Applications from the Genomics and Proteomics Domains

May 3 , 2011

Genome Analysis Toolkit (GATK)

- Structured Programming Framework
  - Uses the functional programming paradigm of map reduce
  - Traversals and walkers in place of map and reduce
  - Traversals provide the division and preparation of data
  - Walkers provide analysis modules

SAM(Sequence Alignment/Map) Format

@SQ SN:>AssembledLN:16608 → Header

Query/Read name **bitwiseFlag** **Reference** **Position** **Mapping quality** **CIGAR string** **MateRef**

HWI-ST632_0045:1:1101:1212:2108#0 16 >Assembled 5326 37 101M *
Mate Position Inferred insert size
0 0
Sequence of read

CCTAACGCTATTCCCTACATCCAACAACACAAAAATAACATGACAACTCGAAAATACAAAACCCACACTCCTTATCCCCCCACTCATCAT
CACCTCCACTC febededfbfbceggedggagegeggcgbcggggddgdggdeeegaeadeegggeegggecddd_d]
dZddeegZgggggcgdggdggdgaffdffcfef XT:A:U NM:i:4 X0:i:1 X1:i:0 XM:i:4 XO:i:0
XG:i:0 MD:Z:2G5G1G88C1

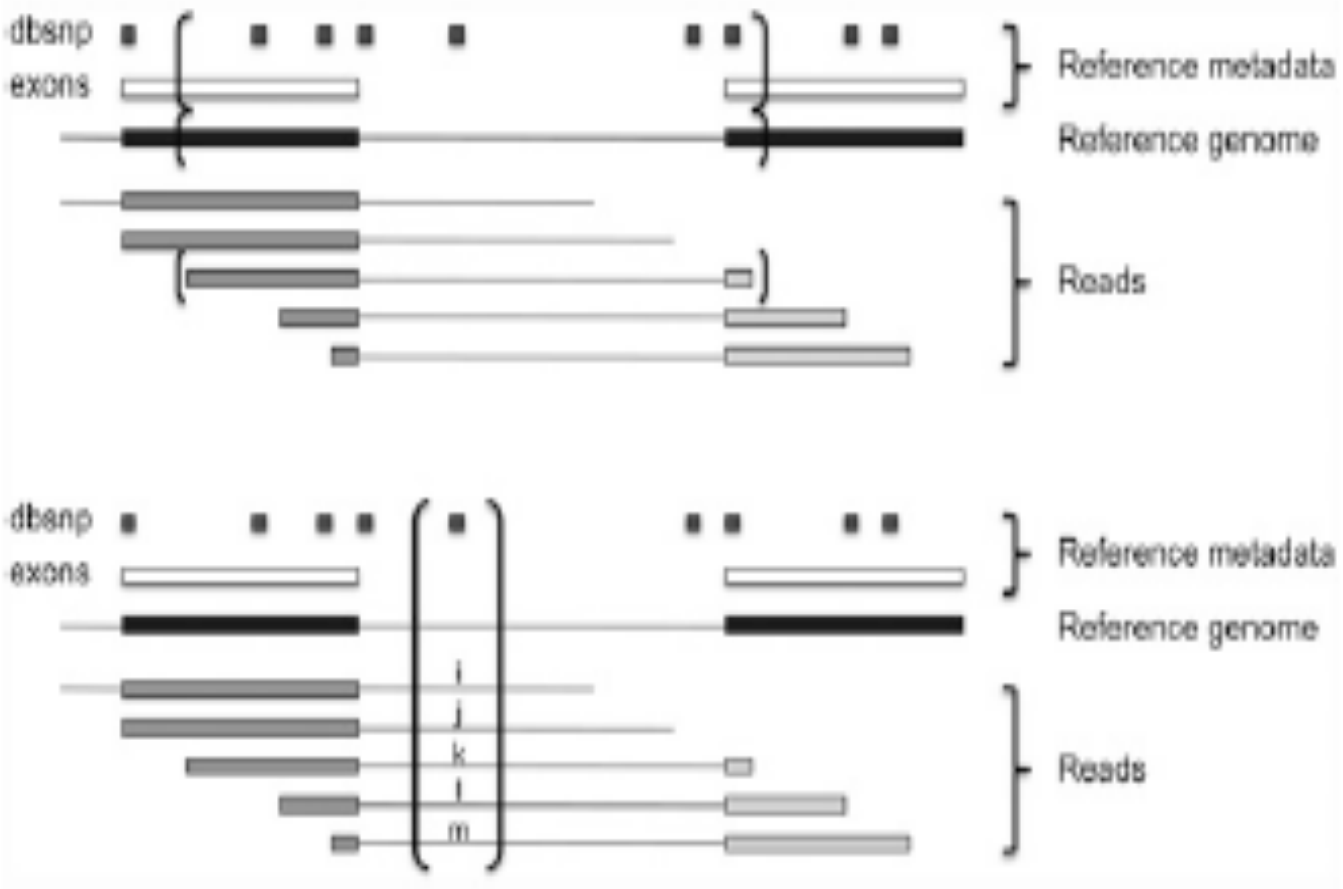
HWI-ST632_0045:1:1101:1174:2139#0 4 * 0 0 * *
0 0

AAAGAGTCAGCGATTAATGAGCATAAGTGGTAAAATGGCTGAGTAAGTATTAGGCTGTAAACCTAAAGACGGGGGTTTAGTCCCCCTT
TTACCAGCCCCGA fgfgggegfgggggggfgggggfggggeggggggfgggegfcfddedbdecebgggddcgedgeebeeae
\ebeeeffcefcacefffff`^dccdeefbO
HWI-ST632_0045:1:1101:1246:2155#0 16 >Assembled 10295 37 101M *
0 0

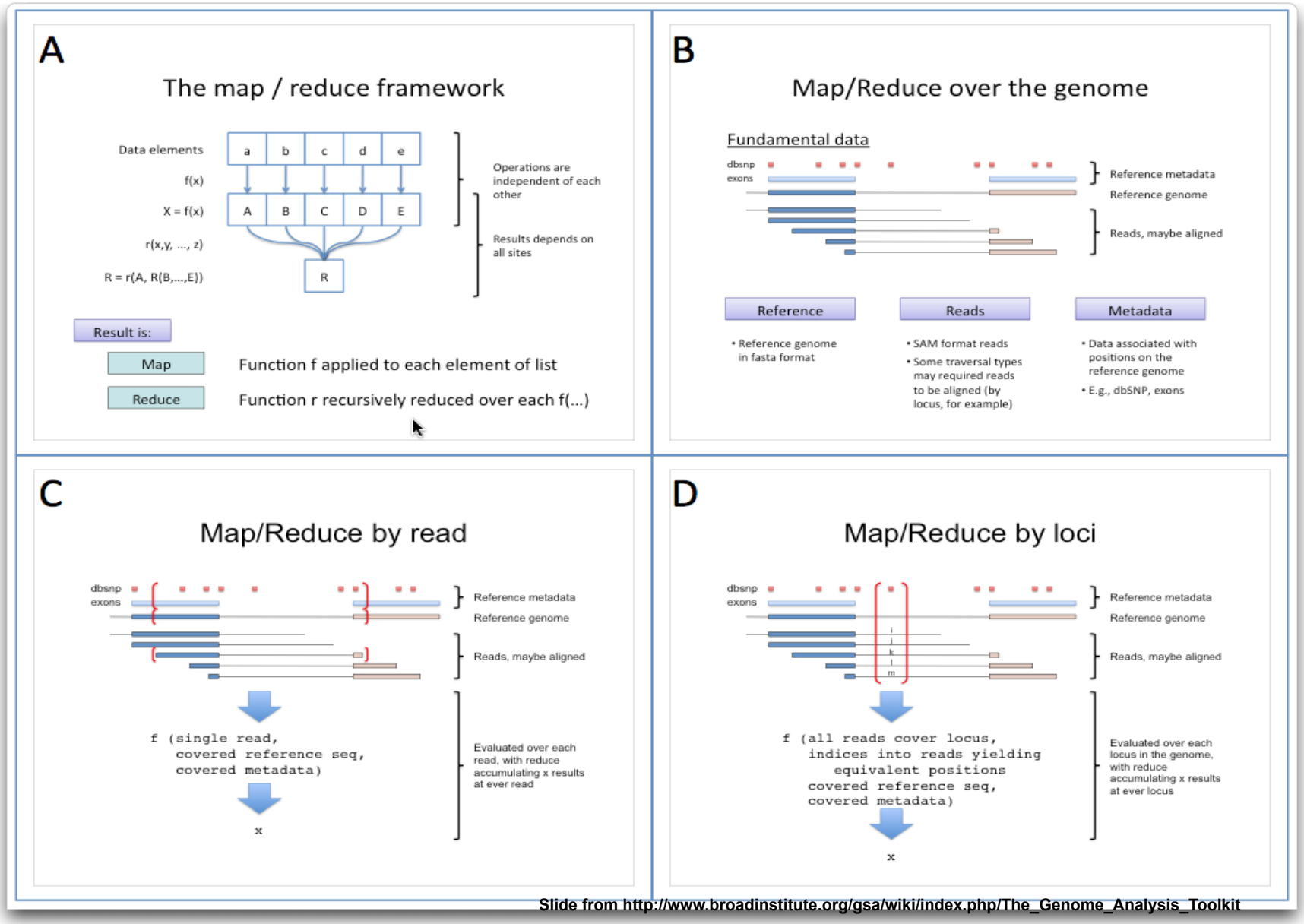
ATTCAAACAACAAACCTCCCAATAATAATCAAATCGACCATTGCCTTTATTATTATCCTAATTCTCAGTTTAACCTATGAATGAACCCAAAA
AGGGCTAGA eg]
gggdfdefeeUde^fggggbggeegacggecgedgggdeggggggggeggggggggggedgdgggggggggggggeggegggggggggggggggggg
XT:A:U NM:i:1 X0:i:1 X1:i:0 XM:i:1 XO:i:0 XG:i:0 MD:Z:
96T4
HWI-ST632_0045:1:1101:1235:2209#0 4 * 0 0 * *
0 0

GACTTACTCCCATCCTTAACCCCTTAACAGAATACATAGCTTACCCATTCCTCATATTATCCTTATGGGGCATAATCATAACAGGCCTAAC
ATGCCTCCGA

Pileup Format



Types of traversals



Slide from http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit

Types of Traversals

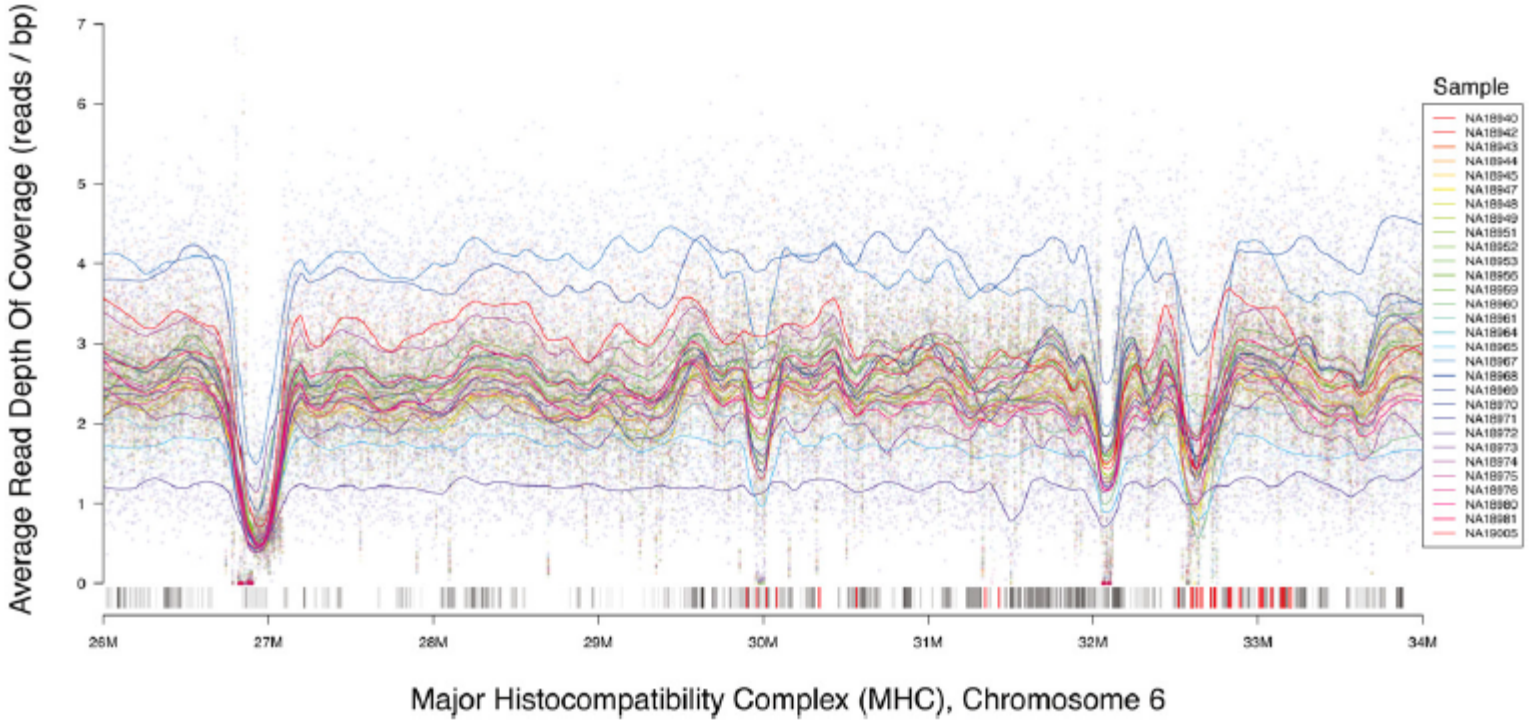
Table 1. Traversal types available in the Genome Analysis Toolkit

TraverseLoci	Each single base locus in the genome, with its associated read, reference ordered data, and reference base are presented to the analysis walker.
TraverseReads	Each read is presented to the analysis walker, once and only once, with its associated reference bases.
TraverseDuplicates	The walker is supplied with a list of duplicate reads and unique reads at each reference locus.
TraverseLocusWindows	Walkers are supplied the reads, reference ordered data, and reference bases for a whole interval of the genome, as opposed to a single base as in TraverseLoci.

Data Processing Functionality

- Sharding
  - Data is split into multi kilobase pieces
  - All associated metadata information for the region is available
  - Includes all the associated reads from the SAM file
- Interval Processing
  - Allows users to bracket the region presented to walkers
- Merging input files
  - To obtain biologically meaningful results
- Parallelization
  - Allows for interval processing or shared memory utilization

Example 1: Depth of Coverage Walker



Walker receives a list of reads covering the reference and emits the size of the pileup

GATK single sample genotype likelihoods

Bayesian model

Likelihood for the genotype | Prior for the genotype | Likelihood of the data given the genotype | Independent base model

$$L(G \mid D) = P(G)P(D \mid G) = \prod_{b \in \{good\_bases\}} P(b \mid G)$$

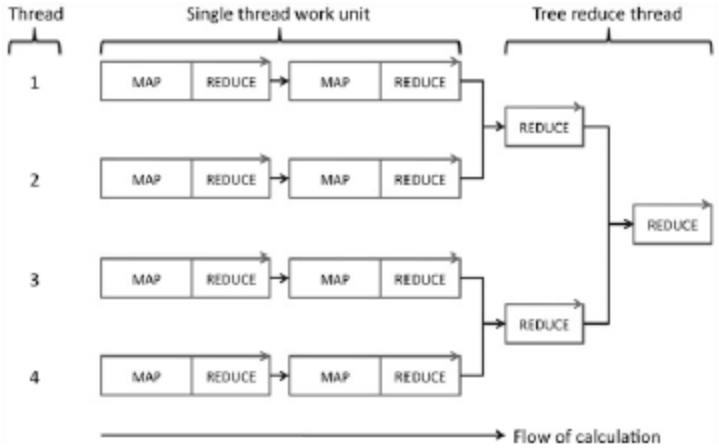
- Priors applied during multi-sample calculation; P(G) = 1
- Likelihood of data computed using pileup of bases and associated quality scores at given locus
- Only "good bases" are included: those satisfying minimum base quality, mapping read quality, pair mapping quality, NQS
- P(b | G) uses a platform-specific confusion matrix
- L(G|D) computed for all 10 genotypes

Slide from http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit

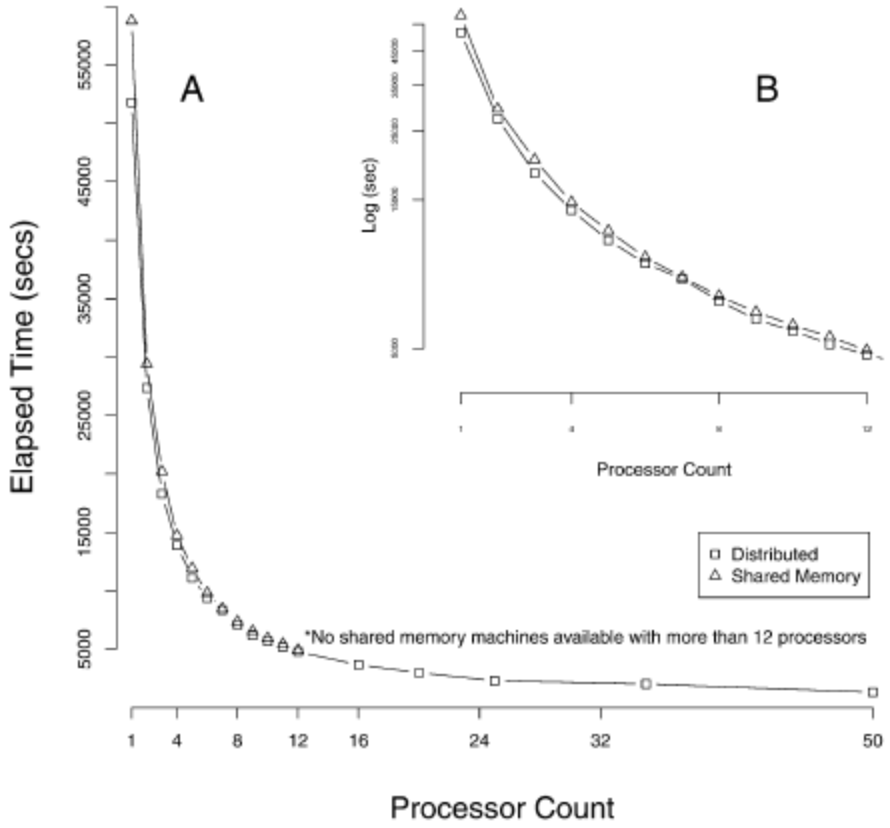
Example 2: Simple Genotype Walker

```
public SimpleCall map(RefMetaDataTracker tracker, ReferenceContext ref, AlignmentContext context) {
// we don't deal with the N ref base case
    if (ref.getBase() == 'N' || ref.getBase() == 'n') return null;
    ReadBackedPileup pileup = context.getBasePileup();
    double likelihoods[] = DiploidGenotypePriors.getReferencePolarizedPriors(ref.getBase(),
                                    DiploidGenotypePriors.HUMAN_HETEROZYGOSITY,
                                    0.01);
    // get the bases and qualities from the pileup
    byte bases[] = pileup.getBases();
    byte quals[] = pileup.getQuals();

    // for each genotype, determine it's likelihood value
    for (GENOTYPE genotype : GENOTYPE.values())
      for (int index = 0; index < bases.length; index++) {
        if (quals[index] > 0) {
          // our epsilon is the de-Phred scored base quality
          double epsilon = Math.pow(10, quals[index] / -10.0);

          byte pileupBase = bases[index];
          double p = 0;
          for (char r : genotype.toString().toCharArray())
            p += r == pileupBase ? 1 - epsilon : epsilon / 3;
          likelihoods[genotype.ordinal()] += Math.log10(p / genotype.toString().length());
        }
      }

    Integer sortedList[] = MathUtils.sortPermutation(likelihoods);

    // create call using the best genotype (GENOTYPE.values()[sortedList[9]].toString())
    // and calculate the LOD score from best - next best (9 and 8 in the sorted list, since the best likelihoods are closest to zero)
    return new SimpleCall(context.getLocation(),
                GENOTYPE.values()[sortedList[9]].toString(),
                likelihoods[sortedList[9]] - likelihoods[sortedList[8]],
                ref.getBase());
  }

  public Integer reduceInit() {
    return 0;
  }

  public Integer reduce(SimpleCall value, Integer sum) {
    if (value != null && value.LOD > LODScore) outputStream.println(value.toString());
    return sum + 1;
  }


  public Integer treeReduce(Integer lhs, Integer rhs) {
    return lhs + rhs;
  }


  public void onTraversalDone(Integer result) {
    out.println("Simple Genotyper genotyped " + result + "Loci.");
  }
}
```

Shared Memory parallel tree-reduction



Parallelization of genotyping



Spectra for PRIDE Experiment # 9169

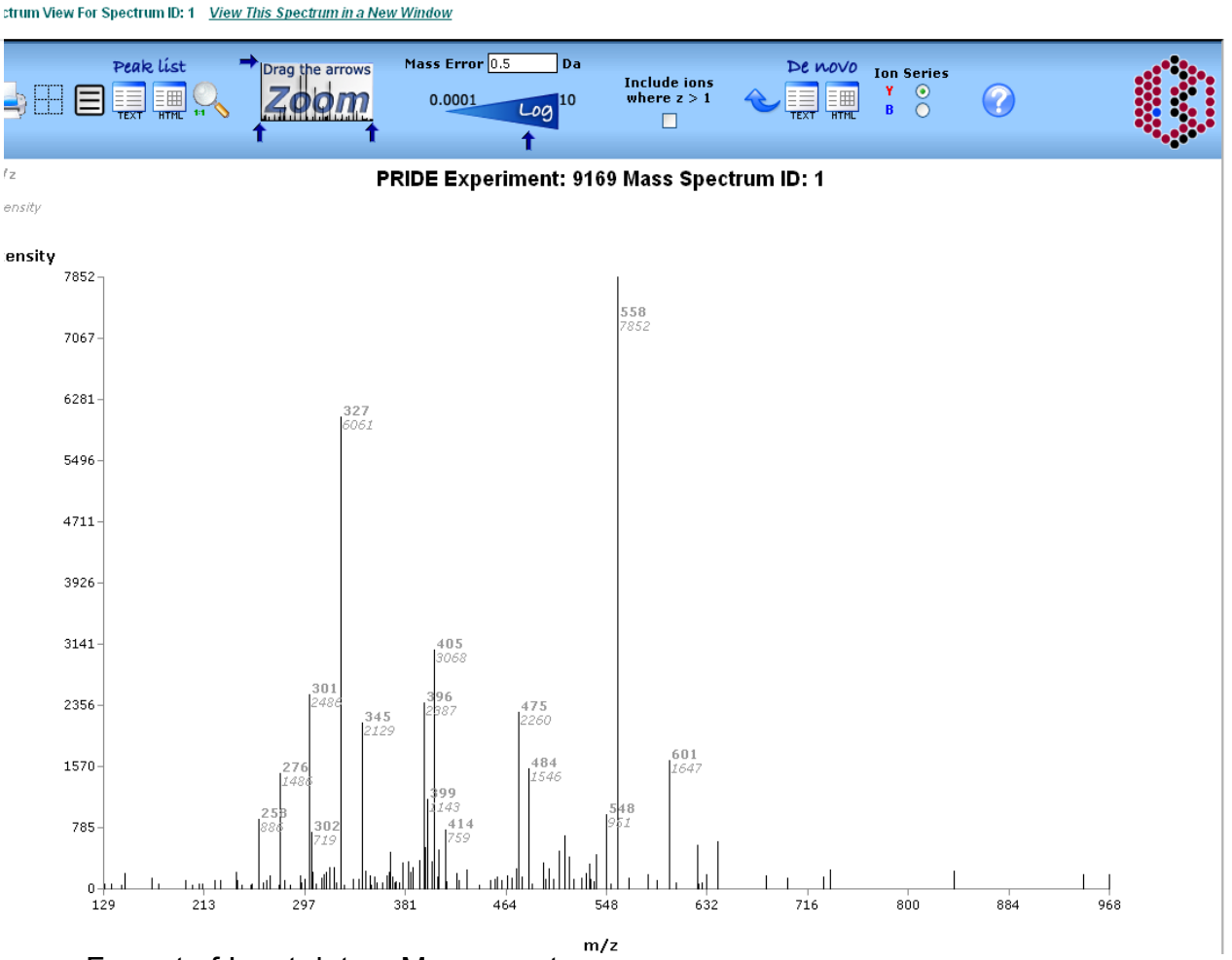
1 2 ... 291

Spectrum ID	m/z Range Start	m/z Range Stop	MS Level
1	129.070490	945.716190	2
2	181.051540	1173.660000	2
3	200.244920	923.556380	2
4	130.194660	789.450320	2
5	175.093970	649.400300	2
6	140.147810	833.783880	2
7	140.147810	833.783880	2
8	129.143390	746.406690	2
9	138.108530	526.147000	2
10	129.091220	677.286460	2
11	137.980710	736.243980	2
12	166.070880	918.476560	2
13	198.183210	841.414690	2
14	167.172210	809.284480	2
15	181.197890	899.346760	2
16	175.147780	926.396450	2
17	110.264740	586.291180	2
18	129.151710	750.384110	2
19	129.311450	1034.517900	2
20	129.089790	680.343380	2
21	129.173880	600.363830	2
22	129.173880	600.363830	2
23	120.140030	710.452570	2
24	120.140030	710.452570	2
25	129.070760	798.134360	2
26	120.270860	696.218030	2
27	148.303440	867.528080	2
28	136.174110	947.301950	2
29	130.156990	824.283200	2
30	232.191400	1120.384300	2

1 2 ... 291

1 to 30 of 8714 Mass Spectra

Data from http://www.ebi.ac.uk/pride/showExperiment.do

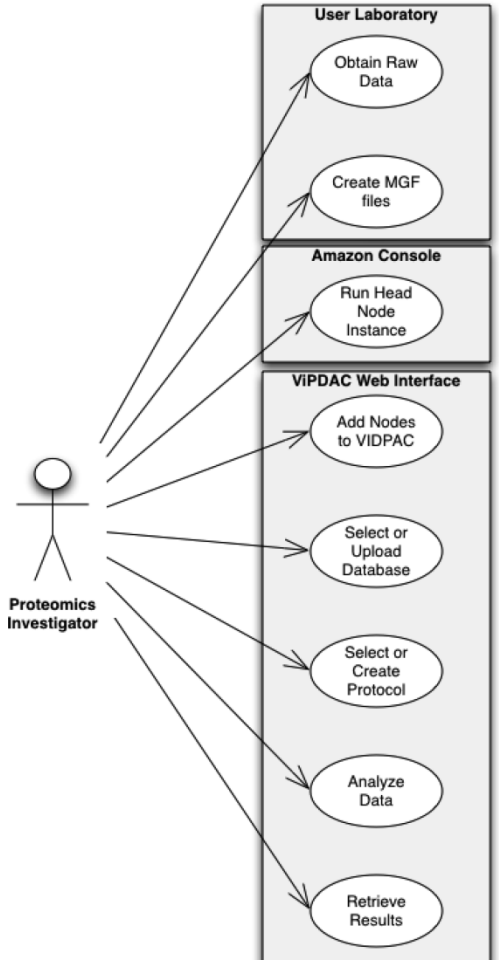


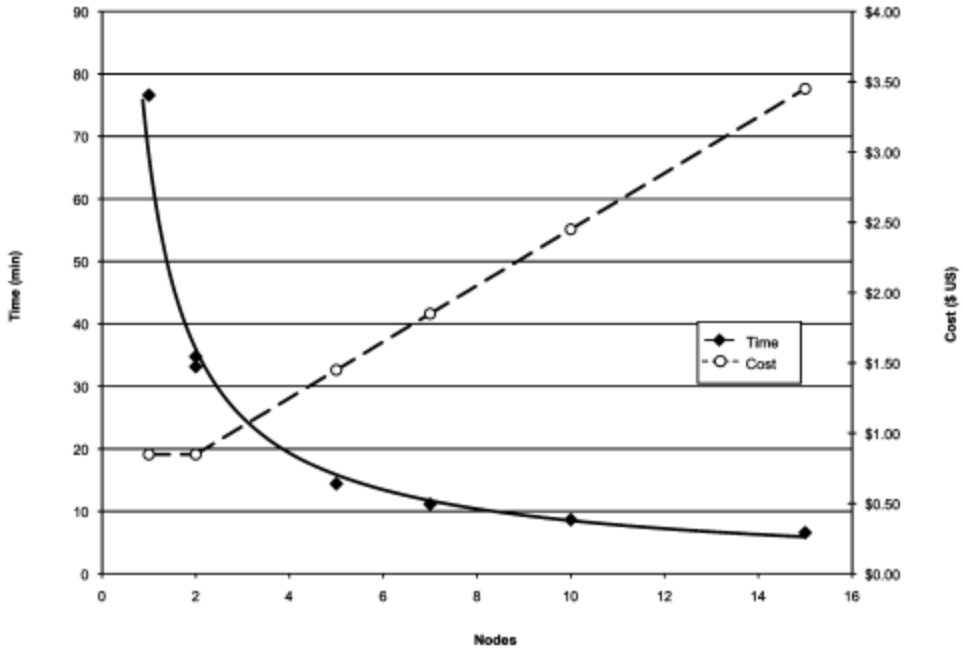
Format of Input data – Mass spectra

Goal: Is to identify the composition of the peptides in sample based on the m/z values.

Method: OMSSA takes experimental ms/ms spectra, filters noise peaks, extracts *m*/*z* values, and then compares these *m*/*z* values to calculated *m*/*z* values derived from peptides produced by an in-silico digestion of a protein sequence library. The theoretical peptides must have a mass within a user specified tolerance of the precursor mass. The resulting search hits are then statistically scored.

Steps for Launching the Virtual protein Data Analysis Cluster





Time and Costs for running a an LC-MS/MS analysis of a human protein reference sample. 22,385 spectra were searched against 39514 proteins

Case Study

- 1000 spectra portion of 81000 spectra used
  - Time on single node cluster 1.6
  - On desktop computer 2.1 h
- Entire dataset of 81000 spectra
  - 6.8 h of 20 node cluster
- For entire experiment (with 20 samples)
  - Estimate 5.7 days on cluster
  - 140 days on desktop machine