

pyLM
v1.1

Generated by Doxygen 1.8.6

Wed Nov 9 2016 10:47:46

Contents

1	Description	1
2	Namespace Index	3
2.1	Packages	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	peekFile Namespace Reference	11
6.1.1	Variable Documentation	11
6.1.1.1	args	11
6.1.1.2	n	11
6.1.1.3	parser	11
6.1.1.4	t	11
6.1.1.5	tf	11
6.2	pyLM Namespace Reference	11
6.2.1	Detailed Description	12
6.2.2	Variable Documentation	12
6.2.2.1	__all__	12
6.3	pyLM.CME Namespace Reference	12
6.3.1	Function Documentation	12
6.3.1.1	tqdm	12
6.4	pyLM.ipynInterface Namespace Reference	12
6.4.1	Function Documentation	12
6.4.1.1	displayRDMESetup	12
6.4.1.2	getReactionString	12

6.4.1.3	writeTable	13
6.5	pyLM.LMLogger Namespace Reference	13
6.5.1	Function Documentation	13
6.5.1.1	setLMLogConsole	13
6.5.1.2	setLMLogFile	14
6.5.1.3	setLMLoggerLevel	14
6.5.2	Variable Documentation	14
6.5.2.1	LMformatter	14
6.5.2.2	LMLogger	14
6.5.2.3	nullHandlerLM	14
6.6	pyLM.RDME Namespace Reference	14
6.6.1	Function Documentation	14
6.6.1.1	tqdm	14
6.7	pyLM.units Namespace Reference	14
6.7.1	Function Documentation	15
6.7.1.1	angstrom	15
6.7.1.2	cm	15
6.7.1.3	day	15
6.7.1.4	hr	16
6.7.1.5	micron	16
6.7.1.6	microsecond	16
6.7.1.7	minute	16
6.7.1.8	mm	16
6.7.1.9	ms	16
6.7.1.10	nm	16
6.7.1.11	ns	17
6.7.1.12	second	17
6.8	pySTDLM Namespace Reference	17
6.8.1	Detailed Description	17
6.8.2	Variable Documentation	18
6.8.2.1	__all__	18
6.9	pySTDLM.CellArranger Namespace Reference	18
6.10	pySTDLM.Cells Namespace Reference	18
6.11	pySTDLM.Distributions Namespace Reference	18
6.12	pySTDLM.Fitting Namespace Reference	18
6.12.1	Function Documentation	19
6.12.1.1	DistributionComparator	19
6.12.2	Variable Documentation	19
6.12.2.1	binwidth	19
6.12.2.2	counts	19

6.12.2.3	measure	19
6.13	pySTDLM.NetworkVisualization Namespace Reference	19
6.13.1	Function Documentation	19
6.13.1.1	plotCMEDynamicReactionNetwork	19
6.13.1.2	plotCMEReactionNetwork	20
6.13.1.3	plotRDMEReactionNetwork	20
6.13.1.4	plotReactionNetwork	20
6.13.2	Variable Documentation	20
6.13.2.1	hasGexf	20
6.14	pySTDLM.OnTheFly Namespace Reference	20
6.15	pySTDLM.PostProcessing Namespace Reference	21
6.15.1	Function Documentation	21
6.15.1.1	closeLMFile	21
6.15.1.2	getAvgVarTrace	22
6.15.1.3	getHistogram	22
6.15.1.4	getOccupancyKymograph	22
6.15.1.5	getPhaseSpace	22
6.15.1.6	getSpecieTrace	23
6.15.1.7	getTimesteps	23
6.15.1.8	openLMFile	23
6.15.1.9	plotAvgVar	23
6.15.1.10	plotAvgVarFromFile	24
6.15.1.11	plotOccupancyKymograph	24
6.15.1.12	plotPhaseSpace	24
6.15.1.13	plotTrace	24
6.15.1.14	plotTraceFromFile	25
6.15.1.15	showAvgVar	25
6.15.1.16	showAvgVarFromFile	25
6.15.1.17	showOccupancyKymograph	25
6.15.1.18	showTrace	26
6.15.1.19	showTraceFromFile	26
6.16	pySTDLM.SBMLReader Namespace Reference	26
6.16.1	Function Documentation	26
6.16.1.1	getRate	26
6.16.1.2	importSBMLModel	26
6.16.1.3	importSBMLModelL3V1	27
6.16.1.4	readSBMLtoCME	27
6.16.1.5	readSBMLtoRDME	27
6.16.1.6	readSBMLtoRegion	27
6.17	pySTDLM.StandardCells Namespace Reference	27

6.17.1	Function Documentation	28
6.17.1.1	buildDividingEColiCell	28
6.17.1.2	buildEColiCell	28
6.17.1.3	buildFilamentousEColiCell	28
6.17.1.4	buildSphericalEColiCell	28
6.17.1.5	packFastGrowingEcoli	29
6.17.1.6	packSlowGrowingEcoli	30
6.18	pySTDLM.StandardReactions Namespace Reference	30
6.18.1	Function Documentation	30
6.18.1.1	addMembraneTransporter	30
6.18.1.2	addMichaelisMenten	31
6.18.1.3	addPassiveTransport	31
6.18.1.4	addReversibleMichaelisMenten	31
6.18.1.5	createExpressionModel	32
6.18.2	Variable Documentation	32
6.18.2.1	hasBio	32
6.19	pySTDLM.StandardReactionSystems Namespace Reference	32
6.19.1	Function Documentation	33
6.19.1.1	addLacTwoStateSystem	33
6.19.1.2	addMinDESystem	33
6.19.1.3	addPTSPathway	33
7	Class Documentation	35
7.1	pySTDLM.Cells.BoxCell Class Reference	35
7.1.1	Detailed Description	35
7.1.2	Constructor & Destructor Documentation	36
7.1.2.1	__init__	36
7.1.3	Member Function Documentation	36
7.1.3.1	computeVolume	36
7.1.3.2	internalAddToSim	36
7.1.3.3	internalModelCell	36
7.1.3.4	internalTranslateCell	36
7.1.4	Member Data Documentation	36
7.1.4.1	depth	36
7.1.4.2	height	36
7.1.4.3	maxs	36
7.1.4.4	membraneThickness	36
7.1.4.5	mins	36
7.1.4.6	name	36
7.1.4.7	p1	36

7.1.4.8	p2	36
7.1.4.9	width	36
7.2	pySTDLM.Cells.CapsuleCell Class Reference	36
7.2.1	Detailed Description	37
7.2.2	Constructor & Destructor Documentation	37
7.2.2.1	__init__	37
7.2.3	Member Function Documentation	37
7.2.3.1	computeVolume	37
7.2.3.2	internalAddToSim	37
7.2.3.3	internalModelCell	37
7.2.3.4	internalTranslateCell	37
7.2.4	Member Data Documentation	37
7.2.4.1	length	37
7.2.4.2	maxs	37
7.2.4.3	membraneThickness	37
7.2.4.4	mins	37
7.2.4.5	name	37
7.2.4.6	p1	37
7.2.4.7	p2	37
7.2.4.8	radius	38
7.3	pySTDLM.Cells.CapsuleShellCell Class Reference	38
7.3.1	Detailed Description	38
7.3.2	Constructor & Destructor Documentation	38
7.3.2.1	__init__	38
7.3.3	Member Function Documentation	38
7.3.3.1	computeVolume	38
7.3.3.2	internalAddToSim	39
7.3.3.3	internalModelCell	39
7.3.4	Member Data Documentation	39
7.3.4.1	length	39
7.3.4.2	maxs	39
7.3.4.3	membraneThickness	39
7.3.4.4	mins	39
7.3.4.5	name	39
7.3.4.6	p1	39
7.3.4.7	p2	39
7.3.4.8	radius	39
7.4	pySTDLM.CellArranger.CellArranger Class Reference	39
7.4.1	Detailed Description	40
7.4.2	Constructor & Destructor Documentation	40

7.4.2.1	<code>__init__</code>	40
7.4.3	Member Function Documentation	40
7.4.3.1	<code>addToSimulation</code>	40
7.4.3.2	<code>cellFactory</code>	40
7.4.3.3	<code>packVolume</code>	40
7.4.4	Member Data Documentation	40
7.4.4.1	<code>allowedCellTypes</code>	40
7.4.4.2	<code>allowedForms</code>	41
7.4.4.3	<code>arrangerAttributes</code>	41
7.4.4.4	<code>cellAttributes</code>	41
7.4.4.5	<code>cells</code>	41
7.4.4.6	<code>cellType</code>	41
7.4.4.7	<code>packerName</code>	41
7.5	<code>pySTDLM.Cells.CellShape</code> Class Reference	41
7.5.1	Detailed Description	42
7.5.2	Constructor & Destructor Documentation	42
7.5.2.1	<code>__init__</code>	42
7.5.3	Member Function Documentation	42
7.5.3.1	<code>addToSimulation</code>	42
7.5.3.2	<code>boundingBox</code>	42
7.5.3.3	<code>createModelCell</code>	42
7.5.3.4	<code>getVolume</code>	42
7.5.3.5	<code>setRegions</code>	42
7.5.3.6	<code>translateCell</code>	43
7.5.4	Member Data Documentation	43
7.5.4.1	<code>cyttype</code>	43
7.5.4.2	<code>maxs</code>	43
7.5.4.3	<code>membraneThickness</code>	43
7.5.4.4	<code>memtype</code>	43
7.5.4.5	<code>mins</code>	43
7.5.4.6	<code>name</code>	43
7.5.4.7	<code>volume</code>	43
7.6	<code>pyLM.CME.CMESimulation</code> Class Reference	43
7.6.1	Detailed Description	44
7.6.2	Constructor & Destructor Documentation	44
7.6.2.1	<code>__init__</code>	44
7.6.3	Member Function Documentation	44
7.6.3.1	<code>addConcentration</code>	44
7.6.3.2	<code>addParticles</code>	45
7.6.3.3	<code>addReaction</code>	45

7.6.3.4	buildReactionModel	45
7.6.3.5	defineSpecies	45
7.6.3.6	run	45
7.6.3.7	runMPI	46
7.6.3.8	runSolver	46
7.6.3.9	save	46
7.6.3.10	setRandomSeed	46
7.6.3.11	setSimulationTime	46
7.6.3.12	setTimestep	46
7.6.3.13	setWriteInterval	46
7.6.4	Member Data Documentation	47
7.6.4.1	filename	47
7.6.4.2	initial_counts	47
7.6.4.3	name	47
7.6.4.4	parameters	47
7.6.4.5	particleMap	47
7.6.4.6	reactions	47
7.6.4.7	replicates	47
7.6.4.8	species_id	47
7.6.4.9	volume	47
7.7	pyLM.RDME.CMESimulation Class Reference	47
7.7.1	Detailed Description	48
7.7.2	Constructor & Destructor Documentation	48
7.7.2.1	__init__	48
7.7.3	Member Function Documentation	48
7.7.3.1	addConcentration	48
7.7.3.2	addParticles	48
7.7.3.3	addReaction	48
7.7.3.4	defineSpecies	49
7.7.4	Member Data Documentation	49
7.7.4.1	filename	49
7.7.4.2	initial_counts	49
7.7.4.3	name	49
7.7.4.4	parameters	49
7.7.4.5	particleMap	49
7.7.4.6	reactions	49
7.7.4.7	replicates	49
7.7.4.8	species_id	49
7.7.4.9	volume	49
7.8	pySTDLM.Fitting.Comparator Class Reference	49

7.8.1	Constructor & Destructor Documentation	50
7.8.1.1	__init__	50
7.8.2	Member Function Documentation	50
7.8.2.1	computeMetric	50
7.8.3	Member Data Documentation	50
7.8.3.1	specie	50
7.9	pySTDLM.Fitting.Fitter Class Reference	50
7.9.1	Constructor & Destructor Documentation	51
7.9.1.1	__init__	51
7.9.2	Member Function Documentation	51
7.9.2.1	addExperiment	51
7.9.2.2	addParameter	51
7.9.2.3	performFit	51
7.9.2.4	saveFit	51
7.9.3	Member Data Documentation	51
7.9.3.1	experiments	51
7.9.3.2	hasFit	51
7.9.3.3	parameters	51
7.10	pySTDLM.CellArranger.LooseGridArranger Class Reference	51
7.10.1	Detailed Description	51
7.10.2	Constructor & Destructor Documentation	52
7.10.2.1	__init__	52
7.10.3	Member Function Documentation	52
7.10.3.1	intPackVolume	52
7.11	pySTDLM.Fitting.MomentComparator Class Reference	52
7.11.1	Constructor & Destructor Documentation	52
7.11.1.1	__init__	52
7.11.2	Member Function Documentation	52
7.11.2.1	computeMetric	52
7.11.3	Member Data Documentation	52
7.11.3.1	moment	52
7.11.3.2	sem	52
7.11.3.3	value	52
7.12	pyLM.LMLogger.NullHandler Class Reference	53
7.12.1	Member Function Documentation	53
7.12.1.1	emit	53
7.13	pySTDLM.Distributions.ParticleGenerator Class Reference	53
7.13.1	Detailed Description	53
7.13.2	Constructor & Destructor Documentation	53
7.13.2.1	__init__	53

7.13.3	Member Function Documentation	54
7.13.3.1	addParticlesToRegion	54
7.13.4	Member Data Documentation	54
7.13.4.1	addedLast	54
7.13.4.2	generator	54
7.13.4.3	particlename	54
7.13.4.4	totalAdded	54
7.14	pySTDLM.CellArranger.RandomArranger Class Reference	54
7.14.1	Detailed Description	55
7.14.2	Constructor & Destructor Documentation	55
7.14.2.1	__init__	55
7.14.3	Member Function Documentation	55
7.14.3.1	doesIntersect	55
7.14.3.2	intPackVolume	55
7.14.3.3	newBBox	55
7.14.3.4	outsideDomain	55
7.15	pySTDLM.CellArranger.RandomSphericalArranger Class Reference	55
7.15.1	Detailed Description	56
7.15.2	Constructor & Destructor Documentation	56
7.15.2.1	__init__	56
7.15.3	Member Function Documentation	56
7.15.3.1	intPackVolume	56
7.16	pyLM.RDME.RDMERegion Class Reference	56
7.16.1	Detailed Description	56
7.16.2	Constructor & Destructor Documentation	57
7.16.2.1	__init__	57
7.16.3	Member Function Documentation	58
7.16.3.1	addReaction	58
7.16.3.2	getReactionCount	58
7.16.3.3	setDefaultDiffusionRate	58
7.16.3.4	setDiffusionRate	58
7.16.4	Member Data Documentation	59
7.16.4.1	defaultDiffusionRate	59
7.16.4.2	diffusionRate	59
7.16.4.3	name	59
7.16.4.4	reactions	59
7.17	pyLM.RDME.RDMESimulation Class Reference	59
7.17.1	Detailed Description	61
7.17.2	Constructor & Destructor Documentation	61
7.17.2.1	__init__	61

7.17.3	Member Function Documentation	61
7.17.3.1	addCuboidRegion	61
7.17.3.2	addParticleAt	61
7.17.3.3	addParticleAtIdx	61
7.17.3.4	addParticles	62
7.17.3.5	addRegion	62
7.17.3.6	addShape	62
7.17.3.7	buildCapsidCell	62
7.17.3.8	buildDifference	63
7.17.3.9	buildDiffusionModel	63
7.17.3.10	buildEllipse	63
7.17.3.11	buildPoint	63
7.17.3.12	buildReactionModel	63
7.17.3.13	buildSpatialModel	63
7.17.3.14	buildSphere	63
7.17.3.15	buildSphericalCell	63
7.17.3.16	buildVector	64
7.17.3.17	defineSpecies	64
7.17.3.18	getLattice	64
7.17.3.19	getLatticeSite	64
7.17.3.20	modifyRegion	64
7.17.3.21	packRegion	65
7.17.3.22	run	65
7.17.3.23	runMPI	65
7.17.3.24	runSolver	65
7.17.3.25	save	65
7.17.3.26	setLatticeSite	66
7.17.3.27	setLatticeWriteInterval	66
7.17.3.28	setRandomSeed	66
7.17.3.29	setSimulationTime	66
7.17.3.30	setTimestep	66
7.17.3.31	setTransitionRate	66
7.17.3.32	setTwoWayTransitionRate	66
7.17.3.33	setWriteInterval	67
7.17.3.34	siteVolume	67
7.17.4	Member Data Documentation	67
7.17.4.1	continousDimensions	67
7.17.4.2	customAddedParticleList	67
7.17.4.3	filename	67
7.17.4.4	hasBeenDiscretized	67

7.17.4.5	<code>initial_counts</code>	67
7.17.4.6	<code>lattice</code>	67
7.17.4.7	<code>latticeSpacing</code>	67
7.17.4.8	<code>lm_builder</code>	67
7.17.4.9	<code>name</code>	67
7.17.4.10	<code>parameters</code>	67
7.17.4.11	<code>particleMap</code>	67
7.17.4.12	<code>regions</code>	67
7.17.4.13	<code>replicates</code>	67
7.17.4.14	<code>siteTypes</code>	67
7.17.4.15	<code>species_id</code>	67
7.17.4.16	<code>transitionRates</code>	67
7.18	<code>pySTDLM.Fitting.SimulatedAnnealingFitter</code> Class Reference	67
7.18.1	Constructor & Destructor Documentation	68
7.18.1.1	<code>__init__</code>	68
7.18.2	Member Function Documentation	68
7.18.2.1	<code>performFit</code>	68
7.18.2.2	<code>saveFit</code>	68
7.19	<code>pySTDLM.Cells.SphereCell</code> Class Reference	68
7.19.1	Detailed Description	69
7.19.2	Constructor & Destructor Documentation	69
7.19.2.1	<code>__init__</code>	69
7.19.3	Member Function Documentation	69
7.19.3.1	<code>computeVolume</code>	69
7.19.3.2	<code>internalAddToSim</code>	69
7.19.3.3	<code>internalModelCell</code>	69
7.19.3.4	<code>internalTranslateCell</code>	69
7.19.4	Member Data Documentation	69
7.19.4.1	<code>maxs</code>	69
7.19.4.2	<code>membraneThickness</code>	69
7.19.4.3	<code>mins</code>	69
7.19.4.4	<code>name</code>	69
7.19.4.5	<code>origin</code>	69
7.19.4.6	<code>radius</code>	69
7.20	<code>pySTDLM.Fitting.SweepFitter</code> Class Reference	69
7.20.1	Constructor & Destructor Documentation	70
7.20.1.1	<code>__init__</code>	70
7.20.2	Member Function Documentation	70
7.20.2.1	<code>performFit</code>	70
7.20.2.2	<code>saveFit</code>	70

7.20.2.3	setParameterDivisions	70
7.20.3	Member Data Documentation	70
7.20.3.1	divisions	70
7.20.3.2	hasFit	70
7.21	pySTDLM.CellArranger.TightGridArranger Class Reference	70
7.21.1	Detailed Description	71
7.21.2	Constructor & Destructor Documentation	71
7.21.2.1	__init__	71
7.21.3	Member Function Documentation	71
7.21.3.1	intPackVolume	71
7.22	pySTDLM.Fitting.TraceComparator Class Reference	71
7.22.1	Constructor & Destructor Documentation	71
7.22.1.1	__init__	71
7.22.2	Member Function Documentation	71
7.22.2.1	computeMetric	72
7.22.3	Member Data Documentation	72
7.22.3.1	comparator	72
7.22.3.2	data	72
7.22.3.3	times	72
8	File Documentation	73
8.1	__init__.py File Reference	73
8.2	__init__.py File Reference	73
8.3	CellArranger.py File Reference	73
8.4	Cells.py File Reference	74
8.5	CME.py File Reference	74
8.6	Distributions.py File Reference	74
8.7	Fitting.py File Reference	75
8.8	ipyInterface.py File Reference	75
8.9	LMLogger.py File Reference	75
8.10	NetworkVisualization.py File Reference	76
8.11	OnTheFly.py File Reference	76
8.12	peekFile.py File Reference	77
8.13	PostProcessing.py File Reference	77
8.14	RDME.py File Reference	78
8.15	SBMLReader.py File Reference	78
8.16	StandardCells.py File Reference	79
8.17	StandardReactions.py File Reference	79
8.18	StandardReactionSystems.py File Reference	79
8.19	units.py File Reference	80

Chapter 1

Description

[pyLM](#) is a Problem Solving Environment (PSE) for biological simulations [1]. Written in Python, it wraps and extends the highly optimized multi-GPU [Lattice Microbes](#) stochastic simulation software [2,3,4]. The PSE is comprised of a base set of functionality to set up, monitor and modify simulations, as well as a set of standard post-processing routines that interface to other Python packages, including NumPy, SciPy, H5py, iGraph to name a few. See [1] for additional information as well as the user guide on the main [website](#). If you use [pyLM](#) in your simulations, please cite references [1] and [3] below.

References

1. J.R. Peterson, M.J. Hallock, J.A. Cole and Z. Luthey-Schulten. A Problem Solving Environment for Stochastic Biological Simulations. *PyHPC 2013 at Supercomputing 2013*, 2013.
2. E. Roberts, J.E. Stone, L. Sepulveda, W.W. Hwu, and Z. Luthey-Schulten. Long time-scale simulations of in vivo diffusion using GPU hardware. In *Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing*, 2009.
3. E. Roberts, J. E. Stone, and Z. Luthey-Schulten. Lattice microbes: high-performance stochastic simulation method for the reaction-diffusion master equation. *J. Comp. Chem.*, 32(3), 245-55, 2013.
4. M.J. Hallock, J.E. Stone, E. Roberts, C. Fry and Z. Luthey-Schulten. Simulation of reaction diffusion processes over biologically-relevant size and time scales using multi-GPU workstations *Parallel Comput.* 40:86-99, 2014, doi:10.1016/j.parco.2014.03.009.

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

peekFile	11
pyLM	
The problem solving environment which interfaces to Lattice Microbes simulation software	11
pyLM.CME	12
pyLM.ipynInterface	12
pyLM.LMLogger	13
pyLM.RDME	14
pyLM.units	14
pySTDLM	
A standard library of functionality that is commonly used in biological simulations	17
pySTDLM.CellArranger	18
pySTDLM.Cells	18
pySTDLM.Distributions	18
pySTDLM.Fitting	18
pySTDLM.NetworkVisualization	19
pySTDLM.OnTheFly	20
pySTDLM.PostProcessing	21
pySTDLM.SBMLReader	26
pySTDLM.StandardCells	27
pySTDLM.StandardReactions	30
pySTDLM.StandardReactionSystems	32

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

pySTDLM.CellArranger.CellArranger	39
pySTDLM.CellArranger.LooseGridArranger	51
pySTDLM.CellArranger.RandomArranger	54
pySTDLM.CellArranger.RandomSphericalArranger	55
pySTDLM.CellArranger.TightGridArranger	70
pySTDLM.Cells.CellShape	41
pySTDLM.Cells.BoxCell	35
pySTDLM.Cells.CapsuleCell	36
pySTDLM.Cells.CapsuleShellCell	38
pySTDLM.Cells.SphereCell	68
pyLM.CME.CMESimulation	43
pyLM.RDME.CMESimulation	47
pySTDLM.Fitting.Comparator	49
pySTDLM.Fitting.MomentComparator	52
pySTDLM.Fitting.TraceComparator	71
pySTDLM.Fitting.Fitter	50
pySTDLM.Fitting.SimulatedAnnealingFitter	67
pySTDLM.Fitting.SweepFitter	69
Handler	
pyLM.LMLogger.NullHandler	53
pySTDLM.Distributions.ParticleGenerator	53
pyLM.RDME.RDMERegion	56
pyLM.RDME.RDMESimulation	59

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

pySTDLM.Cells.BoxCell	
A representation for a spherical cell	35
pySTDLM.Cells.CapsuleCell	
A representation for a spherical cell	36
pySTDLM.Cells.CapsuleShellCell	
A representation for a spherical cell	38
pySTDLM.CellArranger.CellArranger	
A base class for laying out cells in a 3D space	39
pySTDLM.Cells.CellShape	
Base class for a particular cell type for use in CellArranger (and later more)	41
pyLM.CME.CMESimulation	
A CME simulation that contains reactions and species	43
pyLM.RDME.CMESimulation	
A CME simulation that contains reactions and species	47
pySTDLM.Fitting.Comparator	49
pySTDLM.Fitting.Fitter	50
pySTDLM.CellArranger.LooseGridArranger	
An arranger that orients cells in square grid with padding in between	51
pySTDLM.Fitting.MomentComparator	52
pyLM.LMLogger.NullHandler	53
pySTDLM.Distributions.ParticleGenerator	
A particle generator that adds particles to a particular region of a simulation	53
pySTDLM.CellArranger.RandomArranger	
An arranger that randomly orients a number of cells in a regin until either a count or a volume fraction target is met	54
pySTDLM.CellArranger.RandomSphericalArranger	
An arranger that randomly orients a number of cells in a sphere inside a regin until either a count or a volume fraction target is met	55
pyLM.RDME.RDMERegion	
A class that represents a type region of an RDME simulation	56
pyLM.RDME.RDMESimulation	
A class that contains all regions, reactions, diffusions and rules for a RDME simulation	59
pySTDLM.Fitting.SimulatedAnnealingFitter	67
pySTDLM.Cells.SphereCell	
A representation for a spherical cell	68
pySTDLM.Fitting.SweepFitter	69
pySTDLM.CellArranger.TightGridArranger	
An arranger that orients cells in square grid	70

[pySTDLM.Fitting.TraceComparator](#) 71

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

pyLM/__init__.py	73
pySTDLM/__init__.py	73
CellArranger.py	73
Cells.py	74
CME.py	74
Distributions.py	74
Fitting.py	75
ipyInterface.py	75
LMLogger.py	75
NetworkVisualization.py	76
OnTheFly.py	76
peekFile.py	77
PostProcessing.py	77
RDME.py	78
SBMLReader.py	78
StandardCells.py	79
StandardReactions.py	79
StandardReactionSystems.py	79
units.py	80

Chapter 6

Namespace Documentation

6.1 peekFile Namespace Reference

Variables

- tuple `parser` = `argparse.ArgumentParser(description="Print info on simulation file")`
- tuple `args` = `parser.parse_args()`
- tuple `tf` = `float(hdf['Parameters'].attrs['maxTime'])`
- list `t` = `sim['LatticeTimes']`
- tuple `n` = `len(sim['Lattice'])`

6.1.1 Variable Documentation

6.1.1.1 tuple `peekFile.args` = `parser.parse_args()`

6.1.1.2 tuple `peekFile.n` = `len(sim['Lattice'])`

6.1.1.3 tuple `peekFile.parser` = `argparse.ArgumentParser(description="Print info on simulation file")`

6.1.1.4 list `peekFile.t` = `sim['LatticeTimes']`

6.1.1.5 tuple `peekFile.tf` = `float(hdf['Parameters'].attrs['maxTime'])`

6.2 pyLM Namespace Reference

The problem solving environment which interfaces to Lattice Microbes simulation software.

Namespaces

- [CME](#)
- [ipyInterface](#)
- [LMLogger](#)
- [RDME](#)
- [units](#)

Variables

- list `__all__` = `['LMLogger', 'CME', 'RDME', 'units', 'ipyInterface']`

6.2.1 Detailed Description

The problem solving environment which interfaces to Lattice Microbes simulation software.

6.2.2 Variable Documentation

6.2.2.1 list `pyLM.__all__ = ['LMLogger', 'CME', 'RDME', 'units', 'ipyInterface']`

6.3 pyLM.CME Namespace Reference

Classes

- class [CMESimulation](#)
A *CME* simulation that contains reactions and species.

Functions

- def [tqdm](#)

6.3.1 Function Documentation

6.3.1.1 `def pyLM.CME.tqdm (x, ascii=False)`

6.4 pyLM.ipynInterface Namespace Reference

Functions

- def [getReactionString](#)
Write a reaction to a string with the correct units.
- def [writeTable](#)
Write an HTML formatted table to a string.
- def [displayRDMESetup](#)
Display an *RDME* initial conditions in a Jupyter Notebook using ipywidgets.

6.4.1 Function Documentation

6.4.1.1 `def pyLM.ipynInterface.displayRDMESetup (filename, continuous_update=True)`

Display an *RDME* initial conditions in a Jupyter Notebook using ipywidgets.

Parameters

<i>filename</i>	The name of a LM file encoding an <i>RDME</i> simulation
<i>continuous_update</i>	Whether to update the view continuously or only during; default: True

6.4.1.2 `def pyLM.ipynInterface.getReactionString (rct, prd, rate)`

Write a reaction to a string with the correct units.

Parameters

<i>rct</i>	Reactants tuple or string
<i>prd</i>	Products tuple or string
<i>rate</i>	The rate constant

Returns

Three strings: 1) Reaction, 2) rate constant, and 3) units of the rate constant

6.4.1.3 def pyLM.ipylInterface.writeTable (*columnNames*, *rows*)

Write an HTML formatted table to a string.

Parameters

<i>columnNames</i>	The name of the columns for the table
<i>rows</i>	A list of tuples each of the same length as <i>columnNames</i>

6.5 pyLM.LMLogger Namespace Reference

Classes

- class [NullHandler](#)

Functions

- def [setLMLoggerLevel](#)
Set the level of the logger for the application.
- def [setLMLogFile](#)
Set up file handler to print log to file.
- def [setLMLogConsole](#)
Set the logger to write to the console as the code is working.

Variables

- tuple [LMLogger](#) = logging.getLogger('LMLogger')
- tuple [nullHandlerLM](#) = logging.NullHandler()
- tuple [LMformatter](#) = logging.Formatter('%(asctime)s: %(levelname)s: %(message)s')

6.5.1 Function Documentation

6.5.1.1 def pyLM.LMLogger.setLMLogConsole (*level* = logging.DEBUG)

Set the logger to write to the console as the code is working.

Parameters

<i>level</i>	The level of information to log
--------------	---------------------------------

6.5.1.2 `def pyLM.LMLogger.setLMLogFile(filename, level = logging.DEBUG)`

Set up file handler to print log to file.

Parameters

<i>filename</i>	The name of the file to log information
<i>level</i>	The level of information to log

6.5.1.3 `def pyLM.LMLogger.setLMLoggerLevel(level)`

Set the level of the logger for the application.

Parameters

<i>level</i>	The level the logger should report (e.g. <code>logger.WARNING</code> , <code>logger.INFO</code> , etc.)
--------------	---

6.5.2 Variable Documentation

6.5.2.1 `tuple pyLM.LMLogger.LMformatter = logging.Formatter('%(asctime)s: %(levelname)s: %(message)s')`

6.5.2.2 `tuple pyLM.LMLogger.LMLogger = logging.getLogger('LMLogger')`

6.5.2.3 `tuple pyLM.LMLogger.nullHandlerLM = logging.NullHandler()`

6.6 pyLM.RDME Namespace Reference

Classes

- class [CMESimulation](#)
A *CME* simulation that contains reactions and species.
- class [RDMERegion](#)
A class that represents a type region of an *RDME* simulation.
- class [RDMESimulation](#)
A class that contains all regions, reactions, diffusions and rules for a *RDME* simulation.

Functions

- def [tqdm](#)

6.6.1 Function Documentation

6.6.1.1 `def pyLM.RDME.tqdm(x, ascii = False)`

6.7 pyLM.units Namespace Reference

Functions

- def [angstrom](#)

Length Wrapper Functions #.

- def [nm](#)
Returns a representation of a number in nanometers.
- def [micron](#)
Returns a representation of a number in micrometers.
- def [mm](#)
Returns a representation of a number in millimeter.
- def [cm](#)
Returns a representation of a number in centimeter.
- def [ns](#)

Time Wrapper Functions #.

- def [microsecond](#)
Returns a representation of a number in microsecond.
- def [ms](#)
Returns a representation of a number in millisecond.
- def [second](#)
Returns seconds in seconds.
- def [minute](#)
Returns a representation of a number in minutes.
- def [hr](#)
Returns a representation of a number in hours.
- def [day](#)
Returns a representation of a number in days.

6.7.1 Function Documentation

6.7.1.1 `def pyLM.units.angstrom (qty)`

Length Wrapper Functions #.

Returns a representation of a number in angstroms

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.2 `def pyLM.units.cm (qty)`

Returns a representation of a number in centimeter.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.3 `def pyLM.units.day (qty)`

Returns a representation of a number in days.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.4 def pyLM.units.hr (*qty*)

Returns a representation of a number in hours.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.5 def pyLM.units.micron (*qty*)

Returns a representation of a number in micrometers.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.6 def pyLM.units.microsecond (*qty*)

Returns a representation of a number in microsecond.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.7 def pyLM.units.minute (*qty*)

Returns a representation of a number in minutes.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.8 def pyLM.units.mm (*qty*)

Returns a representation of a number in millimeter.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.9 def pyLM.units.ms (*qty*)

Returns a representation of a number in millisecond.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.10 def pyLM.units.nm (*qty*)

Returns a representation of a number in nanometers.

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.11 `def pyLM.units.ns (qty)`

Time Wrapper Functions #.

Returns a representation of a number in nanosecond

Parameters

<i>qty</i>	A list or singleton of a number
------------	---------------------------------

6.7.1.12 `def pyLM.units.second (qty)`

Returns seconds in seconds.

Seems silly, but for completeness and ability to annotate the unit in code.

6.8 pySTDLM Namespace Reference

A standard library of functionality that is commonly used in biological simulations.

Namespaces

- [CellArranger](#)
- [Cells](#)
- [Distributions](#)
- [Fitting](#)
- [NetworkVisualization](#)
- [OnTheFly](#)
- [PostProcessing](#)
- [SBMLReader](#)
- [StandardCells](#)
- [StandardReactions](#)
- [StandardReactionSystems](#)

Variables

- list `__all__` = ['StandardReactions', 'StandardReactionSystems', 'StandardCells', 'Cells', 'CellArranger', 'Distributions', 'PostProcessing', 'NetworkVisualization']

6.8.1 Detailed Description

A standard library of functionality that is commonly used in biological simulations. Three sets of functionality exist including "StandardReactions" which include standard reactions such as Michealis-Menton, membrane transport, transporters, etc. "StandardReactionSystems" includes published reaction systems from a variety of cells including the MinDE system, Lac switch systems in E. coli etc. Finally, "StandardCells" includes a number of standard cell geometries.

6.8.2 Variable Documentation

6.8.2.1 list pySTDLM.__all__ = ['StandardReactions', 'StandardReactionSystems', 'StandardCells', 'Cells', 'CellArranger', 'Distributions', 'PostProcessing', 'NetworkVisualization']

6.9 pySTDLM.CellArranger Namespace Reference

Classes

- class [CellArranger](#)
A base class for laying out cells in a 3D space.
- class [RandomArranger](#)
An arranger that randomly orients a number of cells in a regin until either a count or a volume fraction target is met.
- class [RandomSphericalArranger](#)
An arranger that randomly orients a number of cells in a sphere inside a regin until either a count or a volume fraction target is met.
- class [TightGridArranger](#)
An arranger that orients cells in square grid.
- class [LooseGridArranger](#)
An arranger that orients cells in square grid with padding in between.

6.10 pySTDLM.Cells Namespace Reference

Classes

- class [CellShape](#)
Base class for a particular cell type for use in [CellArranger](#) (and later more)
- class [SphereCell](#)
A representation for a spherical cell.
- class [BoxCell](#)
A representation for a spherical cell.
- class [CapsuleCell](#)
A representation for a spherical cell.
- class [CapsuleShellCell](#)
A representation for a spherical cell.

6.11 pySTDLM.Distributions Namespace Reference

Classes

- class [ParticleGenerator](#)
A particle generator that adds particles to a particular region of a simulation.

6.12 pySTDLM.Fitting Namespace Reference

Classes

- class [Comparator](#)
- class [MomentComparator](#)

- class [TraceComparator](#)
- class [Fitter](#)
- class [SweepFitter](#)
- class [SimulatedAnnealingFitter](#)

Functions

- def [DistributionComparator](#)

Variables

- [counts](#)
- [binwidth](#)
- [measure](#)

6.12.1 Function Documentation

6.12.1.1 def pySTDLM.Fitting.DistributionComparator (*Comparator*)

6.12.2 Variable Documentation

6.12.2.1 pySTDLM.Fitting.binwidth

6.12.2.2 pySTDLM.Fitting.counts

6.12.2.3 pySTDLM.Fitting.measure

6.13 pySTDLM.NetworkVisualization Namespace Reference

Functions

- def [plotReactionNetwork](#)
Plot the reaction scheme as a network.
- def [plotCMEReactionNetwork](#)
Plot the reaction scheme as a network.
- def [plotRDMEReactionNetwork](#)
Plot the reaction scheme as a network.
- def [plotCMEDynamicReactionNetwork](#)
Plot the dynamics of the species on the network into a dynamic graph file (extension: .gexf)

Variables

- [hasGexf](#) = True

6.13.1 Function Documentation

6.13.1.1 def pySTDLM.NetworkVisualization.plotCMEDynamicReactionNetwork (*sim*, *filename*, *outfile*, *stride* = 1, *showMax* = False, *showMin* = False, *threshold* = -1, *replicate* = 1)

Plot the dynamics of the species on the network into a dynamic graph file (extension: .gexf)

Parameters

<i>sim</i>	A CMESimulation object
<i>filename</i>	An file from which to read
<i>outfile</i>	A filename for which to output without the extension (.gexf)
<i>stride</i>	Stride through the times (Default=1)
<i>showMax</i>	Show the maximum value achieved over the timecourse on the node (Default=False)
<i>showMin</i>	Show the maximum value achieved over the timecourse on the node (Default=False)
<i>threshold</i>	Only show nodes that attain this value for at least one timepoint of the simulation. Set -1 to show all nodes (Default=-1)
<i>replicate</i>	The replicate to show (Default=1)

```
6.13.1.2 def pySTDLM.NetworkVisualization.plotCMEReactionNetwork ( sim, filename, withRxnNodes = False,
collapseReversible = False )
```

Plot the reaction scheme as a network.

Parameters

<i>sim</i>	An CMESimulation object
<i>filename</i>	A file to which to output
<i>withRxnNodes</i>	Plot the graph with reactions as nodes (default: false)
<i>collapse-Reversible</i>	Collapse reversible reactions into one node (default: false)

```
6.13.1.3 def pySTDLM.NetworkVisualization.plotRDMEReactionNetwork ( sim, filename, collapseReversible = False )
```

Plot the reaction scheme as a network.

Parameters

<i>sim</i>	An RDMESimulation object
<i>filename</i>	A file to which to output
<i>collapse-Reversible</i>	Collapse reversible reactions into one node

```
6.13.1.4 def pySTDLM.NetworkVisualization.plotReactionNetwork ( sim, filename, collapseReversible = False )
```

Plot the reaction scheme as a network.

Parameters

<i>sim</i>	An RDMESimulation or CMESimulation object
<i>filename</i>	A file to which to output
<i>collapse-Reversible</i>	Collapse reversible reactions into one node

6.13.2 Variable Documentation

```
6.13.2.1 pySTDLM.NetworkVisualization.hasGexf = True
```

6.14 pySTDLM.OnTheFly Namespace Reference

6.15 pySTDLM.PostProcessing Namespace Reference

Functions

- def [openLMFile](#)
Open a Lattice Microbes File for reading.
- def [closeLMFile](#)
Close a Lattice Microbes File.
- def [showTraceFromFile](#)
Show species trace from a particular replicate.
- def [plotTraceFromFile](#)
Plot species from an output file.
- def [showTrace](#)
Show a specific species trace.
- def [plotTrace](#)
Plot a specific species trace.
- def [showAvgVarFromFile](#)
Show species from an output file.
- def [plotAvgVarFromFile](#)
Plot species from an output file.
- def [showAvgVar](#)
Show a specific species average over time and variance.
- def [plotAvgVar](#)
Plot a specific species average over time and variance.
- def [getOccupancyKymograph](#)
Compute the specie density(occupancy) among a slice of the simulation domain as a function over time for the given direction.
- def [plotOccupancyKymograph](#)
Plot a specific specie density(occupancy) among a slice of the simulation domain as a function of a direction over time.
- def [showOccupancyKymograph](#)
Show a specific specie density(occupancy) among a slice of the simulation domain as a function of a direction over time.
- def [plotPhaseSpace](#)
Plot the 2D or 3D phase space associated with the given species over the replicates indicated.
- def [getTimeSteps](#)
Extract the timestep times.
- def [getSpecieTrace](#)
Extract data for a particular specie for the specified replicate.
- def [getAvgVarTrace](#)
Get the average and variance of the specie trace over time.
- def [getHistogram](#)
Get the histogram for a specie.
- def [getPhaseSpace](#)
Get the nD phase space associated with the traces of species.

6.15.1 Function Documentation

6.15.1.1 def pySTDLM.PostProcessing.closeLMFile (f)

Close a Lattice Microbes File.

Parameters

<i>f</i>	A previously opened lattice microbes file
----------	---

6.15.1.2 `def pySTDLM.PostProcessing.getAvgVarTrace (f, specie, doublingTime = None)`

Get the average and variance of the specie trace over time.

Parameters

<i>f</i>	The HDF5 file handle to extract from or the name of a file to open
<i>specie</i>	The specie to extract
<i>doublingTime</i>	An optional doubling time parameter that will normalize average each trace by time in the cell cycle assuming exponentially growing cell ($2\log_2 * x/2^{(t/DT)}$), effectively normalizing against cell size. Default: no averaging is performed

Returns

avg, var, time

6.15.1.3 `def pySTDLM.PostProcessing.getHistogram (f, species)`

Get the histogram for a specie.

Parameters

<i>f</i>	The HDF5 file handle to extract from or the name of a file to open
<i>specie</i>	An array of specie/s to extract. returns a nD histogram of the data.

Returns

bins, edges (len(bins)+1)

6.15.1.4 `def pySTDLM.PostProcessing.getOccupancyKymograph (f, species = None, replicate = 1)`

Compute the specie density(occupancy) among a slice of the simulation domain as a function over time for the given direction.

Parameters

<i>filename</i>	Name of file to extract data from
<i>specie</i>	A particular specie to plot density for
<i>replicate</i>	The replicate to show trace for

6.15.1.5 `def pySTDLM.PostProcessing.getPhaseSpace (f, species, replicate)`

Get the nD phase space associated with the traces of species.

If a single replicate is specified, a single trace will be returned, otherwise a 2/3D density matrix will be returned.

Parameters

<i>f</i>	The HDF5 file handle to extract from or the name of a file to open
<i>specie</i>	An iterable of 2 more specie names
<i>replicate</i>	The replicate for which to extract the phase space

Returns

A numpy array of a trace from a single replicate (specie1, specie2, ...)

6.15.1.6 `def pySTDLM.PostProcessing.getSpecieTrace (f, specie, replicate = 1, doublingTime = None)`

Extract data for a particular specie for the specified replicate.

Parameters

<i>f</i>	The HDF5 file handle to extract from or the name of a file to open
<i>specie</i>	The specie to extract data
<i>replicate</i>	The number of the replicate to extract from (default: 1)
<i>doublingTime</i>	An optional doubling time parameter that will normalize average each trace by time in the cell cycle assuming exponentially growing cell ($2\log_2 * x/2^{(t/DT)}$), effectively normalizing against cell size. Default: no averaging is performed

Returns

The species time trace in a numpy array

6.15.1.7 `def pySTDLM.PostProcessing.getTimesteps (f)`

Extract the timestep times.

Parameters

<i>f</i>	The HDF5 file handle to extract from or the name of a file to open
----------	--

Returns

The timestep times in a numpy array

6.15.1.8 `def pySTDLM.PostProcessing.openLMFile (filename)`

Open a Lattice Microbes File for reading.

Parameters

<i>filename</i>	Name of the file
-----------------	------------------

Returns

a handle to the file

6.15.1.9 `def pySTDLM.PostProcessing.plotAvgVar (f, species = None, filename = None, kwargs)`

Plot a specific species average over time and variance.

Parameters

<i>f</i>	An h5py object handle
<i>species</i>	A list of species to show
<i>filename</i>	A filename to print to (default None, gives the same behavior as showAvgVar(...))
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot

Returns

A handle to the figure object created which allows customization of plot attributes

6.15.1.10 `def pySTDLM.PostProcessing.plotAvgVarFromFile (filename, species, outfile, kwargs)`

Plot species from an output file.

Parameters

<i>filename</i>	The name of an HDF5 output file generated by LatticeMicrobes
<i>species</i>	A list of species to plot
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot
<i>outfile</i>	A filename to plot to

6.15.1.11 `def pySTDLM.PostProcessing.plotOccupancyKymograph (f, species=None, replicate=1, filename=None)`

Plot a specific specie density(occupancy) among a slice of the simulation domain as a function of a direction over time.

Parameters

<i>f</i>	The name of the LM file
<i>specie</i>	A particular specie to plot density for
<i>filename</i>	A filename to print to (default None, gives the same behavior as showAvgVar(...))
<i>replicate</i>	The replicate to show trace for

6.15.1.12 `def pySTDLM.PostProcessing.plotPhaseSpace (f, species=None, replicate=1, withHistogram=False)`

Plot the 2D or 3D phase space associated with the given species over the replicates indicated.

Parameters

<i>f</i>	An h5py object handle
<i>specie</i>	An iterable of 2 or 3 specie names indicating whether to plot in 2D or 3D space
<i>replicate</i>	The replicate to show trace for
<i>withHistogram</i>	If set to true, a heatmap of the phase space over all replicates will be plotted in the background. (NOTE: this only works in 2D.)

6.15.1.13 `def pySTDLM.PostProcessing.plotTrace (f, species=None, replicate=1, filename=None, kwargs)`

Plot a specific species trace.

Parameters

<i>f</i>	An h5py object handle
<i>species</i>	A specie name or a list of species to show; can be a single string or a iterable list of species
<i>filename</i>	A filename to print to (default None, gives the same behavior as showAvgVar(...))
<i>replicate</i>	The replicate to show trace for. Can be an integer or an iterable list of replicte numbers
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot

Returns

A handle to the figure object created which allows customization of plot attributes

6.15.1.14 `def pySTDLM.PostProcessing.plotTraceFromFile (filename, species, replicate, outfile, kwargs)`

Plot species from an output file.

Parameters

<i>filename</i>	The patch to an HDF5 output file generated by LatticeMicrobes
<i>species</i>	A list of species to plot
<i>replicate</i>	The replicate to show trace for
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot
<i>outfile</i>	A filename to plot to

6.15.1.15 `def pySTDLM.PostProcessing.showAvgVar (f, species, kwargs)`

Show a specific species average over time and variance.

Parameters

<i>f</i>	An h5py object handle
<i>species</i>	A list of species to show
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot

6.15.1.16 `def pySTDLM.PostProcessing.showAvgVarFromFile (filename, species, kwargs)`

Show species from an output file.

Parameters

<i>filename</i>	The name of an HDF5 output file generated by LatticeMicrobes
<i>species</i>	A list of species to plot
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot

6.15.1.17 `def pySTDLM.PostProcessing.showOccupancyKymograph (f, species = None, replicate = 1)`

Show a specific specie density(occupancy) among a slice of the simulation domain as a function of a direction over time.

Parameters

<i>f</i>	An h5py object handle
<i>specie</i>	A particular specie to plot density for
<i>replicate</i>	The replicate to show trace for

6.15.1.18 `def pySTDLM.PostProcessing.showTrace (f, species, replicate, kwargs)`

Show a specific species trace.

Parameters

<i>f</i>	An h5py object handle
<i>species</i>	A list of species to show
<i>replicate</i>	The replicate to show trace for
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot

6.15.1.19 `def pySTDLM.PostProcessing.showTraceFromFile (filename, species, replicate, kwargs)`

Show species trace from a particular replicate.

Parameters

<i>filename</i>	The patch to an HDF5 output file generated by LatticeMicrobes
<i>species</i>	A list of species to plot
<i>replicate</i>	The replicate to show trace for
<i>kwargs</i>	Additional arguments to be passed on to matplotlib.plot. These are any arguments that are valid with matplotlib.plot

6.16 pySTDLM.SBMLReader Namespace Reference

Functions

- `def importSBMLModel`
Helper Functions ##.
- `def importSBMLModelL3V1`
- `def getRate`
- `def readSBMLtoCME`
Read an SBML file for the reaction model for a CME simulation.
- `def readSBMLtoRDME`
Read an SBML file for the reaction model for a RDME simulation.
- `def readSBMLtoRegion`
Read an SBML file for the reaction model into a specific region of the RDME simulation.

6.16.1 Function Documentation

6.16.1.1 `def pySTDLM.SBMLReader.getRate (rxn, rcts, prds, sim, globalP, globalPV)`

6.16.1.2 `def pySTDLM.SBMLReader.importSBMLModel (filename)`

Helper Functions ##.

6.16.1.3 `def pySTDLM.SBMLReader.importSBMLModelL3V1 (model, sim, region = None)`

6.16.1.4 `def pySTDLM.SBMLReader.readSBMLtoCME (sim, filename)`

Read an SBML file for the reaction model for a CME simulation.

Parameters

<i>sim</i>	A CMESimulation object
<i>filename</i>	A SBML filename/filepath

Returns

The simulation object

6.16.1.5 `def pySTDLM.SBMLReader.readSBMLtoRDME (sim, filename)`

Read an SBML file for the reaction model for a RDME simulation.

Parameters

<i>sim</i>	A RDMESimulation object
<i>filename</i>	A SBML filename/filepath

Returns

The simulation object

6.16.1.6 `def pySTDLM.SBMLReader.readSBMLtoRegion (sim, region, filename)`

Read an SBML file for the reaction model into a specific region of the RDME simulation.

Parameters

<i>sim</i>	A RDMESimulation object
<i>region</i>	A region name that already exists in the RDMESimulation
<i>filename</i>	A SBML filename/filepath

Returns

The simulation object

6.17 pySTDLM.StandardCells Namespace Reference

Functions

- `def buildEColiCell`
Build a standard E coli cell 0.5 microns across and 2 microns long.
- `def buildDividingEColiCell`
Build a standard E coli cell 0.5 microns across and 4 microns long.
- `def buildSphericalEColiCell`
Build a Spherical E coli cell, one that lacks the gene for elongation.
- `def buildFilamentousEColiCell`
Build a long filamentous E coli cell where the user can specify the length.
- `def packFastGrowingEcoli`

Pack a cell with the protein distribution for fast growing E coli.

- def `packSlowGrowingEcoli`

Pack a cell with the protein distribution for fast growing E coli.

6.17.1 Function Documentation

6.17.1.1 def `pySTDLM.StandardCells.buildDividingEColiCell (sim, crowded = False, crowdedMembrane = False)`

Build a standard E coli cell 0.5 microns across and 4 microns long.

Parameters

<i>sim</i>	An RDMESimulation object
<i>crowded</i>	Should the cytosol be crowded (default: false)
<i>crowded-Membrane</i>	Should the membrane be crowded (default: false)

6.17.1.2 def `pySTDLM.StandardCells.buildEColiCell (sim, crowded = False, crowdedMembrane = False)`

Build a standard E coli cell 0.5 microns across and 2 microns long.

Parameters

<i>sim</i>	An RDMESimulation object
<i>crowded</i>	Should the cytosol be crowded (default: false)
<i>crowded-Membrane</i>	Should the membrane be crowded (default: false)

6.17.1.3 def `pySTDLM.StandardCells.buildFilamentousEColiCell (sim, length = 8.0, crowded = False, crowdedMembrane = False)`

Build a long filamentous E coli cell where the user can specify the length.

Parameters

<i>sim</i>	An RDMESimulation object
<i>length</i>	The length of the cell in microns (default: 8)
<i>crowded</i>	Should the cytosol be crowded (default: false)
<i>crowded-Membrane</i>	Should the membrane be crowded (default: false)

6.17.1.4 def `pySTDLM.StandardCells.buildSphericalEColiCell (crowded = False, latticeSpacing = 16, crowdedMembrane = False)`

Build a Spherical E coli cell, one that lacks the gene for elongation.

Parameters

<i>sim</i>	An RDMESimulation object
<i>crowded</i>	Should the cytosol be crowded (default: false)
<i>crowded-Membrane</i>	Should the membrane be crowded (default: false)

6.17.1.5 `def pySTDLM.StandardCells.packFastGrowingEcoli (sim)`

Pack a cell with the protein distribution for fast growing E coli.

Parameters

<i>sim</i>	A RDMESimulation object
------------	-------------------------

6.17.1.6 def pySTDLM.StandardCells.packSlowGrowingEcoli (*sim*)

Pack a cell with the protein distribution for fast growing E coli.

Parameters

<i>sim</i>	A RDMESimulation object
------------	-------------------------

6.18 pySTDLM.StandardReactions Namespace Reference

Functions

- def [addMembraneTransporter](#)
Adds a membrane transport system to an RDME system.
- def [addPassiveTransport](#)
Adds a passive transport system to an RDME system.
- def [addMichaelisMenten](#)
Adds a Michaelis Menten Reaction.
- def [addReversibleMichaelisMenten](#)
Add a Reversible Michaelis Menten Reaction.
- def [createExpressionModel](#)
Create a set of gene/mRNA/protein reactions based on a genebank file and a set of rates assuming constitutive expression.

Variables

- [hasBio](#) = True

6.18.1 Function Documentation

6.18.1.1 def pySTDLM.StandardReactions.addMembraneTransporter (*sim*, *transporter*, *number*, *name1*, *name2*, *dRate*, *kf*, *kr*, *region1* = 'cytoplasm', *region2* = 'default', *intoRegion* = 'membrane')

Adds a membrane transport system to an RDME system.

Parameters

<i>sim</i>	An RDMESimulation to add the transporter system to
<i>transporter</i>	Name of the transporter particle
<i>number</i>	The number of transporter molecules
<i>name1</i>	Name of the transported particle in region1
<i>name2</i>	Name of the transported particle in region2
<i>dRate</i>	The rate of diffusion from region1 <->intoRegion and region2<->intoRegion
<i>kf</i>	The forward reaction rate

<i>kr</i>	The reverse reaction rate
<i>region1</i>	The region on the first side of the membrane
<i>region2</i>	The region on the second side of the membrane
<i>intoRegion</i>	The region representing the membrane

Returns

The simulation object so this can be a chained call

```
6.18.1.2 def pySTDLM.StandardReactions.addMichaelisMenten ( sim, reactant, enzyme, product, k1f, k1b, k2, region =
'cytoplasm' )
```

Adds a Michaelis Menten Reaction.

Parameters

<i>sim</i>	The RDME or CME reaction
<i>reactant</i>	The reactant that reacts with the enzyme
<i>enzyme</i>	The enzyme catalyzing the reaction
<i>product</i>	The product of the reaction
<i>k1f</i>	The forward reaction rate
<i>k1b</i>	The backward reaction rate
<i>k2</i>	The second forward rate
<i>region</i>	The region in which the reaction should occur (RDME only)

Returns

The simulation object so this can be a chained call

```
6.18.1.3 def pySTDLM.StandardReactions.addPassiveTransport ( sim, specie, dRate, region1 = 'cytoplasm', region2
= 'default', region3 = 'membrane' )
```

Adds a passive transport system to an RDME system.

Parameters

<i>sim</i>	An RDMESimulation to add the passive transport to
<i>specie</i>	The species that can diffuse across the membrane
<i>dRate</i>	The rate of diffusion across the membrane
<i>region1</i>	The region on the first side of the membrane
<i>region2</i>	The region on the second side of the membrane
<i>region3</i>	The region representing the membrane

Returns

The simulation object so this can be a chained call

```
6.18.1.4 def pySTDLM.StandardReactions.addReversibleMichaelisMenten ( sim, reactant, enzyme, product, k1f, k1b, k2f,
k2b, k3f, k3b, region = 'cytoplasm' )
```

Add a Reversible Michaelis Menten Reaction.

Parameters

<i>sim</i>	The RDME or CME reaction
<i>reactant</i>	The reactant that reacts with the enzyme
<i>enzyme</i>	The enzyme catalyzing the reaction
<i>product</i>	The product of the reaction
<i>k1f</i>	The forward reaction rate
<i>k1b</i>	The backward reaction rate
<i>k2f</i>	The second forward rate
<i>k2b</i>	the second backward rate
<i>k3f</i>	The third forward rate
<i>k3b</i>	the third backward rate
<i>region</i>	the region in which the reaction should occur (RDME only)

Returns

The simulation object so this can be a chained call

6.18.1.5 `def pySTDLM.StandardReactions.createExpressionModel (sim, gb, kt, kd, kr, kdil=None, regions=None)`

Create a set of gene/mRNA/protein reactions based on a genbank file and a set of rates assuming constitutive expression.

Parameters

<i>sim</i>	The RDME or CME simulation
<i>gb</i>	The genbank filename. File should be readable by BioPython.
<i>kt</i>	mRNA transcription rate dictionary { locusTag -> rate }
<i>kd</i>	mRNA degradation rate dictionary { locusTag -> rate }
<i>kr</i>	Protein transcription rate dictionary {locusTag -> rate }
<i>kdil</i>	Protein dillution/degradation rate dictionary {locusTag -> rate} (Optional; Default "None", meaning no dillution reaction will be specified)
<i>regions</i>	Regions for the reactions to occur. Degradation is allowed in both regions. {locusTag -> (transcriptionRegion, translationRegion)} (required for RDMEsimulations)

Returns

SeqIO representation of the Genbank file with qualifiers added: `qualifiers["dna_id" -> (str,int), "rna_id" -> (str,int), "protein_id" -> (str,int)]`

6.18.2 Variable Documentation

6.18.2.1 `pySTDLM.StandardReactions.hasBio = True`

6.19 `pySTDLM.StandardReactionSystems` Namespace Reference

Functions

- def [addMinDESystem](#)
Adds the MinDE system in Ecoli as a standard reaction network.
- def [addLacTwoStateSystem](#)
Adds the Lac switch reaction with two states of the DNA.
- def [addPTSPathway](#)
Adds the Phosphoenolpyruvate-depenedent phosphotransferase reaction system.

6.19.1 Function Documentation

6.19.1.1 `def pySTDLM.StandardReactionSystems.addLacTwoStateSystem (sim, inducerType = ' TMG')`

Adds the Lac switch reaction with two states of the DNA.

Reference: E. Roberts, A. Magis, J.O. Ortiz, W. Baumeister, Z. Luthey-Schulten. Noise Contributions in an Inducible Genetic Switch: A Whole-Cell Simulation Study. PLoS Comput. Bio. 7(3): 2011, e1002010.

Parameters

<i>sim</i>	A CMESimulation or RDMESimulation object with the "cytoplasm" region defined
<i>sim</i>	inducerType The type of inducer used for the switch, either TMG or IPTG

Returns

A pointer to the simulation object that was passed in

6.19.1.2 `def pySTDLM.StandardReactionSystems.addMinDESystem (sim)`

Adds the MinDE system in Ecoli as a standard reaction network.

Reference:

Parameters

<i>sim</i>	A RDMESimulation object with the "cytoplasm" and "membrane" regions defined
------------	---

Returns

A pointer to the simulation object that was passed in

6.19.1.3 `def pySTDLM.StandardReactionSystems.addPTSPathway (sim)`

Adds the Phosphoenolpyruvate-depenedent phosphotransferase reaction system.

Reference: J.V. Rodriguez, J.A. Kaandorp, M. Dobrzynski, J.G. Blom Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in Escherichia coli. Bioinform. 22:15 (2006), pp. 1895-1901.

Parameters

<i>sim</i>	A RDMESimulation object with the "cytoplasm", "default" and "membrane" regions defined
------------	--

Returns

A pointer to the simulation object that was passed in

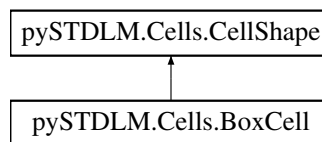
Chapter 7

Class Documentation

7.1 pySTDLM.Cells.BoxCell Class Reference

A representation for a spherical cell.

Inheritance diagram for pySTDLM.Cells.BoxCell:



Public Member Functions

- def [__init__](#)
- def [internalModelCell](#)
- def [computeVolume](#)
- def [internalAddToSim](#)
- def [internalTranslateCell](#)

Public Attributes

- [name](#)
- [p1](#)
- [p2](#)
- [mins](#)
- [maxs](#)
- [width](#)
- [height](#)
- [depth](#)
- [membraneThickness](#)

7.1.1 Detailed Description

A representation for a spherical cell.

This cell requires an attribute dictionary that includes "width", "height", "depth" and "membraneThickness"

7.1.2 Constructor & Destructor Documentation

7.1.2.1 `def pySTDLM.Cells.BoxCell.__init__(self, p1 = 3*[0.], p2 = 3*[0.])`

7.1.3 Member Function Documentation

7.1.3.1 `def pySTDLM.Cells.BoxCell.computeVolume(self)`

7.1.3.2 `def pySTDLM.Cells.BoxCell.internalAddToSim(self, sim)`

7.1.3.3 `def pySTDLM.Cells.BoxCell.internalModelCell(self, attr)`

7.1.3.4 `def pySTDLM.Cells.BoxCell.internalTranslateCell(self, point)`

7.1.4 Member Data Documentation

7.1.4.1 `pySTDLM.Cells.BoxCell.depth`

7.1.4.2 `pySTDLM.Cells.BoxCell.height`

7.1.4.3 `pySTDLM.Cells.BoxCell.maxs`

7.1.4.4 `pySTDLM.Cells.BoxCell.membraneThickness`

7.1.4.5 `pySTDLM.Cells.BoxCell.mins`

7.1.4.6 `pySTDLM.Cells.BoxCell.name`

7.1.4.7 `pySTDLM.Cells.BoxCell.p1`

7.1.4.8 `pySTDLM.Cells.BoxCell.p2`

7.1.4.9 `pySTDLM.Cells.BoxCell.width`

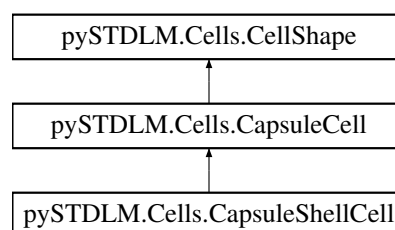
The documentation for this class was generated from the following file:

- [Cells.py](#)

7.2 pySTDLM.Cells.CapsuleCell Class Reference

A representation for a spherical cell.

Inheritance diagram for `pySTDLM.Cells.CapsuleCell`:



Public Member Functions

- def `__init__`
- def `internalModelCell`
- def `computeVolume`
- def `internalAddToSim`
- def `internalTranslateCell`

Public Attributes

- `name`
- `p1`
- `p2`
- `radius`
- `length`
- `mins`
- `maxs`
- `membraneThickness`

7.2.1 Detailed Description

A representation for a spherical cell.

This cell requires an attribute dictionary that includes "radius", "length", and "membraneThickness"

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `def pySTDLM.Cells.CapsuleCell.__init__(self, p1 = 3*[0.], p2 = 3*[0.], radius = 0., length = 0.)`

7.2.3 Member Function Documentation

7.2.3.1 `def pySTDLM.Cells.CapsuleCell.computeVolume (self)`

7.2.3.2 `def pySTDLM.Cells.CapsuleCell.internalAddToSim (self, sim)`

7.2.3.3 `def pySTDLM.Cells.CapsuleCell.internalModelCell (self, attr)`

7.2.3.4 `def pySTDLM.Cells.CapsuleCell.internalTranslateCell (self, point)`

7.2.4 Member Data Documentation

7.2.4.1 `pySTDLM.Cells.CapsuleCell.length`

7.2.4.2 `pySTDLM.Cells.CapsuleCell.maxs`

7.2.4.3 `pySTDLM.Cells.CapsuleCell.membraneThickness`

7.2.4.4 `pySTDLM.Cells.CapsuleCell.mins`

7.2.4.5 `pySTDLM.Cells.CapsuleCell.name`

7.2.4.6 `pySTDLM.Cells.CapsuleCell.p1`

7.2.4.7 `pySTDLM.Cells.CapsuleCell.p2`

7.2.4.8 pySTDLM.Cells.CapsuleCell.radius

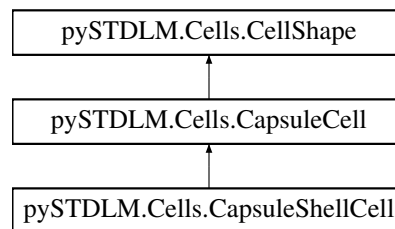
The documentation for this class was generated from the following file:

- [Cells.py](#)

7.3 pySTDLM.Cells.CapsuleShellCell Class Reference

A representation for a spherical cell.

Inheritance diagram for pySTDLM.Cells.CapsuleShellCell:



Public Member Functions

- def `__init__`
- def `internalModelCell`
- def `computeVolume`
- def `internalAddToSim`

Public Attributes

- `name`
- `radius`
- `length`
- `p1`
- `p2`
- `membraneThickness`
- `mins`
- `maxs`

7.3.1 Detailed Description

A representation for a spherical cell.

This cell requires an attribute dictionary that includes "radius", "length" and "membraneThickness"

7.3.2 Constructor & Destructor Documentation

7.3.2.1 `def pySTDLM.Cells.CapsuleShellCell.__init__(self, p1 = 3*[0.], p2 = 3*[0.], radius = 0., length = 0.)`

7.3.3 Member Function Documentation

7.3.3.1 `def pySTDLM.Cells.CapsuleShellCell.computeVolume (self)`

7.3.3.2 `def pySTDLM.Cells.CapsuleShellCell.internalAddToSim (self, sim)`

7.3.3.3 `def pySTDLM.Cells.CapsuleShellCell.internalModelCell (self, attr)`

7.3.4 Member Data Documentation

7.3.4.1 `pySTDLM.Cells.CapsuleShellCell.length`

7.3.4.2 `pySTDLM.Cells.CapsuleShellCell.maxs`

7.3.4.3 `pySTDLM.Cells.CapsuleShellCell.membraneThickness`

7.3.4.4 `pySTDLM.Cells.CapsuleShellCell.mins`

7.3.4.5 `pySTDLM.Cells.CapsuleShellCell.name`

7.3.4.6 `pySTDLM.Cells.CapsuleShellCell.p1`

7.3.4.7 `pySTDLM.Cells.CapsuleShellCell.p2`

7.3.4.8 `pySTDLM.Cells.CapsuleShellCell.radius`

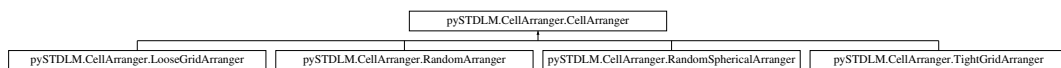
The documentation for this class was generated from the following file:

- [Cells.py](#)

7.4 pySTDLM.CellArranger.CellArranger Class Reference

A base class for laying out cells in a 3D space.

Inheritance diagram for `pySTDLM.CellArranger.CellArranger`:



Public Member Functions

- `def __init__`
Create a *CellArranger*.
- `def cellFactory`
- `def packVolume`
Pack a specified volume with cells.
- `def addToSimulation`
Add packed volume to RDME simulation.

Public Attributes

- `allowedCellTypes`
- `allowedForms`
- `cellType`
- `cellAttributes`
- `cells`

- [packerName](#)
- [arrangerAttributes](#)

7.4.1 Detailed Description

A base class for laying out cells in a 3D space.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 `def pySTDLM.CellArranger.CellArranger.__init__(self, cellType=None, cellAttributes=None, arrangerAttributes=None, name='Unknown')`

Create a [CellArranger](#).

Parameters

<i>self</i>	
<i>cellType</i>	The type of cell this packer represents, can be one of: ["Sphere", "Capsule", "Box", "Capsule-Shell"]
<i>cellAttributes</i>	A dictionary with the required attributes specified for the cell and the form
<i>arranger-Attributes</i>	A dictionary with the required/optional requirements for the individual arranger

7.4.3 Member Function Documentation

7.4.3.1 `def pySTDLM.CellArranger.CellArranger.addToSimulation(self, sim)`

Add packed volume to RDME simulation.

Parameters

<i>self</i>	
<i>sim</i>	An RDMESimulation object

7.4.3.2 `def pySTDLM.CellArranger.CellArranger.cellFactory(self)`

7.4.3.3 `def pySTDLM.CellArranger.CellArranger.packVolume(self, sim, volume=None)`

Pack a specified volume with cells.

Parameters

<i>self</i>	
<i>volume</i>	A volume specified as [[xmin,ymin,zmin],[xmax,ymax,zmax]] (TODO: Add angles)
<i>form</i>	The form of the packing, which can one of: ["Random", "RandomSpherical", "TightGrid", "LooseGrid", "SkewedGrid"]

Returns

A tuple of the form [volumePacked, volumePercent, numberCells]

7.4.4 Member Data Documentation

7.4.4.1 `pySTDLM.CellArranger.CellArranger.allowedCellTypes`

7.4.4.2 [pySTDLM.CellArranger.CellArranger.allowedForms](#)

7.4.4.3 [pySTDLM.CellArranger.CellArranger.arrangerAttributes](#)

7.4.4.4 [pySTDLM.CellArranger.CellArranger.cellAttributes](#)

7.4.4.5 [pySTDLM.CellArranger.CellArranger.cells](#)

7.4.4.6 [pySTDLM.CellArranger.CellArranger.cellType](#)

7.4.4.7 [pySTDLM.CellArranger.CellArranger.packerName](#)

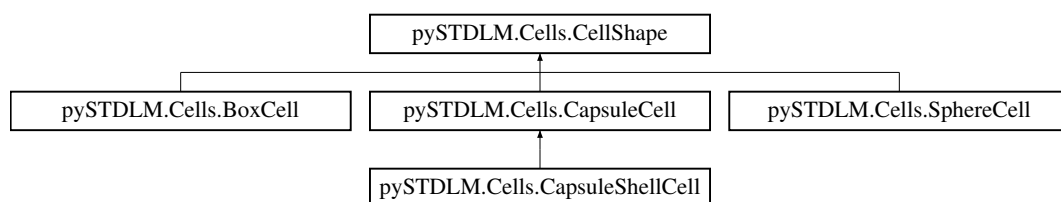
The documentation for this class was generated from the following file:

- [CellArranger.py](#)

7.5 pySTDLM.Cells.CellShape Class Reference

Base class for a particular cell type for use in [CellArranger](#) (and later more)

Inheritance diagram for `pySTDLM.Cells.CellShape`:



Public Member Functions

- def [__init__](#)
- def [getVolume](#)
Volume occupied by cell NOTE: When overriding this class, you must specify a "computeVolume()" function of no arguments.
- def [boundingBox](#)
Return a bounding box for the cell TODO: Make it return an orientation as well.
- def [addToSimulation](#)
Add the cell to the simulation.
- def [translateCell](#)
Shift a cell in space by the specified amount.
- def [createModelCell](#)
Create model cell.
- def [setRegions](#)
Set regions.

Public Attributes

- [name](#)
- [volume](#)
- [mins](#)
- [maxs](#)

- [membraneThickness](#)
- [memtype](#)
- [cyttype](#)

7.5.1 Detailed Description

Base class for a particular cell type for use in [CellArranger](#) (and later more)

7.5.2 Constructor & Destructor Documentation

7.5.2.1 `def pySTDLM.Cells.CellShape.__init__(self)`

7.5.3 Member Function Documentation

7.5.3.1 `def pySTDLM.Cells.CellShape.addToSimulation(self, sim)`

Add the cell to the simulation.

Parameters

<i>self</i>	
<i>sim</i>	An RDMESimulation object

7.5.3.2 `def pySTDLM.Cells.CellShape.boundingBox(self)`

Return a bounding box for the cell TODO: Make it return an orientation as well.

7.5.3.3 `def pySTDLM.Cells.CellShape.createModelCell(self, attr)`

Create model cell.

Parameters

<i>self</i>	
<i>attrs</i>	A dictionary of attributes

7.5.3.4 `def pySTDLM.Cells.CellShape.getVolume(self)`

Volume occupied by cell NOTE: When overriding this class, you must specify a "computeVolume()" function of no arguments.

Returns

volume In intrinsic units

7.5.3.5 `def pySTDLM.Cells.CellShape.setRegions(self, membrane, cytoplasm)`

Set regions.

Parameters

<i>self</i>	
<i>membrane</i>	The name of the region in which the membrane should be considered
<i>cytoplasm</i>	The name of the region in which the cytoplasm should be considered

7.5.3.6 def pySTDLM.Cells.CellShape.translateCell (*self*, *point*)

Shift a cell in space by the specified amount.

Parameters

<i>self</i>	
<i>point</i>	[x,y,z] translation in space

7.5.4 Member Data Documentation

7.5.4.1 pySTDLM.Cells.CellShape.cyttype

7.5.4.2 pySTDLM.Cells.CellShape.maxs

7.5.4.3 pySTDLM.Cells.CellShape.membraneThickness

7.5.4.4 pySTDLM.Cells.CellShape.memtype

7.5.4.5 pySTDLM.Cells.CellShape.mins

7.5.4.6 pySTDLM.Cells.CellShape.name

7.5.4.7 pySTDLM.Cells.CellShape.volume

The documentation for this class was generated from the following file:

- [Cells.py](#)

7.6 pyLM.CME.CMESimulation Class Reference

A [CME](#) simulation that contains reactions and species.

Public Member Functions

- def [__init__](#)
A constructor for the [CMESimulation](#).
- def [defineSpecies](#)
Define a specie/s of particles that exist in the simulation.
- def [addParticles](#)
Add a specified number of particles of the specified type to the specified region.
- def [addConcentration](#)
Add a concentration of particles of the specified type to the simulation.
- def [addReaction](#)
Adds a 0th, 1st or 2nd order reaction.
- def [buildReactionModel](#)

- *Return the Lattice Microbes ReactionModel object for fine-tuning.*
- def [setRandomSeed](#)
- def [setWriteInterval](#)
 - *Set the simulation state write-to-disk interval.*
- def [setSimulationTime](#)
 - *Set the total simulation time.*
- def [setTimestep](#)
- def [save](#)
 - *Create an HDF5 version of the simulation amenable for later running or stand-alone running.*
- def [run](#)
 - *Run the simulation using the specified solver the specified amount of time.*
- def [runMPI](#)
 - *Run the simulation using a call to mpirun with the given options.*
- def [runSolver](#)

Public Attributes

- [particleMap](#)
- [species_id](#)
- [initial_counts](#)
- [reactions](#)
- [parameters](#)
- [volume](#)
- [name](#)
- [replicates](#)
- [filename](#)

7.6.1 Detailed Description

A [CME](#) simulation that contains reactions and species.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 `def pyLM.CME.CMESimulation.__init__(self, volume = None, name = "unnamed")`

A constructor for the [CMESimulation](#).

Parameters

<i>volume</i>	The reaction vessel volume (in Litres). Specifying 'None' signifies that the user has already accounted for volume in rate constants.
<i>name</i>	The name of the CME simulation; Default: "unnamed"

Returns

self

7.6.3 Member Function Documentation

7.6.3.1 `def pyLM.CME.CMESimulation.addConcentration(self, species = 'unknown', conc = 0.0)`

Add a concentration of particles of the specified type to the simulation.

Parameters

<i>self</i>	
<i>species</i>	The name of the specie to add
<i>concentration</i>	The concentration of the species (Molar). Particle count is rounded to nearest integer.

Exceptions

<i>An</i>	error if the volume is not specified
-----------	--------------------------------------

7.6.3.2 `def pyLM.CME.CMESimulation.addParticles (self, species = 'unknown', count = 1)`

Add a specified number of particles of the specified type to the specified region.

Parameters

<i>self</i>	
<i>species</i>	The name of the specie to add particles to
<i>count</i>	The number of that particle to start with (default 1)

7.6.3.3 `def pyLM.CME.CMESimulation.addReaction (self, reactant, product, rate)`

Adds a 0th, 1st or 2nd order reaction.

Parameters

<i>self</i>	
<i>reactant</i>	The list or reactants
<i>product</i>	The list of products
<i>rate</i>	The rate of reaction

7.6.3.4 `def pyLM.CME.CMESimulation.buildReactionModel (self)`

Return the Lattice Microbes ReactionModel object for fine-tuning.

Parameters

<i>self</i>	
-------------	--

Returns

The reaction model for the simulation

7.6.3.5 `def pyLM.CME.CMESimulation.defineSpecies (self, species)`

Define a specie/s of particles that exist in the simulation.

Parameters

<i>self</i>	
<i>species</i>	A list of species to add to the simulation

7.6.3.6 `def pyLM.CME.CMESimulation.run (self, filename, method, replicates = 1, seed = None)`

Run the simulation using the specified solver the specified amount of time.

Parameters

<i>self</i>	
<i>filename</i>	The HDF file to write to
<i>method</i>	The class name for the solver to use (e.g., <code>lm::cme::GillespieDSolver</code>)
<i>replicates</i>	The number of replicates to serially run
<i>seed</i>	A seed for the random number generator to use when running the simulation; None indicates default

```
7.6.3.7 def pyLM.CME.CMESimulation.runMPI ( self, filename, method, replicates = 1, driver = "mpirun", ppe = 1,
seed = None )
```

Run the simulation using a call to mpirun with the given options.

Parameters

<i>self</i>	
<i>filename</i>	The HDF file to write to
<i>method</i>	The class name for the solver to use (e.g., <code>lm::cme::GillespieDSolver</code>)
<i>replicates</i>	The number of replicates to serially run
<i>driver</i>	The program to execute the parallel run, e.g. "mpirun", "aprun", "ibrun", etc.
<i>ppe</i>	The number of processing elements to use
<i>seed</i>	A seed for the random number generator to use when running the simulation; None indicates default

```
7.6.3.8 def pyLM.CME.CMESimulation.runSolver ( self, filename, solver, replicates = 1 )
```

```
7.6.3.9 def pyLM.CME.CMESimulation.save ( self, filename )
```

Create an HDF5 version of the simulation amenable for later running or stand-alone running.

Parameters

<i>self</i>	
<i>filename</i>	The filename to save the simulation setup in

```
7.6.3.10 def pyLM.CME.CMESimulation.setRandomSeed ( self, seed )
```

```
7.6.3.11 def pyLM.CME.CMESimulation.setSimulationTime ( self, time )
```

Set the total simulation time.

Parameters

<i>self</i>	
<i>time</i>	Time length of the simulation

```
7.6.3.12 def pyLM.CME.CMESimulation.setTimeStep ( self, time )
```

```
7.6.3.13 def pyLM.CME.CMESimulation.setWriteInterval ( self, time )
```

Set the simulation state write-to-disk interval.

Parameters

<i>self</i>	
<i>time</i>	Time length between writes

7.6.4 Member Data Documentation

7.6.4.1 [pyLM.CME.CMESimulation.filename](#)7.6.4.2 [pyLM.CME.CMESimulation.initial_counts](#)7.6.4.3 [pyLM.CME.CMESimulation.name](#)7.6.4.4 [pyLM.CME.CMESimulation.parameters](#)7.6.4.5 [pyLM.CME.CMESimulation.particleMap](#)7.6.4.6 [pyLM.CME.CMESimulation.reactions](#)7.6.4.7 [pyLM.CME.CMESimulation.replicates](#)7.6.4.8 [pyLM.CME.CMESimulation.species_id](#)7.6.4.9 [pyLM.CME.CMESimulation.volume](#)

The documentation for this class was generated from the following file:

- [CME.py](#)

7.7 pyLM.RDME.CMESimulation Class Reference

A [CME](#) simulation that contains reactions and species.

Public Member Functions

- def [__init__](#)
A constructor for the [CMESimulation](#).
- def [defineSpecies](#)
Define a specie/s of particles that exist in the simulation.
- def [addParticles](#)
Add a specified number of particles of the specified type to the specified region.
- def [addConcentration](#)
Add a concentration of particles of the specified type to the simulation.
- def [addReaction](#)
Adds a 0th, 1st or 2nd order reaction.

Public Attributes

- [particleMap](#)
- [species_id](#)
- [initial_counts](#)
- [reactions](#)

- [parameters](#)
- [volume](#)
- [name](#)
- [replicates](#)
- [filename](#)

7.7.1 Detailed Description

A [CME](#) simulation that contains reactions and species.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 `def pyLM.RDME.CMESimulation.__init__(self, volume = None, name = "unnamed")`

A constructor for the [CMESimulation](#).

Parameters

<i>volume</i>	The reaction vessel volume (in Litres). Specifying 'None' signifies that the user has already accounted for volume in rate constants.
<i>name</i>	The name of the CME simulation; Default: "unnamed"

Returns

`self`

7.7.3 Member Function Documentation

7.7.3.1 `def pyLM.RDME.CMESimulation.addConcentration(self, species = 'unknown', conc = 0.0)`

Add a concentration of particles of the specified type to the simulation.

Parameters

<i>self</i>	
<i>species</i>	The name of the specie to add
<i>concentration</i>	The concentration of the species (Molar). Particle count is rounded to nearest integer.

Exceptions

<i>An</i>	error if the volume is not specified
-----------	--------------------------------------

7.7.3.2 `def pyLM.RDME.CMESimulation.addParticles(self, species = 'unknown', count = 1)`

Add a specified number of particles of the specified type to the specified region.

Parameters

<i>self</i>	
<i>species</i>	The name of the specie to add particles to
<i>count</i>	The number of that particle to start with (default 1)

7.7.3.3 `def pyLM.RDME.CMESimulation.addReaction(self, reactant, product, rate)`

Adds a 0th, 1st or 2nd order reaction.

Parameters

<i>self</i>	
<i>reactant</i>	The list or reactants
<i>product</i>	The list of products
<i>rate</i>	The rate of reaction

7.7.3.4 def pyLM.RDME.CMESimulation.defineSpecies (*self*, *species*)

Define a specie/s of particles that exist in the simulation.

Parameters

<i>self</i>	
<i>species</i>	A list of species to add to the simulation

7.7.4 Member Data Documentation

7.7.4.1 pyLM.RDME.CMESimulation.filename

7.7.4.2 pyLM.RDME.CMESimulation.initial_counts

7.7.4.3 pyLM.RDME.CMESimulation.name

7.7.4.4 pyLM.RDME.CMESimulation.parameters

7.7.4.5 pyLM.RDME.CMESimulation.particleMap

7.7.4.6 pyLM.RDME.CMESimulation.reactions

7.7.4.7 pyLM.RDME.CMESimulation.replicates

7.7.4.8 pyLM.RDME.CMESimulation.species_id

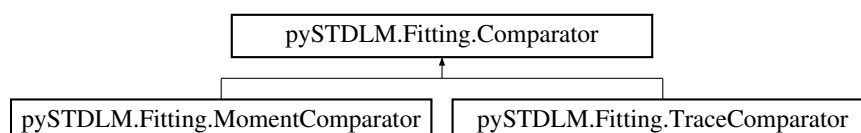
7.7.4.9 pyLM.RDME.CMESimulation.volume

The documentation for this class was generated from the following file:

- [RDME.py](#)

7.8 pySTDLM.Fitting.Comparator Class Reference

Inheritance diagram for pySTDLM.Fitting.Comparator:



Public Member Functions

- def [__init__](#)

- def [computeMetric](#)

Public Attributes

- [specie](#)

7.8.1 Constructor & Destructor Documentation

7.8.1.1 def `pySTDLM.Fitting.Comparator.__init__(self, specie)`

7.8.2 Member Function Documentation

7.8.2.1 def `pySTDLM.Fitting.Comparator.computeMetric (self, fileHandle, time =None, data =None)`

7.8.3 Member Data Documentation

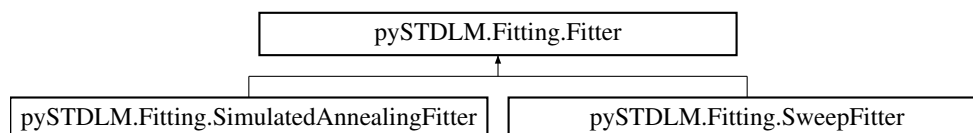
7.8.3.1 `pySTDLM.Fitting.Comparator.specie`

The documentation for this class was generated from the following file:

- [Fitting.py](#)

7.9 pySTDLM.Fitting.Fitter Class Reference

Inheritance diagram for `pySTDLM.Fitting.Fitter`:



Public Member Functions

- def `__init__`
- def [addExperiment](#)
- def [addParameter](#)
- def [performFit](#)
- def [saveFit](#)

Public Attributes

- [experiments](#)
- [parameters](#)
- [hasFit](#)

7.9.1 Constructor & Destructor Documentation

7.9.1.1 `def pySTDLM.Fitting.Fitter.__init__(self)`

7.9.2 Member Function Documentation

7.9.2.1 `def pySTDLM.Fitting.Fitter.addExperiment(self, simulation, comparisons, type, weight=None, name=None)`

7.9.2.2 `def pySTDLM.Fitting.Fitter.addParameter(self, parameter, bounds, initialValue)`

7.9.2.3 `def pySTDLM.Fitting.Fitter.performFit(self, numberProcElements)`

7.9.2.4 `def pySTDLM.Fitting.Fitter.saveFit(self, filebase)`

7.9.3 Member Data Documentation

7.9.3.1 `pySTDLM.Fitting.Fitter.experiments`

7.9.3.2 `pySTDLM.Fitting.Fitter.hasFit`

7.9.3.3 `pySTDLM.Fitting.Fitter.parameters`

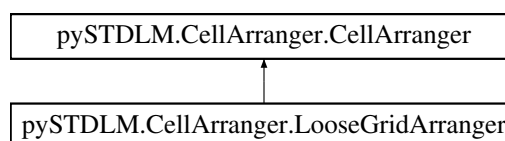
The documentation for this class was generated from the following file:

- [Fitting.py](#)

7.10 pySTDLM.CellArranger.LooseGridArranger Class Reference

An arranger that orients cells in square grid with padding in between.

Inheritance diagram for `pySTDLM.CellArranger.LooseGridArranger`:



Public Member Functions

- `def __init__`
- `def intPackVolume`

Additional Inherited Members

7.10.1 Detailed Description

An arranger that orients cells in square grid with padding in between.

The attribute dictionary requires "offset" which is a list of the form [dx, dy, dz].

7.10.2 Constructor & Destructor Documentation

7.10.2.1 `def pySTDLM.CellArranger.LooseGridArranger.__init__(self, cellType, cellAttributes, arrangerAttributes)`

7.10.3 Member Function Documentation

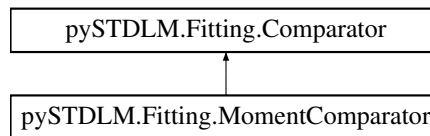
7.10.3.1 `def pySTDLM.CellArranger.LooseGridArranger.intPackVolume(self, sim, mins, maxs)`

The documentation for this class was generated from the following file:

- [CellArranger.py](#)

7.11 pySTDLM.Fitting.MomentComparator Class Reference

Inheritance diagram for pySTDLM.Fitting.MomentComparator:



Public Member Functions

- `def __init__`
- `def computeMetric`

Public Attributes

- `moment`
- `value`
- `sem`

7.11.1 Constructor & Destructor Documentation

7.11.1.1 `def pySTDLM.Fitting.MomentComparator.__init__(self, specie, moment, value = None, sem = None)`

7.11.2 Member Function Documentation

7.11.2.1 `def pySTDLM.Fitting.MomentComparator.computeMetric(self, fileHandle, time = None, data = None)`

7.11.3 Member Data Documentation

7.11.3.1 `pySTDLM.Fitting.MomentComparator.moment`

7.11.3.2 `pySTDLM.Fitting.MomentComparator.sem`

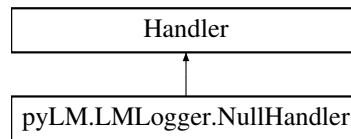
7.11.3.3 `pySTDLM.Fitting.MomentComparator.value`

The documentation for this class was generated from the following file:

- [Fitting.py](#)

7.12 pyLM.LMLogger.NullHandler Class Reference

Inheritance diagram for pyLM.LMLogger.NullHandler:



Public Member Functions

- def [emit](#)

7.12.1 Member Function Documentation

7.12.1.1 def pyLM.LMLogger.NullHandler.emit (*self*, *record*)

The documentation for this class was generated from the following file:

- [LMLogger.py](#)

7.13 pySTDLM.Distributions.ParticleGenerator Class Reference

A particle generator that adds particles to a particular region of a simulation.

Public Member Functions

- def [__init__](#)
Constructor.
- def [addParticlesToRegion](#)
Add a number of particles to a region based on Monte-Carlo sampling of the generator.

Public Attributes

- [generator](#)
- [particlename](#)
- [totalAdded](#)
- [addedLast](#)

7.13.1 Detailed Description

A particle generator that adds particles to a particular region of a simulation.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 def pySTDLM.Distributions.ParticleGenerator.__init__(*self*, *gen*, *ptype*)

Constructor.

Parameters

<i>gen</i>	A PDF set for x,y,z that on the domain [-1,1] and range [0,1) where x is the range of interest
<i>p_{type}</i>	The name of the particle

7.13.3 Member Function Documentation

7.13.3.1 `def pySTDLM.Distributions.ParticleGenerator.addParticlesToRegion (self, rdmesim, region, number, directions, spatial = None)`

Add a number of particles to a region based on Monte-Carlo sampling of the generator.

Parameters

<i>self</i>	
<i>rdmesim</i>	An RDME simulation object that has already been "discretized"
<i>region</i>	A region name of the simulation to add particles to
<i>number</i>	The number of the particle to add
<i>directions</i>	A list of the form (x,y,z) where a 0 indicates a uniform distribution in that direction and a 1 indicates the particles are sampled from the generator in that direction
<i>spatial</i>	(OPTIONAL) A definition of a subsection of the domain that this generation should occur in as a list of lists [[xl,yl,zl],[xh,yh,zh]], otherwise the generator is scaled from (0,0,0) to domain extends (w,h,d)

7.13.4 Member Data Documentation

7.13.4.1 `pySTDLM.Distributions.ParticleGenerator.addedLast`

7.13.4.2 `pySTDLM.Distributions.ParticleGenerator.generator`

7.13.4.3 `pySTDLM.Distributions.ParticleGenerator.particleName`

7.13.4.4 `pySTDLM.Distributions.ParticleGenerator.totalAdded`

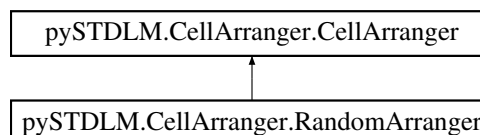
The documentation for this class was generated from the following file:

- [Distributions.py](#)

7.14 pySTDLM.CellArranger.RandomArranger Class Reference

An arranger that randomly orients a number of cells in a region until either a count or a volume fraction target is met.

Inheritance diagram for `pySTDLM.CellArranger.RandomArranger`:



Public Member Functions

- `def __init__`
- `def newBBox`

- def [doesIntersect](#)
- def [outsideDomain](#)
- def [intPackVolume](#)

Additional Inherited Members

7.14.1 Detailed Description

An arranger that randomly orients a number of cells in a regin until either a count or a volume fraction target is met. This class requires either "fraction" or "number" attributes specified in the arrangerAttributes. It also has an optional parameter "padding" which maintains that much padding between the cells and their neighbors.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 def pySTDLM.CellArranger.RandomArranger.__init__(*self*, *cellType*, *cellAttributes*, *arrangerAttributes*)

7.14.3 Member Function Documentation

7.14.3.1 def pySTDLM.CellArranger.RandomArranger.doesIntersect (*self*, *b1*, *b2*)

7.14.3.2 def pySTDLM.CellArranger.RandomArranger.intPackVolume (*self*, *sim*, *mins*, *maxs*)

7.14.3.3 def pySTDLM.CellArranger.RandomArranger.newBBox (*self*, *bbox*, *padding*)

7.14.3.4 def pySTDLM.CellArranger.RandomArranger.outsideDomain (*self*, *b1*, *mins*, *maxs*)

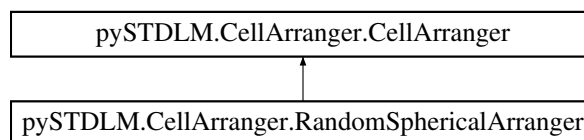
The documentation for this class was generated from the following file:

- [CellArranger.py](#)

7.15 pySTDLM.CellArranger.RandomSphericalArranger Class Reference

An arranger that randomly orients a number of cells in a sphere inside a regin until either a count or a volume fraction target is met.

Inheritance diagram for pySTDLM.CellArranger.RandomSphericalArranger:



Public Member Functions

- def [__init__](#)
- def [intPackVolume](#)

Additional Inherited Members

7.15.1 Detailed Description

An arranger that randomly orients a number of cells in a sphere inside a region until either a count or a volume fraction target is met.

This class requires either "fraction" or "number" attributes specified in the `arrangerAttributes`. It also has an optional parameter "padding" which maintains that much padding between the cells and their neighbors.

7.15.2 Constructor & Destructor Documentation

7.15.2.1 `def pySTDLM.CellArranger.RandomSphericalArranger.__init__(self, cellType, cellAttributes, arrangerAttributes)`

7.15.3 Member Function Documentation

7.15.3.1 `def pySTDLM.CellArranger.RandomSphericalArranger.intPackVolume (self, sim, mins, maxs)`

The documentation for this class was generated from the following file:

- [CellArranger.py](#)

7.16 pyLM.RDME.RDMERegion Class Reference

A class that represents a type region of an [RDME](#) simulation.

Public Member Functions

- `def __init__`
Creates a new [RDMERegion](#).
- `def addReaction`
Adds a 0th, 1st or 2nd order reaction that can occur in the region.
- `def setDefaultDiffusionRate`
Specifies the default diffusion rate of all particles in the region.
- `def setDiffusionRate`
Specify the diffusion rate for a particular particle type.
- `def getReactionCount`
Return the number of reactions defined in this region.

Public Attributes

- `name`
- `reactions`
- `defaultDiffusionRate`
- `diffusionRate`

7.16.1 Detailed Description

A class that represents a type region of an [RDME](#) simulation.

Reactions may be specified that live within a region. In addition a specie's diffusion constant is region dependent and diffusion between regions must be specified. For example: cytosol, membrane, extracellular, nucleoid, etc.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 `def pyLM.RDME.RDMERegion.__init__(self, name)`

Creates a new [RDMERegion](#).

Parameters

<i>self</i>	
<i>name</i>	The name of the region (e.g. cytosol)

7.16.3 Member Function Documentation

7.16.3.1 `def pyLM.RDME.RDMERegion.addReaction (self, reactant, product, rate)`

Adds a 0th, 1st or 2nd order reaction that can occur in the region.

Parameters

<i>self</i>	
<i>reactant</i>	A set of reactants either as a singleton or a list
<i>product</i>	A set of products either as a singleton or a list

Returns

`self`

7.16.3.2 `def pyLM.RDME.RDMERegion.getReactionCount (self)`

Return the number of reactions defined in this region.

Parameters

<i>self</i>	
-------------	--

Returns

Get the number of reactions in the region

7.16.3.3 `def pyLM.RDME.RDMERegion.setDefaultDiffusionRate (self, rate)`

Specifies the default diffusion rate of all particles in the region.

Parameters

<i>self</i>	
<i>rate</i>	The rate of diffusion in um^2/s or um/s for 3D or 2D diffusion

Returns

`self`

7.16.3.4 `def pyLM.RDME.RDMERegion.setDiffusionRate (self, species, rate)`

Specify the diffusion rate for a particular particle type.

Parameters

<i>self</i>	
<i>species</i>	The particle type
<i>rate</i>	The rate of diffusion in um^2/s or um/s for 3D or 2D diffusion

Returns

`self`

7.16.4 Member Data Documentation

7.16.4.1 `pyLM.RDME.RDMERegion.defaultDiffusionRate`

7.16.4.2 `pyLM.RDME.RDMERegion.diffusionRate`

7.16.4.3 `pyLM.RDME.RDMERegion.name`

7.16.4.4 `pyLM.RDME.RDMERegion.reactions`

The documentation for this class was generated from the following file:

- [RDME.py](#)

7.17 pyLM.RDME.RDMESimulation Class Reference

A class that contains all regions, reactions, diffusions and rules for a [RDME](#) simulation.

Public Member Functions

- def `__init__`
Specify a cuboid region that represents the extents to the reaction region as well as the lattice spacing.
- def `addRegion`
Add a region to the simulation.
- def `addCuboidRegion`
Add a cuboid to the builder.
- def `addShape`
- def `modifyRegion`
Return a pointer to a region so that it may be modified.
- def `defineSpecies`
Define a specie/s of particles that exist in the simulation.
- def `siteVolume`
Get the actual volume of a specific site in L.
- def `buildCapsidCell`
Build a capsule based shell in this [RDMESimulation](#) centered within the simulation domain that includes a membrane and cytoplasm.
- def `buildSphericalCell`
Build a spherical based shell in this [RDMESimulation](#) centered within the simulation domain that includes a membrane and cytoplasm.
- def `addParticles`
Add a specified number of particles of the specified type to the specified region.
- def `packRegion`
Add nonmoving obstacles to a particular region.

- def [setTransitionRate](#)
Specify the diffusion rate between species; this is a one directional rate e.g.
- def [setTwoWayTransitionRate](#)
Specify the diffusion rate between species; this is a two directional rate.
- def [buildDiffusionModel](#)
Return the Lattice Microbes DiffusionModel object for fine-tuning.
- def [buildReactionModel](#)
Return the Lattice Microbes ReactionModel object for fine-tuning.
- def [buildSpatialModel](#)
Return the Lattice Microbes SpatialModel object for fine-tuning.
- def [setWriteInterval](#)
- def [setLatticeWriteInterval](#)
- def [setSimulationTime](#)
- def [setTimestep](#)
- def [setRandomSeed](#)
- def [getLattice](#)
Get a discretized version of the simulation domain.
- def [setLatticeSite](#)
Set a particular lattice site type.
- def [getLatticeSite](#)
Get a particular lattice site.
- def [addParticleAt](#)
Add a particle/ to a particular site.
- def [addParticleAtIdx](#)
Create an HDF5 version of the simulation amenable for later running or stand-alone running.
- def [save](#)
Create an HDF5 version of the simulation amenable for later running or stand-alone running.
- def [run](#)
- def [runMPI](#)
Run the simulation using a call to mpirun with the given options.
- def [runSolver](#)
- def [buildVector](#)
- def [buildPoint](#)
- def [buildSphere](#)
- def [buildEllipse](#)
- def [buildDifference](#)

Public Attributes

- [siteTypes](#)
- [particleMap](#)
- [customAddedParticleList](#)
- [species_id](#)
- [regions](#)
- [transitionRates](#)
- [initial_counts](#)
- [parameters](#)
- [continousDimensions](#)
- [latticeSpacing](#)
- [lm_builder](#)
- [lattice](#)
- [hasBeenDiscretized](#)
- [name](#)
- [replicates](#)
- [filename](#)

7.17.1 Detailed Description

A class that contains all regions, reactions, diffusions and rules for a [RDME](#) simulation.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 `def pyLM.RDME.RDMESimulation.__init__(self, dimensions, spacing, name = "unnamed", defaultRegion = "default")`

Specify a cuboid region that represents the extents to the reaction region as well as the lattice spacing.

Parameters

<i>self</i>	
<i>dimensions</i>	A list of [x,y,z]
<i>spacing</i>	Lattice spacing
<i>name</i>	The name of the RDME simulation; default: "unnamed"
<i>defaultRegion</i>	The name of the region that is associated with the lattice sites before any other regions are added; default:"default"

Returns

self

7.17.3 Member Function Documentation

7.17.3.1 `def pyLM.RDME.RDMESimulation.addCuboidRegion(self, name, a, b)`

Add a cuboid to the builder.

Parameters

<i>self</i>	
<i>name</i>	Name of the site type for this region
<i>a</i>	tuple for the first corner in continuous space
<i>b</i>	tuple for the second corner in continuous space

7.17.3.2 `def pyLM.RDME.RDMESimulation.addParticleAt(self, index, particleType)`

Add a particle/ to a particular site.

Parameters

<i>index</i>	(x,y,z) a list of spatial location
<i>specie</i>	The specie type to add

7.17.3.3 `def pyLM.RDME.RDMESimulation.addParticleAtIdx(self, index, particleType)`

Create an HDF5 version of the simulation amenable for later running or stand-alone running.

Parameters

<i>self</i>	
-------------	--

<i>filename</i>	A file to write the simulation to Add a particle/ to a particular site
<i>index</i>	(x,y,z) a list of lattice site indices
<i>specie</i>	The specie type to add

7.17.3.4 `def pyLM.RDME.RDMESimulation.addParticles (self, species = ' unknown', region = ' default', count = 1)`

Add a specified number of particles of the specified type to the specified region.

Parameters

<i>self</i>	
<i>species</i>	The species to add to the region
<i>region</i>	The region to add particles to
<i>count</i>	Number of particles to add (default: 1)

Returns

The simulation object

7.17.3.5 `def pyLM.RDME.RDMESimulation.addRegion (self, region)`

Add a region to the simulation.

Parameters

<i>self</i>	
<i>region</i>	The region to add to the simulation

Returns

The region just added

7.17.3.6 `def pyLM.RDME.RDMESimulation.addShape (self, shape)`

7.17.3.7 `def pyLM.RDME.RDMESimulation.buildCapsidCell (self, length, diameter, membraneThickness, points = False)`

Build a capsule based shell in this [RDMESimulation](#) centered within the simulation domain that includes a membrane and cytoplasm.

Parameters

<i>self</i>	
<i>length</i>	The length of the capsule from one sphere origin to the other
<i>diameter</i>	The diameter of the cell
<i>membrane-Thickness</i>	The thickness of the membrane
<i>points</i>	OPTIONAL: List of lists containing the coordinates of the centers of the spheroids that cap the capsid cell, e.g. [[x1, y1, z1], [x2, y2, z2]]. If not given, cell is centered in the volume and aligned in the z-direction

Returns

The simulation object

7.17.3.8 `def pyLM.RDME.RDMESimulation.buildDifference (self, arg1, arg2, arg3)`

7.17.3.9 `def pyLM.RDME.RDMESimulation.buildDiffusionModel (self)`

Return the Lattice Microbes DiffusionModel object for fine-tuning.

Parameters

<i>self</i>	The diffusion mode for the simulation
-------------	---------------------------------------

7.17.3.10 `def pyLM.RDME.RDMESimulation.buildEllipse (self, arg1, arg2, arg3, arg4, arg5, arg6, arg7)`

7.17.3.11 `def pyLM.RDME.RDMESimulation.buildPoint (self, arg1, arg2, arg3)`

7.17.3.12 `def pyLM.RDME.RDMESimulation.buildReactionModel (self)`

Return the Lattice Microbes ReactionModel object for fine-tuning.

Parameters

<i>self</i>	
-------------	--

Returns

The reaction model for this simulation

7.17.3.13 `def pyLM.RDME.RDMESimulation.buildSpatialModel (self)`

Return the Lattice Microbes SpatialModel object for fine-tuning.

Parameters

<i>self</i>	
-------------	--

Returns

The spatial model (i.e. obstacles, sites, etc.) for this simulation

7.17.3.14 `def pyLM.RDME.RDMESimulation.buildSphere (self, arg1, arg2, arg3)`

7.17.3.15 `def pyLM.RDME.RDMESimulation.buildSphericalCell (self, diameter, membraneThickness, point = False)`

Build a spherical based shell in this [RDMESimulation](#) centered within the simulation domain that includes a membrane and cytoplasm.

Parameters

<i>self</i>	
<i>diameter</i>	The diameter of the cell
<i>membrane-Thickness</i>	The thickness of the membrane

<i>point</i>	The center of the spherical cell
--------------	----------------------------------

Returns

The simulation object

7.17.3.16 `def pyLM.RDME.RDMESimulation.buildVector (self, arg1, arg2, arg3)`

7.17.3.17 `def pyLM.RDME.RDMESimulation.defineSpecies (self, species)`

Define a specie/s of particles that exist in the simulation.

Parameters

<i>self</i>	
<i>species</i>	A list of species to add to the simulation

Returns

The simulation object

7.17.3.18 `def pyLM.RDME.RDMESimulation.getLattice (self)`

Get a discretized version of the simulation domain.

Call this after building all spherical and capsule cells

Parameters

<i>self</i>	
-------------	--

Returns

A lattice object. This function should only be called once.

7.17.3.19 `def pyLM.RDME.RDMESimulation.getLatticeSite (self, index)`

Get a particular lattice site.

Parameters

<i>index</i>	(x,y,z) a list of coordinates
--------------	-------------------------------

Returns

The type of the lattice site (string)

7.17.3.20 `def pyLM.RDME.RDMESimulation.modifyRegion (self, region)`

Return a pointer to a region so that it may be modified.

Parameters

<i>self</i>	
<i>region</i>	Get a region that is attached to the simulation for modification

Returns

The region to modify

7.17.3.21 `def pyLM.RDME.RDMESimulation.packRegion (self, region, radius, percentage, obstacleID)`

Add nonmoving obstacles to a particular region.

Parameters

<i>self</i>	
<i>region</i>	The name of the region in which to add particles to
<i>radius</i>	The radius of the particles
<i>percentage</i>	The percentage of the total region volume that should be packed
<i>obstacleID</i>	an identifier for the obstacle

Returns

The simulation object

7.17.3.22 `def pyLM.RDME.RDMESimulation.run (self, filename, method, replicates = 1, seed = None)`

7.17.3.23 `def pyLM.RDME.RDMESimulation.runMPI (self, filename, method, replicates = 1, driver = "mpirun", ppe = 1, seed = None)`

Run the simulation using a call to mpirun with the given options.

Parameters

<i>self</i>	
<i>filename</i>	The HDF file to write to
<i>method</i>	The class name for the solver to use (e.g., <code>lm::cme::GillespieDSolver</code>)
<i>replicates</i>	The number of replicates to serially run
<i>driver</i>	The program to execute the parallel run, e.g. "mpirun", "aprun", "ibrun", etc.
<i>ppe</i>	The number of processing elements to use
<i>seed</i>	A seed for the random number generator to use when running the simulation; None indicates default

7.17.3.24 `def pyLM.RDME.RDMESimulation.runSolver (self, filename, solver, replicates = 1)`

7.17.3.25 `def pyLM.RDME.RDMESimulation.save (self, filename)`

Create an HDF5 version of the simulation amenable for later running or stand-alone running.

Parameters

<i>self</i>	
-------------	--

<i>filename</i>	A file to write the simulation to
-----------------	-----------------------------------

7.17.3.26 `def pyLM.RDME.RDMESimulation.setLatticeSite (self, index, siteType)`

Set a particular lattice site type.

Parameters

<i>index</i>	(x,y,z) a list of coordinates
<i>siteType</i>	The type to set the lattice point to. This would be the name of a region that has previously been performed"

7.17.3.27 `def pyLM.RDME.RDMESimulation.setLatticeWriteInterval (self, time)`

7.17.3.28 `def pyLM.RDME.RDMESimulation.setRandomSeed (self, seed)`

7.17.3.29 `def pyLM.RDME.RDMESimulation.setSimulationTime (self, time)`

7.17.3.30 `def pyLM.RDME.RDMESimulation.setTimeStep (self, time)`

7.17.3.31 `def pyLM.RDME.RDMESimulation.setTransitionRate (self, species, via, to, rate)`

Specify the diffusion rate between species; this is a one directional rate e.g.

membrane->cytosol or extracellular->membrane

Parameters

<i>self</i>	
<i>species</i>	The specie that can transition between regions
<i>via</i>	From this region
<i>to</i>	To this region
<i>rate</i>	Diffusion rate between regions

Returns

The simulation object

7.17.3.32 `def pyLM.RDME.RDMESimulation.setTwoWayTransitionRate (self, species, one, two, rate)`

Specify the diffusion rate between species; this is a two directional rate.

Parameters

<i>self</i>	
<i>species</i>	The specie that can transition between regions
<i>one</i>	A region
<i>two</i>	The other region
<i>rate</i>	Diffusion rate between regions

Returns

The simulation object

7.17.3.33 `def pyLM.RDME.RDMESimulation.setWriteInterval (self, time)`

7.17.3.34 `def pyLM.RDME.RDMESimulation.siteVolume (self)`

Get the actual volume of a specific site in L.

Returns

the volume of the site in L

7.17.4 Member Data Documentation

7.17.4.1 `pyLM.RDME.RDMESimulation.continuousDimensions`

7.17.4.2 `pyLM.RDME.RDMESimulation.customAddedParticleList`

7.17.4.3 `pyLM.RDME.RDMESimulation.filename`

7.17.4.4 `pyLM.RDME.RDMESimulation.hasBeenDiscretized`

7.17.4.5 `pyLM.RDME.RDMESimulation.initial_counts`

7.17.4.6 `pyLM.RDME.RDMESimulation.lattice`

7.17.4.7 `pyLM.RDME.RDMESimulation.latticeSpacing`

7.17.4.8 `pyLM.RDME.RDMESimulation.lm_builder`

7.17.4.9 `pyLM.RDME.RDMESimulation.name`

7.17.4.10 `pyLM.RDME.RDMESimulation.parameters`

7.17.4.11 `pyLM.RDME.RDMESimulation.particleMap`

7.17.4.12 `pyLM.RDME.RDMESimulation.regions`

7.17.4.13 `pyLM.RDME.RDMESimulation.replicates`

7.17.4.14 `pyLM.RDME.RDMESimulation.siteTypes`

7.17.4.15 `pyLM.RDME.RDMESimulation.species_id`

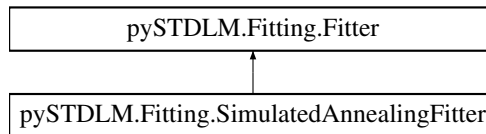
7.17.4.16 `pyLM.RDME.RDMESimulation.transitionRates`

The documentation for this class was generated from the following file:

- [RDME.py](#)

7.18 pySTDLM.Fitting.SimulatedAnnealingFitter Class Reference

Inheritance diagram for pySTDLM.Fitting.SimulatedAnnealingFitter:



Public Member Functions

- def [__init__](#)
- def [performFit](#)
- def [saveFit](#)

Additional Inherited Members

7.18.1 Constructor & Destructor Documentation

7.18.1.1 def `pySTDLM.Fitting.SimulatedAnnealingFitter.__init__(self)`

7.18.2 Member Function Documentation

7.18.2.1 def `pySTDLM.Fitting.SimulatedAnnealingFitter.performFit(self, numberProcElements)`

7.18.2.2 def `pySTDLM.Fitting.SimulatedAnnealingFitter.saveFit(self, filebase)`

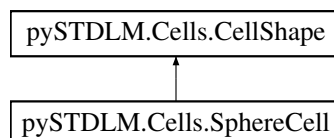
The documentation for this class was generated from the following file:

- [Fitting.py](#)

7.19 pySTDLM.Cells.SphereCell Class Reference

A representation for a spherical cell.

Inheritance diagram for `pySTDLM.Cells.SphereCell`:



Public Member Functions

- def [__init__](#)
- def [internalModelCell](#)
- def [computeVolume](#)
- def [internalAddToSim](#)
- def [internalTranslateCell](#)

Public Attributes

- [name](#)
- [origin](#)
- [radius](#)
- [mins](#)
- [maxs](#)
- [membraneThickness](#)

7.19.1 Detailed Description

A representation for a spherical cell.

This cell requires an attribute dictionary that includes "radius" and "membraneThickness"

7.19.2 Constructor & Destructor Documentation

7.19.2.1 `def pySTDLM.Cells.SphereCell.__init__(self, origin = 3*[0.], radius = 0.)`

7.19.3 Member Function Documentation

7.19.3.1 `def pySTDLM.Cells.SphereCell.computeVolume(self)`

7.19.3.2 `def pySTDLM.Cells.SphereCell.internalAddToSim(self, sim)`

7.19.3.3 `def pySTDLM.Cells.SphereCell.internalModelCell(self, attr)`

7.19.3.4 `def pySTDLM.Cells.SphereCell.internalTranslateCell(self, point)`

7.19.4 Member Data Documentation

7.19.4.1 `pySTDLM.Cells.SphereCell.maxs`

7.19.4.2 `pySTDLM.Cells.SphereCell.membraneThickness`

7.19.4.3 `pySTDLM.Cells.SphereCell.mins`

7.19.4.4 `pySTDLM.Cells.SphereCell.name`

7.19.4.5 `pySTDLM.Cells.SphereCell.origin`

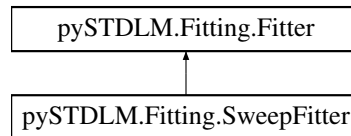
7.19.4.6 `pySTDLM.Cells.SphereCell.radius`

The documentation for this class was generated from the following file:

- [Cells.py](#)

7.20 pySTDLM.Fitting.SweepFitter Class Reference

Inheritance diagram for pySTDLM.Fitting.SweepFitter:



Public Member Functions

- def [__init__](#)
- def [setParameterDivisions](#)
- def [performFit](#)
- def [saveFit](#)

Public Attributes

- [divisions](#)
- [hasFit](#)

7.20.1 Constructor & Destructor Documentation

7.20.1.1 def `pySTDLM.Fitting.SweepFitter.__init__(self)`

7.20.2 Member Function Documentation

7.20.2.1 def `pySTDLM.Fitting.SweepFitter.performFit(self, numberProcElements)`

7.20.2.2 def `pySTDLM.Fitting.SweepFitter.saveFit(self, filebase, plot)`

7.20.2.3 def `pySTDLM.Fitting.SweepFitter.setParameterDivisions(self, parameter, divisions, scale)`

7.20.3 Member Data Documentation

7.20.3.1 `pySTDLM.Fitting.SweepFitter.divisions`

7.20.3.2 `pySTDLM.Fitting.SweepFitter.hasFit`

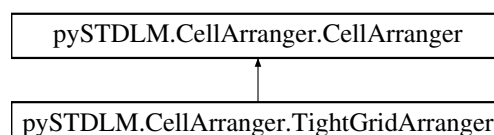
The documentation for this class was generated from the following file:

- [Fitting.py](#)

7.21 pySTDLM.CellArranger.TightGridArranger Class Reference

An arranger that orients cells in square grid.

Inheritance diagram for `pySTDLM.CellArranger.TightGridArranger`:



Public Member Functions

- def [__init__](#)
- def [intPackVolume](#)

Additional Inherited Members

7.21.1 Detailed Description

An arranger that orients cells in square grid.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 `def pySTDLM.CellArranger.TightGridArranger.__init__(self, cellType, cellAttributes, arrangerAttributes)`

7.21.3 Member Function Documentation

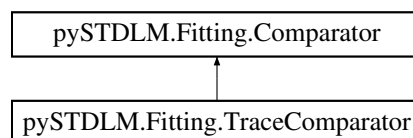
7.21.3.1 `def pySTDLM.CellArranger.TightGridArranger.intPackVolume (self, sim, mins, maxs)`

The documentation for this class was generated from the following file:

- [CellArranger.py](#)

7.22 pySTDLM.Fitting.TraceComparator Class Reference

Inheritance diagram for pySTDLM.Fitting.TraceComparator:



Public Member Functions

- def [__init__](#)
- def [computeMetric](#)

Public Attributes

- [comparator](#)
- [times](#)
- [data](#)

7.22.1 Constructor & Destructor Documentation

7.22.1.1 `def pySTDLM.Fitting.TraceComparator.__init__(self, comp, times, data)`

7.22.2 Member Function Documentation

7.22.2.1 `def pySTDLM.Fitting.TraceComparator.computeMetric (self, fileHandle, time = None, data = None)`

7.22.3 Member Data Documentation

7.22.3.1 `pySTDLM.Fitting.TraceComparator.comparator`

7.22.3.2 `pySTDLM.Fitting.TraceComparator.data`

7.22.3.3 `pySTDLM.Fitting.TraceComparator.times`

The documentation for this class was generated from the following file:

- [Fitting.py](#)

Chapter 8

File Documentation

8.1 `__init__.py` File Reference

Namespaces

- [pyLM](#)

The problem solving environment which interfaces to Lattice Microbes simulation software.

Variables

- list [pyLM.__all__](#) = ['LMLogger', 'CME', 'RDME', 'units', 'ipyInterface']

8.2 `__init__.py` File Reference

Namespaces

- [pySTDLM](#)

A standard library of functionality that is commonly used in biological simulations.

Variables

- list [pySTDLM.__all__](#) = ['StandardReactions', 'StandardReactionSystems', 'StandardCells', 'Cells', 'CellArranger', 'Distributions', 'PostProcessing', 'NetworkVisualization']

8.3 `CellArranger.py` File Reference

Classes

- class [pySTDLM.CellArranger.CellArranger](#)
A base class for laying out cells in a 3D space.
- class [pySTDLM.CellArranger.RandomArranger](#)
An arranger that randomly orients a number of cells in a regin until either a count or a volume fraction target is met.
- class [pySTDLM.CellArranger.RandomSphericalArranger](#)
An arranger that randomly orients a number of cells in a sphere inside a regin until either a count or a volume fraction target is met.
- class [pySTDLM.CellArranger.TightGridArranger](#)

An arranger that orients cells in square grid.

- class [pySTDLM.CellArranger.LooseGridArranger](#)

An arranger that orients cells in square grid with padding in between.

Namespaces

- [pySTDLM.CellArranger](#)

8.4 Cells.py File Reference

Classes

- class [pySTDLM.Cells.CellShape](#)
Base class for a particular cell type for use in [CellArranger](#) (and later more)
- class [pySTDLM.Cells.SphereCell](#)
A representation for a spherical cell.
- class [pySTDLM.Cells.BoxCell](#)
A representation for a spherical cell.
- class [pySTDLM.Cells.CapsuleCell](#)
A representation for a spherical cell.
- class [pySTDLM.Cells.CapsuleShellCell](#)
A representation for a spherical cell.

Namespaces

- [pySTDLM.Cells](#)

8.5 CME.py File Reference

Classes

- class [pyLM.CME.CMESimulation](#)
A CME simulation that contains reactions and species.

Namespaces

- [pyLM.CME](#)

Functions

- def [pyLM.CME.tqdm](#)

8.6 Distributions.py File Reference

Classes

- class [pySTDLM.Distributions.ParticleGenerator](#)
A particle generator that adds particles to a particular region of a simulation.

Namespaces

- [pySTDLM.Distributions](#)

8.7 Fitting.py File Reference

Classes

- class [pySTDLM.Fitting.Comparator](#)
- class [pySTDLM.Fitting.MomentComparator](#)
- class [pySTDLM.Fitting.TraceComparator](#)
- class [pySTDLM.Fitting.Fitter](#)
- class [pySTDLM.Fitting.SweepFitter](#)
- class [pySTDLM.Fitting.SimulatedAnnealingFitter](#)

Namespaces

- [pySTDLM.Fitting](#)

Functions

- def [pySTDLM.Fitting.DistributionComparator](#)

Variables

- [pySTDLM.Fitting.counts](#)
- [pySTDLM.Fitting.binwidth](#)
- [pySTDLM.Fitting.measure](#)

8.8 ipyInterface.py File Reference

Namespaces

- [pyLM.ipyInterface](#)

Functions

- def [pyLM.ipyInterface.getReactionString](#)
Write a reaction to a string with the correct units.
- def [pyLM.ipyInterface.writeTable](#)
Write an HTML formatted table to a string.
- def [pyLM.ipyInterface.displayRDMESetup](#)
Display an [RDME](#) initial conditions in a Jupyter Notebook using ipywidgets.

8.9 LMLogger.py File Reference

Classes

- class [pyLM.LMLogger.NullHandler](#)

Namespaces

- [pyLM.LMLogger](#)

Functions

- def [pyLM.LMLogger.setLMLoggerLevel](#)
Set the level of the logger for the application.
- def [pyLM.LMLogger.setLMLogFile](#)
Set up file handler to print log to file.
- def [pyLM.LMLogger.setLMLogConsole](#)
Set the logger to write to the console as the code is working.

Variables

- tuple [pyLM.LMLogger.LMLogger](#) = logging.getLogger('LMLogger')
- tuple [pyLM.LMLogger.nullHandlerLM](#) = logging.NullHandler()
- tuple [pyLM.LMLogger.LMformatter](#) = logging.Formatter('%(asctime)s: %(levelname)s: %(message)s')

8.10 NetworkVisualization.py File Reference

Namespaces

- [pySTDLM.NetworkVisualization](#)

Functions

- def [pySTDLM.NetworkVisualization.plotReactionNetwork](#)
Plot the reaction scheme as a network.
- def [pySTDLM.NetworkVisualization.plotCMEReactionNetwork](#)
Plot the reaction scheme as a network.
- def [pySTDLM.NetworkVisualization.plotRDMEReactionNetwork](#)
Plot the reaction scheme as a network.
- def [pySTDLM.NetworkVisualization.plotCMEDynamicReactionNetwork](#)
Plot the dynamics of the species on the network into a dynamic graph file (extension: .gexf)

Variables

- [pySTDLM.NetworkVisualization.hasGexf](#) = True

8.11 OnTheFly.py File Reference

Namespaces

- [pySTDLM.OnTheFly](#)

8.12 peekFile.py File Reference

Namespaces

- [peekFile](#)

Variables

- tuple [peekFile.parser](#) = argparse.ArgumentParser(description="Print info on simulation file")
- tuple [peekFile.args](#) = parser.parse_args()
- tuple [peekFile.tf](#) = float(hdf['Parameters'].attrs['maxTime'])
- list [peekFile.t](#) = sim['LatticeTimes']
- tuple [peekFile.n](#) = len(sim['Lattice'])

8.13 PostProcessing.py File Reference

Namespaces

- [pySTDLM.PostProcessing](#)

Functions

- def [pySTDLM.PostProcessing.openLMFile](#)
Open a Lattice Microbes File for reading.
- def [pySTDLM.PostProcessing.closeLMFile](#)
Close a Lattice Microbes File.
- def [pySTDLM.PostProcessing.showTraceFromFile](#)
Show species trace from a particular replicate.
- def [pySTDLM.PostProcessing.plotTraceFromFile](#)
Plot species from an output file.
- def [pySTDLM.PostProcessing.showTrace](#)
Show a specific species trace.
- def [pySTDLM.PostProcessing.plotTrace](#)
Plot a specific species trace.
- def [pySTDLM.PostProcessing.showAvgVarFromFile](#)
Show species from an output file.
- def [pySTDLM.PostProcessing.plotAvgVarFromFile](#)
Plot species from an output file.
- def [pySTDLM.PostProcessing.showAvgVar](#)
Show a specific species average over time and variance.
- def [pySTDLM.PostProcessing.plotAvgVar](#)
Plot a specific species average over time and variance.
- def [pySTDLM.PostProcessing.getOccupancyKymograph](#)
Compute the specie density(occupancy) among a slice of the simulation domain as a function over time for the given direction.
- def [pySTDLM.PostProcessing.plotOccupancyKymograph](#)
Plot a specific specie density(occupancy) among a slice of the simulation domain as a function of a direction over time.
- def [pySTDLM.PostProcessing.showOccupancyKymograph](#)
Show a specific specie density(occupancy) among a slice of the simulation domain as a function of a direction over time.

- def [pySTDLM.PostProcessing.plotPhaseSpace](#)
Plot the 2D or 3D phase space associated with the given species over the replicates indicated.
- def [pySTDLM.PostProcessing.getTimeSteps](#)
Extract the timestep times.
- def [pySTDLM.PostProcessing.getSpecieTrace](#)
Extract data for a particular specie for the specified replicate.
- def [pySTDLM.PostProcessing.getAvgVarTrace](#)
Get the average and variance of the specie trace over time.
- def [pySTDLM.PostProcessing.getHistogram](#)
Get the histogram for a specie.
- def [pySTDLM.PostProcessing.getPhaseSpace](#)
Get the nD phase space associated with the traces of species.

8.14 RDME.py File Reference

Classes

- class [pyLM.RDME.CMESimulation](#)
A CME simulation that contains reactions and species.
- class [pyLM.RDME.RDMERegion](#)
A class that represents a type region of an RDME simulation.
- class [pyLM.RDME.RDMESimulation](#)
A class that contains all regions, reactions, diffusions and rules for a RDME simulation.

Namespaces

- [pyLM.RDME](#)

Functions

- def [pyLM.RDME.tqdm](#)

8.15 SBMLReader.py File Reference

Namespaces

- [pySTDLM.SBMLReader](#)

Functions

- def [pySTDLM.SBMLReader.importSBMLModel](#)
Helper Functions ##.
- def [pySTDLM.SBMLReader.importSBMLModelL3V1](#)
- def [pySTDLM.SBMLReader.getRate](#)
- def [pySTDLM.SBMLReader.readSBMLtoCME](#)
Read an SBML file for the reaction model for a CME simulation.
- def [pySTDLM.SBMLReader.readSBMLtoRDME](#)
Read an SBML file for the reaction model for a RDME simulation.
- def [pySTDLM.SBMLReader.readSBMLtoRegion](#)
Read an SBML file for the reaction model into a specific region of the RDME simulation.

8.16 StandardCells.py File Reference

Namespaces

- [pySTDLM.StandardCells](#)

Functions

- def [pySTDLM.StandardCells.buildEColiCell](#)
Build a standard E coli cell 0.5 microns across and 2 microns long.
- def [pySTDLM.StandardCells.buildDividingEColiCell](#)
Build a standard E coli cell 0.5 microns across and 4 microns long.
- def [pySTDLM.StandardCells.buildSphericalEColiCell](#)
Build a Spherical E coli cell, one that lacks the gene for elongation.
- def [pySTDLM.StandardCells.buildFilamentousEColiCell](#)
Build a long filamentous E coli cell where the user can specify the length.
- def [pySTDLM.StandardCells.packFastGrowingEcoli](#)
Pack a cell with the protein distribution for fast growing E coli.
- def [pySTDLM.StandardCells.packSlowGrowingEcoli](#)
Pack a cell with the protein distribution for fast growing E coli.

8.17 StandardReactions.py File Reference

Namespaces

- [pySTDLM.StandardReactions](#)

Functions

- def [pySTDLM.StandardReactions.addMembraneTransporter](#)
Adds a membrane transport system to an RDME system.
- def [pySTDLM.StandardReactions.addPassiveTransport](#)
Adds a passive transport system to an RDME system.
- def [pySTDLM.StandardReactions.addMichaelisMenten](#)
Adds a Michaelis Menten Reaction.
- def [pySTDLM.StandardReactions.addReversibleMichaelisMenten](#)
Add a Reversible Michaelis Menten Reaction.
- def [pySTDLM.StandardReactions.createExpressionModel](#)
Create a set of gene/mRNA/protein reactions based on a genebank file and a set of rates assuming constitutive expression.

Variables

- [pySTDLM.StandardReactions.hasBio](#) = True

8.18 StandardReactionSystems.py File Reference

Namespaces

- [pySTDLM.StandardReactionSystems](#)

Functions

- def [pySTDLM.StandardReactionSystems.addMinDESystem](#)
Adds the MinDE system in Ecoli as a standard reaction network.
- def [pySTDLM.StandardReactionSystems.addLacTwoStateSystem](#)
Adds the Lac switch reaction with two states of the DNA.
- def [pySTDLM.StandardReactionSystems.addPTSPathway](#)
Adds the Phosphoenolpyruvate-depenedent phosphotransferase reaction system.

8.19 units.py File Reference

Namespaces

- [pyLM.units](#)

Functions

- def [pyLM.units.angstrom](#)
Length Wrapper Functions #.
- def [pyLM.units.nm](#)
Returns a representation of a number in nanometers.
- def [pyLM.units.micron](#)
Returns a representation of a number in micrometers.
- def [pyLM.units.mm](#)
Returns a representation of a number in millimeter.
- def [pyLM.units.cm](#)
Returns a representation of a number in centimeter.
- def [pyLM.units.ns](#)
Time Wrapper Functions #.
- def [pyLM.units.microsecond](#)
Returns a representation of a number in microsecond.
- def [pyLM.units.ms](#)
Returns a representation of a number in millisecond.
- def [pyLM.units.second](#)
Returns seconds in seconds.
- def [pyLM.units.minute](#)
Returns a representation of a number in minutes.
- def [pyLM.units.hr](#)
Returns a representation of a number in hours.
- def [pyLM.units.day](#)
Returns a representation of a number in days.

Index

- `__all__`
 - `pyLM`, 12
 - `pySTDLM`, 18
- `__init__`
 - `pyLM::CME::CMESimulation`, 44
 - `pyLM::RDME::CMESimulation`, 48
 - `pyLM::RDME::RDMERegion`, 57
 - `pyLM::RDME::RDMESimulation`, 61
 - `pySTDLM::CellArranger::CellArranger`, 40
 - `pySTDLM::CellArranger::LooseGridArranger`, 52
 - `pySTDLM::CellArranger::RandomArranger`, 55
 - `pySTDLM::CellArranger::RandomSphericalArranger`, 56
 - `pySTDLM::CellArranger::TightGridArranger`, 71
 - `pySTDLM::Cells::BoxCell`, 36
 - `pySTDLM::Cells::CapsuleCell`, 37
 - `pySTDLM::Cells::CapsuleShellCell`, 38
 - `pySTDLM::Cells::CellShape`, 42
 - `pySTDLM::Cells::SphereCell`, 69
 - `pySTDLM::Distributions::ParticleGenerator`, 53
 - `pySTDLM::Fitting::Comparator`, 50
 - `pySTDLM::Fitting::Fitter`, 51
 - `pySTDLM::Fitting::MomentComparator`, 52
 - `pySTDLM::Fitting::SimulatedAnnealingFitter`, 68
 - `pySTDLM::Fitting::SweepFitter`, 70
 - `pySTDLM::Fitting::TraceComparator`, 71
- `__init__.py`, 73
- `addConcentration`
 - `pyLM::CME::CMESimulation`, 44
 - `pyLM::RDME::CMESimulation`, 48
- `addCuboidRegion`
 - `pyLM::RDME::RDMESimulation`, 61
- `addExperiment`
 - `pySTDLM::Fitting::Fitter`, 51
- `addLacTwoStateSystem`
 - `pySTDLM::StandardReactionSystems`, 33
- `addMembraneTransporter`
 - `pySTDLM::StandardReactions`, 30
- `addMichaelisMenten`
 - `pySTDLM::StandardReactions`, 31
- `addMinDESsystem`
 - `pySTDLM::StandardReactionSystems`, 33
- `addPTSPathway`
 - `pySTDLM::StandardReactionSystems`, 33
- `addParameter`
 - `pySTDLM::Fitting::Fitter`, 51
- `addParticleAt`
 - `pyLM::RDME::RDMESimulation`, 61
- `addParticleAtIndex`
 - `pyLM::RDME::RDMESimulation`, 61
- `addParticles`
 - `pyLM::CME::CMESimulation`, 45
 - `pyLM::RDME::CMESimulation`, 48
 - `pyLM::RDME::RDMESimulation`, 62
- `addParticlesToRegion`
 - `pySTDLM::Distributions::ParticleGenerator`, 54
- `addPassiveTransport`
 - `pySTDLM::StandardReactions`, 31
- `addReaction`
 - `pyLM::CME::CMESimulation`, 45
 - `pyLM::RDME::CMESimulation`, 48
 - `pyLM::RDME::RDMERegion`, 58
- `addRegion`
 - `pyLM::RDME::RDMESimulation`, 62
- `addReversibleMichaelisMenten`
 - `pySTDLM::StandardReactions`, 31
- `addShape`
 - `pyLM::RDME::RDMESimulation`, 62
- `addToSimulation`
 - `pySTDLM::CellArranger::CellArranger`, 40
 - `pySTDLM::Cells::CellShape`, 42
- `addedLast`
 - `pySTDLM::Distributions::ParticleGenerator`, 54
- `allowedCellTypes`
 - `pySTDLM::CellArranger::CellArranger`, 40
- `allowedForms`
 - `pySTDLM::CellArranger::CellArranger`, 40
- `angstrom`
 - `pyLM::units`, 15
- `args`
 - `peekFile`, 11
- `arrangerAttributes`
 - `pySTDLM::CellArranger::CellArranger`, 41
- `binwidth`
 - `pySTDLM::Fitting`, 19
- `boundingBox`
 - `pySTDLM::Cells::CellShape`, 42
- `buildCapsidCell`
 - `pyLM::RDME::RDMESimulation`, 62
- `buildDifference`
 - `pyLM::RDME::RDMESimulation`, 62
- `buildDiffusionModel`
 - `pyLM::RDME::RDMESimulation`, 63
- `buildDividingEColiCell`
 - `pySTDLM::StandardCells`, 28
- `buildEColiCell`
 - `pySTDLM::StandardCells`, 28
- `buildEllipse`

- pyLM::RDME::RDMESimulation, 63
- buildFilamentousEColiCell
 - pySTDLM::StandardCells, 28
- buildPoint
 - pyLM::RDME::RDMESimulation, 63
- buildReactionModel
 - pyLM::CME::CMESimulation, 45
 - pyLM::RDME::RDMESimulation, 63
- buildSpatialModel
 - pyLM::RDME::RDMESimulation, 63
- buildSphere
 - pyLM::RDME::RDMESimulation, 63
- buildSphericalCell
 - pyLM::RDME::RDMESimulation, 63
- buildSphericalEColiCell
 - pySTDLM::StandardCells, 28
- buildVector
 - pyLM::RDME::RDMESimulation, 64
- CME.py, 74
- CellArranger.py, 73
- cellAttributes
 - pySTDLM::CellArranger::CellArranger, 41
- cellFactory
 - pySTDLM::CellArranger::CellArranger, 40
- cellType
 - pySTDLM::CellArranger::CellArranger, 41
- cells
 - pySTDLM::CellArranger::CellArranger, 41
- Cells.py, 74
- closeLMFile
 - pySTDLM::PostProcessing, 21
- cm
 - pyLM::units, 15
- comparator
 - pySTDLM::Fitting::TraceComparator, 72
- computeMetric
 - pySTDLM::Fitting::Comparator, 50
 - pySTDLM::Fitting::MomentComparator, 52
 - pySTDLM::Fitting::TraceComparator, 71
- computeVolume
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 38
 - pySTDLM::Cells::SphereCell, 69
- continuousDimensions
 - pyLM::RDME::RDMESimulation, 67
- counts
 - pySTDLM::Fitting, 19
- createExpressionModel
 - pySTDLM::StandardReactions, 32
- createModelCell
 - pySTDLM::Cells::CellShape, 42
- customAddedParticleList
 - pyLM::RDME::RDMESimulation, 67
- cyttype
 - pySTDLM::Cells::CellShape, 43
- data
 - pySTDLM::Fitting::TraceComparator, 72
- day
 - pyLM::units, 15
- defaultDiffusionRate
 - pyLM::RDME::RDMERegion, 59
- defineSpecies
 - pyLM::CME::CMESimulation, 45
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMESimulation, 64
- depth
 - pySTDLM::Cells::BoxCell, 36
- diffusionRate
 - pyLM::RDME::RDMERegion, 59
- displayRDMESetup
 - pyLM::ipyInterface, 12
- DistributionComparator
 - pySTDLM::Fitting, 19
- Distributions.py, 74
- divisions
 - pySTDLM::Fitting::SweepFitter, 70
- doesIntersect
 - pySTDLM::CellArranger::RandomArranger, 55
- emit
 - pyLM::LMLogger::NullHandler, 53
- experiments
 - pySTDLM::Fitting::Fitter, 51
- filename
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMESimulation, 67
- Fitting.py, 75
- generator
 - pySTDLM::Distributions::ParticleGenerator, 54
- getAvgVarTrace
 - pySTDLM::PostProcessing, 22
- getHistogram
 - pySTDLM::PostProcessing, 22
- getLattice
 - pyLM::RDME::RDMESimulation, 64
- getLatticeSite
 - pyLM::RDME::RDMESimulation, 64
- getOccupancyKymograph
 - pySTDLM::PostProcessing, 22
- getPhaseSpace
 - pySTDLM::PostProcessing, 22
- getRate
 - pySTDLM::SBMLReader, 26
- getReactionCount
 - pyLM::RDME::RDMERegion, 58
- getReactionString
 - pyLM::ipyInterface, 12
- getSpecieTrace
 - pySTDLM::PostProcessing, 23
- getTimeSteps
 - pySTDLM::PostProcessing, 23
- getVolume

- pySTDLM::Cells::CellShape, 42
- hasBeenDiscretized
 - pyLM::RDME::RDMESimulation, 67
- hasBio
 - pySTDLM::StandardReactions, 32
- hasFit
 - pySTDLM::Fitting::Fitter, 51
 - pySTDLM::Fitting::SweepFitter, 70
- hasGexf
 - pySTDLM::NetworkVisualization, 20
- height
 - pySTDLM::Cells::BoxCell, 36
- hr
 - pyLM::units, 16
- importSBMLModel
 - pySTDLM::SBMLReader, 26
- importSBMLModelL3V1
 - pySTDLM::SBMLReader, 26
- initial_counts
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMESimulation, 67
- intPackVolume
 - pySTDLM::CellArranger::LooseGridArranger, 52
 - pySTDLM::CellArranger::RandomArranger, 55
 - pySTDLM::CellArranger::RandomSpherical-
Arranger, 56
 - pySTDLM::CellArranger::TightGridArranger, 71
- internalAddToSim
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 38
 - pySTDLM::Cells::SphereCell, 69
- internalModelCell
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
 - pySTDLM::Cells::SphereCell, 69
- internalTranslateCell
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::SphereCell, 69
- ipyInterface.py, 75
- LMLogger
 - pyLM::LMLogger, 14
- LMLogger.py, 75
- LMformatter
 - pyLM::LMLogger, 14
- lattice
 - pyLM::RDME::RDMESimulation, 67
- latticeSpacing
 - pyLM::RDME::RDMESimulation, 67
- length
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
- lm_builder
 - pyLM::RDME::RDMESimulation, 67
- maxs
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
 - pySTDLM::Cells::CellShape, 43
 - pySTDLM::Cells::SphereCell, 69
- measure
 - pySTDLM::Fitting, 19
- membraneThickness
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
 - pySTDLM::Cells::CellShape, 43
 - pySTDLM::Cells::SphereCell, 69
- mementype
 - pySTDLM::Cells::CellShape, 43
- micron
 - pyLM::units, 16
- microsecond
 - pyLM::units, 16
- mins
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
 - pySTDLM::Cells::CellShape, 43
 - pySTDLM::Cells::SphereCell, 69
- minute
 - pyLM::units, 16
- mm
 - pyLM::units, 16
- modifyRegion
 - pyLM::RDME::RDMESimulation, 64
- moment
 - pySTDLM::Fitting::MomentComparator, 52
- ms
 - pyLM::units, 16
- n
 - peekFile, 11
- name
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMERegion, 59
 - pyLM::RDME::RDMESimulation, 67
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
 - pySTDLM::Cells::CellShape, 43
 - pySTDLM::Cells::SphereCell, 69
- NetworkVisualization.py, 76
- newBBox
 - pySTDLM::CellArranger::RandomArranger, 55
- nm
 - pyLM::units, 16
- ns
 - pyLM::units, 17
- nullHandlerLM

- pyLM::LMLogger, 14
- OnTheFly.py, 76
- openLMFile
 - pySTDLM::PostProcessing, 23
- origin
 - pySTDLM::Cells::SphereCell, 69
- outsideDomain
 - pySTDLM::CellArranger::RandomArranger, 55
- p1
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
- p2
 - pySTDLM::Cells::BoxCell, 36
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
- packFastGrowingEcoli
 - pySTDLM::StandardCells, 28
- packRegion
 - pyLM::RDME::RDMESimulation, 65
- packSlowGrowingEcoli
 - pySTDLM::StandardCells, 30
- packVolume
 - pySTDLM::CellArranger::CellArranger, 40
- packerName
 - pySTDLM::CellArranger::CellArranger, 41
- parameters
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMESimulation, 67
 - pySTDLM::Fitting::Fitter, 51
- parser
 - peekFile, 11
- particleMap
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMESimulation, 67
- particlename
 - pySTDLM::Distributions::ParticleGenerator, 54
- peekFile, 11
 - args, 11
 - n, 11
 - parser, 11
 - t, 11
 - tf, 11
- peekFile.py, 77
- performFit
 - pySTDLM::Fitting::Fitter, 51
 - pySTDLM::Fitting::SimulatedAnnealingFitter, 68
 - pySTDLM::Fitting::SweepFitter, 70
- plotAvgVar
 - pySTDLM::PostProcessing, 23
- plotAvgVarFromFile
 - pySTDLM::PostProcessing, 24
- plotCMEDynamicReactionNetwork
 - pySTDLM::NetworkVisualization, 19
- plotCMEReactionNetwork
 - pySTDLM::NetworkVisualization, 20
- plotOccupancyKymograph
 - pySTDLM::PostProcessing, 24
- plotPhaseSpace
 - pySTDLM::PostProcessing, 24
- plotRDMEReactionNetwork
 - pySTDLM::NetworkVisualization, 20
- plotReactionNetwork
 - pySTDLM::NetworkVisualization, 20
- plotTrace
 - pySTDLM::PostProcessing, 24
- plotTraceFromFile
 - pySTDLM::PostProcessing, 25
- PostProcessing.py, 77
- pyLM, 11
 - __all__, 12
- pyLM.CME, 12
 - pyLM.CME.CMESimulation, 43
- pyLM.ipynbInterface, 12
- pyLM.LMLogger, 13
 - pyLM.LMLogger.NullHandler, 53
- pyLM.RDME, 14
 - pyLM.RDME.CMESimulation, 47
 - pyLM.RDME.RDMERegion, 56
 - pyLM.RDME.RDMESimulation, 59
- pyLM.units, 14
- pyLM::CME
 - tqdm, 12
- pyLM::CME::CMESimulation
 - __init__, 44
 - addConcentration, 44
 - addParticles, 45
 - addReaction, 45
 - buildReactionModel, 45
 - defineSpecies, 45
 - filename, 47
 - initial_counts, 47
 - name, 47
 - parameters, 47
 - particleMap, 47
 - reactions, 47
 - replicates, 47
 - run, 45
 - runMPI, 46
 - runSolver, 46
 - save, 46
 - setRandomSeed, 46
 - setSimulationTime, 46
 - setTimestep, 46
 - setWriteInterval, 46
 - species_id, 47
 - volume, 47
- pyLM::LMLogger
 - LMLogger, 14
 - LMformatter, 14
 - nullHandlerLM, 14
 - setLMLogConsole, 13
 - setLMLogFile, 14

- setLMLoggerLevel, 14
- pyLM::LMLogger::NullHandler
 - emit, 53
- pyLM::RDME
 - tqdm, 14
- pyLM::RDME::CMESimulation
 - __init__, 48
 - addConcentration, 48
 - addParticles, 48
 - addReaction, 48
 - defineSpecies, 49
 - filename, 49
 - initial_counts, 49
 - name, 49
 - parameters, 49
 - particleMap, 49
 - reactions, 49
 - replicates, 49
 - species_id, 49
 - volume, 49
- pyLM::RDME::RDMERegion
 - __init__, 57
 - addReaction, 58
 - defaultDiffusionRate, 59
 - diffusionRate, 59
 - getReactionCount, 58
 - name, 59
 - reactions, 59
 - setDefaultDiffusionRate, 58
 - setDiffusionRate, 58
- pyLM::RDME::RDMESimulation
 - __init__, 61
 - addCuboidRegion, 61
 - addParticleAt, 61
 - addParticleAtIdx, 61
 - addParticles, 62
 - addRegion, 62
 - addShape, 62
 - buildCapsidCell, 62
 - buildDifference, 62
 - buildDiffusionModel, 63
 - buildEllipse, 63
 - buildPoint, 63
 - buildReactionModel, 63
 - buildSpatialModel, 63
 - buildSphere, 63
 - buildSphericalCell, 63
 - buildVector, 64
 - continousDimensions, 67
 - customAddedParticleList, 67
 - defineSpecies, 64
 - filename, 67
 - getLattice, 64
 - getLatticeSite, 64
 - hasBeenDiscretized, 67
 - initial_counts, 67
 - lattice, 67
 - latticeSpacing, 67
 - lm_builder, 67
 - modifyRegion, 64
 - name, 67
 - packRegion, 65
 - parameters, 67
 - particleMap, 67
 - regions, 67
 - replicates, 67
 - run, 65
 - runMPI, 65
 - runSolver, 65
 - save, 65
 - setLatticeSite, 66
 - setLatticeWriteInterval, 66
 - setRandomSeed, 66
 - setSimulationTime, 66
 - setTimestep, 66
 - setTransitionRate, 66
 - setTwoWayTransitionRate, 66
 - setWriteInterval, 66
 - siteTypes, 67
 - siteVolume, 67
 - species_id, 67
 - transitionRates, 67
- pyLM::ipyInterface
 - displayRDMESetup, 12
 - getReactionString, 12
 - writeTable, 13
- pyLM::units
 - angstrom, 15
 - cm, 15
 - day, 15
 - hr, 16
 - micron, 16
 - microsecond, 16
 - minute, 16
 - mm, 16
 - ms, 16
 - nm, 16
 - ns, 17
 - second, 17
- pySTDLM, 17
 - __all__, 18
- pySTDLM.CellArranger, 18
 - pySTDLM.CellArranger.CellArranger, 39
 - pySTDLM.CellArranger.LooseGridArranger, 51
 - pySTDLM.CellArranger.RandomArranger, 54
 - pySTDLM.CellArranger.RandomSphericalArranger, 55
 - pySTDLM.CellArranger.TightGridArranger, 70
- pySTDLM.Cells, 18
 - pySTDLM.Cells.BoxCell, 35
 - pySTDLM.Cells.CapsuleCell, 36
 - pySTDLM.Cells.CapsuleShellCell, 38
 - pySTDLM.Cells.CellShape, 41
 - pySTDLM.Cells.SphereCell, 68
- pySTDLM.Distributions, 18
 - pySTDLM.Distributions.ParticleGenerator, 53
- pySTDLM.Fitting, 18

- pySTDLM.Fitting.Comparator, 49
- pySTDLM.Fitting.Fitter, 50
- pySTDLM.Fitting.MomentComparator, 52
- pySTDLM.Fitting.SimulatedAnnealingFitter, 67
- pySTDLM.Fitting.SweepFitter, 69
- pySTDLM.Fitting.TraceComparator, 71
- pySTDLM.NetworkVisualization, 19
- pySTDLM.OnTheFly, 20
- pySTDLM.PostProcessing, 21
- pySTDLM.SBMLReader, 26
- pySTDLM.StandardCells, 27
- pySTDLM.StandardReactionSystems, 32
- pySTDLM.StandardReactions, 30
- pySTDLM::CellArranger::CellArranger
 - __init__, 40
 - addToSimulation, 40
 - allowedCellTypes, 40
 - allowedForms, 40
 - arrangerAttributes, 41
 - cellAttributes, 41
 - cellFactory, 40
 - cellType, 41
 - cells, 41
 - packVolume, 40
 - packerName, 41
- pySTDLM::CellArranger::LooseGridArranger
 - __init__, 52
 - intPackVolume, 52
- pySTDLM::CellArranger::RandomArranger
 - __init__, 55
 - doesIntersect, 55
 - intPackVolume, 55
 - newBBox, 55
 - outsideDomain, 55
- pySTDLM::CellArranger::RandomSphericalArranger
 - __init__, 56
 - intPackVolume, 56
- pySTDLM::CellArranger::TightGridArranger
 - __init__, 71
 - intPackVolume, 71
- pySTDLM::Cells::BoxCell
 - __init__, 36
 - computeVolume, 36
 - depth, 36
 - height, 36
 - internalAddToSim, 36
 - internalModelCell, 36
 - internalTranslateCell, 36
 - maxs, 36
 - membraneThickness, 36
 - mins, 36
 - name, 36
 - p1, 36
 - p2, 36
 - width, 36
- pySTDLM::Cells::CapsuleCell
 - __init__, 37
 - computeVolume, 37
 - internalAddToSim, 37
 - internalModelCell, 37
 - internalTranslateCell, 37
 - length, 37
 - maxs, 37
 - membraneThickness, 37
 - mins, 37
 - name, 37
 - p1, 37
 - p2, 37
 - radius, 37
- pySTDLM::Cells::CapsuleShellCell
 - __init__, 38
 - computeVolume, 38
 - internalAddToSim, 38
 - internalModelCell, 39
 - length, 39
 - maxs, 39
 - membraneThickness, 39
 - mins, 39
 - name, 39
 - p1, 39
 - p2, 39
 - radius, 39
- pySTDLM::Cells::CellShape
 - __init__, 42
 - addToSimulation, 42
 - boundingBox, 42
 - createModelCell, 42
 - cyttype, 43
 - getVolume, 42
 - maxs, 43
 - membraneThickness, 43
 - memtype, 43
 - mins, 43
 - name, 43
 - setRegions, 42
 - translateCell, 43
 - volume, 43
- pySTDLM::Cells::SphereCell
 - __init__, 69
 - computeVolume, 69
 - internalAddToSim, 69
 - internalModelCell, 69
 - internalTranslateCell, 69
 - maxs, 69
 - membraneThickness, 69
 - mins, 69
 - name, 69
 - origin, 69
 - radius, 69
- pySTDLM::Distributions::ParticleGenerator
 - __init__, 53
 - addParticlesToRegion, 54
 - addedLast, 54
 - generator, 54
 - particlename, 54
 - totalAdded, 54

- pySTDLM::Fitting
 - binwidth, 19
 - counts, 19
 - DistributionComparator, 19
 - measure, 19
- pySTDLM::Fitting::Comparator
 - __init__, 50
 - computeMetric, 50
 - specie, 50
- pySTDLM::Fitting::Fitter
 - __init__, 51
 - addExperiment, 51
 - addParameter, 51
 - experiments, 51
 - hasFit, 51
 - parameters, 51
 - performFit, 51
 - saveFit, 51
- pySTDLM::Fitting::MomentComparator
 - __init__, 52
 - computeMetric, 52
 - moment, 52
 - sem, 52
 - value, 52
- pySTDLM::Fitting::SimulatedAnnealingFitter
 - __init__, 68
 - performFit, 68
 - saveFit, 68
- pySTDLM::Fitting::SweepFitter
 - __init__, 70
 - divisions, 70
 - hasFit, 70
 - performFit, 70
 - saveFit, 70
 - setParameterDivisions, 70
- pySTDLM::Fitting::TraceComparator
 - __init__, 71
 - comparator, 72
 - computeMetric, 71
 - data, 72
 - times, 72
- pySTDLM::NetworkVisualization
 - hasGexf, 20
 - plotCMEDynamicReactionNetwork, 19
 - plotCMEReactionNetwork, 20
 - plotRDMEReactionNetwork, 20
 - plotReactionNetwork, 20
- pySTDLM::PostProcessing
 - closeLMFile, 21
 - getAvgVarTrace, 22
 - getHistogram, 22
 - getOccupancyKymograph, 22
 - getPhaseSpace, 22
 - getSpecieTrace, 23
 - getTimeSteps, 23
 - openLMFile, 23
 - plotAvgVar, 23
 - plotAvgVarFromFile, 24
 - plotOccupancyKymograph, 24
 - plotPhaseSpace, 24
 - plotTrace, 24
 - plotTraceFromFile, 25
 - showAvgVar, 25
 - showAvgVarFromFile, 25
 - showOccupancyKymograph, 25
 - showTrace, 26
 - showTraceFromFile, 26
- pySTDLM::SBMLReader
 - getRate, 26
 - importSBMLModel, 26
 - importSBMLModelL3V1, 26
 - readSBMLtoCME, 27
 - readSBMLtoRDME, 27
 - readSBMLtoRegion, 27
- pySTDLM::StandardCells
 - buildDividingEColiCell, 28
 - buildEColiCell, 28
 - buildFilamentousEColiCell, 28
 - buildSphericalEColiCell, 28
 - packFastGrowingEcoli, 28
 - packSlowGrowingEcoli, 30
- pySTDLM::StandardReactionSystems
 - addLacTwoStateSystem, 33
 - addMinDESsystem, 33
 - addPTSPathway, 33
- pySTDLM::StandardReactions
 - addMembraneTransporter, 30
 - addMichaelisMenten, 31
 - addPassiveTransport, 31
 - addReversibleMichaelisMenten, 31
 - createExpressionModel, 32
 - hasBio, 32
- RDME.py, 78
- radius
 - pySTDLM::Cells::CapsuleCell, 37
 - pySTDLM::Cells::CapsuleShellCell, 39
 - pySTDLM::Cells::SphereCell, 69
- reactions
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMERegion, 59
- readSBMLtoCME
 - pySTDLM::SBMLReader, 27
- readSBMLtoRDME
 - pySTDLM::SBMLReader, 27
- readSBMLtoRegion
 - pySTDLM::SBMLReader, 27
- regions
 - pyLM::RDME::RDMESimulation, 67
- replicates
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMESimulation, 67
- run
 - pyLM::CME::CMESimulation, 45
 - pyLM::RDME::RDMESimulation, 65

- runMPI
 - pyLM::CME::CMESimulation, 46
 - pyLM::RDME::RDMESimulation, 65
- runSolver
 - pyLM::CME::CMESimulation, 46
 - pyLM::RDME::RDMESimulation, 65
- SBMLReader.py, 78
- save
 - pyLM::CME::CMESimulation, 46
 - pyLM::RDME::RDMESimulation, 65
- saveFit
 - pySTDLM::Fitting::Fitter, 51
 - pySTDLM::Fitting::SimulatedAnnealingFitter, 68
 - pySTDLM::Fitting::SweepFitter, 70
- second
 - pyLM::units, 17
- sem
 - pySTDLM::Fitting::MomentComparator, 52
- setDefaultDiffusionRate
 - pyLM::RDME::RDMERegion, 58
- setDiffusionRate
 - pyLM::RDME::RDMERegion, 58
- setLMLogConsole
 - pyLM::LMLogger, 13
- setLMLogFile
 - pyLM::LMLogger, 14
- setLMLoggerLevel
 - pyLM::LMLogger, 14
- setLatticeSite
 - pyLM::RDME::RDMESimulation, 66
- setLatticeWriteInterval
 - pyLM::RDME::RDMESimulation, 66
- setParameterDivisions
 - pySTDLM::Fitting::SweepFitter, 70
- setRandomSeed
 - pyLM::CME::CMESimulation, 46
 - pyLM::RDME::RDMESimulation, 66
- setRegions
 - pySTDLM::Cells::CellShape, 42
- setSimulationTime
 - pyLM::CME::CMESimulation, 46
 - pyLM::RDME::RDMESimulation, 66
- setTimestep
 - pyLM::CME::CMESimulation, 46
 - pyLM::RDME::RDMESimulation, 66
- setTransitionRate
 - pyLM::RDME::RDMESimulation, 66
- setTwoWayTransitionRate
 - pyLM::RDME::RDMESimulation, 66
- setWriteInterval
 - pyLM::CME::CMESimulation, 46
 - pyLM::RDME::RDMESimulation, 66
- showAvgVar
 - pySTDLM::PostProcessing, 25
- showAvgVarFromFile
 - pySTDLM::PostProcessing, 25
- showOccupancyKymograph
 - pySTDLM::PostProcessing, 25
- showTrace
 - pySTDLM::PostProcessing, 26
- showTraceFromFile
 - pySTDLM::PostProcessing, 26
- siteTypes
 - pyLM::RDME::RDMESimulation, 67
- siteVolume
 - pyLM::RDME::RDMESimulation, 67
- specie
 - pySTDLM::Fitting::Comparator, 50
- species_id
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pyLM::RDME::RDMESimulation, 67
- StandardCells.py, 79
- StandardReactionSystems.py, 79
- StandardReactions.py, 79
- t
 - peekFile, 11
- tf
 - peekFile, 11
- times
 - pySTDLM::Fitting::TraceComparator, 72
- totalAdded
 - pySTDLM::Distributions::ParticleGenerator, 54
- tqdm
 - pyLM::CME, 12
 - pyLM::RDME, 14
- transitionRates
 - pyLM::RDME::RDMESimulation, 67
- translateCell
 - pySTDLM::Cells::CellShape, 43
- units.py, 80
- value
 - pySTDLM::Fitting::MomentComparator, 52
- volume
 - pyLM::CME::CMESimulation, 47
 - pyLM::RDME::CMESimulation, 49
 - pySTDLM::Cells::CellShape, 43
- width
 - pySTDLM::Cells::BoxCell, 36
- writeTable
 - pyLM::ipyInterface, 13