CBMPY

cbmpy.sourceforge.net

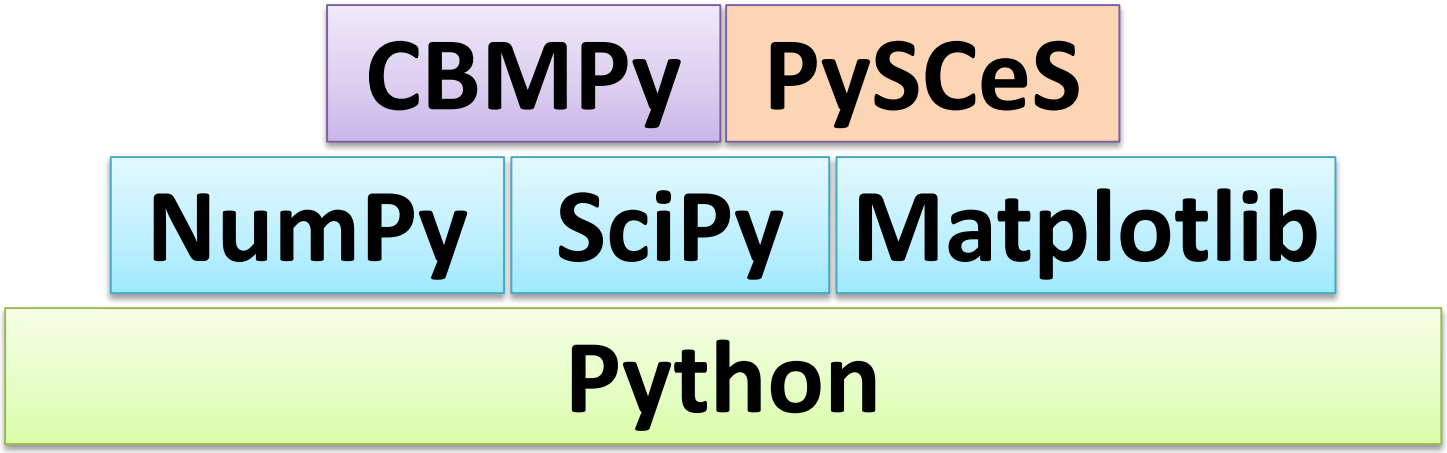
Brett G. Olivier

b.g.olivier@vu.nl

@Brett_Olivier

Systems Bioinformatics, VU University Amsterdam

A CMPy quick reference guide



Startup

```
In [1]: import pyscescbm as cbm

*****
Using CPLEX
*****

WX GUI tools available.
Qt4 GUI tools available

CBMPy environment
******************
Release: 0.7.0
Revision: r279

*****************************************************************************
* Welcome to CBMPy (0.7.2) - PySCeS Constraint Based Modelling              *
*                http://cbmpy.sourceforge.net                               *
*                     Somewhere In Time                                     *
* Copyright(C) Brett G. Olivier 2010 - 2015                                 *
* Dept. of Systems Bioinformatics                                           *
* Vrije Universiteit Amsterdam, Amsterdam, The Netherlands                  *
* CBMPy is distributed under the GNU GPL v 3.0 licence, see                 *
* LICENCE (supplied with this release) for details                          *
*****************************************************************************
```

Module

```
In [3]: cbm.<tab>
cbm.CBCPLEX                    cbm.NINF
cbm.CBCommon                   cbm.absolute_import
cbm.CBConfig                   cbm.analyzeModel
cbm.CBDataStruct               cbm.createReaction
cbm.CBGLPK                     cbm.division
cbm.CBGUI                      cbm.doFBA
cbm.CBModel                    cbm.doFBAMinSum
cbm.CBModelTools               cbm.doFVA
cbm.CBMultiCore                cbm.loadCBGUI
cbm.CBMultiEnv                 cbm.miriamids
cbm.CBNetDB                    cbm.openFileName
cbm.CBPlot                     cbm.os
cbm.CBQt4                      cbm.print_function
cbm.CBRead                     cbm.pyparsing
cbm.CBReadtxt                  cbm.readCOBRASBML
cbm.CBSolver                   cbm.readSBML2FBA
cbm.CBTools                    cbm.readSBML3FBC
cbm.CBVersion                  cbm.rev
cbm.CBWrite                    cbm.saveFileName
cbm.CBWx                       cbm.writeCOBRASBML
cbm.CBXML                      cbm.writeFVAtoCSV
cbm.FluxVariabilityAnalysis    cbm.writeModelToCOMBINEarchive
cbm.INF                        cbm.writeModelToExcel97
cbm.MinimizeSumOfAbsFluxes     cbm.writeSBML3FBC

In [3]:
```

```
In [6]: cmod = cbm.readSBML3FBC('core_memesa_model.l3.xml')
Adding objective: objMaxJ25
SBML3 load time: 0.025

In [7]: cmod2 = cbm.readSBML2FBA('core_memesa_model.xml')
objMaxJ25
Adding objective: objMaxJ25

In [9]: cmod3 = cbm.readCOBRASBML('Ecoli_iJR904.cobra.xml')
INFO: successfully converted file Ecoli_iJR904.cobra.xml to
f:\testmodels\Ecoli_Ijr904
Active objective:
Adding objective: obj
SBML3 load time: 1.279

Writing file: f:\testmodels\Ecoli_iJR904.cobra.xml.l3fbc.xml
Model exported as: f:\testmodels\Ecoli_iJR904.cobra.xml.l3fbc.xml

INFO: SBML Level 3 + FBC file generated as:
f:\testmodels\Ecoli_iJR904.cobra.xml.l3fbc
```

ReadPython

```
def Define_milp_model_1():
    """\nOriginal MILP model\n"""

    model_name = 'core_model_1'

    Reactions ={'RA' : {'id' : 'RA', 'reversible' : True,
                        'reagents' : [(1, 'A')], 'SUBSYSTEM' : 'b1'},
                'RB' : {'id' : 'RB', 'reversible' : True,
                        'reagents' : [(1, 'B')], 'SUBSYSTEM' : 'b2'},
                'R03' : {'id' : 'R03', 'reversible' : True,
                         'reagents' : [(-1, 'A'),(1, 'C')], 'SUBSYSTEM' : 'b1'},
                'R04' : {'id' : 'R04', 'reversible' : True,
                         'reagents' : [(-1, 'B'), 1, 'C')], 'SUBSYSTEM' : 'b2'},
                'R05' : {'id' : 'R05', 'reversible' : False,
                         'reagents' : [(-1, 'C')], 'SUBSYSTEM' : 'b3‘} }

    Species = { 'A' : {'id' : 'A', 'boundary' : False, 'SUBSYSTEM' : 'b1'},
                'B' : {'id' : 'B', 'boundary' : False, 'SUBSYSTEM' : 'b2'},
                'C' : {'id' : 'C', 'boundary' : False, 'SUBSYSTEM' : 'b3‘} }

    Bounds = {'RA' : {'lower' : 2, 'upper' : 10},
              'RB' : {'lower' : -10, 'upper' : 0},
              'R03' : {'lower' : -10, 'upper' : 10},
              'R04' : {'lower' : -10, 'upper' : 10},
              'R05' : {'lower' : 0, 'upper' : 0} }

    Objective_function = {'objMaxR05' : {'id' : 'objMaxR05', 'flux' : 'R05',
                          'coefficient' : 1, 'sense' : 'Maximize', 'active' : True}}

    return model_name, Reactions, Species, Bounds, Objective_function
```

```
In [14]: from CoreModelDefinitions import Define_milp_model_1

In [15]: name, react, spec, bnds, of = Define_milp_model_1()

In [16]: cmod4 = cbm.CBModelTools.quickDefaultBuild(name, react, spec,
bnds, of)
Adding objective: objMaxR05
Reaction R03 already has bounds: {'SUBSYSTEM': 'b1', 'reagents': [(-1, 'A'), (1, 'C')], '
eversible': True, 'id': 'R03'}
Reaction R05 already has bounds: {'SUBSYSTEM': 'b3', 'reagents': [(-1, 'C')], 'reversible
: False, 'id': 'R05'}
Reaction RA already has bounds: {'SUBSYSTEM': 'b1', 'reagents': [(1, 'A')], 'reversible':
True, 'id': 'RA'}
Reaction RB already has bounds: {'SUBSYSTEM': 'b2', 'reagents': [(1, 'B')], 'reversible':
True, 'id': 'RB'}
Reaction R04 already has bounds: {'SUBSYSTEM': 'b2', 'reagents': [(-1, 'B'), (1, 'C')], '
eversible': True, 'id': 'R04'}


In [17]: cbm.CBModelTools.<tab>
cbm.CBModelTools.addBounds
cbm.CBModelTools.addObjectiveFunction
cbm.CBModelTools.addReactions
cbm.CBModelTools.addReversibilityBounds
cbm.CBModelTools.addReversibilityBoundsIgnoreReversible
cbm.CBModelTools.addSpecies
cbm.CBModelTools.quickDefaultBuild
```

```
“Abbreviation”,”equation,”officialName”
SERTRS,[c] : atp + ser-L + trnaser --> amp + h + ppi + sertrna,
                                Seryl-tRNA synthetase
FRUpts,fru[e] + pep[c] --> f1p[c] + pyr[c],
                                D-fructose transport via PEP:Pyr PTS
GLCpts,glc-D[e] + pep[c] --> g6p[c] + pyr[c],
                                D-glucose transport via PEP:Pyr PTS
TDPDRR,[c] : dtdp6dm + nadp <==> dtdpddm + h + nadph,
                                dTDP-4-dehydrorhamnose reductase
ALCD19,[c] : glyald + h + nadh <==> glyc + nad,
                                alcohol dehydrogenase (glycerol)
ALCD2x,[c] : etoh + nad <==> acald + h + nadh,
                                alcohol dehydrogenase (ethanol: NAD)
GLYCK,[c] : atp + glyc-R --> 3pg + adp + h,glycerate kinase
```

```
"Reaction ID","lower boundary","upper boundary“
"EX_glc(e)","-2.5947","-2.4929“
"EX_etoh(e)","0.7865","0.8186“
"EX_ac(e)","0.9351","0.9733"
"EX_lac-L(e)","2.8571","2.9737“
"EX_for(e)","1.4459","2“
"EX_succ(e)",0,0
"EX_pyr(e)","0.0613","0.0648"
```

ReadTXT

```
In [19]: cbm.CBReadtxt.SYMB_SPLIT = ',‘

In [20]: cbm.CBReadtxt.SYMB_IRR = ‘-->’

In [28]: cmod5 = cbm.CBReadtxt.readCSV('Spy_reactions.csv‘,
                'Spy_bounds.csv', biomass_flux=‘R_biomass_LLA1',
                 model_id='TestModel', has_header=True)

Complete duplicate skipping reaction R_PIt6
Complete duplicate skipping reaction R_PNTOt2
Complete duplicate skipping reaction R_PYDAMt
Complete duplicate skipping reaction R_PYRt2
Complete duplicate skipping reaction R_RIBFLVt2
Complete duplicate skipping reaction R_SUCCt6
Complete duplicate skipping reaction R_TRPt6
Complete duplicate skipping reaction R_TYRt6
Duplicate reaction creating new reaction: R_HISt6_1
Duplicate reaction creating new reaction: R_METt6_1
Complete duplicate skipping reaction R_PROt6
Complete duplicate skipping reaction R_THRA
Complete duplicate skipping reaction R_THRt6
Complete duplicate skipping reaction R_TMPKr
['Reaction ID', 'lower boundary', 'upper boundary']

Bounds loaded from file: Spy_bounds.csv

Adding objective: objective1
```

ReadTXT2

WriteSBML

```
In [45]: cbm.writeSBML3FBC(cmod, 'xtest.xml')
INFO: Compartment "Cell" used by species "A" is not defined, creating.
Writing file: xtest.xml
Model exported as: xtest.xml

In [47]: cbm.CBWrite.writeCOBRASBML(cmod, 'xtest2.xml')
WARNING: saving in COBRA format may result in a loss of model information!
INFO: successfully converted file xtest2.xml to f:\DSM\testmodels\xtest2.xml
Model exported as: xtest2.xml

In [48]: cbm.writeModelToExcel97(cmod, 'xtest')

In [51]: cbm.writeModelToCOMBINEarchive(cmod, 'xtest')
Writing file: f:\DSM\testmodels\sedxtmp\xtest.xml
Model exported as: f:\DSM\testmodels\sedxtmp\xtest.xml
COMBINE archive created: xtest.zip
```

```
# CBMPy produced workbooks can also be read back into model objects

In [50]: cmod6 = cbm.CBRead.readExcel97Model('xtest.xls')
Sheet: info
Sheet: solution
Successfully wrote model SBML3FBC file: "xtest.xls.l3.xml“
```

A1 | reaction

reaction	flux	lower	upper	reduced cost	FVA min	FVA max	FVA span	exchange	info	stoichiometry
R01	1	0	1	1	1	1	0	yes	info	stoich
R02	-999	-1000	1000	0	-999	1000	1999		info	stoich
R03	1000	-1000	1000	0	-999	1000	1999		info	stoich
R04	1000	-1000	1000	0	-999	1000	1999		info	stoich
R05	1	0	1000	0	1	1	0		info	stoich
R06	1	0	1000	0	0	1	1		info	stoich
R07	1	0	1000	0	0	1	1		info	stoich
R08	1	0	1000	0	0	1	1		info	stoich
R09	0								fo	stoich
R10	0								fo	stoich
R11	0								fo	stoich
R12	1								fo	stoich
R13	1								fo	stoich
R14	1000	-1							fo	stoich
R15	1									
R16	0									
R17	0									
R18	0									
R19	-999	-1								
R20	-999	-1								
R21	-999	-1								
R22	1									
R23	-999	-1								
R24	1000									
R25	0									
R26	1									

info | solution | re

id	name	reversible	lowerbound	upperbound	compartment	SUBSYSTEM	equation
R01	R01	FALSE	0	1		L	X > A
R02	R02	TRUE	-1000	1000		C1	A = B
R03	R03	TRUE	-1000	1000		C1	A = C
R04	R04	TRUE	-1000	1000		C1	C = B
R05	R05	FALSE	0	1000		L	B > D
R06	R06	FALSE	0	1000		C2	D > E
R07	R07	FALSE	0	1000			
R08	R08	FALSE	0	1000			
R09	R09	FALSE	0	1000			
R10	R10	FALSE	0	1000			
R11	R11	FALSE	0	1000			
R12	R12	FALSE	0	1000			
R13	R13	FALSE	0	1000			
R14	R14	TRUE	-1000	1000			
R15	R15	FALSE	0	1000			
R16	R16	FALSE	0	1000			
R17	R17	FALSE	0	1000			
R18	R18	FALSE	0	1000			
R19	R19	TRUE	-1000	1000			
R20	R20	TRUE	-1000	1000			
R21	R21	TRUE	-1000	1000			
R22	R22	FALSE	0	1000			
R23	R23	TRUE	-1000	1000			
R24	R24	FALSE	0	1000			
R25	R25	FALSE	0	1000			
R26	R26	FALSE	0	1000			

info | solution | reactions | metabolites | network_react

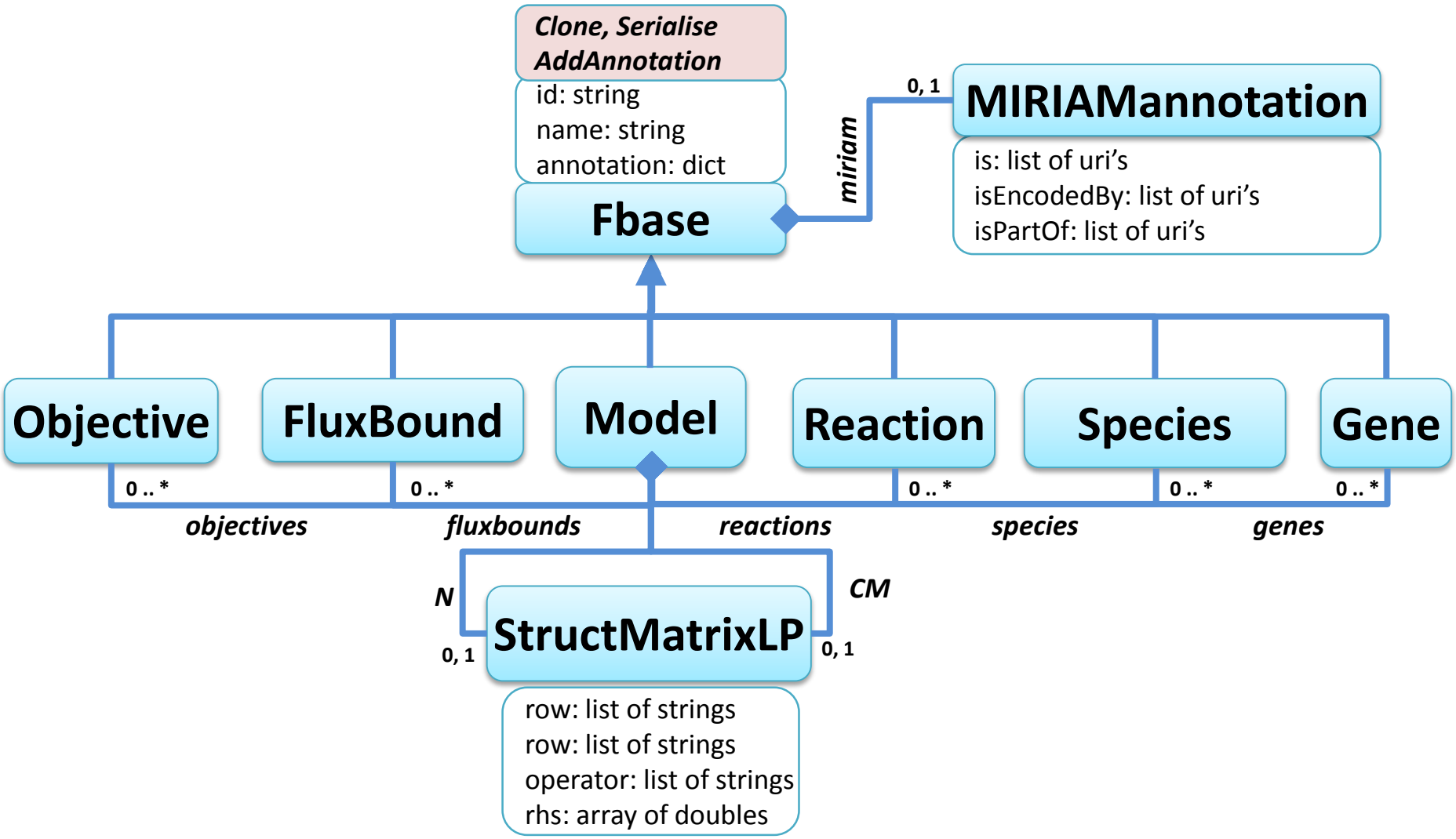
E8 | Cell

id	name	charge	chemformula	compartment	fixed	SUBSYSTEM
A	A	0		Cell	FALSE	C1
B	B	0		Cell	FALSE	C1
C	C	0		Cell	FALSE	C1
D	D	0		Cell	FALSE	C2
E	E	0		Cell	FALSE	C2
F	F	0		Cell	FALSE	C2
G	G	0		Cell	FALSE	C2
H	H	0		Cell	FALSE	C2
I	I	0		Cell	FALSE	C2
J	J	0		Cell	FALSE	C3
K	K	0		Cell	FALSE	C3
L	L	0		Cell	FALSE	C3
M	M	0		Cell	FALSE	C3
N	N	0		Cell	FALSE	C3
O	O	0		Cell	FALSE	C3
P	P	0		Cell	FALSE	C3
Q	Q	0		Cell	FALSE	C3
R	R	0		Cell	FALSE	C4
S	S	0		Cell	FALSE	C4
T	T	0		Cell	TRUE	L
U	U	0		Cell	TRUE	L
X	X	0		Cell	TRUE	L
Y	Y	0		Cell	TRUE	L

solution | reactions | metabolites | network_react | network_metab | miriam

ExcelFormat

ClassDiagram



ModelGet

```
In [27]: cmod.get.<tan>
cmod.getActiveObjective
cmod.getPid
cmod.getFluxBoundIds
cmod.getAllGeneActivities
cmod.getReactionActivity
cmod.getFluxesAssociatedWithSpecies
cmod.getAllProteinActivities
cmod.getReactionIds
cmod.getGPRforReaction
cmod.getAnnotation
cmod.getReactionNames
cmod.getGeneIds
cmod.getBoundarySpeciesIds
cmod.getReactionValues
cmod.getIrreversibleReactionIds
cmod.getCompartmentIds
cmod.getSolutionVector
cmod.getModelCreators
cmod.getExchangeReactionIds
cmod.getSpeciesIds
cmod.getObjFuncValue
cmod.getObjectiveIds
cmod.getFluxBoundByReactionID
cmod.getAllFluxBounds
cmod.getReaction
cmod.getFluxBoundsByReactionID
cmod.getAllGeneProteinAssociations
cmod.getReactionBounds
cmod.getGPRassociation
cmod.getAllProteinGeneAssociations
cmod.getReactionLowerBound
cmod.getGene
cmod.getAnnotations
cmod.getReactionUpperBound
cmod.getId
cmod.getCompartment
cmod.getReversibleReactionIds
cmod.getMIRIAMannotations
cmod.getDescription
cmod.getSpecies
cmod.getName
cmod.getExchangeReactions
cmod.getFluxBoundByID
```

```
In [27]: cmod.create.<tab>
cmod.createGeneAssociationsFromAnnotations  cmod.createReactionReagent
cmod.createGeneProteinAssociation          cmod.createReactionUpperBound
cmod.createObjectiveFunction              cmod.createSingleGeneEffectMap
cmod.createReaction                         cmod.createReactionLowerBound
cmod.createSpecies

In [27]: cmod.set.<tab>
cmod.setActiveObjective      cmod.setGeneActive
cmod.setPrefix               cmod.setAllFluxBounds
cmod.setGeneInactive         cmod.setReactionBound
cmod.setAllProteinActivities cmod.setId
cmod.setReactionBounds       cmod.setAnnotation
cmod.setModifiedDate         cmod.setReactionLowerBound
cmod.setBoundValueByName     cmod.setName
cmod.setReactionUpperBound   cmod.setCreatedDate
cmod.setObjectiveFlux        cmod.setSuffix
cmod.setDescription          cmod.setPid

In [28]: cmod.createGeneAssociationsFromAnnotations()
INFO: used key(s) '['gene_association']'
INFO: Added 902 new genes and 1066 associations to model
```

```
In [4]: cbm.doFBA(cmod)

cplx_constructLPfromFBA time: 0.0789999961853
cplx_analyzeModel FBA --> LP time: 0.0799999237061
Status: LPS_OPT
Model is optimal
analyzeModel objective value: 1.0

Out[4]: 1.0000000000000027

In [5]: cmod.getObjFuncValue()
Objective obj1: "maximize“

Out[5]: 1.00000

In [6]: cmod.getActiveObjective().getValue()
Out[6]: 1.00000

In [12]: cmod.getActiveObjective().getFluxObjectiveReactions()
Out[12]: ['R_BiomassEcoli']

In [13]: cmod.getReaction('R_BiomassEcoli').getValue()
Out[13]: 1.00000
```

FBAminsum

```
In [14]: cbm.doFBAMinSum(cmod)
INFO: Model is optimal: 1
Solution status =  1 : optimal
Solution method =  2 : dual
Objective value =  1.0
Model is optimal
Valid Presolution

RHS sense ok.
Total number of reactions: 1066

Objective value =  629.3125535
Model is optimal
Status: LPS_OPT
Model is optimal

MinimizeSumOfAbsFluxes objective value: 629.3125535

Out[14]: 629.3125535000029

In [19]: cmod.getObjFuncValue()
Objective obj1: "maximize"
Out[19]: 1.00000
```

```
In [20]: f,n = cbm.doFVA(cmod)
Valid Presolution
RHS sense ok.
Number of user selected variables: 1066
FVA has processed 200 of 1066 reactions
FVA has processed 400 of 1066 reactions
FVA has processed 600 of 1066 reactions
FVA has processed 800 of 1066 reactions
FVA has processed 1000 of 1066 reactions
Singlecore FVA took: 0.281049998601 min (1 process)

Output array has columns:
Reaction, Reduced Costs, Variability Min, Variability Max, abs(Max-Min), MinStatus, MaxStatus

In [24]: cbm.writeFVAtoCSV(f, n, 'xtest', fbaObj=cmod)
FVA results written to: xtest.fva.csv
name     optval   min      max      diff     red cost minstat  maxstat
R_F6PA   0.922    0        0.922    0.922    0        1        1
R_FBA    7.143    7.143    8.065    0.922    0        1        1
R_DHAPT  0.922    0        0.922    0.922    0        1        1
R_FRD2   0        0        0.735    0.735    0        1        1

In [36]: cmod.getReaction('R_FBA').getFVAdata()
R_FBA
Flux:   7.14339
FVAmin: 7.14339
FVAmax: 8.06593
Span:   0.92253

Out[36]: (7.14339, 7.14339, 8.06593, 0.92253)
```

```
In [38]: cmod.getReactionIds('PFK')
Out[38]: ['R_PFK', 'R_PFK_2']

In [39]: r = cmod.getReaction('R_PFK')
r.addMIRIAMannotation                    r.getSpeciesIds
r.addReagent                             r.getSpeciesObj
r.annotation                             r.getStoichiometry
r.changeId                               r.getSubstrateIds
r.changeReagentCoefficientForSpecies     r.getUpperBound
r.clone                                  r.getValue
r.createReagent                           r.is_balanced
r.deleteAnnotation                       r.is_exchange
r.deleteMIRIAMannotation                 r.miriam
r.deleteReagentWithSpeciesRef            r.getAnnotation
r.getAnnotations                         r.getEquation
r.getFVAdata                             r.serializeToDisk
r.getId                                  r.setAnnotation
r.getLowerBound                          r.setId
r.getMIRIAMannotations                   r.setLowerBound
r.getName                                r.setName
r.getPid                                 r.setPid
r.getProductIds                          r.setStoichCoefficient
r.getReagent                             r.setUpperBound
r.getReagentObjIds                       r.setValue
r.getReagentRefs                         r.undeleteReagentWithSpeciesRef
r.getReagentWithSpeciesRef

In [42]: r.getSpeciesIds()
Out[42]: ['M_atp_c', 'M_f6p_c', 'M_adp_c', 'M_fdp_c', 'M_h_c']
```

```
In [44]: r.getLowerBound()
Out[44]: 0.0

In [45]: r.setUpperBound(cbm.INF)

In [46]: r.getUpperBound()
Out[46]: inf

In [47]: cmod.getReactionBounds('R_PFK')
Out[47]: ('R_PFK', 0.0, inf, None)

In [52]: cmod.setReactionBound('R_PFK', 0.0001, 'lower')

In [53]: cmod.setReactionLowerBound('R_PFK', 0.0)

In [54]: rr = r.getReagentWithSpeciesRef('M_f6p_c')

In [55]: rr.<tab>
rr.addMIRIAMannotation    rr.getAnnotations         rr.hasAnnotation
rr.setCoefficient         rr.annotation             rr.getCoefficient
rr.id                     rr.setId                  rr.clone
rr.getId                  rr.miriam                 rr.setName
rr.coefficient            rr.getMIRIAMannotations   rr.name
rr.setPid                 rr.compartment            rr.getName
rr.role                   rr.setSpecies             rr.deleteAnnotation
rr.getPid                 rr.serialize              rr.species_ref
rr.deleteMIRIAMannotation rr.getRole                rr.serializeToDisk
rr.getAnnotation          rr.getSpecies             rr.setAnnotation
```

Species

```
In [9]: s = cmod.getSpecies('M_f6p_c')

In [10]: s.<tab>
s.addMIRIAMannotation    s.getCharge              s.isReagentOf
s.setChemFormula         s.annotation             s.getChemFormula
s.is_boundary            s.setId                  s.charge
s.getId                  s.miriam                 s.setName
s.chemFormula            s.getMIRIAMannotations   s.name
s.setPid                 s.clone                  s.getName
s.reagent_of             s.setReagentOf           s.compartment
s.getPid                 s.serialize              s.setValue
s.deleteAnnotation       s.getReagentOf           s.serializeToDisk
s.shadow_price           s.deleteMIRIAMannotation s.getValue
s.setAnnotation          s.unsetBoundary          s.getAnnotation
s.hasAnnotation          s.setBoundary            s.value
s.getAnnotations         s.id                     s.setCharge

In [6]: s.getCharge()
Out[6]: -2

In [7]: s.getChemFormula()
Out[7]: 'C6H11O9P‘

In [15]: s.is_boundary
Out[15]: True

In [16]: s.unsetBoundary()
In [17]: s.is_boundary
Out[17]: False
```

MIRIAM

```
In [18]: s.getId()
Out[18]: 'M_f6p_c'

In [19]: s.getName()
Out[19]: 'D-Fructose 6-phosphate‘

In [24]: s.addMIRIAMannotation('is', 'ChEBI', 'CHEBI:57579')

In [25]: s.getMIRIAMannotations()
Out[25]:
{'encodes': (),
 'hasPart': (),
 'hasProperty': (),
 'hasTaxon': (),
 'hasVersion': (),
 'is': ('http://identifiers.org/chebi/CHEBI:57579',),
 'isDescribedBy': (),
 'isEncodedBy': (),
 'isHomologTo': (),
 'isPartOf': (),
 'isPropertyOf': (),
 'isVersionOf': (),
 'occursIn': ()}

In [26]: s.miriam.getAndViewUrisForQualifier('is')
```

Genes

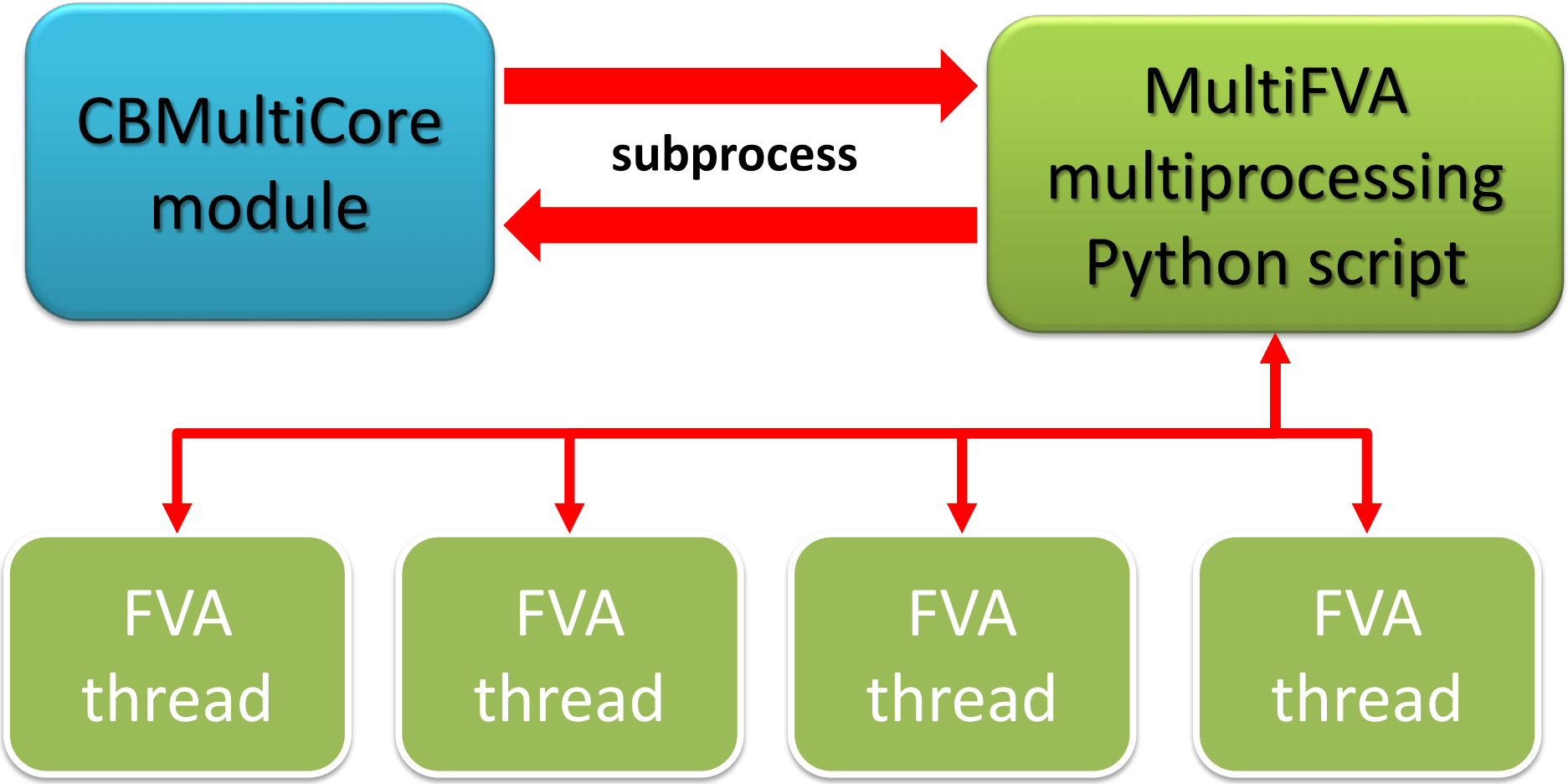
```
In [33]: cmod.getGPRforReaction('R_PFK').getGeneIds()
Out[33]: ['b3916', 'b1723']

In [35]: cmod.getGPRforReaction('R_PFK').getAssociationStr()
Out[35]: '(b3916) or (b1723)‘

In [37]: cbm.doFBA(cmod)
Out[37]: 1.00000

In [48]: cmod.setGeneInactive('b3916', update_reactions=True)
Out[48]: True

In [49]: cbm.doFBA(cmod)
Out[49]: 1.00000

In [50]: cmod.setGeneInactive('b1723', update_reactions=True)
Reaction R_PFK bounds set to [0.0 : 0.0]
Out[50]: True

In [49]: cbm.doFBA(cmod)
Out[49]: 0.982941

In [52]: cmod.resetAllGenes()
Reaction R_PFK bounds set to [0.0 : 999999.0]
```

```
In [59]: cbm.CBTools.scanForReactionDuplicates(cmod)
Found 0 pairs of duplicate reactions
Out[59]: []

In [61]: cbm.CBTools.checkFluxBoundConsistency(cmod)
Out[61]:
{'duplicate_ids': [],
 'eq+lb': [],
 'eq+ub': [],
 'lb>ub': [],
 'multiple_defines': {'equality': {}, 'lower': {}, 'upper': {}},
 'no_reaction': [],
 'rev_contradict': [],
 'undefined': {'no_lower': [], 'no_upper': [], 'no_upper_lower': []}}

In [63]: cbm.CBTools.findDeadEndMetabolites(cmod)
Out[63]:
[('M_23doguln_c', 'R_DOGULNR'),
 ('M_2pglyc_c', 'R_PGLYCP'),

In [66]: cmods = cbm.CBTools.splitReversibleReactions(cmod)
Model clone time: 0.50200009346
Reversible reaction splitter is processing: R_GLYCt
Deleting reaction R_GLYCt and 2 associated bounds
Reversible reaction splitter is processing: R_MME
Deleting reaction R_MME and 2 associated bounds
```

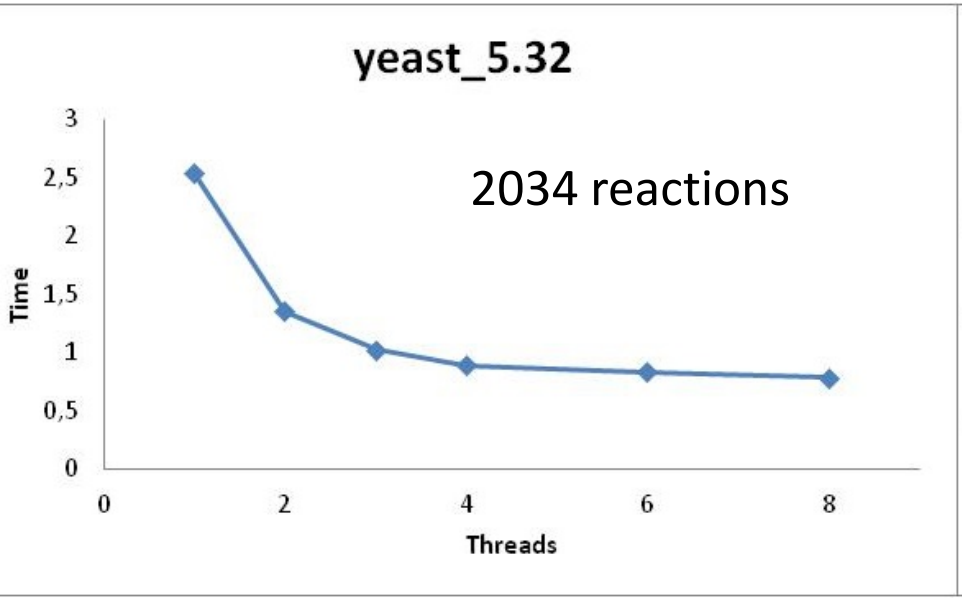
Parallel FVA: **cbm.CBMultiCore**

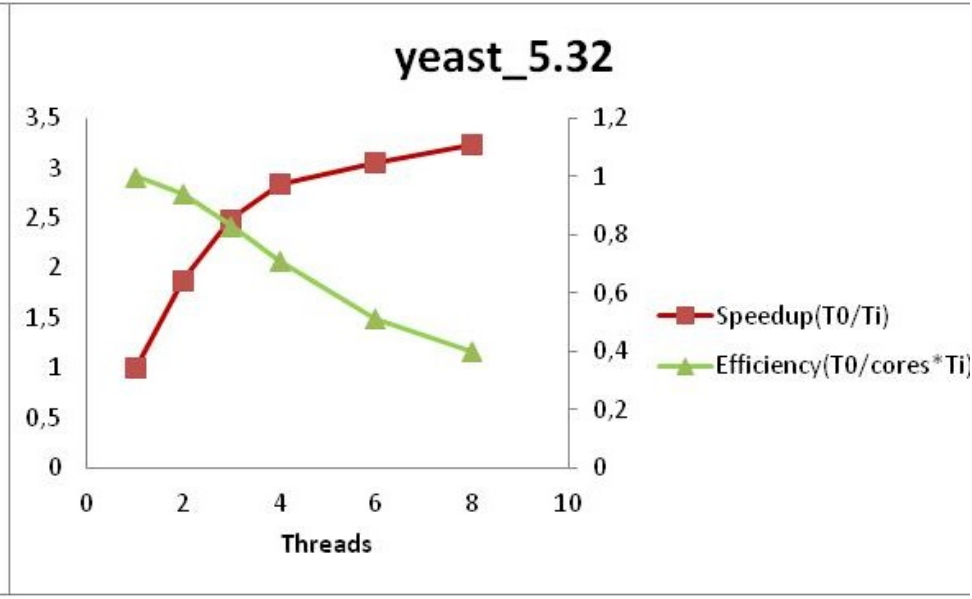
```
f, a = cbm.CBMultiCore.runMultiCoreFVA(cmod, procs=6)
```

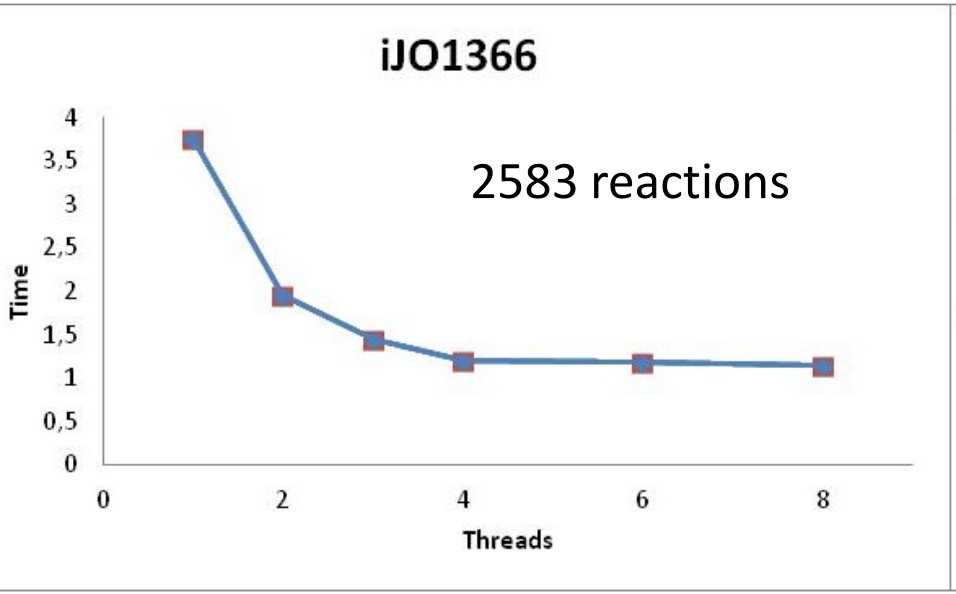
Multicore FVA took: 0.198 min (6 processes)

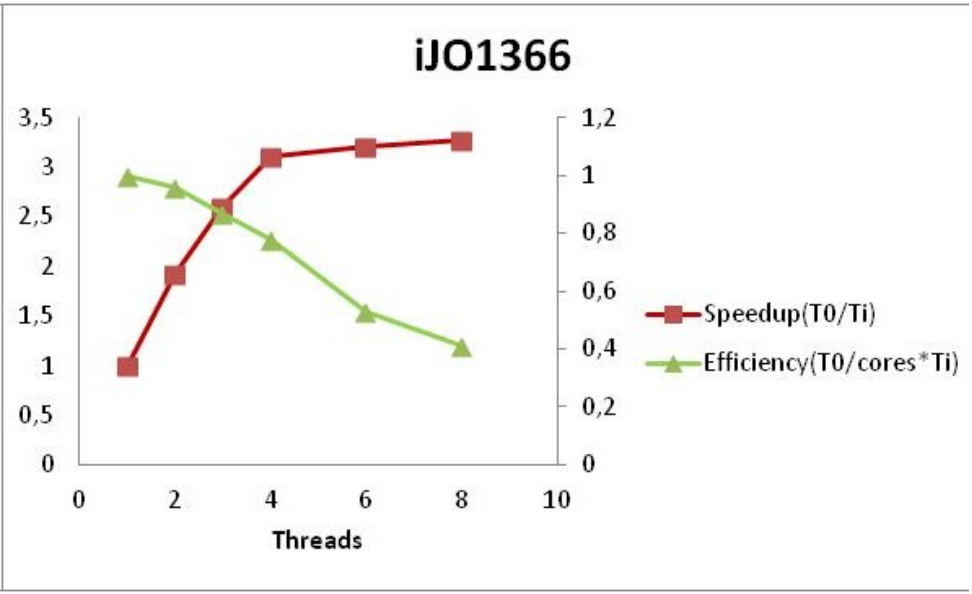


Multithreaded FVA







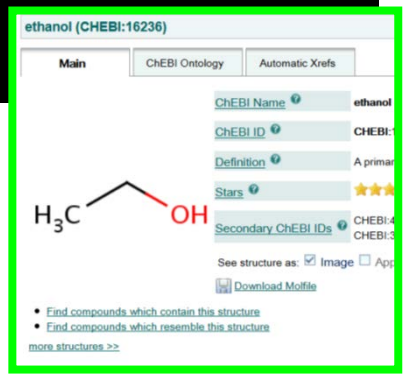


CBMPy console/interactive/scripted

```
> cmod = readSBML3FBC('iJR904.glc.xml, os.getcwd())

> cmod.setReactionLowerBound('R_EX_glc_e_', -10)
> cmod.setReactionUpperBound('R_EX_glc_e_', 0)

> cbm.doFBA(cmod)
   Objective value =  0.92194809505

> S = cmod.getSpecies('M_etoh_c')

> S.addMIRIAMannotation('is', 'ChEBI', 'CHEBI:16236')

> s.miriam.getAndViewUrisForQualifier('is')
```

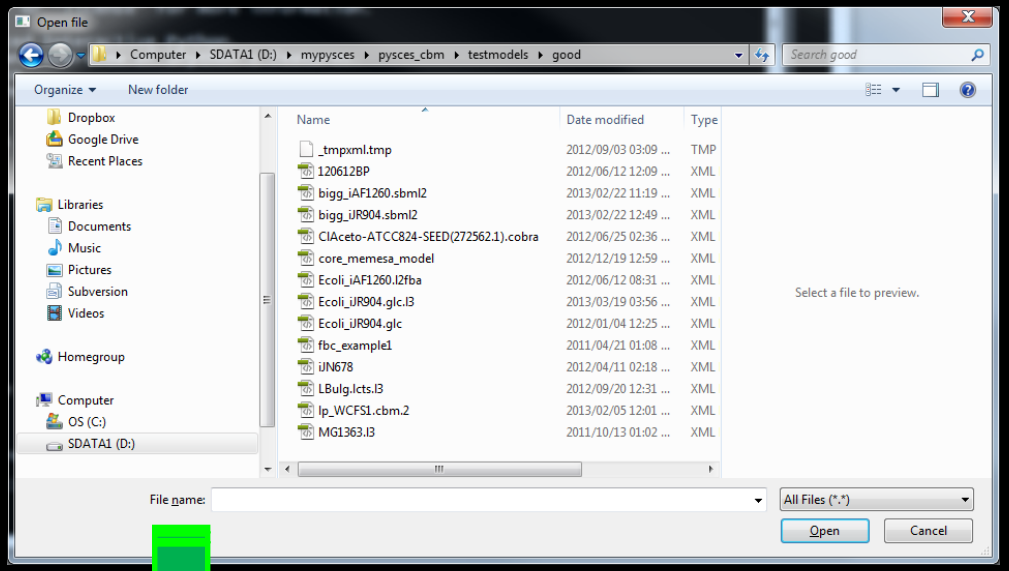
File Edit Search View Tools Options Language Buffers Help

```
R2 = revRr[r_]
R2.setAnnotation('bgoli', '%s split backward half-reaction' % R2.getPid())
rId = R2.getPid()+'_r'
R2.setPid(rId)
R2.reversible = False
cmod.addFluxBound(cbm.CBModel.FluxBound(rId+'_lower', rId, 'GE', 0.0))
cmod.addFluxBound(cbm.CBModel.FluxBound(rId+'_upper', rId, 'LE', abs(ub0)))
```

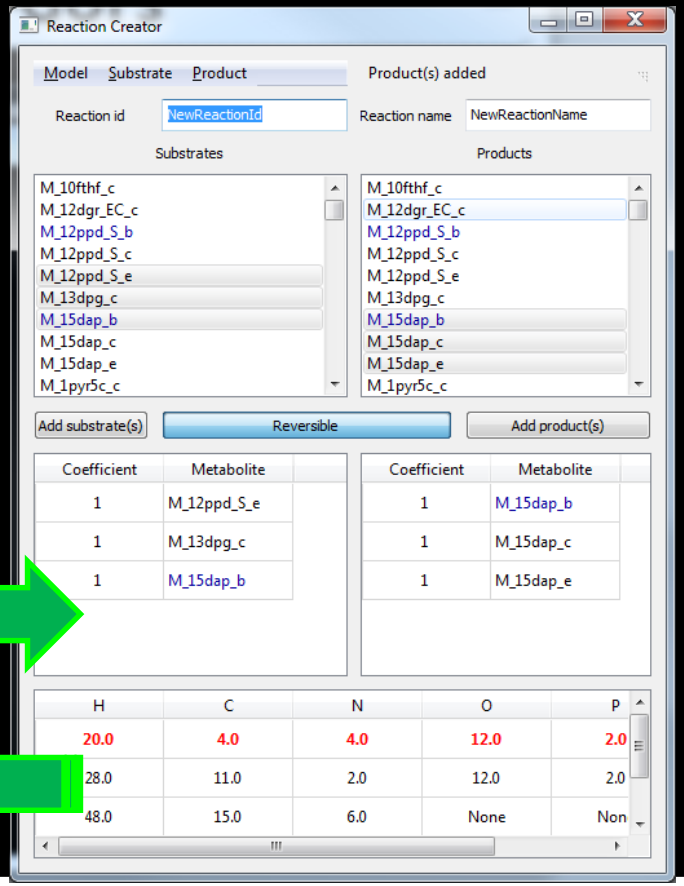


CBMPy: micro-GUI's

```
> cmod = cbm.readSBML3FBC(openFileName())
```




```
> cbm.createReaction(cmod)
```




```
> cbm.doFBA(cmod)
```

CBMPy model GUI

PySCeS-CBM Model Editor - editing: MODELID_1461534 (E. coli iJR904)

File Options Action

ObjValue | ObjStatus | SearchReactions

Run Balance Checker
Run Semantic SBML ▸ Name / Id

Optimize | Min. SumAbsFlux

	Reaction	Name	Flux	d	LB
64	R_ADSK	adenylyl-sulfate kinase	0.233059		0.0
65	R_ADSL1r	adenylsuccinate lyase	0.2894989999		-999999.0
66	R_ADSL2r	adenylosuccinate lyase	0.4779089999		-999999.0
67	R_ADSS	adenylosuccinate synthase	0.2894989999		0.0
68	R_AGDC	N-acetylglucosamine-6-phospha	0.0		0.0
69	R_AGMHE	ADP-D-glycero-D-manno-heptos	0.0252		0.0
74	R_AICART	phosphoribosylaminoimidazoleca	0.5679089999		-999999.0
75	R_AIRC2	phosphoribosylaminoimidazole c	0.4779089999		0.0
76	R_AIRC3	phosphoribosylaminoimidazole c	-0.4779089999		-999999.0
77	R_AKGDH	_2-Oxogluterate dehydrogenas	2.4514793226		0.0
78	R_AKGt2r	_2-oxoglutarate reversible tran	0.0		-999999.0
79	R_ALAALAr	D-alanine-D-alanine ligase (reve	0.0276		-999999.0
80	R_ALAR	alanine racemase	0.0552		-999999.0
81	R_ALATA_D2	D-alanine transaminase	0.0		0.0
82	R_ALATA_L	L-alanine transaminase	-999999.0		-999999.0
83	R_ALATA_L2	alanine transaminase	0.0		0.0
84	R_ALAabc	L-alanine transport via ABC syst	0.0		0.0
85	R_ALAt2r	L-alanine reversible transport vi	0.0		-999999.0
86	R_ALCD19	alcohol dehydrogenase (glycero	0.0		-999999.0
87	R_ALDD19x	aldehyde dehydrogenase (phen	0.0		0.0
88	R_ALDD2x	aldehyde dehydrogenase (aceta	0.0		0.0
89	R_ALLTAH	Allantoate amidinohydrolase	0.0		0.0

Look up reaction name

Session | Reaction | Metabolism | Genes | ReacEdt | MIRIAM

SemanticSBML query

adenylsuccinate lyase

urn:miriam:interpro:IPR004769	http://identifiers.org/interpro/IPR004769
urn:miriam:interpro:IPR019468	http://identifiers.org/interpro/IPR019468
urn:miriam:obo.go:GO:0004019	http://identifiers.org/obo.go/GO:0004019
urn:miriam:ec-code:6.3.4.4	http://identifiers.org/ec-code/6.3.4.4
urn:miriam:insdc:BAH46894.1	http://identifiers.org/insdc/BAH46894.1
urn:miriam:uniprot:C0ZA40	http://identifiers.org/uniprot/C0ZA40
urn:miriam:obo.chebi:CHEBI:22262	http://identifiers.org/obo.chebi/CHEBI:222
urn:miriam:uniprot:Q05911	http://identifiers.org/uniprot/Q05911
urn:miriam:uniprot:P54822	http://identifiers.org/uni
urn:miriam:uniprot:Q21774	http://identifiers.org/uni
urn:miriam:uniprot:P30566	http://identifiers.org/uni
urn:miriam:uniprot:P74384	http://identifiers.org/uni
urn:miriam:uniprot:Q8N142	http://identifiers.org/uni
urn:miriam:uniprot:P30520	http://identifiers.org/uni
urn:miriam:obo.go:GO:0016879	http://identifiers.org/obo

EMBL-EBI — Databases | Tools | Research | Training | Industry

EBI › Databases › InterPro

Jump to: InterProScan Databases Documentation FTP site Help
Advanced search

IPR004769 Adenylosuccinate lyase

Protein matches

UniProtKB Matches: 4985 proteins

Overview: sorted by AC, sorted
Detailed: sorted by AC, sorted
Table: For all matching proteins, of know
Architectures
Accession List
Matches in BioMart

MGI — MGI has a job opening for a biologist.

About Help FAQ

Home | Genes | Phenotypes | Expression | Recombinases

Search | Download | More Resources | Submit Data | Find Mice (IMSR)

Gene Ontology Browser
Term Detail

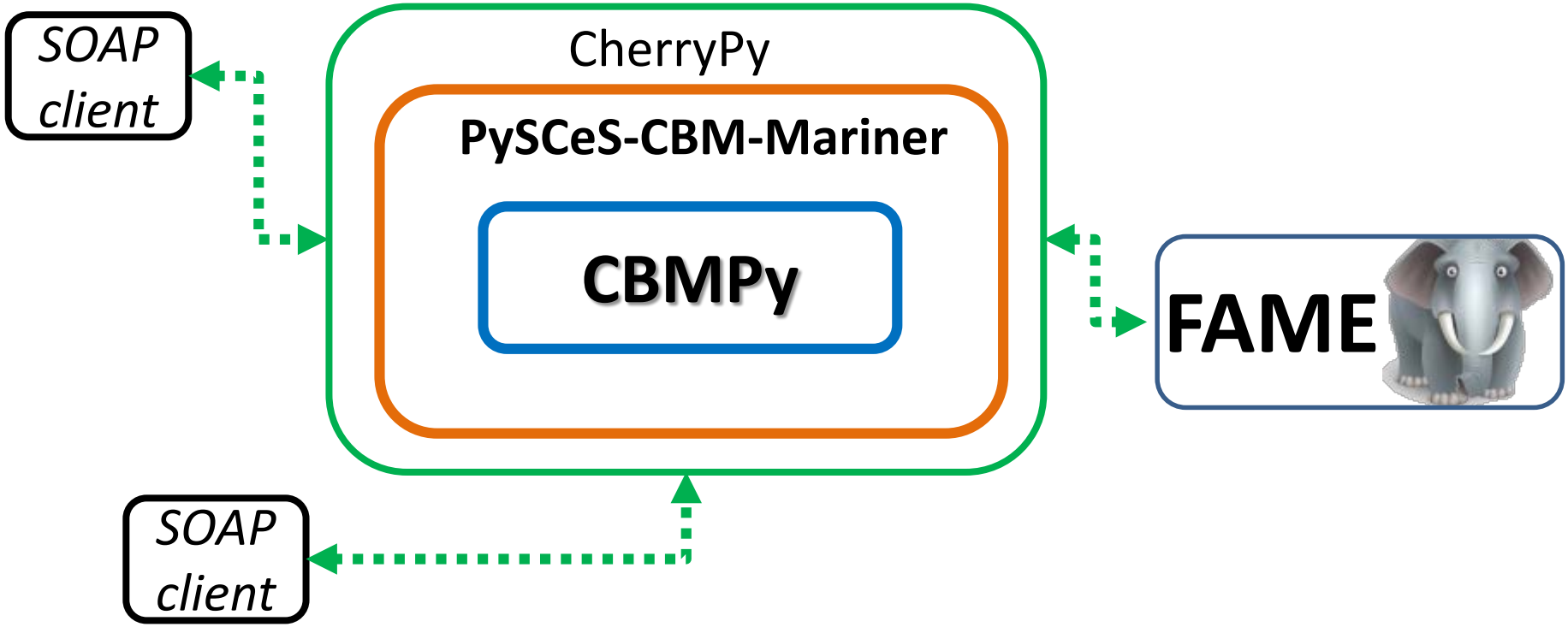
GO term:	adenylosuccinate synthase activity
Synonym:	adenylosuccinate synthetase activity
Synonym:	IMP--aspartate ligase activity
Synonym:	IMP:L-aspartate ligase (GDP-forming)
Synonym:	succinoadenylic kinosynthetase activity
Synonym:	succino-AMP synthetase activity
GO id:	GO:0004019
Definition:	Catalysis of the reaction: L-aspartate + GTP + IMP = N(6)-(1,2-dicarboxyet
Number of paths to term:	1

Resolve urn's to identifiers.org url's

cbm.loadCBGUI(cmod)

CBMPy: web services

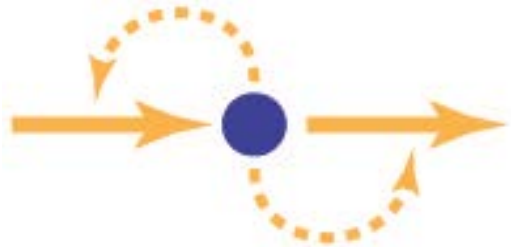
- Web services API exposes core functionality as SOAP web-services in a portable, extensible way.
- Allows for rapid integration with other software, without restricting core development



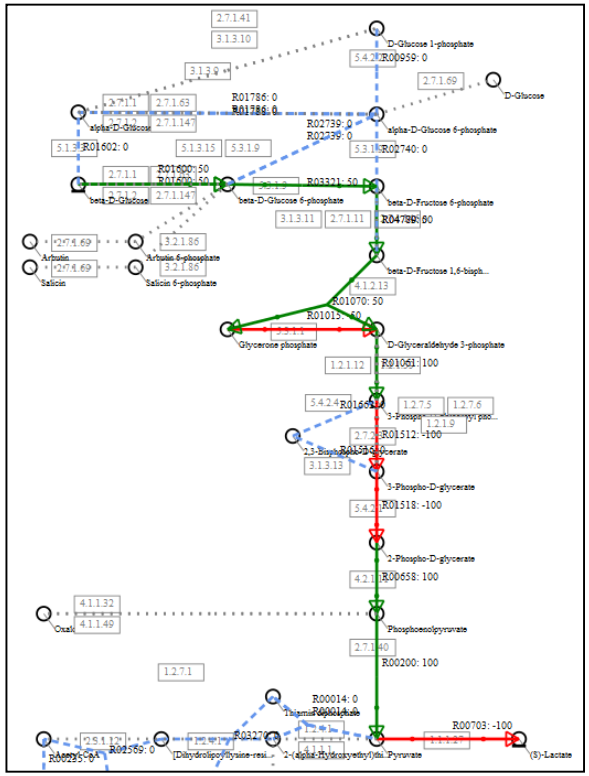
FAME

The **F**lux **A**nalysis and **M**odeling **E**nvironment

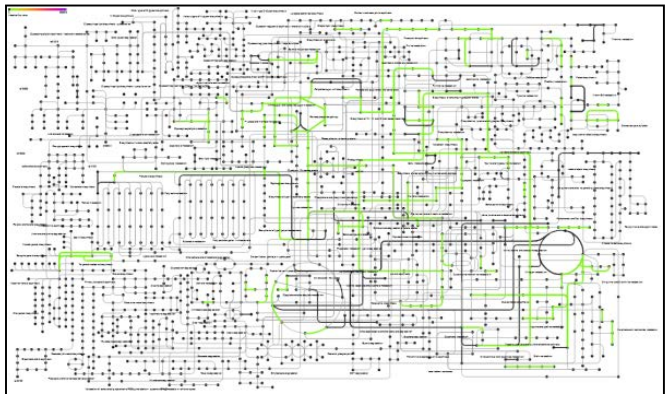
FAME forwards model to **PySCeS-CBM** for solving



PySCeS-CBM solves model; returns results to FAME



Visualization on KEGG maps or user-supplied SVG



Software

Highly accessed Open Access

FAME, the Flux Analysis and Modeling Environment

Joost Boele[1,2], Brett G Olivier[1,2,3] and Bas Teusink[1,2*]

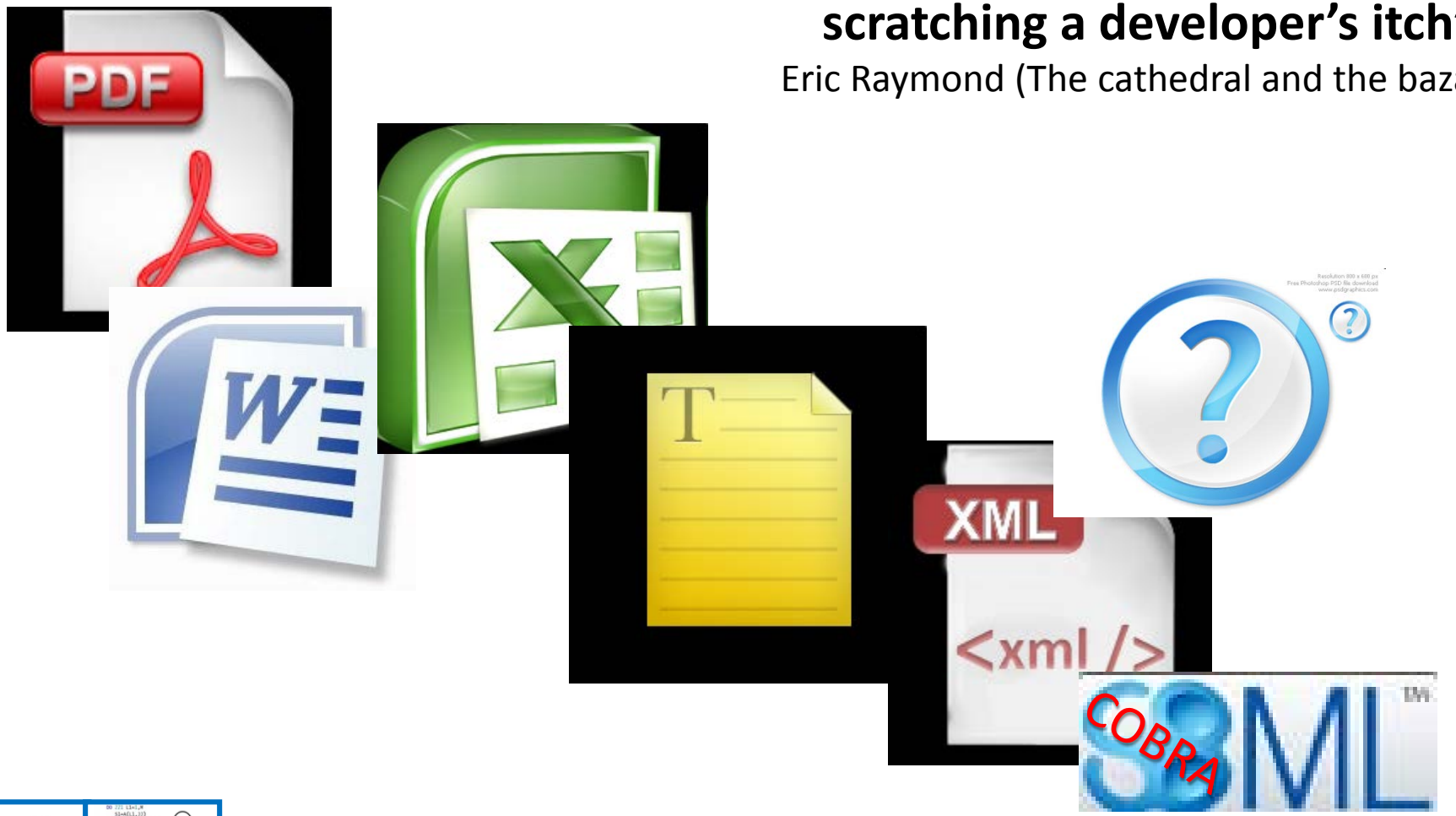
BMC Systems Biology 2012, **6**:8 doi:10.1186/1752-0509-6-8
Published: 30 January 2012

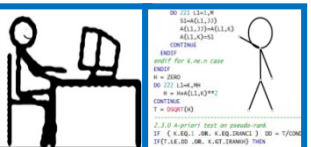
http://f-a-m-e.org/

FBA/GSR models < 2010

“every work of software starts by scratching a developer’s itch”

Eric Raymond (The cathedral and the bazaar)

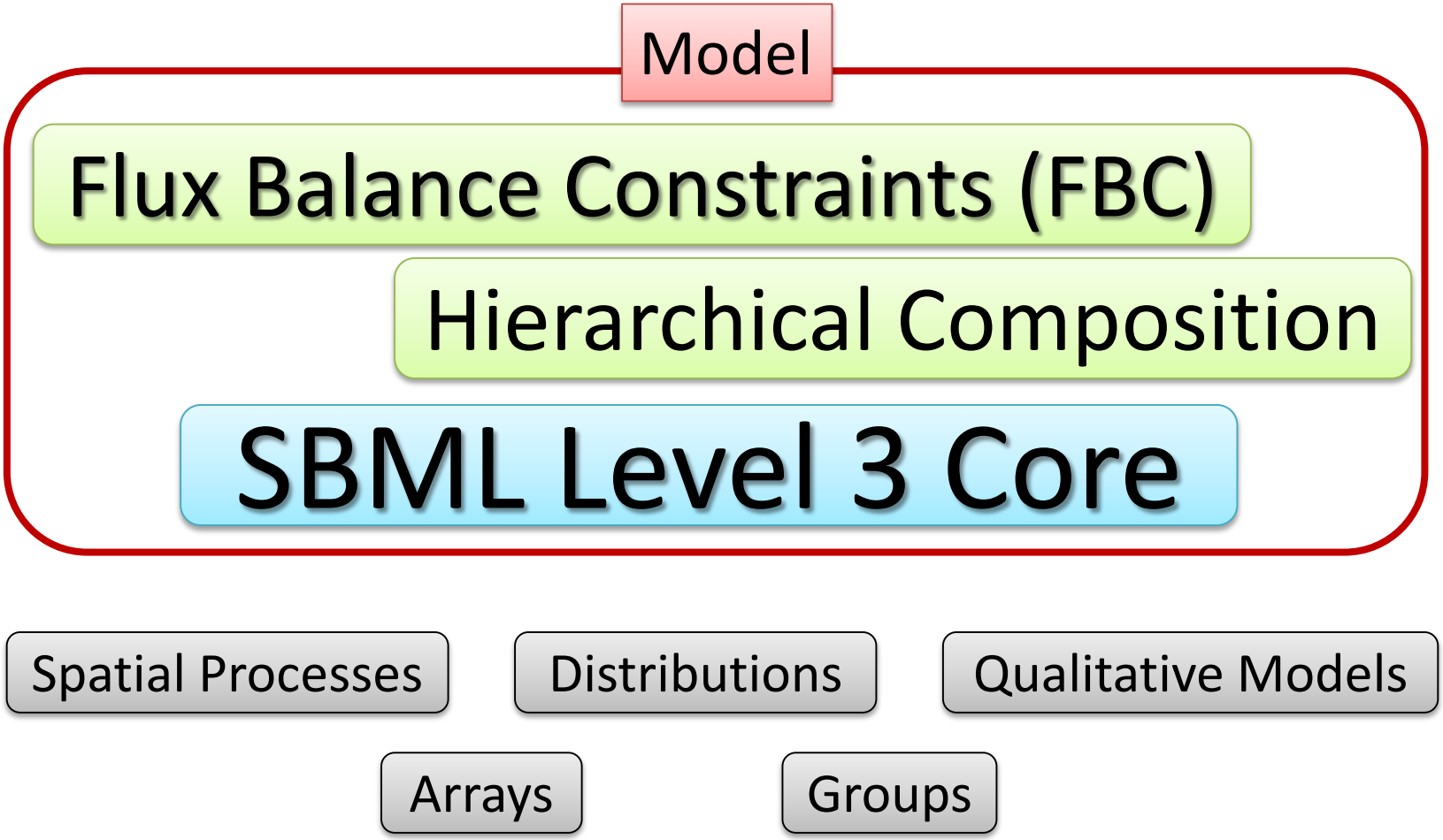




COBRA SBML Level 2: a tool specific dialect!

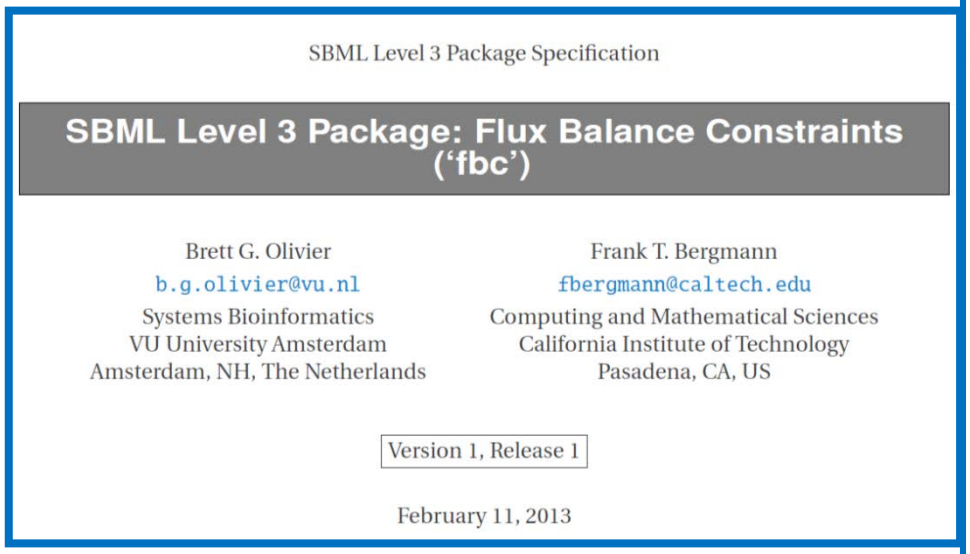
SBML Level 3 Packages

- libSBML has API & bindings for C, Python, C#, Java, MATLAB, R



SBML Level 3 FBC

- Proposed in 2009, **Version 1** specification accepted **March 2013**
- Community driven development process, both **SBML** and **FBA**
- Included in official **libSBML 5.8.0+** release (**sbml.org/downloads**)



SBML Level 3 Package Specification

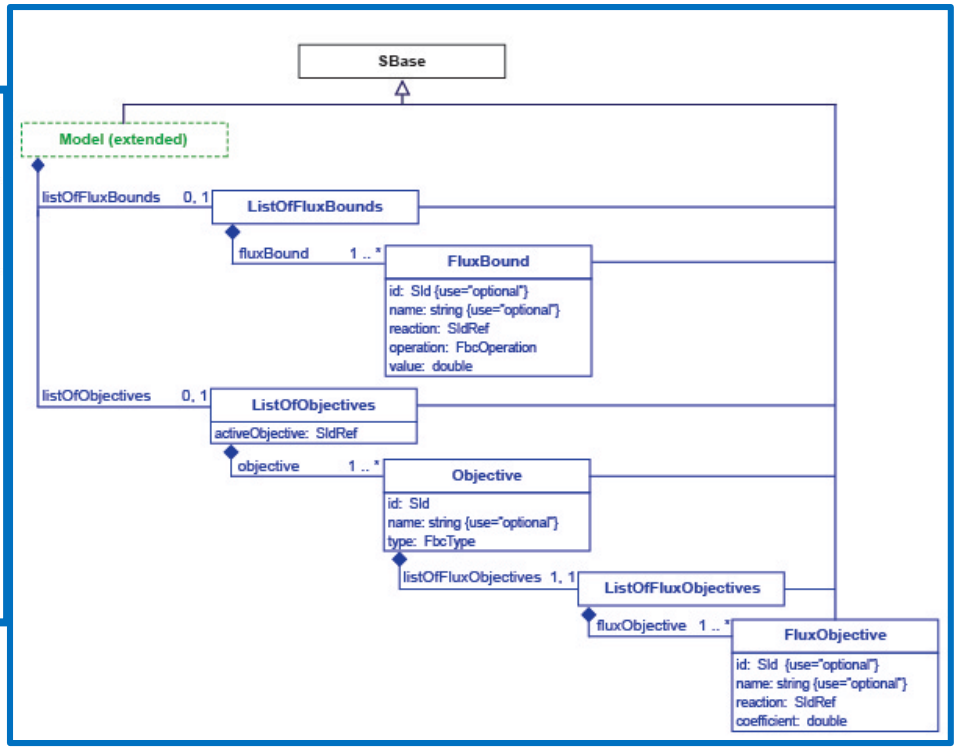
SBML Level 3 Package: Flux Balance Constraints ('fbc')

Brett G. Olivier
b.g.olivier@vu.nl
Systems Bioinformatics
VU University Amsterdam
Amsterdam, NH, The Netherlands

Frank T. Bergmann
fbergmann@caltech.edu
Computing and Mathematical Sciences
California Institute of Technology
Pasadena, CA, US

Version 1, Release 1

February 11, 2013



Available from CO.MBINE.org
http://identifiers.org/combine.specifications/sbml.level-3.version-1.fbc.version-1.release-1

A CMPy quick reference guide