This **Supplementary Material** features the article

"SPATKIN: A simulator for rule-based modeling of biomolecular site dynamics on surfaces"

by Kochańczyk *et al.*, *Bioinformatics*, 2017

This document is intended as a practical introduction to SPATKIN, that demonstrates its capabilities by providing 10 annotated tutorial "μmodels":

All "μmodels" are shown in the form of complete, runnable inputs, and are included as separate files in the source code distribution of SPATKIN (in the directory `doc/examples`). An archive containing these files can be also downloaded directly from the software Web site. Additional SPATKIN documentation can be found in the User Manual.

Tutorial μmodel 1: Heterogeneous initial location of molecules

This example (see Listing 1) shows how user-defined regions can be used to place molecules in a non-homogeneous manner on the lattice. One region is used to constrain the initial placement of molecules A(x∼U,y∼U) (drawn grey), at time=0; other regions are filled with molecules B (red) and C (blue) in the course of simulation according to a temporally constrained emergence rule. Molecules B modify molecules A(x∼U,y∼U) into A(x∼P,y∼U) – green. Molecules C modify molecules A(x∼U,y∼U) into A(x∼U,y∼P) – yellow. Molecules A and B degrade slowly. Simulation shows how polarization can be introduced to the system and how it vanishes due to diffusion.

```
begin parameters
  m 3
  m2 10   k 0.1    # This is a free-format text, meaning that line breaks
  r 10   q 0.01    #  do not matter.
end

begin world
  topology plane  size 200 200
end

begin regions
  # Two basic primitives for defining regions are circles and rectangles.
  CircRgn  circle    100 100 50      # centerX centerY radius
  RectRgn  rectangle 100 150 200 100 # centerX centerY width height

  # Typical constructive geometry operations are supported:
  RgnX  !CircRgn                    # !a =: complement of set a
  RgnY  (CircRgn * RectRgn)         # (a * b) =: intersection of a and b
  RgnZ  (CircRgn - RectRgn)         # (a - b) =: subtraction of b from a
end

begin molecule types
  A(x,y)   B()   C()
end

begin seed species                              # Number of molecules or frac-
  A(x∼U,y∼U)  occupancy 0.1  in region RgnX   #  tional (region) occupancy
end                                             #  should be given here.
```

```
begin event rules
  # Diffusion, molecule emergence and degradation rules have prefix syntax.
  >> A()  m    #  |
  >> B()  m2   #  >- diffusion
  >> C()  m2   # |

  # Following 2 rules for molecule insertion are both spatially and
  #  temporally constrained; effective rate of insertion is proportional
  #  to the number of unoccupied lattice nodes (here, in a region):
  ++ B()  k  in region RgnY  since 5  until 8
  ++ C()  k  in region RgnZ  since 5  until 8

  B() + A(x~U,y~U) -> B() + A(x~P,y~U)  r
  C() + A(x~U,y~U) -> C() + A(x~U,y~P)  r

  -- B()  q     -- C()  q    # Molecules B, C are degraded with rate q.
end

begin observables
    A  A(x~U,y~U)  color lightgrey          # All observables that
    Ax A(x~P,y~U)  color green              #  have assigned colors
    Ay A(x~U,y~P)  color gold               #  are recorded in the
    B  B()         color red                #  trajectory file.
    C  C()         color blue               #
end

begin simulation
    time end 300            # Total duration (in simulation time units).
    observer intervals 100  # No. time points at which logging occurs.
end
```

Listing 1: Tutorial input file 1 (doc/examples/tutorial/01-regions.spatkin).

Tutorial μmodel 2: Diffusion-limited aggregation

This example (Listing 2) demonstrates diffusion-limited aggregation in just 3 (effectively 2) rules (rules of zero rate are omitted).

```
begin parameters
  m        10
  kfast 10000
end

begin world
  topology plane  size 120 120
end

begin regions
  Seeds cells 20,30; 70,20; 82,82  # Region consists of single lattice nodes.
end

begin molecule types
  Particle(mobile)
end

begin seed species
  Particle(mobile~U)  occupancy 1.0  in region Seeds  # confined to a region
  Particle(mobile~P)  occupancy 0.2                   # distributed uniformly
end

begin event rules
  "Gogogo!":
  >> Particle(mobile~P)  m  # A named rule for diffusion.

  >> Particle(mobile~U)  0  # Anonymous rule (referred to as rule "2").

  # According to the above rules, molecule diffusivity depends on its state.
  Particle(mobile~P) + Particle(mobile~U)  ->
  Particle(mobile~U) + Particle(mobile~U)  kfast
end
```

```
begin observables
  frost Particle(mobile~P)  color lightgreen # An array of observables colors
  snow  Particle(mobile~U)  color orangered  #  has been predefined for user's
end                                          #  convenience (see user manual).

begin simulation
  time end 100          # If there are no more events, a warning will be issued.
  observer intervals 200
end
```

Listing 2: Tutorial input file 2 (doc/examples/tutorial/02-aggregation.spatkin).

Tutorial μmodel 3: State-dependent removal from the membrane

This example (Listing 3) is inspired by the fact that lipid modification, such as palmitoylation or farnesylation, can affect membrane attachment of proteins; for example, G protein α subunit is depalmitoylated upon stimulation and then translocates to cytosol.

```
begin parameters
  m1  10
  b  1
  d  10
  u  10
  x  0.1
end

begin world
  topology plane  size 100 100
end

begin regions  (* none *)  end

begin molecule types          # By defining relative molecular weights in
  Thioesterase(a)  weight 1   #  this manner, we assure that Thioesterase,
  AlphaS(palmito)  weight 0   #  which is assumed immobile, does not move
end                           #  upon binding/unbinding AlphaS.

begin seed species
  AlphaS(palmito~P)  1000     # site 'palmito' defined explicitly as unbound
  Thioesterase(a)      10     # site 'a'       defined explicitly as unbound
end

begin event rules
 >> Thioesterase(a)  0       # assumed immobile
 >> AlphaS(palmito) m1       # assumed mobile
 Thioesterase(a) + AlphaS(palmito~P)  ->Thioesterase(a!1).AlphaS(palmito~P!1) b
 Thioesterase(a!1).AlphaS(palmito~P!1)->Thioesterase(a!1).AlphaS(palmito~U!1) d
 Thioesterase(a!1).AlphaS(palmito~U!1)->Thioesterase(a) + AlphaS(palmito~U)   u
 -- AlphaS(palmito~U) x      # removal of depalmitoylated AlphaS from membrane
end
```

```
begin observables
  A_palmi   AlphaS(palmito~P!?) color lightpink # '?!' means that the status of
  A_depalmi AlphaS(palmito~U!?) color red       #  binding is irrelevant here
  T         Thioesterase()       color darkblue
end

begin simulation
  description "Depalmitoylation v0.1" # Descriptions are copied into results.
  duration 300                        # Bimolecular complexes are occasionally
  observer intervals 100              #  seen in trajectory as split hexagons.
end
```

Listing 3: Tutorial input file 3 (doc/examples/tutorial/03-depalmitoylation.spatkin).

Tutorial μmodel 4: Rule-based capabilities (1)

This example (Listing 4) demonstrates rule-based capabilities.

Each molecule S ("substrate") can be independently phosphorylated on 10 residues, meaning that S may assume one of $2^{10} = 1024$ phosphorylation states. Residues A, B, C, D, E can be phosphorylated by kinase K1 which is recruited and remains tethered in the circular region RgnL; residues F, G, H, I, J can be phosphorylated by kinase K2 that is recruited and remains tethered in the circular region RgnR (in this way, occurrence of several second-order reactions is constrained spatially). To become phosphorylated on all residues (dark red observable), S must visit both regions.

In the absence of phosphatase activity (parameter ku = 0), all S are ultimately phosphorylated but even a weak activity of uniformly distributed phosphatases (parameter ku = 0.01) prevents simultaneous phosphorylation of S on all residues.

```
begin parameters
  m     10.     # diffusivity
  kadd  0.1     # insertion rate
  kp    10.     # kinase activity
  ku    0.0     # phosphatase activity  <-- CHOOSE: ku=0 or ku=0.01
  occuS 0.1     # substrate occupancy
  occuP 0.03    # phosphatase ocupancy
end

begin world
  topology plane  size 200 100
end

begin regions
  RgnL  circle  50 50 30
  RgnR  circle 150 50 30
end

begin molecule types
  S(A,B,C,D,E,F,G,H,I,J) # a multi-site substrate
  K1()                   # a kinase
  K2()                   # another kinase
  P()                    # a phosphatase
end
```

```
begin seed species
  S(A~U,B~U,C~U,D~U,E~U,F~U,G~U,H~U,I~U,J~U)  occupancy occuS
  P()  occupancy occuP
end

begin event rules

  >> S() m  # By omitting diffusive rules for K1 and K2, they are
  >> P() m  #  made immobile.

  K1() + S(A~U) -> K1() + S(A~P) kp
  P()  + S(A~P) -> P()  + S(A~U) ku

  K1() + S(B~U) -> K1() + S(B~P) kp
  P()  + S(B~P) -> P()  + S(B~U) ku

  K1() + S(C~U) -> K1() + S(C~P) kp
  P()  + S(C~P) -> P()  + S(C~U) ku

  K1() + S(D~U) -> K1() + S(D~P) kp
  P()  + S(D~P) -> P()  + S(D~U) ku

  K1() + S(E~U) -> K1() + S(E~P) kp
  P()  + S(E~P) -> P()  + S(E~U) ku

  K2() + S(F~U) -> K2() + S(F~P) kp
  P()  + S(F~P) -> P()  + S(F~U) ku

  K2() + S(G~U) -> K2() + S(G~P) kp
  P()  + S(G~P) -> P()  + S(G~U) ku

  K2() + S(H~U) -> K2() + S(H~P) kp
  P()  + S(H~P) -> P()  + S(H~U) ku

  K2() + S(I~U) -> K2() + S(I~P) kp
  P()  + S(I~P) -> P()  + S(I~U) ku
```

```
  K2() + S(J~U) -> K2() + S(J~P) kp
  P()  + S(J~P) -> P()  + S(J~U) ku

  ++ K1() kadd  in region RgnL  since 1.0  until 3.0
  ++ K2() kadd  in region RgnR  since 5.0  until 7.0
end

begin observables
  S_10u     S(A~U,B~U,C~U,D~U,E~U,F~U,G~U,H~U,I~U,J~U) color blue
  S_5pNterm S(A~P,B~P,C~P,D~P,E~P,F~U,G~U,H~U,I~U,J~U) color gold
  S_5pCterm S(A~U,B~U,C~U,D~U,E~U,F~P,G~P,H~P,I~P,J~P) color pink
  S_10p     S(A~P,B~P,C~P,D~P,E~P,F~P,G~P,H~P,I~P,J~P) color darkred
  K1  K1() color black
  K2  K2() color dimgrey
  P   P()  color green
end

begin simulation
  time end 500
  observer intervals 100
end
```

Listing 4: Tutorial input file 4 (doc/examples/tutorial/04-rule_based_1.spatkin).

Tutorial μmodel 5: Rule-based capabilities (2)

This example (Listing 5) is a further demonstration of rule-based capabilities and extends the previous example. Additionally here, independently of their phosphostate, molecules S can form homodimers, so there are $(2^{10})^2/2 \simeq$ over half a million potential homodimer species (more than molecules in the simulation). Dimers in which one protomer is phosphorylated on A, B, C, D, E and the other is phosphorylated on F, G, H, I, J are stabilized (not allowed to dissociate).

```
begin parameters
  m     10.    # diffusivity
  kadd   0.1   # insertion rate
  kp    10.    # kinase activity
  ku     0.0   # phosphatase activity  <-- CHOOSE: ku=0 or ku=0.01
  occuS  0.1   # S occupancy
  occuP  0.03  # phosphatase ocupancy
  b      1     # S homodimerization
  d      1     # S-S un-dimerization
  stb  100     # S-S homodimer stabilization
end

begin world
    topology plane  size 200 100
end

begin regions
  RgnL  circle  50 50 30
  RgnR  circle 150 50 30
end

begin molecule types
  S(A,B,C,D,E,F,G,H,I,J,dim,stable)
  K1()   K2()   P()
end

begin seed species
  S(A~U,B~U,C~U,D~U,E~U,F~U,G~U,H~U,I~U,J~U,dim,stable~U) occupancy occuS
  P()  occupancy occuP
end
```

```
begin event rules
  >> S() m
  >> S(dim!1).S(dim!1) m  # diffusion of S-S dimer
  >> P() m

  K1() + S(A~U) -> K1() + S(A~P) kp
  P()  + S(A~P) -> P()  + S(A~U) ku

  K1() + S(B~U) -> K1() + S(B~P) kp
  P()  + S(B~P) -> P()  + S(B~U) ku

  K1() + S(C~U) -> K1() + S(C~P) kp
  P()  + S(C~P) -> P()  + S(C~U) ku

  K1() + S(D~U) -> K1() + S(D~P) kp
  P()  + S(D~P) -> P()  + S(D~U) ku

  K1() + S(E~U) -> K1() + S(E~P) kp
  P()  + S(E~P) -> P()  + S(E~U) ku

  K2() + S(F~U) -> K2() + S(F~P) kp
  P()  + S(F~P) -> P()  + S(F~U) ku

  K2() + S(G~U) -> K2() + S(G~P) kp
  P()  + S(G~P) -> P()  + S(G~U) ku

  K2() + S(H~U) -> K2() + S(H~P) kp
  P()  + S(H~P) -> P()  + S(H~U) ku

  K2() + S(I~U) -> K2() + S(I~P) kp
  P()  + S(I~P) -> P()  + S(I~U) ku

  K2() + S(J~U) -> K2() + S(J~P) kp
  P()  + S(J~P) -> P()  + S(J~U) ku

  S(dim) + S(dim) -> S(dim!1).S(dim!1) b
```

```
S(A~P,B~P,C~P,D~P,E~P,dim!1,stable~U).S(F~P,G~P,H~P,I~P,J~P,dim!1,stable~U)
->
S(A~P,B~P,C~P,D~P,E~P,dim!1,stable~U).S(F~P,G~P,H~P,I~P,J~P,dim!1,stable~P)
stb

   S(dim!1,stable~U).S(dim!1,stable~U) -> S(dim,stable~U) + S(dim,stable~U) d

   ++ K1() kadd  in region RgnL  since 1.0  until 3.0
   ++ K2() kadd  in region RgnR  since 5.0  until 7.0
 end

 begin observables
   S_10u      S(A~U,B~U,C~U,D~U,E~U,F~U,G~U,H~U,I~U,J~U) color blue
   S_5pNterm  S(A~P,B~P,C~P,D~P,E~P,F~U,G~U,H~U,I~U,J~U) color gold
   S_5pCterm  S(A~U,B~U,C~U,D~U,E~U,F~P,G~P,H~P,I~P,J~P) color pink
   S_10p      S(A~P,B~P,C~P,D~P,E~P,F~P,G~P,H~P,I~P,J~P) color darkred
   SS_dim_stable S(dim!+,stable~P)  color red  # observing S-S dimer
   K1 K1() color black
   K2 K2() color dimgrey
   P  P()  color green
 end

 begin simulation
   time end 500
   observer intervals 100
 end
```

Listing 5: Tutorial input file 5 (doc/examples/tutorial/05-rule_based_2.spatkin).

Tutorial μmodel 6: Gradient formation

This example (Listing 6) shows how a gradient of phosphorylated substrate can be formed between enzymes tethered to different "compartments" of the reactor.

```
begin parameters
  W  200    # Arithmetic expressions and previously defined parameters
  H  W/3    #  can be used to define parameters (evaluations are always
  A  8      #  performed in double floating-point precision).
  m  3
  p  100
  q  p
end

begin world
  topology planar size W H
end

begin regions
  reservoirK  rectangle W*(    1)/(2*A) H/2 W/A H  # in-place arithmetics
  reservoirP  rectangle W*(2*A-1)/(2*A) H/2 W/A H  #
end

begin molecule types   K() S(s) P()  (* kinase, substrate, phosphatase *)  end

begin seed species
  S(s~P)  occupancy 1/8    # not inserting: S(s~U), S(s~PP)
  K()     occupancy 1/32 in region reservoirK
  P()     occupancy 1/32 in region reservoirP
end

begin event rules
  >> S() m                  # K(), P() are immobile
  K() + S(s~U) -> K() + S(s~P)  2*p
  K() + S(s~P) -> K() + S(s~PP)   p
  P() + S(s~PP)-> P() + S(s~P)  2*q
  P() + S(s~P) -> P() + S(s~U)    q
end
```

```
begin observables
  K    K()     color darkmagenta
  S_u  S(s~U)  color yellow    group S  # Grouping observables may aid
  S_p  S(s~P)  color orange    group S  #  trajectory visualization;
  S_pp S(s~PP) color red       group S  #  group "all" is always created.
  P    P()     color green
end

begin simulation
  duration 10000
  observer intervals 100
end
```

Listing 6: Tutorial input file 6 (doc/examples/tutorial/06-gradient_formation.spatkin).

Tutorial µmodel 7: Steady state controlled by diffusion

In this example (Listing 7), lattice is divided into two regions; molecules located in one of them have significantly reduced diffusivity. By visual inspection of the trajectory it can be observed that diffusion controls the steady state (i.e., the fraction of phosphorylated S molecules – black). This case has been analyzed by Szymańska *et al.* [see Figure 2(b) in *Phys. Rev. E*, 2015, **91**, 022702].

```
begin parameters
  a     120       # -- geometric parameter
  rhoK    0.1     # \
  rhoP  rhoK/10   # -> parameters used to set up initial conditions
  rhoS    0.25    # /
end

begin world
  topology plane  size 2*a a  # width==2*height
end

begin regions
  Left  rectangle a/2 a/2 a a  diffusivity 0.01
end

begin molecule types
   K()  P()  S(s)   # kinase, phosphatase, and their substrate
end

begin seed species
  K()     occupancy rhoK
  P()     occupancy rhoP
  S(s~U)  occupancy rhoS
end

begin event rules
  >> K() m   >> P() m   >> S()  100
  K() + S(s~U) -> K() + S(s~P)    1
  P() + S(s~P) -> P() + S(s~U)  100
end
```

```
begin observables
  K K()       color yellow
  P P()       color red
  Su S(s∼U)   color lightgrey
  Sp S(s∼P)   color black
end

begin settings
  time end 20
  observer intervals 200
end
```

Listing 7: Tutorial input file 7 (doc/examples/tutorial/07-diffusion_controlled_steady_state.spatkin).

Tutorial μmodel 8: Ligand-induced receptor dimerization

This is an example (Listing 8) of excitable system where introduction of trivalent ligands that collocalize bivalent receptors facilitates their activatory autotransphosphorylation. This activity is further enhanced by recruitment and activation of autotransphosphorylating kinases and opposed by phosphatases which bind to and act on both activated receptors and kinases. This example is inspired by early events in B cell receptor signaling.

```
(*
 *  This is a stochastic simulation of an excitable system, which means
 *  that the system occasionally can get activated spontaneously (due to
 *  a fluctuation). In such case, one can change random generator seed(s)
 *  and re-run the simulation.
 *)

begin parameters
  m     10.
  occuR  0.03   # receptor occupancy
  occuK  0.1    # kinase ocupancy
  occuP  0.05   # phosphatase ocupancy
end

begin world
  topology plane  size 64 64
  random seed 12345  # This seed influences initial locations of molecules.
end

begin regions
  patch  circle 32 32 10
end

begin molecule types
  K(P,K,R,A)  # kinase
  P(R,K)      # phosphatase
  Ag[3]       # extracellular antigen, trivalent
  R(P,K,A)[2] # bivalent receptor with 2 additional sites for binding K, P
end
```

```
begin seed species
  R(P,K,A~U)[@,@] occupancy occuR  # Molecules are inserted unbound.
  P(R,K)          occupancy occuP
  K(P,K,R,A~U)    occupancy occuK
end

begin event rules

  >> R() m    >> P() m    >> K() m

  # Immobile binders appear unbound in a lattice dual to the regular lattice.
  "Antigen appearance":
  ++ Ag[@,@,@]  0.01  in region patch  since 200  until 220

  "Receptor-ligand binding", "Receptor-ligand unbinding":
  +-!  R() &  Ag[]  10, 0.01

  # Receptors activation in trans:
  R()    + R(A~U) -> R()     + R(A~P)  0.01
  R()    + R(A~P) -> R()     + R(A~PP) 0.01
  R(A~P) + R(A~U) -> R(A~P)  + R(A~P)  0.02
  R(A~P) + R(A~P) -> R(A~P)  + R(A~PP) 0.02
  R(A~PP)+ R(A~U) -> R(A~PP) + R(A~P)  0.05
  R(A~PP)+ R(A~P) -> R(A~PP) + R(A~PP) 0.05

  R(P,K,A~PP) + K(P,K,R,A~U) -> R(P,K!1,A~PP).K(P,K,R!1,A~U)     1
  R(K!1,A~PP).K(P,K,R!1,A~U) -> R(K!1,A~PP).K(P,K,R!1,A~P)      10
  R(K!1).K(R!1) -> R(K) + K(R)                                   1

  K(P,K,R,A~P) + K(P,K,R,A~U)  -> K(P,K!1,R,A~P).K(P,K!1,R,A~U)  1
  K(P,K!1,R,A~P).K(P,K!1,R,A~U)-> K(P,K!1,R,A~P).K(P,K!1,R,A~P) 10
  K(K!1).K(K!1) -> K(K) + K(K)                                   1

  P(R,K) + R(P,K,A~PP) -> P(R!1,K).R(P!1,K,A~PP)                 3
  P(R,K) + R(P,K,A~P)  -> P(R!1,K).R(P!1,K,A~P)                  3
  P(R!1,K).R(P!1,K,A~PP) -> P(R!1,K).R(P!1,K,A~P)               10
  P(R!1,K).R(P!1,K,A~P)  -> P(R!1,K).R(P!1,K,A~U)               10
  P(R!1).R(P!1) -> P(R) + R(P)                                   1
```

```
  P(R,K) + K(P,K,R,A~P)   -> P(R,K!1).K(P!1,K,R,A~P)              1
  P(R,K!1).K(P!1,K,R,A~P) -> P(R,K!1).K(P!1,K,R,A~U)             10
  P(K!1).K(P!1) -> P(K) + K(P)                                    1

end

begin observables
  Ag_tot          Ag[@!?,@!?,@!?]  color black
  Receptor_U      R(A~U)  color lightgrey
  Receptor_P      R(A~P)  color grey
  Receptor_PP     R(A~PP) color brown
  Kinase_inactive K(A~U)  color gold
  Kinase_active   K(A~P)  color red
  Phosphatase     P()      color green
end

begin simulation
  time end 500
  observer intervals 1000
  snapshots on       # If you do not want snapshots, write: snapshots off
  random seed 12345  # This seed influences the order of events.
end
```

Listing 8: Tutorial input file 8 (doc/examples/tutorial/08-receptors_and_ligands.spatkin).

Tutorial μmodel 9: Crowding-facilitated switch in a bistable system

In this example (Listing 9), kinetics of a bistable system is simulated. In a defined instant, chemically inert molecules ("crowders") are introduced which leads to reduction of reactants' diffusivity. In this system, the presence of crowders favors processive rather than distributive phosphorylation, and in this way favors the steady state of a high amount of doubly phosphorylated kinases. Initially the system is in the steady state of a low amount of phosphorylated kinases; upon recruitment of crowders to the membrane, a transition to the other steady state is observed.

The effect of the presence of crowding molecules and their diffusivity on the effective diffusivity of (other) molecules on the lattice has been analyzed by Szymańska *et al*. [see Figure 9 in *Phys. Rev. E*, 2015, **91**, 022702].

```
(*
 *  This is a stochastic simulation of a bistable system, which means that
 *  the system may occasionally switch to another state (though it is very
 *  implausible). In such case, one can change random generator seed(s) and
 *  re-run the simulation.
 *)

begin parameters
  rhoK  0.4
  rhoP  0.1
  d     1    / (6 * rhoP)
  c1    0.02 / (6 * rhoK)
  c2    0.15 / (6 * rhoK)
  c3    4    / (6 * rhoK)
  m     300
end

begin world
  topology planar
  size 50 50
  random seed 123
end

begin regions
end
```

```
begin molecule types
  K(a)  # self-activating kinase
  P()   # phosphatase acting on the kinase
  C()   # crowder
end

begin seed species
  P()     occupancy rhoP
  K(a~U)  occupancy rhoK
end

begin event rules

  ## Rules for reactants:
  #
  >> K() m   >> P() m

  K(a~U) + K(a~U) -> K(a~U) + K(a~P)   2*c1
  K(a~U) + K(a~P) -> K(a~U) + K(a~PP)    c1

  K(a~P) + K(a~U) -> K(a~P) + K(a~P)   2*c2
  K(a~P) + K(a~P) -> K(a~P) + K(a~PP)    c2

  K(a~PP) + K(a~U) -> K(a~PP) + K(a~P) 2*c3
  K(a~PP) + K(a~P) -> K(a~PP) + K(a~PP)  c3

  P() + K(a~P)  -> P() + K(a~U)  d
  P() + K(a~PP) -> P() + K(a~P)  2*d

  ## Rules for crowders:
  #
  ++ C()  0.15  since 36  until 40
  >> C()  m/10
end

begin observables
  K    K(a~U)  color yellow
  K_p  K(a~P)  color orange
```

```
  K_pp K(a~PP) color red
  P    P()     color lime
  C    C()     color dimgray  # crowder
end

begin simulation
  duration 100
  observer intervals 1000
  random seed 123
  snapshots on
end
```

Listing 9: Tutorial input file 9 (doc/examples/tutorial/09-crowding_facilitated_switch.spatkin).

Tutorial μmodel 10: Traveling wave

The considered system contains phosphatases and auto-phosphorylating kinases reacting in a long cylindrical domain. This prototypical bistable system is the subject of the analysis described by Zuk *et al.* [*Phys. Biol.*, 2012, **5**, 055002] and Kochańczyk *et al.* [*J. R. Soc. Interface*, 2013, **10**, 20130151], where the concordance of particle-based simulations in SPATKIN and finite-element method-based simulations of a corresponding partial differential equation system is demonstrated.

```
(*
 *  Please note that the simulation can take about two hours. Occasionally,
 *  the stochastic traveling wave may fail to propagate or spontaneous self-
 *  activation may occur in another part of the lattice -- in such case one
 *  can change random number generator seed(s) and re-run the simulation.
 *)
begin parameters
  n_stat_K    183  #  |
  n_stat_Kp   571  #  > high-phospholevel steady state (calculated from ODEs)
  n_stat_Kpp  439  # |
  rhoK  0.4
  rhoP  0.1
  d     1    / (6 * rhoP)
  c1    0.02 / (6 * rhoK)
  c2    0.18 / (6 * rhoK)
  c3    4    / (6 * rhoK)
  m   1000
end

begin world
  topology planar  size 404 30  random seed 123456789
end

begin regions
  Barrier   rectangle  402  15    4 30  diffusivity 0  # reflective boundary
  Ignition  rectangle   50  15  100 30
  Rest      rectangle  250  15  300 30
end

begin molecule types   K(a)  P()   end
```

```
begin seed species
  P()      occupancy  rhoP   in region Ignition
  K(a~U)   n_stat_K          in region Ignition
  K(a~P)   n_stat_Kp         in region Ignition
  K(a~PP)  n_stat_Kpp        in region Ignition
  P()      occupancy  rhoP    in region Rest
  K(a~U)   occupancy  rhoK   in region Rest
end

begin event rules
  K(a~U)  + K(a~U) -> K(a~U)  + K(a~P)   2*c1
  K(a~U)  + K(a~P) -> K(a~U)  + K(a~PP)    c1
  K(a~P)  + K(a~U) -> K(a~P)  + K(a~P)   2*c2
  K(a~P)  + K(a~P) -> K(a~P)  + K(a~PP)    c2
  K(a~PP) + K(a~U) -> K(a~PP) + K(a~P)   2*c3
  K(a~PP) + K(a~P) -> K(a~PP) + K(a~PP)    c3
  P() + K(a~P)  -> P() + K(a~U)   d
  P() + K(a~PP) -> P() + K(a~P) 2*d
  >> K() m
  >> P() m
end

begin observables
  K    K(a~U)   color yellow
  K_p  K(a~P)   color orange
  K_pp K(a~PP)  color red
  P    P()       color lime
end

begin simulation
  description "Induced chemical travelling wave"
  duration 100   observer intervals 200
  random seed 987654321
end
```

Listing 10: Tutorial input file 10 (doc/examples/tutorial/10-traveling_wave.spatkin).