HPC in Biology

Wei Feinstein, Ph.D.

HPC User Services
LSU HPC/LONI

Louisiana State University
Agenda

• High Performance Computing
• Bioinformatics/computational biology
• Bioinformatics’ tools
  • Sequence assembly
  • Sequence alignment
• Simulation tools
  • Molecular Dynamics
  • Molecular docking
Few things they agreed on

President Bush 2004
President Obama 2010
Few things they agreed on

President Bush 2004
President Obama 2010

... Health Care can only improve with the innovative application of Information Technology.
Scope of Computational Biology

• Mathematical Biology
• Biostatistics
• Biomathematics
• Quantitative Biology
• Biophysics
• Systems biology
Computational Biology at LSU

Support core set of software tools

Provide training – in person and online

Collaborate to better leverage advanced computing for biology
## Available LSU HPC resources

<table>
<thead>
<tr>
<th>SuperMIC</th>
<th>SuperMike II</th>
<th>Philip</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hostname</strong></td>
<td>mikel.hpc.lsu.edu</td>
<td>philip.hpc.lsu.edu</td>
</tr>
<tr>
<td><strong>Peak Performance/TFlops</strong></td>
<td>1000</td>
<td>146</td>
</tr>
<tr>
<td><strong>Compute nodes</strong></td>
<td>360</td>
<td>440</td>
</tr>
<tr>
<td><strong>Processor/node</strong></td>
<td>2 Deca-core</td>
<td>2 Octa-core</td>
</tr>
<tr>
<td><strong>Processor Speed</strong></td>
<td>2.8GHz</td>
<td>2.6GHz</td>
</tr>
<tr>
<td><strong>Processor Type</strong></td>
<td>Intel Xeon 64bit</td>
<td>Intel Xeon 64bit</td>
</tr>
<tr>
<td><strong>Nodes with Accelerators</strong></td>
<td>360</td>
<td>50</td>
</tr>
<tr>
<td><strong>Accelerator Type</strong></td>
<td>Xeon Phi 7120P</td>
<td>2 nVidia M2090</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td>RHEL v6</td>
<td>RHEL v6</td>
</tr>
<tr>
<td><strong>Vendor</strong></td>
<td>Dell</td>
<td>Dell</td>
</tr>
<tr>
<td><strong>Memory per node</strong></td>
<td>64 GB</td>
<td>32/64/256 GB</td>
</tr>
<tr>
<td><strong>Detailed Cluster Description</strong></td>
<td>User Guide</td>
<td>User Guide</td>
</tr>
<tr>
<td><strong>Available Software</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

4/20/2016

**HPC in Biology**
# Available LONI HPC resources

<table>
<thead>
<tr>
<th>QB2</th>
<th>Eric</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hostname</strong></td>
<td>qb2.loni.org</td>
</tr>
<tr>
<td><strong>Peak Performance/TFlops</strong></td>
<td>1,500</td>
</tr>
<tr>
<td><strong>Compute nodes</strong></td>
<td>504</td>
</tr>
<tr>
<td><strong>Processor/node</strong></td>
<td>2.10-Core</td>
</tr>
<tr>
<td><strong>Processor Speed</strong></td>
<td>2.8GHz</td>
</tr>
<tr>
<td><strong>Processor Type</strong></td>
<td>Intel Ivy Bridge–EP Xeon 64bit</td>
</tr>
<tr>
<td><strong>Nodes with Accelerators</strong></td>
<td>480</td>
</tr>
<tr>
<td><strong>Accelerator Type</strong></td>
<td>NVIDIA Tesla K20x</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td>RHEL v6</td>
</tr>
<tr>
<td><strong>Vendor</strong></td>
<td>Dell</td>
</tr>
<tr>
<td><strong>Memory per node</strong></td>
<td>64 GB</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>Information Systems Building, Baton Rouge</td>
</tr>
</tbody>
</table>

**Detailed Cluster Description**

**User Guide**

**Available Software**
## Installed Bio Software Stack

<table>
<thead>
<tr>
<th>Genomics Bioinformatics</th>
<th>Molecular Dynamics Structural Biology</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST+</td>
<td>NAMD</td>
</tr>
<tr>
<td>mpiBLAST</td>
<td>Amber</td>
</tr>
<tr>
<td>HMMER</td>
<td>GROMACS</td>
</tr>
<tr>
<td>MAFFT</td>
<td>Desmond</td>
</tr>
<tr>
<td>MUSCLE</td>
<td>VASP</td>
</tr>
<tr>
<td>R</td>
<td>LAMMPS</td>
</tr>
<tr>
<td>BioPerl</td>
<td>APBS</td>
</tr>
<tr>
<td>FASTX-Toolkit</td>
<td>NWChem</td>
</tr>
<tr>
<td>Picard</td>
<td>GAMESS</td>
</tr>
<tr>
<td>SAMtools</td>
<td>AutoDock_Vina</td>
</tr>
<tr>
<td>SHRImp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>BLAST+</td>
<td>VEILVET</td>
</tr>
<tr>
<td>mpiBLAST</td>
<td>ABYSS</td>
</tr>
<tr>
<td>HMMER</td>
<td>TRINITY</td>
</tr>
<tr>
<td>MAFFT</td>
<td>BWA</td>
</tr>
<tr>
<td>MUSCLE</td>
<td>SSAKE</td>
</tr>
<tr>
<td>R</td>
<td>SOAP de novo</td>
</tr>
<tr>
<td>BioPerl</td>
<td>AMOS</td>
</tr>
<tr>
<td>FASTX-Toolkit</td>
<td>MAQ</td>
</tr>
<tr>
<td>Picard</td>
<td>BOWTIE</td>
</tr>
<tr>
<td>SAMtools</td>
<td>CUFFLINKS</td>
</tr>
<tr>
<td>SHRImp</td>
<td>TOPHAT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Where to Start

• Apply HPC/LONI account
• Apply/Join allocation(s)
• Training of HPC User Environment 1 & 2

http://www.hpc.lsuc.edu/training/archive/tutorials.php
How to Login

• Unix and Mac
  • ssh on a terminal to connect

• Windows box (ssh client):
  • Putty
    http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
  • MobaXterm
    http://mobaxterm.mobatek.net/
Accessing cluster on Windows - Putty
Enable X11 forwarding

- On Linux or Mac, simply pass the -X option to the ssh command line
  - ssh -X username@mike.hpc.lsu.edu
- On windows using putty
  - Connection->SSH->X11->Enable X11 forwarding
  - Install X server (e.g. Xming)
MobaXterm supports

- command line scp and rsync
- sftp file transfer through GUI
- Built-in X11 forwarding

MobaXterm Personal Edition v7.1
(Unix utilities and X-server on Gnu/Cygwin)

- Your computer drives are accessible through the /drives path
- Your DISPLAY is set to 192.168.1.115:0.0
- When using SSH, your remote DISPLAY is automatically forwarded
- Each command status is specified by a special symbol (v or x)

Important:
This is MobaXterm Personal Edition. The Professional edition allows you to customize MobaXterm for your company: you can add your own logo, your parameters, your welcome message and generate either an MSI installation package or a portable executable. We can also modify MobaXterm or develop the plugins you need.

For more information: http://mobaxterm.mobatek.net/versions.php
Software Stack Management

• SoftEnv
  – A software used to manage software package
  – SuperMike2 and Eric
  – softenv -- list of software packages
  – soft add +xxx -- add to user working env

• Modules
  – Most supercomputing sites including XSEDE
  – SuperMIC, Philip and QB2
  – module av
  – module load xxx
How to Submit Jobs

- **Interactive job**
  - Set up an interactive environment on compute nodes
  - Purpose: testing and debugging

- **Batch job**
  - Executed without user intervention using a job script
    - Advantage: the system takes care of everything
    - Disadvantage: can only execute one sequence of commands which cannot be changed after submission
  - Purpose: production run

- Do not run on head nodes!!!
Submitting Jobs on Linux Clusters

• Interactive job example:
  
  qsub -I -V -I -A <Allocation> -q <queue name> \ 
  walltime=<hh:mm:ss>,nodes=<num_nodes>:ppn=<num_cores>

  Add -X to enable X11 forwarding

• Batch Job example:
  
  qsub job_script

http://www.hpc.lsu.edu/docs/pbs.php
Example of pbs.script

#!/bin/bash
#PBS –A allocation
#PBS –q workq
#PBS -l nodes=1:ppn=16
#PBS –N jobName
#PBS -j oe
#PBS -M myEmail@lsu.edu

module load blast-xxx / soft add +xxxx
cd $PBS_O_WORKDIR

blastn -query seq100.ffn -db db/scaffold -out result/ \ result.100 -outfmt 7 -max_target_seqs 100 -num_threads 16
Monitor Your Jobs

- showq –q: list of available queues
- qstat [–u username –n]:
  info about active, eligible, blocked job on a cluster
- qdel #jobid: delete a job
- checkjob #jobid
Agenda

• High Performance Computing
• Bioinformatics/computational biology
• Bioinformatics’ tools
  • Sequence assembly
  • Sequence alignment
• Simulation tools
  • Molecular Dynamics
  • Molecular docking
Definitions

• **What is Bioinformatics?** - Interdisciplinary science of using computational approaches to analyze, classify, collect, represent and store biological data to better understand DNA, RNA and protein molecules.

• **What is Computational Biology?** - Development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to study biological, behavioral, and social systems.
The DNA Sequencing Revolution
Impact on nearly every field of biological research

Human Genetics & Genomics

Plants & Agriculture

Microbes, Viruses & Infectious Diseases

Environmental Genomics

www.roche-applied-science.com
Sequencing

• What is DNA/[RNA] sequencing?
  • Determine/decode the precise order of nucleotides in DNA /RNA molecules
    • DNA: adenine(A), guanine(G), thymine(T) and cytosine(C)
    • RNA: adenine(A), guanine(G), uracil(U) and cytosine(C)
  • Why?
    • Initial step towards understanding the target molecule
    • Blue print for protein translation
• How?
  • Electrophoresis: to separate pieces of DNA differing in length by only one base
DNA Sequencing

• Short reads of DNA [20-30,000] base pairs
• Manual: slow, tedious, error prone and short length
• NGS: next generation sequencing
• Human genome: \( \sim 3,238,830 \) bps
  • An entire human genome can be sequenced within one single day using NGS
  • Sanger sequencing technology would need over a decade to complete
Sequencers

Sanger

1998
ABI 3700
6 MB/Day
$500/MB

Next Generation Sequencers

2005
Roche 454
750 MB/Day
$20/MB

2007
Illumina GAII
5000 Mb/Day
$0.50/MB

2010
Illumina Hiseq
75000 Mb/Day
$0.02/MB

Public domain images from NIH/NHGRI Digital Media Database (www.genome.gov)
Sequence Assembly

- Fragments of DNA sequenced from sequencers
- Construct the original sequence by aligning/merging DNA fragments in the proper order
- Error checking and close gaps

Terms:
- Sequence: genetic order in biological letters (DNA/RNA, amino acids)
- Reads: short pieces of a sequence
- Contigs: assembled reads
Sequence Assembly

• How assemblers work?

finding and analyzing overlaps, which are identical DNA sequences at either end of two different reads
Assembling Methods

• De Novo Assembly
  – Piecing together reads into a larger sequence without relying on an already-assembled sequence of a related organism.

  **Nucleotid Reads**

  **Assembled Segments**

  **Original Sequence**
Assembling Methods

• Reference Assembly/Mapping
  – Using an already-assembled sequence as a guide to match target reads.

Reference Sequence

Reads Mapped to Reference Sequence
Agenda

• High Performance Computing
• Bioinformatics/computational biology
• Bioinformatics’ tools
  • Sequence assembly
  • Sequence alignment
• Simulation tools
  • Molecular Dynamics
  • Molecular docking
HPC Applications

• Sequence assemblers
  • Abyss/A5/mothur
• Sequence alignment
  • Blast/bowtie
• Molecular Dynamics (MD)
  • NAMD (VMD)
  • Gromacs
• Molecular docking
  • Autodock Vina
How does ABySS work?

• Assemble read sequences without paired-end information for an initial assembly
• Map the reads back to the initial assembly
• Use the paired-end information to merge contigs from the first stage into larger sequences
• Output the final assembly
What are included in ABySS

- abyss-pe is a driver script
- ABYSS - the single-end assembler
- AdjList - finds overlaps of length k-1 between contigs
- KAligner** - aligns reads to contigs
- ParseAligns** - finds pairs of reads in alignments
- DistanceEst** - estimates distances between contigs
- Overlap - find overlaps between blunt contigs
- SimpleGraph - finds paths between pairs of contigs
- MergePaths - merges consistent paths
- Consensus - for a colour-space assembly, convert the colour-space contigs to nucleotide contigs
- abyss-fac: calculate assembly contiguity statistics
- abyss-filtergraph: remove shim contigs from the overlap graph
- abyss-fixmate: fill the paired-end fields of SAM alignments
- abyss-map: map reads to a reference sequence
- abyss-scaffold: scaffold contigs using distance estimates
- abyss-todot: convert graph formats and merge graphs
abyss-pe Parameters

a: maximum number of branches of a bubble [2]
b: maximum length of a bubble (bp) [10000]
c: minimum mean k-mer coverage of a unitig [sqrt(median)]
d: allowable error of a distance estimate (bp) [6]
e: minimum erosion k-mer coverage [sqrt(median)]
E: minimum erosion k-mer coverage per strand [1]
j: number of threads [2]
k: size of k-mer (bp)
l: minimum alignment length of a read (bp) [k]
m: minimum overlap of two unitigs (bp) [30]
n: minimum number of pairs required for building contigs [10]
N: minimum number of pairs required for building scaffolds [n]
p: minimum sequence identity of a bubble [0.9]
q: minimum base quality [3]
s: minimum unitig size required for building contigs (bp) [200]
S: minimum contig size required for building scaffolds (bp) [s]
t: minimum tip size (bp) [2k]
v: use v=-v for verbose logging, v=-vv for extra verbose [disabled]
Running ABySS

- Assemble single-end reads
  ```bash
  ABYSS -k20 read.fa -o contigs.fa
  ```

- Assemble paired-end reads
  ```bash
  abyss-pe name=abyss -C result k=32 n=10 \
  in='../reads_1.fa ../reads_2.fa'
  ```
  **K**: size of k-mer (bp)
  **n**: minimum # pairs required to build contigs [10]

- Parallel abyss (MPI)
  ```bash
  abyss-pe np=8 name=test k=32 n=10 \
  in='reads_1.fa reads_2.fa'
  ```
Output files of ABySS

- `${name}-contigs.fa`
  The final contigs in FASTA format
- `${name}-bubbles.fa`
  The equal-length variant sequences (FASTA)
- `${name}-indel.fa`
  The different-length variant sequences (FASTA)
- `${name}-contigs.dot`
  The contig overlap graph in Graphviz format
Optimize k-mer

• Run multiple assemblies using k [20-40]
  
  ```bash
  export k
  for k in {20..40}; do
    mkdir k$k
    abyss-pe -C k$k name=ecoli in=../reads.fa
  done
  ```

• Inspect the assembly contiguity statistics
  
  ```bash
  abyss-fac -t length k*/contigs.fa
  ```

  k = 64 as default
# Memory usage of ABysS

<table>
<thead>
<tr>
<th>Genome size</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 kbp</td>
<td>( \frac{1}{4} ) GB</td>
</tr>
<tr>
<td>5 Mbp</td>
<td>1 GB</td>
</tr>
<tr>
<td>200 Mbp</td>
<td>32 GB</td>
</tr>
<tr>
<td>3 Gbp</td>
<td>128 GB</td>
</tr>
</tbody>
</table>
A5-pipeline

• **Andrew And Aaron's Awesome Assembly pipeline**
• Assembling DNA sequence data generated on Illumina sequencing platform
• Can’t do:
  • Illumina reads shorter than 80nt
  • Base quality is low in all/most reads before 60nt
  • For homozygous haploid genomes.
A5-pipeline Assembly Stages

1: Remove ambiguous and low quality portions of reads
2: Using assembler IDBA, a de Bruijn graph-based algorithm, to assemble contigs
3: Contigs are scaffolded and extended using the software SSPACE, a stand-alone program for scaffolding pre-assembled contigs
4: Crude scaffolds are subjected to quality control check for misassemblies, also use BWA (Burrows-Wheeler Aligner) to map low-divergent sequences against a large reference genome
5: Broken-up scaffolds are rescaffolded using SSPACE.
How to Run A5-pipeline

[@mike021 ~]$ soft add +bio-pipeline

[@mike021 ~]$ a5_pipeline.pl phiX_p1.fastq phiX_p2.fastq a5

....

[samopen] SAM header is present: 1 sequences.
[a5] java -Xmx42490m -jar A5qc.jar a5.s4/a5.qc.libraw1.sam a5.crude.scaffolds.fasta a5.s4/a5.qc.libraw1.broken.fasta 1 > a5.s4/a5.qc.libraw1.qc.out
[a5_s5] No misassemblies found.
[a5] Final assembly in a5.final.scaffolds.fasta

Output: a5.final.scaffolds.fasta
Mothur

• A comprehensive software package that allows users to use a single piece of software to analyze microbial ecology community sequence data
• Initiated by Dr. Patrick Schloss and his software development team in the dept. of Microbiology & Immunology at the University of Michigan
• Offers the ability to go from raw sequences to the generation of visualization tools to describe α and β diversity.
How to Run Mothur

[@mike021 ~]$ soft add +bio-pipeline

[@mike021 ~]$ cd biology/mothur/MiSeq_SOP

[@mike021 ~]$ mothur
mothur v.1.33.3
Last updated: 4/4/2014
...
Distributed under the GNU General Public License
Type 'help()' for information on the commands that are available

Type 'quit()' to exit program

mothur > ...

http://www.mothur.org/wiki/MiSeq_SOP
RAST

Submit assembly to RAST (Rapid Annotation using Subsystem Technology) to be annotated at:
http://rast.nmpdr.org
Agenda

• High Performance Computing
• Bioinformatics/computational biology
• Bioinformatics’ tools
  • Sequence assembly
  • Sequence alignment
• Simulation tools
  • Molecular Dynamics
  • Molecular docking
What is Sequence Alignment?

Align the sequences of DNA, RNA or protein to identify regions of similarity that may be a consequence of functional, structural or evolutionary relationships between the sequences.
Sequence Alignment

- Global sequence alignment  Needleman–Wunsch
- Local sequence alignment  Smith–Waterman
- Glocal sequence alignment
- Short sequence aligners e.g., bowtie
- Long sequence aligners e.g., Blast
BLAST (Basic Local Alignment Search Tool)

• What is BLAST?
  ▪ Basic Local Alignment Search Tool
  ▪ Algorithm for comparing biological sequence information with a database of known sequences

• Basic BLAST terminology:
  ▪ **Query**: sequence to be compared to database
  ▪ **Sequence database**: a collection of known sequences (nucleotides or amino acids)
  ▪ **Alignment (hit)**: when the query matches a database sequence at an acceptable similarity threshold
  ▪ **E-value**: score of an alignment, the lower the better
### BLAST

#### Basic BLAST

Choose a BLAST program to run.

<table>
<thead>
<tr>
<th></th>
<th>Search</th>
<th>Database Type</th>
<th>Query Type</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nucleotide blast</strong></td>
<td>a nucleotide database</td>
<td>using a nucleotide query</td>
<td></td>
<td>blastn, megablast, discontiguous megablast</td>
</tr>
<tr>
<td><strong>protein blast</strong></td>
<td>protein database</td>
<td>using a protein query</td>
<td></td>
<td>blastp, psi-blast, phi-blast, delta-blast</td>
</tr>
<tr>
<td><strong>blastx</strong></td>
<td>protein database</td>
<td>using a translated nucleotide query</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>tblastn</strong></td>
<td>translated nucleotide database</td>
<td>using a protein query</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>tblastx</strong></td>
<td>translated nucleotide database</td>
<td>using a translated nucleotide query</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How to Run BLAST

• Reference blast DB


• Self create db:

  makeblastdb -in scaffold.fna –dbtype nucl/proc –parse_seqids \\
  -out db/scaffold

• Run blast

  blastn -query seq100.ffn -db db/scaffold -out result/result.100 \\
  -outfmt 7 -max_target_seqs 100 -num_threads 16

• File formats:

  .faa: protein sequence in fasta format
  .ffn: protein coding portion of a genome segment
  .fna: genome fasta sequence
Submit a BLAST job

```bash
#!/bin/bash
#PBS -A allocation
#PBS -q workq
#PBS -l nodes=1:ppn=16
#PBS -N jobName
#PBS -j oe
#PBS -M myEmail@lsu.edu

module load blast-xxx / soft add +xxxx

cd $PBS_O_WORKDIR

blastn -query seq100.ffn -db db/scaffold -out result/ result.100 -outfmt 7 -max_target_seqs 100 -num_threads 16
```
Bowtie / Bowtie2

- **Bowtie**: an ultrafast memory-efficient short read aligner. Use BWT burrows-Wheeler Transform index to keep its memory footprint small.
- **Bowtie2**: alignment seq longer than 50-1000bps. Use FM index (based on BWT) to keep memory small.
How to run Bowtie

- Using pre-built *E. coli* index, which sit in indexes/ folder
  `bowtie -t indexes/e_coli reads/e_coli_1000.fq e_coli_1000.map`

- Build new index
  `bowtie-build genomes/NC_008253.fna e_coli_new`

- Convert xxx.fastq format to SAM format
  `bowtie -S e_coli reads/e_coli_10000snp.fq ec_snp.sam`
samtools

A suite of tools for sorting, manipulating, and analyzing alignments, such as output by bowtie

- SAM format: human-readable sequence format
- Binary BAM format
Usage:  samtools <command> [options]

Command:
- view: SAM<->BAM conversion
- sort: sort alignment file
- pileup: generate pileup output
- mpileup: multi-way pileup
- faidx: index/extract FASTA
- tview: text alignment viewer
- index: index alignment
- idxstats: BAM index stats (r595 or later)
- fixmate: fix mate information
- glfview: print GLFv3 file
- flagstat: simple stats
- calmd: recalculate MD/NM tags and '=' bases
- merge: merge sorted alignments
- rmdup: remove PCR duplicates
- reheader: replace BAM header
How to Use samtools

bowtie -S e_coli reads/e_coli_10000snp.fq ec_snp.sam

samtools view -bS -o ec_snp.bam ec_snp.sam

samtools sort ec_snp.bam ec_snp.sorted

samtools pileup -cv -f genomes/NC_008253.fna\ ec_snp.sorted.bam

where pileup command prints records for 10 distinct SNPs, 1st at position 541 in the reference
Agenda

• High Performance Computing
• Bioinformatics/computational biology
• Bioinformatics’ tools
  • Sequence assembly
  • Sequence alignment
• Simulation tools
  • Molecular Dynamics
  • Molecular docking
MD Applications

NAMD
GROMACS
Molecular Dynamics (MD)

• One of the principal tools in the theoretical study of biological molecules
• Calculates time dependent behavior of a molecular system
• Provide detailed information on conformational changes of proteins and nucleic acids
• Routinely used to investigate the structure, dynamics and thermodynamics of biological molecules and their complexes.
• Force fields are approximate and pair-additive
• Long range interactions are cut off
• Boundary conditions are unnatural
NAMD

• Nanoscale Molecular Dynamics (NAMD)
  • Not (just) Another Molecular Dynamics Program
• A open source parallel molecular dynamics code designed for high-performance simulation of large bimolecular systems.
• Joint collaboration of the Theoretical and Computational Biophysics Group (Klaus Schulten) and the Parallel Programming Laboratory (PPL) at the University of Illinois at Urbana-Champaign, 1999
• Charm++ parallel objects, a machine independent parallel programming system, scales to hundred of thousand cores
• Force fields compatible with AMBER, CHARMM and X-PLOR
• Provide limited interface to Tcl (tool command language)
How does it work

• Protein: a chain of amino acids connected by chemical bonds
• Atomic connectivity (topology)
  • bonds, angles, dihedrals among atoms
• Force field parameter file: equations of force field potential energy among atoms
  • e.g. spring stiffness and equilibrium bond length
NAMD Input

- Config file: simulation parameters including:
  - Protein data bank file (.pdb): atomic coordinates
  - Protein structure file (.psf), which is created from
    - .pdb
    - Known topology file (atom types/charges/bonds..)
  - Force field parameter file: atomic potential equations to evaluate forces and energies
    - CHARMM, X-PLOR, AMBER, and GROMACS
  - Downloadable: [http://mackerell.umaryland.edu/charmm_ff.shtml](http://mackerell.umaryland.edu/charmm_ff.shtml)
VMD (Visual Molecular Dynamics)

- Assist NAMD simulation setup
- Display MD results and analyze MD trajectory
- Installed on qb/eric
Prepare files for simulations

• Solvate protein – cellular environment
• Generate structure (.psf) file from .pdb and topology file
• Modify config file (.conf/.namd) to run NAMD
• Example in folder
  biology/namd
• More tutorial
  http://www.ks.uiuc.edu/Training/Tutorials/
## Available NAMD Installations

<table>
<thead>
<tr>
<th>Machine</th>
<th>Version</th>
<th>Softenv Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>eric</td>
<td>2.6</td>
<td>+NAMD-2.6-intel-11.1-mvapich-1.1</td>
</tr>
<tr>
<td>eric</td>
<td>2.7b2</td>
<td>+NAMD-2.7b2-intel-11.1-mvapich-1.1</td>
</tr>
<tr>
<td>eric</td>
<td>2.7b4</td>
<td>+NAMD-2.7b4-intel-11.1-mvapich-1.1</td>
</tr>
<tr>
<td>eric</td>
<td>2.9</td>
<td>+NAMD-2.9-intel-11.1-openmpi-1.3.4</td>
</tr>
<tr>
<td>pandora</td>
<td>2.7</td>
<td>+namd-2.7</td>
</tr>
<tr>
<td>pandora</td>
<td>2.8</td>
<td>+namd-2.8</td>
</tr>
<tr>
<td>supermike2</td>
<td>2.9</td>
<td>+NAMD-2.9-Intel-13.0.0-openmpi-1.6.2</td>
</tr>
<tr>
<td>supermike2</td>
<td>2.9</td>
<td>+NAMD-2.9-Intel-13.0.0-openmpi-1.6.2-CUDA-4.2.9</td>
</tr>
</tbody>
</table>
# Available NAMD Installations

<table>
<thead>
<tr>
<th>Machine</th>
<th>Version</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>qb2</td>
<td>2.10b1</td>
<td>namd/2.10b1/CUDA-65-INTEL-140-MVAPICH2-</td>
</tr>
<tr>
<td>qb2</td>
<td>2.9</td>
<td>namd/2.9/INTEL-14.0.2-ibverbs</td>
</tr>
<tr>
<td>smic</td>
<td>2.10</td>
<td>namd/2.10/INTEL-14.0.2-ibverbs</td>
</tr>
<tr>
<td>smic</td>
<td>2.10</td>
<td>namd/2.10/INTEL-14.0.2-ibverbs-mic</td>
</tr>
</tbody>
</table>
How to run NAMD (qb)

• **Serial version**
  • `namd2 <config-file>`

• **Ibverbs parallel version**
  • `namd2 +p <procs> <config-file>`
  • `charmrun ++local ++p <procs> `which namd2` <config-file>`
  • `charmrun ++nodelist <nodefile> ++p<procs> \ 
    ++remote-shell ssh `which namd2` <config-file>`
  
  Note: nodefile: host qb122
  host qb123

• **MPI parallel version**
  • `mpirun –np/-ppn <procs> -hostfile $PBS_NODEFILE \ 
    `which namd2` <config-file>`
NAMD Output

- COOR: final coordinate file
- VEL: final velocity file
- DCD: trajectory file
- DAT: run information (energies...)
- XSC: system configuration output
Simulation Time

• Serial version
  • 1m34s

• Ivverbs parallel version
  • 2 procs: 51s
  • 20 procs: 9.8s

• MPI parallel version (GPU)
  • 10 procs: 3m32s
  • 20 procs: 21s
NAMD on GPU (qb cluster)

- MPI parallel version (GPU) (10 processes)

```
[wfeinste@qb497 1-2-sphere]$ nvidia-smi
Mon Apr 18 11:54:34 2016

NVIDIA-SMI 352.55  Driver Version: 352.55
------------------------------------------------------------------------------------------------------------------------
<table>
<thead>
<tr>
<th>GPU Name</th>
<th>Fan</th>
<th>Temp</th>
<th>Perf</th>
<th>Persistence-M</th>
<th>Bus-Id</th>
<th>Disp.A</th>
<th>Volatile Uncorr. ECC</th>
<th>GPU-Util</th>
<th>Compute M.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Tesla K20Xm</td>
<td>27C</td>
<td>P0</td>
<td>56W / 235W</td>
<td>0000:03:00.0</td>
<td>0ff</td>
<td>382MiB / 5759MiB</td>
<td>2%</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>N/A 30C</td>
<td>P0</td>
<td>57W / 235W</td>
<td>0000:03:00.0</td>
<td>0ff</td>
<td>381MiB / 5759MiB</td>
<td>3%</td>
<td>Default</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Processes:
<table>
<thead>
<tr>
<th>GPU</th>
<th>PID</th>
<th>Type</th>
<th>Process name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16151</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>0</td>
<td>16152</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>0</td>
<td>16153</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>0</td>
<td>16154</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>0</td>
<td>16155</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>1</td>
<td>16156</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>1</td>
<td>16157</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>1</td>
<td>16158</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>1</td>
<td>16159</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
<tr>
<td>1</td>
<td>16160</td>
<td>C</td>
<td>10b1/CUDA-65-INTEL-140-MVAPICH2-2.0/namd2</td>
</tr>
</tbody>
</table>
```
GROMACS

• Groningen Machine for Chemical Simulations
• Developed by the Berendsen Group, Department of Biophysical Chemistry, University of Groningen, Netherlands, 2001
Gromacs

Input:
• .gro: coordinate, velocity of the system
• .top: topology, i.e. bonds/pairs/angles of atoms
• .mdp: simulation parameters

Preprocessed:
• .tpr: simulation input file generated by grompp

Output:
• .gro: final configuration and velocities
• .trr/.xtc: trajectory
• .edr: energies
• .log: run information
How to Use Gromacs

• Solvate the molecule
  
gmx solvate -cp protein_box.gro -cs spc216.gro \ 
  -o protein_sol.gro -p topol.top

• Generate the topology and index files
  
pdb2gmx -v -f protein_sol.gro -o \ 
  protein_sol_final.gro -p topol_final.top -n \ 
  protein_sol.ndx

• Generate MD run input file (.tpr)
  
grompp -v -f 1UBQ.mdp -c protein_sol_final.gro \ 
  -p topol_final.top -o 1UBQ_run

• Run MD simulations
  
mdrun -v -s 1UBQ_run.tpr -x -deffnm 1UBQ_output

• Post-simulation analysis
Gromacs result (VMD)
Agenda

• High Performance Computing
• Bioinformatics/computational biology
• Bioinformatics’ tools
  • Sequence assembly
  • Sequence alignment
• Simulation tools
  • Molecular Dynamics
• Molecular docking
Autodock Vina

• Open-source program for molecular docking
• Significant improve the average accuracy of binding mode prediction than AutoDock4
• Easy to use
• Multiple CPU/cores (OpenMP)
What is molecular docking?

- Docked molecules bind to receptor (protein) in a specific confirmation
- Identify the confirmation with which a ligand binds target protein with lowest energy at binding site
- Virtual screening in drug discovery
Prepare Files

- Protein (receptor) structure (.pdb/.pdbqt)
- Known molecule/ligand/drug that bind to the target protein (.sdf/.pdbqt)
- Identify docking center
- Define docking box size


- Set up config file (config.txt)
How to Run Vina

• Installed on SuperMike
• cd biology/vina
• soft add +autodock_vina-1.1.2
• Autodock vina usage:
  vina --config config.txt
Autodock Vina Output

- Vina gives binding affinity estimates
- Result >10: binding is very tight
- Result [6-7]: random binding
More Information

• Software stack
  http://www.hpc.lsu.edu/docs/guides/index.php

• HPC training materials
  http://www.hpc.lsu.edu/training/archive/tutorials.php

• Computational biology user guide
  http://www.hpc.lsu.edu/docs/compbio/index.php

• User guide
  http://www.hpc.lsu.edu/docs/guides.php