SModelS 3.0

Guide for a new Database Entry

Database – Constraints and Conditions

- The constraints and conditions are string expressions for the SMS weights:

constraint: 2*({(PV > …} + { (PV > …)})

condition: Csim({(PV > …},{ (PV > …)})

Example: constraint: {(PV > anyBSM(1),anyBSM(2)), (anyBSM(1) > jet,jet,jet), (anyBSM(2) > jet,jet,jet)}

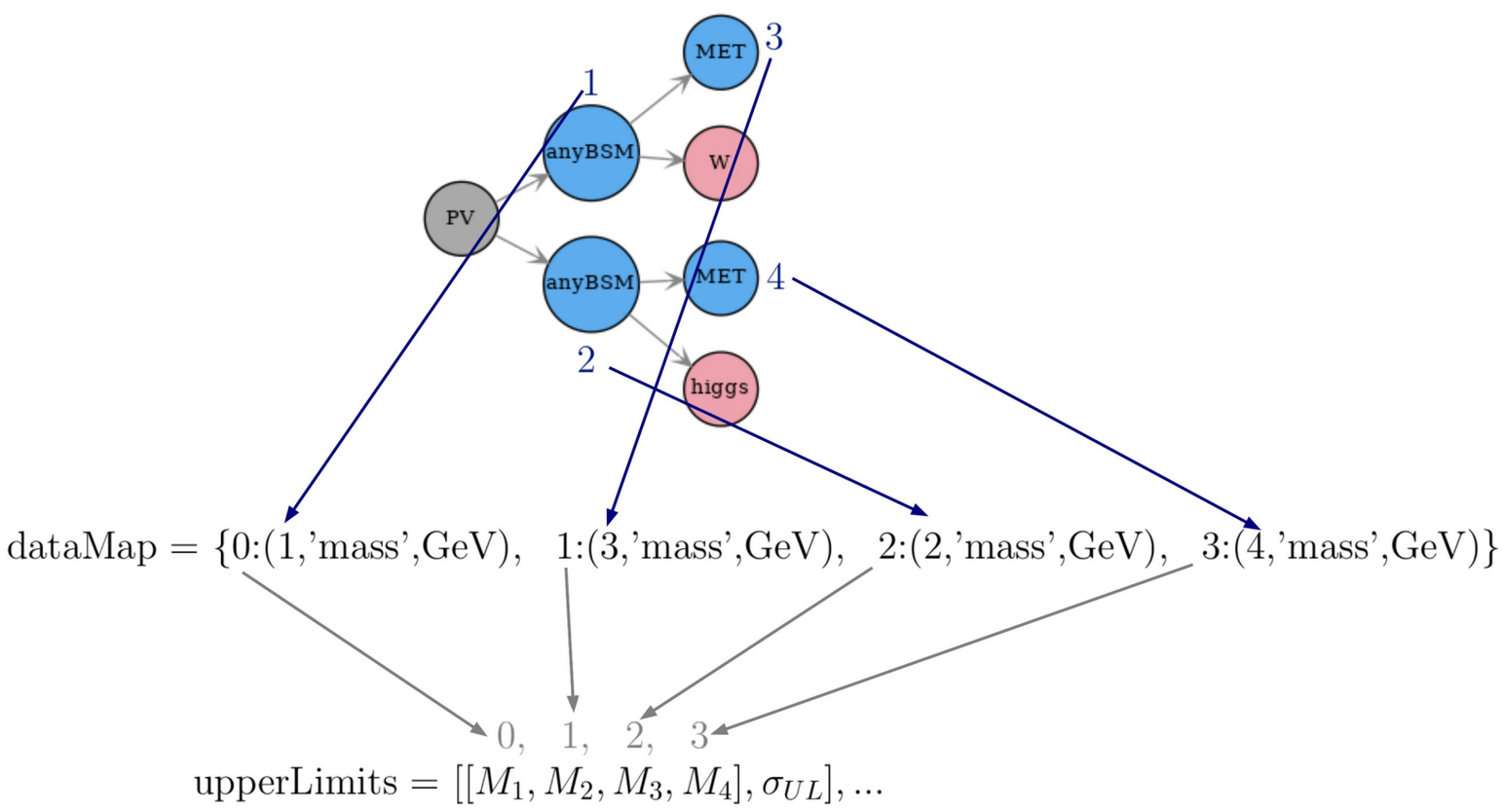
- The labels appearing in constraint and conditions must match one of the particles defined in databaseParticles.py
- The SMS appearing in the constraints are required to be unique, which means that distinct ExpSMS belonging to the same TxName can not match.
- *Note that the strings representing SMS are delimited by curly brackets:* {(PV > …} = {ExpSMS}
- *All BSM nodes appearing in the SMS description should be numbered!*

Database – DataMap

- The dataMap stores the mapping between the node numbers appearing in the constraint (e.g. anyBSM(1),...) to the entries in the flat array of the data grid.

  dataMap = {<data grid index> : (<node index>, <attribute>,<unit>)}

  - Example:

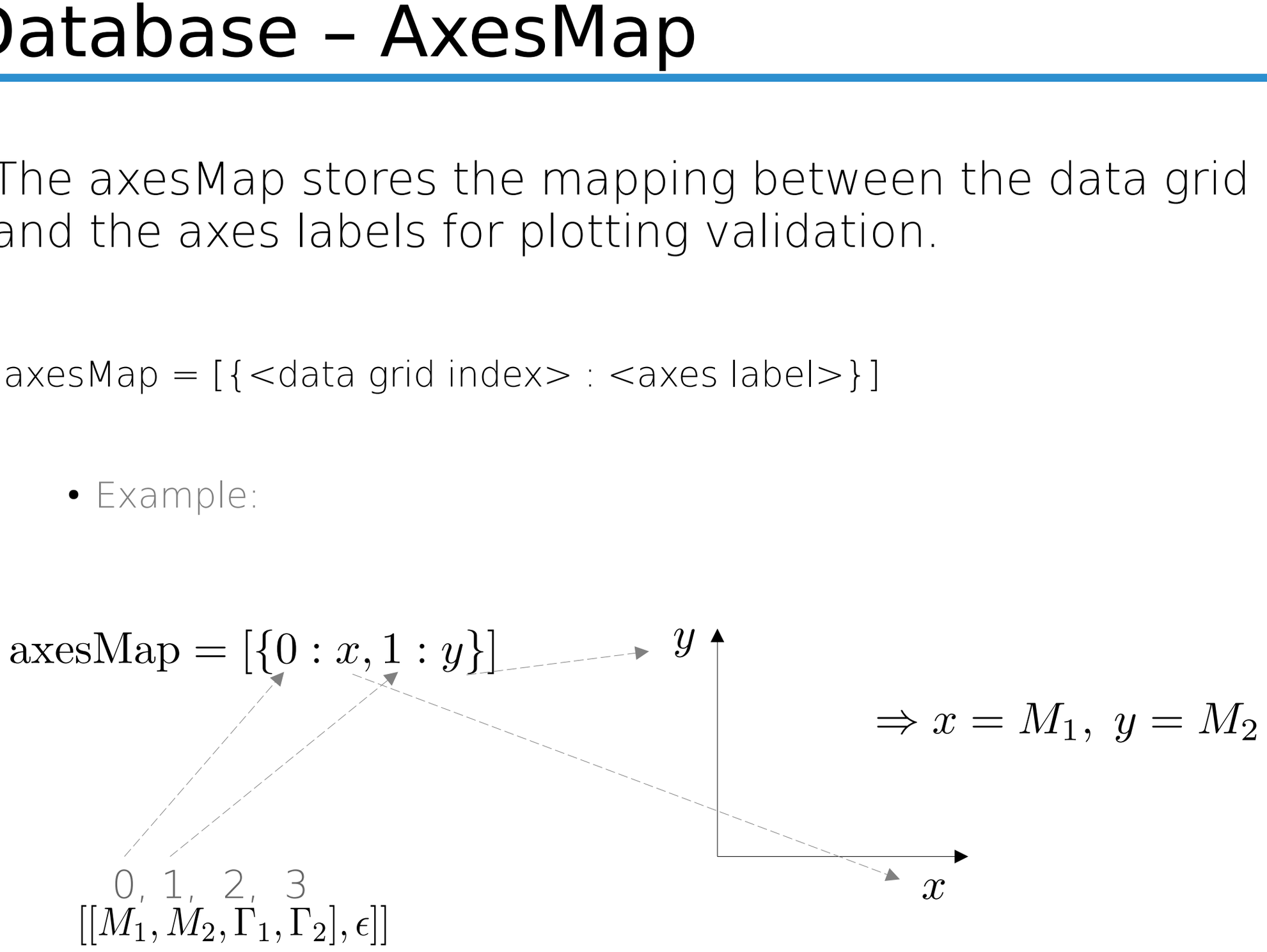


Database – AxesMap

- The axesMap stores the mapping between the data grid indices and the axes labels for plotting validation.

axesMap = [{<data grid index> : <axes label>}]

  - Example:



Database – addMassPlane

- When adding a new mass plane a dictionary must be given defining the mapping between the data grid indices and variables. This mapping can be used to define constraints between the data grid values.

  TxName.addMassPlane({<data grid index> : <variable label>})

- Example:

$$\text{TxName.addMassPlane}(\{0 : 'x', 1 : 'y', 2 : 'z', 3 : 'z'\})$$

0, 1, 2, 3

$$[[M_1, M_2, \Gamma_1, \Gamma_2], \epsilon]]$$

$$\Rightarrow x = M_1,\ y = M_2,\ z = \Gamma_1 = \Gamma_2$$

(setting “2” and “3” to the same variable ‘z’ imposes that the entries 2 and 3 are equal)

Database – Coordinates

- The coordinates argument in setSources defines the mapping between the variables previously defined in addMassPlane to the columns in the input data file.

MassPlane.setSources(coordinates = [{<variable> : <column index in data file>}]

- The upper limit or efficiency value is defined by the “value” string.

- Example:

$$\text{MassPlane.setSources(coordinates} = [\{x : 0, y : 1, z : 2, \text{'value'} : 4\}], ..)$$

0 1 2 3 4 5

```
#  mN2_GeV,   mN1_GeV, width_GeV, tau_ns,     Eff,       EffErr
2.5000e+02,2.5000e+02,1.0616e-13,6.2000e-03,1.4380e-02,3.7350e-04
2.5000e+02,2.4000e+02,6.5820e-13,1.0000e-03,7.5830e-05,2.4590e-05
2.5000e+02,2.5000e+02,3.6567e-13,1.8000e-03,1.0260e-03,9.3370e-05
2.5000e+02,2.4000e+02,3.6567e-13,1.8000e-03,1.0190e-03,9.2370e-05
```