

INFO-F-527 INTERNSHIP FINAL REPORT

This document should be filled by the student, signed by the student and the host company responsible and returned to the MBM Internship Coordinator (Pr. G. Bontempi) at the latest one week after the end of the internship.

STUDENT'S COORDINATES

Family name	Renson
First name	Olivier
Academic year	2020-2021
Matricule	503063
Address (and telephone number)	Priespré, 1 4550 Villers-le-Temple 0479265782
Email	olivier.renson@ulb.be

HOST COMPANY'S COORDINATES

Company name	Université de Liège / CHU de Liège
Official address	Rue de l'Hopital, 1 4000 Liège
Supervisor	1. Alice Mayer 2. Leonor Palmeira
Role in the company	1. Bioinformatics Project Manager at GIGA Research Centre 2. Bioinformatics Project Manager at Human Genetics Department (CHU)
Telephone	1. 04 366 45 42 2. 04 366 91 41
E-mail	1. Alice.mayer@uliege.be 2. lpalmeira@chuliege.be

INFO-F-527 INTERNSHIP FINAL REPORT

DESCRIPTION OF THE ACTIVITIES DONE

TITLE OF THE INTERNSHIP: Investigation of methylation tools in the context of Nanopore sequencing

DESCRIPTION OF THE DOMAIN OF ACTIVITY OF THE HOST COMPANY (max half page)

Established in 2007 at the University of Liège, GIGA is an interdisciplinary research center in biomedical sciences whose mission is advanced medical innovation. The institute encompasses more than 500 members (PI, senior researchers, post-doctoral scientists, thesis students, technicians) with expertise in medical genomics, in silico medicine, neuroscience, oncology, infection and immunity, and cardiovascular sciences. GIGA provides its members access to a broad range of state-of-the-art technologies through core facilities, including genomics, proteomics and imaging platforms (<http://www.giga.uliege.be/>). The CHU Liège is a University Hospital active in all the aspects of health care, including human genetics, oncology, and microbiology. Together the GIGA Research Center and the CHU launched a genomic medicine initiative. This initiative will set up the technological, scientific and medical bases for clinical application of genomics in the fields of genetic diseases, cancer diagnosis and follow-up, and infectious diseases.

INTERNSHIP CONTRIBUTION (max 15 pages) including

- INTRODUCTION

The internship was inscribed in the framework of the genomic medicine initiative hosted by CHU Liège [0] and HRDinONE Télévie project aiming at the simultaneous detection, by **Nanopore sequencing**, of several complex genetical biomarkers linked to homologous recombination deficiency [1]. **Promoter methylation** of genes in the homologous recombination pathway are one of those genetical biomarkers and can be inferred through bioinformatics tools based on **artificial intelligence** and **probabilistic models**. Nanopore sequencing, allows to use raw DNA without the need for any chemical treatment or PCR amplification in comparison to other techniques. By leaving the DNA untouched it is then possible to detect **5-methylcytosines** (5mc) by spotting small changes in the electrical signal of the MinION sequencer [2].

Nanopolish [2] is a software package for the processing and analysis of sequencing data obtained from Oxford Nanopore technology. After the basecalling is done using **Guppy** (proprietary software) [3] or any available tool, methylation events can be detected by doing some **alignment** stuff **in the signal space**.

INFO-F-527 INTERNSHIP FINAL REPORT

However, working in such a signal space is a **computationally complex task** that was solved by a state of the art algorithm based on **hidden Markov model**. The latter was trained with synthetic DNA data to finally be able to differentiate methylated bases from unmethylated ones.

F5c [4] is a **re-implementation** of some modules from Nanopolish in a way that **optimize calculus parallelization** on CPU (multi-threading with POSIX threads, referred as f5c **cpu-opti**) and allows usage of **Nvidia GPU acceleration** (using Cuda, referred as **f5c-cuda**). As shown in [4], f5c is theoretically capable to perform **3 to 5 times faster** than Nanopolish by taking advantage of heterogeneous CPU/GPU architectures (Figure 0). However, for now, f5c only allows usage of 1 GPU at a time, which does not allow us to fully benefit from the cluster's multi-GPU nodes.

- DESCRIPTION OF THE OBJECTIVES & TASKS

The initial and main objective was to **explore methylation tools** (mainly Nanopolish and f5c) that works with Nanopore data, optimize their performances and compare them in terms of **efficiency** (identify computation bottlenecks, find best software/hardware settings) and in terms of **biological meaning** using the default Nanopore dataset.

Then, a second objective we added along the way was to test those algorithms on a **custom dataset (MCF7 cells)**, compare our results with **bisulfite sequencing** (current gold standard for methylation analysis) and see if it is consistent with what we observed previously with the default Nanopore dataset.

- METHODOLOGY USED TO SOLVE THE TASKS

A) The first step was to compare performances of the 3 methylation-calling algorithms (Nanopolish, f5c cpu-opti, f5c Cuda). We used the dragon2 cluster from CECI to run all jobs as it is the only one to provide GPU acceleration (2 nodes with 2 Tesla V100, 2 12-Core Intel Xenon Gold 6126 and 192 Go of RAM). The workflow we followed is detailed in [5] and globally consist of detecting methylated bases and comparing the results obtained with the "bisulfite" gold standard method. We used the \$LOCALSCRATCH as working directory to take profit from its fast RAID0 ssd architecture and get the best possible I/O performances. Then, in order to evaluate the execution time of each step of the process, we simply added the following command line between each statement:

```
echo Current time : $(date +"%T")
```

Each implementation of the algorithm was tested with allocation of 1, 2 and 4 CPUs (we have set the -t option using the formula: #CPU * #Core). The f5c Cuda version required additional parameter tuning to optimally balance the work load between CPU and GPU. Indeed, the GPU is very efficient to align enormous quantity of small reads in parallel while, within the same period of time, the CPU will handle the few

INFO-F-527 INTERNSHIP FINAL REPORT

longer reads. Thus, depending on the hardware specifications (i.e CPU frequency, GPU memory, ...) and on the dataset of read used (length distribution), parameter must be tuned (following guidelines in [7]) in order to obtain the best possible performances. For that, the log file outputted by Cuda (Figure 1) is very useful to get information about how the work load is balanced. We also monitored the GPU activity with "nvidia-smi" command to see how much it was involved in the computations (Figure 2). We finally reported both the global execution time and the specific duration of the methylation calling step in Table 1 in the result section.

B) In the previous step, we found that, by taking advantage of heterogeneous CPU/GPU architectures, f5c Cuda was the fastest tool for methylation detection. It was tested on a GPU node of Dragon2 (Tesla v100 with 16 Go of dedicated memory) and our conclusion was that the algorithm seems to be mainly memory bounded and, to a lesser extent, limited by I/O processes. Thus we found interesting to test it on GPU with 32 Go memory to see if that was really the limiting factor. We kept using the same dataset and the same workflow described in [5]. Unfortunately, Alan cluster (Montefiore ULiège, Tesla v100 with 32 Go of dedicated memory) does not provide an equivalent to the \$LOCALSCRATCH directory of dragon2 we used previously. So we have used the global scratch decentralized directory which may theoretically be slower for I/O processes. Other methods are the same as those described in the previous step. We also decided to run f5c on a standard workstation to check the kind of performances that can be achieved on a simpler hardware architecture. To do so, we used the "taskset" command to allocate to our job only the desired quantity of CPUs among the 20 available and "htop" to visualize CPU usage during the job execution (Figure 4). We tuned parameters depending on the amount of memory available so that the balance between CPU and GPU was optimal and finally reported our results in Table 2.

C) In this third step we further investigate f5c performance by testing some new ideas to fasten calculations and trying to figure out why the workstation gave so good results compared to clusters. As the workstation is equipped with 20 CPUs we wanted to see what kind of performance can be achieved by requesting more than just 4 of them and if it has an impact on the GPU usage. Then we also monitored some potential working directory and their file system speed with:

```
$ df -Th $WORKINGDIR
```

```
$ dd if=/dev/zero of=$WORKINGDIR/test1 bs=1M count=2048 oflag=dsync
```

Finally, we tested the impact of using single- or multi-fast5 files. The original dataset uses single-fast5 so we used the official Nanopore ont-fast5-api [8] to convert it into multi-fast5.

D) So far, we worked with Nanopore data already basecalled. This step aims at converting the MinION electrical signal into DNA bases. Guppy [3], the official Nanopore basecaller, also has an GPU implementation that we wanted to test. So we decided to download raw Nanopore data in order to compare Guppy CPU and GPU

INFO-F-527 INTERNSHIP FINAL REPORT

performances on a dataset of 10 multi-fast5 files with 4000 reads each. Results are reported in Table 5.

E) Finally, we wanted to test f5c on a custom dataset to see if it gave relevant results. We worked with MCF7 cells (breast cancer cell line) whole genome sequencing data (30x depth, 1.5 To). They were also treated using bisulfite sequencing (gold standard for methylation detection) with a focus on 5 regions of interest in the following genes: SEMA3B, RASSF1, KLHL6, INA, PTPRCAP.

Genes	Chr	Start	End	Length	CpG	Function(s)
SEMA3B	3	50275437	50275514	77	6	Inhibits axonal extension, tumor suppressor by inducing apoptosis
RASSF1	3	50340943	50341036	93	6	Tumor suppressor, negatively regulate cell cycle at G1/S-phase
KLHL6	3	183555418	183555536	118	5	B-lymphocyte antigen receptor signalisation
INA	10	103276779	103276913	135	8	Type IV intermediate filament heteropolymers
PTPRCAP	11	67437695	67437765	70	2	T- and B-lymphocyte activation (transmembrane phosphoprotein)

In order to reduce the amount of data to transfer to dragon2 we decided to extract only chromosomes that contains regions of interest (i.e chr 3, 10 and 11). We processed as described on Figure 9 to obtain the inputs files required to build the index and call the methylation (rawSignal.fast5, reads.fastq, sorted.bam, ref.fasta). That enabled us to respectively reduce the total amount of data to only 125, 72 and 55 Go for chromosomes 3, 10 and 11. Then, the methylation was obtained with the following 2 commands (you may need to add/remove "chr" before the chromosome id line in the fa file depending on the annotations used in the bam file):

```
$ f5c_linux_cuda index -t 8 --iop 8 -d fast5_files reads.fastq
$ f5c_linux_cuda call-methylation -r reads.fastq -b
sorted_alignment.bam -g ref.fa -B 8M -K 1024 -t 8 > meth_out.tsv
```

And compute frequencies at each CpG position with:

```
$ f5c_linux_cuda meth-freq -i meth_out.tsv -s > meth_freq.tsv
```

We executed those jobs on dragon2 in order to profit from its Tesla v100 GPU and requested 4 CPUs for the computations. Without precisely tuning parameters it took from around 30 min (chr. 10 and 11) to more than an hour (chr. 3) to obtain the results. Then, we simply used "grep" to extract data about CpGs that are located in our regions of interest. We are now ready to compare the methylation frequencies with the results obtained by bisulfite sequencing (Table 6).

INFO-F-527 INTERNSHIP FINAL REPORT

Workstation hardware specifications:

- Intel Core i9-9900X (3.50GHz, 10 cores)
- Nvidia GTX 1660 Ti (6 Go RAM)
- 233 Go ssd

Dragon2 cluster hardware specifications:

- Intel Xenon Gold 6142 (2.60GHz, 12 cores)
- Tesla v100 (16 Go RAM)
- 40 Go \$Home, 44 Go \$GLOBALSCRATCH, 100 Go \$CECIHOME

Alan cluster hardware specifications:

- Intel Xenon Gold 6248 (2.50GHz, 20 cores)
- Tesla v100 (32 Go RAM)
- 65 To hdd decentralized global scratch

- RESULTS

A) Comparing methylation tools: see Figure 3 for barplot of the execution time and biological comparison.

Global – meth call	Nanopolish	F5c (cpu-opti)	F5c (cuda)
1 cpu	28m 43s – 26m 31s	24m 5s – 21m 48s	8m 59s – 6m 56s
2 cpu	15m 18s – 13m 39s	12m 54s – 11m 7s	7m 22s – 5m 28s
4 cpu	9m 18s – 7m 50s	6m 40s – 5m 27s	4m 18s – 2m 51s

Table 1: execution time for each implementation using successively 1, 2 or 4 CPUs. Expressed in the form (global execution time – methylation step duration).

B) How well does f5c perform on different hardware architectures:

Global – meth call	Workstation	Dragon2	Alan
1 cpu	xxx – 5m 49s	8m 59s – 6m 56s	6m 19s – 4m 38s
2 cpu	xxx – 3m 04s	7m 22s – 5m 28s	6m 06s – 4m 25s
4 cpu	xxx – 1m 43s	4m 18s – 2m 51s	3m 59s – 2m 47s

Table 2: execution time of f5c-Cuda for each hardware configuration using successively 1, 2 or 4 CPUs. Expressed in the form (global execution time – methylation step duration).

C) Requesting more CPUs: see Figure 5 and 6 for GPU monitoring on workstation. Testing different file systems (Figure 7) and input file format:

methylation call	Dragon2	Alan	Workstation
\$GLOBAL	2m 51s – 0.52	2m 47s – 1.1	/
\$HOME	2m 54s – 0.45	2m 45s – 0.59	1m 46s – 0.42
\$LOCAL	2m 54s – 0.39	/	/
/tmp	2m 51s – 0.39	2m 28s – 0.54	1m 44s – 0.4
/dev/shm	2m 52s – 0.39	2m 33s – 2m 53	1m 44s – 0.40

Table 3: monitoring of some available working directories speed using 4 CPUs. (Expressed in the form “methylation calling duration”– “loading time/processing time” ratio).

INFO-F-527 INTERNSHIP FINAL REPORT

Total – meth call	Workstation	Dragon2	Alan
Single fast5	2m 16s – 1m 43s	2m44s – 1m 54s	4m 01s – 2m 29s
Multi fast5	2m 17s – 1m44s	2m 47s – 1m 57s	3m 30s – 2m30s

Table 4: testing single vs multi-fast5 input files

D) Guppy basecalling:

/tmp	Guppy CPU	Guppy GPU
Dragon2 (Tesla 16Go)	5h 25m 52s	31s
Workstation (GeForce GTX 1660 Ti 6 Go)	~ 5h 30m 45s	26m 45s

Table 5: Guppy CPU and GPU run on different setup using the same dataset

E) MCF7 methylation frequency computation using f5c and comparison with bisulfite sequencing.

Chr	Gene	CpG Position	Bisulfite freq	F5c freq
3	SEMA3B	50,275,437	0.91	1.00
3	SEMA3B	50,275,465	0.78	0.80
3	SEMA3B	50,275,471	0.85	0.80
3	SEMA3B	50,275,482	?	0.90
3	SEMA3B	50,275,485	?	0.90
3	SEMA3B	50,275,496	?	0.84
3	RASSF1	50,340,976	0.62	0.94
3	RASSF1	50,340,982	0.63	0.94
3	RASSF1	50,340,992	0.62	0.94
3	RASSF1	50,340,994	0.60	0.94
3	RASSF1	50,341,000	0.61	0.94
3	RASSF1	50,341,014	0.53	0.83
3	KLHL6	183,555,446	0.85	0.86
3	KLHL6	183,555,456	0.92	1.00
3	KLHL6	183,555,498	0.94	1.00
3	KLHL6	183,555,600	0.95	1.00
3	KLHL6	183,555,603	0.93	1.00
10	INA	103,276,801	0.89	0.82
10	INA	103,276,834	0.94	0.89
10	INA	103,276,854	0.93	1.00
10	INA	103,276,872	0.79	1.00
10	INA	103,276,888	0.96	1.00
10	INA	103,276,892	?	1.00
10	INA	103,276,903	?	1.00
10	INA	103,276,912	?	1.00
11	PTPRCAP	60,437,695	0.89	1.00
11	PTPRCAP	60,437,724	?	0.92

Table 6: methylation frequency comparison between bisulfite and f5c

INFO-F-527 INTERNSHIP FINAL REPORT

- DISCUSSION OF THE RESULTS WITH RESPECT TO THE OBJECTIVES

A) Even though they said in [4] that they had not changed the biological part of the algorithm between Nanopolish and f5c, we were able to spot small differences in the output (see Figure 3). However, those differences due floating point approximation are very small and seems to be negligible. In any case, the correlation with bisulfite sequencing data stay strong. About the performances, when using only one CPU, f5c-Cuda performed around 3.5x faster than Nanopolish. This ratio was a little bit decreased when using 4 CPU and fall around 2.5. So in any case where a GPU is available, f5c-Cuda can provide significantly better performances than Nanopolish. If your infrastructure does not provide GPU access it is still recommended to use f5c cpu-opti as it also outperformed Nanopolish in all the situations. From what we saw during the GPU monitoring (f5c-Cuda, Figure 2), nearly all memory is used but GPU activity remains at 0% most of the time. This may be partly due to I/O processes that seems to take around 1/3 of the time. A second hypothesis may be that our GPU are maybe too powerful for this task with only 16 Go of dedicated cache memory. Conclusions about GPU monitoring is that f5c-Cuda is memory bounded and that it may be interesting to run it on setup with 32 Go to see if we can achieve better performances.

B) Overall, the f5c-Cuda implementation running on 32 Go GPU is giving best performances in all categories in comparison to what we obtained with the 16 Go of dragon2. By doubling the memory allocated to the GPU and using the slower file system of Alan, the limiting factor seemed to have changed from the amount of memory available to the I/O processes. Now, it takes indeed longer to load the data than it does to process it while this ratio was about 1/3 before. We were able to obtain a significant gain in time by using GPUs with 32 Go, especially when using only one or two CPUs. When the number of CPU increase, the advantage of having 32 Go became lower. This is due to the fact that the work load is mainly balanced toward the GPU in that case and thus cannot profit from an increase in CPU number to fasten calculations. This is the opposite when using a 16 Go GPU or a CPU-only algorithm that rely more/only on the CPU and therefore can be speed-up by increasing the quantity of CPUs involved in the operation. We were also quite impressed by the good results obtained with the workstation despite its lower end GPU and the only plausible justification we found is that all computations were happening locally and therefore speeding up I/O processes. Another interesting characteristic of the workstation are its CPUs that are much more performant than those in Alan and CECI cluster.

C) Surprisingly, requesting more CPU (Figure 5) on the workstation gave the best performances we ever had. From what we saw on dragon2 and on Alan, one of the main limitation to the GPU seems to be its memory. So here, with only 6 Go, we

INFO-F-527 INTERNSHIP FINAL REPORT

were expecting low performances. At first, with only one working CPU, we got something very similar to the pattern observed on the 2 clusters. As memory usage was already at 90%, we first thought that it was not possible to get more out of the GPU. In contradiction to our expectations, involving more CPUs in the process boosted the GPU computations (Figure 6) showing that finally the memory might not be the limiting factor. Our hypothesis is that each CPU working in parallel delegate some work to the GPU and so by involving more CPU we are able to delegate more work to the GPU. This shows well that f5c is a hybrid software that depend not only on the GPU but also on the CPU. About the different file locations that we checked (Figure 7), /dev/shm was always the fastest one thanks to its temporary file system (tmpfs). It enables copy of files directly into the RAM memory while /tmp and \$LOCALSCRATCH are usually in xfs or ext4 and hence slower according to what is shown on Figure 8.

D) Unlike f5c which relies on both the CPU and GPU, Guppy is a 100% GPU software. That allows massive calculus parallelization and leads to an astonishing speed up of respectively 631x and 12.3x using dragon2 and the workstation.

E) Unfortunately the report on bisulfite sequencing does not contain the frequency for all CpGs so we had to exclude some of them from the comparison (marked as "?" in Table 6). We previously saw that the f5c output correlate quite well with bisulfite sequencing using the Nanopore dataset (Figure 10 left). However, due to the relatively small amount of CpG in our regions of interest that kind of heatmap is not really relevant in our case (Figure 10 right). If we use a simpler violin plot to visualize the distribution, we can see that bisulfite sequencing gives slightly lower and more scattered (between 0.6 and 1) methylation frequencies while f5c outputs higher and more "radical" values around 1.

- CONCLUSIONS

F5c is a **hybrid** software that relies on both the **CPU(s) and GPU**. It is not useful to have a very performant GPU on a simple architecture but we rather recommend to work with a **balanced hardware setup** equipped with powerful CPUs and an average GPU in support (cfr workstation). We also want to highlight the **importance of I/O** processes that can easily become a major bottleneck if not properly monitored and taken into account when choosing a path for storing the dataset. Overall we succeeded in **speeding up** the methylation calling process by a factor of around **4.5 using GPU acceleration**. We also monitored f5c resources usage in order to highlight bottlenecks and find the most appropriate environment for this algorithm. It enables us to reduce again the computation time by a factor of 2 in the best conditions. Finally, we have shown that using multi-fast5 files with small datasets is not mandatory and does not speed up the methylation call at all. From a **biological point of view**, f5c outputs slightly differ from those of Nanopolish due

INFO-F-527 INTERNSHIP FINAL REPORT

to floating point approximations. However, those **changes are minor** and does not seems to be significant.

We succeeded in **splitting MCF7 dataset** and transferring them on dragon2 using basic bioinformatics tools like **bash**, **samtools** and **ont-fast5-api**. Then, we have **extracted methylation data** from it **using f5c**. One drawback of f5c we would highlight is the large amount of disk space required to store (multi-)fast5 files that are used to make an alignment in the signal space. Even if our **results are in the same direction that** the ones obtained by **bisulfite sequencing**, it is difficult to draw any conclusion from the comparison given the relatively small amount of data taken into account in our analysis. Deciding which technique is closer to the biological reality is a complex task that require further investigations. Finally, one thing is certain: methylation calling on Nanopore data allows to **bypass the limitations of bisulfite sequencing** which are incomplete conversion (very specific conditions are required to maintain the DNA in a single- stranded conformation), degradation during treatment (conditions necessary for complete conversion can lead to the degradation of about 90% of the incubated DNA) and no discrimination of 5mC from 5-hydroxymethylcytosine (5hmC).

- REFERENCES

0. https://www.gigauhg.uliege.be/cms/c_4738041/en/gigauhg-genomic-medicine
1. <https://www.nature.com/articles/s41586-020-1943-3>
2. <https://www.nature.com/articles/nmeth.4184>
3. <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1727-y>
4. <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03697-x>
5. https://nanopolish.readthedocs.io/en/latest/quickstart_call_methylation.html
6. <https://www.sciencedirect.com/science/article/pii/S2352914817300916>
7. <https://hasindu2008.github.io/f5c/docs/f5c-perf-hints>
8. https://github.com/nanoporetech/ont_fast5_api

PEDAGOGICAL CONTENT OF THE INTERNSHIP (in terms of soft skills (e.g. learning how to make a presentation) and technical skills (e.g. learning of a new tool, language, operating system)) (max half page)

- Usage of high performance computing clusters
- GPU computing and monitoring
- Methylation detection tools: Nanopolish, f5c, bisulfite sequencing
- Basecalling tool: Guppy
- Presentation and discussion of results
- Docker/Singularity container
- Introduction to spatial transcriptomic & single cell RNAseq

INFO-F-527 INTERNSHIP FINAL REPORT

DETAILED ACTIVITIES CALENDARS OF THE INTERNSHIP

(set of items of the form DD/MM/YY : Activity name, what ? where ?)

18/02/21: First day of the internship, meeting with the team, paperwork, presentation of the project, state of the art.

19/02/21: CECI training session, more state of the art, installation samtools, deepsignal, deepmod, f5c and nanopolish

23/02/21: GIGA cluster and massstorage connection, first jobs, tuto nanopolish, frequency plots in R

25/02/21: f5c cpu-opti and cuda run, selection of best parameters, test on multiple cpu

26/02/21: gpu parameter selection and monitoring, results interpretation, pdf report

02/03/21: pdf report, diapo, presentation of the results, direction for future work, Docker install

04/03/21: Montefiore contact for gpu, Docker tuto, Guppy cpu/gpu installation

05/03/21: Contact with microscopy, Montefiore account setup, Guppy first tests

09/03/21: Guppy tests on gpu, test f5c on Alan cluster, repport, microscopy department lookup

11/03/21: repport end, soft installation on worksatation, guppy & f5c tests

12/03/21: Workstation runs, covid conference, paper club

16/03/21: Dias, workstation cpu/gpu monitoring, docker to singularity attempt

17/03/21: Dias end, guppy gpu monitoring, meeting, single to multifast5 conversion

18/03/21: Paper gpu, test multi fast5

23/03/21: Testing multifast5 and new working directories (/tmp, /dev/shm)

25/03/21: More tests with /tmp and /dev/shm, 20 cpu on alan, file systems speed monitoring

26/03/21: End report, Journal club

30/03/21: Slides and presentation preparation, guppy monitoring on workstation

01/04/21: End of slides, meeting to discuss results and perspectives, mcf7 data download

02/04/21: Bisulfite state of art and mcf7 data exploration

06/04/21: Familiarization with mcf7 nanopore data

08/04/21: Nanopore data analysis and chromosome extraction

INFO-F-527 INTERNSHIP FINAL REPORT

09/04/21: HDF5 download and install, data transfer, read count, split scripts

13/04/21: Split fastq and fast5

15/04/21: End of fast5 splitting

16/04/21: Visite infrastructure avec nouveau candidat, data transfer to dragon2

20/04/21: Run f5c on mcf7 chr10 data, genome annotation correction, meeting NGS

21/04/21: Run f5c on mcf7 chr11 data, Nvidia talk

22/04/21: Run f5c on mcf7 chr3 data, Nvidia talk

27/04/21: Methylation data analysis, SEGI (cluster, massstorage, CECI) visit

29/04/21: Results presentation, test with split option

30/04/21: MCF7 negative control, methylation distribution plot

04/05/21: f5c output and bisulfite comparison, stats and plots

05/05/21: Bisulfite sequencing end of report

06/05/21: Internship final report, webinar genomics platform

10/05/21: Internship final report

11/05/21: Single cell RNAseq

12/05/21: Meeting preparation, closing meeting and discussion

13/05/21: Ascension

14/05/21: Internship final report

INTEGRATION OF THE STUDENT IN THE HOST ORGANIZATION (max 10 lines)

I was very well integrated in the team from the first day and during all the duration of the internship. The fact that I was allowed to physically come to work at the office when I wanted to was a huge plus in these times of massive remote work. A meeting the first day with the team gave me a nice overview of the situation and objectives of the internship. Then we met on regular basis to share my results and discuss what happens next. Working together in a kind of "open-space" makes contacts with colleagues easy and questions quickly answered.

CONCLUSION AND PERSPECTIVES (1 page)

Firstly, from a purely computational standpoint, the comparison between Nanopolish and f5c showed that a significant speed-up of the methylation call can be obtained by spreading computations across CPU and GPU. Indeed, GPU are built in a way that optimize calculus parallelization and more and more tools are now being

INFO-F-527 INTERNSHIP FINAL REPORT

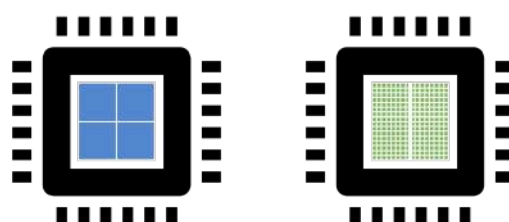
implemented (if possible) to take profit from that kind of hardware architecture. In that context, our results give a nice insight of the type of setup best suited for f5c and the performances that are reachable using GPU. That will be helpful to make the balance between the money to invest in a perspective of cluster upgrade (creating GPU nodes) and the potential gain in computation time.

Then, in biological terms, Nanopore technology enable portable, fast and easy sequencing with on the fly basecalling and methylation calling which makes sequencing data more accessible than ever and allows to bypass bisulfite sequencing limitations. Given the importance of methylation in a variety of biological processes (mismatch repair, gene expression, embryonic development, B-cell differentiation, X-chromosome inactivation, ...), its detection represents an attractive diagnostic and therapeutic target. Indeed, the impact of methylated CpG at certain positions on genes implied in the homologous recombination pathway is still under study and seems to be one of the promising DNA signature for cancer diagnosis and healing.

REMARKS (if any)

Everything went very well in a friendly atmosphere. Flexible working schedule and possibility to work remotely depending on my course and master thesis planning.

SUPPLEMENTARY MATERIAL (including images) (max 5 pages)



CPU	GPU
Central Processing Unit	Graphics Processing Unit
4-8 Cores	100s or 1000s of Cores
Low Latency	High Throughput
Good for Serial Processing	Good for Parallel Processing
Quickly Process Tasks That Require Interactivity	Breaks Jobs Into Separate Tasks To Process Simultaneously
Traditional Programming Are Written For CPU Sequential Execution	Requires Additional Software To Convert CPU Functions to GPU Functions for Parallel Execution

Figure 0: CPU vs GPU

INFO-F-527 INTERNSHIP FINAL REPORT

```
[set_opt_profile] max-lf: 5.0, avg-epk: 2.0, max-epk: 5.0, K: 1024, B: 10.0M, t: 32, ultra-thresh: 100.0k, iop: 64
[init_iop] Spawning 64 I/O processes to circumvent HDF hell
[init_cuda] Running on Tesla V100-PCIE-16GB (device id 0)
[cuda_freemem] 15.42 GB free of total 15.78 GB GPU memory
[init_cuda] Max GPU capacity 9.2M bases
[meth_main::2.335+0.71] 1024 Entries (8.0M bases) loaded
[process_db::5.042+2.25] Events computed
[meth_main::8.073+2.49] 1024 Entries (7.9M bases) loaded
[align_cuda] Load : CPU 31 entries (0.7M bases), GPU 993 entries (7.3M bases)
[load_balance] Processing time : CPU 3.1 sec, GPU 1.8 sec
[process_db::8.190+2.47] Banded alignment done
[process_db::8.794+2.55] Scaling calibration done
[process_db::14.392+3.10] HMM done
[pthread_processor::14.392+3.10] 1024 Entries (8.0M bases) processed
[process_db::17.043+3.21] Events computed
[align_cuda] Load : CPU 44 entries (0.5M bases), GPU 980 entries (7.4M bases)
[load_balance] Processing time : CPU 2.7 sec, GPU 2.0 sec
[process_db::19.722+3.19] Banded alignment done
[meth_main::20.408+3.18] 1024 Entries (7.5M bases) loaded
[process_db::20.423+3.18] Scaling calibration done
[process_db::24.232+3.31] HMM done
[pthread_processor::24.232+3.31] 1024 Entries (7.9M bases) processed
[process_db::26.475+3.36] Events computed
[align_cuda] Load : CPU 32 entries (0.4M bases), GPU 992 entries (7.1M bases)
[load_balance] Processing time : CPU 1.8 sec, GPU 1.4 sec
[process_db::28.250+3.39] Banded alignment done
[process_db::29.082+3.37] Scaling calibration done
```

Figure 1 : f5c Cuda output log file



Figure 2 : GPU monitoring with nvidia-smi during f5c Cuda run

INFO-F-527 INTERNSHIP FINAL REPORT

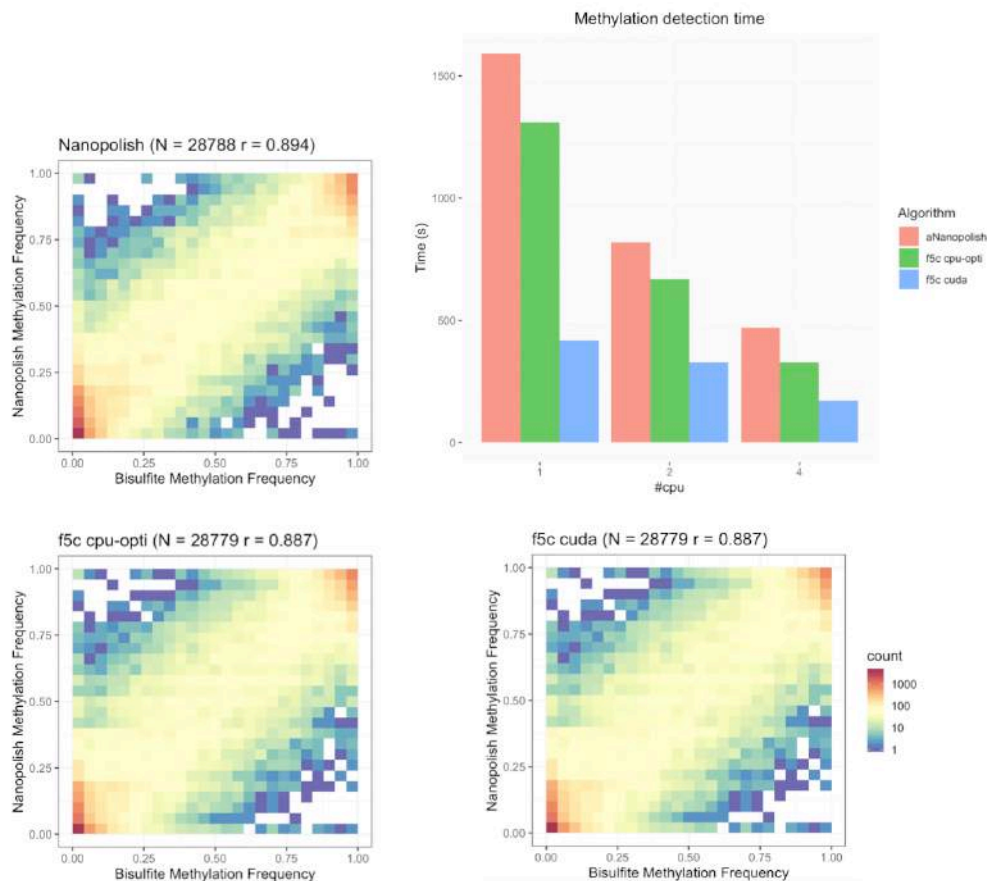
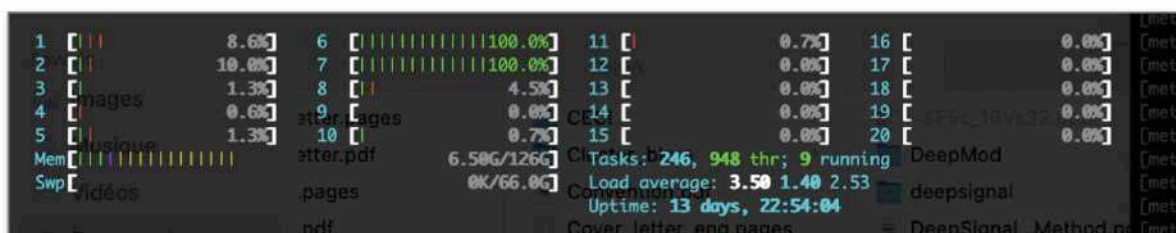
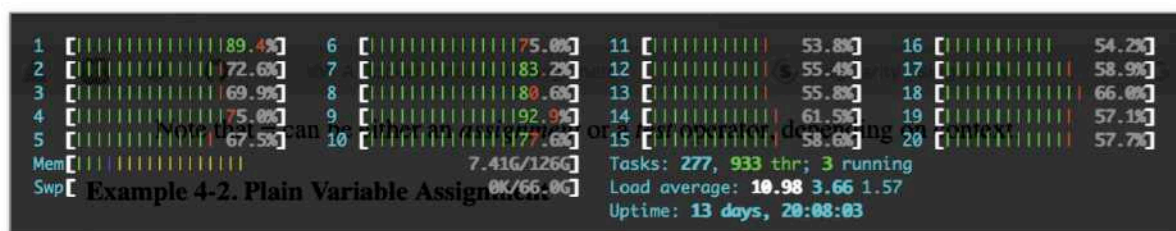


Figure 3 : (upper right) comparison of execution time of the methylation calling step from Table 1. (3 others) biological meaning comparison.

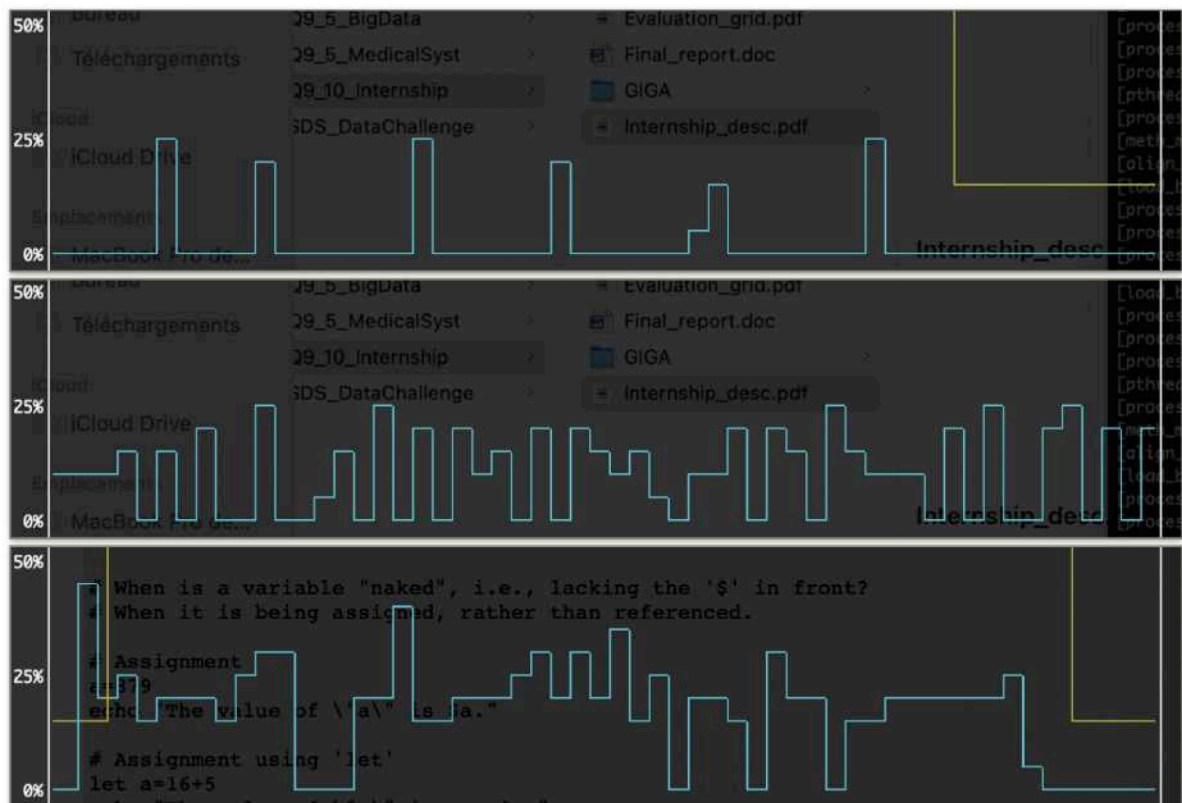


Picture 4 : workstation monitoring (htop) with restricted CPU allocation using "taskset -c 6,7"

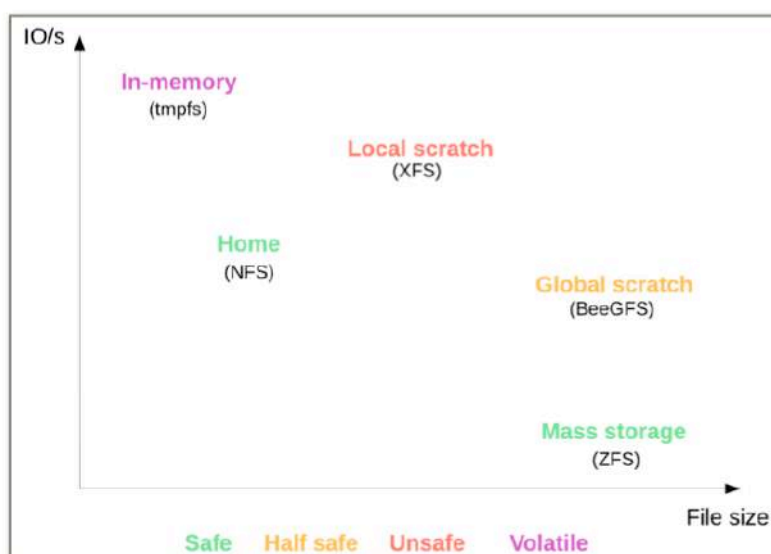


Picture 5 : workstation monitoring (htop) without restrictions, 20 working CPUs

INFO-F-527 INTERNSHIP FINAL REPORT

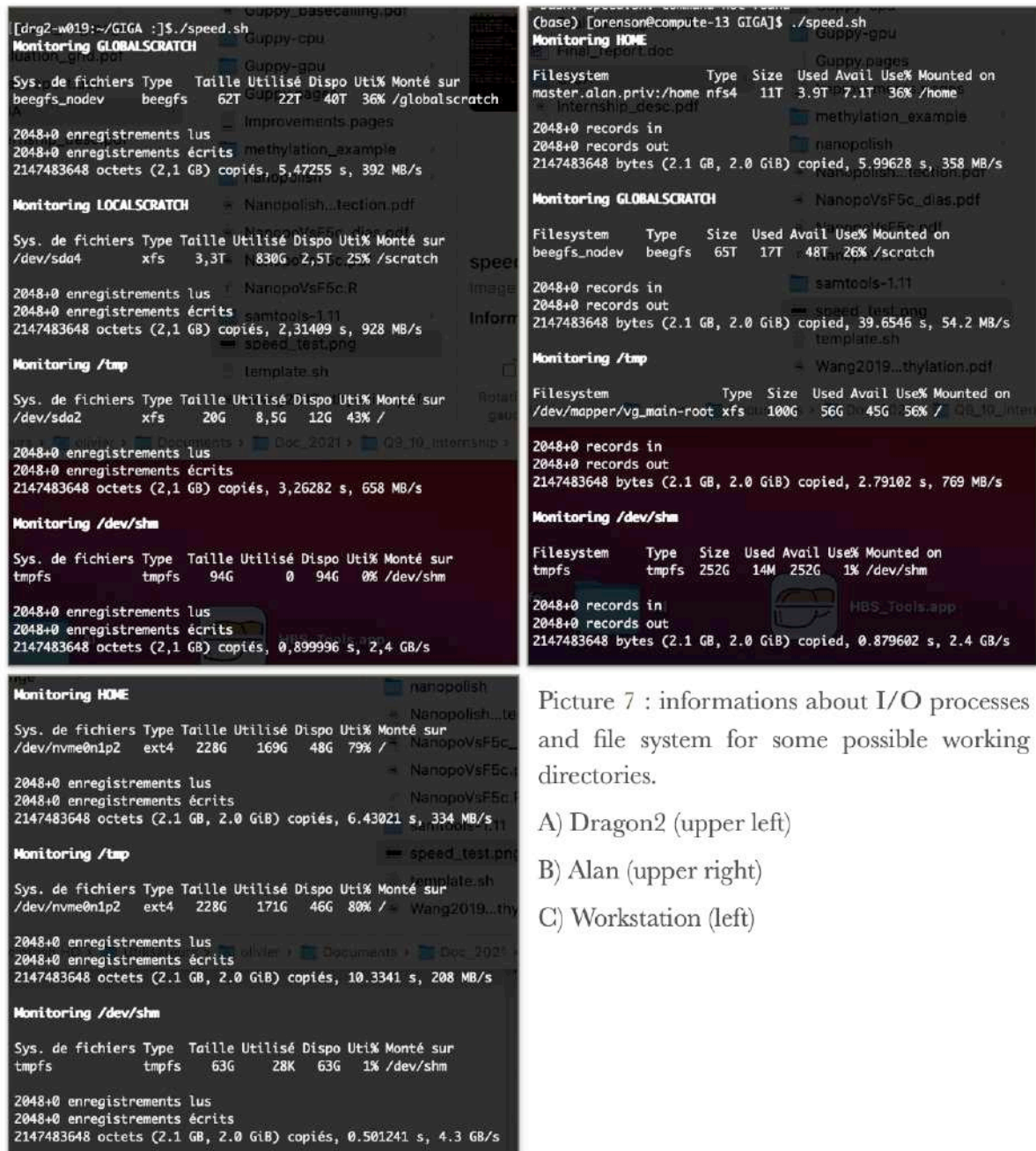


Picture 6 : workstation GPU monitoring (nvidia-smi) respectively with 1, 4 and 20 working CPUs
 Yellow line : GPU memory (constantly at 90%)
 Blue line : GPU computations



Picture 8 : linux file systems performances (from CECI “Scientific Data Management” course)

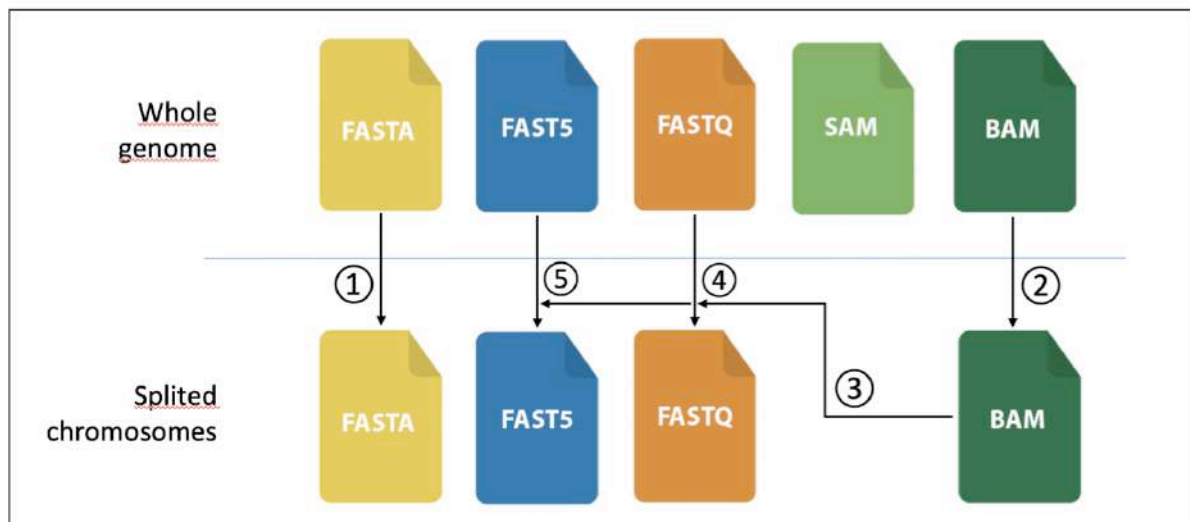
INFO-F-527 INTERNSHIP FINAL REPORT



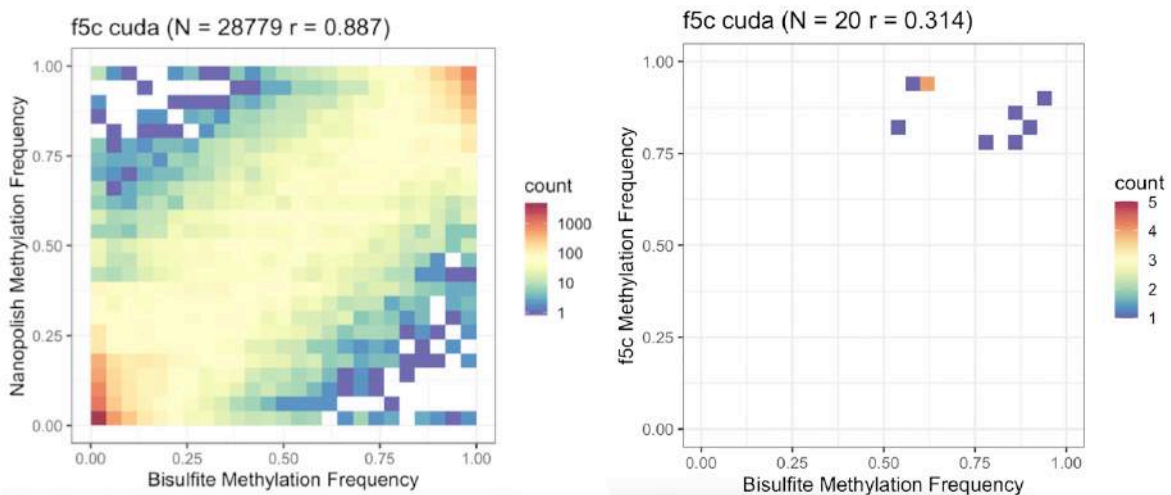
Picture 7 : informations about I/O processes and file system for some possible working directories.

- A) Dragon2 (upper left)
- B) Alan (upper right)
- C) Workstation (left)

INFO-F-527 INTERNSHIP FINAL REPORT



Img 9 : split data workflow. Step 1 : split reference genome file into chromosomes. Step 2 : extract data on chromosome 3, 10 and 11 from bam alignment file. Step 3 : based on the previous file get the id of each read aligned on that region. Step 4 : using the read ids list, extract those reads from the main fastq file. Step 5 : using the read ids list again, extract raw signal of those reads from fast5 directory. See supplementary data for script details.



Img10: correlation between f5c output and bisulfite sequencing with the dataset provided by Oxford Nanopore (left) and with mcf7 data (right).

DATE: 17/05/2021

Signature of the student

Signature of the host company supervisor