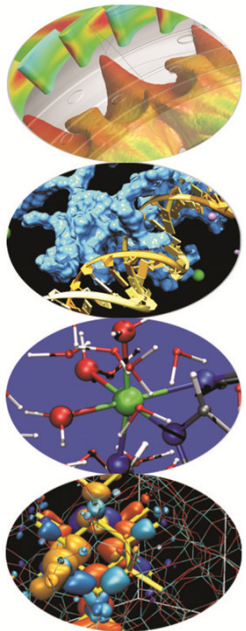
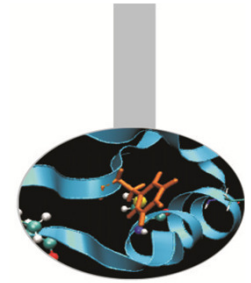


# Librerie matematiche *esercitazioni*

M.Cremonesi, *CINECA*

19 febbraio 2014

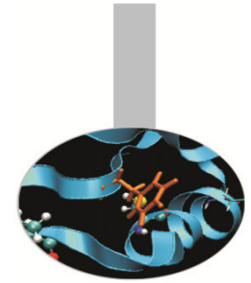




# Installazione

Come primo esercizio utile proviamo a installare una libreria:

- Scaricare la libreria da <http://www.netlib.org/lapack/lapack-3.4.1.tgz>
- Starare in una cartella CorsoHPCtools: `tar -z -xf <libreria>`
- Configurare `make.inc` (basta copiare `make.inc.example`)
- `cd SRC`
- `make`
- `cd ../lapacke`
- `make`
- `cd ../BLAS/SRC`
- `make`

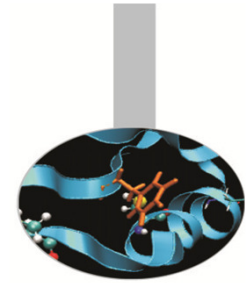


# Moltiplicazione di matrici

È abbastanza facile, benché inefficiente, implementare la moltiplicazione riga-colonna di matrici. Si provi perciò a confrontare le prestazioni di un implementazione manuale con l'uso della funzione BLAS seguente:

```
FUNCTION DDOT(N,X,INCX,Y,INCY) RESULT(XtY)
  REAL(8) :: XtY  ! XtY = Sommatoria(X(i)*Y(i), i=1,N)
  INTEGER, INTENT(IN) :: N, INCX, INCY
  REAL(8), DIMENSION(:), INTENT(IN) :: X,Y
END FUNCTION DDOT
```

Source code: MatrMult



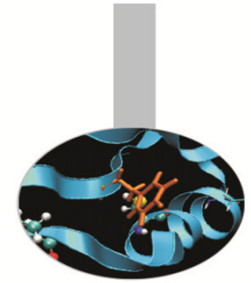
# Moltiplicazione di matrici

Dopo aver inserito la chiamata alla funzione DDOT compilare con:

```
export LIB_ROOT=./lapack-3.4.1
```

```
gfortran -o MatrMult.exe MatrMult.f90 -L$LIB_ROOT -lrefblas
```

```
gcc -o MatrMult.exe MatrMult.c -L$LIB_ROOT -lrefblas
```



# Moltiplicazione di matrici

Un'alternativa valida è rappresentata dall'uso della funzione DGEMM:

$$C(1:M,1:N) = \text{Beta} * C(1:M,1:N) + \text{MatrMult}(\text{Alpha} * A(1:M,1:K), B(1:K,1:N))$$

```
SUBROUTINE DGEMM(TRANSA,TRANSB,M,N,K,ALPHA, &  
&                A,LDA,B,LDB,BETA,C,LDC)
```

```
  CHARACTER(1), INTENT(IN) :: TRANSA, TRANSB
```

```
  INTEGER, INTENT(IN) :: LDA, LDB, LDC, M, K, N
```

```
  REAL(8), INTENT(IN) :: A(LDA,K), B(LDB,N)
```

```
  REAL(8), INTENT(IN OUT) :: C(LDC,N)
```

```
  REAL(8), ALPHA, BETA
```

```
END SUBROUTINE DGEMM
```

Source code: MatrMult1