Appendix

A

Contact is a program and this program is designed to detect whether the docking positions are at a given distance from residues. The source code of this program is described as follow and the usage is wrote following.

Contact [cavity file] [ligand list] [distance]

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

struct liglist{
                    char ligname[40];
                    int    ligatomnum;
                  };

struct ligfile{
                    char atomnum[6];
                    char atomtype[5];
                    char hbtype[1];
                    double x;
                    double y;
                    double z;
                  };

struct cavfile{
                    char type[7];
                    int atomnum;
                    char atomtype[5];
                    char hbtype[1];
                    char res[4];
                    int    resnum;
                    double x;
                    double y;
                    double z;
                  };

struct reslist{
                    char type[7];
                    int atomnum;//for het
                    char res[4];
                    int resnum;//for protein
                    int state;
                  };

//command: contact cav1og5_NEW.pdb docklog.txt        4.5        (5)
```

```
//                          [cavityfile]          [liglist] [D_res] [D_het]
int main (int argc, char *argv[])
{
      struct ligfile lig[100];
      struct reslist reslist[600];
       char cavbuffer[100];
       char ligbuffer[100];
       char listbuffer[100];
       char buffer[7];
       char path[40];//ligand file name
       int count=0;//算 protein file 行數
       int counter=0;//算 liglist 行數
       int ligcount=0;//算 lig 行數
       int rescount=0;//算 protein residue 的數目
       int hetcount=0;//算 protein hetro atom 的數目
       double restrict=0;//跟 protein 的距離限制條件
       double restrict_het=0;//跟 het 的距離限制條件
      double dist=0;//實際距離

       if ( (argv[1] == NULL)||(argv[2] == NULL) )
       {
            printf ( "Please input the cavity file.\n" );
            printf ( "format: contact [cavity file] [distance-protein] ([distance-het])\n" );
            printf ( "please write file names of docking results in docklog.txt\n");
            exit (1);
      }

      restrict = atof(argv[2]);
      restrict = restrict*restrict;
      if ( argv[3] != NULL )
      {
            restrict_het = atof(argv[3]);
            restrict_het = restrict_het*restrict_het;
      }

//清空上一次的  contact-list.txt
      FILE *output;
      output = fopen("contact-list.txt", "w");
      fclose(output);

//read cavity file
      FILE *cavity;
      cavity = fopen (argv[1], "r");
      if (cavity == NULL)
      {
            printf ("Can't open the cavity file %s.\n", argv[1]);
            exit (1);
      }

      while ( ( fgets(cavbuffer, 100, cavity) ) != NULL)//讀 cav 的 ATOM+HETATM 行數
      {
            strncpy (buffer, &cavbuffer[0], 6);
            buffer[6]=0;
            if ( !strncmp(buffer, "ATOM    ", 6) || !strncmp(buffer, "HETATM", 6))
            {
                count++;
            }
```

```
}

struct cavfile cav[count];
count=0;
rewind(cavity);

while ( ( fgets(cavbuffer, 100, cavity) ) != NULL)//把 cav 的資料填到 struct 中
{
      char buff_coor[9];
      char buff_resnum[5];
      strncpy (buffer, &cavbuffer[0], 6);
      buffer[6]=0;

      if ( !strncmp(buffer, "ATOM    ", 6) || !strncmp(buffer, "HETATM", 6))
      {
          char atomnum[6]={0};
          strncpy (cav[count].type, &buffer[0], 7);
          strncpy (atomnum, &cavbuffer[6], 5);
          atomnum[5]=0;
          cav[count].atomnum=atoi(atomnum);
          strncpy (cav[count].atomtype, &cavbuffer[12], 4);
          cav[count].atomtype[4] = 0;
          strncpy (cav[count].res, &cavbuffer[17], 3);
          cav[count].res[3] = 0;

          strncpy (buff_resnum, &cavbuffer[22], 4);
          buff_resnum[4] = 0;
          cav[count].resnum=atoi(buff_resnum);

          strncpy (buff_coor, &cavbuffer[30],8);
          buff_coor[8]=0;
          cav[count].x=atof(buff_coor);
          strncpy (buff_coor, &cavbuffer[38],8);
          buff_coor[8]=0;
          cav[count].y=atof(buff_coor);
          strncpy (buff_coor, &cavbuffer[46],8);
          buff_coor[8]=0;
          cav[count].z=atof(buff_coor);

          if( !strncmp(cav[count].atomtype," CA ",4) ){rescount++;}//找出 res 的數目
          else if(!strncmp(buffer, "HETATM", 6)){hetcount++;}//找出 het 的數目
          else{}
          count++;
          }
}
int j=0;
for(int i=0; i<count; i++)//初始化 reslist
{
      if( !strncmp(cav[i].atomtype, " CA ", 4))
      {
            strncpy (reslist[j].type, cav[i].type, 7);
            strncpy (reslist[j].res, cav[i].res, 4);
            reslist[j].atomnum = cav[i].atomnum;
            reslist[j].resnum = cav[i].resnum;
            reslist[j].state = 0;
            j++;
      }
      else if (!strncmp(cav[i].type, "HETATM", 6))
```

```
            {
                strncpy (reslist[j].type, cav[i].type, 7);
                reslist[j].atomnum = cav[i].atomnum;
                strncpy (reslist[j].res, cav[i].res, 4);
                reslist[j].resnum = cav[i].resnum;
                reslist[j].state = 0;
                j++;
            }
            else{}
        }
//FORMAT:(寫入 contact-list.txt title)
//ligfile_name    ARG_97 GLY_98 ILE_99 PHE100 ...
//mfcd12345678          1          1          1          1 ...
        output = fopen("contact-list.txt", "a");
        fprintf (output,"Set_protein_cavity=%s    #residue=%d    #hetero-atom=%d\n", argv[1],rescount,hetcount);
        fprintf (output,"Set_distance_of_ligand-protein=%4s\n", argv[2]);
        fprintf (output,"Set_distance_of_ligand-hetro=%4s\n", argv[3]);
        fprintf (output, "ligand_file_name                                    ");
        for(int i=0; i<(rescount + hetcount); i++)
        {
            if ( reslist[i].resnum < 10)
            {fprintf (output, " %3s__%1d", reslist[i].res, reslist[i].resnum);}
            else if ( reslist[i].resnum < 100)
            {fprintf (output, " %3s_%2d", reslist[i].res, reslist[i].resnum);}
            else if (!strncmp(reslist[i].type,"HETATM",6))
            {fprintf (output, " %3s%3d", reslist[i].res, reslist[i].atomnum);}
            else
            {fprintf (output, " %3s%3d", reslist[i].res, reslist[i].resnum);}
        }
        fprintf(output, "\n");
        fclose (output);
        fclose(cavity);

//read ligand-list file
        FILE *list;
        list = fopen ("docklog.txt", "r");
        if (list == NULL)
        {
            printf ("Please prepare the ligand file list to docklog.txt\n");
            exit (1);
        }

        while ( ( fgets(listbuffer, 100, list) ) != NULL )//讀出 docklog.txt 行數，設 struct liglist
        {
            if ( !strncmp(&path[0], "pdb", 3) )//compair PDB?
            {counter++;}
            else if ( (listbuffer[16] == 'm')||(listbuffer[16] == 'M') )//Is MFCD?, EX: cav1og5.pdb-MFCD
            {counter++;}
            else if ( (listbuffer[16] == 'e')||(listbuffer[16] == 'E') )//Is EXTREG?, EX: cav1og5.pdb-EXTREG
            {counter++;}
            else //only MDDR number
            {counter++;}
        }
        struct liglist liglist[counter];
        rewind(list);

         FILE *ligand;
//Read PDB docking result
```

```
while ( ( fgets(listbuffer, 100, list) ) != NULL )//Is PDB?
{
	dist=0;
	strncpy(path, &listbuffer[0], 3);//EX: cav1og5.pdb-pdb
	path[3] = '\0';
	if ( !strncmp(&path[0], "pdb", 3) )//compair PDB?
	{
		strncpy(path, &listbuffer[0], 15);//open file
		path[15] = '\0';
		printf("%s\n",path);
		 ligand = fopen (path, "r");
		 if (ligand == NULL)
		 {
			printf ("Can't open the ligand file %s\n", path);
			exit (1);
		}

		ligcount=0;
		while ( ( fgets(ligbuffer, 100, ligand) ) != NULL)//讀出 ligand 檔 ，計算
		{
			strncpy(buffer, &ligbuffer[0], 6);
			buffer[6]=0;
			if ( !strncmp(buffer, "HETATM", 6))
			{
				char buff_coor[9];
				strncpy(lig[ligcount].atomnum, &ligbuffer[6], 5);
				lig[ligcount].atomnum[5]=0;
				strncpy(lig[ligcount].atomtype, &ligbuffer[12],4);
				lig[ligcount].atomtype[4]=0;
				strncpy (buff_coor, &ligbuffer[30],8);
				buff_coor[8]=0;
				lig[ligcount].x=atof(buff_coor);
				strncpy (buff_coor, &ligbuffer[38],8);
				buff_coor[8]=0;
				lig[ligcount].y=atof(buff_coor);
				strncpy (buff_coor, &ligbuffer[46],8);
				buff_coor[8]=0;
				lig[ligcount].z=atof(buff_coor);
				ligcount++;
			}
		}

		fclose(ligand);

		int res_num=0;
		for(int i=0;i<ligcount;i++)/*算距離  */
		{
			for(int j=0;j<count;j++)
			{
				dist = (lig[i].x-cav[j].x)*(lig[i].x-cav[j].x)+
					(lig[i].y-cav[j].y)*(lig[i].y-cav[j].y)+
					(lig[i].z-cav[j].z)*(lig[i].z-cav[j].z);

				if(dist <= restrict)
				{
				    for(int k=0;k<(rescount + hetcount);k++)
				    {
					if ( !strncmp(cav[j].type ,"ATOM    ",6) )
```

```
                                        {
                                            if ( reslist[k].resnum == cav[j].resnum )
                                            {
                                                if ( reslist[k].state==0 )
                                                {
                                                    reslist[k].state=1;
                                                }
                                            }
                                        }
                                        else if (!strncmp(cav[j].type ,"HETATM",6))
                                        {
                                             if ( reslist[k].atomnum == cav[j].atomnum )
                                            {
                                                if ( reslist[k].state==0 )
                                                {
                                                    reslist[k].state=1;
                                                }
                                            }
                                        }
                                        else{}

                                    }
                                }
                            }
                        }
                        FILE *output;
//寫入 contact-list.txt
//ligfile_name arg97 gly98 ...
//mfcd12345678          1          1 ...
                        output = fopen ("contact-list.txt", "a");
                        fprintf (output, "%s    ", path);
                        for(int i=0;i<(rescount + hetcount);i++)
                        {
                            fprintf (output, "%7d", reslist[i].state);
                        }
                        fprintf (output, "\n");
                        fclose (output);

                        for(int i=0;i<(rescount + hetcount);i++)//初始化 reslist
                        {reslist[i].state=0;}
                    }

//Read ACD docking result
                    else if ( (listbuffer[16] == 'm')||(listbuffer[16] == 'M') )//Is MFCD?, EX: cav1og5.pdb-MFCD
                    {
                        dist=0;
                        strncpy (path, &listbuffer[0], 38); //open ACD number file
                        path[38] = '\0';
                        printf("%s\n",path);
                         ligand = fopen (path, "r");
                         if (ligand == NULL)
                         {
                            printf ("Can't open the ligand file %s\n", path);
                            exit (1);
                        }

                        ligcount=0;
                        while ( ( fgets(ligbuffer, 100, ligand) ) != NULL)//讀出 ligand 檔 ，計算
```

```
{
	strncpy(buffer, &ligbuffer[0], 6);
	buffer[6]=0;
	if ( !strncmp(buffer, "HETATM", 6))
	{
		char buff_coor[9];
		strncpy(lig[ligcount].atomnum, &ligbuffer[6], 5);
		lig[ligcount].atomnum[5]=0;
		strncpy(lig[ligcount].atomtype, &ligbuffer[12],4);
		lig[ligcount].atomtype[4]=0;
		strncpy (buff_coor, &ligbuffer[30],8);
		buff_coor[8]=0;
		lig[ligcount].x=atof(buff_coor);
		strncpy (buff_coor, &ligbuffer[38],8);
		buff_coor[8]=0;
		lig[ligcount].y=atof(buff_coor);
		strncpy (buff_coor, &ligbuffer[46],8);
		buff_coor[8]=0;
		lig[ligcount].z=atof(buff_coor);
		ligcount++;
	}
}

fclose(ligand);

int res_num=0;
for(int i=0;i<ligcount;i++)/*算距離  */
{
	for(int j=0;j<count;j++)
	{
		dist = (lig[i].x-cav[j].x)*(lig[i].x-cav[j].x)+
			(lig[i].y-cav[j].y)*(lig[i].y-cav[j].y)+
			(lig[i].z-cav[j].z)*(lig[i].z-cav[j].z);

		if(dist <= restrict)
		{
		    for(int k=0;k<(rescount + hetcount);k++)
		    {
			if ( !strncmp(cav[j].type ,"ATOM    ",6) )
			{
			    if ( reslist[k].resnum==cav[j].resnum )
			    {
				if ( reslist[k].state==0 )
				{
					reslist[k].state=1;
				}
			    }
			}
			else if (!strncmp(cav[j].type ,"HETATM",6))
			{
			     if ( reslist[k].atomnum==cav[j].atomnum )
			    {
				if ( reslist[k].state==0 )
				{
					reslist[k].state=1;
				}
			    }
			}
```

```
                                else{}

                            }
                        }
                    }
                }
                FILE *output;
//寫入 contact-list.txt
//ligfile_name arg97 gly98 ...
//mfcd12345678          1          1 ...
                output = fopen ("contact-list.txt", "a");
                fprintf (output, "%s ", path);
                for(int i=0;i<(rescount + hetcount);i++)
                {
                    fprintf (output, "%7d", reslist[i].state);
                }
                fprintf (output, "\n");
                fclose (output);

                for(int i=0;i<(rescount + hetcount);i++)//初始化 reslist
                {reslist[i].state=0;}
            }

//Read MDDR docking result
            else if ( (listbuffer[16] == 'e')||(listbuffer[16] == 'E') )//Is EXTREG?, EX: cav1og5.pdb-EXTREG
            {
                dist=0;
                strncpy (path, &listbuffer[0], 38); //open MDDR number file
                path[38] = '\0';
                printf("%s\n",path);
                 ligand = fopen (path, "r");
                 if (ligand == NULL)
                 {
                    printf ("Can't open the ligand file %s\n", path);
                    exit (1);
                }

                ligcount=0;
                while ( ( fgets(ligbuffer, 100, ligand) ) != NULL)//讀出 ligand 檔 ，計算
                {
                    strncpy(buffer, &ligbuffer[0], 6);
                    buffer[6]=0;
                    if ( !strncmp(buffer, "HETATM", 6))
                    {
                        char buff_coor[9];
                        strncpy(lig[ligcount].atomnum, &ligbuffer[6], 5);
                        lig[ligcount].atomnum[5]=0;
                        strncpy(lig[ligcount].atomtype, &ligbuffer[12],4);
                        lig[ligcount].atomtype[4]=0;
                        strncpy (buff_coor, &ligbuffer[30],8);
                        buff_coor[8]=0;
                        lig[ligcount].x=atof(buff_coor);
                        strncpy (buff_coor, &ligbuffer[38],8);
                        buff_coor[8]=0;
                        lig[ligcount].y=atof(buff_coor);
                        strncpy (buff_coor, &ligbuffer[46],8);
                        buff_coor[8]=0;
                        lig[ligcount].z=atof(buff_coor);
```

```
                ligcount++;
            }
        }

        fclose(ligand);

        int res_num=0;
        for(int i=0;i<ligcount;i++)/*算距離  */
        {
            for(int j=0;j<count;j++)
            {
                dist = (lig[i].x-cav[j].x)*(lig[i].x-cav[j].x)+
                       (lig[i].y-cav[j].y)*(lig[i].y-cav[j].y)+
                       (lig[i].z-cav[j].z)*(lig[i].z-cav[j].z);

                if(dist <= restrict)
                {
                    for(int k=0;k<(rescount + hetcount);k++)
                    {
                        if ( !strncmp(cav[j].type ,"ATOM    ",6) )
                        {
                            if ( reslist[k].resnum==cav[j].resnum )
                            {
                                if ( reslist[k].state==0 )
                                {
                                    reslist[k].state=1;
                                }
                            }
                        }
                        else if (!strncmp(cav[j].type ,"HETATM",6))
                        {
                            if ( reslist[k].atomnum==cav[j].atomnum )
                            {
                                if ( reslist[k].state==0 )
                                {
                                    reslist[k].state=1;
                                }
                            }
                        }
                        else{}

                    }
                }
            }
        }
        FILE *output;
//寫入 contact-list.txt
//ligfile_name arg97 gly98 ...
//mfcd12345678          1          1 ...
        output = fopen ("contact-list.txt", "a");
        fprintf (output, "%s ", path);
        for(int i=0;i<(rescount + hetcount);i++)
        {
            fprintf (output, "%7d", reslist[i].state);
        }
        fprintf (output, "\n");
        fclose (output);
```

```
            for(int i=0;i<(rescount + hetcount);i++)//初始化 reslist
            {reslist[i].state=0;}
        }

//Read MDDR (only number) file
        else   //MDDR
        {
            dist=0;
            strncpy (path, &listbuffer[0], 28); // MDDR number
            path[28] = '\0';
            printf("%s\n",path);
             ligand = fopen (path, "r");
             if (ligand == NULL)
             {
                printf ("Can't open the ligand file %s\n", path);
                exit (1);
            }

            ligcount=0;
            while ( ( fgets(ligbuffer, 100, ligand) ) != NULL)//讀出 ligand 檔 ，計算
            {
                strncpy(buffer, &ligbuffer[0], 6);
                buffer[6]=0;
                if ( !strncmp(buffer, "HETATM", 6))
                {
                    char buff_coor[9];
                    strncpy(lig[ligcount].atomnum, &ligbuffer[6], 5);
                    lig[ligcount].atomnum[5]=0;
                    strncpy(lig[ligcount].atomtype, &ligbuffer[12],4);
                    lig[ligcount].atomtype[4]=0;
                    strncpy (buff_coor, &ligbuffer[30],8);
                    buff_coor[8]=0;
                    lig[ligcount].x=atof(buff_coor);
                    strncpy (buff_coor, &ligbuffer[38],8);
                    buff_coor[8]=0;
                    lig[ligcount].y=atof(buff_coor);
                    strncpy (buff_coor, &ligbuffer[46],8);
                    buff_coor[8]=0;
                    lig[ligcount].z=atof(buff_coor);
                    ligcount++;
                }
            }

            fclose(ligand);

            int res_num=0;
            for(int i=0;i<ligcount;i++)/*算距離  */
            {
                for(int j=0;j<count;j++)
                {
                    dist = (lig[i].x-cav[j].x)*(lig[i].x-cav[j].x)+
                           (lig[i].y-cav[j].y)*(lig[i].y-cav[j].y)+
                           (lig[i].z-cav[j].z)*(lig[i].z-cav[j].z);

                    if(dist <= restrict)
                    {
                        for(int k=0;k<(rescount + hetcount);k++)
                        {
```

```
                                        if ( !strncmp(cav[j].type ,"ATOM    ",6) )
                                        {
                                            if ( reslist[k].resnum==cav[j].resnum )
                                            {
                                                if ( reslist[k].state==0 )
                                                {
                                                    reslist[k].state=1;
                                                }
                                            }
                                        }
                                        else if (!strncmp(cav[j].type ,"HETATM",6))
                                        {
                                             if ( reslist[k].atomnum==cav[j].atomnum )
                                            {
                                                if ( reslist[k].state==0 )
                                                {
                                                    reslist[k].state=1;
                                                }
                                            }
                                        }
                                        else{}

                                    }
                                }
                            }
                        }
                        FILE *output;
//寫入 contact-list.txt
//ligfile_name arg97 gly98 ...
//mfcd12345678          1          1 ...
                        output = fopen ("contact-list.txt", "a");
                        fprintf (output, "%s ", path);
                        for(int i=0;i<(rescount + hetcount);i++)
                        {
                            fprintf (output, "%7d", reslist[i].state);
                        }
                        fprintf (output, "\n");
                        fclose (output);

                        for(int i=0;i<(rescount + hetcount);i++)//初始化 reslist
                        {reslist[i].state=0;}
                    }

                }
            fclose (list);
            return 0;
}
```