History of Bioinformatics Core at SKCC

- Began in 2002 with grant from Ron and Lucille Neeley
- Linux Cluster in 2002:
  - 34 AMD Athlon CPUs (32-bit)
  - 61 GB total RAM memory
  - 1200 GB of RAID disk storage
- Sustained by endowment from Ron and Lucille Neeley
  - Incremental maintenance costs less than $7,000 a year
  - Bad news: Computers get old and break
    - 10 CPUs have died, 8 CPUs are near death
    - Replacement parts are not available
    - Haven't bought new CPUs since 2006
  - Good news: Every year …
    - Memory becomes cheaper, faster, and bigger
    - Disks become cheaper, faster, and bigger
    - CPUs become cheaper and denser (single core, dual core, and now quad core)
- Linux Cluster today (2008):
  - 24 legacy AMD Athlon CPUs (32-bit)
  - 22 Dell Xeon CPU cores (64-bit)
  - 122 GB total RAM memory
  - 3400 GB of RAID disk storage

Linux Cluster

Q. What is Linux good for?

A. Most popular bioinformatic programs run on Linux:

- **Sequence Alignment (BLAST, BLAT, MUMmer, FASTA)**
- **Whole Genome Assemblers:**
  - **Celera, Phrap, PCAP, Arachne**
- **Genome Browsers and Databases:**
  - **NCBI Map Viewer**
  - **Ensembl Genome Browser**
  - **UCSC Genome Browser**
- **Bioinformatics tool suites (EMBOSS, etc.)**
- **Primer and Oligo Design (Primer3, etc.)**

Q. What is a cluster good for?

A. Our cluster can run jobs 46 times faster than a single PC

- **Days instead of years**
- **Minutes instead of days**

SKCC Bioinformatics Core

What services are provided by the Bioinformatics Core?

- **Internal web site (weber) - web-based bioinformatics tools**
- **External web site (bioinformatics.skcc.org)**
- **Custom Programming (examples):**
  - **Primer Design**
  - **Oligo (Probe) Design**
  - **Sequence Search/Alignment (BLAST, BLAT, MUMmer)**
  - **Visualization (Bacterial genomes, etc.)**
  - **“Can you write me a Perl script that does _____?”**
    - **Over 1000 Perl and Shell scripts written so far**
- **Home Grown Programs (examples):**
  - **Comparative Assembler**
  - **WebArray (Xiao-Qin and Yipeng)**
  - **Multiplex PCR Optimization**
  - **Parallel BLAT sequence alignment**
  - **Job Grabber**

Weber - Internal Web Services

URL: http://weber

Discovering Cures For Cancer In Our Lifetime

SKCC Bioinformatics Internal Web Page

Local Tools

- Simple BLAT Query
- Batch BLAST to NCBI
- Ebi's Peptide Blaster
- FirstEF (First Exon Finder)
- MAFFT Multiple Alignments
- MEME and MAST 3.0.3
- Primer3 batch server
- YPPPPS YAPPPPS
- Primer Express
- RepeatMasker
- TMPred
- Protein Translators
- Glimmer gene prediction

Off-Site Tools

- WU-BLAST Homepage
- EMBL-EBI's WU-Blast2
- PCR Suite based on Primer3
- Pasteur Institute (many useful tools)
- ch.EMBnet.org
- NCBI's Glimmer server

Gene Lookup

- Gene Lynx - gene-lookup tool
- MatchMiner - batch gene-lookup tool
- SOURCE - batch gene-lookup tool
- HomGL - Homology Gene List

Genome Browsers

- Ensembl
- UCSC Genome Browser
- NCBI Map Viewer

Misc. Documents

- MySQL Manual
- Linux Docs
- Perl Tutorial

Core Services

- Genomics Core: PDF PowerPoint
- Bioinformatics Core: PDF PowerPoint

Other Links

- External Bioinformatics Web Site
- System Administration
- Bioinformatics FTP files

Local Database Access:

- phpMyAdmin (read-only access)
- phpMyAdmin-2.5.5
- phpMyAdmin-2.7.0-beta1
- phpMyAdmin-2.11.1

- phpPgAdmin

If you would like a specific public-domain tool installed here let me know.

Weber - Internal Web Services

URL: http://weber

- **Simple BLAT query**
- **Batch BLAST query**
- **MAFFT multiple sequence alignment**
- **Primer and probe design with Primer3**
- **RepeatMasker**
- **TMPred - Prediction of Transmembrane Regions**
- **Protein translators and back-translators**
- **Perl Tutorial**
- **MySQL database access**
- ***Add your own Linux web service – you find it, I'll install it.***

Weber - NCBI *Batch* BLAST

NCBI Batch Blast - Send batch request to NCBI

README

Program | blastn | Databases | nr | + | none | + | none | or special |

Enter sequence below in FASTA format

Clear sequence | Search

The query sequence is filtered for low complexity regions by default.

Filter ☑ Low complexity ☐ Mask for lookup table only

Expect | 10 | Matrix | BLOSUM62 | ☐ Perform ungapped alignment

Blastx: Query Genetic Codes | Standard (1)

Frame shift penalty | No OOF

Other advanced options:

☐ NCBI-gi Alignment view | Pairwise | Descriptions | 100 | Alignments | 50

How patient are you? (Deliver Method) | Send me an email when it's done | Email:

Clear sequence | Search

Weber - Primer3 *Batch* Server

Primer3 Test Pre-Release

pick primers from a DNA sequence

disclaimer | cautions | bugs? suggestions? | source code

Paste source sequence below (5'->3', string of ACGTNacgtn -- other letters treated as N -- numbers and blanks ignored). FASTA format ok.
Please N-out undesirable sequence (vector, ALUs, LINEs, etc.) or use a Mispriming Library (repeat library):

NONE

☑ Pick left primer or use left primer below. | ☐ Pick hybridization probe (internal oligo) or use oligo below. | ☑ Pick right primer or use right primer below (5'->3' on opposite strand).

Pick Primers | Reset Form

Sequence Id: A string to identify your output.

Targets: E.g. 50,2 requires primers to surround the 2 bases at positions 50 and 51. Or mark the source sequence with [and]: e.g. ...ATCT[CCCC]TCAT.. means that primers must flank the central CCCC.

Excluded Regions: E.g. 401,7 68,3 forbids selection of primers in the 7 bases starting at 401 and the 3 bases at 68. Or mark the source sequence with < and >: e.g. ...ATCT<CCCC>TCAT.. forbids primers in the central CCCC.

Product Size Min: 100 Opt: 200 Max: 1000

Number To Return: 5 Max 3' Stability: 9.0

Max Mispriming: 12.00 Pair Max Mispriming: 24.00

Pick Primers | Reset Form

General Primer Picking Conditions

Primer Size	Min: 18	Opt: 20	Max: 27				
Primer Tm	Min: 57.0	Opt: 60.0	Max: 63.0	Max Tm Difference:	100.0		
Product Tm	Min:	Opt:	Max:				
Primer GC%	Min: 20.0	Opt:	Max: 80.0				
Max Self Complementarity:	8.00	Max 3' Self Complementarity:	3.00				
Max #N's:	0	Max Poly-X:	5				
Inside Target Penalty:		Outside Target Penalty:	0	Set Inside Target Penalty to allow primers inside a target.			
First Base Index:	1	CG Clamp:	0				

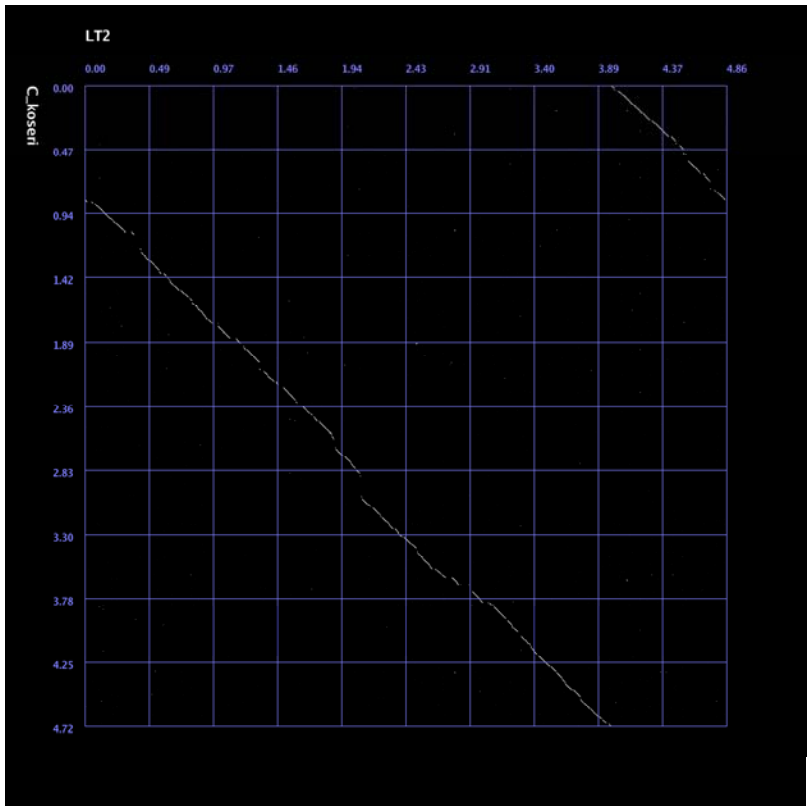
External Web Services

URL: http://bioinformatics.skcc.org

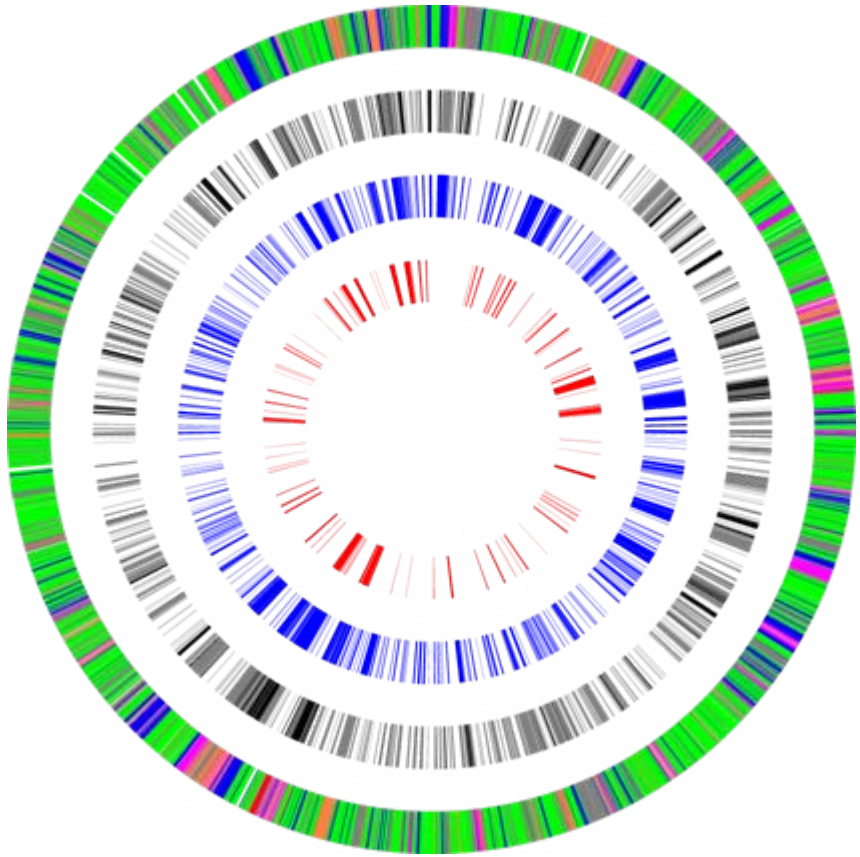
- **WebArray – Xiao-Qin and Yipeng**
- **Permanent file repository**
  - **Supplemental data for your publications**
- **Temporary file repository**
  - **For sharing large files with collaborators**
  - **Password protected**

Visualization Examples

Dot Plot – LT2 vs. C. Koseri

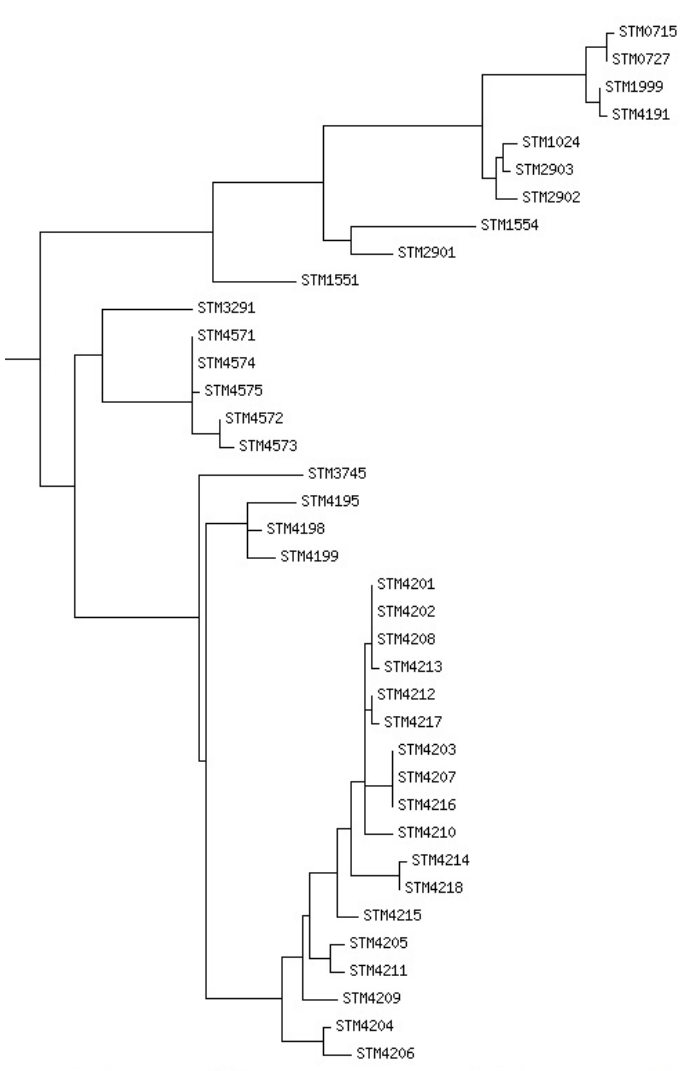


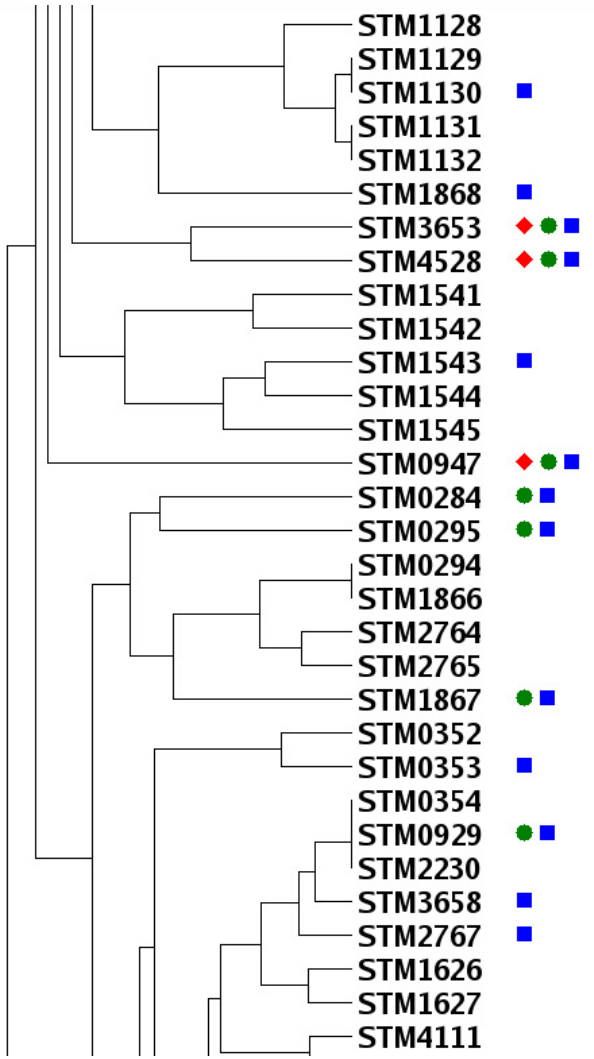
Bacteria Genes



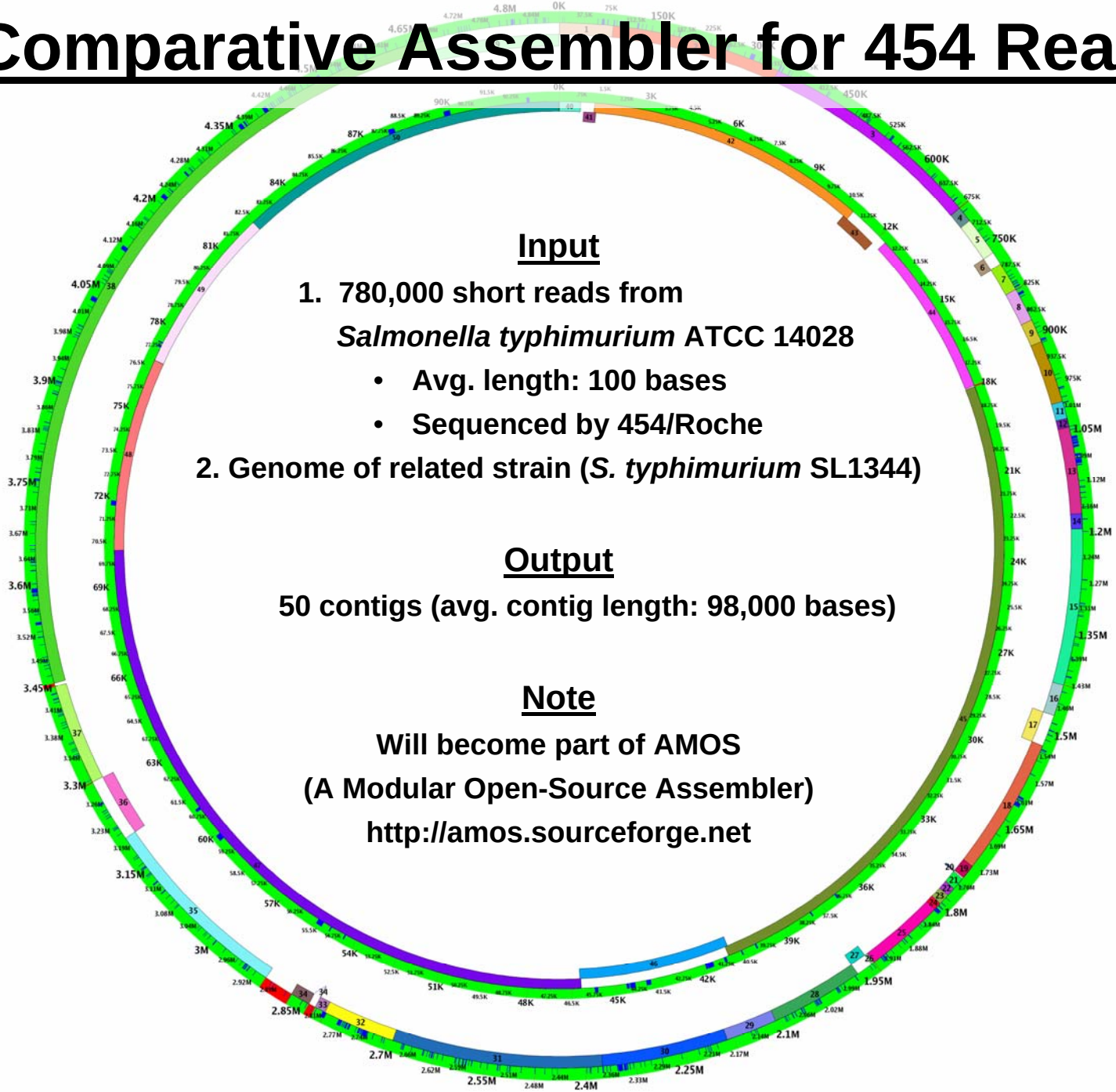
Visualization Examples

Phylogenetic Trees





Comparative Assembler for 454 Reads



Input

1. 780,000 short reads from
   Salmonella typhimurium ATCC 14028
   - Avg. length: 100 bases
   - Sequenced by 454/Roche
2. Genome of related strain (*S. typhimurium* SL1344)

Output

50 contigs (avg. contig length: 98,000 bases)

Note

Will become part of AMOS

(A Modular Open-Source Assembler)

http://amos.sourceforge.net

Job Grabber - Fault Tolerant Job Control System

Waiting:

id	pri	to	user	name	submit_time	status
2672	1	.*	proteomic	sequest.10148 2672	2008-01-20 00:55:15	waiting
2673	1	.*	proteomic	sequest.10148 2673	2008-01-20 00:55:15	waiting
2674	1	.*	proteomic	sequest.10148 2674	2008-01-20 00:55:15	waiting
2675	1	.*	proteomic	sequest.10148 2675	2008-01-20 00:55:15	waiting
2676	1	.*	proteomic	sequest.10148 2676	2008-01-20 00:55:15	waiting
2677	1	.*	proteomic	sequest.10148 2677	2008-01-20 00:55:15	waiting
2678	1	.*	proteomic	sequest.10148 2678	2008-01-20 00:55:15	waiting
2679	1	.*	proteomic	sequest.10148 2679	2008-01-20 00:55:15	waiting
2680	1	.*	proteomic	sequest.10148 2680	2008-01-20 00:55:16	waiting
2681	1	.*	proteomic	sequest.10148 2681	2008-01-20 00:55:16	waiting
2682	1	.*	proteomic	sequest.10148 2682	2008-01-20 00:55:16	waiting
2683	1	.*	proteomic	sequest.10148 2683	2008-01-20 00:55:16	waiting
2684	1	.*	proteomic	sequest.10148 2684	2008-01-20 00:55:16	waiting
2685	1	.*	proteomic	sequest.10148 2685	2008-01-20 00:55:16	waiting
2686	1	.*	proteomic	sequest.10148 2686	2008-01-20 00:55:16	waiting
2687	1	.*	proteomic	sequest.10148 2687	2008-01-20 00:55:17	waiting
2688	1	.*	proteomic	sequest.10148 2688	2008-01-20 00:55:17	waiting
2689	1	.*	proteomic	sequest.10148 2689	2008-01-20 00:55:17	waiting

Running:

id	host	pgid	user	name	start_time	seconds	status
2637	node2	15973	proteomic	sequest.10148 2637	2008-01-23 18:26:50	3864	running
2638	node2	16047	proteomic	sequest.10148 2638	2008-01-23 18:26:51	3863	running
2639	node10	6946	proteomic	sequest.10148 2639	2008-01-23 18:26:52	3862	running
2640	node7	17973	proteomic	sequest.10148 2640	2008-01-23 18:26:55	3859	running
2641	node1	12387	proteomic	sequest.10148 2641	2008-01-23 18:26:55	3859	running
2642	node9	7778	proteomic	sequest.10148 2642	2008-01-23 18:26:56	3858	running
2643	node6	8262	proteomic	sequest.10148 2643	2008-01-23 18:26:56	3858	running
2644	node12	28894	proteomic	sequest.10148 2644	2008-01-23 18:26:57	3857	running
2645	dell1	13103	proteomic	sequest.10148 2645	2008-01-23 18:26:58	3856	running
2646	node1	12465	proteomic	sequest.10148 2646	2008-01-23 18:27:01	3853	running
2647	dell2	25703	proteomic	sequest.10148 2647	2008-01-23 18:27:02	3852	running
2648	node5	21265	proteomic	sequest.10148 2648	2008-01-23 18:27:02	3852	running
2649	node6	8346	proteomic	sequest.10148 2649	2008-01-23 18:27:02	3852	running
2650	node5	21339	proteomic	sequest.10148 2650	2008-01-23 18:27:05	3849	running
2651	dell4	26257	proteomic	sequest.10148 2651	2008-01-23 18:27:05	3849	running
2652	node11	20542	proteomic	sequest.10148 2652	2008-01-23 18:27:06	3848	running
2653	node6	8426	proteomic	sequest.10148 2653	2008-01-23 18:27:08	3846	running
2654	dell4	26359	proteomic	sequest.10148 2654	2008-01-23 18:27:10	3844	running
2655	dell5	13973	proteomic	sequest.10148 2655	2008-01-23 18:27:14	3840	running
2656	dell6	3405	proteomic	sequest.10148 2656	2008-01-23 18:27:18	3836	running
2657	dell4	26486	proteomic	sequest.10148 2657	2008-01-23 18:27:25	3829	running
2658	dell6	3517	proteomic	sequest.10148 2658	2008-01-23 18:27:26	3828	running
2659	dell5	14095	proteomic	sequest.10148 2659	2008-01-23 18:27:29	3825	running
2660	dell5	14197	proteomic	sequest.10148 2660	2008-01-23 18:27:29	3825	running
2661	dell6	3629	proteomic	sequest.10148 2661	2008-01-23 18:27:31	3823	running
2662	dell6	3630	proteomic	sequest.10148 2662	2008-01-23 18:27:31	3823	running
2663	dell5	14309	proteomic	sequest.10148 2663	2008-01-23 18:27:34	3820	running
2664	dell5	14416	proteomic	sequest.10148 2664	2008-01-23 18:27:37	3817	running

Done:

id	host	user	name	end_time	seconds	status
2598	dell3	proteomic	sequest.10148 2598	2008-01-23 19:05:58	2375	done
2608	dell7	proteomic	sequest.10148 2608	2008-01-23 19:12:08	2739	done

Features

- **Queue up thousands of jobs**
- **Each job automatically assigned to next free CPU and run in parallel**
- **Failed jobs can be easily restarted**
- **System crashes are no problem**
- **Jobs are stored in MySQL database, not in RAM**
- **Kill and restart jobs at any time**

Usage Examples

- **Search NCBI “NR” database:**
  - **3 million entries**
  - **19 billion bases**
  - **search time 5 minutes**
- **Design millions of oligos**
- **SEQUEST, DTASelect**
- ***Any other big job that can be split into smaller pieces***

Case Study - SEQUEST

Proteomics has 197,000,000 “DTA” files containing data from mass spec. experiments.

A fast PC running SEQUEST can process:

- **6,000 DTA files in an hour**
- **197,000,000 DTA files in 3.75 years**
- **Can’t easily restart after system crash**

SKCC’s Linux Cluster running SEQUEST through Job Grabber can process:

- **238,000 DTA files per hour**
- **197 million DTA files in 34 days**
- **System crash? No problem. Continue where you left off.**

34 days

3.75 years