Modeling the shedding of EGFR ligands

The model is built for describing the shedding of EGFR ligands and will be integrated into the EGFR signaling network.

The current shedding model:

This model current consists of three parts: Ligands expression and internalization as well recycling; LPA activation of shedding; the main shedding module.

Main shedding module:

Here are what we know: 1) iRhom

What are needed:

1. Previous experimental data that helps to determine the kinetics of different components and estimate parameters
2. Visualization of the detail molecular mechanisms and hypothesis.
3.

TODOs:

20200228: Send Steve the notes about the shedding model

20200221:

1. Incoporate the expression of ligands (Tgfaa)
2. Add the internalization and recycling of memberane ligands

[2]
```
import numpy as np

import tellurium as te
te.setDefaultPlottingEngine('matplotlib')
#te.setDefaultPlottingEngine("plotly")
```

[3]
```
# Shedding module:
model shedding()
    # Activated ERK (pERK) binds to iRhom-TACE complex and
phosphorylate iRhom
    ErkiRhomBind: pErk + iRhomTace -> pErkiRhomTace; kfErkiRhom * pErk
* iRhomTace - kbErkiRhom * pErkiRhomTace
    ErkPhoiRhom: pErkiRhomTace -> pErk + iRhompTace; kpErkiRhom *
pErkiRhomTace
```

```
    # Phosphorylated iRhom binds to 14-3-3 then TACE dissociate from
iRhom
    iRhomp1433Bind: iRhompTace + Pro1433 -> iRhompPro1433Tace;
kfiRhomp1433 * iRhompTace * Pro1433 - kbiRhomp1433 * iRhompPro1433Tace
    iRhompTaceDis: iRhompPro1433Tace -> iRhompPro1433 + Tace; kDisTace
* iRhompPro1433Tace
    # TACE catalyze the ligands at membrane into soluble ligands
    TaceShed: Tace + mTgfa -> Tace + Tgfa; kShedTace * Tace * mTgfa
    # Expression, internalization and degradation of iRhom-TACE
complex, TACE, iRhomP-14-3-3 complex, 14-3-3 proteins
    iRhomTaceExp:  -> iRhomTace; kExpiRhomTace
    Pro1433Exp:  -> Pro1433; kExp1433
    Pro1433Deg: Pro1433 -> ; kDeg1433 * Pro1433
    iRhomp1433Intern: iRhompPro1433 ->  ; kIniRhomp1433 * iRhompPro1433
    TgfaDeg: Tgfa -> ; kDegTgfa * Tgfa
    TaceIntern: Tace -> ; kInTace * Tace
    iRhomTaceIntern: iRhomTace ->  ; kIniRhomTace * iRhomTace
    iRhompTaceIntern: iRhompTace -> ; kIniRhompTace * iRhompTace
    # Phosphorylation of iRhom by Lpa
    LpaActiRhom: Lpa + iRhomTace -> Lpa + iRhompTace; kpLpaiRhom * Lpa
* iRhomTace

end

# ERK activation module:
model erkAct()
    # ERK activation based on the available extracellular ligands
    TgfaActErk: Tgfa + Erk -> Tgfa + pErk; (basal + kAct * (Tgfa /
KD)^n / (1 + (Tgfa / KD)^n) ) * Erk
    ErkDeact: pErk -> Erk; kDephoErk * pErk
    #ErkDeg: pErk -> ; kDegErk * pErk
end

# Ligand expression module:
model tgfExp()
    # TGFa transcription with activated ERK
    TgfaTrc: pErk -> pErk + mRna_Tgfa; basal + kTrc * (pErk / KD)^n /
(1 + (pErk / KD)^n)
    TgfaRnaDeg: mRna_Tgfa -> ; kDegRna * mRna_Tgfa
    # TGFa translation
    TgfaTrl: mRna_Tgfa -> mRna_Tgfa + mTgfa; kTrl * mRna_Tgfa
    # mTGFa transportation
    mTgfaIntern: mTgfa -> mTgfaIn; kIn * mTgfa
    mTgfaRecyc: mTgfaIn -> mTgfa; kRecyc * mTgfaIn
    mTgfaDeg: mTgfaIn -> ; kDegLys * mTgfaIn
end

model sheddingTest

    var species Tgfa
    var species pErk
    var species mTgfa
```

```
    var species Lpa

    pErk = 0.0;
    Tgfa = 0.0;
    mTgfa = 0.0;
    Lpa = 0.0;

    erk: erkAct();
    erk.Tgfa is Tgfa
    erk.pErk is pErk
    erk.basal = 0.0; erk.kAct = 0.1; erk.KD = 1; erk.n = 2;
erk.kDephoErk = 0.01; #erk.kDegErk = 0.01;
    erk.Erk = 1000;


    tgf: tgfExp();
    tgf.mTgfa is mTgfa
    tgf.pErk is pErk
    tgf.basal = 0.0; tgf.kTrc = 1; tgf.KD = 1; tgf.n = 2; tgf.kDegRna
= 0.1; tgf.kTrl = 1; tgf.kIn = 0.001; tgf.kRecyc = 0.01; tgf.kDegLys =
0.02;

    shed: shedding();
    shed.mTgfa is mTgfa
    shed.Tgfa is Tgfa
    shed.pErk is pErk
    shed.Lpa is Lpa

    shed.kfErkiRhom = 0.01; shed.kbErkiRhom = 0.1; shed.kpErkiRhom =
0.1; shed.kfiRhomp1433 = 0.001; shed.kbiRhomp1433 = 0.1; shed.kDisTace
= 0.1; shed.kShedTace = 0.1; shed.kExpiRhomTace = 0.001;
shed.kIniRhomp1433 = 0.05; shed.kExp1433 = 1; shed.kDegTgfa = 0.01;
shed.kInTace = 0.01; shed.kIniRhomTace = 0.01; shed.kpLpaiRhom = 0.01;
shed.kDeg1433 = 0.002; shed.kIniRhompTace = 0.005;

end
```

```
Success: Model can be accessed via variable sheddingTest
```

[4]

```
sheddingTest.draw(width=300)
```

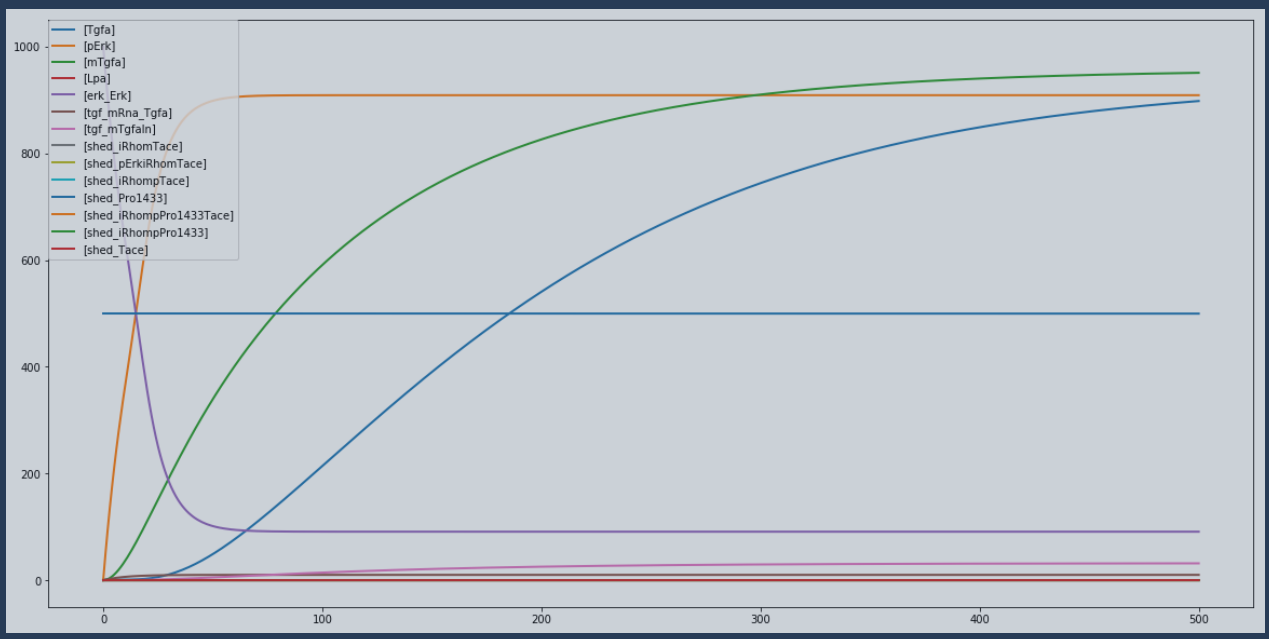
```
/Users/feng626/Library/Application Support/Tellurium/telocal/python-
3.6.3/lib/python3.6/site-packages/ipykernel/__main__.py:1: Warning:

Graphviz is not installed in your machine or could not be found. 'draw'
command cannot produce a diagram.
```

[9]

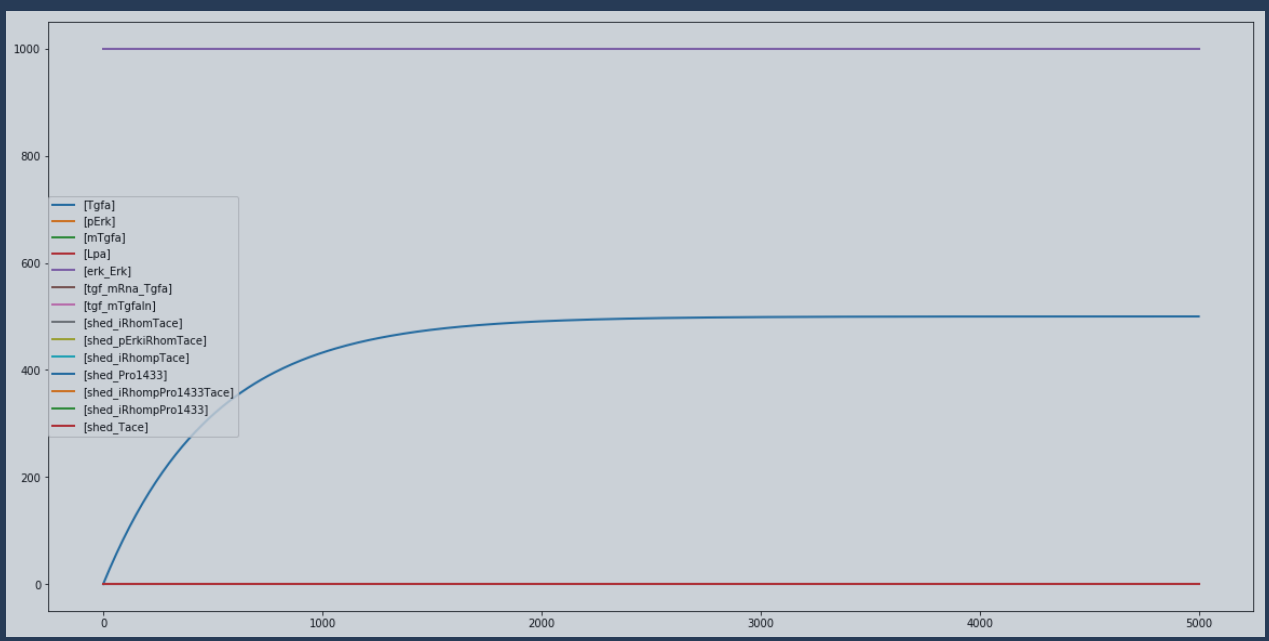
```
sheddingTest.reset()
```

```
sheddingTest.Tgfa = 0.0
sheddingTest.simulate(0,5000,500)
#sheddingTest.plot()
sheddingTest.Tgfa = 1.0
sheddingTest.simulate(0,500,500)
sheddingTest.plot(figsize=(20,10))
```



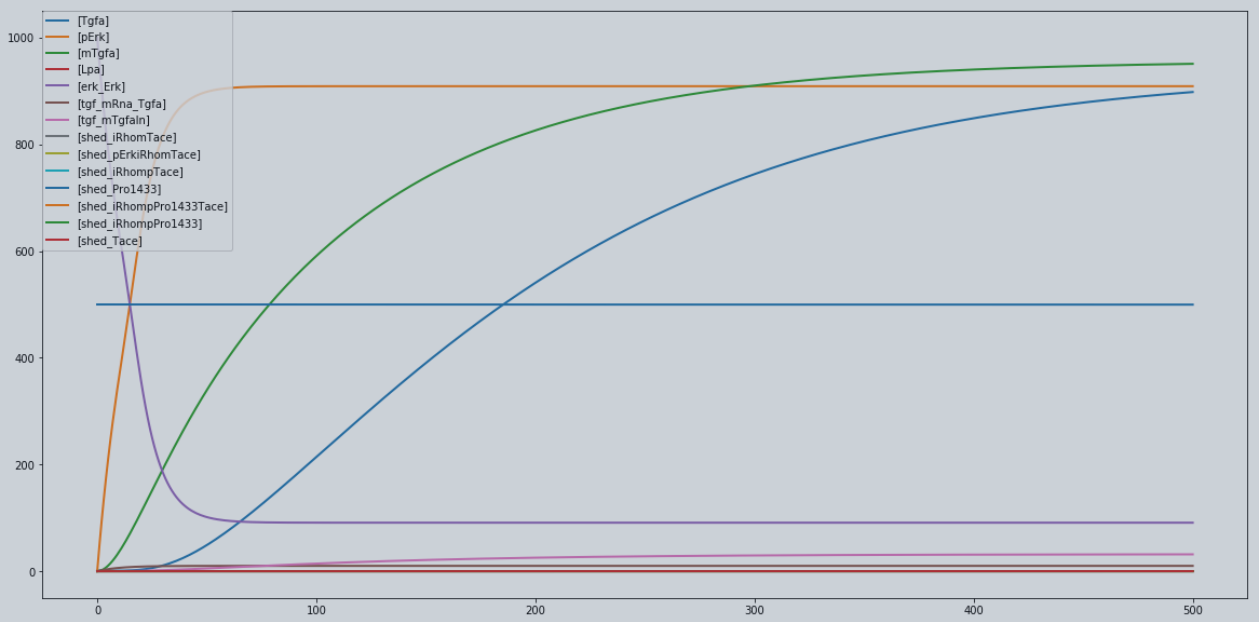
[8]

```
sheddingTest.reset()
sheddingTest.Tgfa = 0.0
sheddingTest.simulate(0,5000,500)
sheddingTest.plot(figsize=(20,10))
```



[7]

```
sheddingTest.Tgfa = 1.0
sheddingTest.simulate(0,500,500)
sheddingTest.plot(figsize=(20,10))
```


[Tgfa]
[pErk]
[mTgfa]
[Lpa]
[erk_Erk]
[tgf_mRna_Tgfa]
[tgf_mTgfaln]
[shed_iRhomTace]
[shed_pErkiRhomTace]
[shed_iRhompTace]
[shed_Pro1433]
[shed_iRhompPro1433Tace]
[shed_iRhompPro1433]
[shed_Tace]
1000
800
600
400
200
0
0
100
200
300
400
500

[8]

[ ]