



---

## Modeling solution conformation of a non-coding RNA molecule using PyRy3D web-server.

During this exercise you will learn how to build a 3D model of an RNA molecule based on secondary structure restraints and Small Angle X-ray Scattering (SAXS) shape reconstruction. The target molecule is an Adenovirus Virus Associated RNA-I (VA<sub>1</sub>ΔTS) which model was recently published at:

Dzananovic E, Patel TR, **Chojnowski G**, Boniecki MJ, Deo S, McEleney K, Harding SE, Bujnicki JM, McKenna SA, *Solution conformation of Adenovirus Virus Associated RNA-I and its interaction with PKR*, J Struct Biol 2014 Jan;185(1):48-57.

This exercise is based on a simplified version of the original modeling procedure. Owing that you will be able to obtain a final model in less than 2 hours using popular modeling tools and web-servers. The exercise concise the following steps:

### Tools

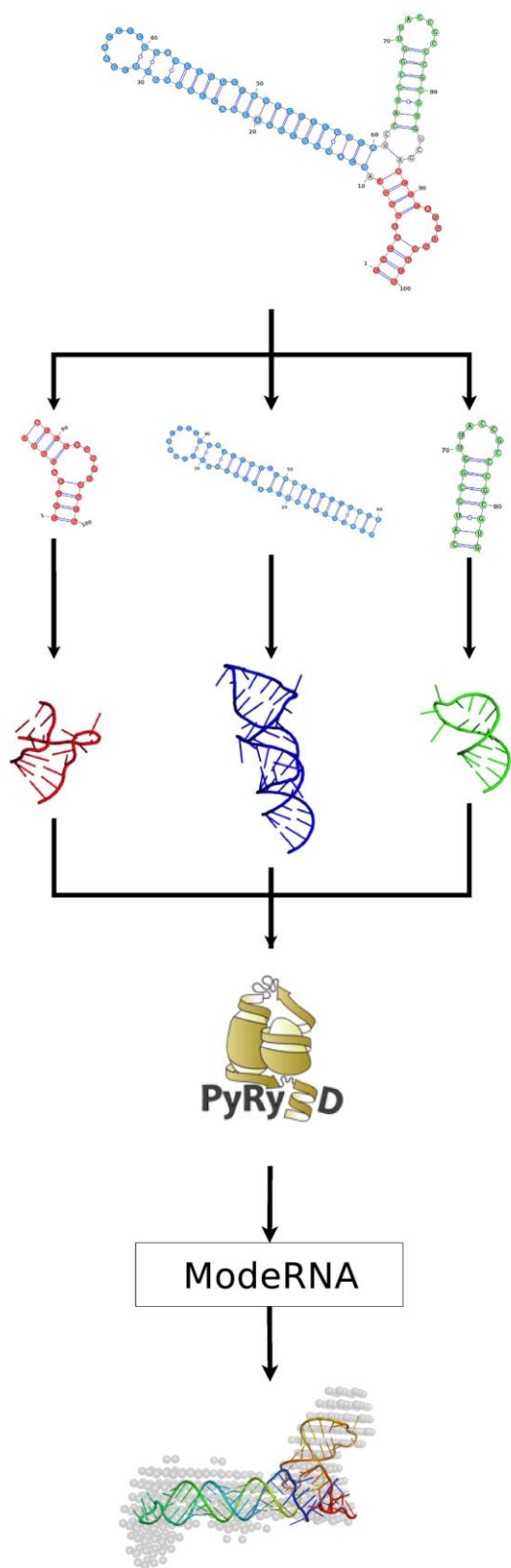
1. Integrative modeling using PyRy3D web interface:  
<http://pyry3d.icm.edu.pl/>
2. 3D structure prediction of RNA with ModeRNA webserver:  
<http://genesilico.pl/modernaserver/submit/>

### Data files used in the tutorial

1. Experimental SAXS curve: *vaids\_saxs.dat*
2. *Ab-initio* shape reconstruction: *vaids\_damfilt.pdb* – *ab-initio* reconstructions of the VA<sub>1</sub>ΔTS solution conformation obtained using DAMMIF (<http://www.embl-hamburg.de/biosaxs/dammif.html/>).

A complete set of input/output files from each step of the tutorial is available at:

[ftp://ftp.genesilico.pl/jczwojdrak/vaids\\_tutorial/vaids\\_input\\_data.tar](ftp://ftp.genesilico.pl/jczwojdrak/vaids_tutorial/vaids_input_data.tar)



**STEP 1** Preparing the initial  $VA_1\Delta TS$  models e.g. using RNA Bricks database and ModeRNA web-server.

**STEP 2** Semi-flexible docking of the RNA model into SAXS *ab-initio* shape reconstruction using PyRy3D

**STEP 3** Completing model using ModeRNA server

**STEP 4** Validating the model quality

---

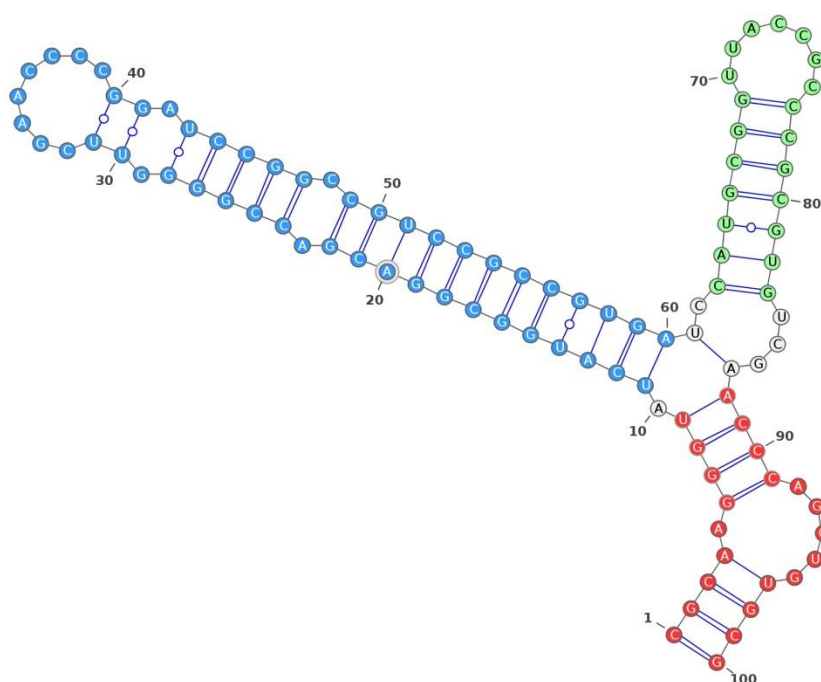
The aim of the tutorial is to build a 3D model of the VA<sub>1</sub>ΔTS based on following data:

- RNA (VA<sub>1</sub>ΔTS) sequence and secondary structure
- Ab-initio shape reconstruction based on SAXS data

Finally, we will validate the obtained model with experimental SAXS data (SAXS curve).

During this tutorial you will learn how to build a model of an RNA molecule that fits a provided SAXS shape reconstruction. In order to avoid over-fitting number of the model degrees of freedom is reduced as much as possible.

First, RNA is divided into rigid regions (stem-loops marked in red, blue and green on Figure 1). Second, individual helices of RNA (together with hairpin loops at their ends) are fitted into the provided SAXS *ab-initio* shape reconstruction as rigid bodies, restrained at the junction to maintain the continuity of the nucleotide chain. Finally the stem-loop fragments are merged to form a complete model.



**Figure 1** Secondary structure of the VA<sub>1</sub>ΔTS. Components that are treated as rigid are marked in red, blue and green. Gray nucleotides (three-way junction on the center) enable sampling of the molecule shape.

---

## STEP I: Preparing PyRy3D input files

To perform flexible fitting of RNA molecule into SAXS shape we will use 3D models of individual helices of RNA generated with ModeRNA server and RNA bricks (for detailed description how to obtain them please go to Supplementary Data section of this tutorial)

Now open PDB files with 3D coordinates of RNA fragments in any text editor. Notice that each RNA fragment has a different chain name (A, B and C respectively). Now copy all the three PDB files into a new directory (e.g. voidts) and prepare input archive for PyRy3D with a command:

```
$ tar cvf voidts.tar voids
```

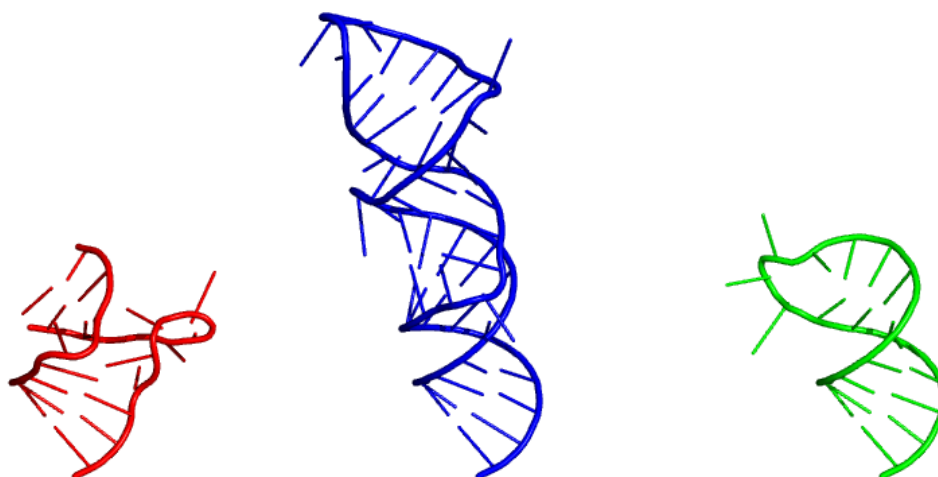


Figure 2 Stem-loop models generated using ModRNA

**STEP 2** Semi-flexible docking of the RNA model into *ab-initio* shape reconstruction using PyRy3D

### 2.1 Defining distance restrains for PyRy3D

Another input file for PyRy3D is a restrains file, with definition of distances between atoms that must be fulfilled in the models.

Restraints for the Pyry3D simulations should be defined in a text file following the rules of the Filtrest3D format (<http://filtrest3d.genesilico.pl/readme.html>). Check the example below:



```
dist (
  (G8) "C" - (U9) "D" (<=1.50)
  (U11) "D" - (C12) "A" (<=1.50)
  (G59) "A" - (A60) "D" (<=1.50)
  (C62) "D" - (C63) "B" (<=1.50)
  (G83) "B" - (U84) "D" (<=1.50)
  (G88) "D" - (C89) "C" (<=1.50)
)
```

residue name and number  
 chain ID  
 distance in Ångströms

To keep RNA fragments close to each other during the simulation, we should define distance restrains proportional to the length of missing chain fragments. As a rule of thumb you can set 3.5 Å per each missing ribonucleotide. The distance restrains definition that will keep input RNA fragments together should look like that:

**File: pyry3d\_vaidts\_restrains.txt**

```
dist (
  (U9) "A" - (U1) "B" (<=3.50)
  (A50) "B" - (C1) "C" (<=7.00)
  (G21) "C" - (A10) "A" (<=14.00)
)
```

Next prepare a file defining sequences of the input stem-loops. You can simply open PDB files in UCSF Chimera (<https://www.cgl.ucsf.edu/chimera/>) and use Tools→Sequence→Sequence to obtain sequence of each fragment.

Then, copy all the sequences into a single fasta file. Start each sequence with “>” and name of the chain used in the corresponding PDB files. As a result of you should obtain a FASTA file containing three sequences:

**File: pyry3d\_vaidts.fasta**

```
>A
CGCAAGGGUACCCAGGUGUGCG
>B
UCAUGGCGGACGACCGGGUUCGAACCCCGGAUCCGGCCGUCGCGUGA
>C
CAUGCGGUUACCGCCCGUG
```

\* note that colors of the sequences refer to colors used for corresponding fragments of the RNA model.



## 2.2 Preparing input configuration file for PyRy3D

PyRy3D input parameters can be defined either as a simple text file, or using the web-interface (<http://pyry3d.icm.edu.pl/>). In this exercise we will use the former option, because it is more transparent and enables easier testing of many different configurations. The parameters file has a very simple format, each line either starts with a parameter name followed with its value:

```
PARAMETER_NAME PARAMETER_VALUE
```

Lines starting with ‘#’ are treated as comments. Below is the list of the most important parameters and their default values, that should yield best results in most of the cases.

Bear in mind, that the more simulation steps (STEPS parameter value) you choose, the more time the modeling process will take, but also you will get more accurate models!

| Simulation parameters                          | What is for?  | Values used in the tutorial |
|--|---|-----------------------------|
| SIMMETHOD                                      | Simulation algorithm (e.g. simulated annealing, genetic algorithm, replica exchange)                                | Simulated annealing         |
| STEPS  | Number of iterations to run   | 20 000                      |
| ANNTEMP*                                       | Simulated annealing temperature parameter   | 10                          |
| <b>Scoring function</b>                        |   |                             |
| MAP_FREESPACE                                  | Weights of a penalty for unoccupied volume inside density map at the beginning and end of simulation                | 10 10                       |
| CLASHES  | Weights of a penalty for collisions between main chain atoms (Ca or C4') at the beginning and end of the simulation | 5 10                        |
| RESTRAINTS                                     | Weights of a penalty for disagreement with user-defined restraints at the beginning and end of the simulation       | 5 5                         |
| <b>Mutation operations</b>                     |   |                             |
| ROTATION_FREQ                                  | Frequency of rotation mutation  | 0.5                         |
| TRANSLATION_FREQ                               | Frequency of translation mutation   | 0.5                         |
| <b>Density map, simulation grid parameters</b> |   |                             |
| SIMBOX   | Parameter indicates how many times simulation box diameter is bigger than the longest distance inside a density map | 1.2                         |
| GRIDRADIUS                                     | Radius of a single grid cell in Å   | 1.0                         |
| SAXSRADIUS                                     | Dammif pseudoatom radius  | 3.0                         |

\* refers to simulated annealing method only



---

Open the `config_file.txt` available in tutorial materials and check whether values of parameters agree with one mentioned in the table above. A part of the configuration file is shown below

**File: `pyry3d_options.txt`**

```
SIMMETHOD SimulatedAnnealing
STEPS 40000
ANNTEMP 10
REHEAT true 1000
MAXROT 5
MAXTRANS 5 5 5

## --- Scoring function parameters -----
OUTBOX 10 10
MAP_FREESPACE 10 10
CLASHES_ALLATOMS 10 10
RESTRAINTS 5 5

SAXSRADIUS 3.0

### ---- Mutation frequencies -----
ROTATION_FREQ 0.4
TRANSLATION_FREQ 0.4
TRANSLATION_ALL_FREQ 0.1
ROTATION_ALL_FREQ 0.1

## ----- Input data descriptors -----
THRESHOLD 0.7
SIMBOX 1.2
GRIDRADIUS 1.0
COMPONENT_REPRESENTATION fa

START_ORIENTATION False
WRITE_N_ITER 40000
```

Now you can run PyRy3D web-server with the configuration files listed above, tar-archived directory containing model fragments, and a shape reconstruction file provided in the start package (*voidts\_damfilt.pdb*).

Calculations may take a while, so be patient PLEASE 😊



## 1. Results analysis

When the program completes, in **results** directory you will get :

- A .log file with information about the modeling process and scores for complex structures generated by the program.

```
Scoring function will penalize shape filling twice since you defined two
parameters: DENSITY and MAP_FREESPACE!
ANNTEMP..... 10.0
CLASHES_PENALTY..... [4.0, 4.0]
COMPONENTS_INDEXES..... {}
DENSITY_PENALTY..... [1.0, 1.0]
DISABLED_MUTATIONS..... ['rotation_cov', 'exchange',
'SimulateDisorder']

cx score for iteration 1 is -262.697 components: restraints: -21.152
collisions: -10.668 map filling: -163.362 outbox atoms: -0.0
density filling: -67.517
ACTUAL WEIGHTS are respectively: 2.0, 4.0, 3.0, 0.0, 1.0|
cx score for iteration 6 is -261.533 components: restraints:-22.42
collisions: -8.0 map filling: -163.551 outbox atoms: -0.0
density filling:-67.562
ACTUAL WEIGHTS are respectively: 2.0, 4.0, 3.0, 0.0, 1.0
```

- A set of PDB files with models coordinates. Each PDB file is named according to the following rule :

**Outname\_score\_iteration\_temperature.pdb**

Where:

- Outname – the name given by the user when running the program, in that case “results”
- Score – penalty for a disagreement with restraints assigned by the program. The closer to 0, the better!
- Iteration – in which simulation step this particular structure was created
- Temperature – simulation annealing temperature parameter

From a set of generated PDB files choose the one with the best PyRy3D score (the closer to 0, the better). Now check:

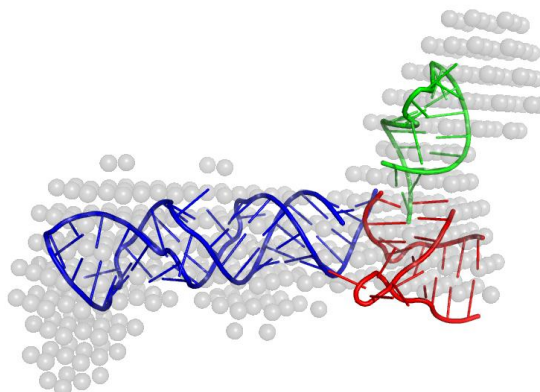




- 
- What is the score of the best model in your simulation?
  - Do other participants have similar/exactly the same score for their models?
  - Do you know why the differences in scores appear?
  - In .log file check what are the penalties for collisions, restraints and fit to density map for the best model.

### STEP 3 Completing model using ModeRNA

Pyry3D output is a PDB file that contains all the RNA fragments provided on input. In most cases they will nicely fit the input SAXS reconstruction, but it may happen that the minimization procedure gets stuck in a local minima. A standard procedure would be then to repeat the fitting many times starting from random starting positions of the subunits, and select a lowest energy solution for further processing. For the purpose of this exercise we will have limit number of simulations to a few (two should be enough). When displayed in PyMOL the best scored result should look more or less like that:



Size and transparency of the SAXS reconstruction spheres (*vaids\_damfilt.pdb*) can be adjusted with following commands:

```
PyMOL> set sphere_scale, 2  
PyMol> set sphere_transparency, 0.5
```

The separate stem-loops have different chain ids and few nucleotides are missing. You can fix that using ModeRNA. First renumber the nucleotides according to their positions in the target VA<sub>1</sub>ΔTS sequence. Use PyMOL again:

```
PyMOL> load results.pdb, results
```



```
PyMOL> alter (chain B), resi=str(int(resi)+10)
PyMOL> alter (chain C), resi=str(int(resi)+62)
PyMOL> alter (chain A and resi 10-), resi=str(int(resi)+78)
PyMOL> alter (all), chain='A'
PyMOL> sort (all)
PyMOL> save results_renumbered.pdb, results
```

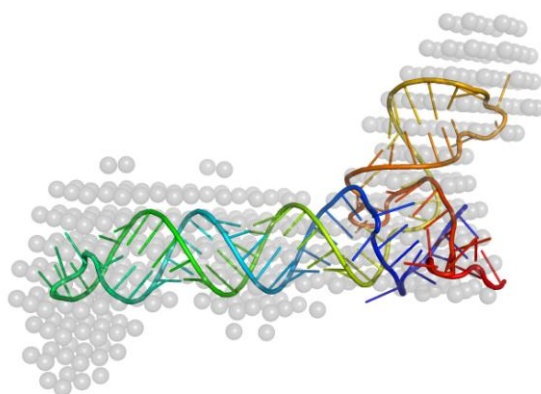
Next prepare an input alignment for ModeRNA. Use full sequence of the VA<sub>1</sub>ΔTS as a target and the PyRy3D model as a template. You can prepare that file from scratch or edit an input fasta file `voidts.fasta`

#### File: `moderna_voidts_final.fasta`

```
>target
CGCAAGGGUAUCAUGGCGGACGACCGGGGUUCGAACCCCGGAUCCGGCCGUCGCGCGU
GAUCCAUGCGGUUACCGCCCGUGUCGAACCCAGGUGUGCG
>model
CGCAAGGGU-UCAUGGCGGACGACCGGGGUUCGAACCCCGGAUCCGGCCGUCGCGCGU
GA--CAUGCGGUUACCGCCCGUG-----ACCCAGGUGUGCG
```

Now you can use ModeRNA server (<http://genesilico.pl/modernaserver/submit/model/>) to prepare the final model.

The output should look more or less like that:



## STEP 4 Validation of the resulting model quality

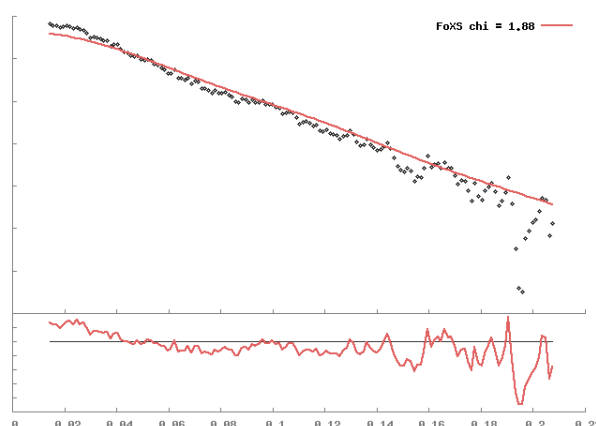
### 4.1 How well the model explains the SAXS data?

All models, in particular those generated using low-resolution approaches should be verified. Therefore in a next step you should check how well your model explains the observed SAXS



data. There are two on-line tools available that can be used for that purpose Foxs web server (<http://modbase.compbio.ucsf.edu/foxs/>) and CRY SOL (<http://www.embl-hamburg.de/biosaxs/atsas-online/>).

A crucial parameter that measures a goodness-of-fit of the experimental and model-derived SAXS curves is the reduced  $\chi^2$  value. If the two curves are almost identical (within the experimental errors) this parameter should be close to 1.0 (or at least less than 1.5). Otherwise the model should be revised. Owing the simplicity of the modeling procedure used in this tutorial results you obtain may be slightly worse.



## 4.2 Model quality validation with Molprobit

Another way to validate model quality is to check reliability of local geometry using Molprobit suite. That tool will compare your model with a large set of crystallographic structures (these are often reg

arded as a reference). Clearly good quality of a model geometry doesn't necessarily mean that it is true, but pretty models are in general easier to analyze.

Go to <http://molprobit.biochem.duke.edu/> and provide ModeRNA model as an input. Next select 'Add hydrogens' option (hydrogen atoms are required for the analysis of clashes) and 'Analyze all-atom contacts and geometry'. After a short time you should get the results more or less like that:



### Summary statistics

|                          |   |           |  |             |
|--------------------------|---|-----------|--|-------------|
| All-Atom<br>Contacts     | Clashscore, all atoms:  | 34.24     | 11 <sup>th</sup> percentile* (N=1784, all resolutions) |             |
|                          | Clashscore is the number of serious steric overlaps (> 0.4 Å) per 1000 atoms. |           |  |             |
| Nucleic Acid<br>Geometry | Probably wrong sugar puckers:   | 2         | 2.00%  | Goal: 0     |
|                          | Bad backbone conformations <sup>#</sup> :                                     | 8         | 8.00%  | Goal: <= 5% |
|                          | Bad bonds:  | 30 / 2384 | 1.26%  | Goal: 0%    |
|                          | Bad angles:   | 96 / 3716 | 2.58%  | Goal: <0.1% |

In the two column results, the left column gives the raw count, right column gives the percentage.

\* 100<sup>th</sup> percentile is the best among structures of comparable resolution; 0<sup>th</sup> percentile is the worst. For clashscore the comparative set of structures was selected in 2004, for MolProbity score in 2006.

<sup>#</sup> RNA backbone was recently shown to be rotameric. Outliers are RNA suites that don't fall into recognized rotamers.

According to the Molprobity tests, geometry of a model we created using the above-given instructions is better than 10% of crystal structures. Pretty good for a 1 hour tutorial, isn't it?



---

## ACKNOWLEDGEMENTS

More information concerning PyRy3D tool and its application in structural biology can be found on our website: <http://genesilico.pl/pyry3d/>

If you have any comments, recommendations, ideas how to improve the workshop or the software please do not hesitate to contact us:

Joanna Kasprzak: [jkasp@amu.edu.pl](mailto:jkasp@amu.edu.pl)

Grzegorz Chojnowski: [gchojnowski@genesilico.pl](mailto:gchojnowski@genesilico.pl)

Janusz Bujnicki: [iamb@genesilico.pl](mailto:iamb@genesilico.pl)

We would like to acknowledge people who helped us to prepare this tutorial for all recommendations, comments and exhaustive testing of our software: **Jagoda Jabłońska** and **Beata Jaroszewska** (Adam Mickiewicz University in Poznań).

**Thank you for attending the workshop!!**

On behalf of program authors, with kind regards,

Joanna Kasprzak & Grzegorz Chojnowski




---

**SUPPLEMENTARY DATA.**

First we need to prepare 3D models of the VA<sub>1</sub>ΔTS molecule rigid fragments. We generously assume here that their geometry doesn't deviate much from an ideal A-RNA double helix. That will allow to prepare simple homology models by adding missing loops and insertions to model A-RNA helices using RNA homology modeling server – ModeRNA (<http://iimcb.genesilico.pl/modernaserver/submit/model/>). Resulting models can be used as a starting point for hybrid modeling with PyRy3D.

A listing below presents sequence and secondary structure of the VA<sub>1</sub>ΔTS molecule with main stems marked as A (red), B (blue), and C (green).

**File: voidts.fasta**

```
> voidts ss extracted from Figure 1
CGCAAGGGUAUCAUGGCGGACGACCGGGGUUCGAACCCCGGAUCCGGCCGUCGCGCGUGA
(((((((.....)))))))))
AAAA-AAAA BBBBBBBBBBBB-BBBBB-----BBBBB-BBBBBBBBBBBB

UCCAUGC GGUUAACGCCCGCGUGUCGAACCCAGGUGUGCG
(.((((.....))))))....))))).....)))))
..CCCCCCC-----CCCCCCC....AAAA-----AAAA
```

### 1.1. Preparing A-RNA models

First get A-RNA double helix template models for homology modeling from RNA Bricks database (<http://iimcb.genesilico.pl/rnabricks>). Use sequence-based search engine to find any helices with appropriate lengths. The are 8, 17, and 7 base-pairs for stem A, B, and C respectively. For example query for the stem A should look that:

```
>s1
.{8}
>s2
.{8}
```


 RNA Bricks

[Home](#) | [Browse](#) | [Search](#) | [Statistics](#) | [Help](#)

## RNA Bricks search form

Structure
Sequence

On this page you can query RNA Bricks with ribonucleotide sequences. Please keep in mind that you must define ALL single-stranded fragments that define a motif. See [help](#) for details.

**Title:**

Here you can shortly describe your query

**Fasta text:**

Query in fasta format ⓘ

**Case sensitive:**  Enable modified (lowercase) residues search ⓘ

**Aggregate results:**  Aggregate results by the 3D structure similarity ⓘ

**Motifs type:** Stems ▼

Select requested RNA motif type.  
It must be consistent with the FASTA query i.e. for stems two segments of equal length must be defined

Show advanced options

Load example data / see example results

Paste query into the *Fasta text* box and select motif type (*stems*). Additionally you can request to *Aggregate results* in order to reduce number of hits. From a results list select PDBID of selected motif. You will be redirected to the listing of motifs from that structure with selected motif being on top of the list. Finally select *download* link from the *Motifs residues* column.

## 1.2 Homology modeling with ModeRNA

Now prepare input files for ModeRNA. These are templates (double helices found in the previous step) and FASTA files defining their alignment with target sequences (included into the zip-archives downloaded from RNA Bricks database)

Next, align the sequences with corresponding parts of the target sequence. You can easily do that manually using a favorite text editor with monospaced fonts (e.g vim, but Courier New font in Word or OpenOffice will be fine too). **Please note that model sequences in the following alignments correspond to the exemplary stems provided in the start package.**

**File:** moderna\_vaidts\_stem\_A.fasta



```
>target
CGCAAGGGUACCCAGGUGUGCG
>model
CGCA-GGGUACCC-----UGCG
```

**File: moderna\_vaidts\_stem\_B.fasta**

```
>target
UCAUGGCGGACGACCGGGGUUCGAACCCCGGAUCCGGCCGUCGCGCGUGA
>model
UCAUGGCGGACG-CCGGG-----CCCGG-CGUCCGCCAUGA
```

**File: moderna\_vaidts\_stem\_C.fasta**

```
>target
CAUGCGGUUACCGCCCGCGUG
>model
CAUGCGG-----CCGCAUG
```

Next use these to generate stem-loop models using ModeRNA server (remember to check ‘clean structure’ option at <http://genesilico.pl/modernaserver/submit/model/>).

ModeRNA uses ‘insertion codes’ to mark residues inserted into the template model. You will have to get rid of them using ModeRNA server again (‘renumber chain’ in the analyze tool at <http://genesilico.pl/modernaserver/submit/analyse/>).





### 1.3 Fixing chain names

During previous step ModeRNA renamed all the chain fragments back to 'A'. In order to be able to define restrains in Pyry3D docking simulation you will have to fix that. Simply run the following PyMOL commands (or use a text editor):

```
PyMOL> load stemA.renumbered_chain.pdb, stem
PyMOL> alter (stem), chain="A"
PyMOL> save stemA.renumbered_chain.pdb, stem
```

```
PyMOL> load stemB.renumbered_chain.pdb, stem
PyMOL> alter (stem), chain="B"
PyMOL> save stemB.renumbered_chain.pdb, stem
```

```
PyMOL> load stemC.renumbered_chain.pdb, stem
PyMOL> alter (stem), chain="C"
PyMOL> save stemC.renumbered_chain.pdb, stem
```