



# Protein Structure Prediction Using Simple Genetic Algorithms

**V. Sundararajan**  
**Scientific and Engineering Computing Group**  
**C-DAC, Pune**  
**vsundar@cdac.in**

## Open Problem

There is no reliable procedure which begins with homologous model of a protein and then relaxes the structure using Molecular Dynamics to yield a conformation close to native. This is an important problem where database analysis cannot help.

## Why Structure Prediction ?

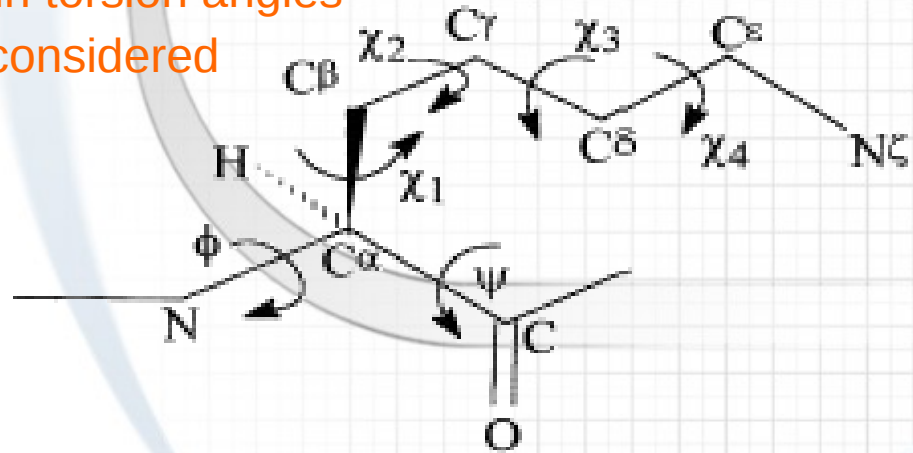
- Protein Structure prediction is one of the most challenging areas of research for the structural biologist.
- For 100 residues  $4^{100}$  ( or  $10^{60}$  ) possible conformations 10 years to search the whole space assuming 1 nanosecond Per energy calculation  
(Levinthal: *T. Creighton (editor), Protein Folding, W. H. Freeman (1992)*)

## Why Structure Prediction ?

- Out of few lakhs protein sequences only about ten thousands have known structures (X-ray, NMR)
- Some proteins cannot be crystallized easily
- About 10% of protein sequences exhibit unrelated and unidentified folds
- It's important to evolve the structures without the help of databases
- Useful in modelling the differences between structures that databases may not facilitate

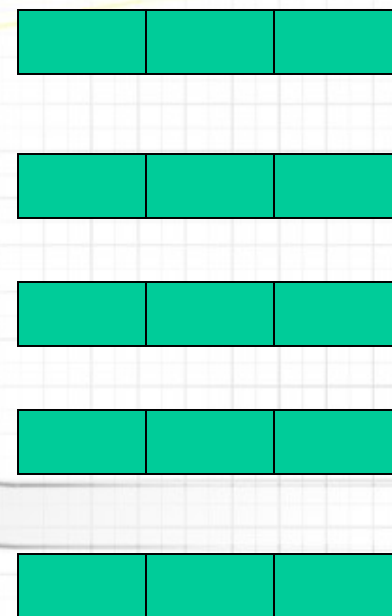
# Model of Protein Structure

- Internal coordinate system restricted to torsion angle variation
- Energy = non-bond + torsion angle interactions (AMBER force field)
- Assumptions:
  - Fixed bond lengths, bond angles
  - Also fix omega and side-chain torsion angles
  - Only phi and psi angles are considered

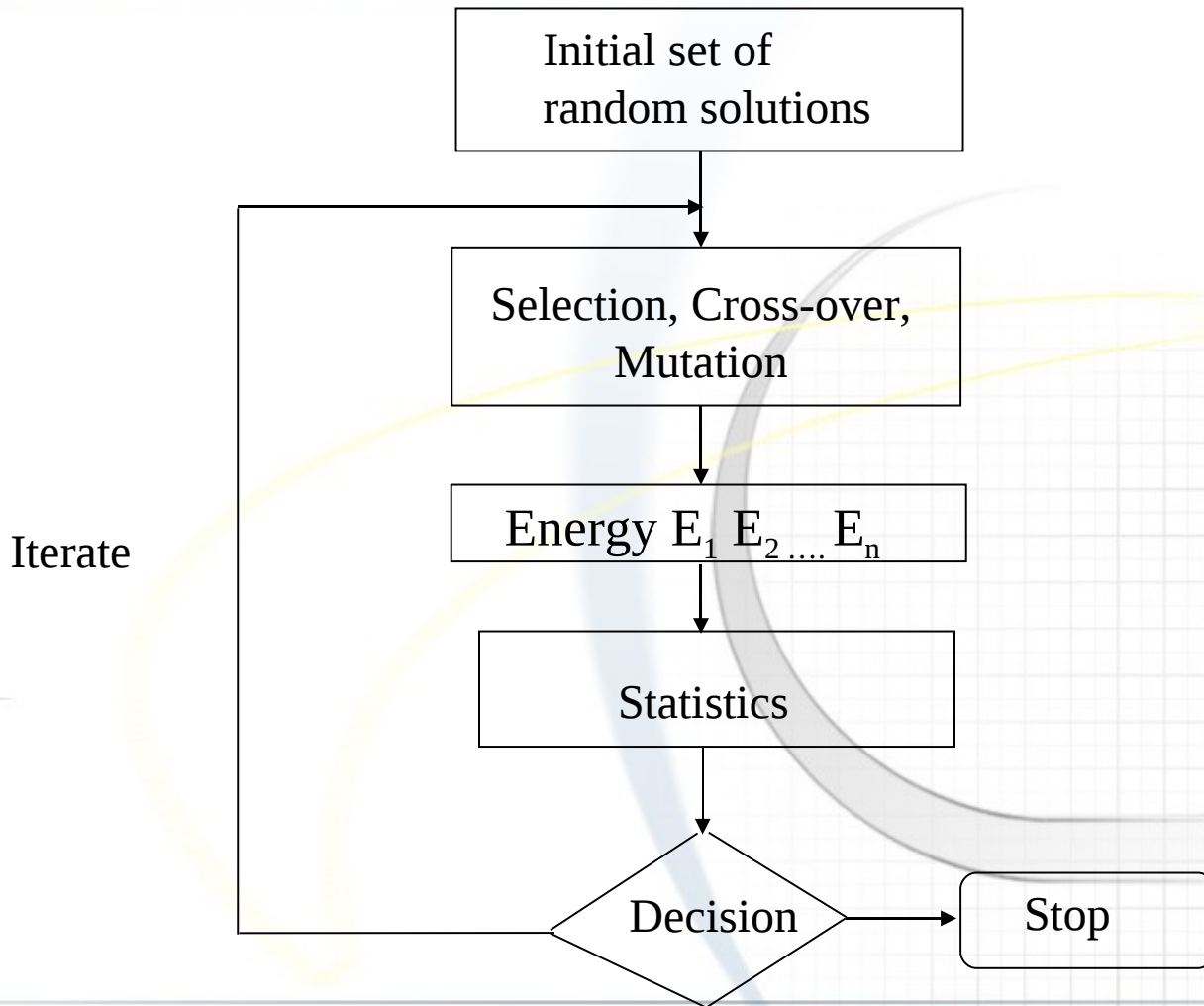


# Implementation

- Torsional angles are represented by binary strings
- Choose many such randomly; each one is a possible structure
- Go through the GA procedure
- Pick up the ones that are having minimum most energy as well as few around that



# Genetic Algorithms (GA)



## Advantages of GA

- Flexibility, robustness with global search characteristics, much less likely to get stuck on local optima
- Easy to implement
- Search from a population of points/solutions
- No derivatives, therefore can solve non-linear, discontinuous in parallel configuration
- Possible to device variety of parallel and distributed algorithms (suited both for HPC and Grid)
- Works on the representations rather than on the variables themselves

## Complexity involved in GA for PSP

- The complexity of GA code increases with the increase in sequence size as it affects chromosome length and population size
- **Length of chromosome (lchrom) =  $9 \times (2 \times \text{seq.length} - 2)$**
- **Population size (popsize) =  $2 \times \text{lchrom}$**
- **Number of function evaluations = popsize  $\times$  number of iterations**  
**where number of iterations (no of iters) =  $5 \times \text{popsize}$**

**Eg: if sequence size is 200 Aminoacids,**

**lchrom =  $9 \times (2 \times 200 - 2) = 3582$**

**popsize =  $2 \times \text{lchrom} = 7164$ ,**

**no.of iter = 35820**

**Number of Function evaluations =  $7164 \times 35820 =$**   
**2,56,61,4480**

Start

**Step1:** Read Population size,  
Number of Runs, Number of Select  
residues and Skip residues

Create a directory with appropriate Input files.  
Input files created according to run number,  
number of runs, Select residues, Skip residues and  
random seed value

Random seed generated for each run and  
input files are modified accordingly

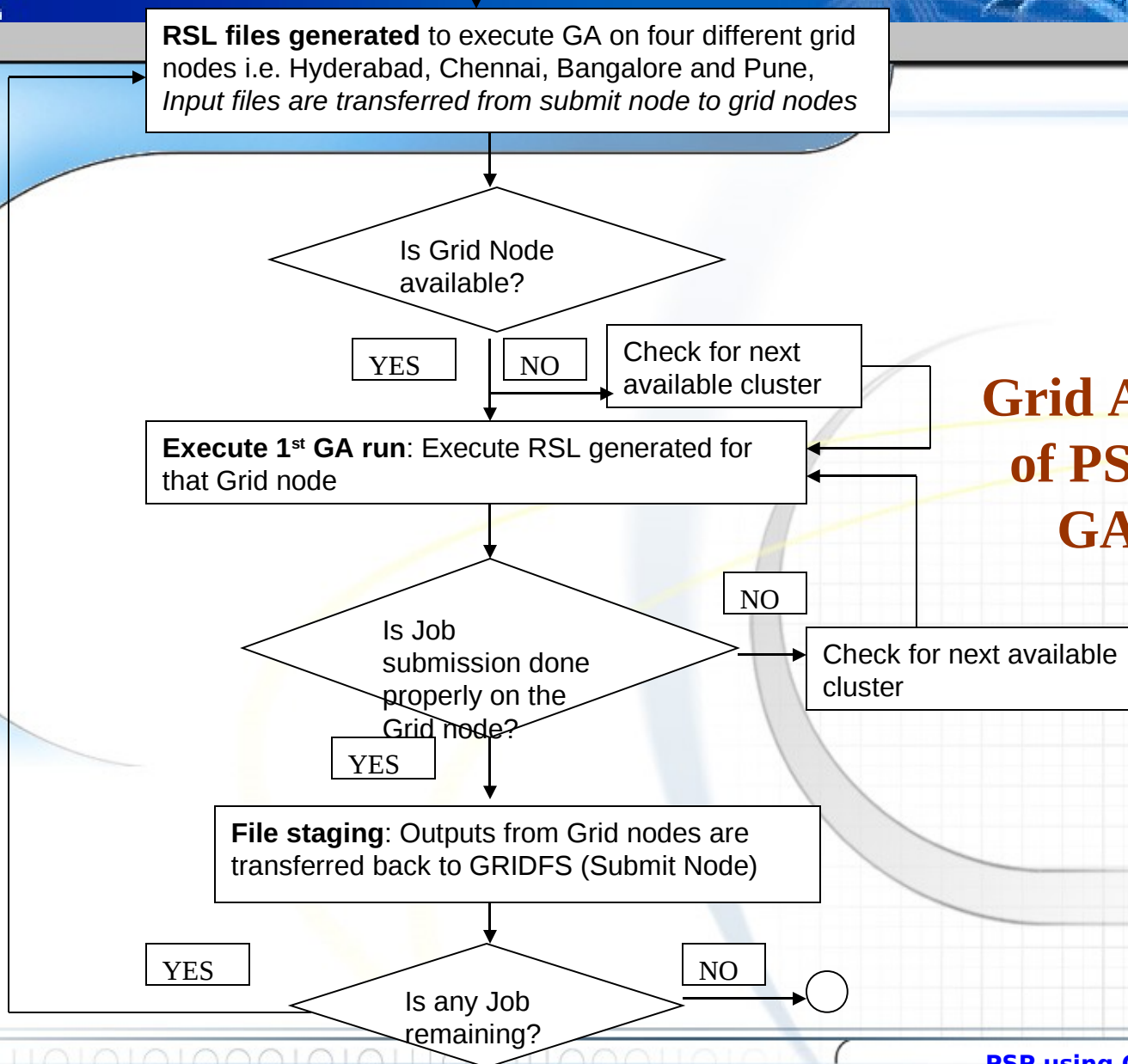
Run Fdivide program on protein file to create sub files  
with protein sequence splitted into multiple parts. Each  
part will have residues of total select+skip residues

On each part of protein molecule, Genetic  
Algorithm (GA) code to be executed  
**Number of GA Jobs to be executed =**  
**Number of Parts X Number of Runs**

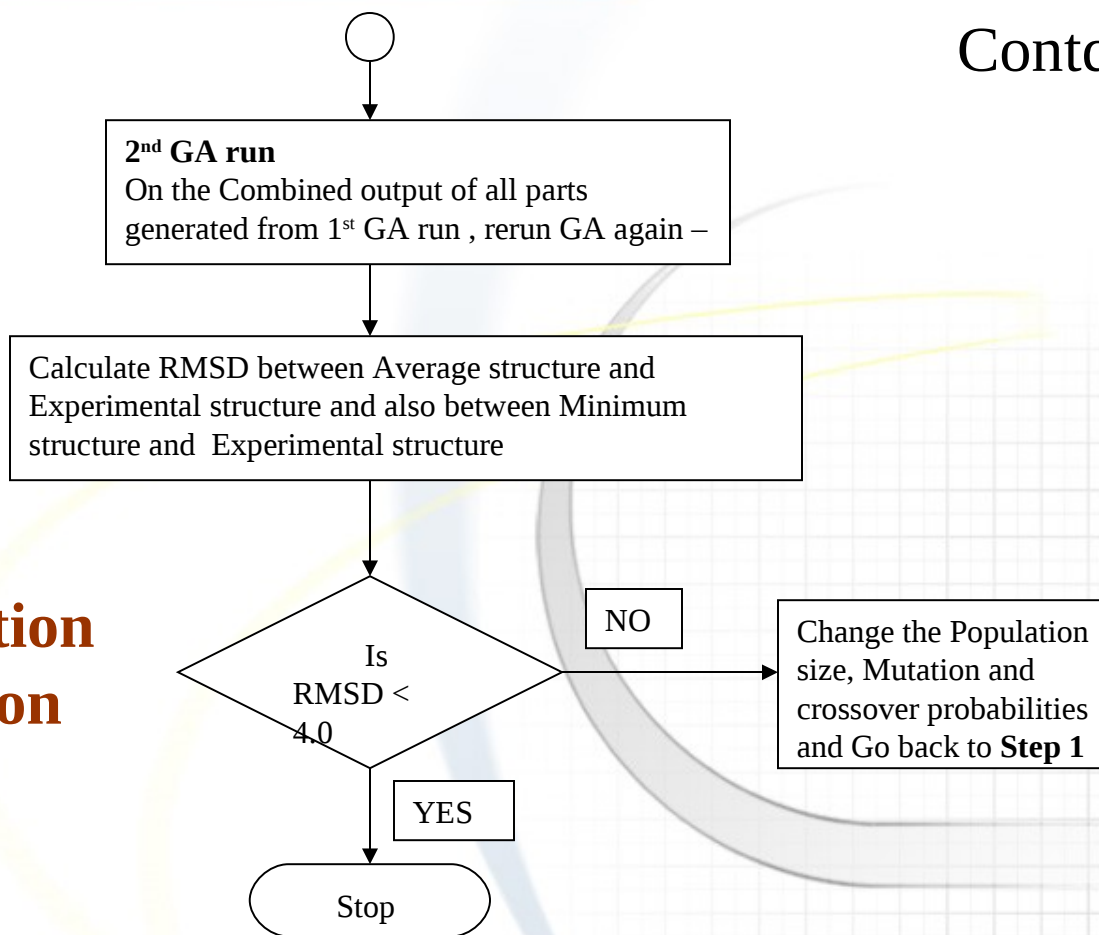
## Grid Automation of PSP code on GARUDA

Contd....

# Grid Automation of PSP code on GARUDA



Contd....



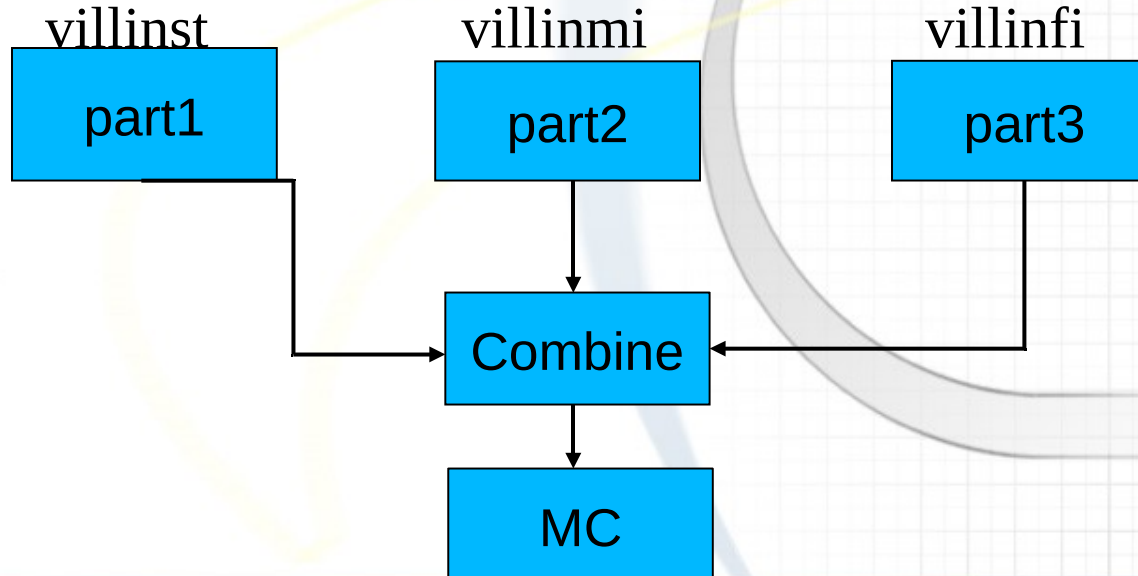
## Grid Automation of PSP code on GARUDA

## Case Study I

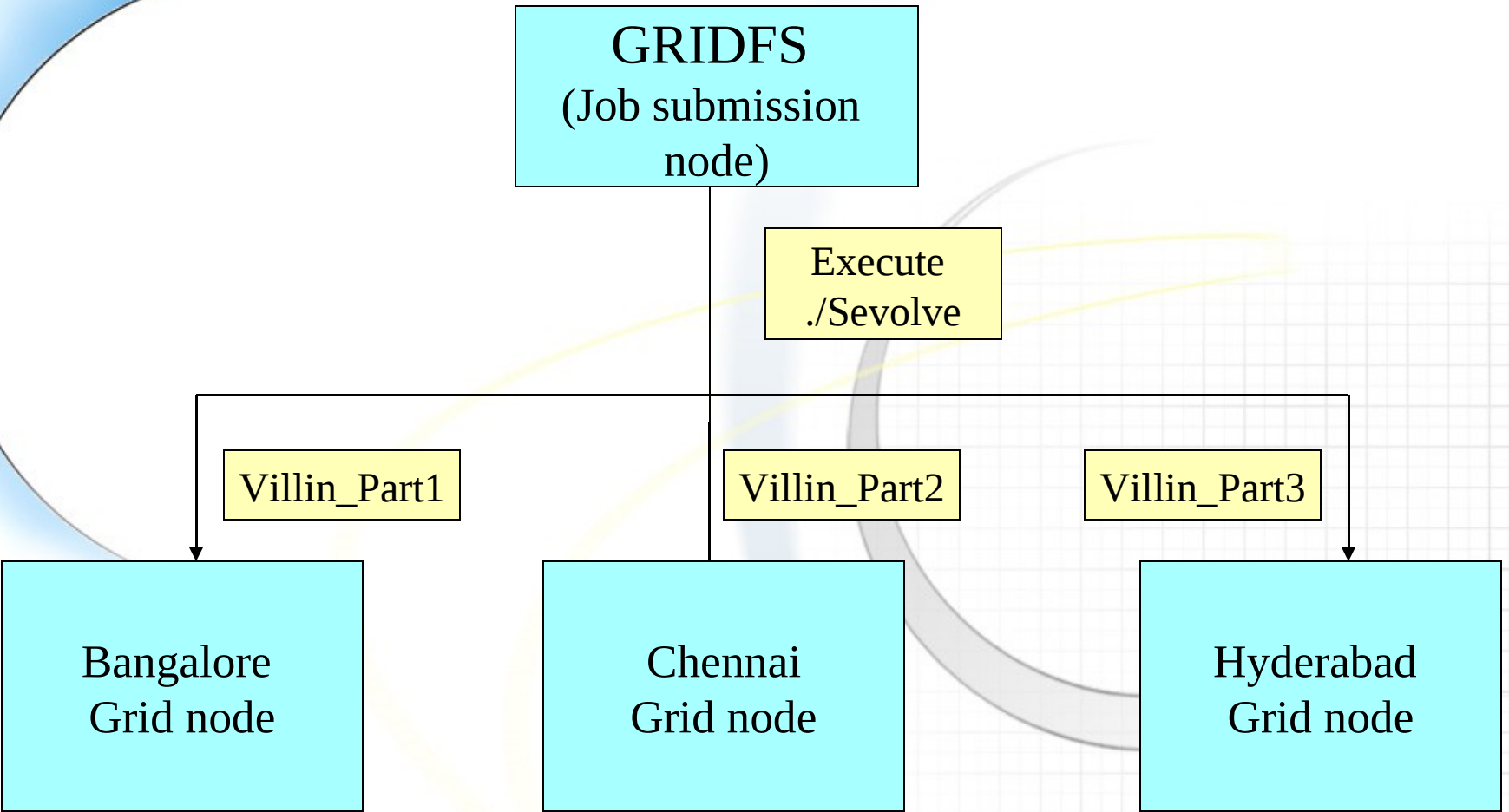
- Simulating protein of 36 amino acids – Villin
- Simple Genetic Algorithm and Monte Carlo methods used
- To make a work flow by splitting the protein into smaller parts and execute GA codes
- Finally the full villin molecule is simulated using MC code

# Implementation

- Implemented a simple workflow
- Parts 1 to 3 execute GA code for parts of the protein
- Advantages:
  - Reduction in Complexity
  - Possibility of concurrency



# Job Submission on Grid



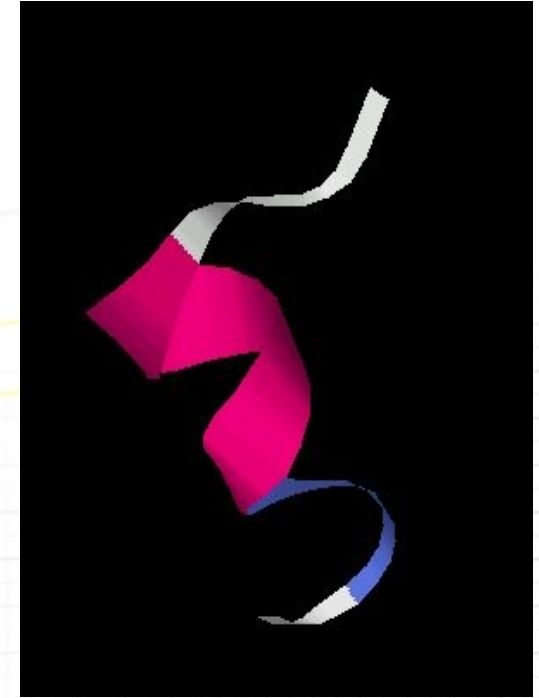
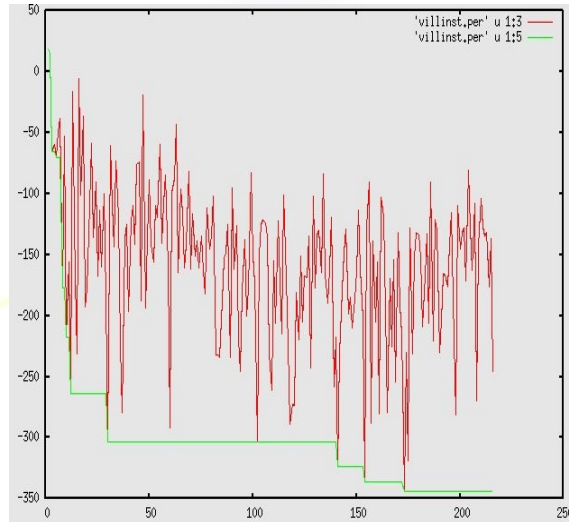
# Submitting across Grid using RSL

```
+  
(  
  &(resourceManagerContact="che01/jobmanager-pbs")  
  (count=1)  
  (label="subjob 0")  
  (environment=(GLOBUS_DUROC_SUBJOB_INDEX 0)  
                (LD_LIBRARY_PATH /usr/local/globus/lib/))  
  (directory="/home/garuda/garuda1/NEW_proteinmod/NEW_proteinmod/data")  
  (executable="/bin/sh")  
  (arguments="villin_part1.sh")  
  
  (file_stage_out=(/home/garuda/garuda1/NEW_proteinmod/NEW_proteinmod/data  
                  /villprt1.tar.gz $(GLOBUSRUN_GASS_URL)villprt1.tar.gz  
  (stderr=vill_part1.err)  
)
```

Shell script: villin\_part1.sh

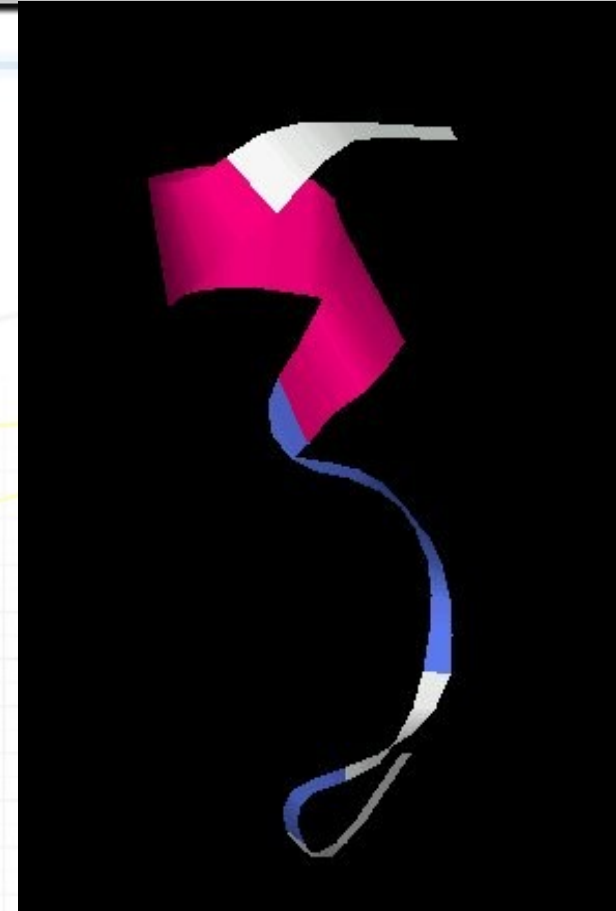
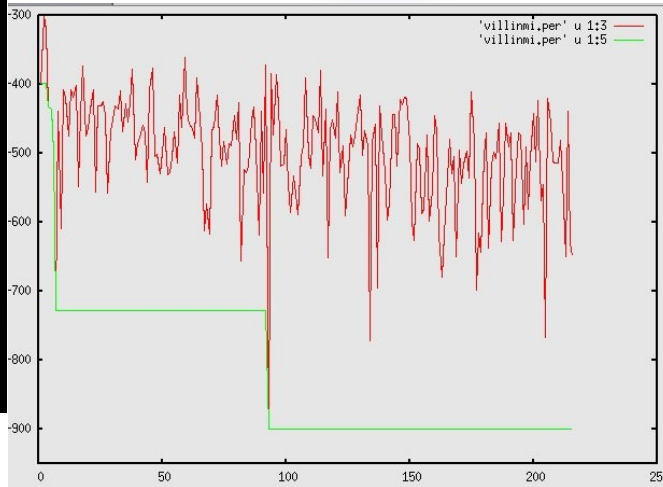
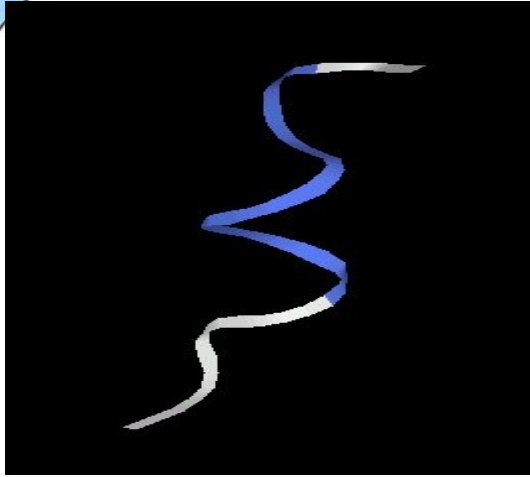
```
#!/bin/sh  
tar xvzf AAA.tar.gz  
chmod ugo+x ./Sevolve  
./Sevolve < vill1  
tar -czf villprt1.tar.gz villprt1.*
```

# Results from first part



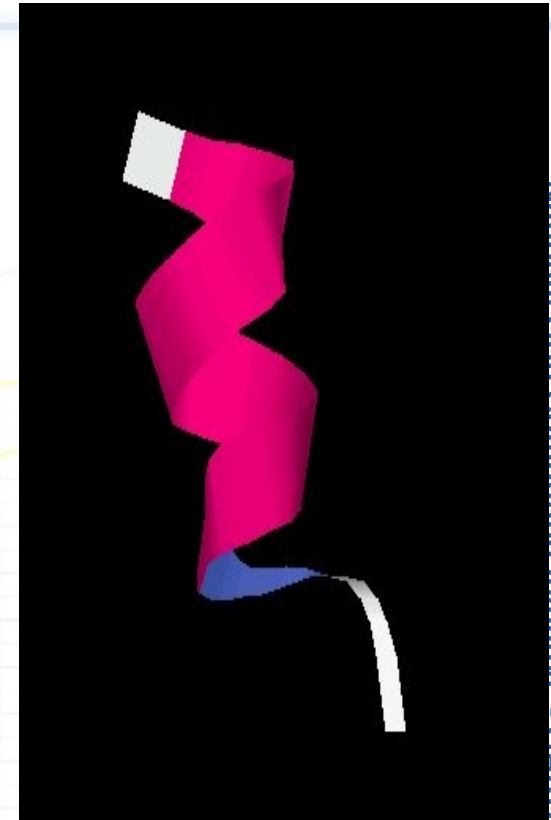
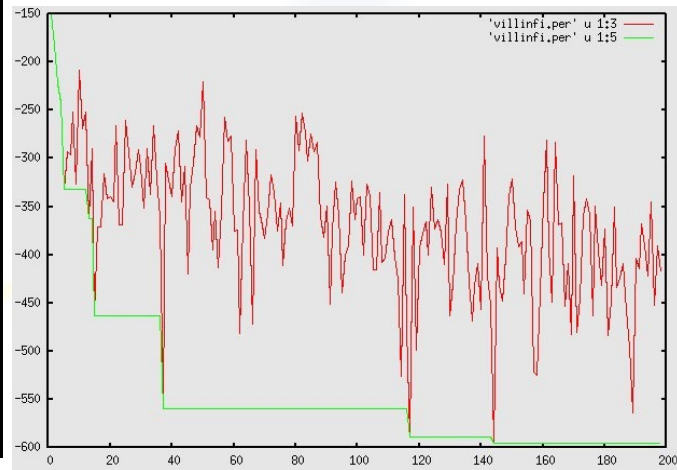
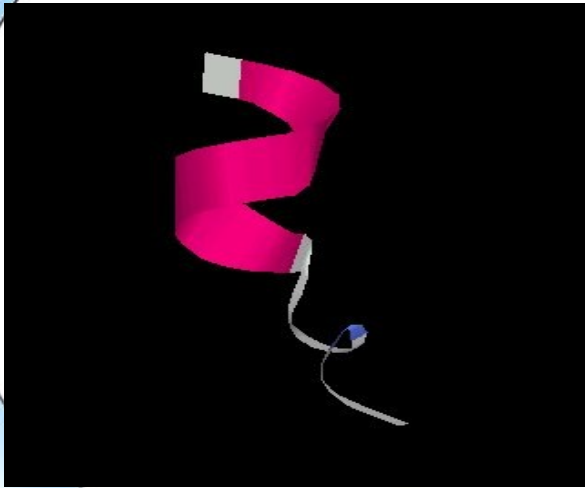
Time: 00:06:00  
2:57 + 3:03

# Results from second part



Time: 00:22:09  
19:15 + 2:54

# Results from third part



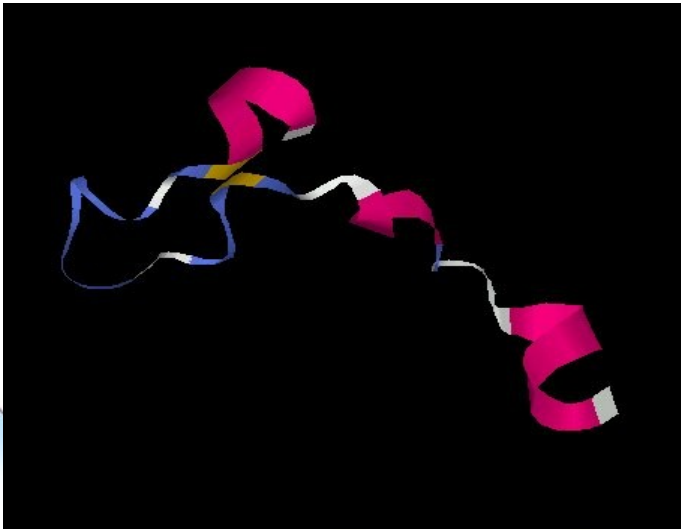
Workshop on Developing Applications on Grid - GARUDA

National Grid Computing Initiative - GARUDA

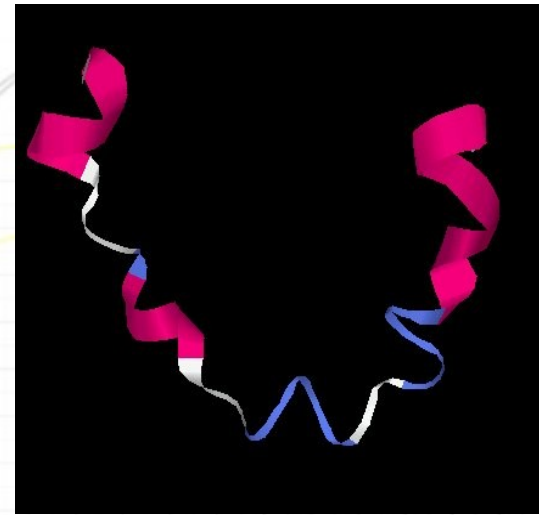
Time: 00:20:12  
12:28 + 7:44

# Result of the final MC program

Time : 00:05:12  
1:50 + 3:22



After 5000 MC  
steps

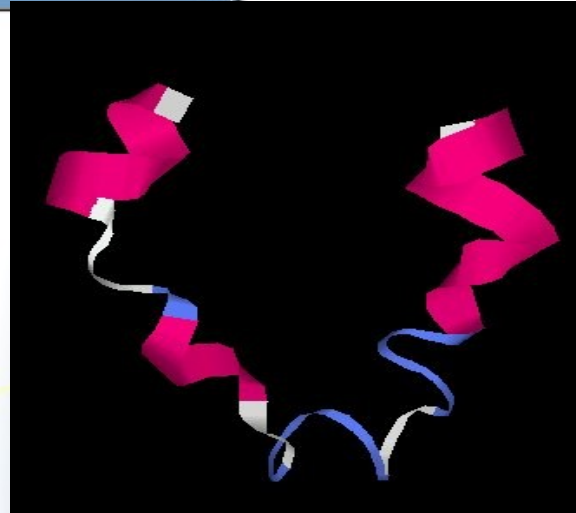


# Experimental result

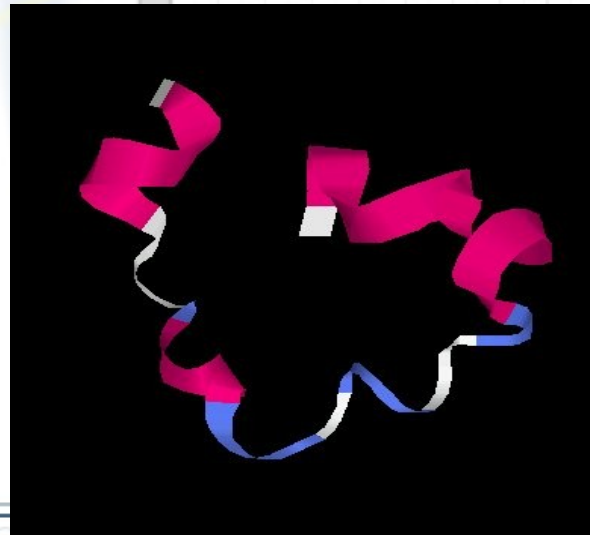


Time: 00:15:37  
11:56 + 03:41

# Predicted result

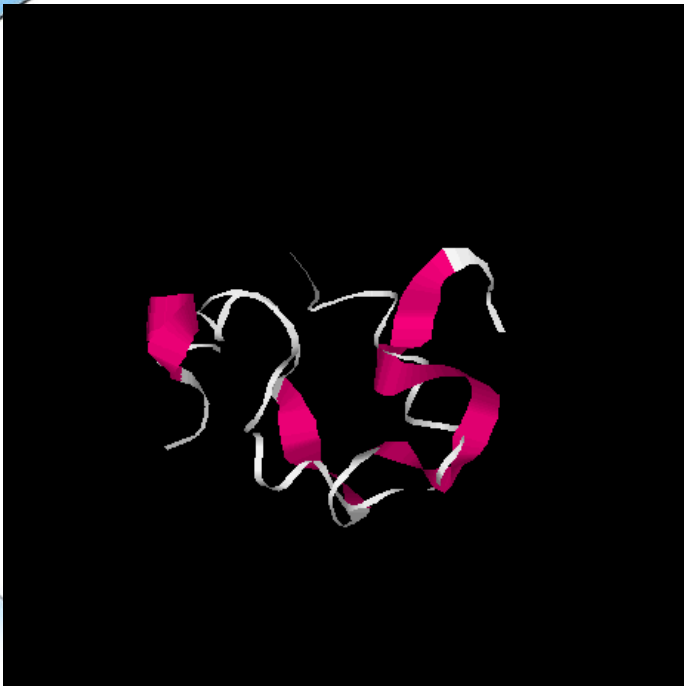


Single AA  
junction

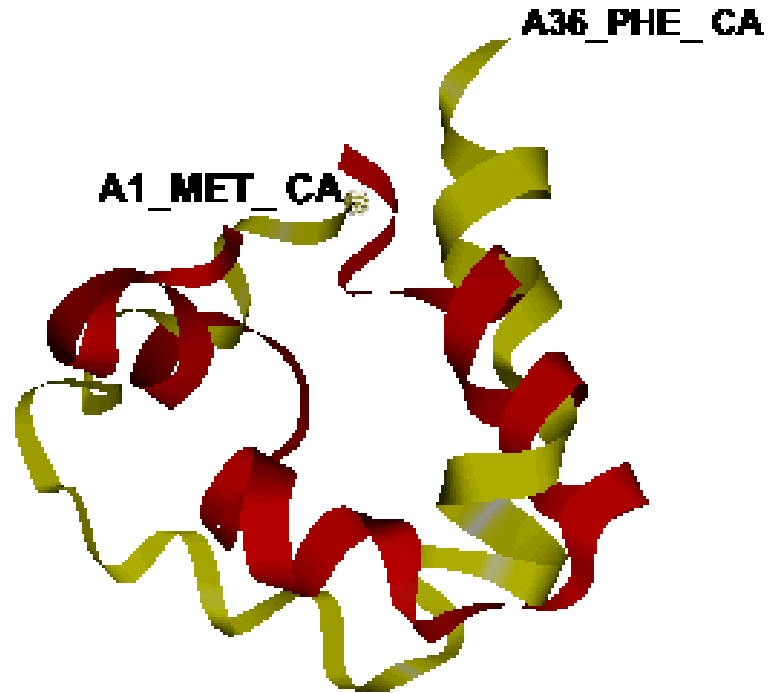


Three AA  
junction

# Divide & Construct method : Villin head piece



simple optimisation



Divide & Construct

Main Chain atoms aligned RMSD: 7.0900  
All atoms aligned RMSD: 8.7340

## Case study II: Tumor suppressor Protein

- Tumor suppressor p53 complexed with DNA - 1tup.pdb
- A 219 amino acid sequence contains more of  $\beta$ -sheets

| Number of AAs | Population size | Number of function evaluations | Time taken on 1CPU |
|---------------|-----------------|--------------------------------|--------------------|
| 219           | 100             | 50000                          | 10hrs              |
| 219           | 300             | 450000                         | 52hrs              |

**Expected time for population size of 7848 (lchrom = 3924 i.e.  $(9*(2*219-2)) = 684$  hrs = 28.5 days**

## Parallel job execution on GRID GARUDA

- As the complexity of GA increases with increase in Population size , the 1tup molecule is splitted into 9 parts each having 30 amino acids.
- Two Amino acids are overlapped between 1st and 2nd part, 2<sup>nd</sup> and 3<sup>rd</sup> part, 3<sup>rd</sup> part and 4<sup>th</sup> part and so on.
- Each part is searched using GA code on Grid nodes parallely.

| Number of AAs | Population size | Number of functions evaluated | Time taken on 1CPU |
|---------------|-----------------|-------------------------------|--------------------|
| 30            | 100             | 50000                         | 19mts              |
| 30            | 300             | 450000                        | 3hrs               |
| 30            | 600             | 1800000                       | 11hrs              |

The time taken is linear with the number of function evaluations

## What to do with Grids?

- Sequential application any where any time
- Multiple runs of applications (Monte Carlo type)
- Multiple simulations (N-body type )
- Distributed algorithms with minimal communication
- Parallel algorithms preferably on a specific cluster (but, any where)
- More effective utilisation of resources
- More effective web-services

## What not to do with Grids ?

- Frequently communicating parallel codes
- A flow that has interdependencies among jobs

Bart Jacob (<http://www-128.ibm.com/developerworks/grid/library/gr-design.html>)

## Conclusion

- GA converged to energies which are lower than that for the experimentally determined structure
- Major problem lies with the fitness since there are no clues to know about the native structures from the single force field function
- These methods are also highly applicable to other application areas as well