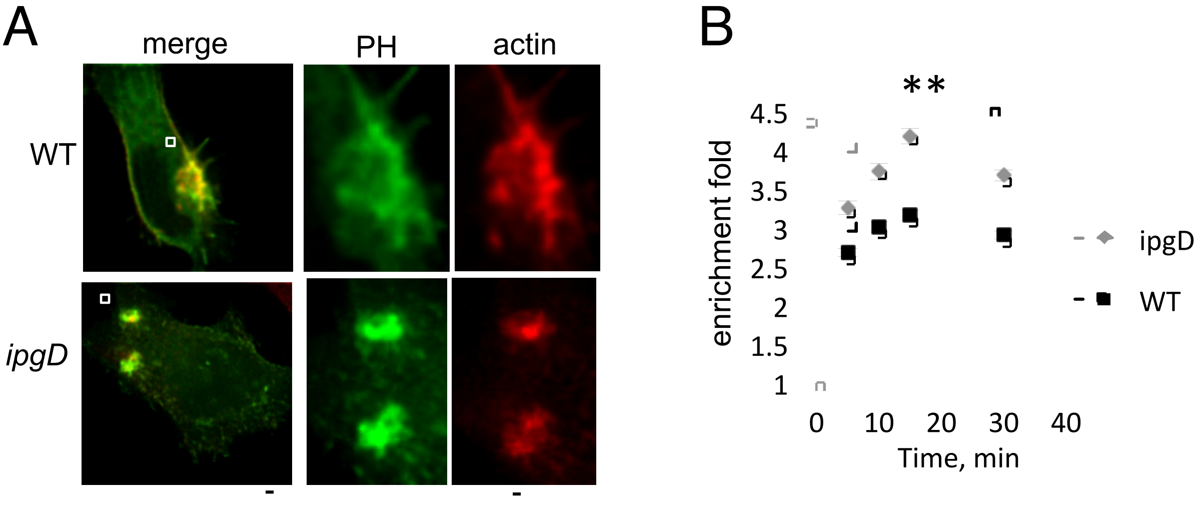
Appendix

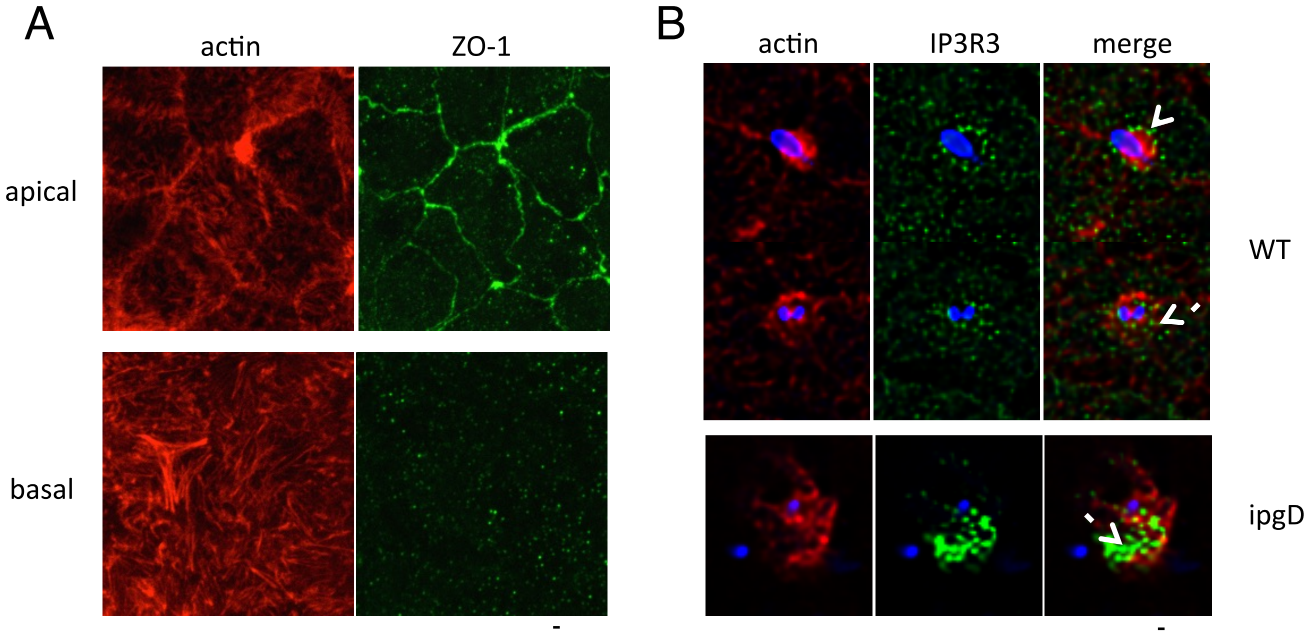
Table of contents



Appendix Fig S1. IpgD down-regulates $PI(4,5)P_2$ levels at *Shigella* invasion sites.

A Representative confocal fluorescence images of cells transfected with the $\text{GFP-PH}_{\text{PLC}\delta}$ probe, challenged with the indicated bacterial strains for 15 min at 37°C and processed for fluorescence staining. Red: actin; green: $\text{GFP-PH}_{\text{PLC}\delta}$. Scale bar = 5 µm.

B The enrichment fold of $PI(4,5)P_2$ at invasion foci ± SEM was determined from the levels of the $\text{GFP-PH}_{\text{PLC}\delta}$ probe at the indicated time points (Materials and Methods). WT *Shigella*: black squares bars; *ipgD* mutant: empty diamonds. N = 3, > 30 foci per sample. Wilcoxon sum rank test, **: $p < 0.01$

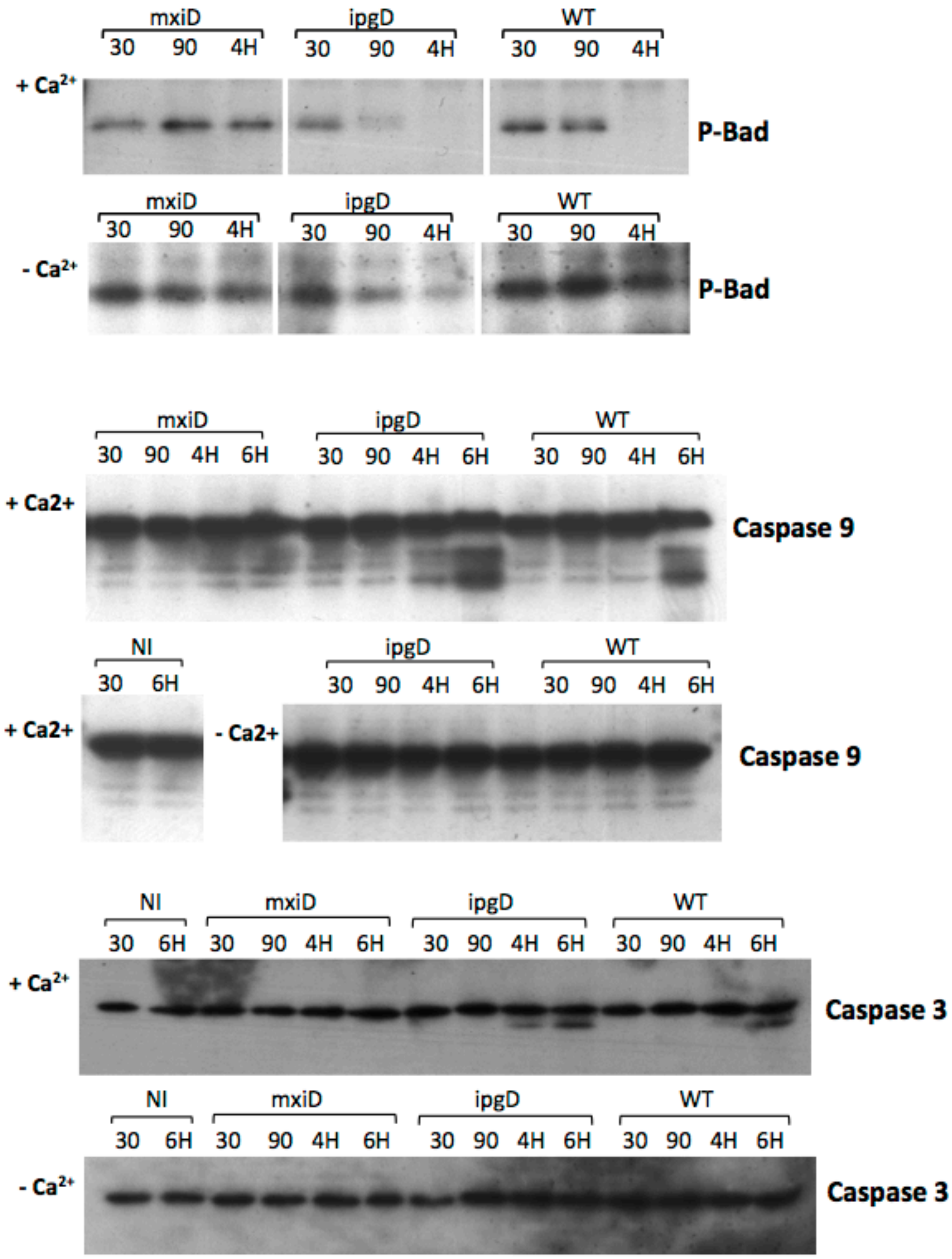


Appendix Fig S2. IpgD down-regulates the recruitment of IP3R3 at invasion sites in polarized intestinal epithelial cells.

Representative confocal fluorescence micrographs of polarized TC7 cells fixed and processed for immunofluorescence staining.

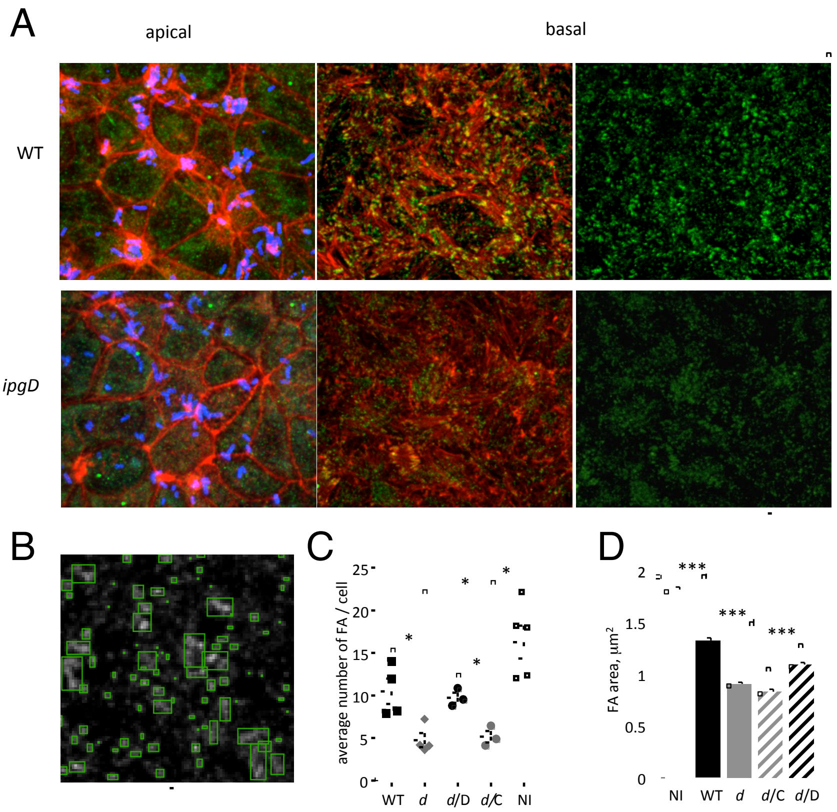
A Projection of confocal planes corresponding to the apical or basal surfaces of the same field. Red: actin; Green: ZO-1. Scale bar = 5 μm.

B TC-7 cells were challenged with the indicated bacterial strains for 20 min at 37°C and processed for immunofluorescence staining. Green: IP3R3; red: actin; blue: bacterial LPS. Scale bar = 5 μm. IP3R3-contanining granules are detected entrapped (arrows) in- or surrounding (arrowhead) actin foci.



Appendix Fig S3. Effects of IpgD on the Ca^{2+}-independent phosphorylation of BAD and Ca^{2+}-dependent degradation of caspases 9 and 3.

Cells were challenged with the indicated bacterial strains. At indicated times, cells were scraped and lysates were analysed by anti-phospho-BAD, -caspase 9, or -caspase 3, Western blotting. Cells were incubated in buffer in the presence (+Ca^{2+}) or absence of 2 mM EGTA (-Ca^{2+}).



Appendix Fig S4. IpgD delays FA disassembly during *Shigella* invasion of polarized intestinal epithelial TC-7 cells.

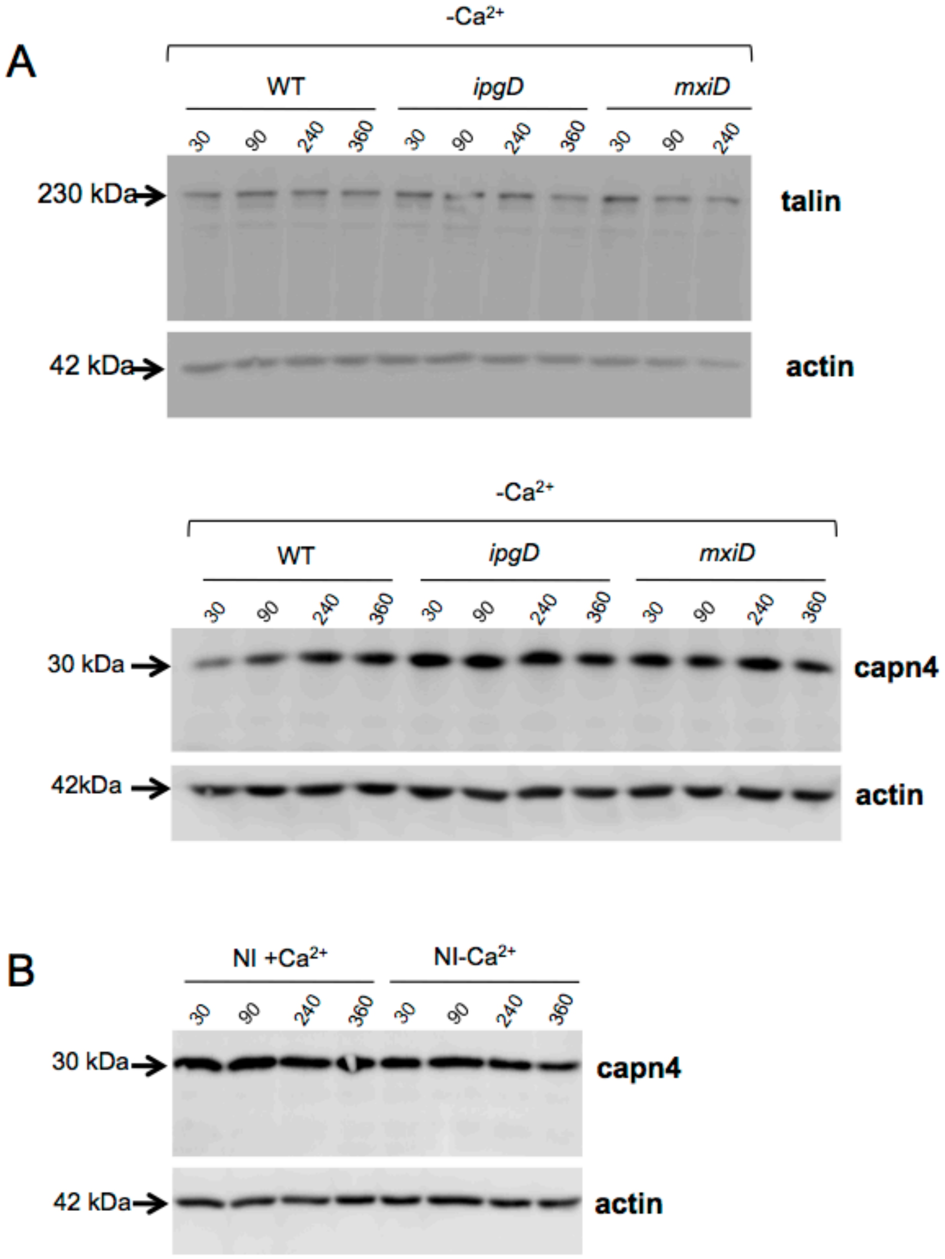
Polarized TC-7 cells were challenged with the indicated bacterial strains for 90 min. Samples were fixed and processed for immuno-fluorescence staining. Red: actin; green: talin; blue: bacterial LPS. Scale bar = 5 µm.

A Representative confocal micrographs of the apical or basal regions of the same field, with a Z-interspacing of 11 µm. Note that at this time point, while bacteria were mainly detected at the cell apical region, different effects on FAs were observed in the cell basal region.

B Talin-containing FAs were scored using an automated algorithm (Materials and Methods). Scored FAs are detected in green boxes in the representative image. The FA average size was determined for NI: non-infected cells, or cells infected with WT: wild-type *Shigella*; *d*: *ipgD* mutant; *d*/C: *ipgD*/IpgDC438S; *d*/D: *ipgD*/IpgD

C The average number of FAs / cell ± SEM was determined for cells infected with the indicated bacteria as in B. Each value corresponds to a determination in a distinct field. For each sample, N = 2, > 50 cells, > 3500 FAs. Mann-Whitney test, *: p value for: WT vs ipgD = 0.0143; *d*/D vs *d*/C = 0.05; *ipgD* vs NI = 0.0079 ≤ 0.005; *d*/C vs NI = 0.0179.

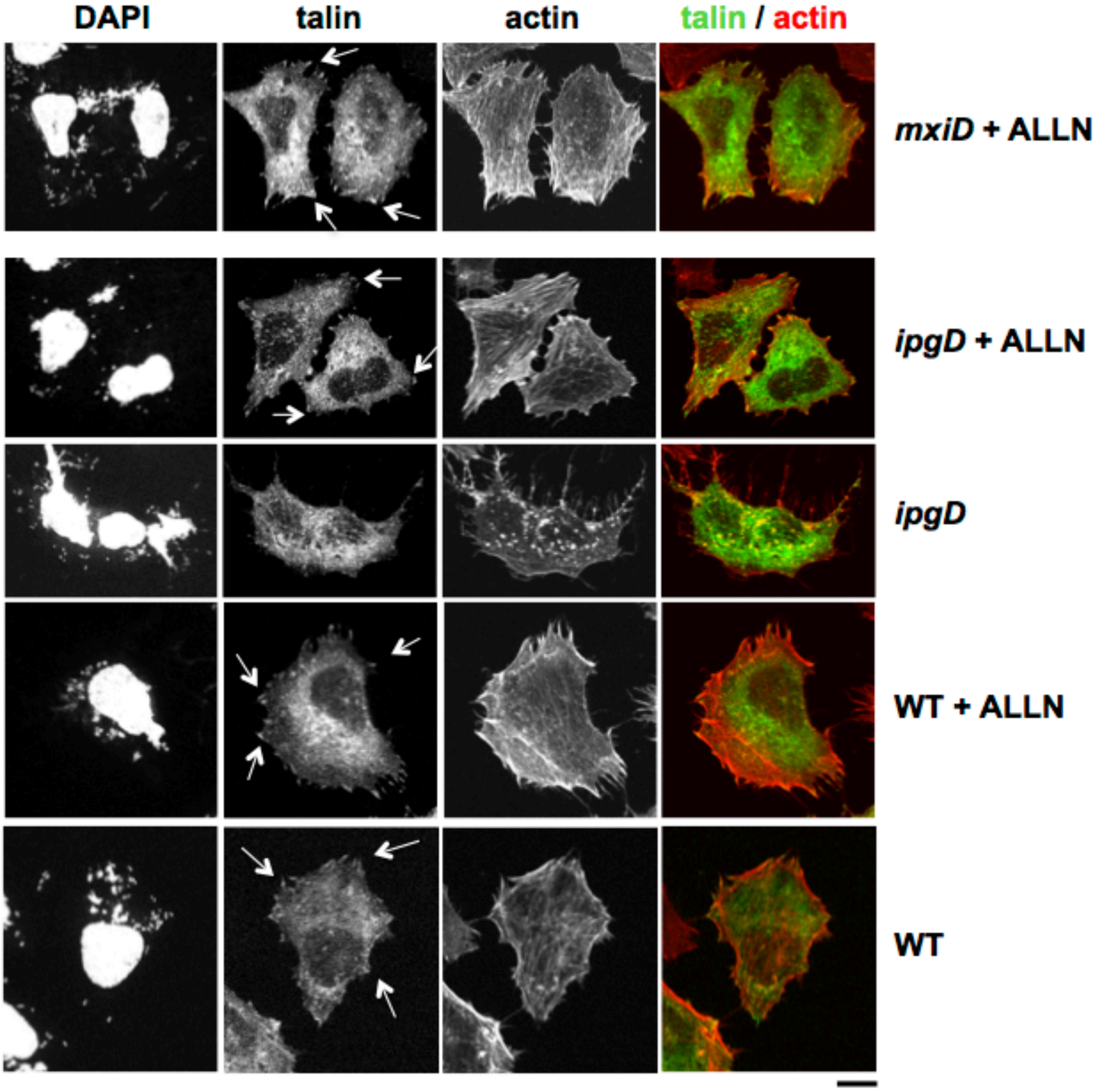
D The average cell area ± SEM was determined for cells infected with the indicated bacteria as in B. One-way Anova test, N = 2, > 30 cells. ***: $p < 0.001$



Appendix Figure S5. Talin degradation and calpain activation are inhibited in the absence of extracellular Ca^{2+} during bacterial infection.

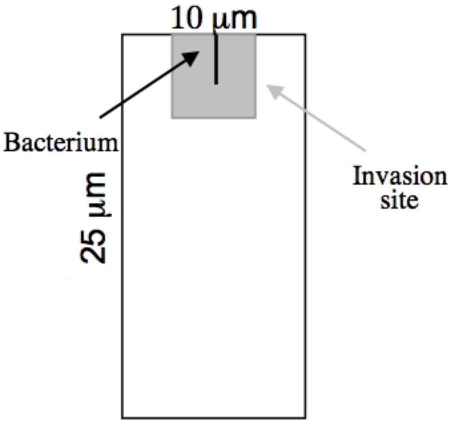
A Cells were challenged with the indicated bacterial strains for the time indicated in minutes in the presence of 2 mM EGTA (-Ca^{2+}). At indicated times, cells were scraped and lysates were analysed by anti-talin or anti-capn4 Western blotting.

B Anti-capn4 Western-blotting analysis of non-infected (NI) control cells incubated for the time indicated in minutes in the presence (+Ca^{2+}) or absence of 2 mM EGTA (-Ca^{2+}).



Appendix Figure S6. The calpain inhibitor ALLN prevents FA disassembly during Shigella infection.

Talin-GFP transfected cells were pre-incubated with the calpain inhibitor ALLN at a final concentration of 10 μM for 60 min prior to challenge with bacteria for 30 min at 37°C and an additional 60 min in the presence of gentamicin at a final concentration of 50 μg/ml. Samples were fixed and processed for immunofluorescence staining of DNA (DAPI) and actin. Representative micrographs of cells challenged with the indicated bacterial strains. The arrows point to FAs. Scale bar, 5 μm.



	D_{Focus}/D_{Cyto}	J_{IP}/V_b
WT	0.62	12.50
ipgD	0.77	16.25

Appendix Figure S7. Geometry and parameters used in the simulations of Ca^{2+} responses induced by *Shigella* during cell invasion.

The cell is simulated as a 2-dimensional 26 x 10 µm system. The *Shigella* invasion focus corresponds to the 6 x 6 µm top central region (grey square). The bacterium is schematized as the black rod. The ratios of the diffusion coefficients of $InsP_3$ and of Ca^{2+} in the focus and in the cytosol (D_{Focus}/D_{Cyto}) and of the rates of $InsP_3$ synthesis at the invasion site relative to the rest of the cell (J_{IP}/V_b) are indicated in the Table. Ca^{2+} responses are considered as global if the peak Ca^{2+} concentration in the middle of the cell exceeds the third of the peak Ca^{2+} response at the invasion site.

Stochastic simulations of *Shigella*-induced Ca2+ responses

```
#include <iostream>

#include <math.h>

#include <cmath>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <vector>

#include <fstream>

using namespace std;

int main()

{

srand(time(0));


//////////////////////////////////////

/////Parameters for Ca2+ dynamics/////

//////////////////////////////////////


double const alpha(0.05), kplus(0.1822), kmoins(0.005),
kact(0.4),ni(3),na(2),b(0.007),vmp(1.2),kp(0.35),k(1),kdeg(0.1),vip(0.1);


////////////////////////////

/////Size of the system/////

////////////////////////////


int const N1(13),N2(5),NN(N1*N2);
```

```
double const  omega(10000/(N1*N2)),Rtot(6000/(N1*N2));


///////////////////////////////////////////////////////////////
/////Some parameters are not homogeneous in the whole cell/////
///////////////////////////////////////////////////////////////


double jip[NN];


for(int i(0);i<NN;i++)
{
   jip[i]=0;
}


jip[N1]=1.25;
jip[N1+1]=1.25;
jip[N1+2]=1.25;
jip[2*N1+1]=1.25;
jip[2*N1+2]=1.25;
jip[2*N1]=1.25;
jip[3*N1]=1.25;
jip[3*N1+2]=1.25;
jip[3*N1+1]=1.25;


double k1[NN];
for(int i(0);i<NN;i++)
```

```
{
   k1[i]=35;
}
k1[N1]=60;
k1[N1+2]=60;
k1[N1+1]=60;
k1[2*N1+1]=60;
k1[2*N1+2]=60;
k1[2*N1]=60;
k1[3*N1+1]=60;
k1[3*N1+2]=60;
k1[3*N1]=60;
double Di[NN];
for(int i(0);i<NN;i++)
{
   Di[i]=280/4;
}
Di[N1+1]=175/4;
Di[N1+2]=175/4;
Di[N1]=175/4;
Di[2*N1+1]=175/4;
Di[2*N1+2]=175/4;
Di[2*N1]=175/4;
Di[3*N1+1]=175/4;
Di[3*N1+2]=175/4;
```

```
Di[3*N1]=175/4;

double D[NN];

for(int i(0);i<NN;i++)

{

   D[i]=30/4;

}

D[N1+1]=19/4;

D[N1+2]=19/4;

D[N1]=19/4;

D[2*N1+1]=19/4;

D[2*N1+2]=19/4;

D[2*N1]=19/4;

D[3*N1+1]=19/4;

D[3*N1+2]=19/4;

D[3*N1]=19/4;


////////////////////////////

/////Initial Conditions/////

////////////////////////////


 double I0(round(1.6*omega));

double I[NN];

for(int i(0);i<NN;i++)

{

   I[i]=I0;
```

```
}

double C0(round(0.064153*omega));

double C[NN];

for(int i(0);i<NN;i++)

{

   C[i]=C0;

}

double R0(round(0.0092926*Rtot));

double R[NN];

for(int i(0);i<NN;i++)

{

   R[i]=R0;



}


/////////////////////////////////////////////

/////Parameters for Gillespie's algorithm/////

/////////////////////////////////////////////


double const trans(0),tend(800),tech(0.1);

double t(0),told(0);

double w[NN*15];

for(int i(0);i<NN*15;i++)

{
    w[i]=0;
}


string V("test2.dat");
ofstream myfile;
myfile.open (V.c_str());



double IR[NN];


///////////////////////////////
/////Gillespie's algorithm/////
///////////////////////////////


while (t<tend+trans)
{
////////////////////////////////
/////Calculing propensities/////
////////////////////////////////


    for(int i(0); i<NN;i++)
    {


        IR[i]=(Rtot-
R[i])*(I[i]/(I[i]+k*omega))*pow(C[i],na)/(pow(C[i],na)+pow((kact*omega),na));
```

```
    w[i]=alpha*b*(k1[i]*omega);
    w[i+NN*1]=alpha*(k1[i]*omega)*IR[i]/Rtot;
    w[i+NN*2]=vmp*omega*pow(C[i],2)/(pow(C[i],2)+pow(kp*omega,2));
    w[i+NN*3]=kmoins*R[i];

    w[i+NN*4]=(kplus/pow(omega,3))*pow(C[i],ni)*(Rtot-
R[i])*pow(kact*omega,na)/(pow(C[i],na)+pow(kact*omega,na));
    w[i+NN*5]=0;
    w[i+NN*6]=0;
    w[i+NN*7]=0;
    w[i+NN*8]=0;
    w[i+NN*11]=0;
    w[i+NN*12]=0;
    w[i+NN*13]=0;
    w[i+NN*14]=0;
    w[i+NN*9]=jip[i]*omega+vip*omega;
    w[i+NN*10]=kdeg*I[i];
  }
  for (int i(0);i<N1*(N2-1);i++)
  {
    if (I[i+N1]<I[i])
      w[i+NN*13]=D[i]*(I[i]-I[i+N1]);
    if (C[i+N1]<C[i])
      w[i+NN*7]=D[i]*(C[i]-C[i+N1]);
    if (I[i]<I[i+N1])
      w[i+N1+NN*14]=D[i+N1]*(I[i+N1]-I[i]);
```

```
        if (C[i]<C[i+N1])
           w[i+N1+NN*8]=D[i+N1]*(C[i+N1]-C[i]);
    }
for (int i(0);i<N1-1;i++)
    {
        for (int j(0); j<N2;j++)
        {
           if (I[N1*(j)+i]< I[N1*(j)+i+1])
           {
              w[N1*(j)+i+1+NN*11]=D[i+1]*(I[N1*(j)+i+1]-I[N1*(j)+i]);
           }

           if (C[N1*(j)+i]< C[N1*(j)+i+1])
           {
              w[N1*(j)+i+1+NN*5]=D[i+1]*(C[N1*(j)+i+1]-C[N1*(j)+i]);
           }

           if (I[N1*(j)+i+1]< I[N1*(j)+i])
           {
              w[N1*(j)+i+NN*12]=D[i]*(I[N1*(j)+i]-I[N1*(j)+i+1]);
           }
           if (C[N1*(j)+i+1]< C[N1*(j)+i])
           {
              w[N1*(j)+i+NN*6]=D[i]*(C[N1*(j)+i]-C[N1*(j)+i+1]);
           }
        }
     }
////////////////////////////////////////
/////Choosing which reaction occurs/////
////////////////////////////////////////
 for (int i(1);i<NN*15;i++)
 {
    w[i]=w[i-1]+w[i];
 }
   double ct(0),z1(0),z2(0),tau(0),uct(0);
   int bip(1);
   ct=w[NN*15-1];
   z1=(rand()/(double)RAND_MAX)+0.0000001 ;
   z2=rand()/(double)RAND_MAX;
   tau=(-log(z1))/ct;
   t=t+tau;
   uct=z2*ct;
   bip=0;

   if (uct<w[NN-1])
   {
      while (uct>w[bip])
      {
         bip=bip+1;
      }
    C[bip]=C[bip]+1;
}
else if (uct<w[2*NN-1])
{
    while (uct>w[NN+bip])
    {
        bip=bip+1;
    }
    C[bip]=C[bip]+1;
}
else if (uct<w[3*NN-1])
{
    while (uct>w[NN*2+bip])
    {
        bip=bip+1;
    }
    C[bip]=C[bip]-1;
}
else if (uct<w[4*NN-1])
{
    while (uct>w[NN*3+bip])
    {
        bip=bip+1;
    }
    R[bip]=R[bip]-1;
```

```
}
else if (uct<w[5*NN-1])
{
   while (uct>w[NN*4+bip])
   {
      bip=bip+1;
   }
   R[bip]=R[bip]+1;
}
else if (uct<w[6*NN-1])
{
   while (uct>w[NN*5+bip])
   {
      bip=bip+1;
   }
   C[bip]=C[bip]-1;
   C[bip-1]=C[bip-1]+1;
}
else if (uct<w[7*NN-1])
{
   while (uct>w[NN*6+bip])
   {
      bip=bip+1;
   }
   C[bip]=C[bip]-1;
```

```
    C[bip+1]=C[bip+1]+1;
}
else if (uct<w[8*NN-1])
{
    while (uct>w[NN*7+bip])
    {
        bip=bip+1;
    }
    C[bip]=C[bip]-1;
    C[bip+N1]=C[bip+N1]+1;
}
else if (uct<w[9*NN-1])
{
    while (uct>w[NN*8+bip])
    {
        bip=bip+1;
    }
    C[bip]=C[bip]-1;
    C[bip-N1]=C[bip-N1]+1;
}
else if (uct<w[10*NN-1])
{
    while (uct>w[NN*9+bip])
    {
        bip=bip+1;
    }
    I[bip]=I[bip]+1;
}
else if (uct<w[11*NN-1])
{
    while (uct>w[NN*10+bip])
    {
        bip=bip+1;
    }
    I[bip]=I[bip]-1;
}
else if (uct<w[12*NN-1])
{
    while (uct>w[NN*11+bip])
    {
        bip=bip+1;
    }
    I[bip]=I[bip]-1;
    I[bip-1]=I[bip-1]+1;
}
else if (uct<w[13*NN-1])
{
    while (uct>w[NN*12+bip])
    {
        bip=bip+1;
```

```
      }
      I[bip]=I[bip]-1;
      I[bip+1]=I[bip+1]+1;
   }
   else if (uct<w[14*NN-1])
   {
      while (uct>w[NN*13+bip])
      {
         bip=bip+1;
      }
      I[bip]=I[bip]-1;
      I[bip+N1]=I[bip+N1]+1;
   }
   else
   {
      while (uct>w[NN*14+bip])
      {
         bip=bip+1;
      }
      I[bip]=I[bip]-1;
      I[bip-N1]=I[bip-N1]+1;
   }

/////////////////////
/////Saving data/////
```

```
/////////////////////

    if (t>told+tech)
        {

            told=t;
            myfile << t << "\t";
            for (int i(0);i<NN;i++)
            {
                myfile << C[i] << "\t";
            }
            for (int i(0);i<NN;i++)
            {
                myfile << R[i] << "\t";
            }
            for (int i(0);i<NN;i++)
            {
                myfile << I[i] << "\t";
            }
            myfile << "\n";
            }
myfile.close();
}
    return 0;
}
```