nextflow not being able to create

It took me a fine moment to scroll through all of the output to the end and see the error...

Command executed:

```
check_samplesheet_fastq.py samplesheet_C_elegans.csv samplesheet.valid.csv
cat <<-END_VERSIONS > versions.yml
"NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK":
    python: ${python --version | sed 's/Python //g'}
END_VERSIONS
```

Command exit status:
1

Command output:
(empty)

Command error:
Unable to find image 'quay.io/biocontainers/python:3.9--1' locally
3.9--1: Pulling from biocontainers/python
c1a16a04cedd: Pulling fs layer
4ca545ee6d5d: Pulling fs layer
5f7a6762e5ca: Pulling fs layer
4ca545ee6d5d: Verifying Checksum
4ca545ee6d5d: Download complete
c1a16a04cedd: Verifying Checksum
c1a16a04cedd: Download complete
c1a16a04cedd: Pull complete
4ca545ee6d5d: Pull complete
5f7a6762e5ca: Download complete
5f7a6762e5ca: Pull complete
Digest: sha256:d97d2b329b4e44d2e07a9737ba348b185d6a47f34fba0ef301d44d11669cac60
Status: Downloaded newer image for quay.io/biocontainers/python:3.9--1
[CRITICAL] The FASTQ file has an unrecognized extension: SRR13761520_1.fastq
It should be one of: .fq.gz, .fastq.gz On line 2.

Embarassing, but true: I had created the sample sheet for this pilot experiment in Notepad++ and forgot to hit save after changing the file names to the ones from the C. elegans experiment **insert monkey covering its eyes smiley** I hit save, and ran the same command again....

(Now, the .csv samplesheet looks as such:

```
sample,fastq_1,fastq_2,strandedness,condition
CONTROL_REPO1,/mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Fastq_original_C_elegans/CHS_1063_R1.fastq.gz,/mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Fastq_original_C_elegans/CHS_1063_R2.fastq.gz,reverse,CONTROL
TREATMENT_REPO1,/mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Fastq_original_C_elegans/CHS_1109_R1.fastq.gz,/mnt/mnt-win-ubu-
shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Fastq_original_C_elegans/CHS_1109_R2.fastq.gz,reverse,TREATMENT
```

)

The next error on the line was "ERROR ~ Error executing process >

'NFCORE_RNASPlice:RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:TRIMGALORE (CONTROL_REPO1)'

Caused by:

Process requirement exceeds available CPUs -- req: 12; avail: 6

Command executed:

```
[ ! -f CONTROL_REPO1_1.fastq.gz ] && ln -s CHS_1063_R1.fastq.gz CONTROL_REPO1_1.fastq.gz
[ ! -f CONTROL_REPO1_2.fastq.gz ] && ln -s CHS_1063_R2.fastq.gz CONTROL_REPO1_2.fastq.gz
trim_galore
--fastqc
--cores 8
--paired
--gzip
CONTROL_REPO1_1.fastq.gz
CONTROL_REPO1_2.fastq.gz

cat <<-END_VERSIONS > versions.yml
"NFCORE_RNASPlice:RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:TRIMGALORE":
trimgalore: $(echo $(trim_galore --version 2>&1) | sed 's/^.*version //; s>Last.*//')
cutadapt: $(cutadapt --version)
END_VERSIONS
```

Command exit status:

*

Command output:

(empty)

Work dir:
/home/iweber/Documents/work/28/40f5dd10796fb6d7bf9a03cf39569b

Tip: you can replicate the issue by changing to the process work dir and entering the command `bash .command.run`

-- Check '.nextflow.log' file for details

"

This clearly says that, normally, the pipeline would try to use 12 cores, and I only have 6 available on my virtual machine. Therefore, I need to limit the number of cores for all processes using `--max_cpus 6`:

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 --input /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/samplesheet_C_elegans.csv --contrasts /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/contrastssheet_C_elegans.csv --fasta /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna --gff /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gff --save-trimmed true --aligner star_salmon --star_index /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/STAR_index_C_elegans/ --salmon_index /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/salmon_index/ --save-unaligned true --save-align-intermeds true --rmats true --outdir p/mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Nextflow_results_test_pipeline/ --max-cpus 6 -profile docker
```

This did not work, raising the exact same error as before. I read on [Nextflow's website that this means I will have to modify its configuration files](#), and, since the page relates to the latest version of the program, I figured that, before that, I'd rather heed its warning to update the software in its entirety (when running the pipelines, it kept telling me I'm running a version 3 versions older than the current one, 24.04). I used `sudo nextflow self-update`, and it worked.

Configuration to use fewer cores

Nextflow has a basic configuration file, but can use customized configuration files as well to get parameters for how it runs pipelines.

In the so-called scopes, Nextflow can get grouped info about the variables related to a certain topic. For example, there is a scope/group of variables pertaining specifically controls [conda-related variables](#) when running the pipeline in conda. However, what I am likely needing are the variables related to the [executor scope](#). Specifically, I need to set `executor.cpus = 6` and `executor.memory = 94 GB`. There are other neat settings that only work on HPCs managed by, for example, the [SLURM](#) resource manager, such as `executor.perCpuMemAllocation`.

I also noticed that Nextflow tried to run each workflow three times. This is likely set by `executor.retry.maxAttempt`, that has a default of 3. However, I'm pretty sure that, if my PC can't handle something in the first go, it also won't in the second or third, so I will set `executor.retry.maxAttempt = 1`.

What could also be interesting is the [timeline scope](#). Here, one can ask for an execution timeline file to be generated with `timeline.enabled = true`, and that might give me more information

So, to sum up, what I need to change is:

```
executor{
    $local{
        cpus = 6
        memory = '94 GB'
        retry.maxAttempt = 1
    }
}

timeline.enabled = true
```

(I could also write this in the dot notation as `executor.$local.cpus = 6`, but why type more :)

Where does one find the configuration file of an nf-core pipeline, though? I checked here https://nf-co.re/docs/usage/getting_started/configuration (and I wish I had done sooner, I just saw now that there is the possibility to test pipelines with a minimal or large public dataset, using profile `test` or profile `test_full`).

When looking for the config files, I remembered I had also seen some warnings related to invalid options:

```
WARN: The following invalid input values have been detected:
```

```
* --save-trimmed: true
* --saveTrimmed: true
* --save-unaligned: true
* --saveUnaligned: true
* --save-align-intermeds: true
* --saveAlignIntermeds: true
```

```
* --max-cpus: 6
* --maxCpus: 6
```

For some reason, some of the underscores in the options have been replaced by regular dashes! So there's actually no need to tinker with the configuration file, just to use the correct parameter names **facepalm**

I restarted the pipeline at 8:54 with:

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 --input /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/samplesheet_C_elegans.csv --contrasts /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/contrastssheet_C_elegans.csv --fasta /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna --gff /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gff --save_trimmed true --aligner star_salmon --star_index /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/STAR_index_C_elegans/ --salmon_index /mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Genomes/genome_C_elegans/salmon_index/ --save_unaligned true --save_align_intermeds true --rmats true --outdir p/mnt/mnt-win-ubuntu-shared/Win_Ubuntu_shared/Pipeline_test_C_elegans/Nextflow_results_test_pipeline/ --max_cpus 6 -profile docker
```

It ran for around half an hour and failed at the edgeR step, I believe because of the GTF annotation file (GCF_000002985.6_WBcel235_genomic.gtf). The error message said:

```
ERROR ~ Error executing process > 'NFCORE_RNASPlice:RNASPLICE:EDGER_DEU:SUBREAD_FLATTENGTF (GCF_000002985.6_WBcel235_genomic.gtf)'
```

Caused by:

```
Process `NFCORE_RNASPlice:RNASPLICE:EDGER_DEU:SUBREAD_FLATTENGTF (GCF_000002985.6_WBcel235_genomic.gtf)` terminated with an error exit status (1)
```

Command executed:

```
flattenGTF -t exon -g gene_id -C -a GCF_000002985.6_WBcel235_genomic.gtf -o annotation.saf

cat <<-END_VERSIONS > versions.yml
"NFCORE_RNASPlice:RNASPLICE:EDGER_DEU:SUBREAD_FLATTENGTF":
    subread: $( echo $(flattenGTF -v 2>&1) | sed -e "s/flattenGTF v//g")
END_VERSIONS
```

Command exit status:

```
1
```

Command output:

```
(empty)
```

Command error:

```
Unable to find image 'quay.io/biocontainers/subread:2.0.1--hed695b0_0' locally
2.0.1--hed695b0_0: Pulling from biocontainers/subread
1dbcab28ce46: Already exists
cfb1ba34637d: Already exists
ace2d8a63dd5: Already exists
75c080ef15eb: Already exists
316957f8baaf: Already exists
dbd31e1d863d: Already exists
2f8531d5a6ec: Already exists
1dbcab28ce46: Already exists
2e178fd72baf: Already exists
2912e793bf3d: Pulling fs layer
2912e793bf3d: Download complete
2912e793bf3d: Pull complete
Digest: sha256:ccee1f6ebb924fd0b3b6db646a51dabc905697aa25be132386dd490c2318286c
Status: Downloaded newer image for quay.io/biocontainers/subread:2.0.1--hed695b0_0
```

```
Flattening GTF file: GCF_000002985.6_WBcel235_genomic.gtf
Output SAF file: annotation.saf
```

```
Looking for 'exon' features... (grouped by 'gene_id')
```

```
ERROR: failed to find the gene identifier attribute in the 9th column of the provided GTF file.
The specified gene identifier attribute is 'gene_id'.
```

```
An example of attributes included in your GTF annotation is 'transcript_id "id=CELE_T22C1.13"; gene_name "mir-8209";'.
The program has to terminate.
```

```
ERROR: Unable to open the GTF file.
```

```
Work dir:  
/home/iweber/work/b5/5edcf637a9a3d52ac7d2a792c1227
```

Tip: you can replicate the issue by changing to the process work dir and entering the command `bash .command.run`

-- Check `'.nextflow.log'` file for details

Fixing GTF file?

I also saw a message saying "unable to open the GTF file" above. To address this, I decided to copy the contents of my Genomes folder into my Documents folder so that Nextflow can run without any further issues of not being able to create symlinks or opening files.

I used my tried and tested friend, [FreeFileSync](#) (open source, with nice perks upon donation of any amount).

I tried running the pipeline with the GTF file instead of GFF

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \  
--max_cpus 6 \  
-w /home/iweber/Documents/Backup_shared_folder/Pipeline_test_C_elegans/ \  
--outdir /home/iweber/Documents/Backup_shared_folder/Pipeline_test_C_elegans/Nextflow_results_test_pipeline/ \  
--input /home/iweber/Documents/Backup_shared_folder/Pipeline_test_C_elegans/samplesheet_C_elegans.csv \  
--contrasts /home/iweber/Documents/Backup_shared_folder/Pipeline_test_C_elegans/contrastssheet_C_elegans.csv \  
--fasta /home/iweber/Documents/Backup_shared_folder/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.fna \  
--gtf /home/iweber/Documents/Backup_shared_folder/Genomes/genome_C_elegans/GCF_000002985.6_WBcel235_genomic.gtf \  
--save_trimmed true \  
--aligner star_salmon \  
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_C_elegans/STAR_index_C_elegans \  
--salmon_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_C_elegans/salmon_index \  
--save_unaligned true \  
--save_align_intermeds true \  
--rmats true \  
-profile docker
```

Annnnd it still didn't work. BUT. Knowing that the pipeline works in principle, I think I can now move on with my main mouse pipeline.

In case still needed:

So, how do I need to modify the GTF file to continue?

I found this [Biostars thread](#) that shares several possible solutions (also, the problem is apparently in featureCounts, not edgeR).

The first (and easiest) proposed solution is to remove any empty fields in the gene_id columns.

Running pipeline for actual experiment

Pigz compression of FastQ files

I used the script mentioned [**Gzipping FastQ files for pipeline with pigz**](#) with minor alterations for the new location:

/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/

```
gzip_several_fastqs.sh  
#!/bin/bash  
  
# Specify directory containing FASTQ files  
DIRECTORY="/mnt-win-uba-shared/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq"  
  
# Find all .fastq files in the specified directory and subdirectories. Write line by line, pipe each line with one file name  
# into while loop that compresses it.  
find "$DIRECTORY" -type f -name "*.fastq" | while read -r FILE; do  
    echo "Compressing $FILE"  
    gzip "$FILE"  
done
```

It completed successfully, and now I have .fastq.gz files instead of the FastQ ones. I also updated the sample sheet:

```
sample,fastq_1,fastq_2,strandedness,condition  
CONTROL_REP1,/home/iweber/Documents/Backup_shared_folder/Pre-  
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/SRR13761520_1.fastq.gz,/home/iweber/Documents/Backup_shared_f  
older/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/SRR13761520_1.fastq.gz,reverse,CONTROL
```

```

CONTROL_REPO2,/home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761521_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761521_2.fastq.gz,reverse,CONTROL
CONTROL_REPO3,/home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761522_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761522_2.fastq.gz,reverse,CONTROL
CONTROL_REPO4,/home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761523_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761523_2.fastq.gz,reverse,CONTROL
PREECLAMPSIA_REPO1,/home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761524_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761524_2.fastq.gz,reverse,PREECLAMPSIA
PREECLAMPSIA_REPO2,/home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761525_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761525_2.fastq.gz,reverse,PREECLAMPSIA
PREECLAMPSIA_REPO3,/home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761526_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761526_2.fastq.gz,reverse,PREECLAMPSIA
PREECLAMPSIA_REPO4,/home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761527_1.fastq.gz,/home/iweber/Documents/Backup_shared_f
older/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/SRR13761527_2.fastq.gz,reverse,PREECLAMPSIA

```

Unzipping .tgz Salmon index and BSOD

I realized that the Salmon index I had downloaded for the mouse was still stored as a .tgz archive, and I started unpacking it with the usual `tar -vxzf` command. I am not sure if it was connected, but, at some point in the process, my PC crashed with a BSOD. This time, space was not the issue - I still have 458 GB free on the partition I allotted to the virtual machine, and I can't believe the index is that large (compressed, it was around 12 GB, so I suspect it will be at most around 50 GB unzipped).

I made backups of everything:

1. restore points for ~~Windows OS partition, VM partition, and data drive~~
2. ~~File History~~
3. Windows recovery drive
4. Folder backups external HDD

I also changed the VM settings:

- Hardware -> Processors: for reasons I can't explain (probably transferred from VirtualBox), I had these options set to 6 processors and 1 core per processor. However, in reality, I have only 1 processor, the one from my laptop, with its 8 cores and 16 threads, so I set the options to "Number of processors: 1" and "Number of cores per processor: 6"
- Options -> Advanced -> activated "Log virtual machine progress periodically"
- Display -> Graphics memory: downsized max amount of guest memory that can be used for graphics memory from 8 GB to 4 GB
-

In general VM settings (Edit -> Preferences):

- Memory -> Additional memory: activated "fit all virtual machine memory into reserved host RAM"
- Memory -> Reserved memory: downsized from 114432 MB to 98000 MB (98 GB)

Things I DID NOT change yet, because they seem too risky to my beginner mind:

- there is a possibility to limit the number of disk input/output operations to prevent the VM from overwhelming the disk it operates on (which, in this case, is the same disk that my Windows host is running on - I really should get a second ultrafast 2+ TB SSD and move it to this one...). This would involve changing the virtual machine's .vmx file and adding a line to limit the operations (disk.maxIOPS = "500")

I tried unzipping the file again with the new settings: `tar -vxzf` and it worked! No more BSOD, and I now have a folder called "default" with all of the index components.

Pipeline parameters

Initial

I changed:

- `--max_memory 90GB` to make sure the pipeline leaves 5 GB for Ubuntu to run properly (apparently, [Linux needs far less RAM than Windows](#) - I know this being news to me probably makes the Linux pros smile and go "aww!" :))
- to cap Nextflow resource usage overall and be on the safe side, I followed the instructions from the [rnasplice parameters page](#) and added to my .bashrc file: `'NXF_OPTS=-Xms1g -Xmx4g'`

- `--save_reference true` saves anything the pipeline downloads by itself, e.g. STAR indices, so that they can be re-used in further runs

To be sure-sure I am working in the work folder I want to set:

```
cd /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/
```

Activate the Nextflow conda environment:

```
conda activate Nextflow
```

Actual pipeline parameters

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results \
--input /home/iweber/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/samplesheet_preeclampnia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclampnia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gtf
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gtf \
--clip_R1 10 \
--clip_R2 10 \
--stringency 5 \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--sjdb_overhang 149 \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/correct_Salmon_index_M_musculus_mm10/de
fault \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker
```

IF NEEDED LATER: `-resume [run_name]` to not repeat already finalized analysis steps! [Documentation for -resume](#) Also see [here](#): " Nextflow generates a unique task hash, e.g. 22/7548fa... for each task in a workflow. The hash takes into account the complete file path, the last modified timestamp, container ID, content of script directive among other factors. If any of these change, the task will be re-executed. Nextflow maintains a list of task hashes for caching and traceability purposes.". Here's the complete list of things that factor into resuming: [caching and resuming](#)

I immediately saw the warning:

```
WARN: The following invalid input values have been detected:
```

```
* --clip_R1: 10
* --clip_R2: 10
* --stringency: 5
* --sjdb_overhang: 149
```

So I suppose TrimGalore and STAR don't take the values for these parameters from the CLI - I'd probably need to modify the nextflow.config file directly to achieve this.

If still needed later

Easy fix for the clipping (I mis-wrote the Rs as capital R instead of minuscule):

```
--clip_r1 10
--clip_r2 10
```

And I got the first error:

```
ERROR ~ Error executing process > 'NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE
(GCF_000001635.27_GRCm39_genomic.gtf)'
```

Caused by:

```
Process `NFCORE_RNASPlice:RNASplice:TX2Gene_TxImport_star_Salmon:gffread_tx2Gene` (GCF_000001635.27_GRCm39_genomic.gtf)`  
terminated with an error exit status (1)
```

Command executed:

```
gffread --table transcript_id,gene_id,gene_name GCF_000001635.27_GRCm39_genomic.gtf | sort -u 1>  
GCF_000001635.27_GRCm39_genomic.tx2gene.tsv  
  
cat <<-END_VERSIONS > versions.yml  
"NFCore_RNASplice:RNASplice:TX2Gene_TxImport_star_Salmon:gffread_tx2Gene":  
    gffread: $(gffread --version 2>&1)  
END_VERSIONS
```

Command **exit** status:

```
1
```

Command output:

```
(empty)
```

Command error:

```
Error: no valid ID found for GFF record
```

This one's new and not encountered when trying to run the pipeline on the C. elegans data. This seems, once more, to be an issue due to differences between versions of the GTF format. As I also can get the gff file for the GRCm39 from [the NCBI FTP](#), I will attempt to run the pipeline using this one instead of the GTF file.

GFF instead of GTF

I used wget

```
https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF\_000001635.27\_GRCm39/GCF\_000001635.27\_GRCm39\_genomic.gff.gz to get the  
file, and `pigz -d GCF_000001635.27_GRCm39_genomic.gff.gz` to decompress it.
```

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \  
--max_cpus 6 \  
--max_memory 90.GB \  
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder \  
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-  
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results \  
--input /home/iweber/Documents/Backup_shared_folder/Pre-  
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/samplesheet_preeclampia.csv \  
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-  
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclampia.csv \  
--fasta  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \  
--gff  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \  
--clip_R1 10 \  
--clip_R2 10 \  
--stringency 5 \  
--save_trimmed true \  
--aligner star_salmon \  
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \  
--sjdb_overhang 149 \  
--salmon_index  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/correct_Salmon_index_M_musculus_mm10/de  
fault \  
--save_unaligned true \  
--save_align_intermeds true \  
--save_reference true \  
-profile docker
```

Started around 15:15.

Slow progress and crash/shutdown by Windows

Around 18:15, TrimGalore was about to end. I saw that I had around 300 GB still free on the VM partition, and started to worry that it might fill and crash the partition if it manages to get to the STAR and Salmon part of the pipeline. So I synced the folder to my backup folder in OneDrive (yes, not the best backup solution most likely, but it works for now, since it's publicly available data) and...deleted anything but the mouse genome and the pre-eclampsia experiment files the shared inter-OS folder *insert sweat smiley* This now leaves 343 GB free on the partition for the virtual machine, and I

dearly hope this will be sufficient to store all of the results from the analyses. I suddenly regret telling the pipeline to save alignment intermediate files such as SAM files...

Hmmm, at 19:24, the pipeline had still not finished the TrimGalore/FastQC part, and still saying: 42 processes waiting for input. However, looking in the results folder of trimgalore, I am seeing reads in fastq.gz format for all of the original fastq files, and the latest ones, judged by the "modified" time point, are from 18:21. It's been an hour since, and the pipeline does not seem to be progressing. Perhaps this is a good chance to stop it and re-start it, this time indicating to not save alignment intermediate files?

The results folder was, at 19:31, at 33,6 GB size. The "work" folder of Nextflow was 34 GB at 19:35.

This did not change around 9 PM, either (I checked over TeamViewer, Nextflow was showing the exact same thing as before, 42 processes later).

I let the pipeline work overnight, but, the next morning (15.8.), I realized that something had happened overnight and my PC had restarted itself. I don't know what crashed, but I can't even see the pipeline appear at all with `nextflow log`. In fact, I can't see any of the pipelines I ran yesterday, regardless of whether I call Nextflow from the conda environment I created specifically for it or from the base environment. The last pipelines that ran which I can see in the log are the C. elegans ones from 9.8.! BUT I can see them if I change into the work directory for the pre-eclampsia experiment xD (good to know!). My last pipeline was "awesome_jepsen" with the session ID d2143e7b-2e6a-41e4-850d-cb8af9596f82, which I can use to eventually resume the pipeline.

Figuring out cause of shutdown

However, I also wanted to see what happened last night and why my PC was restarted, especially whether it crashed or just tried to update. I went to the Windows Event Viewer and went through the messages from last night.

Looks like my Windows system has been very busy. I have some potentially relevant warnings and error messages since yesterday evening from before the pipeline froze:

- At 6 PM, TPM-WMI Event 1800 "A reboot is required before installing the update. Reason: 6". - At 1:43 AM, DistributedCOM, EventID 10010, "The server {9BA05972-F6A8-11CF-A442-00A0C90A8F39} did not register with DCOM within the required timeout." - At 1:44 AM, ServiceControlManager EventID 7043 "The Delivery Optimization service did not shut down properly after receiving a preshutdown control.",
- At 1:48 AM, TPM-WMI EventID 1796 "The Secure Boot update failed to update a Secure Boot variable with error Secure Boot is not enabled on this machine.. For more information, please see <https://go.microsoft.com/fwlink/?linkid=2169931> ,
- At 6 AM, EventID TPM-WMI EventID 1796 (as the one before). But I don't have anything with EventIDs 1074, 6008, or 41. The most I see is a 7034 from ServiceControlManager with the description "The System Update service terminated unexpectedly. It has done this 1 time(s)."
- However, lots of things seem to have happened around 1:44 AM. In sequence, the messages start with
- an error 7043 "The Delivery Optimization service did not shut down properly after receiving a preshutdown control.",
- 6006 "The Event log service was stopped.",
- 109 "The kernel power manager has initiated a shutdown transition. Shutdown Reason: Kernel API",
- 13 "The operating system is shutting down at system time 2024-08-14T23:44:19.417683100Z.",
- 12 "The operating system started at system time 2024-08-14T23:44:42.500000000Z.",
- 20 "The last shutdown's success status was true. The last boot's success status was true.".

What happened here? It looks like my Windows system tried to install some update related to TPM-WMI and restarted my system around 1:43 AM.

About the messages in the event log, ChatGPT said:

Chatgpt

The sequence of events you've provided indicates that your Windows system went through an unexpected or forced shutdown and reboot around 1:44 AM. Let's break down what likely happened based on the event log messages:

1. TPM-WMI Events and Reboot Requirement (6 PM)

- **Event 1800:** "A reboot is required before installing the update. Reason: 6."
- This indicates that earlier in the evening, Windows identified that a reboot was necessary to complete some updates, likely related to TPM (Trusted Platform Module) or Secure Boot.

3. Service Control Manager Errors (1:44 AM)

- **EventID 7043:** "The Delivery Optimization service did not shut down properly after receiving a preshutdown control."
- **EventID 7031:** "The Autodesk Desktop Licensing Service service terminated unexpectedly."
- **EventID 7034:** "The System Update service terminated unexpectedly."
- These events suggest that Windows initiated a shutdown process at 1:44 AM, but certain services did not shut down properly or crashed. The "preshutdown control" refers to the command sent to services to gracefully stop before the system shuts down.

5. Kernel Power Shutdown Initiated (1:44 AM)

- **EventID 109:** "The kernel power manager has initiated a shutdown transition. Shutdown Reason: Kernel API."
- **EventID 13:** "The operating system is shutting down at system time 2024-08-14T23:44:19.417683100Z."

- **EventID 12:** "The operating system started at system time 2024-08-14T23:44:42.500000000Z."
- These logs indicate that the kernel (core of the operating system) initiated a shutdown. The "Kernel API" reason suggests that this was not a user-initiated shutdown but likely caused by an internal system command or process (possibly related to the TPM/Secure Boot update).

6. System Reboot (1:44 AM)

- The system logs show that the system successfully restarted shortly after shutting down at 1:44 AM.
- **EventID 20:** "The last shutdown's success status was true. The last boot's success status was true."
- This suggests that the shutdown and subsequent reboot were completed without errors, but the timing and the unexpected nature of the shutdown indicate it was likely triggered by an automated process, potentially related to updates.

What Likely Happened

- Your system was likely processing a critical update related to Secure Boot or TPM, as indicated by the TPM-WMI events. Windows required a reboot to complete these updates, and it seems that the system initiated this reboot automatically in the early hours of the morning.
- The DistributedCOM error and service shutdown errors suggest that not all processes could cleanly stop before the shutdown, which is typical during an abrupt system shutdown.
- The "Kernel API" shutdown reason indicates that this was likely an automated process rather than something you initiated manually.

To prevent any potential automated restarts by Windows, I changed my active hours to between 3 PM and 9 AM. I also paused updates until September 19th - hope this isn't a terrible idea.

Reactivate 3D acceleration (GPU usage) by the VM

I hoped that changing this will make the pipeline running a lot faster, and hopefully prevent any further issues. However, I saw that the "accelerate 3D graphics" options

I also changed the VM settings again:

- Hardware -> Memory -> Memory for this virtual machine: lowered to 92500 MB (90.5 GB) to avoid memory swapping
- Hardware -> Processors: left as "Number of processors: 1" and "Number of cores per processor: 6"
- Hardware -> Display -> Graphics memory: increased max amount of guest memory back to 8 GB
- Options -> Advanced -> activated "Log virtual machine progress periodically"

In general VM settings (Edit -> Preferences):

- Memory -> Additional memory: left "fit all virtual machine memory into reserved host RAM" activated
- Memory -> Reserved memory: left at 98000 MB (90.5 GB)

Things I still DID NOT change, because they seem too risky to my beginner mind:

- still did not limit the number of disk input/output operations (disk.maxIOPS)

Pipeline re-run with GPU usage activated

To be sure-sure I am working in the work folder I want to set:

```
cd /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/
```

Activate the Nextflow conda environment:

```
conda activate Nextflow
```

I tried re-running the same pipeline as yesterday with `sudo nextflow run nf-core/rnasplice -r 1.0.4 --resume d2143e7b-2e6a-41e4-850d-cb8af9596f82`, but that did not actually restart the pipeline (it called it a different name, "magical_thompson", and gave an error about missing inputs, outdir, contrasts).

I tried to re-run the exact same command as yesterday, including the not-yet corrected parameters for clipping to be sure Nextflow correctly identifies it as the same pipeline, and appended `--resume d2143e7b-2e6a-41e4-850d-cb8af9596f82`.

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplice_results \
--input /home/iweber/Documents/Backup_shared_folder/Pre-
```

```

eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/samplesheet_preeclamp sia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--clip_R1 10 \
--clip_R2 10 \
--stringency 5 \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--sjdb_overhang 149 \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/correct_Salmon_index_M_musculus_mm10/de
fault \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-resume d2143e7b-2e6a-41e4-850d-cb8af9596f82

```

I am not sure if this resumed the exact same run. I now have a run called "intergalactic_woese". However, when looking at what Nextflow currently displays, I'm seeing what I saw last night as well:

Activities Terminal Do Aug 15 12:14

iweber@iweber-VirtualBox: ~/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_wo...

```

WARN: Both --aligner=star salmon and --pseudo aligner=salmon specified. Downstream analyses will be performed on both salmon output files.
[4f/788081] NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1, cached: 1 ✓
[85/e3b1ab] NFC..ASPLICE:RNASplice:PREPARE_GENOME:GTf_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[ ] NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA
[4f/788081] NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1, cached: 1 ✓
[85/e3b1ab] NFC..ASPLICE:RNASplice:PREPARE_GENOME:GTf_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[ ] NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA
[ ] [ 0%] 0 of 1
[c4/ed4701] NFC..CE:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[6a/018809] NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclamp sia.csv) [100%] 1 of 1, cached: 1 ✓
[93/953d3c] NFC..SPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastssheet_preeclamp sia.csv) [100%] 1 of 1, cached: 1 ✓
executor > local (1)
[4f/788081] NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1, cached: 1 ✓
[85/e3b1ab] NFC..ASPLICE:RNASplice:PREPARE_GENOME:GTf_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[15/9138d2] NFC..PLICE:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna) [ 0%] 0 of 1
[c4/ed4701] NFC..CE:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[6a/018809] NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclamp sia.csv) [100%] 1 of 1, cached: 1 ✓
[93/953d3c] NFC..SPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastssheet_preeclamp sia.csv) [100%] 1 of 1, cached: 1 ✓
[ ] NFCORE_RNASPlice:RNASplice:CAT_FASTQ
[ec/63d490] NFCORE_RNASPlice:RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:FASTQC (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
[ee/7705ef] NFCORE_RNASPlice:RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:TRIMGALORE (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
[ ] NFCORE_RNASPlice:RNASplice:ALIGN_STAR:STAR_ALIGN [ 0%] 0 of 8
[ ] NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_SORT
[ ] NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_SAMTOOLS_INDEX
[ ] NFC..ASPLICE:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_STATS
[ ] NFC..LICE:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT
[ ] NFC..LICE:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_IDXSTATS
[a7/435c5b] NFC..RNASplice:RNASplice:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
executor > local (2)
[4f/788081] NFC..E:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1, cached: 1 ✓
[85/e3b1ab] NFC..ICE:PREPARE_GENOME:GTf_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[15/9138d2] NFC..GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1 ✓
[c4/ed4701] NFC..REPAIR_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[6a/018809] NFC..E:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclamp sia.csv) [100%] 1 of 1, cached: 1 ✓
[93/953d3c] NFC..CE:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastssheet_preeclamp sia.csv) [100%] 1 of 1, cached: 1 ✓
[ ] NFCORE_RNASPlice:RNASplice:CAT_FASTQ
[ec/63d490] NFC..CE:RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:FASTQC (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
[ee/7705ef] NFC..RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:TRIMGALORE (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
[ ] NFCORE_RNASPlice:RNASplice:ALIGN_STAR:STAR_ALIGN [ 0%] 0 of 8
[ ] NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_SORT
[ ] NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_SAMTOOLS_INDEX
[ ] NFC..ICE:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS_SAMTOOLS_STATS
[ ] NFC..:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT
[ ] NFC..:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_IDXSTATS
[a7/435c5b] NFC..PLICE:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[ ] NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_COUNT
[ ] NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_EXON
[b2/338963] NFC..PLICE:EDGER_DEU:SUBREAD_FLATTENGTf (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[ ] NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FEATURECOUNTS
[ ] NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON
[ ] NFCORE_RNASPlice:RNASplice:RMATS:CREATE_BAMLIST
[ ] NFCORE_RNASPlice:RNASplice:RMATS:RMATS_PREP
[ ] NFCORE_RNASPlice:RNASplice:RMATS:RMATS_POST

```

This "Plus 33 more processes waiting for tasks" actually changed, but not for better: I could swear it was initially at "32 more", not 33!

But, while making the screenshot, I heard the ventilators ventilate hard, and, suddenly, the picture changed:

Activities Terminal

Do Aug 15 12:18

```
iweber@iweber-VirtualBox: ~/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_wo...  
[c4/ed4701] NFC_CE:RNASPLICER:PREPARE_GENOME:CUSTOM_GETCHROMSIZES [GCF_000001635.27_GRCh39_genomic.fna] [100%] 1 of 1, cached: 1 ✓  
[6a/018809] NFCORE_RNASPLICER:RNASPLICER:INPUT_CHECK:SAMPLESHEET_CHECK [samplesheet_preeclampsia.csv] [100%] 1 of 1, cached: 1 ✓  
[93/953d0c] NFC_SPlice:RNASPLICER:CONTRASTS_CHECK:CONTRASTSHEET_CHECK [contrastsheet_preeclampsia.csv] [100%] 1 of 1, cached: 1 ✓  
[-]  
[ec/63d490] NFCORE_RNASPLICER:RNASPLICER:FASTQ_FASTQC:UNITTOOLS_TRIMGALORE_FASTQC [PREECLAMPSIA REP4] [100%] 8 of 8, cached: 8 ✓  
[ee/7705ef] NFCORE_RNASPLICER:RNASPLICER:FASTQ_FASTQC:UNITTOOLS_TRIMGALORE [PREECLAMPSIA REP4] [100%] 8 of 8, cached: 8 ✓  
[-]  
[a7/435c5b] NFCORE_RNASPLICER:RNASPLICER:CAT_FASTQ [NFCORE_RNASPLICER:RNASPLICER:ALIGN_STAR:STAR_ALIGN] [0%] 0 of 8  
[executer local (3)]  
[47/788081] process > NFCORE_RNASPLICER:RNASPLICER:PREPARE_GENOME:GFFREAD [GCF_000001635.27_GRCh39_genomic.gff] [100%] 1 of 1, cached: 1 ✓  
[85/e318ab] process > NFCORE_RNASPLICER:RNASPLICER:PREPARE_GENOME:GTF_GENE_FILTER [GCF_000001635.27_GRCh39_genomic.fna] [100%] 1 of 1, cached: 1 ✓  
[15/9138d2] process > NFCORE_RNASPLICER:RNASPLICER:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA [rsem/GCF_000001635.27_GRCh39_genomic.fna] [100%] 1 of 1 ✓  
[c4/ed4701] process > NFCORE_RNASPLICER:RNASPLICER:PREPARE_GENOME:CUSTOM_GETCHROMSIZES [GCF_000001635.27_GRCh39_genomic.fna] [100%] 1 of 1, cached: 1 ✓  
[6a/018809] process > NFCORE_RNASPLICER:RNASPLICER:INPUT_CHECK:SAMPLESHEET_CHECK [samplesheet_preeclampsia.csv] [100%] 1 of 1, cached: 1 ✓  
[93/953d0c] process > NFCORE_RNASPLICER:RNASPLICER:CONTROTS_CHECK:CONTRASTSHEET_CHECK [contrastsheet_preeclampsia.csv] [100%] 1 of 1, cached: 1 ✓  
[-]  
[ec/63d490] process > NFCORE_RNASPLICER:RNASPLICER:FASTQ_FASTQC:UNITTOOLS_TRIMGALORE_FASTQC [PREECLAMPSIA REP4] [100%] 8 of 8, cached: 8 ✓  
[ee/7705ef] process > NFCORE_RNASPLICER:RNASPLICER:FASTQ_FASTQC:UNITTOOLS_TRIMGALORE [PREECLAMPSIA REP4] [100%] 8 of 8, cached: 8 ✓  
[09/389016] process > NFCORE_RNASPLICER:RNASPLICER:CAT_FASTQ [NFCORE_RNASPLICER:RNASPLICER:ALIGN_STAR:STAR_ALIGN_PREECLAMPSIA REP2] [0%] 0 of 8  
[process > NFCORE_RNASPLICER:RNASPLICER:ALIGN_STAR:STAR_ALIGN_STATS_SAMTOOLS_SAMTOOLS_SORT]  
[process > NFCORE_RNASPLICER:RNASPLICER:ALIGN_STAR:STAR_ALIGN_STATS_SAMTOOLS_SAMTOOLS_INDEX]  
[process > NFCORE_RNASPLICER:RNASPLICER:ALIGN_STAR:STAR_ALIGN_STATS_SAMTOOLS_BAM_STATS_SAMTOOLS_SAMTOOLS_STATS]  
[process > NFCORE_RNASPLICER:RNASPLICER:ALIGN_STAR:STAR_ALIGN_STATS_SAMTOOLS_BAM_STATS_SAMTOOLS_SAMTOOLS_FLAGSTAT]  
[process > NFCORE_RNASPLICER:RNASPLICER:ALIGN_STAR:STAR_ALIGN_STATS_SAMTOOLS_BAM_STATS_SAMTOOLS_SAMTOOLS_IDXSTATS]  
[a7/435c5b] process > NFCORE_RNASPLICER:RNASPLICER:DEU_DEXSEQ_ANNOTATION [GCF_000001635.27_GRCh39_genomic.gtf] [100%] 1 of 1, cached: 1 ✓  
[process > NFCORE_RNASPLICER:RNASPLICER:DEU_DEXSEQ_DEU_DEXSEQ_COUNT]  
[process > NFCORE_RNASPLICER:RNASPLICER:DEU_DEXSEQ_EXON]  
[b2/338963] process > NFCORE_RNASPLICER:RNASPLICER:EDGER_DEU:SUPERREAD_FLATTENTGT [GCF_000001635.27_GRCh39_genomic.gtf] [100%] 1 of 1, cached: 1 ✓  
[process > NFCORE_RNASPLICER:RNASPLICER:EDGER_DEU:SUPERREAD_FEATURECOUNTS]  
[process > NFCORE_RNASPLICER:RNASPLICER:EDGER_DEU:EDGER_EXON]  
[process > NFCORE_RNASPLICER:RNASPLICER:HM3S_CREATE_BAMLIST]  
[process > NFCORE_RNASPLICER:RNASPLICER:HM3S_RMATS_PREP]  
[process > NFCORE_RNASPLICER:RNASPLICER:HM3S_RMATS_POST]  
[87/9c14e3] process > NFCORE_RNASPLICER:RNASPLICER:VISUALISE_MISO_GTF_2_GFF3 [100%] 1 of 1, cached: 1 ✓  
[5e/89ed70] process > NFCORE_RNASPLICER:RNASPLICER:VISUALISE_MISO_MISO_INDEX [100%] 1 of 1 ✓  
[process > NFCORE_RNASPLICER:RNASPLICER:VISUALISE_MISO_MISO_RUN]  
[process > NFCORE_RNASPLICER:RNASPLICER:VISUALISE_MISO_MISO_SETTINGS]  
[process > NFCORE_RNASPLICER:RNASPLICER:VISUALISE_MISO_MISO_SASHIMI]  
[process > NFCORE_RNASPLICER:RNASPLICER:VISUALISE_MISO_SALMON_QUANT_STAR]  
[a7/b0f3a3] process > NFCORE_RNASPLICER:RNASPLICER:TAXGENE_TXIMPORT_STAR_SALMON_UNTAR [100%] 1 of 1, cached: 1 ✓  
[process > NFCORE_RNASPLICER:RNASPLICER:TAXGENE_TXIMPORT_STAR_SALMON_GFFREAD_TX2GENE [GCF_000001635.27_GRCh39_genomic.gtf]] [100%] 1 of 1, cached: 1 ✓  
[process > NFCORE_RNASPLICER:RNASPLICER:TAXGENE_TXIMPORT_STAR_SALMON_DRIMSEQ_FILTER]  
[process > NFCORE_RNASPLICER:RNASPLICER:DRIMSEQ_DESER_DTM_STAR_SALMON_DRIMSEQ_DTM]  
[process > NFCORE_RNASPLICER:RNASPLICER:DRIMSEQ_DESER_DTM_STAR_SALMON_DEXSEQ_DTM]  
[process > NFCORE_RNASPLICER:RNASPLICER:DRIMSEQ_DESER_DTM_STAR_SALMON_STAGER]  
[60/92c17f] process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_SPLIT_FILES_I0E [GCF_000001635.27_GRCh39_genomic.gtf] [100%] 1 of 1, cached: 1 ✓  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_GENERATE_EVENTS_I0E]  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_PS1PERVENT]  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_SPLIT_FILES_I0E]  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_DIFFSPICE_I0E]  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_CLUSTERGROUPS_I0E]  
[fc/ffa5ce] process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_GENERATE_EVENTS_I0I [GCF_000001635.27_GRCh39_genomic.gtf] [100%] 1 of 1, cached: 1 ✓  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_PS1PERISIFORM]  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_SPLIT_FILES_I0I]  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_DIFFSPICE_I0I]  
[process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_STAR_SALMON_CLUSTERGROUPS_I0I]  
[process > NFCORE_RNASPLICER:RNASPLICER:SALMON_QUANT_SALMON]  
[fe/354fd4] process > NFCORE_RNASPLICER:RNASPLICER:TAXGENE_TXIMPORT_SALMON_GFFREAD_TX2GENE [GCF_000001635.27_GRCh39_genomic.gtf] [0%] 0 of 8  
[8f/b25573] process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_SALMON_GENERATE_EVENTS_I0E [GCF_000001635.27_GRCh39_genomic.gtf] [100%] 1 of 1, cached: 1 ✓  
[df/8384fe] process > NFCORE_RNASPLICER:RNASPLICER:SUPPA_SALMON_GENERATE_EVENTS_I0I [GCF_000001635.27_GRCh39_genomic.gtf] [100%] 1 of 1, cached: 1 ✓  
Plus 19 more processes waiting for tasks...
```

One of the "STAR" processes (09/389016) now has files associated with it, PREECLAMPSIA _REP2. The size of my "results" folder has now also increased to 39,8 GB!

At 12:38, the STAR process for the first file reached 12% (20 min) -> if the speed is constant, I'd need around 2.8 h per sample for it to complete. For 7 more samples, that's 20 hours more -> it'll likely need until tomorrow to finish. My best bet is to probably stop anything else that's particularly resource-intensive on both the host and the guest OS and let it do its job.

At 1:05, we're at 25% of a second sample, CONTROL _REP2! So, in around 50 min, we processed 125% of a file (yes, silly calculation) -> it took around 40 min to complete one sample. The size of the results folder jumped to 66 GB -> looking at around 20 GB per sample -> will need around 160 GB total solely for the SAM files resulting from STAR. It's good that my inter-OS shared folder is now entirely saved in OneDrive and on the other hard drive, because I could delete all of its non-git-related contents, so now I have 480 GB empty disk space for the VM to go nuts with.

What's also great, as for most Nextflow pipelines: it looks like some of the samples are also undergoing the quantification with Salmon now! (4th to last row)

```
iweber@iweber-VirtualBox: ~/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nex
15/9138d2] process > NFCORE RNASPLICE:RNASPLICE:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1 ✓
c4/ed4761] process > NFCORE RNASPLICE:RNASPLICE:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
6a/018809] process > NFCORE RNASPLICE:RNASPLICE:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampsia.csv) [100%] 1 of 1, cached: 1 ✓
93/953d3c] process > NFCORE RNASPLICE:RNASPLICE:CONTRASTS_CHECK:CONTRASTSHEET_CHECK (contrastssheet_preeclampsia.csv) [100%] 1 of 1, cached: 1 ✓
[...] process > NFCORE RNASPLICE:RNASPLICE:CAT_FASTQ -
ec/63d490] process > NFCORE RNASPLICE:RNASPLICE:FASTQC UMITOOLS TRIMGALORE:FASTQC (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
ee/7705ef] process > NFCORE RNASPLICE:RNASPLICE:FASTQC UMITOOLS TRIMGALORE:TRIMGALORE (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
37/e013aa] process > NFCORE RNASPLICE:RNASPLICE:ALIGN_STAR:STAR_ALIGN (CONTROL REP2) [25%] 2 of 8
[...] process > NFCORE RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:SAMTOOLS_SORT [0%] 0 of 2
[...] process > NFCORE RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:SAMTOOLS_INDEX -
[...] process > NFCORE RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:BAM_STATS SAMTOOLS:SAMTOOLS_STATS -
[...] process > NFCORE RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:BAM_STATS SAMTOOLS:SAMTOOLS_FLAGSTAT -
[...] process > NFCORE RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:BAM_STATS SAMTOOLS:SAMTOOLS_IDXSTATS -
a7/435c5b] process > NFCORE RNASPLICE:RNASPLICE:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[...] process > NFCORE RNASPLICE:RNASPLICE:DEXSEQ_DEU:DEXSEQ_COUNT -
[...] process > NFCORE RNASPLICE:RNASPLICE:DEXSEQ_DEU:DEXSEQ_EXON -
b2/338963] process > NFCORE RNASPLICE:RNASPLICE:EDGER_DEU:SUBREAD_FLATTENGT (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[...] process > NFCORE RNASPLICE:RNASPLICE:EDGER_DEU:SUBREAD_FEATURECOUNTS -
[...] process > NFCORE RNASPLICE:RNASPLICE:EDGER_DEU:EDGER_EXON -
[...] process > NFCORE RNASPLICE:RNASPLICE:RMATS:CREATE_BAMLIST -
[...] process > NFCORE RNASPLICE:RNASPLICE:RMATS:RMATS_PREP -
[...] process > NFCORE RNASPLICE:RNASPLICE:RMATS:RMATS_POST -
87/9c14e3] process > NFCORE RNASPLICE:RNASPLICE:VISUALISE_MISO:GTF_2_GFF3 [100%] 1 of 1, cached: 1 ✓
5e/89ed70] process > NFCORE RNASPLICE:RNASPLICE:VISUALISE_MISO:MISO_INDEX [100%] 1 of 1 ✓
[...] process > NFCORE RNASPLICE:RNASPLICE:VISUALISE_MISO:MISO_RUN -
[...] process > NFCORE RNASPLICE:RNASPLICE:VISUALISE_MISO:MISO_SETTINGS -
[...] process > NFCORE RNASPLICE:RNASPLICE:VISUALISE_MISO:MISO_SASHIMI -
[...] process > NFCORE RNASPLICE:RNASPLICE:SALMON_QUANT_STAR [0%] 0 of 2
[...] process > NFCORE RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:UNTAR -
a7/b0f3a3] process > NFCORE RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[...] process > NFCORE RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON:DRIMSEQ_FILTER -
[...] process > NFCORE RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON:DEXSEQ_DTU -
[...] process > NFCORE RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON:STAGER -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:SPLIT_FILES_TPM -
69/92c17f] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:PSIPEREVENT -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:SPLIT_FILES_IOE -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:DIFFSPLICE_IOE -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:CLUSTERGROUPS_IOE -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:CLUSTEREVENTS_IOE -
fc/ffa5ce] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:PSIPERISOFORM -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:SPLIT_FILES_IOI -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:DIFFSPLICE_IOI -
[...] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON:CLUSTERGROUPS_IOI -
9c/00b355] process > NFCORE RNASPLICE:RNASPLICE:SALMON_QUANT_SALMON (CONTROL REP3) [37%] 3 of 8
fe/3541da] process > NFCORE RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
8f/025573] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
df/8384fe] process > NFCORE RNASPLICE:RNASPLICE:SUPPA_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
Plus 19 more processes waiting for tasks...
```

The pipeline ran for some 3h30, and finished running the alignment with STAR (STAR_ALIGN), the sorting of the SAM files with Samtools (SAMTOOLS_SORT), the quantification of the BAM files produced by STAR using Salmon (SALMON_QUANT_STAR), and the [pseudomapping] is **this correct?** and quantification with Salmon alone (SALMON_QUANT_SALMON). Many other smaller steps completed too, which, I'd wager, are

```

genome-specific, since they seem to consist of only 1 step and not 8 (number of samples).
[4f4e1070] Waiting
executor > local (38)
|4f/788081] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1, cached: 1 ✓
|85/e30lab] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GTf_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
|15/9138d2] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
|c4/ed4701] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
|6a/018809] process > NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampsia.csv) [100%] 1 of 1, cached: 1 ✓
|93/953d3c] process > NFCORE_RNASPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSHEET_CHECK (contrastssheet_preeclampsia.csv) [100%] 1 of 1, cached: 1 ✓
| ec/63d490] process > NFCORE_RNASPlice:RNASplice:FASTQC_UMITOOLS_TRIMGALORE:FASTQC (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
| ee/7705ef] process > NFCORE_RNASPlice:RNASplice:FASTQC_UMITOOLS_TRIMGALORE:(PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
| e9/c0e68c] process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:STAR_ALIGN (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
| 85/78622c] process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
| 7f/ba18a] process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_INDEX (CONTROL REP2) [100%] 3 of 8
| | process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_STATS
| | process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_STATS
| | process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT
| | process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_IDXSTATS
| a7/435c5b] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| | process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_COUNT [0%] 0 of 8
| | process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_EXON
| b2/338963] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FLATTENGT (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| | process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FEATURECOUNTS [0%] 0 of 8
| | process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON
| | process > NFCORE_RNASPlice:RNASplice:RMATS:CREATE_BAMLIST [0%] 0 of 1
| | process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_PREP
| | process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_POST
| 87/9c14e3] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:GTF_2_GFF3 [100%] 1 of 1, cached: 1 ✓
| 5e/89ed70] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_INDEX [100%] 1 of 1 ✓
| | process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_RUN
| | process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SETTINGS
| | process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI
| 2a/1984a7] process > NFCORE_RNASPlice:RNASplice:SALMON_QUANT_STAR (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
| | process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_STAR_SALMON:UNTAR
| a7/b0f3a3] process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| | process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_STAR_SALMON:TXIMPORT [0%] 0 of 1
| | process > NFCORE_RNASPlice:RNASplice:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON:DRIMSEQ_FILTER
| | process > NFCORE_RNASPlice:RNASplice:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON:DEXSEQ_DTU
| | process > NFCORE_RNASPlice:RNASplice:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON:STAGER
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:SPLIT_FILES_TPM
| 69/92c17f] process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:PSIPEREVENT
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:SPLIT_FILES_IOE
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:DIFFSPICE_IOE
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:CLUSTERGROUPS_IOE
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:CLUSTEREVENTS_IOE
| fc/ffa5ce] process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:PSIPERISOFORM
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:SPLIT_FILES_IOI
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:DIFFSPICE_IOI
| | process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:CLUSTERGROUPS_IOI
| e8/8ad965] process > NFCORE_RNASPlice:RNASplice:SALMON_QUANT_SALMON (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
| fe/354fd4] process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| f9/fa69fe] process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON:TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv) [100%] 1 of 1, failed: 1 ✗
| 8f/025573] process > NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| df/8384fe] process > NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
| | process > NFCORE_RNASPlice:RNASplice:MULTIQC [0%] 0 of 1
Plus 17 more processes waiting for tasks...

```

Sadly, it failed due to an error stemming from `tximport`.

tximport error with identifiers

The tximport error read:

```
Error in .local(object, ...):
  None of the transcripts in the quantification files are present
  in the first column of tx2gene. Check to see that you are using
  the same annotation for both.
```

Example IDs (file): [ENSMUST00000196221.1, ENSMUST00000179664.1, ENSMUST00000177564.1, ...]

Example IDs (tx2gene): [gene-Bod1-ps, gene-Btf3-ps10, gene-Btf3-ps11, ...]

This can sometimes (not always) be fixed using '`ignoreTxVersion`' or '`ignoreAfterBar`'.

```
Calls: <Anonymous> -> <Anonymous> -> .local
Execution halted
```

Work dir:

```
/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/f9/fa69fe2c73e138ff019c77f1df833e
```

Tip: when you have fixed the problem you can `continue` the execution adding the option '`-resume`' to the run command line

-- Check '`.nextflow.log`' file for details

There's a thread relating to txmeta throwing a similar error: <https://github.com/thelovelab/txmeta/issues/63> Something may be amiss with the metadata from NCBI that messes up how Salmon works with the annotations overall.

There is something potentially awry with the GFF file that makes these odd gene-Bod1-ps show up. I didn't want to tamper with the file I used for the pipeline in order to be able to resume the pipeline later, so I made a copy. I reasoned that, if I have these odd identifiers, they should sort before the usual NCBI transcript identifiers (NM, NR,...). So I sorted the file alphabetically, and displayed the first 100 rows:

```
sort GCF_000001635.27_GRCm39_genomic.gff | head -n 100
```

What I got was, to my understanding, a regular GFF format from RefSeq, and no sign of the odd identifiers:

```
#!/annotation-date 02/01/2024
#!/annotation-source NCBI_RefSeq GCF_000001635.27-RS_2024_02
#!/genome-build-accession NCBI_Assembly:GCF_000001635.27
#!/genome-build GRCm39
#!/gff-spec-version 1.21
#!/gff-version 3
NC_000067.7 BestRefSeq2CGnomon gene 10030388 10038207 . . . ID=gene-Tcf24;Dbxref=GeneID:100039596,MGI:MGI:3780500;Name=Tcf24;description=transcription factor 24;gbkey=Gene;gen
e=Tcf24;gene_biotype=protein_coding;gene_synonym=100039596,Gm10567,Gm2330
NC_000067.7 BestRefSeq2CGnomon gene 10038847 10083766 . . . ID=gene-Ppp1r42;Dbxref=GeneID:69312,MGI:MGI:1921138;Name=Ppp1r42;description=protein phosphatase 1%2C regulatory su
bunit 42;gbkey=Gene;gene=Ppp1r42;gene_biotype=protein_coding;gene_synonym=170001118Rik,4930418G15Rik,Lrrc67,TLR
NC_000067.7 BestRefSeq2CGnomon gene 10108194 10206996 . . . ID=gene-Csppl;Dbxref=GeneID:211660,MGI:MGI:2681832;Name=Csppl;description=centrosome and spindle pole associated pr
otein 1;gbkey=Gene;gene=Csppl;gene_biotype=protein_coding;gene_synonym=2310020J12Rik,4930413022Rik,Csppl,Csp3
NC_000067.7 BestRefSeq2CGnomon gene 10207732 10303306 . . . ID=gene-Arfgef1;Dbxref=GeneID:211673,MGI:MGI:2442988;Name=Arfgef1;description=ADP ribosylation factor guanine nucle
otide exchange factor 1;gbkey=Gene;gene=Arfgef1;gene_biotype=protein_coding;gene_synonym=ARFGEF1,BIG1,D130059805Rik,D730028018Rik,P200
NC_000067.7 BestRefSeq2CGnomon gene 10366016 10790277 . . . ID=gene-Cpa6;Dbxref=GeneID:320993,MGI:MGI:3045348;Name=Cpa6;description=carboxypeptidase A6;gbkey=Gene;gene=Cpa6;ge
ne_biotype=protein_coding;gene_synonym=9030616013Rik
NC_000067.7 BestRefSeq2CGnomon gene 105204642 105284665 . . . ID=gene-Rnf152;Dbxref=GeneID:320311,MGI:MGI:2443787;Name=Rnf152;description=ring finger protein 152;gbkey=Gene;gene
=Rnf152;gene_biotype=protein_coding;gene_synonym=A930029802Rik
NC_000067.7 BestRefSeq2CGnomon gene 105446147 105591466 . . . ID=gene-Pign;Dbxref=GeneID:27392,MGI:MGI:1351629;Name=Pign;description=phosphatidylinositol glycan anchor biosynthe
sis%2C class N;gbkey=Gene;gene=Pign;gene_biotype=protein_coding;gene_synonym=Gm20308,PIG-N
NC_000067.7 BestRefSeq2CGnomon gene 105682856 . . . ID=gene-Relch;Dbxref=GeneID:227446,MGI:MGI:1922832;Name=Relch;description=RAB11 binding and LisH domain%2C coiled-c
oil and HEAT repeat containing;gbkey=Gene;gene=Relch;gene_biotype=protein_coding;gene_synonym=2310035C2Rik,643040IN10,Kiaa1468,mIAA1468
NC_000067.7 BestRefSeq2CGnomon gene 105708443 105777172 . . . ID=gene-Tnfrsf1la;Dbxref=GeneID:21934,MGI:MGI:1314891;Name=Tnfrsf1la;description=tumor necrosis factor receptor sup
erfamily%2C member 11a%2C_NFKB activator;gbkey=Gene;gene=Tnfrsf1la;gene_biotype=protein_coding;gene_synonym=Ly109,mRANK,ODFR,OFE,Rank,TRANCE-R
NC_000067.7 BestRefSeq2CGnomon gene 105916929 105961807 . . . ID=gene-Zcchc2;Dbxref=GeneID:227449,MGI:MGI:2444114;Name=Zcchc2;description=zinc finger%2C CCHC domain containing 2
;gbkey=Gene;gene=Zcchc2;gene_biotype=protein_coding;gene_synonym=9930114B20Rik
NC_000067.7 BestRefSeq2CGnomon gene 106648140 106689544 . . . ID=gene-Kdsr;Dbxref=GeneID:70750,MGI:MGI:1918000;Name=Kdsr;description=3-ketodihydrophosphingosine reductase;gbkey=Gen
e;gene=Kdsr;gene_biotype=protein_coding;gene_synonym=6330410P18Rik,9430079808Rik,Fvt1
NC_000067.7 BestRefSeq2CGnomon gene 107005893 107011218 . . . ID=gene-Serpinb3d;Dbxref=GeneID:394252,MGI:MGI:2683295;Name=Serpinb3d;description=serine (or cysteine) peptidase in
hibitor%2C clade B (ovalbumin)%2C member 3D;gbkey=Gene;gene=Serpinb3d;gene_biotype=protein_coding;gene_synonym=9330156F07Rik
NC_000067.7 BestRefSeq2CGnomon gene 107456724 107488469 . . . ID=gene-Serpinb10;Dbxref=GeneID:241197,MGI:MGI:2138648;Name=Serpinb10;description=serine (or cysteine) peptidase in
hibitor%2C clade B (ovalbumin)%2C member 10;gbkey=Gene;gene=Serpinb10;gene_biotype=protein_coding;gene_synonym=9830131G07,Serpinb10-ps
NC_000067.7 BestRefSeq2CGnomon gene 107517668 107536708 . . . ID=gene-Serpinb8;Dbxref=GeneID:20725,MGI:MGI:894657;Name=Serpinb8;description=serine (or cysteine) peptidase inhibi
tor%2C clade B (ovalbumin)%2C member 8;gbkey=Gene;gene=Serpinb8;gene_biotype=protein_coding;gene_synonym=CAP-2,CAP-2,NK10,ovalbumin,Sp18
NC_000067.7 BestRefSeq2CGnomon gene 109909967 110073002 . . . ID=gene-Cdh7;Dbxref=GeneID:241201,MGI:MGI:2442792;Name=Cdh7;description=cadherin 7%2C type 2;gbkey=Gene;gene=Cdh7;
gene_biotype=protein_coding;gene_synonym=9330156F07Rik,CDH7L1
NC_000067.7 BestRefSeq2CGnomon gene 11063496 111373907 . . . ID=gene-Prex2;Dbxref=GeneID:109294,MGI:MGI:1923385;Name=Prex2;description=phosphatidylinositol-%2C4%2C5-trisphosph
ate-dependent Rac exchange factor 2;gbkey=Gene;gene=Prex2;gene_biotype=protein_coding;gene_synonym=6230420N16Rik,C030645D06Rik,D430013K02,Depd2,P-Rex2
NC_000067.7 BestRefSeq2CGnomon gene 110811802 110905331 . . . ID=gene-Cdh19;Dbxref=GeneID:227485,MGI:MGI:3588198;Name=Cdh19;description=cadherin 19%2C type 2;gbkey=Gene;gene=Cdh
19;gene_biotype=protein_coding
NC_000067.7 BestRefSeq2CGnomon gene 111786431 111792782 . . . ID=gene-Dsel;Dbxref=GeneID:319901,MGI:MGI:2442948;Name=Dsel;description=dermatan sulfate epimerase-like;gbkey=Gene;
gene=Dsel;gene_biotype=protein_coding;gene_synonym=9330132E09Rik,D5-ep1.2
NC_000067.7 BestRefSeq2CGnomon gene 11484121 12046126 . . . ID=gene-A830018L16Rik;Dbxref=GeneID:320492,MGI:MGI:2444149;Name=A830018L16Rik;description=RIKEN cDNA A830018L16 gen
e;gbkey=Gene;gene=A830018L16Rik;gene_biotype=protein_coding
NC_000067.7 BestRefSeq2CGnomon gene 115612467 116515053 . . . ID=gene-Cntnap5a;Dbxref=GeneID:636808,MGI:MGI:3643623;Name=Cntnap5a;description=contactin associated protein-like 5
;gbkey=Gene;gene=Cntnap5a;gene_biotype=protein_coding;gene_synonym=Caspri-5,EG636808
NC_000067.7 BestRefSeq2CGnomon gene 118226244 118239463 . . . ID=gene-Tsn;Dbxref=GeneID:22099,MGI:MGI:109263;Name=Tsn;description=translin;gbkey=Gene;gene=Tsn;gene_biotype=proto
tein coding;gene_synonym=2610034C24Rik,C3P0,TB-RBP
NC_000067.7 BestRefSeq2CGnomon gene 118314976 118537192 . . . ID=gene-Claspl;Dbxref=GeneID:76707,MGI:MGI:1923957;Name=Claspl;description=CLIP associating protein 1;gbkey=Gene;ge
ne=Claspl;gene_biotype=protein_coding;gene_synonym=1700030C23Rik,5730583a19Rik,B130045P17Rik,CLASPLalpha
NC_000067.7 BestRefSeq2CGnomon gene 118555646 118612898 . . . ID=gene-Tfcp2l1;Dbxref=GeneID:81879,MGI:MGI:2444691;Name=Tfcp2l1;description=transcription factor CP2-like 1;gbkey=Gene;ge
ne=Tfcp2l1;gene_biotype=protein_coding;gene_synonym=1810030F05Rik,4932442M07Rik,Cp2l1,Crrt-1,D930018N21Rik,LBP_9,Tfcp2l1
```

Next, I wanted to see what the tx2gene file contains. I went to the work folder I specified to Nextflow and searched for files with that pattern in their name:

```
(base) iweber@iweber-VirtualBox:~/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder$ sudo find -name "*tx2gene*"
[sudo] password for iweber:
./fe/354fd8ac110a2cfffb1588d5cf8ce6/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv # modified 14.8., 15:17
./a7/b0f3a354121a73f22dcc404dd0bce3/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv # modified 14.8., 15:18
./2a/be42ba8de16fcc5c20c7cbbd5075c9/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv # modified 14.8., 14:37
./f9/fa69fe2c73e138ff019c77f1df833e/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv # modified 14.8., 15:17
```

They indeed all contain these odd identifier names, such as the ones shown in the error ([gene-Bod1-ps](#), an existing pseudogene), but also others, like [Gene-Gm13602](#) (a predicted gene).

When opened with LibreOffice, the files all contain 3 columns, which start at the top with

"gene-Bod1-ps gene-Bod1-ps Bod1-ps

gene-Btf3-ps10 gene-Btf3-ps10 Btf3-ps10

gene-Btf3-ps11 gene-Btf3-ps11 Btf3-ps11

gene-Btf3-ps3 gene-Btf3-ps3 Btf3-ps3

gene-Btf3-ps4 gene-Btf3-ps4 Btf3-ps4

gene-Btf3-ps6 gene-Btf3-ps6 Btf3-ps6

gene-Btf3-ps7 gene-Btf3-ps7 Btf3-ps7

gene-Btf3-ps8 gene-Btf3-ps8 Btf3-ps8 " and then, at some point, switch over to something more like

"id-lgha gene-lgha lgha

id-lghd gene-lghd lghd

id-lghd1-1 gene-lghd1-1 lghd1-1

id-lghd2-3 gene-lghd2-3 lghd2-3

id-lghd2-4 gene-lghd2-4 lghd2-4

id-lghd2-5 gene-lghd2-5 lghd2-5

id-lghd2-6 gene-lghd2-6 lghd2-6

id-lghd2-7 gene-lghd2-7 lghd2-7

...,

then,

"rna-AF357341 gene-AF357341 AF357341

rna-AF357428 gene-AF357428 AF357428

rna-ATP6 gene-ATP6 ATP6

rna-ATP8 gene-ATP8 ATP8

rna-COX1 gene-COX1 COX1
rna-COX2 gene-COX2 COX2

...

...

...

rna-Mir100 gene-Mir100 Mir100

rna-Mir100-2 gene-Mir100 Mir100

rna-Mir101a gene-Mir101a Mir101a

rna-Mir101a-2 gene-Mir101a Mir101a

rna-Mir101b gene-Mir101b Mir101b

..."

and, finally, the ones with regular NR NCBI identifiers:

"rna-NM_001001130.3 gene-Zfp85 Zfp85

rna-NM_001001144.3 gene-Scap Scap

rna-NM_001001152.2 gene-Zfp458 Zfp458

rna-NM_001001160.3 gene-Fbxo41 Fbxo41

rna-NM_001001176.2 gene-Taf9b Taf9b

rna-NM_001001177.2 gene-Tsbp1 Tsbp1

rna-NM_001001178.2 gene-Ccdc148"

What is the cause of the tx2gene files containing these identifiers? Can I somehow avoid such identifiers in the creation of the annotations that Salmon needs for quantification so that they don't show up?

And: where do the ENSMUS (Ensembl) type of identifiers come from? In the [tximport error thread](#) I mentioned above, I noticed that the person had this type of identifier in their quant.sf file generated by Salmon, and I went looking for the quant.sf files that Salmon had produced for each of my 8 samples in the results folder. And look what I found:

quant.sf [Read-Only]					
	Name	Length	EffectiveLength	TPM	NumReads
1	ENSMUST00000196221.1	9	9.000	0.000000	0.000
2	ENSMUST00000179664.1	11	11.000	0.000000	0.000
4	ENSMUST00000177564.1	16	16.000	0.000000	0.000
5	ENSMUST00000178537.1	12	12.000	0.000000	0.000
6	ENSMUST00000178862.1	14	14.000	0.000000	0.000
7	ENSMUST00000179520.1	11	11.000	0.000000	0.000
8	ENSMUST00000179883.1	16	16.000	0.000000	0.000
9	ENSMUST00000195858.1	10	10.000	0.000000	0.000
10	ENSMUST00000179932.1	12	12.000	0.000000	0.000
11	ENSMUST00000180001.1	17	17.000	0.000000	0.000
12	ENSMUST00000178815.1	10	10.000	0.000000	0.000
13	ENSMUST00000177965.1	17	17.000	0.000000	0.000
14	ENSMUST00000178909.1	29	29.000	0.000000	0.000
15	ENSMUST00000178230.1	17	17.000	0.000000	0.000
16	ENSMUST00000178483.1	29	29.000	0.000000	0.000
17	ENSMUST00000179560.1	17	17.000	0.000000	0.000
18	ENSMUST00000177839.1	17	17.000	0.000000	0.000
19	ENSMUST00000103439.1	23	23.000	0.000000	0.000
20	ENSMUST00000180266.1	17	17.000	0.000000	0.000
21	ENSMUST00000103441.1	23	23.000	0.000000	0.000
22	ENSMUST00000103567.5	371	68.195	0.000000	0.000
23	ENSMUST00000185651.1	329	51.042	0.000000	0.000
24	ENSMUST00000196939.1	423	95.992	0.000000	0.000
25	ENSMUST00000103568.2	325	49.712	0.000000	0.000
26	ENSMUST00000103569.2	385	74.994	1.547257	1.000
27	ENSMUST00000103570.1	343	56.135	0.000000	0.000
28	ENSMUST00000103571.1	338	54.217	0.000000	0.000
29	ENSMUST00000198019.1	337	53.848	0.000000	0.000
30	ENSMUST00000198058.1	341	55.365	0.000000	0.000
31	ENSMUST00000103573.2	338	54.217	0.000000	0.000
32	ENSMUST00000200127.1	337	53.848	0.000000	0.000
33	ENSMUST00000181483.2	347	57.760	0.000000	0.000
34	ENSMUST00000184883.1	308	44.620	0.000000	0.000
35	ENSMUST00000179789.3	402	83.987	0.000000	0.000
36	ENSMUST00000180717.2	378	71.527	0.000000	0.000

I never used any annotations from Ensembl, but it is possible that the Salmon index I downloaded for the mouse is based on Ensembl/Gencode identifiers! Where is this stored? ChatGPT said it's the txpInfo.bin file, which I cannot directly look into because it's an executable in machine language.

Which means that, sadly, my next step is to build the mouse Salmon index from scratch, using the same GFF file that goes into the pipeline, and then re-run the pipeline.

Build new Salmon index for the mouse genome based on NCBI data

So, back to building Salmon indices: <https://salmon.readthedocs.io/en/latest/salmon.html#preparing-transcriptome-indices-mapping-based-mode>

Since I had previously built the C. elegans index ([Getting dependencies for index building](#)) I have MashMap, I have bedtools, and SalmonTools. I went back to the other [tutorial](#) from Salmon/Alevin regarding how to create the decoys for the index to do the same for the mouse:

I first created the decoys from the NCBI mouse genome:

```
grep '^>' GCF_000001635.27_GRCm39_genomic.fna | cut -d " " -f 1 > decoys.txt  
sed -i.bak -e 's/>//g' decoys.txt
```

and then had a file that looked like this:

```
quant.sf          decoys.txt  
~/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27  
1 NC_000067.7  
2 NT_166280.1  
3 NT_166281.1  
4 NT_166282.1  
5 NT_162750.1  
6 NW_023337852.1  
7 NT_166338.1  
8 NC_000068.8  
9 NC_000069.7  
10 NC_000070.7  
11 NT_187055.1  
12 NC_000071.7  
13 NT_187056.1  
14 NT_187057.1  
15 NT_187058.1  
16 NT_166438.1  
17 NT_187059.1  
18 NC_000072.7  
19 NC_000073.7  
20 NT_166307.1  
21 NC_000074.7  
22 NC_000075.7
```

As I suspect this will cause similar issues as I had with the C. elegans decoys ([Building the decoy-aware transcriptome index for *C. elegans*](#)) with warnings that involved having non-decoy sequences (NC_... ones) after decoy ones -> must make sure that all of the non-decoy sequences (chromosome identifiers and their sequences) appear **first** in the file, and **then** the real decoys (NT, NR, etc). This is also the case in the Fasta file I generated for C. elegans (gentrome_corrected.fa.gz), and the index building worked in this case. So let's sort the decoys.txt I just generated alphabetically:

```
sort decoys.txt > decoys_sorted.txt
```

I could now see that it consists of NC, NT, and NW_ sequences, in this order ([overview transcript identifiers from NCBI RefSeq](#)). But how to make sure they are assembled in the correct order in the "gentrome" Fasta file (the concatenated transcriptome and genome reference file), aka decoys last? I don't know of an easy way to do so directly when creating this concatenated reference file, so I'll do it the same way as for C. elegans before.

I first got the rna.fna file for the mouse transcripts with `wget`

`https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF_000001635.27_GRCm39/GCF_000001635.27_GRCm39_rna.fna.gz` into the same directory with the rest of this genome assembly, and then unzipped it with `pigz -d GCF_000001635.27_GRCm39_rna.fna.gz`. Finally, I ran

```
cat GCF_000001635.27_GRCm39_rna.fna GCF_000001635.27_GRCm39_genomic.fna| pigz > gentrome.fa.gz
```

With Seqkit, I performed

```
seqkit grep -v -f decoys.txt gentrome.fa.gz -o non_decoy.fa
```

to extract the non-decoy information from the compressed file I created before (concatenated transcriptome and genome reference file), and

```
seqkit grep -f decoys.txt gentrome.fa.gz -o decoy.fa
```

to extract specifically the decoys.

To re-assemble the compressed file in the correct order, I ran

```
cat non_decoy.fa decoy.fa | pigz > gentrome_corrected.fa.gz
```

And then attempted to build the index with the corrected file:

```
salmon index -t gentrome_corrected.fa.gz -d decoys.txt -p 12 -i salmon_index
```

Half an hour later, I had the index ("NCBI_Salmon_index_M_musculus_mm10"), with very few warnings (streaks of red equals signs):

```
Version Info: This is the most recent version of salmon.
index ["salmon_index"] did not previously exist . . . creating it
[2024-08-16 14:32:43.574] [jLog] [info] building index
out : salmon_index
[2024-08-16 14:32:43.577] [puff::index::jointLog] [info] Running fixFasta

=====
[Step 1 of 4] : counting k-mers
[2024-08-16 14:32:45.967] [puff::index::jointLog] [warning] Entry with header [NR_028434.1], had length less than equal to the
k-mer length of 31 (perhaps after poly-A clipping)
[2024-08-16 14:32:45.967] [puff::index::jointLog] [warning] Entry with header [NR_028557.1], had length less than equal to the
k-mer length of 31 (perhaps after poly-A clipping)
=====

[2024-08-16 14:33:49.291] [puff::index::jointLog] [warning] Removed 625 transcripts that were sequence duplicates of indexed
transcripts.
[2024-08-16 14:33:49.291] [puff::index::jointLog] [warning] If you wish to retain duplicate transcripts, please use the '--
keepDuplicates' flag
=====

[2024-08-16 14:33:49.291] [puff::index::jointLog] [info] Replaced 73,600,697 non-ATCG nucleotides
[2024-08-16 14:33:49.291] [puff::index::jointLog] [info] Clipped poly-A tails from 9,635 transcripts
wrote 135657 cleaned references
[2024-08-16 14:34:01.734] [puff::index::jointLog] [info] Filter size not provided; estimating from number of distinct k-mers
[2024-08-16 14:34:22.039] [puff::index::jointLog] [info] ntHll estimated 2269740504 distinct k-mers, setting filter size to
2^36
Threads = 12
Vertex length = 31
Hash functions = 5
Filter size = 68719476736
Capacity = 2
Files:
salmon_index/ref_k31_fixed.fa

-----
Round 0, 0:68719476736
Pass      Filling Filtering
1         226      484
2         56       4
True junctions count = 14898458
False junctions count = 4965943
Hash table size = 19864401
Candidate marks count = 347350773

-----
TBB Warning: The number of workers is currently limited to 5. The request for 11 workers is ignored. Further requests for more
workers will be silently ignored until the limit changes.

Reallocating bifurcations time: 5
True marks count: 340969289
Edges construction time: 227

-----
Distinct junctions = 14898458

TwoPaCo::buildGraphMain:: allocated with scalable_malloc; freeing.
TwoPaCo::buildGraphMain:: Calling scalable_allocation_command(TBBMALLOC_CLEAN_ALL_BUFFERS, 0);
allowedIn: 14
Max Junction ID: 14900429
seen.size():119203441 kmerInfo.size():14900430
approximateContigTotalLength: 1121476560
counters for complex kmers:
(prec>1 & succ>1)=3126999 | (succ>1 & isStart)=444 | (prec>1 & isEnd)=457 | (isStart & isEnd)=16
contig count: 25016334 element count: 3022906452 complex nodes: 3127916
# of ones in rank vector: 25016333
[2024-08-16 14:55:26.193] [puff::index::jointLog] [info] Starting the Pufferfish indexing by reading the GFA binary file.
[2024-08-16 14:55:26.193] [puff::index::jointLog] [info] Setting the index/BinaryGfa directory salmon_index
size = 3022906452

-----
| Loading contigs | Time = 261.54 ms

-----
size = 3022906452
```

| Loading contig boundaries | Time = 136.48 ms

```
Number of ones: 25016333
Number of ones per inventory item: 512
Inventory entries filled: 48861
25016333
[2024-08-16 14:55:29.836] [puff::index::jointLog] [info] Done wrapping the rank vector with a rank9sel structure.
[2024-08-16 14:55:30.071] [puff::index::jointLog] [info] contig count for validation: 25,016,333
[2024-08-16 14:55:39.047] [puff::index::jointLog] [info] Total # of Contigs : 25,016,333
[2024-08-16 14:55:39.047] [puff::index::jointLog] [info] Total # of numerical Contigs : 25,016,333
[2024-08-16 14:55:40.249] [puff::index::jointLog] [info] Total # of contig vec entries: 350,903,889
[2024-08-16 14:55:40.249] [puff::index::jointLog] [info] bits per offset entry 29
[2024-08-16 14:55:57.916] [puff::index::jointLog] [info] Done constructing the contig vector. 25016334
[2024-08-16 14:56:14.747] [puff::index::jointLog] [info] # segments = 25,016,333
[2024-08-16 14:56:14.747] [puff::index::jointLog] [info] total length = 3,022,906,452
[2024-08-16 14:56:16.348] [puff::index::jointLog] [info] Reading the reference files ...
[2024-08-16 14:56:34.545] [puff::index::jointLog] [info] positional integer width = 32
[2024-08-16 14:56:34.545] [puff::index::jointLog] [info] seqSize = 3,022,906,452
[2024-08-16 14:56:34.545] [puff::index::jointLog] [info] rankSize = 3,022,906,452
[2024-08-16 14:56:34.545] [puff::index::jointLog] [info] edgeVecSize = 0
[2024-08-16 14:56:34.545] [puff::index::jointLog] [info] num keys = 2,272,416,462
for info, total work write each : 2.331 total work inram from level 3 : 4.322 total work raw : 25.000
[Building BooPHF] 99.9 % elapsed: 1 min 19 sec remaining: 0 min 0 sec
Bitarray 11906737600 bits (100.00 %) (array + ranks )
final hash 218736 bits (0.00 %) (nb in final hash 651)
[2024-08-16 14:57:53.689] [puff::index::jointLog] [info] mphf size = 1419.42 MB
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk size = 251,908,871
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 0 = [0, 251,908,871)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 1 = [251,908,871, 503,817,770)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 2 = [503,817,770, 755,726,641)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 3 = [755,726,641, 1,007,635,512)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 4 = [1,007,635,512, 1,259,544,383)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 5 = [1,259,544,383, 1,511,453,254)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 6 = [1,511,453,254, 1,763,362,125)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 7 = [1,763,362,125, 2,015,270,996)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 8 = [2,015,270,996, 2,267,179,890)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 9 = [2,267,179,890, 2,519,088,761)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 10 = [2,519,088,761, 2,770,997,632)
[2024-08-16 14:57:59.691] [puff::index::jointLog] [info] chunk 11 = [2,770,997,632, 3,022,906,422)
[2024-08-16 15:00:20.657] [puff::index::jointLog] [info] finished populating pos vector
[2024-08-16 15:00:20.657] [puff::index::jointLog] [info] writing index components
[2024-08-16 15:00:56.028] [puff::index::jointLog] [info] finished writing dense pufferfish index
[2024-08-16 15:00:56.806] [jLog] [info] done building index
```

Re-run pipeline

How can I now re-run the pipeline without re-doing the trimming, alignment, and so on?

There is the possibility to [start the pipeline anew from already mapped reads](#): source configuration from genome BAM files or from transcriptome BAM files. Which ones do I have, though? Apparently, I already have both:

Activities Files Fr Aug 16 15:51

	Name	Size	Type	Modified
Recent				
Starred				
Home				
Desktop				
Documents				
Downloads				
Music				
Pictures				
Videos				
Trash				
VMware Tools				
Backup_shared_folder				
Win_Ubuntu_shared				
iweber				
Software				
ABI_files_NGS				
	CONTROL_REPO1.Aligned.out.bam	5,7 GB	Archive	Yesterday
	CONTROL_REPO1.Aligned.toTranscriptome.out.bam	7,1 GB	Archive	Yesterday
	CONTROL_REPO1_sorted.bam	3,1 GB	Archive	Yesterday
	CONTROL_REPO2.Aligned.out.bam	5,9 GB	Archive	Yesterday
	CONTROL_REPO2.Aligned.toTranscriptome.out.bam	7,4 GB	Archive	Yesterday
	CONTROL_REPO2_sorted.bam	3,2 GB	Archive	Yesterday
	CONTROL_REPO2_sorted.bam.bai	2,8 MB	Binary	Yesterday
	CONTROL_REPO3.Aligned.out.bam	5,7 GB	Archive	Yesterday
	CONTROL_REPO3.Aligned.toTranscriptome.out.bam	7,2 GB	Archive	Yesterday
	CONTROL_REPO3_sorted.bam	3,1 GB	Archive	Yesterday

My only concern is that the two kinds of input files completely split what I am able to do with the pipeline: the genome BAM files only let me run the "dexseq_exon", "edger_exon" and "rmats" analysis methods, whereas the transcriptome ones only let me run the "dexseq_dtu" and "suppa" analysis methods. This means I will have to run the remainder of the pipeline as two separate branches, the rMATS/edgeR/DEXseq (exon) branch and the SUPPA/DEXseq branch. Or I could try resuming the previous run with a different salmon_index indicator? I don't think I can do that - when the pipeline stopped yesterday, the processes "SALMON_QUANT_STAR" and "SALMON_QUANT_SALMON" were shown as completed. I suspect I have to run Salmon all over again for both scenarios in order to have it work properly with the new index.

rMATS/edgeR/DEXseq (exon) branch (from genome BAM files)

To use this, I will have to use the `--source genome_bam` argument and the same contrasts sheet but a different sample sheet ("samplesheet_preeclampsia_genomic.csv"):

```
sample,condition,genome_bam
CONTROL_REPO1,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/CONTROL_REPO1.Aligned.out.bam
CONTROL_REPO2,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/CONTROL_REPO2.Aligned.out.bam
CONTROL_REPO3,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/CONTROL_REPO3.Aligned.out.bam
CONTROL_REPO4,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/CONTROL_REPO4.Aligned.out.bam
PREECLAMPSIA_REPO1,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/PREECLAMPSIA_REPO1.Aligned.out.bam
PREECLAMPSIA_REPO2,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/PREECLAMPSIA_REPO2.Aligned.out.bam
PREECLAMPSIA_REPO3,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/PREECLAMPSIA_REPO3.Aligned.out.bam
PREECLAMPSIA_REPO4,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results/star_salmon/PREECLAMPSIA_REPO4.Aligned.out.bam
```

Parameters rMATS/edgeR/DEXseq (exon) branch

I changed folders into the new work folder for the branch with `/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_genomic/`.

I tried to leave as many of the original parameters in place as possible, with the exception of the new work and results folders I created for each of the two branches:

```
sudo nextflow run nf-core/rnas splice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--source genome_bam \
```

```

-w /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_genomic \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnas splice_results_genomic \
--input /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/samplesheet_preeclamp sia_genomic.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker

```

Started at 16:55, crashed around 17:15.

Contrasts sheet issue

The pipeline (hash 941267b9-a228-4895-886f-b62db6df05cf, "happy_lattes") ran a big chunk of the processes contained within:

```

executor > local (60)
[82/668ae4] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1 ✓
[bc/983f70] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GTF_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1 ✓
[60/07d5c2] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1 ✓
[88/713fb1] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1 ✓
[0a/db2bf8] process > NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclamp sia_genomic.csv) [100%] 1 of 1 ✓
[af/b2caf3] process > NFCORE_RNASPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastssheet_preeclamp sia.csv) [100%] 1 of 1 ✓
[41/a10233] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[83/66f690] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_INDEX (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[7d/4df5b7] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_STATS (CONTROL REP1) [100%] 8 of 8 ✓
[ad/3e1437] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT (CONTROL REP1) [100%] 8 of 8 ✓
[7b/b1fe01] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_IDXSTATS (CONTROL REP4) [100%] 8 of 8 ✓
[d5/b86159] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1 ✓
[5e/06e9b6] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_COUNT (CONTROL REP2) [  0%] 0 of 8
[ ] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_EXON
[7f/05a095] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FLATTENTGT (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1 ✓
[d4/785f10] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FEATURECOUNTS (PREECLAMPSIA REP3) [100%] 8 of 8 ✓
[32/43838c] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON (samplesheet_preeclamp sia_genomic.csv) [100%] 1 of 1, failed: 1 ✗
[ ] process > NFCORE_RNASPlice:RNASplice:RMATS:CREATE_BAMLIST
[ ] process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_PREP
[ ] process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_POST
[a9/120e6e] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:GTF_2_GFF3 [100%] 1 of 1 ✓
[d9/7e47cc] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_INDEX [100%] 1 of 1 ✓
[ ] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_RUN [  0%] 0 of 8
[ ] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SETTINGS
[ ] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI
[ ] process > NFCORE_RNASPlice:RNASplice:CUSTOM_DUMPSOFTWAREVERSIONS
[ ] process > NFCORE_RNASPlice:RNASplice:MULTIQC [  0%] 0 of 1
Execution cancelled -- Finishing pending tasks before exit
ERROR ~ Error executing process > 'NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON (samplesheet_preeclamp sia_genomic.csv)'

```

However, it crashed with an error due to edgeR_exon:

```

ERROR ~ Error executing process > 'NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON (samplesheet_preeclamp sia_genomic.csv)'

```

Caused by:

```

Process `NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON (samplesheet_preeclamp sia_genomic.csv)` terminated with an error
exit status (1)

```

Command executed:

```

run_edger_exon.R featurecounts samplesheet_preeclamp sia_genomic.csv contrastssheet_preeclamp sia.csv 10

cat <<-END_VERSIONS > versions.yml
"NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON":
  r-base: $(echo $R --version 2>&1 | sed 's/^.*R version //; s/ .*$///')
  bioconductor-edger: $(Rscript -e "library(edgeR); cat(as.character(packageVersion('edgeR'))))")
END_VERSIONS

```

Command exit status:

1

Command output:

(empty)

```

Command error:
  Unable to find image 'quay.io/biocontainers/mulled-v2-
419bd7f10b2b902489ac63bbaafc7db76f8e0ae1:709335c37934db1b481054cbec637c6e5b5971cb-0' locally
  709335c37934db1b481054cbec637c6e5b5971cb-0: Pulling from biocontainers/mulled-v2-419bd7f10b2b902489ac63bbaafc7db76f8e0ae1
  c1a16a04cedd: Already exists
  4ca545ee6d5d: Already exists
  77553dce0681: Pulling fs layer
  77553dce0681: Verifying Checksum
  77553dce0681: Download complete
  77553dce0681: Pull complete
  Digest: sha256:63ebd4fa165ac8b0a37ee8c5ddd35a0c0abceb191375ffa5155bd7c90811b1bf
  Status: Downloaded newer image for quay.io/biocontainers/mulled-v2-
419bd7f10b2b902489ac63bbaafc7db76f8e0ae1:709335c37934db1b481054cbec637c6e5b5971cb-0
  Loading required package: limma

Warning message:
In read.table(file = file, header = header, sep = sep, quote = quote, :
incomplete final line found by readTableHeader on 'contrastsheet_preeclampsia.csv'
=====
Error in eval(ej, envir = levelsenv) : object 'PREECLAMPSIA' not found
Calls: makeContrasts -> eval -> eval
Execution halted

```

There seem to be issues with the contrasts sheet. The first one is a warning from the side of R that [the last line in the file is not empty](#), as it would expect it to, which is probably not what caused the pipeline to stop. What did, however, cause it to stop, is likely the actual error about the missing object "PREECLAMPSIA". I suspect this is due to how I defined the sample sheet in this case (in the background, R is trying to fit a regression model, and the column names don't fit between the data frame used to build the model vs the one used for the prediction - <https://www.statology.org/error-in-evalpredvars-data-env-object-not-found/>). There may be other issues in the formatting of the sample sheet that cause the problem (<https://help.galaxyproject.org/t/edger-multiple-contrasts-not-working/9707>). I will try changing the sample sheet (that's where I had "control" and "treatment" this time instead of "CONTROL" and "PREECLAMPSIA" like in my previous run of the pipeline), and will try resuming it (I still am in the "genomic" work folder):

```

sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--source genome_bam \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_genomic \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplice_results_genomic \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/samplesheet_preeclamp sia_genomic.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastsheet_preeclamp sia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-resume 941267b9-a228-4895-886f-b62db6df05cf

```

It seems to still want to re-sort the BAM files :(

```
WARN: Both --aligner=star_salmon and --pseudo_aligner=salmon specified. Downstream analyses will be performed on both salmon output files.
executor > local (3)
[82/668ae4] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27 GRCm39 genomic.gff) [100%] 1 of 1, cached: 1 ✓
[bc/983f70] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GTF_GENE_FILTER (GCF_000001635.27 GRCm39 genomic.fna) [100%] 1 of 1, cached: 1 ✓
[60/07d5c2] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27 GRCm39 genomic.fna) [100%] 1 of 1, cached: 1 ✓
[88/713fb1] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27 GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[e8/1a3a83] process > NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampsia_genomic.csv) [100%] 1 of 1 ✓
[b7/60f7fa] process > NFCORE_RNASPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastssheet_preeclampsia.csv) [100%] 1 of 1 ✓
[84/83f8c1] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT (CONTROL_REPI) [  0%] 0 of 8
[ ] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_INDEX
[ ] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_STATS
[ ] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT
[ ] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_IDXSTATS
[d5/b86159] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[ ] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_COUNT
[ ] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_EXON
[7f/05a095] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FLATTENGTF (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[ ] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FEATURECOUNTS
[ ] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON
[ ] process > NFCORE_RNASPlice:RNASplice:RMATS:CREATE_BAMLIST
[ ] process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_PREP
[ ] process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_POST
[a9/120e6e] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:GTF_2_GFF3 [100%] 1 of 1, cached: 1 ✓
[d9/7e47cc] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_INDEX [100%] 1 of 1, cached: 1 ✓
[ ] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_RUN
[ ] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SETTINGS
[ ] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI
[ ] process > NFCORE_RNASPlice:RNASplice:CUSTOM_DUMP SOFTWAREREVERSIONS
[ ] process > NFCORE_RNASPlice:RNASplice:MULTIQC
WARN: The operator 'first' is useless when applied to a value channel which returns a single value by definition
```

Run up to Sashimi crash

```
executor > local (73)
[82/668ae4] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27 GRCm39 genomic.gff) [100%] 1 of 1, cached: 1 ✓
[bc/983f70] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GTF_GENE_FILTER (GCF_000001635.27 GRCm39 genomic.fna) [100%] 1 of 1, cached: 1 ✓
[60/07d5c2] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27 GRCm39 genomic.fna) [100%] 1 of 1, cached: 1 ✓
[88/713fb1] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27 GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[e8/1a3a83] process > NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampsia_genomic.csv) [100%] 1 of 1 ✓
[b7/60f7fa] process > NFCORE_RNASPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastssheet_preeclampsia.csv) [100%] 1 of 1 ✓
[7c/f5a087] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[90/072cec] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_INDEX (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[41/61301a] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_STATS (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[04/02f996] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[8d/ae95b0] process > NFCORE_RNASPlice:RNASplice:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_IDXSTATS (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[d5/b86159] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[23/072064] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_COUNT (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[ ] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_EXON [  0%] 0 of 1
[7f/05a095] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FLATTENGTF (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[55/7d795] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FEATURECOUNTS (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[c9/af3eal] process > NFCORE_RNASPlice:RNASplice:EDGER_EXON (samplesheet_preeclampsia_genomic.csv) [100%] 1 of 1 ✓
[84/ba560d] process > NFCORE_RNASPlice:RNASplice:RMATS:CREATE_BAMLIST (1) [100%] 1 of 1 ✓
[71/3a6b2f] process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_PREP (PREECLAMPSIA-CONTROL) [100%] 1 of 1 ✓
[ ] process > NFCORE_RNASPlice:RNASplice:RMATS_POST [  0%] 0 of 1
[a9/120e6e] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:GTF_2_GFF3 [100%] 1 of 1, cached: 1 ✓
[d9/7e47cc] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_INDEX [100%] 1 of 1, cached: 1 ✓
[e5/6ec8df] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_RUN (8) [100%] 8 of 8 ✓
[1f/d6b6f8] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SETTINGS [100%] 1 of 1 ✓
[f1/2acac7] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI (3) [100%] 3 of 3, failed: 3 ✗
[ ] process > NFCORE_RNASPlice:RNASplice:CUSTOM_DUMP SOFTWAREREVERSIONS
[ ] process > NFCORE_RNASPlice:RNASplice:MULTIQC [  0%] 0 of 1
Execution cancelled -- Finishing pending tasks before exit
-[nf-core/rnasplice] Pipeline completed with errors-
ERROR ~ Error executing process > 'NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI (2)'
```

ERROR ~ Error executing process > 'NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI (2)'

Caused by:

Process `NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI (2)` terminated with an error exit status (1)

Command executed:

```
sashimi_plot --plot-event ENSG00000005302 index miso_settings.txt --output-dir sashimi
```

```
cat <<-END_VERSIONS > versions.yml
```

```
"NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI":
```

```
    python: $(python --version | sed "s/Python //g")
```

```
    misopy: $(python -c "import pkg_resources; print(pkg_resources.get_distribution('misopy').version)")
```

```
END_VERSIONS
```

Command exit status:

1

Command output:

(empty)

Command error:

```
/usr/local/lib/python2.7/site-packages/matplotlib/cbook/deprecation.py:107: MatplotlibDeprecationWarning: The
mpl_toolkits.axes_grid module was deprecated in version 2.1. Use mpl_toolkits.axes_grid1 and mpl_toolkits.axisartist provies
the same functionality instead.
```

```

warnings.warn(message, mplDeprecation, stacklevel=1)
Traceback (most recent call last):
  File "/usr/local/bin/sashimi_plot", line 11, in <module>
    sys.exit(main())
  File "/usr/local/lib/python2.7/site-packages/misopy/sashimi_plot/sashimi_plot.py", line 276, in main
    plot_label=plot_label)
  File "/usr/local/lib/python2.7/site-packages/misopy/sashimi_plot/sashimi_plot.py", line 142, in plot_event
    %(event_name, pickle_dir)
Exception: Event ENSG00000005302 not found in pickled directory index. Are you sure this is the right directory for the event?

Work dir:
/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder_genomic/3c/f8c32ad33e440a86f7b6bf467f5279

```

I tried re-starting it in the same work dir with --sashimi_plot false:

```

sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--sashimi_plot false \
--source genome_bam \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder_genomic \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results_genomic \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/samplesheet_preeclamp sia_genomic.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
--profile docker \
-resume 941267b9-a228-4895-886f-b62db6df05cf

```

SUCCESS! It ran all the way down to creating a MultiQC report about all of the reads it aligned/mapped! [what else?]

SUPPA/DEXseq_dtu branch (from transcriptome BAM files)

To use this, I will have to use the --source transcriptome_bam argument and the same contrasts sheet but a different sample sheet ("samplesheet_preeclamp sia_transcriptomic.csv"):

```

sample,condition,transcriptome_bam
CONTROL_REPO1,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/CONTROL_REPO1_aligned.toTranscriptome.out.bam
CONTROL_REPO2,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/CONTROL_REPO2_aligned.toTranscriptome.out.bam
CONTROL_REPO3,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/CONTROL_REPO3_aligned.toTranscriptome.out.bam
CONTROL_REPO4,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/CONTROL_REPO4_aligned.toTranscriptome.out.bam
PREECLAMPSIA_REPO1,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/PREECLAMPSIA_REPO1_aligned.toTranscriptome.out.bam
PREECLAMPSIA_REPO2,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/PREECLAMPSIA_REPO2_aligned.toTranscriptome.out.bam
PREECLAMPSIA_REPO3,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/PREECLAMPSIA_REPO3_aligned.toTranscriptome.out.bam

```

```
.out.bam  
PREECLAMPSIA_REPO4, PREECLAMPSIA, /home/iweber/Documents/Backup_shared_folder/Pre-  
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnaspli ce_results/star_salmon/PREECLAMPSIA_REPO4.Aligned.toTranscriptome  
.out.bam
```

Parameters SUPPA/DEXseq_dtu branch

I changed folders into the new work folder for the branch with /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_transcriptomic/.

I tried to leave as many of the original parameters in place as possible, with the exception of the new work and results folders I created for each of the two branches:

```
sudo nextflow run nf-core/rnaspli ce -r 1.0.4 \  
--max_cpus 6 \  
--max_memory 90.GB \  
--source transcriptome_bam \  
-w /home/iweber/Documents/Backup_shared_folder/Pre-  
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_transcriptomic \  
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-  
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnaspli ce_results_transcriptomic \  
--input /home/iweber/Documents/Backup_shared_folder/Pre-  
eclamp sia_dataset_raw_and_processed/samplesheet_preeclamp sia_transcriptomic.csv \  
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-  
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \  
--fasta  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \  
--gff  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \  
--save_trimmed true \  
--aligner star_salmon \  
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \  
--salmon_index  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \  
--save_unaligned true \  
--save_align_intermeds true \  
--save_reference true \  
-profile docker
```

...which failed fairly fast with an error:

```
ERROR ~ Error executing process > 'NFCore_RNASPLICE:RNASPLICE:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT (PREECLAMPSIA_REPO2)'  
  
Caused by:  
Process 'NFCore_RNASPLICE:RNASPLICE:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT (PREECLAMPSIA_REPO2)' terminated with an error exit  
status (137)
```

Command executed:

```
 samtools sort \  
 \\  
 -@ 6 \  
 -m 12288M \  
 -o PREECLAMPSIA_REPO2_sorted.bam \  
 -T PREECLAMPSIA_REPO2_sorted \  
 PREECLAMPSIA_REPO2Aligned.toTranscriptome.out.bam  
  
 cat <<-END VERSIONS > versions.yml  
 "NFCore_RNASPLICE:RNASPLICE:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT":  
     samtools: $(echo $(samtools --version 2>&1) | sed 's/^.*samtools //; s/Using.*$/')  
 END VERSIONS
```

Command exit status:

137

Command output:
(empty)

Command error:

```
.command.sh: line 8: 33 Killed samtools sort -@ 6 -m 12288M -o PREECLAMPSIA_REPO2_sorted.bam -T  
PREECLAMPSIA_REPO2_sorted PREECLAMPSIA_REPO2Aligned.toTranscriptome.out.bam
```

Work dir:
/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder_transcriptomic/19/69cc0da8f00c5757bce5c3b01b150c

Fixing 137 error

The exit code 137 seems to have something to do with processes being out of memory, and, looking at the Command error "33 Killed samtools sort -@6 -m 12288M -o PREECLAMPSIA REP2_sorted.bam -T PREECLAMPSIA REP2_sorted PREECLAMPSIA REP2.Aligned.toTranscriptome.out.bam", I think the way I indicated max memory until now isn't working: there's a parameter that says `-m 12288M`. The way the pipeline creators do on the Custom Configuration page is without the dot notation, so let's try again, resuming the same run (this one was 10497c48-b4b0-4990-a919-31eaa1fb3a29 , ecstatic_mercator):

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90GB \
--source transcriptome_bam \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder_transcriptomic \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results_transcriptomic \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/samplesheet_preeclamp sia_transcriptomic.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-resume 10497c48-b4b0-4990-a919-31eaa1fb3a29
```

...but this didn't change the outcome. I got the exact same error, with `samtools_sort` still having the parameter `-m 12288M`. Somehow, the limits I set to the pipeline overall don't seem to work for `samtools`, oddly enough. This means I have to create a custom configuration file to run the pipeline with.

Custom configuration file to fix memory usage issue for Samtools

The process I want to change the options for is one of the "nf-core" subworkflows, "bam_sort_stats_samtools". The "main.nf" Nextflow script in the [bam_sort_stats_samtools GitHub folder](#) seems to not be the script that causes the issue itself (description of the [bam_sort_stats_samtools subworkflow](#)), but to just be used to call another one from a module, [nf-core/samtools/sort](#) and another module that makes the index, [samtools_index](#). Both modules could be causing issues with the memory use, and I don't understand why limiting the memory overall doesn't do it for these two modules as well.

So, since I don't want to create a config file from scratch, not knowing how the original one is set up for the pipeline, perhaps going to the roots is the answer: where is the config file used to set how much memory processes are allowed to use? The scripts themselves listed above don't contain parameter settings for memory use. I then read here about [how to tune pipeline resources](https://nf-co.re/docs/usage/getting_started/configuration#tuning-workflow-resources) and found out that all of the pipelines from nf-core do so using a `base.config` file. This means that I need to find out 1. where this is, 2. copy its contents with the memory adjustment to fit the maximal resources my PC can use, and 3. make the pipeline use the new config file instead of the `base.config`.

After some searching on my system, I found that the `base.config` file is located in `/home/iweber/.nextflow/assets/nf-core/rnasplice/` and looks like this:

Open

+1

base.config
~/.nextflow/assets/nf-core/rnasplice/conf

samplesheet_preecl...a_transcriptomic.csv

X

```

21 // Process-specific resource requirements
22 // NOTE - Please try and re-use the labels below as much as possible.
23 // These labels are used and recognised by default in DSL2 files hosted on nf-core/modules.
24 // If possible, it would be nice to keep the same label naming convention when
25 // adding in your local modules too.
26 // TODO nf-core: Customise requirements for specific processes.
27 // See https://www.nextflow.io/docs/latest/config.html#config-process-selectors
28 withLabel:process_single {
29     cpus = { check_max( 1 , 'cpus' ) }
30     memory = { check_max( 6.0GB * task.attempt, 'memory' ) }
31     time = { check_max( 4.h * task.attempt, 'time' ) }
32 }
33 withLabel:process_low {
34     cpus = { check_max( 2 * task.attempt, 'cpus' ) }
35     memory = { check_max( 12.0GB * task.attempt, 'memory' ) }
36     time = { check_max( 4.h * task.attempt, 'time' ) }
37 }
38 withLabel:process_medium {
39     cpus = { check_max( 6 * task.attempt, 'cpus' ) }
40     memory = { check_max( 36.0GB * task.attempt, 'memory' ) }
41     time = { check_max( 8.h * task.attempt, 'time' ) }
42 }
43 withLabel:process_high {
44     cpus = { check_max( 12 * task.attempt, 'cpus' ) }
45     memory = { check_max( 60.0GB * task.attempt, 'memory' ) }
46     time = { check_max( 16.h * task.attempt, 'time' ) }
47 }
48 withLabel:process_long {
49     time = { check_max( 20.h * task.attempt, 'time' ) }
50 }
51 withLabel:process_high_memory {
52     memory = { check_max( 200.0GB * task.attempt, 'memory' ) }
53 }
54 withLabel:error_ignore {
55     ...
56 }
```

There's indeed an option for processes using high memory to be allowed to go up to 200 GB of RAM usage...which, of course, is not doable on my virtual machine that only allows 95 GB, and of which the pipeline should not use more than 90 GB.

I changed the process_high_memory to

```

withLabel:process_high_memory {
    memory = { check_max( 90.0GB * task.attempt, 'memory' ) }
}
```

and saved the file. I closed and re-opened the terminal, and re-ran the previous command in the transcriptomic work folder....no improvement - I still got the "terminated with an error exit status (137)" errors. Is it possible that the settings in the base.config file are overridden by settings set in some other config file?

Or maybe the issue is even simpler: Looking at the error more closely, I saw that it says "Killed samtools sort -@ 6 -m 12288M". For some reason, my brain read the amount megabyte as 128 GB, which, of course, it is not: it's 12 GB! This means that Samtools sort likely does not fall under the label of process_high_memory within the Nextflow base configuration file! In its own [main.nf](#) file among the nf-core modules, it is labelled as a process_medium! So I went back to my base.config file and changed the memory allotted to process_medium to 90 GB. Let's see if this does the trick...

```

withLabel:process_medium {
    cpus = { check_max( 6 * task.attempt, 'cpus' ) }
    memory = { check_max( 90.0GB * task.attempt, 'memory' ) }
    time = { check_max( 8.h * task.attempt, 'time' ) }
}
```

....nope, it did not. Same error.

Updating modules and subworkflows to fix Samtools memory issues

I did some more research online, and found this as an open issue here: <https://github.com/nf-core/rnasplice/issues/145>, which pointed me to a potential solution: <https://github.com/nf-core/methylseq/pull/330> simply use nf-core modules update to update all of the core modules, such as Samtools, used by this pipeline and others.

I got a prompt saying

```

  ,--./,-.
  / -.-.---\
  } { 
 \``-.-,-`-
 ``-,-,-'

```

nf-core/tools version 2.14.1 – <https://nf-co.re>

```

WARNING 'repository_type' not defined in .nf-core.yml
? Is this repository an nf-core pipeline or a fork of nf-core/modules?
==Pipeline==

```

```

INFO To avoid this prompt in the future, add the 'repository_type' key to your .nf-core.yml file.

```

```

? Would you like me to add this config now? [y/n] (y):
==y==

```

After selecting that this is a pipeline and yes, I want to add the config, I get an error "PermissionError: [Errno 13] Permission denied: './.nf-core.yml'". When I tried to repeat the command with sudo (`sudo nf-core modules update`), it tells me: "sudo: nf-core: command not found". Why is this happening, and how can I run the command with sudo privileges?

Turns out, using `sudo` doesn't just give you localized privileges for that one command you want to execute. `sudo`, in fact, creates a whole new shell, even though it works in the same terminal, and that shell doesn't always have identical settings to the one you use as a non-super-user! ([source](#)). That also means that commands your regular shell knows how to find because of the addresses to their binaries being stored in the PATH variable don't necessarily translate to the sudo shell, because it uses a different PATH variable! It's like activating a different user account on the computer, where not all of the software from other user accounts is accessible.

To make sudo use the exact same path, I could add the option `-E` to my command, which I will do for now:

```

sudo -E nf-core modules update

```

...and which didn't work.

A more permanent solution is to modify the sudoers file (open `/etc/sudoers` with `sudo visudo`, [source](#)) and add to it

```

# Keep path variable from current shell
Defaults env_keep+="PATH"

```

(([source](#)).

I opened the sudoers file with sudo visudo, then tried typing the command above in it, which gave me an odd error message "no string under the cursor" (thanks, Vim, I guess? LE: I hadn't activated "insert" mode). Then, I performed a paste operation with Ctrl+V and accidentally pasted in something utterly unnecessary (a link to a website). When I wanted to undo it with Ctrl+Z, it threw me out of the document and back to the shell. If I now try to re-open the sudoers file again, it tells me "visudo: /etc/sudoers busy, try again later". How to cancel everything I did and make sure I don't break my sudoers file and my system with it?

I used `ps aux | grep visudo` to check for running visudo processes, and found four, three from the root user and one from my personal user:

```

root      6148  0.0  0.0  14700  6144 pts/0    T   11:13   0:00 sudo visudo
root      6149  0.0  0.0  14700  2484 pts/1    Ss  11:13   0:00 sudo visudo
root      6150  0.0  0.0   9044  2816 pts/1    T+  11:13   0:00 visudo
iweber    6167  0.0  0.0   9564  2688 pts/0    S+  11:20   0:00 grep --color=auto visudo

```

They have little status indicators: T = stopped, T+ = stopped, Ss = sleeping, and the grep command I just ran, which isn't relevant. So I could kill the first three processes, but I want more safety operations before that, such as trying to close the file without saving. To switch to the still running visudo process, I ran `jobs`, which showed me

```

[1]+  Stopped                  sudo visudo

```

I then ran `fg %1` to bring this process into the foreground, and could see the contents of the sudoers file again. I deleted what I had previously accidentally inserted, but also chose to exit without saving anything, just to be on the extra safe side, by typing `:q!`. I got back a message saying `visudo: /etc/sudoers.tmp unchanged`, which sounds promising. I ran the `ps aux | grep visudo` command again to check for any leftover running processes, and got only the grep as a running process for my user:

```

iweber    6181  0.0  0.0   9564  2688 pts/0    S+  11:34   0:00 grep --color=auto visudo

```

Because once burnt, twice shy, I made a backup of the sudoers file with

```
sudo cp sudoers sudoers.bak
```

, then opened it with `sudo visudo`. I moved up and down with the arrows to find the other Defaults, hit "i" to enter the INSERT mode of Vim, and then typed `Defaults env_keep+="PATH"`. I clicked `Esc` to exit the inserting mode, then typed `:wq` to write and quit.

And it was all in vain. I then re-started my terminal and typed the nf-core update command again, only to, again, get `sudo: nf-core: command not found`. I activated the Nextflow conda environment I had created, that also didn't help.

I then tried running the command with the full path to the executable. With the Conda environment activated, I ran `which nf-core` and got back `/home/iweber/anaconda3/envs/Nextflow/bin/nf-core`. I can now try running the command with sudo but using the full path:

```
sudo /home/iweber/anaconda3/envs/Nextflow/bin/nf-core modules update
```

I immediately got these strange prompts:

```
WARNING 'repository_type' not defined in .nf-core.yml
? Is this repository an nf-core pipeline or a fork of nf-core/modules?
```

I selected "Pipeline", and then it proceeded:

```
INFO To avoid this prompt in the future, add the 'repository_type' key to
your .nf-core.yml file.
? Would you like me to add this config now? [y/n] (y): y
INFO Config added to '.nf-core.yml'
WARNING Could not find a 'main.nf' or 'nextflow.config' file in '/'
INFO The 'modules.json' file is not up to date. Recreating the
'modules.json' file.
? Can't find a ./modules directory. Would you like me to create one? [y/n] (y):
```

I wasn't sure why the `./modules` directory wasn't where it was supposed to be, so I typed n to cancel the operation. I thought that perhaps the problem is that I am not working in the correct location for the update, and perhaps I need to switch to the pipeline's folder, which I did with `cd /home/iweber/.nextflow/assets/nf-core/rnasplice/`. I tried repeating the update command here, which gave me an error saying "FileNotFoundException: [Errno 2] No such file or directory: 'pre-commit'" and a giant traceback:

```
          ,--./,--.
          /,-.-,-~\ \
        } {
\`-.,-`-
`.-,-`-
```

nf-core/tools version 2.14.1 - <https://nf-co.re>

```
Traceback (most recent call last)
/home/iweber/anaconda3/envs/Nextflow/bin/nf-core:10 in <module>
    7
    8 if __name__ == '__main__':
    9 |     sys.argv[0] = re.sub(r'(-script\.pyw)?|\.exe)?$', '', sys.argv[0])
  > 10 |     sys.exit(run_nf_core())
    11

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/nf_core/__main__.py:150 in
run_nf_core
    147 |     |     log.debug(f"Could not check latest version: {e}")
    148 |     |     stderr.print("\n")
    149 |     # Launch the click cli
  > 150 |     nf_core_cli(auto_envvar_prefix="NFCORE")
    151
    152
    153 @tui()

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/rich_click/rich_command.py:367 in __call__
    364 |     |     # Include this here because I run into a false warning
    365 |     |     # in the PyCharm IDE otherwise; for some reason PyCharm doesn't
    366 |     |     # seem to think RichGroups are callable. (No issues with Mypy, though.)
  > 367 |     |     return super().__call__(*args, **kwargs)
    368
```

```
369
370 class RichCommandCollection(CommandCollection, RichGroup):
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/click/core.py:1157 in __call__
1154 |
1155     def __call__(self, *args: t.Any, **kwargs: t.Any) -> t.Any:
1156         """Alias for :meth:`main`"""
1157         return self.main(*args, **kwargs)
1158
1159
1160 class Command(BaseCommand):
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/rich_click/rich_command.py:152 in main
149     try:
150         try:
151             with self.make_context(prog_name, args, **extra) as ctx:
152                 rv = self.invoke(ctx)
153                 if not standalone_mode:
154                     |   return rv
155             # it's not safe to `ctx.exit(rv)` here!
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/click/core.py:1688 in invoke
1685     super().invoke(ctx)
1686     sub_ctx = cmd.make_context(cmd_name, args, parent=ctx)
1687     with sub_ctx:
1688         |   return _process_result(sub_ctx.command.invoke(sub_ctx))
1689
1690     # In chain mode we create the contexts step by step, but after the
1691     # base command has been invoked. Because at that point we do not
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/click/core.py:1688 in invoke
1685     super().invoke(ctx)
1686     sub_ctx = cmd.make_context(cmd_name, args, parent=ctx)
1687     with sub_ctx:
1688         |   return _process_result(sub_ctx.command.invoke(sub_ctx))
1689
1690     # In chain mode we create the contexts step by step, but after the
1691     # base command has been invoked. Because at that point we do not
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/click/core.py:1434 in invoke
1431     |   echo(style(message, fg="red"), err=True)
1432
1433     if self.callback is not None:
1434         |   return ctx.invoke(self.callback, **ctx.params)
1435
1436     def shell_complete(self, ctx: Context, incomplete: str) -> t.List["CompletionItem"]:
1437         """Return a list of completions for the incomplete value. Looks
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/click/core.py:783 in invoke
780     |
781     with augment_usage_errors(__self__):
782         |   with ctx:
1) 783         |   |   return __callback(*args, **kwargs)
784
785     def forward(
786         |   __self__, __cmd: "Command", *args: t.Any, **kwargs: t.Any # noqa: B902
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/click/decorators.py:33 in new_func
30     """
31
32     def new_func(*args: "P.args", **kwargs: "P.kwargs") -> "R":
1) 33         |   return f(get_current_context(), *args, **kwargs)
34
35     return update_wrapper(new_func, f)
36
    ...
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/nf_core/__main__.py:940 in
```

```
modules_update

937 |     |     |     ctx.obj["modules_repo_branch"],
938 |     |     |     ctx.obj["modules_repo_no_pull"],
939 |     |
940 |     |     exit_status = module_install.update(tool)
941 |     |     if not exit_status and install_all:
942 |     |         sys.exit(1)
943 |     except (UserWarning, LookupError) as e:

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/nf_core/components/update.py:1
05 in update

102 |     |
103 |     # Verify that 'modules.json' is consistent with the installed modules
104 |     if not silent:
105 |         self.modules_json.check_up_to_date()
106 |
107 |     if not self.update_all and component is None:
108 |         choices = [f"All {self.component_type}", f"Named {self.component_type[:-1]}"]

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/nf_core/modules/modules_json.p
y:627 in check_up_to_date

624 |     self.pipeline_components = original_pipeline_components
625 |
626 |     if dump_modules_json:
627 |         self.dump(run_prettier=True)
628 |
629 |     def load(self):
630 |         """

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/nf_core/modules/modules_json.p
y:1061 in dump

1058 |     |     # Sort the modules.json
1059 |     |     self.modules_json["repos"] = nf_core.utils.sort_dictionary(self.modules_json)
1060 |     |     if run_prettier:
1061 |     |         dump_json_with_prettier(self.modules_json_path, self.modules_json)
1062 |     |     else:
1063 |     |         with open(self.modules_json_path, "w") as fh:
1064 |     |             json.dump(self.modules_json, fh, indent=4)

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/nf_core/lint_utils.py:104 in
dump_json_with_prettier

101 |     """
102 |     with open(file_name, "w") as fh:
103 |         json.dump(file_content, fh, indent=4)
104 |     run_prettier_on_file(file_name)
105 |
106 |
107 def ignore_file(lint_name: str, file_path: Path, dir_path: Path) -> List[List[str]]:

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/site-packages/nf_core/lint_utils.py:77 in
run_prettier_on_file

74 |
75 |     nf_core_pre_commit_config = Path(nf_core.__file__).parent / ".pre-commit-prettier-co
76 |     try:
77 |         subprocess.run(
78 |             ["pre-commit", "run", "--config", nf_core_pre_commit_config, "prettier", "--",
79 |             capture_output=True,
80 |             check=True,

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/subprocess.py:548 in run

545 |     |     kwargs['stdout'] = PIPE
546 |     |     kwargs['stderr'] = PIPE
547 |
548 |     with Popen(*popenargs, **kwargs) as process:
549 |         try:
550 |             |             stdout, stderr = process.communicate(input, timeout=timeout)
551 |             except TimeoutExpired as exc:

/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/subprocess.py:1026 in __init__
```

```

1023 |     |     |     |     self.stderr = io.TextIOWrapper(self.stderr,
1024 |     |     |     |     |     encoding=encoding, errors=errors)
1025 |
1026 |     |     |     self._execute_child(args, executable, preexec_fn, close_fds,
1027 |     |     |     |     |     pass_fds, cwd, env,
1028 |     |     |     |     |     startupinfo, creationflags, shell,
1029 |     |     |     |     p2cread, p2cwrite,

```

```
/home/iweber/anaconda3/envs/Nextflow/lib/python3.12/subprocess.py:1955 in _execute_child
```

```

1952 |     |     |     if errno_num != 0:
1953 |     |     |     |     err_msg = os.strerror(errno_num)
1954 |     |     |     if err_filename is not None:
1955 |     |     |     |     raise child_exception_type(errno_num, err_msg, err_filename)
1956 |     |     |     else:
1957 |     |     |     |     raise child_exception_type(errno_num, err_msg)
1958 |     |     |     raise child_exception_type(err_msg)

```

```
FileNotFoundException: [Errno 2] No such file or directory: 'pre-commit'
```

Where could this file or directory named **pre-commit** be located? This might give me an indication of where I need to run the update modules command for it to take effect. A simple "find /home -name pre-commit" returned a bunch of files from my initial Nextflow work folder ("Pre_eclampsia_Nextflow_work_folder") that are locked and give me a "permission denied" error. Using `sudo find /home -name pre-commit` returned exactly one folder, "/home/iweber/anaconda3/envs/Nextflow/bin/pre-commit". So maybe this is where I need to change to in order to run the update command?

Let's try `cd /home/iweber/anaconda3/envs/Nextflow/bin/` and then `sudo /home/iweber/anaconda3/envs/Nextflow/bin/nf-core modules update`. Which means it's still not the right folder. I still think that the `/home/iweber/.nextflow/assets/nf-core/rnasplice/` folder is the best bet: this is for sure where the modules folder is. The file nf-core itself is a Python script that imports a module from the `nf_core.__main__` library called `run_nf_core` and then proceeds to call whichever script one specifies when one calls the nf-core scripts to be executed from the command lines. This means that, in my case, what I am trying to do is to execute the nf-core script with the option "modules", which, in turn, will call the "modules" script, that probably includes a function or link to another script called "update".

I realized at some point that I can try updating the thing itself that causes the issue, in this case, the subworkflow that the module samtools is included in, namely `bam_sort_stats_samtools`.

I changed to the directory where the rnasplice pipeline is stored, `/home/.nextflow/assets/nf-core/rnasplice`, and then ran

```
nf-core subworkflows update bam_sort_stats_samtools
```

I got a message asking if I wanted to see what had changed between my version and the current version:

```

? Do you want to view diffs of the proposed changes? Preview diff in terminal, choose whether to update files
INFO    Changes in module 'nf-core/bam_sort_stats_samtools' between (735e1e04e7e01751d2d6e97055bbdb6f70683cc1) and
(46eca555142d6e597729fc682adcc791796f514)
INFO    Changes in 'bam_sort_stats_samtools/main.nf':
--- subworkflows/nf-core/bam_sort_stats_samtools/main.nf
+++ subworkflows/nf-core/bam_sort_stats_samtools/main.nf
@@ -15,7 +15,7 @@
     ch_versions = Channel.empty()
-
 SAMTOOLS_SORT ( ch_bam )
+
 SAMTOOLS_SORT ( ch_bam, ch_fasta )
     ch_versions = ch_versions.mix(SAMTOOLS_SORT.out.versions.first())
     SAMTOOLS_INDEX ( SAMTOOLS_SORT.out.bam )
INFO    Changes in 'bam_sort_stats_samtools/meta.yml':
--- subworkflows/nf-core/bam_sort_stats_samtools/meta.yml
+++ subworkflows/nf-core/bam_sort_stats_samtools/meta.yml
@@ -6,12 +6,13 @@
     - bam
     - sam
     - cram
-modules:
+components:
     - samtools/sort
     - samtools/index
     - samtools/stats
     - samtools/idxstats
     - samtools/flagstat
+    - bam_stats_samtools
input:
     - meta:
         type: map

```

```

@@ -64,3 +65,6 @@
authors:
- "@drpatelh"
- "@ewels"
+maintainers:
+ - "@drpatelh" + - "@ewels"
INFO    'subworkflows/nf-core/bam_sort_stats_samtools/tests/main.nf.test.snap' was created
INFO    'subworkflows/nf-core/bam_sort_stats_samtools/tests/tags.yml' was created
INFO    'subworkflows/nf-core/bam_sort_stats_samtools/tests/main.nf.test' was created
? Update subworkflow 'bam_sort_stats_samtools'? (y/N)

```

There are indeed changes to the samtools module, so I chose "yes". I got another prompt:

```

WARNING All modules and subworkflows linked to the updated subworkflow will be asked for update.
It is advised to keep all your modules and subworkflows up to date.
It is not guaranteed that a subworkflow will continue working as expected if all modules/subworkflows used in it are
not up to date.
? Would you like to continue updating all modules and subworkflows? (Y/n)

```

and, since it sounds sensible, I chose "yes" again. I had to confirm individually if I want to update each individual module (and, strangely enough, I was not asked for any sudo permissions this time around - maybe because I had already "logged in" as a sudo user earlier in the same shell?). I saved all of the messages in a text file called "2024-08-18_update_subworkflows_completed".

Re-run pipeline after subworkflow/module update

I attempted to re-run the pipeline from yesterday, before I had updated all of the subworkflows. I double-checked I am resuming the correct pipeline by looking up the hash with `sudo nextflow log`

```

sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90GB \
--source transcriptome_bam \
-w /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_transcriptomic \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplice_results_transcriptomic \
--input /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/samplesheet_preeclamp sia_transcriptomic.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-resume 10497c48-b4b0-4990-a919-31eaa1fb3a29

```

Annnnd same problem as before.

I tried updating absolutely all of the modules with `nf-core` modules update --all from within the rnasplice folder, then `nf-core` subworkflows update --all, which told me that the samtools-related ones were already up to date:

```

? Do you want to view diffs of the proposed changes? No previews, just update everything
# Modules
INFO Updating 'nf-core/cat/fastq'
INFO Updating 'nf-core/custom/dumpsoftwareversions'
INFO Updating 'nf-core/custom/getchromsizes'
INFO Updating 'nf-core/fastqc'
INFO Updating 'nf-core/fastqc_fastqc_umitools_trimgalore'
INFO Updating 'nf-core/trimgalore'
INFO Updating 'nf-core/umitools/extract'
INFO Updating 'nf-core/gffread'
INFO Updating 'nf-core/gunzip'

```

```

INFO Updating 'nf-core/multiqc'
INFO Updating 'nf-core/rsem/preparereference'
INFO Updating 'nf-core/salmon/index'
INFO Updating 'nf-core/salmon/quant'
INFO 'modules/nf-core/samtools/flagstat' is already up to date
INFO 'modules/nf-core/samtools/idxstats' is already up to date
INFO 'modules/nf-core/samtools/index' is already up to date
INFO 'modules/nf-core/samtools/sort' is already up to date
INFO 'modules/nf-core/samtools/stats' is already up to date
INFO Updating 'nf-core/star/align'
INFO Updating 'nf-core/subread/featurecounts'
INFO 'modules/nf-core/trimgalore' is already up to date
INFO Updating 'nf-core/ucsc/bedclip'
INFO Updating 'nf-core/bedgraph_bedclip_bedgraphtobigwig'
INFO Updating 'nf-core/ucsc/bedgraphtobigwig'
INFO 'modules/nf-core/ucsc/bedgraphtobigwig' is already up to date
INFO 'modules/nf-core/umitools/extract' is already up to date
INFO Updating 'nf-core/untar'
INFO Updates complete ✨

```

Subworkflows

```

INFO 'subworkflows/nf-core/bam_sort_stats_samtools' is already up to date
INFO 'subworkflows/nf-core/bam_stats_samtools' is already up to date
INFO 'subworkflows/nf-core/bedgraph_bedclip_bedgraphtobigwig' is already up to date
INFO 'subworkflows/nf-core/fastq_fastqc_umitools_trimgalore' is already up to date
INFO Updates complete ✨

```

I decided it might make more sense to simply try the entirety of the pipeline again, since I did not have this issue neither in previous runs, nor when running the genomic branch.

Maybe interesting for some point in the future: [optimizing memory settings for Samtools sort](#)

Attempt to re-run pipeline in its entirety in hopes it will recognize the already aligned files

I could try to run the pipeline again just using `--pseudo_aligner salmon` to re-generate the Salmon quantification files based on the new index, in addition to `--skip_alignment` parameter set to `true` to not re-create the alignments. However, I am not sure if this will fix the issue with `SAMTOOLS_SORT`, which was not an issue in the genomic branch of the pipeline.

To be sure-sure I am working in the work folder I want to set:

```
cd /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/
```

Activate the Nextflow conda environment:

```
conda activate Nextflow
```

I tried to re-run the exact same command as on August 15th, skipping the parameters for clipping, including the path to the new Salmon index, and appended `-resume d2143e7b-2e6a-41e4-850d-cb8af9596f82`.

```

sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre- \
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplice_results \
--input /home/iweber/Documents/Backup_shared_folder/Pre- \
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/samplesheet_preeclamp sia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre- \
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \

```

```
-profile docker \
-resume d2143e7b-2e6a-41e4-850d-cb8af9596f82
```

As an agnostic, I prayed to my pantheon of deities that the pipeline would recognize already generated results and that space on my hard drive will be enough for the new results. Started around 15:20 PM.

It failed around 7 PM due to the same error around tximport.

Activities Terminal So Aug 18 22:37

iweber@iweber-VirtualBox: ~/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nex

```
executor > local (37)
[4f788081] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1, cached: 1 ✓
[85e3b1ab] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GTF_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[159138d2] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[c4/ed4701] process > NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
[6a/018809] process > NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampsia.csv) [100%] 1 of 1, cached: 1 ✓
[3d/4f37d7] process > NFCORE_RNASPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastsheet_preeclampsia.csv) [100%] 1 of 1, cached: 1 ✓
- process > NFCORE_RNASPlice:RNASplice:CAT_FASTQ -
[ec/63d490] process > NFCORE_RNASPlice:RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:FASTQC (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
[ee/7705ef] process > NFCORE_RNASPlice:RNASplice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:TRIMGALORE (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 8 ✓
[5a/858213] process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:STAR_ALIGN (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
[87/ddf40] process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_SORT (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
[d2/e28bcd] process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:SAMTOOLS_INDEX (PREECLAMPSIA REP4) [100%] 8 of 8, cached: 3 ✓
- process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_STATS [0%] 0 of 8
- process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT [0%] 0 of 8
- process > NFCORE_RNASPlice:RNASplice:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_IDXSTATS [0%] 0 of 8
[a7/435c5b] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[c5/4e272d] process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_COUNT (PREECLAMPSIA REP3) [100%] 8 of 8 ✓
- process > NFCORE_RNASPlice:RNASplice:DEXSEQ_DEU:DEXSEQ_EXON [0%] 0 of 1
[b2/338963] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FLATTENTGT (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[02/78452c] process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:SUBREAD_FEATURECOUNTS (PREECLAMPSIA REP3) [100%] 8 of 8 ✓
- process > NFCORE_RNASPlice:RNASplice:EDGER_DEU:EDGER_EXON [0%] 0 of 1
[b9/ble07d] process > NFCORE_RNASPlice:RNASplice:RMATS:CREATE_BAMLIST (1) [100%] 1 of 1 ✓
- process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_PREP [0%] 0 of 1
- process > NFCORE_RNASPlice:RNASplice:RMATS:RMATS_POST -
[87/9c14e3] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:GTF_2_GFF3 [100%] 1 of 1, cached: 1 ✓
[5e/89ed70] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_INDEX [100%] 1 of 1, cached: 1 ✓
[60/0607b3] process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_RUN (4) [50%] 4 of 8
- process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SETTINGS -
- process > NFCORE_RNASPlice:RNASplice:VISUALISE_MISO:MISO_SASHIMI -
[99/5490e2] process > NFCORE_RNASPlice:RNASplice:SALMON_QUANT_STAR (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
- process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_STAR_SALMON_UNTAR -
[a7/b0f3a3] process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_STAR_SALMON_GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[3c/3593b1] process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_STAR_SALMON_TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv) [100%] 1 of 1, cached: 1 ✓
- process > NFCORE_RNASPlice:RNASplice:DRIMSEQ_DEXSEQ_DU_STAR_SALMON:DRIMSEQ_FILTER [0%] 0 of 1
- process > NFCORE_RNASPlice:RNASplice:DRIMSEQ_DEXSEQ_DU_STAR_SALMON:STAGER -
- process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:SPLIT_FILES_TPM [0%] 0 of 1
[69/92c17f] process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
- process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:PSIPEREVENT [0%] 0 of 1
- process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:SPLIT_FILES_IOE -
- process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:DIFFSPLICE_IOE -
- process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:CLUSTERGROUPS_IOE -
- process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:CLUSTEREVENTS_IOE -
[fc/ffa5ce] process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
- process > NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:PSIPERISOFORM [0%] 0 of 1
[d6/2b2e91] process > NFCORE_RNASPlice:RNASplice:SALMON_QUANT_SALMON (PREECLAMPSIA REP4) [100%] 8 of 8 ✓
[fe/354fda] process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON_GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[e4/91ebd6] process > NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON_TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv) [100%] 1 of 1, failed: 1 ✘
[8f/025573] process > NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
[df/8384fe] process > NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
```

ERROR ~ Error executing process > 'NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON:TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv)'

Caused by:

Process `NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON:TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv)` terminated with an error exit status (1)

Command executed:

```
tximport.R GCF_000001635.27_GRCm39_genomic.tx2gene.tsv salmon salmon.merged
```

```
cat <<-END_VERSIONS > versions.yml
"NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON:TXIMPORT":
  r-base: $(echo $(R --version 2>&1) | sed 's/^.*R version //; s/ .*$/\\\'')
  bioconductor-tximeta: $(Rscript -e "library(tximeta); cat(as.character(packageVersion('tximeta'))))")
END_VERSIONS
```

Command exit status:

```
1
```

Command output:

```
(empty)
```

Command error:

```
reading in files with read.delim (install 'readr' package for speed up)
1 2 3 4 5 6 7 8
reading in files with read.delim (install 'readr' package for speed up)
1 2 3 4 5 6 7 8
reading in files with read.delim (install 'readr' package for speed up)
```

```

1 2 3 4 5 6 7 8
Error in .local(object, ...):
  None of the transcripts in the quantification files are present
  in the first column of tx2gene. Check to see that you are using
  the same annotation for both.

Example IDs (file): [NM_001001130.3, NM_001001144.3, NM_001001152.2, ...]
Example IDs (tx2gene): [gene-Bod1-ps, gene-Btf3-ps10, gene-Btf3-ps11, ...]

  This can sometimes (not always) be fixed using 'ignoreTxVersion' or 'ignoreAfterBar'.

Calls: <Anonymous> -> <Anonymous> -> .local
Execution halted

Work dir:
/home/iweber/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/e4/91ebd6e629d26b810b86688181306f

```

The last folder is also where I have a symlink to a tx2gene.tsv file. The symlink points to an address /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/fe/354fd8c110a2cfffb1588d5cf8ce6, where there is indeed a file called "GCF_000001635.27_GRCm39_genomic.tx2gene.tsv" which, indeed, contains again these non-NCBI identifiers ("gene-Bod-etc"). This means the pipeline did not regenerate this file simply because I am now indicating a different Salmon index for the mouse, so this is the file that I will have to regenerate.

Regenerate a correct tx2gene file

The tx2gene file contains information about what transcripts are transcribed from each gene. In essence, it is a small relational table with one column for transcripts and one column for genes, and the information can be extracted from the GFF annotation file associated with most genome assemblies.

To create the files, I used a suite of tools called [gffutils](#). Following the instructions, I installed gffutils in a conda environment with Python 3.8 activated (the latest Python version it works with):

```

conda create -n gffutils python=3.8
conda activate gffutils
cd /home/iweber/Documents/Software
wget https://github.com/fls-bioinformatics-core/GFFUtils/archive/refs/tags/v0.12.0.tar.gz
tar -xvzf v0.12.0.tar.gz

```

This created a folder called "GFFUtils-0.12.0" that contains a whole bunch of scripts. I changed into the directory with `cd GFFUtils-0.12.0/`. The gffutils website says to first install requirements by using pip, so I ran `pip install -r requirements.txt`, which told me "Successfully installed genomics-bcftbx-1.13.1 xlrd-2.0.1 xlsxwriter-3.2.0 xlutils-2.0.0 xlwt-1.3.". I then ran the installation with `pip install GFFUtils` (interestingly, this actually started an installation process, even though it looks like a folder - I presume it has something to do with the `__init__.py` script inside, whose name reminds me of a magic method). I got a message saying "Successfully installed GFFUtils-0.13 argcomplete-3.5.0 argh-0.31.3 importlib-metadata-8.2.0 packaging-24.1 pyfaidx-0.8.1.2 simplejson-3.19.3 zipp-3.20.0".

I created a new directory in my mouse genome folder

(`/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/tx2gene_new/`) and changed to it. In it, I created a script, `create_tx2gene.py` ([inspiration](#)):

```

#!/usr/bin/env python3
import gffutils
import pandas as pd

# Path to GFF file
gff_file =
'/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff'

# Create a gffutils database in memory
db = gffutils.create_db(gff_file, dbfn=':memory:', force=True, keep_order=True, merge_strategy='merge',
sort_attribute_values=True)

# Extract transcript-to-gene mappings
tx2gene = []

for transcript in db.features_of_type('mRNA'): # 'mRNA' corresponds to transcripts in GFF
    gene_id = transcript.attributes.get('Parent')[0] # Assuming 'Parent' contains the gene ID
    transcript_id = transcript.id
    tx2gene.append((transcript_id, gene_id))

# Convert to DataFrame and save to a tsv file

```

```
df = pd.DataFrame(tx2gene, columns=['TXNAME', 'GENEID'])
df.to_csv('tx2gene.tsv', sep='\t', index=False)
```

I wondered if the quant.sf files that tximport will use the tx2gene files for really only contain mRNA identifiers (the NM kind, which I easily saw are present with `head -n 100 quant.sf`, or more. I ran `sudo grep -E '^NR|^XR_ /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnas splice_results/salmon/CONTROL_REPO/quant.sf`, and saw that a whole bunch of other transcript types are present, such as

- NM_ : mRNA (protein-coding transcripts)
- NR_ : Non-coding RNA (often long non-coding RNAs, lncRNAs, or small nuclear RNAs, snRNAs)
- XR_ : Predicted or provisional non-coding RNA
- XM_ : Predicted mRNA (not experimentally confirmed)
- YP_ : Predicted protein (rare in transcript quantification data)
- NP_ : Protein-coding (typically used for proteins, rare in transcript quantification)
- AP_ or WP_ : Protein (alternative locus, rarely used in RNA-seq)
- R_ : Other potential types depending on specific databases or organisms (e.g., ribosomal RNAs might use RR_, but these are typically not included in standard transcriptomes).

And, sure enough, I found lots of NR, XR, XM transcripts. I did not find any of the other types (a call to `sudo grep -E '^YP|^NP|^AP|^WP|^R|^RR_ /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnas splice_results/salmon/CONTROL_REPO/quant.sf` did not return any results, and it did not do so for any of the quant.sf files of the other samples).

I modified the Python script to contain GFF feature names that capture these identifiers:

```
#!/usr/bin/env python3

import gffutils
import pandas as pd

# Specify the path to GFF file
gff_file =
'/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff'

# Create a gffutils database in memory
db = gffutils.create_db(gff_file, dbfn=':memory:', force=True, keep_order=True, merge_strategy='merge',
sort_attribute_values=True)

# Extract transcript-to-gene mappings
tx2gene = []

# Loop through all relevant RNA types based on identifiers found with grep
for transcript in db.features_of_type(['mRNA', 'snRNA', 'lncRNA', 'ncRNA']):
    gene_id = transcript.attributes.get('Parent')[0]
    transcript_id = transcript.id
    tx2gene.append((transcript_id, gene_id))

# Convert to DataFrame and save to a tab-separated value file
df = pd.DataFrame(tx2gene, columns=['TXNAME', 'GENEID'])
df.to_csv('GCF_000001635.27_GRCm39_genomic.tx2gene.tsv', sep='\t', index=False)
```

I chose to name the file exactly as the one I want to replace in the Nextflow work directory. Closed the file and made it executable with `chmod u+x create_tx2gene.py`.

I made a backup copy of the original (misconstructed) tx2gene file:

```
sudo cp /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/fe/354fda8c110a2cfffb1588d5cf8ce6/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/tx2gene_new/GCF_000001635.27_GRCm39_genomic.tx2geneFAULTY.tsv
```

And then I ran the script to create a new tx2gene file with `./create_tx2gene.py`. I checked with `stat GCF_000001635.27_GRCm39_genomic.tx2gene.tsv` when the file was created and finished, and it took around 6.5 hours to create!

The file now looks like this:

GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

File Edit View Insert Format Styles Sheet Data Tools Window Help

Liberation Sans 10 pt B I U T T T A B C D E F G

E9 f Σ =

	A	B	C	D	E	F	G
1	TXNAME	GENEID					
2	rna-XM_006495550.5	gene-Xkr4					
3	rna-NM_001011874.1	gene-Xkr4					
4	rna-XM_011238395.4	gene-Xkr4					
5	rna-XM_036152287.1	gene-Xkr4					
6	rna-XM_030252325.2	gene-Rp1					
7	rna-NM_001370921.1	gene-Rp1					
8	rna-NM_001195662.2	gene-Rp1					
9	rna-NM_011283.3	gene-Rp1					
10	rna-XM_036162991.1	gene-Sox17					
11	rna-NM_001289467.2	gene-Sox17					
12	rna-NM_001289464.2	gene-Sox17					
13	rna-NM_001289466.2	gene-Sox17					
14	rna-NM_001420509.1	gene-Sox17					
15	rna-NM_001289465.2	gene-Sox17					
16	rna-NM_011441.6	gene-Sox17					
17	rna-XM_030252405.1	gene-Sox17					
18	rna-XM_036162992.1	gene-Sox17					
19	rna-NM_001420508.1	gene-Sox17					

File Edit View Insert Format Styles Sheet Data Tools Window Help

Liberation Sans 10 pt B I U T T B U

D98385 f Σ =

	A	B	C	D	E	F	G
98359	ma-XR_004940566.1	gene-Gm24831					
98360	ma-XR_004940562.1	gene-Gm24624					
98361	ma-XR_004940504.1	gene-Gm22139					
98362	ma-XR_004940489.1	gene-Gm25061					
98363	ma-XR_004940474.1	gene-Gm24522					
98364	ma-XR_004940498.1	gene-Gm23124					
98365	ma-XR_004940584.1	gene-Gm22057					
98366	ma-XR_004940496.1	gene-Gm23958					
98367	ma-XR_004940500.1	gene-Gm23199					
98368	ma-XR_004940525.1	gene-Gm25097					
98369	ma-XR_004940481.1	gene-Gm26441					
98370	ma-XR_004940585.1	gene-Gm25793					
98371	ma-XR_004940482.1	gene-Gm22359					
98372	ma-XR_004940471.1	gene-Gm25257					
98373	ma-XR_004940475.1	gene-Gm24268					
98374	ma-XR_004940580.1	gene-Gm57682					
98375	ma-XR_004940582.1	gene-Gm26317					
98376	ma-XR_004940491.1	gene-Gm23794					
98377	ma-XR_004940506.1	gene-Gm23901					
98378	ma-XR_004940472.1	gene-Gm22746					
98379	ma-XR_004940567.1	gene-Gm26007					
98380	ma-XR_004940468.1	gene-Gm22062					
98381	ma-XR_004940605.1	gene-Gm25565					
98382							
98383							
98384							
98385							

It's good, but...the identifiers still don't look like the ones in the quant.sf files. They all have a prefix indicating whether it's a gene or a transcript.

I first modified the TSV file name to "GCF_000001635.27_GRCm39_genomic.tx2gene_regeneration.tsv", and I then fixed the identifiers with a small Python script, "fix_identifiers_tx2gene.py".

```
#!/usr/bin/env python3

import pandas as pd

# Load the tx2gene file
tx2gene = pd.read_csv('GCF_000001635.27_GRCm39_genomic.tx2gene_regeneration.tsv', sep='\t', header=0)

# Remove the 'rna-' prefix from the TXNAME column
tx2gene['TXNAME'] = tx2gene['TXNAME'].str.replace('rna-', '', regex=False)

# Remove the 'gene-' prefix from the GENEID column
tx2gene['GENEID'] = tx2gene['GENEID'].str.replace('gene-', '', regex=False)

# Save the modified tx2gene file
tx2gene.to_csv('GCF_000001635.27_GRCm39_genomic.tx2gene.tsv', sep='\t', index=False)
```

I made the script executable and ran it, which took seconds. Et voila!

GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

A	TXNAME	GENEID
1	XM_006495550.5	Xkr4
2	NM_001011874.1	Xkr4
3	XM_011238395.4	Xkr4
4	XM_036152287.1	Xkr4
5	XM_030252325.2	Rp1
6	NM_001370921.1	Rp1
7	NM_001195662.2	Rp1
8	NM_011283.3	Rp1
9	XM_036162991.1	Sox17
10	NM_001289467.2	Sox17
11	NM_001289464.2	Sox17
12	NM_001289466.2	Sox17
13	NM_001420509.1	Sox17
14	NM_001289465.2	Sox17
15	NM_011441.6	Sox17
16	XM_030252405.1	Sox17
17	XM_036162992.1	Sox17
18	NM_001420508.1	Sox17
19		

Replace reconstructed tx2gene file

Now, to copy this file into the work directories where the still-faulty version is present, I searched again for where all of these files were lurking with
sudo find /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/ -type f -name "*tx2gene*" And found several files:

- 1 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnassplice_results/star_salmon/tximport/tximport.tx2gene.tsv
- 2 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnassplice_results/genome/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
- 3 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnassplice_results_transcriptomic/genome/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
- 4 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/3c/3593b1fa74aefe7ba55a9c5f6549be/tximport.tx2gene.tsv
- 5 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/fe/354fd8c110a2cfffb1588d5cf8ce6/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
- 6 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
- 7 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder_transcriptomic/e5/928599fba977ddf9e27d86a4c83419/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
- 8 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/2a/be42ba8de16fcc5c20c7cbb5075c9/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

Interesting that, now that I searched in the entirety of my project folder, I can see that these .tsv files are present in both the work and the results directories, sometimes with a name like the file I regenerated just now (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv), sometimes as tximport.tx2gene.tsv. I wondered if these two old file types have the exact same contents.

I compared this one

1 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnassplice_results/star_salmon/tximport/tximport.tx2gene.tsv

with this one:

2 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnassplice_results/genome/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

I immediately saw that the tximport.tx2gene.tsv has 141.228 lines (4.2 MB), as opposed to the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv (141.898 lines, 5.2 MB), so both are much larger in comparison to my 2.1 MB regenerated version. I suspect Salmon/tximport created this file including absolutely every kind of transcript, not as narrowly defined as I did when I used gffutils.

Additionally, the files called "GCF_000001635.27_GRCm39_genomic.tx2gene.tsv" are larger at least in part because they contain a third column!

GCF_000001635.27_GRCm39_genomic.tx2gene.tsv (read-only)							
	A	B	C	D	E	F	G
5230	rna-Mirlet7f-2-2	gene-Mirlet7f-2	Mirlet7f-2				
5231	rna-Mirlet7g	gene-Mirlet7g	Mirlet7g				
5232	rna-Mirlet7g-2	gene-Mirlet7g	Mirlet7g				
5233	rna-Mirlet7i	gene-Mirlet7i	Mirlet7i				
5234	rna-Mirlet7i-2	gene-Mirlet7i	Mirlet7i				
5235	rna-ND1	gene-ND1	ND1				
5236	rna-ND2	gene-ND2	ND2				
5237	rna-ND3	gene-ND3	ND3				
5238	rna-ND4	gene-ND4	ND4				
5239	rna-ND4L	gene-ND4L	ND4L				
5240	rna-ND5	gene-ND5	ND5				
5241	rna-ND6	gene-ND6	ND6				
5242	rna-NM_001001130.2	gene-Zfp85	Zfp85				
5243	rna-NM_001001144.2	gene-Scap	Scap				
5244	rna-NM_001001152.2	gene-Zfp458	Zfp458				
5245	rna-NM_001001160.2	gene-Fbxo41	Fbxo41				
5246	rna-NM_001001176.2	gene-Taf9b	Taf9b				
5247	rna-NM_001001177.2	gene-Tsbp1	Tsbp1				
5248	rna-NM_001001178.2	gene-Ccdc148	Ccdc148				
5249	rna-NM_001001179.2	gene-A2ml1	A2ml1				
5250	rna-NM_001001180.2	gene-Zfp941	Zfp941				
5251	rna-NM_001001181.2	gene-BC031181	BC031181				
5252	rna-NM_001001182.2	gene-Baz2b	Baz2b				
5253	rna-NM_001001183.2	gene-Tmem204	Tmem204				
5254	rna-NM_001001184.2	gene-Primpol	Primpol				

I checked what the other files are:

In 3,

/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnassplice_results_transcriptomic/genome/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

there is a 5.2 MB file.

This one

4 /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/3c/3593b1fa74aefe7ba55a9c5f6549be/tximport.tx2gene.tsv is a 4.2 MB file like the other tximport.tx2gene.tsv I saw before.

The same folder contains a symlink to

6 /home/iweber/Documents/Backup_shared_folder/Pre-

eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv, the 5.2 MB file.

Number 5 contains a proper .tsv file (no symlink), 5.2 MB

5 /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/fe/354fd8c110a2cfffb1588d5cf8ce6/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

This one

7 /home/iweber/Documents/Backup_shared_folder/Pre-

eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_transcriptomic/e5/928599fba977ddf9e27d86a4c83419/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

contains a regular .tsv file of 5.2 MB.

Finally, this one

8 /home/iweber/Documents/Backup_shared_folder/Pre-

eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/2a/be42ba8de16fcc5c20c7cb5075c9/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

is 0 bytes, so I presume the process this folder was created for was stopped before the .tsv file was populated.

But the basic question still remains: which file do I need to replace for the pipeline to continue?

To get a better idea of what processes used which files, I looked for the execution_trace txt files. I went to the pipeline_info folder in the output directory of my complete pipeline (so not the separate genomic and transcriptomic branches I ran as well)

(/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplie_results/pipeline_info/).

I peeked into one of them manually and saw mention of "TX2GENE", so I searched through all of the text files in that folder (all are execution traces) to see where it contains mentions of tx2gene.tsv files. I did so with grep -Hn 'TX2GENE' *.txt, and what I got back was:

```
execution_trace_2024-08-14_14-37-07.txt:3:6      2a/be42ba      32450  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)  FAILED  1  
2024-08-14 14:37:49.993 515ms  421ms  -  -  -  -  
  
execution_trace_2024-08-15-17-06.txt:10:10    fe/354fd8c      42739  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)  COMPLETED  
0      2024-08-14 15:17:43.142 11.8s  10s  101.7%  431 MB  449.1 MB  964.5 MB  10 MB  
execution_trace_2024-08-14_15-17-06.txt:12:9      a7/b0f3a3      43907  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)  COMPLETED  
0      2024-08-14 15:17:52.800 11.1s  9s  99.6%  439 MB  457.2 MB  964.5 MB  10 MB  
  
execution_trace_2024-08-15_12-10-30.txt:7:11      fe/354fd8c      42739  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)  CACHED  0  
2024-08-14 15:17:43.142 11.8s  10s  101.7%  431 MB  449.1 MB  964.5 MB  10 MB  
execution_trace_2024-08-15_12-10-30.txt:8:6      a7/b0f3a3      43907  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)  CACHED  0  
2024-08-14 15:17:52.800 11.1s  9s  99.6%  439 MB  457.2 MB  964.5 MB  10 MB  
execution_trace_2024-08-15_12-10-30.txt:66:65     f9/fa69fe      246777  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv)  FAILED  1  
2024-08-15 15:40:36.894 1m 48s  1m 48s  -  -  -  -  
  
execution_trace_2024-08-18_15-22-19.txt:12:11    fe/354fd8c      42739  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)  CACHED  0  
2024-08-14 15:17:43.142 11.8s  10s  101.7%  431 MB  449.1 MB  964.5 MB  10 MB  
execution_trace_2024-08-18_15-22-19.txt:13:9      a7/b0f3a3      43907  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)  CACHED  0  
2024-08-14 15:17:52.800 11.1s  9s  99.6%  439 MB  457.2 MB  964.5 MB  10 MB  
execution_trace_2024-08-18_15-22-19.txt:94:92     3c/3593b1      144000  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv)  COMPLETED  
0      2024-08-18 18:51:08.257 33.2s  32s  97.5%  526.1 MB  1.4 GB  259 MB 162.5 MB  
execution_trace_2024-08-18_15-22-19.txt:95:94     e4/91ebd6      145127  
NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON:TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv)  FAILED  1  
2024-08-18 18:51:41.449 7.2s  7.1s  -  -  -  -
```

I have several execution trace files, with the most recent one for the complete pipeline run being the one from Sunday (execution_trace_2024-08-18_15-22-19). Here, I can see that the pipeline seems to rather use the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file, and not the simple tximport.tx2gene.tsv file I've seen before, and TXIMPORT failed using this particular file (last line in the trace).

It also looks like this tsv file is generated by the process TX2GENE_TXIMPORT_STAR_SALMON:GFFREAD_TX2GENE based on a gtf file (GCF_000001635.27_GRCm39_genomic.gtf). This is quite odd, since I gave the pipeline a GFF file as input, but perhaps it converted it to GTF itself?

And also, if a step, such as this TX2GENE_TXIMPORT_SALMON:GFFREAD_TX2GENE process has results that are cached, can I find and alter those results by going to the work folder called fe/354fda....? (/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/fe/354fda8c110a2cfffb1588d5cf8ce6/) Or would I need to go to the folder from which the GCF file is then used by TXIMPORT, e4/91ebd6...? (This one points to the same work subfolder as above, /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/fe/354fda8c110a2cfffb1588d5cf8ce6/)

In the folder mentioned in the last entry of the trace, /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/e4/91ebd6e629d26b810b86688181306f/ folder, I indeed have yet another symlink to a GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file, namely located in /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/fe/354fda8c110a2cfffb1588d5cf8ce6/ This is the folder mentioned in the second-to-last command, which does contain a proper GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file, and a symlink to a GTF file, GCF_000001635.27_GRCm39_genomic.gtf.

What I make of this is that I'm looking at some error that was likely propagated down different runs of the pipeline. The run from Sunday, with the execution trace I mention, refers to the results of a previous run, stored, in this case, in the "fe" folder. This is where the initial .tsv file was generated from a GTF file that is only referenced here as a symlink. This, in turn, was generated in another process, stored in folder 4f/788081b04c8224dcce0ad7247538b2/. Here, aside from the original GTF file, there is a GFF file, which is present as a symlink to the input GFF file I gave the pipeline, from my "Genomes/genome_M_musculus" folder. I peeked into this GTF file to see if it's the root of all evil, and it doesn't look like it:

```
sudo head -n 100 GCF_000001635.27_GRCm39_genomic.gtf
NC_000067.7      cmsearch      transcript      3172239 3172348 .      +      .      transcript_id "rna-XR_004936710.1";
gene_id "gene-Gm26206"; gene_name "Gm26206"; Dbxref "GeneID:115487594,RFAM:RF00026,GenBank:XR_004936710.1,MGI:MGI:5455983";
Name "XR_004936710.1"; gbkey "ncRNA"; gene "Gm26206"; inference "COORDINATES: profile:INFERNAL:1.1.1"; product "U6
spliceosomal RNA"; transcriptID "XR_004936710.1"; description "predicted gene%2C 26206"; gene_biotype "snRNA";

NC_000067.7      cmsearch      exon      3172239 3172348 .      +      .      transcript_id "rna-XR_004936710.1"; gene_id
"gene-Gm26206"; gene_name "Gm26206"; Dbxref "GeneID:115487594,RFAM:RF00026,GenBank:XR_004936710.1,MGI:MGI:5455983"; gbkey
"ncRNA"; gene "Gm26206"; inference "COORDINATES: profile:INFERNAL:1.1.1"; product "U6 spliceosomal RNA"; transcriptID
"XR_004936710.1";

NC_000067.7      Gnomon      transcript      3269956 3741733 .      -      .      transcript_id "rna-XM_006495550.5"; gene_id
"gene-Xkr4"; gene_name "Xkr4"; Dbxref "GeneID:497097,GenBank:XM_006495550.5,MGI:MGI:3528744"; Name "XM_006495550.5"; gbkey
"mRNA"; gene "Xkr4"; model_evidence "Supporting evidence includes similarity to: 3 mRNAs%2C 3 ESTs%2C 4 Proteins%2C 766 long
SRA reads%2C and 100%25 coverage of the annotated genomic feature by RNAseq alignments%2C including 17 samples with support
for all annotated introns"; product "X-linked Kx blood group related 4%2C transcript variant X1"; transcriptID
"XM_006495550.5"; description "X-linked Kx blood group related 4"; gene_biotype "protein_coding"; gene_synonym
"Gm210,mKIAA1889,XRG4";

NC_000067.7      Gnomon      exon      3269956 3277540 .      -      .      transcript_id "rna-XM_006495550.5"; gene_id "gene-
Xkr4"; gene_name "Xkr4"; Dbxref "GeneID:497097,GenBank:XM_006495550.5,MGI:MGI:3528744"; gbkey "mRNA"; gene "Xkr4"; product "X-
linked Kx blood group related 4%2C transcript variant X1"; transcriptID "XM_006495550.5";
```

The problems seem to have started with the conversion of this GTF file into the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file, located in /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/fe/354fda8c110a2cfffb1588d5cf8ce6/. This tx2gene file has the three-column format with identifiers formatted like:

gene-Bod1-ps	gene-Bod1-ps	Bod1-ps
gene-Btf3-ps10	gene-Btf3-ps10	Btf3-ps10
gene-Btf3-ps11	gene-Btf3-ps11	Btf3-ps11
id-Igkv4-79	gene-Igkv4-79	Igkv4-79
id-Igkv4-80	gene-Igkv4-80	Igkv4-80
rna-Mir30f	gene-Mir30f	Mir30f
rna-Mir31	gene-Mir31	Mir31
rna-NM_001005767.5	gene-Parl	Parl
rna-NM_001005784.1	gene-Patj	Patj
rna-NM_001005787.1	gene-Patj	Patj

This was created last Wednesday, so it must have been right at the start of the pipeline run when I accidentally used the wrong index. Which means it's been cached ever since and the pipeline has been attempting to use this file ever since. Also, considering that it has so many more lines than the file I generated and this particular format, I think I should target this file.

Blessed be those that read, but I also realized that the [pipeline output description page](#) also mentions the tximport.tx2gene.tsv file as being a **filtered** file containing gene to transcripts ids mappings. This would explain why the file I myself generated from the GFF with Gffutils is smaller than the one I just found in the work directory of the pipeline. So is it possible that these smaller tximport.tx2gene.tsv files are built in sequence, based on the initially faulty GCF_000001635.27_GRCm39_genomic.tx2gene.tsv files? I searched my project directory to see where these tximport.tx2gene.tsv pop up with sudo find -name "*tximport**" in my project directory, and found:

```
./Pre_eclampsia_mice_rnaslice_results/star_salmon/tximport/tximport.tx2gene.tsv
./Pre_eclampsia_Nextflow_work_folder/3c/3593b1fa74ae7ba55a9c5f6549be/tximport.tx2gene.tsv
```

I suspect the file in the results folder is the same as the one on the work folder, and it's being saved because I activated options to save intermediates for nearly everything. In the work folder that was returned above, there isn't only the tximport.tx2gene.tsv file, but also a symlink to a GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file, which leads to /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/. This is precisely the third-from-last entry in the execution trace, the one from the process GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) that completed on 2024-08-14 15:17:52.800, so last Wednesday. Which means that it's *this* is the root of all evil *takes out silver weapons*.

Can I force the pipeline to re-do this step that created this tsv file, even though this step is cached? All roads lead to the folder a7/b0f3a354121a73f22dcc404dd0bce3, so I suspect that deleting this will then make quick work of forcing the pipeline to regenerate the file. But is there any warranty that it will do so correctly this time around? Likely not, and I don't know which parameters I have to change for it to stop adding these prefixes to the transcript identifiers. So, what I can do is to delete all of the files that have dependencies to this one. Which ones would those be? ChatGPT suggested to use find with -type l and -lname:

```
sudo find /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/ -type l -lname "/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv"
```

The only result is

```
/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/3c/3593b1fa74aefe7ba55a9c5f6549be/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

What symlinks point to this file?

```
sudo find /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/ -type l -lname "/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/3c/3593b1fa74aefe7ba55a9c5f6549be/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv"
```

Apparently, none other. So, the rabbit hole we're down is:

4f: folder where the pipeline first created the GTF file from the GFF file I provided it with
a7/b0f3a354121a73f22dcc404dd0bce3: folder where the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file was created (also contains symlink to the GTF file the pipeline created)
3c: most recent use of the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file

Which means that... I need to modify the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file in a7 and pray that the symlink holds. I will need to also remove the tximport.tx2gene.tsv file from /Pre_eclampsia_Nextflow_work_folder/3c/3593b1fa74aefe7ba55a9c5f6549be/tximport.tx2gene.tsv because I want the pipeline to re-generate that based on the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file I now generate.

I will make a backup copy of the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv file,:;

```
sudo cp /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27_tx2gene_new/GCF_000001635.27_GRCm39_genomic_FAULTY.tx2gene.tsv
```

To find out exactly which prefixes I have in the huge file, I changed into the a7/b0f3a354121a73f22dcc404dd0bce3 folder and used:

```
sudo grep -o -E '^(|[\\t])[^,;[:space:]]+' GCF_000001635.27_GRCm39_genomic.tx2gene.tsv | awk -F'-' '{print $1}' | tr -d '\t' | sort | uniq
```

This returned a list of identifiers I don't quite believe is right:

```
gene
id
rna
t14
t15
t4
t7a
t7c
t7f
ta3
tap1
tap10
tap11
tap12
```

```
tap13
tap15
tap16
tap17
tap19
tap2
tap20
tap21
tap22
tap24
tap26
tap27
tap28
tap29
tap3
tap31
tap4
tap5
tap6
tap7
tap8
tap9
tf1
tf3
tges3
thl17
tk
tl1
tma
tmn1
tnl5
to4
tp
tp6v0c
ts8
txbp3
```

Searching in the file for, for example, all instances of "tap31-" immediately showed that the regex was still catching parts of gene names that contained a dash for reasons other than having a prefix, e.g. gene-Krtap31-2. So I asked ChatGPT for an updated regex:

```
sudo grep -o -E '^(?![\t])[^\t][[:space:]]+-[^-]*[^0-9]-' GCF_000001635.27_GRCm39_genomic.tx2gene.tsv | awk -F'-' '{print $1}' | tr -d '\t' | sort | uniq
```

and now I only got

```
gene
id
rna
t7a (happens due to some miRNA gene names called "Mirlet7a-1, 2 etc")
t7c
t7f
tk (gene-Stk-ps2)
tma (gene-Ptma-ps2)
tp (still a lot of identifiers containing tp-)
tp6v0c (gene-Atp6v0c-ps2 and Atp6v0c-ps2)
```

The hits that aren't gene-, id-, or rna- all start with a "t," so somehow the regular expression is interpreted as looking for something following the letter t and not the tab character.

The good thing is: I know I only need to worry about gene-, id-, and rna- as prefixes that need to be removed, and I can look specifically for them, not for a regex as I did above. Let's make sure they don't appear anywhere that's *not* at the start of the line:

```
grep -E 'gene-|id-|rna-' GCF_000001635.27_GRCm39_genomic.tx2gene.tsv | grep -v -E '^gene-|id-|rna-'
```

And the command returned nothing, so nope. Anywhere they are, it's at the start of the line.

To remove the gene-, id-, rna- prefixes, I used SED with regular expressions ([tutorial](#), here in more details for [regex with grep](#)):

```
sudo sed -i 's/^gene-|id-|rna-//; s/[\t](gene-|id-|rna-)/\t/g' GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

This replaces any occurrence of the pattern at the start of the line with nothing (hence having just a slash after the slash separating the pattern to be replaced from what to replace it with), and also looks after tab characters and replaces pattern hits with exactly only a tab character. And the option "-i" makes sure it does so in-place, meaning modifying the file without creating a copy, which I want in order to maintain any symlinks and the possibility to resume the pipeline. "-E" enables use of Extended RegEx so that I can have the different variants of the pattern to be replaced.

With a slight tremor, I opened the file and saw:

Bod1-ps	Bod1-ps	Bod1-ps
Btf3-ps10	Btf3-ps10	Btf3-ps10
Btf3-ps11	Btf3-ps11	Btf3-ps11
Btf3-ps3	Btf3-ps3	Btf3-ps3
NM_001014973.3	Snx13	Snx13
NM_001014974.2	Ttl4	Ttl4
NM_001014976.2	Espl1	Espl1

Success!!! And now let's clean up the rest of the mess.

I moved the tximport.tx2gene.tsv file from /Pre_eclamp sia_Nextflow_work_folder/3c/3593b1fa74aefe7ba55a9c5f6549be/tximport.tx2gene.tsv out and into my tx2gene backup directory

```
sudo mv /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/3c/3593b1fa74aefe7ba55a9c5f6549be/tximport.tx2gene.tsv /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/tx2gene_new/FAULTYfrom3c_tximport.tx2gene.tsv
```

However, I realized I had the same genomic.tsv file in other places, too, such as:

/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/fe/354fd8c110a2cfffb1588d5cf8ce6/. Just to be on the safe side, I switched to this folder and ran the same sed command:

```
sudo sed -E -i 's/^gene-[id-|rna-)//; s/[\t](gene-[id-|rna-])/\t/g' GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

which worked as expected.

I though, for good measure, to delete the parent folder of the last genomic.tx2gene.tsv, /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/2a/be42ba8de16fcc5c20c7cbb5075c9/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv, so as to make the pipeline re-do the entire operation that didn't finish and left this file at 0 bytes, but reconsidered and decided against it. Modifying the contents of the other .tsv files already runs a risk of the pipeline not being able to resume. Deleting a full folder like this poses an even higher risk, so I will try to see if it works first.

I really, really wonder, and really, really hope that this will have been the end of the story, but let's run the pipeline first and fix the inevitable mistakes after.

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplice_results \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/samplesheet_preeclamp sia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-resume d2143e7b-2e6a-41e4-850d-cb8af9596f82
```

Started around 16:30.

It finished quantifying with Salmon around 17:20, but then failed again at the tximport step, this time with the error:

```
-[nf-core/rnasplice] Pipeline completed with errors-
```

```
ERROR ~ Error executing process > 'NFCore_RNAsPlice:RNAsPlice:TX2Gene_TxImport_star_Salmon:TxImport  
(GCF_000001635.27_GRCm39_genomic.tx2gene.tsv)'
```

```
Error in .local(object, ...):
```

```
None of the transcripts in the quantification files are present  
in the first column of tx2gene. Check to see that you are using  
the same annotation for both.
```

```
Example IDs (file): [rna-XR_004936710.1, rna-XM_006495550.5, rna-NM_001011874.1, ...]
```

```
Example IDs (tx2gene): [Bod1-ps, Btf3-ps10, Btf3-ps11, ...]
```

```
Work dir:
```

```
/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/18/e6b6c2d71c95100ffda7594b692c83
```

It tells me the problem is in folder 18/e6b6c2d71c95100ffda7594b692c83 .

I looked into the quant.sf files that Salmon just generated and which are symlinked into this folder, and somehow, these quant.sf files now have the indices that the files *didn't* initially have, but which the tx2gene files had. And now of course my tx2gene file

GCF_000001635.27_GRCm39_genomic.tx2gene.tsv in this folder symlinks to the one in the a7 folder, which I had modified to remove the identifiers from (why oh why...).

```
/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/
```

What happens if I restart the pipeline right now? Just to make sure Salmon won't try to requantify the files...No, it fortunately doesn't:

```
Activities Terminal Di Aug 20 17:23

+ iweber@iweber-VirtualBox: ~/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_v

ARN: Killing running tasks (1)ASPLICE:RNAsPlice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:FASTQC (CONTROL REP1) [100%] 8 of 8, cached: 8 ✓
executor > local (1)
4f/788081] process > NFCore_RNAsPlice:RNAsPlice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff) [100%] 1 of 1, cached: 1 ✓
85/e3b1ab] process > NFCore_RNAsPlice:RNAsPlice:PREPARE_GENOME:GTF_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
15/9138d2] process > NFCore_RNAsPlice:RNAsPlice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
c4/ed4701] process > NFCore_RNAsPlice:RNAsPlice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna) [100%] 1 of 1, cached: 1 ✓
6a/018899] process > NFCore_RNAsPlice:RNAsPlice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampnia.csv) [100%] 1 of 1, cached: 1 ✓
3d/4f37d7] process > NFCore_RNAsPlice:RNAsPlice:CONTRASTS_CHECK:CONTRASTSHEET_CHECK (contrastssheet_preeclampnia.csv) [100%] 1 of 1, cached: 1 ✓
-] process > NFCore_RNAsPlice:RNAsPlice:CAT_FASTQ -
b8/f0dc89] process > NFCore_RNAsPlice:RNAsPlice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:FASTQC (CONTROL REP1) [100%] 8 of 8, cached: 8 ✓
7d/87d4af] process > NFCore_RNAsPlice:RNAsPlice:FASTQ_FASTQC_UMITOOLS_TRIMGALORE:TRIMGALORE (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
5a/858213] process > NFCore_RNAsPlice:RNAsPlice:ALIGN_STAR:STAR_ALIGN (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
87/ddff40] process > NFCore_RNAsPlice:RNAsPlice:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS_SORT (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
d2/68be01] process > NFCore_RNAsPlice:RNAsPlice:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:SAMTOOLINDEX (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
-] process > NFCore_RNAsPlice:RNAsPlice:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_STATS [100%] 8 of 8, cached: 8 ✓
8e/8d3cc0] process > NFCore_RNAsPlice:RNAsPlice:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:BAM_STATS_SAMTOOLS:SAMTOOLS_FLAGSTAT (PREECLAMPSIA REP2) [25%] 2 of 8, cached: 2
26/2028c2] process > NFCore_RNAsPlice:RNAsPlice:ALIGN_STAR:BAM_SORT_STATS SAMTOOLS:SAMTOOLS_IDXSTATS (CONTROL REP3) [25%] 2 of 8, cached: 2
a7/435c5b] process > NFCore_RNAsPlice:RNAsPlice:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
c5/4e272d] process > NFCore_RNAsPlice:RNAsPlice:DEXSEQ_DEU:DEXSEQ_COUNT (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
-] process > NFCore_RNAsPlice:RNAsPlice:DEXSEQ_DEU:DEXSEQ_EXON [100%] 0 of 1
b2/338963] process > NFCore_RNAsPlice:RNAsPlice:EDGER_DEU:SUBREAD_FLATTENGTF (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
02/78452c] process > NFCore_RNAsPlice:RNAsPlice:EDGER_DEU:SUBREAD_FEATURECOUNTS (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
bb/c182b1] process > NFCore_RNAsPlice:RNAsPlice:EDGER_DEU:EDGER_EXON (samplesheet_preeclampnia.csv) [100%] 1 of 1, cached: 1 ✓
b9/b1e07d] process > NFCore_RNAsPlice:RNAsPlice:RMATS:CREATE_BAMILIST (1) [100%] 1 of 1, cached: 1 ✓
-] process > NFCore_RNAsPlice:RNAsPlice:RMATS:RMATS_PREP [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:RMATS:RMATS_POST [100%] 0 of 1
87/9c14e3] process > NFCore_RNAsPlice:RNAsPlice:VISUALISE_MISO:GTF_2_GFF3 [100%] 1 of 1, cached: 1 ✓
5e/89ed70] process > NFCore_RNAsPlice:RNAsPlice:VISUALISE_MISO:MISO_INDEX [100%] 1 of 1, cached: 1 ✓
60/0667b3] process > NFCore_RNAsPlice:RNAsPlice:VISUALISE_MISO:MISO_RUN (8) [87%] 7 of 8, cached: 7
-] process > NFCore_RNAsPlice:RNAsPlice:VISUALISE_MISO:MISO_SETTINGS [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:VISUALISE_MISO:MISO_SASHIMI [100%] 0 of 1
99/5490e2] process > NFCore_RNAsPlice:RNAsPlice:Salmon_Quant_Star (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
a7/b0f3a3] process > NFCore_RNAsPlice:RNAsPlice:Tx2Gene_TxImport_Star_Salmon:UNtar [100%] 1 of 1, cached: 1 ✓
-] process > NFCore_RNAsPlice:RNAsPlice:Tx2Gene_TxImport_Star_Salmon:TxImport [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:DrimSeq_Dexseq_Dtu_Star_Salmon:DrimSeq_Filter [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:DrimSeq_Dexseq_Dtu_Star_Salmon:DrimSeq_Dtu [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:DrimSeq_Dexseq_Dtu_Star_Salmon:Stager [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:Split_Files_Tpm [100%] 0 of 1
69/92c17f] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:Generate_Events_Ioe (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
-] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:PsiperEvent [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:Split_Files_Ioe [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:DiffSplice_Ioe [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:ClusterGroups_Ioe [100%] 0 of 1
-] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:ClusterEvents_Ioe [100%] 0 of 1
fc/ffa5ce] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:Generate_Events_Ioi (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
-] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Star_Salmon:PsiperIsoform [100%] 0 of 1
c1/0babac5] process > NFCore_RNAsPlice:RNAsPlice:Salmon_Quant_Salmon (PREECLAMPSIA REP3) [100%] 8 of 8, cached: 8 ✓
fe/354fda] process > NFCore_RNAsPlice:RNAsPlice:Tx2Gene_TxImport_Salmon:Gffread_Tx2Gene (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
0a/e90e48] process > NFCore_RNAsPlice:RNAsPlice:Tx2Gene_TxImport_Salmon:Txiimport (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv) [100%] 0 of 1
8f/025573] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Salmon:Generate_Events_Ioe (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
df/8384fe] process > NFCore_RNAsPlice:RNAsPlice:Suppa_Salmon:Generate_Events_Ioi (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
lus 21 more processes waiting for tasks...
ARN: The operator 'first' is useless when applied to a value channel which returns a single value by definition

Nextflow iweber@iweber-VirtualBox: ~/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder$
```

So I need to fix the GCF_000001635.27_GRCm39_genomic.tx2gene.tsv in

/home/iweber/Documents/Backup_shared_folder/Pre-

eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/ by...replacing its contents with the *original* contents (I can't copy and paste so as to not ruin the symlinks). Apparently, cat does change the contents of files directly, without affecting the so-called inode of the file. I [checked](#) the inode of this file with the "-i" flag of ls:

```
ls -i GCF_000001635.27_GRCm39_genomic.tx2gene.tsv  
90058394 GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

I will first run a test drive with something else to make sure cat works as such on my system as well (yes, I know it's silly, but my anxiety at needing to re-run the entirety of the pipeline is too high).

I went to my scripts folder and made a copy of one of the files there, wrote the contents of another script into it, then tested if it affects its inode:

```
ls -i "trimmomatic_script (copy).sh"  
8965035 'trimmomatic_script (copy).sh'  
cat pigz_several_fastqs.sh > 'trimmomatic_script (copy).sh'  
8965035 'trimmomatic_script (copy).sh'
```

and could confirm the inode was still 89655035 for 'trimmomatic_script (copy).sh'.

Okay, now for my .tsv files. I checked that my file GCF_000001635.27_GRCm39_genomic_FAULTY.tx2gene.tsv still have the prefixes that I now apparently need. Then, I ran the cat command to replace the contents of the file that's causing the issues:

```
sudo cat  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/tx2gene_new/GCF_000001635.27_GRCm39_genomic_FAULTY.tx2gene.tsv > /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/a7/b0f3a354121a73f22dcc404dd0bce3/GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

And I got a "permission denied" error, even with sudo! The file's permissions look like

-rw-r--r-- 1 root root 3956126 Aug 20 15:46 GCF_000001635.27_GRCm39_genomic.tx2gene.tsv , meaning it should work just fine with sudo.

I tried changing the file owner to me for both files:

```
sudo chown iweber:root GCF_000001635.27_GRCm39_genomic.tx2gene.tsv  
sudo chown iweber:root  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/tx2gene_new/GCF_000001635.27_GRCm39_genomic_FAULTY.tx2gene.tsv
```

I'm now the owner of the files:

```
ls -lat  
-rw-r--r-- 1 iweber root 3956126 20. Aug 15:46 GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

I gave myself read and write permissions:

```
sudo chmod 664  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/tx2gene_new/GCF_000001635.27_GRCm39_genomic_FAULTY.tx2gene.tsv
```

..., saw that I now have read and write permissions:

-rw-rw-r-- 1 iweber root 3956126 20. Aug 15:46 GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

and then re-ran the cat command. And it worked, now the file in a7/b0f3a354121a73f22dcc404dd0bce3/ is again 5.2 MB and contains the prefixes!

Undoing the permissions changes:

```
sudo chmod 644 GCF_000001635.27_GRCm39_genomic.tx2gene.tsv sudo chown root:root GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

Initial permissions and ownership is restored:

-rw-r--r-- 1 root root 5234991 20. Aug 18:55 GCF_000001635.27_GRCm39_genomic.tx2gene.tsv

And the file's inode is:

```
ls -i GCF_000001635.27_GRCm39_genomic.tx2gene.tsv  
90058394 GCF_000001635.27_GRCm39_genomic.tx2gene.tsv
```

So I managed to modify it without changing the inode! Let's see what the pipeline has to say to that - re-running with same parameters as above.

It passed the tximport steps, finally!!!

Re-run pipeline after tx2gene fix

It failed, as I expected, at the Sashimi plot generation step, but that's okay, I had simply not changed that before because I was afraid it would prevent the pipeline from resuming before I could see the tximport progress.

```

sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--sashimi_plot false \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/samplesheet_preeclampia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclampia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-resume d2143e7b-2e6a-41e4-850d-cb8af9596f82

```

started around 7 PM.

8 PM:

work folder: 165.7 GB

results folder: 166.1 GB

20.30: results folder 166.9 GB

```

Activities Terminal Di Aug 20 22:58
iweber@iweber-VirtualBox: ~/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_wo.

6a/018809] process > NFCORE_RNASPLICE:RNASPLICE:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampia.csv) [100%] 1 of 1, cached: 1 ✓
3d/4f37d7] process > NFCORE_RNASPLICE:RNASPLICE:CONTRASTS_CHECK:CONTRASTSSHEET_CHECK (contrastssheet_preeclampia.csv) [100%] 1 of 1, cached: 1 ✓
[...] process > NFCORE_RNASPLICE:RNASPLICE:CAT_FASTQ
5b/7caef] process > NFCORE_RNASPLICE:RNASPLICE:FASTQ_FASTQC_UNITOOLS_TRIMGALORE:FASTQC (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
7d/87d40] process > NFCORE_RNASPLICE:RNASPLICE:FASTQ_FASTQC_UNITOOLS_TRIMGALORE:TRIMGALORE (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
5a/858213] process > NFCORE_RNASPLICE:RNASPLICE:ALIGN_STAR:STAR_ALIGN (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
87/dfdf49] process > NFCORE_RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_SORT (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
d2/688e01] process > NFCORE_RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_SAMTOOLS_INDEX (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
19/e509fc] process > NFCORE_RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_BAM_STATS_SAMTOOLS_SAMTOOLS_STATS (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
lc/e47447] process > NFCORE_RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_BAM_STATS_SAMTOOLS_SAMTOOLS_FLAGSTAT (PREECLAMPSIA_REPO4) [100%] 8 of 8, cached: 8 ✓
7d/228a60] process > NFCORE_RNASPLICE:RNASPLICE:ALIGN_STAR:BAM_SORT_STATS_SAMTOOLS_BAM_STATS_SAMTOOLS_SAMTOOLS_IDXSTATS (PREECLAMPSIA_REPO4) [100%] 8 of 8, cached: 8 ✓
a7/435c5b] process > NFCORE_RNASPLICE:RNASPLICE:DEXSEQ_DEU:DEXSEQ_ANNOTATION (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
5/4e272d] process > NFCORE_RNASPLICE:RNASPLICE:DEXSEQ_DEU:DEXSEQ_COUNT (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
36/8f03be] process > NFCORE_RNASPLICE:RNASPLICE:DEXSEQ_DEU:DEXSEQ_EXON (1) [100%] 1 of 1 ✓
b2/338963] process > NFCORE_RNASPLICE:RNASPLICE:EDGER_DEU:SUBREAD_FLATTENGT (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
0/78452c] process > NFCORE_RNASPLICE:RNASPLICE:EDGER_DEU:SUBREAD_FEATURECOUNTS (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
bb/c182b1] process > NFCORE_RNASPLICE:RNASPLICE:EDGER_DEU:EDGER_EXON (samplesheet_preeclampia.csv) [100%] 1 of 1, cached: 1 ✓
b9/b1e07d] process > NFCORE_RNASPLICE:RNASPLICE:RMATS:CREATE_BAMLIST (1) [100%] 1 of 1, cached: 1 ✓
88/964859] process > NFCORE_RNASPLICE:RNASPLICE:RMATS_RMATS_PREP (PREECLAMPSIA-CONTROL) [100%] 1 of 1 ✓
[...] process > NFCORE_RNASPLICE:RNASPLICE:RMATS_RMATS_POST
2a/1984a7] process > NFCORE_RNASPLICE:RNASPLICE:RMATS_RMATS_SALMON_QUANT_STAR (PREECLAMPSIA_REPO4) [100%] 8 of 8, cached: 8 ✓
[...] process > NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON_UNTAR
a7/b0f3a3] process > NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON_GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
4b/be2073] process > NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_STAR_SALMON_GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv) [100%] 1 of 1, cached: 1 ✓
98/f90cc8] process > NFCORE_RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON_DRIMSEQ_FILTER (1) [100%] 1 of 1 ✓
[...] process > NFCORE_RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON_DRIMSEQ_DTU
[...] process > NFCORE_RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_STAR_SALMON_DRIMSEQ_DTU_STAR_SALMON_STAGER
b0/f0e975] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_SPLIT_FILES_TPM (suppa_tpm.txt) [100%] 1 of 1 ✓
69/92c17f] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
94/f58eb1] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_PSIPIREVENT (suppa_tpm.txt) [100%] 1 of 1 ✓
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_SPLIT_FILES_IOE
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_DIFFSPICE_IOE
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_CLUSTERGROUPS_IOE
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_CLUSTEREVENTS_IOE
fc/ffa5ce] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
2e/e85475] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_PSIPIROIFORM (suppa_tpm.txt) [100%] 1 of 1 ✓
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_SPLIT_FILES_IOI
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_DIFFSPICE_IOI
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_CLUSTERGROUPS_IOI
[...] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_STAR_SALMON_CLUSTEREVENTS_IOI
c1/0abac5] process > NFCORE_RNASPLICE:RNASPLICE:SALMON_QUANT_SALMON (PREECLAMPSIA_REPO3) [100%] 8 of 8, cached: 8 ✓
fe/354fda] process > NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_SALMON_GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
a6/26f0de] process > NFCORE_RNASPLICE:RNASPLICE:TX2GENE_TXIMPORT_SALMON_TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv) [100%] 1 of 1, cached: 1 ✓
0f/e0dc18] process > NFCORE_RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_SALMON_DRIMSEQ_FILTER (1) [100%] 1 of 1 ✓
[...] process > NFCORE_RNASPLICE:RNASPLICE:DRIMSEQ_DEXSEQ_DTU_SALMON_DRIMSEQ_DTU
b1/401dd8] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_SPLIT_FILES_TPM (suppa_tpm.txt) [100%] 1 of 1 ✓
6f/025373] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
a3/25537d] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_PSIPIREVENT (suppa_tpm.txt) [100%] 1 of 1 ✓
5c/3033d5] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_SPLIT_FILES_IOE (suppa_local.psi) [100%] 1 of 1 ✓
3b/4ceb9] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_DIFFSPICE_IOE (PREECLAMPSIA-CONTROL) [100%] 1 of 1, failed: 1 ✘
df/8384fe] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf) [100%] 1 of 1, cached: 1 ✓
1c/c65274] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_PSIPIROIFORM (suppa_tpm.txt) [100%] 1 of 1 ✓
d0/285534] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_SPLIT_FILES_IOI (suppa_isoform.psi) [100%] 1 of 1 ✓
70/2bt3a9] process > NFCORE_RNASPLICE:RNASPLICE:SUPPA_SALMON_DIFFSPICE_IOI (PREECLAMPSIA-CONTROL) [100%] 1 of 1, failed: 1 ✘
[... 8 more processes waiting] ... Fin Visual Studio Code ... exit

```

State at crash

Pipeline crashed with an error from the SUPPA Python script:

ERROR ~ Error executing process > 'NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:DIFFSplice_IOI (PREECLAMPSIA-CONTROL)'

Caused by:

Process `NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:DIFFSplice_IOI (PREECLAMPSIA-CONTROL)` terminated with an error **exit** status (1)

Command executed:

```
suppa.py \
    diffSplice \
    -m empirical \
    -gc -pa -s -c \
    -a 1000 \
    -l 0 \
    -al 0.05 \
    -th 0 \
    -nan 0 \
    -i events.ioi \
    -p transcript_PREECLAMPSIA.psi transcript_CONTROL.psi \
    -e PREECLAMPSIA.tpm CONTROL.tpm \
    -o PREECLAMPSIA-CONTROL_transcript_diffsplice

cat <<-END_VERSIONS > versions.yml
"NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:DIFFSplice_IOI":
    suppa: ${python -c "import pkg_resources; print(pkg_resources.get_distribution('suppa').version)"}
END_VERSIONS
```

Command **exit** status:

1

Command output:

```
Calculating differential analysis between conditions: transcript_PREECLAMPSIA and transcript_CONTROL
Saving between_conditions_avglogtpm...
Saved PREECLAMPSIA-CONTROL_transcript_diffsplice_avglogtpm.tab
```

Command error:

```
ERROR:__main__:Unknown error: (<class 'UnboundLocalError'>, UnboundLocalError("local variable 'i' referenced before assignment"), <traceback object at 0x7f8aa5b52d88>
Calculating differential analysis between conditions: transcript_PREECLAMPSIA and transcript_CONTROL
Saving between_conditions_avglogtpm...
Saved PREECLAMPSIA-CONTROL_transcript_diffsplice_avglogtpm.tab
```

Work dir:

```
/home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder/70/2bf3a959d4216c3e18ced0c84206f9
```

Potential problem with NAs in files? Or some other formatting issues? See <https://github.com/comprna/SUPPA/issues/140>

Fix SUPPA issues

The error:

Command executed:

```
suppa.py \
    diffSplice \
    -m empirical \
    -gc -pa -s -c \
    -a 1000 \
    -l 0 \
    -al 0.05 \
    -th 0 \
    -nan 0 \
    -i events.ioi \
    -p transcript_PREECLAMPSIA.psi transcript_CONTROL.psi \
    -e PREECLAMPSIA.tpm CONTROL.tpm \
    -o PREECLAMPSIA-CONTROL_transcript_diffsplice
```

seems to stem from a part of the SUPPA script called diffSplice. The processes that failed are called "DIFFSplice_IOE" and "DIFFSplice_IOI"

I checked [the SUPPA tutorial](#) to better understand what all of the options do and how files such as the "events.ioi" file are supposed to look like. In principle, it works on the basis of the transcript quantification completed in a previous step of the pipeline by Salmon. It uses a script,

multipleFieldSelection.py , to extract the TPMs of transcripts from the Salmon quant.sf files and write them to an "iso_tpm.txt" file, and then uses the GTF annotation file to calculate the abundance of splicing events (so of course the IDs have to match).

I checked the quant.sf file from /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results/star_salmon/salmon/CONTROL_REP1/:

```
Name Length EffectiveLength TPM NumReads
rna-XR_004936710.1 110 16.709 0.000000 0.000
rna-XM_006495550.5 12274 11879.509 23.364650 2076.226
rna-NM_001011874.1 3634 3239.509 0.000000 0.000
rna-XR_004935518.1 10074 9679.509 0.000000 0.000
rna-XM_011238395.4 2104 1709.509 0.952673 12.182
rna-XM_036152287.1 2764 2369.509 0.484718 8.591
rna-XR_004935517.1 24629 24234.509 0.011033 2.000
gene-Gm57489 969 574.710 0.000000 0.000
gene-Gm27396 58 11.497 0.000000 0.000
rna-XM_030252325.2 3710 3315.509 0.000000 0.000
```

It looks, in my beginner's eyes, [as I'd expect it to](#). How do the other ones look like? I suspect they'll all symlink to the same files in the results folder.

```
/ab/c61dc5c2641fa81cdf6f92cd0d8bd1/CONTROL_REP1/quant.sf - proper file. PREFIXES.
/ed/2dde806049b50d88f814ac5e86c19c/CONTROL_REP1/quant.sf - proper file. no prefixes.
/eb/f89d2deaafc4be440ddb966ba6523e/CONTROL_REP1/quant.sf - proper file. no prefixes.
/0d/947583d9f0b8520c8c05bbef3862e9/CONTROL_REP1/quant.sf - proper file, ENSEMBL IDs!!!
```

```
/9b/db66786facbbb6da1b9cf03e513b51/CONTROL_REP2/quant.sf
/94/308f503d9805bc2550cb967da885/CONTROL_REP2/quant.sf
/ba/4c947178f55e00ad8587e0aefbe7bf/CONTROL_REP2/quant.sf
```

```
/67/63a35fc7486b580597422d6c1f9941/CONTROL_REP3/quant.sf
/9c/00b355eb2fb1502013c25167b7e1ec/CONTROL_REP3/quant.sf
/c0/27d16ed774cb1ce2260ed19b38cbd6/CONTROL_REP3/quant.sf
/16/9d22682e0c70d983d6fa6b2190fa5d/CONTROL_REP3/quant.sf
```

```
/11/a5ffd28abb00b85ee642b57a88226c/CONTROL_REP4/quant.sf
/7a/e172c9b55d3c5eeeca0be1623d578c6/CONTROL_REP4/quant.sf
/89/a17e8c1dee5f264ab0f1883cbe3c1a/CONTROL_REP4/quant.sf
/86/8d2678d785ab216e3cc81eaebab5b6/CONTROL_REP4/quant.sf
```

```
/af/285ff63164456209bb1a0025119ec1/PREECLAMPSIA_REP1/quant.sf
/c1/5ce88b2fb4e0d6396e0356f61d18e1/PREECLAMPSIA_REP1/quant.sf
/47/8761a0dc2b9c1fa8d98bff4554baa9/PREECLAMPSIA_REP1/quant.sf
/c6/de63fccef598cea3cb78aa5db47188/PREECLAMPSIA_REP1/quant.sf
```

```
/3b/7c98bef7fe6d32e7224ab6579a603d/PREECLAMPSIA_REP2/quant.sf
/ee/2d458a66b1c9c884fcc483f1cf3a9d/PREECLAMPSIA_REP2/quant.sf
/ae/08fad176e185b39a563143e9b113e2/PREECLAMPSIA_REP2/quant.sf
/31/4c3480a59ac76de6e9dd7089ddffa1/PREECLAMPSIA_REP2/quant.sf
```

```
/81/6919e25d0e14867f32f8d5badbb744/PREECLAMPSIA_REP3/quant.sf
/99/5490e2e6cbf7c23e2f0548a2468829/PREECLAMPSIA_REP3/quant.sf
/bd/075e2e22cbe4725b8c874351101707/PREECLAMPSIA_REP3/quant.sf
/c1/0abac52c6dbe552a4c069b48d10fba/PREECLAMPSIA_REP3/quant.sf
```

```
/e0/8ad9657cccd23e71b5ce33adae4bb8d/PREECLAMPSIA_REP4/quant.sf
/2a/1984a7b79b83028bc746e5ecff4639/PREECLAMPSIA_REP4/quant.sf
/6d/73353a23db5e4d7cb7d9e2c92a3d65/PREECLAMPSIA_REP4/quant.sf
/d6/2b2e91c96f22feb414198b7404c536/PREECLAMPSIA_REP4/quant.sf
```

I have two "suppa" folders in my results, once in the "salmon" folder (263.7 MB) and once in the "star_salmon" folder (263.2 MB). They each consist of:

```
generate_events
    per_isofrom
        events.ioi
    per_local_event
        events.ioe
            GTF and .ioe files for each type of splicing event      (A3, A5, AL, MX, RI, SE)
psi_per_isofrom
    suppa_isofrom.psi
psi_per_local_event
    suppa_local.psi
```

```

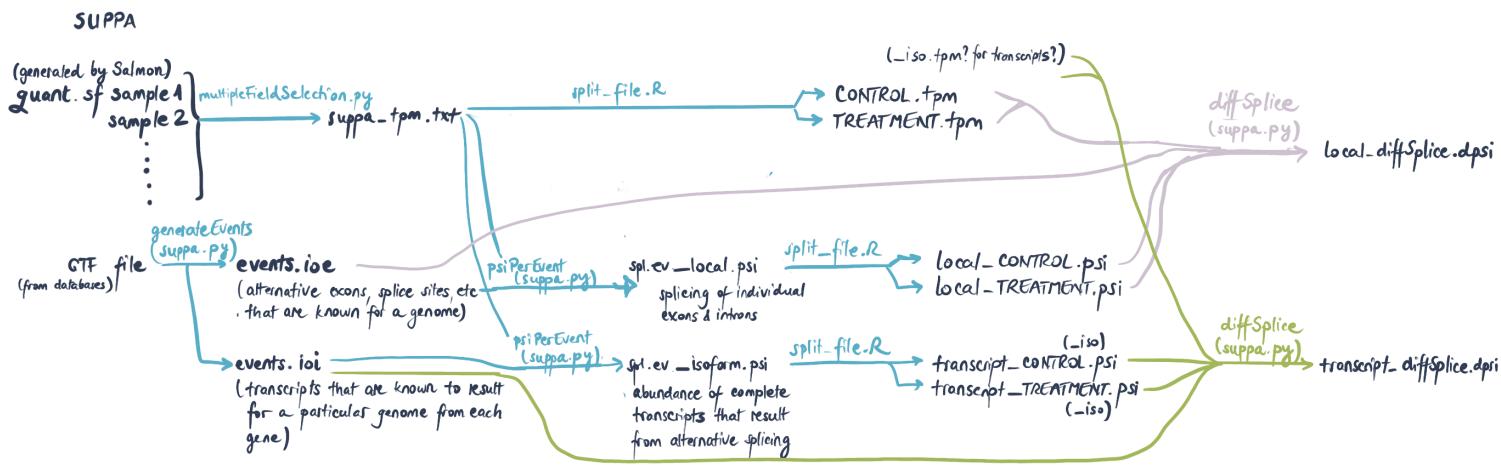
split_files
  per_isoform
    transcript_CONTROL.psi
    transcript_PREECLAMPSIA.psi
  per_local_event
    local_CONTROL.psi
    local_PREECLAMPSIA.psi
  tpms
    CONTROL.tpm
    PREECLAMPSIA.tpm

```

It looks like the two folders in salmon and salmon_star were created at different times (salmon_star later) and are otherwise mostly identical. The only differences (salmon left, star_salmon right):

Drag & drop		Drag & drop			
	Relative path		Relative path		
	Size		Size		
1	psi_per_isoform	suppa_isoform.psi	psi_per_isoform	suppa_isoform.psi	13.814.416
2	psi_per_local_event	suppa_local.psi	psi_per_local_event	suppa_local.psi	17.418.867
3	per_isoform	transcript_CONTROL.psi	per_isoform		
4	split_files/	transcript_PREECLAMP...	split_files/		
5	per_local_event	local_CONTROL.psi	per_local_event		
6	split_files/	local_PREECLAMPSI...	split_files/		
7	split_files/tpms	CONTROL.tpm	tpms	CONTROL.tpm	5.108.786
8		PREECLAMPSIA.tpm		PREECLAMPSIA.tpm	5.114.462

This is what the pipeline does (from the [tutorial](#)):



The files that diffSplice uses (as per [dedicated tutorial section](#)) are the .ioe/.ioi, the .psi, and the .tpm files.

The IOE files contain information taken from the GTF annotation file about where exons are located in the genome, and what transcripts and genes they belong to. I immediately saw something I will need to check in the options the pipeline ran with:

Check!!!!

IMPORTANT: On this occasion, we have used Ensembl. If the user decides to use RefSeq annotation, we suggest using the =====pooled-genes*** option==. RefSeq genes are labeled according to the mRNA sequences they contain, not by the gene locus. This could cause that the same id is mapped to two different places in the genome. Also, sometimes two overlapping isoforms that share some exon or splice-sites are labeled as different genes. The pooled-genes option redefines a gene as a locus with transcripts on the same strand that overlap each other in genome extent and share at least one splice-site pairwise. These issues are a big deal to calculate events and estimate PSIs, since you would be assigning wrong contributions to the events because those transcripts should be considered together as part of the same gene.

Luckily, the [parameters page](#) of the pipeline says that this parameter, `--generateevents_pool_genes`, is set to true by default (yup, I should've checked that more carefully beforehand.)

tpm files

The iso TPM example file from the SUPPA developers looks like this:

iso_tpm_formatted.txt - Notepad

File Edit Format View Help

SRR1513329	SRR1513330	SRR1513331	SRR1513332	SRR1513333	SRR1513334
ENST00000589600	9.23266	7.71646	9.76008	5.1669	5.04695 5.48603
ENST00000414692	0.0989011		0.160883	0.239351	0.293452
ENST00000527740	0	0	0	0	0.184978
ENST00000439648	0	0	0	0	0.228384
ENST00000354495	0.0422224		0.185901	0	1.04344e-10

In the rnaslice pipeline, the file that's generated is called suppa_tpm.txt instead. The one from the folder a6/26f0de2c425f9543d3a9feb3819fd1/ looks similar to the one the developers provide:

suppa_tpm.txt [Read-Only]								
1	CONTROL_REPO1	CONTROL_REPO2	CONTROL_REPO3	CONTROL_REPO4	PREECLAMPSIA_REPO1	PREECLAMPSIA_REPO2	PREECLAMPSIA_REPO3	
2	NM_001001130.3	8.845698	7.142599	8.902068	7.902335	9.010423	7.570207	8.089645 10.12241
3	NM_001001144.3	49.205319	49.010801	49.164584	50.909595	49.190755	46.899712	48.618636 57.294214
4	NM_001001152.2	0	0	0	0	0		
5	NM_001001160.3	7.703279	7.419889	4.50712 9.406561	11.204657	12.601548	14.875179	9.976357
6	NM_001001176.2	0	1.382469	1.417861	0	0	1.494939	1.805488
7	NM_001001177.2	0	0	0.488914	0	0	0	
8	NM_001001178.2	0	2.544779	1.671167	0.219691	0.567723	0	0
9	NM_001001179.3	0.161254	0.145942	0.065627	0.052854	0.108751	0.377394	0.16572 0.164114
10	NM_001001180.2	6.380829	6.092594	4.779937	4.523753	8.283101	10.24307	8.761333 6.42877
11	NM_001001181.4	156.536587	131.325696	141.414752	143.949427	155.466113	133.751355	143.064411 159.761412
12	NM_001001182.3	4.12449	2.26014	5.800339	5.542344	2.026359	4.613256	3.028501 4.927015
13	NM_001001183.1	5.918164	3.783734	3.663662	2.498502	7.028632	4.813887	5.660751 5.014812
14	NM_001001184.2	4.941395	2.696974	2.763057	5.081084	0	3.990487	3.770843 3.514689
15	NM_001001185.3	0	0	0.627949	0	0	0.602629	0

This one doesn't have the prefixes to the identifiers, as opposed to the salmon quant.sf file I just opened. But do none of them have them?

events.ioe files

This is how the events.ioe file looks like from the salmon directory:

events.ioe [Read-Only]								
Open	Save							
1	seqname	gene_id	event_id	alternative_transcripts	total_transcripts			
2	NC_000067.7	gene-Tceal1	A3:NC_000067.7:4957054-4959680:4957054-4959683:+	rna-NM_011541.4,rna-NM_001159751.1	rna-NM_011541.4,rna-NM_001159751.1,rna-NM_001204371.1,rna-NM_001318735.1			
3	NC_000067.7	gene-Oprkl1	gene-Oprkl1;A3:NC_000067.7:5658886-5659228:5658886-5659258:+	rna-NM_001204371.1,rna-NM_001318735.1	rna-NM_011011.2,rna-NM_001204371.1,rna-XR_004934097.1,rna-XM_006495441.5,rna-XM_036153075.1			
4	NC_000067.7	gene-Prex2	gene-Prex2;A3:NC_000067.7:11252258-11254613:11252258-11254670:+	rna-XR_004934097.1,rna-XM_006495441.5,rna-XM_036153075.1	rna-XR_373211.2,rna-XR_373210.1,rna-XR_373212.1,rna-XR_373211.2			
5	NC_000067.7	gene-Gm31195	gene-Gm31195;A3:NC_000067.7:13951612-13952027:13951612-13952047:+	rna-XR_373211.2,rna-XR_373210.1,rna-XR_373212.1,rna-XR_373211.2	rna-XR_004935543.1,rna-XR_003947743.2			
6	NC_000067.7	gene-Gm9947	gene-Gm9947;A3:NC_000067.7:14826461-14845646:14826461-14845750:+	rna-XR_004935542.1,rna-XR_004941784.1	rna-XR_865195.3,rna-XR_004935542.1			
7	NC_000067.7	gene-Gm9947	gene-Gm9947;A3:NC_000067.7:14844161-14845646:14844161-14845750:+	rna-XR_004935542.1,rna-XR_004941784.1	rna-XR_004941784.1,rna-NM_001420525.1			
8	NC_000067.7	gene-Terf1	gene-Terf1;A3:NC_000067.7:15883386-15889158:15883386-15889875:+	rna-XR_001778825.3,rna-XR_001778825.3	rna-XR_001778825.3,rna-XR_865202.3			
9	NC_000067.7	gene-Gm7568	gene-Gm7568;A3:NC_000067.7:16324269-16333232:16324269-16333389:+	rna-XR_026392.1,rna-NM_026392.1	rna-NM_026392.1,rna-NM_027415.2			
10	NC_000067.7	gene-Tmem70	gene-Tmem70;A3:NC_000067.7:16735729-16737889:16735729-16737892:+	rna-NM_016923.2,rna-NM_001159711.1,rna-NM_016923.2	rna-NM_001159711.1,rna-NM_016923.2			
11	NC_000067.7	gene-Ly96	gene-Ly96;A3:NC_000067.7:16761997-16771019:16761997-16771073:+	rna-XM_030252837.1,rna-XM_006495495.3	rna-XM_030252837.1,rna-XM_006495495.3			
12	NC_000067.7	gene-Tfap2b	gene-Tfap2b;A3:NC_000067.7:19304530-19305588:19304530-19306832:+	rna-XR_003946579.2,rna-XR_003946579.2	rna-XR_003946579.2,rna-XR_003946579.2,rna-NM_027974.1			
13	NC_000067.7	gene-Efhc1	gene-Efhc1;A3:NC_000067.7:21049080-21049560:21049080-21049634:+	rna-XM_006495579.3,rna-NM_001290679.1	rna-XM_006495579.3,rna-NM_001290679.1			
14	NC_000067.7	gene-Tmem14a	gene-Tmem14a;A3:NC_000067.7:212996903-21299632:212996903-21299815:+					

There's the name of the chromosome in the first column, the identifier of the gene, an ID for the splicing event itself (containing one of the codes A3, A5, MX, AL, AF, SE, etc for the type of splicing event), which includes two pairs of genomic coordinates (I'm guessing which part of the "genome" is included based on the event happening or not). Finally, there are two more columns: one called "alternative_transcripts" (??containing possible transcripts resulting from this event?), and a fourth one called "total_transcripts" (??).

Again, to my untrained eye, this looks okay.

I ran `sudo find /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/ -name "events.ioe"` to find all of the files.

3b/4eebe93ef7279d6eed34bdf001da0e/events.ioe - symlink to 8f/02557324e537987e9d004

94/f58eb18a126bfe5fb1f4f039b04e0/events.ioe - symlink to 69/92c17f8bb1c99bb34d143392

a3/25537db88f97da97a364ac984e73b5/events.ioe - symlink to 8f/02557324e537987e9d0048b9

1b/e0d509741ed62da01ca01becfdac10/events.ioe - proper file. no prefixes.

69/92c17f8bb1c99bb34d143392f0a3a/events.ioe - proper file. PREFIXES.

f7/39b5eeca178b9e5e88d3c979671f9/events.ioe - proper file. no prefixes.

8f/02557324e537987e9d0048b909a142/events.ioe - proper file. PREFIXES.

events.ioi files

I ran `sudo find /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/ -name "events.ioi"`, and found four in the work folders (which makes sense retrospectively, since I had two completed pipeline runs):

63/a29e5ff0d741615e2f2b88816b8284/events.ioi - no prefixes.

df/8384fea229fab3ca6f03b6283d52b7/events.ioi - PREFIXES

fc/ffa5ce1850fc6f3d5ed20a545f82e6/events.ioi - PREFIXES

70/2bf3a959d4216e3e18ced0e84206f9/events.ioi - this one's the only symlink, and it links to the df/8384fea229fab3ca6f03b6283d52b7 folder.

PSI files

From either the .ioe or the .ioi coordinates plus the tpm file (suppa_tpm.txt in the case of the pipeline), SUPPA generates the PSI files, which show how often an event happens ("Percent Spliced In" is the usual metric for quantifying changes in alternative splicing. Easiest to understand for an exon - in what percentage of all transcripts is it included?).

The PSI file should look like

```
Event_ID      SRR1513329      SRR1513330      SRR1513331      SRR1513332      SRR1513333      SRR1513334  
ENSG00000149554;SE:chr11:125496728-125497502:125497725-125499127:+      0.7471194793      0.7892328982      0.781078454  
0.4554597402    0.4493233894    0.4237912839
```

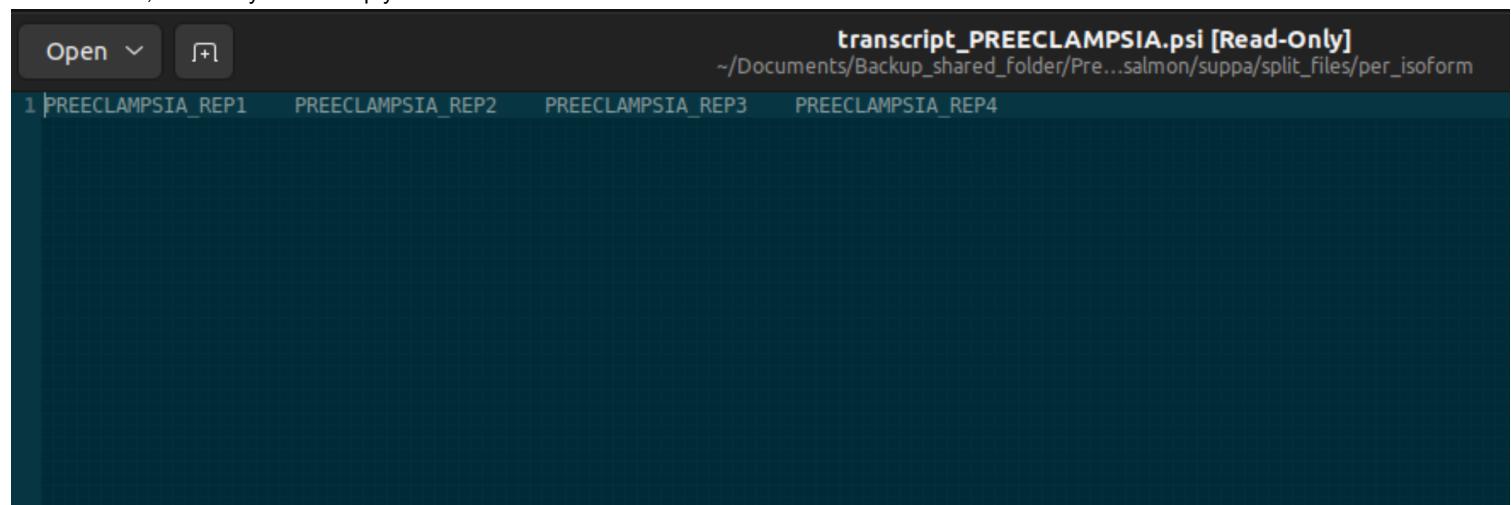
It looks like the Salmon part of the pipeline (salmon folder) managed to obtain the PSI files "per local event" (split_files -> per_local_event) but not the ones "per isoform" (split_files -> per_isoform).

PSI files in salmon folder:

Name	Size	Location
suppa_local.psi	11,9 MB	suppa/psi_per_local_event
suppa_isoform.psi	124 bytes	suppa/psi_per_isoform
local_CONTROL.psi	10,5 MB	suppa/split_files/per_local_event
transcript_CONTROL.psi	52 bytes	suppa/split_files/per_isoform
local_PREECLAMPSIA.psi	10,5 MB	suppa/split_files/per_local_event
transcript_PREECLAMPSIA.psi	72 bytes	suppa/split_files/per_isoform

suppa_local.psi, local_CONTROL.psi, and local_PREECLAMPSIA.psi are all rather large, around 11 MB, whereas suppa_isoform.psi, transcript_CONTROL.psi, and transcript_PREECLAMPSIA.psi are all tiny.

Small wonder, since they are all empty:



The tutorial says that the suppa_local.psi and suppa_isoform.psi files get split into the different conditions in order to let the diffSplice script quantify the differences between treated-untreated. So generating the suppa_local.psi file apparently worked, but not generating the suppa_isoform.psi, and that's where at least some of the problems stem from.

How does the suppa_local.psi from the salmon folder look like?

suppa_local.psi [Read-Only]									
~/Documents/Backup_shared_folder/Pre-eclampsia_results/salmon/suppa/psi_per_local_event									
1	CONTROL REP1	CONTROL REP2	CONTROL REP3	CONTROL REP4	PREECLAMPSIA REP1	PREECLAMPSIA REP2	PREECLAMPSIA REP3	PREECLAMPSIA REP4	
2	gene-0610005C13Rik;SE:NC_000073.7:45220416-45223837:-				NA	NA	NA	NA	NA
3	gene-0610006L08Rik;SE:NC_000073.7:74469101-74469451:74469552-74474468:-				NA	NA	NA	NA	NA
4	gene-0610006L08Rik;SE:NC_000073.7:74474592-74502200:74502294-74503295:-				NA	NA	NA	NA	NA
5	gene-0610010K14Rik;SE:NC_000077.7:70126294-70126290-70126522:-				NA	NA	NA	NA	NA
6	gene-0610010K14Rik;A3:NC_000077.7:70126294-70126823:70126820-70126823:-				NA	NA	NA	NA	NA
7	gene-0610010K14Rik;A5:NC_000077.7:70126294-70126823:70126294-70126850:-				NA	NA	NA	NA	NA
8	gene-0610010K14Rik;A5:NC_000077.7:70126626-70126823:70126850:-				NA	NA	NA	NA	NA
9	gene-0610010K14Rik;SE:NC_000077.7:70126290-70126522:70126626-70126823:-				NA	NA	NA	NA	NA
10	gene-0610010K14Rik;SE:NC_000077.7:70126294-70126522:70126626-70126823:-				NA	NA	NA	NA	NA
11	gene-0610010K14Rik;SE:NC_000077.7:70126294-70126522:70126626-70126850:-				NA	NA	NA	NA	NA
12	gene-0610010K14Rik;SE:NC_000077.7:70127037-70127703:70127804-70128357:-				NA	NA	NA	NA	NA
13	gene-0610025J13Rik;AF:NC_000070.7:98209593:98210546:98210196:98210410-98210546:+				NA	NA	NA	NA	NA
14	gene-061040F04Rik;AF:NC_000072.7:108590954-108600002:108600128:1086000002-1086000002:1086000002-1086000002:+				NA	NA	NA	NA	NA
15	gene-061040J01Rik;AF:NC_000071.7:63969833:63969938-64053773:64034285-64034930-64053773:+				NA	NA	NA	NA	NA
16	gene-061040J01Rik;SE:NC_000071.7:63969938-64033793:64033842-64053773:+				NA	NA	NA	NA	NA
17	gene-061040J01Rik;SE:NC_000071.7:64033842-64041300:64041420-64045388:+				NA	NA	NA	NA	NA
18	gene-061040J01Rik;SE:NC_000071.7:64033842-64045388:64045529-64053773:+				NA	NA	NA	NA	NA
19	gene-1110002E22Rik;AF:NC_000069.7:137768010:137768059-137770782:137769871:137769891-137770782:+				NA	NA	NA	NA	NA
20	gene-1110002307Rik;A3:NC_000076.7:7:66754774-66755949:66754168-66755949:-				NA	NA	NA	NA	NA
21	gene-1110004F10Rik;A3:NC_000073.7:115692760-115698675:115692760-115698686:+				NA	NA	NA	NA	NA
22	gene-1110004F10Rik;A3:NC_000073.7:115701239-115702430:115701239-115702446:+				NA	NA	NA	NA	NA
23	gene-1110004F10Rik;A3:NC_000073.7:115701239-115702430:115701239-115702452:+				NA	NA	NA	NA	NA
24	gene-1110004F10Rik;A3:NC_000073.7:115701239-115702446:115701239-115702452:+				NA	NA	NA	NA	NA
25	gene-1110004F10Rik;A5:NC_000073.7:115692760-115696997:115692622-115696997:+				NA	NA	NA	NA	NA
26	gene-1110004F10Rik;A5:NC_000073.7:115692760-115698675:115692622-115698675:+				NA	NA	NA	NA	NA
27	gene-1110004F10Rik;SE:NC_000073.7:115692762-115696997:115697188-115698675:+				NA	NA	NA	NA	NA
28	gene-1110004F10Rik;SE:NC_000073.7:115692760-115696997:115697188-115698675:+				NA	NA	NA	NA	NA
29	gene-1110018N20Rik;AF:NC_000068.8:167030977:167031326-167033750:1670333101:167033335-167033750:+				NA	NA	NA	NA	NA
30	gene-1110018N20Rik;SE:NC_000068.8:167031326-1670333212:167033335-167033750:+				NA	NA	NA	NA	NA
31	gene-1110018N20Rik;SE:NC_000068.8:167031639-167033242:167032498-167033212:+				NA	NA	NA	NA	NA
32	gene-1110038B12Rik_and_gene-Snord52_and_gene-Gm25744;SE:NC_000083.7:35169451-35169674:35169823-35170861:-				NA	NA	NA	NA	NA
33	gene-1110038F14Rik;A5:NC_000081.7:76833383-76833744:76833352-76833744:+				NA	NA	NA	NA	NA

Not good, it's all NAs everywhere.

So somehow, getting the PSI values is the part that didn't work properly in the Salmon part.

Is it possible that the problem is again a mismatch between identifiers? The SUPPA tutorial has a note "Transcript_IDs in the expression file must be the same as the transcript_IDs in the .ioe file, including the version numbers, otherwise it won't work.". Could it be that the issue is having the non-prefixed IDs in the suppa_tpm.txt but having prefixes in the events.ioe file? But no, I've already seen that that one contains proper-looking NCBI identifiers.

However, I spotted the problem, and I think it's the last two columns in the events.ioe file:

events.ioe [Read-Only]									
~/Documents/Backup_shared_folder/Pre-eclampsia_results/salmon/suppa/generate_events/per_local_event									
1	seqname	gene_id	event_id	alternative_transcripts	total_transcripts				
2	NC_000067.7	gene-Tceal	gene-Tceal;A3:NC_000067.7:4957054-4959680:4957054-4959683:+		rna-NM_011541.4,rna-NM_01159751.1 rna-NM_011541.4,rna-NM_01159751.1,rna-NM_01204371.1,rna-NM_01318735.1				
3	NC_000067.7	gene-Oprk1	gene-Oprk1;A3:NC_000067.7:5658886-5659228:5658886-5659258:+		rna-XR_004934097.1,rna-XM_006495441.5,rna-XM_036153075.1				
4	NC_000067.7	gene-Prex2	gene-Prex2;A3:NC_000067.7:11252258-11254613:11252258-11254670:+		rna-XR_004934097.1,rna-XM_006495441.5,rna-XM_036153075.1,rna-XR_373211.2				
5	NC_000067.7	gene-Gm31195	gene-Gm31195;A3:NC_000067.7:13951612-13952027:13951612-13952047:+		rna-XR_373211.2,rna-XR_373210.1,rna-XR_373212.1,rna-XR_373211				
6	NC_000067.7	gene-Gm9947	gene-Gm9947;A3:NC_000067.7:14826461-14845646:14826461-14845750:+		rna-XR_003947743.2,rna-XR_004935543.1,rna-XR_03947743.2,rna-XR_004935542.1				
7	NC_000067.7	gene-Gm9947	gene-Gm9947;A3:NC_000067.7:14844161-14845646:14844161-14845750:+		rna-XR_003947743.2,rna-XR_004935543.1,rna-XR_03947743.2,rna-XR_004935542.1				
8	NC_000067.7	gene-Terf1	gene-Terf1;A3:NC_000067.7:15883386-15889158:15883386-15889875:+		rna-XR_004941784.1,rna-XR_004941784.1,rna-XR_004941784.1,rna-XM_001420525.1				
9	NC_000067.7	gene-Gm7568	gene-Gm7568;A3:NC_000067.7:16324269-16333223:16324269-16333389:+		rna-XR_001778825.3,rna-XR_001778825.3,rna-XR_001778825.3,rna-XR_865202.3				
10	NC_000067.7	gene-Tmem70	gene-Tmem70;A3:NC_000067.7:16737529-16737889:16737529-16737892:+		rna-NM_026392.1,rna-NM_026392.1,rna-NM_027415.2				
11	NC_000067.7	gene-Ly96	gene-Ly96;A3:NC_000067.7:16761997-16771019:16761997-16771073:+		rna-NM_016923.2,rna-NM_016923.2,rna-NM_016923.2,rna-NM_016923.2				
12	NC_000067.7	gene-Tfap2b	gene-Tfap2b;A3:NC_000067.7:19304530-19305588:19304530-19306832:+		rna-XM_030252837.1,rna-XM_030252837.1,rna-XM_030252837.1,rna-XM_030252837.1				
13	NC_000067.7	gene-Efch1	gene-Efch1;A3:NC_000067.7:21049080-21049560:21049080-21049634:+		rna-XR_003946579.2,rna-XR_003946579.2,rna-XR_003946579.2,rna-XM_027974.1				
14	NC_000067.7	gene-Tmem14a	gene-Tmem14a;A3:NC_000067.7:21296903-21299632:21296903-21299815:+		rna-XM_006495579.3,rna-NM_001290679.1,rna-NM_001290679.1,rna-NM_001290679.1				

To do

Here, I still have those rna- prefixes in front of all of the transcript IDs!!! This is probably why the pipeline can't quantify the PSIs properly -> I will have to remove all of these prefixes and then restart this part of the pipeline! The events.ioe of course has the very same issue!

PSI files in star_salmon folder:

.psi			
Name	Size	Location	
suppa_local.psi	17,4 MB	suppa/psi_per_local_event	
suppa_isoform.psi	13,8 MB	suppa/psi_per_isoform	

The star_salmon part only generated two files, suppa_local.psi and suppa_isoform.psi, which do seem to both contain lots of info but also lots of "nan" values. They're not **only** NAs like the .psi files from the salmon part, so here, some exons/splice sites might just not get used at all...or is it so simple? If I look at the values, in some samples it appears as 1 vs 0, which strikes me as rather odd.

suppa_local.psi:

*suppa_local.psi										
~/Documents/Backup_shared_folder/Pre-eclam...ults/star_salmon/suppa/psi_per_local_event										
1	CONTROL REP1	CONTROL REP2	CONTROL REP3	CONTROL REP4	PREECLAMPSIA REP1	PREECLAMPSIA REP2	PREECLAMPSIA REP3	PREECLAMPSIA REP4		
2	gene-0610005C13rik;SE:NC_000073.7:45217803-45220410:45220616-45223837:-				nan 1.0 nan nan nan nan	nan 1.0 nan nan nan	nan 1.0 nan nan nan	nan 1.0 nan nan nan		
3	gene-0610006L08Rik;SE:NC_000073.7:74469101-74469451:74469552-74474468:-				nan nan nan nan nan nan					
4	gene-0610006L08Rik;SE:NC_000073.7:74474592-74502200:74502294-74503295:-				nan nan nan nan nan nan					
5	gene-0610010K14Rik;A3:NC_000077.7:70126294-70126522:70126290-70126522:-				0.36237985741632	0.39842286109646097	0.2966690113282723	0.3715960157074983	0.32	
6	gene-0610010K14Rik;A3:NC_000077.7:70126294-70126823:70126290-70126823:-				0.5074183414423623	0.7605924601936709	0.4412847583363537	0.6595874237469798	1.0	
7	gene-0610010K14Rik;A5:NC_000077.7:70126294-70126823:70126290-70126850:-				0.07654152010390355	0.09390962706748793	0.050624904769660727	0.1221008707507892	0.06	
8	gene-0610010K14Rik;A5:NC_000077.7:70126626-70126823:70126626-70126823:-				0.009441730595943432	0.0 0.0 0.0 0.0 0.0	0.022726417501698	0.02180144440045283	0.036129388	
9	gene-0610010K14Rik;SE:NC_000077.7:70126290-70126522:70126626-70126823:-				0.06983407943483946	0.0 0.0 0.0 0.0 0.0	0.16831993804042075	1.0	0.3379514982532557	0.37
10	gene-0610010K14Rik;SE:NC_000077.7:70126294-70126522:70126626-70126823:-				0.0 0.0 0.0 0.0 0.0	0.08394141853172771	0.0 0.0 0.0 0.0 0.0			
11	gene-0610010K14Rik;SE:NC_000077.7:70126294-70126522:70126626-70126850:-				0.18821429953454596	0.16084738155230238	0.17661870138901148	0.19156134631451363	0.16	
12	gene-0610010K14Rik;SE:NC_000077.7:70127037-70127703:70127804-70128357:-				0.5926975895360438	0.5606249849108814	0.6827043034349142	0.6309902772995214	0.71	
13	gene-061002513Rik;AF:NC_000070.7:98209593-98210546:98210196-98210410-98210546:+				0.0 0.0 0.0 0.0 0.0	nan 1.0 nan nan 1.0	1.0 1.0 1.0 1.0 1.0			
14	gene-0610040F04Rik;AF:NC_000072.7:108590954-108600002:108600128-108590954-108637778:108637895:-				0.0 0.0 0.0 0.0 0.0	nan 1.0 nan nan 1.0	0.0 0.0 0.0 0.0 0.0			
15	gene-0610040J01Rik;AF:NC_000071.7:63969833-63969938-64053773:64034825-64034930-64053773:+				nan nan nan nan nan	nan nan nan nan nan	nan nan nan nan nan			

suppa_isoform.psi:

*suppa_isoform.psi										
~/Documents/Backup_shared_folder/Pre-eclam...results/star_salmon/suppa/psi_per_isoform										
1	CONTROL REP1	CONTROL REP2	CONTROL REP3	CONTROL REP4	PREECLAMPSIA REP1	PREECLAMPSIA REP2	PREECLAMPSIA REP3	PREECLAMPSIA REP4		
2	gene-Gm26206;rna-XR_004936710.1	nan	nan	nan	nan 0.0	0.0 0.0	nan 1.0	nan 0.0		
3	gene-Gm53491;rna-XR_004935518.1	0.0	nan	nan	nan 1.0	1.0 1.0	nan 0.0	nan 0.0		
4	gene-Gm53491;rna-XR_004935517.1	1.0	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
5	gene-Gm22307;rna-XR_004936894.1	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
6	gene-Gm39586;rna-XR_003947708.1	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
7	gene-Gm29874;rna-XR_001778816.2	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
8	gene-Gm29874;rna-XR_001778815.2	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
9	gene-Gm29874;rna-XR_001778819.2	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
10	gene-Gm29874;rna-XR_373196.3	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
11	gene-Gm7369;gene-Gm7369	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
12	gene-Lyplal1;rna-NM_001355712.1	0.041433128357916474	0.0	0.0	0.012343305956902813	0.015744693725033884	0.015871675262633116	0.0	0.06251964612212481	0.032723676
13	gene-Lyplal1;rna-NM_008866.3	0.9585668716420835	1.0	0.0	0.987656940430971	0.984255306274966	0.9481283247373669	1.0	0.9374803538778751	0.9672763230418453
14	gene-Gm57794;rna-XR_004935520.1	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
15	gene-Tceal1;rna-NM_001159750.1	0.11215239416510886	0.16721451947651078	0.12095780164053374	0.1256038687913864	0.11826446155517824	0.13036734748016476	0.14		
16	gene-Tceal1;rna-NM_011541.4	0.8834285907348687	0.8327854805234892	0.8754390122436836	0.8743961312086136	0.8817355384448219	0.8696326525198353	0.846067571		
17	gene-Tceal1;rna-NM_001159751.1	0.008441901510022401	0.0	0.0	0.003603186115782701	0.0	0.0	0.0	0.008070724961290214	0.0
18	gene-Gm16041;rna-XR_373197.3	nan	nan	nan	nan 0.0	0.0 0.0	nan 0.0	nan 0.0		
19	gene-Atp6v1h;rna-XM_006495433.4	0.0	0.0	0.0	0.013246627536890817	0.009962792984243369	0.0	0.0	0.016047565752173705	
20	gene-Atp6v1h;rna-XM_006495436.5	0.0	0.0	0.0	0.008938008128900164	0.0	0.0	0.0009330529345777168	0.0	
21	gene-Atp6v1h;rna-XM_030255207.2	0.0	0.0	0.0	0.0026417169192436227	0.0	0.0	0.0	0.0	
22	gene-Atp6v1h;rna-XM_030255244.2	0.0	0.0	0.0	0.007238485577201244	0.0	0.0	0.0	0.0	

Interestingly, this part of the pipeline generated some somewhat more sensible-looking psi files, in spite of also having the prefixes in the .ioe and .ioi files.

The part of the pipeline that gave the error, diffSplice, should use the PSI files and the .tpm files, PREECLAMPSIA.tpm and CONTROL.tpm. The same type of split applied to the psi files happens to the suppa_tpm.txt file, resulting in the CONTROL.tpm and PREECLAMPSIA.tpm files, so that's where they come from.

Fix mismatch between suppa_tpm.txt files and .ioe and .ioi files

tpm files and prefixes

Double checking where I have the suppa_tpm.txt files, and whether they have or don't have the prefixed IDs:

```

sudo find -name "suppa_tpm.txt"
1 ./Pre_eclamp...mice_rnas.../salmon/tximport/suppa_tpm.txt -no prefixes
2 ./Pre_eclamp...mice_rnas.../star_salmon/tximport/suppa_tpm.txt -PREFIXES

3 ./Pre_eclamp...Nextflow_work_folder/94/f58eb18a126bfe5f5bf14f039b04e0/suppa_tpm.txt -PREFIXES

4 ./Pre_eclamp...Nextflow_work_folder/a6/26f0de2c425f9543d3a9feb3819fd1/suppa_tpm.txt -no prefixes

5 ./Pre_eclamp...Nextflow_work_folder/2e/e854754f8b8356ae2d0484f72704ae/suppa_tpm.txt -PREFIXES

6 ./Pre_eclamp...Nextflow_work_folder/a3/25537db88f97da97a364ac984e73b5/suppa_tpm.txt -no prefixes

7 ./Pre_eclamp...Nextflow_work_folder/1c/c6527462913e81a675c76bda7b5b85/suppa_tpm.txt -no prefixes

8 ./Pre_eclamp...Nextflow_work_folder/4b/be20738da4815ab9d8e928e633da53/suppa_tpm.txt -PREFIXES

9 ./Pre_eclamp...Nextflow_work_folder/3c/3593b1fa74aef...7ba55a9c5f6549be/suppa_tpm.txt -PREFIXES

10 ./Pre_eclamp...Nextflow_work_folder/b1/401dd8bb1ca47549d3bc1f7d8deed1/suppa_tpm.txt -no prefixes

```

11

```
./Pre_eclampsia_Nextflow_work_folder/b0/f0e97555e06ec3fd61e8aac89f3e9e/suppa_tpm.txt -PREFIXES
```

I have two files in the "results" folder. The suppa_tpm.txt file without prefixes is in the "salmon" branch, the one without prefixes is in the "star_salmon" branch. Interestingly, the "salmon" branch, where the tpm file has no prefixes, is the one with more problems upon quantification.

In the work folder, I have WITH prefixes the work folders

```
3. /04/f58eb18a126bfe5f5bf14f039b04e0 —— symlink to /4b/be20738da4815ab9d8e928e633da53  
5. —/2e/e954754f8b8356ae2d0484f72704ae - symlink to /4b/be20738da4815ab9d8e928e633da53  
8. /4b/be20738da4815ab9d8e928e633da53 - proper file, PREFIXES.  
9. /3c/3593b1fa74aef7ba55a9c5f6549be - proper file, PREFIXES.  
11. #b0/f0e97555e06ee3fd61e8aae89f3e9e symlink to /4b/be20738da4815ab9d8e928e633da53
```

WITHOUT prefixes:

```
4. /a6/26f0de2c425f9543d3a9feb3819fd1 - proper file, no prefixes.  
6. /a3/25597db88f97da97a364ac984e73b5 - symlink to /a6/26f0de2c425f9543d3a9feb3819fd1  
7. /4c/e6527462913e81a675e76bda7b5b85 - symlink to /a6/26f0de2c425f9543d3a9feb3819fd1  
10. #b1/401dd8bb1ea47549d3be1f7d8deed4 symlink to /a6/26f0de2c425f9543d3a9feb3819fd1
```

That means I have 1 tpm file without prefixes, the one in /a6/26f0de2c425f9543d3a9feb3819fd1, and two with prefixes, /4b/be20738da4815ab9d8e928e633da53 and /3c/3593b1fa74aef7ba55a9c5f6549be

So the problem seems to arise from the fact that I have a number of quantification files from Salmon, some from an old run of the pipeline where I specified the wrong index (Ensembl identifiers), some that have the prefixes in front of the NCBI IDs, and some that don't have the prefixes. And then the same happens for the .ioe and .ioi annotation files and the tpm files.

This means that the easiest way out would probably be to start SUPPA from scratch, using its own script and giving it a GTF file and Salmon quantification files that I know all have matching identifiers, either with or without prefixes. But I'd still prefer to do this in the context of the rnas splice pipeline, because that makes it a lot easier to trace back and fix errors, as I've done so far. So I need to make sure that I start the same run, but delete the GTF and quant.sf files that have the prefixes in their work folders in the hope that the pipeline will now build them all correctly.

Let's check out the traces to better understand which process generated which files, especially for my last pipeline run:

```
/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnas splice_results/pipeline_info
```

Find entries related to SUPPA in the execution traces in the pipeline_info folder:

```
sudo grep -Hni "SUPPA" /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnas splice_results/pipeline_info/*.txt
```

-H = print file name

-n = print line number and contents

-i = case-insensitive search

(side note, I found out that Nextflow also allows one to specify an [option "-with-report"](#) that creates an html file where the information from the traces is gathered in a much more human-friendly way, with plots about performance, etc)

Pasted results into "2024-08-22_SUPPA_tasks_execution_traces.xlsx".

I have two batches of processes related to SUPPA that are marked "COMPLETED" (and not "CACHED", which indicates that these processes picked up the data from one of their predecessors that was COMPLETED).

One is from the 14th of August (my very first run of the pipeline):

```
63/a29e5f 29773 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf)  
COMPLETED  
1b/e0d509 29732 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf)  
COMPLETED  
f7/39b5ee 29751 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOE  
(GCF_000001635.27_GRCm39_genomic.gtf) COMPLETED  
fc/ffa5ce 38378 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOI  
(GCF_000001635.27_GRCm39_genomic.gtf) COMPLETED  
69/92c17f 38393 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOE  
(GCF_000001635.27_GRCm39_genomic.gtf) COMPLETED  
df/8384fe 44336 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf)  
COMPLETED  
8f/025573 45724 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf)  
COMPLETED
```

And one is from the 20th:

```
1c/c65274 99337 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:PSIPERISOFORM (suppa_tpm.txt) COMPLETED
```

a3/25537d 100294 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:PSIPEREVENT (suppa_tpm.txt) COMPLETED
d0/285534 101163 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:SPLIT_FILES_IOC (suppa_isoform.psi) COMPLETED
b0/f0e975 101137 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:SPLIT_FILES TPM (suppa_tpm.txt) COMPLETED
b1/401dd8 101130 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:SPLIT_FILES TPM (suppa_tpm.txt) COMPLETED
5c/3033d5 101911 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:SPLIT_FILES IOE (suppa_local.psi) COMPLETED
2e/e85475 120384 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:PSIPERISOFORM (suppa_tpm.txt) COMPLETED
94/f58eb1 121280 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:PSIPEREVENT (suppa_tpm.txt) COMPLETED
70/2bf3a9 121738 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:DIFFSPLICE_IOC (PREECLAMPSIA-CONTROL) FAILED

Everything else was used from a cache, e.g. on the 20th:

8f/025573 45724 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOC (GCF_000001635.27_GRCm39_genomic.gtf)
CACHED
fc/ffa5ce 38378 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOC
(GCF_000001635.27_GRCm39_genomic.gtf) CACHED
69/92c17f 38393 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOC
(GCF_000001635.27_GRCm39_genomic.gtf) CACHED
df/8384fe 44336 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOC (GCF_000001635.27_GRCm39_genomic.gtf)
CACHED
8f/025573 45724 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOC (GCF_000001635.27_GRCm39_genomic.gtf)
CACHED
fc/ffa5ce 38378 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOC
(GCF_000001635.27_GRCm39_genomic.gtf) CACHED
df/8384fe 44336 NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOC (GCF_000001635.27_GRCm39_genomic.gtf)
CACHED
69/92c17f 38393 NFCORE_RNASPlice:RNASplice:SUPPA_STAR_SALMON:GENERATE_EVENTS_IOC
(GCF_000001635.27_GRCm39_genomic.gtf) CACHED

It looks like I might still have the IOE and IOI files I generated the first time around, before I had to modify any of the identifiers for tximport to work, and the pipeline used these when calculating the PSI per isoform, per event, etc. I'll check the work folder for some of the cached operations, to see if they symlink to the files created on the 14th. And no, it is the actual folder that was created on the 14th! This is true for all of the other cached results. So the pipeline simply used the cached files stored in these directories. Which means this is likely the root of all evil in this case **[chuckle]**.

I now also know where are the .ioi and .ioe files for this run are located:

8f/025573 - events.ioe, plus all of the parts relating to specific exons, alternative splice site, etc. Contains prefixes.
69/92c17f - events.ioe, plus all of the parts relating to specific exons, alternative splice site, etc. Contains prefixes.
df/8384fe - events.ioi. Contains prefixes.
fc/ffa5ce - events.ioi. Contains prefixes.

This is all getting extremely convoluted. The easiest way would probably be to delete all of the folders in the work folders that I identified as containing "COMPLETED" processes, and starting the pipeline again with only SUPPA activated and all other outcome options set to false, to not have it run any other parts of the pipeline.

I went systematically through the work folder, based on the Excel file, and deleted the files one by one with `sudo rm -r` to avoid any other issues.

Interim pipeline run continuation without SUPPA

started around 11:30 AM

12:24: results folder = 167 GB, work folder = 166.6 GB

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--sashimi_plot false \
--suppa false \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results \
--input /home/iweber/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/samplesheet_preeclampsia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclampsia.csv \
--fasta \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
```

```
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-resume d2143e7b-2e6a-41e4-850d-cb8af9596f82
```

Resume pipeline with new SUPPA run

I will try to let the pipeline run SUPPA only and directly from Salmon quant.sf files, with the option --source salmon_results activated.

The sample sheet needs to look like:

```
sample,condition,salmon
CONTROL_REPO1,control,AEG588A1.tar.gz
CONTROL_REPO2,control,AEG588A2.tar.gz
CONTROL_REPO3,control,AEG588A3.tar.gz
TREATMENT_REPO1,treatment,AEG588A4.tar.gz
TREATMENT_REPO2,treatment,AEG588A5.tar.gz
TREATMENT_REPO3,treatment,AEG588A6.tar.gz
```

The quant.sf files that seem to have worked were the ones used in the star_salmon run of the pipeline. The ones in the results file that fit it are the ones WITH the prefixes, so that's what I should give the pipeline.

I made the file samplesheet_preeclampsia_SalmonSUPPA.csv:

```
sample,condition,salmon_results
CONTROL_REPO1,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/CONTROL_REPO1
CONTROL_REPO2,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/CONTROL_REPO2
CONTROL_REPO3,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/CONTROL_REPO3
CONTROL_REPO4,CONTROL,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/CONTROL_REPO4
PREECLAMPSIA_REPO1,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/PREECLAMPSIA_REPO1
PREECLAMPSIA_REPO2,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/PREECLAMPSIA_REPO2
PREECLAMPSIA_REPO3,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/PREECLAMPSIA_REPO3
PREECLAMPSIA_REPO4,PREECLAMPSIA,/home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/salmon/PREECLAMPSIA_REPO4
```

I also used --suppa_tpm to give the pipeline a distinct tpm file, one of the ones with prefixes, and set --diffsplice_paired to false to better reflect the experimental setup.

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--source salmon_results \
--rmats false \
--dexseq_dtu false \
--dexseq_exon false \
--edger_exon false \
--suppa_tpm /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/suppa_tpm.txt \
--diffsplice_paired false \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder_SalmonSUPPA \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results_SalmonSUPPA \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/samplesheet_preeclampsia_SalmonSUPPA.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclampsia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--aligner star_salmon \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--salmon_index
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
```

```
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
--profile docker \
-with-report
```

Annd it completed successfully :) I tried looking into the execution_timeline and execution_report files, and initially couldn't, because, apparently, I am not the owner. Rectified that for the files execution_timeline_2024-08-22_17-46-37.html and execution_report_2024-08-22_17-46-37.html using sudo chown , and I could then open the reports.

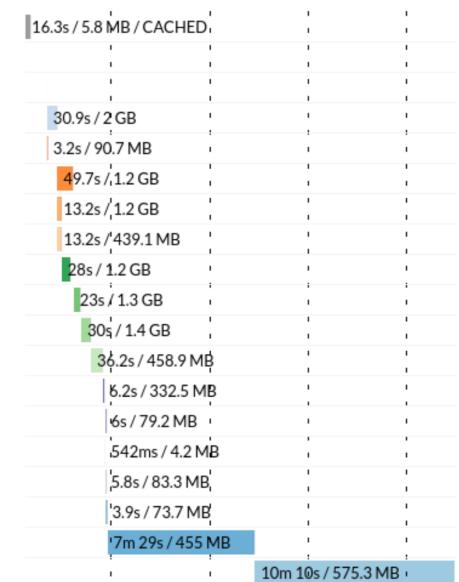
They are a dream to look at. They even contain the info that's in the trace files, with even more detailed information on how long the tasks ran, how much resources they used up, etc.

Here's a snippet from the timeline:

Processes execution timeline

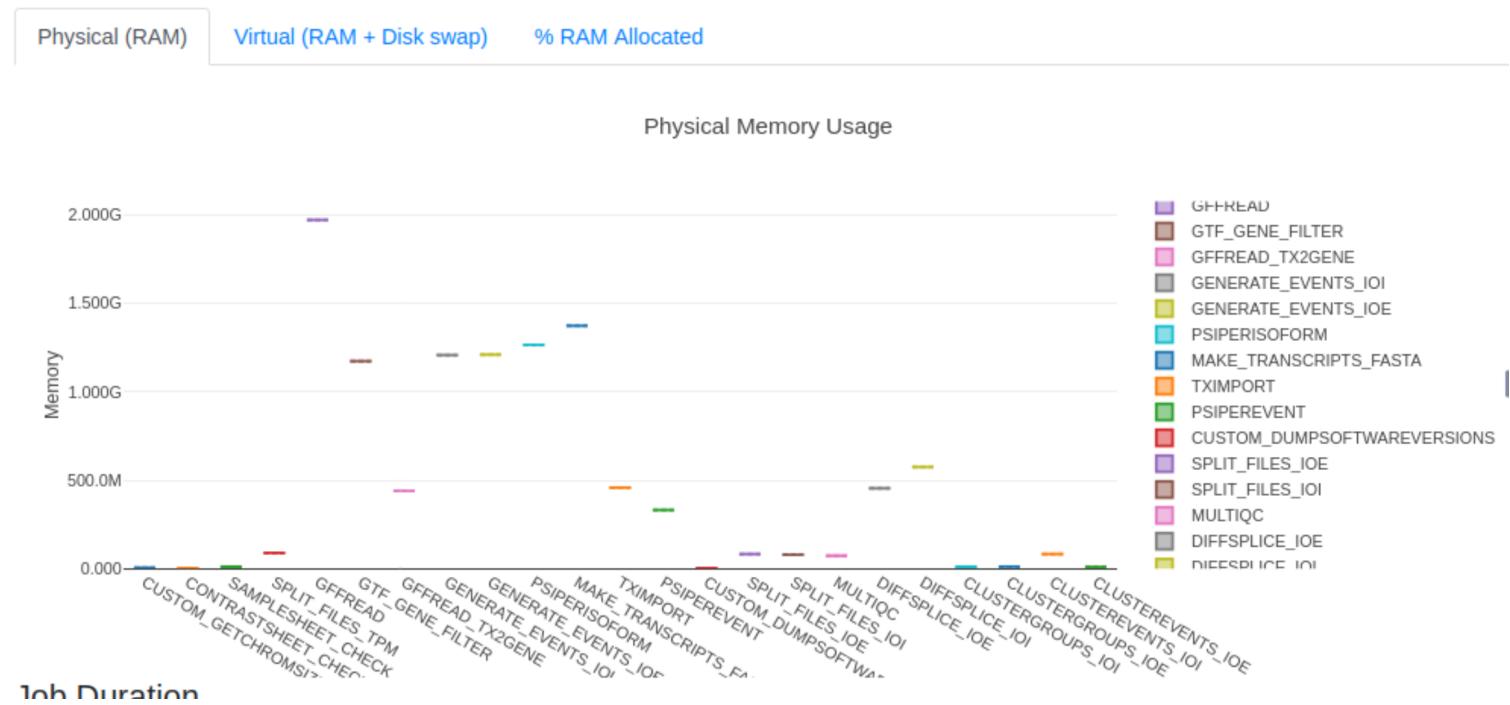
Launch time: 22 Aug 2024 17:45
Elapsed time: 20m 55s
Legend: job wall time / memory usage (RAM)

```
NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:CUSTOM_GETCHROMSIZES (GCF_000001635.27_GRCm39_genomic.fna)
NFCORE_RNASPlice:RNASplice:CONTRASTS_CHECK:CONTRASTSHEET_CHECK (contrastsheet_preeclampsia.csv)
NFCORE_RNASPlice:RNASplice:INPUT_CHECK:SAMPLESHEET_CHECK (samplesheet_preeclampsia_SalmonSUPPA.csv)
NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GFFREAD (GCF_000001635.27_GRCm39_genomic.gff)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:SPLIT_FILES TPM (suppa_tpm.txt)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOE (GCF_000001635.27_GRCm39_genomic.gtf)
NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:GTF_GENE_FILTER (GCF_000001635.27_GRCm39_genomic.fna)
NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON:GFFREAD_TX2GENE (GCF_000001635.27_GRCm39_genomic.gtf)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:GENERATE_EVENTS_IOI (GCF_000001635.27_GRCm39_genomic.gtf)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:PSIPERISOFORM (suppa_tpm.txt)
NFCORE_RNASPlice:RNASplice:PREPARE_GENOME:MAKE_TRANSCRIPTS_FASTA (rsem/GCF_000001635.27_GRCm39_genomic.fna)
NFCORE_RNASPlice:RNASplice:TX2GENE_TXIMPORT_SALMON:TXIMPORT (GCF_000001635.27_GRCm39_genomic.tx2gene.tsv)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:PSIPEREVENT (suppa_tpm.txt)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:SPLIT_FILES_IOC (suppa_isoform.psi)
NFCORE_RNASPlice:RNASplice:CUSTOM_DUMP SOFTWAREVERSIONS (1)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:SPLIT_FILES_IOE (suppa_local.psi)
NFCORE_RNASPlice:RNASplice:MULTIQC
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:DIFFSPlice_IOE (PREECLAMPSIA-CONTROL)
NFCORE_RNASPlice:RNASplice:SUPPA_SALMON:DIFFSPlice_IOC (PREECLAMPSIA-CONTROL)
```



And here from the report (it's interactive!)

Memory



Tasks

This table shows information about each task in the workflow. Use the search box on the right to filter rows for specific values. Clicking headers will sort the table by that value and scrolling side to side will reveal more columns.

Values shown as: Human readable ▾

Show All entries

Filter: Metrics Metadata All Search:

task_id	process	tag	status	hash	allocated cpus	%cpu	allocated memory	%mem
1	NFCORE_RNASPlice:RNASplice:PREP_RIBOGENOME_GERMLINE_genome	COMPLETED	61/cf36c8	2	100.4	12.000 GB	2.2	
2	NFCORE_RNASPlice:RNASplice:SUBP_SALMON:SPLIT_FILES TPM	COMPLETED	e1/83ec7d	2	100.9	12.000 GB	0.1	
3	NFCORE_RNASPlice:RNASplice:INPUT_CHECK_SAMPLESHEET_CHECK	COMPLETED	ae/0c3399	1	92.3	6.000 GB	0.0	
4	NFCORE_RNASPlice:RNASplice:PREP_RIBOGENOME_GERMLINE_GTF_CDS_CACHED_ES	CACHED_ES	39/edd456	1	99.3	6.000 GB	0.0	

Nb for future use:

They're using something called Plotly to create the interactive plots, which is Python-based

Original authors' results

From GEO page "Methods: E17.5 cortex mRNA profiles of offspring from control and preeclampsia mother mice were generated by deep sequencing, in triplicate, using Illumina GAIix. The sequence reads that passed quality filters were analyzed at the transcript isoform level with two methods: Burrows–Wheeler Aligner (BWA) followed by ANOVA (ANOVA) and TopHat followed by Cufflinks. Results: Using an optimized data analysis workflow, we **mapped about 30 million sequence reads** per sample to the mouse genome (build mm9) and **identified 16,014 transcripts in the E17.5 cortices of offspring from control and preeclampsia mother mice with BWA workflow** and **34,115 transcripts with TopHat workflow.**"

General stats

By analyzing the BAM files, Samtools Stats returned a total of around 40 million reads mapped per sample. From these 40 million, STAR returned 20 million uniquely mapped reads. Knowing that I have a read length of 150 bp, what's the coverage and depth I reached for this dataset? How does this fit with what people usually use for alternative splicing analysis? Also: it looks like I lost around 20% of most uniquely mapped reads in the Salmon step, leaving me with only around 15-16 million reads per sample. This may be due to the fact that the NCBI annotation file (GTF) I used is not as exhaustive as the GENCODE annotation file.

To do

coverage calculation. A depth of 50-100x coverage is often desired for confident detection of **rare or tissue-specific splicing events**.

Running DESeq2 to get gene expression results

For quantifying gene expression in a non-splice-sensitive way, I went with DESeq2, and decided to run it in a separate R project, called "". I made a script called "2024-08-22_DESeq2_pre-eclampsia", based on the one from the course from Ioana Lemnian[link to Ioana's page], but updated to reflect the DEseq2 [tutorial](#).

Correspondence my variables ~ variables official tutorial

my_ddstxi ~ dds and ddsTxi ~ : the DESeqDataSet which DESeq2 performs the analysis on
my_dds_filtered ~ : version of the DESeq2DataSet after rows with low counts are removed
my_DESeq_gene : raw DESeq2 result with nominal p values
my_DESeqGene_result ~ res: DESeq2 result with p values adjusted for multiple comparisons with the Benjamini-Hochberg method (FDR)

Run DESeq2

I used the Salmon quant.sf files generated by the star_salmon run of the pipeline, the ones containing prefixes for the identifiers. For the tx2gene file from the pipeline, I used "GCF_000001635.27_GRCm39_genomic_PREFIXES.tx2gene.tsv", which, in essence, is a copy of the file I initially termed "FAULTY" because it contained the prefixes. But it will do for the DESeq quantification.

I ran all of the DESeq2 steps and then tried the interactive plotting with the plotMA function from the DESeq2 package, but was not satisfied with the interactive selection of data points and how the gene names displayed. I therefore tried using Plotly (see above), but the plots would not display neither in the viewer nor open in a web browser. I read that this is an [issue that the latest version of R causes with Rstudio's viewer](#) and that a patched version (R-4.4.1patched) would fix it. I installed the new version from <https://cran.r-project.org/bin/windows/base/rpatched.html> and restarted RStudio. I then reopened RStudio, went to Tools -> Global Options -> General, and set the R installation to be used to the new, patched version. And now it works - I have Plotly plots in which I can click on individual dots to see the gene name associated with it.

Authors' DEG analyses from source publication

"the clean reads were obtained and aligned to the mouse genome (mm10, NCBI) using **Hisat2** (82). **HTSeq** was applied to obtain gene counts, and gene expression was determined using the RPKM method (83)."

"For the analysis of DEGs, the **EBSeq algorithm** (85) was applied to filter DEGs, after the significant analysis, P-value, and false discovery rates (FDRs) were calculated. The following criteria were used: (i) **log2(fold change) >0.585 or <-0.585** for E17.5 RNA-seq and >0.263 or <-0.263 for adult hippocampus RNA-seq; and (ii) **FDR<0.05** (86)."

By setting these criteria, the authors found a total of 256 DEGs with log2FCs between -4.61 and 4.68 (0.04 to 25-fold change). They set cutoffs at 0.585 and -0.585, which corresponds to a 1.5-fold increase or decrease down to 66%, respectively.

I am not convinced by the authors' use of EBSeq instead of DESeq2. Its latest update is 8 years ago, whereas DESeq2 is regularly maintained (last update is from a month ago, July 2024). Additionally, it uses RPKM, which is suitable for intra-sample gene expression level comparison, but NOT for DEG analysis between samples. EBSeq does have the advantage of having been developed specifically for identifying alternative transcript usage, so not just at the gene expression level, but it performs better on complex experimental setups (multiple intra- and inter-subject comparisons), whereas DESeq can also perform transcript-level analysis if the tx object it uses is previously generated with the "txOut = TRUE" argument. Interestingly, the authors still report their results at gene level, not transcript level.

```
txi <- tximport(files, type = "salmon", txOut = TRUE)
```

EBSeq: <https://academic.oup.com/bioinformatics/article/29/8/1035/228913> and <https://bioconductor.org/packages/release/bioc/html/EBSeq.html>

Tutorial: <https://github.com/lengning/EBSeq>

Authors' GO analyses from source publication

"**GO analysis** was performed to facilitate elucidating the biological implications of the DEGs in the experiment. DEGs in the cortices of E17.5 embryos exposed to PE were compared with the genes in the Human Gene module of Simons Foundation for Autism Research Initiative using enrichment analysis via **phyper function in R**."

Completing git repo with results

I tried adding all of the results to the git repo in one go, and that was too much data at once, and I think my connection kept dropping during the upload process. So, I want to now backroll the commit with all of these files, and try adding the results folders from the different parts of the pipeline separately. I used [this tutorial](#) to better understand the reset options for Git. It says:

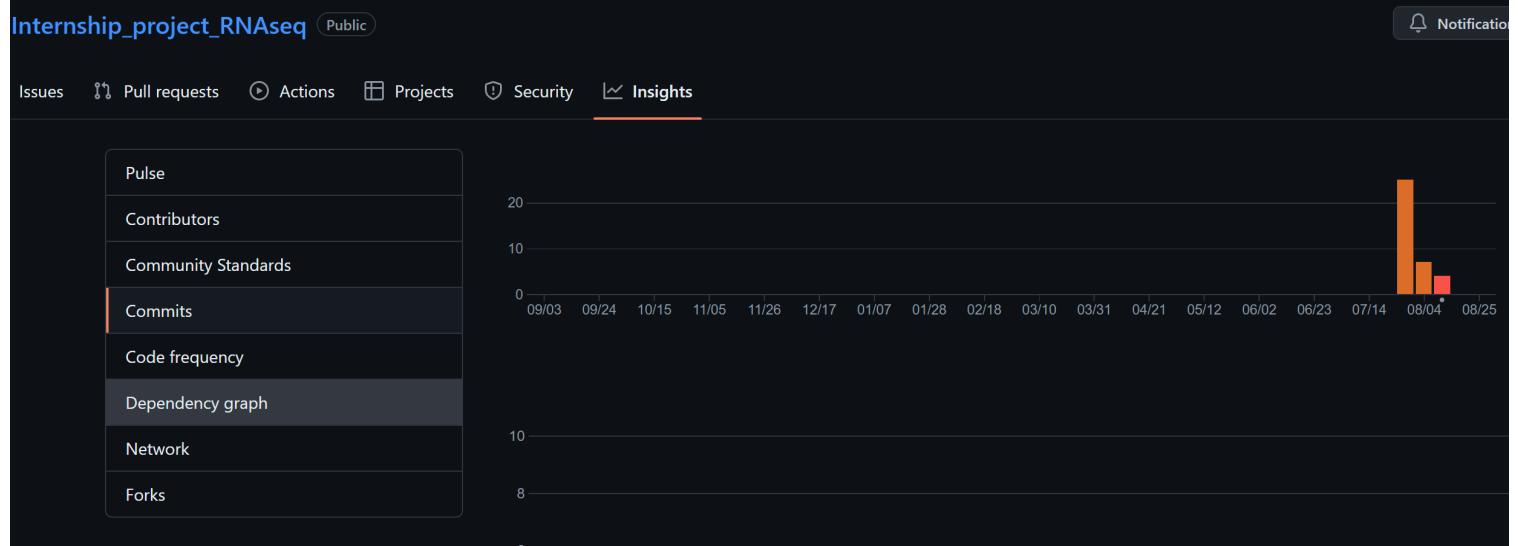
Note

```
### Overview of git reset
```

- **git reset --mixed:** Resets the HEAD and updates the staging area, but leaves the working directory unchanged.
- **git reset --soft:** Only resets the HEAD, leaving both the staging area and the working directory unchanged.
- **git reset --hard:** Resets the HEAD, updates the staging area, and resets the working directory to match the specified commit."

So what I need to do is a mixed reset. I initially thought I might need to use `git revert`, since I am using an online repository and that would create a fully new commit when reverting to a previous state, so that the entire history of changes is preserved, including this change (doesn't lose commits

between the current one and the one I'd be reverting to). However, I checked my online repository, and none of the results files seem to have uploaded. Plus, my last commit is from the week of August 11th:



Based on [this tutorial](#), it's safe to do a default (mixed) reset, with which I can set the head to point at the previous commit. To find out which one that was, I used `git log --oneline` in the Visual Studio Code terminal to display all of my commits. What I get is:

```
PS C:\Win_Ubuntu_shared> git log --oneline
273c660 (HEAD -> main) Rnasplice pipeline results and first analyses (DESeq2)
fa817ae (origin/main) Deleted unnecessary files, e.g. supplementary from original pre=eclampsia paper
8df2686 Deleted unnecessary files, e.g. supplementary from original pre=eclampsia paper
03abed5 Final parameter descriptions for pipeline
22e2f07 Safety changes to VM, new sample sheet pre-eclampsia data
68512bc After Nextflow update to 24.04
95e3f8c STAR Salmon indices, incl for C. elegans test
540c495 After installation STAR Salmon
9a4f924 Pipeline pull and further parameter readup
8d16a20 Progress Cuda Docker shared folder
c2df828 Success Docker pigz progress
7da8d10 Modify GPU and CPU options; update samplesheet reverse
ca62e75 Added progress imgs VMware Toolbox and updated overview files
90e5d29 Overview file update - transition to VMware
61e22ca Added .fastq.gz to gitignore; added contrasts sheet
a39c376 Project overview - pipeline parameters, sample sheet, strandedness
0ee20fa Updated project progress overview PDF, info on nvidia and cuda errors and git LFS plus large file handling
2d182eb Removed fastqs from gitattributes and added to gitignore
e30ec90 Restructured dataset folder to include raw FastQs
9483fc3 Set .gitattributes to track FastQ files with git LFS
```

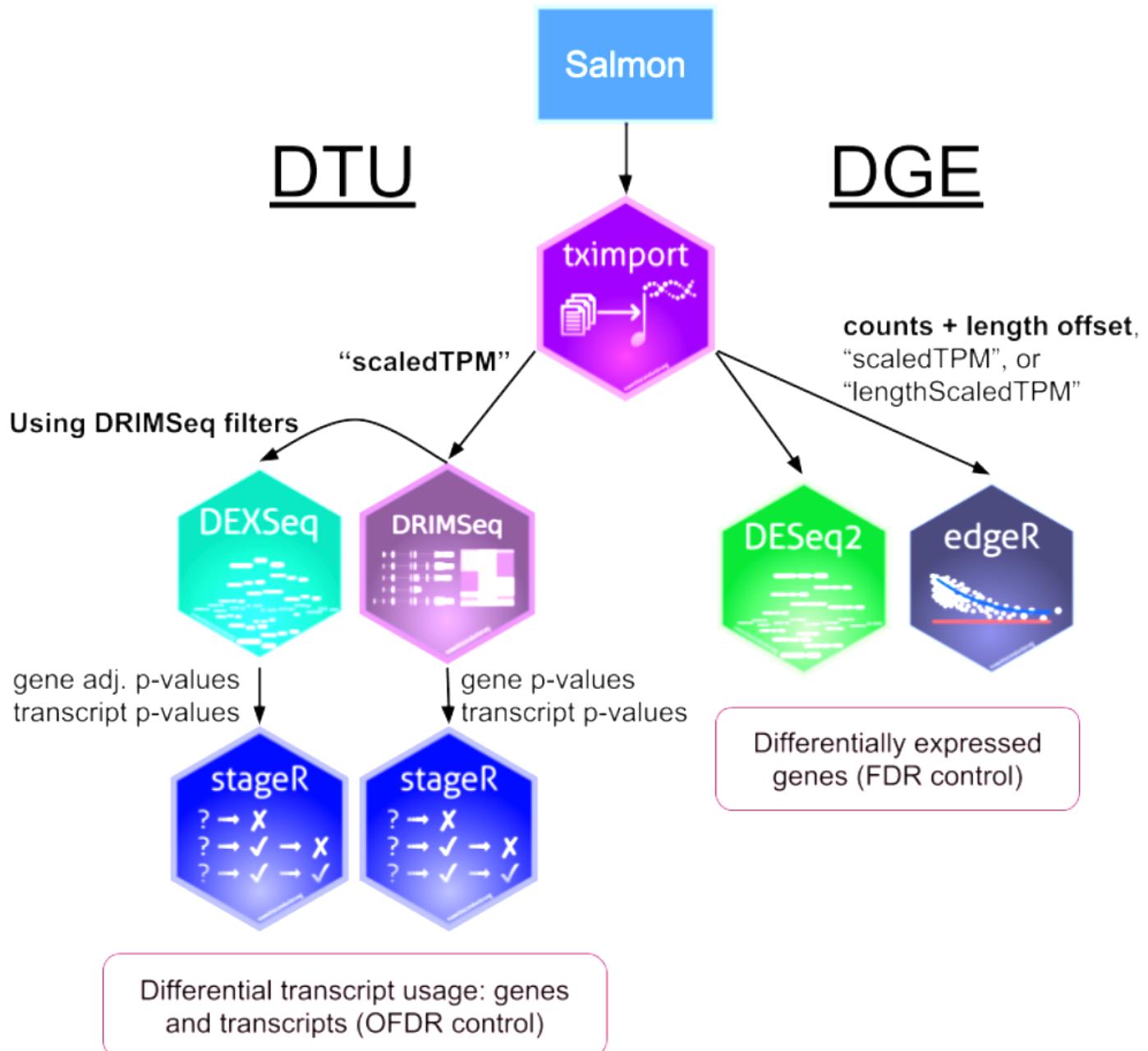
So the commit I need to reset to is fa817ae, the one right before the current one. This will perform a mixed reset, after which I can add the changed files in smaller batches to the staging area and commit them in these smaller packets. I did it with `git reset fa817ae`, and all was good, and I could finally upload the results files in small batches and avoiding large ones, such as the .rds files that rMATS and DEXSeq exon produce.

Check results of other pipeline components

For all of these, I used the results I got from the star_salmon arm of the pipeline (I copied the results folders to my Win-Ubu shared folder)

Checking DEXSeq differential transcript usage (DTU) results

DEXSeq was originally created for the quantification of how often an exon is used in all of the transcripts that a gene locus generates. Like DESeq2, it uses tximport to create an object containing the reads for all genes across samples, imported from the Salmon quantification files.



The DEXSeq author's recommendation is to use the `scaledTPM` method to generate the counts from the abundance data [how is this done? What are the differences between the scaling methods? See Trapnell et al., 2013]. The pipeline does this automatically by setting a parameter called `dtu_txi = 'dtuScaledTPM'` in the config file.

Next, the pipeline used [DRIMSeq](#), just as per the [recommendations](#), as a statistical model to identify differentially expressed transcripts from the `cts` object output by `tximport`. This complements the DEXSeq algorithm, which was originally created for estimating differences in exon usage, and can return lots of false positives in terms of significant FDRs when it comes to complete transcripts. Instead, DRIMSeq assumes that the counts for all of the transcripts that a gene produces, taken together, stay constant, but what differs is the proportion in which each transcript contributes to the total counts. This way, DRIMSeq can detect differential transcript usage between conditions when these ratios vary more strongly between conditions than they do between the samples of a single condition.

! In the `cts` object, the column sums should be equal to the number of reads that were mapped by Salmon !!!

The only objects I can find in the "results" folder are the `DEXSeqDataSet`, the results, and `perGeneQValue` in RDS format, and then the results and `perGeneQValue` in TSV format as well. So, if I want to look into what the pipeline did before that, such as the actual transcript reads generated by `tximport`, I'll have to find it in the "work" folder.

I have two folders where `tximport` worked, both from 20.8., and they both contain a variety of tsv files. The folders are
`a6/26f0de2c425f9543d3a9feb3819fd1` from "salmon" part of pipeline

`4b/be20738da4815ab9d8e928e633da53` from "star_salmon" part of pipeline. These files are slightly larger, but also contain the prefixes.

I imported the TSV file `DEXSeqResults.PREECLAMPSIA-CONTROL.tsv` from the `star_salmon` part of the pipeline. It has several columns:

- "groupId" - the gene the transcripts belong to
- "featureID" - the transcript identifier (I double checked with `n_distinct(DEXSeq_res$featureID)`, and they're indeed all unique)
- "exonBaseMean" - confusingly, in this case, how highly expressed a transcript is across all of the samples
- "dispersion" - degree of variability in transcript usage between samples

```
"stat" - usually, the Wald test statistic calculated for this transcript  
"pvalue" -  
"padj" -  
"PREECLAMPSIA" -  
"CONTROL" -  
"log2fold_CONTROL_PREECLAMPSIA" -
```

Checking DEXSeq differential exon usage (DEU) results

perGeneQValue() function calculates a so-called Q value. This is a FDR calculated for the entirety of a gene, which tells us if at least one exon in that gene undergoes alternative usage (note: authors specifically use "differential usage" because exons at the 5' end of a gene can be differentially used for reasons other than alternative splicing, such as using different transcription start sites when alternative promoters get activated). To get the number of genes with at least one differentially used exon:

```
numbOfGenes = sum( perGeneQValue(dxr) < 0.1)
```

Checking edgeR differential exon usage results

Tutorial: <https://bioconductor.org/packages/devel/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf>

edgeR is usually used for DEG analysis, but is used in the context of the rnassplice pipeline to perform differential exon usage analysis.

The rnassplice pipeline used featureCounts to generate the table of read counts (=reads aligning to each gene or exon, derived from the BAM files produced by STAR) that edgeR works on. edgeR can also work on the transcript level if the estimated abundances are input from Salmon quantification files via tximport.

From the pipeline, I got several result files from edgeR: exprs.csv, usage.exon.csv, a usage.gene.csv, and a usage.simes.csv. The ouput page of the pipeline says the following:

" Results CSV files are also output per comparison as {contrast}.usage.usage.csv [I suspect they mean usage.exon.csv] which contain the differential exon usage results, {contrast}.usage.gene.csv which contain results at a gene level identified with F-tests, and {contrast}.usage.simes.csv which contain results at a gene level using the Simes adjustment."

The edgeR manual says:

"Both exon-level and gene-level tests can be performed simultaneously using the diffSpliceDGE() function in edgeR. The exon-level test tests for the significant difference between the exon's logFC and the overall logFC for the gene. Two testing methods at the gene-level are provided. The first is to conduct a gene-level statistical test using the exon-level test statistics. Whether it is a likelihood ratio test or a QL [quasi-likelihood] F-test depends on the pipeline chosen. The second is to convert the exon-level p-values into a genewise p-value by the Simes' method. The first method is likely to be powerful for genes in which several exons are differentially spliced. The Simes' method is likely to be more powerful when only a minority of the exons for a gene are differentially spliced."

"The quasi-likelihood (QL) method is highly recommended for differential expression analyses of bulk RNA-seq data as it gives stricter error rate control by accounting for the uncertainty in dispersion estimation. The likelihood ratio test can be useful in some special cases such as single cell RNA-seq and datasets with no replicates."

- exprs.csv - appears to include values related to exons (several data points per gene)
- usage.exon.csv - exon-level FDR and log2 fold changes
- usage.gene.csv - I assume these are the gene-level statistical tests using the exon-level test statistics
- usage.simes.csv - these have to be the genewise p-values, converted from the exon-level p-values by the Simes' method

Checking rMATS alternative splicing results

Tutorial explaining the output: <https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2023/rmats/>

...I just saw on the rnassplice pipeline page something that eluded me before: "Furthermore, --rmats_read_len has to be set by the user and if the read length is variable, an average or median read length has to be specified." I checked, and I had not set such a parameter before. I know from the [parameters page](#) that the pipeline sets the default to 40, and, even though I do see that NOT_EXPECTED_READ_LENGTH is 0 for all samples in the read_outcomes_by_bam.txt file, the analysis might not be correct. This means I need to re-run the rMATS part of the pipeline with --rmats_read_len 150 . In the meanwhile, I can set up my scripts to check the analysis quality and the results, as I will be able to reuse these once I have the new rmats results.

To do

- check info from C:\Win_Ubuntu_shared\Pre-eclampsia_dataset_raw_and_processed\Pre_eclampsia_mice_rnassplice_results\rmats\PREECLAMPSIA-CONTROL\rmats_temp\2024-08-20-18_20_27_773885_read_outcomes_by_bam.txt
- check read coverage

check distribution of p-values/FDR

check JCEC datasets for each of the splicing event types

rMATS pipeline part re-run with adjusted -- parameter

I checked the TSV file with the read length counts from the MultiQC report that the previously completed pipeline run had generated. It does indeed look like the median read length is 150 bp across all of the samples, with 518871962 reads from the total of 650708713 across all samples still being of this length after trimming (79%). This is why I set the --rmats_read_len parameter to 150. I first ran it without resuming the run that had successfully completed without SUPPA and soon realized that it was trying to re-trim all of the reads and re-align them, so I cancelled that run and added the "-resume d2143e7b-2e6a-41e4-850d-cb8af9596f82" to try to re-run the previous run. However, that gave me an error saying "Report file already exists: /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder/true -- enable the 'report.overwrite' option in your config file to overwrite existing files

" and then interrupted, so I understand that it considers the pipeline run completed and just wanted to rewrite the report, meaning it did not account for the change in the rMATS parameter. So I decided to continue today's run of the pipeline, 4718b1b9-5107-4741-a4b8-dd9b14854130. And whoops, got the same error again. So I set --report.overwrite=true ...which did not work, in any format whatsoever.

I could set the parameter to false in the nextflow.config files - I found many of them on my system, but also one in "/home/iweber/.nextflow/assets/nf-core/rnasplice/", which I suspect is the one the pipeline actually uses.

To be extra sure that this is the right config file, I accessed the nextflow.log files in my work directory, and they all have a line saying "Sep-04 11:07:50.453 [main] DEBUG nextflow.config.ConfigBuilder - Found config base: /root/.nextflow/assets/nf-core/rnasplice/nextflow.config", so I suspect that's the real location. I wanted to access it, and tried from the command line with sudo cd root/.nextflow/assets/nf-core/rnasplice, but was denied permission regardless. This is when I learned about how to open an elevated/root shell, in which I am indeed acting as the "root" user: just run sudo -i ! I then used the cd command without the sudo to get to the rnasplice folder, and accessed the nextflow.config file via nano.

Instead of changing the report.overwrite parameter to false, I simply changed the name that the report will have. It usually is

```
report {  
    enabled = true  
    file    = "${params.outdir}/pipeline_info/execution_report_${trace_timestamp}.html"
```

so I changed it to

```
report {  
    enabled = true  
    file    = "${params.outdir}/pipeline_info/execution_report_NR2${trace_timestamp}.html"
```

saved, and then exited the root shell with exit . I tried re-running:

```
cd /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder  
conda activate Nextflow  
  
sudo nextflow run nf-core/rnasplice -r 1.0.4 \  
--max_cpus 6 \  
--max_memory 90.GB \  
--multiqc_title "rMATS_run_multiQC" \  
--rmats true \  
--rmats_read_len 150 \  
--sashimi_plot false \  
--suppa false \  
--dexseq_dtu false \  
--edger_exon false \  
--dexseq_exon false \  
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclampsia_dataset_raw_and_processed/Pre_eclampsia_Nextflow_work_folder \  
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-  
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_rnasplice_results_rMATS \  
--input /home/iweber/Documents/Backup_shared_folder/Pre-  
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/samplesheet_preeclampnia.csv \  
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-  
eclampsia_dataset_raw_and_processed/Pre_eclampsia_mice_raw_fastq/contrastssheet_preeclampnia.csv \  
--fasta  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \  
--gff  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \  
--save_trimmed true \  
--aligner star_salmon \  
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \  
--salmon_index  
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/NCBI_Salmon_index_M_musculus_mm10 \
```

```
--save_unaligned true \
--save_align_intermeds true \
--save_reference true \
-profile docker \
-with-report true \
-resume 4718b1b9-5107-4741-a4b8-dd9b14854130
```

but it still didn't fix the issue, so I realized I have no time for this. I decided to start a fully new pipeline run while also kicking the Salmon run out, since this will not be needed for the rMATS output. I also created a new work folder to avoid any other collisions with the previous run of the pipeline.

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--multiqc_title "rMATS_run_multiQC" \
--rmats true \
--rmats_read_len 150 \
--sashimi_plot false \
--suppa false \
--dexseq_dtu false \
--edger_exon false \
--dexseq_exon false \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_rMATS \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplice_results_rMATS \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/samplesheet_preeclamp sia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--save_reference true \
-profile docker \
-with-report true
```

This indeed restarted the entirety of the pipeline, which ran through the STAR part and reached the Samtools sort part. However, this is where my virtual machine crashed, because of lack of space (it would seem it produced over 200 GB of results!). In the error message the machine showed me, related to the vmdk disk, I chose the option to power off the VM, as I saw from Windows Explorer that space was the issue (there were something like 45 KB left on the VM partition *sweat smile*). I went to the VM folder and saw that I had three-four older snapshots, each some 94 GB in size, so I went to the Snapshot Manager in VMware and attempted to delete the old snapshots from there. I left the last two snapshots in place, an auto-snapshot and one I had created manually on August 23rd.

Upon hitting "delete" on the old snapshots, it showed a progress bar for a fraction of a second, then gave me an error "An error occurred while consolidating disks: There is not enough space on the file system for the selected operation." After I hit ok, the snapshots vanished from the Snapshot Manager. I now had around 29 GB free on my drive. I noticed there is also a recycle bin folder outside of the VMware folder, and I emptied that one, leaving me with a total of 94 GB of space on the disk. I wanted to expand the virtual machine's disk size further, but it told me I cannot do that while there still are snapshots on the disk. So I went in to delete the last snapshots in the Snapshot Manager, only to now get an error "An error occurred while consolidating disks: The specified virtual disk needs repair."

However, I simply tried my luck and started the virtual machine, which... worked. I ran the log command for Nextflow, and found that my previous run had the hash 4b5f396a-a8f4-47b8-8a72-f78513d90127, so I used

```
sudo nextflow run nf-core/rnasplice -r 1.0.4 \
--max_cpus 6 \
--max_memory 90.GB \
--multiqc_title "rMATS_run_multiQC" \
--rmats true \
--rmats_read_len 150 \
--sashimi_plot false \
--suppa false \
--dexseq_dtu false \
--edger_exon false \
--dexseq_exon false \
-w /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_Nextflow_work_folder_rMATS \
--outdir /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_rnasplice_results_rMATS \
--input /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/samplesheet_preeclamp sia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff \
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--save_reference true \
-profile docker \
-with-report true
```

```

eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/samplesheet_preeclamp sia.csv \
--contrasts /home/iweber/Documents/Backup_shared_folder/Pre-
eclamp sia_dataset_raw_and_processed/Pre_eclamp sia_mice_raw_fastq/contrastssheet_preeclamp sia.csv \
--fasta
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.fna \
--gff
/home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/GCF_000001635.27_GRCm39_genomic.gff \
--save_trimmed true \
--aligner star \
--star_index /home/iweber/Documents/Backup_shared_folder/Genomes/genome_M_musculus/GCF_000001635.27/STAR_index_M_musculus \
--save_reference true \
-profile docker \
-with-report true \
-resume 4b5f396a-a8f4-47b8-8a72-f78513d90127

```

read outcomes check

Columns Breakdown:

- **USED**: The number of reads that were successfully used in the splicing analysis.
- **NOT_PAIED**: The number of reads that were discarded because they were not part of a properly paired-end read. In paired-end sequencing, both reads from a pair should map correctly for inclusion.
- **NOT_NH_1**: Reads that mapped to multiple locations in the genome (multimapped reads). These reads are often discarded since they do not provide unambiguous information about splicing.
- **NOT_EXPECTED_CIGAR**: Reads with an unexpected CIGAR string, indicating alignment issues. These reads may not support the expected splicing or exon-exon junctions.
- **NOT_EXPECTED_READ_LENGTH**: Reads that were discarded due to unexpected read length (should usually be 0, indicating consistent read length).
- **NOT_EXPECTED_STRAND**: Reads discarded due to unexpected strand orientation (may apply in strand-specific RNA-seq protocols).
- **EXON_NOT_MATCHED_TO_ANNOTATION**: Reads that mapped to exons but were discarded because they didn't match the provided annotation.
- **JUNCTION_NOT_MATCHED_TO_ANNOTATION**: Reads mapping to junctions that didn't match the annotation.
- **CLIPPED**: Reads with significant soft-clipping, where parts of the read weren't aligned (generally removed from analysis).
- **TOTAL_FOR_BAM**: The total number of reads in the BAM file.

Steps to Use This Information for Quality Check:

1. **Proportion of "USED" Reads**: The **USED** reads are the reads that were actually included in the rMATS splicing analysis. A high proportion of used reads relative to the total reads (given in **TOTAL_FOR_BAM**) suggests good data quality.

You can calculate the proportion of used reads as:

$$\text{Proportion used} = \frac{\text{USED}}{\text{TOTAL_FOR_BAM}}$$

If this value is low (e.g., below 80%), you may need to investigate the reasons why so many reads were discarded.

2. **Reasons for Discarded Reads**: Check the categories that account for reads that were discarded, such as:

- **NOT_PAIED**: If this number is high, it suggests issues with paired-end sequencing data.
- **NOT_NH_1**: A high count here suggests that many reads map to multiple locations, possibly due to repetitive regions in the genome.
- **NOT_EXPECTED_CIGAR** and **EXON_NOT_MATCHED_TO_ANNOTATION**: These categories could indicate issues with alignment quality or annotation mismatches. For example, if you're using an incorrect or incomplete annotation file, many reads might not match the expected exons or junctions.
- **JUNCTION_NOT_MATCHED_TO_ANNOTATION**: This could be a sign that the annotations are incomplete, or the splicing patterns are highly novel or uncharacterized.

3. **Comparing Across Samples**: You can compare the read outcome statistics across different BAM files (representing different replicates or conditions) to ensure consistency. Large discrepancies between samples may indicate differences in sequencing quality or alignment quality.

Visualizing the Results in R

To systematically check this information and make it easier to interpret, you can load the file into R and calculate the proportion of reads used, along with a breakdown of discarded reads across categories.

1. Load the File into R

Here's how you can read in the `read_outcomes_by_bam.txt` file and calculate the statistics:

```

library(tidyverse) # Load the data
read_outcomes <- read_delim("path/to/read_outcomes_by_bam.txt", delim = "\t", col_names = FALSE) # Reshape the data into a usable format # Assuming the file structure alternates between a BAM file name and corresponding stats
read_outcomes <- read_outcomes %>% mutate(group = rep(1:(n()/10), each = 10)) %>% # Each BAM file has 10 lines of stats

```

```

pivot_wider(names_from = X1, values_from = X2) %>% rename(BAM = BAM) # Convert relevant columns to numeric
read_outcomes %>% mutate(across(c(USED, NOT_PAIED, NOT_NH_1, NOT_EXPECTED_CIGAR, NOT_EXPECTED_READ_LENGTH, NOT_EXPECTED_STRAND, EXON_NOT_MATCHED_TO_ANNOTATION, JUNCTION_NOT_MATCHED_TO_ANNOTATION, CLIPPED, TOTAL_FOR_BAM), as.numeric)) # Calculate the proportion of reads used
read_outcomes <- read_outcomes %>% mutate(Proportion_Used = USED / TOTAL_FOR_BAM) # View the cleaned data
head(read_outcomes)

```

2. Check the Proportion of Used Reads

You can now visualize the proportion of reads used for each BAM file:

```

# Bar plot of the proportion of reads used across BAM files ggplot(read_outcomes, aes(x = BAM, y = Proportion_Used)) +
geom_bar(stat = "identity", fill = "blue") + theme_minimal() + labs(title = "Proportion of Used Reads Across BAM Files", x = "BAM File", y = "Proportion of Reads Used") + coord_flip() # Flip coordinates for easier reading if many BAM files

```

This will help you quickly identify whether certain samples have a low proportion of reads used.

3. Visualize Discarded Reads by Category

You can also visualize the number of discarded reads across various categories (e.g., not paired, multimapped, not matching exons/junctions):

```

# Reshape the data for plotting discarded reads by category read_discarded <- read_outcomes %>% select(BAM, NOT_PAIED, NOT_NH_1, NOT_EXPECTED_CIGAR, EXON_NOT_MATCHED_TO_ANNOTATION, JUNCTION_NOT_MATCHED_TO_ANNOTATION) %>% pivot_longer(cols = -BAM, names_to = "Category", values_to = "Count") # Plot discarded reads by category ggplot(read_discarded, aes(x = BAM, y = Count, fill = Category)) + geom_bar(stat = "identity", position = "stack") + theme_minimal() + labs(title = "Discarded Reads by Category Across BAM Files", x = "BAM File", y = "Number of Discarded Reads") + coord_flip() # Flip coordinates for easier reading

```

This stacked bar plot will help you see which types of discarded reads are most prevalent for each sample.

Summary of Quality Checks:

- **Proportion of Reads Used:** A high proportion of used reads indicates good quality data. You can set a threshold for what you consider acceptable (e.g., above 80% used).
- **Discarded Reads:** Check the reasons for discarded reads (e.g., unpaired, multimapped, CIGAR mismatch, not matching annotation). If a specific category accounts for many discarded reads, it may highlight issues with data quality, annotation, or alignment.
- **Comparing Samples:** Large differences in read usage across samples may indicate quality issues that need further investigation (e.g., library preparation issues, sequencing depth variation, alignment problems).

By using these insights from `read_outcomes_by_bam.txt`, you can ensure the quality of the data going into the rMATS analysis and better understand any potential sources of splicing noise or error.

read coverage check

p-value/FDR distribution check

Checking JCEC files with the actual splicing events

Example for SE data:

Imported JCEC file as tibble. The most relevant columns are, aside from the FDR, IncLevelDifference (this is the delta percent spliced in, dPSI, showing what the difference between the usage of an exon is between conditions), and IncLevel1 and IncLevel2, which show the inclusion values for control vs pre-eclampsia, each of them for the four replicates.

Many of the IncLevel1 and IncLevel2 observations contain NA values. Since rMATS needs a certain read coverage specifically across splice sites at the ends of exons to be able to calculate PSIs and dPSIs, not having enough coverage at these locations might be causing these results.

Checking SUPPA alternative splicing results

Interactive Visualizations with Plotly

Things I learned:

- Color mappings are mandatorily inherited from one Plotly trace in a plot to the ones that come after/above it. If wanting to force a trace to only use points of one color, use `color = l("color_of_choice")`
- Plotly plot objects are potentially not saved properly in the RDS format. Even though I could open them in the Shiny app.R script and they seemed okay, once I launched the app, it briefly flickered open and immediately crashed with an error message about the file not being accessible.
- Shiny processes working directories differently than what `getwd()` returns. Even though one can manually import certain plots from the script itself outside of

Shiny app

Citations

"If you use nf-core/rnasplice for your analysis please cite:

- The pipeline
<https://doi.org/10.5281/zenodo.8424632>
- The nf-core framework
<https://doi.org/10.1038/s41587-020-0439-x>
- Software dependencies
[https://github.com/nf-core/rnasplice/blob/master/CITATIONS.md"](https://github.com/nf-core/rnasplice/blob/master/CITATIONS.md)

Robinson, MD, McCarthy, DJ, Smyth, GK (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26, 139–140.

ACTIVELY WORKING ON

To dos

check how authors

- did their gene expression analysis
- filtered for lowly expressed genes —INFO NOT FOUND
and that's because they likely never did it and that's why they have fourfold increases/decreases for some genes —they have fields with 0 reads!
- filtered DEGs for FDR and fold change
- generated Z-scores for the heatmap no info on this in paper

find

- their result tables of DEGs, their P values, and Z scores
- RNA-Seq data from adult hippocampi

test

- expression levels and significance I find for the genes they highlight in figure 2G
- same for any other data they have in result tables

git

- change commit to exclude results and then add piecemeal before pushing to repo

change

- everything that's a data frame to a tibble
- everything that isn't Plotly or ggplot2 to one of them
- exploratory graphs to be nicer

make

- MA plots for both LFC shrinkage and non-LFC shrinkage data
- volcano plots for both LFC shrinkage and non-LFC shrinkage data

run pipeline

- adult hippocampi RNA-seq data

compare

- genes and splicing de-regulated in embryonic cortices vs same in adult hippocampi - any overlap?

Other useful Nextflow parameters

How Nextflow interprets strings in scripts:

<https://www.nextflow.io/docs/latest/script.html#string-interpolation>

basics:

1. double quotation marks, but not single, allow Nextflow to read values stored in variables, e.g. \$my_variable
2. using \ at the end of a line is a non-breakable space -> can be used to spread a command across multiple lines while Nextflow still interprets it as one single line
3. blocks of text spanning several lines can be put in triple quotation marks
4. it also, of course, supports regular expressions <https://www.nextflow.io/docs/latest/script.html#regular-expressions>

The names of parameters that Nextflow knows and that can be set in scripts or otherwise: <https://www.nextflow.io/docs/latest/script.html#regular-expressions> launchDir, projectDir

For parameters to be used in case one ever publishes a pipeline, even if only just on GitHub or some other repository:
<https://www.nextflow.io/docs/latest/config.html#scope-manifest>

Installing R and RStudio on the virtual machine, plus relevant packages

From <https://cran.r-project.org/>

```
# update indices
sudo apt update -qq
# install two helper packages we need
sudo apt install --no-install-recommends software-properties-common dirmngr
# add the signing key (by Michael Rutter) for these repos
# To verify key, run gpg --show-keys /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# Fingerprint: E298A3A825C0D65DFD57CBB651716619E084DAB9
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | sudo tee -a
/etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# add the R 4.0 repo from CRAN -- adjust 'focal' to 'groovy' or 'bionic' as needed
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"
```

I updated R Studio to the current version (rstudio-2024.04.2-764-amd64.deb), installed with sudo dpkg -i rstudio-2024.04.2-764-amd64.deb). I then installed the Bioconductur package manager with

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("tximport")
```

and chose *not* to update any packages it attempted to update. (I also installed DESeq2 by accident because it was in the same script). I also installed the R package `readr` using `pacman::p_install(readr)`, as this apparently accelerates `tximport` quite a lot.

To regenerate the tx2gene TSV file, ([install and use tximport](#) in R), I thought I'd need a so-called TxDb object:
<https://bioconductor.org/packages/release/bioc/vignettes/GenomicFeatures/inst/doc/GenomicFeatures.html>

As per the tutorial, I installed the package `GenomicFeatures` with `BiocManager::install("GenomicFeatures")`. I got some errors at the end of the process:

```
ERROR: dependency ‘png’ is not available for package ‘KEGGREST’
* removing ‘/home/iweber/R/x86_64-pc-linux-gnu-library/4.4/KEGGREST’
ERROR: dependency ‘Rhtslib’ is not available for package ‘Rsamtools’
* removing ‘/home/iweber/R/x86_64-pc-linux-gnu-library/4.4/Rsamtools’
ERROR: dependency ‘Rsamtools’ is not available for package ‘GenomicAlignments’
* removing ‘/home/iweber/R/x86_64-pc-linux-gnu-library/4.4/GenomicAlignments’
ERROR: dependencies ‘XML’, ‘RCurl’ are not available for package ‘restfulr’
* removing ‘/home/iweber/R/x86_64-pc-linux-gnu-library/4.4/restfulr’
ERROR: dependency ‘KEGGREST’ is not available for package ‘AnnotationDbi’
* removing ‘/home/iweber/R/x86_64-pc-linux-gnu-library/4.4/AnnotationDbi’
ERROR: dependencies ‘XML’, ‘Rsamtools’, ‘GenomicAlignments’, ‘restfulr’ are not available for package ‘rtracklayer’
* removing ‘/home/iweber/R/x86_64-pc-linux-gnu-library/4.4/rtracklayer’
ERROR: dependencies ‘AnnotationDbi’, ‘rtracklayer’ are not available for package ‘GenomicFeatures’
* removing ‘/home/iweber/R/x86_64-pc-linux-gnu-library/4.4/GenomicFeatures’
```

License

<http://creativecommons.org/licenses/by/4.0/> probably best, version with attribution.