

CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
 SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
 Integrated Infrastructure Initiative

JRA4 – Production status of second set of applications

Deliverable ID: D-JRA4-5

Due date: April 30, 2006

Actual delivery date: May 15, 2006

Lead contractor for this deliverable: IDRIS – CNRS, France

Project start date : May 1st, 2004

Duration: 4 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	X
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Content

Table of Content.....	2
1. Executive Summary	3
2. Introduction.....	4
3. GeneHunter-TwoLocus port and installation	4
4. Molecular Dynamics software: installation, optimization and development.....	5
5. Automatic MD Set-Up procedure for massive MD simulations: MoDIG	12

1. Executive Summary

This document is the PM24 deliverable of the Joint Research Activity in Life Sciences. It contains a progress report on the support activities related to the deployment and operation of the two genomic projects initiated at PM18; one at IDRIS – in the area of human genetics of strong familial myopia - and the other at BCS – in the area of protein dynamics.

This document is publicly available.

2. Introduction

This deliverable constitutes a progress report on the new application selected by IDRIS at PM18 to continue with the genomics activity after the final report on the initial set of applications (the port of the GeneHunter-TwoLocus parallel code).

It also reports on the ongoing work at BSC, that joined the project in May 2005 and the Life Sciences activity in October 2005, in the field of protein dynamics:

- Automatic Molecular Dynamics Set-Up procedure for massive MD simulations (MoDIG), where BSC has developed a SETUP procedure that allows any user to launch MD simulations in an almost automatic way, saving days or weeks or tedious and error-prone work. Overall, MoDIG allows us to make a much efficient use of MareNostrum or any other supercomputer facility.
- Molecular dynamics software (installation, optimization and development), where BSC has gotten running in an optimized and fast manner programs, like Amber, Gromacs and NAMD, in addition to corrections made to the formulas computed by these programs.

3. GeneHunter-TwoLocus port and installation (Olivier Glorieux, IDRIS – CNRS)

3.1 - Background

This project arises from a member of the Toulouse U563 INSERM team (Dr Sandrine Paget), who is working on the subject of the human genetics of the strong familial myopia. The requested access to the DEISA supercomputing infrastructure and the life science support services results from her need to perform a large scale analysis using GeneHunter-TwoLocus, a MPI parallelized code. Her objective is to use this code to locate the genes linked to the disease, testing 8 loci to one another in a 700 people crowd. This program is too demanding to be run in the PC cluster available at Toulouse's Genopole. Even a very partial calculation does not fit into the maximum 24 hours on 20 processors. Access to larger supercomputers is needed; particularly view of a sustained activity in this field by a larger scientific community

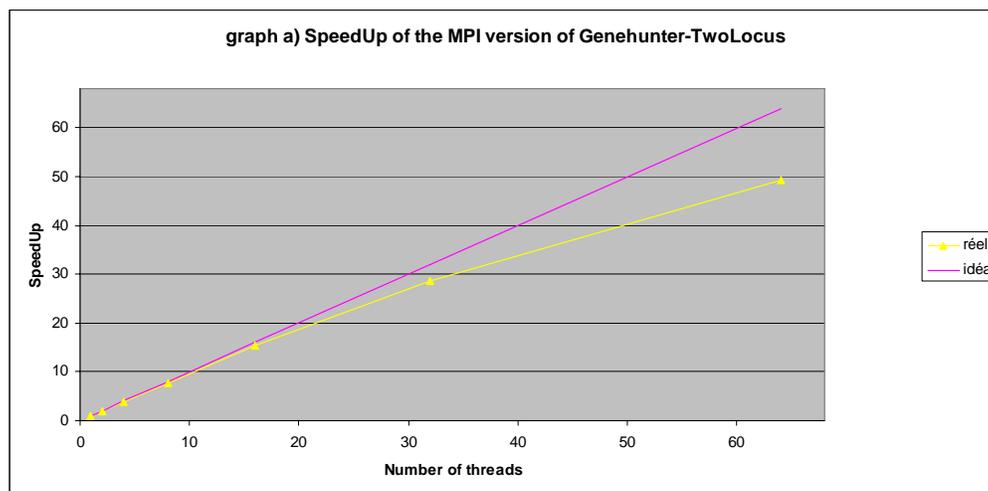
The purpose of this activity is to install and optimize the software, initially for the IB SP4 systems of the DEISA distributed environment. Besides operating this particular project, the purpose is also to make the software available to a wider user community.

3.2 – Progress report

Genehunter-TwoLocus has been successfully recompiled on the IBM SP4 platform Zahir. However, I turned out that, when the software was tested with its own dataset, it did not execute correctly when compiled with a64 bits addressing.

It took some time to identify the source of the error. Following discussions with the developers of the MPI version of the code, we identified the source of the error and corrected it by modifying the MPI function calls.

The following graph shows the speedup obtained on the “high performance” partition of the IBM SP4 platform at IDRIS (P655 processors, 4 processors per node without sharing of the L2 cache) using the datasets relevant for Dr Paget application (Figure A).



The two courbes represent the speedup for two different datasets relevant for the project. Excellent scalability is observed for a moderate number of processors. Work is in progress to push this scalability as far as possible, before launching the production runs.

4 - Molecular dynamics software: installation, optimization and development. (Jordi Camps (BSC), Albert Pérez (IRB-BSC), Carles Ferrer (IRB-BSC))

4.1 - Abstract

Mare Nostrum is a supercomputer that is being used for a large variety of calculations. Some of these calculations are related to Molecular Dynamics simulation.

There are a number of programs related to this subject. Amber, Gromacs and NAMD are some of them. This report exposes the steps followed in order to get these programs running in an optimized and fast manner. Also there are corrections made to the formulas computed by these programs.

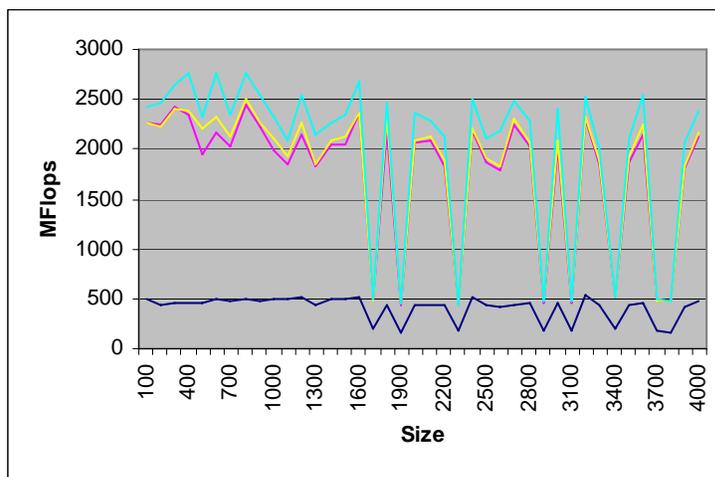
This report also includes information about some software developed in order to process the files generated with one of the packages with another.

4.2 - Gromacs installation

Gromacs comes in different formats: source and binary. Binary format is useful for a quick deployment and use of the software, but binary distributions can not be compiled with the most optimum flags for every architecture. Therefore, we used the source distribution to be able to fine tune the execution speed.

One of the requirements of Gromacs software is the availability of a Fast Fourier Transform library called FFTW (Fast Fourier Transform in the West). This library had to be downloaded and compiled in a similar manner than the Gromacs software itself.

In the graphic to the right we can see the megaflops obtained with a basic, non optimized, compilation (shown in dark blue), and the different results obtained with different compilation options. We can see that the light blue line is better than the other ones, so the compilation flags used to compile this package were the corresponding to that line. Similar tests have been performed with all the software compiled in order to get the quickest execution time possible for each program.



The first step to compile FFTW was to set up correctly the environment variables. This was the result:

```

MPICH_HOME=/opt/osshpc/mpich-gm/64/ssh
MPICH_CC=xlc_r
MPICH_CCC=xlc_r
MPICH_F77=mpif77_r
MPICH_F90=mpif90_r
MPICH_CLINKER=xlc_r
MPICH_CCLINKER=xlc_r
MPICH_F77LINKER=mpif77_r
MPICH_F90LINKER=mpif90_r
PATH=$MPICH_HOME/bin:/opt/ibmcmp/xlf/9.1/bin:/opt/ibm
    cmp/vac/7.0/bin:/opt/ibmcmp/vacpp/7.0/bin:$PATH
CC=xlc
CCC=xlc
FC=xlf
F77=f77
F90=xlf90
CFLAGS="-qarch=ppc970 -qtune=ppc970 -qaltivec -
    qenablevmx -qhot=arraypad -qhot=simd -
    qhot=vector -qinline -qipa=level=2 -qstrict -O5"
FFLAGS="-qarch=ppc970 -qtune=ppc970"
OBJECT_MODE=64

```

```
P4_GLOBMEMSIZE=33554432
```

Now we were prepared to launch the compilation in single and double precision. The commands used were:

```
$ make distclean
$ ./configure --prefix=$HOME/installed --enable-mpi -
  -enable-threads --enable-type-prefix --enable-
  float
$ make
$ make install
$ make distclean
$ ./configure --prefix=$HOME/installed --enable-mpi -
  -enable-threads --enable-type-prefix
$ make
$ make install
```

Once the FFTW libraries were installed, we could start the Gromacs compilation itself. In this case we added some environment variables more:

```
LDFLAGS=-L/home/bsc23089/installed/fftw/lib
CPPFLAGS=-I/home/bsc23089/installed/fftw/include
CFLAGS='-qarch=ppc970 -qtune=ppc970 -qenablevmx -
  qhot=arraypad -qhot=simd -qhot=vector -qinline -
  qipa=level=2 -qstrict -O5'
```

The more interesting part is the CFLAGS. Heavy optimizations are applied: full InterProcedural Analysis, High Order Transforms, vectorization...

Benchmarks have been executed changing some optimization options in order to see the performance impact and choose the best combination.

Once I get the best flags for FFTW and Gromacs, I made the definitive compilation of the package using the following commands:

```
$ make distclean
$ ./configure --prefix=$HOME/installed/gromacs --
  exec-prefix=$HOME/installed/gromacs --x-
  libraries=/usr/X11R6/lib64 --enable-fortran --
  enable-mpi --enable-ppc-altivec --with-fftw --
  with-x
$ make
$ make install
```

The most interesting details are the `--enable-mpi` flag, in order to use more than one processor, and the `--enable-ppc-altivec` for the use of the AltiVec vectorization unit provided by the PowerPC architecture.

This compilation steps take a long time due to the Interprocedural Analysis, but the results are slightly better than without IPA. I considered that is a good trade-off to compile one time and execute lots of times.

4.3 - Amber installation

The steps followed in the Amber installation procedure are similar to the ones followed in the Gromacs installation procedure. The first step was to get the source code and

configure the basic environment variables. Second one was to patch the source code with the web distributed patches in order to eliminate known bugs.

Because our architecture was not common, the configuration script was not tested with our hardware and I had to debug it in order to get the MPI version compiled and working. Once the *configure* script was patched, it was executed and the *config.h* file was generated. Some changes were made to this file to achieve our goal: compile the code with the optimizer compilers of IBM, the xl series. Finally we could launch a compilation for the serial version of the programs.

After this compilation, the built-in tests were executed to test the correctness of the results. Being this results correct, we launch a compilation for the parallel version and the corresponding tests, which were also correct.

Apart from the main package, the *pmemd* executable had to be compiled separately because is the newest addition to the package and is not completely integrated. But the process is quite similar.

In the first place, a machine-specific file must be created (like the *config.h* file). For a successful compilation, two pre-processor flags must be defined:

- `-traditional-cpp`
- `-DNO_C_UNDERSCORE`

And then the classic `./configure; make; make install` procedure. The second flag is specially important because without it, the code compiled can not be linked. It took a long time to figure out what was happening until I realized that different compilers were used simultaneously (Fortran and C compilers) and that they generate the object files with different nomenclatures. Once the program is compiled, some tests are performed, and if the results are correct we continue with the compilation of the parallel version. A new pre-processor flags must be added: `-DMPI`

Also, the environment must be configured:

```
$ MPICH_INCLUDE=$MPICH_HOME/include
$ MPICH_LIBDIR=$MPICH_HOME/lib
$ MPILIB="-L$MPICH_LIBDIR -lmpich"
```

And now we can repeat the `./configure; make; make install` procedure. The last step was to test again the result executable to make sure that the calculus are being computed properly.

4.4 - NAMD installation

The third Molecular Dynamics simulation package was installed by the system administrators of the Mare Nostrum, but the testing of the installation was done by us. The program work as expected in its native force field, but when we switch to the OPLS force field, the results were wrong. This leads us to investigate the reason for the incorrectness and we discovered some errors in the parameter files.

OPLS functionality was added in the latest iteration of NAMD. It was not so much tested and the force field parameters were partially wrong and incomplete. It was necessary to comment out some lines in the files and add the parameters for some standard ions. These parameters were extracted from the Charmm software force field parameters.

Once the parameters were corrected, the new test simulations were executed and the results were correct. At this moment, we are the only centre in the world with the corrected parameters and we are able to run OPLS force field simulations with NAMD.

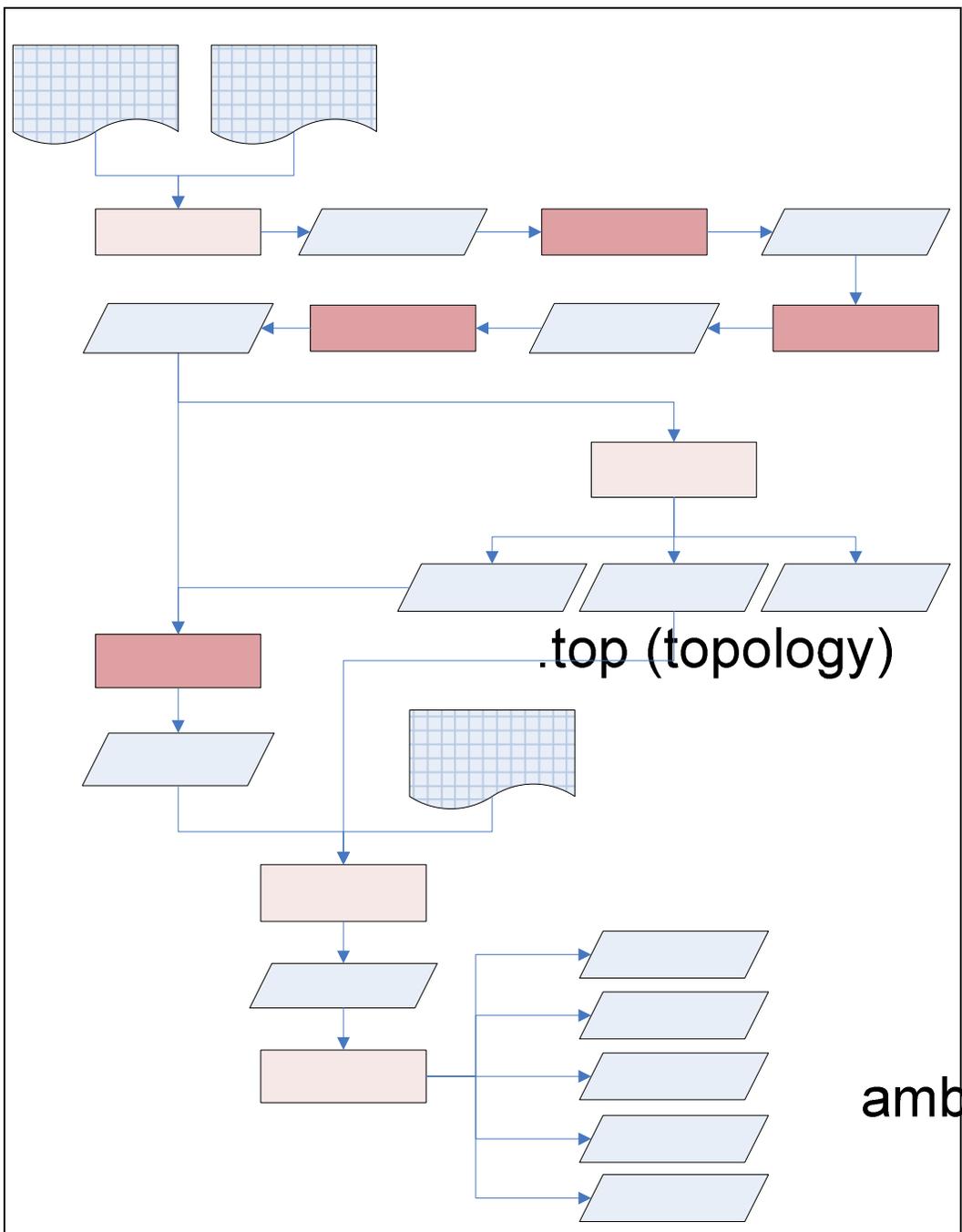
4.5 - Interformat converter

A necessity to use Amber files with Gromacs aroused in the group. This leads to the development of a set of scripts for converting Amber restart files to Gromacs restart files for the simulation of the same structure with another package.

A series of scripts has been developed to aid in every step of the conversion process. The scripts are the following:

- `convert.pl`: This is the main tool. Its purpose is to execute the entire conversion pipeline. It calls the rest of the programs used in the process, the new ones developed by ourselves and the utilities used belonging to the MD simulations package.
- `pdb2gro.pl`: This script converts a PDB input file into the Gromacs input format.
- `adaptgro.pl`: This script changes the residue names from the ones used by Amber to the conventions of Gromacs. It also removes the atoms not needed if a unified force field (like G43a1) is used.
- `addvelocities.pl`: This script is the responsible for the addition of the velocities needed for a restart in the same conditions (temperature, direction of movement...) as the source files.
- `renum.pl`: This script renumbers the atoms and residues of a `.gro` file. This is necessary since in the adaptation process we delete some atoms and the number correlation is lost.
-

The complete pipeline is shown here:



.r (Amber)
(x)

ambpdb

Diagram 1: Execution flow for a Amber to Gromacs conversion

.gro (x, v)

Similar procedures have been established in order to convert amber systems to charmm and OPLS force fields. These two force fields can be run in NAMD program suite. Basically we developed different scripts that read a restart file from amber trajectory to

generate psf topology and velocities, coordinates and xsc files files based on Charmm22 or OPLS force field needed in NAMD program. Also a NAMD parameter file is generated satisfying all conditions suitable to run Molecular Dynamics in a comparable way that amber uses.

In the next diagram it can be noted the flowchart of work for the conversion.

We start with the amber topology file and the equilibrated restart file from the MoDEL setup platform. With these files we obtain two pdb files one for the coordinates and the second for the velocities of the system, and xsc format file is also generated with the box parameters. In the next step the pdb files are translated from amber atom and residues names domain to charmm or OPLS name domains using the suitable dictionary. Those dictionaries have been obtained comparing aminoacid and ions libraries from amber charmm and opls force fields. With the coordinates and velocities translated pdb using perl scripting and psfgen program from NAMD suite we obtain psf files for every system and input files in order to run Molecular Dynamics under NAMD platform.

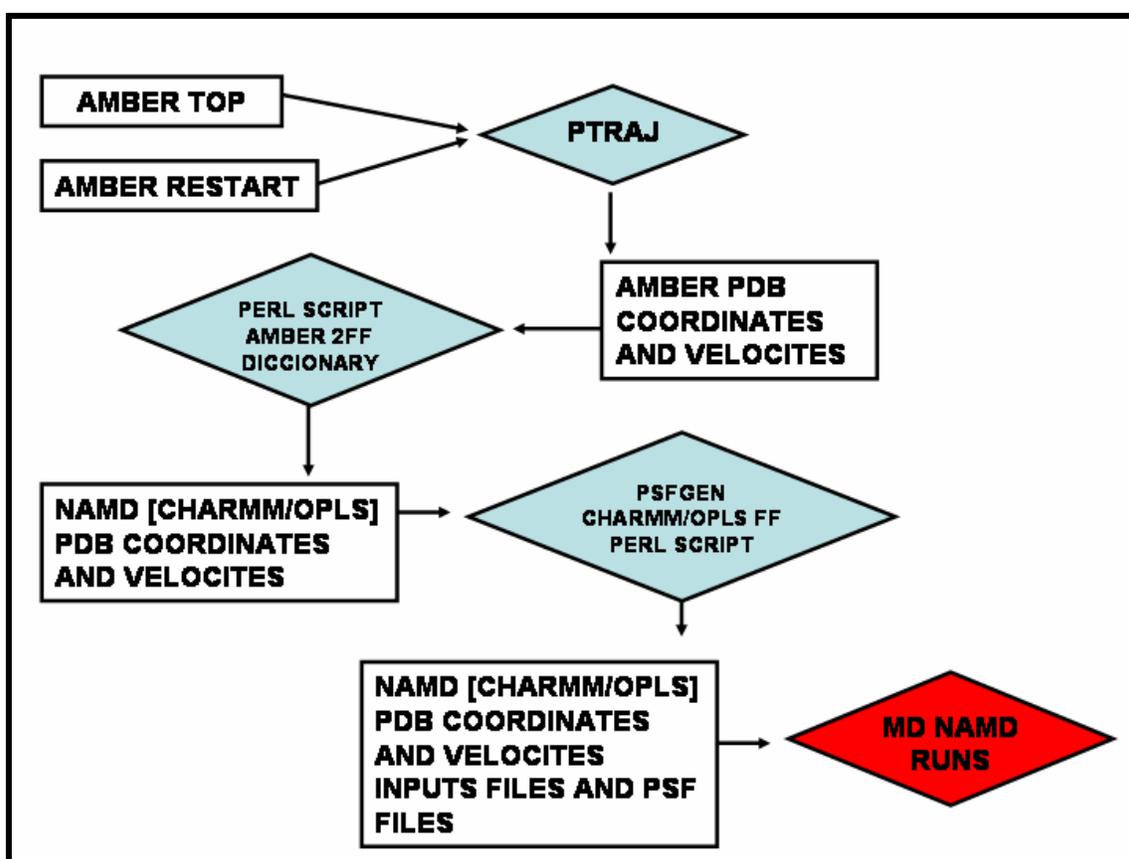


Diagram2: Work flowchart in setup charmm and opls MD under NAMD from amber

4.6 - Trajectories and output transformation for analysis.

The final objective for the work developed in this project is to compare different force fields for protein molecular dynamics. Our idea is to analyze all force fields under the analysis flowchart developed in MoDEL database. To accomplish this requirement we need all trajectories under the same amber trajectory format with the same amber atom order. To this end we develop several perl scripts that read a restart and topology files

from amber and a topology and pdb files from other force platform reorder the trajectories in an amber way

In the diagram 3 it is represented the flow of work in the reordering of trajectories from charmm and OPLS force fields run under NAMD suite and gromos force field under GROMACS package. Using the same set of dictionaries used in the amber to FF translation we obtain the right amber order with a set of perl scripts that read amber topologies, amber restart and pdbs and psf from charmm OPLS or gromos trajectories. With this new order we can reorder the original trajectory from NAMD or GROMACS converted previously to a trajectory format using PTRAJ program. With these trajectories in an amber order and format we can submit the analysis and comparison under MoDEL analysis suite.

5. Automatic Molecular Dynamics Set-Up procedure for massive MD simulations: MoDIG. (Tim Meyer, Manuel Rueda, IRB-BSC)

5.1 – Introduction

Molecular Dynamics Packages such as AMBER have evolved to a state where even non-experts can run simulations of almost any protein whose atomic structure is known. But any simulation is only as good as its input data and many articles, and tutorials on simulation setup have been published. Simulation setup is usually a multi-step procedure that requires numerous user inputs and expert decisions. Interactive or semi-automatic front-end programs for molecular dynamics simulations that include structure setup tools exist but to our knowledge there is no program that does a completely autonomous simulation setup including state of the art protonation analysis and ligand parameter setup.

MoDIG has been developed and tested on over 1.000 Proteins of all folds during the creation of the Molecular Dynamics Library MoDEL at the Barcelona Supercomputing Center. The SETUP procedure developed here allows any user to launch MD simulation in an almost automatic way, saving days or week or tedious and error-prone work. Overall, MoDIG allows us to make a much efficient use of Mare Nostrum or any other supercomputer facility

5.2 - Programs and Forcefields used

5.2.1 – AMBER

AMBER (Assisted Model Building with Energy Refinement) is the collective name for a suite of programs that allows users to carry out and analyze molecular dynamics simulations. The programs used during setup are listed below. The AMBER package includes programs also a variety of forcefield. Per default simulations are setup using the well established parm99 forcefield for the protein- and nucleic acid molecules and the gaff⁷ forcefield for ligand molecules.

Antechamber

Antechamber is a set of tools to generate input files for organic molecules, which can then be read into LeEaP, the main simulation setup program in the AMBER suite. We use in conjunction with the gaff forcefield to automatically generate parameters for so-far unparameterized ligand molecules.

LeAP

LeAP is the besides Antechamber the main preparation program in AMBER. It constructs the biopolymers from a pdb file and adds missing atoms according to its residue libraries. It then solvates the system using pre-equilibrated water boxes and writes atom positions and all forcefield parameters to two files which serve as input for the simulation programs sander and pmemd.

Sander

Sander is the original molecular dynamics program which performs the simulation. Besides it has the ability to energy minimize a given structure which is required in various point during the setup.

5.2.2 – CMIP

CMIP (Classical Molecular Interaction Potential) developed locally that allows us to calculate the interaction energy between two molecules at great accuracy and speed. Its functional defines the potential around a molecule as the sum of steric and electrostatic contributions where the steric term is defined by a 12-6 Lennard-Jones expression and the electrostatic term from solving numerically the Poisson equation.

Titration of structural waters

CMIP has shown excellent results in predicting the positions of structural water molecules in high resolution protein crystals. These water molecules play an important role in protein stability and their absence complicates or even impedes full equilibration of the protein and leads to artifactual results. Crystal waters can be detected in high resolution xray experiments but are mostly absent in lower resolution crystal structures and completely absent in NMR structures.

Titration of Counter Ions and Salt

Many proteins have a net-overall charge that needs to be neutralized before simulation. Monovalent ions are typically used for balancing since they equilibrate much faster than higher-valent ions due to lesser electrostatic interactions. In addition to the neutralizing ions we also add the equivalent of 50mM Salt ions to simulate a physiologic environment. Correct initial placement of these so-called bulk-ions can shorten the time required to equilibrate the ion atmosphere.

Protonation states of Protonatable Residues

Various Amino acids can exhibit different protonation states depending on their environment. Histidine for example can be either neutral in two different tautomeric forms or positively charged. Since protons do not reflect x-rays they can not be detected

by x-ray crystallography and their positions have thus be determined by other means since wrongly protonated residues can significantly destabilize a protein. We used CMIP to calculate the solvation- and interaction energies of all possible protonation states of all titrable side chains to find the best possible combinations.

5.3 - MoIDIG PROGRAM

The PDB Structure undergoes a sequence of preparation steps as illustrated in . The program requires as only input parameter the PDB identifier of the structure to be simulated. Changes to the structure can be made manually beforehand in which case the program uses the existing file as starting point instead of retrieving the structure from the PDB database.

Our program is object orientated and based on the Datastructures known from the MMTSB Toolset (for details please refer to [http://mmtsб.scripps.edu/](http://mmtsب.scripps.edu/)). On reading Hydrogen Atoms are identified and stripped to avoid naming problems. Residues are then classified into 6 basic residue types which are amino-acid, nucleic-acid, ligand, monovalent ion, divalent ion, and water. If ligands and/or ions are present they are checked against the ligand library an if needed input parameters are generated using the Antechamber⁸ program and stored to the ligand library for future use. Ligands are parametrized in the Gaff⁷ forcefield and partial charges are calculated on the AM1-BCC^{10,11} level of theory. All water molecules that do not belong to the first solvation shell of any divalent ions are then stripped from the structure, as are all monovalent ions. A simple distance search between all present cysteine residues is the performed to detect sulfur bridges and modify residues accordingly. The LeAP program then adds all missing atoms to the structure and creates coordinate and parameter files required for a short minimization by Sander to fix hydrogen positions and steric clashes. CMIP then adds counter ions required to neutralize the protein. In the computationally most costly step now all possible protonation states of all basic or acid sites are probed to find the most favorable configuration. After each modification a short minimization is performed to correctly accommodate modified side chains.

In case the total charge has changed the protein is again neutralized and structural waters that account for roughly 0.2% of all water molecules and the equivalent amount of 50mM NaCl are titrated using CMIP. A octahedral waterbox with 12A minimum distance from any protein or ligand atom is then created and filled with TIP3P water molecules using LeaP. The final coordinate- and parameter file are created.

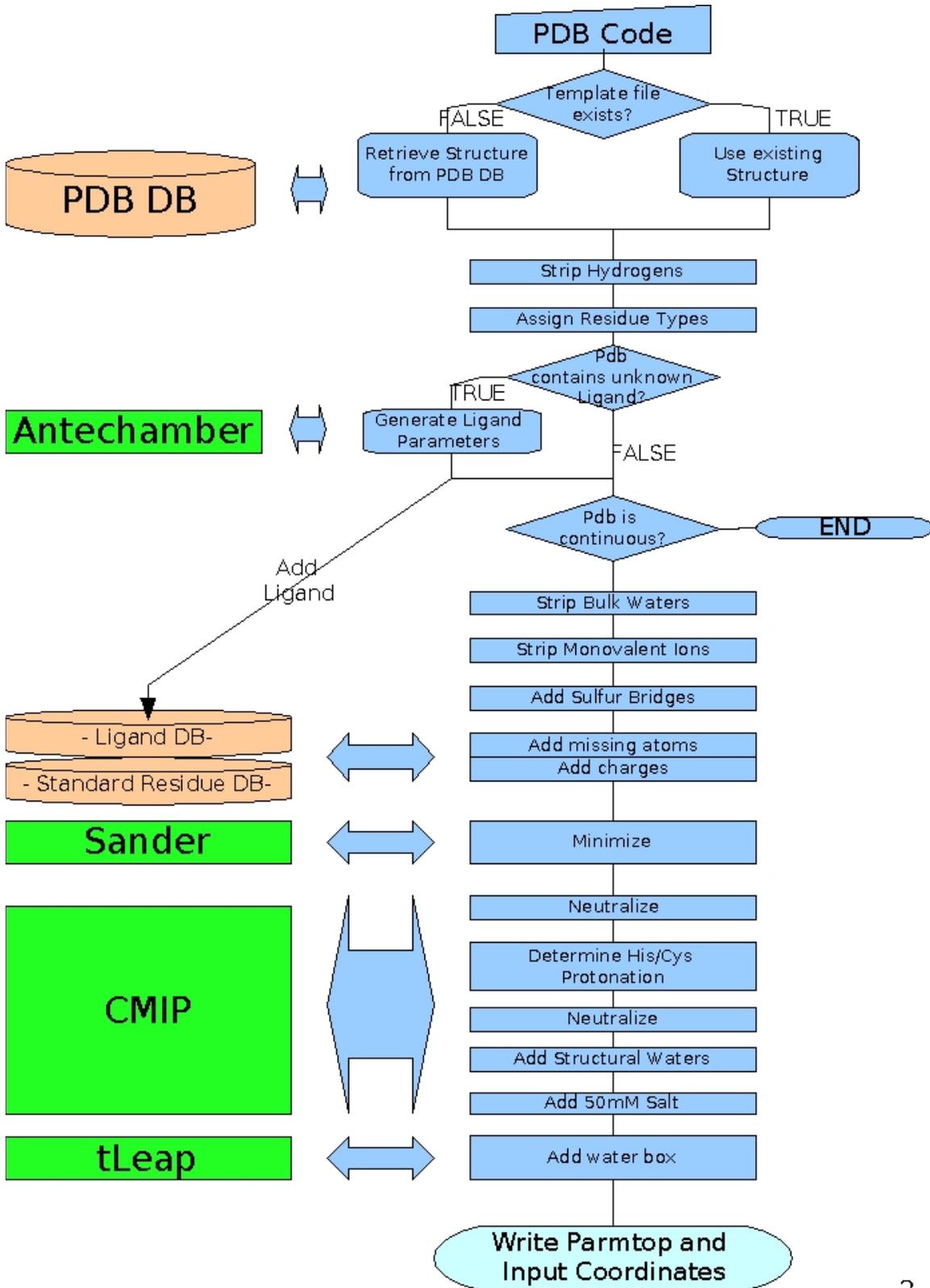


Figure 1 Flow chart of the MolDIG utility.

6. List of Acronyms and Abbreviations

INSERM	Institut National de la Santé et de la Recherche Médicale
CMIP.	Classical Molecular Interaction Potential
MD.	Molecular Dynamics
MoDIG.	Molecular Dynamics Input Generator.
PTRAJ.	Analysis module from the University of California.
MoDel.	Molecular Dynamics Extended Library.
NAMD, AMBER, GROMACS.	Molecular dynamics computer programs
OPLS. GROMOS, PARM.	Force-Fields used in MD simulations.