1 Sample "bt1"

1.1 General Information

This sample has the code `bt1` and is named "BT1". It is part of the project code `under_ice_rerun_bt` called "Rerun of under ice depth profile of lake BT" along with 7 other samples.

1.2 Meta-data details

The meta data associated with this sample can be found in the following JSON file:

http://github.com/xapple/gefes/tree/master/json/under_ice_rerun_bt.json

Here is a summary of all the information pertaining to this sample:


```
"project_name": "under_ice_rerun_bt",
"project_long_name": "Rerun of under ice depth profile of lake BT",
"project_num": 11,
"uppmax_project_id": "b2014083",
"illumina_run_id": "150611_D00457_0101_AC6VPBANXX",
"library_strategy": "WGS",
"library_source": "METAGENOMIC",
"library_selection": "RANDOM",
"library_layout": "Paired-end",
"platform": "ILLUMINA",
"instrument_model": "Illumina HiSeq",
"instrument_software": "v2.2.58",
"forward_read_length": 124,
"reverse_read_length": 124,
"date": "2014-03-18",
"latitude": [
    63.582139,
    "N"
],
"longitude": [
    12.270827,
    "E"
],
"country": "Sweden",
"location": "Lake near Skalstugan",
"bioproject": "PRJNAXXXXXX",
"gefes_settings": {
    "quality_checker": {
        "object": {
            "source": "gefes.preprocess.sickle",
            "name": "Sickle"
        }
    }
},
"samples_base_dir": "~/GEFES/raw/projects/under_ice_rerun/",
"sample_name": "bt1",
```

```
"sample_long_name": "BT1",
"sample_directory": "Sample_BT1",
"sample_num": 1,
"forward_reads": "fwd.fastq.gz",
"reverse_reads": "rev.fastq.gz",
"forward_md5": "2f43253d68a8a0ffaf0c47d13b4181ea",
"reverse_md5": "8506cce1a9c4bf5ab5a8ef1630bc01ea",
"forward_read_count": 26132363,
"reverse_read_count": 26132363,
"organism": "aquatic metagenome",
"env_biome": "freshwater lake",
"env_feature": "lake water",
"env_material": "water",
"design_description": "Water was sampled from under the lake's ice cover using Limnos sampler. Water was pr
"biosample": "SAMNXXXXXXXX",
"depth": [
    0.5,
    "m"
],
"ph": 5.37,
"toc": [
    7.76666666667,
    "mg/l"
],
"ton": [
    0.474655233333,
    "mg/l"
],
"top": [
    9.45945945946,
    "µg/l"
],
"sulfate": [
    0.472656019,
    "mg/l"
],
"oxygen": [
    6.12,
    "mg/l"
],
"conductance": [
    17.88,
    "µS/cm"
],
"temperature": [
    0.3,
    "Celsius"
],
"filtered_volume": [
    0.75,
    "ml"
```


```
],
"cell_counts": [
    3492510,
    "cells/ml"
],
"co2": [
    101.8,
    "µM"
],
"ch4": [
    11.5,
    "µM"
],
"feII": [
    0.493358351617,
    "µM"
],
"feIII": [
    1.66048780223,
    "µM"
],
"fe_total": [
    2.15384615385,
    "µM"
],
"suva": [
    3.32188841202,
    "mg/l*m"
]
```

1.3 Processing

- This report (and all the analysis) was generated using the GEFES project at: http://xapple.github.io/gefes/
- Version `0.1.5` of the pipeline was used.
- This document was generated at `2016-12-15 21:06:16 CET+0100`.
- The results and all the files generated for this sample can be found at: ssh://ww-hmem02.climb.cluster/home/lucas/GEFES/views/projects/under_ice_rerun_bt/samples/bt1/
- The exact git hash of the latest commit was: 78325bb6760553550cb31b2adbd888f0395d099b

• 1.4 Also more simply referred to by its shortened tag `78325bb`.

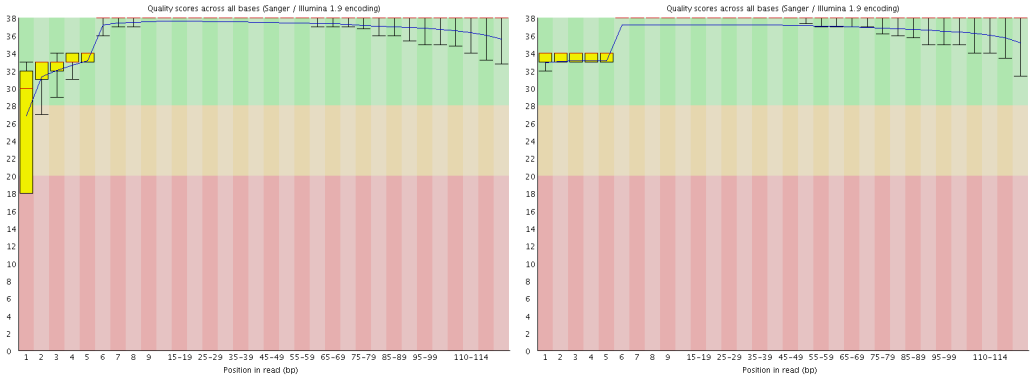
1.5 Raw data

- The forward read file weighed *1.84 GiB* and contained *26'132'363* reads.
- The reverse read file weighed *1.80 GiB* and contained *26'132'363* reads.

More information about the raw output of the sequencer for this sample can be found in the XML report generated by the Illumina software here:

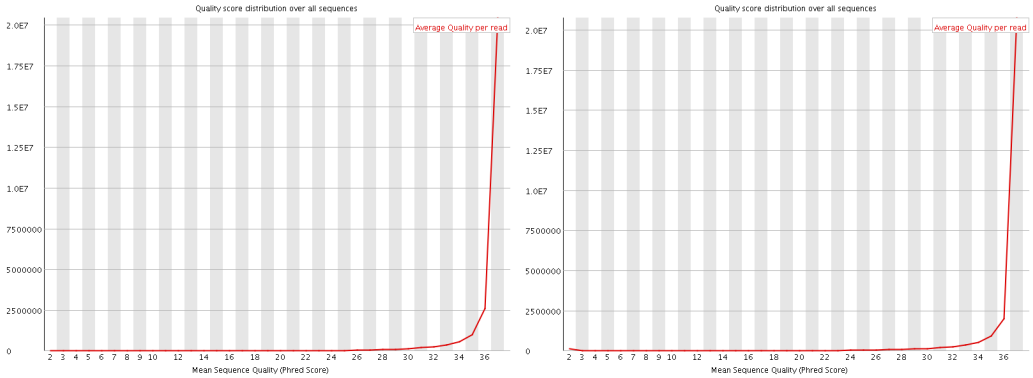
`ssh://ww-hmem02.climb.cluster/home/lucas/GEFES/raw/projects/under_ice_rerun/report.xml`

The average quality per base can be seen in figure 1 and the average quality per sequence in figure 2.



(a) Forward **(b)** Reverse

Figure 1. Raw per base quality



(a) Forward **(b)** Reverse

Figure 2. Raw per sequence quality

1.6 Preprocessing

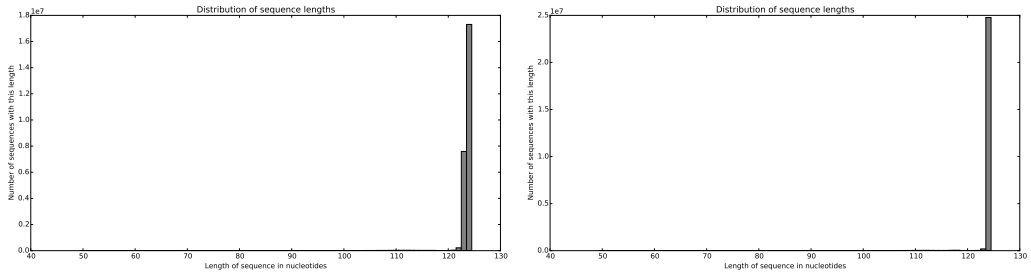
Next, we filter the sequences based on the following rules:

- We remove any bases from the extremities of each sequence if they are under the quality threshold of 20.
- We run a 12.0 base pair window over the sequences and check that the quality doesn't drop below the same threshold of 20. If it does, we trim the read and keep the longest stretch.

- If the resulting trimmed read is shorter than 50 base pairs or contains any undetermined "N" bases, we keep discard it.
- In the event that one of the reads in a pair gets discarded while the other one does not, we place the resulting singleton in a separate "singles" FASTQ file.

This leaves us with 98.54% of the original sequences organized in 25'749'925 pairs and 365'983 singletons.

Of course, now, not all sequences have the same length and we have created a distribution of read sizes as seen in figure 3.



(a) Forward **(b)** Reverse

Figure 3. Distribution of sequence lengths after quality control

The singletons have their own length distribution shown in figure 4:

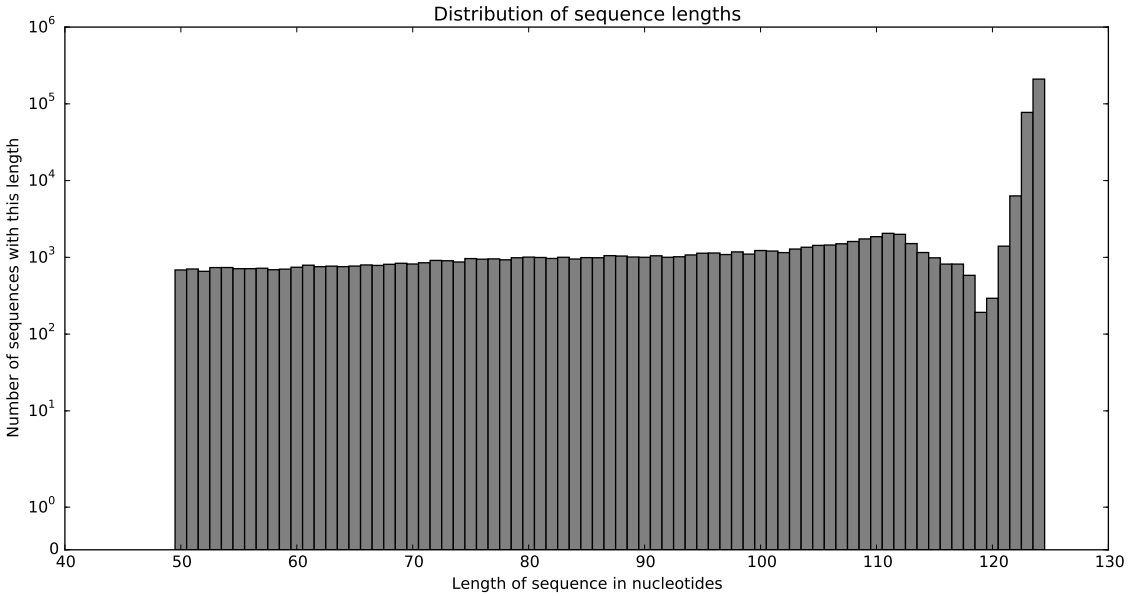
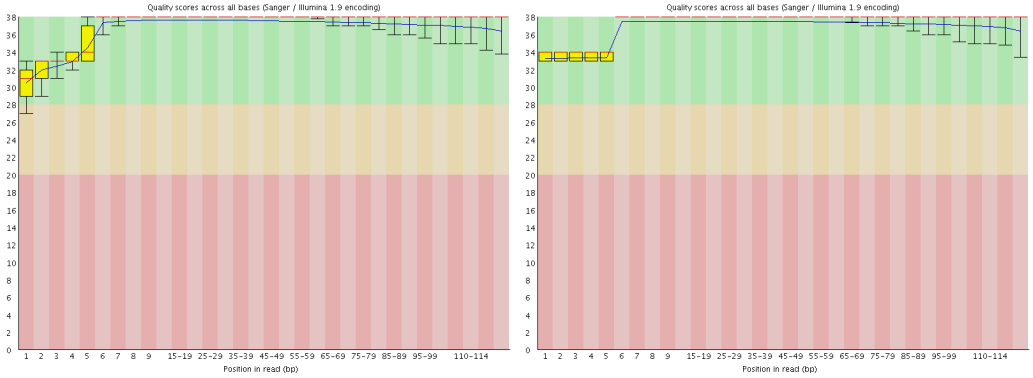


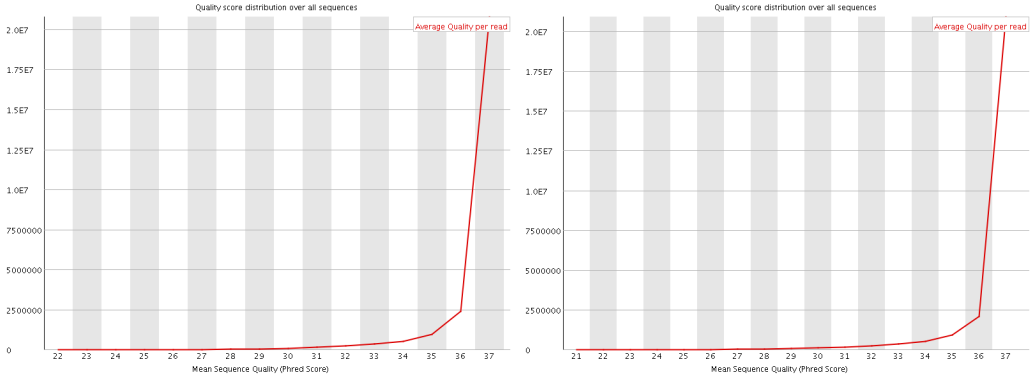
Figure 4. Singletons length distribution

We can look again at the per base quality and the per sequence quality of the pairs after cleaning in figure 5 and 6:



(a) Forward **(b)** Reverse

Figure 5. Per base quality after quality control



(a) Forward **(b)** Reverse

Figure 6. Per sequence quality after quality control

1.7 Mono Assembly

Most often, at this point the next step in the pipeline is to combine several samples together in an "aggregate" object and co-assemble them. The result of this operation is available in the report of the corresponding aggregate. But before that, we can try to simply run this sample into the assembler all by itself. For this we use the "Ray assembler v2.3.1" software with a kmer size of 71 and a contig length cutoff of 1000 base pairs.

This results in the production of 4'368 contigs, of which the length distribution can be seen in figure 7:

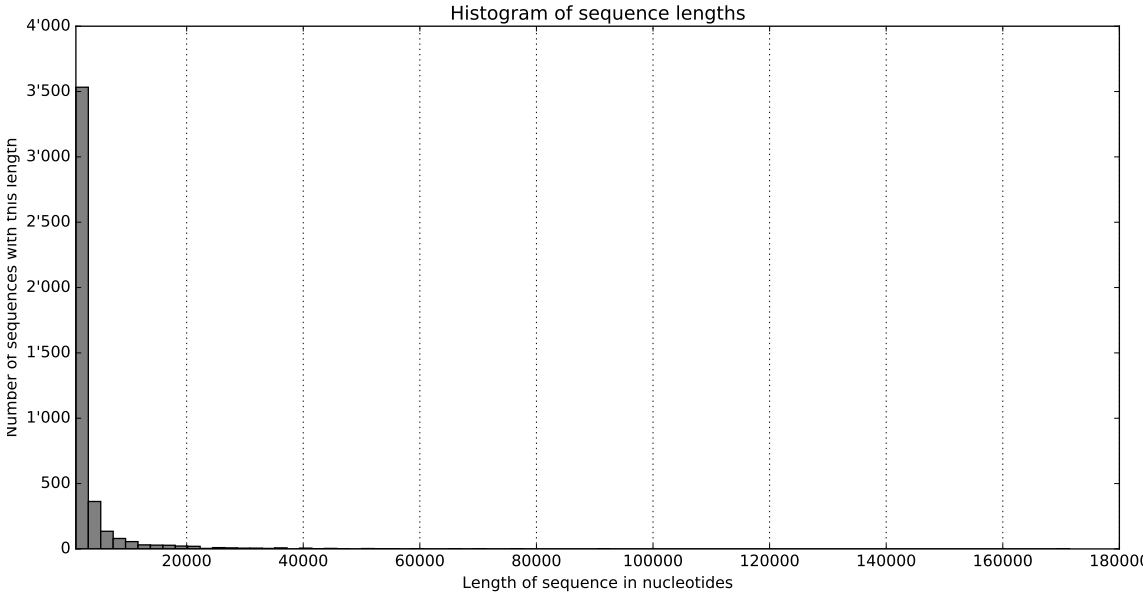


Figure 7. Mono-assembly length histogram

The total amount of base pairs generated is 13'967'331.

1.8 Mono Mapping

Now we will attempt to map every (cleaned) read (excluding singletons) of this sample back to the contigs that were generated in the previous mono-assembly step using the "Bowtie2 2.2.5" software. We will remove the predicted PCR duplicates, leaving us with 51'272'844 reads to map. Exactly 9.72% of the reads map back to the contigs generated in the mono-assembly and 90.28% have no match.

For every of the 4'368 contigs, we can now compute a mean coverage, as well as the covered fraction. The distribution of these two variables can be seen in figure 8 and 9:

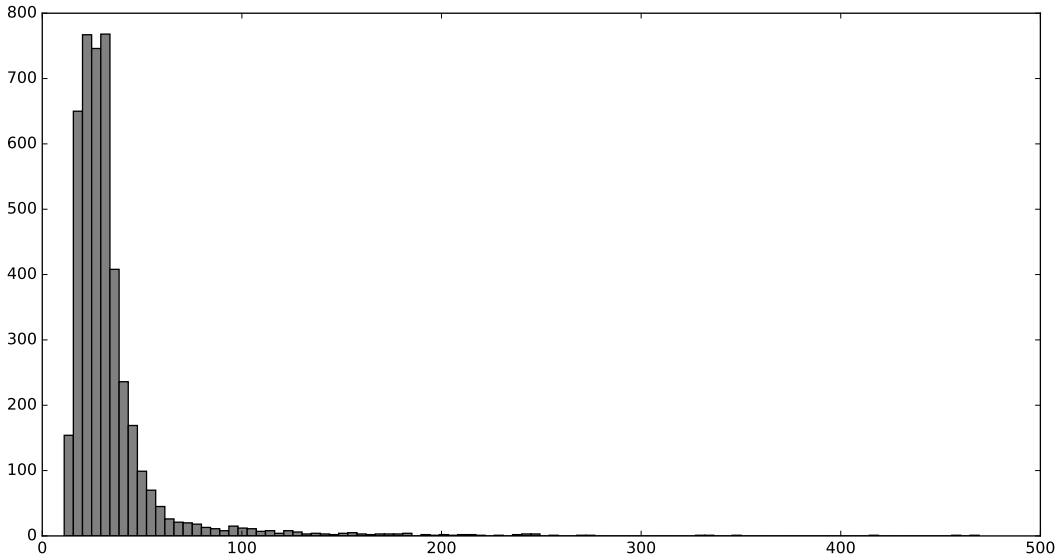


Figure 8. Mono-mapping mean coverage distribution

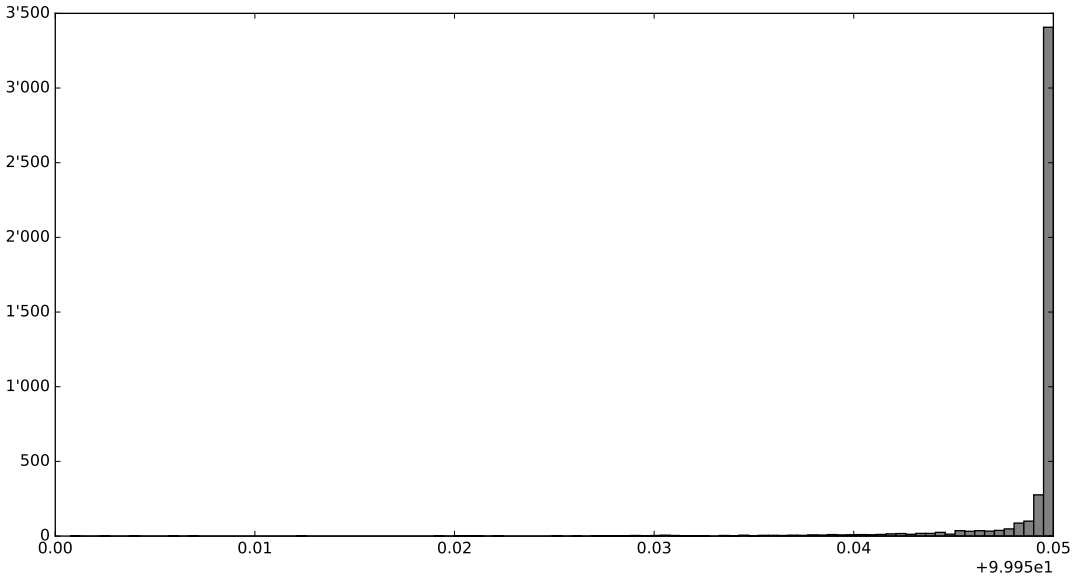
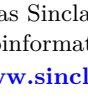


Figure 9. Mono-mapping percent covered distribution



1.9 Merged-Mapping

We can also attempt to map every (cleaned) read (excluding singletons) of this sample back to the contigs that were generated in the merged-assembly step using the "Newbler from Roche GS De Novo Assembler software v2.9" software. Again, we will remove the predicted PCR duplicates, leaving us with 50'515'584 reads to map. Exactly 45.15% of the reads map back to the contigs generated in the co-assembly and 54.85% have no match.

For every of the 72'730 contigs, we can now compute a mean coverage, as well as the covered fraction. The distribution of these two variables can be seen in figure 10 and 11:

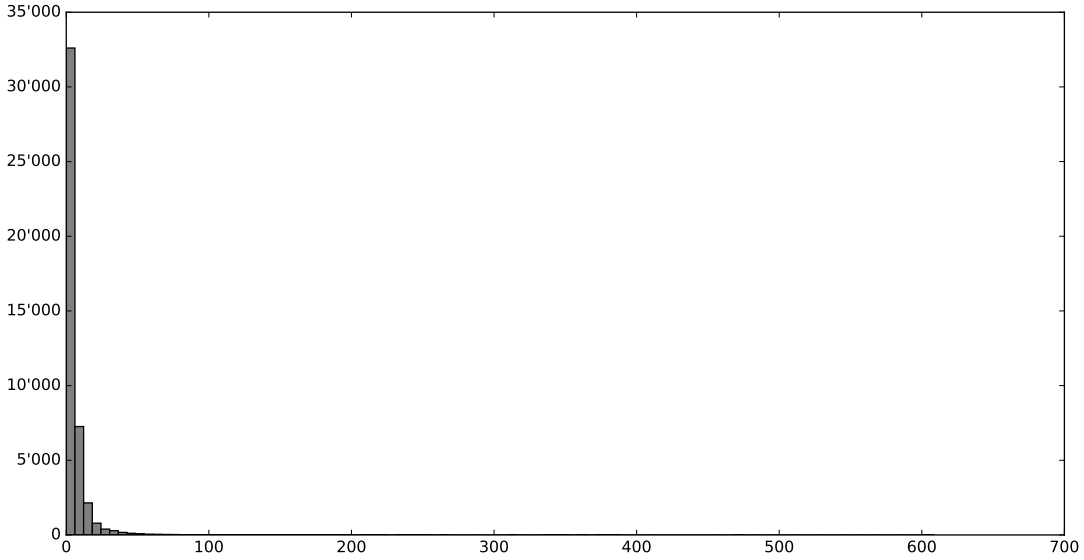


Figure 10. Merged-mapping mean coverage distribution

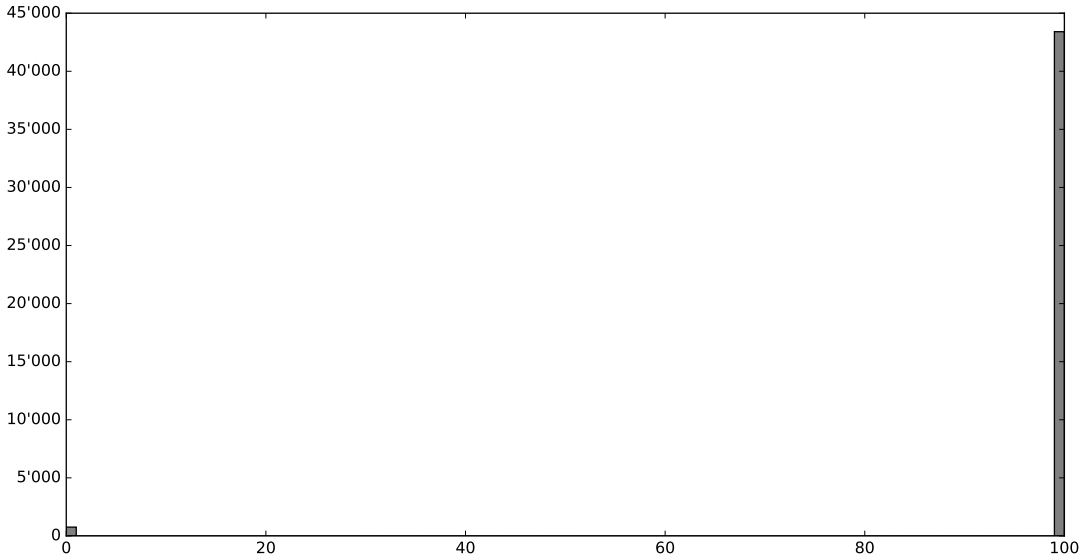


Figure 11. Merged-mapping percent covered distribution