

# Εισαγωγή στην Υπολογιστική Βιολογία

Διάλεξη 10η :

Εφαρμογές της C : Δομική Βιολογία

# Δομική βιολογία: ορισμός

---

Ο κλάδος της βιολογίας ο οποίος ασχολείται με τον προσδιορισμό και την ανάλυση των δομών βιολογικών μακρομορίων και των συμπλόκων τους.

# Δομή μορίου: ορισμοί

---

Αυστηρός ορισμός :

Δομή ενός μορίου είναι η σχετική διευθέτηση στον τρισδιάστατο χώρο των ατόμων από τα οποία αποτελείται το μόριο.

Με άλλα λόγια, γνώση της δομής ενός μορίου σημαίνει ότι γνωρίζουμε που βρίσκεται κάθε άτομο του μορίου σε σχέση με όλα τα άλλα άτομα.

# Protein Data Bank (PDB)

---

Είναι η μοναδική (πρωτοταγής) βάση δεδομένων της δομικής βιολογίας. Στις επόμενες διαφάνειες θα παρουσιαστεί η μορφή μιας καταχώρησης της. Η διεύθυνση της βάσης είναι :

<http://www.pdb.org/>

# PDB : παράδειγμα

---

```
HEADER      TRANSCRIPTION REGULATION      16-JAN-99      1B6Q
TITLE      ALANINE 31 PROLINE MUTANT OF ROP PROTEIN
COMPND     MOL_ID: 1;
COMPND     2 MOLECULE: ROP;
COMPND     3 CHAIN: NULL;
COMPND     4 SYNONYM: ROM;
COMPND     5 ENGINEERED: YES;
COMPND     6 MUTATION: A31P;
COMPND     7 BIOLOGICAL_UNIT: HOMODIMER
SOURCE     MOL_ID: 1;
SOURCE     2 ORGANISM_SCIENTIFIC: ESCHERICHIA COLI;
SOURCE     3 CELLULAR_LOCATION: CYTOPLASM;
SOURCE     4 PLASMID: COLE1;
SOURCE     5 GENE: ROP;
SOURCE     6 EXPRESSION_SYSTEM: ESCHERICHIA COLI;
SOURCE     7 EXPRESSION_SYSTEM_STRAIN: 71/72 (71/18 PLUS PCI857);
SOURCE     8 EXPRESSION_SYSTEM_PLASMID: PEX43
```

# PDB : παράδειγμα

```
KEYWDS      TRANSCRIPTION REGULATION
EXPDTA      X-RAY DIFFRACTION
AUTHOR      N.GLYKOS,G.CESARENI,M.KOKKINIDIS
REVDAT      1   09-JUL-99 1B6Q   0
JRNL        AUTH   N.M.GLYKOS,G.CESARENI,M.KOKKINIDIS
JRNL        TITL   PROTEIN PLASTICITY TO THE EXTREME: CHANGING THE
JRNL        TITL 2 TOPOLOGY OF A 4-ALPHA-HELICAL BUNDLE WITH A SINGLE
JRNL        TITL 3 AMINO-ACID SUBSTITUTION
JRNL        REF    STRUCTURE (LONDON)          V.   7   597 1999
JRNL        REFN   ASTM STRUE6  UK ISSN 0969-2126          2005
REMARK      1
REMARK      1 REFERENCE 1
REMARK      1 TITL  MEANINGFUL REFINEMENT OF POLY-ALANINE MODELS USING
REMARK      1 TITL 2 RIGID-BODY SIMULATED ANNEALING : APPLICATION TO
REMARK      1 TITL 3 THE STRUCTURE DETERMINATION OF THE A31P ROP MUTANT
REMARK      1 REF    ACTA CRYSTALLOGR., SECT.D          V.  55  1301 1999
REMARK      1 REFN   ASTM ABCRE6  DK ISSN 0907-4449          0766
```

# PDB : παράδειγμα

```
REMARK 2 RESOLUTION. 1.80 ANGSTROMS.
REMARK 3
REMARK 3 REFINEMENT.
REMARK 3 PROGRAM : X-PLOR 3.851
REMARK 3 AUTHORS : BRUNGER
REMARK 3
REMARK 3 DATA USED IN REFINEMENT.
REMARK 3 RESOLUTION RANGE HIGH (ANGSTROMS) : 1.80
REMARK 3 RESOLUTION RANGE LOW (ANGSTROMS) : 40.825
REMARK 3 DATA CUTOFF (SIGMA(F)) : 0.0
REMARK 3 DATA CUTOFF HIGH (ABS(F)) : NULL
REMARK 3 DATA CUTOFF LOW (ABS(F)) : NULL
REMARK 3 COMPLETENESS (WORKING+TEST) (%) : 99.9
REMARK 3 NUMBER OF REFLECTIONS : 5103
REMARK 3
REMARK 3 FIT TO DATA USED IN REFINEMENT.
REMARK 3 CROSS-VALIDATION METHOD : THROUGHOUT
REMARK 3 FREE R VALUE TEST SET SELECTION : RANDOM
REMARK 3 R VALUE (WORKING SET) : 0.189
REMARK 3 FREE R VALUE : 0.240
REMARK 3 FREE R VALUE TEST SET SIZE (%) : 5.0
REMARK 3 FREE R VALUE TEST SET COUNT : 286
```

# PDB : παράδειγμα

---

```
REMARK 200 EXPERIMENTAL DETAILS
REMARK 200 EXPERIMENT TYPE           : X-RAY DIFFRACTION
REMARK 200 DATE OF DATA COLLECTION  : NULL
REMARK 200 TEMPERATURE                (KELVIN) : 293
REMARK 200 PH                          : 4.3
REMARK 200 NUMBER OF CRYSTALS USED    : 1
REMARK 200
REMARK 200 SYNCHROTRON                 (Y/N) : N
REMARK 200 RADIATION SOURCE            : NULL
REMARK 200 BEAMLINE                    : NULL
REMARK 200 X-RAY GENERATOR MODEL       : SEALED TUBE
REMARK 200 MONOCHROMATIC OR LAUE       (M/L) : M
REMARK 200 WAVELENGTH OR RANGE         (A)  : 1.5418
REMARK 200 MONOCHROMATOR               : GRAPHITE MONOCHROMATOR
REMARK 200 OPTICS                      : MONOCHROMATOR
REMARK 200
```



# PDB : παράδειγμα

```
REMARK 999 SEQUENCE
REMARK 999 1B6Q          SWS      P03051      57 -      63 NOT IN ATOMS LIST
DBREF  1B6Q           1      56 SWS      P03051      ROP_ECOLI           1      56
SEQADV 1B6Q PRO           31 SWS      P03051      ALA           31 ENGINEERED MUTATION
SEQRES  1           63 MET THR LYS GLN GLU LYS THR ALA LEU ASN MET ALA ARG
SEQRES  2           63 PHE ILE ARG SER GLN THR LEU THR LEU LEU GLU LYS LEU
SEQRES  3           63 ASN GLU LEU ASP PRO ASP GLU GLN ALA ASP ILE CYS GLU
SEQRES  4           63 SER LEU HIS ASP HIS ALA ASP GLU LEU TYR ARG SER CYS
SEQRES  5           63 LEU ALA ARG PHE GLY ASP ASP GLY GLU ASN LEU
FORMUL  2 HOH      *56(H2 O1)
HELIX   1   1 LYS           3 GLU           28 1                               26
HELIX   2   2 PRO           31 ALA           54 1                               24
CRYST1  30.400  42.100  81.400  90.00  90.00  90.00 C 2 2 21           8
ORIGX1  1.000000  0.000000  0.000000           0.000000
ORIGX2  0.000000  1.000000  0.000000           0.000000
ORIGX3  0.000000  0.000000  1.000000           0.000000
SCALE1  0.032895  0.000000  0.000000           0.000000
SCALE2  0.000000  0.023753  0.000000           0.000000
SCALE3  0.000000  0.000000  0.012285           0.000000
```

# PDB : παράδειγμα

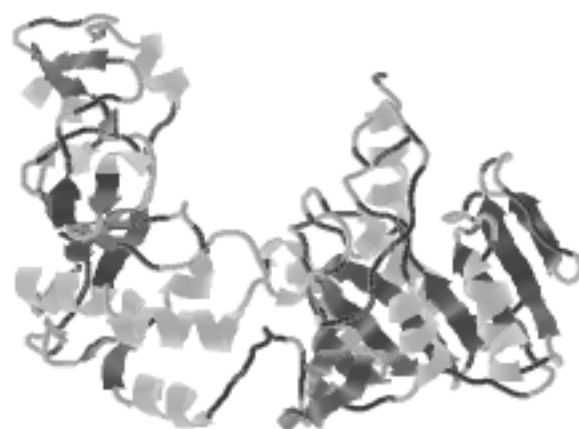
---

ATOM	1	N	MET	1	-2.053	13.510	-6.199	1.00	47.14
ATOM	2	CA	MET	1	-1.894	13.110	-4.767	1.00	49.45
ATOM	3	C	MET	1	-0.688	12.186	-4.582	1.00	47.94
ATOM	4	O	MET	1	-0.774	10.982	-4.841	1.00	51.90
ATOM	5	CB	MET	1	-3.163	12.402	-4.276	1.00	51.84
ATOM	6	CG	MET	1	-3.059	11.814	-2.871	1.00	57.35
ATOM	7	SD	MET	1	-4.121	12.669	-1.683	1.00	62.43
ATOM	8	CE	MET	1	-2.938	13.041	-0.373	1.00	61.79
ATOM	9	N	THR	2	0.434	12.748	-4.134	1.00	42.11
ATOM	10	CA	THR	2	1.636	11.954	-3.914	1.00	34.48
ATOM	11	C	THR	2	1.551	11.168	-2.615	1.00	32.97
ATOM	12	O	THR	2	0.726	11.447	-1.749	1.00	32.39

# Το πρόβλημα : επικράτειες

---

Οι πρωτεΐνες είναι συνηθέστατα οργανωμένες σε επικράτειες (domains), δηλ. σε δομικά (και ίσως λειτουργικά) διακριτές περιοχές. Σε πολλές περιπτώσεις η διάκριση μεταξύ των domains είναι οφθαλμοφανής, π.χ.



# Το πρόβλημα : επικράτειες

---

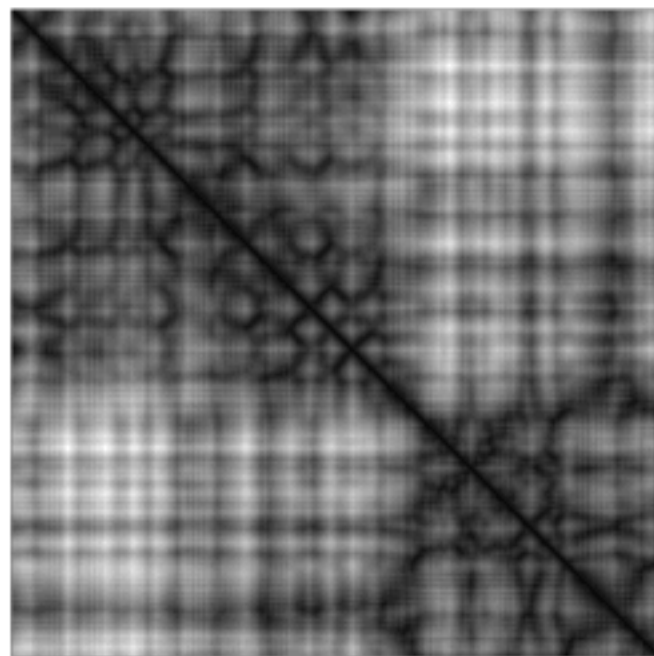
... ενώ σε άλλες όχι, π.χ.



# Το πρόβλημα : επικράτειες

---

Μία απλή μέθοδος για τη διάκριση επικρατειών είναι τα Ca-Ca distance maps :



# Το πρόβλημα :

---

Αυτό που θέλουμε είναι ένα πρόγραμμα σε C το οποίο να διαβάζει (από την καθιερωμένη είσοδο) ένα PDB αρχείο για μια πρωτεΐνη, και να γράφει στην καθιερωμένη έξοδο τις αποστάσεις (σε Angstrom) μεταξύ όλων των ζευγών ατομών Ca της πρωτεΐνης. Δηλ.

# Είσοδος :

---

ATOM	1	N	LEU	12	-23.473	-11.736	-9.882	1.00	65.95	N
ATOM	2	CA	LEU	12	-23.078	-12.360	-8.620	1.00	65.23	C
ATOM	3	C	LEU	12	-21.709	-13.037	-8.739	1.00	64.24	C
ATOM	4	O	LEU	12	-21.296	-13.759	-7.818	1.00	65.26	O
ATOM	5	CB	LEU	12	-24.085	-13.425	-8.194	1.00	66.24	C
ATOM	6	CG	LEU	12	-25.506	-13.030	-7.805	1.00	67.06	C
ATOM	7	CD1	LEU	12	-26.538	-13.970	-8.439	1.00	66.68	C
ATOM	8	CD2	LEU	12	-25.685	-13.060	-6.286	1.00	66.70	C
ATOM	9	N	ALA	13	-21.026	-12.860	-9.866	1.00	60.78	N
ATOM	10	CA	ALA	13	-19.736	-13.518	-10.047	1.00	58.43	C

.....

# Έξοδος :

---

0 0 0.0000000

0 1 3.8139548

0 2 6.2237191

0 3 9.8304338

.....

0 163 40.3271141

0 164 43.9487190

1 0 3.8139548

1 1 0.0000000

1 2 3.8132520

.....

164 163 3.7846403

164 164 0.0000000



# Το πρόγραμμα :

---

Η κεντρική ιδέα είναι απλή : διαβάζουμε από την είσοδο συντεταγμένες x,y,z για κάθε άτομο του PDB αρχείου και εάν το άτομο είναι "CA", αποθηκεύουμε τις συντεταγμένες του σε διαδοχικές θέσεις τριών πινάκων (ένα για τις x συντεταγμένες, ένα για τις y, και ένα για τις z). Δεδομένου ότι η απόσταση μεταξύ δύο ατόμων είναι

$$D = \text{sqrt}[(x_1-x_2)^2 + (y_1-y_2)^2 + (z_1-z_2)^2]$$

το μόνο που μένει να υπολογίσουμε τα D μέσω δυο ένθετων for().

# Ο καθ'αυτός υπολογισμός :

---

```
for ( i=0 ; i < tot_atoms ; i++)
  for ( m=0 ; m < tot_atoms ; m++ )
  {
    distance = sqrt( (x[i]-x[m])*(x[i]-x[m]) +
                     (y[i]-y[m])*(y[i]-y[m]) +
                     (z[i]-z[m])*(z[i]-z[m]) );
    printf("%5d %5d %15.7f\n", i, m, distance );
  }
}
```

# Ανάγνωση δεδομένων :

---

... και προβλήματα της.

```
tot_atoms = 0;

while( scanf("%s %d %s %s %d %f %f %f %f %f %s", &s, &atid,
&atname, &resname, &resid, &xx, &yy, &zz, &j, &j, &s) != EOF)
{
    if ( atname[0] == 'C' && atname[1] == 'A' )
    {
        x[tot_atoms] = xx;
        y[tot_atoms] = yy;
        z[tot_atoms] = zz;
        tot_atoms++;
    }
}
```

# Το πρόγραμμα :

---

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

```
main()
{
char s[20];
int atid;
char atname[5];
char resname[5];
int resid;
float xx, yy, zz;
float j;
float x[5000];
float y[5000];
float z[5000];
float distance;
int tot_atoms;
int i, m;
```

# Το πρόγραμμα :

```
tot_atoms = 0;
while( scanf("%s %d %s %s %d %f %f %f %f %f %s", &s, &atid,
&atname, &resname, &resid, &xx, &yy, &zz, &j, &j, &s) != EOF)
{
    if ( atname[0] == 'C' && atname[1] == 'A' )
    {
        x[tot_atoms] = xx;
        y[tot_atoms] = yy;
        z[tot_atoms] = zz;
        tot_atoms++;
    }
}

for ( i=0 ; i < tot_atoms ; i++)
    for ( m=0 ; m < tot_atoms ; m++)
    {
        distance = sqrt( (x[i]-x[m])*(x[i]-x[m]) +
            (y[i]-y[m])*(y[i]-y[m]) +
            (z[i]-z[m])*(z[i]-z[m]) );
        printf("%5d %5d %15.7f\n", i, m, distance );
    }
}
```

# Επίδειξη χρήσης ...



```

#include <stdio.h>

main()
{

char    s[1000];
char    t[10000];
int     i, k, tot;

if ( scanf("%s", s ) != 1 )
    exit(1);

tot = 0;
while ( scanf("%s", t ) != EOF )
    {
        i = 0;
        while( t[i] != 0 )
            {
                k = 0;
                while( s[k] == t[i+k] && s[k] != 0 && t[i+k] != 0 )
                    k++;

                if ( s[k] == 0 )
                    printf(" ----> %5d\n", tot );

                tot++;
                i++;
            }
    }
}

```