Parallel Shell Cut

Yongchang JI (yji@cs.ucf.edu)

(Last update November 10, 2004)

CONTENTS

A. INTRODUCTION

Parallel Shell Cut (PCUT) program is used to cut 3D density map's inside nucleic acid and outside empty noise. PCUT use as input: 3D density maps generated by P3DR, output: the cut 3D density map with only the needed shell, it has the same header as the old one. The 3D maps have been stored in the Purdue *.PIF format.

PCUT is used in conjunction with our Parallel Structure Factors (PSF) to get better resolution.

Contact us: yji@cs.ucf.edu, dcm@cs.ucf.edu

B. PROGRAM INPUT

Input parameters of PCUT are as follows :

```
------------------------------------------------------------------------
inputmapname                                    (C)
outputcutmapname                                (C)
inner_rad, outer_rad, cut_step, drop_weight     (I, I, I, F)
------------------------------------------------------------------------
```

1. inputmapname   (C)

Specify the name of the PIF format 3D density map to be used as input.

2. outputcutmapname   (C)

Specify the name of the PIF format 3D map to be used after cut.

3. inner_rad, outer_rad, cut_step, drop_weight   (I, I, I, F)

inner_rad specifies the inner radius that will be cut from center to this radius.

outer_rad specifies the outer radius that will be cut from this radius to the map edge.

cut_step is the soft cut size. The shell from **inner_rad** - **cut_step** to **inner_rad** will use the soft cut method (Gaussian function). It is the same soft cut is also used from **outer_rad** to **outer_rad** + **cut_step**.

Drop_weight is the weight used for soft cut. In the cutting range, from **inner_rad** or **outer_rad** up to **cut_step** pixels, the weight will drop from 1 to **Drop_weight**. In the cutting range but beyond the **cut_step**, the values will be set to be 0.0.

C. EXAMPLE PSF INPUT CONTROL PARAMETERS FILE

rec.pif.even

cut1.even

40, 99, 12, 0.001

D. PROGRAM EXECUTION

Normal operation of PCUT will generate a new map which gets rid of the inside nucleic acid and outside empty noise. The new map has the same header as the old one.

For example:

Execute PCUT with 8 nodes using MPI, the command will be:

% mpirun -nolocal -machinefile mach -np 8 PCUT < Pcut.in > Output

or

% mpirun -np 8 PCUT < Pcut.in > Result

Pcut.in is the script of the input control parameters file, and the **Output** file will be the standard output. We use I/O redirect methods. The **mach** file is used for specifying the hosts

of the cluster that you want to use. Please modify it to indicate the host names of your cluster. If use second command, it will use the default machine configuration under MPICH directory.

For details of running MPI and specifying hosts in the host file, see user guide of MPICH at http://www-unix.mcs.anl.gov/mpi/mpich/.

E. PROGRAM NOTES

The directories of PCUT source code are:

    |-- Commpk   !common routine directory
    |-- PCUTsrc   !PCUT source codes directory
    |-- include   !include files directory
    |-- Makefile

1. Weight function in soft cut region (gaussian):

    f(x,y,z) = f(x,y,z) * exp(- dist**2 / (- cut_step**2 / log(cut_weight)))
    dist $\in$ (1, cut_step) is the distance from the cut radius in the cut region.

F. REFERENCES

G. FLOW CHART FOR PSF PROGRAM

The algorithm for PCUT is as follows:
Step 1: Read the input data;
Step 2: Read 3D map and distribute different slabs to different nodes;
Step 3: Each node cuts its own slab based on input parameters;
Step 4: node 0 gather the result and write out the new map.

PCUT   !! main program
    |--- call mpi_init   !! mpi initialization
    |--- call check_map   !! check the input 3D map
    |--- call bcast_parameters   !! broadcast parameters
    |--- call set_indices   !! initialize parameters
    |--- call cut_map   !! cut the map
    |--- call mpi_finalize   !! finalize mpi

```
CUT_MAP                             !! read, cut, and write map
   |--- call pif_open               !! open the input pif file for read
   |--- call pif_read_gh            !! read the global header of the input pif file
   |--- call pif_open               !! open the output pif file for write
   |--- call pif_write_gh            !! write the global header of the output pif file
   |--- call pif_read_dh             !! read the data header of the input pif file
   |--- call pif_write_dh            !! write the data header of the output pif file
   |--- for each node do
        |--- call pif_read_mapi4     !! read 1 slab of pif file in 4-byte
        |--- call pif_read_mapi2     !! read 1 slab of pif file in 2-byte
   |--- call pif_close              !! close input pif file
   |--- cut the slabs
   |--- for each node do
        |--- call pif_write_mapi4    !! read 1 slab of pif file in 4-byte
        |--- call pif_write_mapi2    !! read 1 slab of pif file in 2-byte
   |--- call pif_close              !! close output pif file
```