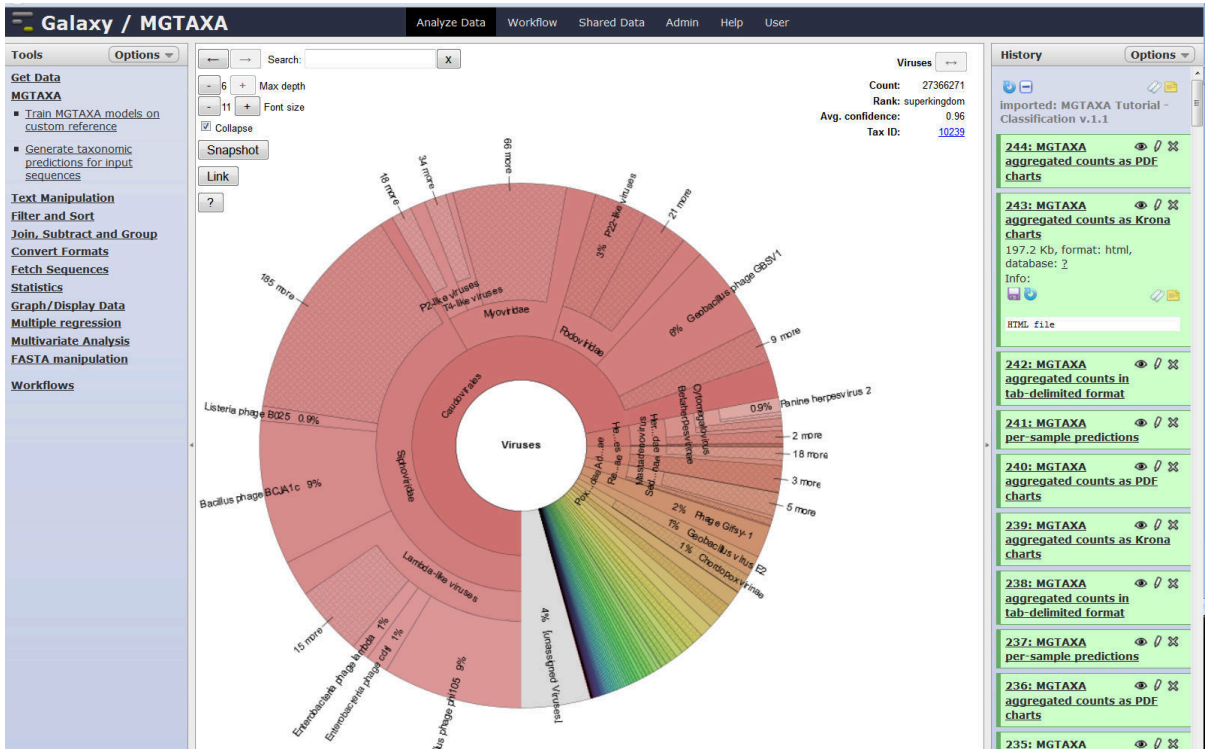
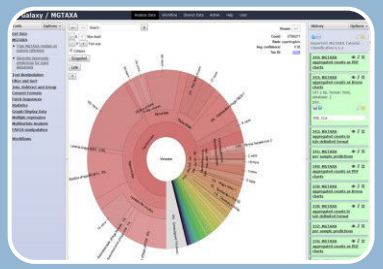
MGTAXA – a toolkit and a Web server for predicting taxonomy of the metagenomic sequences with Galaxy front-end and parallel computational back-end

Andrey Tovchigrechko, Seung-Jin Sul, Timothy Prindle, Shannon J. Williamson and Shibu Yooseph
http://andreyto.github.com/mgtaxa/ atovtchi@jcvi.org



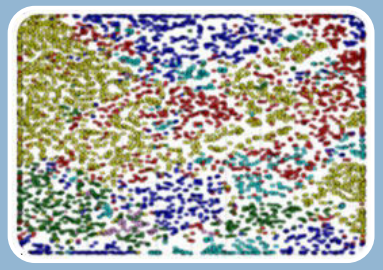
J. Craig Venter INSTITUTE

MGTAXA Components



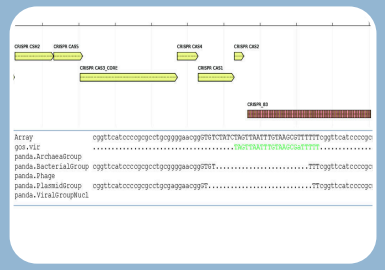
Predict taxonomy for bacterial metagenomic sequences

- Glimmer ICM based classifier (similar to Phymm approach). Parallelized on HTC and HPC clusters, command line and Galaxy Web interface. Sequences above 300 bp.
- BLAST+ and Batch SOM implementations for HPC clusters with MPI-MapReduce framework. Calls to pristine NCBI BLAST+ API – full compatibility. Scales to 2000 cores on XSEDE (former TeraGrid) (TACC Ranger).
- Assembly contig consensus classifier from ORF BLASTP or APIS assignments



First algorithm to predict hosts for bacteriophages in metagenomes

- Explores compositional similarity between phage and host
- ICM and SOM for prediction and visualization
- Assigns phage scaffolds (5Kbp+) to bacterial sequences
- Uses infrastructure and interfaces of the bacterial classifier



CRISPR pipeline

- Scans genomes & metagenomes for CRISPR arrays and genes
- Connects with viral metagenomes through spacer matches – alternative way to establish the bacteriophage-host relationship
- Parallelized on HTC and HPC clusters

ICMs for more sensitive compositional models with less dependency on the length of training sequence

- Interpolated Context Model (ICM) developed for Glimmer (Delcher 1999) to find correct reading frame based on nucleotide composition
- ICMs used as a basis of Phymm taxonomic prokaryotic classification algorithm (Brady & Salzberg, 2009)
- The main idea of ICM is to use longer k-mers when there is a statistically significant frequency of those specific k-mers in the training sample
- This helps extract compositional signal from sequences shorter than would be required by a fixed k-mer model
- Also less bias toward longer training sequences – critical for classification of viruses
- We use ICMs to assign sample taxonomy as well as bacteriophage host taxonomy

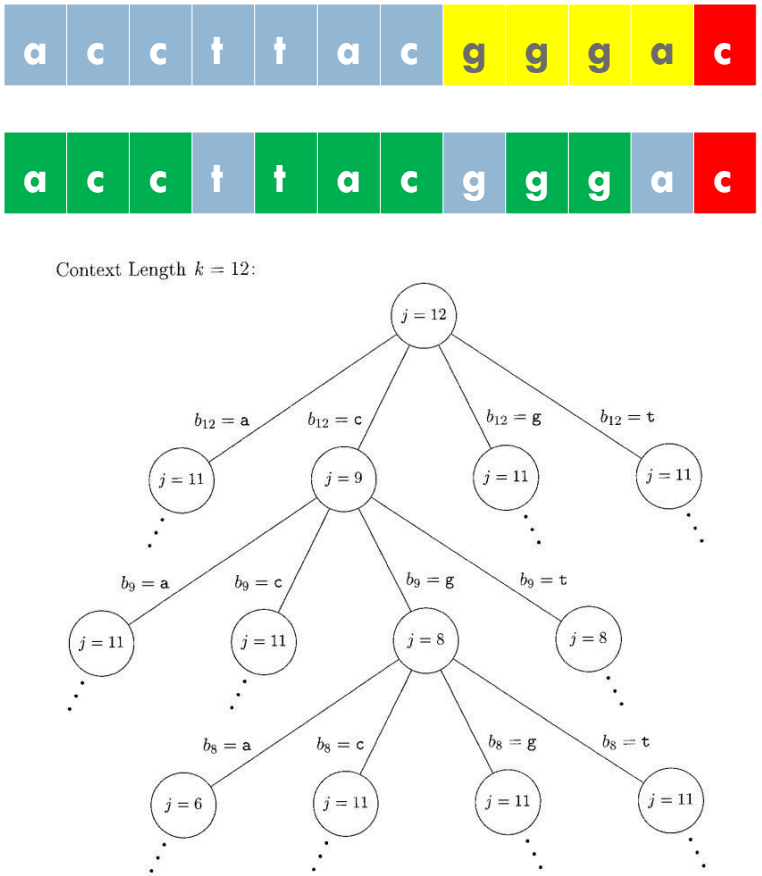
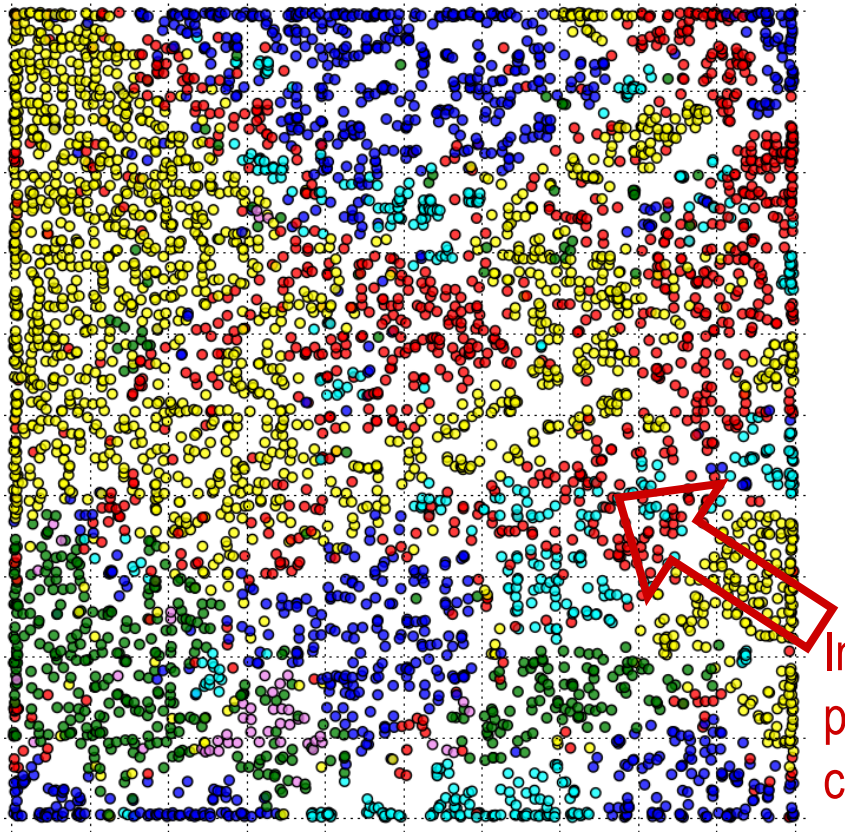
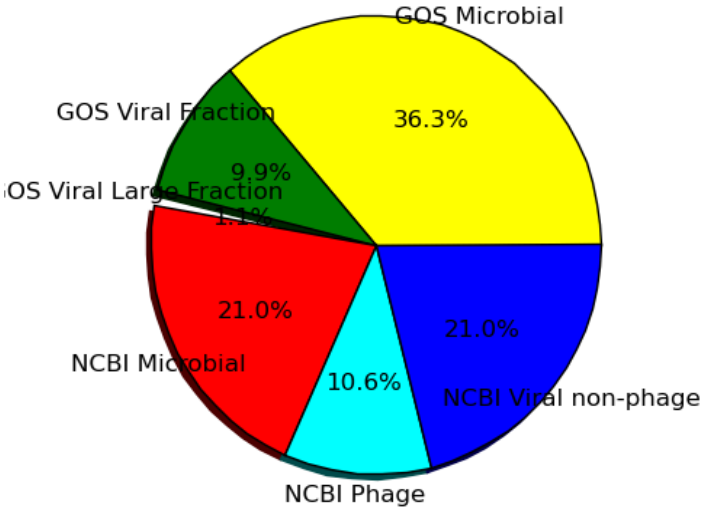


Image from Delcher 1999

How predicting the viral host taxonomy relates to predicting the bacterial taxonomy



Color legend and proportions of inputs to the SOM



In this region, enterobacteria and their phages cluster together. Many other clades behave like that on this map.

Self-organizing Map (SOM) of tetranucleotide (4-mers) frequency vectors
Mix of NCBI RefSeq 5 Kb chunks and GOS scaffolds

Phage host prediction algorithm

- Viruses exhibit huge diversity and fast evolution. Homology based methods for assigning taxonomy to viral metagenomic sequences suffer from the lack of sufficiently close sequences in the reference databases
- Although it has been shown by several groups that viruses tend to adopt polynucleotide (*k*-mer) composition of their hosts, regardless of homology, ours is the first practical method that uses this tendency to build a classifier for metagenomic viral contigs (5Kbp or longer)
- We score viral contigs against prokaryotic ICMs for the entire RefSeq and pick the ICM with the best score as the host prediction
- We also have a mixed mode classifier, where we score both against ICMs trained on viral sequences and prokaryotic ICMs. Unlike the only other reported compositional taxonomic classifier of viral sequences (NBC, fixed *k* models), ICM scoring does not exhibit any strong preference to longer reference sequences (NBC reported to misclassify a lot of its benchmark sequences as giant mimivirus).

Phage host prediction benchmarking

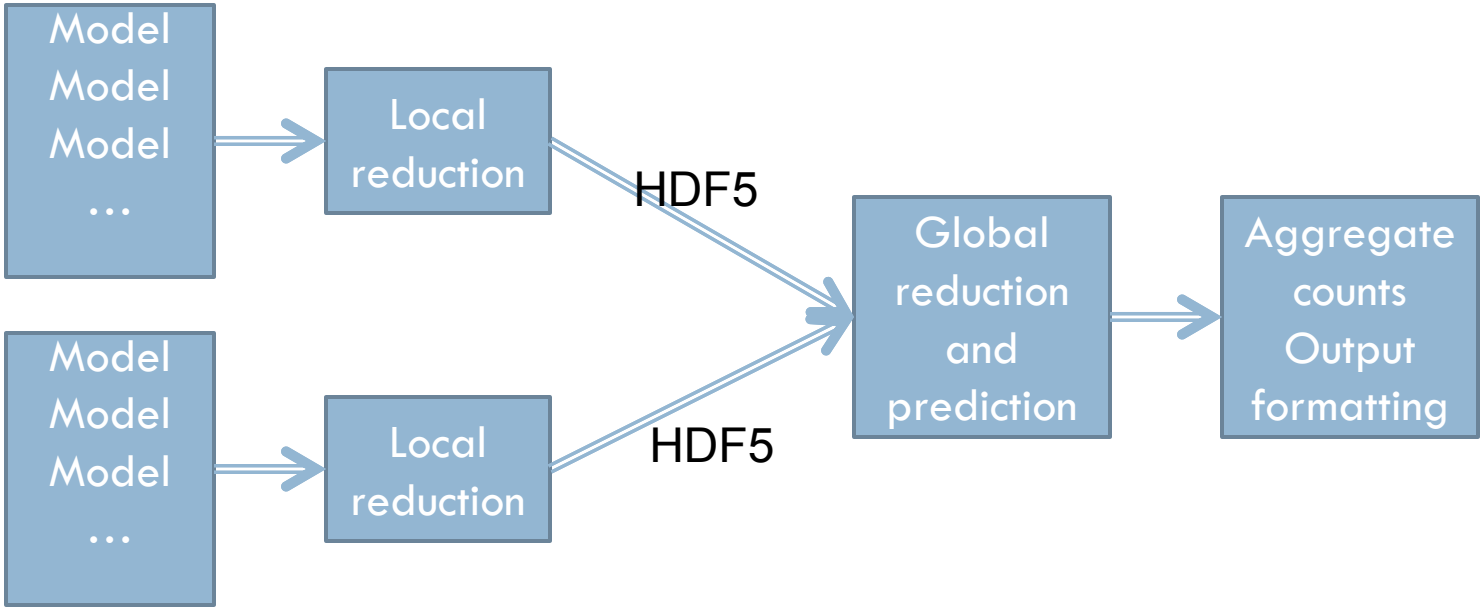
- Benchmark was compiled from all NCBI RefSeq Genbank records (Nov 2010) where a virus had a prokaryotic species identified as a host either by host record by the naming of the virus
- Among those, lysogenic (those that integrate into host DNA during their lifecycle) viruses were identified by literature curation for viruses that named a specific host genome

A. Phages with lysogenic cycle (24 viral species, 10 host genera)						
Exclude\Predict Host Rank	**genus**	**family**	**order**	**class**	**phylum**	**rejected**
none	89%	96%	94%	99%	97%	2%
species	85%	90%	90%	97%	95%	5%
B. All phages (294 viral species, 33 host genera)						
Exclude\Predict Host Rank	**genus**	**family**	**order**	**class**	**phylum**	**rejected**
none	50%	55%	59%	71%	76%	6%
species	43%	49%	54%	68%	75%	7%

- Random assignment would result in 0.2% accuracy at a genus level
- Algorithm, benchmarking and results on Global Ocean Sampling data are described in (Williamson et al, Metagenomic Investigation of Viruses throughout the Indian Ocean. PLoS One 2012; accepted)

Parallelization approach

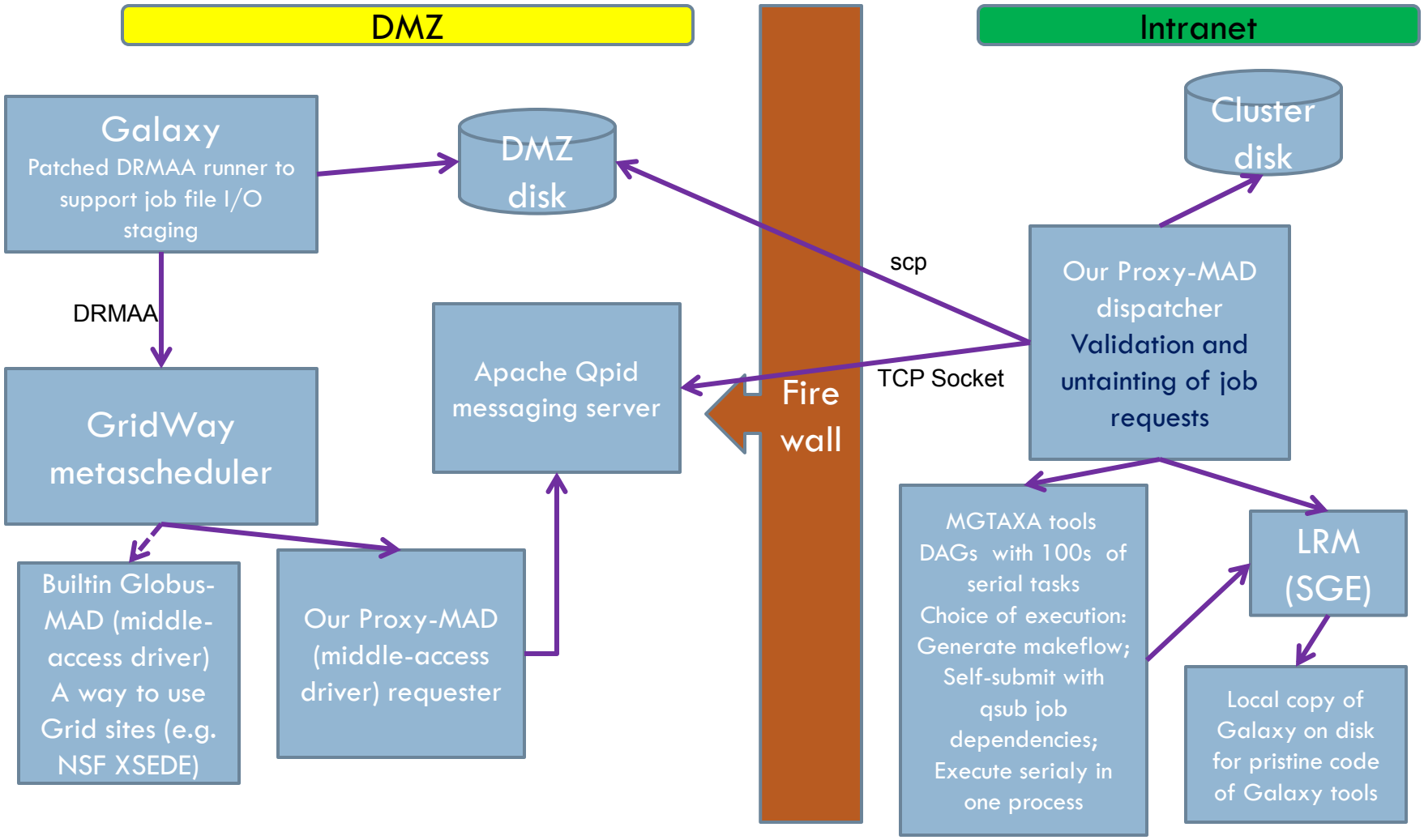
- 2500 models, 50G on disk, each input sequence has to be scored against each model
- Coarse-grained parallelism
  - Training – one task is to build one ICM (1 min to build one model for a 4.7Mbp genome, creating 29MB model file)
  - Scoring – one task is to score input sequences against one model (12.5 min to score 1Gbp against one model)



Backends for parallel execution

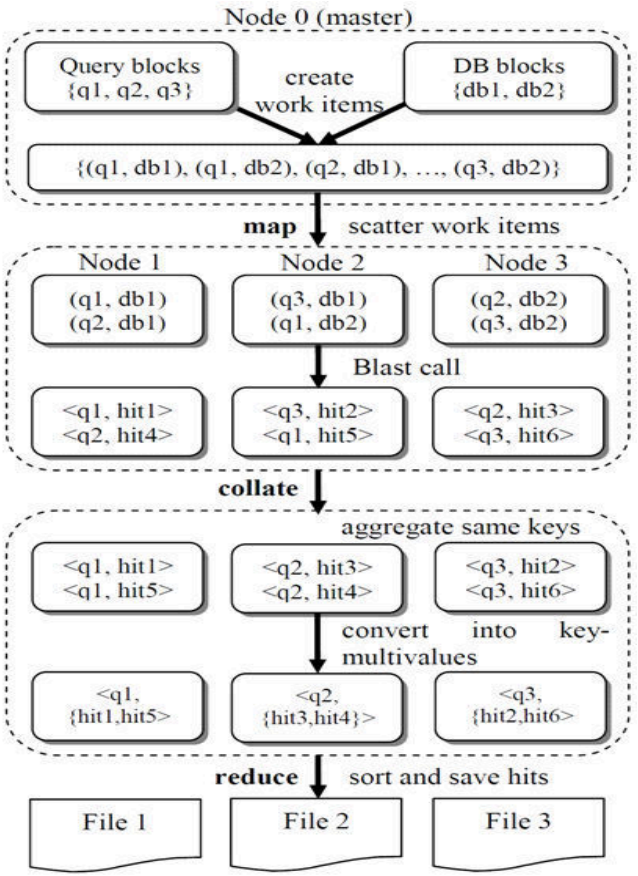
- How MGTAXA workflows (DAGs) will be executed is selected at run-time by a command-line switch: mgt-classifier –run-mode batchDep –batch-backend makeflow [other options …]
- Implemented choices:
  - Serially in one process. No cluster or external workflow support is needed
  - Submit itself as a DAG of SGE jobs using qsub job dependency option
  - Generate a "make file" for the Makeflow workflow execution engine. Then run makeflow on it with a choice of its own back-ends:
    - Parallel multi-processing on a single node (normally N processes == N cores)
    - Submitting multiple SGE jobs while satisfying the dependencies defined by the DAG
    - Glide-in mechanism ("dual-level scheduling") within one or more large MPI jobs. With that, MGTAXA can run on large XSEDE clusters that only schedule efficiently large parallel MPI jobs
    - WorkQueue with a shared file-system option. Advantage for large number of very short jobs.
    - Generic LRM interface (define job submit command for the current LRM)
  - Makeflow is tolerant to compute node failures, and also has restart capability (like make)
- As a general note, Makeflow seems like a great fit to design Galaxy tools that need to execute their own massively parallel programmatic workflows (as opposed to interactively user-defined Galaxy built-in workflows). Makeflow is serverless, zero-administration, and appears to Galaxy as a single serial job that can be submitted to SGE like any other tool.

Web server - architecture



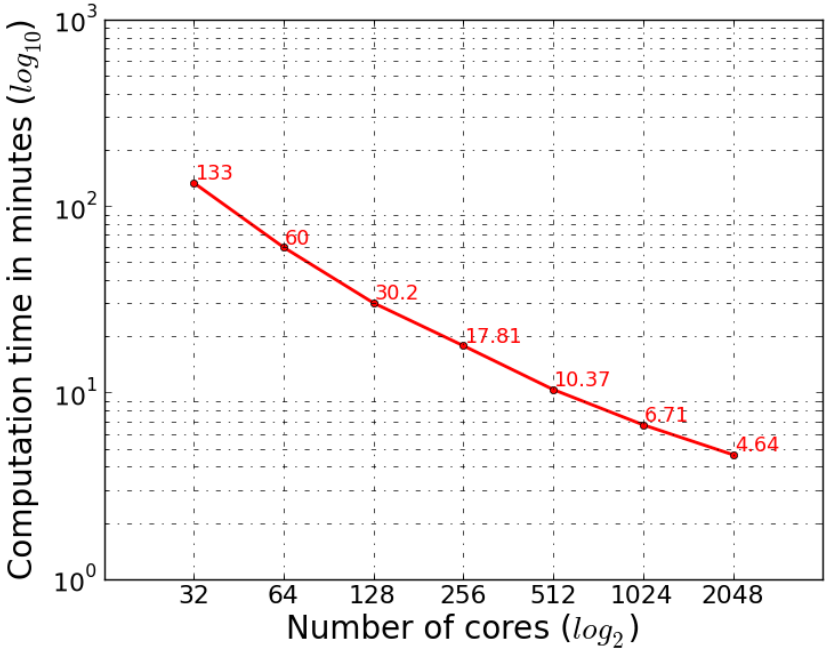
MR-MPI BLAST+ implementation

- Runs as a regular MPI program, on any supercomputer with a shared file system
- Makes high-level API calls to unmodified NCBI C++ Toolkit – results are fully compatible with the upstream NCBI code; easy to keep up to date and support any options of the NCBI BLAST+
- Implemented with MapReduce MPI (MR-MPI) library from Sandia Lab that helps organizing computations and data flow
- The library scheduler was modified to solve the problem of maintaining context between map() calls – a common problem with the classical MapReduce algorithm
- Parallel sort for the final output results
- Output: Tabular, NCBI XML, internal binary format, HDF5
- Special mode for BLASTN metagenomic read recruitment (compatible with implementation in Rusch et al, PLoS Biol 2007)

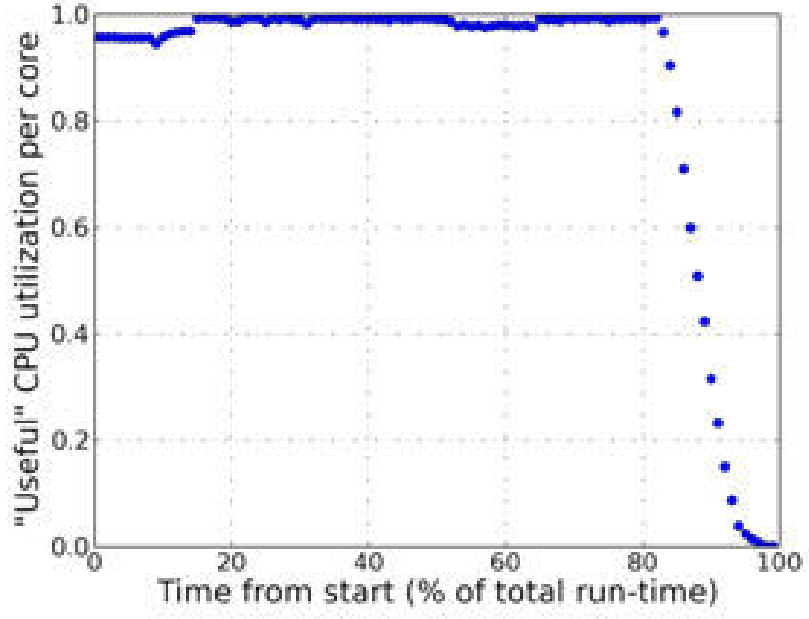


Control flow of the MR-MPI BLAST

MR-MPI BLASTN and BLASTP scaling



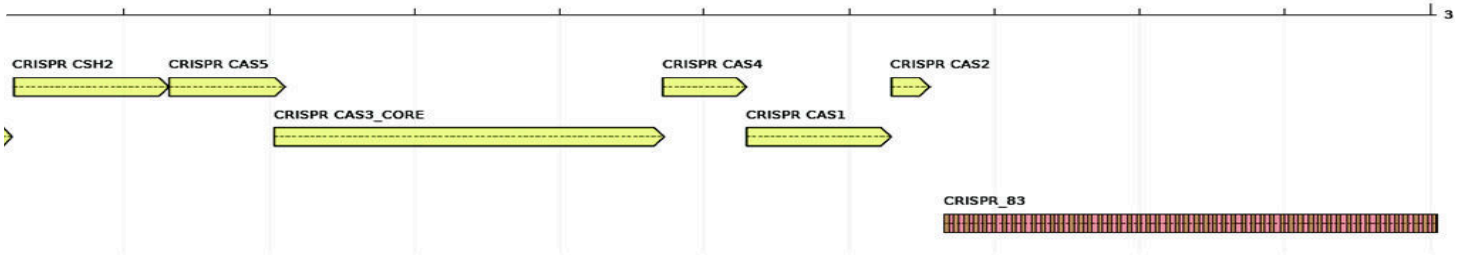
Scaling chart for MR-MPI BLASTN showing process wall clock time at different total core counts in MPI job. The total number of query sequences is 40,000. The sequences are split into 40 blocks of 400 kbp. Each block, when combined with one DB partition, forms a sequential work unit for the MapReduce algorithm. The data point labels represent time in minutes.



"Useful" CPU utilization per core during the course of the computation for the MR-MPI BLASTP run with 1024 cores. CPU user time used at any given moment within a BLAST call was divided by the corresponding wall clock time, summed over all concurrent calls, and divided by a total number of cores allocated to the MPI program. From (*Sul and Tovchigrechko, IPDPS, 2011)*

CRISPR Analysis Pipeline

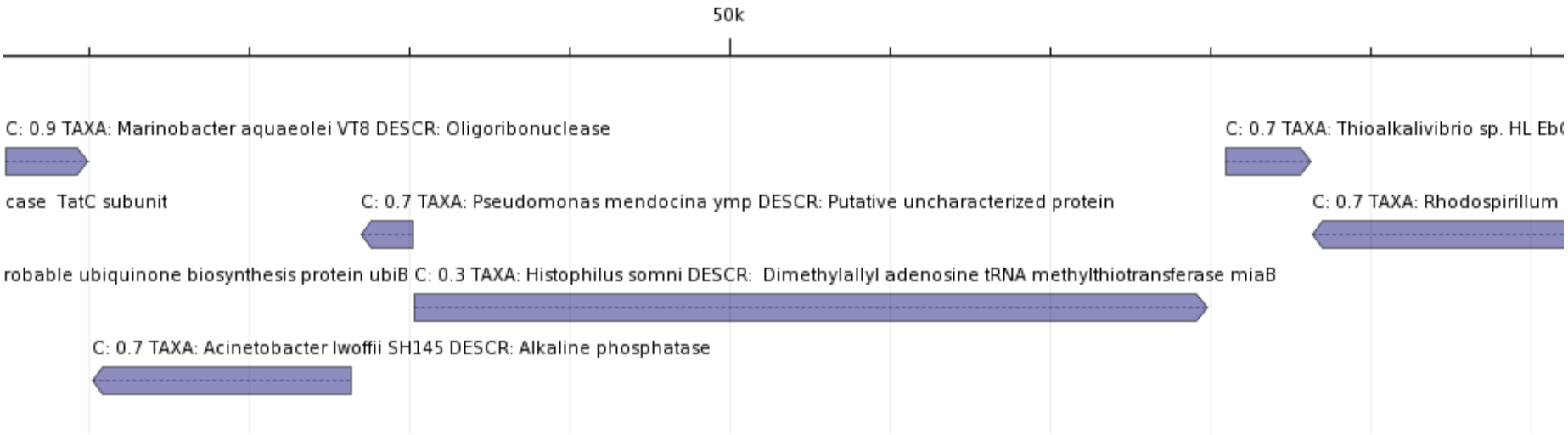
- Detects CRISPR arrays and CAS genes (genes as annotated by JCVI protein annotation pipeline), builds genome diagrams, finds spacer matches to the viral sequences
- Applied in (Yooseph et al, Nature 2010) for the Moore marine genomes collection, 137 marine genomes sequenced and annotated by JCVI (and previously sequenced 60 other marine genomes). Demonstrated that the presence of CRISPR system is clearly associated with the life-style of the organism as related to the chances of its repeated encounters with the same types of viruses.
- Applied to GOS bacterial assembly and viral reads, several smaller datasets
- The CRISPR finder is batched PILERCR with our post-processing to remove false positives




```
Array                cggttcatccccgcgcctgcggggaacggGTGTCTATCTAGTTAATTTGTAAGCGTTTTTTcggttcatccccgcg
gos.vir              ....................................TAGTTAATTTGTAAGCGaTTTTT..............
panda.ArchaeaGroup
panda.BacterialGroup cggttcatccccgcgcctgcggggaacggGTGT.......................TTTcggttcatccccgcg
panda.Phage
panda.PlasmidGroup   cggttcatccccgcgcctgcgaggaacggGT.........................TTcggttcatccccgcg
panda.ViralGroupNucl
```

Assembly contig ORF consensus classifier

- Suppose that we have applied a homology-based metagenomic classification pipeline such as JCVI pipeline working on best BLASTP hits (Tanenbaum et al, Stand. Genomic Sci. 2010) or APIS (Badger et al, J. Bacteriol. 2006; Allen et al, ISME J. 2012) working on phylogenetic inference.
- And we identify taxonomic composition of our sample by counting individual gene assignments
- In the example below, we would decide that we have all these bugs in whatever proportions
- But, if the annotation was done on sufficiently large metagenomic contigs, we can classify each contig by a lowest common ancestor node of its ORF assignments where a specified majority vote (e.g. 75%) is reached.
- Example below is a fragment of a globally abundant SAR86 genome (published in 2011) searched against the version of Uniref100 from 2010. Our consensus classifier correctly called in gamma-proteobacteria.



Team

- Seung-Jin Sul (HPC tools)
- Tim Prindle (Proxy MAD)
- Shibu Yooseph (co-PI, metagenomics expertise)
- Shannon Williamson (original idea to work on bacteriophage-host prediction tool)

- Funding: NSF 0850256, DOE DE-FC02-02ER63453