Nanopore Sequencing Technology and Tools for Genome Assembly:

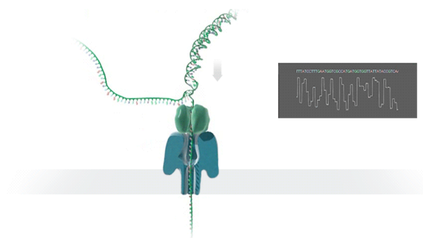
Computational Analysis of the Current State, Bottlenecks and Future Directions

Damla Senol Cali[1], Jeremie S. Kim[1,3], Saugata Ghose[1], Can Alkan[2] and Onur Mutlu[3,1]

[1] Carnegie Mellon University, Pittsburgh, PA, USA [2] Bilkent University, Ankara, Turkey [3] ETH Zürich, Zürich, Switzerland

Nanopore Sequencing

Nanopore sequencing is an emerging and a promising single-molecule DNA sequencing technology.



Nanopore is a nano-scale hole. In nanopore sequencers, an **ionic current** passes through the nanopores. When the DNA strand passes through the nanopore, the sequencer measures the **change in current**. This change is used to identify the bases in the strand with the help of **different electrochemical structures** of the different bases.

Advantages

- Does *not* require nucleotide labeling for detection during sequencing,
- Relies on the electronic or chemical structure of the different nucleotides for identification,
- Allows generating **very long reads**, and
- Provides **portability**, **low cost**, and **high throughput**.

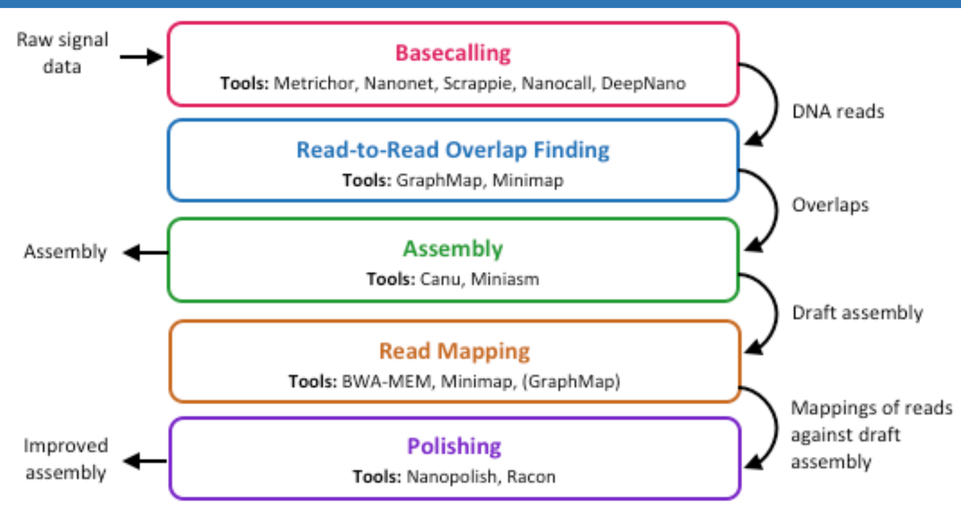
Challenges

- One major drawback: **high error rates**
- Nanopore sequence analysis tools need to:
  - **overcome high error rates**, and
  - take better advantage of the technology
- **Faster tools** are critically needed to:
  - take better advantage of the **real-time data production** capability of MinION, and
  - enable **fast, real-time data analysis**

Our Goal

- Comprehensively analyze the multiple steps and the associated state-of-the-art tools in genome assembly pipelines using nanopore sequence data in terms of **accuracy**, **performance**, **memory usage**, and **scalability**.
- Reveal **bottlenecks** and **trade-offs** that different combinations of tools lead to.
- Provide **guidelines** for both **practitioners**, such that they can determine the appropriate tools and tool combinations that can satisfy their goals, and **tool developers**, such that they can make design choices to improve current and future tools.

Nanopore Genome Assembly Pipeline



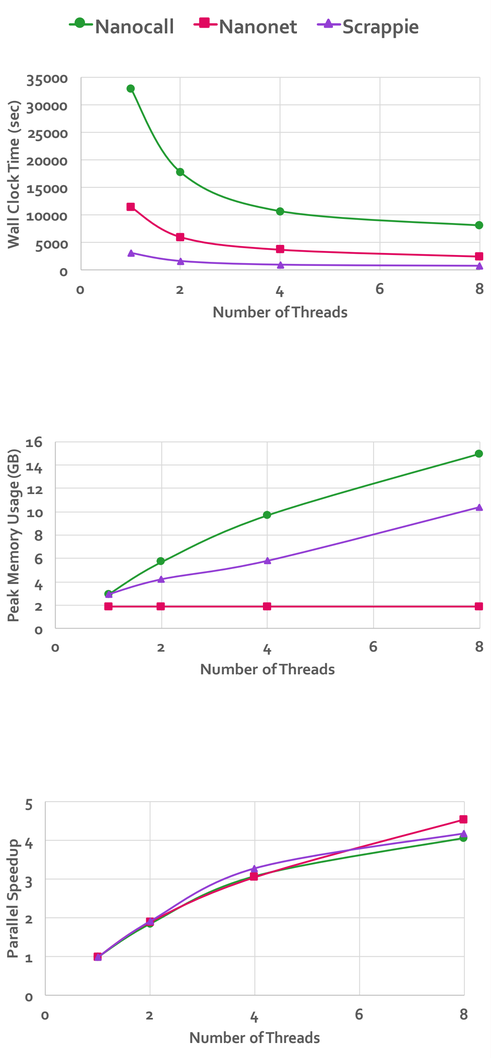
Results and Analysis

	Step 1 Wall Clock Time (h:m:s)	Step 1 Memory Usage (GB)	Step 2 Wall Clock Time (h:m:s)	Step 2 Memory Usage (GB)	Step 3 Wall Clock Time (h:m:s)	Step 3 Memory Usage (GB)	Number of Contigs	Identity (%)	Coverage (%)
Metrichor + Canu	–	–	–	–	44:12:31	5.76	1	98.05	99.92
Metrichor + Minimap + Miniasm			2:15	12.30	00:01:09	1.96	1	87.71	94.85
Metrichor + GraphMap + Miniasm			6:14	56.58	00:01:05	1.82	2	86.22	96.95
Nanonet + Canu	17:52:42	1.89	–	–	11:32:40	5.27	1	97.92	99.97
Nanonet + Minimap + Miniasm			1:13	9.45	00:00:33	0.69	1	85.50	92.76
Nanonet + GraphMap + Miniasm			3:18	29.16	00:00:32	0.65	1	85.36	91.16
Scrappie + Canu	03:11:41	13.36	–	–	33:47:41	5.75	1	98.46	99.90
Scrappie + Minimap + Miniasm			2:52	12.40	00:01:29	1.98	8	86.94	90.04
Scrappie + GraphMap + Miniasm			7:26	38.31	00:01:23	1.87	1	86.78	89.86
Nanocall + Canu	47:04:53	37.73	–	–	01:35:23	3.77	86	93.33	28.93
Nanocall + Minimap + Miniasm			1:15	12.19	00:00:20	0.47	5	80.52	42.92
Nanocall + GraphMap + Miniasm			5:14	56.78	00:00:16	0.30	3	80.51	41.32
DeepNano + Canu	23:54:34	8.38	–	–	01:15:48	3.61	106	92.75	99.16
DeepNano + Minimap + Miniasm			1:50	11.71	00:01:03	1.31	1	82.38	65.00
DeepNano + GraphMap + Miniasm			5:18	54.64	00:00:58	1.10	1	82.39	64.92

OBSERVATION 1: The choice of the tool for the **basecalling step plays an important role to overcome the high error rates** of nanopore sequencing technology. **Basecalling with RNNs** (e.g., Metrichor, Nanonet, Scrappie) provides **higher accuracy** and **higher speed** than **basecalling with HMMs.** Also, the newest basecaller of ONT, **Scrappie**, has the potential to overcome the **homopolymer basecalling problem**.

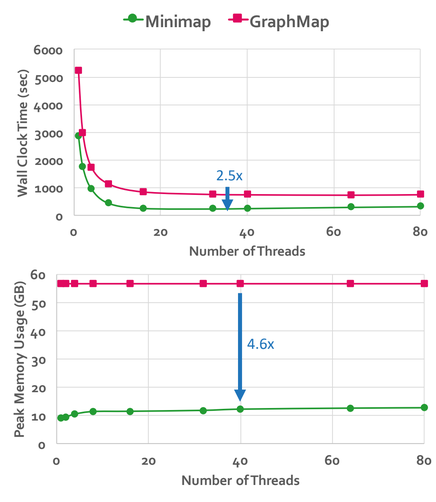
OBSERVATION 2: **Scrappie** and **Nanocall** have a **linear increase in memory usage** when number of threads increases. In contrast, **Nanonet** has a **constant memory usage** for all evaluated thread units.

OBSERVATION 3: When the number of threads exceeds the number of physical cores, **the simultaneous multithreading (SMT) overhead** prevents continued linear speedup of Nanonet, Scrappie and Nanocall.



OBSERVATION 4: Using **minimizers** instead of all k-mers, as done by Minimap, does *not* affect **the overall accuracy** of the first three steps of the pipeline.

OBSERVATION 5: By storing minimizers, Minimap has a much **lower memory usage** and thus much **higher performance** than GraphMap.



OBSERVATION 6: There is a **trade-off between accuracy and performance** when deciding on the appropriate tool for the assembly step. Canu provides **higher accuracy** than Miniasm, with the help of the error-correction step that is present in its own pipeline. However, Canu is much **more computationally intensive** and **greatly slower** (i.e., by 1096.3x) than Miniasm.

Miniasm is suitable for **fast initial analysis**, and the quality of its assembly can be increased with **an additional polishing step.**

OBSERVATION 7: The choice of BWA-MEM and Minimap for the read mapping step does *not* affect **the accuracy of the polishing step**. However, BWA-MEM is **computationally more expensive** than Minimap.

OBSERVATION 8: Both Nanopolish and Racon **significantly increase the accuracy of the draft assemblies**. However, Nanopolish is **computationally much more intensive and greatly slower** than Racon.

For more results, analysis and recommendations, please refer to:

OXFORD

Briefings in Bioinformatics, 2018, 1–18

doi: 10.1093/bib/bby017
Review Article

Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

Damla Senol Cali, Jeremie S. Kim, Saugata Ghose, Can Alkan and Onur Mutlu

BiB version *arXiv version*

Contact: Damla Senol Cali, *dsenol@andrew.cmu.edu*