



# MetaHipMer2

A metagenome assembly application written in UPC++

---

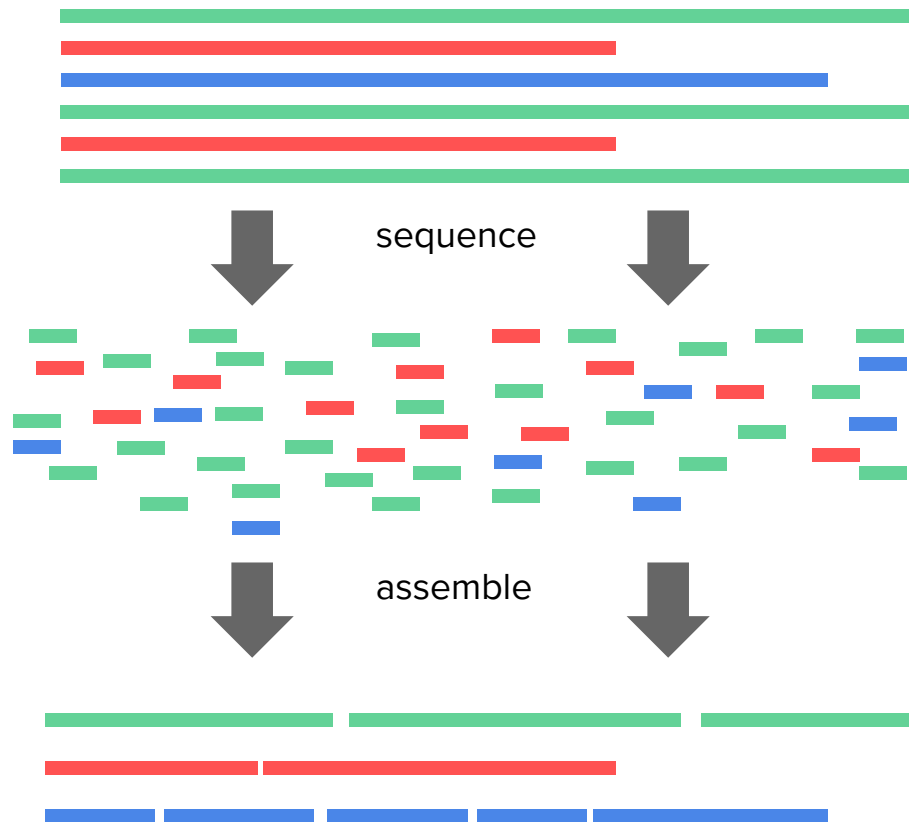
Steven Hofmeyr  
April 2021

# MetaHipMer2



- Part of Exascale Computing Project [Exabiome](#) project developed at Lawrence Berkeley National Laboratory (LBL)
- Original MetaHipMer (v1) was released in 2017 (written in UPC/MPI)
- Released [version 2](#) of MHM on [Sept 30th 2020](#)
  - Entirely rewritten in UPC++ (developed by Pagoda ECP project at LBL)
- Runs [2x to 10x faster](#) than MHM v1 and uses 2x less memory

# Metagenome Assembly



## microbial genomes (1000s)

- varying abundance (frequency)
- sequence depth (how many times sequenced, e.g. 50x)

## reads

- typical length for short reads 150-250
- error prone, e.g 0.24% per base
- number of reads for a genome dependant on sequence depth and abundance

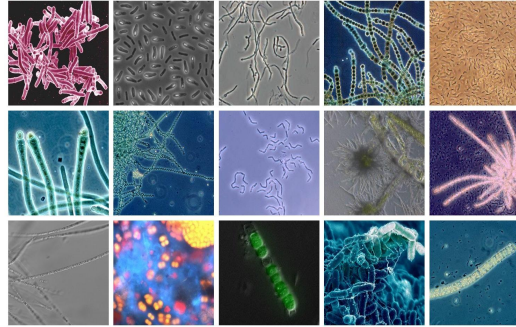
## contigs

- contiguous sequences
- the longer the better
- the fewer errors the better

# Science Analysis with MetaHipMer



What happens to microbes after a wildfire?



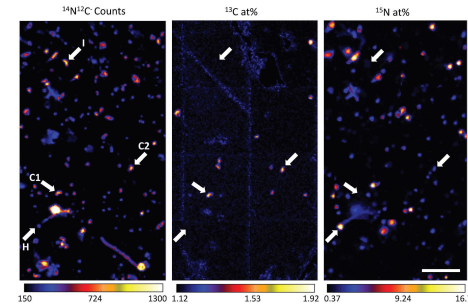
What are the microbial dynamics of soil carbon cycling?



What are the seasonal fluctuations in a wetland mangrove?



How do microbes affect disease and growth of switchgrass for biofuels

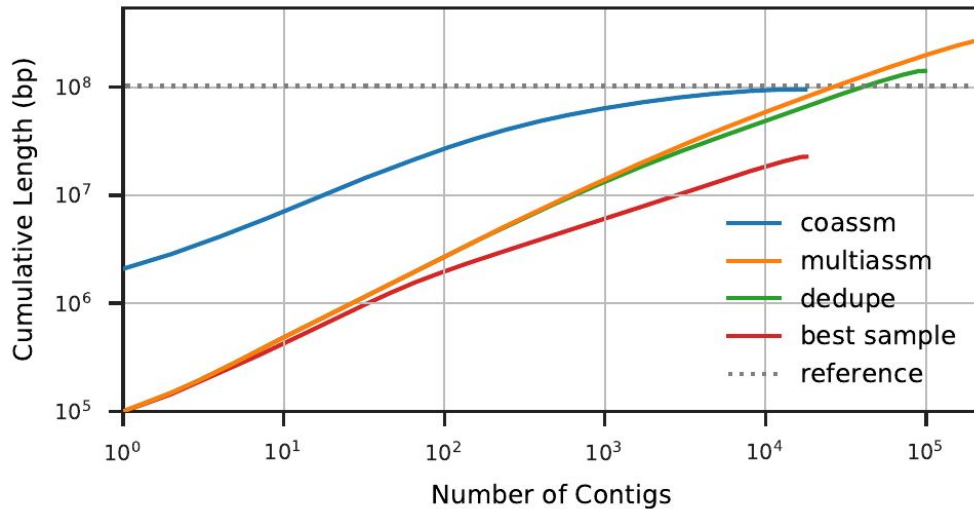


Combine genomics with isotope tracing methods for improved functional understanding

# Big Data, Big Compute → Better Science



Study: assemble an 813 GB Arctic ocean dataset of 12 samples plus synthetic data from 25 reference genomes injected at varying abundances

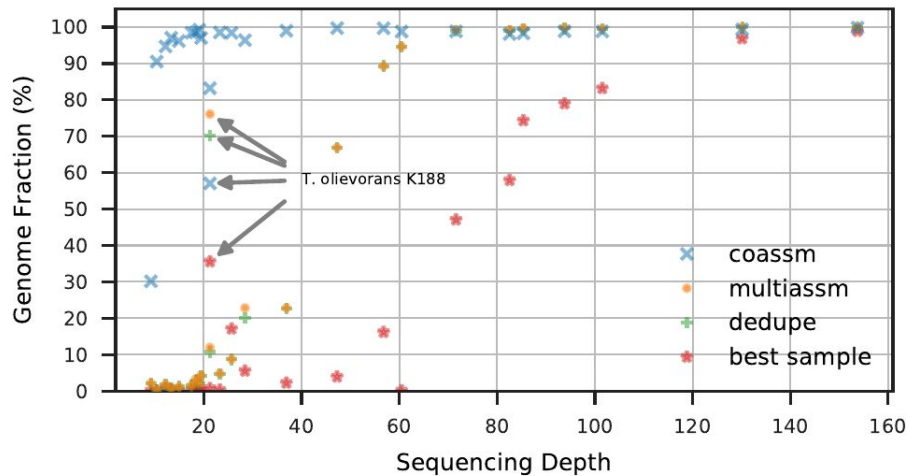


Approaches:

- **Coassembly**: entire data set
- **Multiassembly**: lane at a time
- **Dedup**: remove duplicates from multiassembly

**Coassembly** gives longer, less redundant assemblies, and is only possible with HPC (currently only MHM)

# Coassembly Finds Rare Genomes



- Low depth not recovered with **multiassembly/dedup** (< 5% genome fraction)
- Low depth mostly recovered by **coassembly** (> 90% genome fraction)
- **Rare genomes** are often of **more interest** than abundant genomes (which tend to be well known)

# UPC++ and GASNet-EX



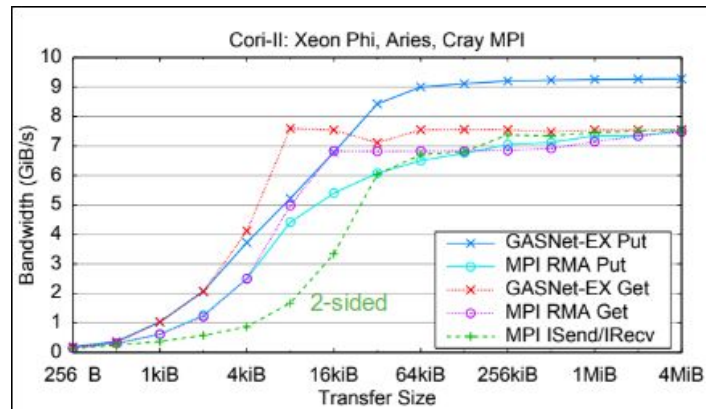
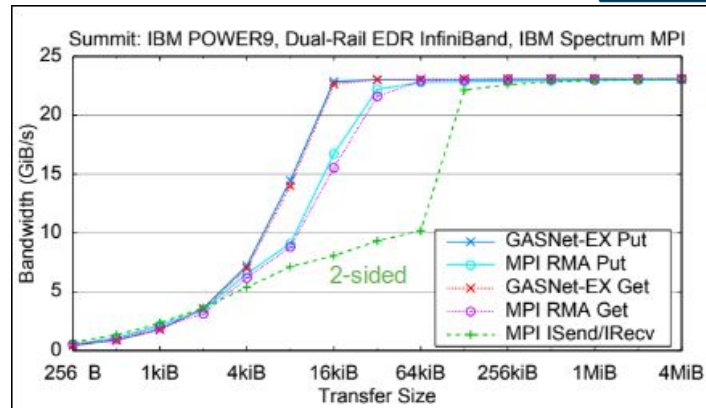
MHM2 is written in UPC++

- UPC++ is a C++ library for distributed memory systems that supports one-sided communication, remote procedure calls (RPCs), asynchronous communication and collectives
- UPC++ runs on GASNet-EX, a low-level communication layer that provides a network-independent interface that runs on several different hardware platforms
- Neither UPC++ nor GASNet-EX use MPI, although they are interoperable with MPI
- MHM2 does not use MPI (nor OpenMP)
- Depending on the system, the job launcher may use MPI

# Performance of GASNet-EX



- Plot shows one-sided vs two-sided message performance
- MPI ISend/Irecv 2-sided (+) gives lowest bandwidth
- MPI RMA (○, ○) outperforms 2-sided
- GASNet-Ex (x, x) consistently outperforms MPI (both 2-sided and RMA)



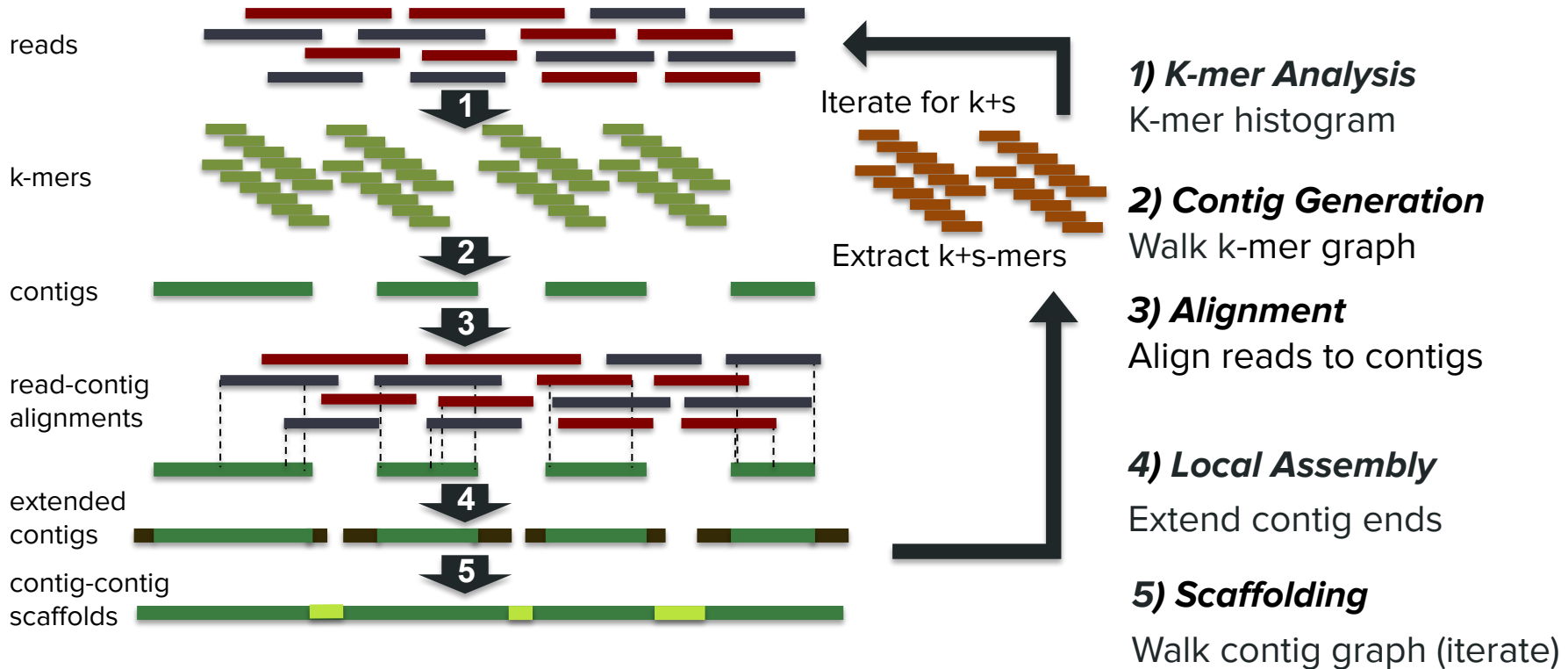


# MetaHipMer Computational Motifs



- **Distributed hash tables** everywhere
  - random memory access with **little locality**
  - communication is **irregular point-to-point**
  - scaling is achieved through good load balance
  - **accesses** generally happen separately from **updates**
- Computation mainly **string manipulation**
  - negligible floating point
- **Memory intensive**
  - coassembly of large datasets requires supercomputers
- Challenging for GPU optimization
  - we're working on this - for this contest, there is no GPU support

# MetaHipMer Assembly Pipeline

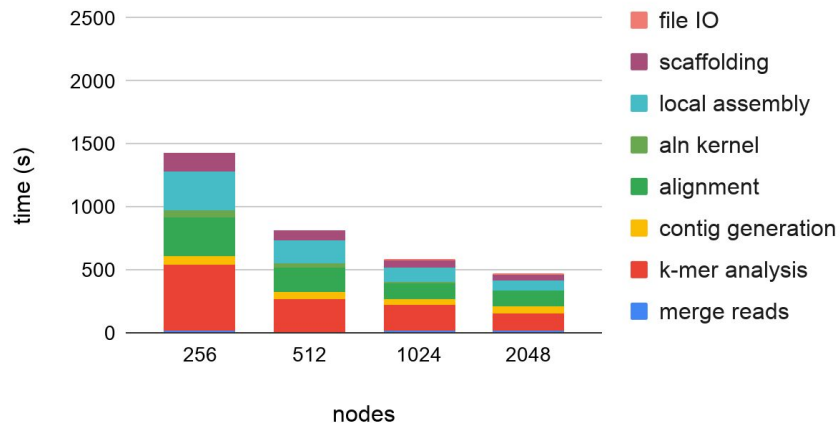


*Actual pipeline is more complex, simplified for purpose of presentation*

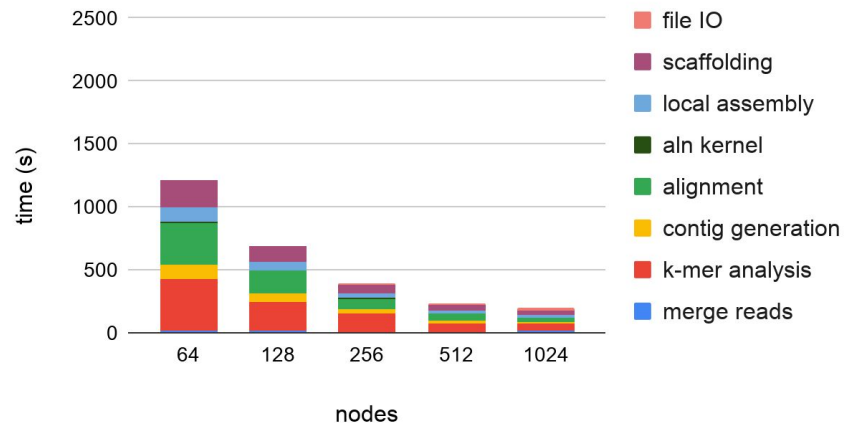
# MetaHipMer Stage Timings



Cori KNL



Summit

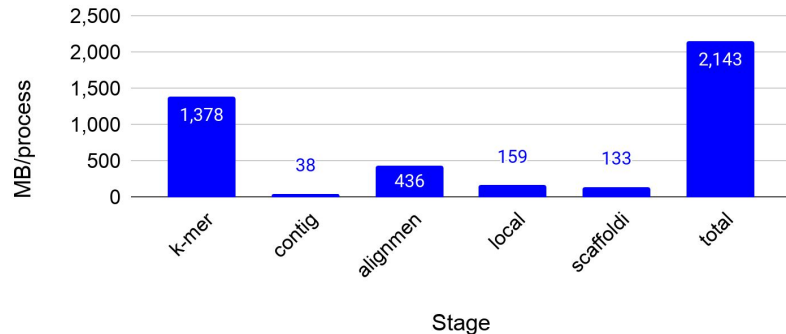


- Assembly of **822GB** dataset
- Decent scaling of all stages on both **NERSC Cori KNL** and **OLCF Summit**
- Exact breakdown of **timing depends on dataset** as well as machine/software (e.g. higher depth increases k-mer analysis time compared to contig generation)

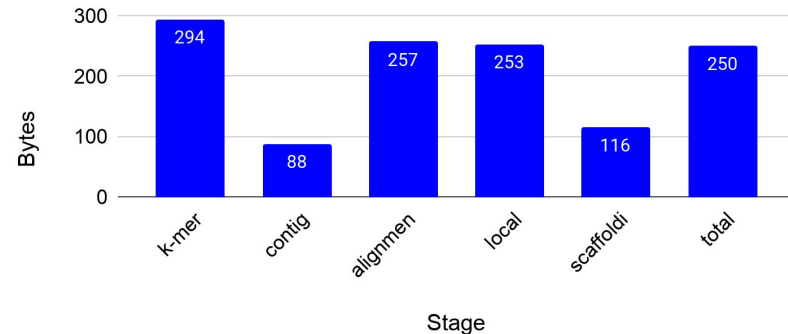
# MetaHipMer Communication Patterns



Communication Volume



Average Message Size

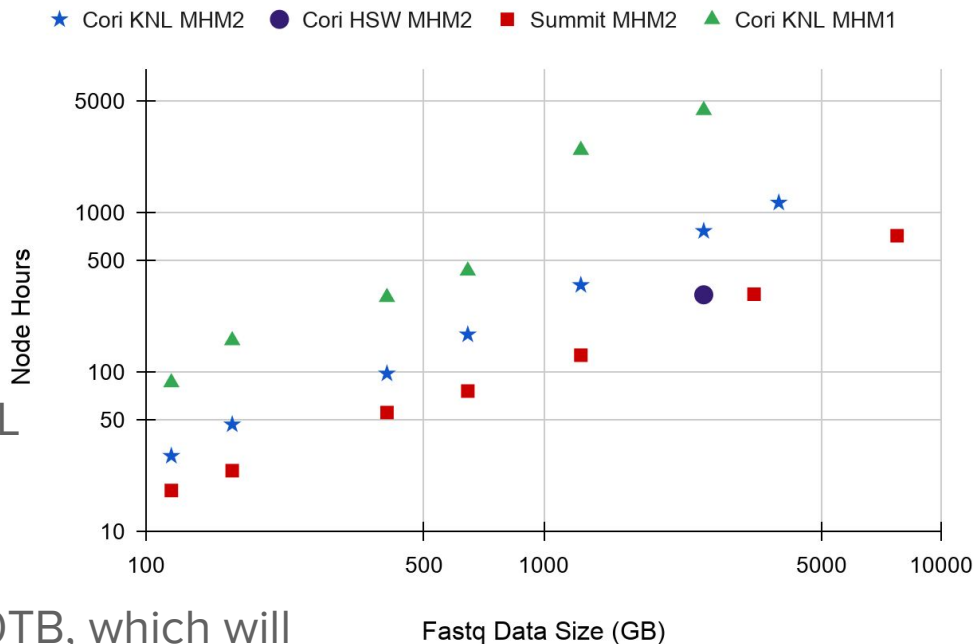


- Assembly of **822GB** dataset on 512 Summit nodes
- Two typical **patterns of communication**:
  - random point-to-point with **small active messages** (e.g. graph traversal in contig generation)
  - random point-to-point with larger, **aggregated messages** (e.g. k-mer analysis)
- Actual number and size of messages depends on both dataset (size and composition) and number of processes and nodes

# MetaHipMer Scaling on Large Datasets



- Assemblies of increasing fraction of **7.7TB dataset**
- Comparing Cori KNL, Haswell and Summit
- **Summit** is about **2-3x better** than Cori KNL
- Limited Cori Haswell results (similar to Summit)
- Up to **10x faster than version 1** on KNL
- The 7.7TB assembly on Summit is by far the **biggest metagenome ever assembled** - science to follow
- Eventual Exabiome project goal is 50TB, which will take *at least* 4650 node hours on Summit, i.e. full Summit for an hour



# MetaHipMer Tuning



- Look at ISC SCC webpage for guidance
  - <https://hpcadvisorycouncil.atlassian.net/wiki/spaces/HPCWORKS/pages/1827307543/MetaHipMer+2.0>
- Basic usage (unzip the input file):  
`mhm2.py -r competition1.fastq --checkpoint=no`
- Parameters that can be changed:
  - `--max-kmer-store`
  - `--max-rpcs-in-flight`
  - `--pin`
  - `--shared-heap`
  - `--procs`(check [https://bitbucket.org/berkeleylab/mhm2/src/master/docs/mhm\\_guide.md](https://bitbucket.org/berkeleylab/mhm2/src/master/docs/mhm_guide.md) for parameter descriptions)
- Change any UPC++ or GASNet options
  - <https://gasnet.lbl.gov/dist-ex/>
  - e.g. `GASNET_USE_ODP` or `GASNET_USE_XRC`

# MetaHipMer Results



- Output of MHM2 is non-deterministic, but overall statistics should not vary significantly from one run to the next:

```
Assembly statistics (contig lengths >= 500)
Number of contigs:      18360
Total assembled length: 27651381
Average contig depth:   7.21741
Number of Ns/100kbp:   0.0759456 (21)
Max. contig length:    71617
Contig lengths:
  > 1kbp:                20642175 (74.65%)
  > 5kbp:                 6289130 (22.74%)
  > 10kbp:                3476308 (12.57%)
  > 25kbp:                997631 (3.61%)
  > 50kbp:               240837 (0.87%)
```

???