Tutorial on HYPERSPACE

From Zamora-Sillero *et al.*: **Efficient Exploration of Parameter Spaces in Systems Biology**, *BMC Systems Biology* 2011

Elias Zamora-Sillero [1,2]

[1] Institute of Evolutionary Biology and Environmental Studies
University of Zurich

[2] Computational System Biology Group
ETH Zurich

For queries contact `ezamora80@hotmail.com`

August 2011

Outline

1. Introduction
2. Installation
3. MCexp
4. ELexp
5. Volint

Introduction

- HYPERSPACE
  - Powerful tool for exploring parameter spaces in systems biology
  - Platform independent since based on MATLAB
- Requirements
  - MATLAB version 7.1 (or later)
  - Statistics Toolbox
- HYPERSPACE includes the following functions
  - MCexp carries out an Out-of-Equilibrium Adaptive Monte Carlo Sampling of a parameter space
  - ELexp carries out a Multiple Ellipsoid Based Sampling of a parameter space
  - Volint carries out a viable volume computation

Installation

- Download the HYPERSPACE.zip file from www.ieu.uzh.ch/wagner/software or www.csb.ethz.ch/tools
- Unzip the HYPERSPACE.zip file to the desired location
- Create a path within MATLAB to the HYPERSPACE location

Out-of-Equilibrium Adaptive Monte Carlo Sampling

- MCexp
  - Carries out an Out-of-Equilibrium Adaptive Monte Carlo Sampling of a parameter space
  - Gives as an output a set of viable parameter points from a coarse-grained exploration of a parameter space
  - Should be the first function used to explore the parameter space
  - For details see Zamora-Sillero *et al.*: **Efficient Exploration of Parameter Spaces in Systems Biology**, *BMC Systems Biology* 2011

Out-of-Equilibrium Adaptive Monte Carlo Sampling

- MCexp

OutM = MCexp(function,threshold,x0,bmax,bmin,n)

- Input:

  function : a function that takes a parameter point and evaluates its cost

  threshold : a scalar that defines the maximum value of the cost such that the parameter points is viable

  x0 : a vector with the coordinates of a viable parameter point

  bmax : a vector with the upper bounds of the parameter space

  bmin : a vector with the lower bounds of the parameter space

  n : a scalar that contains the maximum number of parameter evaluations

Out-of-Equilibrium Adaptive Monte Carlo Sampling

- MCexp

OutM = MCexp(function,threshold,x0,bmax,bmin,n)

- Output:
  OutM : a data structure with three fields
  OutM.V : a matrix whose rows contain all the viable parameter points found by the algorithm
  OutM.cost : a column vector with the cost of the viable parameter points present in Out.V
  OutM.flag : a data structure with two fields
  OutM.flag.vol : a vector that contains the volume covered by the enclosing ellipsoids in every iteration
  OutM.flag.conv : a binary scalar that equals 1(0) if the algorithm converged (did not converge)

Out-of-Equilibrium Adaptive Monte Carlo Sampling

- MCexp

OutM = MCexp(function,threshold,x0,bmax,bmin,n)

- For more documentation type help MCexp

MCexp: example

OutM = MCexp(function,threshold,x0,bmax,bmin,n)

MCexp: example

OutM = MCexp('funtest', threshold, x0, bmax, bmin, n)

function: funtest.m

- funtest is a function that takes as an argument a parameter point in two dimensions and gives as an output the distance of that parameter point to the origin [0 0]

MCexp: example

OutM = MCexp('funtest', 1, x0, bmax, bmin, n)

threshold: 1

- The parameter points for which funtest is smaller than 1 are considered viable

MCexp: example

OutM = MCexp('funtest', 1, [0 0], bmax, bmin, n)

x0: [0 0]

- The sampling starts from [0 0]

MCexp: example

OutM = MCexp('funtest', 1, [0 0], [100 100], bmin, n)

bmax: [100 100]

- The upper bound of the parameter space on each axis is [100 100]

MCexp: example

OutM = MCexp('funtest', 1, [0 0], [100 100], [-100 -100], n)

bmin: [-100 -100]

- The lower bound of the parameter space on each axis is [-100 -100]

MCexp: example

OutM = MCexp('funtest', 1, [0 0], [100 100], [-100 -100], 100000)

n: 100000

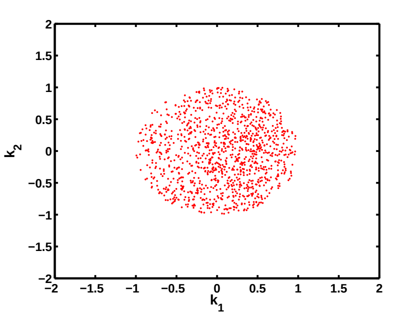
- The maximum number of parameter evaluations is 100000

MCexp: example

OutM = MCexp('funtest', 1, [0 0], [100 100], [-100 -100], 100000)

OutM.V

- OutM.V contains all the viable parameter points found
- To visualize them type: plot(OutM.V(:,1), OutM.V(:,2),'.r');



MCexp: example

OutM = MCexp('funtest', 1, [0 0], [100 100], [-100 -100], 100000)

OutM.flag.conv: 1

- The method converged before reaching the maximum number of evaluations

Multiple Ellipsoids Based Sampling

- ELexp
  - Carries out a Multiple Ellipsoids Based Sampling of the parameter space
  - Takes as an argument a set of viable parameter points
  - Gives as an output a set of viable parameter points from a detailed exploration of the parameter space
  - Is convenient to seed with the viable parameter points found by MCexp
  - For details see Zamora-Sillero *et al.*: **Efficient Exploration of Parameter Spaces in Systems Biology**, *BMC Systems Biology*. 2011

Multiple Ellipsoids Based Sampling

- ELexp

OutE = ELexp(function,threshold,Vo, bmax, bmin, n)

- Input:

  function : a function that takes a parameter point and evaluates its cost
  threshold : a scalar that defines the maximum value of the cost such that the parameter points is viable
  Vo : a matrix whose rows contain viable parameter points
  bmax : a vector with the upper bounds of the parameter space
  bmin : a vector with the lower bounds of the parameter space
  n : a scalar that contains the maximum number of parameter evaluations

Multiple Ellipsoids Based Sampling

- ELexp

OutE = ELexp(function,threshold,Vo, bmax, bmin, n)

- Output:
  OutE : a data structure with three fields
  OutE.V : a matrix whose rows contain all the viable parameter points found by the algorithm
  OutE.cost : a column vector with the cost of the viable parameter points present in OutE.V
  OutE.flag : a data structure with two fields
  OutE.flag.vol : a vector that contains the volume covered by the enclosing ellipsoids in every iteration
  OutE.flag.conv : a binary scalar that equals 1(0) if the algorithm converged (did not converge)

Multiple Ellipsoids Based Sampling

- ELexp

OutE = ELexp(function,threshold,Vo, bmax, bmin, n)

- For more documentation type help ELexp

ELexp: example

OutE = ELexp(function,threshold,Vo, bmax, bmin, n)

ELexp: example

OutE = ELexp('funtest', threshold, Vo, bmax, bmin, n)

function: funtest.m

- funtest is a function that takes as an argument a parameter point in two dimensions and gives as an output the distance of that parameter point to the origin [0 0]

ELexp: example

OutE = ELexp('funtest', 1, Vo, bmax, bmin, n)

threshold: 1

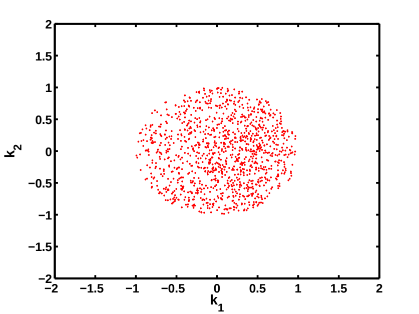
- The parameter points for which funtest is smaller than 1 are considered viable

ELexp: example

OutE = ELexp('funtest', 1, OutM.V, bmax, bmin, n)

Vo: OutM.V

- OutM.V is the set of viable parameter points found by MCexp



ELexp: example

OutE = ELexp('funtest', 1, OutM.V, [100 100], bmin, n)

bmax: [100 100]

- The upper bound of the parameter space on each axis is [100 100]

ELexp: example

OutE = ELexp('funtest', 1, OutM.V, [100 100], [-100 -100], n)

bmin: [-100 -100]

- The lower bound of the parameter space on each axis is [-100 -100]

ELexp: example

```
OutE = ELexp('funtest', 1, OutM.V, [100 100], [-100 -100],...
100000)
```

n: 100000

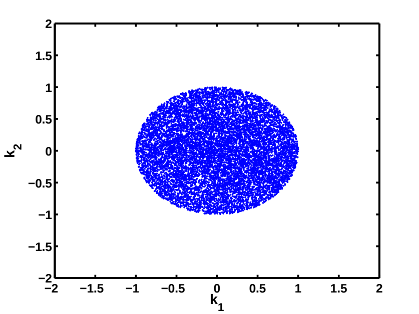
- The maximum number of parameter evaluations is 100000

ELexp: example

OutE = ELexp('funtest', 1, OutM.V, [100 100], [-100 -100],...
100000)

OutE.V

- OutE.V contains all the viable parameter points found
- To visualize them type: plot(OutE.V(:,1), OutE.V(:,2),'.b');



ELexp: example

```
OutE = ELexp(’funtest’, 1, OutM.V, [100 100], [-100 -100],...
100000)
```

OutE.flag.conv: 1

- The method converged before reaching the maximum number of evaluations

Viable volume estimation

- Volint
  - Estimates through a Monte Carlo integration the volume of the parameter space filled by viable points
  - Takes as an argument a set of viable parameter points in order to define the integration domain. It is convenient to use the set of viable parameter points found by MCexp and ELexp
  - Gives as an output the estimation of the viable volume and a set of parameter points uniformly sampled from the viable space
  - For details see Zamora-Sillero *et al.*: **Efficient Exploration of Parameter Spaces in Systems Biology**, *BMC Systems Biology*. 2011

Viable volume estimation

- Volint

OutV = Volint(function, threshold, Vo, bmax, bmin, n)

- Input:

  function : a function that takes a parameter point and evaluates its cost

  threshold : a scalar that defines the maximum value of the cost such that a parameter point is viable

  Vo : a matrix whose rows contain viable parameter points

  bmax : a vector with the upper bounds of the parameter space

  bmin : a vector with the lower bounds of the parameter space

  n : a scalar that contains the maximum number of parameter evaluations

Viable volume estimation

- Volint

OutV = Volint(function, threshold, Vo, bmax, bmin, n)

- Output:
  OutV : a data structure with four fields
  OutV.V : a matrix whose rows contain all the viable parameter points found by the algorithm
  OutV.cost : a column vector with the cost of the viable parameter points present in OutV.V
  OutV.vol : a scalar that contains the estimation of the viable volume
  OutV.err : a scalar that contains the estimation of the error in the volume calculation

Viable volume estimation

- Volint

OutV = Volint(function, threshold, Vo, bmax, bmin, n)

- For more documentation type help Volint

Volint: example

OutV = Volint(function, threshold, Vo, bmax, bmin, n)

Volint: example

OutV = Volint('funtest', threshold, Vo, bmax, bmin, n)

function: funtest.m

- funtest is a function that takes as an argument a parameter point in two dimensions and gives as an output the distance of that parameter to the origin [0 0]

Volint: example

OutV = Volint('funtest', 1, Vo, bmax, bmin, n)

threshold: 1

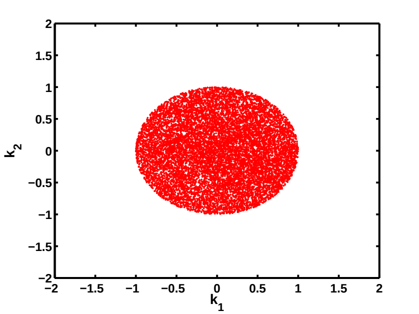
- The parameter points for which funtest is smaller than 1 are considered viable

Volint: example

OutV = Volint('funtest', 1, Vtotal, bmax, bmin, n)

Vo: Vtotal

- Vtotal is the set of viable parameter points found by MCexp and ELexp: Vtotal=vertcat(OutM.V,OutE.V);



Volint: example

OutV = Volint('funtest', 1, Vtotal, [100 100], bmin, n)

bmax: [100 100]

- The upper bound of the parameter space on each axis is [100 100]

Volint: example

OutV = Volint('funtest', 1, Vtotal , [100 100], [-100 -100], n)

bmin: [-100 -100]

- The lower bound of the parameter space on each axis is [-100 -100]

Volint: example

OutV = Volint('funtest', 1, Vtotal , [100 100], [-100 -100],...
100000)

n: 100000

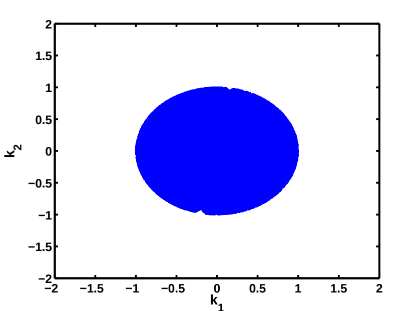
- The maximum number of parameter evaluations is 100000

Volint: example

OutV = Volint(’funtest’, 1, Vtotal, [100 100], [-100 -100],...
100000)

OutV.V

- The set of uniformly distributed viable parameter points computed by Volint
- To visualize them type: plot(OutV.V(:,1), OutV.V(:,2),’.b’);



Volint: example

```
OutV = Volint('funtest', 1, Vtotal, [100 100], [-100 -100],...
100000)
```

OutV.vol: 3.1320

- The estimated volume is equal to 3.1320. It differs less than 1 percent from the exact one

OutV.err: 0.0079

- The estimated error of the volume estimation