Bayesian inference GW1 data

In[∘]:=

```
ClearAll;
Clear["Global`*"]
```

Import data

```
lcol = 6; lrow = 0;
a = {0.1, 0., 0.2, 0.7} (*true values: stem cells*)
b = {0.2, 0.8, 0., 0.}  (*true values: differentiated cells*)
Zt = Import[
   "C:\\Users\\mw\\Documents\\dottorato\\branching_process_programs\\GW_type2\\
     GW_type2_data\\GW2_Zs_gen8_num_sim20_a[0.1 0.  0.2 0.7 0.
     0. ]_b[0.2 0.8 0.  0.  0.  0. ].txt", "Table", Delimiter → ","];
Zt = Zt⟦1 ;; Dimensions[Zt]⟦1⟧ - lrow, 1 ;; Dimensions[Zt]⟦2⟧ - lcol⟧;
(* select part of the dataset*)
Print["dimension matrix Zt: ", Dimensions[Zt]]
Print[Zt // MatrixForm]

Z1 = Import[
   "C:\\Users\\mw\\Documents\\dottorato\\branching_process_programs\\GW_type2\\
     GW_type2_data\\GW2_Z1s_gen8_num_sim20_a[0.1 0.  0.2 0.7 0.
     0. ]_b[0.2 0.8 0.  0.  0.  0. ].txt", "Table", Delimiter → ","];
Z1 = Z1⟦1 ;; Dimensions[Z1]⟦1⟧ - lrow, 1 ;; Dimensions[Z1]⟦2⟧ - lcol⟧;
(* select part of the dataset*)
Print["dimension matrix Z1: ", Dimensions[Z1]]
Print[Z1 // MatrixForm]

Z2 = Import[
   "C:\\Users\\mw\\Documents\\dottorato\\branching_process_programs\\GW_type2\\
     GW_type2_data\\GW2_Z2s_gen8_num_sim20_a[0.1 0.  0.2 0.7 0.
     0. ]_b[0.2 0.8 0.  0.  0.  0. ].txt", "Table", Delimiter → ","];
Z2 = Z2⟦1 ;; Dimensions[Z1]⟦1⟧ - lrow, 1 ;; Dimensions[Z2]⟦2⟧ - lcol⟧;
(* select part of the dataset*)
Print["dimension matrix Z2: ", Dimensions[Z2]]
Print[Z2 // MatrixForm]
Length[Z2]
```

Out[∘]=

```
{0.1, 0., 0.2, 0.7}
```

Out[∘]=

```
{0.2, 0.8, 0., 0.}
```

dimension matrix Zt: {9, 14}

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 2 & 2 \\ 2 & 4 & 4 & 2 & 1 & 4 & 3 & 3 & 1 & 2 & 4 & 3 & 4 & 0 \\ 1 & 6 & 6 & 2 & 1 & 8 & 3 & 1 & 1 & 4 & 7 & 3 & 6 & 0 \\ 0 & 6 & 12 & 3 & 0 & 14 & 5 & 1 & 1 & 8 & 9 & 5 & 9 & 0 \\ 0 & 7 & 15 & 5 & 0 & 24 & 9 & 1 & 1 & 13 & 9 & 9 & 10 & 0 \\ 0 & 11 & 16 & 7 & 0 & 37 & 12 & 1 & 1 & 17 & 13 & 13 & 13 & 0 \\ 0 & 11 & 21 & 11 & 0 & 49 & 20 & 1 & 1 & 25 & 20 & 17 & 16 & 0 \\ 0 & 13 & 31 & 14 & 0 & 75 & 33 & 1 & 1 & 29 & 23 & 21 & 21 & 0 \end{pmatrix}$$

dimension matrix Z1: {9, 14}

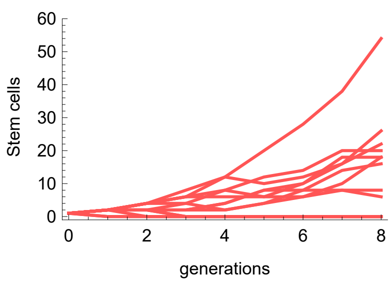
$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 0 & 2 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 \\ 2 & 4 & 4 & 2 & 0 & 4 & 2 & 2 & 0 & 2 & 4 & 2 & 4 & 0 \\ 0 & 4 & 6 & 2 & 0 & 8 & 2 & 0 & 0 & 4 & 6 & 2 & 6 & 0 \\ 0 & 2 & 12 & 2 & 0 & 12 & 4 & 0 & 0 & 8 & 8 & 4 & 6 & 0 \\ 0 & 4 & 10 & 4 & 0 & 20 & 8 & 0 & 0 & 12 & 6 & 8 & 6 & 0 \\ 0 & 8 & 12 & 6 & 0 & 28 & 10 & 0 & 0 & 14 & 10 & 8 & 6 & 0 \\ 0 & 8 & 16 & 8 & 0 & 38 & 16 & 0 & 0 & 20 & 18 & 14 & 10 & 0 \\ 0 & 6 & 22 & 8 & 0 & 54 & 26 & 0 & 0 & 20 & 18 & 16 & 18 & 0 \end{pmatrix}$$

dimension matrix Z2: {9, 14}

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 4 & 0 & 1 & 0 & 2 & 1 & 1 & 1 & 0 & 1 & 1 & 3 & 0 \\ 0 & 3 & 5 & 1 & 0 & 4 & 1 & 1 & 1 & 1 & 3 & 1 & 4 & 0 \\ 0 & 3 & 4 & 1 & 0 & 9 & 2 & 1 & 1 & 3 & 3 & 5 & 7 & 0 \\ 0 & 3 & 5 & 3 & 0 & 11 & 4 & 1 & 1 & 5 & 2 & 3 & 6 & 0 \\ 0 & 7 & 9 & 6 & 0 & 21 & 7 & 1 & 1 & 9 & 5 & 5 & 3 & 0 \end{pmatrix}$$

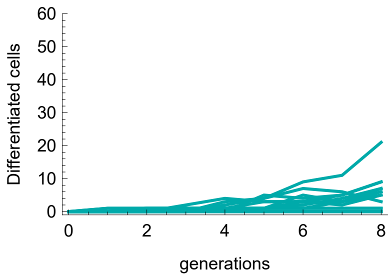
```
(*plot trajectories*)
ListLinePlot[Transpose[Z1],
 PlotRange → {-1, 60}, Frame → {{True, False}, {True, False}},
 FrameLabel → {{" Stem cells", None}, {"generations", None}},
 ImageSize → 200, PlotStyle → Lighter[Red], DataRange → {0, Length[Z2] - 1}]
```

Out[∘]=



In[ ]:=
```
ListLinePlot[Transpose[Z2],
 PlotRange → {-1, 60}, Frame → {{True, False}, {True, False}},
 FrameLabel → {{"Differentiated cells", None}, {"generations", None}},
 ImageSize → 200, PlotStyle → Darker[Cyan], DataRange → {0, Length[Z2] - 1}]
```

Out[ ]=



Useful Functions

```
(*find conditional likelihood*)
(*ji_ = start with ji type-1 cells,
jf_=end up with jf type-1 cells, same thing with ki_, kf_*)
CondLikelihood[ji_, ki_, jf_, kf_] := \
  (PGFa = a0 + a2 * w + a3 * s^2 ;
   PGFb = b0 + b1 * w + b2 * s ;
   D[(PGFa)^ji * (PGFb)^ki, {s, jf}, {w, kf}] / (kf! * jf!) /. {s → 0, w → 0 }
  )

 (*------------------Define Definite Polynomial Integration-------
 --------------------------------------------------------------*)
PolyDInt[y_ + z_, x_, a_, b_] := PolyDInt[y, x, a, b] + PolyDInt[z, x, a, b];
PolyDInt[c_ y_, x_, a_, b_] := c * PolyDInt[y, x, a, b] /; FreeQ[c, x];
PolyDInt[x_^n_., x_ , a_, b_] := b^(n + 1) / (n + 1) - a^(n + 1) / (n + 1);
PolyDInt[c_, x_, a_, b_] := c * b - c * a /; FreeQ[c, x];
PolyDInt[y_, x_, a_, b_] := Block[{expanded}, expanded = Expand[y];
  If[PolynomialQ[expanded, x], PolyDInt[expanded, x, a, b]]]

(*------------------------------------------------------------------------
                                             ---------------------------------------
                                             -------*)
```

Posteriors

```
organoids = Dimensions[Zt]⟦2⟧;                    (*num organoids *)
ngen = Dimensions[Zt]⟦1⟧ - 1;                     (*generation*)
ndata = organoids * ngen ;                         (*num of data*)
groupby = 1;                                       (*num of data to use together *)
ndataseq = ndata / groupby ;                        (*num of groups*)

prior = 1;
posteriors = ConstantArray[0, {organoids}];
(*Cell counts of the same trajectory are analyzed
 together and then we loop over all the trajectories.*)
For[org = 1, org ≤ organoids, org ++,
      likelihood = 1;
      (*create the joint likelihood*)
      For[i = 1, i ≤ ngen, i++,
   likelihood *= CondLikelihood[Z1⟦i, org⟧, Z2⟦i, org⟧, Z1⟦i + 1, org⟧, Z2⟦i + 1, org⟧]];
      likelihood = likelihood /. {a0 → 1 - a2 - a3, b0 → 1 - b1 - b2};
  (*apply constrains*)

      priorlikelihood = MonomialList[prior * likelihood];
  (*Parallelization: integration on many monomial instead of a single polynomial*)
      Print["org: ", org, "   likelihood length:", Length[priorlikelihood]];
      (*\ symbol is not recognize from the PolyDInt code!*)
      (*calculate the evidence*)
      normalization = Simplify[ParallelTable[PolyDInt[PolyDInt[PolyDInt[ PolyDInt[
          priorlikelihood [[k]], a3, 0, 1 - a2], a2, 0, 1], b2, 0, 1 - b1], b1, 0, 1],
     {k, 1, Length[priorlikelihood ]}]];
      (*calculate the posterior*)
      posterior = prior * likelihood / Total[normalization];
      posteriors⟦org⟧ = posterior;
      prior = posterior;

 ] // AbsoluteTiming
```

```
org: 5   likelihood length:46 893
org: 6   likelihood length:152 101
org: 7   likelihood length:202 985
org: 8   likelihood length:241 840
org: 9   likelihood length:241 840
org: 10   likelihood length:663 978
org: 11   likelihood length:1 539 245
org: 12   likelihood length:2 429 898
org: 13   likelihood length:3 920 180
org: 14   likelihood length:4 220 479
```

Out[ ]=
```
{57 056.1, Null}
```

Marginals

In[ ]:=
```
lpost = Length[posteriors];
pa0s = ConstantArray[0, {lpost}];
pa2s = ConstantArray[0, {lpost}]; pa3s = ConstantArray[0, {lpost}];
pb0s = ConstantArray[0, {lpost}];
pb1s = ConstantArray[0, {lpost}]; pb2s = ConstantArray[0, {lpost}];

For[i = 1, i ≤ lpost, i++,
  posterior = MonomialList[Simplify[posteriors[[i]]]];
  (*find marginals ai*)
  Print["i: ", i];
  (*Print["posterior:  ", posterior];*)
  (*Exploiting parallel computing*)
  pai = ParallelTable[PolyDInt[PolyDInt[posterior[[k]], b2, 0, 1 - b1], b1, 0, 1],
    {k, 1, Length[posterior]}];
  len = Length[pai];
  pa2s[[i]] = Total[ParallelTable[PolyDInt[pai[[k]], a3, 0, 1 - a2], {k, 1, len}]];
  pa3s[[i]] = Total[ParallelTable[PolyDInt[pai[[k]], a2, 0, 1 - a3], {k, 1, len}]];
  pai = MonomialList[Total[pai /. a2 → 1 - a0 - a3]];
  pa0s[[i]] = Total[ParallelTable[PolyDInt[pai[[k]], a3, 0, 1 - a0], {k, 1, Length[pai]}]];

  (*find marginals bi*)
  pbi = ParallelTable[PolyDInt[PolyDInt[posterior[[k]], a3, 0, 1 - a2], a2, 0, 1],
    {k, 1, Length[posterior]}];
  len = Length[pbi];
  pb1s[[i]] = Total[ParallelTable[PolyDInt[pbi[[k]], b2, 0, 1 - b1], {k, 1, len}]];
  pb2s[[i]] = Total[ParallelTable[PolyDInt[pbi[[k]], b1, 0, 1 - b2], {k, 1, len}]];
  pbi = MonomialList[Total[pbi /. b2 → 1 - b0 - b1]];
  pb0s[[i]] = Total[ParallelTable[PolyDInt[pbi[[k]], b1, 0, 1 - b0], {k, 1, Length[pbi]}]];
 ] // AbsoluteTiming
```

```
i: 1
i: 2
i: 3
i: 4
i: 5
i: 6
i: 7
i: 8
i: 9
i: 10
i: 11
i: 12
i: 13
i: 14
```

In[ ]:=

Find estimates

In[•]:=
```
npar = 3;
as = {a0, a2, a3}; bs = {b0, b1, b2};
pas = {pa0s, pa2s, pa3s};     pbs = {pb0s, pb1s, pb2s};
Eas = ConstantArray[0, Dimensions[pas]]; sas = ConstantArray[0, Dimensions[pas]];
Ebs = ConstantArray[0, Dimensions[pbs]]; sbs = ConstantArray[0, Dimensions[pbs]];
For[j = 1, j ≤ npar, j++,
    Print["paramter (j) : ", j];
    For[i = 1, i ≤ lpost, i++,

   apas = MonomialList[as⟦j⟧ * Simplify[pas⟦j, i⟧]];
   Eas⟦j, i⟧ =
    Total[ParallelTable[PolyDInt[apas⟦k⟧ , as⟦j⟧, 0, 1], {k, 1, Length[apas ]}]];
   apas = MonomialList[as⟦j⟧^2 * Simplify[pas⟦j, i⟧]];
   sas⟦j, i⟧ =
    Sqrt[Total[ParallelTable[PolyDInt[apas⟦k⟧ , as⟦j⟧, 0, 1], {k, 1, Length[apas ]}]] -
      Eas⟦j, i⟧^2];

   bpbs = MonomialList[bs⟦j⟧ * Simplify[pbs⟦j, i⟧]];
   Ebs⟦j, i⟧ =
    Total[ParallelTable[PolyDInt[bpbs⟦k⟧ , bs⟦j⟧, 0, 1], {k, 1, Length[bpbs ]}]];
   bpbs = MonomialList[bs⟦j⟧^2 * Simplify[pbs⟦j, i⟧]];
   sbs⟦j, i⟧ = Sqrt[Total[ParallelTable[
        PolyDInt[bpbs⟦k⟧ , bs⟦j⟧, 0, 1], {k, 1, Length[bpbs ]}]] - Ebs⟦j, i⟧^2];

   ];
 ] // AbsoluteTiming
```

```
paramter (j) : 1
paramter (j) : 2
paramter (j) : 3
```

Out[•]=
```
{322.068, Null}
```

```
paramter (j) : 1
paramter (j) : 2
paramter (j) : 3
```

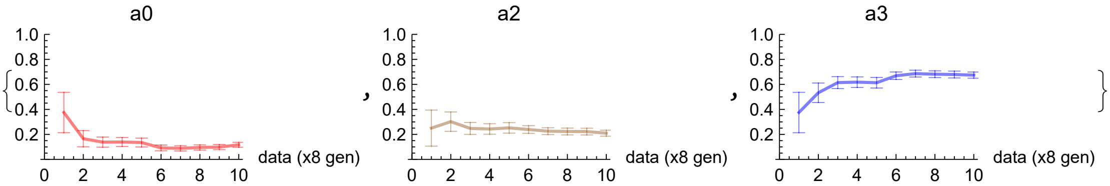
Out[•]=
```
{656.359, Null}
```

Plot

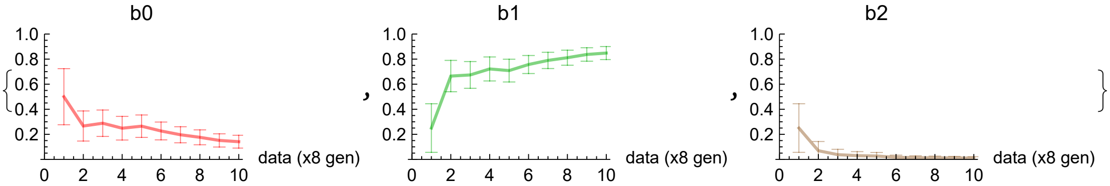
Out[•]=
```
{3, 10}
```

In[•]:=
```
(*make plots*)
acolors = {Red, Brown, Blue};
bcolors = {Red, Darker[Green], Brown};
pltEas = Table[ListLinePlot[Table[Around[Eas⟦j⟧⟦k⟧, sas⟦j⟧⟦k⟧], {k, 1, organoids}],
   PlotStyle → Directive[Opacity[0.5], acolors⟦j⟧], PlotRange → {0, 1},
   AxesLabel → {"data (x" <> ToString[ngen] <> " gen)"},
   PlotLabel → TextString[as⟦j⟧]], {j, 1, npar}]
pltEbs = Table[ListLinePlot[Table[Around[Ebs⟦j⟧⟦k⟧, sbs⟦j⟧⟦k⟧], {k, 1, organoids}],
   PlotStyle → Directive[Opacity[0.5], bcolors⟦j⟧], PlotRange → {0, 1},
   AxesLabel → {"data (x" <> ToString[ngen] <> " gen)"},
   PlotLabel → TextString[bs⟦j⟧]], {j, 1, npar}]
```

Out[•]=



Out[•]=



Comparison with true values

In[ ]:=

```
ParametersMatrix = {{0.1, 0.2, 0.7}, {0.2, 0.8, 0.0}};
truea = Table[
  ListLinePlot[ConstantArray[ParametersMatrix[[1, k]], {organoids}], PlotRange → {0, 1},
   AxesLabel → {"data (x" <> ToString[ngen] <> " gen)", "PDF stem cells"},
   PlotStyle → acolors[[k]]], {k, 1, npar}];
trueb = Table[ListLinePlot[ConstantArray[ParametersMatrix[[2, k]], {organoids}],
   PlotRange → {-.09, 1}, PlotStyle → bcolors[[k]], AxesLabel → {"data (x" <>
      ToString[ngen] <> " gen)", "PDF differentiated cells"}], {k, 1, npar}];
Show[truea, pltEas, ImageSize → 300] × Show[trueb, pltEbs, ImageSize → 300]
(*{Show[truea[[1]], pltEas[[1]]],Show[truea[[2]], pltEas[[2]]],Show[truea[[3]], pltEas[[3]]]}
 {Show[trueb[[1]], pltEbs[[1]]],Show[trueb[[2]], pltEbs[[2]]],Show[trueb[[3]], pltEbs[[3]]]}
*)
```

Out[ ]=