MMMx: In silico spin labeling, distance distributions, preparing constructs

G. Jeschke

ETH Zürich, Dep. Chemistry & Applied Biosciences

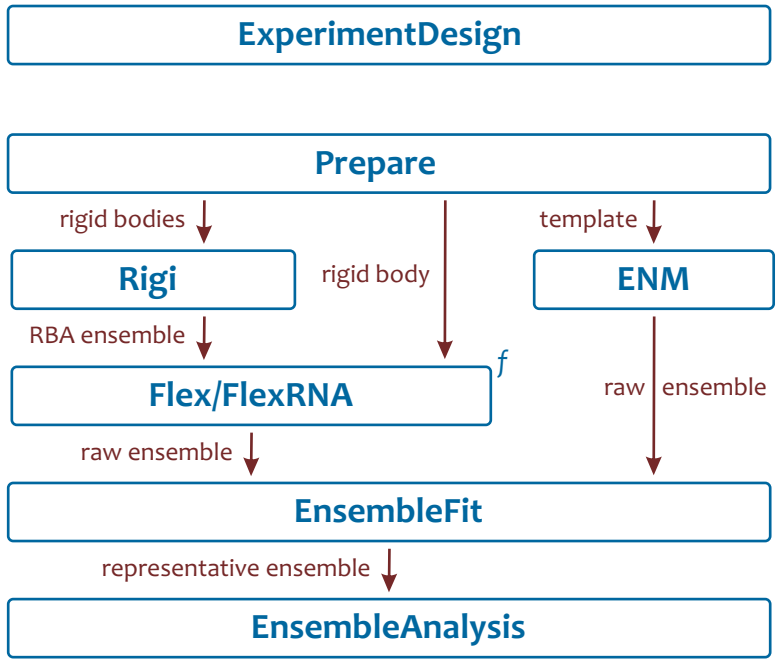
General concept of MMMx

As a Matlab toolbox

- collection of Matlab functions for handling of protein and nucleic acid structures and for integrative structural biology
- (ensemble) structures are stored in an internal format (`entity`)
- the entity format automatically stores spin label rotamers, when they are computed

As a script language for modelling

- typical modelling tasks do not require Matlab programming
- script language is documented at **mmmx.info**
- access is via *modules* that have only a small number of *keywords*
- all modules, except for ExperimentDesign, can be part of a *modelling pipeline*
- modules exchange models (mainly) in PDB format
- steps in a modeling pipeline can also be performed with external software
- MMMx scripts can implement a whole pipeline or only parts of it



Loading a structure

MMMx structures are ensembles of *C* conformers

- a single *conformer* (single MODEL in PDB) is a special case (ensemble with a single member)
- conformers have *populations*, which add to 1; the case of uniform populations $p_c = 1/C$ is a special case
- MMMx stores populations in `REMARK 400` records of PDB files and can read them
- the native ensemble format of MMMx is a list of PDB file names of individual conformers with associated populations

Four ways of constructing an ensemble in the ExperimentDesign module

`import` reads a whole ensemble with uniform populations from a single PDB file (use for NMR structures)

`addpdb` construct an ensemble from a PDB file name with wildcards (uniform populations)

☝ all files must correspond to the same primary structure (number and type of chains, residues, and atoms)

`getens` construct an ensemble from an MMMx ensemble list, including populations

`expand` construct an ensemble from rigid-body arrangement output of the Rigi module

Module-internal variable names

- upon loading or constructing an ensemble, it is assigned and internal *identifier* that acts like a variable name in a programming language

Computation of a few distance distributions for T4 lysozyme

```
% MMMx Demo: Computation of selected distance distributions for T4 Lysozyme

# log % opens a log file with the same name as the control file

!ExperimentDesign % starts the ExperimentDesign module
   import 2LZM T4L % loads the T4 lysozyme crystal structure from the PDB and assigns identifier T4L
   sitescan mtsl T4L T4L_sites % site scan with label MTSL, default set CILMSTV of residues
   pairlist T4L_sites T4L T4L_pairs 15 80 % list of all pairs of sites for T4 lysozyme
   plot T4L_distribution pdf % specifies a basis file name and the graphics format PDF for plots
   distributions mtsl T4L T4L_distribution % block key, computes selected distributions
       68   131 % address line, here we have a single chain and two residue numbers suffice
       72   131
       75   131
       76   131
       79   131
       82   131
       86   131
       89   131
      109   131
   .distributions % block keys must be explicitly closed
.ExperimentDesign % the module must be explicitly closed

# report % open the log file in the system editor (in Windows usually Notepad)
```

Distance distributions for orthogonal spin labelling

```
% MMMx Demo: Computation of distributions for orthogonal spin labelling on protein and RNA

# log

!ExperimentDesign
   import 2ADC RRM % load NMR structure of RBD34 of PTBP1 complexed with RNA CUCUCU
   sitescan mtsl RRM RRM_sites I % sitescan for the protein part of the RRM, only isoleucine
   sitescan iap-4tu RRM SL_sites u 3 0.25 % sitescan (only uracil) for the RNA part, at least 3
                                          % rotamers, partition function of at least 0.25=
   hetpairlist RRM_sites SL_sites RRM RRM_pairs 15 80 % pair list with minimum distance of 15 Å and
                                                      % maximum distance of 80 Å
   plot RRM_distribution pdf % specifies a basis file name and the graphics format PDF for plots
   distributions mtsl|iap-4tu RRM RRM_distribution % block key for orthogonal labelling
      RRM_pairs % instead of site pair addresses, we specify a complete pair list
   .distributions % close block key
.ExperimentDesign % close module

# report
```

- in the T4 Lysozyme example, distributions could have been computed without the previous site scan and pair list generation
- in this example, the pair list is used and must be generated first

Importing an MMMx ensemble and computing a trivariate distribution

```
% MMMx Demo: Computing a trivariate distance distribution from an MMMx generated ensemble

# log

!ExperimentDesign

   getens SRSF1_DEER_PREratio.ens SRSF1_16_107_RNA12 % load SRSF1 in complex with RNA12 ensemble

   trivariate mtsl|mtsl|r5p SRSF1_16_107_RNA12 % MTSL on the first two sites, r5p on the third site
       (A)16   (A)107   (C)1 % addressing including a chain identifier
   .trivariate % close block key

.ExperimentDesign % close module

# report
```

- because the ensemble is large, this example takes rather long to compute
- this example demonstrates computation of a distribution without previous site scan

Merging a rigid-body template from several PDB structures

```
% MMMx Demo: Building a rigid-body template from several NMR structures
% and searching for optimum reference sites

# log

!prepare
   getpdb 2AD9 RRM1 % a single NMR structure that specifies RRM1 and an RNA binding motif
   getpdb 2ADB RRM2 % a single NMR structure that specifies RRM2 and an RNA binding motif
   getpdb 2ADC RRM34 % a single NMR structure that specifies RRM3/4 and two RNA binding motifs
   merge PTBP1 % generate the new entity with identifier 'PTBP1' from subsets of residues/nucleotides
      RRM1 {11}(A)58-146 % will be new chain A, model 11 of the NMR structure is used
      RRM1 {11}(B)147-152 % will be new chain B, model 11 of the NMR structure is used
      RRM2 {13}(A)182-283 % will be new chain C, model 13 of the NMR structure is used
      RRM2 {13}(B)299-304 % will be new chain D, model 13 of the NMR structure is used
       RRM34 {3}(A)337-531 % will be new chain E, model 3 of the NMR structure is used
       RRM34 {3}(B)532-537 % will be new chain F, model 3 of the NMR structure is used
       RRM34 {3}(C)538-543 % will be new chain G, model 3 of the NMR structure is used
   .merge % close block key
   save PTBP1_EMCV_IRES_rb PTBP1 PTB1 % save chimera generated by merging
.prepare % close module
```

- if necessary, residues can be renumbered by the `prepare` module

Searching for optimum reference points in the rigid bodies

```
!ExperimentDesign
   addpdb PTBP1_EMCV_IRES_rb PTBP1 % load complete rigid-body definition
                                   % generated by the previous module
   sitescan mtsl PTBP1 RB1_sites CILMSTVAN 5 0.1 A % >=5 rotamers, partition function >=0.1,
                                                   % only chain A
   sitescan mtsl PTBP1 RB2_sites CILMSTVAN 5 0.1 C % >=5 rotamers, partition function >=0.1,
                                                   % only chain C
   sitescan mtsl PTBP1 RB34_sites CILMSTVAN 5 0.1 E % >=5 rotamers, partition function >=0.1,
                                                    % only chain A
   rbReference PTBP1 15 80 RB1_sites % generate reference sites for first rigid body
   rbReference PTBP1 15 80 RB2_sites % generate reference sites for second rigid body
   rbReference PTBP1 15 80 RB34_sites % generate reference sites for third rigid body
.ExperimentDesign % close module

# report
```

- command `rbReference` reads the site lists from a file (and command `sitescan` stores them as files)
- thus, you can use `rbReference` with manually edited site lists, for instance, after deleting mutants that did not work out or that you do not like after visual inspection