



Computational grid for comparative genomics to identify conserved non-coding (CNS) regions

Jagath C. Rajapakse, Ph.D.

*Associate Professor, School of Computer Engineering
Deputy Director, Bioinformatics Research Centre
Nanyang Technological University, Singapore*

*Visiting Professor, Biological Engineering Division
Massachusetts Institute of Technology, USA*

Outline

- Comparative genomic approach for characterizing non-coding sequences
- Computational pipeline for identifying conserved non-coding sequences
- Implementation on a grid system
- Comparison of human and mouse chromosome 12
- Conclusions

Comparative genomics

- Comparative analyses of genomes to study evolutionary characteristics
- Segments of functional significance are often conserved over the evolution
- Comparative genomics can help understand function of genes, SNPS, non-coding regions, repeats, etc.
- Identify segments of non-coding regions, that are of functional significance such as gene regulation



Non-coding sequences

- Non-coding sequences do not code for proteins and stable RNAs
- They account for 95% of genome. Why present over the evolution?
- Conserved non-coding segments (CNS) are implicated in the regulation of genes
- Objectives:
 - identify CNS by taking a comparative genomic approach;
 - create databases at different levels of significance
 - characterize their possible functions

Limitations of existing conserved non-coding databases

- Out-of-date data-sources: Penn State, Berkeley and UCSC databases use Human Dec 2001 and MGSCv3 mouse sequences
- Not all UTRs are considered as source sequences to align.
- Not consider repetitive sequences
- Incremental database

Motivation

Pipeline for identifying CNS

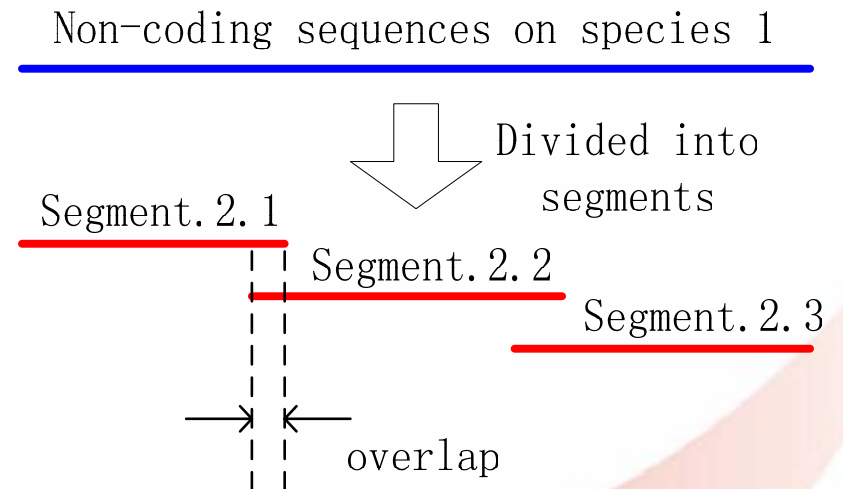
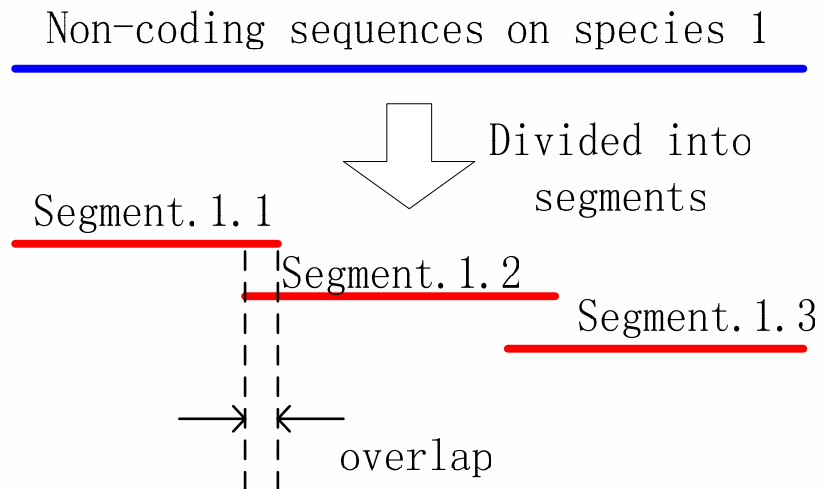
- Genomes and annotations
 - Ensembl database (version 36, 2006)
 - Uses MySQL relational databases
 - A set of APIs serve as a middle-layer for accessing its databases.
- Extracting non-coding sequences
 - repetitive sequences; introns; untranslated regions; pseudogenes; conserved intergenic sequences.
- Preprocessing for BLASTZ
 - form segments of non-coding sequences of length of 1Mbp with 100bp overlap between adjacent segments

Pipeline for identifying CNS

- Sequence comparison with BLASTZ
 - BLASTZ aligns an appreciable fraction of the neutrally evolving DNA in the human and mouse genomes
- Identifying CNS
 - Selection criteria: species, identity, and length
 - gap-free alignments with at least 70% identity and 100 bp in length
- CNS database
 - CNS are stored in relational databases (MySQL).
 - The quality of the segments are assessed and fed back to improve identification

Parallelization of Pipeline

- BLASTZ accounts for most of computation time.
- Parallelization is achieved by dividing totally independent subtasks.



Overlaps aim to guarantee that no alignments will be lost by the segmentation

BLAST vs. BlastZ

- BLAST is a heuristic method to find the local alignments between two sequences.
 - For a given word (segment pair) and score matrix, a list of all words that can produce scores larger than a threshold is created.
 - Search individual long sequences for occurrences of words in the list (hits).
 - Extending a hit to find a locally maximal segment pair containing that hit.
- BLASTZ is designed to align an appreciable fraction of the neutrally evolving DNA in the human and mouse genomes.
 - Finding short near-exact matches.
 - Extending each short match without allowing gaps.
 - Extending each gap-free match that exceeds a certain threshold by a dynamic programming procedure that permits gaps.

Grid-BLAST vs. Grid-BlastZ

- GridBlastZ use dynamic load balance concept

Advantage: keep all clusters busy and make the application's runtime as short as possible.

Disadvantage: Not all clusters permit a user occupy resources as large as possible.

- GridBlast dynamically download sequences to local machines; GridBlastz statically distribute sequences.

GridBlast it is not suitable for large size sequence comparison, because downloading large size sequences for each comparison is low efficient.

GridBlastz is suitable for large size tasks.

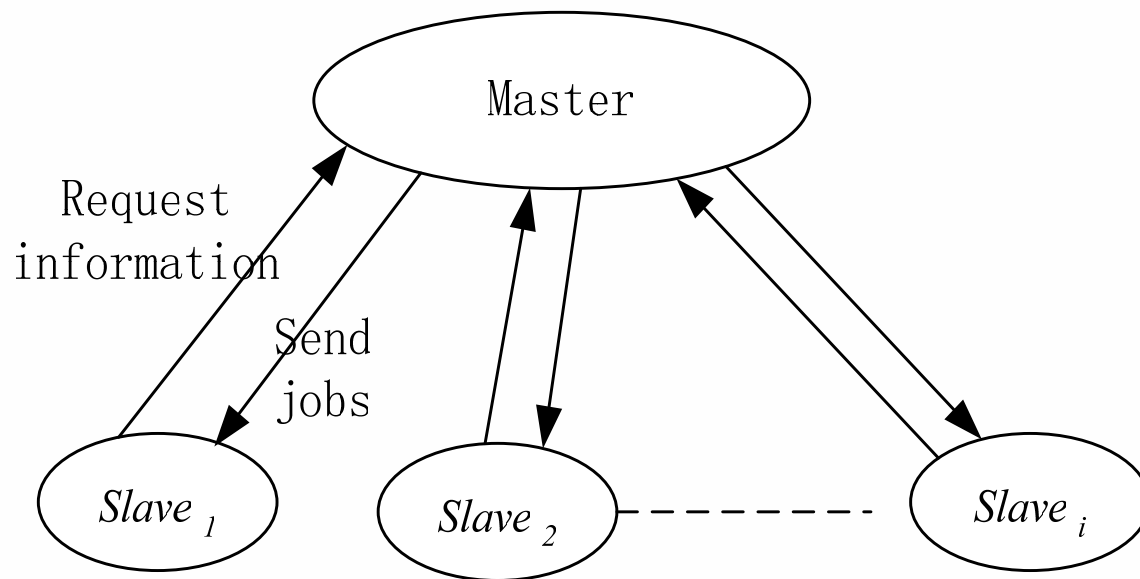
Why introduce grid computing for comparative genomics?

- Inherent parallelism, storage of genomic data in different machines, and the immense computation required
- e.g.: to compare 2.8 Gb of human sequence versus 2.5 Gb of mouse sequence, 481 days for a 833-Mhz Pentium III CPU.
- Effective R&D platform
- Grid computing achieves computing power by combing existing computational resources instead of building anew one. Therefore, high performance/cost ratio.
- Efficiently scale to meet higher computing and storage requirements
- Enable collaboration and promote operational flexibility

Mapping applications onto grid systems

- Heterogeneous characteristics of grid systems
 - Resources have different computational power and shared by all users.
 - They usually are connected by networks with widely varying performance characteristics
- To run the pipeline efficiently on grid systems, should:
 - The amount of work allocated to a computing resource should depend on the computational power that resource allocates to the application at that time. This assures that no processor becomes the bottleneck.

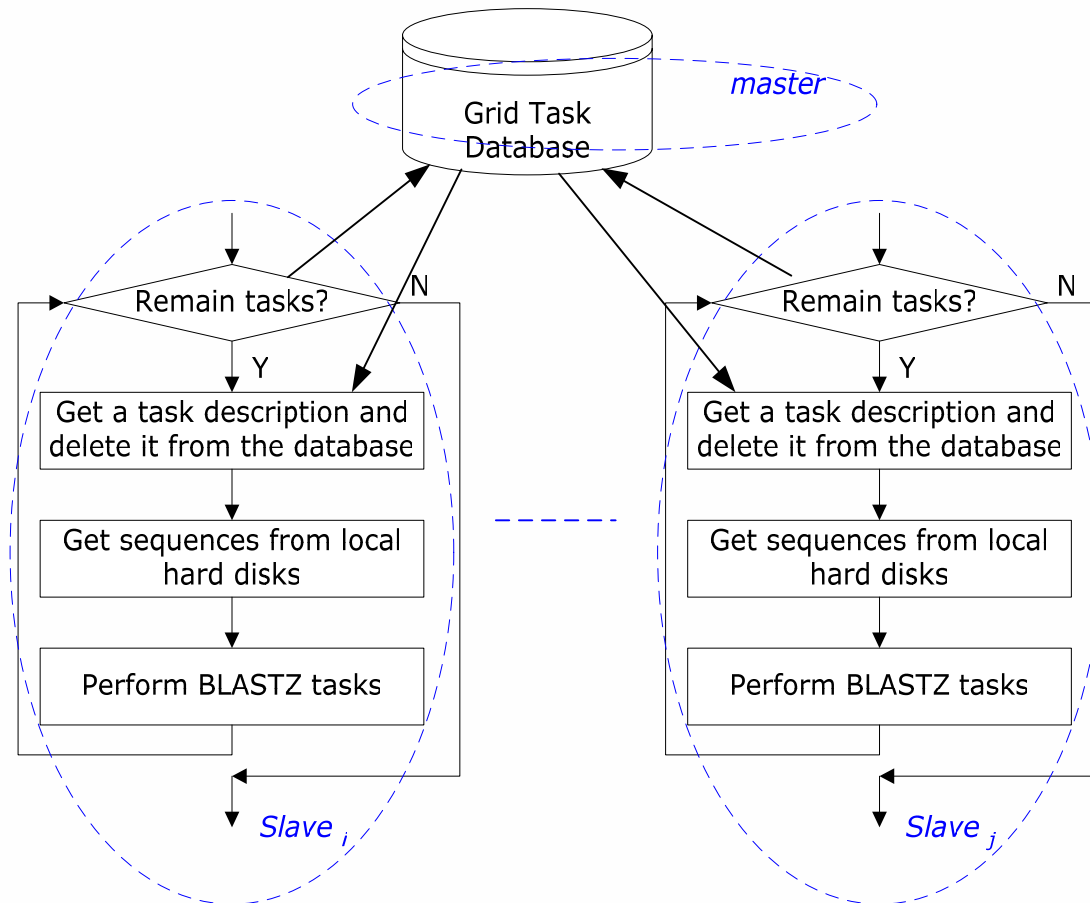
Master-slave paradigm: dynamic load balancing



Static vs. dynamic load balancing

Once the slave node completes a job, it sends a request to the master process. The master responds by sending back a new task

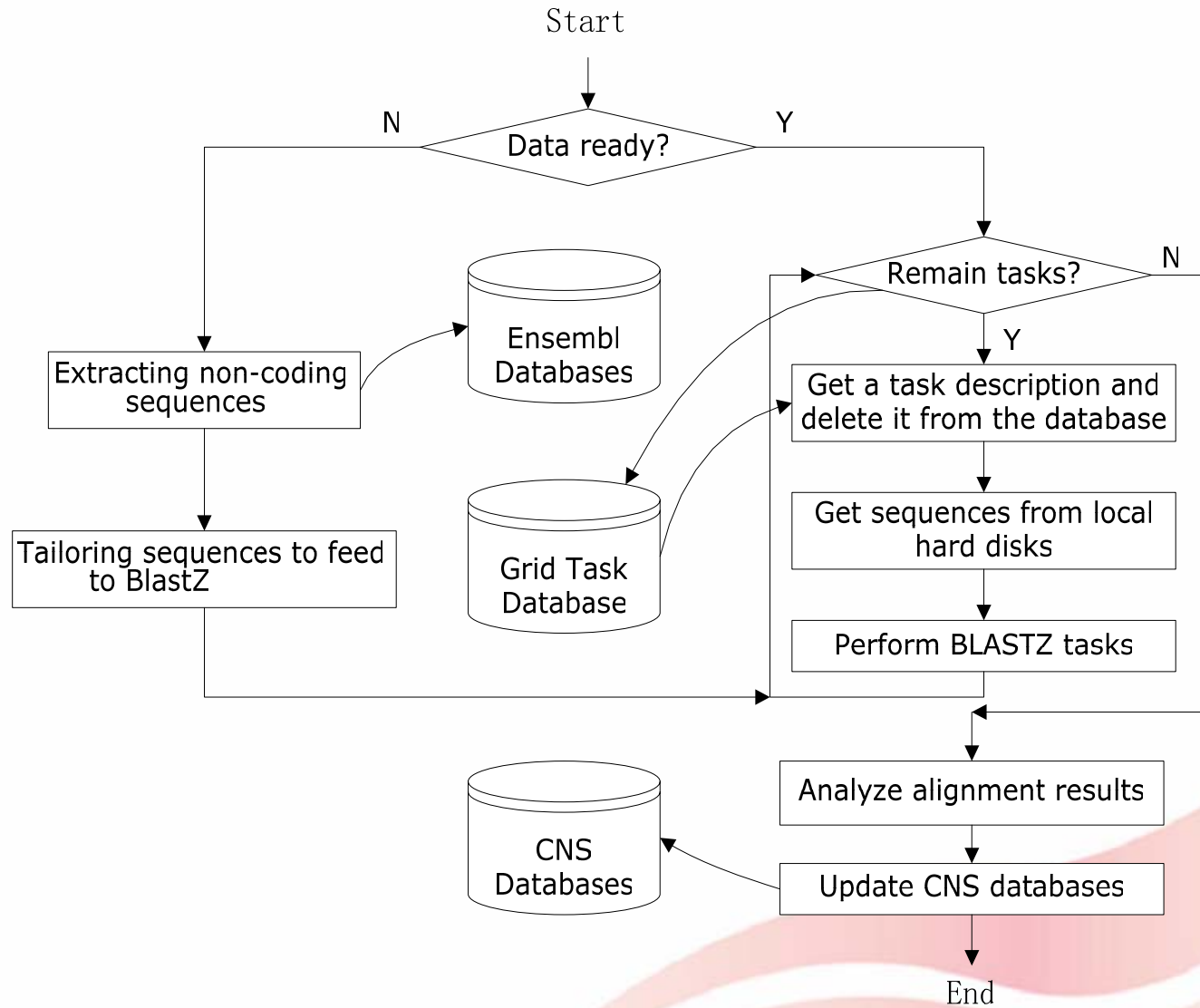
Grid Implementation



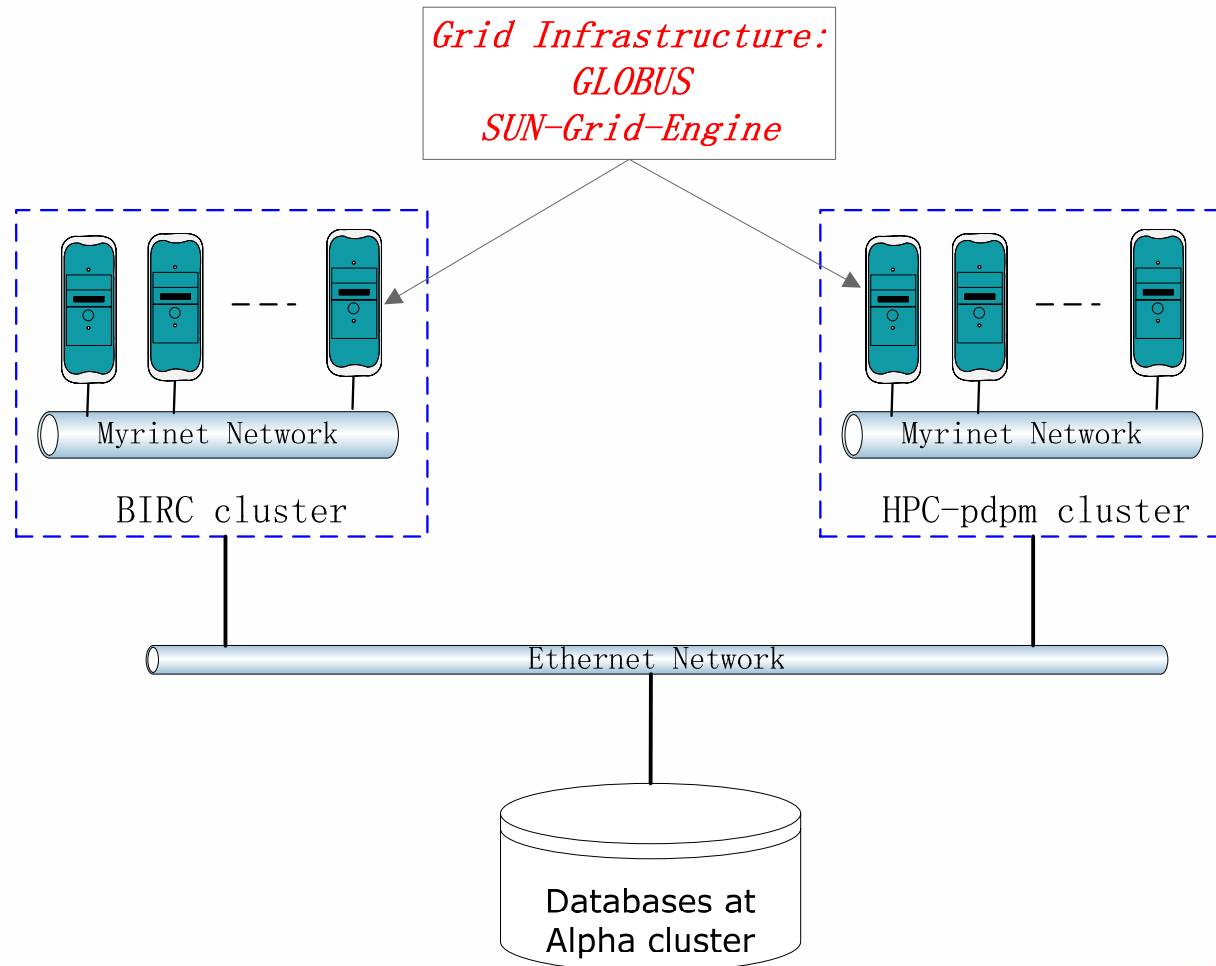
Example task: hum_chro_21_seg_1.mou_chro_16_seg_3

- Comparison between segment 1 of human chromosome 21 and segment 3 of the mouse chromosome 16

Grid-enabled pipeline



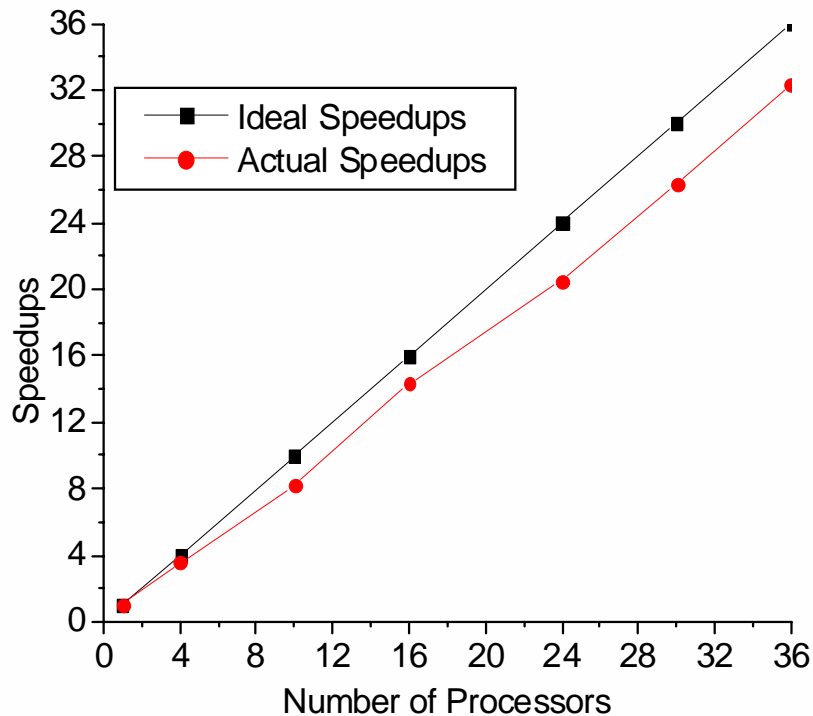
Test environment



Two Linux clusters: 16 Itanium 733MHz CPUs (BIRC) and 20 Intel 2.60GHz PCs (PDCC);

Unix64 cluster: 44 node Alpha cluster (BIRC)

Speedup



Speedups for aligning non-coding sequences on human chromosome 21 and mouse chromosome 16.

$$Speedup = \frac{RT_1 + RT_2}{2RT}$$

RT_1 and RT_2 are runtimes per processor of cluster1 and cluster2, respectively;

RT the runtime for each experiment using different number of processors.

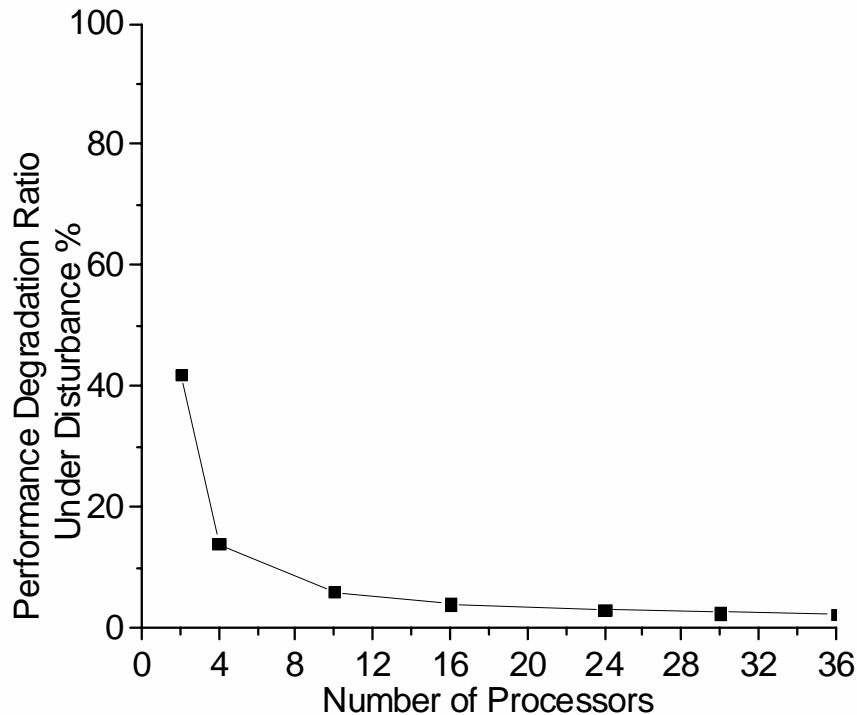
Overhead includes:

launching processes in clusters

accessing Gridtask database to get a task

accessing hard disks to get sequences

PDRD (*performance degradation ratio under disturbance*)



PDRDs for aligning non-coding sequences on human chromosome 21 and mouse chromosome 16.

Disturbance: effect by another applications running on the node to an application

$$PDRD = \left(\frac{T' - T}{T} \right) \times 100\%$$

where T' denotes the execution time under disturbance and T denotes the execution time without any disturbance

Uniqueness of our CNS databases

- Untranslated regions (UTRs) and repetitive regions are considered as source sequences to align for identifying the conserved regions;
- Contain more information, such as with which regions on which genomes a conserved non-coding region is associated;
- will cover the cross-species comparisons among all available vertebrate genomes such as human, mouse, chimp, chicken, zebrafish, tetraodon, and fugu.

Example database

REGIONS	TYPE	LENGTHS	ASSOCIATED REGIONS
9865681–9866001	interExons	320	mou_chro_16_interExons_6606058–6606170 mou_chro_16_interExons_11766093–11766193 mou_chro_16_interExons_60668073–60668181
13341775–13341882	pseudoGene	107	mou_chro_16_pseudoGene_29623039–29623146
14379144–14379261	pseudoGene	117	mou_chro_16_interExons_10498570–10498687
15313659–15313764	interExons	105	mou_chro_16_threeUTR_95583097–95595538
9758757–9759568	interExons	811	mou_chro_16_interExons_8772815–8772916 mou_chro_16_interExons_13057804–13058031 mou_chro_16_interExons_46949517–46949777 mou_chro_16_interExons_63522651–63522852

Example conserved non-coding database on human chromosome 21.

Future work

--- Characterizing CNS

- **Genome-wide statistical and computational analysis. It involves :**
 - finding how statistically significant are these potential regulatory elements present within and across genomes;
 - assessing the frequency of all elements in the database across genomes;
 - finding the similarity among elements;
 - creating new potential elements by merging the elements that are similar.
- **These findings can be verified using previously studied elements and regions, such as those reported in databases like TRANSFAC and BIND, to aid in the assignment of biological functions to conserved non-coding sequences.**

Thank you.

