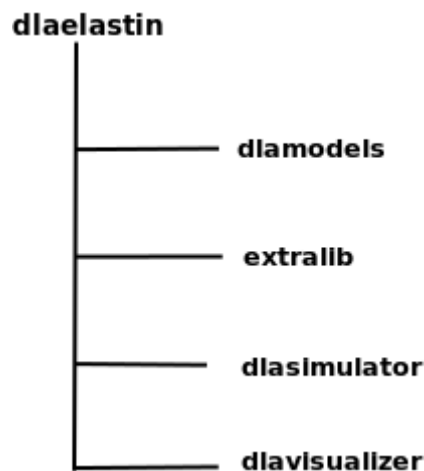


Simulation Programs for Modelling Elastin Assembly

1. The folder structure

The simulation package includes four directories, dlamodels, extralib, dlasimulator, dlavisualizer. The dlamodels directory contains the common code used by both dlasimulator and dlavisualizer. The extralib directory contains the Apache xerces xml parser, which is required by the dlamodels. The dlasimulator directory contains the code for building the simulator, and the dlavisualizer directory contains the code to build a 3D trace visualizer.



2. Compile and run the program

2.1. Compiling

To compile the simulator, first compile the dlamodels, then compile the dlasimulator. Compiling the dlavisualizer follows the same steps as compiling dlasimulator. However, the QT library 4.3+ version is required by the dlavisualizer. The QT library can be downloaded at <http://qt.nokia.com/downloads>.

2.2. Run the simulator

To run the simulator, type “dlasimulator modelfile tracefile” on the command line, where the modelfile specifies the parameters of the simulation model, and the tracefile is used to save the output simulation trace. How to write the modelfile will be explained in section 2.4. The simulator can be restarted from previous simulations. This means that when it runs with a non-existing trace file, the simulation will run start from scratch, when it runs with an existing trace file, the simulator will load the trace file and continue running the existing simulation.

2.3. Run dlavisualizer

To run the simulator, simply type “dlvisualizer” or “dlvisualizer tracefile” on the command line. If the tracefile argument is specified, the trace in the file will be visualized. Or one can open a trace file in the visualizer’s window after it starts.

2.4. Write a simulation specification file

The simulation specification file is a XML file specifying the parameters of the simulation. The simulation specification includes several sections.

Termination Condition

This section specifies how the simulation is terminated. A simulation can be normally terminated in two ways, by specifying a certain amount of time and by specifying the number of particles in the aggregate. The simulation will be terminated by the first satisfied condition.

To generate an aggregate containing 100 particles, the termination condition can be specified as below.

```
<number-of-elements>100</number-of-elements>  
<days-of-simu-time>0.5</days-of-simu-time>
```

The simulation will be terminated either when the aggregate including 100 particles has been generated or the the time limit has been reached. If the time limit has been reached before enough particles included in the aggregate. The simulation can be re-run using the same specification file and trace file. The program will continue the simulation. This process can be repeated many times until the number of particles in the aggregate satisfied.

Release Template

This section specifies attributes of the release template when the simulation begins. For a three dimensional template, it includes the original width, height, and depth of the template. The central position of the template is supposed at the centre of the aggregate (normally [0, 0, 0]). Besides the dimensions, it also specifies the distance between the surface of the template and the surface of the aggregate, and the expanding step of the template. For the elastin simulation, the expansion of the release template depends on the dimension of the aggregate. So only template-distance and template-expand-step are used.

```
<template-distance>30</template-distance>  
<template-expand-step>1</template-expand-step>  
<template-adjust-ratio>0.0</template-adjust-ratio>  
<template-width>30</template-width>  
<template-height>30</template-height>  
<template-depth>30</template-depth>
```

Particle Specification

This part specifies the attributes of the simulation particles. The simulation particle can be defined by five attributes: element-orientation, element-width, element-height, element-depth, and element-

speed. The element-orientation specifies where the direction of a particle is fixed. The element-width, element-height, element-depth specify three dimensions. The element-speed means the speed of the particle when it does random walk.

```
<element-orientation>1</element-orientation>
<element-width>20</element-width>
<element-height>1</element-height>
<element-depth>1</element-depth>
<element-speed>1</element-speed>
```

Surface Diffusion

This section specifies the parameters used for surface diffusion. The surface-diffusion element has two attributes, 'enabled' and 'level'. 'enabled' can be set to 0 and 1, 1 means surface diffusion is enabled and 0 means disabled. 'level' was used to test for different surface diffusion methods, for current elastin simulation, 'level' is set to 1. There are 5 sub-elements for controlling the surface diffusion behaviour. 'diffusion-step' specifies the angle in radian each time a particle going left or right. 'kt-constant' defines the temperature parameter for computing free energy. 'try-times' specifies the times of movement allowed for diffusion. 'distance-threshold' specifies the centre distance of neighbour rods for computing free energy.

How diffusion-step works

The elastin rod diffuses on the surface of the aggregate randomly. Since the rod is not allowed to move along the length direction of the rod, it can only move left or right on the surface of the aggregate. Each time the rod can move left or right by a random angle between (0, 0.78) radian. The moving angle is generated randomly using random number generator. It is very possible that if the rod finished the movement one time it will go inside its neighbour rods rather than moving on the surface. To prevent this happens, the rod is controlled to move left or right with small steps. For example, if the generated random angle is 5 radian and the diffusion-step is 0.01, then the rod will take $5/0.01=500$ steps to finish this diffusion rather than 1.

```
<surface-diffusion enabled="0" level="1">
  <diffusion-step>0.01</diffusion-step>
  <kt-constant>20</kt-constant>
  <try-times>1000</try-times>
  <distance-threshold>1.2</distance-threshold>
  <influence-coefficient>0.01</influence-coefficient>
</surface-diffusion>
```

Hydrophobic Constraints

This part specifies the conditions on which a moving particle is attached to the aggregate. The probability of the moving particle attaching to the aggregate is computed based on the percentage of the overlapped hydrophobic length of the two most close particles.

There are two attributes and two sub-elements for specifying the attaching probability. The attribute 'enabled' defines whether the constraint is enforced. If this constraint is disabled, the moving particle will be attached to the aggregate when it collides with the aggregate unconditionally. The

'type' attribute specifies which method is used to calculate the attaching probability. Based on different computing methods, the sub-elements 'coefficient' and 'constant' have different meanings.

type:

1: The probability is linear to percentage of the overlapped hydrophobic parts of the two tropoelastins. $p = a*x + b$, where a is the coefficient and b is the constant.

2: The probability is exponential density distribution to the overlapped hydrophobic parts of the two tropoelastins. $p = a * e^{(-a*x)}$, where a is the coefficient.

3: The probability is exponential cumulative distribution to the overlapped hydrophobic parts of the two tropoelastins. $p = 1 - e^{(-a*x)}$, where a is the coefficient.

4: Power distribution, $p = x**a$, where a is the coefficient.

5: Normal distribution, $p = 1/\sqrt{2*\pi*\delta\text{Square}}*\exp(-(x-0.5)**2/(2*\delta\text{Square})) - \text{constant}$, where δSquare is coefficient, and μ is constant. And $\delta\text{Square} = 0.09f$, $\text{constant} = 0.33f$.

Below is an example of type 1 hydrophobic constraint:

```
<hydrophobic-constraints enabled="1" type="1">  
  <coefficient>1.0</coefficient>  
  <constant>0</constant>  
</hydrophobic-constraints>
```

Element Template

This section defines the domain structure of elastin models. Each domain is specified by two attributes: type and length. Type can be 'h' or 'c', where 'h' represents hydrophobic, and 'c' represents cross-linking domains. Length of the domain here means the proportional length not the absolute length. For example, the domain structure below:

```
<element-template>  
  <exon type="h" length="2" />  
  <exon type="c" length="2" />  
  <exon type="h" length="2" />  
  <exon type="c" length="2" />  
  <exon type="h" length="2" />  
</element-template>
```

The proportional length of each domain is 2. If the element-width in the previous section is 20, then the domain length of each part is $20*(2/(2+2+2+2+2))=4$.

Centre Adjustment

Centre adjustment was used for testing purpose. It is not used for final simulation. However, the program will try to read it in the specification file. So, it still need to be presented in the specification file.

Sample of Simulation Specification

A typical model specification file looks as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<simu-context>
  <number-of-elements>100</number-of-elements>
  <days-of-simu-time>7</days-of-simu-time>

  <!--
    Specifications for the release template.
  -->
  <template-distance>30</template-distance>
  <template-expand-step>1</template-expand-step>
  <template-adjust-ratio>0.0</template-adjust-ratio>
  <template-width>30</template-width>
  <template-height>30</template-height>
  <template-depth>30</template-depth>

  <!--
    Properties of the particles.
  -->
  <element-orientation>1</element-orientation>
  <element-width>20</element-width>
  <element-height>1</element-height>
  <element-depth>1</element-depth>
  <element-speed>1</element-speed>

  <center-adjustment enabled="0">
</center-adjustment>

  <surface-diffusion enabled="0" level="1">
  <diffusion-step>0.01</diffusion-step>
  <kt-constant>20</kt-constant>
  <try-times>1000</try-times>
  <distance-threshold>1.2</distance-threshold>
  <influence-coefficient>0.01</influence-coefficient>
</surface-diffusion>

  <hydrophobic-constraints enabled="1" type="1">
  <coefficient>1.0</coefficient>
  <constant>0</constant>
</hydrophobic-constraints>

  <element-template>
  <exon type="h" length="2" />
  <exon type="c" length="2" />
  <exon type="h" length="2" />
  <exon type="c" length="2" />
  <exon type="h" length="2" />
  </element-template>
</simu-context>
```

3. Major Programming Components

The major class association diagram is shown as below.

