Introduction

Targeted sequencing

In some sequencing applications, the focus of study — a single gene, or a selection of genomic regions — comprises a small fraction of the genome/sample. In these cases, characterisation through whole-genome sequencing can be inefficient and costly. Targeted sequencing is a term used to describe strategies that reduce the time spent sequencing regions that are not of interest, thereby significantly reducing the amount of data required to achieve the desired depth of the regions of interest. As well as reducing sequencing cost, this reduces the data analysis burden and enables a quicker workflow. Targeted sequencing using nanopore technology can be achieved in several ways:

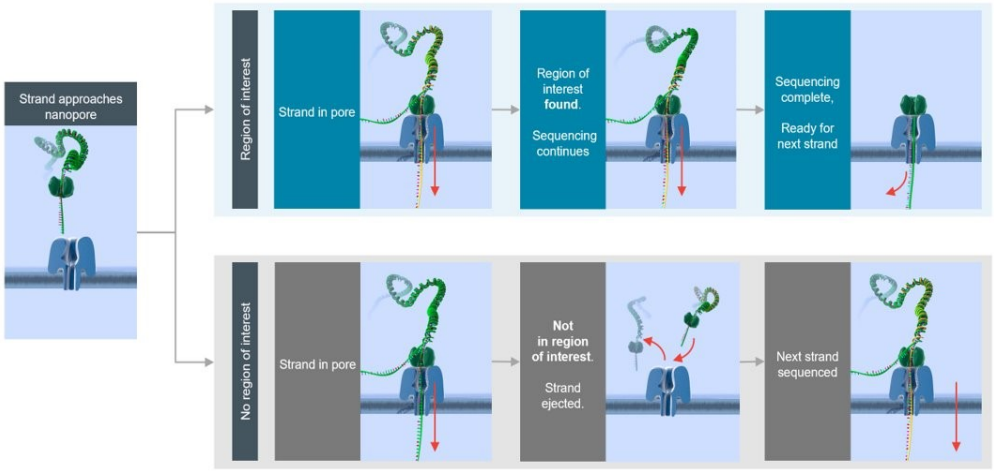
- amplicon sequencing
- pull-down
- Cas9-based enrichment
- adaptive sampling (AS)

Advantages of nanopore sequencing for a targeted approach

- Using methods such as Cas9-based enrichment and adaptive sampling allows enriching for regions inaccessible to traditional technologies, e.g. repetitive or GC-rich regions that cannot be amplified with PCR
- Characterisation of very large regions of the interest: it is possible to enrich and sequence targets spanning tens of kilobases or more in single reads
- Base modifications are retained without the need for any further library preparation
- Information-rich data: detection of single nucleotide variants (SNVs), structural variants, repeat expansions and base modifications in a single sequencing run
- On-demand sequencing, rapid access to results, simple end-to-end workflows and minimal start-up costs

Introduction to adaptive sampling

Adaptive sampling offers a fast and flexible method to enrich regions of interest by depleting off-target regions: target selection takes place during sequencing itself, with no requirement for upfront sample manipulation. The library is prepared and loaded as normal and “adaptive sampling” is selected in MinKNOW (typically a .bed file detailing target regions will need to be uploaded). Once the run commences, sequencing will begin and due to the real-time nature of nanopore sequencing, it is possible to identify whether or not the strand that is being sequenced is within the region of interest (ROI). If the read does not map to the ROI, MinKNOW will reverse the polarity of the applied potential, ejecting the strand from the pore so it is able to accept a new, different template strand. Off-target strands are continually rejected until a strand from the ROI is detected and sequencing is allowed to proceed.



Adaptive sampling can run in two different modes: “enrichment” and “depletion”. In “enrichment”, ROIs are uploaded to MinKNOW and strands that fall outside of this are rejected. In “depletion” mode, targets that are not of interest (e.g. host DNA in a host:microbiome metagenomic analysis) are uploaded to MinKNOW and strands that fall within these regions are rejected. In this document, we discuss “enrichment” strategies only. We generally observe an enrichment for ROI of ~5-10-fold when targeting using AS, and the sections below outline our advice on how this can be achieved. For targeting regions within human genomes, we find this level of enrichment is robust as long as the total fraction that is being targeted is <10% of the total genome, and can enable users to obtain a mean depth >20-40x of ROI on a MinION flow cell.

Considerations for experimental design

Library input

For conventional (non-AS sequencing runs) sequencing runs, we recommend adding ~5-50 fmol of DNA library to obtain optimum sequencing yield. During an adaptive sampling run, most of the template strands that enter the pores are NOT from the region of interest – these strands are thus quickly rejected/ejected from the pore and the pore returns to the open pore state: this can lead to an overall increase in open pore time. To mitigate this, we recommend adding 50 fmol of library. However, we find that adding more than this does not provide additional benefit.

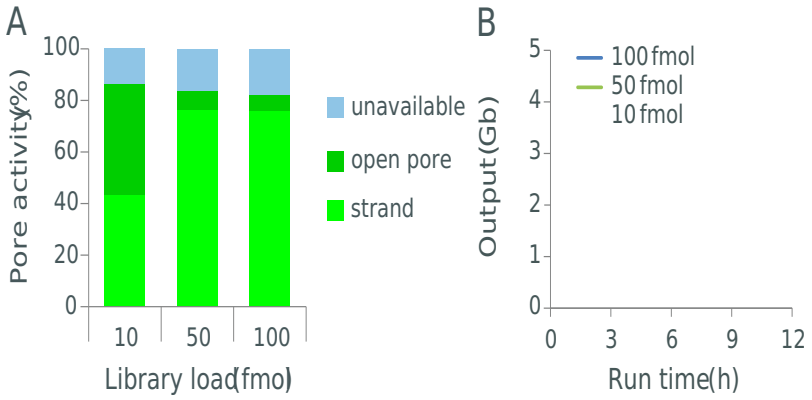


Figure 1. Impact of library loading amount on flow cell output in adaptive sampling. To demonstrate the impact of library

input amount on flow cell output in AS runs, we generated a library from human genomic DNA. 10, 50 or 100 fmol of the library were run on GridION under AS conditions (targeting ~1% of the genome, split into 800 40,000 bp targets) and the output recorded. Panel A) displays the health of each flow cell (as represented in the GUI on MinKNOW) and shows that at a library load of 10 fmol, there is a significant number of pores in the open pore state. At 50 fmol, the pores are almost fully occupied. Panel B) displays the rate of data acquisition (Gb) over time and shows as the library load increases from 10 fmol to 50 fmol the output increases. However, little increase in output is observed by exceeding this.

Sequencing Kit

To date, our work with AS has focussed primarily on libraries generated using the Ligation Sequencing Kit. We have found that due to the improvements in sequencing kit components, the performance/output of AS is improved as you progress through the kit iterations: LSK109 < LSK110. Therefore, we recommend using the most up-to-date version of the kit that is available.

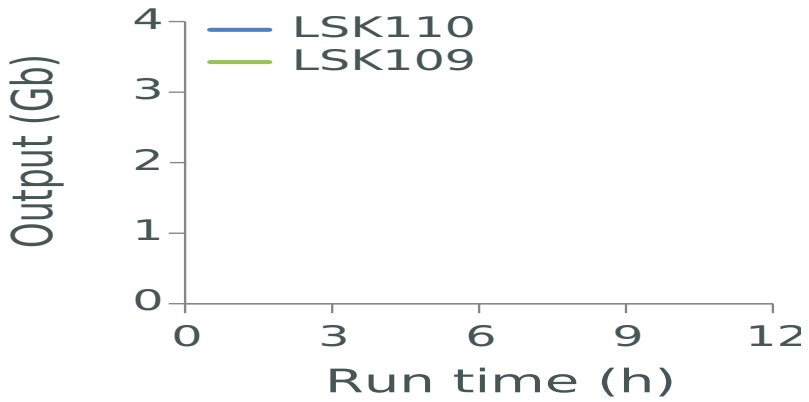


Figure 2. Impact of sequencing kit chemistry on flow cell output in adaptive sampling. To demonstrate the impact of sequencing kit chemistry on flow cell output in AS runs, we generated libraries from unfragmented and size selected human genomic DNA using various iterations of the Ligation Sequencing Kit: SQK-LSK109 and SQK-LSK110. 50 fmol of each library was run on GridION under AS conditions (targeting ~1% of the genome, split into 800 40,000 bp targets) and the output recorded.

Fragment length

The reversal of the potential difference applied across the membrane to eject off-target reads is the same mechanism by which MinKNOW “unblocks” pores. On rare occasions, pores are not successfully unblocked by this unblock mechanism: this is documented elsewhere. In this situation, the pore becomes “unavailable” for sequencing (terminally blocked) and over time, as more pores become unavailable, the rate of data acquisition begins to slow. Due to the persistent application of the “unblock” mechanism in AS runs (to reject off-target reads), the rate of pores becoming terminally blocked is often higher when compared with conventional runs. This effect can be exacerbated when input fragment lengths are longer. In order to maintain high data outputs with longer read libraries in AS runs, we recommend performing flow cell washes (using EXP-WSH004) to clear terminally blocked pores, and then re-loading library.

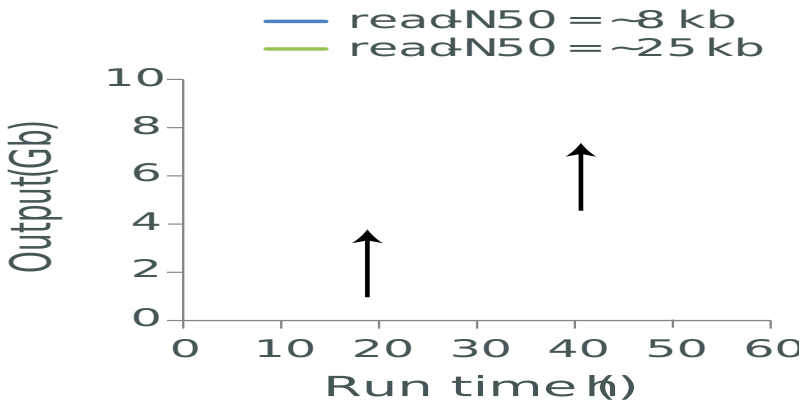


Figure 3. Impact of fragment length on flow cell output in adaptive sampling. To demonstrate the impact on flow cell output in AS runs of different length libraries, we generated two libraries from human genomic DNA; one was fragmented using a Covaris g-TUBE to produce a read-N50 ~8 kb and the other was unfragmented and size selected to generate a read-N50 of 25 kb. The libraries were run on GridION under AS conditions (targeting ~1% of the genome, split into 800 40,000 bp targets). The rate of data acquisition (Gb) slows over time as an increased number of pores become unavailable: this rate of decay is faster with the longer libraries. Output can be increased by performing flow cell washes every ~20 hours (denoted by vertical arrows).

"Buffer" size

“Buffer” regions are flanking regions added to the side of every single target described in the .bed file. These regions allow reads which begin with a sequence that may not initially map to our target region, but may extend into our target region, to be accepted. By accepting reads which map into these flanking regions we increase the number of accepted reads that hit our target: while increasing the size of this buffer does lead to an increase in off-target bases sequenced, it can also increase the depth of the regions of interest. The size of the buffer chosen relates to the read length of the underlying library – we recommend setting the buffer size to the read-N10 of the library (this is the point at which 10% of the bases sequenced are from reads that are this length or longer). Having a buffer size that exceeds the length of all reads present in the library will lead to reads being accepted and sequenced that never enter the target region.

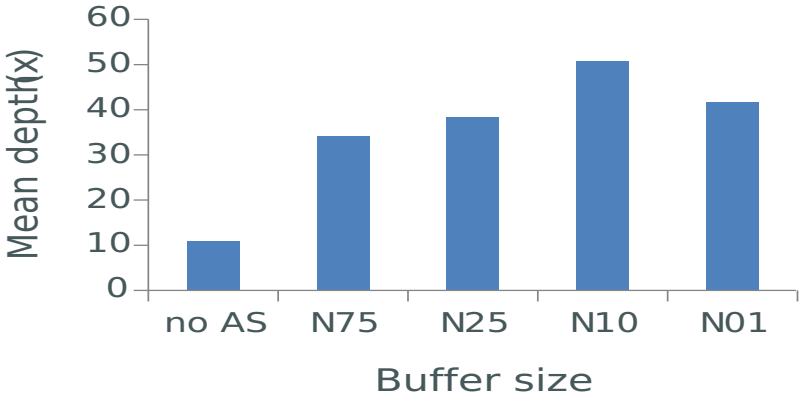


Figure 4. Impact of buffer size on sequencing depth in adaptive sampling. To demonstrate the impact of buffer size on sequencing depth in AS runs, we generated libraries from unsheared and size-selected human genomic DNA (to produce a read-N50 ~25 kb; the read length of the library was determined by running on GridION under non-AS conditions). We then performed the AS runs: we targeted (at random) 800 40,000 bp targets (this equates to 1% of the genome). The buffer size upstream and downstream of each target was set to ~20 kb, ~47 kb, ~62 kb and ~88 kb to represent the read-N75, -N25, -N10 and -N01 of the library. We observed that maximum target depth was obtained when the buffer size was set to the read-N10 of the library.

Note, as the number of targets increases, the percentage of the genome that is being targeted will increase. Once the total fraction being targeted breaches 10-15% (ROI + buffer), we find that the level of enrichment can drop – see Figure 5. Therefore, it may be necessary to either fragment the library (this has the impact of reducing the buffer size around each target), or reduce the buffer size from read-N10 (e.g. to read-N75).

Target region

In preliminary AS runs we typically observed 5-10 fold enrichment of the ROI: this equates to ~20-40x sequencing depth of the targets in human genome tests. To test how robust this level of enrichment was (for human genome studies), we titrated the number of targets and the size of the targets in various scenarios and recorded the sequencing depth obtained in control runs (no AS) and in AS runs. We found that this level of enrichment/depth is observed when the fraction of the genome targeted is <10% regardless of the number of targets or the size of the targets.

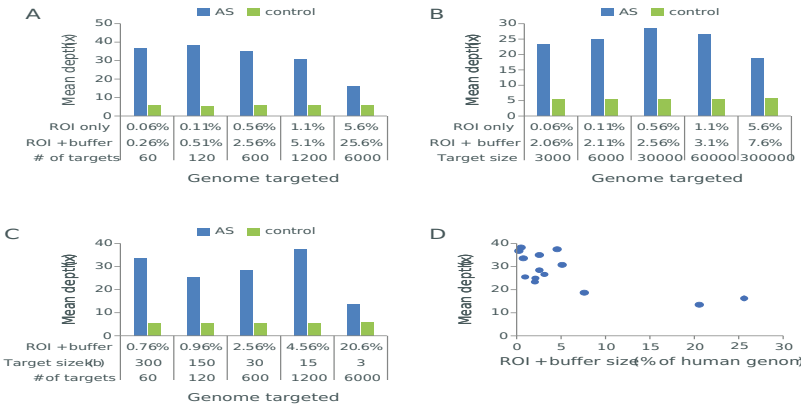


Figure 5. Impact of configuration of target region on sequencing depth in adaptive sampling. To demonstrate the effect of different target configurations on sequencing depth in AS runs, we generated libraries from unsheared and size-selected human genomic DNA (to produce a read-N50 ~25 kb; the read length of the library was determined by running on GridION under non-AS conditions). We then performed the AS runs with various configurations of target regions (different numbers of targets, different sizes of targets, with different fractions of the genome targeted) and recorded the sequencing depth of our ROIs. The upstream and downstream buffer size was fixed as the read-N10 of the library (~50 kb) and this value added to both side of each individual ROI in all conditions. **Panel A)** In this experiment, we enriched for different numbers of 30,000 bp targets to cover ~0.06% (60 targets), ~0.11% (120 targets), ~0.56% (600 targets), ~1.1% (1,200 targets) and ~5.6% (6,000x targets) of the genome. **Panel B)** In this experiment, the number of targets is set at 600, but the size of each target is altered to cover ~0.06% (3,000 bp targets), ~0.11% (6,000 bp targets), ~0.56% (30,000 bp targets), ~1.1% (60,000 bp targets) and ~5.6% (300,000 bp targets) of the genome. **Panel C)** In this experiment, the fraction of the genome targeted was set at ~0.56% (1.8 Mb), but the number and size of targets was altered. **Panel D)** The data from panels A, B and C is combined to show how the size of the AS.bed file (which comprises ROI + buffer regions) relates to depth of target.

As the % of the genome targeted increases (particularly above 20%), the depth will drop to 10-20x. However, it may be possible to recover performance by either fragmenting the library (this has the impact of reducing the buffer size around each target by reducing the length of the read-N90), or reducing the buffer size from the read-N90 value (e.g. to read-N25): these strategies can reduce the % of the genome targeted and may increase sequencing depth of the target region.

Chromosome enrichment

AS experiments were also performed to target whole chromosomes: no buffer region is required in such a scenario. We targeted chromosome 7 (159 Mb) and chromosome 21 (47 Mb) by AS and recorded the depth obtained compared with control runs: the results obtained are in line with previous observations – sequencing depth increases ~5-10-fold to around 30x.

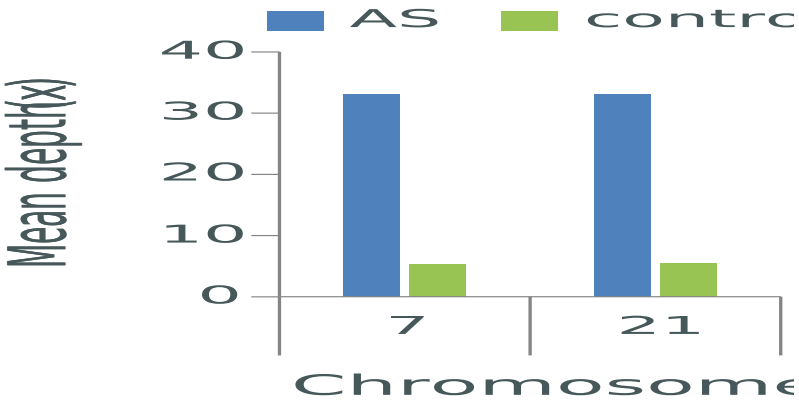


Figure 6. Targeting whole chromosomes by AS. We prepared sequencing libraries from unsheared and size selected human genomic DNA (to produce a read-N50 ~25 kb; the read length of the library was determined by running on GridION under non-AS conditions). We then performed the AS runs to target either chromosome 7 or chromosome 21 and recorded the depth of the targeted chromosome.

Designing and running an experiment with adaptive sampling

Adaptive sampling inputs: creating a .bed file

After considering factors like % of the genome to enrich for, read lengths and response time, the next step is to collate the input files needed for an adaptive sampling experiment. These are:

- A genomic reference: this must be a FASTA or a pre-calculated minimap2 index
- Optionally, a .bed file with the genomic coordinates of the target regions

Note that if you do not input a .bed file, then the entire FASTA/minimap2 index file will be used for enrichment or depletion.

.bed files can be acquired in several ways.

Note: We recommend saving your alignment files in a folder with the prefix/data or the default location MinKNOW saves your reads after a sequencing run, to avoid issues when performing adaptive sampling in MinKNOW:

e.g.
Windows: C:\data\
Mac: /Library/MinKNOW/data/
Ubuntu: /var/lib/MinKNOW/data/

If you are targeting an exome

The EPI2ME Labs analysis suite includes a "Curating Adaptive Sampling input files for MinKNOW" tutorial. The tutorial allows users to prepare and download the necessary files to perform an adaptive sampling experiment selecting for reads that span genes, transcripts, exons etc. stored within ensembl. The workflow outputs:

- A reference genome file
- The source .gtf file from which target regions were produced
- The .bed file containing target regions to provide to MinKNOW

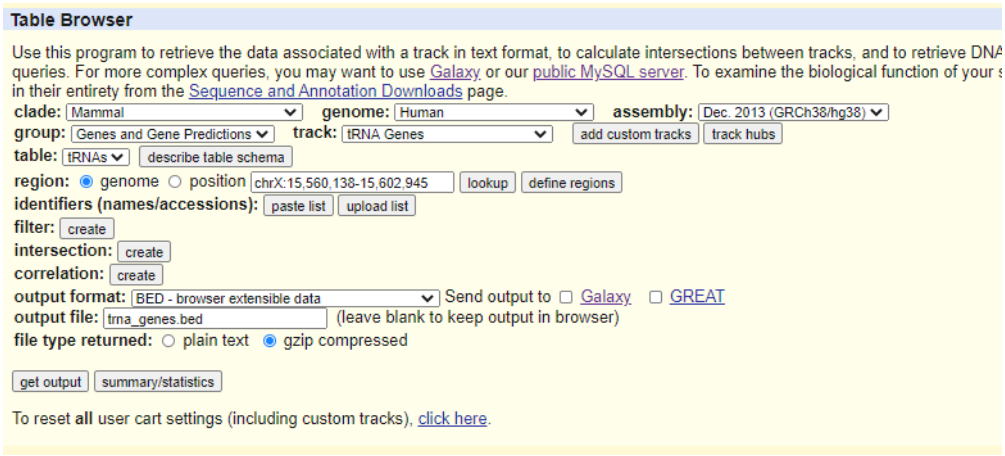
To use EPI2ME Labs, refer to the Curating Adaptive Sampling input files for MinKNOW workflow.

If you are targeting other genomic regions

You will also need a .bed file and the reference sequence. Reference sequences can be downloaded from ensembl, UCSC Genome Browser, or other databases. For more information about .bed files and their required fields, see BED File Format - Definition and supported options.

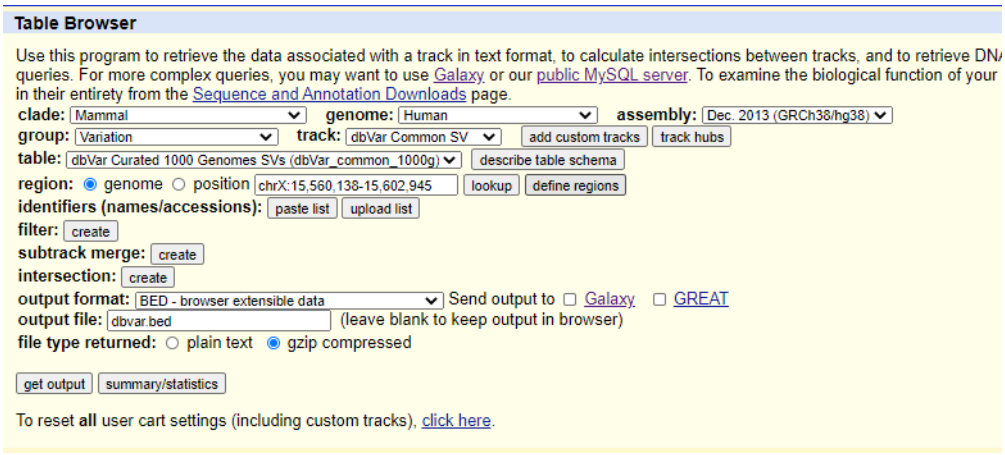
.bed files can be downloaded from the UCSC Table Browser. For example, to select human tRNA genes only:

1. Select the following in the dropdown menus: a. clade: Mammal b. genome: Human c. assembly: Dec_2013(GRCh38/hg38) d. group: Genes and Gene Predictions e. track: tRNA Genes
2. Select **BED - browser extensible data** in the output format.
3. Click **get output**



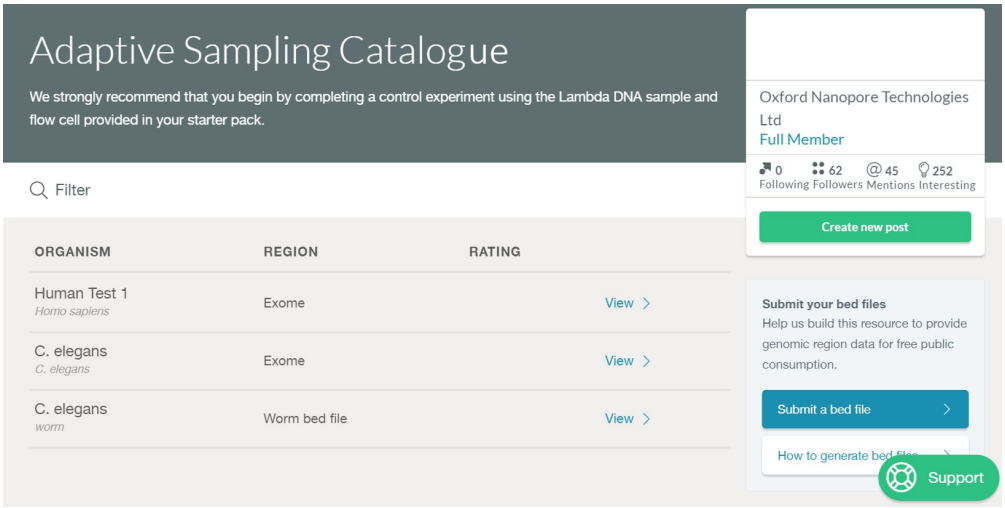
As another example, to select common structural variants in the human genome:

1. Select the following in the dropdown menus: a. clade: Mammal b. genome: Human c. assembly: Dec_2013(GRCh38/hg38) d. group: Variation e. track: dbVar_Common_SV
2. Select **BED - browser extensible data** in the output format.
3. Click **get output**



Sharing .bed files on the Nanopore Community

To browse adaptive sampling .bed files submitted by other Community members, or to submit your own, please visit the Adaptive Sampling Catalogue. Instructions are provided for how to add a .bed file to the catalogue.



Starting an adaptive sampling experiment in MinKNOW

Instructions for setting up an adaptive sampling run are provided in the MinKNOW protocol. During an enrichment experiment, adaptive sampling will reject all sequences that are not present in the reference/.bed file, whilst in a depletion experiment, only sequences that are present in the reference/.bed file will be rejected. Currently, adaptive sampling is enabled on MinION Mk1C, GridION and PromethION (for more details, see the "Release caveats" section below).

With adaptive sampling, barcode balancing is available as a beta feature which allows users to preferentially sequence under-represented barcodes in their samples to balance the read data across the barcodes based on the reference file provided. Please note, with increasing the number of reads sequenced for these barcodes, the overall data output for all reads may be reduced.

Output files

The files that MinKNOW outputs during a sequencing experiment are described in the MinKNOW protocol.

For adaptive sampling experiments, there is an additional CSV file named `adaptive_sampling.csv` that is saved in `other_reports` in the run folder, which can be used for troubleshooting.

The file has the following fields:

Field	Description	Example value
batch_time	The epoch time in seconds when the batch was processed	1595496307.7494152
read_number	The order of the read within the channel	14
channel	The channel on the flow cell that the read passed through	512
num_samples	The number of samples in the read	4000
read_id	The id of the read	d605d893-44a3-4648-a5c3-50928267678f
sequence_length	The number of bases in the read. For stop_receiving reads, this is the number of bases recorded until the decision was made to continue sequencing the rest of the read.	324
decision	Which decision was taken	**enrich:** unblock, stop_receiving, unblock_hit_outside_bed **deplete:** unblock, stop_receiving, stop_receiving_hit_outside_bed An "unblock" decision means that adaptive sampling decided to reject the read.

IMPORTANT

Release caveats

The adaptive sampling release is fully supported on GridION.

- GridION Mk1 allows adaptive sampling on 3-5 flow cells simultaneously
- GridION X5 allows adaptive sampling on a maximum of three flow cells simultaneously

The Fast basecalling model is used with adaptive sampling on the GridION.

Adaptive sampling on MinION Mk1B is supported for users who have enabled GPU basecalling on their system. More information is available in the Community Support post: How do I enable live GPU basecalling on MinION Mk1B?

Adaptive sampling support for the beta release is available on PromethION and MinION Mk1C. A new 'sketch' model is used with the MinION Mk1C for basecalling small chunks to make decisions to eject or keep reads. This model provides low-latency basecalling, making the most of the available on-board compute.

On PromethION 48, we recommend the following:

- **Fast: 16** flow cells. We recommend no more than 10 flow cells for the best response time.
- **HAC: 6** flow cells.
- **SUP:** We do **not recommend** using SUP basecaller when running adaptive sampling

On PromethION 24, we recommend the following:

- **Fast: 8** flow cells. We recommend no more than 8 flow cells for the best response time.
- **HAC: 3** flow cells.
- **SUP:** We do **not recommend** using SUP basecaller when running adaptive sampling.

MinION Mk1C does not support running live basecalling alongside adaptive sampling.

Currently, we only recommend using alignment on bacterial-sized genomes.

Future developments

- Improvements to reference upload

Troubleshooting

Are too many reads being rejected?

You can determine the number of reads that are being rejecting due to adaptive sampling via two methods:

- During the run: using the read length histogram in MinKNOW by ticking the box **Split by read end reason**. If you are running in a mode where a sufficient number of reads should be rejected to produce a peak of short reads, there should be a bimodal distribution with a peak at a base-count much lower than what you expect from the sample. The reads rejected by read until will have the label **Data Service Unblock Mux Change**.
- After the run: using the adaptive_sampling.csv. This file will contain information about the final decision that was taken by adaptive sampling for each read.

Am I not getting a sufficient enrichment from my sample?

- Are your reads long enough? Short reads will spend most of the time in the decision phase, which means the software will select for a low fraction of bases.
- Is your input amount sufficient? Adaptive sampling increases the number of transitions from strand to pore to allow recapture. Therefore, having a low input amount with an associated long time to capture the strands can exacerbate this.
- Does your output decrease quickly over time? You may need to wash your flow cell using the Flow Cell Wash Kit to recover some channels, then reload more of your sample.

Am I not getting my entire region of interest enriched?

If you are using a .bed file to perform enrichment, you may need to extend the regions of interest to include more bases at the beginning and end, where you are not seeing sufficient enrichment.

How many bases are read before accepting or rejecting a strand?

For enrichment, 200 bases are required before a decision is made. However in practice, due to MinKNOW read detection the number is typically ~450 bases for R9.4.1 chemistry.

For depletion, the minimum number of bases required is ~450, but up to 4000 bases can be read before a strand is ejected.

What happens to rejected reads?

If a strand has already translocated through the pore and its sequence has been aligned to make the rejection decision, the read is output to the fast_q pass folder if it passes the Q-score filter, or to thefast_q fail folder if it falls below the Q-score filter.

Rejected reads will not be sequenced again. Rejected reads are included in the total outputs reported by MinKNOW, however as these reads are typically short in a good adaptive sampling run, this should make a minor difference to the enriched output value.

Should I use a reference index or reference sequence?

Either is suitable, however if using a reference sequence, the run setup may take longer as MinKNOW builds the reference index (minimap2 file). You can use MinKNOW to build the minimap2 index ahead of run setup if required - see the MinKNOW protocol for details.

Is adaptive sampling impacted by barcodes, especially if I have custom barcodes that increase the length of the sequence before my material? Can I de-multiplex while running adaptive sampling?

Adaptive sampling should not be impacted by barcodes, and demutiplexing is possible in adaptive sampling mode.

Should I use FASTA alone or also a .bed file? Does this make a difference to performance in different circumstances?

Either option will work, however inputting both a FASTA and .bed file means that reads can be potentially aligned to either file, and therefore be easier to reject.

If I have a GPU on my computer for use with the MinION, and select HAC basecalling while running adaptive sampling, can this lead to issues in the performance of adaptive sampling or basecalling?

Yes - both adaptive sampling and basecalling use the GPU, so this can lead to performance issues such as adaptive sampling falling

behind and reads not being rejected and fully sequenced.

References

References

- Miller, D et al. (2020) Targeted long-read sequencing resolves complex structural variants and identifies missing disease-causing variants, *bioRxiv*, Nov 2020, https://www.biorxiv.org/content/10.1101/2020.11.03.365395v1
- Kovaka, S; Fan, Y; Ni, B; Timp, W & Schatz, MC. (2020) Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED, *bioRxiv*, Feb 2020, https://www.biorxiv.org/node/1132895.abstract
- Payne, A; Holmes, N; Clarke,T; Munro, R; Debebe, B; Loose, M (2020) Nanopore adaptive sequencing for mixed samples, whole exome capture and targeted panels, *bioRxiv*, Feb 2020, https://www.biorxiv.org/content/10.1101/2020.02.03.926956v2
- De Maio, N; Manser, C; Munro, R; Birney, E; Loose, M & Goldman, N (2020) BOSS-RUNS: a flexible and practical dynamic read sampling framework for nanopore sequencing, *bioRxiv*, Feb 2020, https://www.biorxiv.org/content/10.1101/2020.02.07.938670v2
- Loose, M; Malla, M & Stout, M (2016) Real Time Selective Sequencing using Nanopore Technology,*Nat Methods* **13**, 751–754 (2016). https://doi.org/10.1038/nmeth.3930